Burroughs

B 5281

PROCESSOR

TRAINING MANUAL

PROPERTY OF AND TO BE RETURNED TO

Burroughs

Printed in U.S. America Revised 8-1-66

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

INTRODUCTION

1.1 GENERAL

The purpose of this manual is to provide a detailed description of the internal functioning of the B 5281 Data Processor. It is written with the assumption that a knowledge of the B 5500 System Concept has been attained, prior to reading this manual. Its intention is, therefore, not to duplicate the System Concept material previously covered, but to complement existing material. For these reasons the system operating characteristics relative to the Processor are not described. Instead, the specific operation of the Processor is detailed.

To accomplish its purpose, it is intended that this manual be utilized in conjunction with the B 5500 Processor Flow Charts, which are contained in a separate binder. As each operator, that the Processor is mechanized to execute, is detailed in a logical form on the Flow Charts, a description of the operator flow charts will suffice to explain the operator complement which is available within the Processor. To facilitate the description of the operators, the major logical elements provided as part of the Processor logic are described. These include:

1. The Parallel Adder
2. The Serial Decimal Adder
3. The Input/Output Conversion
4. B Register manipulation
5. The Processor to Memory Timing relationship
6. The Operator Control Level Coding
7. Maintenance Test Logic

To facilitate and complement the use of this manual, the separate binder containing the Operator Flow Charts also contains the Glossary of terms which are utilized, a Cross Index to the D.A. Logical Schematics, and a Cross Index to the Logic Book.

1.2 PHYSICAL ORIENTATION

The physical location of the B 5281 Data Processor within the B 5120 Main Frame is illustrated in Figure 1.2-1. When a single data processor is utilized, its location is at the left end of the main frame and is designated as DP-A. When two data processors are utilized, DP-A is at the left and DP-B is at the right. Also illustrated in Figure 1.2-1 are the Gate (Rack) pivot points. The gate pivot point arrangement facilitates the cabling connections to the other units of the Main Frame.

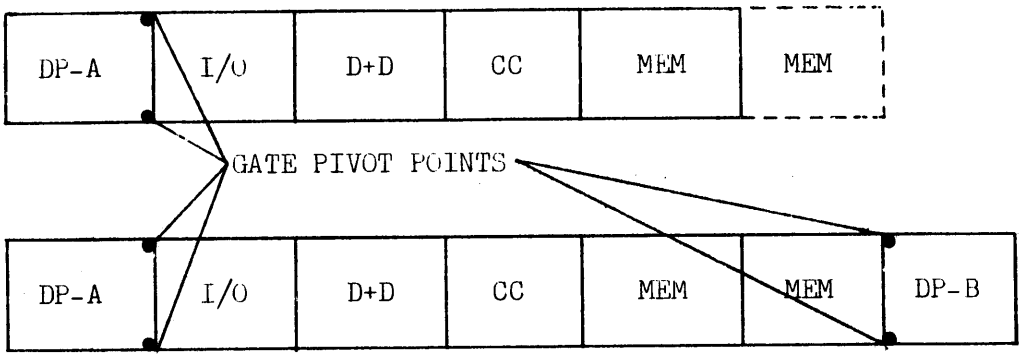


FIGURE 1.2-1. DP-A AND DP-B MAIN FRAME LOCATION

Figure 1.2-2 shows a general view of the Main Frame (front view) with the gates of the Processor opened. The Power Pack and Rack J can be observed within the Processor. The Power Pack is described in the B 5370 Power Supply Manual.

The physical construction of the two Processors (DP-A and DP-B) is the same. The rack layout as illustrated in Figure 1.2-3 is therefore applicable to either Processor The layout of DP-B is rotated relative to DP-A due to the physical location of the two Processors. When a single Processor is utilized, the Rack layout as illustrated in Figure 1.2-3 is valid for DP-A when viewed from the front. It is noted that Rack J is stationary as opposed to Racks A, B, D, and E which are swing-out gates.

Figure 1.2-4 illustrates the panel layout for the various racks within the Processor. The numbering layout (0 through 9) as illustrated implies that the racks are viewed from the wiring side of the rack. The numbering is reversed when viewed from the package insertion side. The panel layout is the same for Racks A, B, D, and E. Rack J is merely a smaller version of the swing out gates. Note that the cabling, to and from the Processor racks, plugs into column 4 of Panels B, D and F.

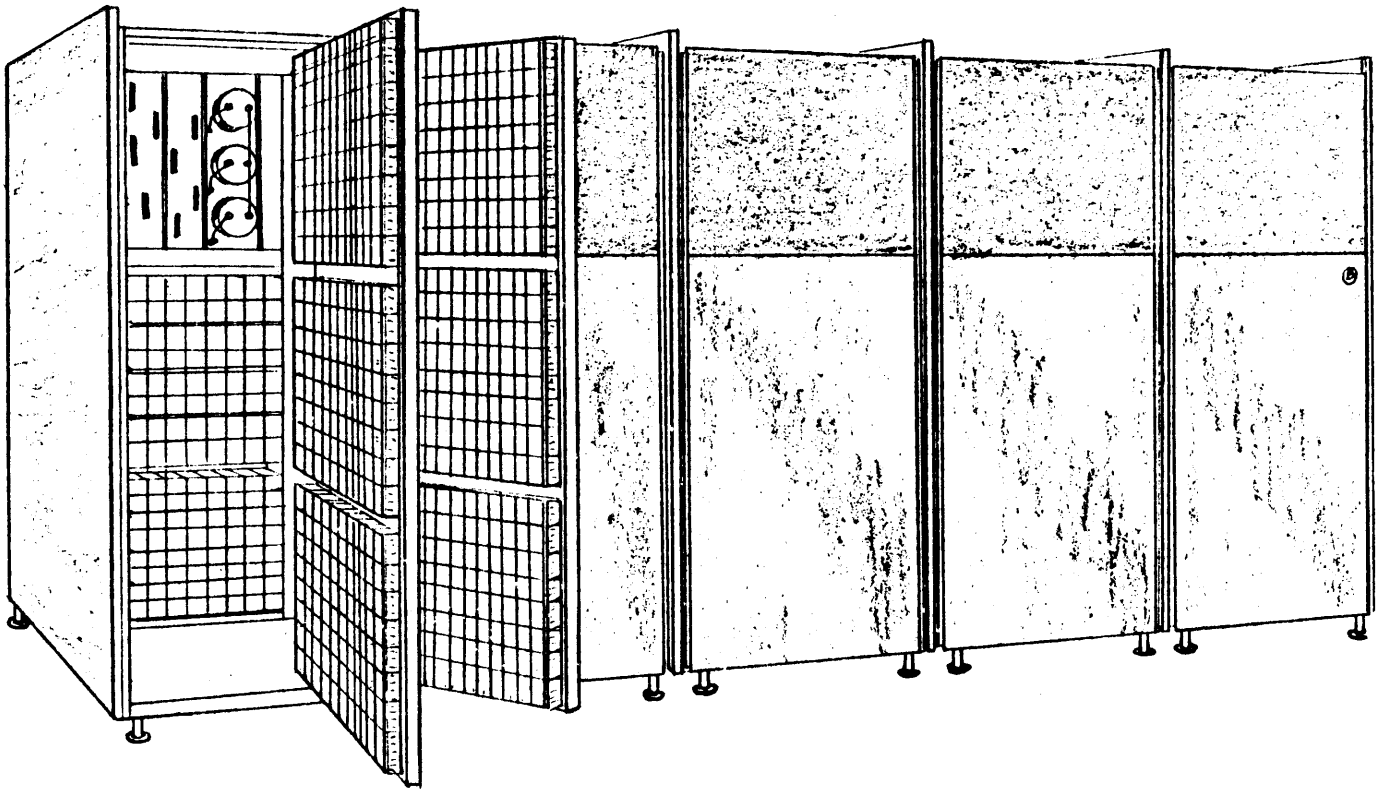


FIGURE 1.2-2. MAIN FRAME WITH PROCESSOR OPENED

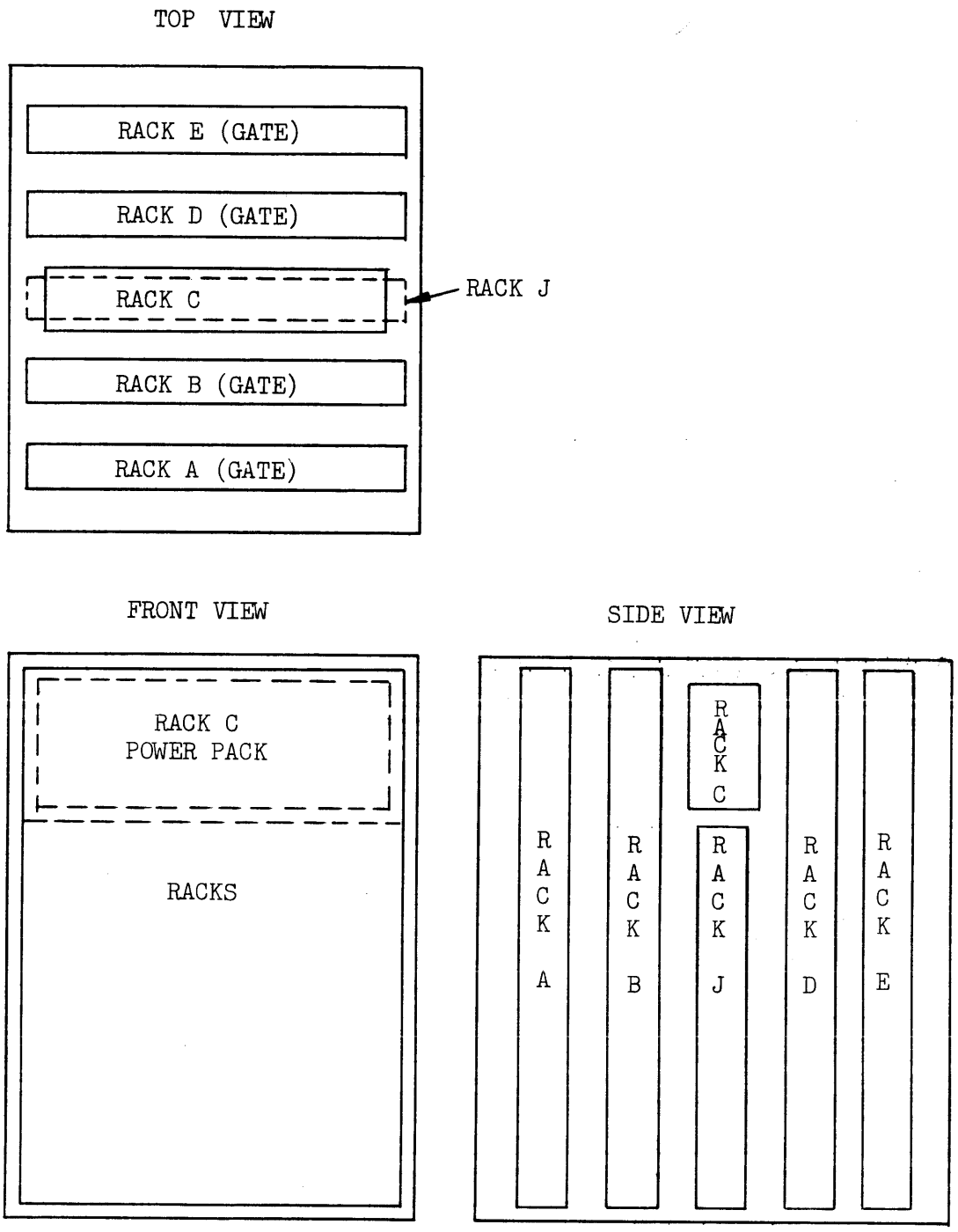
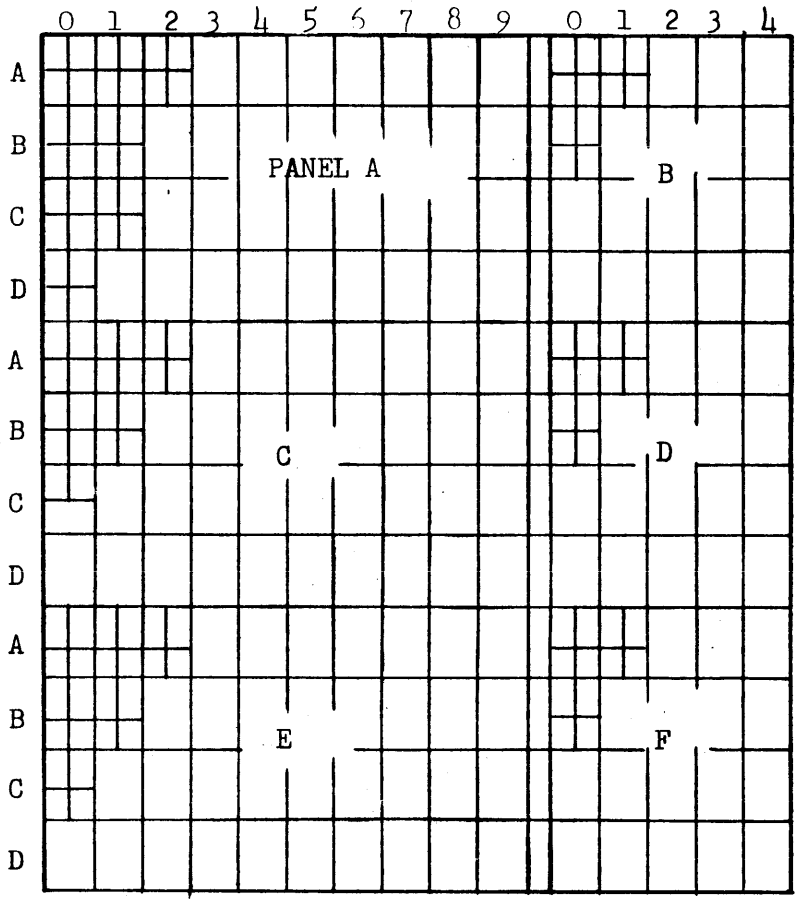


FIGURE 1.2-3. PROCESSOR RACK LAYOUT

RACKS A, B, D & E



RACK J

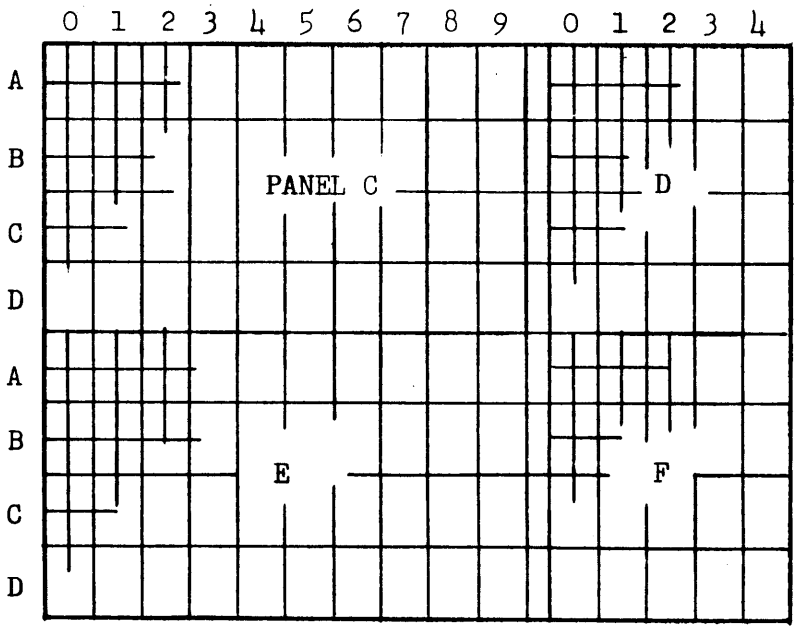


FIGURE 1.2-4. PANEL LAYOUT WIRING SIDE

RACK CONTENT

The rack content in terms of the various registers and logical flip-flops of the Processor are listed below; the Q flip-flops appear on various racks:

A RACK

1. A Register
2. X Register
3. AROF
4. BROF
5. SALF

B RACK

1. B Register
2. "W" Drivers (Adder Logic)

E RACK

1. P Register
2. T Register
3. J Register
4. CWMF
5. PROF
6. TROF
7. MRAF
8. MROF
9. MWOF
10. HLTF
11. TM Register
12. CCCF

D RACK

1. K Register
2. V Register
3. N Register
4. L Register
5. G Register
6. H Register
7. E Register
8. Y Register
9. Z Register
10. I Register
11. NCSF
12. "U" Drivers (output drivers)
13. VARF

J RACK

1. F Register
2. S Register
3. C Register
4. M Register
5. R Register

C RACK

1. Power Pack

SCHEMATIC PAGE NUMBER

The various registers and their associated drivers and switches are contained on the D.A. Logic Schematics in a grouped numbered sequence. The D.A. Logic Schematics are grouped together according to rack content; that is, all logic contained on a specific rack is grouped together to form a set of schematics for that rack. The list which follows indicates the D.A. Logic Schematic numbering groups according to the indicated circuitry. Because the page numbers, as indicated on the following list, refer to common circuitry which may be contained on more than one rack, the same page number group may be found within the schematics for more than one specific rack. The indicated page number groups with an (nn), imply a series of pages relative to that logic.

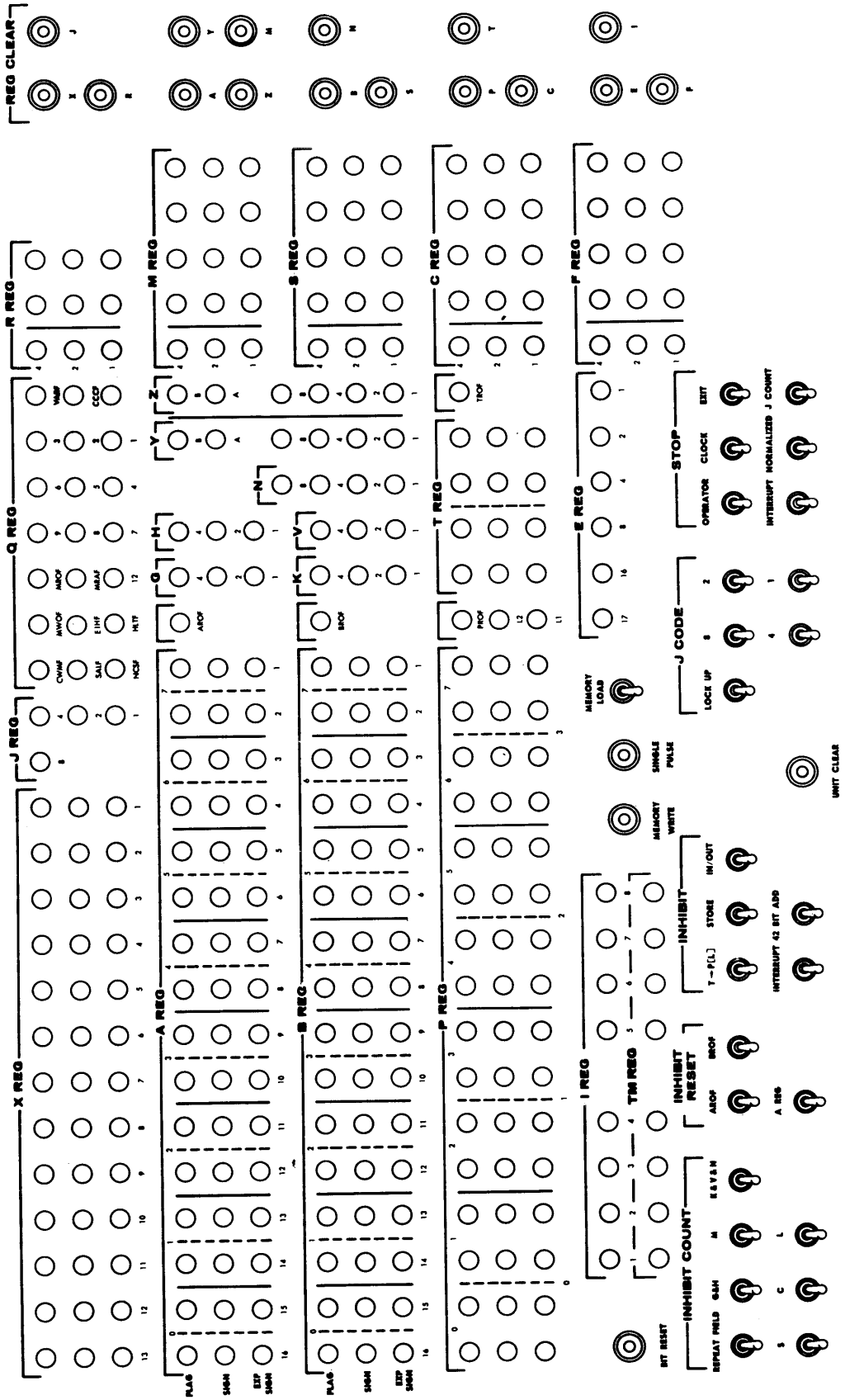


FIGURE 1.2-5 PROCESSOR DISPLAY PANEL

PROCESSOR DISPLAY PANEL

The Processor display panel is located in Display and Distribution as illustrated in Figure 1.2-5. If the system consists of two Processors, then there will be two such display panels. The display panel for Processor A will be located on Gate A (outer gate) of Display and Distribution and display panel for Processor B will be located on Gate B (inner gate).

The display panels for either Processors A or B are identical and are as illustrated in Figure 1.2-5. Mounted on the display panel are the neon indicators for each flip-flop in the Processor. Each of these indicators contains the capability of manually setting or resetting its respective flip-flop. Also contained on the display panel are all of the Processor maintenance switches. The following description will give a basic description of each of the registers (and flip-flops) and maintenance switches on the Processor maintenance panel.

Registers and Flip-Flops

A REGISTER. The A register is a 48 bit register which is used to contain the top word of the stack for word mode or a word of the source string in character mode. The bits are numbered 1 thru 48 (bottom to top, right to left). These 48 bits can be subdivided into 16 octal digits or 8 six bit characters. The octal digits (octades) are numbered right to left as indicated at the bottom of the A register neons with octade 1 consisting of bits 1, 2 and 3 of the A register. The six bit characters are numbered, as indicated above the A register, 0 thru 7 from left to right, with character 0 (zero) containing the six high order bits of the A register (48 ⇒ 43).

AROF. This is the "A" Register Occupied Flip-flop. When it is set it indicates that the contents of the A register is valid; when reset, it indicates that the A register contents are invalid.

B REGISTER. Is of the same construct as the A register. The B register will contain (in word mode) the top word of the stack if the A register is empty (indicated by $\overline{\text{AROF}}$) or the second word of the stack if the A register is occupied (AROF). In Character Mode the B register will contain one word of the destination string.

BROF. This is the B Register Occupied Flip-flop. When it is ON, it indicates that the contents of the B register are valid. When it is OFF, it indicates that the contents of the B register are invalid.

C REG. This register contains 15 neons which indicate the status of the flip-flops in the C register. In either the Word Mode or Character Mode, this register holds the core address of the program word which is in the P register.

CCCF. Logical flip-flop for the maintenance Test Routines. Controls the Processor clock.

E REG. There are six neons in this register which are used to indicate the type of memory access needed by the Processor.

F REG. There are 15 indicators in this register. In Word Mode, this register holds the address of the top Return Control Word or top Mark Stack Control Word in the stack. In Character Mode, this register holds the address of the top Return Control Word in the stack.

G REG. There are three flip-flops in this register. In Word Mode or Character Mode, this register points to a character within the A register.

H REG. There are three flip-flops in this register. In either Word Mode or Character Mode, this register points to the bit of the character in the A register pointed to by the G register.

I REG. There are seven flip-flops in this register. Bits 1, 2 and 3 are for the Non-Syllable Interrupts, bits 5, 6, 7 and 8 are used to develop binary values for syllable interrupts.

J REG. There are four neons in this register. This register contains the primary control setting used during the execution of a syllable.

K REG. There are three neons in this register. In either the Word Mode or Character Mode, this register points to a character within the B register.

L REG. (Located between the P and T registers) - There are two neons in this register. In either the Word Mode or Character Mode, this register points to the next operator (to be executed) within the P register.

M REG. There are 15 neons within this register. In Word Mode, this register holds the memory address of the word placed in the A register other than by stack adjustment. In Character Mode, this register holds the address of the current word in the source string.

N REG. There are four neons in this register. In either the Word Mode or Character Mode this register is used as a special counter. The flow chart should be used to obtain its specific use.

P REG. There are 48 indicators in this register. They are divided into four groups of 12, with each group representing one operator. The operators are selected by the L registers.

PROF. This is the P Register Occupied Flip-flop which indicates whether or not the information in the P register is valid. When True (indicator On) the information in the P register is valid, when False the information in the P register is invalid.

Q REG. There are 18 flip-flops in this register. The Q register flip-flops are distributed throughout the Processor unit and they are used as logical flip-flops as needed by the operators. Bits 1 through 9 are not specified as to particular use. Bits 12 through 18 are used as follows:

Q12F - In Character Mode it is used as the True-False indicator. In Word Mode it is the Mark Stack Flip-flop (MSFF).

MRAF - This flip-flop is on when the memory operation is completed for a memory fetch access.

MROF - This flip-flop is on when the memory operation is completed for a memory read access.

HLTF - This bit is used to provide special stops during maintenance operations.

EIHF - Control flip-flops for a memory read access.

MWOF - This bit is on when the memory operation is completed for a memory write access.

SALF - Control flip-flop used to indicate if a program is in sublevel or program level. When this bit is on it indicates sublevel.

CWMF - Character Word Mode Flip-flop, when on indicates program is in character mode.

R REG. There are nine bits in this register. In word mode, this register contains the base address (three high order digits only) of the Program Reference Table (PRT). In Character Mode the six-bits are used as a tally register.

S REG. There are 15 bits within this register. In Word Mode, this register contains the address of the top word in the stack in core memory. In Character Mode, this register holds the address of the current word in the destination string.

T REG. There are 12 bits in this register. This register contains the syllable currently being executed by the Processor.

TM REG. Eight bit register used for Temporary storage for the Maintenance Test Routines.

V REG. There are three bits within this register. In either the Word Mode or Character Mode, this register points to the bit of the character in the B register pointed to by the K register.

VARF. When set will allow the Processor to access the entire PRT when in sublevel.

X REG. There are 39 bits in this register. In Word Mode, this register is used as an extension of the B register during mathematical computations. In Character Mode, this register is used as storage for Loop Control Words.

Y REG. There are six bits in this register. In either the Word Mode or Character Mode, the register is used for temporary storage of a character from the A register.

Z REG. There are six bits in this register. In either the Word Mode or Character Mode, the register is used for temporary storage of a character from the B register.

PUSH BUTTON SWITCHES

This is a description of the push button switches (non-indicating) located on the Processor Display Panel.

These switches are the push type and maintain contact during the duration that the switch is held in.

BIT RESET. When this switch is held in, it switches the input voltage to the neon switches from +100V to -100V. This causes the neon switch to apply a reset voltage to the flip-flop when the neon switch button is pressed. The BIT RESET switch is common to all of the neon switches on the Processor Panel.

MEMORY WRITE (US25X). To use this switch the MEMORY LOAD switch must be in the TEST position (up) and TROF must be Off.

When depressed, this switch causes the contents of the B register to be written into memory under the control of the S register. The S register is incremented by one.

REG CLEAR. There are 16 buttons in this group. When a button is depressed its associated register is completely cleared (all bits reset).

SINGLE PULSE (US24X). This button is used to generate one clock pulse to the Processor. It is normally used in conjunction with one of the Processor stop switches.

UNIT CLEAR. Pressing this button resets all of the flip-flops in the Processor. This is the general Clear button for the Processor only.

INHIBIT TOGGLE SWITCHES

This is a description of the toggle switches located on the Processor Display Panel. Listed below is a table of all toggle switches on the Processor Display Panel. This table is a listing by switch number sequence. The following write up is in alphabetical sequence and describes the use of the switches.

B 5000 PROCESSOR - MAINTENANCE PANEL SWITCHES

SWITCH NUMBER	ACTION
US01X	INHIBIT COUNT REPEAT FIELD
US02X	INHIBIT COUNT G AND H
US03X	INHIBIT COUNT M
US04X	INHIBIT COUNT K AND V AND N
US05X	INHIBIT COUNT S
US06X	INHIBIT COUNT C
US07X	INHIBIT COUNT L
US08X	INHIBIT RESET AROF
US09X	INHIBIT RESET BROF
US10X	INHIBIT T <= P [L]
US11X	INHIBIT STORE
US12X	INHIBIT IN/OUT
US13X	INHIBIT INTERRUPT
US14X	STOP OPERATOR
US15X	STOP CLOCK
US16X	STOP ON EXIT
US17X	STOP ON INTERRUPT (PROCESSOR TYPE)
US18X	STOP WHEN NORMALIZED
US19X	STOP ON J COUNT
US20X => US23X	J-CODE SWITCHES (1, 2, 4 and 8)
US26X	INHIBIT RESET A REGISTER
US27X	INHIBIT 42 BIT ADD
US28X	LOCK-UP ON J CODE
US29X	MEMORY LOAD

Inhibit 42 Bit Add (US27X). This switch in the TEST position inhibits the level from the outputs of the adder circuit. This prevents the sum of the addition from being fed into the B register.

Inhibit In/Out (US12X). This switch in TEST position forces the Initiate Input/Output operators to become NO OPS, thus eliminating any I/O operations.

Inhibit Interrupt (US13X). In TEST position, this switch inhibits the set of any interrupt in the Processor, by inhibiting the setting of any bits in the I register.

Inhibit Store (US11X). When this switch is in TEST position, any memory write access by the Processor is inhibited.

Inhibit T <= P (L) (US10X). When this switch is in the TEST position it inhibits the transfer of the operators from the P register to the T register. This switch will not inhibit the character transfer from the P register to the T register when called for by the CALL REPEAT FIELD operator.

Inhibit Count C (US06X). When this switch is in the TEST position, it inhibits incrementing the C register. This prevents the fetch of a new program word, allowing the Processor to recycle on the instruction word addressed by the C register.

Inhibit Count G & H (US02X). When this switch is in the TEST position it inhibits the incrementing or decrementing of the G and H registers.

Inhibit Count K, V & N (US04X). When this switch is in the TEST position, it inhibits the incrementing or decrementing of the K, V and N registers. These registers may be either set or reset.

Inhibit Count L (US07X). When this switch is in the TEST position, it inhibits the incrementing or decrementing of the L register. This locks the Processor on a single operator unless an interrupt is encountered.

Inhibit Count M (US03X). When this switch is in the TEST position, it inhibits the incrementing or decrementing of the M register.

Inhibit Count Repeat Field (US01X). When this switch is in the TEST position, it inhibits the decrementing of the REPEAT COUNT FIELD in the T register.

Inhibit Count S (US05X). When this switch is in the TEST position, it inhibits the incrementing or decrementing of the S register.

Inhibit Reset of AROF (US08X). When this switch is in the TEST position, it inhibits the logical resetting of the AROF Flip-flop.

Inhibit Reset of BROF (US09X). When this switch is in the TEST position, it inhibits the logical resetting of the BROF Flip-flop.

Inhibit Reset of A Reg (US26X). When this switch is in the TEST position, it inhibits the logical resetting of the Mantissa in the A register for all operators.

J-Code Switches 1, 2, 4 and 8 (US20X ⇒ US23X). These switches are used in conjunction with the STOP ON COUNT and the J CODE LOCK UP switches. These four switches (1, 2, 4 and 8) are set to indicate the desired J register setting to stop or lock up on.

Lock-Up on J (US28X). This switch is used in conjunction with the J-CODE SWITCHES. When this switch is in TEST position, it inhibits further changes to the J REG once it has attained a setting equal to the value set in the J-CODE SWITCHES.

Stop Clock (US15X). When this switch is turned on, it inhibits the clock pulses to this Processor from Central Control when a memory cycle is not in progress. This allows for single pulsing the Processor while the remainder of the system is running with the clock normal.

Stop Exit (US16X). When this switch is in TEST position, it will inhibit the clock pulses at SECL time, which is at the end of the operator that is presently being executed.

Stop on Interrupts (US17X). When this switch is in TEST position, it will inhibit the clock when any Processor Interrupt appears in the Processor Interrupt Register.

Stop on J Count (US19X). When this switch is in TEST position, it will inhibit the clock pulses to the Processor when the value of the J register equals the value encoded by the J-CODE switches.

Stop Normalized (US18X). When this switch is in the TEST position, it will stop the Processor clock pulses when both operands in the A and B register are aligned (exponents are equal).

Stop Operator (US14X). When this switch is in TEST position, it provides a level which, in conjunction with the Conditional Halt Operator and the J register equal to zero, will stop all Processor action. Processor operation will be resumed when the CLOCK START button is pushed. When this switch is in NORMAL position the Conditional Halt Operator is treated as a No Op.

B 5000 PROCESSOR SCHEMATIC PAGE NUMBERS

65 • 10 • nn • 0	A Register FF's, drivers and switches
65 • 14 • nn • 0	F Register FF's, drivers and switches
65 • 18 • nn • 0	X Register FF's, drivers and switches
65 • 22 • nn • 0	K Register FF's, drivers and switches
65 • 26 • nn • 0	V Register FF's, drivers and switches
65 • 30 • nn • 0	N Register FF's, drivers and switches
65 • 34 • nn • 0	S Register FF's, drivers and switches
65 • 38 • nn • 0	C Register FF's, drivers and switches
65 • 38 • 50	D Drivers
65 • 42 • nn • 0	L Register FF's, drivers and switches
65 • 46 • nn • 0	P Register FF's, drivers and switches
65 • 50 • nn • 0	T Register FF's, drivers and switches
65 • 54 • nn • 0	G Register FF's, drivers and switches
65 • 58 • nn • 0	H Register FF's, drivers and switches
65 • 62 • nn • 0	M Register FF's, drivers and switches
65 • 66 • nn • 0	B Register FF's, drivers and switches
65 • 68 • nn • 0	"W" Drivers and Switches
65 • 70 • nn • 0	E Register FF's, drivers and switches
65 • 72 • nn • 0	"U" Drivers and Switches
65 • 74 • nn • 0	R Register FF's, drivers and switches
65 • 78 • nn • 0	J Register FF's, drivers and switches, TM Register, CCCF
65 • 82 • nn • 0	Y Register FF's, drivers and switches
65 • 86 • nn • 0	Z Register FF's, drivers and switches
65 • 88 • nn • 0	I Register FF's, drivers and switches
65 • 90 • nn • 0	Q Register FF's, drivers and switches, VARF
65 • 92 • nn • 0	Common AND Gates
65 • 94 • nn • 0	Misc. "L" Levels
65 • 95 • nn • 0	Intergate Cables
65 • 96 • nn • 0	Interunit Cables
65 • 97 • nn • 0	Filters (Power Distribution)
65 • 97 • 60 • 0	
65 • 97 • 61 • 0	E Rack Ground
65 • 97 • 62 • 0	
65 • 97 • 62 • 5	Inter-Lace Ground
65 • 97 • 63 • 0	
65 • 97 • 64 • 0	J Rack Ground
65 • 97 • 65 • 0	
65 • 97 • 66 • 0	
65 • 97 • 67 • 0	B Rack Ground
65 • 97 • 68 • 0	
65 • 97 • 68 • 5	Inter-Lace Ground
65 • 97 • 69 • 0	
65 • 97 • 70 • 0	A Rack Ground
65 • 97 • 71 • 0	
65 • 97 • 71 • 5	Inter-Lace Ground
65 • 97 • 72 • 0	
65 • 97 • 73 • 0	D Rack Ground
65 • 97 • 74 • 0	
65 • 97 • 74 • 5	Inter-Lace Ground
65 • 98 • nn • 0	Terminations
65 • 99 • nn • 0	Inter Frame Jumpers

SECTION 2

LOGICAL IMPLEMENTATION

2.1 OPERATOR CONTROL LEVEL CODING

The operator code for a specific operator, when in the T register, is decoded to provide a level or levels to enable the logic relative to that specific operator. In some cases the level developed has the same mnemonic designation as the operator itself. In other cases, the level developed is not readily identifiable to the specific operator. In addition, certain operators are decoded such that more than one level is developed; in this case, the levels developed are common to more than one specific operator.

The information which follows indicates the mnemonic name of the operator and the associated bit configuration decoded in the T register. Not all bit positions of the T register are decoded in order to develop the various operator control levels. Only those bits necessary to accomplish the required decoding are actually decoded. For those bit combinations which do not decode to a valid operator, the bit configuration is decoded as Syllable Execute Complete Level and, therefore, act as a NO-OP if placed in the T register.

WORD MODE OPERATORS

Refer to the B 5500 Handbook, Section 2, for a listing of the word mode operators with their associated mnemonic code and octal code. Section 9 of the handbook contains a listing of the T register setting assigned to each word mode operator, including Operand Call, Descriptor Call, Literal Syllables and Control operators.

The common levels developed from these operators are indicated in the margins and footnotes. For those states which are assigned, all blanks are zero, although only those flip-flops whose state is designated are tested by the decoding gates. Where a flip-flop is indicated as "X", a "don't care" condition exists.

CHARACTER MODE OPERATORS

Refer to the B 5500 Handbook, Section 2, for a listing of the Character Mode Operators with their associated mnemonic code and octal code. Section 9 of the handbook contains the Character Mode listing the T register states assigned to each character mode operator, including reserved and "not assigned" states.

In all assigned cases, the literal portion of the syllable is represented as a "don't care" condition. The common levels developed from these operators are indicated in the margins.

2.2 MEMORY ACCESS

Any time the Processor requires a memory access it must provide Central Control with the necessary data to allow the memory exchange portion of Central Control to select the correct Memory Module (0 thru 7) and to subsequently enable the address, control, and information lines between the Processor and the Memory Module. The addressing data provided by the Processor consists of 15 bits of addressing data to Central Control. The information lines between the Processor and Core Memory consist of 48 write information lines and 48 read information lines. Control data to Central Control is a memory write level to indicate a memory write operation and an inhibit level to prevent crosspoint selection when necessary.

The following is a glossary of the levels between Processor and Central Control that are pertinent to a memory access.

PROCESSOR TO CENTRAL CONTROL

UMAIS/. Memory Request Not, when false inhibits the setting of the crosspoint flip-flop in Central Control. This level is primarily used in case of an address error, an address less than 01000 when in Normal State.

UMWRS. Memory Write Level, when true indicates that the subsequent memory access is a memory write access.

UWnnS. 48 write information lines from the Processor.

UM01S/ thru UM12S/. Address lines from the 12 low order bits of the addressing register. These lines are routed thru memory exchange to the Memory Module.

UM13D thru UM15D and UM13D/ thru UM15D/. Six lines indicating the status of the 3 high order bits of the addressing register. These bits are used in Central Control to select a crosspoint flip-flop.

CENTRAL CONTROL TO PROCESSOR

DnnS. 48 read information lines from Core Memory to Processor via memory exchange.

MAED. Memory Address Error indicates an attempt to access either a non-existant Memory Module or a Memory Module that is in local.

MPEX. Memory Parity Error, used during a memory read operation to indicate that a memory parity exists. This level is used to set the Memory Parity Error Interrupt in the Processor.

MTOD. Memory Time Zero Driver, originates in Central Control, indicates the memory cycle is at memory time zero. This level is used to set MWOF (Memory Write Obtained Flip-flop) in the Processor.

MT2S. Memory Time Two, originates in the Memory Module, indicates when the memory cycle is at memory time two. This level is only used during a read access to signify when the Processor can sample the read information lines.

No detailed explanation will be given in this manual of the operation of memory exchange. For this description refer to the Central Control Technical Manual, Subject 7.6.

ADDRESS REGISTERS

With each memory access the Processor uses one of three 15 bit registers for addressing memory; the S register, M register, or C register. Of the 15 bits, the three high order bits are decoded in Central Control and used to select the correct Memory Module. The 12 low order bits are routed thru memory exchange to the Memory Address Register in the Memory Module. For the usage of the three addressing registers refer to the Processor Flow Charts, page 4.02.0.

INFORMATION REGISTERS

For each memory access the Processor uses one of four registers to contain the information to be written into or read from memory. These registers are the A register, B register, M register, and the P register. The A and B registers are 48 bit registers used to contain the information being acted upon by the Processor. Any information to be written into memory must be in one of these two registers.

The P register is a 48 bit register used to contain the program word presently being executed. Use of the P register as an information register is only during a fetch access, the addressing register will be the C register.

The M register is normally considered to be an addressing register but can be used as an information register. This register, being only 15 bits long, will only receive bits 16 thru 30 of the word read from core memory.

CONTROL FLIP-FLOPS

The Processor uses several flip-flops to control its memory access. A description of these flip-flops and their functions follows.

MWOF. Memory Write Obtained Flip-flop, is set by MTOD and reset by the next clock pulse. Its function during a write access is to indicate the termination of the write access, during a read access it serves to clear the information register in the Processor.

MROF. Memory Read Obtained Flip-flop, set by MT2S, it is on for 1 μs. This flip-flop is not used during a write access. During a read access it indicates the termination of the read access. MROF serves also to enable the read information lines into the appropriate information register.

MRAF. Memory Read Access Flip-flop, its use is the same as MROF except that it is only used during a fetch access. It is used to gate the information into the P register and to indicate the termination of a fetch access.

EIHF. Inhibit Address Flip-flop is set during a memory read, other than a fetch access, to prevent erroneously selecting more than one Memory Module during one read access. Its use will take affect in such a case as the addressing register being counted up or down to address the next Memory Module in sequence prior to completion of a memory cycle.

E REGISTER

The E register is set by the Processor when initiating a memory access. This register is a six bit register that is used to select the desired address register, information register, and to indicate a read or write access. A description of the E register bit usage follows.

E01F. When ON indicates that the B register is the information register, when OFF, the A register is the information register.

E02F. When ON indicates that the S register will be the addressing register.

E04F. When ON indicates that the M register will be the addressing register.

E08F. When ON indicates a memory write access, when OFF, a memory read access.

E02F • E04F. When the E register is equal to six, the access will be a read, the M register is the addressing register and also the information register.

E16F • E17F. Used to initiate a fetch access in the Processor. E16F is set by the Processor to request a fetch access, E17F is then set to execute the fetch access.

Figure 2.2-1 is a simplified diagram of the E register gating for selection of the various address and information registers. The selection of any of these registers is possible with any value in the E register. For example, if the E register was equal to 3, then the S register will be the addressing register, selected via gates 2 and 12. The B register will be selected as the information register via gate 6 when MROF is true.

MROF is set, via gate 10, at memory time two (MT2S) with EIHF. EIHF will be set if the memory access is a read access, but not a fetch access.

If the E register is equal to eleven (E01F • E02F • E08F) then the memory access is a write. The S register is still the addressing register, via gates 2 and 12, and the B register is still the information register, now selected by gate 15, to the write information lines (UWnnS).

Gate 15 will be active during both a read and write access by the Processor, but only during a write access will the Memory Module sample the status of the write lines. The write level from the Processor, UMWRS, will be true with E08F • E17F/, enabling the Memory Write Level (WOOD) to the memory module.

Coming into the Processor during a memory write access is the timing level MTOD. With E17F/, indicating that this is not a fetch access, the output of gate 8 will set MWOF. MWOF in turn indicates to the Processor the termination of the write access in the Processor.

Two levels are produced as a result of a memory access by the Processor. These two levels, E Register Equal Zero (EEZL) and E Register Equal Zero and MWOF/ (EWZL), are for use only within the Processor to indicate when a Processor memory access is in progress, other than a fetch access. EEZL is true when the E register equals zero; any access will set the E register bits 1, 2, or 4 making EEZL False. EWZL, also normally True, will go False if EEZL is False or MWOF (Memory Write Obtained Flip-flop) is set. These two levels are primarily used to gate logical functions within the Processor.

A fetch access, using E16F and E17F, can be attempted simultaneous with a read or write access. Normally the fetch access will wait for the data access to take place. For a description of the type of access and priority of accesses, refer to the Processor Flow Charts, 4.01.0.

PROCESSOR ACCESS TIMING

The timing diagram in Figure 2.2-2 is for the Processor accessing Core Memory, write and read. The timing diagram will accommodate both the B 460 (6 μs memory) and the B 461 (4 μs memory) timing. The differences in timing which are pertinent to the B 461 Memory Module are indicated by the dashed lines.

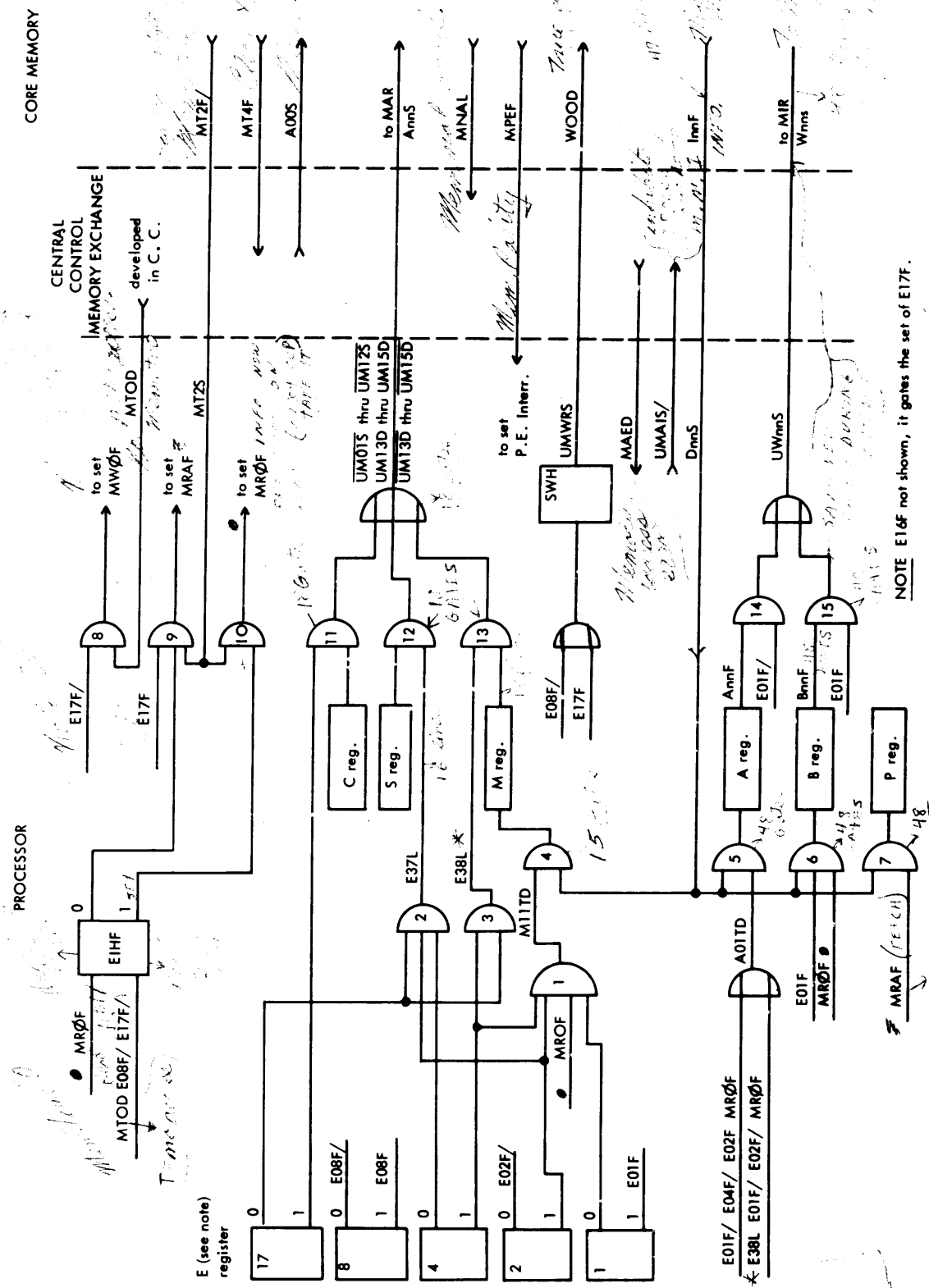
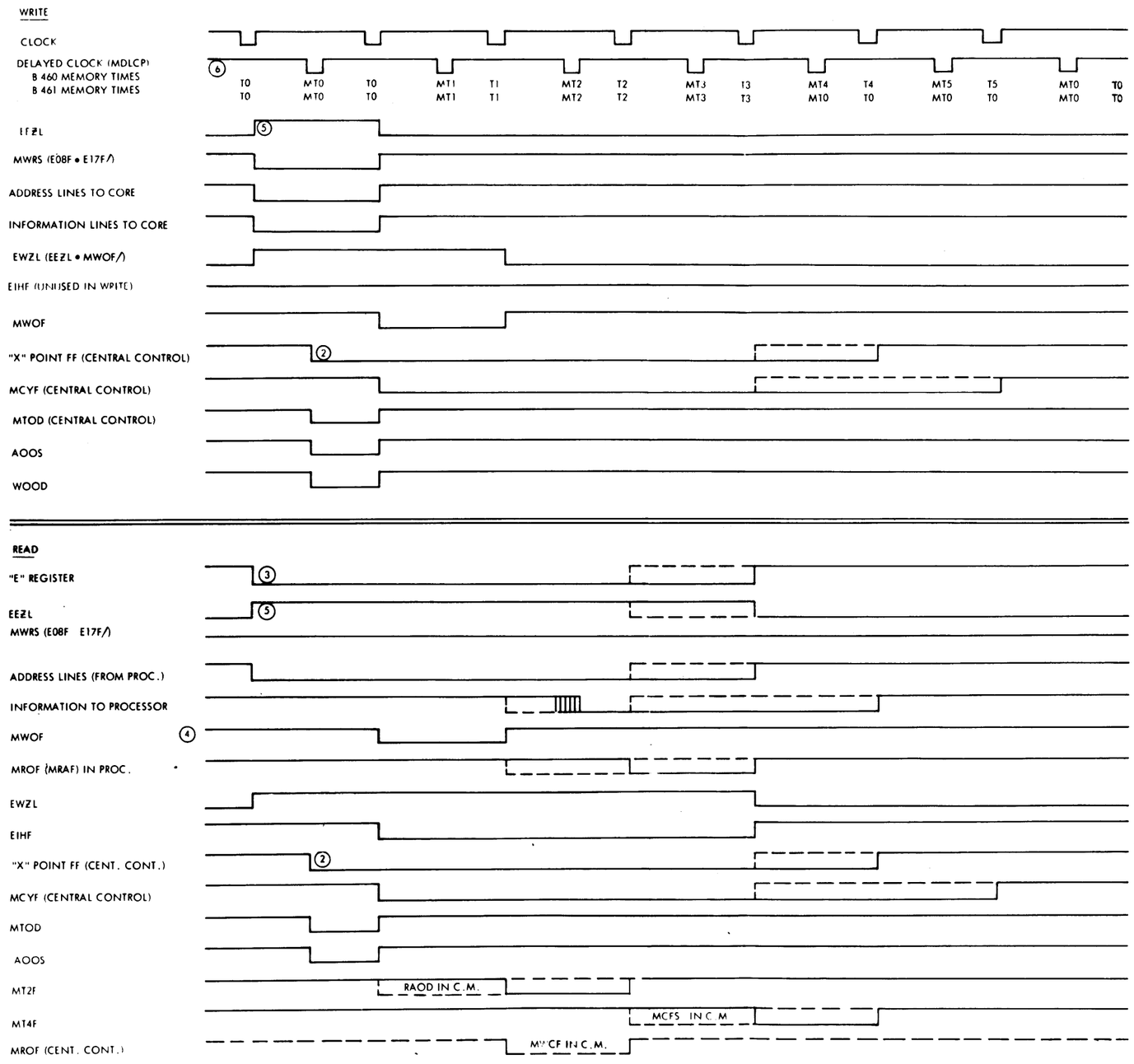


FIGURE 2.2-1 "E" REGISTER GATING



NOTES:
1. DASHED LINES INDICATE TIMINGS UNIQUE TO B 461 (4 μs CORE).
2. "X" POINT SET IS UNCLOCKED SET, GATED BY MDCLP. "X" POINT FF ARE FFN7 TYPES, NO DELAY.
3. COULD BE E17F IF PROCESSOR DOING A FETCH ACCESS.
4. MRAF INSTEAD OF MROF FOR FETCH ACCESS.
5. EEZL = E04F/ • E02F/ • E01F/
6. DELAYED CLOCK IN CENTRAL CONTROL ONLY.

2.3 B REGISTER ALIGNMENT

The B register, when handling character type information, can only transfer characters to or from the B register via specified register locations. In this respect, the B register differs from the A register in that any specified character position of the register cannot be directly accessed. In the B register any specified character that must be accessed requires that the specified character position be shifted to the proper register location. When a specified character position is in the proper register location to be accessed, the character position is said to be "aligned" in the alignment station. Figure 2.3-1 illustrates the B register with the alignment stations indicated. If it is desired that a character be transferred from the B register, the Normal output register location is the 16th and 15th octade positions of the B register; that is, the most significant character position (0) of the B register. If it is desired that a character be transferred to the B register, the Normal input register location is the 1st and 2nd octade positions of the B register; that is, the least significant character position (7) of the B register. These two alignment stations are utilized whenever characters are to be transferred or compared. In addition to the normal input and output alignment stations, the Field Add/Subtract operator utilizes two additional alignment stations. During Field Add/Subtract, the output alignment station is the 2nd and 3rd octade positions of the B register; the input alignment station is the 16th and 15th octade positions of the B register. In this case, the input alignment station during Field Add/Subtract occupies the same register location as the normal output alignment station.

When character type information is accessed from core memory and placed in the B register, the characters of the word just accessed may or not be in their normal configuration. In this sense, normal configuration implies that the most significant character position of the word occupies the most significant character position of the B register and the least significant character position of the word occupied the least significant character position of the B register. The N register is utilized to determine whether the characters of the word currently occupy their normal character positions within the word. If the N register equals zero, the word currently occupying the B register is said to be "restored" to its normal configuration. Thus, if the N register is not equal to zero, the B register contents are not in their normal configuration; that is, the characters of the word in the B register have been shifted.

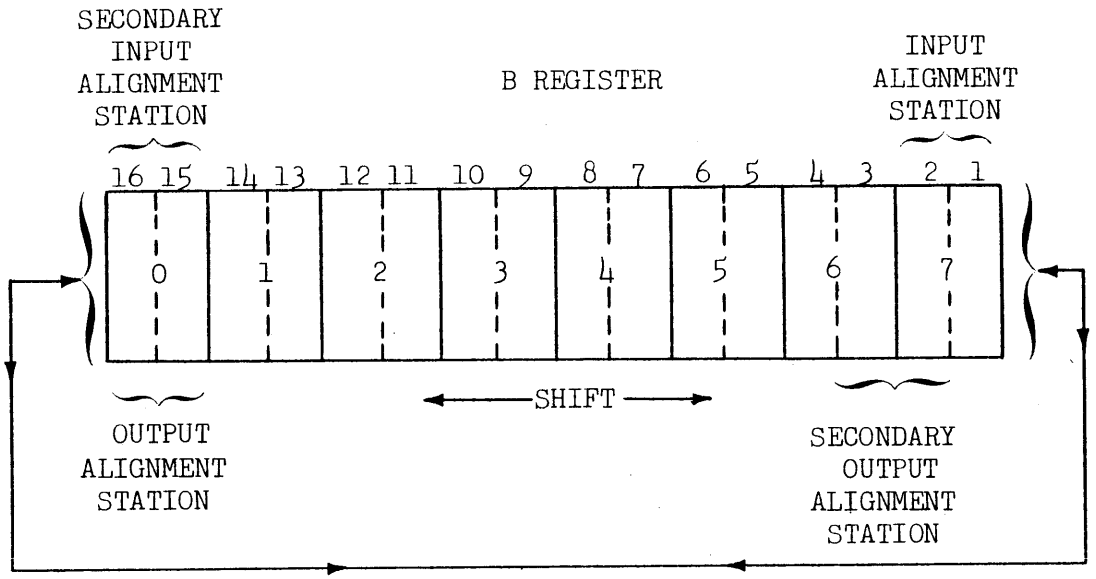
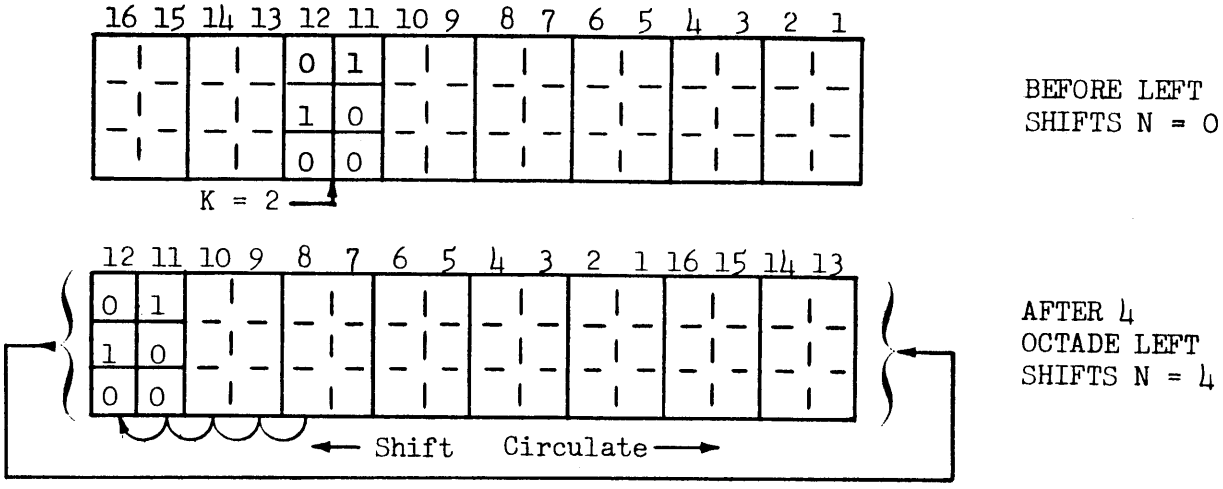


FIGURE 2.3-1. B REGISTER ALIGNMENT STATIONS

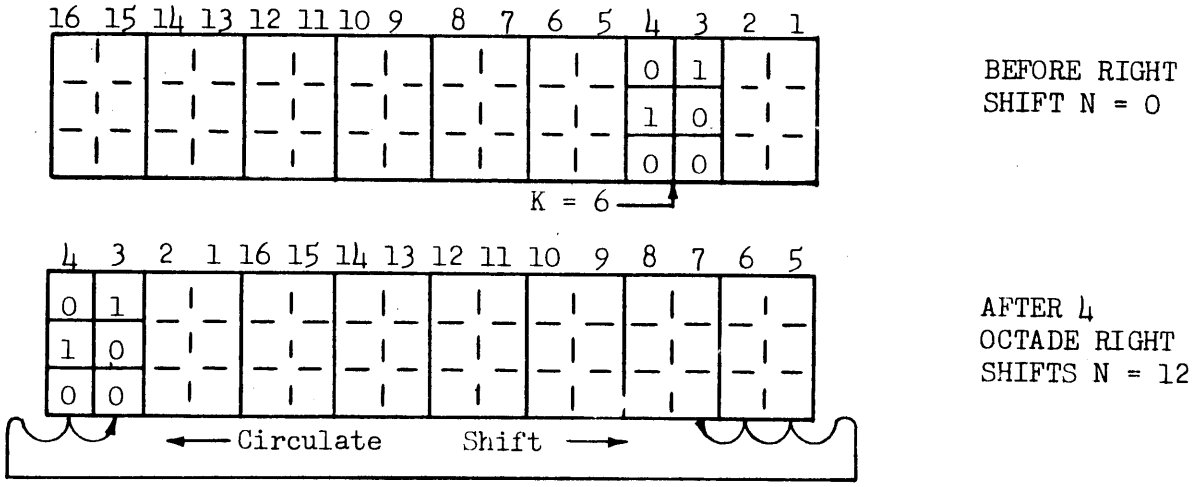
CHARACTER ALIGNMENT

When it is desired to either transfer a character to or from the B register, the specified character position must first be placed in the proper alignment station. The K register performs the function of specifying the specific character position to be placed in the proper alignment station. Normally the K register points to the character position that is to be aligned to the output alignment station. In the process of aligning a particular character into the output alignment station, the B register contents are either shifted right or left by octades (one-half character shifts) until the specified character position is in the output alignment station. During the shifting of the B register contents, the octade that is shifted out of either end of the register is circulated to the other end of the B register. In conjunction with the octade shifts of the B register, the N register tallies the octade shifts; minus one for each right shift and plus one for each left shift.

Assume that it is desired to transfer a character out of the B register. The desired character must be shifted to the output alignment station if not currently there. Output alignment stations, as referred to here, imply the normal alignment station (16th and 15th octade positions). The output alignment station utilized during Field Add/Subtract is referred to as the secondary output alignment station. Figure 2.3-2 illustrates two cases, where initially the B register contents are in their normal configuration (N = 0) and thereafter the specified character is shifted to the output alignment station. In case 1, the desired character is the 2nd character position of the B register and is subsequently shifted left by octades until it occupies the 16th and 15th octade positions of the B register; note that the N register is incremented to a value of four. In case 2, the desired character is in the 6th character position of the B register, and is subsequently shifted right by octades until it occupies the 16th and 15th octade positions of the B register; note that the N register is decremented to a value of 12.



CASE 1. DESIRED CHARACTER IN 12 AND 11 OCTADE POSITIONS



CASE 2. DESIRED CHARACTER IN 4 AND 3 OCTADE POSITIONS

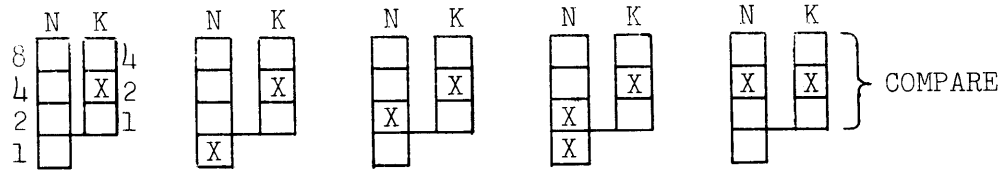
FIGURE 2.3-2. CHARACTER ALIGNMENT

ALIGNMENT SHIFT CONTROL

In order to determine the proper direction for shifting the B register contents to align the specified character in the output alignment station, the 4's bit of the K register is interrogated. If the K register equals four or more (4's bit on), the B register is shifted to the right until alignment is attained. If the K register equals less than four (4's bit is off), the B register is shifted left until alignment is attained.

To recognize that a specified character is in the output alignment station, the logical condition (K = N) is interrogated. In case 1 of Figure 2.3-2, the character pointer K is specifying the 2nd character position of the B register. Also, it is illustrated that the word in the B register is initially in its original configuration as indicated by the N register being equal to zero. Examining the K and N registers for equality, and thus alignment, is accomplished through the comparison of the 3 most significant bits (2, 4 and 8) of the N register with the 3 bits of the K register (1, 2 and 4). Effectively, the comparison is recognizing double counts of the N register and thereby recognizing whole character shifts of the B register.

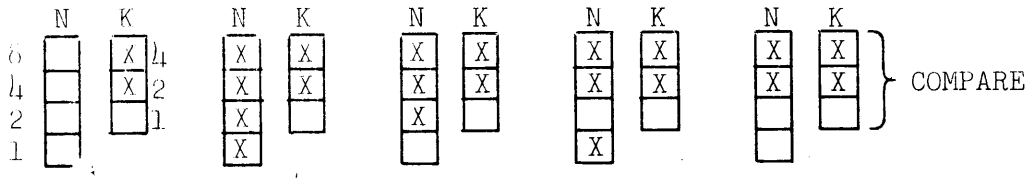
For example: If initially K = 2 and N = 0 (Case 1), the illustration of Figure 2.3-3 shows the conditions existing during the shifting of the specified character into the output alignment station. When the condition K = N is recognized (the logical level KENL goes true), as indicated in step 5, the following clock pulse transfers the specified character out of the output alignment station. Simultaneously, if additional characters require transfer, the K register is incremented, the B register is shifted another octade position and the N register is incremented to tally the shift. Thereafter, with an additional shift of the B register, K will again equal N and the process is repeated until all required characters are transferred.



1. INITIAL CONDITION K = 2 N = 0	2. AFTER FIRST LEFT SHIFT N = 1	3. AFTER SECOND LEFT SHIFT N = 2	4. AFTER THIRD LEFT SHIFT N = 3	5. AFTER FOURTH LEFT SHIFT N = 4 K = N (KENL)	6. TRANSFER CHARACTER

FIGURE 2.3-3. LEFT SHIFT ALIGNMENT

Similarly, if a right shift is indicated, as in Case 2 of Figure 2.3-2, the N register will be decremented during the alignment procedure as shown in Figure 2.3-4. Note that in this case, when the logical condition of K = N is first recognized (KENL is true), only one-half of the desired character is in the output alignment station. This condition is recognized by the 1's bit of the N register being in the set state. Therefore, even though the logical condition of KENL is true, one additional shift of the B register is required to place the complete character in the output alignment station. When shifting to the right, the logical condition of (KENL • NO1F') indicates the alignment of the specified character is completed.



1. INITIAL CONDITION K = 6 N = 0
2. AFTER FIRST RIGHT SHIFT N = 15
3. AFTER SECOND RIGHT SHIFT N = 14
4. AFTER THIRD RIGHT SHIFT N = 13 K = N (KENL)
5. AFTER FOURTH RIGHT SHIFT N = 12 K = N (KENL)
6. TRANSFER CHARACTER

FIGURE 2.3-4. RIGHT SHIFT ALIGNMENT

INPUT ALIGNMENT STATION

From the preceding discussion, it is observed that the logical condition of KENL indicates the specified character is in the output alignment station. Therefore, in order to place the specified character position into the input alignment station, two left shifts and a corresponding circulate of the B register contents will place the specified character in the input alignment station. This is the normal procedure utilized.

RESTORATION OF THE B REGISTER

During the execution of certain operators the contents of the B register must be restored to their original configuration. In this case, the status of the 8's bit of the N register is interrogated to determine the most expedient method to restore the word. If the 8's bit is set (NO8F), the B register contents are shifted left by octades, with a corresponding increment of the N register, until the N register equals zero. If the 8's bit is reset (NO8F'), the B register contents are shifted to the right, with a corresponding decrement of the N register, until the N register equals zero. When NEZL is true (N equals zero), the contents of the B register are in their original configuration.

ALIGNMENT DURING FIELD ADD/SUBTRACT

During the field Add/Subtract Operation, the same logic is utilized to place the specified character in the secondary alignment station as is utilized to align to the normal alignment station. If, when the B register contents are in their normal configuration (N equals zero), the N register is preset to the value of 13, the alignment procedure, when executed, will place the desired character in the secondary output alignment station. During the restoration process in this case, the B register is shifted and the N register counted until the N register equals 13.

2.4 PARALLEL ADDER

The parallel adder is a straight binary D.C. level adder. The parallel adder is mechanized in two configurations:

1. The 42 Bit Mantissa Adder
2. The 10 Bit Address Adder

The 42 bit mantissa adder may also function as a 39 bit adder. The basic add logic of both the mantissa and address adder is the same, with the primary difference in the number of bits which are added and the specific registers which are utilized.

The parallel adder is capable of addition in only one clock pulse time. To facilitate the single pulse addition, logic is provided to determine the presence of a carry prior to the actual addition. Because the parallel adder is only capable of addition, subtraction is a function of complement addition.

Figure 2.4-1 shows the 42 Bit Mantissa Adder and the 10 Bit Address Adder in block diagram form. The 42 bits of the mantissa adder are made up from the 39 bits of the A and B registers, respectively, plus their respective 3 bit extension which is contained within the M register. The following two equations state the action performed by the parallel mantissa and address adder, respectively.

Logic for the Mantissa Adder

$$B \leftarrow B+A \equiv M[10 \Rightarrow 8]\, B[39 \Rightarrow 1] \leftarrow M[10 \Rightarrow 8]\, B[39 \Rightarrow 1] + M[6 \Rightarrow 4]\, A[39 \Rightarrow 1] + Q01F$$

Logic for the Address Adder

$$M \leftarrow M+A \equiv M[10 \Rightarrow 1] \leftarrow M[10 \Rightarrow 1] + A[10 \Rightarrow 1] + Q01F$$

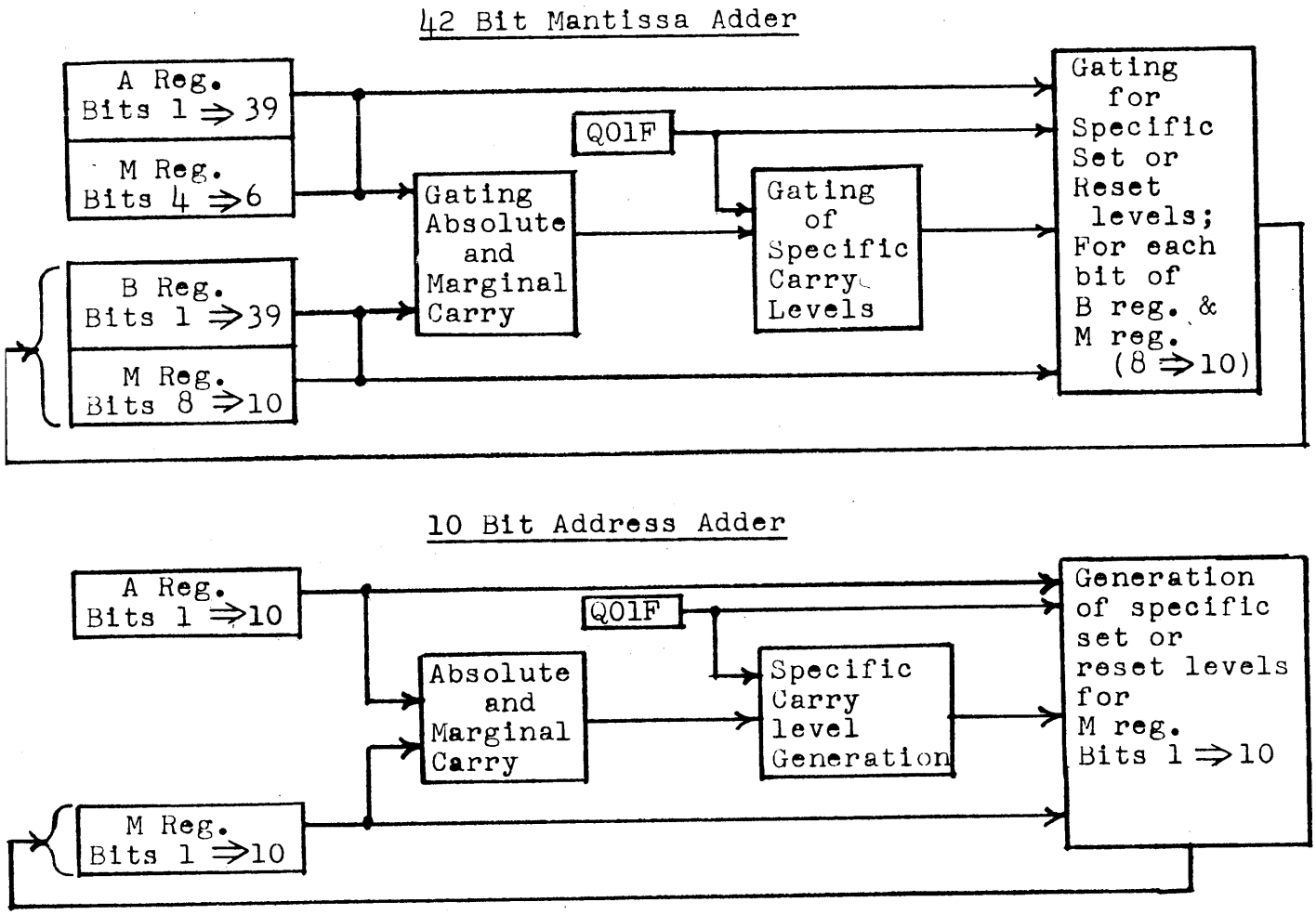


FIGURE 2.4-1. 42 BIT MANTISSA/10 BIT ADDRESS ADDER

BASIC ADDER LOGIC

Due to the binary nature of the parallel adder, it is sufficient to explain the addition of only two binary bits. Each succeeding binary bit position has a similar bit configuration.

Consider the addition of two binary bits, which we will call A and B, plus a possible carry input. The only possible bit configurations that can occur are:

1. A and B zero, with or without a carry
2. A and B one, with or without a carry
3. A zero and B one, with or without a carry
4. A one and B zero, with or without a carry

With these the only possible bit configurations, the logic necessary to sum the A and B bits (with or without a carry) and then place the sum in the B bit position, can be stated simply as:

$$\Delta B = (A \bullet CARRY') + (A' \bullet CARRY)$$

Note that the logic is only written in the form to complement the B bit if required. If the B bit is already in the correct state, no further action is necessary. Figure 2.4-2 shows the possible bit configurations previously stated. Note that the equation for ΔB is valid.

Common Bit Positions Indicated								
C	1	1	0	0	0	0	1	1
A	0	0	1	1	0	0	1	1
B	0	1	0	1	0	1	0	1
	ΔB				Do Not ΔB			

FIGURE 2.4-2. B BIT COMPLEMENT

To mechanize an adder to sum a series of bit positions in parallel requires a knowledge of a carry from the previous bit position. An equation for the add logic of the second bit of a two bit configuration follows. Note that this equation is similar to the equation for ΔB with the carry portion of the equation expanded to include the specific terms of the preceeding bit position.

$$\begin{aligned}\Delta BmnF = AmnF \bullet &(Amn\text{-}1F' \bullet Bmn\text{-}1F' \\ &+ Amn\text{-}1F' \bullet Wmn\text{-}2C' \\ &+ Bmn\text{-}1F' \bullet Wmn\text{-}2C') \\ + AmnF' \bullet &(Amn\text{-}1F \bullet Bmn\text{-}1F \\ &+ Amn\text{-}1F \bullet Wmn\text{-}2C \\ &+ Bmn\text{-}1F \bullet Wmn\text{-}2C)\end{aligned}$$

NOTE

In order to facilitate writing the equation for a number of bit positions, the notation of mn is utilized:
mn = EVEN bit position, ex. 02, 04, 06, etc.
mn-1 = ODD bit position, ex. 01, 03, 05, etc.
W = Adder Logic

In Figure 2.4-3, an example of two bit positions of the A and B register illustrate the usage of the equation for ΔBmnF. Note the equation is valid for all possible bit combinations that will complement the B bit.

Bit	mn	mn-1	mn	mn-1	mn	mn-1	mn	mn-1	mn	mn-1	mn	mn-1
C				0		0				1		1
A	1	0	1	0	1		0	1	0	1	0	
B	Δ	0	Δ		Δ	0	Δ	1	Δ		Δ	1

FIGURE 2.4-3. COMPLEMENT EVEN BIT POSITION

· CARRY LOGIC

From the two previously stated equations to set the sum in the B register (ΔBmn-1F and ΔBmnF), knowledge of a carry from a previous bit position is required. In order to add all bit positions at one clock pulse time, knowledge of the existance of a carry into ALL bit positions is required prior to the actual summing process.

Absolute Carry

It is not always necessary to know if a carry exists from a previous bit position in order to determine if a subsequent carry will result from a given bit position. For those carries which do not require information relative to a prior carry, the carry logic can be written directly. All that is necessary is to know the status of the preceeding bit position. The following equation for WmnR states the logic for an Absolute Carry. The letter W indicates adder logics while the letter R indicates absolute carry.

$$\begin{aligned} WmnR = {} & AmnF \cdot BmnF \\ & + AmnF \cdot Amn\text{-}1F \cdot Bmn\text{-}1F \\ & + BmnF \cdot Bmn\text{-}1F \cdot Amn\text{-}1F \end{aligned}$$

Figure 2.4-4 illustrates three examples of an absolute carry, corresponding to each of the OR terms of the equation for an absolute carry.

Bit	mn	mn-1	mn	mn-1	mn	mn-1
A	1		1	1		1
B	1			1	1	1

FIGURE 2.4-4. BIT COMBINATION FOR WmnR

Marginal Carry

For those bit positions which do require a carry from a prior bit position to develop a carry from their own bit position, an equation is written to indicate that the specific bit position in question has a marginal carry. The interpretation of a marginal carry is as follows: There is a carry from the bit position in question if there is a marginal carry preceeded by an absolute carry or a series of marginal carries preceeded by an absolute carry. The equation for a marginal carry, WmnM, is as follows; M indicates a marginal carry.

$$\begin{aligned} WmnM = {} & AmnF \cdot Amn\text{-}1F \\ & + AmnF \cdot Bmn\text{-}1F \\ & + BmnF \cdot Amn\text{-}1F \\ & + BmnF \cdot Bmn\text{-}1F \end{aligned}$$

Figure 2.4-5 illustrates four examples of a marginal carry corresponding to each of the OR terms of the equation for a marginal carry.

Bit	mn mn-1	mn mn-1	mn mn-1	mn mn-1
A	1 1	1	1	
B		1	1	1 1

FIGURE 2.4-5. MARGINAL CARRY FROM POSITION mn

Carry

With the knowledge of an absolute or marginal carry for each even bit position, it can then be determined if a carry from any even bit position does indeed exist. An example of the logical equation stating the existence of an actual carry for bit position 6 is as follows: C indicates a carry level. The term W00C indicates a carry into the low order end of the register.

$$\begin{aligned} W06C = {} & W06R \\ & + W06M \cdot W04R \\ & + W06M \cdot W04M \cdot W02R \\ & + W06M \cdot W04M \cdot W02M \cdot W00C \end{aligned}$$

The existence of an actual carry is seen to result from:

1. An absolute carry
2. A marginal carry preceeded by an absolute carry
3. A series of marginal carries preceeded by an absolute carry

PARALLEL ADDER LOGIC

The parallel adder logic makes redundant use of all the previously stated equations. In the mechanization of the carry logic, the various registers have their bits grouped in pairs of two. Thus, carry levels are only developed for even bit positions.

A consolidation of the basic adder logic which is utilized by the parallel adder follows:

1. All ODD bit positions use the equation in the form of Δ Bmn-1F
2. All EVEN bit positions use the equation in the form of Δ BmnF
3. The carry level WmnC is developed only for EVEN bit positions
4. The carry logic from ODD numbered bit positions is developed within the equation for Δ BmnF
5. The status of all absolute and marginal carries (true and false) is known prior to the actual add
6. The presence of all carry levels (WmnC) is known prior to the actual add

42 Bit Mantissa Adder

Figure 2.4-6 shows the two registers of the 42 bit mantissa adder.

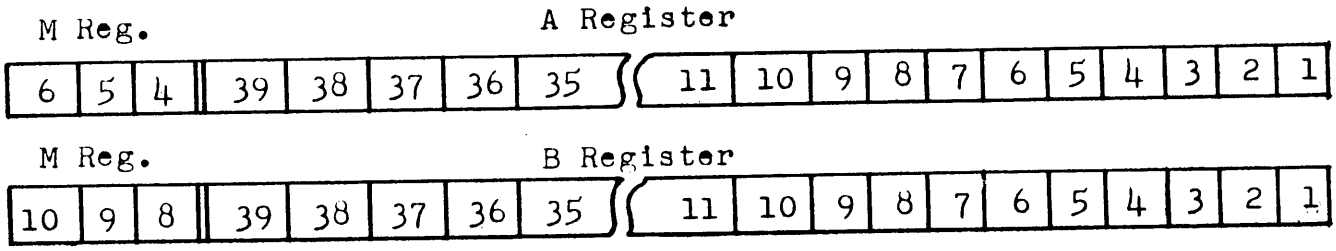


FIGURE 2.4-6. 42 BIT MANTISSA ADDER REGISTERS

The specific equations for setting the sum into the B register are as follows. Note that the equations are written in the general form, however, in the logic book some of the equations may appear in an inverted form. The term B02Z allows the gating of the sum into the B register whenever a 42 bit mantissa add is required. The term T52L' appears in the gating of the M register extension of the B register. The term T52L' functions to convert the 42 bit mantissa adder into a 39 bit mantissa adder, which is required during the Integer Store Operators for rounding off the B register.

Δ Bmn-1F = BO2Z • Wmn-1A 1 thru 39; ODD
Δ BmnF = BO2Z • WmnA 2 thru 38; EVEN
Δ MO8F = BO2Z • T52L' • W3WW
Δ MO9F = BO2Z • T52L' • W5WW
Δ M10F = BO2Z • T52L' • W2WW

Wmn-1A = Amn-1F • Wmn-2C'
+ Amn-1F' • Wmn-2C

WmnA = AmnF • (Amn-1F' • Bmn-1F'
+ Amn-1F' • Wmn-2C'
+ Bmn-1F' • Wmn-2C'

W5WW = MO5F • W4OC'
+ MO5F' • W4OC

+ AmnF' • (Amn-1F • Bmn-1F
+Amn-1F • Wmn-2C
+Bmn-1F • Wmn-2C)

W3WW = MO4F' • A39F • B39F
+ MO4F • A39F' • B39F'
+ MO4F • A39F' • W38C'
+ MO4F • B39F' • W38C'
+ MO4F' • B39F • W38C
+ MO4F' • A39F • W38C

W2WW = MO5F • MO9F • MO6F'
+ MO5F • W4OC • MO6F'
+ MO9F • W4OC • MO6F'
+ MO5F' • MO9F' • MO6F
+ MO5F' • W4OC' • MO6F
+ MO9F' • W4OC' • MO6F

Notice that the equations for W3WW and W2WW have the same form as WmnA.

10 Bit Address Adder

Figure 2.4-7 shows the two registers of the 10 bit address adder. The term M10Z allows the gating of the sum into the M register when the address adder is required. The equations of W1WW and W4WW have the same form as the equation for WmnW.

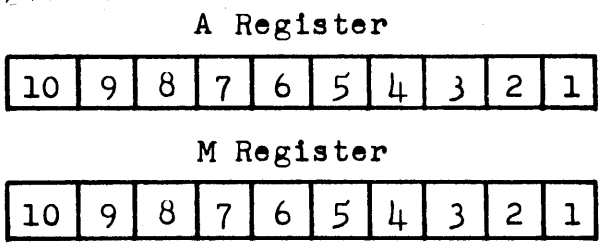


FIGURE 2.4-7. 10 BIT ADDRESS ADDER REGISTERS

Δ Mmn-1F = M10Z • Wmn-1A 1, 3, 5, 7
Δ MmnF = M10Z • WmnW 2, 4, 6
Δ M08F = M10Z • W1WW
Δ M09F = M10Z • W09A
Δ M10F = M10Z • W4WW

Wmn-1A = Amn-1F • Wmn-2C'
+ Amn-1F' • Wmn-2C

WmnW = AmnF • (Amn-1F' • Mmn-1F'
+ Amn-1F' • Wmn-2C'
+ Mmn-1F' • Wmn-2C'
+ AmnF' • (Amn-1F • Mmn-1F
+ Amn-1F • Wmn-2C
+ Mmn-1F • Wmn-2C)

W1WW = A08F' • A07F • M07F
+ A08F' • A07F • W06C
+ A08F' • W07F • W06C
+ A08F • A07F' • M07F'
+ A08F • A07F' • M06C'
+ A08F • M07F' • W06C'

W4WW = A10F' • M09F • A09F
+ A10F' • W08C • A09F
+ A10F' • M09F • W08C
+ A10F • A09F' • M09F'
+ A10F • A09F' • W08C'
+ A10F • M09F' • W08C'

Carry Levels

The development of the various carry levels WmnC, WmnR, and WmnM have been described. Because the carry levels are common to both the mantissa and address adder, the term Q09F is added to the carry level logic. When Q09F is SET the adder is functioning as an address adder; carry levels are developed from the A and M registers. When Q09F is RESET, the adder is functioning as a 42 bit mantissa adder; carry levels are developed from the A and B registers. The general form for a carry level follows:

WmnC = WmnR
+ WmnM • WmnR
+ WmnM • WmnM • MmnR
+ WmnM • WmnM • WmnM • W00C

NOTE

W00C ≡ Q01F. Q01F is set for a PRE-CARRY during subtraction or is set for adding in a one during a round operation.

WmnC is developed for each even bit position, W02C through W40C. For example, the carry level W40C is from the M register, bit positions 4 and 8; see Figure 2.4-6, the 42 Bit Mantissa Adder.

The absolute carry level logic, WmnR, has the term Q09F included for bit positions 02 through 10 and 40. It is not required of bit positions 12 through 38. Q09F indicates address adder and Q09F' indicates the mantissa adder.

Bit positions mn (02 through 10):

```
WmnR = AmnF • BmnF • Q09F'
     + AmnF • Amn-1F • Bmn-1F   Q09F'
     + BmnF • Bmn-1F • Amn-1F   Q09F'
     + AmnF • MmnF • Q09F
     + AmnF • Amn-1F • Mmn-1F • Q09F
     + MmnF • Mmn-1F • Amn-1F • Q09F
```

Bit position 40:

```
W40R = Q09F' • (M04F • M08F
               + M04F • A39F • B39F
               + M08F • A39F • B39F)
```

Bit positions mn (12 through 38); note that the equation is written in the Switched Form.

```
-0'-  WmnR
-I -  + AmnF' • Amn-1F'
      + AmnF' • BmnF'
      + AmnF' • Bmn-1F'
      + Amn-1F' • BmnF'
      + BmnF' • Bmn-1F'
```

The marginal carry logic, WmnM, has the term Q09F included for bit positions 02 through 10 and 40; it is not required for bit positions 12 through 38.

Bit positions mn (02 through 10):

```
WmnM = AmnF • Amn-1F
     + AmnF • Bmn-1F • Q09F'
     + AmnF • Mmn-1F • Q09F
     + Amn-1F • BmnF • Q09F'
     + Amn-1F • MmnF • Q09F
     + BmnF • Bmn-1F • Q09F'
     + MmnF • Mmn-1F • Q09F
```

Bit position mn (12 through 38);

```
-0 -  WmnM
-I-   + AmnF • Amn-1F
      + AmnF • Bmn-1F
      + Amn-1F • BmnF
      + BmnF • Bmn-1F
```

ADDER LEVELS

The levels W10L, W13L and W14L are developed for use externally in the adder logic. The levels W10L and W14L are carry levels from EVEN bit positions. The level W10L indicates a carry from the 10 bit position of the address adder. W10L has the same development as other carry logic.

NOTE

The level W10C is not developed for general adder usage, therefore, W10L is utilized.

W10L = W10R
+ W10M • W08R
+ etc.

The level W14L indicates a carry from the 14th octal sum digit. Its development is similar to other carry logic.

W14L = M06F • M10F
+ M06F • M05F • M09F
+ M06F • M05F • W40C
+ M06F • M09F • W40C
+ M05F • M10F • M09F
+ M05F • M10F • W40C
+ M10F • M09F • W40C

The level W13L indicates a carry from the 13th octal sum digit. The bit position corresponding to the 13th octal sum digit does not correspond to an even bit position. Therefore, the carry level W13L is from bit position 39, an odd bit position.

W13L = A39F • B39F
+ W50B • W38R
+ W50B • W38M • W36R
+ etc.
W50B = A39F + B39F

2.5 SERIAL DECIMAL ADDER

The Serial Decimal Adder is a D.C. level binary adder which sums two binary coded decimal digits in the Y and Z registers and places the straight binary sum into the Z register. Figure 2.5-1 shows the Serial Decimal Adder in a block diagram form. The inputs to the Y and Z register are from the A and B registers, respectively; a character at a time. Because the resulting sum of the addition of the Y and Z registers is in a straight binary form, it is necessary to convert the sum into a binary coded decimal digit before storing the sum back into the B register.

ADDER INPUTS

The input to the Y register is from the A register. Each character that is transferred to the Y register is the character specified by the G register. Because the A and B bit position of the least significant character in a field contains the sign, all bits of a character are transferred to the Y register.

The input to the Z register is from the second and third octade positions of the B register. All bit positions of a character are transferred to the Z register to facilitate sign comparison.

The decimal adder is only mechanized for addition, therefore, subtraction is accomplished via complement addition. In the performance of complement addition it is necessary to complement either the Y or Z register input levels to the adder logics. Addition logic states that the smaller of the two fields (in magnitude) is complemented. If it is required that the Y input be complemented, the output levels of the Y register are gated into the adder logics in complement form. If it is required that the Z input be complemented, the character transferred to the Z register from the B register is in complement form.

When complement addition is required, the status of Q02F and Q04F states which of the two inputs is to be complemented. Refer to Field Add flow. The various states of Q02F and Q04F are as follows:

Q02F' • Q04F' = No Complement
Q02F • Q04F' = Complement Y
Q02F • Q04F = Complement Z

Figure 2.5-2 shows the decimal adder in block form with specific levels labeled. The Z register is shown in two places for clarity.

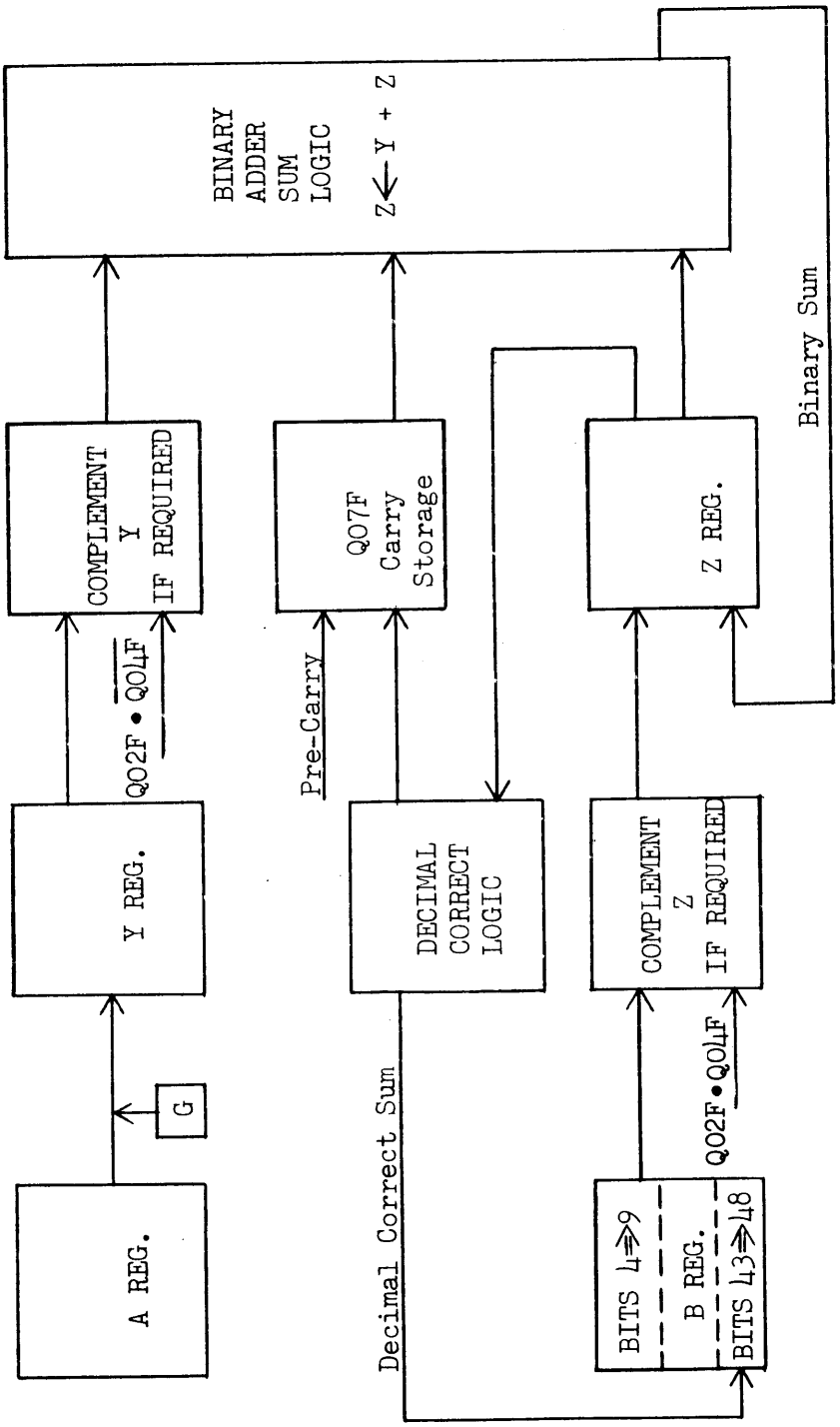
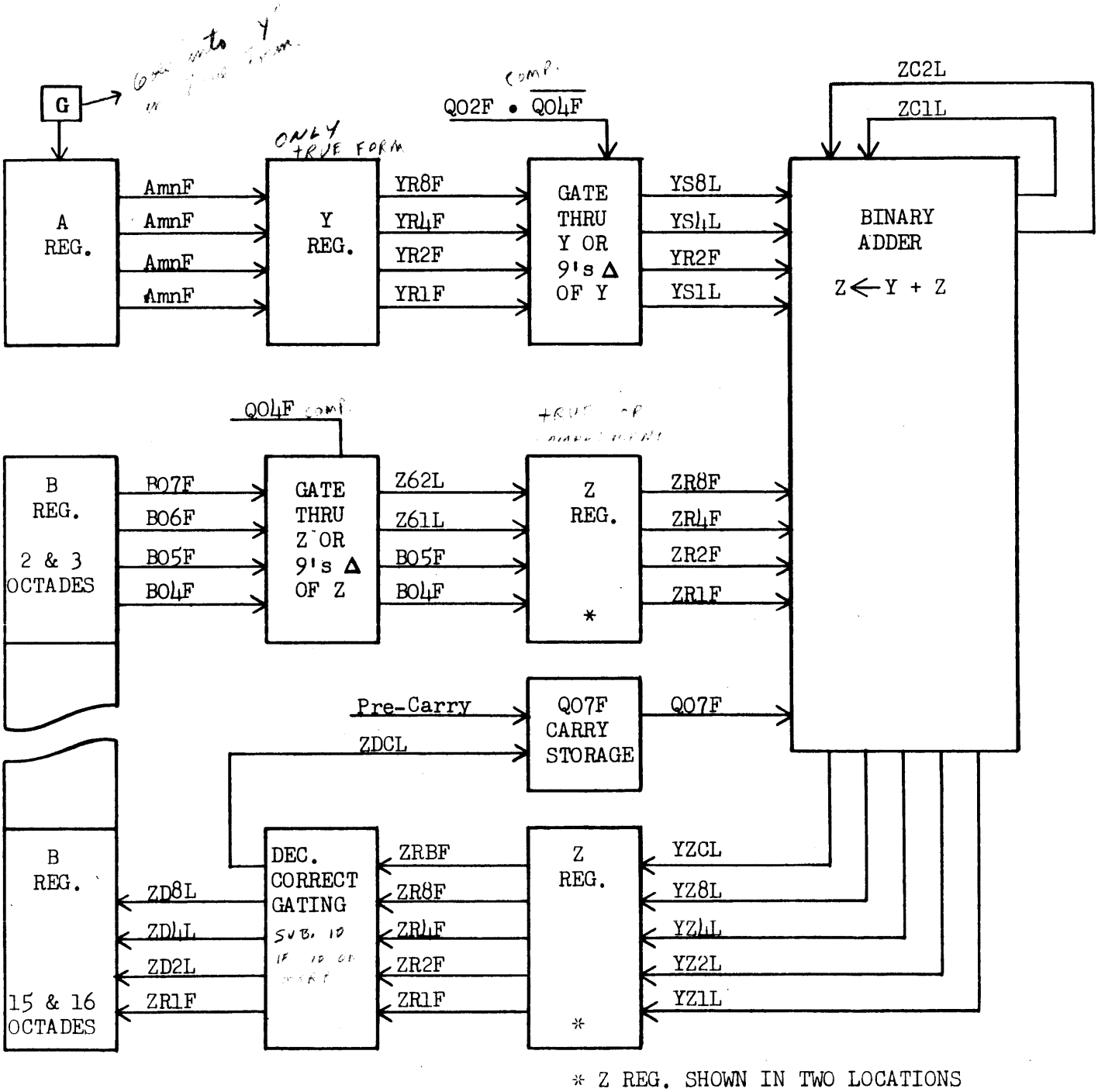


FIGURE 2.5-1. SERIAL DECIMAL ADDER



* Z REG. SHOWN IN TWO LOCATIONS

FIGURE 2.5-2. ADDER LEVELS

COMPLEMENT INPUT

When a complement input to the adder is required, the nines complement of each binary coded decimal digit is developed. Figure 2.5-3 shows the binary coded decimal digit input characters with their corresponding nines complement. Note that decimal digits 0 through 9 are the only possible input digits.

DECIMAL	INPUT	9's COMPLEMENT
	8 4 2 1	8 4 2 1
0	0 0 0 0	1 0 0 1
1	0 0 0 1	1 0 0 0
2	0 0 1 0	0 1 1 1
3	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 0 1
5	0 1 0 1	0 1 0 0
6	0 1 1 0	0 0 1 1
7	0 1 1 1	0 0 1 0
8	1 0 0 0	0 0 0 1
9	1 0 0 1	0 0 0 0

FIGURE 2.5-3. NINES COMPLEMENT

The logic rules which state when to complement a specific bit to develop a nines complement are as follows. If none of the rules apply, no complement is required.

1's bit = Always Complement
2's bit = Never Complement
4's bit = Only if 2's bit is on
8's bit = Only if both the 2's and 4's bit are off

Z Register Inputs. The input to the Z register may be in complement or non-complement form. The following equations state the input gating to the Z register. Q04F on, indicates a nines complement input. The term Z02TD1 allows the Z register set and reset levels to be effective.

ZR1F = Z02TD1 • B04F • Q04F'
+ Z02TD1 • B04F' • Q04F

ZR1F'= Z02TD1 • B04F' • Q04F'
+ Z02TD1 • B04F • Q04F

ZR2F = Z02TD1 • B05F
ZR2F'= Z02TD1 • B05F'

ZR4F = Z02TD1 • Z61L
ZR4F'= Z02TD1 • Z61L'

ZR8F = Z02TD1 • Z62L
ZR8F'= Z02TD1 • Z62L'

ZRAF = Z02TD1 • B08F
ZRAF'= Z02TD1 • B08F'

ZRBF = Z02TD1 • B09F
ZRBF'= Z02TD1 • B09F'

Z61L = B06F • B05F'
+ B06F' • B05F • Q04F
+ B06F • Q04F'

Z62L = B07F' • B06F' • B05F' • Q04F
+ B07F • Q04F'

Y Register Input

The Y register input is a direct transfer from the A register. The G register points to the character that is transferred. The following equation shows the transfer gating input to the Y register in simplified form. Note that the Common And Gate (CAG) levels correspond to the G register setting.

m = 1, 2, 4, 8, A or B

YRmF =	CAGY07 • AnnF	nn = 01⇒06
+	CAGY06 • AnnF	nn = 07⇒12
+	CAGY05 • AnnF	nn = 13⇒18
+	CAGY04 • AnnF	nn = 19⇒24
+	CAGY03 • AnnF	nn = 25⇒30
+	CAGY02 • AnnF	nn = 31⇒36
+	CAGY01 • AnnF	nn = 37⇒42
+	CAGY00 • AnnF	nn = 43⇒48

Y Register Output

The Y register output levels which are input levels to adder logics may be in complement or non-complement form. The following logical equations state the gating of the Y register output levels in complement or non-complement form. Note that gating for the 2's bit is absent, because it is never required to complement the 2's bit to obtain the nines complement.

Q02F' • Q04F' = No Complement
Q02F • Q04F' = Complement Y

YS1L =	YR1F • Q02F'	No Δ
+	YR1F • Q04F	Δ Z
+	YR1F' • Q02F • Q04F'	Δ Y

YS4L =	YR4F • YR2F'	No Δ
+	YR4F • Q02F'	No Δ
+	YR4F • Q04F	Δ Z
+	YR4F' • YR2F • 211L	Δ Y

211L = Q02F • Q04F'

ADDER LOGIC

The adder logic accepts the input levels from the Y and Z registers as binary coded decimal digits, and develops complement logic to place the sum in the Z register in straight binary form. Figure 2.5-4 shows the Y and Z registers with the corresponding carry levels indicated. Q07F is the carry storage input from a previous decimal digit or is a pre-carry in complement addition. The carry levels ZC1L, ZC2L and YZCL are developed from bit positions 1, 2 and 8, respectively. A carry level from bit position four is not developed.

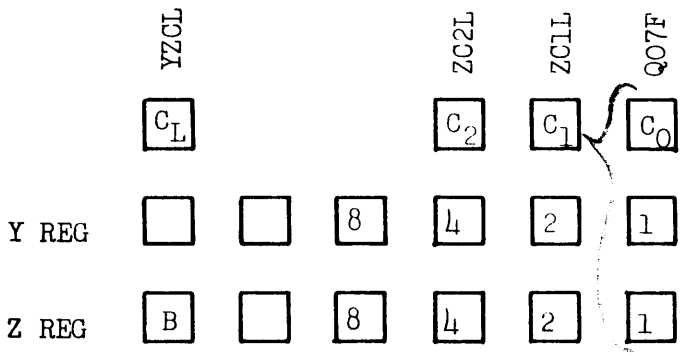


FIGURE 2.5-4. CARRY LEVELS

The carry level YZCL from the 8 bit position is a binary carry and therefore has a decimal value of 16. When the sum of Y plus Z is placed in Z, the carry level YZCL is stored in the B bit position of the Z register.

The basic add logic of placing the sum in the Z register by complementing or non-complementing the Z register is similar to the logic utilized in the Parallel Adder. The basic logical equation in simplified form is as follows:

$$\Delta Z \text{ bit} = (Y \text{ bit} \bullet \text{Bit CARRY}') + (Y \text{ bit}' \bullet \text{Bit CARRY})$$

Z Register Sum Input

The equations for complementing the Z register and thus setting the binary sum in the Z register are as follows. In the logic book, the equations are in switched form. The level 614L allows gating the sum into the Z register during Field Add/Sub.

$$\Delta ZR1F = 614L \bullet YZ1L$$
$$\Delta ZR2F = 614L \bullet YZ2L$$
$$\Delta ZR4F = 614L \bullet YZ4L$$
$$\Delta ZR8F = 614L \bullet YZ8L$$
$$\Delta ZRBF = 614L \bullet YZCL$$

$$YZ1L = (YS1L \bullet QO7F') + (\overline{YS1L} \bullet QO7F)$$
$$YZ2L = (YR2F \bullet ZC1L') + (YR2F' \bullet ZC1L)$$
$$YZ4L = (YS4L \bullet ZC2L') + (YS4L' \bullet ZC2L)$$
$$YZ8L = YS8L \bullet ZR4F' + YS8L \bullet ZC2L' + YS8L' \bullet YS4L \bullet ZR4F + YS8L' \bullet YS4L \bullet ZC2L + YS8L' \bullet ZR4F \bullet ZC2L$$

In the interpretation of the equation for YZ8L, it must be remembered that the only possible inputs to the adder are the digits 0 through 9. Therefore, the 8 and 4 bits will never be true at the same time.

Carry Logic

The equations to develop carry levels from the first, second and eighth bit positions are as follows. The carry from the fourth bit position is a part of the equation for YZ8L and YZCL.

ZC1L = YS1L • ZR1F
+ YS1L • QO7F
+ ZR1F • QO7F

ZC2L = YR2F • ZR2F
+ YR2F • ZC1L
+ ZR2F • ZC1L

YZCL = YS8L • ZR8F
+ YS8L • ZR4F • ZC2L
+ ZR8F • YS4L • ZC2L

BINARY SUM CORRECTION

The binary sum in the Z register passes through decimal correct gating before being stored in the 15th and 16th octade positions of the B register. Figure 2.5-5 contains a truth table for decimal correcting of binary digits 0 through 19.

DECIMAL DIGIT	Z REG. BIT B 8 4 2 1	QO7F	B REG. BITS 46/8 45/4 44/2 43/1
0	0 0 0 0 0	0	0 0 0 0
1	0 0 0 0 1	0	0 0 0 1
2	0 0 0 1 0	0	0 0 1 0
3	0 0 0 1 1	0	0 0 1 1
4	0 0 1 0 0	0	0 1 0 0
5	0 0 1 0 1	0	0 1 0 1
6	0 0 1 1 0	0	0 1 1 0
7	0 0 1 1 1	0	0 1 1 1
8	0 1 0 0 0	0	1 0 0 0
9	0 1 0 0 1	0	1 0 0 1
10	0 1 0 1 0	1	0 0 0 0
11	0 1 0 1 1	1	0 0 0 1
12	0 1 1 0 0	1	0 0 1 0
13	0 1 1 0 1	1	0 0 1 1
14	0 1 1 1 0	1	0 1 0 0
15	0 1 1 1 1	1	0 1 0 1
16	1 0 0 0 0	1	0 1 1 0
17	1 0 0 0 1	1	0 1 1 1
18	1 0 0 1 0	1	1 0 0 0
19	1 0 0 1 1	1	1 0 0 1

FIGURE 2.5-5. DECIMAL CORRECTION TRUTH TABLE

The equations for setting the decimal corrected binary coded decimal digit sum into the 15th and 16th octade positions of the B register are as follows.

NOTE

During Field Add, the 16th octade is cleared prior to setting the sum into the 15th and 16th octades. The 16th octade is reset at the time the binary sum is placed in the Z register.

The level E31L permits gating the sum into the B register during Field Add/Sub.

B43F = E31L • ZR1F
B43F' = E31L • ZR1F'

B44F = E31L • ZD2L
B44F' = E31L • ZD2L'

B45F = E31L • ZD4L
B45F' = E31L • ZD4L'

B46F = E31L • ZD8L
Q07F = E31L • ZDCL

	Decimal Digits
ZD2L = ZRBF' • ZR8F' • ZR2F	2, 3, 6, 7
+ ZR8F • ZR4F • ZR2F'	12, 13
+ ZRBF • ZR2F'	16, 17
ZD4L = ZR4F • ZR2F	6, 7, 15, 16
+ ZR8F' • ZR4F	4, 5, 6, 7
+ ZRBF • ZR2F'	16, 17
ZD8L = ZR8F • ZR4F' • ZR2F'	8, 9
+ ZRBF • ZR4F' • ZR2F	18, 19
ZDCL = ZRBF	16, 17, 18, 19
+ ZR8F • ZR4F	12, 13, 14, 15
+ ZR8F • ZR2F	10, 11

2.6 INPUT CONVERT: B29T Flow Chart

The B29T Flow Chart is to be used in conjunction with the INPUT CONVERT Flow Chart. (See write up on INPUT CONVERT Flow Chart.) The B29T Flow Chart illustrates the logic for shifting the B register in the attainment of the conversion of binary coded decimal to straight binary. The actions described by the B29T Flow Chart occur during J = 3 of INPUT CONVERT.

CONVERSION PRINCIPAL

The conversion from binary coded decimal to straight binary is based on the principal of repeated division of the decimal number by 2, with the successive remainders forming the binary number. The first remainder to be developed forms the least significant binary bit, successive remainders form the more significant binary bits. An example of the conversion of the decimal number 151 to binary is shown in Figure 2.6-1.

151	Div. by 2 =	75 -	Rem. of	1	1
75	" =	37 -	"	1	1 1
37	" =	18 -	"	1	1 1 1
18	" =	9 -	"	0	0 1 1 1
9	" =	4 -	"	1	1 0 1 1 1
4	" =	2 -	"	0	0 1 0 1 1 1
2	" =	1 -	"	0	0 0 1 0 1 1 1
1	" =	0 -	"	1	1 0 0 1 0 1 1 1

Decimal 151 = Binary 10010111

FIGURE 2.6-1. DECIMAL 151 TO BINARY

SHIFT REGISTER

The process of repeated division by 2 is accomplished by repeated shifting of a register containing the number to be converted. With the binary coded decimal number entered into a shift register divided in decades, the division by 2 is performed by shifting towards the least significant digit position. However, when a one is shifted from the 1's position of a decade to the 8's position of the next lower decade, the value of this one changes from 10 to 8 rather than 10 to 5. This means that when shifting a binary one from a decade to the next lower decade, a 3 must be subtracted to effect the division by 2.

An example of the conversion of the binary coded decimal number 27 via a shift register is illustrated in Figure 2.6-2.

TENS	UNITS		
8 4 2 1	8 4 2 1		
0 0 1 0	0 1 1 1		shift
0 0 0 1	0 0 1 1	1	shift
0 0 0 0	1 0 0 1	1 1	shift
0 0 0 0	0 1 1 0	1 1	subtract 3
0 0 0 0	0 0 1 1	0 1 1	shift
0 0 0 0	0 0 0 1	1 0 1 1	shift
0 0 0 0	0 0 0 0	1 1 0 1 1	

B.C.D. 27 = Binary 11011

FIGURE 2.6-2. B.C.D. CONVERSION VIA SHIFTING

In the B 5500 each decade is checked by each clock pulse. When a "0" is stored in the least significant bit position of the (n+1)th digit, the 3 most significant bits of the nth digit and the least significant bit of the (n+1)th digit are shifted one position to the right.

If a "1" is stored in the least significant bit position of the (n+1)th digit, the bits of the nth digit are modified through a switching network. This network performs the action of shifting to the right and simultaneously subtracting 3.

NOTE

The logic of the switching network is the same for each decade. The most significant decade needs no switching network since for this decade the least significant bit of the next higher decade is always zero.

Logic

The logic for shifting and subtracting 3 in any one decade is as follows: Let FF's n equal the four binary bit positions of any one decade; i.e., FFn1 = 1's bit; FFn2 = 2's bit, FFn3 = 4's bit; and FFn4 = 8's bit. The equations which follow are the input equations for the flip-flops of decade n, assuming that FF(n+1)1 equals one. J equals set input and K equals reset input.

J-FFn1 = FFn2'
K-FFn1 = FFn2
J-FFn2 = FFn3
K-FFn2 = FFn3
J-FFn3 = FFn4'
K-FFn3 = FFn2
J-FFn4 = FFn3 • FFn2
K-FFn4 = 0

Figure 2.6-3 illustrates the application of the equations for setting the flip-flop of the nth decade when the 1's bit of the (n+1) decade equals one. The nth decade (before) illustrates values of 0 through 9; the nth decade (after) illustrates the values after shifting and subtracting 3 is accomplished.

n+1	nth Before	nth After	
3 2 1	4 3 2 1	4 3 2 1	
4 2 1	8 4 2 1	8 4 2 1	
1	0 0 0 0	0 1 0 1	8-3 = 5
1	0 0 0 1	0 1 0 1	8-3 = 5
1	0 0 1 0	0 1 1 0	9-3 = 6
1	0 0 1 1	0 1 1 0	9-3 = 6
1	0 1 0 0	0 1 1 1	10-3 = 7
1	0 1 0 1	0 1 1 1	10-3 = 7
1	0 1 1 0	1 0 0 0	11-3 = 8
1	0 1 1 1	1 0 0 0	11-3 = 8
1	1 0 0 0	1 0 0 1	12-3 = 9
1	1 0 0 1	1 0 0 1	12-3 = 9

FIGURE 2.6-3. SETTING FLIP-FLOPS OF NTH DECADE

B29T FLOW CHART

The B29T Input Convert flow chart illustrates the logic for setting and resetting the specific bits of the B register to accomplish the input convert operation. The levels term W1(n)I is true when the corresponding next higher decades 1's bit is equal to zero; in which case the only required actions for the respective decades is to shift. The levels term W0(n)I is true when the corresponding next higher decades 1's bit is equal to one; in which case the required action is to shift and subtract 3.

Note that the input to a specific bit position ending in "X" refers to a set input. The input terms ending in "Y" refer to a reset input. For example B03X implies the B03F is to be set.

2.7 OUTPUT CONVERT: B30T Flow Chart

The B30T Flow Chart is to be used in conjunction with the OUTPUT CONVERT Flow Chart. (See write up on OUTPUT CONVERT Flow Chart.) The B30T Flow Chart illustrates the logic for shifting the B register in the attainment of the conversion of straight binary to binary coded decimal. The actions described by the B30T Flow Chart occur during J = 3 of OUTPUT CONVERT.

CONVERSION PRINCIPAL

The conversion from straight binary to binary coded decimal makes use of the fact that the value of any binary bit is equal to 2^{n-1} where n is the location of the bit in the binary number. Thus, the decimal value of a binary number may be determined by adding appropriate powers of 2 as indicated by the presence of a 1 in the binary number. The power of 2 may be calculated by doubling each binary bit n-1 times. Doubling of the binary bits is accomplished via a shift register, since shifting the entire register once toward the most significant bit position doubles all binary stages in the register.

When incorporating this procedure in a binary coded decimal (B.C.D.) register, it must be noted that shifting a 1 from the 8's bit position of a decade to the 1's bit position of the next higher decade does not double the value of the bit as occurs when shifting between bit position within a decade. That is, when a 1 is shifted from the 8's bit position of a decade to the 1's bit position of the next higher decade, the value of the bit only increases from 8 to 10 rather than 8 to 16. This means in this case, that an additional 6 must be added to the "next higher decade" to account for the missing value of 6. This value of 6 may be added into the "next higher decade" by adding 3 to the lower significant decade prior to shifting; hence, after shifting the value of 6 is effectively added to the "next higher decade".

In a binary coded decimal register, a carry must occur from any one decade whenever the value of the digit in the decade exceeds the value of 9. That is, if the decimal digits in each decade are to remain in proper form and not exceed the value of 9. If the value of the current digit in any one decade equals 6 or more before shifting, its value will exceed the value of 9 after shifting and thereby require a decimal carry to be developed. If it can be determined that the value of decades contents will exceed the value of 9 after shifting (current value equals 5 or more), the previously described problem of a 1 in the 8's bit position of a decade being shifted to the 1's bit position of the next higher decade results; i.e., the loss in the value of 6. Therefore, if the current value of a digit is equal to 5 or more before shifting, it is known that a decimal carry will result, which requires that a value of 3 be added to the digit prior to the shift. The net result is that all digits remain in proper binary coded decimal form; no digit position exceeds the value of 9 either before or after shifting.

In the B 5500, each decade is checked by each clock pulse. If the decades content equals less than 5, the decade is just shifted one bit position to the left. If the decades content equals 5 or more, the decades three most significant bit positions (8, 4 and 2) and the least significant bit position of the next higher decade are modified through a switching network. This switching network has an action which is equivalent to adding in the value of 3 and shifting one position to the left.

Logic

The logic for adding 3 and shifting (switching network) is the same for each decade and is as follows: Let FF's n equal the three most significant binary bit positions of any one decade and FF(n+1) equal the least significant binary bit position of the next more significant decade. That is, FFn2 = 2's bit; FFn3 = 4's bit; FFn4 = 8's bit; and FF(n+1) = 1's bit. The equations which follow are the input equations for the three most significant flip-flops of decade (n) and the least significant flip-flop of decade (n+1). The leading designation by the letter J indicates the set input, and the letter K indicates the reset input. It is assumed that the value of the current digit in the nth decade equals 5 or more; otherwise, the equations are not valid.

J-FFn2 = FFn1'
K-FFn2 = FFn1
J-FFn3 = FFn1'
K-FFn3 = FFn1' + FFn2'
J-FFn4 = 0
K-FFn4 = FFn1'
J-FF(n+1) = FFn4 + FFn3 • FFn2 + FFn3 • FFn1
K-FF(n+1) = FFn4' • FFn3' + FFn3 • FFn2' • FFn1'

Figure 2.7-1 illustrates the application of the equations for setting the three most significant bit positions of decade (n) and the least significant bit position of the (n+1) decade. The nth decade (before) illustrates values 0 through 9; the nth and nth+1 (after) illustrates the values after shifting. Note that for values less than 5 only shifting is illustrated. For values 5 or more, adding of 3 and shifting is illustrated as the final result.

nth+1 After	nth After	nth Before	
3 2 1	4 3 2 1	4 3 2 1	Bit Position
4 2 1	8 4 2 1	8 4 2 1	Binary Value
0	0 0 0 X	0 0 0 0	Less than 5
0	0 0 1 X	0 0 0 1	
0	0 1 0 X	0 0 1 0	
0	0 1 1 X	0 0 1 1	
0	1 0 0 X	0 1 0 0	
1	0 0 0 X	0 1 0 1	5 or more
1	0 0 1 X	0 1 1 0	
1	0 1 0 X	0 1 1 1	
1	0 1 1 X	1 0 0 0	
1	1 0 0 X	1 0 0 1	

FIGURE 2.7-1. DECADE BIT SHIFT

An example of binary to decimal conversion is illustrated in Figure 2.7-2. In Figure 2.7-2 the binary number (1100111) is converted to decimal 103 via shifting according to the principles previously described. Note that the addition of 3 and shifting (when required) is illustrated in two steps to more clearly show the action.

B30T FLOW CHART

The B30T Output Convert Flow Chart illustrates the logic for setting and resetting the specific bits of the B register to accomplish the output convert operation. The levels termed W0(n)Ø are true when the corresponding nth decade is equal to a value of 5 or more. In this case, n decade may have values of 1 through 8, referring to the least significant to the most significant decade and numbered 1 through 8, respectively.

The levels termed W1(n)Ø are true when the corresponding nth decade contains a value less than 5, in which case, only shifting is required.

Note that the input to a specific bit position ending in the letter "X" refers to a set input. The input terms ending in the letter "Y" refer to a reset input.

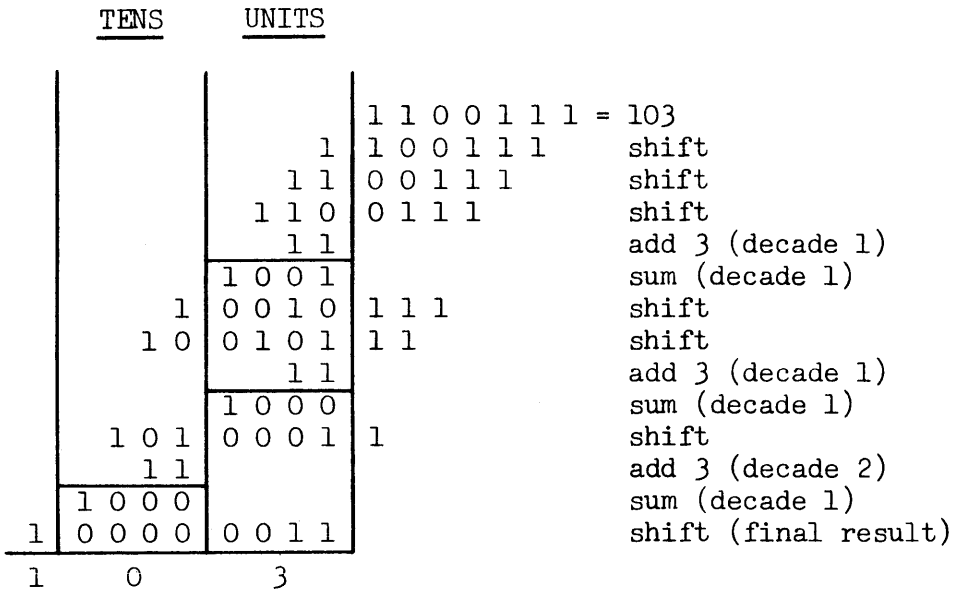


FIGURE 2.7-2. BINARY TO DECIMAL CONVERSION

SECTION 3

WORD MODE OPERATIONAL FLOWS

3.1 LOGICAL AND 0415
LOGICAL OR 0215

The LOGICAL AND operator will set a ZERO (False) into each bit position of the A register, except the flag bit (48), when a ZERO appears in the corresponding bit position of the B register.

The LOGICAL OR operator will set a ONE (True) into each bit position of the A register, except the flag bit, if the corresponding bit position in the B register is a ONE.

SUMMARY OF OPERATION

The operators require both the A register and the B register to be occupied. If they are not, then the required stack push up will take place.

The LOGICAL AND function will result in a True indication if the corresponding bits in the A register and the B register are both True. The LOGICAL OR function will result in a True indication if either or both of the corresponding bits of the A register and B register are True. The result will appear in the A register; the B register will be set to empty. The flag bit of the result in the A register will be the same as the flag bit was in the B register.

DETAILED DESCRIPTION

JOOL

The actions above the double line are for stack push up if required. The actions below the double line are to implement the LOGICAL AND/OR operator.

$\overline{\text{AROF}}$

If the A register is unoccupied, then shift the contents of the B register to the A register, A←B. Mark the B register as empty, BROF to 0.

$\overline{\text{AROF}} + \overline{\text{BROF}}$

If either the A register or the B register are unoccupied, then initiate a memory read operation, E←3, and exit from this J count to J = 1.

$\overline{\text{AROF}} \cdot \text{BROF}$

A shift from B register to A register occurred with $\overline{\text{AROF}}$ so if the B register was occupied then mark the A register occupied.

AROF • BROF

This condition indicates that the A and B registers are full and the actions listed below the double line can occur.

AROF • BROF • LOAL

With both registers occupied and a LOGICAL AND operator, perform a single sided transfer (A' to B') of the False outputs of B[47=>1] into the KCL (reset) inputs of the A[47=>1]. This results in a True indication only if both corresponding bits of the registers are True.

AROF • BROF • LOOL

With both registers occupied and a LOGICAL OR operator, perform a single transfer, A to B, of the True outputs of B[47=>1] into the JCL (set) inputs of A[47=>1]. This results in a True indication if either or both corresponding bits are True.

AROF • BROF

With both registers occupied, mark the B register as empty, BROF to 0, and exit from this operator, (EXIT). The term EXIT will enable the Syllable Execute Complete Level (SECL) and result in the termination of this operator.

Also transfer the flag bit in the B register to the flag bit position of the A register, (A48F<-B48E).

JOIL

MROF

With MROF set the information is in the MIR register of core memory, shift it to the B register. Indicate that B register is occupied by BROF 1. Also count the S register down 1, S-1, and return to JOOL, (J←0). This J register setting (JOIL) is used for stack push up only.

3.2 LOGICAL EQUIVALENCE LOEL-1015

Set a one in each position of the B register, except the flag bit, when the corresponding bit positions of the A and B register are equal. Set a zero in each position of the B register, except the flag bit, when the corresponding bit positions of the A and B register are not equal. The flag bit of the B register is unaltered. The A register is set to empty.

SUMMARY OF OPERATION

The EQUIVALENCE comparison by bit, is performed between the three bit positions of the sixteenth octade of the A and B registers. Bit positions 46, 47 and 48 of the A register are compared against bit positions 46, 47 and 48 of the B register. The results of the EQUIVALENCE comparison are stored in the first octade position of the B register, bit positions 1, 2 and 3. Each time the first octade position of the B register is set to the results of an EQUIVALENCE comparison, both the A and B registers are shifted left one octade position. The left shifts by octade are tallied by the N register.

DETAILED DESCRIPTION

J = 0

The actions above the double line load the A and B registers if they are not already loaded, the actions below the double line perform the logical EQUIVALENCE function.

If either the A or B register is unoccupied, a normal push up from the stack occurs.

ICFL = Q25L • T50L
T50L = Logical EQUIVALENCE
Q25L = (J = 0) (AROF' + BROF')

All of the actions that follow require the A and B registers to be occupied (AROF • BROF).

B03F to 1 or B03F to 0

The first clock pulse which finds the A and B registers occupied is when the A and B registers are in their original form. The term N register equal to zero (NEZL), indicates that no octade shifts have taken place. Therefore, the forty-eighth bit position of the B register (flag bit) is set directly into B03F, thus forming that portion of the new sixteenth octade position.

During the subsequent fifteen octade positions, B03F is set to one if A48F and B48F are EQUAL (A48F • B48F + A48F' • B48F'), otherwise B03F is set to zero. Subsequent octade positions, other than the flag bit position, are indicated by the term NEZL' being true.

B02F is 1 or B02F to 0

During all octade positions, B02F is turned on if both the corresponding two bits being checked (A47F and B47F) are EQUAL. If A47F and B47F are not both set or both reset, B02F is turned off.

B01F to 1 or B01F to 0

During all octade positions, B01F is turned on if both the two corresponding bits (A46F and B46F) being checked are EQUAL. If A46F and B46F are not EQUAL (A46F' • B46F + A46F • B46F'), B01F is turned off.

J to 2

When N15L is true, the N register has been counted up to 15 to tally 15 left shifts of the A and B registers. The clock pulse which sets the J register to 2, also gates the last required EQUIVALENCE comparison and left shift. Thus, the B register is restored to its original form with the flag bit again in the sixteenth octade position.

Shift A and B left by Octades and N + 1

As each EQUIVALENCE comparison is performed, the A and B registers are shifted left by octades. Note that the A and B registers are not circulated, but that new information is developed as the B register is shifted left toward its final position. The last EQUIVALENCE comparison occurs when N equals 15, thereafter the N register is counted up by one, leaving the N register equal to zero.

J = 1

This J register setting is associated with the loading of the B register, at J equals 0, during a normal push up if required.

J = 2

The A register is marked unoccupied by the resetting of AROF, and the Syllable Execute Complete Level (SECL) is gated true to end the logical EQUIVALENCE operator.

3.3 LOGICAL NEGATE

LONL-0115

Complement every bit position of the A register except the flag bit which is left unaltered.

SUMMARY OF OPERATION

If the A register is empty and the B register is full, the B register contents are transferred to A and the B register is marked unoccupied.

If both registers are empty, the A register is loaded directly and the B register remains unoccupied. Thereafter, the A register is complemented and the operator terminated.

DETAILED DESCRIPTION

J = 0

If neither the A or B register is loaded, the E register is set to 2 initiating a load of the A register. Once the A register is occupied, the A register is complemented. The flag bit is left unaltered.

The clock pulse which gates the complement of the A register into the A register also gates the Syllable Execute Complete Level (SECL) true, thus terminating the operator.

A01 ⇒ 39F = A03T
A01 ⇒ 39F' = A03T
A03T = AROF • C03L • T07F
C03L • T07F = Logical Negate

A40 ⇒ 45F = A05T
A40 ⇒ 45F' = A05T
A05T = AROF • C03L • T07F

A46 and 47F = AROF • C03L • T07F
A46 and 47F' = AROF • C03L • T07F

J = 1

Entry to this J count is due to the setting of E to 2 for a load of the A register. When the Memory Read Access obtained level is available, the J register is returned to zero, the S register counted down and the A register marked occupied.

3.4 RELATIONAL OPERATORS

COMPARE:	B Greater than A	BGAL - 0225
	B Greater than or Equal to A	BGEL - 0125
	B Equal to A	BEQL - 4425
	B Less than or Equal to A	BLEL - 4125
	B Less than A	BLAL - 4225
	B Not Equal to A	BNEL - 0425

The relational operators perform comparisons on the two top operands in the stack. The operands are removed from the stack and the result of the comparison is placed in the top of the stack. The operands may be in an un-normalized form and the required scaling will take place in the comparison operation. For the relational operators, operands of zero, minus zero and a zero mantissa, with a non-zero exponent, are considered equal. Flag bits are ignored.

The operand in the B register is algebraically compared with the operand in the A register. Depending on the operator level, if the value of the operand in the B register is algebraically: greater than; greater than or equal to; equal to; less than or equal to; less than; not equal to, the value of the operand in the A register, the low order bit of the B register is set to one and all other bits of the B register are set to zero. Otherwise, all bits of the B register are set to zero. The A register is set to empty.

SUMMARY OF OPERATION

Refer to Figure 3.4-1 for a block diagram flow of the logic utilized in the compare operators. The A and B registers must be loaded if they are not loaded on entry. If the mantissa of either the A or B register is zero, or if the signs of the mantissa are different, the comparison is completed on the basis of the mantissa signs and possible zero mantissa.

If neither mantissa is zero and the mantissa signs of A and B are alike, it is necessary to determine if the exponents are equal, or if both mantissas are normalized. If the exponents are not equal and the A and B registers are not normalized, the A and B registers are normalized before the comparison can continue. When one or both of the conditions are met (exponents equal or normalized registers), the A register is complemented enabling a comparison of the mantissa magnitudes.

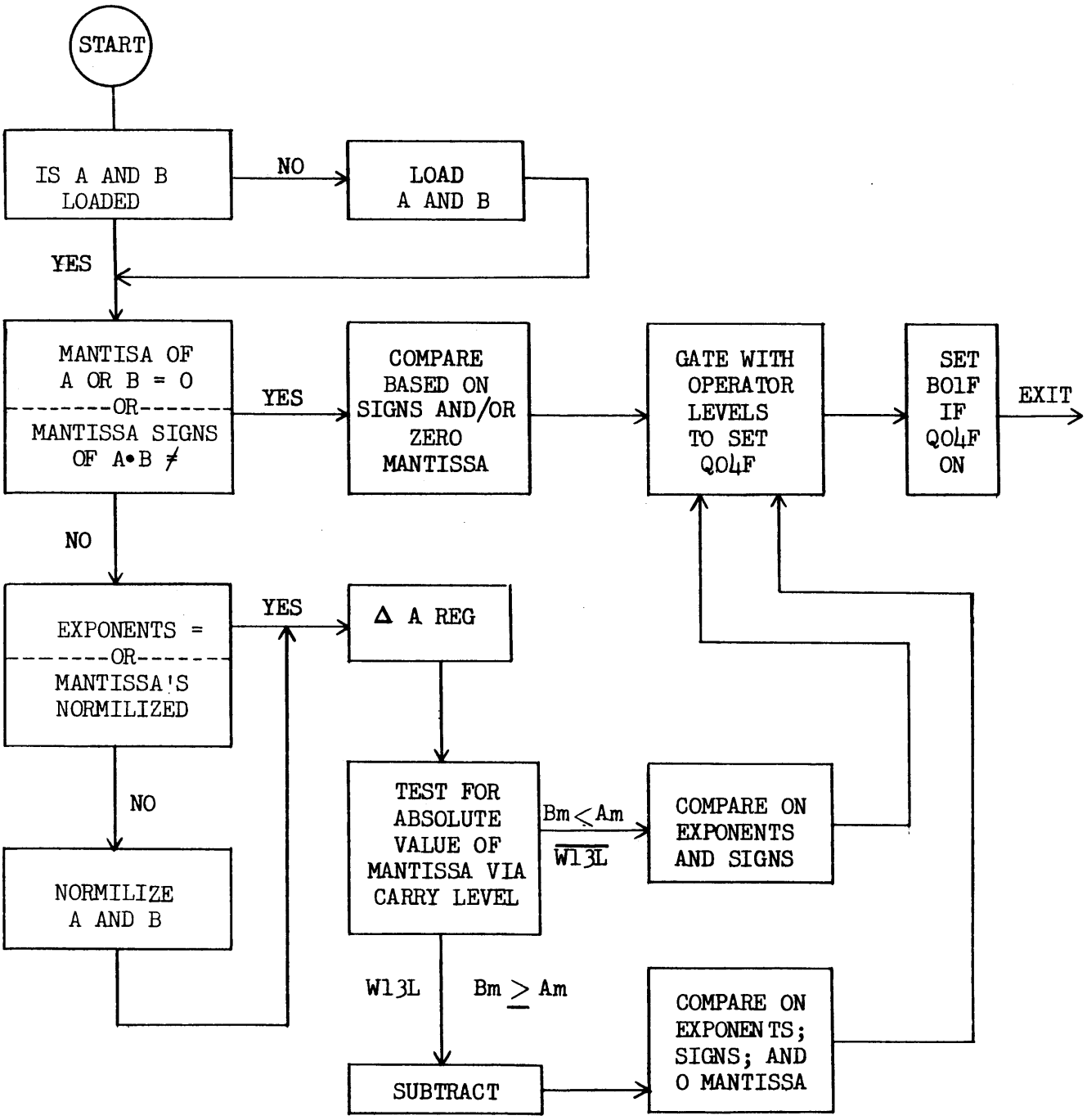


FIGURE 3.4-1 COMPARE OPERATOR LOGIC

If the carry level W13L is true, the B mantissa is either equal to or greater than the A mantissa. If W13L is false, the B mantissa is less than the A mantissa. Figure 3.4-2 illustrates in simple terms how the presence of a carry indicates the larger of the two mantissas.

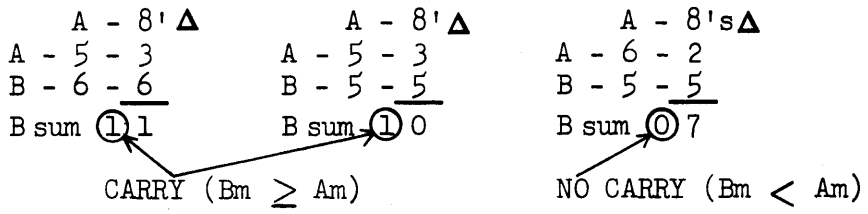


FIGURE 3.4-2. MANTISSA COMPARISON

If the carry level is false (W13L' is true), the B mantissa is less than the A mantissa. The comparison is then completed on the basis of the mantissa signs and algebraic exponent magnitude.

If the carry level is true, it is necessary to complete the subtract operation, after which the B mantissa may be interrogated to determine if the two mantissas were equal or not. Refer to Figure 3.4-2. The comparison is then completed on the basis of mantissa signs and the algebraic exponent magnitude.

Depending on the specific operator being utilized and the results of the comparison, Q04F is turned on if the condition is true or left off if the required conditions are false. Q04F is serving as temporary storage to indicate that the required comparison has been met. Subsequently, the state of Q04F is transferred to the least significant bit of the B register (B01F) to indicate the results of the operator. The operation is then terminated.

DETAILED DESCRIPTION

J = 0

If either the A or B register is unoccupied, a normal load from the stack occurs. All of the actions which follow require the A and B registers to be occupied (AROF • BROF).

Δ A to A, Q01F to 1

If the A and B exponents are equal (W73L true), or if both the mantissas are normalized, the A mantissa is complemented. Q01F is set to one, thus changing the 7's complement to an 8's complement, for the possible subtract operation to follow.

Q04F to 1

The setting of the logical toggle Q04F is a result of a compare operator ANDED with comparison logic. Q04F serves as temporary storage, until J = 6, at which time B01F is set. The following paragraphs explain each gate condition for setting Q04F.

BGEL + BGAL + BNEL

If the operator is either:
1. B greater than or equal to A
2. B greater than A
3. B not equal to A

If the following OR logic is true when:

W06L • W07L' • B47F' - A mantissa is zero and B is a positive number, therefore, B is greater than A.

W06L' • W07L • A47F - A is a negative number and the B mantissa is zero, therefore, B is greater than A.

W06L' • A47F • B47F' - A is a negative number and B is positive, therefore, B is greater than A.

BLEL + BLAL + BNEL

If the operator is either:
1. B less than or equal to A
2. B less than A
3. B not equal to A

If the following OR logic is true when:

W06L • W07L' • B47F - A mantissa is zero and B is a negative number, therefore, B is less than A.

W06L' • W07L • B47F' - A is a positive number and B mantissa is zero, therefore, B is less than A.

W06L' • A47F' • B47F - A is a positive number and B is negative, therefore, B is less than A.

BEQL + BGEL + BLEL

If the operator is either:
1. B equal to A
2. B greater than or equal to A
3. B less than or equal to A

If the following logic is true when:

W06L • W07L - A mantissa is zero and the B mantissa is zero, therefore, A and B are equal. (Negative and positive zeros are considered the same.)

B01F to 0, J to 6

If either of the mantissas are zero or the signs of the mantissa are opposite, the compare would have been complete via the previous logic. B01F is reset in case the compare is not met. The J register is set to six to allow the setting of B01F if the compare was met.

AROF • BROF • W06L' • W07L' • (B47F' • A47F' + B47F • A47F)

The above gating conditions define that both the A and B registers are not zero and that their signs of the mantissas are alike, which will allow the following action based on additional gating logic:

J to 2

With the above gating conditions and the exponents of A and B equal (W73L) or the mantissas of both A and B normalized, a possible subtract operation is required to complete the compare operation. Note that W73L refers to an algebraic equality; the signs of the exponents (46th bit) must be alike. This same logic caused the complement of A to be set into A.

J to 4

If the exponents of A and B are not equal (W73L') and either the A or B mantissa is not normalized, the A and/or B registers must be normalized before further comparison can be made.

J = 1

BROF to 1, S-1, J to 0

Entry to this J register setting would only occur if the registers required loading. Upon completion of a normal load, a return to J equals 0, allows the previously described logic to occur.

J = 2

Entry into this J register setting is required when a subtract (complement add) must be performed to decide the comparison.

Due to the carry propagation feature of the 42 bit mantissa adder, the presence or absence of a carry level from the 13th octade (W13L + W13L') is determined and available immediately, once the A and B registers are occupied. The state of this level immediately indicates which mantissa is greater than or equal to in absolute value.

B + A to B, J to 3

If W13L is true, the mantissa (the mantissa alone and not its signs or exponents) of B is greater than or equal to in absolute value than the mantissa of A. The subtract operation is performed, after which, at J = 3, the magnitude in the B mantissa is interrogated to determine if the two mantissas were equal.

B01F to 0, J to 6

If W13L' is true, the mantissa of B is less in absolute value than the mantissa of A. Therefore, the compare operation will be completed at this J register setting.

Q04F to 1

With W13L' true, the operators are ANDED with the mantissa signs and exponents to determine if the compare is true. In order to arrive at this point, the B mantissa is less than the A mantissa and the mantissa signs are alike.

BGEL + BGAL + BNEL

These operators check for B greater than A. The following logic ANDED with the operators determines a true comparison:

A47F' • W75L - A is a positive number and B is also a positive number (like signs) with the exponent of B greater than the exponent of A (W75L). Therefore, B is greater than A.

A47F • W74L - A is a negative number and the exponent of B is less than the exponent of A (W74L). The mantissa of B is less in absolute value than the mantissa of A (W13L'), that is, A contains a larger negative number than B. Therefore, B is greater than A.

A47F • W73L - A is a negative number and the exponent of A equals B (W73L). The mantissa of B less than the mantissa of A (W13L'), that is, A contains a larger negative number than B with exponents equal. Therefore, B is greater than A.

BLEL + BLAL + BNEL

These operators check for B less than A. The following logic ANDED with the operators determines the comparison:

W47F • W75L - A is a negative number and the exponent of B is greater than the exponent of A with the mantissa of B less than the mantissa of A. Both numbers are negative, with B more negative than A. Therefore, B is less than A.

A47F' • W74L - A is a positive number and the exponent of B is less than the exponent of A. Both numbers are positive with both the mantissa and exponent of B less than the mantissa and exponent of A. Therefore, B is less than A.

A47F' • W73L - A is a positive number and the exponents of A and B are equal. Both numbers are positive with equal exponents and the mantissa of B is less than the mantissa of A (W13L'). Therefore, B is less than A.

J = 3

Entry to this logical state is from the J = 2 box where the subtract (complement add) operation had been performed. The difference of this subtract operation is in the B register.

B01F to 0, J to 6

The compare will be completed in this box, therefore, B01F is set to zero. (The content of B is no longer valid after this clock pulse time.) Entry into this state requires that the signs of the mantissa be alike.

Q04F to 1

The operators are ANDED with the mantissa signs and exponent to determine the comparison. A true comparison sets Q04F to one.

(BEQL + BGEL + BLEL) • W73L • W07L

These operators are comparing for equality. With the mantissa of B equal to zero (W07L) and the exponents equal, B is equal to A. In Figure 3.4-2 it is illustrated how equal mantissas result in a zero B mantissa.

BGEL + BGAL + BNEL

These operators check for B greater than A. The following logic ANDED with the operators determines the comparison:

A47F' • W75L - The signs of both mantissas are positive (A47F' is required to arrive at this point) and the exponent of B is greater than the exponent of A. Both numbers are larger than zero and B the larger of the two, therefore, B is greater than A.

A47F • W74L - The signs of both mantissas are negative and the exponent of B is less than the exponent of A. Both numbers are less than zero with A more negative than B, therefore, B is greater than A.

A47F' • W73L • W07L' - The signs of both mantissas are positive, the exponents are equal and the mantissa of B is greater than A (W07L'). The B mantissa could only have been greater than or equal to the A mantissa to arrive at this point. Therefore, B is greater than A.

BLEL + BLAL + BNEL

These operators compare for B less than A. The following logic ANDED with the operators determines the comparison:

A47F • W75L - The signs of both mantissas are negative and the exponent of B is greater than the exponent of A. Both numbers are negative with B being the larger negative number. B, then, is less than A.

A47F' • W74L - The signs of both mantissas are positive and the exponent of B is less than the exponent of A. Both numbers are positive with A being the larger of the two. Therefore, B is less than A.

A47F • W73L • W07L' - The signs of both mantissas are negative, the exponents are equal and the mantissa of B is greater than the mantissa of A. The number in B is more negative than the number in A. Therefore, B is less than A.

J = 4

Entry into this J count can only be achieved from the J = 0 state with the exponents not equal (W73L'), neither mantissa normalized (B13L + A13L), equal mantissa signs and neither mantissa zero. In other words, A and/or B must be normalized before the comparison can continue.

NORMALIZE A

Shift the A register left by octades, count down the exponent of A and set the least significant octade of A to zero, until the A mantissa is normalized or the exponents are equal.

NORMALIZE B

Normalize the B register in the same manner as the A register.

J to 0

If both mantissas are normalized or if the exponents become equal, set the J register to zero to continue the compare operation.

J = 6

B to 0, AROF to 0, EXIT

Unconditionally clear the B register, mark the A register as empty and terminate the operation. SECL will be true at this clock pulse time.

B01F to 1

If the logical toggle Q04F is on, indicating that the compare is true, pulse the one side of B01F. Essentially this will cause the bit to complement as the complete B register is being set to zero at this clock pulse time. Because B01F is cleared at each compare logical box, this results in B01F going to the on state.

3.5 SYLLABLE BRANCH FORWARD UNCONDITIONAL — BFUL - 4231
SYLLABLE BRANCH BACKWARD UNCONDITIONAL — BBUL - 4131
WORD BRANCH FORWARD UNCONDITIONAL — JFUL - 6231
WORD BRANCH BACKWARD UNCONDITIONAL — JBUL - 6131

For the branch operators, the top of the stack specifies the cell or syllable to which branching occurs. Branching is either relative to the location of the branch operator or to an absolute address, as determined by the type of word in the top of the stack. An operand specifies the number of syllables or words to be jumped, either forward or backward; a descriptor specifies an address to which the branching occurs.

If the flag bit of the word in the A register is zero with a syllable branch operator, the C and L registers are:

1. Increased for branch forward by the 12 lower order bits of the word in the A register.

2. Decreased for branch backward by the 12 lower order bits of the word in the A register.

If the flag bit of the word in the A register is zero with a word branch operator, the L register is set to zero and the C register is:

1. Increased for branch forward by the 10 lower bits of the word in the A register.

2. Decreased for a branch backward by the 10 lower bits of the word in the A register.

If the flag bit and the presence bit of the word in the A register are both one, the 15 low order bits of the word in the A register are transferred to the C register and the L register is set to zero. In all cases, the A register is set to empty.

If the flag bit of the word in the A register is a one and the presence bit is a zero, the presence bit is set in the interrupt register if the Processor is in normal state. The contents of the A register are retained and the operation terminated.

SUMMARY OF OPERATION

If the A register is not loaded on entry, adjustment of the stack occurs to load the A register. When the A register is loaded, or if occupied on entry, the status of the A register is checked to determine whether it is a descriptor or an operand.

If the word in the A register is a descriptor, the presence bit is checked; and if the information is present, the A and B registers are interchanged, thus placing the descriptor in the B register. The contents of B[15=>1] are transferred to the C register and a fetch from this address is initiated. The original contents of the B register are restored from temporary storage in the A register.

If the presence bit of the descriptor in the A register is off (A46F'), no branch takes place and the operation is terminated. If the Processor is in normal state (NCSF) then the presence bit interrupt is set.

If the word in the A register is an operand and the operator is a syllable branch, the contents of the C register are transferred to the M register. The M register and the L register are then modified by the 12 least significant bits of the A register (A12 through A01). Therefore, the number of syllables that may be transferred (forward or reverse) is 0 through 4095. The modification of the M and L registers is accomplished by incrementing the L register by the 2 least significant bits of the A register and then using the 10 bit address adder to increment the M register by the next ten least significant bits of the A register (A12 through A03). The modified address in the M register is then placed in the B register, and subsequently transferred to the C register. A fetch is initiated to load the P register from the modified address and the operator terminates.

Syllable branching either forward or reverse is considered to be relative to the syllable following the branch operator syllable. Figure 3.5-1 illustrates branching forward and/or backwards 4 syllables from an assumed branch operator specified by L = 2.

BRANCH FORWARD 4 SYLLABLES

BRANCH BACKWARD 4 SYLLABLES

FIGURE 3.5-1 BRANCH 4 SYLLABLES

If the word in the A register is an operand and the operator is a word branch, the actions are similar to a syllable branch with the exception that the L register is set to zero, and the A register bits 10 thru 1 (only) are used to modify the C register contents. Therefore, syllable and word branches are the same with regard to total number of syllables that may be spanned. Word branching, either forward or reverse, is relative to the word which contains the branching instruction.

DETAILED DESCRIPTION

J = 0

If the A register is not occupied on entry, the A register is loaded via a normal stack adjustment, resulting in the A register being occupied and the B register

being unoccupied. If A and B are occupied on entry, no stack adjustment occurs. Fetch is inhibited by the term ICFL' being false, thus giving precedence to the loading of the A register. The term ICFL' is false for the duration of the operator.

ICFL = T57L • T12F

T57L • T12F = Branch Forward/Backward Unconditional

Program Word in A

Once the A register is occupied and the word in A is a descriptor (A48F), the presence bit is checked. If the information is present (A46F) in core memory, the A and B registers are interchanged; thus enabling the transfer of the core address at J = 5. The L register is set to zero, indicating a branch to the first syllable of the address contained in the descriptor. The A register is marked empty as it will no longer contain valid information once the B register is restored at J = 5. E16F is set to 1 to indicate that a new program word is needed in the P register and to initiate a Fetch operation.

Presence Interrupt to 1

If the program word addressed by the descriptor in the A register is not present in core memory (A46F'), the presence bit interrupt is set by the setting of U15F, U16F and U17F. The operator is terminated by the setting of J to 15.

U15F, U16F, U17F = NCSF • I08Z • US13X'
I08Z = A03L • T57L • T12F • A46F'
A03L = AROF • A48F • J = 0
T57L • T12F = Unconditional Branch
US13X' = Inhibit Interrupt Switch in D. D.

C to M

If the A register contains an operand (A48F'), the contents of the C register are transferred to the M register for subsequent modification by the operand in the A register.

J to 2

If the T register contains a syllable branch operator and the A register an operand, the J register is set to 2 for subsequent L register modification.

J to 6

If the T register contains a word branch and the A register an operand, the J register is set to 6. This will by-pass the L register modification and subsequent actions will set the L register to zero.

L - 1

The L register is counted down 1 to point at the branch instruction. The L register had been counted up by 1 at the time that SECL had transferred the branch instruction to the T register.

J = 1

This J count is part of the stack adjustment to load the A register.

J = 2

If the A register contains an operand, modification of the L and M registers occurs at this J count by the two least significant bits of the A register. In other words, the two least significant bits of the A register are used to set the L register to the correct syllable within an address word. The N register is used as a tally to gate the required clock pulses at this J count. The following paragraphs describe the occurrences at J = 2, relative to the setting of the N register and Branch Forward Unconditional.

N = 0, Branch Forward

The N register is equal to zero at the first clock pulse of J = 2. If A01F is set, the L register is incremented by one. If, in addition to A01F being on, the L register is equal to three, the M register is incremented by one to carry on into the next word. The A register is shifted right one binary bit and the N register counted up one.

N = 1, Branch Forward

At the second clock pulse of J = 2, the A01F bit has the binary value of 2 (it was shifted into the A01F position from the A02F position). Therefore, if the A01F bit is on at the second clock pulse time, the L register must then be incremented twice. The first incrementation occurs at this clock pulse, and the second incrementation occurs at the following clock pulse when the N register equals three. If the L register is equal to three when A01F is on, the M register is advanced by one to carry into the next word. The A register is not shifted at this clock pulse because the A01F bit position will have to be interrogated at the next clock pulse. A01F at this time has a value of 2.

N = 2, Branch Forward

At the third clock pulse of J = 2, the L register is counted up one if A01F is on. This accounts for the second count of the L register due to the original second bit position of the A register. If LE3L is true, the M register is counted up one. The A register is shifted right one binary bit position, thus placing the original third bit position of the A register into A01F.

N = 3, Branch Forward

At the fourth clock pulse of J = 2, the L register is counted up one by the logic N is equal to 3. The L register is incremented at this time because the L register was decremented at J = 0 of this operator. The reference for branch is the syllable following the Branch syllable. The L register is not allowed to count during fetch because the logic at fetch time does not provide for incrementing the M register. Refer to Figure 3.5-1 and note the zero reference point following the branch operator syllable. If the L register is equal to three, the M register is incremented to carry into the next word.

Q09F to 1, J to 3

At this fourth clock pulse, Q09F is set to enable the adder logic to function as an address adder. The J register is set to 3 where the M register is incremented via the address adder.

J = 2, Branch Reverse

The following paragraphs describe the occurrences at J = 2, relative to the setting of the N register and Branch Reverse Unconditional.

N = 0, Branch Reverse

The N register is equal to zero at the first clock pulse of J = 2. If A01F is on, the L register is decremented by one, provided the N register does not equal three (N03L'). If the L register equals zero, when A01F is set, the M register is also decremented to carry back into the previous word. The A register is shifted right one binary bit, to bring the second binary bit into the A01F position.

N = 1, Branch Reverse

At the second clock pulse of J = 2, the A01F bit has a binary value of 2. Therefore, the L register must be decremented twice if A01F is on. The first decrementation occurs at this clock pulse. If the L register is equal to zero and the N register is not equal to three, the M register is also counted down when A01F is on. No shift of the A register occurs at this clock pulse, since the A01F bit will be interrogated at the next clock pulse, because A01F has a value of 2 at this clock pulse.

N = 2, Branch Reverse

At the third clock pulse of J = 2, the L register is counted up for the second count if A01F is on. The M register is decremented if A01F is on, provided the N register does not equal three and the L register is equal to zero. This provides a carry to the prior word. The A register is shifted right one binary bit position, placing the original third bit position of the A register (A03F) into A01F.

N = 3, Branch Reverse

At the fourth clock pulse of J = 2, the L register is counted up one from the logic that N is equal to 3. The M register is also incremented if the L register is equal to three.

The L register, and possibly the M register, is incremented at this time because Inhibit Fetch at J = 0 prevented the normal count of L. The count of L is required because the reference for branch reverse is the syllable following the branch syllable. The L register was not allowed to count at fetch time because Fetch does not provide logic to carry into the M register.

N = 3, Forward or Reverse Branch

At the fourth clock pulse of J = 2, (N = 3) Q09F is set to one, enabling the parallel adder to function as an address adder. The J register is set to three where the add occurs.

BBUL

If the operator is Branch Backward Unconditional, the mantissa of the A register is complemented to provide for a complement add. Q01F is set to on for a pre-carry, converting the seven's complement to an eight's complement. Q03F is set to one (at J = 3) for decrementation logic of the high order bits of the M register.

J = 3

AR0F to 0, B to A

The A register is marked unoccupied, as the add occurs at this clock pulse time. Therefore, the contents of the A register are no longer required. The B register contents are transferred to the A register for temporary storage so that the B register can be used in the transfer to the A register of the modified address in the M register to the C register. Saving the B register content is only significant if the A and B registers were loaded on entry to the operator. Otherwise the B register contains non-valid information.

M plus A, J to 4

Add the 10 low order bits of the A register to the M register and place the sum in the M register. The J register is set to 4 to start the transfer of M to C.

Increment M15F through M11F

If Q03F is off, (indicating a branch forward operator), and a carry exists from the address adder (W10L), the M register, bits 15 through 11, are incremented. The incrementing of M15 through 11 provides for branching to the next higher segment of memory. This is in conjunction with the results of the add of bits 10 through 1.

Decrement M15F through M11F

If Q03F is on, indicating a branch reverse operation and a borrow occurred during the complement add (no carry; W01L'), the M register bits 15 through 11 are decremented. The decrementing of M15 through 11 provides for branching to the next lower segment of memory in conjunction with the results of the subtract of bits 0 through 1.

J = 4

The contents of the modified M register are transferred to the B register in preparation for the subsequent transfer to the C register. The J register is set to 5 to complete the transfer, and E16F is set to 1 to initiate a Fetch.

J = 5

Transfer B to C

The 15 least significant bits of the B register are transferred to the C register. The C register will now contain the modified address. A fetch was initiated by previously setting the E register to 16. The fetch will load the P register with the new program word.

Transfer A to B, Exit

The original contents of the B register, which were placed in the A register for temporary storage, are returned to the B register. The syllable execute complete level is true at this clock pulse time, terminating the operator.

L to 0

If the operator is a word branch (T11F) then set the L register to zero to indicate a branch to syllable zero of the word branching to.

J = 6

Q09F to 1, J to 3

At this time Q09F is set to enable the adder logic to function as an address adder. The J register is set to 3 where the M register is incremented via the address adder.

A[39=>1] to Δ A[39=>1], Q01F to 1, Q03F to 1

If the operator is a branch backwards words (JFBL), the mantissa of the A register is complemented to provide for a complement add. Q01F is set to one for a pre-carry, converting the seven's complement to an eight's complement. Q03F is set to one for decrementation logic of the high order bits of the M register at J = 3.

J = 15

If the word in the A register at J = 0 was a descriptor and if the information was not present in core memory, a presence interrupt would occur if in normal state. The syllable execute complete level is true at this clock pulse to terminate the operator.

3.6 SYLLABLE BRANCH FORWARD CONDITIONAL — BFCL - 0231
SYLLABLE BRANCH BACKWARD CONDITIONAL — BBCL - 0131
WORD BRANCH FORWARD CONDITIONAL — JFCL - 2231
WORD BRANCH BACKWARD CONDITIONAL — JBCL - 2131

For the branch operators, the top of the stack specifies the cell or syllable to which branching occurs. Branching is either relative to the location of the branch operator or to an absolute address, as determined by the type of word in the top of the stack. An operand specifies the number of syllables or words to be jumped, either forward or backward; a descriptor specifies an address to which branching occurs.

Conditional branch operators use the low-order bit of the second word in the stack as the true or false condition on which to branch. Conditional branches take place on a false condition.

If the low order bit of the word in the B register is a one, the A and B registers are set to empty and the operation terminated.

If the low order bit of the word in the B register is a zero and the flag bit of the word in the A register is a zero, the C and L registers are:

Syllable Branch

1. Increased for branch forward

2. Decreased for branch backward

By the 12 low order bits of the word in the A register.

Word Branch

1. For branch forward the L register is set to zero and the C register is increased

2. For branch backward the L register is set to zero and the C register is decreased

By the 10 low order bits of the word in the A register.

If the low order bit of the word in the B register is a zero and the flag bit and the presence bit of the word in the A register are both one, the 15 low order bits of the word in the A register are transferred to the C register and the L register is set to zero. In both of the cases, the A and B registers are set to empty.

If the low order bit of the word in the B register is a zero, the flag bit of the word in the A register is a one and the presence bit of the word in the A register is a zero, the presence bit is set in the interrupt register if the Processor is in normal state. The contents of the A and B registers are retained and the operation is terminated.

SUMMARY OF OPERATION

If either the A or B register is not loaded on entry, stack adjustment occurs to load the A and B registers. When the A and B registers are loaded, or if occupied on entry, the status of the least significant bit of the B register (B01F) is checked to determine whether a branch is called for. If B01F is on, a true condition is indicated and no branch occurs. Therefore, the A and B registers are marked unoccupied and the operator terminated.

If a false condition of B01F is indicated, the A register is checked for a descriptor or an operand. If the A register contains a descriptor, the presence bit is checked. If the information is present, the A register is transferred to the B register. Subsequently, the contents of B15 through B1 are transferred to the C register and a fetch initiated. If the information is not present, a presence interrupt condition is set in normal state and the operator terminated.

If B01F is false and the A register contains an operand, the contents of the C register is transferred to the M register. If the operator is a syllable branch, the M register and the L register are then modified by the 12 least significant bits of the A register (A12 through A01). Therefore, the number of syllables that may be transferred (forward or reverse) is 0 through 4095. The modification of the M and L registers is accomplished by incrementing the L register by the 2 least significant bits of the A register and then using the 10 bit address adder to increment the M register by the next ten least significant bits of the A register (A12 through A03). The modified address in the M register is then placed in the B register and, subsequently, transferred to the C register. A fetch is initiated to load the P register from the modified address and the operator terminated. If the operator is a word branch, then the L register is set to zero and the 10 low order bit of the A register (only) are used to modify the C register contents.

DETAILED DESCRIPTION

J = 0

Normal stack adjustment occurs on entry if either the A or B register or both are empty. Fetch is inhibited by the term ICFL' being false, giving precedence to the loading of the A and B registers.

ICFL = Q25L • T57L
T57L = Branch Forward/Backward Conditional
Q25L = (J = 0) • AROF' + BROF'

No Branch

Once the A and B registers are loaded, the least significant bit of the B register is checked. If B01F is on, no branching occurs. The A and B registers are marked unoccupied and the J register is set to 15 to terminate the operator. Fetch is allowed at this clock pulse time, because B01F is on (ICFL' goes true).

ICFL = T57L • Q21L • B01F'
T57L = Branch Forward/Backward Conditional
Q21L = (J = 0) • AROF • BROF

Branch - Program Word in A

If B01F is reset, branching will occur. Therefore, Fetch is inhibited by ICFL' being false. If the word in the A register is a descriptor (A48F) and the information is present (A46F), the A register contents are transferred to the B register; this enables the transfer of the core address to the C register at J = 5. The A register is marked unoccupied and the L register is set to zero. E16F is set to one to initiate the Fetch of a new program word to the P register.

Presence Interrupt to 1

If the program word, addressed by the descriptor in the A register, is not present in core memory (A46F'), the presence bit interrupt is set by the setting of U15F, U16F and U17F. The operator is terminated by the setting of J to fifteen.

U15F, U16F, U17F = NCSF • I08Z • US13X'
I08Z = A03L • T57L • B01F' • A46F' • BROF
A03L = AROF • A48F • (J = 0)
T57L = Conditional Branch
US13X' = Inhibit Interrupt Switch in D.D.

C to M

If the A register contains an operand (A48F'), the contents of the C register are transferred to the M register for subsequent modification by the operand in the A register.

J to 2

If the T register contains a syllable branch operator (A48F' • T11F') and the A register an operand, the J register is set to 2 for subsequent L register modification.

J to 6

If the T register contains a word branch (A48F' • T11F) and the A register an operand, the J register is set to 6 and the L register will subsequently be set to zero.

L - 1

The L register, having been counted up during SECL time when transferring the branch instruction of the T register, is counted down 1 to point to the branch instruction in preparation for modifying the L register.

J = 1

This J count is part of the stack adjustment.

J = 2

If the A register contains an operand, modification of the L and M register occurs at this J count by the two least significant bits of the A register. In other words, the two least significant bits of the A register are used to set the L register to the correct syllable within an address word. The N register is used as a tally to gate the required clock pulses at this J count. The following paragraphs describe the occurrences at J = 2 relative to the setting of the N register and Branch Forward Conditional.

N = 0, Branch Forward

The N register is equal to zero at the first clock pulse of J = 2. If A01F is set, the L register is incremented by one. If, in addition to A01F being on, the L register is equal to three, the M register is incremented by one to carry on into the next word. The A register is shifted right one binary bit and the N register counted up one.

N = 1, Branch Forward

At the second clock pulse of J = 2, the A01F bit has the binary value of 2 (it was shifted into the A01F position from the A02F position). Therefore, if the A01F bit is on at the second clock pulse time, the L register must be incremented twice. The first incrementation occurs at this clock pulse, and the second incrementation occurs at the following clock pulse when the N register equals three. If the L register is equal to three when A01F is on, the M register is advanced by one to carry into the next word. The A register is not shifted at this clock pulse because the A01F bit position will have to be interrogated at the next clock pulse and A01F has a value of 2.

N = 2, Branch Forward

At the third clock pulse of J = 2 the L register is counted up one, if A01F is on. This accounts for the second count of the L register due to the original second bit position of the A register. Again, if LE3L is true, the M register is counted up one. The A register is shifted right one binary bit position, placing the original third bit position of the A register into A01F.

N = 3, Branch Forward

At the fourth clock pulse of J = 2, the L register is counted up by the logic N is equal to 3. The L register is incremented at this time because the L register was decremented at J = 0 of this operator. The reference for branch is the syllable following the Branch syllable. The L register is decremented at J = 0 because the logic at fetch time does not provide for incrementing the M register. If the L register is equal to three, the M register is incremented to carry into the next word.

Q09F to 1, J to 3

At this fourth clock pulse, Q09F is set to enable the adder logic to function as an address adder. The J register is set to 3 where the M register is incremented via the address adder.

J = 2, Branch Backward

The following paragraphs describe the occurrences at J = 2, relative to the setting of the N register and Branch Backward Conditional.

N = 0, Branch Backward

The N register is equal to zero at the first clock pulse of J = 2. If A01F is on, the L register is decremented by one provided the N register does not equal three (N03L'). If the L register equals zero when A01F is set, the M register is also decremented to carry back into the previous word. The A register is shifted right one binary bit, to bring the second binary bit into the A01F position.

N = 1, Branch Backward

At the second clock pulse of J = 2, the A01F bit has a binary value of 2. Therefore, the L register must be decremented twice if A01F is on. The first decrementation occurs at this clock pulse. If the L register is equal to zero and the N register is not equal to three, the M register is also counted down when A01F is on. No shift of the A register occurs at this clock pulse since the A01F bit will be interrogated at the next clock pulse because A01F has a value of 2 at this clock pulse.

N = 2, Branch Backward

At the third clock pulse of J = 2, the L register is counted up for the second count if A01F is on. The M register is decremented if A01F is on, provided the N register does not equal three and the L register is equal to zero. This provides a carry to the prior word. The A register is shifted right one binary bit position placing the original third bit position of the A register (A03F) into A01F.

N = 3, Branch Backward

At the fourth clock pulse of J = 2, the L register is counted up one from the logic that N is equal to 3. The M register is also decremented if the L register is equal to zero.

The L register and possibly the M register is incremented at this time because Inhibit Fetch at J = 0 prevented the normal count of L. The count of L is required because the reference for branch reverse is the syllable following the branch syllable. The L register was not allowed to count at fetch time because Fetch does not provide logic to carry into the M register.

N = 3, Forward or Backward Branch

At the fourth clock pulse of J = 2 (N = 3), Q09F is set to one enabling the parallel adder to function as an address adder. The J register is set to three where the add occurs.

BBCL

If the operator is Branch Backward Conditional, the mantissa of the A register is complemented providing for a complement add. Q01F is set on for a pre-carry converting the seven's complement to an eight's complement. Q03F is set to one for decrementation logic (at J = 3) of the high order bits of the M register.

J = 3

AROF to 0, B to A

The A register is marked unoccupied as the add occurs at this clock pulse time. Therefore, the contents of the A register are no longer required.

M plus A, J to 4

Add the 10 low order bits of the A register to the M register and place the sum in the M register. The J register is set to 4 to start the transfer of the M register contents to the C register.

Increment M15F through M11F

If Q03F is off, indicating a branch forward operator and a carry exists from the address adder (W10L), the M register bits 15 through 11 are incremented. The incrementing of M14 through 11 provides for branching to the next higher segment of memory in conjunction with the results of the add of bits 10 through 1.

Decrement M15F through M11F

If Q03F is on, indicating a branch reverse operation and a borrow occurred during the complement add (no carry; W10F'), the M register bits 15 through 11 are decremented. The decrementing of M15 through 11 provides for branching to the next lower segment of memory in conjunction with the results of the subtract of bits 10 through 1.

J = 4

The contents of the modified M register are transferred to the B register in preparation for the subsequent transfer to the C register. The J register is set to 5 to complete the transfer and E16F is set to initiate a Fetch.

J = 5

Transfer B to C

The 15 least significant bits of the B register are transferred to the C register. The C register will now contain the modified address. The fetch will load the P register with the new program word. The B register is marked unoccupied. The syllable Execute Complete Level is true at this clock pulse time, terminating the operator.

L to 0

If the operator is a word branch (T11F), then set the L register to zero to indicate a branch to syllable zero of the word branching to.

J = 6

Q09F to 1, J to 3

Q09F is set to enable the adder logic to function as an address adder. The J register is set to 3 where the M register is incremented via the address adder.

Δ A[39=>1], Q01F to 1, Q03F to 1

If the operator is a branch backward words (JBCL), the mantissa of the A register is complemented to provide for a complement add. Q01F is set for a pre-carry, converting the seven's complement to the eight's complement. Q03F is set to 1 for decrementation of the high order bits of the M register at J = 3.

J = 15

If the word in the A register at J = 0 was a descriptor and the information was not present in core memory, a presence interrupt would occur if in normal state. The syllable execute complete level is true at this clock pulse to terminate the operator

3.7 BRANCH RETURN OPERATOR RJPL-0135

For the branch return operator, the top of the stack specifies the word to which branching occurs. If the presence bit of the word in the A register is zero, the presence bit is set in the interrupt register and the operation is terminated.

If the presence bit of the word in the A register is one, the operation is continued as follows: The S register is set to the contents of bit positions 30 through 16 of the word in the A register. The C register is set to the contents of bit positions 15 through 1 of the word in the A register. The L register is set to zero.

The contents of the cell addressed by the S register, the mark stack control word, is read from memory. The R and F registers are set to the contents of their respective fields of the mark stack control word. The mark stack flip-flop and the program level flip-flop are set to the contents of their respective positions of the mark stack control word. The S register is decreased by one. The A and B registers are set to empty.

SUMMARY OF OPERATION

Upon entry into the branch return operator, the top word of the stack is placed in the B register if not already there. The presence bit of the descriptor in the B register is checked and if not present, a presence bit interrupt is set. If however, the information is present (presence bit on), the address of the mark stack control word is transferred from the B register to the S register. A memory cycle is initiated to load the mark stack control word into the B register. From the descriptor in the B register, the address of the word, to which branching occurs, is transferred to the C register. A fetch is initiated to load the program word into the P register. The syllable control register (L), the mark stack flip-flop and sublevel flip-flop are cleared.

When the mark stack control word in the B register is transferred to the F and R registers, the mark stack flip-flop and the sublevel flip-flop, the operator is terminated and the next syllable to be executed is the first syllable of the new program word previously fetched.

DETAILED DESCRIPTION

J = 0

Unconditionally inhibit fetch count up. Mark the B register as occupied and the A register as empty. If the A register is loaded on entry, the contents of the A register is transferred to the B register.

E to 3, Q03F to 1

If both the A and B registers are empty, a load of the B register is initiated by setting the E register to 3. The logical flip-flop, Q03F, is set to one to indicate that the stack address must be adjusted prior to terminating the operator, if a presence interrupt is set. The S register must be counted down, on a presence interrupt, because "Storage For Interrupt" logic counts the S register up, if the B register contains valid information, (BROF).

Set Presence Bit Interrupt

If the presence bit is off (B46F'), the presence bit flip-flops, U15F, U16F and U17F, are set. The J register is set to 3 to terminate the operator.

U15F, U16F, U17F = NCSF • I08Z • US13X'
I08Z = Q23L • RJPL • B46F' • EEZL
Q23L = AROF' • BROF • (J = 0)
US13X' = Inhibit Interrupt Switch in D.D.

B30 through 16F to S

If the presence bit is on (B46F), the address of the mark stack control word in the B register (bits 30 through 16), is transferred to the S register. The J register is set to 1 and a memory access is initiated to load the mark stack control word, by setting E to 3.

J = 1

The contents of the B register (bits 15 through 1) are transferred to the C register and a load of the P register is initiated by the setting of E to 16. The L register is cleared, thus pointing at the first syllable of the branched to program word. The mark stack flip-flop and the sublevel flip-flop are cleared in preparation for their new status, from the mark stack control word, at J = 2.

J = 2

Upon completion of the memory access, which loaded the mark stack control word into the B register, the entry stack location is restored to the F register from the B register (bits 30 through 16). The base address of the Program Reference Table is restored to the R register from the B register (bits 42 through 34).

The S register is decremented so it will point to the word prior to the mark stack control word. The B register is marked empty and the syllable execute complete level is true, to terminate the operator.

SALF and MSFF to 1

If the sublevel bit is on (B31F), the sublevel flip-flop (SALF) is set. If the mark stack bit is on (B32F), the mark stack flip-flop is set. The mark stack flip-flop is equivalent to Q12F.

SALF = 909L • B31F • RJPL
909L = EEZL • J02L

Q12F = Q10I
Q10I = 909L • B32F • RJPL

J = 3

Entry into this J count occurs only if the presence bit was not on in the initial word in the B register. Upon entry to the operator the B register may have been occupied or empty. If the B register had been empty, it would have been loaded and the logical flip-flop Q03F, turned on. Q03F is turned on to indicate that the loading of the B register occurred without a corresponding decrement of the S register. If, however, the B register was loaded on entry to the operator, the S register would have already been decremented and Q03F would not be turned on.

The syllable execute complete level is true at the last pulse of the operator. If Q03F is on, the stack address is decremented.

3.8 STORE DESTRUCTIVE / STORE NON-DESTRUCTIVE

BSDL-0421
BSNL-1021

The store operators operate on the two top words in the stack. The top word in the stack specifies the cell into which the second word in the stack is stored. The destructive store operators remove both the address and the information stored from the stack. The non-destructive store operators remove only the address from the stack.

If the VARF Flip-flop is set, the Processor is set to sub-program level after the relative address operation and VARF is reset.

If the flag bit and the presence bit of the word in the A register are both one, the contents of the B register are stored in the memory cell addressed by the 15 low order bits of the A register. The A and B registers are set to empty for store destructive and only the A register set to empty for store non-destructive.

If the flag bit of the word in the A register is one and the presence bit is zero, the presence bit code is set in the interrupt register if NCSF is set and the operation terminated.

If the flag bit of the word in the A register is zero, the ten low order bits of the word in the A register are used as a relative address, except that no addressing relative to the C register takes place. If the syllable calls for addressing relative to the C register, the absolute address is constructed relative to the R register instead. The contents of the B register are stored in the memory cell addressed after appropriate indexing of the relative address. The A and B registers are set to empty for store destructive and only the A register set to empty for non-destructive.

SUMMARY OF OPERATION

If the word in the A register is a descriptor, the presence bit is checked to determine if the storage area is present in core memory. If the presence bit is on, the B register contents are stored in the cell addressed by the 15 low order bits of the descriptor in the A register. If the presence bit is off, the storage area is not in core memory and a presence interrupt is set.

If the word in the A register is an operand, the status of the sub-level flip-flop, mark stack flip-flop and the A register bits (10, 9 and 8) determine whether the storage address is to be developed by indexing R relative or F relative. The following chart specifies the flip-flop and bit combination required for the indexing of a relative address.

SALF	BIT 10	BIT 9	BIT 8	MSFF	ABSOLUTE ADDRESS	
					Base	Relative Address
0	-	-	-	-	R+	Bits 10 thru 1
1	0	-	-	-	R+	Bits 9 thru 1
1	1	0	-	0	F+	Bits 8 thru 1
1	1	0	-	1	(R+7) +	Bits 8 thru 1
1	1	1	0	-	* C+	Bits 7 thru 1
1	1	1	1	0	F-	Bits 7 thru 1
1	1	1	1	1	(R+7) F-	Bits 7 thru 1

* Forced to R relative for Store Operator

When indexing is F relative, either the present F register contents are utilized as the base or the F register address contained in the mark stack control word at location R+7 is utilized as the base. If the F register address contained in the mark stack control word is utilized it must first be accessed and brought to the M register before the final storage address may be indexed.

The R register is a nine bit register whose bits occupy high order significance in a 15-bit address. Therefore, an R register address is specifying a block area of 64 words. The 6 low order bits of a 15-bit address, specify which one of the 64 cells (in the block) is being addressed.

DETAILED DESCRIPTION

If either the A or B register is empty, a push up from the stack loads the A and/or B registers. Once the A and B registers are loaded or if loaded on entry to the operator, the flag and presence bits are interrogated.

Presence Interrupt

If the word in the A register is a descriptor (A48F) and the storage area is not present in core memory (A46F'), a presence interrupt is set by turning on U15F, U16F and U17F. The J register is set to 15 to terminate the operation.

U15F, U16F, U17F = NCSF • I08Z • US13X'
I08Z = A03L • C04L • BROF • T11F' • A46F'
A03L = AROF • (J = 0) • A48F
C04L • T11F' = Store Operators
US13X' = Inhibit Interrupt SW in D.D.

Descriptor is Present

If the word in the A register is a descriptor and the storage area is present in core memory (A46F), the low order bits of the A register (15 through 1) are transferred to the M register. A store of the B register contents is initiated by the setting of the E register to 13. The J register is set to 5 for unconditional clearing of the A register. The B register is cleared if the operator is store destructive.

AROF • BROF • A48F' • SALF • A10F

The following paragraphs apply if the A register contains an operand with the sub-level flip-flop and A10F both on.

A10F and A09F to 0

If the index operation is F+ relative (SALF • A10F • A09F'), A10F and A09F are reset, limiting indexing to one of the 225 locations above the address contained in the F register. If the index operation is decoded as C relative (SALF • A10F • A09F), it is forced to R relative for the store operator. A10F and A09F are reset, limiting the indexing to one of the 127 locations above the address contained in the R register.

F to M

If an F relative index is based on the present contents of the F register (MSFF' • A09F' + MSFF' • A08F), the contents of the F register are transferred to the M register, establishing the F relative base in the M register.

7 to M [3 ⇒ 1], E to 6

If an F relative index is based on the F register contents presently stored in the mark stack control word (MSFF • A09F' + MSFF • A08F) at location R + 7, the three low order bits of the M register are set to 7. The high order bits of the M register (bits 5 through 7) were set to the contents of the R register at SECL time when this operator was set into the T register. Thus, the M register contains the address of the mark stack control word. A memory load cycle is initiated by setting E to 6, which transfers bits 30 through 16 of the mark stack control word to the M register. Thus, after the memory cycle, the M register will contain the F register setting stored at location R + 7.

Δ A [7 ⇒ 1] to A [7 ⇒ 1], Q01F and Q03F to 1

If the index operation is F- relative (SALF • A10F • A09F • A08F), the A register bits 7 through 1 are complemented for the ensuing subtract operation. Q01F is set to convert the 7's complement to an 8's complement. Q03F is set to remember that a subtract operation is to take place. Because seven bits are involved in the subtraction, the indexed address may be up to 127 locations below the F register.

J to 4

If the word in the A register is an operand (A48F'), the J register is set to 4 for either the addition or subtraction.

SALF to 1, VARF to 1

If the preceding operator was VARL, then SALF was reset to allow R+ of 1023 words. VARF was set to remember that SALF was reset. With AROF • BROF • VARF restore SALF to 1 and reset VARF.

Q09F to 1

If the top word in the stack is an operand (AROF • BROF • A48F'), then set Q09F to convert the parallel adder logics to a 10 bit address adder.

M + A to M, E to 13, J to 5

If the operation is R+ relative (AROF • BROF • A48F' • (SALF' + A10F'), then add M to A with the result in M. The M register had been set to the R register contents during the SECL time that transferred this operator to the T register. The E register is set to 13, thus initiating a store of the B register contents at the indexed address in the M register. The J register is set to 5 in preparation of termination of this operator.

M [15 ⇒ 11] + 1

If during the add a carry from the 10's position of the adder is indicated (W10L), a one is added to the high order bit positions of the M register.

J = 1

This J register setting is for the initial loading of the A and/or B registers, if not loaded on entry.

J = 4

If the desired F register setting is in R + 7, the addition of the A register to the M register must wait until E is equal to zero (M is loaded with the F contents). If, however, indexing is relative to the R register or to the present F register contents, E is equal to zero on entry to J = 4. If Q03F is off (arithmetic operation is addition) and a carry from the 10's position of the adder is indicated (W10L), a one is added to the high order bits of the M register. If Q03F is on (arithmetic operation is subtraction via complement addition) and a carry from the 10's position is not developed (W10L'), a one is subtracted from the high order bits of the M register. The absence of a carry indicates a larger number was

subtracted from a smaller number. Therefore, an artificial borrow is required to address the next lower segment of memory.

The E register is set to 13, thus initiating a store of the B register contents at the indexed address in the M register. The J register is set to 5 for termination of the operation.

J = 5

Upon completion of the memory write cycle (EEZL), the A register is marked empty and the syllable execute complete level (SECL) is true to terminate the operator. If the operator is store destructive, the B register is also marked empty.

3.9 INTEGER STORE DESTRUCTIVE — ISDL-4121
INTEGER STORE NON-DESTRUCTIVE — ISNL-4221
CONDITIONAL INTEGER STORE DESTRUCTIVE — CSDL-0121
CONDITIONAL INTEGER STORE NON-DESTRUCTIVE — CSNL-0221

The store operators operate on the two top words in the stack. The top word in the stack specifies the cell into which the second word in the stack is stored. The destructive store operators remove both the address and the information stored from the stack. The non-destructive store operators remove only the address from the stack. The A and B registers are set to empty for store destructive. Only the A register is set to empty for store non-destructive.

Integer Store operator or Conditional Integer Store (if the integer bit of the word in the A register is a one): If the flag bit and the presence bit of the word in the A register are both one, or if the flag bit of the word in the A register is a zero, the word in the B register is made an integer as follows:

If the exponent is zero, the operand is not changed.

If the exponent is positive and non-zero, the operand is normalized and the exponent reduced. If the exponent is not reduced to zero as a result of normalizing, the integer overflow interrupt bit is set.

If the exponent is negative and non-zero, the operand is shifted to the right until the exponent equals zero. The mantissa is rounded according to the following rules:

a. For positive operands, the mantissa is increased in magnitude by one if the fractional part of the operand (i.e., the digits shifted out of the register) was greater than or equal to one-half.

b. For negative operands, the mantissa is increased in magnitude by one if the fractional part of the operand was greater in magnitude than one-half.

If an integer overflow occurs, the integer overflow bit is set in the interrupt register, the contents of the A and B registers is retained and the operation is terminated.

If integer overflow does not occur or if the operator is Conditional Integer Store with the integer bit of the word in the A register off and if the flag bit and the presence bit of the word in the A register are both one, the contents of the B register are stored in the memory cell addressed by the 15 low order bits of the A register. The A and B registers are set to empty.

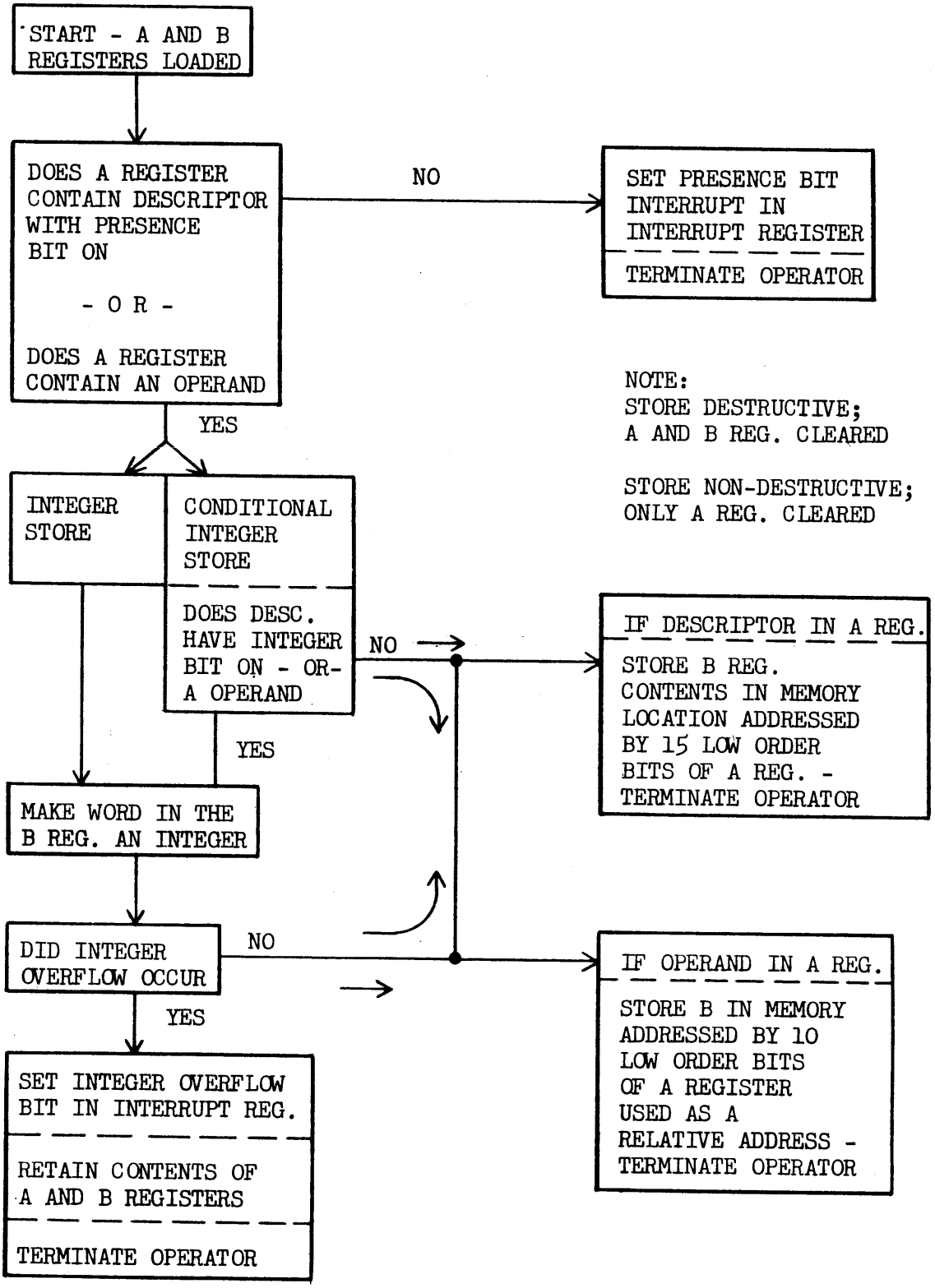


FIGURE 3.9-1. INTEGER/CONDITIONAL INTEGER STORE - DESTRUCTIVE/NON-DESTRUCTIVE

If the flag bit of the word in the A register is one and the presence bit is zero, the presence bit is set in the interrupt register and the operation terminates.

If the flag bit of the word in the A register is zero, the ten low order bits of the word in the A register are used as a relative address, except that no addressing relative to the C register takes place. If the syllable calls for addressing relative to the C register, the absolute address is constructed relative to the R register instead. The contents of the B register are stored in the memory cell addressed after appropriate indexing of the relative address.

SUMMARY OF OPERATION

Refer to Figure 3.9-1 for a flow diagram of the basic logic utilized in the Integer Store operators. If the A register contains a descriptor, its presence bit is checked. If the presence bit is off, a presence bit interrupt is set into the interrupt register and the operator is terminated. (The contents of the A and B registers are retained in the A and B registers). If the presence bit of the descriptor in the A register is on, the address contained in the 15 low order bits of the A register is utilized to store the contents of the B register.

If the contents of the A register are an operand, the contents of the B register are stored at a relative address. The status of the sub-level flip-flop, mark stack flip-flop and the A register bits (10, 9 and 8) determine whether the storage address is to be developed by indexing R relative or F relative. The following chart specifies the flip-flop and A register bit combinations required for the indexing of a relative address.

SALF	BIT 10	BIT 9	BIT 8	MSFF	ABSOLUTE ADDRESS	
					Base	Relative Address
0	-	-	-	-	R+	Bits 10 thru 1
1	0	-	-	-	R+	Bits 9 thru 1
1	1	0	-	0	F+	Bits 8 thru 1
1	1	0	-	1	(R+7) +	Bits 8 thru 1
1	1	1	0	-	* C+	Bits 7 thru 1
1	1	1	1	0	F-	Bits 7 thru 1
1	1	1	1	1	(R+7) -	Bits 7 thru 1

* Forced to R relative for Store Operator.

When indexing is F relative, either the present F register contents are utilized as the base or the F register address contained in the mark stack control word at location R+7 is utilized as the base. If the F register address contained in the mark stack control word is utilized, it must first be accessed and brought to the M register before the final storage address may be indexed.

If the operator is:

1. An Integer Store operator,
2. A Conditional Integer Store operator and the A register contains a descriptor with its Integer bit (A29F) on,
3. A Conditional Integer Store operator and the A register contains an operand,

the contents of the B register is made into type integer (if not already an integer) prior to the storage operation.

If the operator is a Conditional Integer Store operator and the A register contains a descriptor with its Integer bit off, the contents of the B register is stored without any modifications.

DETAILED DESCRIPTION

J = 0

The occurrences at the J register equals zero setting is applicable to all of the Integer Store Operators. The occurrence above the first double line functions to load the A and B registers, if not already loaded, on entry. The occurrences between the first and second double lines are applicable to descriptors only. The occurrences below the second double line are applicable to operands only.

Handling Descriptors

If the A and B registers are loaded, the flag bit of the word in the A register is checked. If the flag bit (A48F) is on, the word in the A register is a descriptor. The address described by the descriptor (in bits 1 through 15) is shifted to the M register. At the same time, the presence bit is checked. If the presence bit (A46F) is off, the storage area addressed by the descriptor is not presently contained in core memory. Therefore, a presence bit interrupt is set into the interrupt register by turning on U15F, U16F and U17F. If the presence bit is off, the J register is set to 15 to terminate the operator.

If the presence bit is on, the J register is set to 4 to continue the store operator. The logical flip-flop Q06F is turned on to remember the fact that the A register contained a descriptor. Q06F being on, causes the relative addressing index logic to be non-effective.

Handling Operands

If the flag bit (A48F) is off, the word in the A register is an operand and a relative address (as contrasted to the absolute address used with descriptors) is developed in the M register.

F to M

If SALF and A10F are on with the mark stack flip-flop (MSFF') off, and either A09F' is off or A08F is on, the storage address will be relative to the address contained in the F register. As a result, the contents of the F register are transferred to the M register. In other words, the contents of the B register will be stored relative to either the address of a return control word or a mark stack control word.

7 to M [3 ⇒ 1], E to 6

If an F relative index is based on the F register contents, presently stored in the mark stack control word (MSFF • A09F' + MSFF • A08F) at location R + 7, the three low order bits of the M register are set to 7. The high order bits of the M register (bits 15 through 7) were set to the contents of the R register at the previous SECL time and the M register will contain the address of the mark stack control word. A memory load cycle is initiated by setting E to 6, which transfers bits 30 through 16 of the mark stack control word to the M register. After the memory cycle, the M register will contain the F register setting stored at location R + 7.

A10F and A09F to 0

If the index operation is F+ relative (SALF • A10F • A09F'), A10F and A09F are reset, thus limiting indexing to one of the 225 locations above the address contained in the F register. If the index operation is decoded as C relative (SALF • A10F • A09F), it is forced to R relative for the store operator. A10F and A09F are reset, thus limiting the indexing to one of the 127 locations above the address contained in the R register.

Δ A [7 ⇒ 1] to A [7 ⇒ 1], Q01F and Q03F to 1

If the index operation is F- relative (SALF • A10F • A09F • A08F), the A register bits 7 through 1 are complemented for the ensuing subtract operation. Q01F is set to convert the 7's complement to an 8's complement. Q03F is set to remember that a subtract operation is to take place. Because seven bits are involved in the subtraction, the indexed address may be up to 127 locations below the F register.

Q09F to 1

Q09F is set, to convert the parallel adder logics to a 10 bit address adder.

J to 4

If the operation is F relative (SALF • A10F) or the B register contents is not an integer (W72L') and the A register contains an operand (A48F'), then set the J register to 4 for either an addition or subtraction and prepare to make B an integer.

SALF to 1, VARF to 0

If the preceding operator was VARL, then SALF was reset to allow R+ of 1023 words. VARF was set to remember that SALF was reset. Restore SALF to 1 and reset VARF.

A + M to M, E to 13, J to 5

If the operator is R+ relative (AROF • BROF • A48F' • [SALF + A10F']) and the word in the B register is an integer (W72L), then add M + A with the result in the M register. The M register had been set to the contents of the R register during the previous SECL time. The E register is set to 13 to initiate the store of the B register contents and the J register is set to 5 to terminate this operator.

M [15 ⇒ 11] + 1

If during the add a carry from the 10's position of the adder is indicated (W10L), a one is added to the high order bit position of the M register.

J = 1

This J register setting is for the initial loading of the A and/or B registers if not loaded on entry.

J = 4 • EEZL

If the logical flip-flop Q06F is off, the A register contains an operand. Therefore, the indexing logic between the first and second double lines is allowed to occur.

If the desired F register setting is in R + 7, the addition of the A register to the M register must wait until E is equal to zero (M becomes loaded with the F contents). If, however, indexing is relative to the R register or to the present F register contents, E is equal to zero on entry to J = 4. If Q03F is off (arithmetic operation is addition) and a carry from the 10's position of the adder is indicated (W10L), a one is added to the high order bits of the M register. If Q03F is on (arithmetic operation is subtraction via complement addition) and a carry from the 10's position is not developed (W10L'), a one is subtracted from the high order bits of the M register. The absence of a carry indicates a larger number was subtracted from a smaller number and, therefore, an artificial borrow is required to address the next lower segment of memory.

J = 4 B Ready for Storage

The word in the B register is ready for storage if its exponent is zero (BEXL) or if the operator is a Conditional Integer Store and the A register contains a descriptor with its Integer bit reset (A29F'). The level BEXL is equivalent to W72L in the logic book. The word in the B register is stored at the address specified by the M register via the setting of E to 13. The J register is set to 5 to terminate the operation.

J = 4 B Not Ready for Storage

The word in the B register is not ready to be stored if the following conditions exist: BEXL' • (A29F + ISDL + ISNL). This logic indicates that the exponent of the word in the B register is not zero and either:

1. An Integer is required (A29F)
2. The operator is an Integer Store operator (ISDL or ISNL)

If these conditions exist, the sign of the exponent and the contents of the mantissa are checked to determine what further manipulations of the B register are required.

J to 6

The J register is set to 6 to modify a word in the B register which has a non-zero mantissa (W07L') and the exponent sign is positive (B46F'). The B register will be shifted left.

J to 7

The J register is set to 7 to modify a word in the B register which has a non-zero mantissa (W07L') and the exponent sign is negative (B46F). The B register will be shifted right.

Clear B [47 ⇒ 1], E to 13, J to 5

If the B register mantissa is zero (W07L) and the exponent is not zero (W72L'), the B register is cleared to zero. The E register is set to 13 to store the zero B register contents. The J register is set to 5 to terminate the operator.

J = 5

The operator waits until the store memory cycle is complete (EEZL). The B register is marked empty if the operator is Store Destructive. The A register is marked unoccupied regardless of which store operator is used. The syllable execute complete level (SECL) is gated true to end the operator.

J = 6

This J register count occurs when the word to be stored has a positive exponent sign and a non-zero mantissa. The mantissa of the word in the B register is shifted left until the exponent is reduced to zero. If a non-zero octal digit is encountered, before the exponent is reduced to zero, the number cannot be integerized and an integer overflow condition is indicated.

Shift B Mantissa Left, Count Exponent Down

If the thirteenth octade position of the B register mantissa is zero (B13L) and the exponent is not zero (W72L'), the mantissa of the B register is shifted left by octades and the exponent counted down. This continues until the number is integerized (exponent reduced to zero).

Integer Overflow, J to 15

If, during the process of integerizing, the mantissa becomes normalized (B13L' no more leading zeros), the integer overflow is set into the interrupt register. The interrupt register flip-flops U17F and U18F are set. The J register is set to 15 to terminate the operator.

E to 13, J to 5

If the exponent is reduced to zero before the mantissa becomes normalized, the E register is set to 13 to complete the store of the B register contents at the address contained in the M register. The J register is set to five to terminate the operator.

J = 7

This J register setting occurs when the word to be stored has a non-zero mantissa and a negative exponent sign. The word in the B register is integerized by shifting right into the X register and decreasing the exponent. During the process of integerizing either one of two things happens:

1. The exponent of the B register is reduced to zero. In this case the portion of the mantissa still in the B register is possibly rounded and then stored at the address indicated by the M register.
2. The mantissa of the B register becomes zero before the exponent of the B register becomes zero. In this case the B register is cleared and stored in the address indicated by the M register.

B 13 through 1 to X13 through 1

The B and X registers are effectively joined to handle octal right shifts. The exponent of B is counted up by one with each shift. This occurs as long as the mantissa is not zero (W07L') and the exponent is not zero (BEXL'). The term BEXL is equivalent to W72L in the logic book.

Q04F to 1

The logical flip-flop (Q04F) is set if a bit is shifted from the B register to the X register. The status of Q04F will be a factor to determine whether a round of the B register mantissa is required.

J equals 7, No Round Required

The rules for rounding the mantissa of the word in the B register are as follows:

1. If the sign of the mantissa is positive, increase the mantissa by one if the portion of the mantissa to be dropped (that portion shifted to the X register) is greater than or equal to one-half.
2. If the sign of the mantissa is negative, increase the mantissa by one if the portion of the mantissa to be dropped is greater than one-half.

NOTE

In the octal system, a 4 is representative of 1/2.

When the exponent becomes equal to zero, the status of X39F and Q04F are checked for a possible round. Rounding is not required when X39F is reset or the sign of the B mantissa is negative (B47F) with Q04F reset. If X39F is reset, the portion of the mantissa shifted out of the B register is less than 4. If Q04F is reset, the portion of the B register mantissa shifted out is not greater than 4. It must be greater than 4 if the mantissa sign is negative. Therefore, with a round operation not required, the B register contents are stored by setting E to 13. If, in the process of shifting right to reduce the exponent to zero, the mantissa of the B register word becomes zero (W07L), the sign of the mantissa is made positive (B47F = 0).

J equals 7, Round Required

The rules for rounding are listed in the preceding paragraph. Checking for a possible round occurs when the exponent equals zero (W72L). Rounding is required when X39F is set and either the sign of the mantissa is positive (B47F') or Q04F is set. If X39F is set, a 4 was shifted out of the B register. If the sign of the mantissa is positive, a 4 shifted out is sufficient to require a round. If Q04F is set, the portion shifted out is greater than 4, and, therefore, a round is required.

The procedure for adding a 1 to the portion of the mantissa still in the B register is as follows:

1. The A register mantissa is cleared.
2. Q01F is set so that a 1 will be added in during the subsequent add.
3. Q09F is reset so the parallel adder will function as such. If it is on, the parallel adder functions as an address adder.
4. The J register is set to 8, where the zero in the A register mantissa and the 1 via Q01F are added to the B register mantissa.

J = 7, Clear B

If the mantissa of the B register is reduced to zero (W07L), while the exponent of B is still not zero (W72L'), the B register is cleared and then stored at the address specified by the M register (E is set to 13). The J register is set to 5 to end the operator.

J = 7, Clear Exponent Sign

If the exponent of the word in the B register is zero, the sign of the exponent is cleared (B46F is reset).

J = 8

The content of the A register mantissa (zero) plus Q01F are added to the B register mantissa, increasing the mantissa in the B register by 1. Note that the parallel adder is functioning as a 39 bit mantissa adder (not a 42 bit adder). The E register is set to 13 to store the word in the B register at the address indicated by M. The J register is set to 5 to end the operator.

J = 15

At J equals 15, the syllable execute complete level is gated true to end the operator. The J register setting occurs if a presence bit or integer overflow interrupt resulted during the operator.

3.10 DIAL A
WORD MODE NOOP
DIAL B
SET VARIANT

DIAL - XX55
NOPL - 0055
DIBL - XX61
VARL - 0061

If the six high order bits of the operator (repeat count field) are not zero, the three most significant bits of the operator are placed in the:

1. G register for DIAL
2. K register for DIBL

and the three next most significant bits of the operator are placed in the:

1. H register for DIAL
2. V register for DIBL.

If all of the six high order bits are zero, then the following action takes place:

1. Nothing happens for a NOPL
2. VARF is set and SALF reset for VARL.

SUMMARY OF OPERATION

The character and bit pointers of either the A or B register (depending on the operator) are set to the value contained in the six high order bits of the operator. If the six high order bits are equal to zero, then there will either be no action or VARF will be set and SALF reset depending upon the operator.

DETAILED DESCRIPTION

J = 0

DIAL A: If the six high order bits of the operator in the T register are not equal to zero (TEZL'), the contents of the T register [12 ⇒ 7] are transferred to the G and H registers. The Syllable Execute Complete Level (SECL) is true with the term EXIT to terminate the operator.

NOPL: If the operator in the T register is 0055, that is, the six high order bits of the T register are equal to zero, then only the term EXIT is true to terminate the operator. No other logical functions occur.

DIAL B: If the six high order bits of the operator in the T register are not equal to zero (TEZL'), the contents of the repeat count field of the T register are transferred to the K and V registers. SECL is true with the term EXIT to terminate the operator.

VARL: If the operator in the T register is a 0061 (VARL), then the following actions will occur:

EXIT

The term EXIT will enable the SECL logic to terminate the operator.

VARF to 1

If the Processor is in sublevel (SALF) and the six high order bits of the T register are zero (TEZL), then set VARF. This is to remember that SALF was turned off.

SALF to 0

If the six high order bits of the T register are zero (TEZL), then reset the sublevel flip-flop (SALF). This will place the Processor in program level to allow relative addressing of R+ 1023 words.

3.11 TRANSFER BITS TRFL-XX65

A field in the A register, starting at the bit position addressed by the G and H registers, replaces a corresponding length field in the B register, starting at the bit position addressed by the K and V registers, and proceeding towards the lower order bit positions.

The length of the field transferred is specified by the six high order bits of the operator. The transfer of bits is terminated by the transfer of the specified number of bits or when either the A or B registers have been exhausted.

The contents of G, H, K and V registers after the operation are the same as prior to the operation. The A register is set to empty.

SUMMARY OF OPERATION

Bits are transferred from the A register to the B register via the Y register. Character transfer from the A register to the Y register is specified by the G register. The desired receiving character position in the B register, specified by the K register, is shifted (aligned) into the first and second octade position of the B register. Thereafter bit transfer commences from the Y register to the B register. Bit transfer from the Y register, specified by the H register bit pointer, is into a bit position of the first and second octade positions of the B register, specified by the V register bit pointer. As each bit is transferred, the H and V registers are incremented to point at the next bit position of a character. As the bits of a character position become exhausted, the G register is incremented to address the next lower significant character of the A register and/or the K register is incremented, resulting in a shift of the B register to bring the next receiving character position into the first and second octade positions. Upon transfer of all the required bits the B register is restored, aligning the character positions to their original configuration.

DETAILED DESCRIPTION

J = 0

If the six high order bits of the operator are equal to zero (TEZL) on entry or after the A and B registers are loaded (if required), the J register is set to 4 to terminate the operator.

Store Dials

The character and bit pointers of the A and B registers are stored in the X register, so they can be restored to their present setting upon completion of the operator.

Q09F to 1, N to 14

The logical flip-flop Q07F is off, on entry to the operator. Therefore, the N register is set to 14 and Q07F is set on the first clock pulse. Q07F's only function is to prevent the N register from being jammed to 14 more than once. The N register is pre-set to 14, resulting in a simulated left character shift of the B register. Thus, when alignment of the B register occurs, the desired character position will be in the first and second octade positions.

NOTE

Normal alignment logic provides for alignment into the 15th and 16th octade positions, which is one character position displaced from the position required in this operator.

A [G] to Y

The first clock pulse of the operator, that finds the A register occupied, transfers the character pointed to by the G register to the Y register. Only the first character transfer occurs at this time, subsequent character transfers from the A register occur at J equals 3.

Load A and/or B

The A and/or B registers are loaded if not occupied on entry to the operator. A normal push-up from the stack occurs if required.

Alignment of B Register

The A and B registers must be loaded, there must be bits to transfer (TEZL') and Q07F must be set, indicating that the N register has been pre-set to 14 for the alignment to commence.

Shift B right by octade. If the K register is not equal to the three high order bits of the N register (KENL'), the B register must be shifted to place the character pointed to by K, into the first and second octade positions of B. If K04F is true, K is pointing at one of the 4 least significant character positions of the B register, it is faster to shift right to attain alignment. If N01F is true, an odd number of shifts are indicated, and additional shifting is required.

Shift B left by octade. If K04F is reset, K is pointing at one of the 4 most significant character positions of the B register, it is faster to shift left to attain alignment.

J to 3. As soon as KENL is true (K equals the 2, 4, 8 bit configuration in the N register), control is set to J equals 3. Figure 3.11-1 shows an example of the B register before and after alignment. In the example, K equals 5 and the N register is shown in its pre-set condition of equal to 14 at the start.

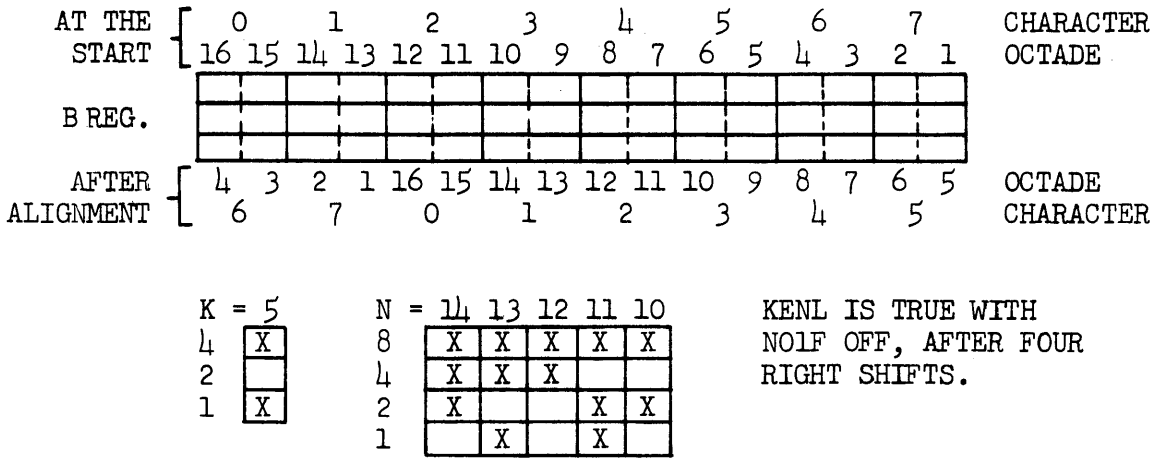


FIGURE 3.11-1. B REGISTER ALIGNMENT

J = 1

This J register setting is attained by a load of either the A or B register, required on entry.

J = 2

This J register setting is arrived at from J equals 3. Going to J equals 2 implies that all bit positions of the present character in the first and second octade positions of the B register have been transferred. The next character position must be shifted into the first and second octade positions of the B register (provided more characters are to be transferred).

Q03F was set at J equals 3 to cause transfer to J equals 2, therefore, Q03F is reset, as control is returned to J equals 3. The B register is shifted left and circulated one octade position to complete the second of two octade shifts required. (The first octade shift occurs at J equals 3). The N register is incremented to tally the left shift of the B register.

J = 3

A [G] to Y, Q04F to 0

If Q04F is on, a character from the A register (specified by G) is transferred to the Y register. Q04F would have been on if the last bit of the present character in the Y register had been transferred, thus the Y register required reloading. Q04F is reset to permit the bit transfer to again commence.

Shift B, N + 1, J to 2

If Q03F is on, all bit positions of the present character in the first and second octade positions of the B register have been transferred into. To bring the next lower significant character into the first and second octade positions of the B register the B register is shifted left and circulated one octade. This is the first half of the character shift; the second half occurs at J = 2. The N register is incremented to tally the shift and the J register is set to two.

Q04F to 1, Clear Y

If HE5L is true, the H register is pointing at the last bit of the character being pointed at by the G register. Q04F is set to 1 to enable the reloading of the Y register at the next clock pulse. The Y register is cleared for the same reason.

Q03F to 1

Q03F is turned on if VE5L is true. VE5L is true if the V register is pointing at the last bit of the character in the first and second octade positions of the B registers, i.e., the character being pointed to by the K register.

J equals 3 - Q04F' • Q03F'

Both Q04F and Q03F must be off for the following actions to occur. These flip-flops are off as long as the bit pointers of the A and B registers are not pointing to the last bits of their respective character. Q04F, when set, inhibits bit transfer to allow reloading of the Y register. Q03F, when set, inhibits bit transfer to allow the required character shift of the B register.

Y [H] to B [KV]

The H register bit pointer indicates which bit of the Y register to transfer to the first and second octades of the B register. The V register bit pointer indicates which bit position of the first and second octades is to receive the bit from the Y register. The following logical equations show the set and reset logic of the first and second octade positions of the B register (bits B01F through B06F).

B01F = J42L • VE5L • YOHL
B01F'= J42L • VE5L • YOHL'

B02F = J42L • VE4L • YOHL
B02F'= J42L • VE4L • YOHL'

B03F = J42L • VE3L • YOHL
B03F'= J42L • VE3L • YOHL'

```
B04F = J42L • VE2L • YOHL
B04F'= J42L • VE2L • YOHL'

B05F = J42L • VE1L • YOHL
B05F'= J42L • VE1L • YOHL'

B06F = J42L • VEZL • YOHL
B06F'= J42L • VEZL • YOHL'

J42L = (J = 3) • TRFL • Q04F' • Q03F'

YOHL = YR1F • H04F • H01F              H = 5
     + YR2F • H04F • H01F'               = 4
     + YR4F • H02F • H01F                = 3
     + YR8F • H02F • H01F'               = 2
     + YRAF • H04F' • H02F' • H01F       = 1
     + YRBF • H04F' • H-2F' • H01F'      = 0
```

H and V plus 1, T minus 1

The H register is counted up by one to point at the next lower significant bit positions of the Y register. The V register is counted up by one to point at the next lower significant bit position of the first and second octades of the B register. The six high order bits of the T register are counted down by one at each bit transfer to tally the bit transfers.

K plus 1, G plus 1

If VE5L is true, the last bit of the character, presently in the first and second octades of the B register, is being loaded. Hence, K is incremented to point at the next character of the B register. If HE5L is true, the last bit of the character in the Y register is being transferred out. Hence, the G register is incremented to point at the next character in the A register.

J to 4

The operator specifications state that the operator is terminated if either the A or B registers exceed the least significant bit position of their respective register or if the six high order bit positions of the T register are counted down to zero. Hence, if G equals 7 and H equals 5, the last bit of the least significant character of the A register is being transferred. If K equals 7 and V equals 5, the last bit of the least significant character of the B register is being transferred. TE1L being true indicates the last bit (of the specified number) is about to be transferred. The J register is set to 4 to terminate the operator.

J = 4

Restore Dials, Clear Top of Stack

The G, H, K and V register settings, which were formerly stored in the X register, are returned to their respective registers. AROF is cleared to indicate that the top address of the stack no longer contains valid information.

Restore B Register

If either the 8th bit of the N register is off (N08F') or the level N15L is true, it is most expeditious to restore the B register to its original configuration by shifting RIGHT and circulating. The N register is decremented to tally the right shifts. When the N register equals 14, the B register is in its original configuration.

NOTE

Normally, when the N register is equal to zero, the B register is in its original configuration. However, at J = 0 of this operator, the N register was pre-set to 14 prior to any shifts of the B register. Thus in this case, when N14L is true, the B register is in its original configuration.

If the 8th bit of the N register is on (N08F) and neither N14L nor N15L is true (N14L' • N15L'), the most expeditious way of restoring the B register is to shift LEFT and circulate. The N register is incremented to tally the right shifts.

EXIT

When N14L becomes true, indicating that the B register has been restored to its original alignment, the Syllable Execute Complete Level is gated true to terminate the operator.

3.12 COMPARE FIELD EQUAL CFEL-XX75

A field in the A register, starting at the bit position addressed by the B and H registers, is compared with a corresponding length field in the B register, starting at the bit position addressed by the K and V registers and proceeding towards the lower order bit positions.

The length of the fields in the registers is specified by the six high order bits of the operator. The comparison is terminated by the comparison of the number of bits specified or by the comparison of the low order bit of either register.

If all of the corresponding bits of the fields compared are equal, the low order bit of the A register is set to one and other bit positions of the A register are set to zero. If any of the corresponding bit positions of the fields compared are not equal, all bit positions of the A register are set to zero. The contents of the B, G, H, K and V registers after the operation are the same as prior to the operation.

SUMMARY OF OPERATION

Upon entry into the operator, if the six high order bits of the operator are equal to zero and the B and A registers are occupied, the A register is cleared, A01F is set (thus marking the operation as true), and the operator is terminated.

If the six high order bits of the operator are not zero, and either or both registers are not loaded, load them and continue. The dial pointers (bit and character pointer registers) are temporarily stored in the X register for preservation. The B register is aligned to the output alignment station. The character containing the first bits to be compared is transferred from the A to the Y register and from the B to the Z register.

Logical toggle Q05F is turned on, this results in starting the compare with an assumption of equality. The comparison of bits, starts at the most significant bit position of both fields and continues until a false condition (the field in B is not equal to the field in A) is found, or either the end of the word is reached or the T register is counted down to zero. As the bits of a character of either field are exhausted, the next character of that field is transferred into the compare registers. Logical toggles are employed to inhibit comparison during the shifting of new characters into the compare registers. As soon as a false condition is found, the operation terminates leaving the false condition (A register is cleared) in the A register. The dial settings are restored. All logical toggles and compare registers are cleared. The B register is restored, and the operation terminates. If the fields do not find a false condition for a word length, the operation is terminated leaving the A register marked as a true condition (A is cleared except for the least significant bit). B is restored.

DETAILED DESCRIPTION

J = 0

If the six high order bits of the operator are equal to zero, and the A and B registers are loaded, the A register is cleared while the J register is set to 4 to terminate the operator. When this operator is used with the six high order bits of the operator equal to zero, the top word of the stack is cleared and is subsequently marked with a true comparison indication

Q05F to 1

The logical flip-flop Q05F is unconditionally turned on to assume equality. If, during the compare phase of the operator, an unequal compare is encountered, Q05F is turned off. The status of Q05F at the termination of the operator determines whether a true or false condition is set into the A register.

Store Dials

The contents of the G, H, K and V registers are temporarily stored in the X register bit positions X39 and X28. They are restored at J = 4.

Push-Up

If either the A and/or B registers are not loaded on entry to the operator, they are loaded in the normal fashion.

A [G] to Y

If or when the A register is loaded, a character from the A register is transferred to the Y register as defined by the G register. This transfers the most significant character (containing bits to be compared) of the word in the A register to one of the compare registers (Y).

Alignment of B Register

When the A and B registers are loaded, and the six high order bits of the operator are not zero (TEZL'), the most significant character position of the B register (that has bits to be compared) is shifted to the output alignment station (the 15th and 16th octade positions), if necessary.

If the K register is not equal to the three high order bits of the N register (KENL'), indicating that the specified character position is not in the output alignment station, the status of K04F is checked. If

K04F is on, the B register is shifted RIGHT and circulated while counting the N register down. If K04F is off, the B register is shifted LEFT and circulated while counting the N register up.

If the B register is shifted right (K04F is on) to attain alignment and N01F is On (When KENL is true), an additional left octade shift is required to place the complete character position in the output alignment station.

Transfer B to Z

When KENL is true and N01F is off, the specified character in the output alignment station is transferred to the Z register. At the same time, the B register is shifted one octade position to bring one half of the next character into the output alignment station. The N register is counted up by one to tally the left shift. Q02F is set to remember an octade shift of the B register has occurred, so that at J = 3 the second half of the complete character shift will take place. The J register is set to 3 for continuation of the operator.

J = 1

This J count is part of a normal push-up of the stack.

J = 3

During this control state the comparison takes place, bit by bit, from the most significant to the least significant bit. The first compare that finds the specified bit in Z not equal to the specified bit in Y (BENL) turns Q05F off. Upon encountering a not equal comparison, the A register is cleared and the J register is set to 4 to terminate the operator. Q05F is turned off to inhibit the setting of A01F at J = 4, and the operator is ended with the A register marked to a false condition.

```
BENL = YOHL • ZOVL'
     + YOHL' • ZOVL

- O' - ZOVL'
- O - ZOVL

- I - + ZR1F • V04F • V01F          V = 5
      + ZR2F • V04F • V01F'             4
      + ZR4F • V02F • V01F              3
      + ZR8F • V02F • V01F'             2
      + ZRAF • V04F' • V02F' • V01F     1
      + ZRBF • V04F' • V02F' • V01F'    0
```

```
- 0' - YOHL'
- 0  - YOHL
- I  - + YR1F • HO4F • HO1F          H = 5
       + YR2F • HO4F • HO1F'             4
       + YR4F • HO2F • HO1F              3
       + YR8F • HO2F • HO1F'             2
       + YRAF • HO4F'• HO2F' • HO1F      1
       + YRBF • HO4F'• HO2F' • HO1F'     0
```

Count Bit and Character Pointers

With Q04F off and either Q03F off or Q02F on, the bit pointers are advanced. Q04F being off indicates that the Y register is not in the process of being loaded. Q03F off, or Q02F on indicates the Z register is not in the process of being loaded. Thus, if both the Y and Z registers are loaded, a bit comparison is taking place; therefore the H and V registers bit pointers are incremented and the six high order bits of the operator in the T register are decremented.

If the H register (bit pointer for the character in the Y register) is equal to 5, the G register is incremented, the Y register is cleared and Q04F is turned on. Q04F is set on to inhibit compare while the next character in the A register is transferred to Y.

If the V register (bit pointer for the character in the Z register) is equal to 5, the K register is incremented, the Z register is cleared and Q03F is turned on. Q03F is set on to inhibit compare while the next character in the B register alignment station is shifted to the Z register. Q03F is also set on to implement the next character shift of the B register.

B Register Shift

If Q03F is on, the present character in the output alignment station of the B register (15th and 16th octades) is shifted to the Z register. At the same time, Q03F gates the first half of a character shift (one octade shift and circulate). The N register is incremented to tally the shift of B. Q03F also gates the setting of Q02F to the 1 state. Q02F going on subsequently gates the second half of the required character shift (second octade shift and circulate). Q02F is unconditionally reset, allowing it to gate only one octade shift. During the first character transfer to the Z register (at J = 0), Q02F is set. Thus, on entry to this J = 3 count, Q02F is on to gate the second half of a character shift. All subsequent settings of Q02F occur at J = 3.

J = 4

If Q05F is on, indicating that the field in the B register is equal to the field in the A register, the least significant bit of the A register is turned on (A01F). With A01F on and the remainder of the A register cleared, the compare field equal operator indicates a true comparison.

The bit and character pointer values which were stored in the X register at J = 0, are restored by transferring the X register bit (X39 through X28) to the G, H, K and V registers.

Alignment of B Register

If the N register does not equal zero, the B register is out of alignment relative to its original position and must therefore be restored.

If the N Greater than 8 level (NG8L) is true, it is more expedient to bring the B register into alignment by shifting the B register left by octades and circulating it.

If the N Less than 8 level (NL8L) is true, B register alignment is accomplished by shifting right by octades and circulating it.

The N register is counted to tally the shifting of the B register, and when the N register equals zero (NEZL) the B register is in its original position. The Syllable Execute Complete level is true when NEZL is true to terminate the operator.

NG8L = NO8F
NL8L = NO8F' • (NO4F + NO2F + NO1F)

3.13 COMPARE FIELD LOW CFLL-XX71

A field in the A register, starting at the bit position addressed by G and H registers, is compared with a field in the B register, starting at the bit position addressed by the K and V registers, and proceeding towards the low order bit positions.

The length of the fields in the registers is specified by the six high order bits of the operator. The comparison is terminated by the comparison of the number of bits specified or by the comparison of the low order bit position of either register.

If the magnitude of the field compared in the B register is less than the magnitude of the field compared in the A register, the low order bit of the A register is set to one and all other bit positions of the A register are set to zero; otherwise all bit positions of the A register are set to zero. The contents of the B, G, H, K and V registers after the operation are the same as prior to the operation.

SUMMARY OF OPERATION

Upon entry into the operator, if the six high order bits of the operator are equal to zero and the A and B registers are occupied, the A register is cleared (thus marking the operation as false) and the operator is terminated. If the six high order bits of the operator are equal to zero and either or both registers are empty, either or both are loaded and then the A register is cleared (marking the operation as false) and the operator is terminated.

If the six high order bits of the operator are not equal to zero, and both registers are loaded: The dial pointers (bit and character pointer registers) are temporarily stored in the X register for preservation. The B register is aligned to the output alignment station and the characters containing the first bits to be compared are transferred, from the A to the Y register and from the B to the Z register.

The comparison of bits starts at the most significant bit position of both fields and continues until a true condition is found; that is, the field in B is less than the field in A, or either the end of the word is reached, or the specified number of bits have been compared. As the bits of a character of either field are exhausted, the next character of either field is transferred into the compare registers. Logical toggles are employed to inhibit comparison during the shifting of new characters into the compare registers.

As soon as a true condition is found, the operation terminates leaving the true indication (A register is cleared except for the least significant bit which is turned on) in the A register, the dial settings are restored, all logical toggles and compare registers are cleared, the B register is restored, and the operation terminates.

If the fields do not find a true condition by the end of a word or after the specified number of bits have been compared, the operation is terminated, leaving the A register marked as a false condition (all bits of A are cleared) and the B register is restored.

DETAILED DESCRIPTION

J = 0

If the six high order bits of the operator are equal to zero, and the A and B registers are loaded, the A register is cleared while the J register is set to 4 to terminate the operator. When this operator is used with the six high order bits of the operator equal to zero, the top word of the stack is cleared to mark it with a false comparison indication.

Store Dials

The contents of the G, H, K and V registers are temporarily stored in the X register bit positions X39 through X28. They are restored at J = 4.

Push-Up

If either the A and/or B registers are not loaded on entry to the operator, they are loaded in the normal fashion.

A [G] to Y

If or when the A register is loaded, a character from the A register is transferred to the Y register as defined by the G register. This transfers the most significant character (containing bits to be compared) of the word in the A register to one of the compare registers (Y).

Alignment of B Register

When the A and B registers are loaded, and the six high order bits of the operator are not zero (TEZL'), the most significant character position of the B register (that has bits to be compared) is shifted to the output alignment station (the 15th and 16th octade positions), if necessary.

If the K register is not equal to the three high order bits of the N register (KENL'), indicating that the specified character position is not in the output alignment station, the status of K04F is checked. If K04F is on, the B register is shifted right and circulated while counting the N register down. If K04F is off, the B register is shifted left and circulated while counting the N register up.

If the B register is shifted right (K04F is on) to attain alignment and N01F is on (when KENL is true), an additional right octade shift is required to place the complete character position in the output alignment station.

Transfer B to Z

When KENL is true and N01F is off, the specified character in the Output alignment station is transferred to the Z register. At the same time, the B register is shifted one octade position to bring one half of the next character into the output alignment station. The N register is counted up by one to tally the left shift. Q02F is set to remember the one octade shift of the B register has occurred, so that at J = 3 the second half of the complete character shift will take place. The J register is set to 3 for continuation of the operator.

J = 1

This J count is part of a normal push-up of the stack.

J = 3

During this control state the comparison takes place, bit by bit, from the most significant to the least significant. The first compare that finds the specified bit in Z not equal to the specified bit in Y (BENL), clears the A register (A to 0) and sets J equal to 4 to terminate the operator.

Upon encountering a not equal comparison, Q05F is turned on if the Z less than Y level (ZLYL) is true; in which case the operator is ended with the A register marked to a true condition. Otherwise the A register is marked to a false condition.

```
BENL = YOHL • ZOVL'                          ZLYL = YOHL • ZOVL'
     + YOHL'• ZOVL

- O' - ZOVL'
- O  - ZOVL
- I  - + ZR1F • VO4F • VO1F                  V = 5
       + ZR2F • VO4F • VO1F'                    4
       + ZR4F • VO2F • VO1F                     3
       + ZR8F • VO2F • VO1F'                    2
       + ZRAF • VO4F'• VO2F' • VO1F             1
       + ZRBF • VO4F'• VO2F' • VO1F'            0

- O' - YOHL'
- O  - YOHL
- I  - + YR1F • HO4F • HO1F                  H = 5
       + YR2F • HO4F • HO1F'                    4
       + YR4F • HO2F • HO1F                     3
       + YR8F • HO2F • HO1F'                    2
       + YRAF • HO4F' • HO2F' • HO1F            1
       + YRBF • HO4F' • HO2F' • HO1F'           0
```

Count Bit and Character Pointers

With Q04F off and either Q03F off or Q02F on, the bit pointers are advanced. Q04F being off indicates that the Y register is not in the process of being loaded. Q03F off or Q02F on indicates the Z register is not in the process of being loaded. Thus, if both the Y and Z registers are loaded, a bit comparison is taking place. The H and V registers' bit pointers are incremented and the six high order bits of the operator in the T register are decremented.

If the H register (bit pointer for the character in the Y register) is equal to 5, the G register is incremented, the Y register is cleared and Q04F is turned on. Q04F is set on to inhibit compare while the next character in the A register is transferred to Y.

If the V register (bit pointer for the character in the Z register) is equal to 5, the K register is incremented, the Z register is cleared and Q03F is turned on. Q03F is set on to inhibit compare while the next character in the B register alignment station is shifted to the Z register. Q03F is also set on to implement the next character shift of the B register.

B Register Shift

If Q03F is on, the present character in the output alignment station of the B register (15th and 16th octades) is shifted to the Z register. At the same time Q03F gates the first half of a character shift (one octade shift and circulate). The N register is incremented to tally the shift of B. Q03F also gates the setting of Q02F to the 1 state. Q02F going on gates the second half of the required character shift (second octade shift and circulate). Q02F is unconditionally reset, allowing it to gate only one octade shift.

During the first character transfer to the Z register (at J = 0), Q02F is set. Thus, on entry to this J = 3 count, Q02F is on to gate the second half of a character shift. All subsequent settings of Q02F occur at J = 3.

J to 4. The operator specifications state that the operator is terminated if either the A or B registers exceed the least significant bit position of their respective register or if the six high order bit positions of the T register are counted down to zero. Hence, if G equals 7 and H equals 5, the last bit of the least significant character of the A register is being transferred. If K equals 7 and V equals 5, the last bit of the least significant character of the B register is being transferred. The level TE1L being true, indicates that the last bit (of the specified number) is about to be transferred. Thus, the J register is set to 4 to terminate the operator and the A register is cleared to terminate the operator.

If the N Greater than 8 level (NG8L) is true, it is quicker to bring the B register into alignment by shifting the B register left by octades and circulating it.

If the N Less than 8 level (NL8L) is true, alignment is accomplished by shifting right by octades and circulating it.

The N register is counted to tally the shifting of the B register, and when the N register equals zero (NEZL) the B register is in its original position. The Syllable Execute Complete level is true when NEZL is true to terminate the operator.

NG8L = NO8F
NL8L = NO8F' • (NO4F + NO2F + NO1F)

3.14 SET FLAG BIT
RESET FLAG BIT

SFBL-4015
RFBL-2015

Set Flag Bit: Set the flag bit of the word in the A register to one.

Reset Flag Bit: Set the flag bit of the word in the A register to zero.

SUMMARY OF OPERATION

If the A Register is empty it is loaded with the top word in the stack. If the top word is in B then B is transferred to A. If neither A nor B registers are loaded, then B is loaded from Memory and transferred to A. With a word in the A register, the 48th bit is set to 1 or 0 as indicated by the operator. If the 48th bit equals 1, the word is considered a Descriptor. If it equals 0, the word is an operand.

DETAILED DESCRIPTION

J = 0

Upon entry, if the A register is empty and the B register is full (AROF' • BROF), the word in B is transferred to A, the B register is marked as empty and A is marked as full.

If both A register and B register are empty, upon entry, (AROF' • BROF') then Memory access to load B is initiated, (E to 3) and primary control counter J is set to 1 where the operator waits for completion of Memory access.

When the A register is loaded and if the operator is Reset Flag Bit A48F is set to 0. If the operator is Set Flag Bit then A48F is set to 1; in either case, the syllable execution complete level becomes true and the operator terminates.

J = 1 Level

Upon receipt of MROF signal from Memory the top word in the stack is placed in "B", BROF is set to 1, marking "B" full, the address register (S) is counted down to address the next word in the stack and J is set to 0.

3.15 TEST FLAG BIT — TFBL - 2031
INTERROGATE PERIPHERAL STATUS — IPSL - 2431
INTERROGATE I/O CHANNEL — TIOL - 6431

If the operator is a Test Flag Bit, and if the flag bit of the word in the B register is zero, the low order bit of the word in the A register is set to one and all other bits of the A register are set to zero; otherwise, all bits of the A register are set to zero.

Interrogate I/O Channel will set a value into the A register (three low order bits) corresponding to the number of the I/O channel that is not busy.

Interrogate Peripheral Status will allow the MCP to determine, very rapidly, which units have changed status since the last interrogation. This bulk interrogation of peripheral units eliminates the need for a separate I/O operation for each unit.

SUMMARY OF OPERATION

Upon entry to the operator the stack is checked to determine if stack adjustment is necessary. If the operator in the T register is Test Flag Bit (TFBL), then the stack is adjusted to place the top word of the stack in the B register. If the operator is interrogate, then the stack must only be adjusted to mark the A register empty.

With the test flag bit operator (TFBL) and the top word in the stack is in the B register, the B register is checked to see if it contains an operand or a descriptor. If the word is an operand (B48F'), the least significant bit of the A register (A01F) is set to one and the rest of the A register is cleared. If the word is a descriptor (B48F), the A register is cleared and the operator is terminated. In either case the A register is marked as valid.

Interrogate I/O Channel: This operator interrogates the I/O Channels to determine which channel is currently in line to be assigned next, that is, which is the lowest-numbered currently available Input-Output Control Unit. A literal is placed in the top of the stack. The literal indicates the next assigned channel in the following way:

Literal	Channel
0	All channels busy
1	Channel one due for assignment
2	Channel two due for assignment
3	Channel three due for assignment
4	Channel four due for assignment

Interrogate Peripheral Units: This operator places in the top of the stack a word representing the current ready status of the peripheral equipment. One bit in the word is associated with each peripheral unit. This bit is set to 1 if the associated unit is ready; to 0 if the associated unit is not ready.

The A register is set to zero. The 31 low order bits are set to reflect the current status of the peripheral units.

The A register is marked full and the operation is terminated.

PERIPHERAL UNIT (PS LINES)	A REG. FF
TSU - A	1
TSU - B	2
TSU - C	3
TSU - D	4
TSU - E	5
TSU - F	6
TSU - H	7
TSU - J	8
TSU - K	9
TSU - L	10
TSU - M	11
TSU - N	12
TSU - P	13
TSU - R	14
TSU - S	15
TSU - T	16
DRUM - 1	17
DRUM - 2	18
DISC FILE - 1	19
DISC FILE - 2	20
PRINTER - 1	21
PRINTER - 2	22
CARD PUNCH	** 23
CARD READER - 1	24
CARD READER - 2	25
SPO - KEYBOARD	26
P.T. PUNCH - 1	27
P.T. READER - 1	28
P.T. READER - 2	29
P.T. PUNCH - 2	30
DATA COM. CONTROL	31

FIGURE 3.15-1 PERIPHERAL TO A REGISTER

Figure 3.15-1 shows the association between the A register bit positions and peripheral units.

DETAILED DESCRIPTION

JOOL

UNCONDITIONALLY indicate A register as occupied (AROF set to 1).

AROF • BROF

With A register and B register occupied, push down the stack by initiating a store of the B register (E to 11). Indicate the storage location of the latter (S + 1). Exit this J count to J2.

AROF • $\overline{\text{BROF}}$

With A register occupied and B register not occupied, shift the contents of A register to B register and mark the B register as occupied.

$\overline{\text{AROF}}$ • $\overline{\text{BROF}}$ • $\overline{\text{T09F}}$ (Test Flag Bit Operator)

With A register and B register not occupied and a Test Flag Bit Operator (T09F') initiate a read out of the top word of the stack from core memory to B register (E to 3). Exit this J count to J1.

$\overline{\text{AROF}}$ + $\overline{\text{BROF}}$

If either A register or B register, or both A register and B register are unoccupied, ensure that the contents of A register are equal to zero due to single-ended set logics at J3.

AROF • BROF + AROF • $\overline{\text{BROF}}$ + $\overline{\text{AROF}}$ • $\overline{\text{BROF}}$ • $\overline{\text{T09F}}$

If either A register or B register is occupied (not both), exit this J count to J3. With A register and B register not occupied and an Interrogate I/O Channel or Peripheral Status operator (T09F), exit this J count to J3. No further stack adjustment is necessary.

J01L

MROF (Memory Read)

With the Memory Information Register occupied (MROF), shift the contents of MIR to B register, mark B register as occupied; Decrement S register; exit this J count to J3.

J02L

MWOF (Memory Store)

With the Memory Information Register occupied (MWOF), Store the contents of the Memory Information Register; shift the contents of A register to B register; clear A register; exit this J count to J3.

J03L

Unconditionally EXIT from this operator.

$\overline{B48F} \bullet \overline{T09F}$ (Test Flag Bit)

If the Flag Bit of the top word of the stack is zero, set A01F bit to 1.

T09F • $\overline{T12F}$ (Interrogate Peripheral Status)

For any peripheral unit that is ready (CC indication), set a corresponding bit in A register to indicate same (A [31 <= 1] by PS). For PS to A register bit see Figure 3.15-1.

T09F • T12F (Interrogate I/O Channel)

Figure 3.15-2 indicates the A register bits set/reset for the I/O Channel interrogation.

A01F	A02F	A03F	CHANNEL
0	0	0	
X	0	0	I1BS(D1RSX)
0	X	0	I2BS(D2RSX)
X	X	0	I3BS(D1RSX • D2RSX)
0	0	X	I4BS(D3RSX)

FIGURE 3.15-2

The Central Control logic will only allow one of the I/O channels at a time be not busy. The I/O not busy lines in CC (InBS') are used to develop the DnRSX lines to the Processor; n indicates any value of 1 to 4. The following is the logic for developing the DnRSX levels.

$$D1RSX = \overline{I1BS} + \overline{I3BS}$$
$$D2RSX = \overline{I2BS} + \overline{I3BS}$$
$$D3RSX = \overline{I4BS}$$

From this logic it is determined that D1RSX is True if I/O 1 or 3 is not busy; D2RSX is True if I/O 2 or 3 is not busy; and D3RSX is True if I/O 4 is not busy.

3.16 RESET SIGN BIT
SET SIGN BIT
CHANGE SIGN BIT
STACK SEARCH FOR FLAG

MSPL - 4431
MSNL - 0431
CSSL - 1031
SSFL - 7031

Reset Sign Bit. Set the sign bit of the word in the A register to zero.

Set Sign Bit. Set the sign bit of the word in the A register to one.

Change Sign Bit. Complement the sign bit of the word in the A register.

Stack Search For Flag. Start at the address specified by the fifteen low order bits of the word in the A register, search consecutive memory locations until a word is found with the flag bit on, a descriptor is left in the top of the stack pointing to the word found with the flag bit on.

SUMMARY OF OPERATION

Stack adjustment is the same for all of these operators; if the top word in the stack is not in the A register, a stack push up is done to mark the A register as full.

The sign of the mantissa of the word in the A register is adjusted (set, reset, or complemented) depending upon which of the sign operators is in the T register.

The Stack Search for Flag operator will address successive memory locations, examining the flag bit of the word in each of these addresses. If the flag bit is on, the address of this word is placed into the 15 low order bits of the A register. The rest of the word is set to zero, and the word is marked as a present data descriptor. The operator is then exited. If the flag bit is off, the addressing register (M) is increased by 1 and the flag bit of the next word is examined. This process continues until a word is found with the flag bit on.

DETAILED DESCRIPTION

J = 0

Stack Adjustment

If both the A and B registers are empty (AROF' • BROF'), a load of the A register is initiated by setting the E register to 2. The J register is set to 1 to complete the load of the A register.

If the A register is empty and the B register is loaded (AROF' • BROF), the B register contents are transferred to the A register and the A register is marked as valid (AROF to 1) and the B register is marked as invalid (BROF to 0).

A47 to 0

If the operator is Reset Sign Bit (MSPL), the sign of the mantissa is made positive.

A47 to 1

If the operator is Set Sign Bit (MSNL), the sign of the mantissa is made negative.

Δ A47 to A47

If the operator is Change Sign Bit (CSSL), the sign of the mantissa is complemented by pulsing both the set and reset sides of A47F.

EXIT

Terminate the operator if it is one of the sign operators (MSPL + MSNL + CSSL).

A [15 ⇒ 1] to M, E to 4, J to 2

A Stack Search for Flag operator (SSFL) causes the 15 low order bits of the A register to be shifted to the M register to address the first Core Memory word which will be placed in the A register (E to 4) and tested for flag bit. Exit from this J count to J = 2.

J = 1

This J register setting only occurs if both the A and B registers were empty on entry to the operator. Upon completion of the memory access to load the A register, control is returned to J = 0. The A register is marked occupied and the stack address register (S) is counted down, which is a normal procedure.

J = 2

EEZL

Upon completion of the memory access request (EEZL) initiated by this operator, unconditionally make the last word searched for (in the A register) a present (A46F to 1) data descriptor (A48F to 1 and A47F to 0). Within this descriptor place the address of the last word searched for (M to A [15 ⇒ 1]. The remainder of the A register is unused, set it to zero (A [45 ⇒ 16] to 0).

A48F

If at the same time that the actions at EEZL take place the flag bit in the A register is off, then the last word accessed is not a descriptor. Therefore, continue the search by setting E to 4 and increase the address register by one (M + 1).

A48F

If the last word accessed was a descriptor, then exit this operator; a descriptor has been found.

3.17 SINGLE PRECISION ADD
SINGLE PRECISION SUBTRACT

ADlL-0101
SUlL-0301

ADD

The operands in the A and B registers are added algebraically and the sum left in the B register. For all conditions, at the end of the operation the A register is set to empty, the B register is set to full and the B register flag bit is set to zero.

If either operand has a mantissa of zero, the non-zero operand is the result. If both operands have a mantissa of zero, the B register is set to all zeros. In either case, the operation is terminated.

If the mantissa signs and the exponents of the operands are equal, the mantissas are added and the sum placed in the B register. If the sum exceeds 13 octal digits, the mantissa of the sum is shifted right one octal place, rounded and the exponent algebraically increased by one.

If the exponents of the operands are equal but the mantissa signs are unequal, the difference of the mantissa with appropriate sign is placed in the B register. If the difference is equal to zero, the B register is set to all zeros.

If the exponents of the operands are unequal, the operands are first aligned. If the alignment causes the smaller operand to be shifted right 14 octal places, the larger operand is the result.

If the alignment causes the smaller operand to be shifted right, but less than 14 octal places, the digits of the smaller operand shifted out of the register are used to obtain the result.

If the signs of the operands are equal, the mantissas are added and the sum placed in the B register. If the sum does not exceed 13 octal digits, the last digit shifted out of the register is used for rounding the result. If the sum is 14 digits, the mantissa in B is rounded to 13 digits.

If the signs of the operands are unequal, the digits are complemented as they are shifted out of the register during alignment. In effect the equivalent of a 15 digit subtraction occurs in this latter case and the result is rounded to the 13 most significant digits of the 15-digit result.

If the result has an exponent greater than $+63_D$ the exponent over-flow bit is set in the Interrupt register. The B register contains the correct mantissa, mantissa sign and exponent sign. The magnitude of the correct exponent is contained in the exponent field of the B register modulo 64.

SUBTRACT

The operand in the A register is algebraically subtracted from the operand in the B register and the difference left in the B register. An add operation is performed as specified for the Add operator, except for the conditions of sign comparison. Effectively, the mantissa sign of the operand in the A register is reversed, and the operands added.

SUMMARY OF OPERATION

The performance of addition or subtraction is accomplished while adhering to the basic rules of algebraic addition and subtraction. The basic rules of algebraic addition and subtraction state that for:

Add

Like signs: Add and use the common sign.
Unlike signs: Subtract the smaller from the larger and use the sign of the larger.

Subtract

Change the sign of the subtrahend and follow the rules of algebraic addition.

A Register contains - Subtrahend
B Register contains - Minuend
B Register - Difference

Restating the basic algebraic rules to define when internal addition or subtraction is to be performed, results in:

Internal Subtract = Unequal signs • add operator
OR Equal signs • subtract operator

Internal Addition = Equal signs • add operator
OR Unequal signs • subtract operator

Prior to the execution of the actual add, or complement add, the two operands must be aligned. Two floating point numbers are aligned if their exponent magnitudes and exponent signs are equal. If the exponents of both operands are zero, the signs of the exponents are ignored i.e., the operands are aligned when their exponents equal zero. The following examples illustrate various types of alignments and results achieved during add and subtract. In the examples the symbols and numbers have the following significance:

Sn Se E Mantissa -- X Register
Sn = Sign of number
Se = Sign of exponent
E = Exponent magnitude

Example 1 - Add; operands aligned by normalizing one of the operands.

A	+	+	03	0012567543210
B	+	+	02	5120000000777
A	+	+	02	012567543210 0
B	+	+	02	5120000000777
B	+	+	02	52456754330 77

Example 2 - Add; operands aligned by scaling one of the operands.

A	+	+	03	1256754321000
B	+	+	02	5120000000070
A	+	+	03	1256754321000
B	+	+	03	0512000000007
B	+	+	03	1770754321007

Example 3 - Add; operands aligned by normalizing one of the operands and scaling the other.

A	+	+	03	0012571234567
B	+	-	01	512007654321 0
A	+	+	01	1257123456700
B	+	+	01	0051200765432
B	+	+	01	1330324444332

Example 4 - Subtract; operands aligned by normalizing one of the operands and scaling the other. Result is normalized and rounded.

A	+	+	02	0100000001434
B	+	-	02	3000000002333
A	+	+	01	1000000014340
B	+	+	01	0003000000002 445

The 333 shifted out of the B register into the X register is in 8's complement form. The 7's complement of the smallest mantissa is placed in the A register. Note that no pre-carry is required in this example (something scaled into X is in 8's complement form):

Example 4 (Con't):

A	+	+	01	7774777777775		
B	+	+	01	1000000014340	445	(445 in X Register)
B	+	+	01	10775000014335	445	
B	+	+	00	7750000143354	45	Normalized
B	+	+	00	7750000143355		Rounded answer

Flow Sequence

The occurrences during single precision add-subtract relative to the J register setting is as follows:

J = 0 Start

1. Initiate loading of the A and B registers if necessary
2. Determine if exponents are unequal
3. If exponents are equal or the operand is an integer, either:

   A. Perform internal add, or
   B. Complement appropriate operand for internal subtract

J = 1 Stack adjustment if required

J = 2 Operand Adjustment

1. Adjust operands if exponents are unequal
2. Perform internal add operation; if round required, add one
3. Complement appropriate operand for internal subtract operation

J = 4 Scale for Overflow

1. Scale results one place
2. Prepare for possible round

J = 5 Subtract

1. Complement addition for internal subtract
2. Set sign of result
3. Determine whether to de-complement, normalize, round or exit

J = 7 Complement

1. Change sign of result
2. Complement result

J = 8 Normalize

1. If the result of subtraction is un-normalized; shift result left one position.

J = 15 Exit, Round

1. If exponent of result is greater than +63, set exponent overflow interrupt
2. If result is zero, clear exponent and sign to zero
3. Add operation for round after internal subtract
4. Add operation for round after scale for overflow
5. Add operation for decomplement
6. Terminate the operator

Q Flip-Flops

The Q flip-flops are utilized as follows:

Q01F - Add a one for either subtraction, rounding or decomplementing.
Q02F - Change sign of result during internal subtract.
Q04F - Indicates a non-zero digit has been scaled into the X register.
Q06F - Indicates operands have been interchanged for scaling.

DETAILED DESCRIPTION

If either the A register, B register or both are empty on entry, the A and/or B registers are loaded.

J = 0 AR0F • BR0F

When the A and B registers are loaded, the A register is marked unoccupied and the B register flag bit (B48F) is set to zero, thus assuring the results to be an operand.

B to 0

If the A and B register mantissas are both zero, (W06L • W07L), the answer is zero, therefore, the B register is cleared to all zeros. (The B register exponent might not have been zero).

Exit

If the A register mantissa equals zero (W06L), the answer is in the B register. The syllable execute complete level is true to terminate the operator.

A [46 through 40] to B [46 through 40]

If the A register mantissa is not zero and the B register mantissa is zero (W06L' • W07L), the answer is in the A register and must be transferred to the B register. The exponent, plus sign of the operand in the A register,

is transferred to the B register. The next clock pulse will find exponents equal; therefore, control is transferred to the W73L • J91L box where the A register mantissa will be transferred to the B register via an addition operation.

J to 2

If the A register mantissa is not zero and the exponents are not algebraically equal (W06L' • W73L'), the J register is set to 2 where the alignment of the operands occurs.

J = 1

Entry into this J register setting is only required if either the A or B register was unoccupied on entry to the operator.

J = 2

If the exponents of the operands were not equal and alignment was required, the J register was set to 2. If during alignment of the operands scaling of the B register is required, the B register mantissa is checked for the possibility of the B register mantissa being cleared to zero.

If the exponent of the A register is greater than the exponent of the B register (W74L) and the B register mantissa is cleared to zero during scaling (W07L), the A register mantissa and sign are transferred to the B register, thus making the exponents equal. This causes immediate transfer to the W73L • J91L box, where the A register mantissa is transferred to the B register via an addition operation. The Q01F and Q04F flip-flops are set to zero, nullifying the effect of any digits that may have been shifted out of the B register during scaling, i.e., a possible round is prevented.

W74L • W07L' • J91L

The alignment procedure is to normalize the operand with the algebraically largest exponent and to scale the operand with the algebraically smaller exponent. W74L • W07L indicates that the A exponent is greater than the B exponent and the B register mantissa is not zero (the A register mantissa is not zero in order to arrive at this point).

Normalize A

If the 13th octade of the A register is equal to zero (A13L), the A register mantissa is normalized by shifting the A register mantissa left by octades and at the same time decrementing the exponent of A. If the exponents become equal (W73L), control is transferred to the W73L • J91L box.

Scale B

When the 13th octade of the A register mantissa becomes not zero (A13L'), the scaling of the B register commences by shifting the B register mantissa right by octades and at the same time incrementing the exponent of B. The scaling of any non-zero digits out of the B register is into the X register in an 8's complement form. The X register is shifted right by octades as a part of scaling the B register.

If any non-zero digit is shifted out of the B register as indicated by the first octade of the B register not being equal to zero (B01L'), the logical flip-flop, Q04F, is set to remember the scaling of a non-zero digit out of the B register. The first non-zero digit shifted out of the B register, (indicated by Q04F being reset), is shifted to the X register in an 8's complement form. Subsequent digits (Q04F set) shifted out of the B register are transferred to the X register in 7's complement form.

Q01F to 1 or 0

If the internal operation is addition (W99L), the status of B03F is checked as the first octade position is shifted to the X register. This is to determine whether a round will be required on the forthcoming addition. If a digit of less than 4 is shifted out of the B register, rounding is not required. If a digit greater than or equal to 4 is shifted out of the B register, rounding is required. Only the last digit that is shifted out of the B register has any significance in determining whether a round is required. As each digit is shifted out of the B register; if B03F is off, Q01F is reset, if B03F is on, Q01F is set. Thus, the last digit shifted out determines the final setting of Q01F.

If the internal operation is subtraction (W98L) and a non-zero digit has been shifted out of the B register as indicated by the level B01L' being true, Q01F is set to the one's state. Q01F is set at this time, if a non-zero digit has been shifted out of the B register, to allow it to be reset when one of the operands is complemented for the forthcoming subtraction. When performing complement addition, a pre-carry is normally required to convert the seven's complement to an eight's complement. However, if any digit is shifted out of the B register, it is shifted out in an eight's complement form. Because the X register contents are an extension of the mantissa contained in the B register, the pre-carry has effectively been added in.

W75L • J91L • W07L'

If, on entry, the exponent of B is greater than the exponent of A and the B mantissa is not normalized (B13L), the B mantissa is normalized and the exponent is decremented to tally the left octade shifts of the B register mantissa. The X register contents are shifted left by octades with the 13th octade of the X register shifted to the B register's first octade position. On entry to the operator the X register is in a cleared status. The only thing accomplished is the shifting of zeros into the low order end of the B register as the B register is normalized. The logic of shifting the X register at this time is redundant (common logic utilized in double precision add-subtract).

If, during the normalizing of the B register, the exponents become equal, control is transferred to the W73L • J91L box. If the B register becomes normalized (B13L') before the exponents become equal, the contents of the A and B registers are interchanged so that the original contents of the A register may be scaled in an attempt to equalize exponents. The logical flip-flop, Q06F, is set to remember that the operands have been interchanged. Once the operands have been interchanged, the exponent of A becomes greater than the exponent of B and control is transferred to the W74L • W07L' • J91L box where scaling of the B register (original contents of A) commences.

W73L • J91L

Where the exponents of the A and B registers become equal, the normal add-subtract cycle commences.

J to 5, Complement Q01F

If the internal operation is subtract (W98L), the J register is set to 5 when the complement addition takes place. The operation is subtract if the operator is add with signs unequal or subtract with signs equal. On entry to this W73L • J91L box the status of Q01F may be either on or off. If Q01F is on, something has been shifted to the X register; therefore, a pre-carry is not required to develop an 8's complement (it already is in 8's complement form). Q01F is reset in this case. If Q01F is off, it is turned on to develop an 8's complement for the forthcoming subtraction.

Complement the Smallest Operand

In order to obtain a true answer (as opposed to a complement answer) in complement addition, the smaller operand of the two must be the one complemented. In determining which mantissa is smallest, logic is only provided to interrogate the 13th octade position of the A and B registers. If W02L is true, the 13th octade of A is less than the 13th octade of B; therefore 39 bits of the A register mantissa are complemented. If W02L' is true, the 13th octade of B is less than or equal to the 13th octade of A. Therefore, the 39 bits of the B register mantissa are complemented and transferred to the B register. The transfer from A to B is required because the largest operand (presently in the A register) contains the resultant sign if the operator is add or contains the complement of the resultant sign if the operator is subtract. If the status of the 13th octade positions of the A and B registers does reflect the true magnitude of the A and B registers, the incorrect mantissa will be complemented, thereby resulting in a complement answer. The answer will require decomplementing.

Q02F to 1

If the operator is Subtract and the operands have not been interchanged and the 13th octade of B is less than the 13th octade of A (SU1L • Q06F' • W02L'), the logical flip-flop, Q02F, is set to remember that the resultant sign

must be complemented. Q02F is utilized at this time because the contents of the A register is being transferred to the B register at this clock pulse. This is explained in the previous paragraph.

ADD B + A, Clear A

If the internal operation is addition (W99L), the contents of the A and B register mantissas are added via the 42 bit mantissa adder. The operation is Add if the operator is Add with equal signs or Subtract with unequal signs. The A register is cleared as its contents are no longer valid.

Complement B47F

If the operator is subtract and the operands have been interchanged (SU1L • Q06F), the sign bit, B47F, is complemented. If the internal operation is addition resulting from a Subtract operator (SU1L), the signs must have been unequal. The basic rule of algebraic subtraction states that the sign of the subtrahend is complemented and the rules of algebraic addition are then followed. Therefore, if the operator is Subtract and the operands have been interchanged, the sign of the B register must be complemented (the A register contained the subtrahend).

J to 4

If the operation is Add and an overflow carry results from the 13th octade position (W13L), the results contained in the B register must be scaled one octade position and possibly rounded. The J register is set to 4 where the scaling occurs.

Exit

If the operation is addition and an overflow carry does not result (W13L'), the operation is complete. The syllable execute complete level is true to terminate the operator.

J = 4

Entry to this J count only occurs if the operation is add and an overflow carry resulted. Therefore, the B register mantissa is shifted one octade position to the right and the exponent is incremented to tally the shift. If a digit of 4 or more is shifted out of the B register during scaling, rounding of the B register is required if Q01F was off during add; if Q01F was on during add, rounding occurs if more than four were shifted out at this time. Q01F is set so a one may be added to the B register mantissa at J = 15. If a digit of 3 or less (B03F' + B02F' • B01F') is shifted out, rounding is not required and Q01F is reset. The J register is set to 15 for a possible round and to terminate the operator.

J = 5

Entry to this J register count only occurs if the internal operation is subtract. The contents of the A and B register mantissas are added plus Q01F via the 42 bit mantissa adder. The Q01F flip-flop may or may not have been set; in either case, it is unconditionally reset. The A register is cleared as it no longer contains valid information. If Q02F is set, the sign bit, B47F, is complemented to establish the correct resultant sign.

J to 7

If in the performance of complement addition an overflow carry does not result (W13L'), the answer is in complement form with the incorrect sign. The wrong (largest) operand was complemented. Control is transferred to J = 7 where the resultant sum and sign are decomplemented.

J to 8

If Q04F, is set, a number of things are known about the operands in the A and B registers:

1. The A register contents must have been normalized.
2. The 13th octade of the A register contained a digit.
3. Something has been shifted out of the B register.
4. The 13th octade of the B register contained a zero.
5. The B register mantissa was complemented and placed in the A register.
6. The A register contents were transferred to the B register.

If in addition to Q04F being set, the two high order bits of the B register equal zero (B39F' • B38F'), the original contents of the A register 13th octade position must have been equal to 1. Therefore, after complementing and the transfer, the B37F bit is set. Also, the 13th octade digit position of the present A register contents equal 7 (complement of zero). If the complement addition results in no carry developed from the 12th octade position of the B register (W36C'), it is known that the 13th octade of the result equals zero and scaling of the B register is required. The following example illustrates the aforementioned conditions relative to the 13th octade position of the A and B register:

```
A = 1   7  - complement of zero
B = 0   1  - A to B transfer
       10  - sum with overflow carry
```

J to 15

With Q04F set and the previously mentioned points kept in mind, if a carry results from the 12th octade position of the B register, the 13th octade position of the B register cannot equal zero. Also, if 107L is true

(B39F + B38F), the 13th octade position of the B register cannot equal zero. Therefore, normalizing is not required and control is transferred to J = 15 to terminate the operator. If X39F is true, rounding of the results is required.

Exit

If 107L is true (B39F + B38F), normalizing is not required. If the 39th bit of the X register is off (X39F'), rounding is not required. Therefore, the operator is terminated and the syllable execute complete level is true.

Q01F to 1

The logic W36C' • B39F' • B38F' • X36F indicates that the 13th octade of the B register will equal zero and normalizing of the B register is required. X36F indicates that rounding will be required after one left octade shift of the B and X registers. Q01F is set to facilitate the adding of a one to the resultant sum.

Exit

If Q04F is reset, nothing has been scaled into the X register. Rounding of the B register is not required. If the 13th octade of the A and B registers do not equal each other (W03L') it is not possible for the resultant sum of the complement addition to equal all zeros and the operator is terminated.

J to 15

If the 13th octades of the A and B register are equal (W03L), it is possible that the resultant sum of the complement addition may equal all zeros. Control is transferred to J = 15 where the B register mantissa may be checked for all zeros. W13L indicates a true answer and decomplementing is not required.

J = 7

Entry to this J count only occurs if decomplementing is required as indicated by a lack of overflow carry during complement addition. The B register mantissa is complemented and transferred to the A register. The B register is also cleared to zero. The sign of the results is complemented to the correct value. Q01F is set to one and the J register is set to 15 where the one is added to the 7's complement of the results to develop a true answer.

J = 8

Entry to this J count only occurs if normalizing of the B register is required. During subtraction, the X register is considered as an extension of the B register. It is effectively subtracted from zero and must be considered a part of the results. Therefore, the B and X registers are shifted left one octade position (the 13th octade of the X register is transferred to the first octade of the B register), and the exponent is decremented to tally the left shift. The J register is set to 15 where a possible round is added to the results.

J = 15

The final add for either rounding or decomplementing takes place at this J count. The syllable execute complete level is true to terminate the operator.

Set Overflow Interrupt

If the exponent is greater than +63, M11F will be set, which causes the exponent overflow bit to be set in the interrupt register.

Clear the B Register

If the B register mantissa is zero (possible during internal subtraction), and the logical flip-flop, Q01F, is reset (no rounding or decomplementing is required), the B register is cleared to all zeros.

3.18 SINGLE PRECISION MULTIPLY

The operands in the A and B registers are algebraically multiplied and the product left in the B register. For all conditions, at the end of the operation, the A register is set to empty, the B register is set to full and the B register flag bit is set to zero.

If the mantissa of either operand is zero, the B register is set to all zeros.

If the exponents of the operands are both zero, the 26-digit product of the mantissas is computed. If the 13 most significant digits of the product are all zero, the 13 least significant digits are the mantissa of the result and the exponent of the result is zero. If the 13 most significant digits of the product are not all zero, the product is normalized and rounded to 13 digits. A mantissa of thirteen sevens is not rounded.

If the exponents of the operands are not both zero, the operands are normalized. The 26 digit product of the mantissas is computed, normalized and rounded to 13 digits.

If the result has an exponent greater than +63 or less than -63 the exponent overflow bit, or exponent underflow bit respectively, is set in the Interrupt register. The B register contains the correct mantissa, mantissa sign and exponent sign. The magnitude of the correct exponent is contained in the exponent field of the B register modulo 64.

SUMMARY OF OPERATION

In Single Precision Multiply, the operand in the A register is considered the multiplicand and the operand in the B register is considered the multiplier. During multiplication it is convenient to consider the mantissa and its sign as an algebraic number. The same is true for the exponent and its sign. This results in the following rules for multiplication:

1. The mantissa of the result is the algebraic product of the mantissas.
2. The exponent of the result is the sum of the exponents.

Since, in the B 5500, both operands are normalized before multiplication is initiated (if both exponents are not equal to zero), the result is either a 25 or 26 digit number in which only the 13 most significant digits are retained. For example, when multiplying two N-digit numbers, the result may be either 2N or 2N-1 digits in length:

```
  543 - N = 3
x 111 - N = 3
-------
  543
 543
543
-------
62473 - 2N-1 = 5
```

```
  543 - N = 3
x 211 - N = 3
-------
  543
 543
1036
-------
136773 - 2N = 6
```

Once normalizing of the two operands is completed (if both exponents equal zero, normalizing does not occur), the multiplier operand in the B register is transferred to the X register while the B register is cleared to zero. Subsequently the exponent add and multiply operation commences. During multiply, the partial product is developed in the B register. And as each digit of the multiplier is utilized to develop the product, it is shifted out of the X register and, at the same time, the partial result is shifted to the X register. This results in the final product (before rounding) being contained in the B and X registers. Because only the 13 most significant digits of the product are retained, an octal 15 (13 in decimal) must be added to the exponent result to compensate for the left shift in the octal point.

Example:

A	+	+	03	0070000000001	
B	+	+	12	0100000000002	
A	+	+	01	7000000000100	
B	+	+	11	1000000000020	
B,X	+	+	12	0700000000026	0000000002000 - un-normalized result
B,X	+	+	11	7000000000260	0000000020000 - normalized result
B	+	+	26	7000000000260	- Final result, with adjusted exponent after normalizing, is that the 13 most significant digits are retained and the 13 least significant digits are lost. This requires a movement of the octal point accomplished by adding an octal 15 (13 in decimal) to the exponent.

Integer Multiplication

If the exponents of both operands are zero before normalization, the operands are not normalized and the product of the mantissas is computed. If the 13 most significant digits of the product are zero, the 13 least significant digits are the result, with the exponent of the result being equal to zero.

A	+	+	00	0000001230000	
B	+	+	00	0000000300000	
B,X	+	+	00	0000000000000	0371000000000. octal point
B	+	+	00	0371000000000.	- Final result

Multiplication Procedure

In the performance of multiplication, it is the normal procedure to add the multiplicand to the partial product the number of times as specified by each digit of the multiplier operand. As each digit of the multiplier operand is utilized, a right shift of the partial product occurs.

In the B 5500, however, in order to expedite the multiply operation, both addition and subtraction are utilized in the development of the product. If the product (or partial product) can be developed faster by subtraction, that is the method utilized. However, if addition is faster, it is utilized. For example, (remember this is octal arithmetic) if we wish to multiply by the digit 2, it is a simple matter to add the multiplicand to the partial product twice. But, if we wish to multiply by the digit 6, it is faster to subtract 2 and then 8, rather than to add the multiplicand 6 times.

In the addition and complementary addition (for subtraction) of the multiplicand to the partial product, use is made of the fact that a binary left shift of the multiplicand doubles the value of that number. And that two left shifts will increase the value of the number by four. In the single precision multiply flow, the current value of the multiplicand in the A register is denoted by:

A - multiplicand is in its original form
A' - multiplicand is in complementary form
2A - multiplicand has been doubled
2A' - multiplicand is doubled and is in complementary form
4A - multiplicand has been increased fourfold (shifted left twice)
4A' - multiplicand increased fourfold and is in complementary form

With the facility of shifting the multiplicand in a binary fashion, the process of multiplication is simplified. For example, if we wish to multiply by the digit 2, the multiplicand is doubled to 2A and is then added to the partial product once. If the current multiplier digit is 4, the multiplicand may be increased to 4A and then added to the partial product once. As a general rule, in the mechanization of single precision multiply, if the current multiplier digit is either a one, two, three, or four, the multiplicand is added to the partial product. If however, the current multiplier digit is a five, six or seven, the multiplicand is subtracted from the partial product via complement addition. There are some exceptions to this rule; for example, if the current multiplier digit is such to cause a subtract operation and the next multiplier digit is a 3 or 4, the operation will be to continue subtraction rather than to add as in the normal case. This is to speed the overall time to perform the multiply operation. Naturally, if the current multiplier digit is a 0, all that is required is to shift the partial product right one digit position. This statement of operation is slightly simplified but serves to introduce the actual procedure utilized.

FLOW SEQUENCE

The occurrences during single precision multiply relative to the J register settings are as follows:

J = 0 Start

1. If either A or B is unoccupied; initiate loading from the stack.
2. If either operand is zero, make answer zero.

3. If either operand is a non-integer and either operand is un-normalized; initiate normalization.
4. If both operands are integers, or both operands are normalized; initiate multiply.

J = 1 Obtain Operands from the Stack

J = 2 Normalize Operands

1. Both operands are normalized
2. Initiate multiply

J = 5 First Add Cycle for Multiplier = $\pm$ 3

J = 6 Main Add Cycle

J = 7 Scale Multiplier and Partial Product

J = 8 Normalize Result

1. If result is integer, do not normalize
2. If result is non-integer; normalize

J = 15 Exit

1. Round result
2. Set exponent overflow or underflow interrupt if valid
3. Return to control

Q Flip-Flops

The Q flip-flops are utilized as follows in single precision multiply:

Q01F - +1 for subtraction; indicates that the multiplicand is contained in the A register in complement form.

+1 for rounding of result.

Q02F - Controls exponent add operation.

Q03F - Set when the next multiplier digit (X*) is $\pm$ 0 or $\pm$ 4.

Q05F - Both operands are integers.

Q06F - Used in exponent arithmetic to memorize that the intermediate result has gone negative.

Q08F - Indicates a negative partial product.

M Register

The M Register is utilized as follows:

M (2 and 1) - Multiplicand Counter
The two low order bits of the M register are used as a counter to indicate the value of the multiplicand in the A register.

M = 0; A or A' is presently in the A register
M = 1; 2A or 2A' is presently in the A register
M = 3; 4A or 4A' is presently in the A register

M (6 through 4) - A mantissa extension

M04F has a value of 2^{39}
M05F has a value of 2^{40}
M06F has a value of 2^{41}

M07F - An exponent extension

M07F has a value of 2^6 ($2^6 = 64$)

M (10 through 8) - B mantissa extension

M08F has a value of 2^{39}
M09F has a value of 2^{40}
M10F has a value of 2^{41}

M11F - B exponent extension

M11F has a value of 2^6

M (15 through 12) - Exponent control counter

The four high order bits of the M register are used as a counter to control the exponent add operation.

Special Levels

W91L' = Q03F (Q01F' • X03F • X02F • X01F + Q01F • X03F' • X02F' • X01F')

W91L = same input as for W91L' switched.
W91L' when true indicates that: The multiplier digit just completed is +0 or +4 and the next multiplier digit is -1 or that the multiplier digit just completed is -0 or -4 and the next multiplier digit is +1.

W86L = X02F • X01F • Q01F + X02F' • X01F' • Q01F'

W86L when true indicates that the multiplier digit (X*) is either plus or minus, zero or four (X* = ± 0 or ± 4)

W86L'= same input as for W86L' switched

W86L' when true indicates that the multiplier digit (X*) is either plus or minus, one, two or three (X* = ± 1 or ± 2 or ± 3) and is <u>not</u> plus or minus zero or four (X* ≠ ± 0 or ± 4)

W87L = X03F' • X02F • X01F • Q01F'
+ X03F' • X02F • X01F' • Q01F
+ X03F • X02F' • X01F' • Q01F
+ X03F • X02F' • X01F • Q01F'

W88L = X03F • X02F (X01F' + Q01F')
+ X03F' • X02F' (X01F + X01F)
+ X01F • Q01F (X03F' + X02F')
+ X01F' • Q01F' (X03F + X02F)

W88L when true indicates that the multiplier digit (X*) is either plus or minus one, two or four (X* = ± 1 or ± 2 or ± 4)

Multiplicand Shift Levels

Upon interrogation of the current multiplier digit, it may be necessary to adjust the current value of the multiplicand in the A register. If it is necessary to increase the value of the multiplicand, the A register left shift level (ALSL) will be true. If it is necessary to decrease the value of the multiplicand, the A register right shift level (ARSL) will be true. And if it is necessary to change from an add operation to a subtract operation, or if it is necessary to change from a subtract operation to an add operation, the complement shift level (ACSL) will be true. The following equations develop the levels of ACSL, ALSL, and ARSL:

- O' - ACSLS1 -- Note that the equation for ACSL is switched.
- I - + M02F
+ X03F • Q01F
+ X03F' • Q01F'
+ M01F' • X01F' • Q01F'
+ M01F' • X01F • Q01F
+ M01F • X01F' • X02F'
+ M01F • X01F • X02F

- O - ALSLD1
- I - + M02F' • M01F' • X01F' • Q01F'
+ M02F' • M01F' • X01F • Q01F
+ M01F • X03F' • X02F • X01F • Q01F
+ M01F • X03F • X02F' • X01F' • Q01F'

```
- 0 -  ARSLD1
- I -  + M01F • Q01F • X01F' • (X03F + X02F')
       + M02F • X03F' • X02F'
       + M02F • X02F • X01F'
       + M02F • X03F • Q01F
       + M02F • X01F • Q01F'
       + M01F • X03F' • X01F • Q01F'
       + M01F • X02F • X01F • Q01F'
```

Multiplicand Adjustment Table

The multiplicand adjustment table illustrates the required adjustment of the multiplicand in the A register as defined by the preceding equations:

X*	X03F	X02F	X01F	Q01F	M = 0 A or A'	M = 1 2A or 2A'	M = 3 4A or 4A'
+0	0	0	0	0	←		→
+1	0	0	0	1	C	→	→
+1	0	0	1	0		→	→
+2	0	0	1	1	←	C	→
+2	0	1	0	0	←		→
+3	0	1	0	1	C	C	→
+3	0	1	1	0		→	→
-4	0	1	1	1	←	←	
+4	1	0	0	0	←	←	
-3	1	0	0	1		→	→
-3	1	0	1	0	C	C	→
-2	1	0	1	1	←		→
-2	1	1	0	0	←	C	→
-1	1	1	0	1		→	→
-1	1	1	1	0	C	→	→
-0	1	1	1	1	←		→

X* = Multiplier digit
→ = Right Shift ARSL
← = Left Shift ALSL
C = Complement ACSL

M = 0 = M02F' • M01F'
M = 1 = M02F' • M01F
M = 3 = M02F • M01F

FIGURE 3.18-1. MULTIPLICAND ADJUSTMENT TABLE

Values of Multiplicand Stored in A

The table below shows the value of the multiplicand for each multiplier value after adjustment according to Figure 3.18-1. It is noted that the multiplier value of +0 will adjust the multiplicand to 2A, and a multiplier value -0 will adjust the multiplicand to 2A'. This has been specified to minimize the adjust time for the next multiplier digit. However, after the multiplication for all multiplier digits is completed, the multiplicand is always returned to +A.

X	+0	+1	+2	+3	+4	-4	-3	-2	-1	-0
Contents of A Register	2A	A	2A	2A or A	4A	4A'	2A' or A'	2A'	A'	2A'

FIGURE 3.18-2. ADJUST MULTIPLICAND VALUES

Algorithm for Adding or Subtracting

Figure 3.18-3 illustrates the required add or subtract cycles according to the values of the multiplicand.

X	FIRST ADD CYCLE	SECOND ADD CYCLE
+0		
+1	B+A → B	
+2	B+2A → B	
+3	B+2A → B or B+A → B	B+A → B or B+2A → B
+4	B+4A → B	
-4	B-4A → B	
-3	B-2A → B or B-A → B	B-A → B or B-2A → B
-2	B-2A → B	
-1	B-A → B	
-0		

FIGURE 3.18-3. ALGORITHM FOR ADDING OR SUBTRACTING

DETAILED DESCRIPTION

J = 0

If either the A and/or B registers are empty on entry to the operator, stack adjustment is initiated to load the A register with the multiplicand and the B register with the multiplier operand. If on entry, either of the registers

is occupied, a memory access is initiated to load the B register. The other operand is transferred to the A register if not already there. If, on entry, both the A and B registers are empty, a memory access is initiated to load the A register. The J register is set to one for continuation of the stack adjustment.

J = 1

If either the A or B register had been empty, on entry to the operator, a memory access would have been initiated. Upon completion of the memory access (MROF), a memory access is initiated to load the B register, if initially both the A and B registers had been empty (indicated by BROF in the reset state). If, however, only one of the registers had been initially unoccupied (indicated by BROF in the set state), a memory access is not required. If a memory access is not required, the J register is set to zero. If a memory access is required, the setting of the J register must await completion of that memory access (the second MROF occurring at J = 1).

J = 0 • AROF • BROF

If either the A or B mantissa is equal to zero (W06L + W07L), the resultant product is zero. Therefore, both the A and B registers are cleared to zero and the J register is set to 15 for termination of the operator.

Indicate Integers

If the exponents of both operands are equal to zero (W71L • W72L), the operands are integers. Q05F is set to indicate that an integer multiply is required.

Normalize Operands

If the mantissa of both operands is not equal to zero (W07L' • W06L') and the exponent of either operand is not equal to zero (W71L' • W72L'), then normalizations of the operands may be required.

Normalize A. If the A register is not normalized (A13L), normalization of the A register is initiated. The J register is set to 2 for continuation of the normalization process and initiation of the multiply operation.

Normalize B. If the B register is not normalized (B13L), normalization of the B register is initiated. Normalization of the B register involves shifting of the X register which in this case is a redundant operation (utilization is made of existing logic). However, shifting of the X13 octade into the first octade of the B register, performs the function of inserting zeros. The J register is set to 2 for continuation of the normalization process and the initiation of the multiply operation.

X13 octade into the first octade of the B register, performs the function of inserting zeros. The J register is set to 2 for continuation of the normalization process and the initiation of the multiply operation.

If the exponents of A and B equal zero and neither mantissa equals zero (W71L • W72L • W06L' • W07L'), or both the A and B operands are normalized, exponent arithmetic is initiated by the setting of Q02F to the one state. In conjunction with the initiation of exponent arithmetic, the exponent of the B register is incremented by one (M [11] B [46 through 40] + 1) as part of the required exponent adjustment. (See write-up on exponent arithmetic). The multiplier operand (mantissa) is transferred to the X register while the B register mantissa (bits 39 through 1) is cleared to zero. The J register is set to 7 to start the main multiplication operation.

Complement B47F

If the sign of the operand in the A register is negative (A47F), the sign of the resultant product is the complement of the sign of the operand in the B register. Therefore, the sign of the operand in the B register is complemented. The rules of multiplication state that if the signs of the operands are equal, the result is positive. And if the signs of the operands are unequal, the sign of the result is negative.

For example:

A	B	
-	-	Complement B sign to +
-	+	Complement B sign to -
+	-	Sign of B OK
+	+	Sign of B OK

Double A

If the least significant multiplier digit is even (B01F'), the multiplicand is doubled to a value of 2A. The doubling of the multiplicand is recorded by the multiplicand counter being incremented by one (M [3 through 1] + 1).

AROF and B48F to Zero

The A register is unconditionally marked as unoccupied, as it will not contain valid information, upon completion of the operator. The B register flag bit is also unconditionally reset to insure that the resultant product is marked as an operand.

J = 2

If normalization of either operand is required, transfer to J = 2 is effective to complete the normalization process. The logical function at J = 2 is the same as at (J = 2 • AROF • BROF) with the exception of the logic relative to an integer operation. The setting of the J register to 2, at J = 2, is redundant and results from the use of common logic.

J = 5

Entry to J = 5 is from J = 7 and only occurs when the current multiplier digit is either a plus or minus three (± 3). The first cycle or two, required for a plus or minus three multiplier digit, occurs at J = 5. The second cycle occurs at J = 6. Unconditionally, the multiplicand in the A register is added to the partial product in the B register and the result placed in the B register.

Halve A

If the least significant bit of the multiplicand counter is on (M01F), thereby indicating that the multiplicand has been doubled to either 2A or 2A', the multiplicand is halved by shifting the A register mantissa and its extension (M [6 through 4]) right one binary bit position. The multiplicand counter is decreased by one, thus indicating that the multiplicand is in its original form of A or A'.

Double A

If the least significant bit of the multiplicand counter is off (M01F'), indicating that the multiplicand is in its original state of 2A or is in the complement of its original state of 2A', the multiplicand is doubled by shifting the A register mantissa and its extension left one binary bit position. The multiplicand counter is incremented to record the new value of the multiplicand (2A or 2A').

A01F to 1 or 0

If Q01F is set (indicating that the multiplicand is in complementary form), the low order bit of the A register (A01F) is set to provide a valid complement when the multiplicand is doubled. If, however, Q01F is reset (indicating that the multiplicand is in its original form), the low order bit of the A register is reset, when the multiplicand is shifted left for doubling of the multiplicand.

Q08F to 1

If Q01F is on (complementary addition is being performed) and there is no carry for the most significant digit position (W14L'), the indication is that the partial product in the B register is in complementary form (the

partial product is less than zero). Therefore, Q08F is set to record this fact so that (at J = 7) when the partial product is shifted right, a 7 may be set into the most significant digit position of the partial product, maintaining the partial product in complementary form.

Q08F to 0

If Q01F is reset (regular addition is being performed) and there is a carry from the most significant digit position (W14L), the indication is that the partial product in the B register is in a true form (the partial product is equal to or greater than zero). Therefore, Q08F is reset to record this fact so that when the partial product is shifted right (at J = 7), a zero may be set into the most significant digit position of the partial product, thus maintaining the partial product as a true number.

J = 6

Entry to J = 6 is from J = 7 when the current multiplier digit is either plus or minus 1, 2 or 4 and is from J = 5 for performance of the second multiply cycle when the multiplier digit is plus or minus three ($\pm$ 3). Unconditionally, the multiplicand is added to the partial product and the result placed in the B register.

J to 8

If the N register has been counted up to 14 (N14L), it indicates that all of the multiplier digits have been utilized. A final addition cycle is performed, to convert the partial product (which is below zero) to a correct product above zero. The J register is set to 8 for a final normalization of the resultant product.

Halve A

At the time the multiplicand is being added to the partial product as a result of the current multiplier digit, the next multiplier digit is interrogated and appropriate adjustment of the multiplicand occurs at this time to minimize the time utilized for adjustment of the multiplicand. The next multiplier digit is presently in the least significant digit position of the X register; therefore, interrogation of the A register Right Shift Level (ARSL) will specify if the next multiplier digit requires halving of the multiplicand. For development of the ARSL, see Multiplicand Shift Levels in the Summary of Operation. The term N14L' indicates that this is not the last addition cycle.

Double A

If interrogation of the next multiplier digit indicates that the multiplicand should be doubled (ALSL), and this is not the last addition cycle (N14L'), the A register is shifted left one binary bit position. For development of ALSL, see Multiplicand Shift Levels, Summary of Operation.

A01F to 1 or 0

If the multiplicand is being doubled (ALSL is true) and the multiplicand is in complementary form (Q01F), the least significant bit of the A register is set to provide a valid doubled complement. However, if the multiplicand is in its original form, the least significant bit of the A register must be reset to provide a valid doubled true number. Doubling, in the sense it is used at this time (J = 6), implies; going from A to 2A or going from 2A to 4A or the complement thereof.

Q08F to 1 or 0

Q08F is either set to one or zero for the same reasons as specified at J = 5. If the state of Q08F does not require altering, it remains in its current state.

J = 7

The first entry to this J register setting is from either J = 0 or J = 2, when multiply is initiated and subsequent entries to J = 7 are from J = 6 after a multiply cycle has been completed. The basic function at J = 7 is to interrogate the current multiplier digit to determine what changes must be made to the value of the multiplicand and to shift the current multiplier digit out of the X register, while simultaneously shifting the partial result to the X register. The first time into J = 7, from either J = 0 or 2, the current multiplier digit is the least significant digit of the multiplier and occupies the least significant position of the X register. Subsequent transfers to J = 7 will find succeedingly more significant multiplier digits in the least significant digit position of the X register.

Basically, there are three primary control levels at this J = 7 state. These are W91L, W91L' and ARSL + N13L • Q01F':

W91L is equal to Q03F' + Q01F' (X03F' + X02F' + X01F') + Q01F (X03F + X02F + X01F) and indicates that an extra adjustment of the multiplicand is not required.

W91L' is equal to W91L switched, and indicates that an extra adjustment of the multiplicand is required. Table 1 on the flow chart illustrates the condition when an extra shift is required.

ARSL + N13L • Q01F' is used to return the multiplicand to its original configuration upon completion of the multiply operation. (N13L • Q01F'), in addition to the normal function of gating the right shift of the multiplicand in the A register (ARSL).

Q03F to 0

If the level W91L' is true, the logical flip-flop Q03F is reset. Q03F being reset causes W91L to go true, thus allowing the normal functions of J = 7 to occur. If W91L' is true, an extra adjustment of the multiplicand is taking place, thus Q03F being reset allows only one clock pulse for the extra adjustment of the multiplicand.

Shift X Register and Increment N Register

Whenever W91L is true at J = 7, the X register is shifted right one octade position, thereby shifting the current multiplier digit out of the X registers, and simultaneously shifting the next multiplier digit into the least significant digit position of the X register. At the same clock pulse at which this shifting occurs, the current multiplier digit is being interrogated and action taken accordingly in a logical box below. The N register is incremented to keep track of the number of multiplier digits utilized.

X [39 through 37] to 0

If the N register is equal to zero (NEZL), the most significant digit position of the X register is cleared to zero at the time the first multiplier digit is shifted out of the X register. This insertion of a zero happens only during the first shift of the X register and functions as a flag between the multiplier digits and the digits of the product that will be shifted to the X register. This zero will gate a multiplier value of +1, during interrogation of a 14th multiplier digit, which will only occur if, after utilization of the 13th multiplier digit, the result in the B register is in complement form, thus requiring an additional add cycle to convert the product to a true number.

Shift Partial Product

If the N register is not equal to zero (NEZL'), the first multiplier digit has already been utilized. The B register and its extension is shifted right one octade position, in conjunction with a digit of the result being shifted to the most significant digit position of the X register. Simultaneously, the X register is shifted right one octade position, thereby shifting the current multiplier digit out of the X register and making room for the partial result presently being shifted to the X register. The right shift of the partial product has the effect of causing the next addition, to the partial product, to be added one digit position to the left, as done in normal arithmetic.

M [10 through 8] to 7

If Q08F is set, it indicates that the partial product in the B register is in a complementary form; therefore, when the partial product is shifted right one octade position, a 7 is inserted into the extension of the B register mantissa to maintain the partial product in complementary form.

M [10 through 8] to 0

If Q08F is reset, it indicates that the partial product is a true number (not a complement); therefore, when the partial product is shifted right one octade position, a 0 is inserted into the extension of the B register mantissa to maintain the partial product as a true number.

J to 8

If the N register is equal to 13 (N13L) and Q01F is reset (Q01F'), all 13 multiplier digits have been utilized and the resultant product in the B register is a true number; therefore, the J register is set to 8 for final normalization of the product.

Complement A

If the A register Complement Shift Level is true (ACSL), when W91L is also true, the A register and its extension (multiplicand) are complemented. The ACSL is developed from the value of the current multiplier digit and the present status of the multiplicand. See the Multiplicand Shift Levels in the Summary of Operation.

Q03F to 0

If W86L' is true, the current multiplier digit (X*) is not a plus or minus zero or four ($X^* \neq \pm 0$ or ± 4); therefore, Q03F is reset. The logic, which had set Q03F, was developed from a previous multiplier digit of ± 0 or ± 4. Q03F was set in anticipation of an extra shift, in conjunction with the current multiplier digit. Thus, during interrogation of the current multiplier digit, with W86L' being true, an extra shift of the multiplicand is not required. If an extra shift had been required, Q03F would be reset via W91L'.

J to 5

If W87L is true, that is, if the current multiplier digit develops a multiplier value of plus or minus three ($X^* = \pm 3$), the J register is set to 5 for execution of the first of two required addition cycles.

Q03F to 1

If W86L is true, the current multiplier digit has developed a multiplier value of plus or minus zero or four ($X^* = \pm 0$ or ± 4). In this case, Q03F is set to allow the W91L' to go true. Q03F performs the function of remembering that the current multiplier digit has a value of ± 0 or ± 4, so that during interrogation of the next multiplier digit it can be determined whether an extra shift is required. If, during interrogation of the next multiplier digit, it is determined that an extra shift of the multiplicand is required, W91L' will go true to allow the extra required shift.

J to 6

If W88L is true, the current multiplier digit has developed a multiplier value of ± 1, ± 2 or ± 4; therefore, the J register is set to 6 for execution of the addition cycle.

Double A

If the A register Left Shift Level (ALSL) is true, the A register and its extension are shifted left one binary bit position. This will double the multiplicand to either 2A or 4A or the complement thereof. The multiplicand counter is incremented to record the new value of the multiplicand.

A01F to 1 or 0

If the multiplicand is in complementary form (Q01F is set), A01F must be turned on when the multiplicand is doubled. If the multiplicand is in the form of a true number, A01F must be turned off when the multiplicand is doubled.

Halve A

If the A register Right Shift Level is true (ARSL) or the N register is equal to 13, and the product in the B register is not in a complementary form (N13L • Q01F'), the A register and its extension are shifted to the right one binary bit position. The multiplicand counter is counted down to record the new value of the multiplicand. N13L • Q01F' indicates that all multiplier digits have been utilized in the development of the product and that portion of the product, in the B register, is in the form of a true number and does not require complementing. The multiplicand is always returned to its original configuration upon completion of the multiply and is only important if the multiply operation is entered via an operator other than multiply.

J = 8 • Q02F'

Entry to this J register setting is from J = 7, when all 13 multiplier digits have been utilized, and decomplementing is not required or from J = 6, when decomplementing was required, in which case N was equal to 14. Q02F must be reset to indicate that the exponent add operation is complemented.

Normalize B,X

If the full product in the B and X registers is not normalized (B13L), and this is not an integer multiply operation (Q05F'), the product is normalized by shifting the B and X registers left by octades until they are normalized. The exponent is decremented for each left shift.

Result is Integer

If the multiply is an integer operation (Q05F), and the mantissa of the B register is equal to zero (W07L), the contents of the X register are transferred to the B register. The exponents of the B register and the M register are cleared to zero.

J to 15

With either an integer multiply operation, and the B register mantissa equal to zero (Q05F • W07L), or a non-integer multiply, with the B register normalized (B13L'), the J register is set to 15 for a possible round and termination of the operation.

Q01F to 1

If, or when, the B register is normalized, and there is a digit in the most significant octade position of the X register that is equal to or greater than 4 (B13L' • X39F), Q01F is set, so a one may be added to the product during rounding at J = 15.

Q05F and A register to Zero

Unconditionally reset Q05F and clear the A register in preparation for the final add of the round operation.

J = 15

Exit

The syllable execute complete level is true to terminate the operator.

B + A to B

If an overflow carry will not result from the rounding addition (W13L'), the addition takes place. Only if the product is equal to all 7's will an overflow carry ever result from the addition of a one; in which case, W13L is true to prevent the rounding operation. If Q01F is reset, the addition cycle is redundant.

B Register to Zero

If the B register mantissa is equal to zero (W07L), the product is zero; therefore, the B register is cleared to zero in case the exponent is other than zero.

Exponent Underflow; Interrupt to 1

If the exponent add operation, or the process of normalization, resulted in an exponent underflow as indicated by M11F, and the sign of the exponent is negative (B46F), an exponent underflow is set into the interrupt address register by the setting of U18F and U16F.

Exponent Overflow; Interrupt to 1

If the exponent add operation resulted in an exponent overflow (M11F), and the sign of the exponent is positive (B46F'), an exponent overflow is set into the interrupt register by the setting of U15F, U16F and U18F.

3.19 SINGLE PRECISION DIVIDE DV1L-1001

The operand in the B register is algebraically divided by the operand in the A register and the quotient is left in the B register. For all conditions, after the operation, the A register is set to empty, the B register is set to full and the B register flag bit is set to zero.

If the mantissa of the B register is zero, the B register is set to all zeros. If the mantissa of the A register is zero, the divide by zero bit in the Interrupt register is set. In either case, the operation is terminated.

If the mantissa of neither operand is zero, both operands are normalized and the operand in the B register is divided by the operand in the A register. Fourteen significant quotient digits are developed. The quotient is rounded to thirteen significant digits and left in the B register.

If the result has an exponent greater than +63 or less than -63, the exponent overflow bit or exponent underflow bit, respectively, is set in the Interrupt register. The B register contains the correct mantissa, mantissa sign and exponent sign. The magnitude of the correct exponent is contained in the exponent field of the B register modulo 64.

SUMMARY OF OPERATION

In Single Precision Divide, the operand in the A register is considered the divisor and the operand in the B register is considered the dividend. Since both operands are normalized before division is initiated, the quotient can have no more than one significant digit to the left of the octal point. Therefore, either a 14 or 15 (12 or 13 in decimal) must be subtracted from the exponent of the result.

The existance of a significant digit to the left of the octal point is illustrated in the following simplified example:

$$\frac{472.}{200.} = 2.35 \text{ (octal)}$$ ← Significant Digit

$$\frac{470.}{600.} = 0.640 \text{ (octal)}$$ ← Non-Significant Digit

The following example illustrates the necessary adjustment of the exponent to obtain the correct exponent value in the final resultant quotient:

NOTE

An octal 14 is subtracted from the resultant exponent value of 42. (42 - 14 = 26)

A	+	-	14	2000000000000.
B	-	+	26	4270000000000.
B	-	+	42	2350000000000.
B	-	+	26	2350000000000. ← octal point

Flow Sequence

The occurrences during single precision divide relative to the J register settings are as follows:

J = 0 and J = 1 - Stack Adjustment
J = 0 • AROF • BROF

1. Check for either operand being equal to zero
2. Normalize operands if required
3. Initiate divide operation
4. Initiate exponent arithmetic
5. Set sign of result

J = 2

1. Continue normalization of operands
2. Initiate divide operation
3. Initiate exponent arithmetic
4. Set sign of result

J = 5

1. Subtract divisor from dividend (remainder)
2. Increment quotient
3. Decomplement divisor

J = 6

1. Add divisor to partial dividend (remainder)
2. Decrement quotient
3. Complement divisor

J = 7

1. If quotient is unnormalized; shift and initiate next divide subcycle
2. If quotient is normalized; initiate termination

J = 15

1. Round and terminate

QUOTIENT PREDICTION TABLE

The process of division is basically similar to the usual principle of repetive subtraction. In the B 5500 both addition and subtraction are utilized in the development of the resultant quotient. The basic division cycle is implemented by logic which roughly approximates the value of the quotient digit prior to its actual formation. As a result of a comparison between the two high order octades of the dividend in the B register with the high order octade of the divisor in the A register, three levels are developed which indicate the status of the next quotient digit:

B/A is equal to or greater than 4
B/A is less than 4
B/A is not equal to zero

These levels are utilized in the formation of the quotient digits to arrive at them in the least number of pulses. Figure 3.19-1 shows the Quotient Prediction Table.

DIVIDE CYCLE

Once both operands are normalized, the divide operation starts out in subtraction. The quotient digit is increased as long as the remainder does not go below zero. Once it becomes apparent that another subtraction will cause the remainder to go below zero, (that quotient digit has been formed) the quotient predictor logic determines whether to gate the operation into another subtract operation or into an add operation. In either case, the objective is to form the next quotient digit in the least number of arithmetic operations. When the 14th digit has been formed, it is examined and subsequently the final quotient is rounded to 13 digit positions.

The main divide cycle primarily consists of repitive iterations between J register setting of 5, 6 and 7. Figure 3.19-2 illustrates the occurances at J equals 5, 6 and 7 in block diagram form.

An example of the division process is as follows:

$$\frac{4720000000000.}{2000000000000.} = 2350000000000.$$

Dividend	0 4720000000000.	B Register and its extension
Divisor	0 2000000000000.	A Register and its extension
Divisor	7 6000000000000.	In eights complement form
	0 4720000000000.	
Subtract	7 6000000000000.	
	0 2720000000000.	- Increment quotient tally plus
Subtract	7 6000000000000.	one to 1
	0 0720000000000.	- Increment quotient tally plus one to 2
Shift	0 7200000000000.	- Clear tally to zero; record quotient digit of 2
Subtract	7 6000000000000.	
	0 5200000000000.	- Increment quotient tally plus one to 1
Subtract	7 6000000000000.	
	0 3200000000000.	- Increment tally plus one to 2
Subtract	7 6000000000000.	
	0 1200000000000.	- Increment tally plus one to 3
Subtract	7 6000000000000.	
	7 7200000000000.	- Do not increment tally

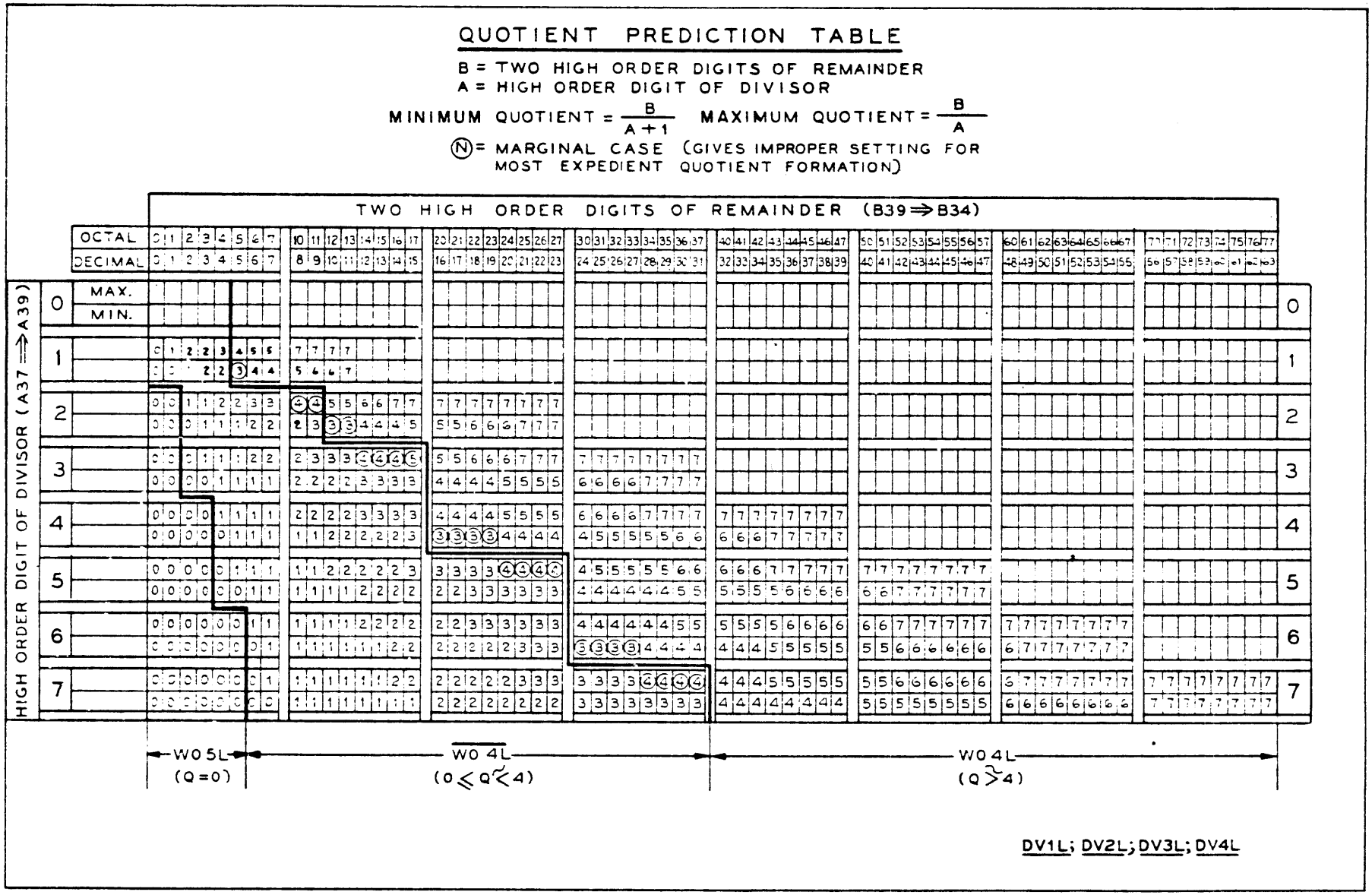


FIGURE 3.19-1. QUOTIENT PREDICTION TABLE

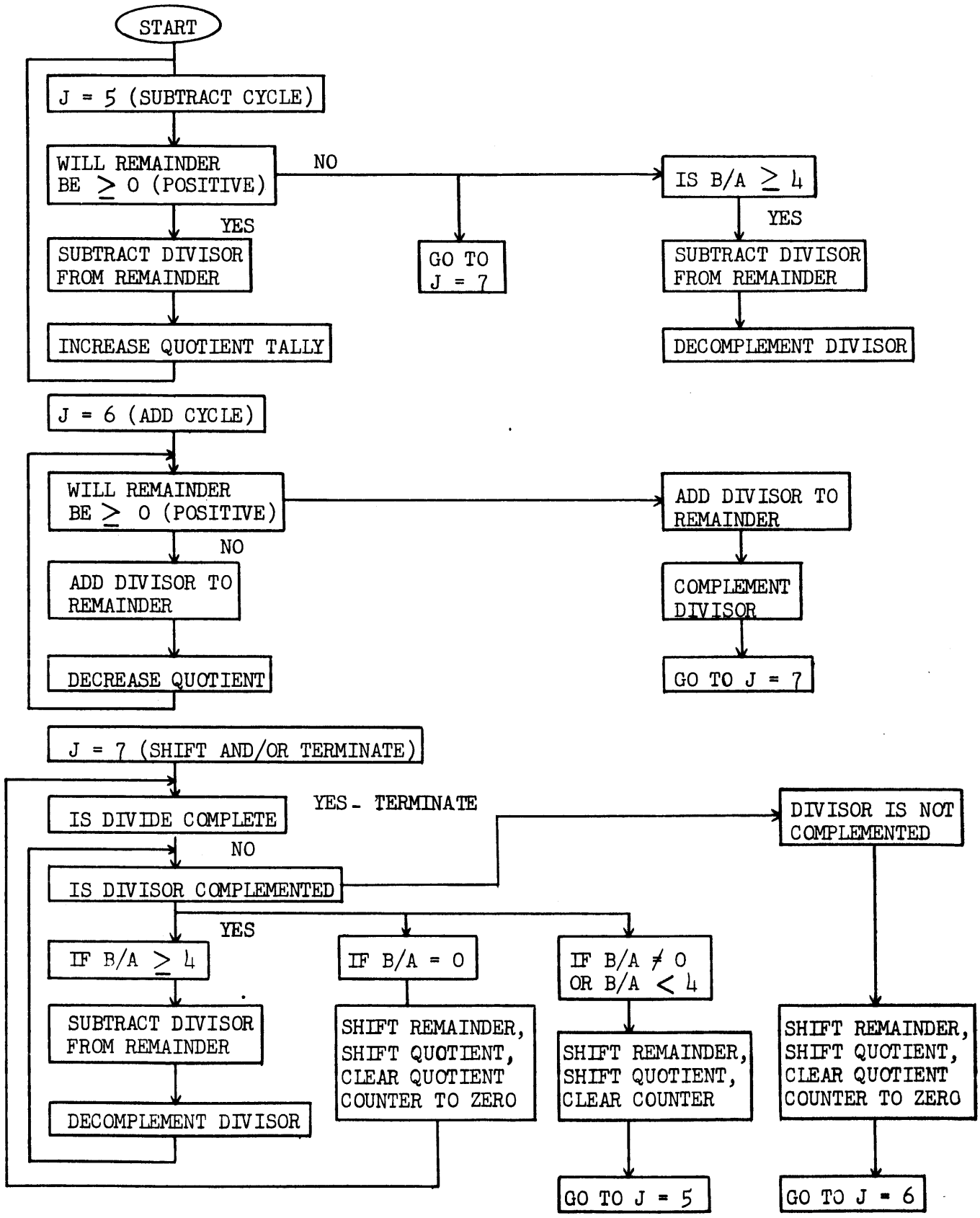


FIGURE 3.19-2. DIVIDE CYCLE BLOCK DIAGRAM

Shift	7 2000000000000.	- Clear tally to zero; record quotient digit of 3
Add	0 2000000000000.	
	7 4000000000000.	- Decrease quotient tally minus one to 7
Add	0 2000000000000.	
	7 6000000000000.	- Decrease quotient tally minus one to 6
Add	0 2000000000000.	
	0 0000000000000.	- Decrease quotient tally minus one to 5
Shift	0 0000000000000.	- Clear tally to zero; record quotient digit of 5
Shift	0 2350000000000.	- Final Quotient

J = 0

If both registers are empty, the memory access will load the A register. If just one of the registers is empty, the memory access will load the B register; the other operand has been transferred to the A register, if not already there. In either case, the J register is set to one for continuation of the stack adjustment.

J = 1

Upon completion of the memory access, the B register is marked as occupied and the stack address counted down. If both registers were initially empty, another memory access is initiated, otherwise, the J register is set to zero.

J = 0 • AROF • BROF

Once the A and B registers are loaded, if not initially loaded, the mantissas of both the A and B registers are checked for zero content. If the A register mantissa (divisor) is equal to zero (W06L), the divide by zero interrupt is set into the Interrupt register by the setting of U15F, U17F and U18F. If the B register mantissa (dividend) is equal to zero (W07L), the A register is cleared to all zeros (the B register is cleared to zeros at J = 15). If either mantissa is equal to all zeros, the J register is set to 15 for termination of the operator.

Normalize Operands

If neither mantissa is equal to zero, both operands are checked to determine if they are normalized. If the A register mantissa is not normalized, the A register mantissa and its extension is shifted left one octade position. At the same time, the J register is set to J = 2 for continuation of the normalization process. If the B register is un-normalized, its mantissa is shifted left one octade position and the J register is set to J = 2 for

continuation of the normalization process. When normalization of the operand in the B register occurs, the X register is also shifted left in conjunction with the left shift of the B register. This results from the utilization of common logic and serves in the simplification of the arithmetic operands. The shifting of the 13th octade of the X register into the low order end of the B register serves the function of inserting zeros into the B register as it is normalized.

Initiate Divide

Once both registers are normalized (A13L' • B13L') the divide operation is initiated. The J register is set to 5 for the start of the subtract operation. The A register and its extension (divisor operand) is complemented in preparation of the forthcoming subtract operation. Q01F is complemented for use as a pre-carry in the conversion of a 7's complement to an 8's complement. Q02F is set to the ones state for initiation of the exponent subtract operation. The sign of the divisor (A46F) is complemented as required in the performance of an algebraic subtraction of the exponents.

Set Sign of Result

If the sign of the operand in the A register is negative (A47F), the sign of the operand in the B register is complemented to the final correct sign. The algebraic rules of division state that: if signs are alike, the sign of the result is positive and if the signs are unlike, the sign of the result is negative. Therefore, only if the sign of the A register operand is negative, is it required to complement the sign of the B register operand. The sign in the B register is the resultant sign.

AROF and B48F to Zero

The A register is unconditionally marked as unoccupied as its contents are not valid upon completion of the divide operation. The B register flag bit is unconditionally reset to insure that the resultant quotient is marked as an operand.

J = 2

If either operand required normalization, the J register would be set to 2 in conjunction with the first left shift of the normalization process. The primary difference between J = 0 and J = 2 is the absence of the logic for the checking of a zero mantissa; otherwise, the basic function is the same.

J = 5

Development of the current quotient digit at J = 5 is via subtraction. Prior to the actual subtraction (complement addition), it is determined if the subtraction will cause the remainder to go negative (below zero). If the remainder will pass through zero as a result of the subtraction, the quotient

digit for that subcycle has already been formed; hence, further subtraction is not required (as far as the current quotient is concerned). The determination, of whether the subtraction will cause the remainder to go negative, is accomplished by the interrogation of an overflow carry from the 13th octade sum digit (W13L) or the extension of the B register mantissa for a not equal to zero condition (W12L). If either W13L or W12L are true, the remainder will continue to be positive (above zero).

Example:

NOTE

Octal subtraction via complement addition.

```
A Reg.    7 5000000000000.
B Reg.    0 4000000000000.
          0 1100000000000.
```

W13L • W12L' = a carry

```
A Reg.    7 5000000000000.
B Reg.    3 1000000000000.
          2.6000000000000.
```

W13L' • W12L = a carry

```
A Reg.    7 5000000000000.
B Reg.    0 1000000000000.
          7.6000000000000.
```

W13L' • W12L' = no carry (remainder negative)

If either W12L or W13L is true, the quotient digit counter (M 3 through 1) is incremented to tally the valid subtraction. If an overflow carry from the 13th octade is not present and the B register extension is equal to zero (W13L' • M10F' • M09F' • M08F'), the J register is set to seven for storage of the just developed quotient digit plus a required shift of the remainder.

When it becomes apparent that the current quotient digit has been formed (the next subtraction will cause a negative remainder), the quotient predictor level W04L ($B/A \geq 4$) is interrogated. If W04L is true, the next quotient digit is probably greater than four; therefore, the most expedient method for obtaining the next quotient digit is via addition rather than subtraction; i.e., if the next quotient digit is a 6 (which would require six subtractions to develop), it is faster to initially subtract 8 (octal) and then add back twice to develop the required 6. If W04L is true, an additional subtraction occurs (making the remainder negative) for subsequent use in the development of the next quotient digit. A subtraction of the remainder at this time is equivalent to a subtraction of eight (ten decimal) because there will be a left shift of the remainder prior to the addition cycle. In conjunction with this subtraction, the divisor in the A register is complemented to facilitate the addition cycles in the formation of the next quotient digit.

J = 6

Entrance to J = 6 results from the assessment that the next quotient digit is 4 or greater. Thus, the operation is to add a true divisor (uncomplemented) to a negative (complemented; below zero) remainder and to count the quotient digit counter down. This may be thought of as a corrective adjustment to the quotient digit counter, based on the number of times the add cycle proves the initial value of the quotient digit counter (zero) to be in error. When the complemented remainder goes positive (above zero), the correction to the value in the quotient digit counter has been achieved and is the correct current quotient digit. The value of the remainder going positive is indicated by a carry from the 13th octade plus a seven (W13L • M08F • M09F • M10F) in the extension of the B register. During the last add (at J = 6), the divisor in the A register and its extension is complemented, with the J register being set to seven.

J = 7

The 13th octade position of the X register is checked to determine if the quotient is normalized. X13L indicates an unnormalized quotient; hence, additional quotient digits remain to be developed.

Shift B and X Left

If the divisor is not complemented, as indicated by Q01F', entry to J = 7 is from J = 5 and a quotient digit has been formed in the quotient counter. It is transferred to the least significant digit position of the X register with the quotient counter being cleared to zero. Simultaneously, the remainder in the B register, and the previously developed quotient digits in the X register, are shifted left one octade position. The N register is incremented to tally the number of shifts required in the development of a quotient to a normalized status.

If Q01F is set, entry to J = 7 is from either J = 5 or J = 6. With Q01F set, shifting of the B and X registers only occurs on the first clock pulse of J = 7, if the level W04L' is true (B/A is less than four). If W04L' is true, the next quotient digit will be formed via subtraction; with Q01F in the set status indicating a positive (above zero) remainder and a complemented divisor, it is permissible to proceed with the shifting of the remainder. If, however, the level W04L is true (with Q01F being true), the shifting of the remainder must be delayed one clock pulse time. This is required for a combination of reasons:

1. If W04L is true, the next quotient digit is predicted to be equal to or greater than four
2. The next quotient digit will be formed via addition
3. Entry to J = 7 was from J = 6, thus the remainder currently in the B register is positive (above zero)

4. The divisor must be subtracted from the remainder prior to shifting the remainder, and
5. The divisor must be decomplemented before entering the next divide cycle

Once the divisor becomes decomplemented, Q01F' goes true. The quotient digit in the quotient digit counter is transferred to the X register in conjunction with a simultaneous left shift of the B and X registers.

J to 6

In either case, as described above, Q01F' being true will cause a transfer to J = 6 where development of the next quotient digit is formed via addition.

Decomplement A

If Q01F and W04L are both true, the next quotient digit will be formed by adding. Entry to J = 7 is from J = 6 as described above. The same conditions could still prevail if the previous quotient digit were a zero (shifting of the B and X registers without going to J = 5 or J = 6).

J to 5. If Q01F is set, entry to J = 7 is from either J = 5 or J = 6 with the remainder being a positive number (greater than zero). If (W05L' • W04L') is true, the quotient predictor logic indicates that the next quotient digit will be less than four and is not a zero. Therefore, the J register is set to 5 for the subtraction sub-cycle to commence.

Quotient Has Been Obtained

X13L' signals that the quotient in the X register has been normalized, and Q02F' signals that the exponent arithmetic has been completed. Therefore, the quotient in the X register is shifted to the B register and the B register extension is cleared to zero. The J register is set to 15 for termination of the operator.

Q01F to 1 or 0

The specifications state the 14 quotient digits are developed and then rounded to 13 digits. If the 14th quotient digit that is developed, is equal to four or more, Q01F is set for subsequent rounding. If the 14th quotient digit is less than four, Q01F is reset.

Decrement B Exponent

Since the divide operator must normalize mantissas prior to the divide cycle, the divide operation will develop a normalized quotient in the X register, by either 13 or 14 shifts of the X register. The N register will be counted to either 13 or 14. If a smaller mantissa is divided into a larger mantissa, 13 shifts of the X register are required to develop a

normalized mantissa. If a larger mantissa is divided into a smaller mantissa, 14 shifts of the X register are required to develop a normalized mantissa and the leading quotient digit was a zero. Because the exponent arithmetic is based upon 13 shifts of the X register (a smaller mantissa into a larger mantissa), the final exponent must be decreased by one, if N equals 14.

J = 15

The syllable execute complete level is true to terminate the operator. The A register, plus Q01F, is added to the B register. The A register was previously cleared to zero at J = 7; therefore, Q01F (if set), will add a one to the final quotient for rounding. If the final developed quotient is equal to zero (W07L), the B register is cleared to zero to insure a zero exponent.

Exponent Interrupt

If the exponent sign is negative (B46F) and the high order exponent bit (M11F) is on, the exponent underflow bit is set into the interrupt register. If the exponent sign is positive (B46F') and the high order exponent bit (M11F) is on, the exponent overflow bit is set into the interrupt register.

3.20 INTEGER DIVIDE

DV3L-3001

The operand in the B register is algebraically divided by the operand in the A register and the integer part of the quotient is left in the B register. For all conditions, after the operation the A register is set to empty, the B register is set to full and the B register flag bit is set to zero.

If the mantissa of the B register is zero, the B register is set to all zeros. If the mantissa of the A register is zero, the divide by zero bit is set in the Interrupt register. In either case, the operation is terminated.

If the mantissa of neither operand is zero, both operands are normalized. If the exponent of the B register is algebraically less than the exponent of the A register after both operands have been normalized, the B register is set to all zeros. If the exponent of the B register is algebraically equal to or greater than the exponent of the A register, the divide operation proceeds until an integer quotient or a quotient of 13 significant digits is calculated.

If an integer quotient is developed, the quotient is left in the B register with zero exponent. If a non-integer quotient is developed, the integer overflow bit is set in the Interrupt register and the 13 significant digit quotient with the correct exponent, modulo 64 is left in the B register.

SUMMARY OF OPERATION

The Integer Divide operator functions in a fashion identical to the Single Precision Divide operator (to which reference may be made), with the exception of the modifications as related to the exponential conditions. The procedure undertaken in the Integer Divide operator (providing neither mantissa is equal to zero) is to first normalize both operands. Once the operands are normalized, the exponent values are interrogated. If the exponent of the operand in the A register (divisor) is algebraically greater than the exponent of the operand in the B register, it is impossible to obtain an integer quotient. Therefore, the operator is terminated with both operands being cleared to zero. If, however, the exponent of the operand in the A register is algebraically less than the exponent of the operand in the B register, it may be possible to obtain an integer quotient. In which case, the divide cycle proceeds in an effort to develop an integer quotient. Note that the exponent arithmetic is not allowed to commence at this time as is done in a normal single precision divide operator.

During the division process, as each quotient digit is generated, the exponent of the operand in the B register is decremented. If the exponents become algebraically equal before 13 quotient digits are developed, the quotient is an integer. If, however, 13 quotient digits are developed and the exponents are still not algebraically equal, an integer quotient cannot be developed. If a non-integer quotient is generated, the exponent subtract

SIMPLIFIED FLOW CHART INTEGER DIVIDE

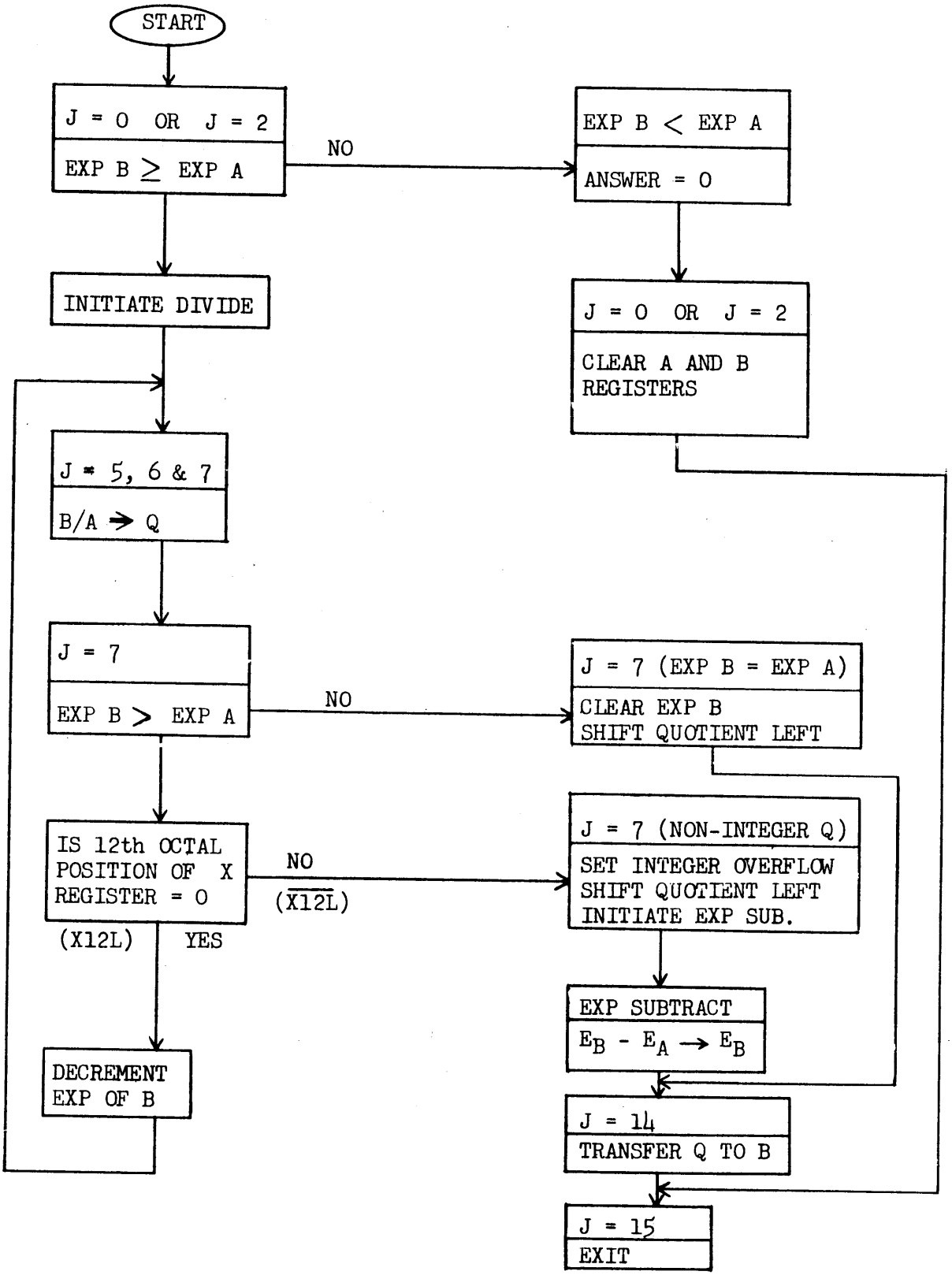


FIGURE 3.20-1. INTEGER DIVIDE

operation is allowed to proceed. It can be seen that the divide cycle may terminate at any time during the development of the quotient (exponents become equal).

Exponent Manipulation - Integer Divide

The following states the considerations relating to exponent manipulation during Integer Divide:

1. If exp. A $>$ exp.B
   a. Result exp. = 0 (answer = 0)

2. If mantissa B $>$ mantissa A and,
   a. If 0 $\leq$ exp. B - exp. A $\leq$ 12
      (1) Result exp. = 0
   b. If exp. B - exp. A $>$ 12
      (1) Result exp. = exp. B - exp. A - 12 (integer overflow)

3. If mantissa A $>$ mantissa B and,
   a. If 0 $\leq$ exp. B - exp. A $\leq$ 13
      (1) Result exp. = 0
   b. If exp. B - exp. A $>$ 13
      (1) Result exp. = exp. B - exp. A - 13 (integer overflow)

DETAILED DESCRIPTION

J = 0 and J = 1

Stack adjustment and normalization of the operands takes place if required. If the operands are normalized on entry to the operator, the operations which examine the exponents are performed. If normalization is required, the operations which examine the exponents are performed at J = 2. The specific actions are described under the J = 2 heading.

J = 2

Once normalization is completed, the exponents are interrogated. If the exponent of the operand in the A register is algebraically greater than the exponent of the operand in the B register (W74L), an integer quotient cannot be obtained. Therefore, both the A and B registers are cleared to zero and the J register is set to 15 for termination of the operator. The A register mantissa extension is also cleared at this time due to redundant logic, but has no significance at this time.

If the exponents are either algebraically equal or the exponent of the operand in the A register is algebraically less than the exponent of the operand in the B register (W73L + W75L), the divide operation proceeds as described in the Single Precision Divide write-up for J register settings 5, 6 and 7.

J = 7

Because a basic comparison of exponents is accomplished prior to or during the divide cycle, exponent arithmetic is not allowed to start when the divide cycle is initiated as is done in the normal single precision divide cycle. Thus exponent arithmetic is not initiated at J = 2. If and when it is determined that a non-integer quotient will be developed (W75L • W12L'), exponent arithmetic is initiated. For the same reason, the sign of the exponent of the operand in the A register is not complemented when the divide cycle is initiated. The exponent signs must remain in their original configuration during the divide cycle, in order to algebraically subtract a 1 from the B operand exponent, as each digit of the quotient is developed. The A operand exponent sign will be complemented (to permit subtraction of exponents) only when it is determined that a non-integer quotient has been developed.

If, at any time during the divide cycle, the exponents become equal (W73L), when the 12th octade position of the X register is equal to zero (X12L), the exponent of the operand in the B register is cleared to zero. The last quotient digit just formed is transferred to the X register and the J register is set to 14 for subsequent termination of the operator.

If, during the divide cycle, a digit occupies the 12th octade position of the X register (X12L') and the exponent of B is still greater than the exponent of A, an integer overflow bit is set into the interrupt register by the setting of U17F and U18F. Exponent arithmetic is initiated by the setting of Q02F. The logical flip-flop, Q04F, is also set to prevent the clearing of the M register at J = 14, thereby allowing the checking of an exponent underflow/overflow condition. The last quotient digit formed is transferred to the X register and the J register is set to 14, for subsequent termination of the operator.

J = 14 • Q02F'

Once the exponent arithmetic is completed (if it was initiated), the final quotient result is transferred from the X to the B register. The A register is cleared for two reasons:

1. The requirement of the operator that the A register be cleared at termination of the operator.
2. It must be cleared because of the redundant add operation that will take place at J = 15.

The Q flip-flops (1 through 9) are cleared to zero because Q01F must be reset during the redundant add operation at J = 15. If a non-integer quotient was developed (Q04F), the M register is prevented from being cleared to allow the M register to be interrogated at J = 15.

J = 15

A redundant add cycle is performed due to the use of common logic. The syllable execute complete level is true to terminate the operator. Exponential interrupts are established as in Single Precision Divide, if required.

3.21 REMAINDER DIVIDE

The operand in the B register is algebraically divided by the operand in the A register to develop an integer quotient. The remainder after the division is left in the B register. For all conditions, after the operation the A register is set to empty, the B register is set to full and the B register flag bit is set to zero.

If the mantissa of the B register is zero, the B register is set to all zeros. If the mantissa of the A register is zero, the divide by zero bit is set in the Interrupt register. In either case, the operation is terminated.

If the mantissa of neither operand is zero, both operands are normalized. If the exponent of the B register is algebraically less than the exponent of the A register, after both operands have been normalized, the operand in the B register is the result. If the exponent of the B register is algebraically equal to, or greater than, the exponent of the A register, the divide operation proceeds until an integer quotient or a quotient of 13 significant digits is calculated.

If an integer quotient is developed and the mantissa of the remainder is not zero, the remainder with exponent modulo 64 is placed in the B register. If an integer quotient is developed and the mantissa of the remainder is zero, the B register is set to all zeros.

If a non-integer quotient is developed, the integer overflow bit is set in the Interrupt register and the B register is set to all zeros.

If the result has an exponent less than -63, the exponent underflow bit is set in the Interrupt register. The B register contains the correct mantissa, mantissa sign and exponent sign. The mantissa of the correct exponent is contained in the exponent field of the B register modulo 64.

SUMMARY OF OPERATION

The Remainder Divide Operator functions in a similar fashion to the Integer Divide Operator with exceptions in the handling of the exponent and certain exponential functions. For a summary of the basic divide operation, refer to the Single Precision Divide Operator (DV1L).

Figure 3.21-1 illustrates the basic procedure utilized in the performance of the Remainder Divide Operator. The following states the considerations relating to the exponent manipulation during Remainder Divide Operator:

1. If exp. A $>$ exp. B
   a. Remainder exp. = exp. of B

2. If mantissa B $>$ mantissa A and,
   a. If $0 \leq$ exp. B - exp. A ≥ 12

SIMPLIFIED FLOW CHART REMAINDER DIVIDE

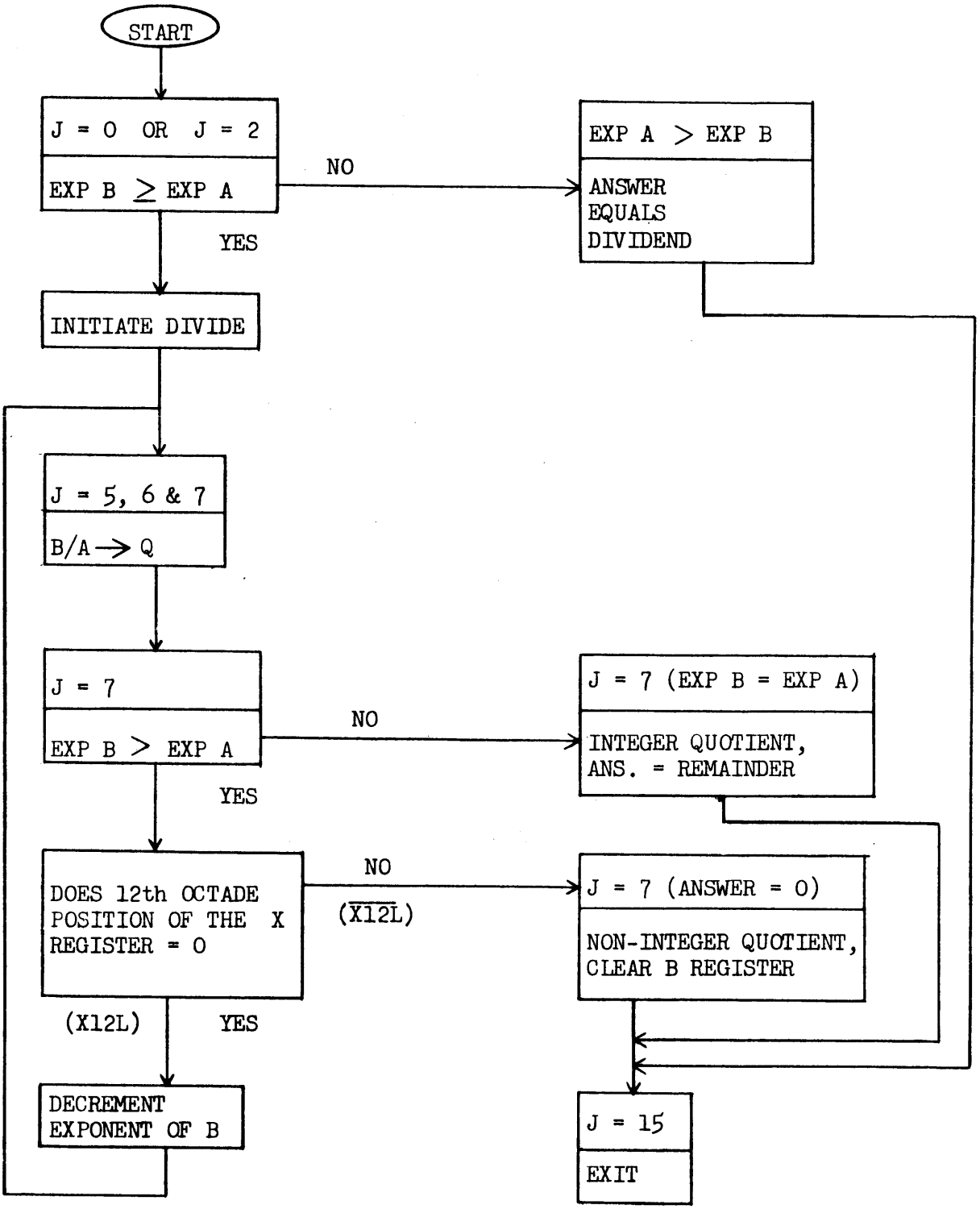


FIGURE 3.21-1. REMAINDER DIVIDE

(1) Remainder exp. = exp. B - exp. A
b. If exp. B - exp. A > 12
(1) Remainder exp. = 0 (remainder = 0)

3. If mantissa A > mantissa B and,
a. If 0 ≤ exp. B - exp. A ≤ 13
(1) Remainder exp. = exp. B - exp. A
b. If exp. B - exp. A > 13
(1) Remainder exp. = 0 (remainder = 0)

DETAILED DESCRIPTION

J = 0 and J = 1

Stack adjustment and normalization of the operands takes place, if required, as described in the Single Precision Divide write up. If the operands are normalized on entry to the operator, the operations which examine exponents are performed. If normalization is required, the operations which examine exponents are performed at J = 2. The specific actions are described under the J = 2 heading.

J = 2

Normalization continues until completed if it is required. Then, if the exponent of the operand in the A register is greater than the exponent of the operand in the B register (W74L), the operand in the B register is the result; i.e., a zero integer quotient would result if the division were allowed to proceed. Therefore, the A register is cleared and the J register is set to 15 for termination of the operator.

If the exponents are either algebraically equal, or the exponent of the operand in the A register is algebraically less than the exponent of the operand in the B register (W73L + W75L), the divide operation is initiated. Because exponent comparison and manipulation is accomplished prior to or during the divide cycle, the exponent arithmetic operation is not required during the Remainder Divide Operator.

J = 7

If during the divide cycle, the exponents become equal, an integer quotient has been obtained. In which case, the A register is cleared (answer is in B) and the J register is set to 15 for termination of the operator. If, at this time, it is observed that Q01F is in the reset state (W73L • Q01F'), the remainder in the B register has been reduced (by the value of the divisor) one too many times (See J = 5 box). Thus an add restore operation is performed to obtain the proper remainder value.

If, during the divide cycle, the 12th octade position of the X register becomes occupied and the exponents are still not equal, an integer quotient cannot be obtained. The A and B registers are cleared to zero and the J register is set to 15, for termination of the operator. The Integer Overflow interrupt is also set into the interrupt register by the setting of U17F and U18F. The Q flip-flops and the M register bits (6 through 4) are cleared because of the redundant add cycle at J = 15.

J = 15

A redundant add cycle is performed due to the use of common logic. The syllable execute complete level is true to terminate the operator. Exponential interrupts are established if required.

3.22 DOUBLE PRECISION ADD
DOUBLE PRECISION SUBTRACT

AD2L-0105
SU2L-0305

GENERAL

Add Double Length

The double length operand in the A and B registers is algebraically added to the double length operand addressed by the S register. The double length result is left in the A and B registers. The S register is reduced by two. Bit positions 48 through 40, of the least significant word of the double length result, are set to zero. The flag bit of the most significant word of the double length result is set to zero. All non-zero results are normalized. The A and B registers are both set to full.

If the exponents of the two operands are equal, the double length mantissas are algebraically added. If the double length mantissa exceeds 26 octal digits, the double length mantissa is shifted right one place and the exponent algebraically increased by one. The result is normalized. If the mantissa of the result is zero, the A and B registers are set to all zeros and the operation is terminated.

If the exponents of the operands are not equal, alignment of the double length operands occurs. The alignment of double length operands is equivalent to the alignment of single length operands. When shifting a double length mantissa, digits are shifted between the least significant digit of the most significant word of the operand and the most significant digit of the mantissa of the least significant word of the operand.

If the mantissa of the smaller operand is shifted right 26 or more octal digits during alignment, the larger operand, normalized, is the result.

After alignment of operands with unequal exponents, the mantissas of the operands are algebraically added. If the double length mantissa exceeds 26 octal digits, the double length mantissa of the result is shifted to the right one place and the exponent algebraically increased by one. The result is normalized.

If the exponent of the result is greater than +63 or less than -63, the exponent overflow bit or exponent underflow bit, respectively, is set in the Interrupt register. The result in the A and B registers contains the correct double length mantissa, mantissa sign and exponent sign. The magnitude of the correct exponent is contained in the exponent field of the A register modulo 64.

Subtract Double Length

The double length operand in the A and B registers is algebraically subtracted from the double length operand addressed by the S register. An addition operation is performed as specified for the Add Double Length operator, except that the conditions of sign comparison are changed to effect an algebraic subtraction.

SUMMARY OF OPERATION

The two double length operands are defined via the following terminology:

$S_1\ e_1\ E_1\ M_1\ m_1$ = Operand at the top of the stack

$S_2\ e_2\ E_2\ M_2\ m_2$ = Other operand

$S_1\ e\ E\ M_3\ m_3$ = First operand normalized or scaled

$S_2\ e\ E\ M_4\ m_4$ = Second operand normalized or scaled

$S\ e\ E\ M\ m$ = Result

S = mantissa sign
e = exponent sign
E = exponent value
M = high order half of mantissa
m = low order half of mantissa

J Register Settings

The occurrences during double precision add-subtract relative to the J register settings are as follows:

J = 0 Start

Initiate loading of the A and B registers from the stack of initially empty.

J = 1 Read $S_1\ e_1\ E_1\ M_1$ and/or m_1 from the stack.

J = 2 Read $S_2\ e_2\ E_2\ M_2$ from the stack.

J = 3, 4, 5, 6, and 7 Adjust Operands

The two double length operands are adjusted until the exponents are equal:

$S_1\ e_1\ E_1\ M_1\ m_1$ redefined as $S_1\ e\ E\ M_3\ m_3$

$S_2\ e_1\ E_2\ M_2\ m_2$ redefined as $S_2\ e\ E\ M_4\ m_4$

J = 8 Add; Subtract m

The low order halves of the operand (m_3 and m_4) are added or subtracted.

J = 9 Obtain M_4 from the stack.

J = 10 Add; Subtract M

The high order halves of the operand (M_3 and M_4) are added or subtracted.

J = 11 Scale Results

If the mantissa result of internal addition is equal or greater than 8^{13}, the result is scaled one position.

J = 12, 13 Decomplement

The mantissa result of internal subtraction is less than 8^{13}.

J = 13 Normalize
1. Normalize result
2. If result = zero; exponent and sign are cleared
3. Place S e E M in the A register
4. Place m in the B register

J = 15 Exit
1. Set exponent overflow or underflow in the Interrupt register if valid
2. Clear flag bits
3. Syllable execute complete level is true

Q Flip-flops

The Q Flip-Flops are utilized as follows:

Q01F - +1 for subtraction
+1 for (m_3 + m_4) overflow
+1 for decomplement
+1 for round

Q02F - On for internal add; also used to control decomplementing.

Q03F - On for internal add; also used to control storing of M_4 and m_4.

Q04F - On, if operands have been interchanged once. Off if operands have not been interchanged, or they have been interchanged twice.

Q05F - Set when scaling operands and M = 0.

Q06F - On if AD2L initiated from MU2L.

DETAILED DESCRIPTION

J = 0

Stack adjustment commences at J equals zero to load the A, B and X registers. Initially it is not known whether the operands have equal exponents; or if not equal, which exponent is algebraically greatest. Therefore, the A, B and X registers are loaded with the assumption that the exponents are not equal and that the top operand in the stack has the greatest exponent. Normalizing the scaling is accomplished via the B and X registers. Therefore, initially the assumed operand with the greatest exponent, is loaded into the B and X register. S_1 e_1 E_1 M_1goes to the B register, m_1 goes to the X register, and S_2 e_2 E_2 M_2 go to the A register.

Because of the necessity of considering four possible conditions of the A and B registers on entry to the operator (A and B loaded, A and B empty, A or B empty or loaded), some of the logic to load the A, B and X registers, to the proper configuration, is redundant. Unconditionally, the E register is set to 2 for loading the A register, which in 3 out of 4 possible initial conditions will load one half of the top operand in the stack. In the fourth condition, it will load the first half. (S_2 e_2 E_2 M_2) of the second operand. The J register is set to one to complete the load and fetch is inhibited to prevent any fetch from delaying the other required accesses.

B to A

If the B register is occupied, the contents of the B register are transferred to the A register. If the A register is initially empty, this puts the first half of the top operand in the stack (S_1 e_1 E_1 M_1) into the A register. If however, the A register is initially occupied, this puts the second half of the top operand in the stack (m_1) in the A register.

A to B

If the A register is initially occupied, its contents are transferred to the B register. This places the first half of the top double length operand (S_1 e_1 E_1 M_1) in the B register. If both the A and B registers had been loaded on entry to the operator, these two actions would have interchanged, S_1 e_1 E_1 M_1 and m_1.

J = 1

The occurrences under this J register setting continue the required stack adjustment. The specific actions accomplished are dependent upon the initial entry conditions at J = 0. The desired end result at J = 1 is to have the B register loaded with S_1 e_1 E_1 M_1, the X register loaded with m_1, and to have the E and S registers set to load S_2 e_2 E_2 M_2 into the A register. The following discussion describes the actions at J = 1 relative to the initial setting of the A and B registers:

A and B Initially Loaded

If the A and B registers were initially occupied, the only thing accomplished at J = 1 is to transfer m_1 from the A register to the X register and to transfer control to J = 2 to await completion of the loading of the A register with S_2 e_2 E_2 M_2.

A Empty and B Loaded Initially

If initially the A register was empty and the B register was loaded, the first occurrence at J = 1 is the completion of the memory cycle to load the A register with m_1. The A register is marked occupied and another memory cycle started to load the A register with S_2 e_2 E_2 M_2. The next clock pulse transfers m_1 from the A register to the X register, with the J register being set to 2 to await completion of loading the A register with S_2 e_2 E_2 M_2.

A Loaded and B Empty Initially

If initially the A register was occupied and the B register was unoccupied, the first occurrence at J = 1 is to transfer the mantissa portion of the A register contents (M_1) to the X register and also to transfer the contents of the complete A register to the B register (S_1 e_1 E_1 M_1). The A to B register transfer is redundant, having been accomplished at J = 0. The A to X register transfer is unnecessary, as the present A register contents are not required in the X register. The only action of importance occurring at the first clock pulse of J = 1 is to mark the B register as occupied and the A register as unoccupied. The second occurrence at J = 1 is the completion of the memory access (MROF), which loaded the A register with the lower half of the top operands mantissa m_1, marking the A register as occupied and starting another memory cycle to load the A register with S_2 e_2 E_2 M_2. The next clock pulse transfers the A register contents (m_1) to the X register and the J register is set to 2 to await completion of the memory access.

A and B Initially Empty

If initially the A and B registers were unoccupied, the first occurrence at J = 1 is an A to B transfer which is unneccessary. At completion of the memory cycle started at J = 0 (MROF) to load the A register with S_1 e_1 E_1 M_1, the A register is marked occupied and a memory cycle is started to load the A register with m_1. The next clock pulse transfers the A register contents to the B register and marks the A register unoccupied. Also, an A to X register transfer occurs, which is unnecessary at this time and the B register is marked occupied, which is necessary at this time. At the completion of the memory cycle (MROF) to load the A register with m_1, the A register is marked occupied and another memory access is initiated to load the A register with S_2 e_2 E_2 M_2. The next clock pulse transfers the A register contents (m_1) to the X register with the J register being set to 2 to await completion of the memory access.

J = 2

Upon completion of the memory access, initiated at either J = 0 or J = 1, the J register is set to 3 to start the exponent comparison. Entrance to J = 3 occurs with the A, B and X registers containing the following information:

A register - Most significant half of the 2nd operand (S_2 e_2 E_2 M_2)
B register - Most significant half of the 1st operand (S_1 e_1 E_1 M_1)
X register - Least significant half of the 1st operand (m_1)

The following chart illustrates the necessary actions during J = 0, 1 and 2 to accomplish the loading of the A, B and X registers, starting from the four possible states of the A and B register on entry to the operator. The circled memory address cells in the stack in the next word brought into the A register (when E is set to 2). The stated condition relative to a J register setting is illustrated as on entry to the J register setting, i.e., on entry to J = 2, the E register is set to load the A register with S_2 e_2 E_2 M_2.

		REGISTERS			STACK			
J		A	B	X	4	3	2	1
0	AROF•BROF	S_1 e_1 E_1 M_1	m_1				(circled) S_2 e_2 E_2 M_2	m_2
1	AROF•BROF	E=2/m_1	S_1 e_1 E_1 M_1				(circled) S_2 e_2 E_2 M_2	m_2
2		E=2	S_1 e_1 E_1 M_1	m_1			(circled) S_2 e_2 E_2 M_2	m_2

		REGISTERS			STACK			
J		A	B	X	4	3	2	1
0	AROF'•BROF		$S_1e_1E_1M_1$			m_1	$S_2e_2E_2M_2$	m_2
1	MROF•AROF'•BROF	E=2	$S_1e_1E_1M_1$			m_1	$S_2e_2E_2M_2$	m_2
	AROF•BROF	E=2 / m_1	$S_1e_1E_1M_1$			m_1	$S_2e_2E_2M_2$	m_2
2		E=2	$S_1e_1E_1M_1$	m_1		m_1	$S_2e_2E_2M_2$	m_2

J		A	B	X	4	3	2	1
0	AROF•BROF'	S_1 e_1 E_1M_1				m_1	$S_2e_2E_2M_2$	m_2
1	AROF•BROF'	E=2/$S_1e_1E_1M_1$				m_1	$S_2e_2E_2M_2$	m_2
	MROF•AROF'•BROF	E=2	$S_1e_1E_1M_1$			m_1	$S_2e_2E_2M_2$	m_2
	AROF•BROF	E=2 / m_1	$S_1e_1E_1M_1$			m_1	$S_2e_2E_2M_2$	m_2
2		E=2	$S_1e_1E_1M_1$	m_1		m_1	$S_2e_2E_2M_2$	m_2

J		A	B	X	4	3	2	1
0	AROF'•BROF				$S_1e_1E_1M_1$	m_1	$S_2e_2E_2M_2$	m_2
1	MROF•AROF'•BROF'	E=2			$S_1e_1E_1M_1$	m_1	$S_2e_2E_2M_2$	m_2
	AROF•BROF'	E=2/$S_1e_1E_1M_1$			$S_1e_1E_1M_1$	m_1	$S_2e_2E_2M_2$	m_2
	MROF•AROF'•BROF	E=2	$S_1e_1E_1M_1$			m_1	$S_2e_2E_2M_2$	m_2
	AROF•BROF	E=2 / m_1	$S_1e_1E_1M_1$			m_1	$S_2e_2E_2M_2$	m_2
2		E=2	$S_1e_1E_1M_1$	m_1		m_1	$S_2e_2E_2M_2$	m_2

J = 3

The primary function of this J register setting is to compare and equalize exponents, if possible. Initial entry into J = 3 occurs with the A register containing S_2 e_2 E_2 M_2, the B register containing S_1 e_1 E_1 M_1, and the X register containing m_1. A number of possible actions may occur at J = 3, based primarily upon a comparison of the exponents algebraic magnitude. Comparison of the two exponents and the actions taken are listed below:

1. Exponents are equal - start normal add/subtract process
2. Exponent A is less than Exponent B, B mantissa is not normalized - normalize the B mantissa
3. Exponent A is less than Exponent B, B mantissa is normalized - interchange operands; go to 5
4. Exponent A is greater than Exponent B, A mantissa is not normalized - interchange operands; go to 2
5. Exponent A is greater than Exponent B, A mantissa is normalized - scale B

The A, B and X registers are provided with the facility of only being able to normalize or scale either operand via the B and X register. Therefore, if either operand requires normalization or scaling, the complete double length operand must be placed into the B and X registers before normalizing or scaling commences. The initial configuration of the operands in the A, B and X registers is such to facilitate normalizing or possibly scaling of the top operand of the stack (presently contained in the B and X registers). If the second double length operand of the stack requires normalizing or scaling, it first must be placed in the B and X registers.

W73L Exponents Equal

If the exponents are equal on entry to J = 3 or adjusted to equality, the following actions occur. Once the operands are normalized or scaled into equality, the mantissa portions of the operands are redefined as M_3 m_3 and M_4 m_4 to differentiate the normalized or scaled mantissa from the original configurations of M_1 m_1 and M_2 m_2. This is just a matter of convenience to help keep track of the operands.

N to Q The N register is set to zero when the exponents become equal to zero. If the exponents were equal on entry, the N register equals zero. The N register is utilized to tally the left or right shifts required when normalizing or scaling the operands.

Exit to Add/Subtract. If either 0 or 2 operand interchanges have occurred (2 operands interchanges occur when both normalizing and scaling is required to equalize exponents), indicated by Q04F being reset; the least significant mantissa (m_3) of the first operand is transferred to the A register and the most significant mantissa (M_3) of the first operand is stored in the X

register. The stack address is counted down and the E register is set to 3 to initiate a memory cycle to load the B register with the least significant mantissa (m_4) of the second operand. The J register is set to 8 to await completion of the memory access before starting the add or subtract cycle of the least significant mantissas.

Store M_4. If the operands have been interchanged only once, in the process of equalizing exponents, the most significant half of the second operand (M_4) is stored back into the stack. The purpose of storing M_4, at this time, is to be able to load the A, B and X registers to the same configuration, as with 0 or 2 operands interchanged, prior to commencing either an Add or Subtract operation.

Q02F and Q03F to 1. If an internal add operation is to be performed (W97L) as indicated by an Add operator with equal signs or a Subtract operator with unequal signs, both Q02F and Q03F are set to one, thereby enabling add logic and excluding complement addition logic.

E to 11, J to 6

If the exponent of B is greater than the exponent of A (W75L), the B register is normalized (B13L'), the operands have been interchanged once (Q04F) and the N register does not equal 14 (N14L'), a second operand interchange is initiated by setting the E register to 11 to store the most significant half of the normalized second operand (M_4). The J register is set to 6 to continue the operand interchange. This operand interchange is required to allow the first operand (presently in the A register), to be scaled in an attempt to equalize exponents.

Normalize B, X

If the exponent of B is greater than the exponent of A (W75L) and the mantissa of the B register contents is not normalized (B13L), the B and X registers are shifted left by octades in an attempt to either normalize the B register or equalize exponents. The N register is counted plus one to tally each left shift of the B and X registers. The N register is utilized to determine if the mantissa is zero.

Decrease B Exponent

In conjunction with each left shift of the B register, in an attempt to normalize or equalize exponents, the B register is decremented as long as the B register mantissa is not zero (W07L'), or the N register does not equal 14 (N14L'). In other words, if the N register is increased to 13 and the B register equals zero, the complete mantissa equals zero. Therefore, W07L' + N14L' indicates that the B register contains information, and decrementing of the B register mantissa is still required.

W76L Exponents Unequal - Operand Interchange Required

If either the exponent of B is greater than the exponent of A, the B mantissa is normalized and the operands have not been interchanged or the exponent of A is greater than the exponent of B and A is not normalized (W76L), the first operand interchange is started. The contents of the X register (m_3 if B normalized or m_1 if B not normalized) is transferred to the A register to permit storage in the stack. The E register is set to 10 to initiate the store memory cycle and the J register is set to 4 to continue the operand interchange.

W74L • A13L' - Exponents Unequal, A is Normalized, Scale B

If the exponent of the operand presently in the A register is greater than the exponent of the operand presently in the B register (W74L) and the A register mantissa is normalized (A13L'), the B and X registers are scaled in an attempt to equalize exponents. (W74L • A13L' is a common gate for the following actions.)

Increase B Exponent. As long as the N register is not equal to 14, the B exponent is incremented for each right shift of the B and X registers. When scaling the B and X registers, the level N14L' being true indicates that the mantissa of the operand presently in the B and X registers is not equal to zero.

Prepare for Round. During scaling of the B and X registers, any digit shifted out of the X register is interrogated to determine whether a round may be required. Only the last digit shifted out of the X register has any significance in determining whether a round may be required. If the last digit shifted out is 4 or greater (X03F), Q01F will be set. If the last digit shifted out is less than 4 (X03F'), Q01F will be reset.

Q05F to 1. If the B register mantissa is equal to zero (W07L), the logical flip-flop Q05F is set to one. This will allow the N register to count the Right shifts of the B and X registers as they are scaled.

N + 1. With Q05F on, the N register is incremented for each shift of the B and X registers. The N register is utilized in this application to determine whether the X register is zero or not.

Exponent of A to B.

If the N register is equal to 14 (N14L) and the B register mantissa is equal to zero (W07L), the entire double length mantissa has been cleared to zero. Therefore, the other operand is the result. The exponent transfer of A to B will equalize exponents and transfer control to the W37L level where an add cycle will be started to put the final result in the A and B registers.

N to Zero

At the completion of normalizing (W75L B13L') the operand in the B and X registers, the N register must be cleared so that the N register may be utilized to tally scaling of the other operand.

J = 4

Entrance to this J register setting only occurs when the first of a possible two operand interchanges is initiated. In this case the operands have not been interchanged, the exponents are unequal and the operands are in the incorrect registers for normalizing or scaling. When the memory access, that stored the least significant half of the mantissa originally in the X register (m_1) is completed (EEZL), the stack address is decremented to address the least significant half of the second operand (m_2) for subsequent insertion into the X register. Actually, the S register will be counted down twice, at this J register setting, in order to address the proper mantissa (m_2). The second clock pulse of this J register setting (EWZL) transfers the B register contents (S_1 e_1 E_1 M_1) to the A register, initiates a load of the B register to load m_2 and sets the J register to 5 to complete the operand interchange.

J = 5 • EEZL

This J register setting that completes the operand interchange may have resulted from either a first or second operand interchange cycle. The state of the logical flip-flop, Q04F, tells whether zero or one operand interchange has taken place. If Q04F is off and is being set, this is the first operand interchange. If Q04F is on and is being reset, this is the second operand interchange. The mantissa portion of the B register contents is transferred unconditionally to the X register, which for the first operand interchange is m_2 and for the second operand interchange is m_1.

Q04F Off. During the first operand interchange, Q04F is in the off state, thereby gating a count up to the S register, plus setting the E register to 3 to load the B register with S_2 e_2 E_2 M_2.

Q04F On. During the second operand interchange, Q04F is in the on state, thereby resulting in the A register contents (S_1 e_1 E_1 M_1) being transferred to the B register, the stack address being decremented and the E register set to 3 to load the A register with S_2 e_2 E_2 M_2.

J = 6

Entry to this J register setting occurs when the second of a possible two operand interchanges is initiated. In this case the operands have been interchanged from their initial configuration (at entrance to J = 3), the exponents are unequal (E_B Greater than E_A) and the B mantissa is normalized.

The entrance to J = 6 follows a memory cycle, initiated at J = 3, to store the most significant half of the normalized second operand (S_2 e_2 E_2 M_4). Upon completion of the memory cycle (MWOF), the X register contents (m_4) are transferred to the B register for subsequent storage in the stack. The S register is decremented and E is set to 11 to initiate the store. The logical flip-flop Q03F is turned on (via MROF) to inhibit the store logic on subsequent clock pulses of this J register setting. Upon completion of the store memory cycle, the S register is incremented twice (via EEZL Q03F) in order to address the least significant half of the first operand (m_1). In conjunction with the second count of the S register (EWZL) a memory cycle is initiated to load the B register with m_1 (E to 3), Q03F is cleared and the J register is set to 5 to complete the operand interchange.

J = 7

Entry to this J register setting is from J = 3 when exponents have been equalized and the operands have been interchanged once. The mechanization of the forthcoming add/subtract logic assumes that the operands have not been interchanged. Therefore, one of the functions at J = 7 is to arrange the operands into the same configuration as with zero or two operand interchanges. Upon completion of the memory cycle, initiated at J = 3 (MROF), to store the most significant half of the equalized second operand M_4, the most significant half of the equalized first operand (M_3) is stored in the X register. The X register contents (m_4) are transferred to the B register, the S register is incremented to address m_3 and a memory cycle is initiated to load the A register with m_3. The J register is set to 8 to proceed with the add/subtract cycle.

J = 8 • EEZL

Entry to this J register setting can either be from the previously described J register setting (J = 7) or from J = 3 (exponents are equal and either 0 or 2 operand interchanges have been accomplished). In either case;a load memory access would have been initiated (E=2 or E=3); therefore, the memory access must be completed before commencing J = 8 (EEZL).

Internal Subtract

If Q03F is off, an internal subtract is indicated. Therefore, the A register mantissa (m_3) is complemented and Q03F is set to allow the complement addition to take place. Q01F is complemented in order to set it to the proper value for rounding the answer. If the last digit scaled out of the X register equaled 3 or less, or nothing was scaled out of the X register, Q01F would be in the reset state on entry to J = 8. However, if the last digit scaled out of the X register was equal to 4 or more, Q01F would be in the set state on entry to J = 8. In other words, if rounding is required, Q01F will be in the reset state during the complement addition cycle and one less will be subtracted. If rounding is not required, Q01F will be in the set state during the complement addition cycle and a normal subtraction occurs.

B + A to B

The logical flip-flop Q03F gates the addition of m_3 and m_4. Q03F is in the set state on entry if the internal operation is addition, or will be set at the first clock pulse if the operation is internal subtraction. The J register is set to 9 to continue the addition. Q03F is a common gate for the following actions.

S + 1, A Exponent to B, A sign to B

The logical flip-flop Q04F being off indicates that either 0 or 2 operand interchanges have occured. Because an initial assumption assumed the second operand to be the largest (presently in the A register) and the basic rules of algebraic subtraction indicate the use of the largest operand to determine the sign of the result, the exponent and sign of the final answer is placed in the B register. The S register is incremented to address the most significant half of the second operand (M_4).

S - 1

If the logical flip-flop Q04F is on, M_4 is located below the present address in the S register. Therefore, S is decremented by one.

Q01F to 1 or 0

If there is a carry from the thirteenth octade (W13L), Q01F is turned on so it can be added to the most significant half of the two mantissas. If no carry results from the addition of the two low order mantissas, Q01F is reset.

Q03F to 0

If the operation is internal subtract, the logical flip-flop Q02F will be reset. Therefore, Q03F is turned off in order to gate a complement addition cycle of the major mantissas at J = 10.

E to 2

The logical flip-flop Q06F is in the reset state if the operator is Add/Subtract. Only if this flow is entered via a Multiply operator will Q06F be set. With Q06F reset, the E register is set to 2 to initiate a load of the A register with the most significant half of the second operand (M_4).

J = 9

At this J register setting the operands are set up for the forthcoming addition of the most significant mantissas. Upon completion of the memory access to load the A register with M_4 (MR0F), the contents of the X register (M_3) is transferred to the B register and the contents of the B register

(m_3 + m_4) are transferred to the X register for temporary storage. The S register is counted down by one in preparation to exit the operator. The J register is set to 10 for continuation of the add cycle.

If the logical flip-flop Q06F is set, the operator was entered via the Multiply operator. In which case the A register is cleared and the B and X registers are interchanged regardless of the level MROF.

J = 10

At this J register setting the addition of the two most significant mantissas is accomplished.

Internal Subtract

If the logical flip-flop Q03F is reset the internal operation is Subtract. Therefore, the B register mantissa (M_3) is complemented and transferred to the A register while the A register contents (M_4) are transferred to the B register. Q03F is set to one to allow the addition to take place.

B + A to B

The logical flip-flop, Q03F, gates the addition of M_3 and M_4. The S register is counted down by one in preparation to exit the operator. The least significant mantissa sum (m_3 + m_4) is transferred to the A register from the X register to facilitate a subsequent interchange of the A and B registers. Q01F is reset as it is not known at this clock pulse whether a decomplement or overflow will be required.

J to 11. If there is an overflow (W13L), on the internal add operation (Q02F), the J register is set to 11 to scale the overflow.

J to 12. If there is no overflow (W13L'), on an internal subtract operation (Q02F'), the J register is set to 12 to decomplement the least significant mantissa.

J to 14. If there is an overflow on subtract (Q02F') • W13L) or if there is no overflow on internal add (Q02F • W13L'), the J register is set to 14 for a final normalizing of the result (if required) and placement of the mantissas in the proper registers.

J = 11

Entry to this J register setting is when an overflow occurs during an internal add operation. The double length result in the B and X registers is unconditionally scaled one octade position to the right. Because this scaling is the result of an overflow, a one is placed into the 13th octade of the B register as it is scaled. B37F is set while B38F and B39F are reset. The exponent is incremented by one to maintain the result at the same value. The J register is set to 14 for the final answer set up.

J = 12

Entry to this J register setting is from a no overflow condition during an internal subtract operation; i.e., decomplementing of the double length result is required to obtain a true result.

Q02F Off

On entry to J = 12, Q02F is off and is unconditionally turned on. The A register mantissa ($m_3 + m_4$) is complemented while the B register mantissa ($M_3 + M_4$) is transferred to the X register for temporary storage. The mantissa portion of the B register is cleared to zero and Q01F is complemented to facilitate the forthcoming add cycle.

Q02F On

The complemented sum in the A register mantissa ($m_3 + m_4$) is added to Q01F and the decomplemented result placed into the B register. If an overflow carry occurs, it must be added into the most significant half of the double length mantissa result. The logic at J = 13 provides for complementing Q01F; therefore, if an overflow carry occurs (W13L), Q01F is reset, otherwise it is left in the set state. The J register is set to 13 where the most significant half of the result is decomplemented. The most significant half of the result ($M_3 + M_4$), presently in temporary storage in the X register, is transferred to the A register.

J = 13

At this J register setting the most significant half of the result is decomplemented. The A register mantissa contents ($M_3 + M_4$) and Q01F are unconditionally complemented. If an overflow carry resulted from the previous addition at J = 12, Q01F will be turned on at this time, otherwise, it will be turned off. On entry to J = 13, Q02F is set and is reset to control the remaining actions at this J register setting.

Q02F On

The B register mantissa contents (decomplemented m_3 + m_4) are transferred to the X register for temporary storage, while the B register mantissa is cleared to zero.

Q02F Off (Second Clock Pulse Of J = 13)

The complemented sum in the A register mantissa (M_3 + M_4) is added to Q01F (Q01F may be on or off) and the decomplemented result placed in the B register. By complementing B47F the J register is set to 14 to complete the final answer set up. The sign of the result is decomplemented.

J = 14

At this J register setting the final normalization of the result occurs (if required) and the two halves of the double length operand are placed into the proper registers.

B13L

If the result is not normalized (B13L is true), the B and X registers are shifted left by octades with a corresponding count down of the exponent. Simultaneously, M(14 through 12) is shifted into the first octade position of the X register, which is Add Subtract, and will shift zeros into the X register. If this flow was entered via the Multiply operator, the most significant digit of m_7 will be shifted into the X register. On subsequent shifts, zeros are shifted into the X register, because M(14 through 12) is cleared to zero with the first clock of this J register setting. The N register is counted for each left shift of the B and X register. Shifting is continued until either the result becomes normalized or the answer is discovered to be equal to zero.

Answer = 0

If the N register reaches a count of 13 and the B register mantissa is equal to zero (N13L • W07L) the answer is zero. In this case the operator level is transferred to Double Precision Multiply (MU2L), where the logic is provided to handle a zero answer. The J register setting remains the same while the S register is counted up by one to compensate for a decrement of the S register at J = 14 • MU2L.

Answer is Normalized

When the result is normalized (B13L'), or if normalized on entry to this J register setting, the least significant half of the double length result ($m_3 + m_4$) in the X register, is transferred to the B register. The most significant half of the double length result (S e E $M_3 + M_4$) in the B register, is transferred to the A register and the J register is set to 15 to terminate the operator.

J = 15

The flag bit is unconditionally cleared to zero thereby insuring the results to be an operand. The flag bit, sign bit and the exponent of the second word of the result are cleared to zero as they no longer have any significance. The A and B registers are marked as occupied, fetch is allowed and the syllable execute level is true to terminate the operator.

Exponent Underflow Interrupt to 1

If the sign of the exponent is negative (B46F) and the exponent magnitude exceeds 63 (77 octal) as indicated by the exponent extension M11F being set, an exponent underflow interrupt is set in the interrupt register.

Exponent Overflow Interrupt to 1

If the sign of the exponent is positive (B46F') and the exponent magnitude exceeds 63 (octal 77), as indicated by the exponent extension M11F being set, an exponent overflow interrupt is set in the interrupt register.

3.23 DOUBLE PRECISION MULTIPLY

For the double precision arithmetic operators, an operand occupies two words. The second word of an operand is considered an extension of the mantissa of the first word of an operand; i.e., the mantissa of the first word of an operand is an integer and the mantissa of the second word of the operand is a fraction. When in the stack, the first word of a double precision operand is in the top of the stack and the second word of a double precision operand is in the second word of the stack. Therefore, double precision arithmetic operators operate on four words in the stack, removing those words from the stack and leaving the result as two words in the stack.

Multiply Double Length

The double length operand in the A and B registers is algebraically multiplied by the double length operand addressed by the S register. The double length result is left in the A and B registers and the S register is reduced by two. Bit positions 48 through 40 of the least significant word of the double length result are set to zero. The flag bit of the most significant word of the double length result is set to zero. All non-zero results are normalized. The A and B registers are both set to full.

Both double length operands are normalized. If either operand has a mantissa of zero, the A and B registers are set to all zeros and the operation is terminated.

If neither double length operand has a mantissa of zero, the normalized double length mantissas of the two operands are multiplied. Twenty seven digits of the 52 digit product are retained. The product is normalized and truncated to a 26 digit result. The least significant two digits are not considered a precise part of the result because there may be a maximum error of 1 in the twenty-fifth digit position.

If the exponent of the result is greater than +63 or less than -63, the exponent overflow bit or exponent underflow bit, respectively, is set in the Interrupt register. The result in the A and B registers contains the correct double length mantissa, mantissa sign and exponent sign. The magnitude of the correct exponent is contained in the exponent field of the A register, modulo 64.

SUMMARY OF OPERATION

In order to describe the procedure by which Double Precision Multiply is accomplished, it is convenient to give mnemonic terms to the various operands. The following is the mnemonic notation utilized throughout this description:

S_1 E_1 M_1 m_1 = Initial double length multiplier (two top operands in the stack).

S_2 E_2 M_2 m_2 = Initial double length multiplicand (third and fourth operands in the stack).

S_1 E_3 M_3 m_3 = Normalized double length multiplier with adjusted exponent.

S_{10} E_{10} M_4 m_4 = Normalized double length multiplicand with result mantissa sign and exponent.

Partial Products and Final Result

$M_6 + m_6 = M_4 \times m_6$ (1st partial product)

$M_7 + m_7 = (M_3 \times m_4) + m_6$ (2nd partial product)

$M_8 + m_8 = (M_4 \times M_3) + M_7$ (3rd partial product)

$M_9 + m_9 = M_8 + (m_8 + M_6)$ (4th partial product)

S_{10} E_{10} M_{10} $m_{10} = M_9 + m_9 + m_7$ (Final result)

S = Mantissa sign

E = Exponent (including exponent sign)

M = Most significant half of double length mantissa

m = Least significant half of double length mantissa

Procedure

The procedure by which double precision multiply is mechanized, is illustrated below; in a manner similar to that utilized with pencil and paper:

M_3 m_3 = Normalized operands

M_4 m_4 = Normalized operands

M_6 m_6 = $M_4 \times m_3$

M_7 m_7 = $(M_3 \times m_4) + m_6$

M_8 m_8 = $(M_4 \times M_3) + M_7$

M_9 m_9 = $M_8 + (m_8 + M_6)$

M_{10} m_{10} = $M_9 + m_9$ + (High Order Digit Of m_7)

NOTE:

High order digit is shifted into the low order end of m_9, if (M_9 + m_9) requires a final normalization.

The execution of Double Precision Multiply utilizes the logic of 3 operators; MU2L, MU1L and AO2L. The execution of the J register settings of 5, 6 and 7 is identical to that utilized in single precision multiply. Branching to the Double Precision Add Operator occurs in the development of M_9 m_9 (see above example) for the addition of M_6 to (M_8 + m_8).

Control of the multiply cycles (after normalization of both sets of operands) is governed by the logical flip-flops AROF and BROF. When utilized in this function, AROF and BROF are not referring to the state of occupancy of the A and B registers. The status of AROF and BROF is utilized at J = 8 and J = 9 to initiate the required multiply subcycles and the subsequent transfer to the double precision add operator. The functions of AROF and BROF, relative to the various multiply subcycles, are illustrated below. The indicated states of AROF and BROF, gate the corresponding action:

$\overline{AROF} \cdot \overline{BROF} = M4 \times m3 = M6 + m6$

$AROF \cdot \overline{BROF} = (M3 \times m4) + m6 = M7 + m7$

$AROF \cdot BROF = (M4 \times M3) + M7 = M8 + m8$

$\overline{AROF} \cdot BROF = M8 + (m8 + M6) = M9 + m9$

Flow Diagram

The actions of Double Precision Multiply (after operand are normalized) relative to the register contents are illustrated in Figure 3.23-1.

	REGISTERS				STACK		
START - (NORMALIZED OPERANDS)	A M4	B --	X m3	M	3 M3	2 M4	1 m4
M4 x m3 = M6 + m6	M4	M6	m6		M3	M4	m4
STORE M6; GET M3 and m4 - m6 to 8	M3	m6	m4		M3	M4	M6
(M3 x m4) + m6 = M7 m7	M3	M7	m7		M3	M4	M6
STORE HIGH ORDER DIGIT OF m7 IN M	M3	M7	m7	m7	M3	M4	M6
GET M4 - M3 to X REG.	M4	M7	M3	m7	M3	M4	M6
(M4 x M3) + M7 = M8 + m8	M4	M8	m8	m7	M3	M4	M6
M8 + (m8 + M6) = M9 + m9		M9	m9	m7	M3	M4	M6
(M9 NORMALIZED; IF NO SHIFT.) B + X, LEFT SHIFT m7 INTO X		M9	m9				
EXCHANGE X - B AND B - A	M10	m10					

FIGURE 3.23-1. MULTIPLY FLOW DIAGRAM

Logical Flip-Flops

Q01F - Used during the single precision multiply sub-cycle phase (J = 5,6 and 7).
Q02F - Controls exponent add operation and internal add during double precision add.
Q03F - Used during single precision multiply sub-cycles and for internal add during double precision add.
Q04F - Used for double precision add operation (must be off when going to AD2L from MU2L)
Q05F - Inhibits exponent add (not used during D.P. Multiply). Utilized when MU2L is initiated from DV2L.
Q06F - Indicates exit from MU2L to AD2L. (Set on at exit from MU2L).
Q08F - Used during S.P. Multiply sub-cycles (indicates negative partial product).

AROF and BROF

1. Controls Stack Adjustment
2. Controls Normalization
3. Controls Multiply Sub-cycles

DETAILED DESCRIPTION

J = 0 - Start, obtain M1 and m1 from the stack.

At this J register setting the two word multiplier operand M1 and m1 is obtained from the stack and placed in the B and X registers to allow normalization at J = 2. Fetch is unconditionally inhibited to give precedent to the loading of A and/or B.

If both registers are loaded on entry, the B register mantissa (m1) is transferred to the X register and the A registers contents (M1) are transferred to the B register. The J register is set to 2 to commence normalization.

If one register is occupied on entry, its contents are transferred to the A register (if not already there) and a memory access is initiated to load the B register with m1. Subsequently the A register contents and the B register mantissa are transferred to the B and X registers respectively. The J register is set to 2 for initiation of the normalization process.

If both registers are unoccupied on entry, two memory accesses are initiated at J = 0. The first to load the A register with M1 and the second to load the B register with m1. Note that EEZL will allow the loading of the A register to occur by preventing the E register from being set to 3 until after the A register is loaded. Once both registers are loaded, the A to B and the B to X transfer takes place, with the J register being set to 2 for initiation of the normalization process.

J = 1 - Obtain M2 and m2 From The Stack

At this J register setting, which follows the normalization of M1 and m1, the double length multiplicand M2 and m2 is placed into the B and X registers for normalization prior to initiation of the multiplication.

On entrance to J = 1, any action must await completion of the store memory access initiated at J = 2 for the storage of M3 (m3 is in the A register). The first memory access initiated at J = 1, loads the B register with m2. Note that the S register is decremented twice in order to address m2. Upon completion of the memory access to load the B register with m2, a second memory access is initiated to load the B register with M2. Simultaneously, the S register is incremented to address M2 and the m2 operand is transferred to the X register. At completion of the memory access to load M2, the J register is set to 2 for normalization of M2 and m2. Note that both AROF and BROF are in reset status when J = 1 is exited.

J = 2 - Normalize Operands, Set Sign Result, Initiate Exponent Add

When J = 2 is entered the first time from J = 0 (AROF is on) the double length operand M1 and m1 is normalized. Once normalized a memory cycle is initiated to store M3 in the stack at the address previously storing m1. At the same time, m3 is transferred to the A register. The exponent of the top operand is transferred to the A register to facilitate the exponent add operation when initiated.

When J = 2 is entered for the second time from J = 1 (AROF is off), the double length operand M2 and m2 is normalized. Once normalized, a memory cycle is initiated to store M4 at the address previously storing M2. Exponent arithmetic is initiated by setting Q02F on. In conjunction with initiation of exponent arithmetic, the exponent of the B register is incremented as part of the required exponent adjustment; see write-up on exponent arithmetic.

The sign of the result is set into the B register via interrogation of the sign in the A register. The rules of multiplication state that like signs equals positive results and unlike signs equals negative results. Therefore, if the sign of A is negative, the sign in the B register (result sign) is in complement form.

During normalization of both the A and B registers, the operands are interrogated via N13L • W07L to determine if the operand is zero. If the N register is counted to 13 (indicating that the least significant half of that operand has been shifted to the B register) and the B register mantissa is zero, the complete operand must be equal to zero. In which case the J register is set to 14 for termination.

J = 3 • MWOF - Place M4 In The Stack

Upon completion of the memory cycle, initiated at J = 2, to place M4 in the stack (MWOF), a memory cycle is initiated to store m4 at the same address which stored m2. The contents of the A register (m3) are transferred to the X register while m4 in the X register is transferred to the A register, from which it will be stored.

J = 4 • MWOF - Place m4 In The Stack

When the memory access initiated at J = 3 for the store of m4 is completed, the B register mantissa contents (M4) are transferred to the A register. The sign of the resultant product is also transferred to the A register. Transfer of the flag bit to the A register has no significance and results from the use of common logic.

J = 5 - 1st Add for Mult. Digit ±3 (Single Precision Multiply Sub-Cycle)

Entry to J = 5 is from J = 7 and only occurs when the current multiplier digit is either a plus or minus three (±3). The first cycle, of two required for a plus or minus three multiplier digit, occurs at J = 5. The second cycle occurs at J = 6. Unconditionally the multiplicand in the A register is added to the partial product in the B register and the result placed in the B register.

Q08F to 1

If Q01F is on (complementary addition is being performed) and there is no carry for the most significant digit position (W14L'), the indication is that the partial product in the B register is in complementary form (the partial product is less than zero). Therefore, Q08F is set to record this fact so that (at J = 7), when the partial product is shifted right, a 7 may be set into the most significant digit position of the partial product maintaining the partial product in complementary form.

Q08F to 0

If Q01F is reset (regular addition is being performed) and there is a carry from the most significant digit position (W14L), the indication is that the partial product in the B register is in a true form (the partial product is equal to or greater than zero). Therefore, Q08F is reset to record this fact so that when the partial product is shifted right (at J = 7), a zero may be set into the most significant digit position of the partial product maintaining the partial product as a true number.

Halve A

If the least significant bit of the multiplicand counter is on (M01F) thereby indicating the multiplicand has been doubled to either 2A or 2A', the multiplicand is halved by shifting the A register mantissa and its extension (M(6 through 4) right one binary bit position. The multiplicand counter is decreased by one thus indicating that the multiplicand is in its original form of A or A'.

Double A

If the least significant bit of the multiplicand counter is off (M01F') thereby indicating that the multiplicand is in its original state of 2A, or is in the complement of its original state of 2A', the multiplicand is doubled by shifting the A register mantissa and its extension left one binary bit position. The multiplicand counter is incremented to record the new value of the multiplicand (2A or 2A').

A01F to 1 or 0

If Q01F is set (indicating that the multiplicand is in complementary form), the low order bit of the A register (A01F) is set to provide a valid complement when the multiplicand is doubled. If, however, Q01F is reset (indicating that the multiplicand is in its original form), the low order bit of the A register is reset when the multiplicand is shifted left for doubling of the multiplicand.

J = 6 - (Single Precision Multiply Sub-Cycle)

Entry to J = 6 is from J = 7 when the current multiplier digit is either plus or minus 1, 2 or 4 and is from J = 5 for performance of the second multiply cycle when the multiplier digit is plus or minus three (±3). The multiplicand is unconditionally added to the partial product and the result placed in the B register.

J to 8

If the N register has been counted up to 14 (N14L), indicating that all the multiplier digits have been utilized plus a final addition cycle is being performed to convert the partial product (which is in complementary form) to a true form, the J register is set to 8 for a final normalization of the resultant product.

Halve A

At the time the multiplicand is being added to the partial product as a result of the current multiplier digit, the next multiplier digit is interrogated. Appropriate adjustment of the multiplicand occurs at this time to minimize the time utilized for adjustment of the multiplicand. The next multiplier digit is presently in the least significant digit position of the X register; therefore, interrogation of the A register Right Shift Level (ARSL) will specify if the next multiplier digit requires halving of the multiplicand. For development of the ARSL, see Multiplicand Shift Levels in the Summary of Operation. The term N14L' indicates that this is not the last addition cycle.

Double A

If interrogation of the next multiplier digit indicates that the multiplicand should be doubled (ALSL) and this is not the last addition cycle (N14L'), the A register is shifted left one binary bit position. For development of ALSL, see Multiplicand Shift Levels in the Summary of Operation.

A01F to 1 or 0

If the multiplicand is being doubled (ALSL is true) and the multiplicand is in complementary form (A01F), the least significant bit of the A register is set to provide a valid doubled complement. However, if the multiplicand is in its original form, the least significant bit of the A register must be reset to provide a valid doubled true number. Doubling, in the sense it is used at this time (J = 6), implies going from A to 2A or going from 2A to 4A or the complement thereof.

Q08F to 1 or 0

Q08F is either set to one or zero for the same reasons as specified at J = 5. If the state of Q08F does not require altering, it remains in its current state.

J = 7 - (Single Precision Multiply Sub-Cycle)

Entry to this J register setting is from J = 9 when a multiply sub-cycle is initiated. Subsequent entry to J = 7 (during single precision multiply interval cycle) is from J = 6, after a multiply cycle has been completed. The basic function at J = 7 is to interrogate the current multiplier digit to determine what changes must be made to the value of the multiplicand and to shift the current multiplier digit out of the X register while simultaneously shifting the partial result to the X register. Each time into J = 7, from J = 9, the current multiplier digit is the least significant digit of the multiplier and occupies the least significant position of the X register. Subsequent transfers to J = 7 will find succeedingly more significant multiplier digits in the least significant digit position of the X register.

Basically there are three primary control levels at this (J = 7) state. There are W91L, W91L' and ARSL + N13L • Q01F':

W91L is equal to Q03F' + Q01F' (X03F' + X02F' + X01F') + Q01F (X03F + X02F + X01F) and indicates that an extra adjustment of the multiplicand is not required.

W01L' is equal to W91L switched, and indicates that an extra adjustment of the multiplicand is required. Table 1 on the flow chart illustrates the conditions when an extra shift is required.

ARSL + N13L • Q01F' is used to return the multiplicand to its original configuration upon completion of the multiply operation (N13L • Q01F'), in addition to the normal function of gating the right shift of the multiplicand in the A register (ARSL).

Q03F to 0

If the level W91L' is true, the logical flip-flop Q03F is reset. Q03F being reset causes W91L to go true, thus allowing the normal functions of J = 7 to occur. If W91L' is true, an extra adjustment of the multiplicand is taking place, thus Q03F being reset allows only one clock pulse for the extra adjustment of the multiplicand.

Shift X Register and Increment N Register

Whenever W91L is true at J = 7, the X register is shifted right one octade position thereby shifting the current multiplier digit out of the X register and simultaneously shifting the next multiplier digit into the least significant digit position of the X register. At the same clock pulse, at which this shifting occurs, the current multiplier digit is being interrogated and action taken accordingly in a logical box below. The N register is incremented to keep track of the number of multiplier digits utilized.

X (39 through 37) to 0

If the N register is equal to zero (NEZL), the most significant digit position of the X register is cleared to zero at the time the first multiplier digit is shifted out of the X register. This insertion of a zero happens only during the first shift of the X register and functions as a flag between the multiplier digits and the digits of the product that will be shifted to the X register. This zero will gate a multiplier value of +1 during interrogation of a 14th multiplier digit which will only occur if, after utilization of the 13th multiplier digit, the result in the B register is in complement form, thus requiring an additional add cycle to convert the product to a true number.

Shift Partial Product

If the N register is not equal to zero (NEZL'), the first multiplier digit has already been utilized. The B register and its extensions are shifted right one octade position in conjunction with a digit of the result being shifted to the most significant digit position of the X register. Simultaneously the X register is shifted right one octade position, thereby shifting the current multiplier digit out of the X register and making room for the partial result presently being shifted to the X register. The right shift of the partial product has the effect of causing the next addition to the partial product to be added one digit position to the left as is done in normal arithmetic.

M (10 Through 8) to 7

If Q08F is set, it indicates that the partial product in the B register is in a complementary form; therefore, when the partial product is shifted right one octade position, a 7 is inserted into the extension of the B register mantissa to maintain the partial product in complementary form.

M (10 Through 8) to 0

If Q08F is reset, it indicates that the partial product is a true number (not a complement); therefore, when the partial product is shifted right one octade position, a 0 is inserted into the extension of the B register mantissa to maintain the partial product as a true number.

J to 8

If the N register is equal to 13 (N13L) and Q01F is reset (Q01F'), all 13 multiplier digits have been utilized and the resultant double length product in the B and X register is a true number; therefore, the J register is set to 8 for adjustment of the operand and subsequent initiation of the next multiply sub-cycle or double precision add.

Complement A

If the A register complement Shift Level is true (ACSL) when W91L is also true, the A register and its extension (multiplicand) are complemented. The ACSL is developed from the value of the current multiplier digit and the present status of the multiplicand. See the Multiplicand Shift Levels in the Summary of Operation.

Q03F to 0

If W86L' is true, the current multiplier digit (X*) is not a plus or minus zero or four (X* ≠ ±0 or ±4), therefore Q03F is reset. The logic which had set Q03F was developed from a previous multiplier digit of ±0 or ±4. Q03F was set in anticipation of an extra shift in conjunction with the current multiplier digit. Thus, during interrogation of the current multiplier digit, with W86L' being true, an extra shift of the multiplicand is not required. If an extra shift had been required, Q03F would be reset via W91L'.

J to 5

If W87L is true, that is, if the current multiplier digit develops a multiplier value of plus or minus three (X* = ±3), the J register is set to 5 for execution of the first of two required addition cycles.

Q03F to 1

If W86L is true, the current multiplier digit has developed a multiplier value of plus or minus zero or four (X* = ±0 or ±4), in which case Q03F is set to allow the W91L' to go true. Q03F performs the function of remembering that the current multiplier digit has a value of ±0 or ±4, so that during interrogation of the next multiplier digit it can be determined whether an extra shift is required. If, during interrogation of the next multiplier digit, it is determined that an extra shift of the multiplicand is required, W91L' will go true to allow the extra required shift.

J to 6

If W88L is true, the current multiplier digit has developed a multiplier value of ±1, ±2 or ±4; therefore, the J register is set to 6 for execution of the addition cycle.

Double A

If the A register Left Shift Level (ALSL) is true, the A register and its extension are shifted left one binary bit position. This will double the multiplicand to either 2A or 4A or the complement thereof. The multiplicand counter is incremented to record the new value of the multiplicand.

A01F to 1 or 0

If the multiplicand is in compelementary form (Q01F is set), A01F must be turned on when the multiplicand is doubled. If the multiplicand is in the form of a true number, A01F must be turned off when the multiplicand is doubled.

Halve A

If the A register Right Shift Level is true (ARSL) or the N register is equal to 13 and the product in the B register is not in a complementary form (N13 • Q01F'), the A register and its extension are shifted to the right one binary bit position. The multiplicand counter is counted down to record the new value of the multiplicand. N13L • Q01F' indicate that all multiplier digits have been utilized in the development of the product and that portion of the product in the B register is in the form of a true number and does not require complementing. The multiplicand is always returned to its original configuration upon completion of the multiply cycle, prior to returning to the main double precision multiply cycle.

J = 8 - Get Operands From Stack; Count Sub-Cycles; Exit to AD2L

Double Precision Multiply consists basically of three single precision multiply sub-cycles and one double precision add sub-cycle. The first entrance to J = 8 is upon completion of the first of the three multiply sub-cycles.

Two subsequent returns to J = 8 occur upon completion of the second and third multiply sub-cycles, thereafter double precision Add is initiated. The logical flip-flops AROF and BROF are used to keep count of the sub-cycles.

Initiation of the first multiply sub-cycle (at J = 9), initially occurs with both AROF and BROF in the reset state and subsequently AROF is set. Therefore, the first entry to J = 8 finds AROF set and BROF reset. The multiply of M4 x m3 is completed. As previously described in the summary of operation, the next sub-cycle is the multiply of M3 x m4. Note that the exponent add operation must be completed (Q02F') prior to proceeding. Therefore, when the exponent add is complete, a memory access is initiated to load the A register with m4 and the J register is set to J = 11 where a memory cycle will be initiated to store M6. Thereafter a fetch of M3 will be initiated. BROF is set to the one state, thus indicating (on the next return to J = 8) the completion of the second single precision multiply sub-cycle.

On the second return to J = 8, with AROF and BROF both set, the multiply of M3 x m4 is completed. The required actions at this return to J = 8 are: 1. Storage of the most significant digit of m7 into the M register (bit positions 14 through 12), 2. Transfer of M3, presently in the A register, to temporary storage in the X register, 3. Reset of AROF, to indicate (on the third return to J = 8) the completion of the M4 x M3 single precision multiply sub-cycle, 4. Decrement of the stack to address M4, 5. Initiation of a memory access to load the A register with M4.

On the third return to J = 8, with AROF reset and BROF set, the multiply sub-cycle of (M4 x M3) + M7, to develop the new partial product of M8 + m8, is complete. The required actions at this return to J = 8 are: 1. Transfer of M8 in the B register to the X register, 2. Transfer of m8 in the X register to the B register, 3. Decrement of the stack address to address M6, 4. Initiation of a memory access to load the A register with M6, 5. Setting of logical flip-flops Q02F, Q03F and Q06F for subsequent use in double precision add, 6. Transfer of the operator level to double precision add for

the performance of M8 + (m8 + M6) to develop the new partial product of M9 + m9. Subsequently, during double precision add, the most significant digit of m7 is shifted to the X register if M9 + m9 is unnormalized.

J = 9 • EEZL - Initiate Multiply Sub-Cycles; Initial Multiplicand Adjust

Initiation of the three required multiply sub-cycles occurs at J = 9. The first entrance to J = 9 is from J = 4 and occurs with M4 in the A register, m3 in the X register and both AROF and BROF are in the reset state. AROF is set to indicate that the first of three multiply sub-cycles has been initiated. The B register mantissa is cleared to zero the first time in J = 9 to allow the first partial product to develop in a cleared register. Adjustment of the multiplicand occurs if required.

The second entrance to J = 9 is from J = 8 and occurs with both AROF and BROF in the set state, m4 in the X register, m6 in the B register, and a memory access in progress to load the A register with M3. Upon completion of the memory access, to load the A register with M3, adjustment of the multiplicand occurs if required. The J register is set to 7 for the multiply sub-cycle of (M3 x m4) + m6 to develop the new partial product of M7 + m7.

The third entry to J = 9 is from J = 8 and occurs with AROF reset and BROF set, M3 in the X register, M7 in the B register, and a memory access in progress to load the A register with M4. Upon completion of the memory access to load the A register with M4, adjustment of the multiplicand occurs if required. The J register is set to 7 for the multiply sub-cycle of (M4 x M3) + M7 to develop the new partial product of M8 + m8.

J = 10 - Store S_{10} E_{10} M6; Fetch M3

This J register setting is a part of the adjustment required after completing the first multiply sub-cycle and before initiating the second multiply sub-cycle. Entrance to this J register setting is from J = 11 where a store of S_{10} E_{10} M6 was initiated. Upon completion of the memory access, the stack address is counted down twice, in order to address M3, and a memory access is initiated to load the A register with M3. At the same time the X register contents (m6) are transferred to B and the A register mantissa contents (m4) are transferred to the X register. The exit from J = 10 to J = 9 occurs with the B register containing m6, the X register containing m4, and a memory access in progress to load the A register with M3.

J = 11 • MROF - Initiate Store of S_{10} E_{10} M6; Get m4

Upon completion of the memory access to load the A register with m4 (initiated at J = 8), a memory access is initiated to store S_{10} E_{10} M6. This J register setting is a part of the register set-up in preparation for the second multiply sub-cycle of (M3 x m4) + m6 to develop the next partial product of M7 + m7.

J = 14 - Answer = 0

This J register setting is arrived at from J = 2 if, during the process of normalization, it is determined that either operand is equal to zero. In which case both the A and B registers are cleared to zero and the stack address is decremented to address the next location below the original four double length operands. The M register is cleared to zero and the J register is set to 15 for termination of the operator.

J = 15 - Exit

The A and B registers are marked as occupied. The zero product is marked as an operand. Fetch is enabled and the Syllable Execute Complete Level is true to terminate the operator. If an exponent underflow or overflow condition exists, the appropriate interrupt is set in the interrupt register.

AD2L

The following actions occur after the T register is changed from MU2L to AD2L at J = 8.

J = 8 (AD2L) - Add M6 + m8

Entrance to this J register setting occurs with M8 in the X register, m8 in the B register, and a memory access in progress to load the A register with M6. Flip-flops Q02F, Q03F and Q06F are in the set state and Q04F is in the reset state. Therefore, the add operation adds M6 to m8 and any carry developed is stored in Q01F. The result exponent and its sign (S_{10} E_{10}) are transferred to the B register in preparation of a possible exponent adjustment. The S register is incremented as part of existing AD2L logic and has no significant function as far as MU2L is concerned.

J = 9 (AD2L) - M8 to B; m9 to X

At this J register setting the A register is cleared. The B and X registers are interchanged placing M8 in the B register and m9 in the X register. The J register is set to 10 where a possible carry is added to M8. The decrement of the S register has no significance for MU2L.

J = 10 (AD2L) - M8 + Carry

The A register is cleared; therefore the addition cycle adds a possible carry (Q01F) to M8. If an overflow carry results (Q02F • W13L) branch to J = 11. If no overflow carry results (Q02F • W13L') branch to J = 14.

J = 11 (AD2L) - Scale for Overflow.

If an overflow carry resulted from the addition of a carry at J = 10, the B and X registers contents M9 + m9 are shifted one position to the right with a corresponding increment of the exponent.

J = 14 (AD2L) - Final Normalization and Answer Set Up

If a final normalization of the product in the B and X registers is required (B13L), the B and X registers are shifted left one octade position with a corresponding decrement of the exponent. Simultaneously the most significant digit of m7 (presently stored in the M register, bits 14 through 12) is transferred to the least significant digit (octade) position of the X register. Note that the logic which indicates a transfer back to MU2L (if the answer is zero) has no significance when AD2L is entered from MU2L. This logic results through the use of common logic.

Once the product is normalized, it is placed in the proper register for termination of the operator; S_{10} E_{10} M10 in the A register and m10 in the B register. M (14 through 12) is cleared.

J = 15 - Exit

The A and B registers are marked as occupied. The product is marked as an operand and the exponent position of the B register is cleared. Fetch is enabled and the Syllable Execute Complete Level is true to terminate the operator. If either an exponent underflow or overflow is indicated, the appropriate interrupt is set in the interrupt register.

3.24 DOUBLE PRECISION DIVIDE

For the double precision arithmetic operators, an operand occupies two words. The second word of an operand is considered an extension of the mantissa of the first word of an operand; i.e., the mantissa of the first word of an operand is an integer and the mantissa of the second word of the operand is a fraction. When in the stack, the first word of a double precision operand is in the top of the stack and the second word of a double precision operand is in the second word of the stack. Therefore, double precision arithmetic operators operate on four words in the stack, removing those words from the stack and leaving the result as two words in the stack.

DIVIDE DOUBLE LENGTH

The double length operand addressed by the S register is algebraically divided by the double length operand in the A and B registers. The double length result is left in the A and B registers and the S register is reduced by two. Bit positions 48 through 40 of the least significant word of the double length result are set to zero. The flag bit of the most significant word of the double length result is set to zero. All non-zero results are normalized. The A and B registers are both set to full.

The double length operand in the A and B registers is normalized. If the operand in the A and B registers has a mantissa of zero, the double length operand addressed by the S register is placed in the A and B registers as the result, the divide by zero bit is set in the Interrupt register and the operation is terminated.

If the double length operand in the A and B registers does not have a mantissa of zero, the double length operand addressed by the S register is normalized. If the double length operand addressed by the S register has a mantissa of zero, the A and B registers are set to all zeros and the operation is terminated.

If neither double length operand has a mantissa of zero, the divide operation on the normalized operands takes place. A result of twenty six significant quotient digits is developed.

If the exponent of the result is greater than +63 or less than -63, the exponent overflow bit or exponent underflow bit, respectively, is set in the Interrupt register. The result in the A and B registers contains the correct double length mantissa, mantissa sign and exponent sign. The magnitude of the correct exponent is contained in the exponent field of the A register modulo 64.

SUMMARY OF OPERATION

The Double Precision Divide operator uses the logics of Single Precision Divide, Double Precision Multiply and Double Precision Add to accomplish the result. These double precision operators utilize the logic of the single precision operators, therefore the entire complement of arithmetic operators is required for reference.

Each double length word contains the exponent for all 26 digits of the mantissa in the exponent field of the high order word. This implies that the low order mantissa is correctly scaled to the high order mantissa prior to entry to the operator. Because the low order mantissa is considered an extension of the high order mantissa; i.e., the low order mantissa is considered a fractional part, the exponent field of the low order double length word is meaningless. For these reasons, the exponent arithmetic of double precision divide is identical to that of single precision divide.

In order to describe the procedure by which Double Precision Divide is accomplished, mnemonic terms are given to the various operands. The following mnemonic notations are utilized throughout this description:

S_1 E_1 M_1 m_1 = Divisor; Initial top double length operand
S_2 E_2 M_2 m_2 = Dividend; Initial second double length operand
S_{10} E_{10} Q_3 q_3 = Final Quotient
S = Sign of Mantissa
E = Exponent and its sign
M; Q = High order half of double length mantissa; Quotient
m; q = Low order half of double length mantissa; Quotient
R = Single length remainder

Subscripts:

1 = Initial divisor; Quotient; Remainder
2 = Initial dividend; Quotient; Second remainder
3 = Normalized mantissa; Adjusted exponent of divisor
4 = Normalized mantissa; Adjusted exponent of dividend
10 = Normalized mantissa; Adjusted exponent and sign of result

Arithmetic Procedure

The arithmetic procedure of double precision divide can be approached in several ways. The method chosen for the B 5500 was based on the available capabilities which resulted in the least error. A brief development follows:

The initial operands may be expressed as:

$$8^{E1}\,(M_1 + m_1 \cdot 8^{-13}) \quad \text{and;} \quad 8^{E2}\,(M_2 + m_2 \cdot 8^{-13})$$

The normalized operands may be expressed as:

$8^{E3}(M_3 + m_3 \cdot 8^{-13})$ and; $8^{E4}(M_4 + m_4 \cdot 8^{-13})$

The divide may be expressed as:

$$\frac{8^{E4}(M_4 + m_4 \cdot 8^{-13})}{8^{E3}(M_3 + m_3 \cdot 8^{-13})} \quad (1)$$

Note that $_8E$ refers to the exponent of the complete operand to the base eight. Therefore, the low order mantissa (m) is illustrated raised to 8^{-13} in order to consider the fractional part of the mantissa as an integer.

An ordinary long division of the factors will show that the result becomes an expansion of a binomial series. This series is mechanized in the performance of double precision divide; however, only the first two factors of the series are utilized (the scaling of the third term is outside machine range). The first two factors of the series expressing the division of equation has the form of:

$$8^{E4-E3-26} \quad \frac{M_4 + m_4 \cdot 8^{-13}}{M_3 \cdot 8^{-13}} \cdot 8^{13} - \frac{m_3}{M_3} \quad (2)$$

Equation (2) redefined in terms of Q and q:

$$8^{E4-E3-26} \quad Q_1 + q_1 \cdot 8^{-13} \cdot Q_2 + q_2 \cdot 8^{-13} \quad (3)$$

The formation of the expression in equation (3) is the only requirement of the double precision divide operator. This is observed to represent a product of two double length operands; hence, the requirement for utilization of the double precision multiply operator. The development of the expression of equation (3) is accomplished in a series of sub-cycles which compute the individual terms of the equation:

$$Q1 = \frac{M_4 + m_4 \cdot 8^{-13}}{M_3 \cdot 8^{-13}} \quad (4) \qquad q_1 = \frac{R_1}{M_3 \cdot 8^{-13}} \quad (5)$$

R_1 = remainder resulting from equation 4.

$$Q_2 = 8^{13} - 1 \quad (6) \qquad q_2 = 8^{13} - \frac{m_3}{M_3 \cdot 8^{-13}} \quad (7)$$

Q_2 is developed by simply complementing a register.

Since each divisor is M_3 and has a scale factor of 8^{-13} in equations 4, 5, and 7 it is unnecessary to remove 26 (decimal) from the final exponent, just the value of 13. However, if Q_1 is developed in 14 shifts, an additional one is subtracted from the final exponent value.

If in the development of q_2 in equation 7, the term $(m_3/M_3 \cdot 8^{-13})$ is equal to zero, the term $Q_2 + q_2 \cdot 8^{-13}$ becomes equal to one and the final quotient is equal to the first factor of equation 3, $(Q_1 + q_1 \cdot 8^{-13})$. Therefore this possibility is checked and if found true, the operator is exited in a fashion to load Q1 and q1 in the A and B registers respectively. In the development of q_2, note that its development is not complete until its 8's complement is obtained. The divide cycle for $(m_3/M_3 \cdot 8^{-13})$ equation 7 develops $\overline{q2}$, thereafter its complement is generated and an end around carry is added to develop the 8's complement (q_2).

Another way of viewing the procedure of double precision divide is as follows: The first division cycle in the development of Q_1 and q_1, divides the integer portion of the divisor into the complete dividend (integer and fractional part). This division will develop a partial quotient which is in excess of the correct final value (provided the fractional part of the divisor is not equal to zero). Thereafter, if it can be determined to what degree this partial quotient is in excess of the correct final quotient, it becomes a simple procedure to reduce the value of the partial quotient to the correct final quotient. In essence, this is the procedure followed in double precision divide. This can be accomplished by taking a ratio of the fractional part of the divisor to the integer part of the divisor, and subsequently reducing the initially developed quotient by an amount as indicated by the ratio. In the second major half of equation 3, the term $(m_3/M_3 \cdot 8^{-13})$ develops a ratio indicating the excess value of the initially developed quotient. In order to be able to reduce the partial quotient value via multiplication, it becomes necessary to obtain the complement of the ratio. Thereafter, by multiplying the initially developed quotient by the complement of the ratio, the adjusted (correct) quotient is obtained. The exponent arithmetic will be the same as that for single precision divide because the double length operand exponent refers to the integer portion of the operand.

Figures 3.24-1 and 3.24-2 illustrate the actions occurring during double precision divide in simplified and detailed form respectively.

DETAILED DESCRIPTION

J = 0 - Obtain Divisor Operand From Stack

At this J register setting the two word divisor operand M1 and m1 is obtained from the stack and placed in the B and X registers to allow normalization at J = 2. Fetch is unconditionally inhibited to give precedence to the loading of A and/or B.

If both registers are loaded on entry, the B mantissa is transferred to the X register and the A register contents are transferred to the B register. The J register is set to 2 to commence normalization.

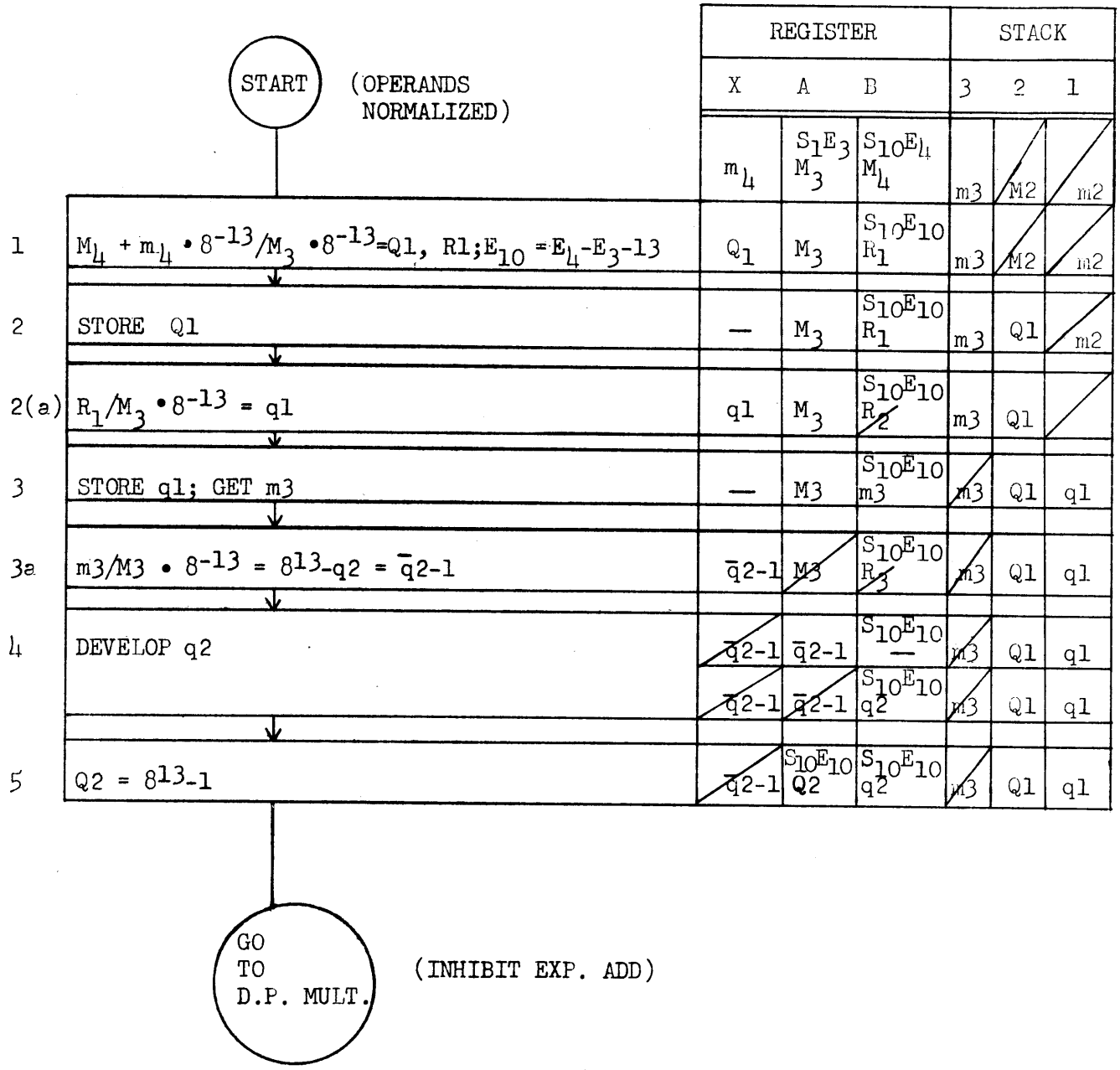
If one register is occupied on entry, its contents are transferred to the A register if not already there, and a memory access is initiated to load the B register with m1. Subsequently the A register contents and the B register mantissa are transferred to the B and X registers respectively. The J register is set to 2 for initiation of the normalization process.

If both registers are unoccupied on entry, two memory accesses are initiated at J = 0, the first to load the A register with M1 and the second to load the B register with m1. Note that EEZL will allow the loading of the A register to occur by preventing the E register from being set to 3 until after the A register is loaded. Once both registers are loaded, the A and B transfer to the B and X registers takes place. The J register is set to 2 for initiation of the normalization process.

J = 1 - Obtain Dividend From The Stack

At this J register setting, which follows the normalization of the divisor, the double length dividends (M2 and m2) are placed in the B and X registers for normalization prior to initiation of the divisior. Note that both AROF and BROF are in the ones state on entrance to J = 1.

On entrance to J = 1, any action must await completion of the store memory access initiated at J = 2 for the storage of M3 (m^3 is in the A register). The first memory access to be initiated at J = 1 loads the B register with m2 (S is decremented twice to address m2). Upon completion of the memory access to load the B register with m2, a second memory access is initiated to load the B register with M2. Simultaneously, the S register is incremented to address M2 with the m2 operand being transferred to the X register. At the completion of the memory access to load M2 into the B register, the J register is set to 2 for normalization of M2 and m2. Note that both AROF and BROF are in a reset status when J = 1 is exited.



3.24-1 SIMPLIFIED ACTIONS OF DOUBLE PRECISION DIVIDE

NOTE: 1. 0 = Setting of Stack Address Register(S) J4 THRU J13
2. ~~~ = Redundant Data
3. - = Cleared Register

DV2L

DIVIDE SUB-CYCLES

			REGISTERS			STACK		
SUB-CYCLE	J		X	A	B	3	2	1
1	4		m_4	$S_1 E_3 M_3$	$S_{10} E_4 M_4$	m_3	~~~	~~~
$\overline{AROF} \cdot \overline{BROF}$	5		m_4	$S_1 E_3 \overline{M_3}$	$S_{10} E_4 M_4$	m_3	~~~	~~~
	5, 6 & 7 S.P. DIVIDE							
	7	$\overline{Q02F} \cdot XI3L$ $N13L \cdot \overline{Q01F}$	Q_1	$\overline{M_3}$	$S_{10} E_{10} R_1$	m_3	~~~	~~~
		$N14L \cdot \overline{Q01F}$	Q_1	$\overline{M_3}$	$S_{10} E_{10+1} R_1$	m_3	~~~	~~~
	8		Q_1	$\overline{M_3}$	$S_{10} E_{10} R_1$	m_3	~~~	~~~
2	9	MWOF	$\overline{M_3}$	$S_{10} E_{10} R_1$	$S_{10} E_{10} Q_1$	m_3	E=11	~~~
$\overline{AROF} \cdot \overline{BROF}$	5		-	$S_{10} E_{10} \overline{M_3}$	$S_{10} E_{10} R_1$	m_3	$S_{10} E_{10} Q_1$	~~~
	5, 6 & 7 S.P. DIVIDE							
	7	N13L · Q01F	q_1	$S_{10} E_{10} M_3$	$S_{10} E_{10} R_2$	m_3	$S_{10} E_{10} Q_1$	~~~
		N13L · $\overline{Q01F}$	q_1	$S_{10} E_{10} M_3$	$S_{10} E_{10} R_2$	m_3	$S_{10} E_{10} Q_1$	~~~
	8		q_1	$S_{10} E_{10} M_3$	~~~	m_3	$S_{10} E_{10} Q_1$	~~~
3	10	EEZL	M_3	$S_{10} E_{10} q_1$	~~~	m_3	$S_{10} E_{10} Q_1$	E=10
$\overline{AROF} \cdot \overline{BROF}$		EWZL	M_3	$S_{10} E_{10} q_1$	~~~	m_3	$S_{10} E_{10} Q_1$	$S_{10} E_{10} q_1$
	11	MROF	M_2	$S_{10} E_{10} \overline{M_3}$	E=3	m_3	$S_{10} E_{10} Q_1$	$S_{10} E_{10} q_1$
	5		-	$S_{10} E_{10} \overline{M_3}$	m_3	m_3	$S_{10} E_{10} Q_1$	$S_{10} E_{10} q_1$
	5, 6 & 7 S.P. DIVIDE							
	7	N13L	$\overline{q}_2$	$S_{10} E_{10}$ ~~~	~~~	m_3	$S_{10} E_{10} Q_1$	$S_{10} E_{10} q_1$
4 A. $\overline{W06L}$ (A≠0)	8	$\overline{AROF} \cdot \overline{BROF}$	$\overline{q}_2$	$S_{10} E_{10} \overline{q}_2$	~~~	m_3	$S_{10} E_{10} Q_1$	$S_{10} E_{10} q_1$
or	12		~~~	$S_{10} E_{10} q_2 - 1$	-	m_3	$S_{10} E_{10} Q_1$	$S_{10} E_{10} q_1$
	13		~~~	$8^{13} - 1$	q2	m3	$S_{10} E_{10} Q_1$	$S_{10} E_{10} q_1$
MU2L→T	0	Q05F · AROF · BROF	~~~	~~~ Q_1	$S_{10} E_{10} q_1$	m_3	$S_{10} E_{10} Q_1$	$S_{10} E_{10} q_1$
4 B.	8	AROF · BROF	~~~	Q	~~~	m_3	$S_{10} E_{10} Q_1$	$S_{10} E_{10} q_1$
W06L	14	MROF · BROF	~~~	E=2	-	~~~	$S_{10} E_{10} Q_1$	$S_{10} E_{10} q_1$
(A=0)		MROF · $\overline{BROF}$	~~~	$S_{10} E_{10} Q_1$	E=3	~~~	$S_{10} E_{10} Q_1$	$S_{10} E_{10} q_1$
EXIT	15			$S_{10} E_{10} Q_1$	$S_{10} E_{10} q_1$	~~~	$S_{10} E_{10} Q_1$	$S_{10} E_{10} q_1$

Printed in U.S.A.

J = 2 - Normalize Operands; Set Result Sign; Initiate Exponent Subtract

This J register setting is entered twice for normalization of operands. When J = 2 is entered the first time from J = 0 (AROF is on) the double length operand M1 and m1 is normalized. If after 13 shifts of the B and X registers (in the process of normalization) the B register mantissa equals zero, the divisor operand must also be equal to zero. In which case, the divide by zero interrupt is placed in the interrupt register by the setting of U15F, U17F and U18F. The E register is set to 2 to load the A register with the most significant half of the dividend operand and the J register is set to 14 where the least significant half of the dividend will be loaded into the B register. Once normalization is completed, m3 is transferred to the B register and a memory cycle is initiated to store m3 in the stack. At the same time the B register contents (M3 and its exponent) are transferred to the A register. The J register is set to 1 where the dividend operand is obtained from the stack.

The second entry to J = 2 is from J = 1, with AROF in the reset state. The double length operand M2 and m2 is normalized. If during the normalization process, it is discovered that the divisor operand is equal to zero (N13L W07L), the operator level is changed to MU2L where the A and B registers will be cleared to zero and the operation terminated. Once the divisor operand is normalized, exponent arithmetic is initiated by the setting of Q02F. The sign of the divisor exponent is complemented because exponents are to be subtracted. The J register is set to 4, at which time the divide opeation is initiated. The resultant sign is set into the B register operand sign bit position (B47F); if A is negative and B is negative, B must be changed to positive, or if A is negative and B is positive, B must be changed to negative, otherwise the sign in B is correct.

J = 4 - Initiate Divide

Entry to J = 4 occurs with M3 in A, M4 in B, and m4 in X. The divisor operand in the A register mantissa plus the A register mantissa extension are complemented in preparation for the first subtraction of the divide cycle. Q01F is complemented to the ones state for the same reason. The N register is cleared to zero because it is utilized to tally the internal divide operation. The J register is set to 5 for the first subtract cycle.

J = 5, 6 And 7

The function of J = 5, 6 and 7 is the performance of a single precision divide cycle. The logic of J = 5 and 6 is identical to that utilized in single precision divide. The major portion of the logic at J = 7 is identical to that of signal precision divide, with some additions necessary as a requirement of double precision divide. Only the additions to the basic single precision divide cycle are described in this write up.

Three single precision divide cycles are performed at J = 5, 6 and 7. The required divide cycle is described in the summary of operation as equations 4, 5 and 7. Each of the divide operations is denoted as a sub-cycle of the total operation. The logical flip-flops AROF and BROF are used to tally the subcycles.

Sub-Cycle One (AROF' • BROF')

In the first sub-cycle, the term Q1 is developed. Note that during the first sub-cycle the divide cycle is terminated only when a 13th quotient digit is formed; whereas during the second and third sub-cycles, the divide cycle is terminated after 13 shifts whether the 13th quotient digit is equal to zero or not. Thus the terms AROF' • X13L maintain the divide cycle even though N = 13. Also note that AROF' forces all quotient digits to be formed by the subtraction process in the first sub-cycle. This will assure that the divisor in the A register, is in complementary form upon completion of the first divide sub-cycle. Exit from the first sub-cycle at J = 7 is via the logic of either N14L • Q02F' or N13L • X13L' • Q02F'. If 14 shifts were required in the development of Q1, the exponent in the B register is decreased by one. Since no other sub-cycle will be allowed to form N14L, the decrement of the exponent only occurs during the first sub-cycle.

Sub-Cycle Two (AROF • BROF')

In the second sub-cycle, the term q1 is developed. The division of the remainder (R1) by M3 • 8^{-13} is performed in the normal manner with no restriction; i.e., it is a regular single precision divide cycle. Exit from this sub-cycle is via the logic of AROF • N13L. The divisor, however, may be in either complement or noncomplement form. If it is in noncomplement form (Q01F is reset), in which case an add restore operation is initiated (which is unnecessary at this time), the divisor is complemented in preparation for the next divide sub-cycle. The J register is set to 8 for register manipulation.

Sub-Cycle Three (AROF • BROF)

In the third sub-cycle, a part of the term q2 is developed. A division of m3/M3 generates $\overline{q2}$, which is a factor described in the summary of operation. The division process is the same as in single precision divide. Exit from this sub-cycle is via the logic of AROF • N13L (the J register is set to 8). At the same time $\overline{q2}$ is transferred to the A register and Q01F is cleared to zero.

J = 8 - Step to Next Sub-Cycle

Sub-Cycle One Exit. Upon completion of the first sub-cycle for the development of Q1, entry to J = 8 occurs with AROF and BROF in the reset state. Unconditionally the N register is cleared in preparation of the next divide sub-cycle. The divisor in the A register is transferred to the X register for temporary storage, the remainder (R1) in the B register is transferred to the A register for temporary storage, and the quotient Q1 is transferred to the B register. A memory access is initiated (E to 11) to store Q1 in the stack. AROF is set to tally the completion of the first sub-cycle. The J register is set to 9 for initiation of the second sub-cycle.

Sub-Cycle Two Exit. Upon completion of the second sub-cycle for the development of q1, entry to J = 8 occurs with AROF set and BROF reset. The A and the X register contents (mantissa only) are interchanged, placing q1 in A and the divisor in X. The B register retains the remainder (R2), which is meaningless at this time since it is not used in any fashion for arithmetic purposes. AROF is set to one redundantly and both the E and J registers are set to 10 to set up the storage of the A register contents (q1). Also, BROF is set to one and the S address is decremented to point at the location previously occupied by m2.

Sub-Cycle Three Exit. Entry to J = 8 at this time occurs with both AROF and BROF set, which is upon completion of the third sub-cycle for the development of $\overline{q2}$. With BROF on, $\overline{q2}$ is checked for a zero mantissa (W06L). If this is the case, the answer is the two previously developed terms (Q1 + q1, see summary of operation). The E register is set to 2 to initiate a memory access to load the A register with S10 E10 Q1 and the J register is set to 14 for subsequent termination of the operator. If the mantissa is not zero (W06L'), the J register is set to 12. The A register mantissa contents are complemented which forms q2-1. The B register is cleared to zero and Q01F is set to form an 8's complement on the subsequent add cycle at J = 12 in order to develop q2 (Q2-1 + 1 = q2). Since AROF is on, BROF is redundantly set and the S register is decremented to point at S10 E10 Q1.

J = 9 - Store Q1; Initiate 2nd Sub-cycle

When the store of Q1 is complete (MR)F), the registers are restored by transferring the X register contents (divisor) back to the A register, the A register contents (R1) are transferred back to the B register and the X register is cleared in preparation of the next quotient development. The J register is set to 5, initiating the divide cycle.

J = 10 - Store q1; Initiate Read of m3

Entrance to J = 10 is from J = 8, where a store of q1 was initiated. Upon completion of the memory access, the stack address is counted up twice via EEZL in order to point at the low order mantissa of the normalized divisor (m3). In conjunction with the second count up of the stack address (EWZL), a memory access is initiated to load the B register with m3. Simultaneously, the X register contents, complemented divisor ($\overline{M3}$), are transferred to the A register. The J register is set to 11 where the 3rd sub-cycle is initiated.

J = 11 - Initiate 3rd Sub-cycle

Upon completion of the memory access (MROF), the X register is cleared in preparation of the divide cycle to develop $\overline{q2}$. The J register is set to 5 to initiate the divide cycle.

J = 12 - Develop Q2 q2

Entry to this J register setting indicates that $\overline{q2}$ was not equal to zero; therefore, a one is added to q2-1 (via Q01F; with B cleared) to develop q2. In essence, the 8's complement of q2 is developed, $\overline{q2}$ was complemented at J = 8 and with the addition of a one at this time its 8's complemented is developed.

Simultaneously, the B register mantissa (equal to zero) is complemented and transferred to the A register. Thus the A register mantissa becomes a complemented extension of the B register mantissa. The A register contains Q2 and the B register contains $q2 \cdot 8^{-13}$. The sign bit A47F is reset to insure that the result sign (already developed) is not altered during double precision multiply. Q05F is set to the one state to inhibit exponential arithmetic during MU2L. The J register is set to 13 where MU2L is initiated.

J = 13 - Change Operator Level

The most significant octade positions of the A register mantissa (bits 39 through 37) are cleared to zero insuring a single position left shift of the $Q2 + q2 \cdot 8^{-13}$ double length operand at the start of MU2L. Q01F is cleared to zero becuase the logic of MU2L assumes it to be initially in a cleared status. T09F is set and T10F is reset changing the operator level to MU2L. The J register is set to zero for commencing MU2L at the beginning of the operator. Note that there will not be a return to DV2L at the completion of the MU2L operator (see MU2L write-up).

J = 14 - Answer = $Q1 + q1 \cdot 8^{-13}$

Entry to this J register setting occurs if the term $\overline{q2}$ is found to be equal to zero (at J = 8), in which case the fractional part of the normalized divisor (m3) was equal to zero. Therefore, the final quotient is the result of the integer division which developed $Q1 + q1 \cdot 8^{-13}$. (See summary of operation.) At the completion of the memory access, initiated at J = 8 to load the A register with S10 E10 Q1, the stack address is decremented and a memory access is initiated to load the B register with $q1 \cdot 8^{-13}$. BR0F is reset so that at the completion of the memory access to load the B register, the J register may be set to 15 for termination of the operator.

J = 15 - Exit; Round

The A register flag bit is set to zero to insure the double length operand is marked as an operand. AR0F and BR0F are set to indicate valid information. The B register flag, sign and exponent bits are cleared since these bit positions are invalid. Fetch is enabled and the syllable execute complete level is true to terminate the operator. If either an exponent underflow or overflow condition is detected, the appropriate interrupt is set into the interrupt register.

3.25 EXPONENTIAL ARITHMETIC

In any Multiply or Divide operator (except Remainder Divide), if the operator logic sets Q02F, the exponential arithmetic is initiated and proceeds without regard to any J register setting. The execution of the exponential arithmetic consists essentially of an algebraic addition of the contents of two binary counters: the A register exponent field with its extension, and the B register exponent field with its extension. In the following discussion, reference to the exponent field, implies the complete exponent field including its extension.

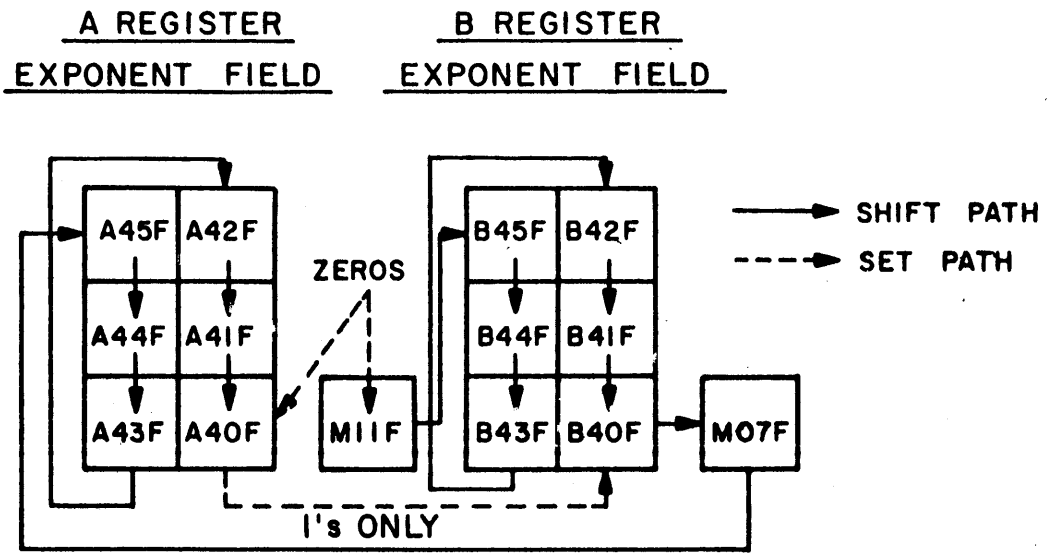


FIGURE 3.25-1 EXPONENT ARITHMETIC REGISTERS

Figure 3.25-1 illustrates the A and B register exponent fields; also indicated are the set and shift paths which are utilized. The procedure used in the performance of the algebraic arithmetic of the exponent fields is to modify (increment/decrement) the B exponent field by the contents of the A exponent field in a serial fashion. This can best be seen by following a simplified sequence of events: The least significant bit position of the A exponent field (A40F) is interrogated. If it is set, the B exponent field is incremented or decremented (depending on a sign comparison). If it (A40F) is reset, no modification of the B exponent field is required. Next, the A and B exponent fields are shifted to the right (one binary bit position), with the contents of B40F being circulated to the A exponent field (M07F bit position). At this time the complete procedure is repeated, only it is noted that now A40F contains the previous contents of A41F; i.e., the contents of A40F have the significance of a binary two. Each time the procedure is repeated, the contents of A40F has a correspondingly more significant binary value. The procedure is completed when all bit positions are utilized, and at this time the sum will be in the A register.

An additional requirement of the exponential arithmetic procedure is that a constant amount, +12 (decimal) for Multiply or -12 (decimal) for Divide must be algebraically added to the B exponent field. In Multiply and Divide, the final exponent value is adjusted to be relative to the least significant position of the register.

The exponent arithmetic is performed under control of a binary counter, the M register, bit positions 15 through 12 which are decoded in a 8, 4, 2, 1 fashion. This counter has two output levels:

W15L indicating the M counter = 2 or 4
W16L indicating the M counter ≠ 2 or 4 or 9

The W15L is used to gate the addition of a 12 (4 and then 8) at the appropriate time. W16L gates the basic increment or decrement of the B exponent field by the contents of the A exponent field. When W16L (M15F • M12F) equals 9, the sum has been assembled in the A exponent field.

DETAILED DESCRIPTION

As soon as Q02F is set to the one state, the exponent arithmetic is started. At the start of the M counter it is equal to zero, thus the level W16L is true. With W16L true, the A4OF bit position is interrogated. If A4OF is set, the B exponent field is either incremented or decremented. This depending on whether the exponent signs are equal or not. Simultaneously, the A4OF bit is reset to the zero state. If, however, the A4OF bit were in the reset state, no modification of the B exponent field is required. In any event, the A4OF bit in the reset status indicates that the bit position has been interrogated and the corresponding action has been taken.

Now that the least significant bit of the A exponent field has been utilized to possibly modify the B exponent field, both the A and B exponent fields are shifted one binary bit position to the right. The bit contents shifted out of B4OF are transferred to M07F (the most significant bit position of the A exponent field). Simultaneously, the M counter is counted up by one.

With the M counter now equal to 1, the logical level W16L is again true. With W16L true, the above described steps are repeated. Only this time the contents of A4OF will have the binary value of 2. Each time the exponent fields are shifted, the A4OF bit position contains a more significant binary value; i.e., 4, 8, 16, 32.

When the M counter becomes equal to 2, the logical level W16L is false and W15L is true. At this time the B exponent field will be unconditionally incremented or decremented in accordance to the following scheme:

Increment B if:	Multiply and B is positive or Divide and B is negative
Decrement B if:	Multiply and B is negative or Divide and B is positive

Since the B exponent field has been shifted twice at this time (M = 2), the B exponent field is actually being incremented or decremented by the value 4. Simultaneously, the M counter is counted up to 3, but neither the A or B exponent fields are shifted. With the M counter equal to 3, the level W16L is again true and the actions resulting from this level are performed. The next setting of the M counter is 4 which again gates the level W15L true. With the level W15L true, the previously described actions are repeated. At this time, however, the B exponent field is incremented or decremented by the value of 8 completing the algebraic addition of a plus or minus 12 (decimal).

When the M counter is equal to 9, the result will be in the A exponent field in either a true or complementary form. The result would be in complementary form if the result had gone negative, in which case it must be decomplemented. The logic of this follows: if, during the exponent arithmetic operation, the B exponent field is equal to zero (W72L • M11F') and the B exponent must be decremented (WBDL), the next intermediate result will appear in complementary form (M11F will be set). In this case, the B exponent field is shifted right, M11F must not be cleared. The logical flip-flop Q06F is set to remember that the result has gone negative, and on subsequent shifts of the B exponent field, Q06F prevents the clearing of M11F. Should the B exponent field go positive again due to a subsequent increment, M11F will automatically be cleared via the internal register count logic. Therefore, Q06F and M11F being set at M = 9 (Q06F • M11F), indicates the result is in complementary form.

WBDL = + W15L • T06L • B46F
+ W15L • T12L • B46F'
+ T13L • A40F • W16L • A46F' • B46F
+ T13L • A40F • W16L • B46F • A46F
T06L - Single and Double Precision Multiply
T12L - Single and Double Precision Divide
T13L - S.P. and D.P. (Multiply and Divide) and Integer Divide

If the result is in a true form, the M11F bit position could again be in the set state. If M11F is set, Q06F being in the reset state indicates that the result is an overflowed sum in a true form and is not a complemented sum. For example, suppose that the initial value of the B exponent field is 077 and the increment value is true. The result would be (octal) 001 + 077 = 100, M11F would be on, and yet the sum in the B exponent field is not in a complementary form. In that case, Q06F being reset, supplies the necessary information for the next setting of M11F; i.e., M11F is to be reset.

M11F'. The result obtained in the A exponent field is in true from and is transferred to the B exponent field. The highest possible value that the result may obtain is 213 (octal). This result cannot be stored in the A exponent field. An additional flip-flop would be required to store the leading 2. Therefore, B40F must be tested to detect an eventual overflow. W02F and M12F are unconditionally cleared, indicating the operation is completed.

M11F. The result is obtained in complementary form, indicated by both M11F and Q06F being in the set state. Q06F is utilized at this time to gate a two-step action. With M11F • Q06F true, B40F is cleared. When the decrement level has been true during exponent add, the result can never overflow into B40F. Simultaneously, the A exponent field is decremented by one, so that a subsequent 7's complement yields an 8's complement. Q06F is unconditionally reset. Thereafter, with M11F • Q06F' true, the A exponent field and the exponent sign of the B exponent field (B46F) are complemented. M11F is unconditionally cleared and the logic previously described with M11F in the reset state prevails.

3.26 MARK STACK
ENTER CHARACTER MODE

MSOL - 0441
ECML - 4441

MARK STACK

The contents, if any, of the A and B registers are pushed into the stack in memory. The mark stack control word is constructed and stored in the top of the stack in memory. The F register is set to the address of the cell in which the mark stack control word has been stored. If the mark stack flip-flop is zero and the Processor is in the sub-program level, the mark stack control word is stored in the cell addressed by the contents of the R register plus seven. The mark stack flip-flop is set to one.

ENTER CHARACTER MODE

In-line Character Mode entry will allow character mode segments to be entered from a program without the necessity of using a program descriptor. After stack adjustment, the operator will construct a Return Control Word (RCW) and place it in the top of the stack and set the Processor in Character Mode. The C register and L register are not affected by this operator. Control will continue in sequence.

SUMMARY OF OPERATION

MARK STACK

The mark stack control word is constructed in a cleared B register and is subsequently stored in the top of the stack. If there have been no previous mark stack control words (mark stack flip-flop is off), this mark stack control word is also stored in the program reference table. The organization of the mark stack control word is as follows:

Identification bits - Positions 48, 47 and 45

Bit 48 = 1
Bit 47 = 1
Bit 45 = 0

R Register contents - Bit positions 42 through 34
Mark Stack flip-flop- Bit position 32

Program Level flip-flop - Bit position 31
0 Program level/1 Sub-program level
F Register contents - Bit positions 30 through 16

The mark stack flip-flop is unconditionally set. The address of the MSCW is placed in the F register and the operation is terminated.

ENTER CHARACTER MODE

Upon entry to this operator there are two requirements which must be met; the top word in the stack must be a destination address, and the previous control word in the stack must be a Mark Stack Control Word.

The top word in the stack is put in the A register and the B register is cleared for the building of a Return Control Word (RCW). A RCW is constructed and pushed down into the stack. The C and L registers are stored in the RCW in the normal way, but it is a redundant operation.

The Processor is placed in sublevel (SALF) and character mode (CWMF), the mark stack flip-flop is reset (MSFF to 0) and the operator is terminated by changing the operator in the T register to a Recall Destination Address (RDAL) with the J register set to two.

DETAILED DESCRIPTION

JOOL

If either or both the A and B registers are occupied on entry, the appropriate memory accesses are initiated to store the contents in the stack. If both registers are initially empty or when the memory access is initiated to store the B register contents (if one is required), the J register is set to 4 to continue the operation.

B to A

If the A register is empty and the B register full (AROF' • BROF) and an ECML operator in the T register (T12F), the contents of the B register (a destination address) are transferred to the A register for later interrogation and use.

E to 2

With both the A and B registers unoccupied (AROF' • BROF') and the ECML operator in the T register (T12F), initiate a memory cycle to obtain the top word of the stack in the A register.

JO4L

B to 0

Upon completion of the memory access initiated at J = 0 or if no memory access was initiated (EEZL), the B register is cleared to facilitate the construction of either the mark stack control word or the return control word, depending upon the operator.

J to 5

If this is the Mark Stack operator (T12F'), proceed to J = 5 to construct the mark stack control word.

J to 7

If this is the Enter Character Mode operator (T12F), proceed to J = 7 to construct the return control word.

J05L

The mark stack flip-flop is unconditionally set indicating that a mark stack control is in the stack. The flag and identification bits are set (B48 and B47) to one (B45 was set to zero at J = 4) to identify this word as a mark stack control word. If the mark stack flip-flop is reset and the Processor is in sub-program level (MSFF' • SALF), the J register is set to 6 for subsequent storage in the PRT. If not in sub-program level or a previous MSCW had been constructed, the J register is set to 1 where the storage address of this MSCW is placed in the F register.

J06L

Upon completion of the memory access initiated at J = 5, for storage of the mark stack control word (EEZL) in the stack, the base address of the PRT plus 7 is placed in the M register. The nine bits of the R register (base of the PRT) are transferred to the high order bit positions of the M register. The M register bits, 6 through 4 are cleared to zero and bits 3 through 1 are set to 7. Thus, the base of the PRT plus 7 is assembled in the M register. A memory access is initiated to store the MSCW at the indicated address. The J register is set to 1 for storage of the MSCW stack address in the S register.

J01L

Upon completion of the memory access initiated at either J = 5 or J = 6, the S register contents are stored in the F register. The syllable execute complete level is true to terminate the mark stack operator.

J07L - Enter Character Mode Operator.

A Return Control Word is constructed by setting the values of the L, C, F, H, V, G, and K registers into the B register. B48F and B47F are set to one to identify the control word. The storing of the C and L registers is not necessary as they continue to count in sequence as they normally would.

S + 1

Increment the S register by 1 to store the RCW at the top of the stack.

E to 11

Set the E register to 11 to initiate the store of the RCW.

J to 9

Set the J register to 9 in preparation of terminating this operator.

J09L - Enter Character Mode Operator

When the RCW is stored in memory (EEZL), the mark stack flip-flop is set to zero (MSFF to 0) and the Processor is set into sublevel (SALF to 1). The contents of the S register are set into the F register to allow the F register to address the RCW. The R register is cleared, R is not used as an address register in character mode, and the Processor is placed in character mode (CWMF to 1). The S register is interchanged with bits 30=> 16 of the X register. This has the effect of:

1. Clearing the S register, X register is equal to zero at this time
2. Placing into bits 30=>16 of the X register the address of the return control word.

Instead of normal operator termination by SECL, control is transferred by forcing a RDAL (recall destination address) operator in the T register and setting the J register to two. Also, the contents of the A register (destination address) are transferred to the B register. At J = 2 of the RDAL operator the contents of the B register are expected to contain the destination address.

Unconditionally the A register is set to empty (AROF to 0) and the V and K registers are cleared.

3.27 EXIT

REWL - 0435

Registers A and B are marked empty. The word addressed by the F register, the return control word, is placed in the B register.

If the flag bit of the word in the B register is 1, the operation is continued. If the flag bit is 0 and the Processor is in the normal state, the Flag Bit Interrupt is set and the operator exited with the Return Control Word left at the top of the stack. If the flag bit is 0 and the Processor is in the control state, the operator is terminated but the interrupt is not set.

The C, L, G, H, K and V registers are set to the contents of their respective fields of the return control word in the B register. The S register is set to the contents of the F register field of the return control word in the B register.

The word now addressed by the S register, the mark stack control word, is read from memory into the B register. The R and F registers are set to the contents of their respective fields, of the mark stack control word. The mark stack flip-flop and the program level flip-flop are set to the contents of their respective positions of the mark stack control word. The S register is decreased by one. The A and B registers are set to empty.

The mark stack bit of the word in the B register is examined. If this bit is zero the operation is completed.

If the mark stack bit is one, the program level bit is examined. If the program level bit is zero, indicating program level, the operation is completed.

If the program level bit is one, indicating sub-program level, the word addressed by the F register field of the mark stack control word, the previous mark stack control word, is placed in the B register. The mark stack bit is examined. If the mark stack bit is one, the process of reading the previous mark stack control word and examining its mark stack bit is repeated until a mark stack control word with the mark stack bit set to zero is placed in the B register. The contents of the B register is stored in the cell addressed by the contents of the R register plus seven. The operation is completed.

SUMMARY OF OPERATION

The Exit operator functions to exit from a sub-routine to either program or sub-program level. The C, L, K, G, V, H, R and F registers are restored to their respective values prior to sub-routine entry. The S register is set to one less than the address containing the mark stack control word.

DETAILED DESCRIPTION

J = 0

Fetch is inhibited (ICFL) to give precedence to loading of the B register. The A and B registers are marked as unoccupied. The address previously in the F register (address of the Return Control Word), is transferred to the S register so that a memory access may be initiated (E to 3) to load the B register with RCW.

J02L

Upon completion of the memory access, if the Flag bit of the word in the B register is on then reset the mark stack flip-flop and the sublevel flip-flop. Transfer the address of the Mark Stack Control Word to the S register, set E to 3 and J to 3 to access the MSCW.

If $\overline{B48F}$ then the area referenced by the Return Control Word is not present, set J to 12 in preparation for an exit with pseudo presence bit interrupt (flag bit interrupt).

J = 3

This J register setting is executed prior to completion of the memory access initiated at J = 2 to load the B register with MSCW. At this time the B register contains the return control word (RCW). The address of the mark stack control word (contained in the RCW, bits 30 through 16) is temporarily stored in the X register. It will be placed in the S register at J = 8. The various registers are restored to their prior value and a fetch is initiated to load the C register with the word containing the syllable following that which initiated the sub-routine. J is set to 4.

J = 4

Upon completion of the memory access, which was initiated at J = 2 to load the B register with the mark stack control word, the R and F registers are restored to their prior contents presently stored in the MSCW. The mark stack and sub-program level flip-flops are restored to the setting contained in the MSCW. If this Exit operator terminates a sub-routine in program level (sub-program level bit B31F is reset) this MSCW was the first one constructed (mark stack flip-flop is reset), the J register is set to 8 and no further action occurs at this J register setting.

If, however, the exit is to sub-program level and the MSCW just accessed is not the first one constructed (B32F • B31F), then the previously constructed MSCW's must be accessed until a MSCW is found that refers to the RCW. In this case, the S register is set from the MSCW contents and a memory access is initiated to load the B register with the previous MSCW. This process is repeated until a MSCW is found that does point to a RCW. If more than one MSCW exits, the logical flip-flop Q06F is set, thus preventing the R and F registers from being set by other than the MSCW first accessed by the exit operator.

J = 8

The address of the MSCW, previously stored in the X register at J = 3, is transferred to the S register. The J register is set to 9 where the S register is decremented. If more than one MSCW were accessed at J = 4, Q06F would be set. Therefore, the last MSCW accessed (which points to a RCW) is stored at the address R + 7. The R contents were obtained from the first MSCW interrogated during the exit operator.

J = 9

Upon completion of the memory access to store the MSCW, or if none were initiated (EEZL), the S register is decremented to point at the address below the MSCW.

J12L

Accessing of an RCW marked not present causes entry to this J register setting, exit this operator and set BROF to 1 in order to keep the RCW in the top of the stack. If in normal state, set the Flag Bit Interrupt, which will be interpreted by the MCP as a Presence Bit Interrupt.

3.28 RETURN NORMAL
RETURN SPECIAL

RNML - 0235
RSPL - 1235

If the A register is empty, a word is placed in the A register by stack adjustment and the A register set to full. If both the A register and the B register are full, the B register is set to empty.

If the operator is Return Normal, the word addressed by the F register, the return control word, is placed in the B register.

If the operator is Return Special, the word addressed by the S register, the return control word, is placed in the B register.

If the flag bit of the word in the B register is 1, the operation is continued. If the flag bit is 0 and the Processor is in the normal state, the Flag Bit Interrupt is set and the operator exited, with the A and B registers marked full. If the flag bit is 0 and the Processor is in the control state, the operator is terminated, but the interrupt is not set.

The C, L, G, H, K, and V registers are set to the contents of their respective fields of the return control word in the B register. The S register is set to the contents of the F register field of the return control word in the B register.

The word addressed by the S register, the mark stack control word, is read from memory. The R and F registers are set to the contents of their respective fields of the mark stack control word. The Mark Stack Flip-flop and the Program Level Flip-flop are set to the contents of their respective positions of the mark stack control word. The S register is decreased by one.

The mark stack bit of the word in the B register is examined. If this bit is zero, the operation is completed.

If the mark stack bit is one, the program level bit is examined. If the program level bit is zero indicating program level, the operation is completed.

If the program level bit is one, indicating sub-program level the word addressed by the F register field of the mark stack control word, the previous mark stack control word, is placed in the B register. The mark stack bit is examined. If the mark stack bit is one, the process of reading the previous mark stack control word and examining its mark stack is repeated until a mark stack control word with the mark stack bit set to zero is placed in the B register. The contents of the B register is stored in the cell addressed by the contents of the R register plus seven. The operation is completed.

The subsequent action of the return operation is similar to that of the operand or descriptor call syllable. If the syllable indication in the return control word indicated an operand call syllable, the subsequent action performed is described by the operand call flow chart. If the syllable indication in the return control word indicates a descriptor call syllable, the subsequent action performed is described by the descriptor call flow chart.

SUMMARY OF OPERATION

The function of the return normal/special operator is to facilitate returning to the main routine following the completion of the sub-routine.

The normal return operator is used in sub-routines where no parameters are needed and there is no nesting of sub-routines (F register contains address of return control word).

The special return operator is used in sub-routines where parameters were inserted in the sub-routine by the main program (at completion of sub-routine the return control word will be the second word in the stack).

The return normal/special operator also places the result of the sub-routine in the A register.

Upon entry into the operator, the stack is adjusted so that the A and B registers are loaded. The B register will be loaded with the return control word addressed by the S register when the operator is a return special, and by the F register when the operator is a return normal. If the flag bit in the return control word is on then restore the H, V, G, K, L and C registers to their configurations prior to entry into the sub-routine. The S register is set to the address of the top mark stack control word and it is subsequently loaded into the B register. It is tested for being the first mark stack control word and its address is temporarily stored in the X register. If this mark stack control word is not the first, the next lower mark stack control is loaded into the B register and tested for being the first. This operation continues with each descending mark stack control word until the first is located. This mark stack control word is then stored in the PRT. The address temporarily stored in the X register (address of top MSCW) is placed in the S register. A descriptor or operand call is set into the T register dependent upon the indicator in the return control word. The operation continues as a descriptor or operand call. If the flag bit of the return control word was off, terminate the operator with the Flag Bit Interrupt set and the A register containing the return control word.

DETAILED DESCRIPTION - RNML + RSPL

J = 0

The fetch operation is inhibited at this time (ICFL), fetch is initiated at J = 9. After stack adjustment the A register will contain the result of the sub-routine. Therefore, it is unconditionally marked as loaded. The B register is marked as empty to allow it to be loaded as necessary during the operation.

Upon entry, if both the A and B registers are empty, the E register is set to 2 to initiate a load of the A register. If the A register is empty and the B register is full, the contents of B are transferred to the A register. The stack address is decremented by one when the memory read access has been obtained (MROF) during a load of either the A or B register.

Once the A register is loaded, the subsequent action loads the B register with the return control word which will facilitate a return to that part of the program which

initiated going to this sub-routine. The address of this return control word, when executing a return normal operator, is contained in the F register. When the operator is a return special, the return control word will be the second word in the stack and is addressed by the S register directly.

The E register is set to 3 to initiate a load of the B register by the S register. If the operator is a return special, the S register contains the correct address at this time. If the operator is a return normal (RNML), the address contained in the F register will be transferred to the S register at this time to provide the address needed to locate the return control word. The J register is set to 2 allowing the next sequence of actions.

J = 2

When the memory cycle for the loading of the B register with the RCW is completed (EEZL) and the flag bit of the RCW is on (B48F), the top mark stack control word address (contained in the return control word) is transferred to the S register. The mark stack flip-flop and the sub-program level flip-flop are cleared prior to obtaining their new settings from the mark stack control word. The J register is set to three to gate the next sequence of actions.

If the flag bit in the RCW is off (EEZL B48F'), then the only action to occur at J = 2 is to set the J register to 12.

J = 3

The H, V, G, K, L, and C registers are restored to their configurations prior to entry into this sub-routine. The E register bit 16 is set to 1 to initiate a Fetch operation. At the same time, 39 bits of the return control word in the B register are temporarily stored in the S register to retain the top mark stack control word address for use at J = 8. The J register is set to 4 to allow the next sequence of actions.

Q05F to 1

Logical flip-flop Q05F is set to one if the descriptor call operator indicator (B46F) is on in the return control word. The setting of Q05F will allow either return operator to be completed as a descriptor call operator. If B46F is off, the return operator will be completed as an operand call operator.

J = 12

Entry to this J count is from J = 2 only in case the flag bit is off in the return control word. If in normal state (NCSF), set the Flag Bit Interrupt and exit this operator. Mark the B register as valid; it contains the RCW with the flag bit off. The flag bit is interpreted by the MCP at a presence bit interrupt.

J = 4

Upon completion of the memory access, which was initiated at J = 2 to load the B register with the mark stack control word (MSCW), the R and F register fields are transferred to the R and F registers, respectively. The mark stack and/or sub-program level flip-flops are set to one, if the mark stack and/or sub-program level bits (B32F, B31F) are on in the mark stack control word. If this MSCW is the first

that was developed during the sub-routine (MSFF bit B32F is reset) or the return is to program level (sub-program level bit B31F is reset), the J register is set to 8 and no further actions occur at this J register setting.

If, however, this is not the first MSCW developed during the sub-routine and exit from the sub-routine is to sub-program level (B32F • B31F), then the previously constructed mark stack control words must be accessed until the first MSCW is located. Therefore, the S register is set from the F register field of the MSCW, presently in the B register, and a memory access is initiated to load the B register with the next lower MSCW. This process is repeated until the first MSCW is located. If more than one MSCW exists, the logical flip-flop Q06F is set to prevent the R and F registers from being set from other than the top MSCW (the first one accessed during this operator). When the first MSCW is located, the J register is set to 8.

J = 8

The address of the top MSCW, previously stored in the X register at J = 3, is transferred to the S register. The X register is cleared to zero. It may be used during the operand/descriptor call operator. The J register is set to 9 where the operator is changed to an operand/descriptor call syllable. If more than one MSCW were accessed at J = 4, Q06F would be set. Therefore, the last MSCW accessed at J = 4 is stored at the address R + 7. The R contents were obtained from the first MSCW interrogated during the return operator.

J = 9

Upon completion of the memory access to store the MSCW, or if none were required (EEZL), the S register is decremented by one to point at the address below the top MSCW; i.e., the top stack address prior to sub-routine entry.

The T register coding for a return normal/special has T01F set and T02F reset. At this time the T register is changed to a call syllable by setting T02F to one. The type of call (operand or descriptor) is dependent upon the status of Q05F, its status was established at J = 3. If Q05F is reset, T01F is reset to indicate an operand call. Otherwise the operator is a descriptor call. The J register is set to 2 to establish the entry point of the operand/descriptor call syllable. (Refer to the Operand/Descriptor Call operator flow chart and write-up).

3.29 EXCHANGE

F FIELD TO CORE FIELD	EXCL - 1025
F FIELD TO F FIELD	FCXL - 1425
CORE FIELD TO C FIELD	FFXL - 3425
CORE FIELD TO F FIELD	CCXL - 5425
	CFXL - 7425

The function of the Exchange operator is to exchange the top two words in the stack.

The remaining operators are referred to as Fixed Field Transfer Operators. Their function is to transfer a 15 bit field from the A register to the B register.

SUMMARY OF OPERATION

Upon entry to the operator, if either the A or B registers are unoccupied, stack adjustment is initiated to load both of the registers. If the operator is the Exchange operator, then the A and B register contents are exchanged and the operator is terminated.

The fixed field transfer operators use the two top words in the stack. The contents of the specified field of the top word is transferred to the specified field of the second (lower) word. The fixed fields are termed either 'F' which encompasses bits 30=>16 of either word or "Core" which encompasses bits 15=>1 of either word. On this basis, two words with two positions in each gives four combinations of exchanging the fields:

Transfer 'F' Field to 'F' Field - FFXL - 3241

The contents of bits 30=>16 of the A register are transferred to bits 30=>16 of the B register.

Transfer 'F' Field to "Core" Field - FCXL - 3241

The contents of bits 30=>16 of the A register are transferred to bits 15=>1 of the B register.

Transfer "Core" Field to "Core" Field - CCXL - 6241

The contents of bits 15=>1 of the A register are transferred to bits 15=>1 of the B register.

Transfer "Core" Field to 'F' Field - CFXL - 7241

The contents of bits 15=>1 of the A register are transferred to bits 30=>16 of the B register.

In all cases, the B register bits not transferred remain unchanged and the A register is marked empty.

DETAILED DESCRIPTION

J = 0

$\overline{\text{BROF}}$

If the B register is unoccupied, initiate a memory read (E to 3) to load the B register. Set J = 1 to complete the memory operation.

$\overline{\text{AROF}}$ BROF

If the A register is unoccupied and the B register is occupied ($\overline{\text{AROF}}$ • BROF), initiate a memory read operation to load the A register (E to 2) and exit this J count to J = 2 to complete the memory access initiated.

The stack adjustment here differs from normal stack adjustment in that if the A register is empty and the B register is loaded, setting the E register to 2 will actually take the second word in the stack and place it in the top of the stack (A register). This accomplishes an exchange of the two top words in the stack.

AROF • BROF

If both the A and B registers are initially occupied, or if only the A register was initially occupied, then the condition of AROF • BROF will be True at J = 0. The A and B register contents are exchanged (B to A).

EXCLD1 • $\overline{\text{T09F}}$

If the operator is an Exchange with the A and B registers loaded, go to J = 3 in preparation to terminate this operator.

EXCLD1 • T09F

If the operator is one of the Fixed Field Transfers, then go to J = 7 to make the transfer.

J = 1

Upon completion of the load B memory access, the stack address is decremented to address the next word in the stack. The B register is marked as occupied and the J register setting is returned to zero.

J = 2

Upon completion of the memory access to load the A register, the A register is marked as occupied and the stack address is decremented. The J register is set to 3 for termination of the operator. The exchange of the top two words in the stack has been accomplished via stack adjustment.

J = 3

The syllable execute complete level is true to terminate the operator. This J register setting is used only for the Exchange operator.

J = 7

Upon entering this J count the top two words in the stack have been exchanged, either at J = 0 with AROF • BROF or via stack adjustment, and the operator is one of the Fixed Field Transfers.

The unconditional transfer of A to B makes both registers identical, containing the second word in the stack. The unconditional transfer of the S register contents to the X register bits 30=>16 is to save the S register contents.

$\overline{\text{T12F}}$ / T12F

Bit 12 of the T register determines the field being transferred from. If it is on (T12F), then the operator is either CFXL or CCXL and the transfer will be from the CORE or C field of the top word in the stack which is in the B register. If bit 12 is off (T12F'), then the operator is either FCXL or FFXL and the transfer will be from the F field of the top word in the stack, bits 30 to 16 of the B register.

$\overline{\text{T11F}}$ / T11F

Bit 11 of the T register determines the field being transferred into. If this bit is on (T11F), then the operator is either CFXL or FFXL and the transfer will be into the F field of the word; set J = 9 to accommodate this transfer. If bit 11 in the T register is off (T11F'), then the operator is either FCXL or CCXL and the transfer will be into the CORE or C field of the word; set J = 8.

J = 8

$\overline{\text{T11F}}$

Entry to this J register setting is from J = 7 if T11F' is true. The from field is now in the S register and the word being transferred to is in the B register; T11F' indicates that the field is to go into the CORE or C field. Set the B register bits 15=>1 with the contents of the S register.

T11F

Entry to this J register setting is from J = 9 if T11F is true. The from field is in the F register and the field being transferred into is bits 30=>16 of the B register, the F field. Therefore, place the contents of the F register into B[30=>16]. Set the contents of the S register into the F register to restore the F register to its original configuration.

Unconditionally set the S register to its original configuration, X[30=>16] to S, mark the A register as empty, AROF to 0, and terminate the operator (EXIT).

J = 9

If the field being transferred into is the F field, interchange the F and S registers. This temporarily stores the contents of the F register in the S register and places the field being transferred into the F register to make use of the F to B[30=>16] transfer logic. Set the J register to 8 to make this transfer and terminate the operator.

3.30 DUPLICATE DUPL - 2025

Stack adjustment occurs to make one register empty and the other full. The contents of the full register are copied into the empty register and the empty register is marked full.

SUMMARY OF OPERATION

If both the A and B registers are loaded, the B register contents are stored in the stack. Then the contents of A are transferred to B. If both registers are empty, the B register is loaded and then the contents of the B register are transferred to the A register. If either register is loaded and the other empty, the contents of the loaded register are transferred to the empty register. Upon completion of the operator, the A and B registers will have the same contents.

DETAILED DESCRIPTION

J = 0

Upon entry into the operator, if both the A and B registers are loaded, a store B memory access is initiated by setting the E register to 11, the stack address is incremented by one and the J register is set to 2. If both the A and B registers are empty, a load B memory access is initiated by setting the E register to 3. The J register is set to one. If the A register is empty and the B register is loaded, the contents of B are transferred to the A register and the A register is marked as loaded. If the A register is loaded and the B register is empty, the contents of A are transferred to the B register and the B register is marked as loaded. The syllable execute complete level (SECL) is obtained and the operator terminates.

J = 1

Upon completion of the Load B memory access the B register is marked as loaded, the stack address is decremented by one and the J register is set to zero.

J = 2

Upon completion of the store B memory access, the B register is marked as empty and the J register is set to zero.

3.31 LOAD

If the flag bit and the presence bit of the word in the A register are both one, the word in the A register is replaced by the contents of the cell addressed by the 15 low order bits of the A register.

If the flag bit of the word in the A register is zero, the 10 low order bits of the word in the A register are used as a relative address. The contents of the A register are replaced by the contents of the memory cell addressed after appropriate indexing of the relative address.

If the flag bit of the word in the A register is one and the presence bit zero, the presence bit in the interrupt register is set and the operation is terminated.

If the VARF Flip-flop is set, the Processor is set to sub-program level after the relative address operation and VARF is reset.

SUMMARY OF OPERATION

The load operator is used to read a word out of memory and store it in the A register. The address for this word to be read out of memory is determined in the following manner:

If the A register contains a:
- Descriptor - the address contained in the 15 low order bits of the A register is utilized.
- Operand - the address is determined by adding the 10 low order bits of the A register to a specified base address.

If the word initially in the A register is a descriptor, the presence must be on or an interrupt is initiated.

DETAILED DESCRIPTION

J = 0

The actions occurring at J = 0 may be separated into three groups according to the double lines. The first group places the top word in the stack into the A register if it is not already there. The second group of actions provides for the handling of descriptors. The third group is for the handling of operands.

Placing Top Word of Stack in A

If both registers are empty, a load of the A register is initiated. If A is empty with B loaded, the B register contents are transferred to the A register.

Handling of Descriptors

If the word in the A register is a descriptor (AROF • A48F) and the presence bit is off (A46F'), a presence interrupt is set into the interrupt register by the setting of U15F, U16F and U17F. If the word addressed is present in core memory (A46F set), the 15 low order bits of the A register are transferred to the M register and a memory access is initiated to load the A register from the cell addressed by the M register.

Handling of Operands

If the word in the A register is an operand (AROF • A48F') the 10 low order bits of the A register are used to develop a relative address, the address of the word to be loaded into the A register. Figure 3.31-1 shows the bit coding utilized to develop a Relative address.

SALF	Bit 10	Bit 9	Bit 8	MS	Absolute Address	
					Base	Relative
0	-	-	-	-	R+	Bits 10 thru 1
1	0	-	-	-	R+	Bits 9 thru 1
1	1	0	-	0	F+	Bits 8 thru 1
1	1	0	-	1	(R+7)+	Bits 8 thru 1
1	1	1	0	-	C+	Bits 7 thru 1
1	1	1	1	0	F-	Bits 7 thru 1
1	1	1	1	1	(R+7)-	Bits 7 thru 1

FIGURE 3.31-1. RELATIVE ADDRESS CODING

R relative. If the Processor is not in sub-program level of operation (SALF') or it is in the sub-program level but the A10F bit is reset (A10F'), the base address is to be R relative. Subsequently, the 10 low order bits of the A register are added to the nine bits of the R register (presently in the high order of M) with the sum placed in the M register.

R to M -	15	14	13	12	11	10	9	8	7							- Base
A -						10	9	8	7	6	5	4	3	2	1	- Index
M -	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	- Relative Address

C Relative. If the Processor is in sub-program level (SALF) and the A10F bit is set (indicating an R relative address is not desired) and A09F is set with A08F reset (A09F • A08F' - indicating a C relative address is desired), the contents of the C register are transferred to the M register. The addition cycle at J = 4 places the C relative address in the M register.

F Plus Relative. If the Processor is in sub-program level (SALF) and the A10F bit is set (R relative not desired) and the Mark Stack Flip-flop is off (an F relative address is desired), with either A09F reset or A08F set (A09F' + A08F - indicating F plus relative), the contents of the F register are placed in the M register. The addition cycle at J = 4 places the F relative address in the M register.

F Minus Relative. If the same conditions prevail as with F+ relative with exception of both A09F and A08F being in the set state (indicating F minus is desired), the seven low order bits of the A register are complemented in addition to transferring the F register contents to the M register. Thus, at J = 4, the addition cycle will subtract A from the M register. Q01F is set to change the complemented A register into an 8's complement. Q03F is set to indicate a borrow in case a carry results during the addition cycle. Note that only the seven low order bits of the A register are complemented. A requirement of F minus relative is that A10F, A09F and A08F be in the set state, which is the required status for the complemented A register; therefore, no action is necessary.

R+7 Relative. If the same conditions prevail as with F+ relative with the exception that the Mark Stack Flip-flop is set, the required F register setting is presently stored in the mark stack control word at location R+7. Therefore, the word at R+7 must first be accessed to obtain the required F register setting. The digit 7 is placed in the M register with the E register being set to 6. This will load the M register with the F register setting presently contained in the mark stack control word at R+7. Once the M register is loaded with the F register setting, the operation proceeds as previously described for either plus or minus F relative, depending on the bit configuration in the A register.

Relative Address Magnitude. If a relative address (other than R relative) is indicated by the bit configuration in the A register (SALF • A10F), the extent to which the address may be relative is restricted. If either A09F or A08F is reset in conjunction with SALF and A10F being set, the A10F and A09F bit positions are reset. Thus, F+ and (R+7) + may be indexed a maximum of 255 locations and C+, F- and (R+7) - may be indexed a maximum of 127 locations.

Q09F to 1. Q09F is set to one so that the 42 bit mantissa adder will function as a 10 bit adder.

J to 4. The J register is set to 4 to gate the addition cycle.

SALF to 1, VARF to 0. If the variant flip-flop is on (VARF) then a previous VARL operator had reset the sub-level flip-flop (SALF). Reset the variant flip-flop (VARF) and set SALF to 1.

J = 1

This J register setting is only reached if initially both the A and B registers were unoccupied. Upon completion of the memory access to load the A register, AROF is

set, the S register is decremented, and the J register is returned to zero.

J = 4

At this J register setting the 10 low order bits of the A register are either added to or subtracted from the M register. Any actions at J = 4 are gated with EEZL, thus insuring that the M register contains the F setting from R+7 (if required), otherwise EEZL is true on entry to J = 4.

The 10 low order bits of the A register are added to or subtracted from the 10 low order bits of the M register. If the A register contents are in complement form (subtract), Q03F will be set, otherwise it is reset. If a subtract operation is being performed and no carry results (Q03F • $\overline{\text{W10L}}$), a one is subtracted from the five high-order bits of the M register to address the next lower segment of memory. If an addition is being performed and a carry results (Q03F' • W10L), a one is added to the five high-order bits of the M register to address the next higher segment of memory.

The E register is set to 4 to load the A register with the word addressed by the relative address presently being developed. The J register is set to 5 to await completion of the memory access and subsequent termination of the operator.

J = 5

Upon completion of the memory access to load the A register, the syllable execute complete level is true to terminate the operator.

3.32 INDEX
F & S SET/STORE
LINK LIST LOOKUP

INDL - 0141
FXSL - 2141
LLLL - 2541

INDEX

The 15 low order bits of the word in the B register are arithmetically added to the 15 low order bits of the word in the A register. Positions 48 through 16 of the A register are unchanged. Overflow is lost. The B register is set to empty.

F & S REG. SET/STORE

Based on the two low order bits in the A register, either the F or S registers will be set from or stored into the B register. The A register is set to empty.

LINK LIST LOOKUP

Bits 30 ⇒ 16 of the A register contain the size of the memory segment required. The B register will contain a storage link word with a segment size in bits 30 ⇒ 16 and an address of the next larger segment in bits 15 ⇒ 1. If the segment size, as specified by the B register, is smaller than the size required, the next storage link word, as specified by B [15 ⇒ 1], is examined. This process will continue until a segment size is found which is equal to or larger than the size required, at which time a data descriptor is placed into the A register pointing to the last storage link word looked at.

SUMMARY OF OPERATION

The A and B registers are loaded, if not initially loaded. The contents of the A register are added to the contents of the B register via the 42 bit mantissa adder and the sum is placed in the B register. Thereafter, the 15 low order bits of the B register are transferred to the M register and subsequently transferred to the A register.

F & S REGISTER SET/STORE

Stack adjustment is initiated, if necessary, to load the A and B registers. The top word in the stack is looked at to determine what the operator will do. Referencing Figure 3.32-1, A01F and A02F are used to determine which register is effected and whether the operation is a set or store. A01F determines which register (F or S) is affected. If A01F is set, then the S register is affected. If A01F is reset, then the F register is the register being worked upon. A02F determines if the operation is to set or store the register. If A02F is set, then the operator will set the register (F or S) from the B register. If A02F is reset, then the operator will store the contents of the F or S register into the B register. In either case the A register is set to empty when the operator is terminated. If the operator was to set the F or S register, then also mark the B register as empty.

A01F	A02F	Resulting Action
0	0	F to B [30 ⇒ 16]
1	0	S to B [15 ⇒ 1]
0	1	B [30 ⇒ 16] to F
1	1	B [15 ⇒ 1] to S

FIGURE 3.32-1 F & S REGISTER REFERENCE

LINK LIST LOOKUP

This operator was designed to scan the available storage link words in memory. Its primary use will be in the storage allocation routine of the MCP. The storage link list words have the following format:

Bits 48 ⇒ 46 — Bits 48 and 46 are set to one and bit 47 is set to zero to indicate a data descriptor, present.

Bits 45 ⇒ 31 — Not used, normally set to zero.

Bits 30 ⇒ 16 — Number of words in this available space in memory. If last link word, then equal to 77777.

Bits 15 ⇒ 1 — Address of next larger space in memory.

The top word in the stack (A register) will contain the size of memory area required in bits 30 ⇒ 16. The storage link words are read out of memory into the B register starting with the smallest available area. If this area is smaller than required, address the next larger area. If this area is equal to or larger than required, build a data descriptor with an address of the last link word found. Terminate the operator with the data descriptor in the top of the stack and the second word will be the storage link word found.

If there is no area of memory large enough available, the last link word accessed is in an MCP table and referred to as omega. This link word will have a maximum size word count (77777) to terminate the operator with a data descriptor in the top of the stack pointing to omega.

DETAILED DESCRIPTION

JOOL

If initially both registers are unoccupied, a memory access is initiated to load the A register with AROF being set to the one state. Upon completion of the memory access (MROF), a memory access is initiated to load the B register.

If just one of the registers is occupied on entry, a memory access is initiated to load the B register. The contents of the B register are transferred to the A register, if the A register is not already loaded. Stack adjustment occurs when required.

Once the A and B registers are loaded (AROF • BROF), the T register contents are looked at to determine what the operation will be.

AROF • BROF • EEZL • $\overline{\text{T11F}}$

If the A and B registers are loaded and the operator in the T register is Index (T11F'), add the A and B registers and place the sum in the B register. The A register remains unchanged. Set the J register to J = 2.

AROF • BROF • EEZL • T11F • $\overline{\text{T09F}}$

If the A and B registers are loaded (AROF • BROF • EEZL) and the operator is F & S Set/Store (T11F • T09F'), mark the A register as empty (AROF to 0) and terminate the operation (EXIT). At the same time examine the two low order bits of the A register to determine the exact operation.

A02F

If the operation is to set one of the registers, then mark the B register as empty (BROF to 0).

A02F • A01F

If the operation is to set (A02F) the S register (A01F), then transfer bits 15 ⇒ 1 of the B register to the S register.

A02F • $\overline{\text{A01F}}$

If the operation is to set the F register (A01F'), then transfer bits 30 ⇒ 16 of the B register to the F register. Set SALF to place the Processor in sub-level.

$\overline{\text{A02F}}$ • A01F

If the operation is to store (A02F') the S register (A01F), then transfer the contents of the S register into B [15 ⇒ 1].

$\overline{\text{A02F}}$ • $\overline{\text{A01F}}$

If the operation is to store (A02F') the F register (A01F'), then transfer the contents of the F register into B [30 ⇒ 16].

AROF • BROF • EEZL • T11F • T09F

With the A and B registers loaded (AROF • BROF • EEZL) and the Link List Lookup operator in the T register (T11F • T09F), complement A [39 ⇒ 1] to make use of the complement add in the mantissa adder. Set Q01F to one (ΔQ01F) to change the 7's complement to the 8's complement. Set the J register equal to 9.

J01L

Entry to this J setting is only for the Index operator. The 15 low order bits of the sum in the B register (Indexed portion) are transferred to the M register. Note that any carries are lost. The B register is marked as unoccupied and the J register is set to 2.

J02L

The indexed result in the M register is transferred to the 15 low order bit positions of the A register. The syllable execute complete level is true to terminate the operator.

J09L

Initial entry to this J register setting will find the A register [30 ⇒ 16] containing the size of an area needed, in complement form, the B register will contain the starting address of the storage link words. Remaining entries to this J count will find the B register containing the size of the memory segment being examined (bits 30 ⇒ 16) and the address of the next larger segment of memory (bits 15 ⇒ 1). Unconditionally transfer the address to the M register (M ⇐ B [15 ⇒ 1]) and set E to 5 to initiate a memory read operation. Change control to J equal 4.

J04L

Once the B register is loaded (EEZL) then the carry level from the 13th octal position of the mantissa adder is examined to see if a carry exists (W13L + W13L'). If no carry level exists from the adder (W13L'), the size of the segment in memory is smaller than required, set the J register to 9 to read the next storage link word. If a carry level does exist (W13L), then the area found is equal to or larger than needed, set the J register to 10 to terminate the operator.

J10L

Unconditionally build a data descriptor (A47 to 0, A48 to 1, A46 to 1), clear the unused bits, A [45 ⇒ 16] to zero, and set the address portion of the descriptor to the address of the last link word looked at, M to A [15 ⇒ 1]. The operator is terminated with the data descriptor in the top of the stack and the second word in the stack contains the last storage link word examined.

3.33 CONSTRUCT OPERAND CALL
CONSTRUCT DESCRIPTOR CALL

MDVL-0241
MDAL-1241

The contents of the A and B registers are exchanged. The flag bit of the word in the A register is set to one. If the operator is:

Construct Operand Call: The subsequent action of this operator is identical to that of an Operand Call Syllable after the Operand Call Syllable has caused a word to be read from memory.

Construct Descriptor Call: The subsequent action of this operator is identical to that of a Descriptor Call Syllable after the Descriptor Call Syllable has caused a word to be read from memory.

SUMMARY OF OPERATION

If the A and B registers are initially loaded, their respective contents are exchanged. Otherwise the registers are loaded in an exchanged fashion. The new contents of the A register are forced to be a descriptor and the operator is changed to either an Operand/Descriptor Call Syllable depending upon the original operator.

DETAILED DESCRIPTION

J = 0

If both the A and B registers are initially occupied, their respective contents are exchanged and the J register is set to 2.

If either the A and/or B register is unoccupied on entry, the A and B registers are loaded in an exchanged fashion; i.e., if A is empty and B is occupied, B is not pushed up. If A is occupied and B is empty, A is pushed down into B. If both registers are empty, the top word in the core stack is loaded into B and the second word of the stack is loaded into A.

J = 1

If both registers were initially empty, a memory access is initiated to load the A register with the second word of the core stack, once the memory access is completed (MROF) to load the B register with the top word of the core stack.

J = 2

Upon completion of the memory access to load the A register (if required; MROF), the S register is decremented.

If both registers were initially occupied or upon completion of the memory access to load the A register, EEZL is true to gate the following required actions:

1. Mark the B register as occupied and the A register as a descriptor.
2. Set either the Operand or Descriptor Call Syllable code to the T register, depending on the current operator.
3. Fetch is enabled. Automatic Count for Fetch Level is true (ACFL). The syllable execute complete level is true to terminate the operator.

3.34 BRANCH FORWARD NON-DESTRUCTIVE
BRANCH BACKWARD NON-DESTRUCTIVE
BRANCH FORWARD DESTRUCTIVE
BRANCH BACKWARD DESTRUCTIVE
DELETE

ZFNL - Y051/X451
ZBNL - X151/X551
ZFDL - X251/X651
ZBDL - X351/X751
DELL - 0051

The above listed branch operators will examine a field of bits within the second word in the stack starting at the bit position specified by the G and H registers. The length of the field is specified by the four high order bits of the T register (T12F ⇒ T09F) with a maximum field length of 15 bits.

If the specified field is equal to zero, the A register is marked as empty and the operator is terminated; no branching occurs. If the specified field is unequal to zero, execute a syllable branch, either forward or backward. In either case, if the operator is Destructive, unconditionally mark the B register as empty.

If the operator is the Delete operator, then the top word in the stack is deleted from the stack.

SUMMARY OF OPERATION

The Delete operator will delete the top word in the stack. Upon entry to this operator the stack can be in one of four configurations, depending upon the status of AROF and BROF. The deletion will occur as follows for the various stack configurations:

AROF	BROF	ACTION
0	0	Count S - 1
0	1	Set BROF to 0
1	0	Set AROF to 0
1	1	Set AROF to 0

The delete will take place if the repeat count field of T is less than four (TL4L), that is, if one of the branch on non-zero field operators has a field length of zero, then the operator will only delete the top word in the stack.

The Branch on Non-zero Field operators will operate as follows:

Branch Forward on Non-Zero Field Non-Destructive

This operator tests a field of the word in the B register for zero. If the field is zero, the A register is marked empty and the operator is terminated. If the field is not zero, the T register is changed to a syllable branch forward unconditional.

Branch Forward on Non-Zero Field Destructive

This operator tests a field of the word in the B register for zero. If the field is zero, the A and B registers are marked empty and the operator is terminated. If the field is not zero, the B register is marked empty and the T register is changed to a syllable branch forward unconditional.

Branch Backwards on Non-Zero Field Non-Destructive

This operator tests a field of the word in the B register for zero. If the field is zero, the A register is marked empty and the operator is terminated. If the field is not zero, the T register is changed to a syllable branch backwards unconditional.

Branch Backwards on Non-Zero Field Destructive

This operator tests a field of the word in the B register for zero. If the field is zero, the A and B registers are marked empty and the operator is terminated. If the field is not zero, the B register is marked empty and the T register is changed to a syllable branch backwards unconditional.

DETAILED DESCRIPTION

JOOL

TL4L

If the repeat count field of the operator in the T register is less than four, the operator is a Delete. The operator is terminated (EXIT) and the A register is set to empty (AROF to 0).

$\overline{\text{AROF}}$

If the A register is empty, then mark the B register as empty (BROF to 0) and transfer the contents of the B register to the A register. The transfer of B to A is redundant and will have no effect as AROF is also being reset.

$\overline{\text{AROF}} \cdot \overline{\text{BROF}}$

If both the A and B registers are empty, the top word in the stack is pointed to by the S register. Delete this word from the stack by decrementing S (S-1).

$\overline{\text{TL4L}}$

If the repeat count field of the operator in the T register is not less than four, then the operator is one of the branch operators. Unconditionally Inhibit Fetch for the possibility of a branch.

A normal stack adjustment occurs, if necessary, to load the A and B registers. The contents of the G, H, K, and V registers are set into the X register as a temporary storage.

AROF • BROF

When the A and B registers are loaded, the J register is set to 9, the A and B registers are exchanged, and the repeat count field of the T register is counted down (T-4). The count down by 4 is actually counting down the field length by one because only the four high order bits of the repeat counted are effected. The count down is in preparation of testing the first character of the field to be tested.

J01L

Upon completion of a memory cycle for stack adjustment count the S register down 1 and return control to J = 0. Mark the B register as valid.

J08L

T08F

Mark the B register as empty, this is a destructive operator.

HE5L

When the H register is equal to 5 the last bit of the character in the Y register is being tested; clear the Y register (Y to 0).

BENL

If the bit in Y, pointed to by H, is unequal to zero, change control to J10. A non-zero bit has been found in the field and there is no need to test the rest of the field; proceed with the branch.

BENL + TL4L

If the bit in Y, pointed to by H, (BENL) is unequal to zero or if the repeat count field of the T register is less than four (TL4L), then this is the end of the test for a non-zero field. Interchange the contents of the A and B registers to restore them to their original configurations.

TL4L

With the repeat count field of the T register less than four (TL4L) the last bit of the field is being checked; restore the Dials to their original settings from the X register.

$\overline{\text{BENL}}$

The bit pointed to by H in the Y register is zero ($\overline{\text{BENL}}$') and the field being compared has been exhausted (TL4L); set the J register to zero in preparation to exit this operator without any branching.

$\overline{\text{TL4L}}$

With $\overline{\text{TL4L}}$' true the repeat counted field has not been exhausted; continue by decrementing the T register repeat count field (T-4).

$\overline{\text{GE7L}}$ + $\overline{\text{HE5L}}$

If the G and H registers are not pointing to the least significant bit of the word, then increment H (H+1) to point to the next bit in sequence.

$\overline{\text{GE7L}}$ • HE5L

With G pointing to any character except the last character of the word ($\overline{\text{GE7L}}$') and H is set to the last bit in Y, H = 5, increment the G register (G+1) to index the next character transferred to the Y register.

$\overline{\text{BENL}}$ • HE5L

If the bit being tested is the least significant (HE5L) and it is equal to zero ($\overline{\text{BENL}}$'), set the J register to 9 to obtain the next character.

J09L

A new character is needed in the Y register; transfer from A to Y, the character pointed to by the G register. Set J to 8.

J10L

Entry to this J register setting is only if a non-zero field has been found and a branch is to take place. A syllable branch operator is set into the T register and the J register is set to zero. T07F' is true if the operation is a branch forward, set a 2 into the third octal position of the T register, T [9 ⇒ 7] to 2; if T07F is true the operation is to branch backward, set a one into the third octal position of the T register, T [9 ⇒ 7] to 1. Set the remainder of the T register to 4x31, and clear the J register. This changes control of the Processor to J = 0 of the syllable branch operator (4231 or 4131), forward or backward.

Unconditionally the G, H, K, and V registers are restored from the X register.

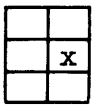
3.35 VARIABLE FIELD ISOLATE

The Variable Field Isolate operator will select a field of bits from the top word in the stack (1 to 39 bits long) and create a new word with this field, right justified. This new word will be an operand and will replace the top word in the stack.

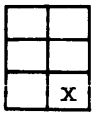
The G and H registers index the high order bit of the field. The operator variants (L located in T [12 ⇒ 10] and S located in T [9 ⇒ 7] 0 indicate the size and right hand offset of this field. The L variant indicates the number of characters that make up this field, including the first and last bits. The S variant indicates the offset, right justification, of the low order bit of the field, that is, how many low order bits in the last character of the field are to be discarded.

The right justification of the field will make use of both octal and binary shifting. If the value in the S variant is less than three, the bit position to be right justified is in the low order octade of the last character of the field. Only binary shifting is necessary, the S variant is counted down by one with each binary shift. Right justification is complete when the S variant is equal to zero.

Example: "x" indicates the bit position right justifying to.

1. 

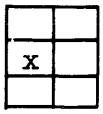
S = 1; S is set to one to indicate the number of binary shifts required to right justify bit "x".

2. 

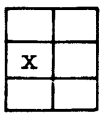
S = 0; One binary right shift places bit "x" into the low order bit position, S is counted down to 0, now indicates that field is right justified.

If the S variant is three or greater, the bit position to be right justified is in the high order octade of the last character of the field. The right justification will be accomplished by one octal shift followed by the required number of binary shifts. Since one octal shift is equal to three binary right shifts, the S variant must be decremented by three. However, existing machine logic makes it more expedient to first adjust the S variant by adding one and then subtract four when the octal shift takes place. This will leave a value in the S variant which is the number of binary shifts required to complete the right justification.

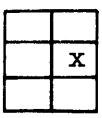
Example: "x" indicates the bit position right justifying to.

1. 

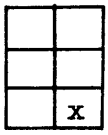
S = 4; S is equal to four to indicate the number of binary shifts required to right justify bit "x".

2. 

S = 5; S is adjusted in preparation for an octal shift by adding one to the S variant.

3. 

S = 1; One octal shift places bit "x" into the low order octade, S variant is decreased by four and now indicates the actual number of binary shifts remaining.

4. 

S = 0; One binary right shift places bit "x" into the low order bit position, S is counted down by one when the shift occurs, now equal to zero to indicate that right justifying is complete.

Figure 3.35-1 depicts the initial offset (S variant) with respect to the adjusted offset and the number and type of shifts required for each setting of S.

INITIAL OFFSET	ADJUSTED OFFSET	OCTAL SHIFTS	BINARY SHIFTS
0	0	0	0
1	1	0	1
2	2	0	2
3	4	1	0
4	5	1	1
5	6	1	2
6	DISALLOWED VALUES OF S		
7			

FIGURE 3.35-1 S VARIANT ACTIONS

The S variant can only contain the values of zero through five to indicate the number of bits that the field is to be right justified. The values of six or seven are disallowed. The reason being that there are only six bits to a character and if the offset is greater than five it will include an entire character, therefore, the L variant would be one less to transfer one less character.

SUMMARY OF OPERATION

If necessary, stack adjustment is initiated to place the top word of the stack in the A register.

To isolate the field of 13 bits in Figure 3.35-2 (A), starting with the bit labeled 1 and ending with the bit labeled 13, the following conditions must exist upon entry to the operator:

1. The G register is equal to 2 to index the third character of the word in the A register.

2. The H register is equal to 1 to index the second bit of this character, the bit labeled as 1.

3. The L variant of the operator is equal to 3 to indicate the number of characters in the field.

4. The S variant of the operator is equal to 4 to indicate the number of bits the field is offset from the right hand bit of the last character. The unwanted bits in Figure 3.35-2 are designated by "u".

To isolate the field in the A register, a transfer, a character at a time, removes the field from the A register and, via the Y register, places it into the X register. Only the low order bits of the first character, starting with the bit indexed by H, are transferred to the X register, the H register is then set to zero. With the H register equal to zero the remaining character positions will transfer as six bit characters.

Each time that a character is transferred from the A register to the X register, the G register is counted up one to index the next character in the A register and the L variant is decremented by one to count the number of characters transferred. At such a time that the L variant of the operator is equal to zero, the transfer of the field from the A register to the X register has been completed, Figure 3.35-2 (B). Also, the S variant had been examined and found to contain a value of three or greater, the required adjustment made by incrementing the S variant by one to a value of five.

The S variant with the adjusted value of five indicates that right justification requires one octal right shift and one binary right shift, reference Figure 3.35-1. The X register contents are shifted right octally once and the S variant is decreased by four to a value of one, Figure 3.35-2 (C).

The A register is now cleared and the contents of the X register are transferred to the A register in preparation for the necessary binary shifting, Figure 3.35-2 (D).

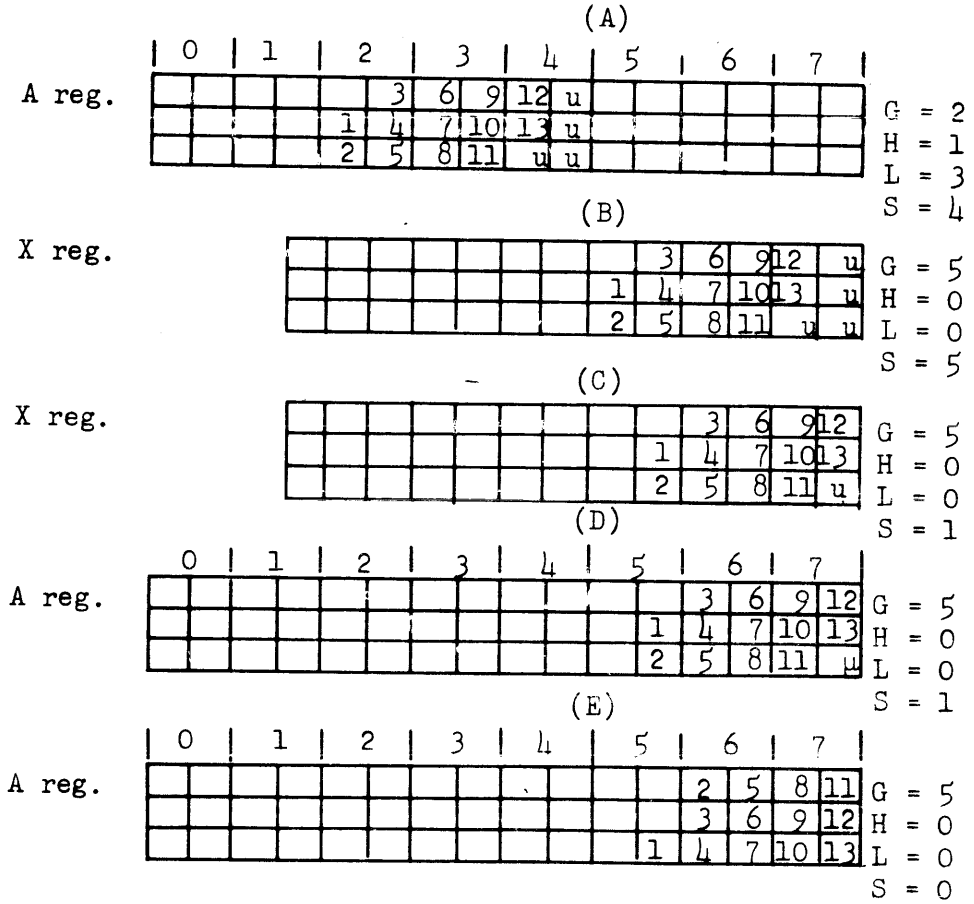


FIGURE 3.35-2 VFIL OPERATIONAL FLOW

The A register contents are binary shifted right, S is counted down by one with each binary shift, until the S variant is equal to zero. The S variant equal to zero indicates that the entire field has been right justified, Figure 3.35-2 (E), and the operator can be terminated. The isolated field has been right justified and has replaced the top word in the stack.

It is important to note that this is the only word mode operator that uses the Dial registers (G and H) and does not restore them to the values they contained upon entry to the operator. The G and H registers now index the character and bit following the last character transferred to the X register.

DETAILED DESCRIPTION

JOOL

The L variant is examined to determine if the operator will be terminated or executed. If the variant L equals zero (TL8L), then the operator is terminated (EXIT). MROF will be true if stack adjustment had occurred and required a memory read; decrement the S register (S - 1).

If the L variant is unequal to zero (TL8L'), then the actions listed below the double line will occur.

$\overline{\text{TL8L}}$

If the L variant is unequal to zero and AROF' is true, the actions listed between the double lines will adjust the stack to place the top word of the stack into the A register.

AROF • EEZL • $\overline{\text{TL8L}}$

With the A register occupied (AROF) and no memory access in progress (EEZL) and the L variant of the operator unequal to zero (TL8L'), change control by setting the J register to 5 and transfer the first character of the field to the Y register (A [G] to Y).

$\overline{\text{TL3L}}$

If the S variant in T [9 ⇒ 7] of the operator is not less than three (TL3L'), then the S variant must be adjusted. Increment the S variant by adding one to the repeat count field of the T register (T + 1).

J05L

The following actions occur unconditionally at J = 5.

Y [H ⇒ 5] to X [6 ⇒ 1]

Transfer the contents of the Y register into the six low order bits of the X register. If this is the first character of the field, the only bits transferred will be the low order bits of Y starting with the bit pointed to by H. The H register will be set to zero after the first character transfer to make all other characters transfer as full six bit characters.

Y to 0

Clear the Y register in preparation for the next character, transferred single ended from the A register.

G + 1

Increment the G register by one to index the next character in sequence.

T - 8

The L variant of the operator is contained in the three high order bits of the repeat count field of the T register. By subtracting eight from the repeat count field the L variant is decreased by one.

X [13 ⇐ 3] ⇐ X [12 ⇐ 2]

The X register is octally shifted left. The first octade shift of a significant character within the X register is performed at J06, and the second octade shift occurs with the above logics.

H to 0

Ensure full character (6 bits) shift from register Y to register X of all characters of the field with the exception of the first character.

J to 6

Exit from this J count to J = 6.

T10F • $\overline{\text{T11F}}$ • $\overline{\text{T12F}}$

If this is the last character of the field, set the contents of the A register to zero (A to 0). The last character of the field is denoted by T10F • T11F' • T12F', or the L variant equal to one.

J06L

TL8L

If the L variant is unequal to zero (TL8L'), octally left shift the X register once (X [13 ⇐ 2] ⇐X [12 ⇐ 1]) in preparation to place the next character of the field in the X register. Clear the X [1] octade (X [1] to 0) because its input from the Y register is single ended. The transfer of Y to the X [2] octade is double ended. Also exit this J count to J equal 5.

Unconditionally at J = 6 transfer the next character from the A register to the Y register indexed by the G register (A [G]) to Y).

TL8L

If the L variant of the operator is equal to zero (TL8L), the entire field is in the X register, the actions listed below the double line will occur.

T09F is reset (T09F to 0) to decrement the S variant by four. If T09F was off, then this action is redundant.

T09F

With T09F true, the value of the adjusted S variant is greater than three, an octal right shift of the X register is executed (X [12 ⇒ 1] ⇐X [13 ⇒ 2]) as the first step of right justifying the field.

T09F

If T09F' is true, the right justification will not require an octal right shift. Transfer the contents of the X register to the A register (X to A [39 ⇒ 1]) and exit this J register setting (J to 7).

J07L

TEZL

If the repeat count field of the T register is unequal to zero (TEZL'), right justify the field in the A register by doing a binary right shift (A [39 ⇒ 2] to A [38 ⇒ 1]), clear the high order bit of the A register mantissa (A39F to 0) and count the S variant down by one (T - 1).

TE1L

With the repeat count field of the T register equal to one the last binary right shift is being executed; terminate the operator (EXIT).

SECTION 4

CHARACTER MODE OPERATIONAL FLOWS

4.1 TRANSFER SOURCE CHARACTERS TSDL-XX77

This operator transfers characters from the source string, starting at the position specified by the M and G registers, to the destination string, starting at the position specified by the S and K registers. The number of characters transferred is specified by the repeat field.

Prior to the execution of the operator, the H and V registers are tested for zero. If either the H and/or V register is not equal to zero, it is set to zero and the G and/or K register is increased by one. Overflow into the M and/or S registers can occur.

Successive characters proceed from left to right within a word and to consecutively higher memory addressed by word. At the completion of the operation, the M, G, S and K registers specify the next source and destination character in sequence.

SUMMARY OF OPERATION

The repeat-count field of the operator in the T register specifies the number of characters that are to be transferred. Any number of characters from zero to a maximum of 63 may be transferred by this operator.

Upon entry to the operator, the bit pointers are interrogated to determine if they point at the first bit of a character (zero). If the bit pointers do not equal zero, they are unconditionally set to zero and their associated character pointers are advanced. If the advancement of the character pointers causes overflow of the character pointers (7 to 0), their associated word address registers are incremented.

Once the bit pointers have been adjusted (if required), the repeat count field is checked for a zero content. If the repeat count field is zero, the operator is terminated. If the repeat count field is not equal to zero, the A and B registers are loaded if either is initially empty. If the registers are initially occupied, and character pointer adjustment is required which overflows into the next word, the A and/or B registers are reloaded.

Once the A and B registers are loaded, the Bit pointers equal zero and if the repeat count field does not equal zero, the B register is checked for alignment. If the three bits of the K register are equal to the three high order bits of the N register (8-4-2), the B register is in alignment. If they are not equal, the B register will be shifted either right or left and the N register counted up or down until K is equal to N. When the B register is in alignment, the character pointed to by the K register is in the output alignment station (the 15 and 16 octade positions of the B register).

Transfer of characters can begin, once the B register is in alignment. The transfer is accomplished by first transferring the character in the A register, pointed to by the G register, to the Y register. Simultaneously, the B

register is shifted one octade position to the left when the N register is being incremented to tally the left shift. Thereafter, on the next clock pulse, the B register is again shifted left one octade position. In conjunction with this second left shift on the B register, the character in the Y register is transferred to the B register input alignment station (the 1st and 2nd octade positions). In other words, the character position pointed to by the K register is presently in the input alignment station and it receives the character from the A register (pointed to by the G register).

As each character is transferred, the source and destination character addresses are increased by one. If either the source and/or destination addresses overflow, their respective word addresses are increased by one. In the case of the destination address, the word in the B register must be stored in memory prior to incrementing the word address. As the character addresses are incremented, the repeat count field is decremented by one. Transfer of characters continues until the repeat count field goes to zero.

DETAILED DESCRIPTION

J = 0

Since there are numerous occurrences at J = 0, reference to the specific actions is made, relative to the double line divisions (according to action sequence).

Unconditionally, the V and H bit pointers are cleared to zero. This only has significance if they are not initially equal to zero. If, on entry to the operator, the six high order bits of the T register (repeat count field) are equal to zero (TEZL), and once the bit pointers are equal to zero (VEZL), the J register is set to 5 for subsequent termination of the operator. Note that HEZL is implied to be true when VEZL is true, and is not actually required as a term of the equation.

Inhibit Fetch

If either bit pointer is not equal to zero (HEZL' + VEZL'), fetch is inhibited to give precedence to the loading of the registers.

HEZL'

If the bit pointer for the A register is not equal to zero (HEZL'), the character pointer is incremented (G + 1) to address the next character in sequence. If at that time the character pointer is pointing at the last character of a word (GE7L), the source address register is incremented to address the next word in sequence. The A register is marked as unoccupied in order to facilitate loading of the A register. In this case, the first character of the next word is the character initially desired.

VEZL'

If the bit pointer for the B register is not equal to zero (VEZL'), the character pointer is incremented (K + 1) to point at the next character in sequence. If, at this clock pulse, the B register is loaded and is in alignment (KENL • BROF), the increment of the K register will cause misalignment of the B register. Therefore, to maintain alignment, the B register is shifted left and circulated two octade positions with a corresponding increment of the N register. The first shift occurs at this clock pulse time, and the second shift occurs at J = 6.

If the character pointer for the destination string is pointing at the last character of a word (KE7L) at the time it is incremented, the B register is marked unoccupied to facilitate reloading. If the B register requires reloading when KENL is true, its present contents must first be stored in memory. If KENL is true when KE7L is true, the B register is not in its original configuration, as obtained from memory (it has been shifted), thus the alignment procedure described above serves to place the B register in its proper form for storage.

If the character pointer for the destination string is pointing at the last character of a word, and the N register is equal to zero (KE7L • NEZL) at the time the character pointer is incremented, the destination string address register is incremented by one (S + 1). In this case, the B register is in its original configuration and does not require storage in memory; i.e., it already is in memory in proper form. Thus the only requirement is to reload the B register with the next word in sequence.

TEZL' • VEZL • HEZL

Once the bit pointers are equal to zero and the repeat count field indicates characters to be transferred, the A and B registers are loaded if required. If, on entry to the operator, the bit pointers were not equal to zero and the subsequent increment of the character pointers causes a word overflow, the A and/or B registers would be loaded (as required) regardless of their entry status to the operator. If both registers are marked unoccupied, the B register is loaded first by the setting of E to 3. This allows the B register to be aligned while the A register is being loaded. Once the B register is occupied, a load of the A register is initiated by the setting of E to 4.

MROF

If MROF is true at J = 0, this can only be the completion of a memory access to load the A register. Therefore, the A register is marked as occupied.

BROF • TEZL' • VEZL

Alignment of the B register occurs when the B register is occupied, there are characters to be transferred and the bit pointers are equal to zero. The specific actions during the alignment process are as follows:

K04F (KENL' + N01F). The B register is not in alignment when KENL' is true; i.e., the character position pointed to by the K register is not in the output alignment station (15th and 16th octade positions of the B register). In this case, alignment of the B register is required. The 4's bit of the K register is interrogated to determine the most expedient method for the alignment procedure. If K04F is set, it is most expedient to shift and circulate the B register to the right while decrementing the N register to tally the octade shifts. Since the N register tallies octade shifts and the K register points to a complete character position, one additional shift and circulate of the B register is required when N01F is true, even though the K equals N level (KENL) is true.

K04F' • KENL'. If the B register is not in alignment (KENL') and the 4's bit of the K register is reset (K04F'), it is most expedient to execute the alignment by shifting and circulating the B register to the left, while incrementing the N register to tally the left shifts.

KENL • N01F' • AROF • HEZL. Once the B register is in complete alignment and the A register is occupied, as indicated by KENL • N01F' • AROF, the first character from the A register to be transferred (as pointed to by the G register) is transferred to the Y register. Simultaneously, the B register is shifted to the left one octade position; i.e., one half character shift. This is the first of the required two octade shifts to place the character position in the B register, as pointed to by the K register, into the input alignment station (1st and 2nd octade positions of the B register).

The N register is incremented to keep track of the shifts and the J register is set to 2, to continue the transfer operation. The shift of a character from the B register output alignment station to the Z register is an unnecessary operation during the Transfer Source Characters Operator.

J = 1

Entry to J = 1 occurs only if the B register required loading at J = 0. When the memory access is complete (MROF) to load the B register with a word from the destination string, the B register is marked occupied and the J register is returned to zero. If, at this time, the A register is marked unoccupied, a memory access is initiated to load the A register with a word from the source string.

J = 2

The character in the Y register is transferred to the B1 and B2 octade positions of the B register. The B register is shifted left one octade position thus completing the second required half character shift to place the character position, as pointed to by the K register, in the input alignment station. The Y and Z registers are cleared to zero; clearing the Z register is an unnecessary function during this operator. The character pointers (G and K), plus the N register, are incremented, while the repeat count field of the T register is decremented.

When the character pointers are incremented, they are tested for overflow. If the K register overflows (KE7L + 1), the B register must be stored in memory regardless of the state of the A register character pointer. In this case,

the E register is set to 11 and the J register is set to 3. If the G register overflows (GE7L + 1) and the K register does not (KE7L'), the source address register is incremented (M + 1) and a load of the A register is initiated. If the G register overflowed, or neither register overflowed, the J register is set to 5. Note that if the B register is stored in memory, it is in the proper configuration for storage at this time and no adjustment of the B register is required prior to storage in the destination string.

J = 3

Upon completion of the B register store memory access (MWRF), the destination string address register is incremented by one (S + 1) to address the next word in sequence. At the same time, the G register is tested to determine if it also overflowed at J = 2. If the G register did overflow, it would not be equal to zero (GE7L + 1 GEZL). Therefore, the source address register is incremented (M + 1), a memory access is initiated to load the A register (E to 4) and the J register is set to 4. If the G register did not overflow, the B register is reloaded, if the repeat count field in the T register is equal to less than eight (TL8L). Since the information in the destination string is normally overwritten completely with new information, the B register only requires reloading if the T register is less than eight; this indicates that less than a complete word is to receive new information. If the G register did not overflow, the J register is set to 5.

J = 4

When the A register is reloaded from the source string (MROF), the J register is set to 5 to continue the operation. If the T register is less than eight (TL8L) the B register requires reloading. Its reloading was delayed at J = 3 to give precedence to the load of the A register, the B register will only require reloading during the last word of the transfer operation. The E register is set to 3 to initiate a load of the B register if required.

J = 5

Upon completion of a memory access to load either the A or B register (if a load was required), the repeat count field of the T register is interrogated. If TEZL is true, the transfer of all characters is complete and the syllable execute complete level is true to terminate the operator. If TEZL' is true,, additional characters require transfer. The next character in the A register, specified by the G register, is transferred to the Y register. The B register is shifted left one-half character position with the N register being incremented. The J register is set to 2 for the other half character shift and the transfer of the character in the Y register to the B register.

J = 6

This J register setting is only attained if, on entry to the operator, the V register was not equal to zero and the B register was initially occupied and in alignment with the K register (KENL). If at J = 0, the K register did not equal seven, the B register is brought back into alignment (K was decremented) by two half character shifts, the first at J = 0 and the second at this J register setting. In this case, BROF remains set and gates a return to J = 0

If, however, the K register did equal seven when KENL was true, BROF would have been reset, indicating that the B register contents are not a part of the destination field and therefore require storage. In this case, the two half character shifts are required to place the B register contents in the proper configuration prior to storage. Thus, on entry to J = 6, the level EWZL gates the left shift of the B register and also initiates a memory access to store the B register contents. Upon completion of the store memory access, the destination string address register is incremented (S + 1) and the J register is returned to zero. (EWZL = EEZL • MWOF')

4.2 TRANSFER SOURCE ZONE TZDL-XX76

This operator transfers the zone portion of the characters in the source string starting at the position specified by the M and G registers, to the zone portion of the characters in the destination string, starting at the position specified by the S and K registers. The numeric portions of the destination string characters are retained. The number of zones transferred is specified by the repeat field.

Prior to the execution of the operator the H and V registers are tested for zero. If either the H and/or V register is not equal to zero, it is set to zero and the G and/or K register is increased by one. Overflow into the M and/or S registers can occur.

Successive characters proceed from left to right, within a word, and to consecutively higher memory, addressed by word. At the completion of the operation, the M, G, S and K registers specify the next source and destination character in sequence.

SUMMARY OF OPERATION

The operation of the Transfer Source Zone Operator is almost identical to the operation of the Transfer Source Character Operator. Therefore, refer to the subject describing the Transfer Source Character Operator for a detailed description of specific actions. Only the minor differences are described in this write-up.

Once the operation proceeds to the point of starting the transfer operation, [B register in alignment and A register loaded (J = 0)], the character in the A register, pointed to by the G register, is transferred to the Y register. The character in the B register output alignment station is transferred to the Z register, and the B register receives the first of two character shifts. Thereafter, when the B register receives its second character shift (J = 2) to place the character position pointed to by the K register in the input alignment station, the zone portion of the character in the Y register is transferred to the B register. At the same time the numeric portion of the character in the Z register is transferred to the B register. The Y and Z registers are cleared, the character pointers are incremented, and the repeat count field is decremented to tally the zone transfer. The operation proceeds until the T register is equal to zero.

The basic difference can be seen in the storage function of the Z register to hold the numeric portion of the character shifted out of the B register for later insertion into the B register input station. Also, the fact that only the zone portion of the character in the Y register is transferred to the B register input alignment station.

DETAILED DESCRIPTION

For the detailed description of the Transfer Source Zone Operator, see subject 4.1 entitled "Transfer Source Characters" and note the exceptions described in the paragraphs under the "Summary of Operation".

4.3 TRANSFER SOURCE NUMERICS TNDL-XX75

This operation transfers the numeric portion of the character in the source string, starting at the position specified by the M and G registers, to the numeric portion of the character in the destination string, starting at the position specified by the S and K registers.

The zone bits of the characters in the destination string are set to zero. If the zone portion of the last source character transferred is minus, (BA = 10), the true/false indicator is set to true, otherwise, the true/false indicator is set to false. The number of numeric characters transferred is specified by the repeat field.

Prior to the execution of the operator, the H and V registers are tested for zero. If either the H and/or V register is not equal to zero, it is set to zero and the G and/or K register is increased by one. Overflow into the M and/or K registers can occur.

Successive characters proceed from left to right, within a word, and to consecutively higher memory addressed by word. At the completion of the operation, the M, G, S and K registers specify the next source and destination character in sequence.

SUMMARY OF OPERATION

The operation of the Transfer Source Numeric Operator is almost identical to the operation of the Transfer Source Character Operator. Therefore, refer to the Transfer Source Character Operator write-up for a detailed description of the specific actions. Only the minor differences are described in this write-up.

When the operation proceeds to the point of transferring the character in the Y register to the input alignment station (at J = 2), only the numeric bits of the character are transferred to the B register, and the zone bits of the B register input station (B05F and B06F) are cleared to zero. When the last character transfer is taking place, as indicated by the repeat count field equal to one, the zone bits are interrogated. If the zone bits indicate a negative sign (YRBF YRAF'; i.e., BA = 10), the true/false flip-flop (TRFF) is set to one state. Otherwise, the TRFF is left in the reset state. The true/false flip-flop is equivalent to Q12F in the logic book.

DETAILED DESCRIPTION

For the detailed description of the Transfer Source Numerics Operator, see subject 4.1 entitled "Transfer Source Characters" and note the exceptions described in the paragraphs under the "Summary of Operation".

4.4 TRANSFER PROGRAM CHARACTERS
TRANSFER BLANKS FOR NON-NUMERICS

TPDL - XX74
TBZL - XX12

The Transfer Program Characters operator transfers characters from the program string, starting at the position specified by the C and L registers, to the destination string, starting at the position specified by the S and K registers.

The number of characters transferred is specified by the repeat field. When the repeat field is odd, the first character in the program string is skipped.

The next program syllable fetched is the syllable following the one containing the last character transferred.

Prior to the execution of the operator, the V register is tested for zero. If the V register is not equal to zero, it is set to zero and the K register increased by one. Overflow into the S register can occur.

The Transfer Blanks for Non-Numerics operator tests a string of characters, starting with the character pointed to by the S and K registers. If a character tested is non-numeric, it is replaced in the string by a blank. The operation continues until the repeat field of T is equal to zero or a numeric character is encountered. If a numeric character is encountered, it is returned to the string and the operator is terminated.

SUMMARY OF OPERATION

Transfer Program Characters

The operation of the Transfer Program Characters Operator is similar to the operation of the Transfer Source Characters Operator. The primary difference lies in the fact that the Source String is not utilized as a source of characters to be transferred to the destination string. The source of characters obtained for subsequent transfer to the destination string is from the Program String. Therefore, the C register is utilized to address the program string for insertion of words into the P register. In this sense, the word in the P register does not contain program syllables, it contains characters to be transferred to the destination string.

In order to specify an individual character in the P register, the existing circuitry is modified; i.e., the L register normally points to a syllable (which occupies two character positions). Therefore, logic is provided (the 7th bit of the T register) to differentiate between the two character positions. For example, when the L register is specifying one of the two character syllable positions of the P register, the T07F bit being reset indicates the most significant character of the syllable, and the T07F bit being set indicates the least significant character of the syllable. Another way of considering the character selection of the P register is to assume that L02F, L01F and T07F have the significance of 4, 2 and 1, respectively. In this sense, the bits act as a binary counter: if L02F, L01F and T07F are all set, the character position of the P register specified is the 7th (the least significant). Thus the counter is cycled zero through seven to specify each of the eight characters of the word.

Transfer Blanks for Non-Numerics

This operator tests characters, specified by the S and K registers, for less than or equal to zero. If the character is less than or equal to zero, it is replaced by a blank character, and the next character is tested. If the character is greater than zero, the operation is terminated. The number of characters tested is specified by the repeat field. If the operator is terminated because a character tests greater than zero, the character is a numeric character and the S and K registers are left pointing at this character. If the operator is terminated because the repeat field has been counted to zero, S and K will point at the character following the field tested.

If the field contains all blanks or if the field length is zero, the true-false flip-flop is set to one. If the field contains a numeric character, the true-false flip-flop is set to zero.

DETAILED DESCRIPTION

Because of the many similarities between this operator flow and the Transfer Source Characters, all items will not be discussed. For a discussion of the items not discussed here refer to Subject 4.1, Detailed Description.

J = 0

The prime difference at this J register setting with the Transfer Source Characters is the elimination of the logic for loading the A register.

Also, if the operator in the T register is the Transfer Program Characters, note that when the B register is aligned and ready to commence character transfers the P register must be occupied as indicated by PROF in the set state. The first character to be transferred to the Y register from the P register is specified by the combination of L + T7, (see Summary Of Operation). The operator specification states that if the repeat field is initially odd, the first character in the program string is skipped. This is in reference to two character positions of a syllable, and requires no additional action on the part of the operator, i.e., on entry the L + 7 combination indicates the desired character position that is the first to be transferred.

If the operator is the Transfer Blanks for Non-Numerics, then the true-false flip-flop is set to one. Also, when the B register is aligned and ready for transfer, there is no character in the Y register, that is, the Y register is equal to zero.

J = 1

Entry to this J register setting occurs only if the B register required loading at J = 0. The B register is marked as occupied and the J register is set to zero.

J = 2

Unconditionally, the contents of the B register are shifted one octade left and the Y and Z registers are cleared. The K register is incremented one to indicate the character position of the B register in the output alignment station. The N register is counted up to count the octal shifts and the repeat field of the T register is counted down (T - 1).

$\overline{KE7L}$ • TBZL

If the character pointer of the B register is not indexing the last character of B, then the J register is set to 5 to obtain the next character from the B register.

KE7L

With the K register equal to 7, the last character of the B register is indexed; set the J register to 3 and the E register to 11 to initiate a store of the word in the B register.

TPDL

If the operator in the P register is Transfer Program Characters (TPDL), the contents of the Y register are transferred to the two low order octades of the B register.

At this time, if the T07F bit is set, the L register is also counted up by one, otherwise, the L register is not disturbed. For example, if the third character of the P register had just been transferred to the B register, L02F would be reset and both L01F and T07F would be set. Then when T is decremented and L incremented, the L02F bit is set and both L01F and T07F would be reset; thereby pointing at the 4th character position of the P register.

When L is equal to three and the 7th bit of T is set (LE3L • T07F), the P register must be reloaded. Therefore, the C register is incremented and E is set to 16 to initiate a fetch for the load of the P register. Note that if the character pointer for the B register also overflows (KE7L), the store of the B register contents takes precedence over the load of the P register. The E register will not be set to 16 until J = 3 in this case.

TBZL

With the Transfer Blanks for Non-Numerics operator in the T register (TBZL), a comparison of the character in the Z register must be made. At this time the Y register contents are equal to zero, comparisons will consider the character zero as non-numeric.

YGZL • $\overline{\text{YNZL}}$

If the character in Y (equal to character code for zero) is greater than the character code in the Z register (YGZL) or if the character code in Z is equal to Y (YNZL'), then the character in Z is non-numeric. Insert the code for a blank into the two low order bits of the B register.

$\overline{\text{YGZL}}$ • YNZL

If the character in Y is not greater than the character in Z (YGZL') and Y is not equal to Z (YNZL), then the character in the Z register is a numeric. Transfer this character back into the B register and set Q03F to one. Q03F indicates that a numeric character has been found.

J = 3

This J register setting is entered only if K was equal to seven at J equal 2. At the time of entry to this J register setting, a memory access is in process to store the contents of the B register.

MWOF • LEZL • $\overline{\text{T07F}}$ • TPDL

With the Transfer Program Characters operator (TPDL) and the store of the B register complete, set the E register to 16 and count the C register up one if L is equal to zero and T07F' is true.

MWOF • (TL8L + TBZL)

If the repeat count field of the T register is less than eight or the operator is Transfer Blanks, set the E register to 3 to reload the B register.

MWOF • $\overline{\text{Q03F}}$

When the memory write access is complete (MWOF) and if no numeric character is found or the operator is a Transfer Program characters (indicated by Q03F'), then increase the address of the destination string (S + 1).

J to 5

Change the J register setting when the memory access is complete (MWOF).

J = 5

If the repeat field of the T register is not zero (TEZL') and no memory access is in progress (EEZL) and the operator is the Transfer Program characters (TPDL) with no fetch in process (E16F') or the Transfer Blanks (TBZL) with no numeric character found (Q03F'), then the B register is shifted left (B [16 ⇒ 2] ⇐ B [15 ⇒ 1]) and

the character in the output alignment station is set into the Z register (B [16 ⇒ 15] to Z). The V register is set to 2. Also the N register is incremented (N + 1) to tally the octal shifts of the B register.

$\overline{TEZL} \bullet EEZL \bullet \overline{E16F} \bullet TPDL$

With the same enabling logic for shifting the B register, if the operator is the Transfer Program characters, then transfer the next character to the Y register (P [L + T7] to Y).

TEZL • KEZL • TPDL

If the repeat field of T is zero (TEZL) and the first character of the word in B is the character indexed (KEZL), then the contents of the word in B are duplicated in memory; mark the B register as empty (BROF to 0).

EXIT

There are three conditions which can cause the termination of this operator.

The first condition (TEZL • EEZL • E16F' • TPDL) is if the operator is Transfer Program characters (TPDL) with no memory access in progress (EEZL • E16F') and the entire string of program characters has been transferred, indicated by the repeat count field of the T register equal to zero (TEZL).

The remaining two conditions are for the Transfer Blanks operator (TBZL). If the repeat field of T is equal to zero (TEZL) and no memory access in progress (EEZL), then the entire string of characters was non-numeric; terminate the operation. If a numeric character was found in the string and the destination index adjusted to index the numeric character (Q08F), then terminate the operator.

Q08F • EEZL • TBZL

If Q08F is set, then a numeric character was found and is in the process of being reindexed; decrement the K register (K-1) to index the numeric character found.

Q03F • TBZL

If a numeric character was found (Q03F), then reset the true-false (TFFF to 0) indicator. The true-false indicator will remain set if the entire field was non-numeric.

Q03F • EEZL • TBZL

If a numeric character was found (Q03F), the K register indexes the character following the numeric. To index the numeric, the B register is shifted right two times, N is decremented with each shift and Q08F is set to one. Q08F indicates that the first shift has occurred and the K register can be decremented and the operation terminated.

J = 6

This J register setting is activated only if, on entrance to the operator, VEZL' was true, B register was occupied and in alignment with the K register (KENL).

4.5 TRANSFER WORDS

TWDL-XX05

This operator transfers words from the source string, starting at the position specified by the M register, to the destination string, starting at the position specified by the S register. The number of words transferred is specified by the repeat field.

Prior to the execution of the operator, the G, H, K and V registers are tested for zero. If the G or H registers are not equal to zero, they are set to zero and the M register increased by one. If either the K or V registers are not equal to zero they are set to zero and the S register increased by one.

At the completion of the operation, the M and S registers specify the next word in sequence.

SUMMARY OF OPERATION

Upon entry to the operator, the B register is checked to determine if it is occupied. If the B register is occupied, its contents are stored in memory prior to proceeding with the operation. Before storage can be effected, the B register contents must be in alignment. Therefore, the B register contents are restored to the proper configuration (if required) and then stored in the destination string. The B register is marked unoccupied.

The bit and character pointers for the source string are unconditionally cleared to zero. If they are not initially equal to zero, the source word address register is increased by one and the A register marked unoccupied.

The bit and character pointers for the destination string are unconditionally cleared to zero. If they are not initially equal to zero, the destination word address register is increased by one.

If the repeat count field is not equal to zero (TEZL'), the process of transferring words commences by loading the A register from the source string if it is unoccupied. If the A register is occupied at this time, the operation proceeds directly into the storage phase.

The contents of the A register are stored at the address specified by the destination string word address register (S). Subsequently, the A register is reloaded from the source string, and then stored into the destination string. The repeat count field tallies the number of words stored. This process continues until terminated by the repeat count field being equal to zero.

DETAILED DESCRIPTION

J = 0

Unconditionally the bit and character pointers for the source string are cleared to zero (H and G to 0). If either the H or G register is not initially equal to zero, the source word address register (M) is incremented by one with the A register being marked unoccupied.

NEZL' • NO8F

If the B register is not restored (not in its original position), as indicated by NEZL', it must be restored. The 8th bit of the N register is interrogated to determine the most expedient method of restoring the B register. If NO8F is set, it is faster to shift left and circulate the B register. The N register is counted up to tally the shifts. The use of NO8F limits the restoration of the B register to four shifts.

NEZL' • NO8F'

If the B register is not restored and the 8th bit of the N register is reset (NO8F'), it is faster to shift right and circulate the B register. The N register is counted down to tally the right shifts.

NEZL • BROF

Once the B register is restored to its original configuration, a store memory access is initiated (E to 11). The J register is unconditionally set to 1 whenever the N register equals zero.

J = 1

If the A register is unoccupied, the repeat count field does not equal zero, and the memory access to store the B register contents is completed (if one were initiated). The E register is set to 4 to initiate a memory access for loading the A register with a word from the source string. The J register is set to 2 to continue the transfer operation.

All actions at J = 1 (except the clearing of BROF) must await the completion of the memory access to store the B register contents, if one were initiated (EEZL).

If the A register is unoccupied and the repeat count field does not equal zero, the E register is set to 4, for initiation of a memory access to load the A register with a word from the source string. The J register is set to 2 to continue the transfer operation.

The character and bit pointers for the destination string are unconditionally (except for EEZL) cleared to zero (K and V to zero). If either the K or V registers are not initially equal to zero, the destination string word address register is increased by one (S+1). The B register is unconditionally marked as unoccupied. The operator terminates with both the A and B registers marked in an empty status.

J = 2

The A register is unconditionally marked as unoccupied.

As long as the repeat count field does not equal zero, the transfer of words continues at this J register setting. If this J register setting is arrived at with the A register occupied, the first action is gated by the term EWZL. Otherwise, the first action at this J register setting is gated by MROF, at the completion of the memory access to load the A register. In either event,

the first action initiates a store of the A register contents in the destination string (E to 10), increments the source string word address register (M+1) and counts the repeat count field down to tally the transfer operation.

Upon completion of the storage operation, the terms EEZL • MWOF initiate a reload of the A register from the source string, while simultaneously counting the destination word address register up by one. Thereafter, upon completion of the memory access to reload the A register (MROF), a memory access is again initiated to store the A register contents in the destination string. This process continues until the repeat count field is counted down to zero (TEZL).

When the level TEZL is true and the memory access is complete, to store the last word to be stored in the destination string, the syllable execute complete level is true to terminate the operator.

4.6 TEST OPERATORS

TEST FOR EQUAL	TEQL-XX24
TEST FOR GREATER THAN OR EQUAL TO	TGEL-XX26
TEST FOR LESS THAN OR EQUAL TO	TLEL-XX34
TEST FOR GREATER	TGTL-XX27
TEST FOR LESS	TLTL-XX35
TEST FOR NOT EQUAL	TNEL-XX25

Test operators provide the ability of testing a character in the source area against a predetermined character. Tests are made on the basis of the collating sequence of characters. The test operators test a single character in the source string against the repeat count field of the operator. If the test is met, the true/false flip-flop is set to true (1). If the test is unsuccessful, the true/false flip-flop is set to false (0). The character in the source string is specified by the M and G registers. The M and G registers are not advanced.

SUMMARY OF OPERATION

Upon entry to the operator, the bit pointer address for the A register is tested for zero. If it is not equal to zero, it is set to zero and the character pointer address is counted plus one. Should the character pointer address overflow, the word address register is incremented by one, and the A register is marked as unoccupied.

With the bit pointer equal to zero, the A register occupancy flip-flop is tested to determine whether or not it is loaded. If the A register is not occupied, a memory read access is initiated to load the A register. When the A register has been loaded (if required), the character specified by the G register is transferred to the Y register, and the repeat count field in the T register is transferred to the Z register. The true/false flip-flop is cleared in preparation for the test operation.

The Y and Z registers feed the comparator circuitry. The two outputs of the comparator determine whether the specified character from the A register is greater or less than the character in the Repeat Count Field, and whether the specified character from the A register is equal or not equal to the character in the Repeat Count Field. These two outputs, along with a decoding of the current operator levels, determine whether or not the true/false flip-flop is set to one. At the same time the operator is terminated. Upon termination of the operator, the G register continues to point at the character addressed during the test.

DETAILED DESCRIPTION

J = 0

Unconditionally the true/false flip-flop (TFFF) is cleared to zero. Logic is provided to set TFFF only if the conditions are met. TFFF is Q12F in the logic book.

If the bit pointer for the source string is not equal to zero initially, it is set to zero and the character pointer for the source string is counted plus one. Should this cause the character pointer to overflow, the word address register for the source string is increased by one (M+1), and the A register is marked as empty.

If the A register is unoccupied initially or is marked as unoccupied, as described above, a load of the A register memory access is initiated by setting the E register to 4. The J register is set to 1 to complete the load operation.

When the A register is loaded and the bit pointer for the source string is equal to zero, the character from the source string specified by the G register is transferred to the Y register. The character in the repeat count field of the operator in the T register is transferred to the Z register. The J register is set to 2 for performance of the comparison.

J = 1

Entry to this J register setting implies that the A register required loading. When the memory read access is obtained (MROF), the A register occupancy flip-flop is set and the J register returned to zero.

J = 2

The true/false flip-flop is set to the one state if the test is met (TFFF to 1). The T register bit combination, with the comparator output, establishes which operator and test conditions allow TFFF to be set. The comparator output Y GREATER THAN Z (YGZL) equals source string character being greater than reference character in repeat count field. The comparator output Y NOT EQUAL TO Z (YNZL) is equivalent to the source string character specified by the G register NOT EQUAL TO the reference character in the repeat count field of the T register.

T04F' • T01F • YGZL

The T register bit combination of T04F' • T01F applies to either of two operators: Test for Greater or Test for Not Equal (TGTL + TNEL). If YGZL is true, the test has been met and TFFF is set.

T04F' • T02F • YGZL

The T register bit combination of T04F' • T02F applies only to either of two operators: Test for Greater, or Test for Greater Than or Equal To (TGTL + TGEL). If YGZL is true the test has been met and TFFF is set.

T01F' • YNZL'

The T register bit T01F being reset indicates any one of three operators: Test for Greater Than or Equal To (TGEL), Test for Equal (TEQL), or Test for Less Than Or Equal To (TLEL). If YNZL' is true, the characters are equal, the test has been met and TFFF is set.

T01F' • T04F • T02F' • YGZL'

The T register bit combination of T04F • T02F' • T01F' can only apply to one operator: Test for Less Than Or Equal To (TLEL). If YGZL' is true, the character in Y is not greater than the character in Z and the test is met; TFFF is set.

T01F • T02F' • YGZL' • YNZL

The T register bit combination of T01F • T02F' can apply only to either of two operators: Test for Less, or Test for Not Equal (TLTL + TNEL). If YGZL' and YNZL are true, the character in Y must be less than the character in Z and the test is met; TFFF is set.

Exit

The Y and Z registers are cleared unconditionally. The syllable execute complete level is true to terminate the operator.

4.7 TEST FOR ALPHANUMERIC TANL-XX36

This operator compares the repeat field and the character in the source string, specified by the M and G registers, for a "greater than or equal" condition.

If the source character is greater than or equal to the repeat field character, and the source character is not the "multiply" character (external code 10 1010) or "not equal" character (external code 01 1010), the true/false indicator is set to true; otherwise, the true/false indicator is set to false. The M and G registers are not advanced.

A "not equal" character in the repeat field compared with a "not equal" character in the source string will result in setting the true/false indicator to true.

A "multiply" character in the repeat field compared with a "multiply" character in the source string will result in setting the true/false indicator to true.

Prior to the execution of the operator the H register is tested for zero. If the H register is not equal to zero, it is set to zero and the G register is increased by one. Overflow into the S register can occur.

SUMMARY OF OPERATION

This operator tests a character from the source string with the character in the repeat count field of the operator in the T register. The initial setting of the repeat count field is determined by the programmer. If the letter A is placed in the repeat count field of the operator, the test is for alphanumeric. If, however, a character other than the letter A is the initial setting of the repeat count field, the alphanumeric test will only hold for that portion of the collating sequence greater than the selected character; i.e., this operator will simply indicate the relative location of a character in the collating sequence with respect to the character it is being tested against.

To test for alphanumeric, the letter A should be in the repeat count field of the T register. This character is placed in the Z register and the character from the source string is placed in the Y register. Subsequently, a comparison output from the comparator of greater than will indicate that the character being tested is possibly an alphanumeric character. To be sure that the character is alphanumeric, it must then be determined that the tested character is not either:

1. A multiply symbol, or
2. A not equal symbol

Therefore, if a greater than comparison is initially encountered, the multiply character is placed into the Z register and the test is repeated. If the character is not a multiply symbol, the not equal character coding is placed in the Z register, with the test being repeated again. If the

character is not a not equal symbol, the character originally tested must be an alphanumeric character and therefore the true/false flip-flop is set to the one state.

Reflecting back to the beginning of the test operation, if in the initial comparison the greater than output was false, the character being tested is either equal to or less than the reference character. Therefore, the not equal comparison output is interrogated, and if false, the two characters must be equal to each other; i.e., equal to the letter A in the normal case. The true/false flip-flop is set to one and the operation terminated. If, however, both the greater than and the not equal outputs of the comparator are false, the character is a special character and the true/false flip-flop is left in the reset state indicating the comparison was not met.

DETAILED DESCRIPTION

J = 0

If the source string bit indicator (H register) is not equal to zero, it is set to zero and the character address register is increased by one. Should the G register equal seven, when it is incremented, the source string word address register is counted up by one (M+1) with the A register being marked unoccupied.

If the A register is initially unoccupied or marked unoccupied on entry, a read memory access is initiated by setting the E register to 4 and the J register is set to 1.

When the bit pointer is zero and the A register is occupied (AROF • HEZL), the character from the source string, as pointed to by the G register, is transferred to the Y register. The character in the repeat count field of the T register is transferred to the Z register. The J register is set to 2 for commencement of the comparison operation.

The true/false flip-flop is unconditionally set to zero because the logic is constructed to assume the comparison will not be met.

J = 1

Entry to this J register setting only occurs if the A register requires loading. When the memory read access is obtained, the A register is marked occupied with the J register returned to a zero setting.

J = 2

The following description assumes the repeat count field initially contains the letter A, which is the normal case.

YGZL'

If the comparator output, YGZL, is false, the source string character in the Y register is not greater than the reference character in the Z register, and is therefore either equal to or less than the reference character. The

logical flip-flop, Q03F, is set to the one state to inhibit the setting of the true/false flip-flop (TFFF) at J = 5. TFFF can still be set at this J register setting if the comparison results in an equality output; i.e., the tested character is the letter A. In any event, if YGZL' is true, the comparison is completed at this clock pulse time. Thus, the J register is set to 5 for subsequent termination of the operator.

YGZL

If the comparator output YGZL is true, the source string character in the Y register is greater than (in collating sequence) the reference character in the Z register. It must now be determined whether the tested character is a multiply symbol; therefore, the Z register is cleared in preparation for the insertion of a multiply symbol. The J register is set to 3 where the multiply symbol is set into the Z register.

YNZL' • T01F'

If the comparator output YNZL' is true, the two characters being compared are equal to each other. Therefore, the true/false flip-flop is set to one (TFFF to 1) and the J register is set to 5 for subsequent termination of the operator. The J register is set to 5 via the term YGZL' as described above. The term T01F is always reset for this operator and has no significance other than being part of common logic utilized by other operators.

J = 3

This J register setting is only arrived at if the initial comparison indicates the tested character is alphanumeric provided it is not either a multiply or not equal character. The Z register was cleared at J = 2, and by setting the B bit of the Z register to one (ZRBF to 1), the coding for the Multiply symbol is placed in the Z register (10 0000). The J register is set to 4 for execution of the comparison.

J = 4

At this time the multiply character in the Z register is compared against the character in the Y register. If the comparison results in the comparator output, YNZL, being true (the tested character is not a multiply symbol), the Z register is changed to end not equal symbol by setting the bits A, 8 and 4; the B bit is not altered (11 1100). The J register is set to 5 for the next comparison.

If, however, the comparison results in the comparator output YNZL' being true, the tested character is equal to a multiply symbol; i.e., it is a special character and is not an alphanumeric character. Therefore, the test is not met. The logical flip-flop Q03F is set to inhibit the setting of the true/false flip-flop at J = 5. It should be remembered that in order to arrive at this point with YNZL' being true, the repeat count field must have initially contained other than a multiply symbol.

J = 5

If this J register setting is arrived at from J = 4 and Q03F is still reset, the comparator output is interrogated. If YNZL is true, the tested character in the Y register is not a not equal symbol and must therefore be an alphanumeric character; TFFF is set to indicate the comparison has been met. The syllable execute complete level is true to terminate the operator.

If this J register setting is arrived at from J = 2, Q03F will be in the set state. The status of the true/false flip-flop would already have been established. Therefore, the syllable execute complete level is true to terminate the operator.

4.8 TEST BIT

TEBL-XX37

This operator tests one bit in the source string at the position specified by the M, G and H registers against the low-order bit of the repeat field. If they are equal, the true/false indicator is set to true; otherwise, the true/false indicator is set to false. The M, G and H registers are not advanced.

SUMMARY OF OPERATION

Upon entry to the operator, the true/false flip-flop is set to zero. The A register is loaded if initially unoccupied.

When the A register is loaded, the character specified by the G register is transferred to the Y register. A comparison is then made between a bit in the Y register (specified by the H register) and the least significant bit of the repeat count field of the operator in the T register. If both bits are set or if both bits are reset, the true/false flip-flop is set to one. The operation is then terminated.

DETAILED DESCRIPTION

J = 0

The true/false flip-flop (TFFF) is unconditionally cleared to zero prior to the comparison.

If the A register is unoccupied, it is loaded from the source string. The E register is set to 4 and the J register is set to 1, to complete the load.

If the A register is initially occupied, the character specified by the G register is transferred to the Y register. The J register is set to 2 for execution of the comparison.

J = 1

Entry to this J register setting only occurs if the A register was initially unoccupied. Upon completion of the load A memory access, the character specified by the G register is transferred to the Y register and the J register is set to 2.

J = 2

If the least significant bit of the repeat count field in the T register is the same as the bit in the Y register, as specified by the H register, (T07F • YOHL + T07F' • YOHL') the true/false flip-flop is set to one to indicate a true comparison. If the two bits are not alike, the true/false flip-flop is left in the reset state.

When the term YOHL is true, the indication is that the bit of the Y register, specified by the H register, is set (equal to one). The logic for YOHL is as follows:

```
YOHL = + YR1F • H04F • H01F                    - 5th bit position
       + YR2F • H04F • H01F'                   - 4th bit position
       + YR4F • H02F • H01F                    - 3rd bit position
       + YR8F • H02F • H01F'                   - 2nd bit position
       + YRAF • H04F' • H02F' • H01F           - 1st bit position
       + YRBF • H04F' • H02F' • H01F'          - 0   bit position
```

The syllable execute complete level is unconditionally true to terminate the operator.

4.9 COMPARISON OPERATORS - COMPARE CHARACTERS

COMPARE FOR EQUAL	SEQL-XX60
COMPARE FOR NOT EQUAL	SNEL-XX61
COMPARE FOR GREATER OR EQUAL	SGEL-XX62
COMPARE FOR GREATER	SGTL-XX63
COMPARE FOR EQUAL OR LESS	SLEL-XX70
COMPARE FOR LESS	SLTL-XX71

The comparison operators are used for comparing two identical length fields of alphanumeric characters. Comparisons are made on the basis of the collating sequence of characters. Fields may start at any position within a word; word boundaries are ignored. Although the result of the comparison may be known, before all characters of the fields have been compared, the address registers are advanced the full amount.

Each operator compares a character field in the source string, starting at the position specified by the M and G registers, with an identical length character field in the destination string, starting at the position specified by the S and K registers.

If the field in the source string is either;

1. Equal
2. Not Equal
3. Greater or Equal
4. Greater
5. Equal or Less
6. Less

Depending on the specific operator, the true/false indicator is set to true; otherwise, the true/false indicator is set to false.

The number of characters in each of the fields to be compared is specified by the repeat field.

SUMMARY OF OPERATION

The repeat count field of the operator in the T register specifies the number of characters that are to be compared. Any number of characters from zero to a maximum of 63 may be specified in the repeat count field.

Upon entry to the operator, the bit pointers are unconditionally cleared to zero. If either, or both, bit pointers are not equal to zero, the appropriate character pointer is incremented. If the character pointers overflow, the appropriate word address register is advanced. When the bit pointers equal zero, the A and/or B registers are loaded if required. With the B register occupied, the bit pointers are equal to zero and, provided the repeat count field does not equal zero, the B register is checked for alignment. If the B register is unaligned, it is aligned.

Comparison of characters can begin once the B register is in alignment and the A register is occupied. The comparison is effected from the most significant to the least significant character positions. The comparison is terminated upon encountering the first unequal pair of characters. When the first unequal comparison is encountered, the true/false flip-flop is set, if the character from the source string is greater than the character from the destination string; otherwise, the true/false flip-flop is left in the reset state. Subsequently, at the end of the operator, the status of the true/false flip-flop, together with the operator levels, determines the final setting of the true/false flip-flop. Although the result of the comparison is known, when the first unequal pair of characters is encountered, the operator is not terminated until the address registers (character and word) point at the character position following the last character position of the field being compared.

DETAILED DESCRIPTION

J = 0

Since there are numerous occurrences at J = 0, reference to the specific actions is made according to action sequence and relative to the double line divisions.

Unconditionally, the V and H bit pointers are cleared to zero. This only has significance if they are not initially equal to zero (see below). If on entry to the operator, the six high order bits of the T register (repeat count field) are equal to zero (TEZL), and once the bit pointers are equal to zero (VEZL), the J register is set to 5 for subsequent termination of the operator. Note that HEZL is implied to be true, when VEZL is true, and is not actually required as a term of the equation. The true/false flip-flop (TFFF) is cleared unconditionally in preparation of the compare operation that follows.

Inhibit Fetch

If either bit pointer is not equal to zero (HEZL' + VEZL'), fetch is inhibited to give precedence to the loading of the registers.

HEZL'

If the bit pointer for the A register is not equal to zero (HEZL'), the character pointer is incremented (G + 1) to address the next character in sequence. If, at that time, the character pointer is pointing at the last character of a word (GE7L), the source address register is incremented to address the next word in sequence. The A register is marked as unoccupied in order to facilitate loading of the A register. In this case, the first character of the next word is the character initially desired.

VEZL'

If the bit pointer for the B register is not equal to zero (VEZL'), the character pointer is incremented (K + 1) to point at the next character in sequence. If, at this clock pulse, the B register is loaded and is in alignment (KENL • BROF), the increment of the K register will cause misalignment of the B register. Therefore, to maintain alignment, the B register is shifted left and circulated two octade positions, with a corresponding increment of the N register. The first shift occurs at this clock pulse time, and the second shift occurs at J = 6.

If the character pointer for the destination string is pointing at the last character of a word (KE7L) at the time it is incremented, the B register is marked unoccupied to facilitate reloading. If the B register requires reloading when KENL is true, its present contents must first be stored in memory. if KENL is true when KE7L is true, the B register is not in its original configuration, as obtained from memory (it has been shifted), thus the alignment procedure described above serves to place the B register in its proper form for storage.

If the character pointer for the destination string is pointing at the last character of a word and the N register is equal to zero (KE7L • NEZL) at the time the character pointer is incremented, the destination string address register is incremented by one (S + 1) and the logical flip-flop Q04F is set. In this case, the B register is in its original configuration and, therefore, does not require storage in memory; i.e., it already is in memory in proper form. Thus, the only requirement is to reload the B register with the next word in sequence. Q04F is set to prevent any further storage of the B register contents. Only the first word of the destination accessed by this operator may require storage.

TEZL' • VEZL • HEZL

Once the bit pointers are equal to zero and the repeat count field indicates characters to be transferred, the A and B registers are loaded if required. If, on entry to the operator, the bit pointers were not equal to zero and the subsequent increment of the character pointers causes a word overflow, the A and/or B registers would be loaded (as required) regardless of their status on entry to the operator. If both registers are marked unoccupied, the B register is loaded first by the setting of E to 3. This allows the B register to be aligned while the A register is being loaded. Once the B register is occupied, a load of the A register is initiated by the setting of E to 4.

MROF

If MROF is true at J = 0, this can only be the completion of a memory access to load the A register. Therefore, the A register is marked as occupied.

BROF • TEZL' • VEZL

Alignment of the B register occurs when the B register is occupied, these are characters to be transferred and the bit pointers are equal to zero. The specific actions during the alignment process are as follows:

K04F (KENL' + N01F). The B register is not in alignment when KENL' is true; i.e., the character position pointed to by the K register is not in the output alignment station (15th and 16th octade positions of the B register). In this case, alignment of the B register is required. The 4's bit of the K register is interrogated to determine the most expedient method for the alignment procedure. If K04F is set, it is most expedient to shift and circulate the B register to the right, while decrementing the N register to tally the octade shifts. Since the N register tallies octade shifts and the K register points to a complete character position, one additional shift and circulation of the B register is required when N01F is true, even though the K equals N level (KENL) is true.

K04F' • KENL'. If the B register is not in alignment (KENL') and the 4's bit of the K register is reset (K04F'), it is most expedient to execute the alignment by shifting and circulating the B register to the left while incrementing the N register to tally the left shift.

KENL • N01F' • AROF • HEZL. Once the B register is in alignment and the A register is occupied, as indicated by KENL • N01F' • AROF, the character from the B register output alignment station is transferred to the Z register. The character from the A register, as pointed to by the G register, is transferred to the Y register. Simultaneously, the B register is shifted to the left and circulated one octade position; i.e., one half character shift. This is the first of the required two octade shifts to place the next character into the output alignment station. The N register is incremented to keep track of the shifts and the J register is set to 2 to commence the comparison operation.

J = 1

Entry to J = 1 only occurs if the B register requires loading at J = 0. When the memory access is complete (MROF) to load the B register with a word from the destination string, the B register is marked occupied and the J register is returned to zero. If, at this time, the A register is marked unoccupied, a memory access is initiated to load the A register with a word from the source string.

J = 2

The occurrences at this J register setting are separated into three groups by two sets of double lines. The middle group is only effective prior to detecting an unequal comparison. The bottom group is only effective when, and after, an unequal comparison has been found.

On entry to J = 2, the Y and Z registers are cleared, because the comparison is performed at this same clock pulse time. The B register is shifted left and circulated one octade position, thus completing the second of the required two shifts to place the next character to be compared into the B register output alignment station. The N register is counted plus one to tally the left shift of the B register. Note that the N register is only incremented if the B register is marked as occupied (BROF). This permits the N register to equal zero at the termination of the operator if a not equal comparison is encountered, prior to the repeat count field being counted down to zero. The

character pointers, G and K, are incremented to point at the next character in sequence, with the repeat count field in the T register being decremented to tally the comparison. The increment of G and K, plus the decrement of T, only occurs if the B register is marked occupied or if the repeat count field does not equal zero, but is less than eight (BROF + TL8L • TEZL'). This logic provides for counting the character pointers to the character position following the field being compared, in the event a not equal comparison is encountered before the end of the field of characters is reached. BROF is reset only if a not equal comparison is encountered (see below).

If the comparison of the current pair of characters in the Y and Z registers results in a not equal condition (YNZL), the logical flip-flop Q03F is set. If in addition to a not equal condition being encountered, the character in the Y register is greater than the character in the Z register (YGZL), the true/false flip-flop is set (TFFF to 1). Note that the true/false flip-flop may only be set upon encountering the first pair of not equal characters, because of Q03F. If the comparison results in an equality, Q03F remains reset and the operation proceeds to obtain the next pair of characters for comparison. Whenever an unequality is reached, comparison ceases and the remaining characters are skipped; i.e., the character and word address registers are advanced to the character position following the end of the field.

Q06F' - Store Repeat Count Field

The first entry to J = 2 occurs with the logical flip-flop Q06F in the reset state. At that time the repeat count field of the T register is stored into the H and V registers. This logic only has significance if the operator is Field Add/Subtract but is not inhibited if a compare operator is being executed. Note that this can only occur for the initial setting in the T register; thereafter, Q06F is set.

Q03F' • YNZL'

This logic exists when an inequality has not previously been detected (Q03F') and the characters presently in the Y and Z registers are equal (YNZL'). Thus, additional characters remain to be compared.

When the character pointers are incremented, they are tested for overflow. If the G register overflows (GE7L + 1) and the K register does not (KE7L'), the source string address register is counted up by one (M + 1) and a read memory access is initiated to load the A register, by setting the E register to 4. The J register is set to 5 to await completion of the memory access and then to continue the compare operation. If the K register overflows (KE7L + 1) and Q04F is reset, the contents of the B register must be stored prior to reloading the B register. Q04F being reset indicates that the word currently in the B register is the first word with characters to be compared. Since a part of this word may belong to another field, it must be stored prior to reloading the B register contents. Note that if the B register character pointer overflows, preference is given to the storage of the B register contents regardless of whether the character pointer for the A register overflowed. The J register is set to 3 for the reloading of the B register or if the G register overflowed at the same time, the A register is loaded first (at J = 3).

Q03F + YNZL

If Q03F is set, the comparison process has already been completed; i.e., the most significant unequal pair of characters had been reached. If YNZL is true and Q03F is reset, the current pair of characters in the Y and Z registers are the first unequal pair of characters to be encountered. In any event, the comparison is completed, either at this clock pulse or prior to this clock pulse. Thus the character pointers and the word address registers, for the source and destination string, are advanced to the character position following the last character position of the specified field. This is accomplished in the following manner: for example, assume a field length of approximately three or four words and that an unequal comparison is encountered.

If an inequality is encountered at any character position other than the last character position of the word in the B register (KE7L'), regardless of the relative character position in the A register, the character pointers are advanced until the last character position of the word containing the unequal comparison in the B register is reached with the repeat count field in the T register being decremented (G+1, K+1, and T-1 gated via BROF). This is accomplished by cycling between J = 2 and J = 5 in order to restore the N register to zero. In this case the J register is set to 5 via BROF • KE7L'.

When the K register points at the last character position of the word (KE7L), the B register is checked to determine if this is the first word of the destination string accessed by the operator. If this is the first word, Q04F will be in the reset state. Therefore, a store memory access is initiated with Q04F being set to the one state. At the same time, the B register is marked as unoccupied. The J register is set to 4 to await the completion of the store memory access, after which a return is made to J = 2. If the first word had already been stored, indicated by Q04F in the set state, and the K register points at the last character position of the word, the B register is marked unoccupied with both the K and S registers being advanced to the first character position of the next word.

Upon a return to J = 2 (completion of the store memory access) or if the first word has already been stored and the character pointer for the B register is pointing at the first character position of the next word, the word address registers are advanced by word and the T register is counted down by eight (M+1, S+1 and T-8 gated via TL8L' • BROF' • TEZL') until the repeat count field becomes less than eight; less than a full word remains. The character pointers are not advanced while this advancement of the word registers is being accomplished. Note that the G register does not have to point at the beginning of a word, in order to increment by words.

When the repeat count field equals less than eight, incrementing of the word address registers ceases and counting of the character pointers (G and K) again commences. When the repeat count field equals one and provided the B register is unoccupied (TE1L • BROF'), the J register is set to 5 for subsequent termination of the operator; i.e., the T register decrement at this clock pulse will cause the repeat count field to go to zero. If, however, the B register is still occupied when the repeat count field is equal to one, the exit to J = 5 must await TEZL.

The actions just described may not all occur as described, being dependent on whether an inequality is encountered and at what point in the string the inequality is detected.

J = 3

If the character pointer for the B register equaled seven (KE7L) at J = 2, the J register would have been set to 3. If a memory access were initiated to store the B register contents, all actions await completion of the access (EEZL). At this time the G register is tested to determine if it also overflowed when the K register overflowed. If the G register did overflow, it would now equal zero (GE7L+1 = GEZL). Therefore, the source address register is incremented (M+1), a load memory access to load the A register is initiated, and the J register is set to 4. If the G register did not overflow, the B register is reloaded, the destination string address register is incremented (S+1), a load B memory access is initiated and the J register is set to 5.

J = 4

Entry to this J register setting occurs under two different circumstances:

1. If an inequality had not yet been encountered and both the A and B registers required reloading at J = 2, a transfer to J = 3 would have occurred to load the A register, after which the J register was set to 4 for subsequent reloading of the B register. In this case, BROF is in the set state. Therefore, upon completion of the load A memory access (MROF), the E register is set to 3 for reloading the B register and the J register is set to 5.

2. If an inequality had been encountered and the first word of the destination string required storing (Q04F was still off at J = 2), a memory access would have been initiated to store the B register contents with BROF being reset. Therefore, J = 4 is entered to await completion of the store memory access of the B register contents. Upon completion of the store memory access (MWOF), the destination string address register is incremented by one (S+1) and the repeat count field is interrogated to determine if it is equal to zero. If the T register is equal to zero, the J register is set to 5 for subsequent termination of the operator. If the T register is not equal to zero (TEZL'), the J register is returned to J = 2 for counting of the character pointers and word address registers to point at the character position following the last character position of the specified field.

J = 5

The occurrences at J = 5 are separated into two groups, depending on whether the repeat count field equals zero or not. This J register setting may be entered with a memory access in progress to load either the A or B register, or it may be entered with no memory access in progress. If entered with a memory access in progress, all actions await completion of the accesses.

TEZL' • EEZL

If the repeat count field has not been reduced to zero, the next pair of characters is transferred to the Y and Z registers for subsequent comparison at J = 2. This action may, however, be redundant if an inequality had already been encountered (see discussion under J = 2 heading). The B register receives the first of the two required octade shifts to accomplish a full character shift. The N register is incremented to tally the shift and the J register is returned to J = 2.

TEZL • EEZL

If the repeat count field is equal to zero and the operator is one of the Compare operators, the Field Add/Subtract level will be false (FASL' will be true). FASL' will gate the bit pointers to a cleared status and the syllable execute complete level is true to terminate the operator. In this case, the setting of the J register to zero has no significance.

If, however, the operator is either Field Add or Field Subtract (FASL' is false), the operator can only be exited at this time if initially the repeat count field of the operator in the T register equals zero. Q06F will be in the reset state in this case. The clearing of Q06F at this time has significance only if the operator is Field Add or Field Subtract and the repeat count field does not initially equal zero. In this case, the next sequence of events occurs on the Field Add-Subtract flow chart (see Field Add-Subtract write up).

TFFF to 0

If during the comparison operation an inequality were encountered, Q03F would be in the set state at this time. If in addition to the inequality, the field of the source string proved to be greater than the field of the destination string (YGZL), TFFF would be in the set state at this time. It must now be determined whether TFFF is in the correct status relative to the operator in the T register. If the operator is:

1. Compare for Less, or Compare for Equal or Less; indicated by T04F • T02F'

2. Compare for Equal, or Compare for Equal or Less; indicated by T02F' • T01F, and the true/false flip-flop is set (Source greater than destination), the test has not been met and therefore the true/false flip-flop is reset, (TFFF to 0).

TFFF to 1

If the test had been met, the true/false flip-flop must be set at this time if it is in the reset state. If the operator is:

1. Compare for Equal or Less; indicated by T04F • T02F' • T01F', and both Q03F and TFFF are reset, the test is met and TFFF is set to the one state. If, however, Q03F is set, the test has not been met, and if TFFF is also in the set state, this logic to set TFFF has no significance as it will complement to the zero state by the above logic (see TFFF to 0).

2. Compare for Not Equal, or Compare for Less; indicated by T02F' • T01F with Q03F in the set state, the test has been met for not equal. If in addition, TFFF is reset, the test has been met for Compare for Less. If, however, TFFF is set, the test has not been met for Compare for Less and therefore, this logic has no significance because TFFF complements to the zero state (see TFFF to 0).

3. Compare for Equal, or Compare for Greater or Equal; indicated by T04F' • T01F' with Q03F in the reset state, the test has been met and TFFF is set to the one state.

Q04F and Q06F to 0

The logical flip-flops Q04F and Q06F are cleared at the termination of the operator. This has significance only if the operator is Field Add/Subtract and the operator level is being changed; i.e., if the operation is Compare, the syllable execute complete level would clear Q06F and Q04F.

J = 6

This J register setting is only attained if, on entry to the operator, the V register was not equal to zero and the B register was initially occupied and in alignment with the K register (KENL). If at J = 0, the K register did not equal seven, the B register is brought back into alignment (K was incremented) by two half character shifts. The first at J = 0 and the second at this J register setting. In this case, BROF remains set and gates a return to J = 0.

If, however, the K register did equal seven when KENL was true, BROF would have been reset, indicating that the B register contents are not a part of the destination field and, therefore, require storage. In this case the two half character shifts are required to place the B register contents in the proper configuration, prior to storage. Thus on entry to J = 6 the level EWZL gates the left shift of the B register and also initiated a memory access to store the B register contents. Upon completion of the store memory access, the destination string address register is incremented (S+1) and the J register is returned to zero. (EWZL = EEZL • MWOF') Q04F is set to remember the first word of the destination has been stored.

4.10 FIELD ADD / FIELD SUBTRACT

FADL-XX73 & FAXL-XX33
FSUL-XX72 & FSXL-XX32

FIELD ADD

This operator algebraically adds a source field, whose most significant position is specified by the M and G registers, to a destination field, whose most significant position is specified by the S and K registers.

The lengths of the two fields are equal and are specified by the repeat field. The result is stored in the destination field. The zone positions of the least significant character of a field contain the sign of the field. (BA = 10 is minus, all other combinations are plus). All zone positions, other than the sign, are set to 00; a plus sign is BA = 00 and a minus sign is BA = 10.

If the result of the addition overflows the destination field, the overflow is lost and the true/false indicator set to true; otherwise, the true/false indicator is set to false.

If the operation is an arithmetic addition and both fields are minus zero, the result is minus zero. If the operation is an arithmetic subtraction, the source field is plus zero and the destination field is minus zero, the result is minus zero. In all other instances, a result of zero is a plus zero.

FIELD SUBTRACT

This operator algebraically subtracts a source field from a destination field in the manner described in the field add operator.

SUMMARY OF OPERATION

Prior to executing a Field Add or Field Subtract operator, the two fields to be algebraically added or subtracted must be compared. The comparison is a numeric comparison and is performed to facilitate the algebraic addition or subtraction without the possibility of obtaining a complement answer. The Compare Characters Operator is utilized in the performance of this comparison. Once the comparison is completed, the status of the true/false flip-flop and Q03F indicate the relative magnitude of the two fields. If Q03F is set, the two fields are unequal. If TFFF is reset (and source field is greater than the destination field. If TFFF is rest (and Q03F is set), the source field is less than the destination field.

Reference should be made to the Compare Characters Flow Chart and the corresponding write-up, prior to proceeding with this write-up. The entire Compare Characters operator is executed as the first phase of Field Add/ Subtract. Upon termination of the compare phase, the character and word registers point at the character position following the last character position of the field just compared. Therefore, the character and word pointers must be decremented by one to point at the least significant character position of the field to be algebraically added or subtracted.

The performance of the algebraic addition or subtraction takes place in serial fashion via the Y and Z registers. To facilitate this action, the first operation of the add/subtract phase of the operator is to load the A and B registers from the source and destination strings, respectively, if they do not initially contain the proper words; i.e., the two words containing the least significant character position of each field. Once the A and B registers are loaded with the proper words, the B register must be brought into alignment. In this case, alignment implies that the character position pointed to by the K register is in the second output alignment station; i.e., the 2nd and 3rd octade positions of the B register. This is required because the normal output alignment station (15th and 16th octade positions) is utilized as an input station for the resultant sum as developed. As each sum digit is developed, the B register is shifted to the right. Thus at the completion of a full word, it is in proper form for storage into the destination string with further shifting being required. Only if the most significant word of the field is partially utilized, will further shifting be required to place the word in proper form for storage in the destination string. Since the normal alignment logic is constructed to have the character pointed to by the K register in the 15th and 16th octade position when the B register is in alignment, the N register must be preset prior to the alignment procedure. Therefore, during the add/subtract operation, the K equals N level (KENL) implies that the character position, specified by the K register, is in the 2nd and 3rd octade positions of the B register.

Once the B register is in alignment, and the A register is loaded, the least significant digit of the two fields is transferred to the Y and Z registers. A sign comparison is then performed to determine whether either of the two fields require complementing for the performance of subtraction. Thereafter, either an add or complement add operation takes place in a serial fashion until all character positions of the field have been exhausted. At the completion of the add or subtract operation, the most significant word is checked for alignment, prior to storage into the destination string. In this case, the N register is already preset to perform the alignment. Also, at completion of the add or subtract operation, the character and word address registers are pointing at the most significant character position of their respective fields. Therefore, the character and word address registers must be decremented to the character position following the least significant character position of the respective fields.

DETAILED DESCRIPTION

Since the Compare Characters Operator is utilized as the first phase of the Field Add/Subtract operator, this description starts at J = 5 of the Compare Characters Operator Flow Chart.

J = 5 - Compare Characters Operator Flow Chart

It is assumed that the last two characters of each field (least significant) have been compared and TEZL is true. This causes the J register to be set to zero (in this case J = 0 of the Field Add/Subtract operator). If, initially, the repeat count field were equal to zero, the logical flip-flop, Q06F, would be in the reset state at this time and would, therefore, gate the syllable execute complete level true to terminate the operator. If the repeat count field is not initially equal to zero, Q06F would be in the set state at this time. Because the operator is either Field Add or Field Subtract, the Field Add/Subtract Level FASL is true (FALS' is false). With FASL' and Q06F' both false, the syllable execute complete level cannot go true. Due to the operator coding, FASL equals XX72 or XX73, and the logic to alter the status of the true/false flip-flop is inhibited. At this clock pulse time, both Q04F and Q06F are reset. Q06F has no further function and Q04F serves in another function during the add/subtract phase of the operation.

The remaining functions, that occur at J = 5, are shown on the Field Add/Subtract Flow chart.

J = 5 - Field Add/Subtract Flow Chart

The function of setting the J register to zero and Q06F' causing an Exit are shown on both flow charts; the significance of this logic has been described above.

If the operator in the T register is either Field Add or Field Subtract, FASL is true, and provided Q06F is set, the following actions occur: The 6th bit of the T register is cleared to zero, thereby causing either the Field Add Aux level or the Field Subtract Aux level to be true. This terminates the Compare Characters flow and commences the Field Add/Subtract operation.

Because the character pointers are addressing the character position, following the least significant digit of the field, the G and K registers are counted down by one to address the least significant digit of their respective fields. If either or both the character pointers equal zero, when the character pointers are decremented, the desired least significant digit is in the prior word of the respective field. If GEZL is true, the source address register is decremented (M-1) and the A register is marked unoccupied to permit reloading. If KEZL is true, the destination address register is decremented (S-1) and the B register is marked unoccupied to permit reloading.

If, on entry to the add/subtract phase of the operation, the B register is occupied with valid information and the K register does not equal zero (BROF • KEZL'), the B register contents may be brought back into alignment; the B register received an extra character shift of its content in conjunction with the increment of the K register during the compare phase. Therefore, the B register is shifted to the right and circulated one octade position, the second octade shift occurs at J = 0. The N register is decremented to tally the right shift. If the B register does not contain valid information at this time, shifting of the B register is unnecessary

but uninhibited, whereas counting of the N register must not occur and is therefore inhibited. Remember, if the B register does not contain valid information, the N register is equal to zero on entry to this phase of the operation.

J = 0

The second octade shift of the B register contents takes place. If the B register contents are invalid, this shift is unnecessary but uninhibited. If the B register contents are valid (BROF), the N register is decremented. Thus, the N register equals zero on exit of J = 0 to J = 1.

The repeat count field value which had been stored in the bit registers, H and V, (during the compare phase) is set back into the repeat count field of the T register. The repeat count field value remains in the H and V registers for further use at the termination of the operator.

J = 1

If either or both the A and B registers are marked unoccupied at this time, the appropriate memory accesses are initiated to load the A and/or B registers.

Once the A and B registers are occupied (they may be on entry to this J register setting), the N register is interrogated to determine if it equals zero. If the N register equals zero (NEZL), the word in the B register is unshifted, the most significant digit occupies the 15th and 16th octade positions while the least significant digit occupies the 1st and 2nd octade positions. In this case, the N register is preset to 13, so that at J = 6 the alignment procedure will place the character pointed to by the K register into the 2nd and 3rd octade positions of the B register.

If, however, the N register is not equal to zero (NEZL'), the B register contents are in alignment relative to the K register; i.e., KENL is true and therefore the character pointed to by the K register is in the 15th and 16th octade positions of the B register. Because the N register setting is not known, the N register cannot be preset as when N equals zero. Instead, all that is required is to shift the B register left and circulate three times by octades. This will place the desired digit into the 2nd and 3rd octade positions of the B register. The first of the required shifts occurs at this J register setting. The other two required shifts occur at J = 3 and J = 4. Note that the K register is not counted for any of the three shifts of the B register, thus KENL will be true when the desired digit is in the second output station.

J = 2

This J register setting occurs if at J = 1 a memory access were initiated to load the B register from the destination string. At the completion of the access the B register is marked occupied and, if the A register requires loading, a memory access is initiated to load the A register from the source string. The J register is set to 1 to continue the operation.

J = 3

At this J register setting the second of the three required octade shifts occurs to place the desired least significant digit into the second output alignment station. See discussion under the paragraphs headed by J = 1.

J = 4

At this J register setting, the third of the required three octade shifts occurs to place the desired least significant digit into the second output alignment station. See discussion under the paragraphs headed by J = 1.

J = 6

This J register setting is utilized to realign the word in the B register so that the least significant character of the destination string is in the second output alignment station, (the 2nd and 3rd octade positions). Thereafter, the two least significant digits from the source and destination strings are transferred to the Y and Z registers, respectively. For the following logic, it must be remembered that initially N = 13 is coming from J = 1.

KENL' • (KO4F' • KE7L)

If KENL' is true, the desired digit is not in the second output station of the B register. If KO4F is reset; the most expedient method to attain alignment is by shifting left and circulating with a corresponding increment of the N register, to tally the shifts. If the K register is equal to seven (KE7L) when the N register equals 13, the desired digit must be in the 1st and 2nd octade positions. Thus, only one octade shift to the left and a corresponding circulate is required to place the least significant digit into the 2nd and 3rd octade positions of the B register.

KENL' • KO4F • KE7L' + KENL • NO1F

If the K register is pointing to either the 4th, 5th or 6th character position of the word, but not the 7th character position (KO4F • KE7L'), it is most expedient to shift to the right and circulate to attain alignment to the second output station. Once KENL is true, an additional right shift and circulate is required, if NO1F is true.

KENL • NO1F'

When KENL and NO1F' is true, the desired least significant digit is in the second output alignment station of the B register. The least significant digit from the source string is pointed to by the G register and is transferred from the A register to the Y register. The desired least significant digit from the destination string, as pointed to by the K register, is transferred from the B register (second output alignment station) to the Z register. The J register is set to 7, where the next operation will be a comparison of the signs of the two fields.

J = 7

At this J register setting the correct sign of the result is determined by a sign comparison of the two fields. It is also determined whether a complement add is to be performed, and if so, which of the two operands require complementing. On the initial entry to J = 7, Q08F is in the reset state and gates the actions above the double line. Unconditionally the logical flip-flop Q08F is set to inhibit the actions above the double line and enable the actions below the double line. Q08F is only in the reset state during the least significant digit and is set for all other digits.

NEGL • Q08F'

If NEGL is true, the resultant sign of the operation for either Field Add or Field Subtract (FAXL or FSXL) is negative. Q09F is set to store this fact for subsequent insertion into the least significant character position of the destination string (at J = 10). The logical equation for NEGL follows (note that the equation for NEGL is switched):

- O'-	NEGL	Significance of Terms
-I-	+ ZRAF • CADL'	(Z+ • CADL')
	+ ZRBF' • CADL'	(Z+ • CADL')
	+ ZRAF • Q12F'	(Z+ • Y less than Z)
	+ ZRBF' • Q12F'	(Z+ • Y less than Z)
	+ ZRAF' • ZRBF • CADL • Q12F	(Z- • CADL • Y greater than Z)
	+ CADL • Q03F'	(CADL • Y equals Z)

The term Q12F is equivalent to TFFF, thus when Q12F is true the source field is greater than the destination field (Y is greater than Z). The term CADL, when true, indicates that a complement addition is to take place. The equation for CADL is as follows:

```
CADL = T01F • YZSL'     (Field Add with sign unequal)
     + T01F' • YZSL     (Field Subtract with signs equal)
```

The term YZSL, when true, indicates that the signs of the two operands are equal. When YZSL' is true, the signs of the two operands are unequal. If T01F is set, the operand is Field Add and when T01F is reset the operand is Field Subtract. The equation for YZSL is as follows:

```
YZSL = + YRBF' • ZRBF'  (both signs positive)
       + YRBF' • ZRAF   (both signs positive)
       + YRAF  • ZRAF   (both signs positive)
       + YRAF  • ZRBF'  (both signs positive)
       + YRBF  • YRAF'  (both signs negative)
        • ZRBF  • ZRAF'
```

The algebraic rules of addition and subtraction state that if the operation is:

ADD - Like Signs; add and use the common sign
- Unlike Signs; subtract smaller from larger and use sign of larger.

SUBTRACT - Change sign of subtrahend (source string, Y register) and proceed as in Add.

It should be noted that the sign of the source string is not actually changed. Instead the logic is constructed to observe the algebraic rules of subtraction without requiring an inversion of the sign. The Truth Table For Result Sign (Figure 4.10-1) illustrates what the result sign will be, based upon the initial operator, the various sign configurations, and the relative magnitudes of the two operand fields.

OPERATOR SYLLABLE	Y SIGN	Z SIGN	CADL	RESULT SIGN - IF		
				Y = Z	Y < Z	Y > Z
ADD	+	+	0	+	+	+
ADD	-	+	1	+	+	-
ADD	+	-	1	+	-	+
ADD	-	-	0	-	-	-
SUB	+	+	1	+	+	-
SUB	-	+	0	+	+	+
SUB	+	-	0	-	-	-
SUB	-	-	1	+	-	+

	Q03F	Q12F
Y = Z	0	0
Y < Z	1	0
Y > Z	1	1

Q12F = TRFF
Q12F = Y > Z
Q03F = Y ≠ Z

FIGURE 4.10-1. TRUTH TABLE FOR RESULT SIGN

CADL • Q08F'

If the Complement Add level (CADL) is true, Q02F and Q07F are set to the one state. Q02F indicates to the adder logic that a complement is required. Q07F is the carry storage input to the serial adder (see the section on Serial Adder). Therefore, Q07F is set for an end around carry function to change a nines complement to a tens complement.

TFFF • CADL • Q08F'

If TFFF is true, in addition to CADL, the logical flip-flop Q04F is set. If complement addition is required, one of the operands will be complemented. If the source string is greater than the destination string (Y is greater than Z), the field of the destination string is the one to be complemented. The following states which field will be complemented and is based on the settings of Q02F and Q04F:

Q02F' • Q04F' = No complement add
Q02F • Q04F' = Complement Y (source string)
Q02F • Q04F = Complement Z (destination string)

Q08F

On the first entry to J = 7, the operation is to execute a sign comparison as described above. Based on the sign comparison, one of the two operands may require complementing. Due to the construction of the adder logic, only the Z register input is gated in complement form. If the Y input requires complementing, it is complemented within the adder logic (see the section on Serial Adder). Therefore, upon initiating the actual add operation, the least significant digit is again transferred to the Z register. If Q04F is set, the input to the Z register is in nines

complement form. If Q04F is reset, this transfer of the least significant digit is redundant. On subsequent entries to J = 7, the input to the Z register is in proper form, based on the status of Q04F.

The digit from the A register, as pointed to by the G register, is transferred to the Y register. During the first entry to J = 7, this is redundant. The J register is set to eight for performance of the binary add. At the same time, the B register is shifted to the right and circulated one octade position with a corresponding decrement of the N register. This places one-half of the next digit, to be added, into the second output alignment station.

J = 8

The numeric bits of the Y register are added to the numeric bits of the Z register with the straight binary sum placed into the Z register. The binary carry results, if any, are placed into the Z register B bit position. The B register is shifted to the right one octade position with a corresponding decrement of the N register. This is the second of two required octade shifts, to place the next digit to be added into the second output alignment station.

The B register 16th octade position is cleared to zero in conjunction with the shift of the B register contents. This is the preparation for receiving the decimal correct sum at J = 10. The repeat count field in the T register is counted down by one to tally the addition. The true/false flip-flop (TFFF) is cleared, as its present function of indicating a greater than is ended and it will be used in a new function at J = 10. Q07F is reset because its carry function has been utilized for the current digit.

As long as the repeat count field is greater than one (TE1L'), the character pointers G and K are decremented by one to address the next more significant digits of their respective strings. If the G register is pointing at the most significant digit of a word (GEZL), the source address register is counted down by one and a memory access is initiated to load the A register. If, at the same time, the K register is pointing at the most significant digit of a word (KEZL), the logical flip-flop, Q05F, is set to remember that the B register contents are to be stored at J = 10. The J register is set to 10 where the straight binary sum in the Z register is decimally corrected and placed into the B register.

J = 9

This J register setting is attained if, at J = 5, Q05F were set to indicate the end of a word. Subsequently, at J = 10 a store memory access would have been initiated and the J register set to 9. Upon completion of the memory access to store the B register contents (MWOF • EEZL), a memory access is initiated to load the B register, provided that the repeat count field does not equal zero (TEZL'). The destination string address register is counted down (S-1) to address the next more significant word of the destination string.

TEZL • MWOF

If there are no more characters to be added (TEZL) and the access for storing the B register contents has been obtained (MWOF), the value of the repeat count field is restored to the T register from the H and V registers. This restoration of the repeat count field is required because the character pointers and word address registers must be counted up to point at the character position, following the least significant position of their respective fields. The J register is set to 12 for the counting process and subsequent termination of the operator.

EWZL • TEZL'

Upon completion of the load B memory access, which was initiated at this J register setting, if the repeat count field did not equal zero, the B register is shifted left and circulated one octade position, with an increment of the N register to tally the shift. When obtaining a new word from memory, a single octade shift to the right is required to place the least significant digit of that word into the second output alignment station (2nd and 3rd octade positions). The J register is set to 7, where the Y and Z registers are loaded for the next addition cycle.

J = 10

This J register setting follows the character addition which developed a binary sum and placed that sum into the Z register, bit positions B, 8, 4, 2 and 1. At this time, via a decimal correct matrix, the straight binary sum is available as a decimally corrected binary coded decimal digit. If entry to this J register setting occurs with a memory access in progress (A register may have required reloading at J = 8), all actions await completion of the access as indicated by EEZL. Upon completion of the access, if initiated, the binary coded decimal corrected sum digit is transferred to the B register output alignment station (15th and 16th octade positions of the B register). Simultaneously, the B register octade positions 1 through 15 (excluding 16th) are shifted to the right, one octade position. Two functions are served by this shift:

1. The contents of the 15th octade position are cleared to provide for insertion of the new information (the 16th octade was cleared at J = 8).
2. This is the first of two octade shifts to place a subsequent digit into the second output alignment station of the B register.

The N register is decremented to tally the right shift of the B register. Note that, if in the process of developing a decimal corrected digit, a decimal carry may have resulted as indicated by the level ZDCL being true. If ZDCL is true, the carry flip-flop Q07F is set so that the carry may be added into the next digit. The true/false flip-flop (TFFF) is also set if the level ZDCL is true. However, this only has significance if it occurs when the most significant digit is being transferred to the B register. In this case, TFFF being set to the one state will indicate that an overflow of the destination has resulted.

At the time the least significant sum digit is placed into the B register, the resultant sign of the algebraic addition or subtraction must be placed into the least significant digit position. Therefore, if Q09F is in the set state (the sign is negative), the 48th bit of the B register is set to one; otherwise, B48F is left in the reset state. Q09F is unconditionally reset to prevent insertion of the sign into any other position of the field.

At the same clock pulse, that transfers the decimally corrected sum digit from the Z register to the B register, the next pair of digits to be algebraically summed are transferred to the Y and Z registers. The digit from the A register, as pointed to by the G register, is placed in the Y register in true form. Based on the status of Q04F, the digit from the B registers second output alignment station is transferred to the Z register, in either complement or non-complement form.

Before exiting this J register setting, the status of Q05F is checked to see if the B register contents require storage. If Q05F is set, a store memory access is initiated (E to 11) and the J register is set to 9 for the subsequent reloading of the B register, provided the repeat count field does not equal zero. If Q05F is in the reset state, the J register is set to 8 to continue the addition operation (if T is not zero); otherwise, the J register is set to 11 for subsequent termination of the operator.

J = 11

When the J register equals 11, the B register contents are placed into proper alignment for storage into memory and the least significant digit of the word must occupy the 1st and 2nd octade positions of the B register prior to storage. Because the N register was preset to the value of 13 (at J = 1), when the B register contents were in proper configuration, the restore logic, at this time, aligns the B register contents until the N register again equals 13. When N equals 13, the B register contents will be in the proper form for storage into memory. If N is not equal to 13 (N13L') and the 4's bit of the K register is reset, it is most expedient to shift right and circulate to attain restoration of the B register contents. The N register is decremented to tally the right shifts. If, however, the 4's bit of the K register is set, it is most expedient to shift left and circulate to attain restoration of the B register contents. The N register is incremented to tally the left shifts. When the N register equals 13, a store memory access is initiated by setting the E register to 11. Note that the initiation of the memory access is gated by the term EWZL (EWZL = EEZL • MWOF'). This prevents a second memory access from being initiated at the completion of this access. At the completion of the access (MWOF) the J register is set to 12 for subsequent termination of the operator.

The repeat count field, which is stored in the H and V registers, is unconditionally stored to the repeat count field of the T register. This will permit gating the character pointers and word address register, so that they point at the character position following the least significant character of their respective fields.

J = 12

The A and B registers are unconditionally marked unoccupied, with the N, H and V registers being cleared to zero.

TFFF to 0

If the logical flip-flop, Q02F, is set at this time, the operation just accomplished was a complement add. Therefore, if TFFF is set, at this time, its significance of indicating an overflow of the destination field is not valid and TFFF must be cleared to zero. If the operation was a normal add (Q02F in the reset state) and TFFF is in the set state, it remains set to indicate an overflow of the destination field.

Restore Pointers

If the repeat count field is greater than eight (TL8L'), the word address registers for the source and destination strings are counted up by words. The repeat count field in the T register is decremented by eight to tally the word counts.

When the T register repeat count field becomes less than eight, count up by word ceases and count down by character commences. If either or both character pointers overflow, the appropriate word address register is increased by one. When the repeat count field equals zero and, provided the memory access is completed to store the B register contents (EEZL • TEZL). the syllable execute complete level is true to terminate the operator.

4.11 JUMP FORWARD UNCONDITIONAL
JUMP FORWARD CONDITIONAL

FWJL-XX47
CFJL-XX45

JUMP FORWARD UNCONDITIONAL

This operator initiates an unconditional forward jump. The C and L registers are increased by the contents of the repeat field. Prior to the addition of the repeat field of this operator, the C and L register effectively contain the address of the next syllable in sequence.

JUMP FORWARD CONDITIONAL

This operator initiates a Jump Forward Unconditional if the true/false indicator is false. If the true/false indicator is true, the next syllable in sequence is fetched. The true/false indicator remains unchanged.

SUMMARY OF OPERATION

Both operators jump over a number of consecutive program syllables in the program string. The number of syllables jumped is specified by the repeat count field and is limited to a maximum of 63.

If the repeat count field is zero for both operators, or if the true/false flip-flop is in the set state for a conditional jump, the operator acts as a NO-OP. The operator is terminated and the next syllable in sequence is executed.

If the repeat count field is not equal to zero, and either the operator is Unconditional Jump or Conditional Jump with the true/false flip-flop in the reset state, the operation proceeds to increment the C and L registers. To speed the jump operation, the program address register is counted by words, with the repeat count field decremented by 4, as long as the repeat count field is greater than four. When the repeat count field is less than four, the syllable address register is counted by one, with the repeat count field decremented by one. If this results in the syllable address register overflowing, the program word address register is increased by one. When the repeat count field equals zero, the syllable address register receives its final count and the operation terminates.

DETAILED DESCRIPTION

J = 0

If the repeat count field is equal to zero (TEZL), or the operator is conditional jump and the true/false flip-flop is set (CFJL • TFFF), the J register is set to 2 for subsequent termination of the operator. This effectively acts as a NO-OP operation.

If the repeat count field is not equal to zero (TEZL'), and the operator is either an unconditional jump or conditional jump with the true/false flip-flop in the reset state, the J register is set to 1 and the inhibit count up for fetch level is true (ICFL' is false). Inhibit Count Up For Fetch is required to prevent an extra undesired count of the L register. It also prevents PROF from being set to the one state (J = 0 of Fetch).

J = 1

If the repeat count field is equal to or greater than four (TL4L'), indicating that at least one word is to be skipped, the program address register (C) is counted up by one and the repeat count field counted down by four.

If the repeat count field is less than four (TL4L), the syllable pointer register (L) is counted up by one and the repeat count field counted down by one. When the repeat count field equals zero, the TL4L level is still true and L is incremented once more with T being decremented. Therefore, the C and L registers receive one count more than indicated by the T register; i.e., if T equals 1, L is counted twice.

When the repeat count field is less than four, and if the syllable pointer register is equal to three (TL4L • LE3L), the program word register (C) is increased by one. This action occurs when L goes from 3 to 0, thereby addressing the first syllable of the next word.

When the repeat count field equals zero, the J register is set to 2 for subsequent termination of the operator and a fetch is initiated by setting E16F to 1.

J = 2

The syllable execute complete level is unconditionally true to terminate the operator. The syllable execute complete level being true, with PROF in the reset state, will reset TROF (see SECL flow). Therefore, all actions await completion of the fetch access to load the P register with the program word addressed by the C register.

4.12 JUMP REVERSE UNCONDITIONAL JUMP REVERSE CONDITIONAL

REJL-XX57
CRJL-XX55

JUMP REVERSE UNCONDITIONAL

This operator initiates an unconditional reverse jump. The C and L registers are decreased by the contents of the repeat field. Prior to the subtraction of the repeat field of this operator, the C and L registers effectively contain the address of the next syllable in sequence.

JUMP REVERSE CONDITIONAL

This operator initiates a Jump Reverse Conditional if the true/false indicator is false. If the true/false indicator is true, the next syllable in sequence is fetched. The true/false indicator remains unchanged.

SUMMARY OF OPERATION

Both operators reverse jump a number of consecutive syllables in the program string. The number of syllables jumped is specified by the repeat count field and is limited to a maximum of 63. The reference point for counting is considered to be the syllable following the Jump operator syllable.

If the repeat count field is zero for both operators, or if the operator is Conditional Jump, with the true/false flip-flop in the set state, the operation is treated as a NO-OP and the next program syllable in sequence is executed.

If the repeat count field is not zero and the operator is unconditional jump or conditional jump with the true/false flip-flop in the reset state, the operator is executed as follows: The repeat count field receives an initial count down of one. If the repeat count field is greater than four (more than one word is to be jumped), the program word address register is decremented by one and the repeat count field by four as long as the repeat count field is greater than four.

If the repeat count field is less than four, or when it becomes less than four, the syllable pointer register and the repeat count field are decremented by one, as long as the repeat count field does not equal zero. If the syllable pointer register overflows, the program address register is decremented to address the next prior word in sequence.

When the repeat count field equals zero, a fetch is initiated to load the P register with the word now addressed by the C register.

DETAILED DESCRIPTION

J = 0

If the repeat count field is equal to zero (TEZL), or the operator is conditional jump, with the true/false flip-flop in the set state (CRJL • TFFF), the J register is set to 2 for subsequent termination of the operator.

If the repeat count field is not equal to zero (TEZL') and the operator is either a unconditional jump or a conditional jump, with the true/false flip-flop in the reset state, the J register is set to 1 and the inhibit count up for fetch level is true (ICFL' is false). Inhibit count up for fetch is required to prevent an undesired count of the L register. It also prevents PROF from being set to the one state at J = 0 of fetch. (See J = 0 of Fetch).

Because the reference for reverse jump is the syllable following the current syllable, the repeat count field is counted down by one (T-1) prior to counting down the L and C registers. This is required because the Inhibit Count Up for Fetch level prevented a normal count up of the L register. The L register was not allowed to count up because a fetch of a program word might have resulted if the L register equalled three.

J = 1

If the repeat count field is equal to or greater than four (TL4L'), indicating at least one word is to be skipped, the program word address register (C) is counted down by one and the repeat count field is counted down by four.

When the repeat count field is less than four but not equal to zero (TL4L • TEZL'), the syllable address register and the repeat count field are counted down by one. If, during this time, the L register overflows (LEZL is true when L is decremented), the C register is decremented to the prior word. When the repeat count field equals zero, the J register is set to 2. A fetch is initiated by setting E16F to 1 when the T register repeat count field is equal to one (TE1L).

J = 2

If the logical flip-flop Q02F is reset, a fetch is initiated by setting the E register to 16. The syllable execute complete level is unconditionally true to terminate the operator. With the syllable execute complete level true and because PROF is in the reset state, TROF will be reset (see SECL flow). This will inhibit all actions until the completion of the fetch access, which loads the P register with the program word now addressed by the C register.

4.13 BEGIN LOOP

BELL-XX52

This operator begins a string of program syllables which is to be repeated. The end of the string is identified by an End Loop Operator.

The number of times this string of syllables is to be repeated is specified by the repeat field of the Begin Loop Operator. If the repeat field is zero or one, the program string is executed once. Any repeated program string may contain within it repeated program strings.

SUMMARY OF OPERATION

The Begin Loop Operator is used to begin the execution of a group of program syllables enclosed between a Begin Loop Operator and an End Loop Operator. This operator generates a new Loop Control Word. The repeat count field placed in the Loop Control Word (from the T register) determines the number of times the loop will be executed. Prior to placing the repeat count field into the Loop Control Word to be developed, the repeat count field is decreased by one to account for the initial execution of the loop, which follows upon termination of this operator.

Upon entry to this operator, the X register will contain either:

1. An F register setting pointing to a Return Control Word, or
2. A Loop Control Word (minus I.D. bits) which in turn contains an F register setting pointing to either a prior Loop Control Word or a Return Control Word.

If the X register contains an F register setting, pointing to a Return Control Word, this Begin Loop Operator is initiating a primary loop, not a loop within a loop. If, however, the X register contains a Loop Control Word, this Begin Loop Operator is initiating a loop within a loop; i.e., a nested loop is being entered.

In any event, this operator will obtain the F register setting from the X register contents (the contents on entry to the operator) and subsequently store the entry contents of the X register, at the address specified by the just obtained F register setting plus one. The Loop Control Word generated by this operator is placed in the X register (the F register setting placed into this Loop Control Word is the address of the S register contents currently being stored in memory).

DETAILED DESCRIPTION

J = 0

The B register contents are temporarily stored in the A register (until the end of this operator) and the A register is marked unoccupied. At the termination of this operator, the A register contents are non-valid.

The X register contents are transferred to the B register. The X register contents may be either:

1. An F register address pointing at a Return Control Word, or
2. A Loop Control Word (minus the I.D. bits) which contain an F register address pointing at either a prior Loop Control Word or a Return Control Word.

The J register is set to one where the new Loop Control Word is generated.

If the repeat count field of the Begin Loop Operator in the T register is not equal to zero (TEZL'), the repeat count field is counted down by one to tally the initial execution of the loop, which occurs upon termination of this operator.

J = 1

A Loop Control Word is generated in the B register at this J register setting. The settings of the C and L registers are placed into the B register. Note that the combination of C and L points to the syllable following the Begin Loop syllable in the program string (L is incremented at J = 0 of Fetch). The repeat count field of the operator is also placed in the B register, and if the repeat count field were not initially equal to zero, the value transferred to the B register is one less than the initial value on entry. The F register setting of the new Loop Control Word being generated is already in the B register, having been placed there at J = 0 of the operator. The B register bits 48 and 47 are set while the 45th is reset, thereby marking this word as a Loop Control Word. The J register is set to 2, where the new Loop Control Word is transferred to the X register.

J = 2

The contents of the X register are transferred to the B register, with the low order 39 bits of the B register being transferred to the X register. Thus, the new Loop Control Word (minus the identification bits) will occupy the X register. The initial contents of the X register (on entry to this operator) will occupy the B register, with the identification bits marking the contents of a Loop Control Word. The J register is set to 3 where the address to store the B register contents is obtained.

J = 3

The current setting of the S register is temporarily stored in the X register, bit positions 16 through 30. It will be restored to the S register upon termination of the operator. Simultaneously, the F register setting occupying the X register bit positions 16 through 30 is transferred to the S register; this is the F register setting initially in the X register on entry to the operator. The J register is set to 4 to continue the operation.

J = 4

The S register is counted up by one and a memory access is initiated (E to 11) to store the B register contents at the address that is one above the preceding Loop Control Word or the Return Control Word. This is depending upon whether or not a nested loop is being entered. The J register is set to 5 to await completion of the store memory access.

J = 5

Upon completion of the store memory access (MWOF), the S register is restored to its initial setting on entry to this operator, by transferring the X register bits 16 through 30 to the S register. Simultaneously, the current setting of the S register (address of the B register contents just stored) is transferred to the X register bit positions 16 through 30. The initial B register contents are restored to the B register, by transferring A to B. The syllable execute complete level is true to terminate the operator.

4.14 END LOOP ENLL-XX51

This operator marks the end of a repeated program string. A test is made of the number of times the program string has been executed. If the number of repetitions is less than the repeat field of the Begin Loop operator that initiated the program string, the unconditional branch occurs to the syllable following the Begin Loop operator; otherwise, control continues in sequence.

SUMMARY OF OPERATION

The End Loop operator is utilized to check the repeat count field portion of the Loop Control Word in the X register, to determine whether the loop is to be executed again or whether the next syllable in sequence is to be executed.

If the repeat count field portion of the Loop Control Word is not equal to zero, the repeat count field portion of the Loop Control Word is decreased by one and the syllable following the Begin Loop operator is the next to be executed.

If the repeat count field portion of the Loop Control portion is equal to zero, the preceding Loop Control Word or the Return Control Word is read out of memory and placed in the X register. In this case, the syllable following the End Loop operator is the next to be executed.

DETAILED DESCRIPTION

J = 0

The B register contents are temporarily stored in the A register (until the end of the operator) and the A register is marked unoccupied. The A register contents are non-valid at termination of the operator.

The Loop Control Word (minus the I.D. bits) in the X register is transferred to the B register. This Loop Control Word contains an F register setting which points to either a prior Loop Control Word or a Return Control Word.

Fetch is inhibited at this time because the P register may require reloading, if the number of specified loops have been executed. The J register is set to 1 to continue the operation.

J = 1

The repeat count field of the Loop Control Word now in the B register is transferred to the repeat count field portion of the T register. The J register is set to 2.

J = 2

If the repeat count field is not equal to zero (TEZL'), the loop must be executed again. Hence, the C and L fields of the Loop Control Word are

shifted to the C and L registers. The C and L registers will now point at the syllable following the Begin Loop Operator. The E register is set to 16, in order to load the P register with the word addressed by the C register. The T register is decremented by one, to tally the loop that will subsequently be executed. The J register is set to 3.

If the repeat count field is equal to zero (TEZL), the X register bit positions 30 through 16 are interchanged with the S register contents. The S register contents are placed in the X register for temporary storage. The S register receives the address of the prior Loop Control Word, or if none exist, the address of the Return Control Word. The E register is set to 3, thereby initiating a memory access to load the B register with the prior Loop Control Word, or the Return Control Word, depending on which one is accessed. In this case, the syllable following the End Loop operator is the next to be executed. Therefore, Fetch is initiated, which causes the automatic count up for fetch level to go true (ACFL; see Fetch flow). The J register is set to 5, where the S register has its contents restored.

J = 3

When the J register equals three, the repeat count field is returned to the Loop Control Word in the B register. The repeat count field. being returned to the Loop Control Word, is now one less than its value on entry to this operator. The J register is set to 4.

J = 4

At this J register setting, the Loop Control Word is returned to the X register prior to the execution of the loop again, if additional executions of the loop are required. If, however, the required number of loops have already been executed, either a prior Loop Control Word or a Return Control Word is transferred to the X register (depending on whether a nested loop is being terminated). The B register contents are unconditionally restored from temporary storage in the A register. The syllable execute complete level is true to terminate the operator.

J = 5

At this J register setting, upon completion of the memory access to load the B register with either a prior Loop Control Word or a Return Control Word (MROF), the S register is restored to its value on entry to this operator. The J register is set to 4, where the Loop Control Word or Return Control Word is placed in the X register.

4.15 JUMP OUT OF LOOP
JUMP OUT OF LOOP CONDITIONAL

JOLL-XX46
CJOL-XX44

JUMP OUT OF LOOP

This operator is used for jumping out of a repeated program string and terminates the repetition of the program string. The operator causes an unconditional forward jump over the number of syllables specified by the repeat count field and deletes the count of the repetitions associated with the program string.

JUMP OUT OF LOOP CONDITIONAL

If the true/false indicator is set to false, a Jump Out of Loop Operator is performed. If the true/false indicator is set to true, control continues in sequence.

SUMMARY OF OPERATION

If the operator is Jump Out of Loop or Jump Out of Loop Conditional, with the true/false flip-flop in the reset state, the F register field of the Loop Control Word in the X register is utilized to obtain the prior Loop Control Word (if any exists) or the Return Control Word. The Loop Control Word or Return Control Word (depending on which is accessed) is placed in the X register. If the repeat count field of the operator in the T register is not equal to zero, the C and L registers are advanced by the amount specified by the repeat count field. Subsequently, a fetch is initiated to load the P register with the program word addressed by the new setting of the C register. If, however, the repeat count field of the operator in the T register is equal to zero, the operator is terminated and the next syllable in sequence is executed.

If the operator is Jump Out of Loop with the true/false flip-flop in the set state, the operator terminates and the next syllable in sequence is executed.

DETAILED DESCRIPTION

J = 0

JOLL + CJOL • TFFF'

If the operator is Jump Out of Loop or Jump Out of Loop Conditional with the true/false flip-flop in the reset state (JOLL + CJOL • TFFF'), the S register contents are placed in temporary storage in the X register. Simultaneously, the F register setting, from the Loop Control Word in the X register, is transferred to the S register. A memory access is initiated by setting the E register to 2, for loading the A register with the prior Loop Control Word or, if none exists, the word accessed will be a Return Control Word. The A register is marked unoccupied as its contents are non-valid upon termination of this operator.

TEZL (JOLL + CJOL • TFFF')

If, in addition to the above stated logic, the repeat count field of the operator in the T register is not equal to zero (TEZL'), Fetch is inhibited to prevent a count up of the L register (see Fetch flow). The J register is set to 1.

TEZL + CJOL • TFFF

If the operator is Jump Out of Loop Conditional or if the repeat count field of the operator in the T register is equal to zero, then the J register is set to 2.

J = 1

At this J register setting, the address of the next syllable to be executed is determined by counting up the C and L registers by the value of the repeat count field of the operator in the T register. If the repeat count field is greater than four (TL4L'), the C register is incremented by one and the repeat count field decremented by four.
When the repeat count field is less than four (TL4L), incrementing of the C register ceases and incrementing of the L register commences. The L register is counted down by one, with an equivalent count down of the T register repeat count field (L + 1 and T-1). If the L register is equal to three (LE3L), when it receives an increment of its value, the C register is also counted up by one to address the next word in sequence. When the T register repeat count field equals one (TE1L), the L register receives its last increment, the J register is set to 2 and E16F is set to 1. (TL4L is true when TEZL is true).

J = 2

Upon completion of the load A register memory access, initiated at J = 0 (EEZL), the Loop Control Word or Return Control Word, depending on which was accessed, is transferred to the X register. Simultaneously, the S register contents are restored from temporary storage in the A register. Note that this action only occurs if the operator is Jump Out of Loop or Jump Out of Loop with the true/false flip-flop in the reset state, in which case a load of the A register would have occurred. If the operator is Jump Out of Loop Conditional, with the true/false flip-flop in the set state, the load A register memory access would not have been initiated at J = 0. In either case, the term EEZL gates the syllable execute complete level true to terminate the operator.

4.16 SKIP FORWARD SOURCE
SKIP REVERSE SOURCE

FSSL-XX31
RSSL-XX30

SKIP FORWARD SOURCE

The contents of the M and G registers are increased by the contents of the repeat field.

SKIP REVERSE SOURCE

The contents of the M and G registers are decreased by the contents of the repeat field.

SUMMARY OF OPERATION

For either operator, if the bit pointer for the source string (H register) is not equal to zero on entry to the operator, the character pointer (G register) is increased by one. If this results in an overflow of the character pointer, the source string word address register (M) is advanced to the next word. The bit pointer is unconditionally cleared to zero.

When the bit pointer is equal to zero, the repeat count field is checked for a zero content. If equal to zero, the operator terminates. If not equal to zero, the M register is either incremented or decremented (depending on the operator) with the repeat count field being counted down by eight, as long as the repeat count field is equal to or greater than eight. When the repeat count field is less than eight, the G register is either incremented or decremented (depending on the operator) with the repeat count field being counted down by one. When the repeat count field equals zero, the G register receives its last count and the operator terminates.

If the skipping operation causes the M and G registers to point at a character position in a word, other than the word initially specified by the M register, the A register is marked unoccupied.

DETAILED DESCRIPTION

J = 0

Unconditionally the bit pointer for the source string is cleared to zero. If the bit pointer is not initially equal to zero, the G register is increased by one. If this results in an overflow of the G register, the M register is advanced to the next source string word and the A register is marked empty.

If the repeat count field is equal to zero on entry to the operator, the syllable execute complete level is true to terminate the operator. If the repeat count field is not equal to zero, exit awaits incrementing or decrementing of the M and G registers.

TEZL' •,HEZL

If the repeat count field of the operator in the T register is equal to or greater than eight (TL8L'), the T register repeat count field is counted down by eight (T-8). In conjunction with the decrement by eight of the T register, the M register is increased by one, if the operator is Skip Forward (FSSL) or decreased by one if the operator is Skip Reverse (RSSL).

If/or when the repeat count field of the operator in the T register is less than eight (TL8L), the T register repeat count field is counted down by one (T-1). At the same time, the G register is incremented by one, if the operator is Skip Forward, or decremented by one, if the operator is Skip Reverse. If the G register is equal to seven when the G register is incremented, during Skip Forward, the M register is counted up by one. If the G register is equal to zero when the G register is decremented, during Skip Reverse, the M register is counted down by one.

RSSL •GEZL + FSSL X GE7L + TL8L'

If the operator is Skip Reverse and the G register is pointing at the most significant character position of a word (RSSL • GEZL), the A register is marked unoccupied. If the operator is Skip Forward and the G register is pointing at the least significant character of a word (FSSL • GE7L), the A register is marked unoccupied. If the repeat count field is equal to or greater than eight, unconditionally mark the A register unoccupied. In this case, either operator results in the character pointer overflowing into another word.

4.17 SKIP FORWARD DESTINATION FSDL-XX16

The contents of the S and K registers are increased by the contents of the repeat field.

Prior to the execution of the operator the V register is tested for zero. If the V register is not equal to zero, it is set to zero and the K register is increased by one. Overflow into the S register can occur.

SUMMARY OF OPERATION

If the destination string bit pointer is not pointing at a full character, the V register is cleared to zero and the character pointer (K) is advanced by one. If the B register contains valid information when the character pointer is incremented, the B register must be shifted one character position to maintain alignment of the K and N registers. If at any time during this operator, the character pointer overflows when the B register contains valid information, the B register contents must be stored in memory. When the character pointer overflows, the destination string word address register is counted up by one. If the B register contains valid information at this time, the increment of the S register awaits the storage of the B register contents.

If, initially, the B register contains valid information, the character pointer is incremented by single character positions even though the repeat count field may be greater than eight. Once the B register contents are stored in memory, the increment is by word until the repeat count field is less than eight. Thereafter, incrementing is by character. When the repeat count field equals zero, the operator terminates.

DETAILED DESCRIPTION

J = 0

The bit pointer for the destination string (V) is unconditionally cleared to zero. Once the bit pointer is cleared to zero, or if initially equal to zero, and the repeat count field is equal to zero (TEZL • VEZL), the syllable execute complete level is true to terminate the operator. The operator is terminated in the same manner, after counting the repeat count field down to zero during the normal execution of the operator.

K + 1

If the bit pointer is initially not equal to zero (VEZL'), the character address is counted plus one (K + 1).

If the repeat count field is not equal to zero (TEZL'), and the number of characters to be skipped is less than eight (TL8L) or the B register contains valid information (BROF), the character pointer is counted by one. When the B register contains valid information (a portion of the destination string) its contents will require storage, if more than eight character positions are to be skipped. Skipping by characters is required until the B register contents are stored in memory.

VEZL' + TEZL'

If the B register contains valid information and is in alignment (KENL • BROF), the B register must be shifted left and circulated, in order to maintain alignment when the character pointer is incremented; as indicated by the logic of VEZL' + TEZL' being true. The N register is incremented to tally the octade shifts of the B register. In this case, the first of two required octade shifts (one character shift) occurs at this J register setting, the second occurs at J = 6.

BROF to 0

The B register is unconditionally marked unoccupied when the character pointer points at the most significant character position of a word (KE7L). If at this time the B register contents are valid, the reset of BROF permits the storage of the B register contents at J = 6; otherwise, the reset of BROF is unnecessary,but is uninhibited.

S + 1

The destination string word address register is counted up by one when either of the following logics is true: 1. Once the bit pointer equals zero, if the repeat count field is greater than eight and provided the B register does not contain valid information as indicated by VEZL • BROF' TL8L', or 2. If the B register is in alignment (restored) as indicated by NEZL and the character address pointer equals seven (KE7L) when an increment of the character pointer occurs as indicated by TEZL' + VEZL'.

In the latter case, the B register is in a restored status (NEZL), which indicates that the B register does not require storage, even if the B register is initially occupied on entry to the operator.

T - 1

With the bit pointer equal to zero, and the repeat count field not equal to zero (VEZL • TEZL'), the repeat count field is counted down by one if either the repeat count field is less than eight or the B register contains valid information (TL8L + BROF). This action occurs simultaneously with the increase of the character pointer skipping the character.

T - 8

Once the B register is marked unoccupied (either on entry or after storage) and with the bit pointer equal to zero, the repeat count field is counted down by eight, as long as the repeat count field is greater than eight (BROF' • TL8L' • VEZL). This occurs in conjunction with skipping by words.

J = 6

Entry to J = 6 only occurs if the B register contains valid information and the B register required shifting to maintain alignment with the K register. On entry to this J register setting, the level EWZL gates the second of two octade shifts to complete a full character shift. If at

this same time the B register requires storage of its contents (BROF' • EWZL), a memory access is initiated to store the B register contents in the destination string at the address specified by the S register. If storage of the B register contents occurs, at the completion of the storage access (MWOF), the destination string word address register is counted up by one (S + 1).

At the completion of the store memory access or if none were required (B register still contains valid information) as indicated by BROF + MWOF, the J register is set to zero to continue the skipping operation.

4.18 SKIP REVERSE DESTINATION RSDL-XX17

The contents of the S and K registers are decreased by the contents of the repeat field.

Prior to the execution of the operator, the V register is tested for zero. If the V register is not equal to zero, it is set to zero and the K register is increased by one. Overflow into the S register can occur.

SUMMARY OF OPERATION

A maximum of 63 character positions can be skipped in the destination string. If, initially, the bit pointer does not point at a full character position, the bit pointer is reset to zero and the character pointer advanced to the next character position. If advancement of the character pointer is into the next higher word, the B register contents must be stored, if it is valid information and is not initially in a restored status.

Once the bit pointer equals zero, the skip reverse operation commences. If the B register contains valid information, alignment of the K and N registers is maintained while the character pointer is counted down. If, or when, the character pointer overflows, while counting the character pointer down, the B register contents must be stored back into the destination string (if valid) prior to proceeding any further with the operation. Once the B register is marked unoccupied, the skipping may be by words, if the repeat count field is greater than eight. When the repeat count field is less than eight, skipping is by character position.

DETAILED DESCRIPTION

J = 0

The bit pointer for the destination string is unconditionally cleared to zero. Once the bit pointer is cleared to zero, or if initially equal to zero, and the repeat count field equals zero (VEZL • TEZL), the syllable execute complete level is true to terminate the operator. The operator is terminated in the same manner, after counting the repeat count field down to zero, during the normal execution of the operation.

VEZL'

If the bit pointer for the destination string is not initially equal to zero (VEZL'), the character pointer is counted up by one (K + 1) to point at the next character in sequence; the skip operation starts with a full character position. If, at this clock pulse time, the B register is loaded and is in alignment (KENL • BROF), the increment of the K register will cause misalignment of K versus N. To maintain alignment, the B register is shifted left and circulated two octade positions with a corresponding increment of the N register. The first shift occurs at this clock pulse time and the second at J = 6.

If the character pointer for the destination string is pointing at the last character of a word (KE7L), at the time the character pointer is incremented, the B register is marked unoccupied to indicate that storage of the B register contents is required at J = 6. However, if KENL is true when KE7L is true, the B register is not in the proper configuration for storage in memory and the alignment procedure described above serves to place the B register in the proper form for storage in the destination string.

If the character pointer is pointing at the last character position of a word, and the N register is equal to zero (KE7L • NEZL), at the time the character pointer is incremented via the VEZL' logic, the B register contents do not require storage. In this case, the only required action is to count the character pointer up by one to address the starting character position for this operation.

VEZL' • TEZL'

The actions between the double lines which are gated by VEZL and TEZL' are described in a logical sequence. The required actions at this time are to count down the character pointer and, if required, the word address register. Normally, if the repeat count field becomes less than eight, the count down is by character position. However, if initially the B register contains valid information and the count down is such to cause an underflow of the character pointer (KEZL to KE7L of the prior word), the B register contents must be stored into the destination string. Because of this, the B register must be shifted and circulated as the character pointer is counted down to preserve the alignment as indicated by KENL. Once the B register contains non-valid information, decrementing is by words as long as the repeat count field is greater than eight. When the repeat count field is less than eight, the decrement is again by character position.

KENL'. If the B register is not in alignment, as indicated by KENL', the B register is marked unoccupied. In this case, the information in the B register was stored by a prior operator and need not be stored again.

BROF' • TL8L + BROF • KENL • NO1F'. If either of these logics exist, the character pointer and repeat count field are counted down by one. If the B register information is non-valid and T is less than eight (BROF' • TL8L), count down must be by single character positions. If the B register contains valid information, that has not been stored as yet (BROF • KENL • NO1F'), count down must be by character position. Note that the count down only occurs when the N register indicates a complete character position as specified by NO1F'. This is to prevent a count down during both of the two required octade shifts of the B register.

BROF' • (KEZL + TL8L'). The S register is counted down by one to address the prior word in the destination string, when the B register contents do not require storage and either the character pointer is pointing at the first character of a word, or the repeat count field is greater than eight. The level KEZL provides for counting down the S register when the count down is character; i.e., the repeat count field is less than eight.

BROF • KENL • NEZL'. If the B register contains valid information and it is in alignment with the K register, the B register is shifted to the right and circulated, providing the N register does not equal zero. When the N register equals zero, the B register contents will be stored in memory and must not be shifted at that time. The N register is counted down by one to tally the right shifts of the B register.

BORF' • TL8L'. When the B register does not contain valid information and the repeat count field is greater than eight, the repeat count field in the T register is decremented by eight to tally skipping by words.

BROF • KENL • NO1F' • KEZL. This logic initiates the store memory access (E to 11), for storage of the B register contents. For proper storage of the word in the B register, the N register must be equal to zero. The term KEZL indirectly indicates that N is equal to zero. NO1F' insures a full character shift of the B register prior to storage.

VEZL

Fetch is inhibited (ICFL' is false) if the bit pointer is not equal to zero. It may be required to store the B register contents, therefore, in order to prevent any delay of the operator execution, Fetch must be inhibited.

J = 1

This J register setting is only attained when the B register contents require storage. At the completion of the store memory cycle (MWOF) the S register is counted down to address the prior word of the destination string. The B register is marked unoccupied and the J register is returned to J = 0.

J = 6

This J register setting is only attained if, on entry to the operator, the bit pointer did not equal zero (VEZL') and the B register was occupied and in alignment with the K register (BROF • KENL). If, at that time the K register did not equal seven, the B register is brought back into alignment via the second of two octade shifts (the K register was incremented at J = 0 to misalign the B register relative to the K register). In this case, the B register continues to have valid information, BROF remains set and gates a return to J = 0.

If, however, the K register did equal seven when KENL was true, BROF would have been reset, indicating that the B register contents are to be stored in memory. In this case, the K register was advanced from seven to zero (at J = 0) and the B register, therefore, requires two octade shifts prior to its storage in memory. On entry to J = 6 the level EWZL gates the second octade left shift and circulates, plus initiating the store memory access (BROF' • EWZL). Upon completion of the store memory access (MWOF), the S register is counted up by one and the J register is returned to zero to continue the operation.

4.19 SKIP BIT SOURCE SBSL-XX03

The contents of the M, G and H registers are increased by the contents of the repeat field.

SUMMARY OF OPERATION

This operator skips bits from the source string. The number of bits that may be skipped is limited to a maximum of 63 (decimal).

Upon entry to the operator, the repeat count field of the operator, in the T register, is checked and if initially equal to either zero or one, the operation terminates. At termination of the operator, if the repeat count field is equal to one, the bit pointer is advanced by one. If the bit pointer overflows (5 to 0), the character pointer is incremented. If the character pointer also overflows, the word address register is incremented.

If, initially, the repeat count field is greater than one, the bit pointer is incremented with a corresponding decrement of the repeat count field. When the bit pointer overflows into the next character position, the character pointer is advanced by one. If the character pointer overflows when the bit pointer overflows, the word address register is advanced by one with the A register being marked in an unoccupied status. The last bit increment occurs with the repeat count field equal to one and the operator terminates at the same clock pulse time.

DETAILED DESCRIPTION

J = 0

Upon entry to the operator, if the repeat count field is equal to either zero or one (TE1L), the syllable execute complete level is true to terminate the operator. The T register repeat count field equal to one level (TE1L), is equivalent to the repeat count field being equal to either zero or one (TE1L + TEZL + TE1L).

$$TE1L = T12F' \cdot T11F' \cdot T10F' \cdot T09F' \cdot T08F'$$

If the repeat count field is greater than one (TE1L'), the bit pointer for the source string is counted up by one (H + 1), with a corresponding count down of the repeat count field (T-1), to tally the increment of the bit pointer Note that the bit pointer receives its last increment when the repeat count field equals one; the count up of the H register is gated by TEZL'.

If the bit pointer is equal to 5 when it is counted up, it will overflow into the next character position (HE5L to HE0L). At that time, the character pointer is also counted up by one (G + 1).

If both the character pointer and bit pointer overflow, when the bit pointer is incremented (GE7L • HE5L), the source string word address register is advanced by one (M + 1) with the A register being marked as unoccupied.

4.20 SKIP BIT DESTINATION

SBDL-XX02

The contents of the S, K and V registers are increased by the contents of the repeat field.

SUMMARY OF OPERATION

This operator skips bits from the destination string. The number of bits that may be skipped is limited to a maximum of 63.

If the repeat count field is equal to zero, the operation terminates. If the repeat count field is not zero, the bit pointer (V register) is counted up by one, while the repeat count field is counted down by one. If the bit pointer is counted through a complete character, the character pointer (K register) is advanced by one. If the character pointer is advanced through a complete word, the word is stored in memory and the destination string word address register (S) is increased by one. The operation continues until the repeat count field is counted down to zero, at which time the operator terminates.

DETAILED DESCRIPTION

J = 0

If the repeat count field of the operator in the T register is equal to zero (TEZL), the syllable execute complete level is true to terminate the operator.

If the repeat count field is not equal to zero (TEZL'), the bit pointer for the destination string is counted up by one (V+1) while the repeat count field is counted down by one (T-1), to tally the increment of the V register.

If the bit pointer is equal to five (VE5L) when the bit pointer is incremented, the character pointer is counted up by one (K+1), to point at the next more significant character position of the destination string.

If the bit pointer is pointing at the last bit of a character (equal to five) and the N register is not equal to zero (VE5L • NEZL'), the B register is shifted left and circulated one octade position. The N register is incremented to tally the left shift of the B register. The logical flip-flop, Q08F, is set to one with the J register being set to 1. The logical flip-flop, Q08F, is set in order to gate the second of two octade shifts (at J=1) to make one complete character shift. Note that if the N register is equal to zero, it is not necessary to shift and circulate the B register (it is already in the proper form for storage). Once the B register is stored, the N register will equal zero and the shifting of the B register is not allowed to occur.

If the character pointer is at the last character position of a word and the bit pointer is at the last bit position of a character (KE7L • VE5L), either of two actions may occur. If the B register is unoccupied (already stored), it is only necessary to increase the destination string word address register. If the B register is still occupied, the B register contents must be stored in the destination string. The J register is set to 1 where a memory access is initiated to store the B register contents.

J = 1

If the logical flip-flop, Q08F, is set, the B register receives the second of two octade shifts to complete a complete character shift. The N register is counted up by one to tally the left shift and circulate of the B register.

If entry to J = 1 occurs with KE7L true, the character pointer overflowed at J = 0. The B register contents require storage. A memory access is initiated to store the B register contents in the destination string. The B register is marked unoccupied and the logical flip-flop, Q08F, is reset as its function is terminated. The J register is set to 2 to await completion of the memory access so the destination string word address register may be incremented.

If entry to J=1 occurs with KEZL' true, two things are known about the B register contents:

1. The B register contents have not as yet been stored;
2. The B register contents are not in proper form for storage (is not restored).

If at this time the repeat count field is not equal to zero, the operation continues by returning to J=0. If the repeat count field is equal to zero, the syllable execute complete level is true to terminate the operator. In this case (KE7L' • TEZL), the operator terminates without storing the B register contents. The B register contents are only stored if sufficient increments of the bit pointer are specified to cause an overflow of the character and word address registers.

J = 2

Upon completion of the memory access to store the B register contents in the destination string (MWOF), the destination string word address register is counted up by one to address the next word in sequence. If the repeat count field is equal to zero at this time, the operator terminates; otherwise, the J register is returned to zero to continue the operation.

4.21 STORE SOURCE ADDRESS STSL-XX15

The contents of the M and G registers are stored in the word at the address formed by reducing the address of the return control word by the repeat field. The contents of the M register are stored in bit positions 15 through 1 and the contents of the G register in bit positions 18 through 16 of the word address. The flag bit of the word address is set to zero. The address of the return control word remains unchanged.

Prior to the execution of the operator, the H register is tested for zero. If the H register is not equal to zero, it is set to zero and the G register is increased by one. Overflow into the M register can occur.

SUMMARY OF OPERATION

This operator stores the address of a character (as specified by the M and G registers) in a specific location, whose address is formed by reducing the address of the return control word by the repeat count field.

The contents of the M and G registers are placed in the B register (least significant bit positions) with the most significant bit positions being cleared to zero. The return control word is obtained from the F register and placed in the M register. The repeat count field is utilized to tally the count down of the M register. When the repeat count field equals one, the last decrement of the M register occurs and a memory access is initiated to store the word. The address of the return control word is then restored to the F register.

DETAILED DESCRIPTION

J = 0

The bit pointer for the source string (H register) is unconditionally cleared to zero. If the H register is not initially equal to zero, the character pointer is advanced by one (G+1). If at this time, the character pointer equals seven, the source string word address register is increased by one (M+1).

When the bit pointer equals zero (HEZL), the B register contents are placed in temporary storage in the A register and restored at termination of the operator. The B register bit positions 19 through 48 are cleared to zero to insure the flag bit being equal to zero. The address of the source string character pointed to, by the M and G register, is transferred to the B register (bit positions 1 through 18). The address of the return control word is obtained from the F register and placed in the M register. The J register is set to 1, where the return control word address in the M register is decremented.

J = 1

If the repeat count field is not equal to zero (TEZL'), both the M register and the T register repeat count field are counted down by one, until the repeat count field equals one (TE1L). When the repeat count field equals one, the M register is decremented for its last count down and a memory access is initiated to store the B register contents at the address specified by the M register. The J register is set to 2 for subsequent termination of the operator.

J = 2

Upon completion of the memory access (MWOF), the B register contents are stored from temporary storage in the A register. The initial source string word address setting is restored in the M register from the B register bit positions 1 through 15. The syllable execute complete level is true to terminate the operator. The A register is marked unoccupied as its contents are no longer valid.

4.22 STORE DESTINATION ADDRESS STDL-XX14

The contents of the S and K registers are stored in the word at the address formed by reducing the address of the return control word by the repeat field. The contents of the S register are stored in bit positions 15 through 1 and the contents of the K register bit positions 18 through 16 of the word address. The flag bit of the word address is set to zero. The address of the return control word remains unchanged.

Prior to the execution of the operator, the V register is tested for zero. If the V register is not equal to zero, it is set to zero and the K register increased by one. Overflow into the S register can occur.

SUMMARY OF OPERATION

This operator stores the address of a character (as specified by the S and K registers) in a specific location, whose address is formed by reducing the address of the return control word by the repeat count field.

If, on entry to the operator, the bit pointer for the destination string does not equal zero, the character pointer is increased by one. If the B register is in alignment relative to the K register; KENL is true, the B register must be shifted to maintain the alignment of K versus N. If the character pointer overflows and the B register contents are in alignment, the B register contents must be stored im memory and the destination string word address register increased by one. If the character pointer overflows, and the B register does not contain valid information, it is only necessary to increase the destination string word address register by one.

If the bit pointer for the destination string is initially equal to zero, none of the aforementioned is required. The operation proceeds when the bit pointer equals zero. The contents of the S and K registers are placed in the B register (least significant bit positions) with the most significant bit positions being cleared to zero. The repeat count field is utilized to tally the count down of the S register. When the repeat count field equals one, the last decrement of the S register occurs and a memory access is initiated to store the word just developed in the B register. The address of the return control word is then returned to the S register.

DETAILED DESCRIPTION

J = 0

The bit pointer for the destination string is unconditionally cleared to zero (V register to 0).

VEZL

If the V register is initially equal to zero, or when cleared to zero, the B register contents are placed in temporary storage in the A register. The B register, bit positions 19 through 48, is cleared to zero to insure that the flag bit is reset. The contents of the S and K registers are transferred

to the B register, bit positions 1 through 18. The address of the return control word in the F register is transferred to the S register. The J register is set to 1, where the return control word address in the S register is counted down.

VEZL'

If the bit pointer is not initially equal to zero (VEZL'), the character pointer is increased by one (K+1). If, at this time, the B register contains valid information and is in alignment relative to the K register (BROF • KENL), the B register is shifted left and circulated one octade position. This is the first of two required octade shifts to maintain alignment of K versus N. The N register is incremented to tally the shift and the J register is set to 6, for the second required shift and circulate of the B register. If the character pointer is specifying the least significant position of the word at this time (KE7L), the B register is marked unoccupied in order to gate a store of the B register contents at J = 6.

If and when the character pointer is incremented, the character pointer overflows and the B register is in a restored status (KE7L • NEZL), it is not required to store the B register contents. In this case, the destination string word address register is increased by one and the operation continues as with VEZL.

J = 1

If a memory access were initiated to store the initial B register contents, all actions at this J register setting await completion of the access, if still in progress at the entry to J = 1. As long as the repeat count field is not equal to one (TE1L'), both the S register and the repeat count field are counted down by one. When the repeat count field equals one, the S register receives its last count down (gated by TEZL') and a memory access is initiated to store the B register contents at the address specified by the S register. The J register is set to 2 for the subsequent termination of the operator.

J = 2

Upon completion of the memory access (MWOF), the B register contents are restored for temporary storage in the A register. The initial destination string word address register setting is restored to the S register from the B register bit positions 1 through 15. The A register is marked unoccupied (as its contents are no longer valid) and the syllable execute complete level is true to terminate the operator.

J = 6

This J register setting is only attained if, on entry to the operator, the bit pointer was not equal to zero and in addition the B register was occupied and in alignment relative to the K register (KENL • BROF). The second of two required octade shifts occurs at this time to complete a full character shift. If at J = 0, the K register did not equal seven, BROF would still be set and gate a return to J = 0. If, however, the K register

did equal seven at J = 0, BROF would be reset and gate a memory access to store the B register contents in the destination string. At the completion of the memory access, MROF gates the increment of the S register and sets the J register to zero, to continue the operation as with VEZL. Note that EWZL only allows a single shift and circulate of the B register. (EWZL = MWOF' • EEZL)

4.23 STORE CONTROL ADDRESS STPL-XX54

The contents of the C and L registers are stored in the word formed by reducing the address of the return control word by the repeat field. The contents of the C register are stored in bit positions 15 through 1 and the contents of the L register in bit positions 38 and 37 of the word address. The flag bit of the word address is set to zero. The address of the return control word remains unchanged.

SUMMARY OF OPERATION

This operator stores the address of the next syllable in sequence, at the address formed by reducing the address of the return control word by the repeat count field.

The return control word address is obtained from the F register and placed in both the S register and the B register. The transfer to the B register is for temporary storage. At the same time, the current S register setting is placed in the F register for temporary storage. Thereafter, the S register is reduced by the value of the repeat count field to form the storage address. The contents of the C and L registers are transferred to the B register. Note that the L register is incremented, during J = 0 of Fetch, to address the syllable following this operator. If the L register overflowed, the C register is incremented during the Fetch operation. A memory access is initiated to store the B register contents. The return control word address is returned to the F register. The initial S register setting is restored into the S register.

This operator is usually followed by a jump-type operator. The subroutine, to which control changes, is ended with a Recall Control Address Operator (RPAL). The Recall Control Address Operator increases the L register contents by one, so that the syllable following the aforementioned jump-type operator can be executed.

DETAILED DESCRIPTION

J = 0

The B register contents are transferred to the A register for temporary storage. The 48th bit of the B register is reset, to mark the word in the B register as an operand. The return control word in the F register is transferred to both the B register, bit positions 16 through 30, and the S register. Simultaneously, the current S register setting is transferred to the F register for temporary storage. The J register is set to one, where the return control word address in the S register is counted down.

J = 1

If the repeat count field is not equal to zero (TEZL'), the S register and the repeat count field of the T register are counted down by one. When the repeat count field equals one (TE1L), the S register receives its last decrement and a memory access is initiated to store the B register contents at the address specified by the S register. Simultaneously, the contents of the C and L registers are transferred to the B register. At this time the C and L registers point at the syllable following this operator. The J register is set to 2 for restoration of the registers.

J = 2

At the completion of the memory access (MWOF), the address of the return control word is restored to the F register. The B register contents are restored, from temporary storage in the A register, and the S register is restored from the F register. The syllable execute complete level is true to terminate the operator. The A register is marked unoccupied, as its contents are no longer valid.

4.24 RECALL SOURCE ADDRESS RSAL-XX53

The word at the address formed by reducing the address of the return control word by the repeat count field is read from memory. If the flag bit of the word is a one and the presence bit is zero, the presence bit is set in the interrupt register and the operation terminated.

If the flag bit and the presence bit of the word are both one, the bit positions 15 through 1 of the word are transferred to the M register and the G and H registers are cleared to zero. If the flag bit is zero, bit positions 15 through 1 are transferred to the M register, bit positions 18 through 16 of the word are transferred to the G register and the H register set to zero.

SUMMARY OF OPERATION

The contents of the B register are temporarily stored in the A register. The address of the return control word, in the F register, is transferred to the M register. The A register is marked unoccupied. The H and G registers are cleared to zero, the H register to point at a full character position, and the G register in preparation of receiving a new setting; otherwise, the G register points at the zero character position of a word. If the repeat count field is not zero, the M register is counted down until the T register goes to zero. When the repeat count field equals one, a memory access is initiated to load the B register from the address specified by the M register.

The M register is set from the 15 low order bit positions of the word brought to the B register. If the word brought to the B register is an operand, the G register is set from the G field in the B register. If the word is a descriptor and the presence bit is reset, a presence bit interrupt is set. The B registers initial contents are restored from the A register and the operator is terminated.

DETAILED DESCRIPTION

J = 0

The B register contents are placed in temporary storage in the A register. The return control word address in the F register is transferred to the M register. The H and G registers are cleared, with the A register being marked unoccupied. The J register is set to 1.

J = 1

If the repeat count field is not equal to zero (TEZL'). the M register and the repeat count field are counted down by one. When the repeat count field equals one (TE1L), the M register receives its final count and a memory access is initiated to load the B register with the word addressed by the M register. The J register is set to 2.

J = 2

Upon completion of the load B memory access (EEZL), the 15 low order bits of the word brought to the B register are transferred to the M register. If the word brought to the B register is an operand (B48F1), the G register is a descriptor with the presence bit reset (B48F • B46F') and a presence bit interrupt is set into the interrupt register. The initial B register contents are restored to the B register from the A register and the J register is set to 15.

J = 15

The syllable execute complete level is true to terminate the operator.

4.25 RECALL DESTINATION ADDRESS

RDAL-XX04

The word, at the address formed by reducing the address of the return control word by the repeat field, is read from memory. If the flag bit of the word is a one and the presence bit is a zero, the presence bit is set in the interrupt register and the operation terminated.

If the flag bit and the presence bit of the word are both one, bit positions 15 through 1 of the word are transferred to the S register and the K and V registers are set to zero.

If the flag bit of the word is a zero, bit positions 15 through 1 of the word are transferred to the S register. Bit positions 18 through 16 are transferred to the K register and the V register is set to zero.

The address of the return control word remains unchanged.

SUMMARY OF OPERATION

If, on entry to the operator, the B register is occupied, its contents must be stored in the destination string. If the B register contents are not in the proper configuration for storage, the B register must be restored prior to storage of its contents.

Once the B register is unoccupied, the address of the return control word in the F register is transferred to the S register. The S register value is then counted down by the amount specified by the repeat count field. When the repeat count field equals one, the S register receives its final count and a memory access is initiated to load the B register with the word specified by the S register.

From the word brought to the B register, the S register is set from the 15 low order bit positions of the B register. If the word is an operand, the K register is set from the word; otherwise, the K register equals zero. If the word is a descriptor, with the presence bit reset, a presence bit interrupt is set. At this point, the operation is terminated.

DETAILED DESCRIPTION

$J = 0$

If the N register is not equal to zero (NEZL'), indicating the B register is in an unrestored status, the 8th bit of the N register is interrogated. If NO8F is set, the B register is shifted left and circulated with a corresponding count up of the N register. If NO8F is reset, the B register is shifted right and circulated with a corresponding count down of the N register. When the N register equals zero, and provided the B register is occupied (NEZL • BROF), a memory access is initiated to store the B register contents.

The N register equal to zero level also gates the J register to 1. If the operator is entered with the B register occupied, the return control word address in the F register is transferred to the S register. If the B register is initially occupied, the return control word is not transferred to the S register until J = 1.

J = 1

Upon completion of the memory access to store the B register contents, the B register is marked unoccupied and the return control word address is transferred to the S register. If no memory access were initiated at J = 0, the S register already contains the return control word address. The clock pulse following completion of the memory access (EWZL) commences the S register decrement. The S and T registers are decremented as long as the repeat count field is not equal to one (TE1L'). When the repeat count field equals one, the S register receives its final count down and a memory access is initiated to load the B register with the word currently addressed by the S register. The K and V registers are cleared to zero. The V register is cleared to point at a complete character position. The J register is set to 2.

J = 2

Upon completion of the load B register memory access, the S register is set from the 15 low order bit positions of the word in the B register. If the word is an operand (B48F'), the K register is set from the B register contents. If the word is a descriptor, with the presence bit off, a presence bit interrupt is set in the interrupt register. The J register is set to 15.

J = 15

The syllable execute complete level is true to terminate the operator.

4.26 RECALL CONTROL ADDRESS RPAL-XX50

The word, at the address that is formed by reducing the address of the return control word by the repeat field, is read from memory. If the flag bit of the word is one and the presence bit is zero, the presence bit is set in the interrupt register and the operation terminated.

If the flag bit and the presence bit of the word are both one, bit positions 15 through 1 of the word are transferred to the C register and the L register is set to zero.

If the flag bit of the word is zero, the bit positions 15 through 1 of the word are transferred to the C register. Bit positions 38 and 37 are transferred to the L register. The C and L registers are advanced by one to specify the next syllable in sequence.

The address of the return control word remains unchanged.

SUMMARY OF OPERATION

The B register contents are placed in the A register for temporary storage. The address of the return control word is temporarily stored in the B register and is also transferred to the S register. The contents of the S register are temporarily stored in the F register. Fetch is inhibited and the L register is cleared to zero. The S register is counted down according to the value of the repeat count field. When the repeat count field equals one, S receives its final count and a load of the B register memory access is initiated.

When the B register is loaded from the address specified by the new S register setting, the C register is set from that word. If the flag bit is off, the L register is set from the word. If the word is a descriptor, with the presence bit off, the presence bit interrupt is set. When the flag bit is off and once the C and L registers are set from the word, the L and possibly C registers are incremented by one. If a Store Control Address Operator stored the address just recalled (flag bit off), one syllable is jumped over. A fetch is initiated to load the P register and the operation terminated.

DETAILED DESCRIPTION

J = 0

The B register contents are placed in temporary storage in the A register, with the A register being marked in an unoccupied status. The return control word address is obtained from the F register and transferred to the S register. It is also placed in temporary storage in the B register bit positions 16 through 30. The current S register is placed in the F register for temporary storage. Fetch is inhibited (ICFL; of fetch is false) and the L register is cleared to zero. The J register is set to 1.

J = 1

As long as the repeat count field is not equal to zero (TEZL'), the S register and the repeat count field are counted down by one. When the repeat count field equals one, the S register receives its final count and a memory access is initiated to load the B register with the word at the address specified by the S register. Upon initiation of the load B register memory access (MWOF), the return control word address is restored to the F register and the entry S register setting is restored from the F register. The J register is set to 2. Note that MWOF is true when initiation of the read memory access is verified to actually be in progress. Therefore, the address in the S register is no longer required.

J = 2

Upon completion of the memory access to load the B register (EEZL), the low order 15 bits of the word are transferred to the C register. If the flag bit of the word is off (B48F'), the 38th and 37th bits of B are transferred to the L register. If the flag bit had been on, the L register would have been left equal to zero. The J register is set to 3. If the presence bit is on or the flag bit is off (B46F + B48F'), initiate a fetch of the program word addressed by the C register (E to 16).

J = 3

If the flag bit is reset, the information addressed by the C register was formed during a Store Control Address operator. Because the L register setting points to the syllable following the Store Control Address syllable, the L register must be counted up by one at this time, to jump over the Jump operator syllable normally following a Store Control Address Operator. See write up for Store Control Address Operator. Therefore, if the flag bit is reset (B48F'), the L register is counted up by one. If the L register equals three (LE3L) at this time, the C register is counted up by one to address the next word in sequence. The initial B register contents are stored to the B register from temporary storage in the A register. The syllable execute complete level is true to terminate the operator.

4.27 SET SOURCE ADDRESS SSPL-XX22

The M register is set to the address formed by reducing the address of the return control word by the repeat field. The address of the return control word remains unchanged. Registers G and H are set to zero.

SUMMARY OF OPERATION

This operator is used to set the source string word address register (M register). The address of the return control word is obtained from the F register and placed in the M register. The contents of the M register are then reduced by the value of the repeat count field of the operator in the T register. The character and bit pointers are cleared to zero, thereby pointing to the most significant character and bit position of the word addressed by the M register. Unconditionally the A register is marked unoccupied.

DETAILED DESCRIPTION

J = 0

The address of the return control word currently in the F register is transferred to the M register and the J register is set to 1.

J = 1

If the repeat count field is greater than one (TE1L'), both the M register and the repeat count field are counted down by one. This process continues until the repeat count field equals one, at which time the M register receives its final count down (TEZL' is still true). The syllable execute complete level is true to terminate the operator.

The G and H registers are unconditionally cleared to zero, in order to point at the most significant character position of the word addressed by the M register. The A register is marked unoccupied by resetting AROF.

4.28 SET DESTINATION ADDRESS SDPL-XX06

The S register is set to the address formed by reducing the address of the return control word by the repeat field. The address of the return control word remains unchanged. Registers K and V are set to zero.

SUMMARY OF OPERATION

If, on entry to the operator, the B register is occupied, its contents must be stored in the destination string, prior to proceeding with the operators prime purpose. If the B register contents are not in the proper configuration for storage, the B register must be restored prior to the storage of its contents.

Once the B register contents are stored, or if the B register is initially unoccupied, the address of the return control word is transferred to the S register. The S register is counted down by the value of the repeat count field of the operator in the T register. The K and V registers are cleared to zero and the B register is marked empty. The operator then terminates.

DETAILED DESCRIPTION

J = 0

If the N register is not equal to zero (NEZL'), indicating the B register contents are not in the proper configuration for storage, the 8th bit of the N register is interrogated. If the 8th bit of the N register is set (N08F), the B register is shifted left and circulated with a corresponding increment of the N register. If the 8th bit of the N register is reset (N08F'), the B register is shifted right and circulated with a corresponding decrement of the N register. When the N register equals zero, and provided the B register is occupied (NEZL • BROF), a memory access is initiated to store the B register contents in the destination string.

When NEZL is true, the J register is set to 1. If the operator is entered with the B register unoccupied (BROF') and the N register equal to zero, the return control word address is obtained from the F register and transferred to the S register. If the B register is initially occupied on entry to the operator, the return control word address is not transferred to the B register until J = 1.

J = 1

Upon completion of the memory access to store the B register contents (when required), the return control word address is transferred from the F register to the S register. This is gated by EEZL • BROF. At this time, the B register is marked unoccupied. If the B register is initially unoccupied, the above actions (at J = 1) do not occur. In either case, the V register is cleared to zero.

On the following clock pulse after EEZL (if a memory access were initiated at J = 0), EWZL gates the following actions; otherwise, the following actions proceed at the first clock pulse of J = 1. If the repeat count field is greater than one (TE1L'), both the S register and the repeat count field are decremented by one. When the repeat count field equals one (TE1L), both the S register and the repeat count field are decremented by one. When the repeat count field equals one (TE1L), the S register is decremented for the final count down and the syllable execute complete level is true to terminate the operator. The H register (character pointer) is unconditionally cleared to zero.

4.29 TRANSFER SOURCE ADDRESS

SSAL-XX56

The eighteen bits of the three characters in the source string, starting at the position specified by the M and G registers, are transferred to the M and G registers. The three most significant bits are transferred to the G register and the remaining 15 bits are transferred to the M register. The H register is set to zero.

Prior to the execution of the operator, the H register is tested for zero. If the H register is not equal to zero, it is set to zero and the G register is increased by one. Overflow into the M register can occur.

SUMMARY OF OPERATION

When the H register is equal to zero, the high order octade of the repeat count field, in the T register, is cleared to zero; thereafter, the binary value of 16 is placed in the repeat count field. If the B register is occupied, its contents are stored in the destination string. If the A register is unoccupied, it is loaded from the source string.

A character (six bits) from the A register, specified by the G register, is transferred to the Y register. The B register is shifted left one complete character position (via two octade shifts) and the character in the Y register is transferred to the least significant character position of the B register. The repeat count field is decremented by eight, to tally the operation and the G register is incremented. The Y register is reloaded from the A register and thus the complete process is repeated until three load and shift operations are completed. Then the 18 least significant bits of the B register are transferred to the G and M registers. The A register is marked empty and the operation is terminated.

DETAILED DESCRIPTION

J = 0

Fetch is inhibited to give precedence to the storing and/or loading of the registers if required. The bit pointer (H register) is cleared to zero. If the H register is not initially equal to zero, the G register is counted up by one. Should the G register overflow, the source string word address register (M) is advanced to the next word and the A register is marked unoccupied.

When the H register is equal to zero (HEZL), its zero content is transferred to the high order octade position of the repeat count field in the T register (T12F through T10F). The J register is set to 1. If the B register is occupied at this time (BROF), its contents are stored in the destination string (E to 11). If when the B register contents are stored, the B register contents are in a restored status (NEXL), the B register is marked unoccupied. If the B register contents are stored in an unrestored status (NEXL'), the B register remains marked occupied, to indicate the B register must be reloaded at the end of this operator.

J = 1

If a store memory access were initiated at J=0, further actions await completion of the access (EEZL). If, at that time, the A register is unoccupied, a load A register memory access is initiated. The J register is set to 2. A binary value of 16 is placed into the repeat count field by the setting of T11F. The Automatic Count for Fetch Level is true (ACFL) to initiate a normal fetch operation. The load A register memory access takes precedence in case the P register also requires reloading (see Fetch Flow).

J = 2

Upon completion of the load A register memory access, if a load were required, the character specified by the G register is transferred from the A register to the Y register. The B register is shifted left one octade position and the J register is set to 3. The shift of the B register serves to clear the low order octade positions of the B register.

J = 3

The character in the Y register (six bits) is transferred to the low order character position of the B register. Simultaneously, the B register is shifted left to complete a full character of the B register contents. The Y register is cleared to zero in preparation for receiving the next six bits from the A register. The G register is counted up by one to address the next bits of the A register. The repeat count field is counted down by eight (T-8) to tally the transfer of six bits of the B register. If the repeat count field is greater than eight (TL8L'), the J register is set to 2 for reloading the Y register. If the G register overflows at this time (GE7L + 1), the source string word address register is counted up by one and a memory access is initiated to reload the A register. If the repeat count field is less than eight (TL8L), all eighteen bits from the A register have been transferred to the B register; the J register is set to 4 in this case.

J = 4

The B register bit positions 15 through 1 are transferred to the M register, with bits 18 through 16 transferred to the G register. The A register is marked unoccupied and the J register is set to 5 for termination of the operator. If, at this time, the B register is marked occupied, indicating its contents were stored in the destination string (at J = 0) in an unrestored status, a memory access is initiated to reload the B register with the contents, on entry to the operator. Note that the N register was not disturbed during the operator and, therefore, continues to maintain its relationship to the intial word in the B register.

J = 5

Upon completion of the memory access to reload the B register with its entry contents (if any were initiated), the syllable execute complete level is true to terminate the operator.

4.30 TRANSFER DESTINATION ADDRESS SDAL-XX07

The eighteen bits of the three characters in the destination string, starting at the position specified by the S and K registers, are transferred to the S and K registers. The three most significant bits are transferred to the K register and the remaining 15 bits are transferred to the S register. The V register is set to zero.

Prior to the execution of the operator, the V register is tested for zero. If the V register is not equal to zero, it is set to zero and the K register increased by one. Overflow into the S register can occur.

SUMMARY OF OPERATION

Upon entry into the operator, the setting of the N register is checked and if not equal to zero, the contents of the B register are shifted either right or left until B is restored to its original configuration. When G is restored, the word is stored in the destination string.

The B register bit pointer is unconditionally cleared. The character address is increased plus one if the bit pointer was not zero, to allow addressing the next full character. If, when increasing the character address, it overflows, the B register is marked as empty and the next word in the destination string is loaded into the A register.

The following registers are transferred to temporarily store their contents while the registers are being used for data manipulation: M to S (destination address to source address register), S to M (source address to destination address register), K to G (B bit pointer to A bit pointer register), G to K (A bit pointer to B bit pointer register), and B to A (the destination string word in B to the A register). The N register is set to 2 to allow the transferring of 3 characters of the destination string word, now in A, via Y, to the B register. During this transfer of 3 characters, if the A character address (G) overflows, the next word in the destination string is loaded into A. The transfer of characters is completed when the N register equals zero. At this time the 15 low order bits (destination address) are transferred to the destination address register (S) and the 3 bits containing the B character address are transferred to K. The A character address, and the source address temporarily stored in K and S, are transferred back to G and M. The B and A registers both are marked as empty when the operator is terminated.

DETAILED DESCRIPTION

J = 0

Upon entry into the operator the state of the N register is checked and if it is equal to zero (NEZL') and the 8's bit of N is on (N08F), the B register is shifted left and circulated to restore its contents to their original configuration within the register.

The N register is incremented for each shift until N equals zero, at which time the B register is restored.

If, upon entry, N is not equal to zero and the 8's bit is off (NO8F') the B register is shifted right and circulated to restore its contents to their original configuration within the register.

The N register is decremented for each shift until N equals zero at which time the B register is restored. The state of N's 8's bit limits the shifting of the B register to a maximum of 4 character shifts to restore the register.

When N equals zero (NEZL) and the B register is loaded, either upon entry or after restoring the B register, the contents of the B register are stored in the destination string, initiated by setting the E register to 11. The J register is set to one, to allow the next sequence of actions, when N = 0.

The B register bit point (V) is set to zero when no memory accesses are in process (EEZL). At the same time, the bit pointer is checked and if it is not equal to zero (VEZL'), the character address (K) is incremented by one. If the character address should overflow (KE7L • VEZL') when incrementing the K register, the destination word address is increased and the B register is marked as empty. The above actions adjust the bit pointer, character address and destination address to a full character in the destination string, in the case where a bit operator had preceded this operator. The J register is set to 2 to allow the next sequence of actions.

J = 2

If, upon entry, the B register was empty or at J = 1 it was marked as empty (BROF'), the A register is loaded with 8 characters from the destination string, by setting the E register to 4 to initiate the access.

Because the A register has the facilities to be transferred character by character to the B register and B can be transferred to S and K, it is necessary for certain registers to be interchanged to make use of these functions. The M register is transferred to S to temporarily store the source address.

The destination address, in S, is transferred to M to address the destination word to be loaded into A, if B were empty upon entry into J = 2 or if character address overflow occurs at J = 4. Because the destination string word, normally in B, is now in the A register (either being loaded there (E to 4) or transferred there (B to A) at J = 2), the B register address is transferred from K to G to address the proper character in A.

The A character address in G is transferred to K for temporary storage. The destination string word in B is transferred to A. If the B register was marked as loaded, the word in B is the one to be acted upon, however, if B was empty the transfer is redundant, since A will be loaded with a new word (E to 4).

The N register is used to count the number of characters shifted from A to B. Therefore, N is set to two and, when counted down through zero, will allow the desired 3 characters to be shifted. The J register is set to 3 to allow the next sequence of actions.

J = 3

With no memory accesses in process (EEZL), a character of the destination string addressed by G is transferred to the Y register. The B register is shifted right octally, to allow transferring of the character being shifted to Y to be transferred to B1 and B2 (low order character position of B). The J register is set to four to allow the next sequence of actions.

J = 4

The destination character, in Y, is transferred to B1 and B2. B2 through B16 are transferred octally to the right in preparation for the next character transfer. The Y register is cleared to allow the next character in A to be transferred to Y. The character address in G is incremented by one, to allow it to address the next destination string character in A. The N register, which is counting the 3 characters to be shifted, is decremented by one to record that a character has been transferred.

If the last character in A has been transferred to Y (GE7L) and the required 3 characters have not been transferred to B (NEZL'), the next destination word in sequence must be loaded into A to complete the transfer. This is accomplished by setting the E register to 4 and increasing the address in the M register by one.

If the 3 characters have not been transferred, indicated by N not being equal to zero (NEZL'), the J register is set back to 3 to allow the next character in sequence to be transferred to B. When the 3 characters have been transferred from A to B, the N register will equal zero (NEZL). At this time, the J register is set to 5 to allow the next sequence of actions and to terminate the operator.

J = 5

The 15 low order bits of B, containing the destination address just transferred there from the destination string via the A register, are now transferred to the S register (destination address register). The character address contained in B18 through 16 is also transferred to the B register character address register (K).

The A character address, temporarily stored in K, is restored to the G register. The source address, temporarily stored in the S register, is restored in the M register. Both the A and B registers are marked as empty prior to exit from this operator. The syllable execute complete level terminates the operator.

4.31 INPUT CONVERT ICOL-XX67

This operator converts decimal characters, in the source string, starting at the position specified by the M and G registers in an octal word in the position specified by the S register. The number of decimal characters converted is specified by the repeat field; the maximum value for the repeat field is 8.

If the value of the field converted is not zero, the sign of the field is obtained from the zone bits of the least significant character of the field (BA = 10 is minus, all other combinations are plus) and is stored in the sign position of the word. All other zone bits of the field are ignored.

If the value of the field converted is zero, the sign of the word is set to plus.

If, prior to the execution of this operator, the K or V registers are not zero, they are set to zero and the S register advanced by one.

If the repeat field is zero, no conversion takes place, but the register manipulation does take place. The conversion process treats the decimal field as an integer. The flag bit, exponent sign and exponent of the octal word are set to zero.

Prior to the execution of the operator, the H register is tested for zero. If the H register is not equal to zero, it is set to zero and the G register is increased by one. Overflow into the M register can occur.

SUMMARY OF OPERATION

J = 0

Upon entry to the operator, the B register is restored if required, and then stored. The A register is loaded, if initially empty or if it is marked empty; the G register may overflow if the H register is not intially equal to zero. If the repeat count field is equal to zero, the operator terminates when the B register is restored and the bit pointers are cleared to zero.

With the A register occupied and the B register empty, a character at a time is transferred to the B register via the Y register. The number of characters transferred to the B register is limited to a maximum of eight; the repeat count field high order octade is cleared to zero after its first tally decrement, consequently, only eight character transfers may be specified.

When all specified characters have been transferred to the B register, the A register is cleared to zero and the decimal to binary conversion is started. The repeat count field is set to 27, to allow the 27 shifts required for the conversion. (For the rules of decimal to binary conversion, see the write-up on decimal to binary conversion.) When the conversion is complete, the repeat count field will equal zero. At this time the sign of the original word is placed in the sign position for the mantissa of the octal word in the B register. The converted word is stored in memory, the S register is incremented and the operation is terminated.

DETAILED DESCRIPTION

J = 0

Due to the numerous occurrences at this J register setting, the described actions are based on the double line divisions.

HEZL'

If the bit point for the A register is not equal to zero (HEZL'), it is cleared to zero and the character pointer (G) is counted plus one. Should the character pointer overflow during the increment of the G register, the M register is advanced, with the A register being marked unoccupied.

NEZL'

Upon entry to the operator, if the B register is not restored (that is, the characters are not in their original positions as when they were initially loaded into B), it must be restored. Based on the status of NO8F, the B register is restored with the minimum number of shifts. If NO8F is set, the B register is shifted left and circulated while counting the N register up until N equals zero. If NO8F is reset, the B register is shifted right and circulated while counting the N register down until N equals zero.

NEZL

When the N register equals zero, or if zero on entry, the K and V registers are cleared to zero. Once the K and V registers equal zero, or if, initially, either equals zero, and provided the B register is occupied, a memory access is initiated (E to 11) to store the B register contents. The J register is set to 4 to await completion of the access and, subsequently, to increment the S register. The B register is marked empty. If, however, the B register is initially unoccupied, the S register is incremented by one. In this case, the S register is incremented to prevent the converted word from being stored over valid information currently addressed by the S register.

HEZL • NEZL • BROF' › TEZL'

If the repeat count field is not equal to zero, the B register marked empty, and both the N and H registers are equal to zero (HEZL • NEZL • BROF' • TEZL'), the B register is cleared to zero and the J register set to 1 to start the character transfer operation. If, at this time, the A register is marked unoccupied, it must be loaded prior to proceeding with the operation. The repeat count field is counted down by one (T-1) at this time (tally the action before the action). This is required in case the repeat count field initially equals eight; the eight's bit (binary value) of the repeat count field is cleared to J = 1, and must, therefore, be tallied prior to the transfer operation.

TEZL'

If the repeat count field is not equal to zero, Fetch is inhibited to give precedence to the memory access described above, plus any required during the transfer of characters to the B register.

J = 1

At the completion of the memory access to load the A register, if access is required, the character from the A register, specified by the G register, is transferred to the Y register. Simultaneously, the B register is shifted to the left one octade position. This is the first of two octade shifts required for one character shift. The high order octade of the repeat count field in the T register is cleared to zero, thus effectively limiting the number of characters to be converted to eight. The J register is set to 2.

J = 2

The character in the Y register is transferred to the B register input alignment station (first and second octade positions). The B register is shifted left for the second of two octade shifts, to complete a full character shift. The Y register is cleared prior to receiving the next character from the A register. The G register is counted up by one, to address the next character in sequence. If the repeat count field is not equal to zero (TEZL'), the repeat count field is counted down by one (T-1), to tally the next character transfer and the J register is set to 1, for the next transfer. If the G register overflows when it is incremented (GE7L + 1), the E register is set to 4 to load the A register with the next word in sequence. When GE7L is true, the M register is counted up by one to address the next word.

If the repeat count field is equal to zero (TEZL), the specified number of characters have been transferred to the B register. The A register is cleared to zero in preparation for further use during the conversion operation. The repeat count field is set to 27 for use during the conversion operation. The J register is set to 3 to start the conversion.

J = 3

Fetch is allowed at this time (ACFL goes true; see Fetch flow) because no additional memory access is required until after the conversion process.

TEZL' - Decimal to Binary Convert

With the repeat count field not equal to zero (it was set to 27 at J = 2), the Conversion begins. The setting of the repeat count field to 27, allows the 27 shifts necessary to convert 8 decimal digits.

B29T. This is a special shift right level, to shift only the Numeric bits of the B register. The Zone bits positions are left as they are. The low order numeric bit of each character is transferred to the high order numeric bit of the next character. The low order bit of the B register is transferred to A27F, with A27F and A02F being shifted one bit position to the right. The repeat count field is counted down by one for each shift. This operation continues until the repeat count field is zero (27 shifts).

TEZL-Conversion Completed

When the repeat count field equals zero, the Converted word in the A register is stored in memory by initiating a store memory access (E to 10). If the sign of the decimal word is minus, and the Converted word in the A register is not zero (B06F • B05F' • W06L'), the sign of the Converted word is set to minus by setting A47F to one. The J register is set to 4.

J = 4

Upon completion of the store memory access, initiated at either J = 0 by setting E to 11 or at J = 3 by setting E to 10, the S register is counted up by one and the J register is set to zero to either continue or terminate the operation.

4.32 OUTPUT CONVERT

OCOL-XX66

This operator converts an octal word in the source string, at the position specified by the M register, to decimal characters, starting at the position specified by the S and K registers. The number of decimal characters resulting from the conversion is specified by the repeat field; the maximum value for the repeat field is 8. If the octal word does not have a value of zero, the sign of the word is stored in the zone bits of the least significant character (for minus BA is set to 10, for plus BA is set to 00). All other zone bits are zero. If the octal word has a value of zero, the sign of the word is stored in the zone bits of the least significant character as a plus.

If, prior to the execution of this operator, the G or H registers are not zero, they are set to zero and the M register advanced by one. In the conversion process, the flag bit, exponent sign and exponent are ignored; the mantissa of the octal word is treated as an integer.

If the value of the mantissa is larger than can be converted to the field size specified by the repeat field, the characters converted will be the least significant characters of the decimal integer equivalent of the octal integer mantissa. If the number being converted is negative and non-zero, the sign of the result is set to negative, even if all of the least significant digits converted are zero. The characters, in excess of the number specified by the repeat field, are lost. The true/false indicator is set to false.

If the word can be converted to the field size specified, or if the repeat field is zero, the true/false indicator is set to true.

Prior to the execution of the operator, the V register is tested for zero. If the V register is not equal to zero, it is set to zero and the K register increased by one. Overflow into the S register can occur.

SUMMARY OF OPERATION

Upon entry to the operator, the bit pointers H and V and the character pointer G are cleared to zero, with the true/false flip-flop being set to the one state (true). If either the bit or character pointer for the source string had been on, the word address register for the source string (M) is advanced. If the bit pointer for the destination had been on, the character pointer for the destination string (K) is incremented. If the B register is occupied and in alignment relative to the K register, a shift and circulate of the B register contents is required if the associated character pointer is incremented (K+1). The repeat count field is checked, and if it is equal to zero, the operation terminates. If, after the bit pointers equal zero, the B register is occupied and in alignment but not restored (N register does not equal zero), the B register is shifted until N equals zero and then its contents are stored in memory to preserve its contents.

If the repeat count field is not equal to zero, the low order octade of the repeat count field is temporarily stored in previously cleared V register. The B register is cleared to zero and the A register loaded, if not occupied at this time. When the A register is loaded and the B register is cleared, the binary to decimal conversion is performed. (See write-up on binary to decimal conversion for specific details of conversion procedure.)

The converted characters are developed in the B register. If the conversion exceeds eight decimal characters (overflows), the true/false flip-flop is set to false. The sign of the converted word is set into the least significant character position of the converted word. The converted word is transferred to the A register and the B register is loaded from memory. The B register is loaded from memory because the converted decimal characters may require storage into a partial field, which in turn may extend into the next adjacent word in memory. The repeat count field is restored from the V register.

If the number of converted characters to be transferred is equal to eight, the B register is aligned, if necessary (K has not been altered since storing the B register contents) and transfer of the converted word commences, character by character to the B register. If the character transfer extends into the next word, the current word in the B register is stored and the next word brought up.

If the number of characters to be transferred is less than eight, the complement of the repeat count field, in the T register, is utilized in a comparison operation with the G register (G equals zero), for transfer alignment. When the G register equals the complement of the low order octade of the repeat count field, the transfer of characters to the B register is subsequently started. If the G register does not equal the complement of T (9 through 7), the G register is incremented with each clock pulse until the G register equals the complement of T (9 through 7). At this time the most significant character positions (which are not to be transferred as part of the final converted result) are checked for a zero content. If these character positions are not zero, the true/false flip-flop is set to false. When the G register points at the first character to be transferred, the transfer operation commences. Upon completion of the transfer of the converted decimal character, the operation terminates.

DETAILED DESCRIPTION

J = 0

The H, V and G registers are unconditonally cleared to zero and the true/false flip-flop is set to the one state (true.) If on entry the V register equals zero or after being cleared to zero, the repeat count field equals zero, the syllable execute complete level is true to terminate the operator. If either the bit pointer or character pointer for the source string does not equal zero (HEZL' + GEZL'), the source string word address register is incremented by one (M+1) and the A register is marked unoccupied.

If the bit pointer for the destination string does not equal zero (VEZL'), the character pointer is incremented (K+1). If, when the character pointer is incremented, the B register contains valid information and the contents are in alignment relative to the K register (BROF • KENL), the B register is shifted left one character position to maintain the alignment. One octade shift occurs at this J register setting and the second octade shift occurs at J = 10, to complete a full character shift. If the K register equals seven (KE7L) at this time, the B register is marked unoccupied as K will now point at the first character of the next word. If the B register is occupied with valid information, the reset of BROF indicates storage of the B register contents is required. If, however, when the character pointer is incremented the N register equals zero (indicating the B register is already in a stored status) and the K register equals seven (KE7L • NEZL), it is only necessary to increment the source string word address register (S+1). In this case, KENL is false and a transfer to J = 10 is not effected.

If, when the V register equals zero, the N register does not equal zero and provided the repeat count field is not zero, the 8's bit of the N register is interrogated (N08F • NEZL' • VEZL • TEZL'), to determine whether to shift the B register contents right or left to restore the B register contents (place in proper form for storage).

When the B register is restored, as indicated by N equal to zero, and the V register equals zero, with the repeat count field not equal to zero (NEZL • VEZL • TEZL'), the J register is set to 1. If the B register is occupied at this time (BROF), a store memory access is initiated to store the B register contents.

Fetch is inhibited to give precedence to memory access that may initially be required by this operator.

J = 1

Upon completion of the store memory access, if initiated (EEZL), the B register is cleared to zero (to provide a register for conversion) and the J register is set to 2. If the A register is marked empty, a memory access is initiated to load the A register. The low order octade of the repeat count field in the T register is transferred to the V register for temporary storage. The low order octade of the repeat count field will be restored to the repeat count field (at J = 3) for use in determining the number of final decimal characters to be accepted (a maximum of eight).

J = 2

At this time Fetch is allowed, as the conversion operation will consume more time than a Fetch operation, and no memory access will be required until the conversion is completed.

Upon completion of any memory access initiated (EEZL), the J register is set to 3 to commence the conversion operation. Simultaneously the mantissa of the word to be converted in the A register is checked. If the four high order digits (octades) of the mantissa in the A register do not equal zero (W08L'), the conversion must start at the high order digit of the mantissa; therefore, the repeat count field of the T register is set to 39 decimal to allow 39 pulses of shifting all 39 bits of the A register mantissa. If the A register

mantissa equals zero (W06L), conversion is not required; therefore, the repeat count field is cleared to zero. If the four high order digits (octades) of the A register mantissa equal zero, but the complete mantissa is not zero (W08L W06L'), the repeat count field is set to 27 decimal to allow for 27 shifts of the A register. Q01F is set to the one state to remember the starting point for the conversion. The source string word address register is counted up by one (M+1), as the word currently in the A register is the one addressed by the M register at this time.

Unconditionally, the B register is marked occupied because the B register should be occupied at termination of the operator.

J = 3

If the repeat count field is not equal to zero (TEZL'), the conversion is started. For details of the principal of the conversion, see write-up on Binary to Decimal conversion.

With the repeat count field not equal to zero, the repeat count field is counted down for each clock pulse and the A register is shifted left one bit position. The B30T transfer level is true to effect the conversion action.

The B30T is a special shift of the register that ignores the zone positions of the B register; i.e., the shift is B01 to B02, B02 to B03, B03 to B04, B04 to B07, etc. This action continues for either 39 or 27 pulses depending on the value set into the repeat count field at J = 2.

In conjunction with the above actions, if Q01F is set, only 27 shifts are required. The most significant bit shifted out of the A register into the B register is A27F. If Q01F is reset, then 39 shifts are required. In this case, the most significant bit of the A register mantissa (A39F) is shifted to the lease significant bit position of the B register.

If the converted number overflows the B register during the conversion process, the true/false flip-flop is set to zero state (false). That is, if during the conversion a number greater than four has been shifted into the numeric portion of the most significant character position of the B register (W080) and the repeat count field still does not equal zero (indicating additional shifts remain), the number to be converted is larger than can be converted.

If the sign of the mantissa of the word being converted is negative (A47F), the converted word is made negative by setting B06F; this is the zone portion of the least significant decimal digit of the converted word. For negative, BA equals 10.

TEZL

When the repeat count field equals zero, which it does after the conversion or it may result if the mantissa to be converted initially equaled zero, the E register is set to 3 to initiate a load of the B register. This load of the B register is required in case the converted digits are to be stored only in part of the word. The converted word in the B register is transferred to the A register as the B register will be loaded from memory. The A register is marked unoccupied as its contents are non-valid upon termination of the operator. The original low order octade of the repeat count field is restored to the T register, from temporary storage in the V register. The J register is set to 4.

J = 4

If the repeat count field is equal to zero and the load B register memory access is completed, the T1OF bit of the repeat count field is set to indicate a full eight characters are to be transferred as the final converted result. If the repeat count field equals zero at this time, it must have initially been greater than seven; otherwise, the operator would have terminated.

If the repeat count field is not equal to zero (TEZL'), less than eight decimal digits were initially specified to be converted as a final result. If less than eight were specified, only the least significant digit is the one to be transferred to the B register. In this case, the most significant digits in the A register (the ones exceeding the specified number) must be checked to determine if they equal zero or not. If they do not equal zero, the converted digits exceeded the specified number, and the true/false flip-flop must be set to false. The digits are checked (those to the left of the first to be transferred) as follows: the character pointed to by the G register is transferred to the Y register and the G register is counted up by one. The G register starts out at zero in this case. Simultaneously, the G register contents are compared with the complement of the repeat count field (low order octade) for equality as indicated by GETL. If GETL is true, the current increment of the G register will cause the G register to point at the first character to be transferred. Therefore, if GETL is true, the J register is set to 5 to start the transfer; otherwise, the operation is repeated. The V register is cleared as its function is finished.

J = 5

If the memory access to load the B register (initiated at J = 3) is completed at this J register setting or if the access is already completed (MROF + EEZL), the Y register is cleared to zero, as its current function is ended and the J register is set to 6. Unconditionally the Y register contents are compared against the Z register contents. The Y register contains an accumulation of the characters that were shifted to it at J = 4 (the shifts were single sided) and the Z register equals zero. If the level Y not equal to Z is true (YNZL), something exceeding the specified number of digits was converted; therefore, the true/false flip-flop is set to false.

J = 6

If the word brought up from memory is not in alignment relative to the K register, the B register is either shifted right or left and circulated with a corresponding decrement or increment of the N register, until the character position specified by the K register is in the output alignment station. When the character position is in the output alignment station (KENL NO1F'), the character pointed to by the G register is transferred from the A register to the Y register. The B register is shifted left one octade position as the first of two octade shifts to place the character position specified by the K register in the input alignment station of the B register. The J register is set to 7.

J = 7

The character in the Y register is transferred to the B register input alignment station and the B register receives its second octade shift, with the N register incremented by one to complete a full character shift. The Y register is cleared to zero in preparation of receiving the next character to be transferred. The character pointers G and K are advanced to address the next lower significant character position of the word, and the repeat count field is counted down by one to tally the character transfer. If the character pointer for the B register does not overflow (KE7L'), the J register is set to 8 to either terminate the operator or to continue transfer of digits.

J = 8

If the repeat count field is not equal and no memory access is in progress (TEZL' • EEZL), the next character from the A register is specified by the G register and is transferred to the Y register. The B register is shifted left for the first of two required octade shifts to complete a full character shift. The N register is incremented to tally the shift. The J register is set to 7.

J = 9

If, at J = 9, the character pointer for the B register overflowed, a memory access would have been initiated to store the B register contents. Upon completion of the store memory access (MWOF), the destination string address register is counted up by one (S+1) and the J register is set to 8, for either termination of the operator or continuation of the transfer operation. If, at this time, it is noted that the character pointer for the A register overflowed (GEZL), indicating all converted characters have been transferred, the B register is marked unoccupied (its contents were stored in order to arrive at this point). Thus, upon the return to J = 8, the operator will terminate. If, however, the character pointer did not overflow, indicating more characters remain to be transferred, a memory access is initiated to reload the B register. The return to J = 8 in this case results in a continuation of the transfer operation.

J = 10

Entry to this J register setting was necessary when at J = 0 the V register did not initially equal zero and the N register did not equal zero. If the V register did not equal zero, the K register was incremented and the B register was shifted one octade position. At this time, the second octade shift occurs in order to maintain the alignment of the B register contents, relative to the K register. If the K register overflows, BROF will be in the reset state at this time. Therefore, (BROF' • EWZL) initiates a memory access to store the B register contents. At the completion of the store memory access, the destination string word address register is counted up by one (S+1) and the return to J = 0 is gated by MWOF. If the K register did not overflow, BROF is in the set state at this time and gates an immediate return to J = 0.

4.33 SET TALLY
INCREASE TALLY
CHARACTER MODE NO-OP

SETL - XX42
INTL - XX40
NOCL - 0040

Set Tally: This operator sets the tally register to the value contained in the repeat count field.

Increase Tally: This operator increases the Tally register by the amount of the repeat count field. The tally is modulo 64; overflows are lost.

Character Mode No-Op: This operator will result in no actions occurring except SECL. It has a repeat field of zero upon entry to the operator.

SUMMARY OF OPERATION

Set Tally: The contents of the repeat count field are transferred to the R register.

Increase Tally: If the repeat count field is not equal to zero, the tally is increased by one and the repeat count field is decreased by one. When the repeat count field is equal to one (or if zero on entry), the syllable execute complete level is true (SECL).

Character Mode No-Op: This operator is the Increase Tally operator with a repeat count field of zero upon entry. The only action to occur is to initiate the syllable execute complete level.

DETAILED DESCRIPTION

J = 0 - Set Tally

Unconditionally transfer the repeat count field of the operator in the T register (T12F through T07F), to the R register bit positions R06F through R01F. The syllable execute complete level is true to terminate the operator.

J = 0 - Increase Tally - No-Op

If the repeat count field of the operator is equal to zero or one (TE1L), terminate the operator (EXIT). If TE1L is false (TE1L'), then decrement the repeat field of T by one; if the repeat field of the operator is unequal to zero (TEZL'), then increase the tally register by one (R + 1).

4.34 STORE TALLY STAL-XX41

This operator stores the 6 bit Tally register at the address formed by reducing the address of the return control word by the repeat field. The value contained in the Tally register is stored as an integer. The contents of the Tally register remain unchanged.

SUMMARY OF OPERATION

Upon entry to the operator, the B register is restored to its original state and then stored in the destination string. If the B register is initially empty, the operator proceeds as after the B register is restored. The address of the return control word in the F register is transferred to the B register and, subsequently, transferred to the A register for temporary storage. The Tally in the R register is then transferred to the B register. The address of the return control word in F is interchanged with the S register contents. The S register is reduced by the repeat count field value. The Tally in the B register is stored at the address specified by the S register. The return control word address temporarily stored in the A register is returned to the F register and the initial S register setting is restored to the S register. The B register is marked empty and the operator terminates.

DETAILED DESCRIPTION

J = 0

If NO8F is set, the B register is shifted left and circulated to restore its original configuration prior to storage in memory. The N register is counted plus one for every octade shift until N equals zero.

If N is not equal to zero (NEZL') and NO8F is reset, the B register is shifted right and circulated to restore it to its original configuration, prior to storing in memory. The N register is counted down by one for each shift until N equals zero.

When the B register is restored (NEZL), the contents of the B register are transferred to the A register as a step in ultimate storage in memory. The A register is marked unoccupied and the address of the return control word in the F register is transferred to the B register, as the first step in transferring it to the A register for temporary storage. If the B register was initially occupied, a memory access is initiated to store the A register contents which now contain the initial B register contents. The J register is set to 1.

J = 1

Upon completion of the store A register contents (initial B register contents, or if the B register was initially unoccupied (MWOF + BROF'), the address of the return control word currently in the B register is transferred to the A register. The B register is marked unoccupied and the J register is set to 2.

J = 2

The Tally in the 6 low order bits of the R register is transferred to the 6 low order bit positions of the B register. The address of the return control word in the F register is transferred to the S register and the destination string word address (S) contents are temporarily stored in the F register. The J register is set to 3.

J = 3

The return control word address, now in the S register, and the repeat count field in the T register are counted by one until the repeat count field equals zero. This action forms an address in which to store the Tally. When the repeat count field equals one (TELL), the S register is counted down for its final count and a memory access is initiated (E to 11) to store the Tally, currently in the B register, at the address specified by the new S register setting. The J register is set to 4 when TELL is true.

J = 4

Upon completion of the store memory access (MWOF), the address of the return control word is transferred to the B register and the J register is set to 5.

J = 5

The address of the return control word is restored in the F register. The initial S register setting is restored from temporary storage in the F register. The B register is marked unoccupied and the syllable execute complete level is true to terminate the operator.

4.35 RESET BIT

REBL-XX65

This operator sets bits to zero in the destination string, starting at the position specified by the S, K and V registers. Successive bits proceed from left to right. The number of bits set to zero is specified by the repeat field. The S, K and V registers, upon the completion of this operator, address the next bit in sequence.

SUMMARY OF OPERATION

Upon entry to the operator the repeat count field is checked. If it is zero, the operator terminates. If it is not zero, the B register is loaded if it is unoccupied. When the B register is loaded, it is aligned so that the character containing the first bit to be reset is in the output alignment station. When the B register is aligned, the character in the output alignment station is transferred to the Z register.

In reference to the character in the Z register, the bit position specified by the V register is reset. The repeat count field is counted down by one as the bit pointer (V register) is counted up by one. Thereafter, the next bit position, specified by the V register, is reset. If the bit pointer is to reset more bits than are contained in one character, successive characters are shifted to the output alignment station, with the current character in the Z register being restored to the B register (via the input alignment station) and the next character transferred to the Z register from the B register output alignment station. If the number of bits to be reset exceeds a full word (may start within a word and extend into the next word) the word in the B register is stored back into the destination string and the next word from the destination is loaded into the B register. The process of resetting bits continues.

When the repeat count field has counted down to zero, the operation terminates; i.e., when the repeat count field equals one, the last bit to be reset is reset and the character in the Z register is returned to the B register via the input alignment station. If the process of resetting bits ends with a bit other than the last bit of the character being reset, the B register will not be in alignment relative to the K register. Therefore, the final action of the operator is to shift the B register, such that the B register is in alignment relative to the K register (KENL is true). The character and bit pointers (K and V) are left pointing at the bit position following the last bit to be reset.

DETAILED DESCRIPTION

J = 0

If the repeat count field is equal to zero (TEZL), the operation is terminated by the syllable execute complete level (SECL) being true. If the repeat count field is not equal to zero and the B register is empty (TEZL' · BROF'), a load B register memory access is initiated by setting the E register to 3. The J register is set to 1 to await completion of the access.

If, or when, the B register is loaded and aligned (TEZL' • BROF • KENL • NO1F'), the status of Q09F is checked. Initially, Q09F is in the reset state. If Q09F is reset, the character in the B register output alignment station (B16 and B15) is transferred to the Z register. Simultaneously the B register is shifted left one octade position, the N register incremented and the logical flip-flop, Q09F, is set to the one state. The B register is shifted to bring the next character into the output alignment station and to make room for receiving the character in Z into the input alignment station. The second octade shift to complete a full character shift occurs at J = 2.

When Q09F is on, the bit position in the Z register, specified by the V register, is set to zero. The repeat count field is decremented to tally the operation and the V register is incremented to address the next bit position. If V is equal to 5 (VE5L), the K register is advanced to address the next character position of the B register, for subsequent reloading of the Z register. When the V register equals 5 or if the repeat count field equals one (VE5L + TE1L), the J register is set to 2. If TE1L is true, the required number of bits have been reset.

If the B register is occupied and the repeat count field is not equal to zero with Q09F in the reset state (BROF • TEZL' • Q09F'), the B register is either shifted right or left depending on the status of KO4F, to place the character specified by the K register in the output alignment station. This action may occur either initially or if the B register is reloaded during the operator. When KENL is true and NO1F is reset, the operation proceeds as described above.

J = 1

Upon completion of a load B register memory access, initiated either at J = 0 or J = 3, the B register is marked occupied and the J register is set to zero.

J = 2

The character in the Z register is transferred to the B register input alignment station. The B register is shifted left (minus the least significant octade) one octade position. This is the second of two octade shifts to complete a full character shift. The N register is incremented to tally the shift. The logical flip-flop Q09F is reset and the Z register is cleared to zero. If more bits remain to be reset, clearing Z and resetting Q09F permits loading the Z register with the next character at J = 0.

If both the K register and V register equal zero (KEZL • VEZL), the first bit of the next word in sequence is being addressed by K and V. Therefore, a memory access is initiated to store the current word in the B register in the destination string.

If the repeat count field is not equal to zero (TEZL') and K and V are not pointing at the first bit of the next word (KEZL' + VEZL') the J register is set to zero to continue the reset bit operation.

If the repeat count field is equal to zero and the K and V registers are not pointing at the first bit of the next word, the J register is set to 4 for termination of the operation.

J = 3

Upon completion of the store B memory access (MWOF), the destination string word address register is advanced by one (+1). If at this time the repeat count field is equal to zero (MWOF • TEZL), the J register is set to 4 for termination of the operator. If the repeat count field is not equal to zero (EEZL • TEZL'), a load B register memory access is initiated and the J register is set to 1, to await completion of the access and subsequent continuation of the operation.

J = 4

If, when the operation is terminated, the B register contents are not in alignment relative to the K register (KENL' + NO1F), the B register is shifted to the right and circulated by octade, with a corresponding decrement of the N register. If the final bit reset during the operator is the last bit position of a character (5th bit, at which time VE5L is true), the operation will terminate with the B register in alignment relative to the K register. In this case, shifting of the B register is not required at J = 4. In either case, when KENL • NO1F' is true, the syllable execute complete level is true to terminate the operator.

4.36 SET BIT

SEBL-XX64

This operator sets bits to one in the destination string starting at the position specified by the S, K and V registers. Successive bits proceed from left to right. The number of bits set to one is specified by the repeat field. The S, K and V registers, upon the completion of this operator, address the next bit in sequence.

SUMMARY OF OPERATION

The occurrences during this operator are identical to the occurrences during the Reset Bit operator with the exception of setting bits versus resetting bits. For a reference to the summary of operation and detailed description, refer to the write up on RESET BIT operator (REBL).

4.37 CALL REPEAT FIELD CLRL-XX43

The six low-order bits of the word, at the address formed by reducing the address of the return control word by the repeat field, is transferred to the repeat field of the T register.

If this field is not zero, the transfer of the repeat field of the subsequent syllable to the T register is suppressed and the present contents of the repeat field of the T register are used as the repeat field of the subsequent syllable.

If the repeat field is zero, the operator of the subsequent syllable is ignored and a Jump Forward Unconditional Operator is executed using the repeat field in the subsequent syllable. No interrupt may occur between the execution of this operator and the execution of the subsequent operator.

SUMMARY OF OPERATION

The address of the return control word is obtained from the F register and placed in the S register. The S register is decremented by the value of the repeat count field of this operator. The word addressed by the return control word address minus the repeat count field is brought to the B register. The low order six bits of this word are then transferred to the repeat count field in the T register. The repeat count field in the T register is then interrogated.

If the repeat count field is equal to zero, a Jump Forward Unconditional Operator code is forced into the T register. Simultaneously the repeat count field of the subsequent syllable is obtained from the P register and transferred to the T register repeat count field. The Jump Forward Unconditional Operator is entered and executed. In this case, the repetitive operation which was being tallied in memory has been completed; hence, a jump to the syllable following the loop just completed occurs.

If the repeat count field is not equal to zero, the repeat count field remains as it is and the operator code of the next syllable in sequence is obtained from the P register and placed in the T register. The next syllable in sequence (now in the T register) is executed with its repeat count field equivalent to the value obtained from the word brought from memory. In this case, the repetitive operation which is being performed must be performed at least one more time.

DETAILED DESCRIPTION

J = 0

The address of the return control word in the F register is placed in temporary storage in the B register. It is also transferred to the S register. The current S register value is transferred to the F register for temporary storage. The J register is set to 1, where the address of the return control word address minus the repeat count field is developed. If, initially, the B register contains valid information, it is transferred to the A register for temporary storage.

J = 1

As long as the repeat count field in the T register is not zero, both the S register and the repeat count field are counted down by one. When the repeat count field equals one (TE1L), the S register receives the final count and a memory access is initiated to load the B register with the word addressed by the S register. Once it is verified that the load B register memory access is in progress, as indicated by MWOF being true, the address of the return control word is restored to the F register (from the B register) and the initial S register setting is restored to the S register (from the F register). At the completion of the memory access to load the B register (MROF), the J register is set to 2.

J = 2

The six low order bits of the word just brought to the B register are transferred to the repeat count field of the T register. The J register is set to 3. If the B register initially contained valid information, BROF would still be in the set state and at this time gate a transfer of A to B, thereby restoring B's original contents.

J = 3

Unconditionally PROF is reset, thereby indicating that the next syllable to be executed is in the T register after this clock pulse time. This is a function normally performed during SECL which does not occur in this case. The J register is set to zero for use in the syllable to be executed next. At this same clock pulse time, the status of the repeat count field is interrogated.

If the repeat count field is equal to zero (TEZL), the third bit of the T register is set to one (T03F to 1) which changes the operator code in the T register to a Jump Forward Unconditional Operator. (CLRL - XX43 to FWJL - XX47). At the same time, the repeat count field of the next syllable in sequence (as specified by the L register) is obtained from the P register and is transferred to the repeat count field of the T register. Note that the L register was incremented during Fetch and thus points at the next syllable in sequence.

If the repeat count field is not equal to zero (TEZL'), the operator code for the next syllable in sequence (as specified by the L register) is transferred from the P register to the T register. The current value of the repeat count field remains as is.

4.38 EXIT CHARACTER MODE
IN-LINE EXIT CHARACTER MODE

RECL - XX00 (XX ≠ 01)
ILEL - 0100

The return control word is read from memory. The C, L, G, H, K and V registers are set to the contents of the respective field of the return control word. The S register is set to the contents of the F register field of the return control word.

The word addressed by the S register, the mark stack control word, is read from memory. The R and F registers are set to the contents of the respective fields of the mark stack control word. The Mark Stack Flip-flop and the Program Level Flip-flop are set to the contents of the respective positions of the mark stack control word. The S register is decreased by one. The A and B registers are set to empty. The Mode Flip-flop is set to word mode.

If the operator is the In-Line Exit Character Mode, then the C and L registers are not set from the return control word, instead the next operator in sequence is executed.

SUMMARY OF OPERATION

This operator restores the contents of the H, V, G, K, L and C registers from the return control word. The F and R registers, mark stack and sub-level flip-flops are restored from the mark stack control word. The S register is set to address the word prior to the mark stack control word. The next syllable to be executed is the one following the Descriptor Call or Operand Call that initiated entry to character mode. The next syllable interpretation is in word mode. If the operator was the In-Line Exit Character Mode, then the C and L registers do a normal count up at the SECL time that this operator was placed in the T register.

If the flag bit of the return control word is off, then terminate the operator with the return control word in the B register and set the flag bit interrupt.

DETAILED DESCRIPTION

J = 0

Fetch is inhibited (ICFL' is false) if the operator is Exit Character Mode (T07F'). A new program address is to be loaded into the C register and, thereafter, the P register is reloaded.

If the N register is not equal to zero (NEZL'), the B register is shifted either right or left and circulated, with the N register decremented or incremented (depending on the status of N08F) until the N register is equal to zero. When the N register equals zero, and provided the B register is occupied, a memory access is initiated to store the B register contents in the destination string. The J register is set to 1.

J = 1

Upon completion of the store B memory access, if initiated, the address of the return control word is transferred from the F register to the S register. A memory access is initiated to load the B register with the Return Control Word and the J register is set to 2.

J = 2

At the completion of the access to load the B register with the Return Control Word (EEZL), if the flag bit of the Return Control Word is one (B48F), the address of the Mark Stack Control Word is obtained from the B register and transferred to the S register. A memory access is initiated to load the B register with the Mark Stack Control Word. The true/false and sub-level flip-flops are set to zero prior to obtaining the new setting from the Mark Stack Control Word at J = 4. The J register is set to 3 where the registers are restored from the Return Control Word.

If the flag bit of the Return Control Word is off (B48F'), then the J register is set to 12 to terminate the operator with a flag bit interrupt.

J = 3

The H, V, G and K registers are set from the contents of the Return Control Word in the B register. If the operator is the Exit Character Mode (T07F'), then the L and C registers are also set and the E register is set to 16, thereby initiating a fetch of the program word (specified by C) containing the syllable following that which initiated entry to character mode. The J register is set to 4.

J = 4

Upon completion of the memory access to load the B register with the Mark Stack Control Word (EEZL), the F and R registers are set from the F and R fields of the Mark Stack Control Word in the B register. The Mark Stack Flip-flop and the Sub-level Flip-flop are set from their respective bit positions of the Mark Stack Control Word. Note that the true/false flip-flop in character mode is equivalent to the Mark Stack Flip-flop in word mode (Q12F). The S register is counted down by one, to address the word below the Mark Stack Control Word. AROF, BROF, the Character Word Mode Flip-flop, the X, N and M registers are all cleared to zero. The syllable execute complete level is true to terminate the operator.

J = 12

Entry to this J register setting occurs only if the flag bit of the Return Control Word is off. Unconditionally, the B register is marked as occupied (BROF to 1) and the operator terminated. If the Processor is in normal state (NCSF), then set the flag bit interrupt.

SECTION 5

CONTROL STATE AND MISCELLANEOUS OPERATORS

5.1 COMMUNICATE OPERATOR

PURPOSE

To allow the Processor to communicate with the M.C.P. when a special subroutine or action by the M.C.P. is needed by the program in process.

SUMMARY

If the Processor is in the control state, exit the operation. If the Processor is in the normal state, check for the location of the top word in the stack. If it is not in either the A or B registers, load it into B from memory. If it is in either A or B, or when placed there, initiate a store memory access and store the word into the Program Reference Table. The particular location in the PRT is addressed by transferring the contents of the R register to the M register and increasing it by a value of nine. The Communication Interrupt bit in the Interrupt register is set to transfer from Normal State to Control State. Upon completion of the store operation, the operation is terminated.

DETAILED DESCRIPTION

J = 0

SECL

Allow the syllable execute level to complete the operation if the Processor is in the Control State as indicated by the Normal Control State Flip-flop being off (NCSF'), essentially this becomes a NO-OP for the Control State .

When the memory operation initiated by this operator is completed, memory write access is obtained (MWOF). MWOF allows an exit from the operation.

If the Processor is in the Normal State as indicated by NCSF being on, one or more of the following actions and the completion phase above will occur:

E to 3

Set the E register to 3 to initiate a load B memory access, if the word in the top of the stack is not in the A or B registers as indicated by the A and B register occupancy flip-flops being off, ($\overline{AROF} \cdot \overline{BROF}$).

J to 1

Advance the Control State to one to complete the memory access initiated as a result of the A and B registers being empty.

E to 13

Set the E register to initiate a store B memory access. B will be stored at the address specified by the contents of the M register (which will contain the address in R, + 9).

If the top word of the stack is in the B register (BROF) and the A register is empty ($\overline{\text{AROF}}$), store this word in the memory area reserved for it in Program Reference Table (R + 9).

BROF to 0

If the B register is occupied and is to be stored, mark it as empty.

E to 12

AROF to 0

Set the E register to 12 to initiate a Store A Memory Access. The top word of the stack is in the A register indicated by the A register occupying flip-flop (AROF) being on. The word will be stored in the Program Reference Table at the address specified by the R register plus nine.

Reset the A register occupancy flip-flop to show that A is empty for the next operator.

R to [M15F through M07F]

Set the Base address of the Program Reference Table contained in the R register in the high order positions of the M register.

M04F & M01F to 1

Increase this address by 9 by turning on the 8's bit and the one's bit. This establishes the address in the Program Reference Table where the word in the top of the stack is to be stored.

UI7F to 1

Set the Communication Interrupt bit on in the Interrupt register of Processor A or B (whichever Processor is being used) to cause a communication interrupt condition.

J = 1

S - 1

BROF to 1

J to 0

Upon completion of Memory access, which was initiated at Control State equal to Zero only if both A and B registers were empty, count the stack address register down one to show that the contents of the stack are decreased by one, Mark the B register as occupied, and reset the control to zero to complete the operation.

5.2 PROGRAM RELEASE
I/O RELEASE

PREL - 0111
IORL - 2111

PURPOSE

Program Release: This operator functions to release the Processor from an object program in order to allow the MCP to initiate an I/O operation. When the object program requires an I/O operation, execution of this operator by the object program will result in placing the address of the I/O descriptor in R + 11 (octal), and set an interrupt which will result in the necessary MCP action to execute the I/O operation. The Program Release Operator will accomplish this in the following manner.

1. Transfer the I/O descriptor address from core memory to the "A" register.

2. If in Control State, reset the presence bit of the Data descriptor, that addresses the I/O descriptor, to mark the input/output area of core memory as unavailable to the object program. If in Normal State, set either the Continuity Bit Interrupt or the Program Release Interrupt depending upon the status of the Continuity Bit in the Data descriptor.

3. Restore the Data descriptor back into its original location in core memory.

4. Terminate the operator, if in Normal State, then store the absolute address of the I/O descriptor into the PRT at R + 11. (The MCP will transfer this address to core memory location 10 for use by the I/O unit.)

I/O Release: This operator is used by the MCP after an I/O operation to mark the input/output area of core memory as available to the object program. The I/O Release operator accomplishes this in the following manner:

1. Transfer the Data descriptor that addresses the I/O descriptor from core memory to the "A" register.

2. Turn on the Presence Bit of the Data descriptor. This marks the input/output area as available to the object program.

3. Restore the Data descriptor to core memory and terminate the operator.

SUMMARY OF OPERATION

Due to the similarity of the two operators, Program Release and I/O Release, the operation of both will be described as one, with the differences in each specified when necessary.

Upon entry to either operator, the top word in the stack must address the Data descriptor that addresses the I/O descriptor to be effected. This word may be either a Data descriptor or an operand.

If the word in the top of the stack is a descriptor, then the 15 low order bits of this word contain the absolute address of the Data descriptor. The Presence Bit of the descriptor is examined if it is found OFF, then the operation is terminated with the non-present descriptor in the top of the stack and the Presence Bit Interrupt is set if the Processor is in Normal State, if it is ON, then the 15 low order bits of the descriptor are used to access core memory and place the Data descriptor into the "A" register.

If, upon entry to the operator, the top word in the stack is an operand, then relative addressing is required. The relative addressing that takes place will be R+, F+, or F-. No "C" relative addressing can take place with this operator. If the bit configuration of the "A" register specifies "C" relative, R+ relative addressing will take place instead.

With the I/O Release operator and the Processor in Normal State, none of the described actions will take place and the operator is treated as a NO OP.

When the Data descriptor that addresses the I/O descriptor is in the "A" register, the actions which take place will depend upon which operator is in the "T" register.

I/O Release: With this operator the Presence Bit of the Data descriptor is turned ON and the Data descriptor is restored into its original location in core memory and the operation is terminated.

Program Release: If the Processor is in Control State then the Continuity Bit of the Data descriptor is examined; if it is ON then the Continuity Bit Interrupt is set; if it is OFF then the Program Release Interrupt is set. The Continuity Bit being ON indicates that this descriptor is linked with one or more other I/O descriptors and will result in the MCP rotating the descriptors prior to initiating the I/O operation. In either case, the Data descriptor will be restored to its original location in core memory and the operation is terminated. If the Processor is in Normal State, then the address of the Data descriptor will be placed into R + 11.

DETAILED DESCRIPTION

For the actions described herein reference the Program-I/O Release Flow Chart, 3.04.0.

JOOL

EXIT

If this is the I/O Release operator and the Processor is in Normal State (IORL • NCSF), the Syllable Execute Complete Level is TRUE to terminate the operator. Thus, the I/O Release operator will function as a NO OP when executed in Normal State.

Q09F to 1

If the operator is the Program Release operator or I/O Release and the Processor is in Control State (PREL + IORL • NCSF'), Q09F is set to one in order to allow the parallel adder to function as a ten bit address adder.

The logical equation below the double line which states (PREL + IORL • NCSF') will also control the following functions:

B to A, BROF to 0, AROF to 1.

If the A register is empty, as indicated by AROF', then stack adjustment is required. The top word of the stack may be located in the B register at this time, therefore transfer the contents of the B register to the A register, and mark the B register as empty and the A register as occupied. If the B register is unoccupied at this time, then only the setting of AROF is of value, the other actions are redundant.

E to 2, J to 1.

If both the A and B registers are empty (AROF' • BROF') then the top word of the stack is in core memory, set E to 2 to initiate a memory cycle, the S register as the addressing register, to place the top word of the stack into the A register. Set the J register to 1 to await completion of this memory cycle.

The following actions are controlled by the logic below the second double line that reads AROF • A48F (PREL + IORL • NCSF'). Again the term within the parenthesis specifies the operator is either the Program Release or the I/O Release and in Control State, which indicates that the operator can be executed. The terms AROF • A48F indicate that the top word of the stack is in the A register and that this word is a descriptor, no relative addressing is necessary and the following actions will occur:

Presence Bit Interrupt

If this is a non-present descriptor and the Processor is in Normal State (A46F' • NCSF) then set the Presence Bit Interrupt; the I/O descriptor pointed to is not available to the object program.

J to 15.

If the descriptor in the A register is non-present, then set the J register to 15 in preparation of termination of this operator.

A [15 ⇒ 1] to M, E to 4, J to 3

If this is a descriptor that is marked as present (A46F), then the operation may proceed as normal. Set the 15 low order bits of this descriptor into the M register, set the E register to 4. This action will result in using the M register as an addressing register to access the I/O descriptor and place it into the A register. The J register is set to 3 to allow the subsequent functions of this operator to take place.

If the operator is the Program Release or I/O Release in Control State and the A register contains an operand, AROF • A48F' (PREL + IORL • NCSF'), then the actions listed below the third double line at JOOL can take place.

J to 2

The J register is set to 2 in order to access the I/O descriptor via relative addressing.

R to M [15 ⇒ 7]

If the Processor is not in sublevel (SALF'), or the tenth bit of the A register is reset (A10F'), or if the configuration of bits in the A register is such that the condition of A10F • A09F • A08F' exists, then the relative addressing will be R relative; transfer the contents of the R register into the 9 high order bits of the M register in preparation of address indexing. The configuration of A10F • A09F • A08F' indicates C relative addressing, but for this operator it is forced to R relative addressing.

A10F to 0

Reset the tenth bit of the A register if the Processor is in sublevel and the A register contains the bit configuration of A10F • A09F'. This indicates that relative addressing is F relative and can only use the 8 low order bits of the A register; reset A10F so that it will have no effect on the address adder.

The actions listed below the fourth double line are controlled by the logic indicated below this double line, SALF • A48F' • A10F • AROF • (PREL + IORL • NCSF'). This logic states that the Processor is in sublevel and the tenth bit of the A register is on, therefore indicating eventual F relative addressing. The following actions will take place:

F to M

With the Mark Stack Flip-flop reset (MSFF'), the F register is addressing the top RCW (Return Control Word) in the stack if the configuration of bits in the A register (A09F' + A08F) indicate F relative addressing; transfer the contents of the F register to the M register.

R + 7 to M, E to 6

If the F register is addressing a Mark Stack Control Word and F relative addressing is required (MSFF • (A09F' + A08F)), the address of the top RCW in the stack must be obtained from R + 7 of the PRT.

Set into the M register the value of R + 7 and set the E register to 6. This E register setting will result in bits 16 ⇒ 30 of the word in R + 7 being transferred into the M register, this will be the address of the top RCW in the stack. This is necessary because any F relative addressing is relative to the top RCW in the stack.

A10F to 0, A09F to 0

With A08F reset, then the relative addressing will only use 7 low order bits of the A register, reset bits 9 and 10 of the A register.

Δ A [7 ⇒ 1] to A [7 ⇒ 1], Q01F to 1, Q03F to 1

If both A08F and A09F are set then the relative addressing will be F-; complement the 7 low order bits of the A register to allow the adder to perform a subtraction. Q01F is set to provide the pre-carry for the adder logic, Q03F is set to allow the adder to include the proper carry level into the 5 high order bits of the address result.

J01L

This J register setting is used for stack adjustment to complete its memory access if required. Upon completion of this memory access (MR0F) the S register is decremented (S - 1) and control is returned to J = 0.

J02L

Control to this J register setting is achieved only if relative addressing was necessary. If a memory access is in progress then all actions at this time will wait for the access to complete (EEZL). The only time a memory access would be in progress at this time is if an access had to be made to R + 7 at J00L. When EEZL goes true, the 10 lower order bits of the M register are added to the 10 low order bits of the A register and to Q01F. Q01F will be on only if the relative addressing is F-. Depending upon the status of Q03F and W10L (carry level from the tenth bit position of the parallel adder) the 5 high order bits of the M register will be either incremented or decremented, as required.

Upon completion of this address addition the M register will contain the absolute address of the I/O descriptor. Initiate a memory cycle (E to 4) and change control to J = 3. This memory cycle will place the I/O descriptor into the A register.

J03L

All actions described within this J register setting are conditioned by the term EEZL which indicates that the memory cycle to obtain the I/O descriptor is completed and the I/O descriptor is now in the A register.

J to 4

If the operator is the Program Release operator and the Processor is in Normal State (NCSF • PREL) then set the J register to 4 in preparation of storing the address of the I/O descriptor into the PRT at R + 11.

E to 12

Setting the E register to 12 will initiate a memory cycle to restore the I/O descriptor back into its original location.

A46F to 1

If the operator is the I/O Release operator (IORL) then set the presence Bit of the I/O descriptor.

Continuity Bit Interrupt to 1, Program Release Interrupt to 1

If the operator is the Program Release operator and the Processor is in Normal State (PREL • NCSF) then set one of the two interrupts indicated as determined by the Continuity Bit of the I/O descriptor, A28F to A28F'.

J to 5

With the Processor in Control State or with the I/O Release operator (NCSF' + IORL) change control to J05L in preparation of termination of this operator.

A46F to 0

Reset the Presence Bit of the I/O descriptor if this is the Program Release operator and the Processor is in Control State (PREL • NCSF').

J04L

This J register setting is only entered if the operator is Program Release and the Processor is in the Normal State. Upon completion of the memory access to store the I/O Descriptor into the cell from which it was read out of (EEZL), the address of the cell in which the I/O Descriptor has just been stored is transferred from the M register to the 15 low order bits of the A register; the remainder of the A register (Bits 16 thru 48) is cleared to zero. The address R + 11 is placed in the M register and a memory access is initiated to store the address of the I/O Descriptor in cell R + 11. The J register is set to 5.

The action of storing the address of the I/O descriptor into cell R + 11 is required for the MCP to initiate an I/O operation.

J05L

This J register setting is for the normal exit from the operator. The A register is cleared and the Syllable Execute Complete Level is true (EXIT).

J15L

This J register setting is achieved only if a non-present descriptor is in the top of the stack at J00L. The only action to occur here is to terminate this operator.

5.3 INTERROGATE INTERRUPT

IINL - 0211

PURPOSE

The Interrogate Interrupt operator is used to check for the presence of an interrupt when the Processor is in Control State. If an interrupt exists then change control to the address specified by the Interrupt Address Register in Central Control.

SUMMARY OF OPERATION

This operator is for control state use only, if used in normal state it is a NO-OP. If in control state and no interrupt exists in the Interrupt Address Register, the operator is a NO-OP.

Any interrupt in the Interrupt Address Register will have IA6F and/or IA5F set. By interrogating these two bits of the interrupt register, it can be determined whether or not an interrupt exists. If no interrupt exists in the IAR's (Interrupt Address Register) then the mnemonic term I03L from CC will be True. I03L in turn is the level that (I03L = IASF' • IA5F' • IA6F') enables the IINL operator to function as a NO-OP.

If an interrupt does exist, it is indicated to the Processor by the mnemonic term I02L. This level (I02L = IASF + IA5F + IA6F) is True if either IA5F or IA6F is set or if IASF (Interrupt Address Sync Flip-flop) is set. IASF is set by the Interrogate Interrupt Operator, it prevents the interrupt priority logic in Central Control from changing the status of the IAR while interrogating an interrupt.

Once IASF is set the term I01L is (I01L = IASF • (IA5F + IA6F)) enabled to allow the transfer of IAR to the C register. The operator will then terminate with the S register set to 100 (octal) and a fetch access being initiated.

DETAILED DESCRIPTION

JOOL

EXIT

If the Processor is in Normal State or if no interrupt exists (NCSF + I03L) the operator will function as a NO-OP; enable SECL to terminate the operator.

INHIBIT COUNT UP FOR FETCH

If an interrupt exists in the IAR and the Processor is in Control State (I02L • NCSF'), inhibit any fetch actions, control will be changed as determined by the IAR.

IASL

Interrupt Address Sync. set Level is sent to CC, + 0 set IASF.

IASF disables the Interrupt Address Register set so that it cannot be changed during the execution of the IINL syllable.

When IASF has been set, the actions below the second double line are enabled. Transfer the Interrupt address to the C register's six low order positions and clear the remaining positions of the C register. Set E16 to one to initiate a fetch access. Set the J count to one to continue the execution of the operator.

JO1L

Unconditionally allow the Syllable Execute Complete Level to terminate this operation.

Set the S register to an address of 100, base address of the MCP stack.

Clear the old Stack Address in preparation for inserting a new address. Clear the L register so that the first syllable of the Interrupt Program Word is referenced.

Clear the Interrupt Address register flip-flops as this information is now contained in the C register and clear the particular interrupt that was interrogated.

5.4 STORE FOR INTERRUPT
STORE FOR TEST

SFIL - 3011
SFTL - 3411

STORE FOR INTERRUPT

This operator can be used programmatically, but is usually hardware generated. It is generated at SECL time if the Processor is operating in Normal State and an interrupt exists. The operator will store the contents of the A and B registers, if they are valid, and construct the necessary control words in the stack for either word mode or character mode.

If this operator occurs in Processor 1, an Interrogate Interrupt operator is forced at the termination of the Store for Interrupt. If this operator occurs in Processor 2, the Processor is placed into an idle condition at the termination of this operator.

If the operator is used programmatically when in normal state the Processor will remain in normal state.

STORE FOR TEST

The Store For Test operator is included for test and diagnostic purposes. It is normally hardware initiated and used as an automatic part of the test procedure but can also be used as a programed operator. This operator can be used in either word or character mode and in either normal or control state, its operation closely related to the Store For Interrupt operator.

SUMMARY OF OPERATION

The Store For Interrupt and Store For Test operators are normally hardware initiated but can be inserted as a programed operator. If the Store For Interrupt operator is used programmatically when in normal state the Processor will remain in normal state, if in character mode then the Processor will remain in character mode, the remainder of a program initiated Store For Interrupt is the same as a hardware initiated Store For Interrupt.

The Store For Interrupt operator will vary, for word mode and character mode, in the construction of the stack. Referencing Figure 5.4-1, the Store For Interrupt stack construction for word mode is as follows:

1. Store the contents of the B register if the B register is marked as occupied.

2. Store the contents of the A register if the A register is marked as occupied.

3. Construct and store an Interrupt Control Word, this word will contain the value of the M register, N register, R register, MSFF, SALF and VARF.

4. Construct and store an Interrupt Return Control Word, this word will contain the contents of the C register, F register, K register, G register, L register, V register, H register and the status of BROF.

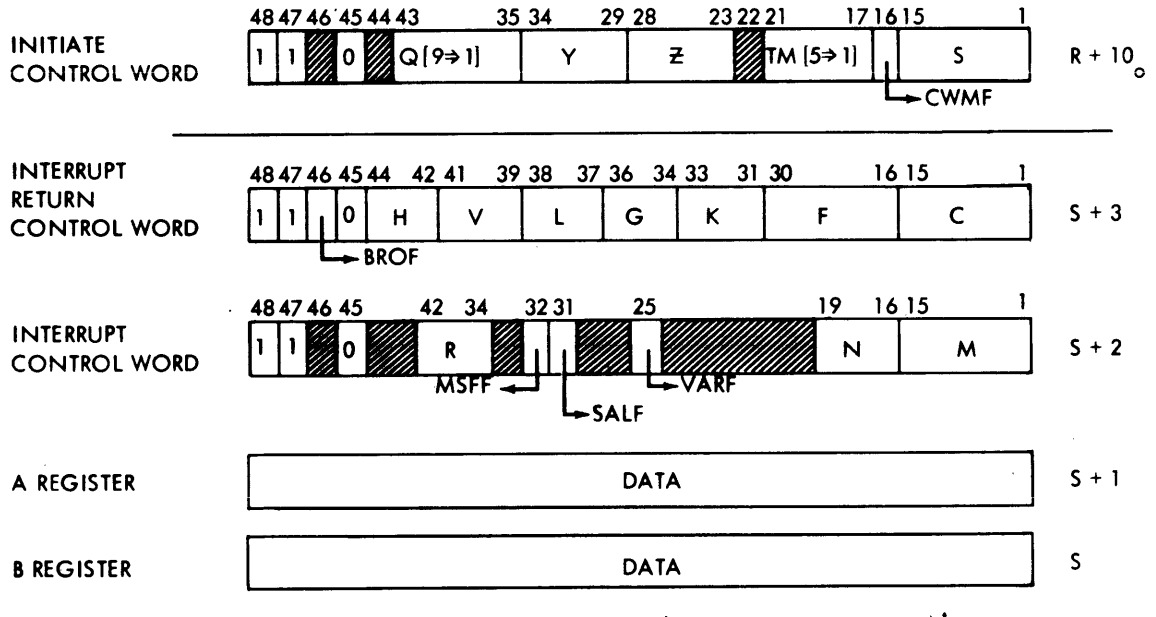


Figure 5.4-1 WORD MODE STACK

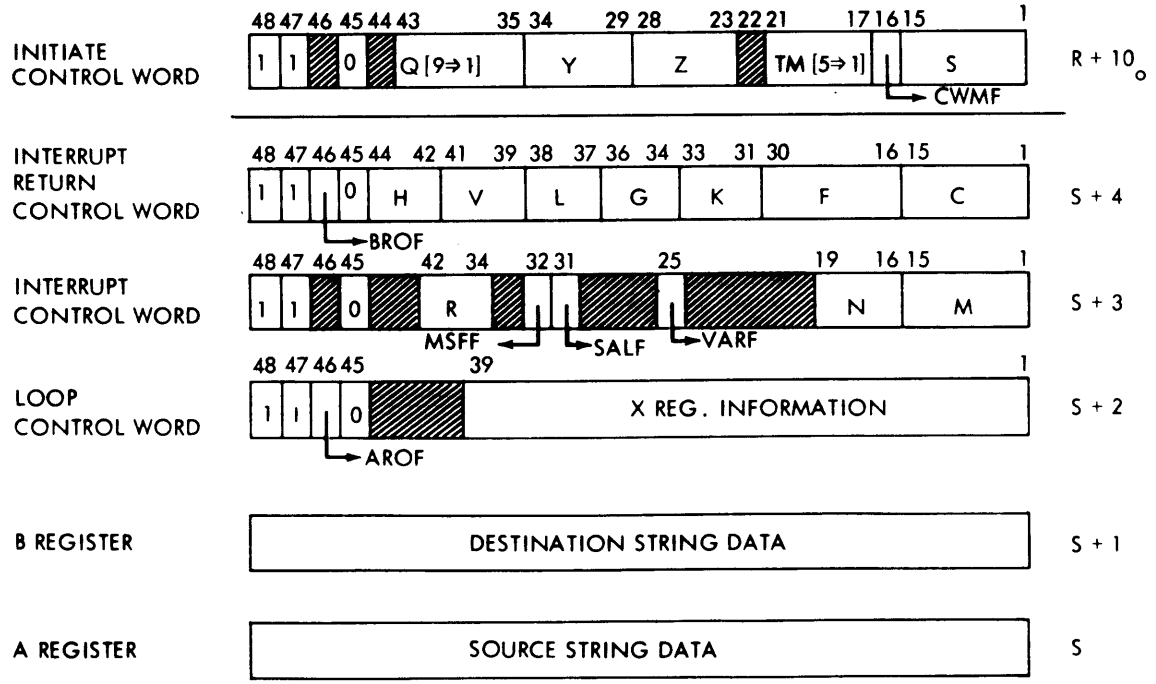


FIGURE 5.4-2 CHARACTER MODE STACK

After the stack is constructed the Store For Interrupt operator will construct and store an Initiate Control Word in the PRT at the location of R + 10. The Initiate Control Word will contain the values of the Q register (bits 9 ⇒ 1), the Y register, the Z register, bits 5 ⇒ 1 of the TM register, CWMF, and the S register. The storing of the Q register and the TM register is not required for the Store For Interrupt operator but is a redundant operation at this time. The S register setting in the Initiate Control Word is the address of the Interrupt Return Control Word that was stored in the stack. The termination of the Store For Interrupt operator will idle the Processor if this is Processor 2; if this is Processor 1 then the T register will be forced to the Interrogate Interrupt operator.

If the Store For Interrupt operator is executed when in character mode then, referencing Figure 5.4-2, the following is a description of the stack construction:

1. If valid then store the contents of the A register.
2. If valid then store the contents of the B register.
3. Transfer the Loop Control word from the X register to the A register, construct and store an Interrupt Loop Control Word.
4. Construct and store the Interrupt Control Word.
5. Construct and store the Interrupt Return Control Word.

The difference between the stack construct for word mode and character mode is that in character mode the A register is stored before the B register, prior to construction the Interrupt Control word an Interrupt Loop Control word is placed into the stack.

Before storing the Initiate Control Word the proper R register setting must be found when in character mode. This is accomplished by accessing the Mark Stack Control Word addressed by the F register and transferring the R field to the R register. The Initiate Control Word is constructed and stored into the PRT at location R + 10.

The termination of the operator is the same as when in word mode except that if the operator was program initiated then the Processor is left in character mode.

The Store For Test operator functions in the same general manner as the Store For Interrupt in character mode. With this operator the contents of the A and B registers are always stored, whether they are marked occupied or not marked as occupied. The termination of the Store For Test operator will access cell zero of core memory to obtain the address of the next operator to follow the Store For Test operator.

DETAILED DESCRIPTION

JOOL

The count up for Fetch is inhibited to prevent the counting of the C register and the initiation of a memory fetch access.

NCSF to 0

If no memory access is being executed (EEZL) and this is a hardware generated Store For Interrupt (Q07F) or the Store For Test operator (T09F) then place the Processor in control state by setting the Normal Control State flip-flop to zero (NCSF to 0).

J to 10

If the A register is valid with the Processor in character mode (AROF • CWMF) or if the operator is the Store For Test (T09F) then change control to J = 10 to store the A register.

The following actions will occur only if the Processor is in character mode. The S register contents are exchanged with bits 30 through 16 of the X register. This action will place the address of the top word of the stack into the S register and the address of the character mode destination string into the X register bits 30 = 16.

J to 1

If the A register is empty and this is not the Store For Test operator (AROF' • T09F') then change control by setting the J register to 1.

J to 2

If the operator is the Store For Interrupt with the Processor in word mode (CWMF' • T09F'), then change control to J = 2.

S + 1, E to 11

When in word mode and the Store For Interrupt operator (CWMF' • T09F' term below the double line) with the B register occupied (BROF), store the B register contents (E to 11) and increment the S register (S + 1) to address the next higher stack location.

J01L

Entry to this J register setting is achieved only if the Processor is in character mode or the operator is the Store For Test operator. This J register setting will, upon the completion of any previous memory access (EEZL), construct an Interrupt Loop Control Word in the A register and, if necessary, store the B register contents.

The Interrupt Loop Control Word will contain the X register contents in A (39 through 1) and the status of AROF at the time of interrupt (AROF to A46F). The J register is set to 2 and AROF is set to 1 in order to store the A register contents.

S + 1, E to 11

If the B register is occupied (BROF) or the operator in the T register is the Store For Test operator (T09F) then increment the S register to address the next stack location in core memory and set the E register to 11 to store the B register contents.

J02L

J to 3

Upon completion of the memory cycle to store the B register contents (EEZL) the J register is set to 3 to prepare for the construction of an Interrupt Control Word.

AROF to 0, S + 1, E to 10

If the A register is occupied, either with an Interrupt Loop Control Word for character mode or with data for word mode, then mark the A register as empty and initiate a memory cycle to store the A register contents (E to 10) and count up the S register to address the next core stack location.

J03L

When no memory accesses are in progress (EEZL) with this Processor, construct an Interrupt Control Word in the B register. This control word will contain the contents of M, N, R, MSFF, SALF and VARF. Set E to 11 to store this word in the stack, increment the S register by one, and control is changed to J = 4.

J04L

Upon completion of the memory cycle that stores the Interrupt Control Word as indicated by EEZL an Interrupt Return Control Word is constructed in the B register. The S register is counted up 1 to point to the next core location of the stack and the E register is set to 11 to store this Return Control Word.

J to 5

If the Processor is in character mode then the J register is set to 5. This will result in the correct R register setting being found in order to locate the address of R + 10. If in character mode the R register is used as a tally register.

J to 11

If in word mode (CWMF) then the R register contains the base address of the PRT. Change control to J = 11 in order to construct and store an Initiate Control Word.

J05L

Upon entry to this J register setting the F register contains the address of the top Return Control Word in the stack (constructed when character mode was entered) and the S register contains the address of the Interrupt Return Control Word just stored at J = 4. Interchange the F and S register contents in order to address the Return Control Word, set E to 3 to initiate the memory cycle and change control to J = 6.

J06L

When the memory access is complete and the Return Control Word is in the B register (EEZL), transfer the address of the Mark Stack Control Word from the B register (B 30 through 16 to S) and initiate a memory cycle to access the MSCW.

J07L

With the MSCW in the B register, transfer the proper address to the R register (B 42 through 34 to R) and restore into the S register the address of the top word in the stack in core memory (F to S), and change control to J = 11.

J08L

With no memory access in progress (EEZL) construct an Initiate Control Word in the B register. The setting of the TM register into the B register will be of value for the Store For Test operator, but for the Store For Interrupt operator it is a redundant operation.

R + 8 to M, E to 13, J to 9

Prepare to store the Initiate Control Word in the PRT by setting the M register to R + 8 (octal), the E register to 13 to initiate a memory cycle and change control to J = 9.

J09L

Entry to this J count is to terminate the operator. When EEZL is true then the Initiate Control Word constructed at J08L is stored and the operator is allowed to continue.

M to 0, R to 0, MSFF to 0, SALF to 0

When EEZL is true then reset the above registers in preparation for going into Control State and subsequent MCP action.

TROF to 0, T to 0, PROF to 0

If this operator is being executed in Processor #2 as indicated by PK1L' then set the Processor to an idle state by clearing the T register and resetting PROF and TROF.

J to 0

If this is not the store for test operator then clear the J register in preparation for the next operator.

T to IINL

If this is Processor #1 and this is not the Store For Test operator (T09F' • PK1L) then set the Interrogate Interrupt Operator into the T register in order to determine which interrupt exists.

TM8F to MR0F, TM to 0, TM7F to MW0F

If the operator is the Store For Test then the TM register is cleared. The setting of MR0F and MW0F is redundant as TM7F and TM8F are off at this time.

J to 14, E to 5

If the operator is the Store For Test and this is Processor #1 (T09F • PK1L) then change control to J = 14 and set the E register to 5 to initiate a memory access of cell zero in core memory. This cell will contain a Data descriptor for the test routine to check the results of a test operation.

CWMF' to 0

If this is not a program initiated Store For Interrupt (Q07F) or if this is the Store For Test operator (T09F) then place the Processor into word mode.

J10L

Entrance to this J register setting is from J01L if the A register is valid when the Processor is in character mode or if this is the Store For Test operator. The purpose of this J count is to store the A register in the stack prior to storing the B register. Increment the S register, set E to 10 to initiate a memory cycle, and change control to J = 1.

J11L

When no memory access is in progress (EEZL) clear the B register in preparation for the construction of an Initiate Control Word and change control to J = 8.

J14L

This J register setting can be attained only if this is the Store For Test operator and Processor #1 (ref. J09L description). At J09L a memory cycle was initiated when the J register was set to 14. This memory cycle accesses cell zero of core and will

place a data descriptor into the B register. This data descriptor contains a branching address for the Processor to enter a test evaluation routine.

E16F to 1

When the memory access is terminating (MROF) set E16F in order to initiate a fetch of a new program word into the P register.

B (15 thru 1) to C, L to 0, EXIT

When the memory access is completed (EEZL) transfer the 15 low order bits of the data descriptor in the B register to the C register, clear the L register as the first syllable to be executed is syllable zero, and terminate this operator (EXIT).

K, V, G, and H to 0

Because the Processor will be entering a new program clear the K, V, G, and H registers as these registers are not cleared with the Syllable Execute Complete Level.

5.5 INITIATE P1
INITIATE TEST

INIL - 4111
IFTL - 5111

PURPOSE

The purpose of the initiate operators is to initiate a Processor into the execution of a program. This starting into some program can be via one of three sources:

1. Programmatic execution of this operator by the MCP (Processor #1 in control state) to initiate Processor #1 into the execution of an object program in normal state.

2. Initial load, via the LOAD BUTTON on the operator console, will result in the load Initiate P1 operator being set into the T register of Processor #1. The object program will run in control state.

3. If the Processor being initiated is Processor #2, then the Initiate P1 operator is set into the T register of Processor #2 to initiate P2 into the execution of an object program in normal state.

The Initiate For Test operator is used in conjunction with the B 5500 Maintenance Test Routine. This operator can only be used programmatically to initiate Processor #1 into a particular test function.

Both the Initiate P1 and Initiate Test operators require the Processor in control state for programmatic use of these operators.

SUMMARY OF OPERATION

Initiate P1: Prior to the programmatic execution of the Initiate P1 operator by the MCP, the top word in the MCP stack must contain an Initiate Control Word designating the program to be initiated. With the Initiate Control Word in the top of the MCP stack, the Initiate P1 operator will initiate the object program (normal state program) on Processor #1 in the following manner:

1. Distribute the Initiate Control Word located at the top of the MCP stack. Primarily the information found in this control word is the S register setting to point the S register to the top of the object program's stack, and the CWMF.

2. Obtain and distribute from the object program's stack the Interrupt Return Control Word, Interrupt Control Word, and, if entry to a program in character mode, the Interrupt Loop Control Word and the A and B register data.

3. Initiate a fetch of the first syllable to be executed in the object program.

4. Terminate this operator, placing the Processor in normal state by setting NCSF.

If Processor #1 is initiating an object program in Processor #2 then the Initiate P2 operator in P1 will generate the level APKD in Central Control. APKD will set into the T register of P2 the Initiate P1 operator which in turn will access core address 10 to obtain the Initiate Control Word of the object program being initiated. The remainder of the operation is the same as a program executed Initiate P1 operator.

Another method of hardware generation of the load Initiate P1 operator is with the LOAD BUTTON located on the operator console. Depression of this button will result in the following actions occurring:

1. Generation of the Master Clear Pulse to reset all flip-flops in the system.

2. Initiate an I/O operation, card or drum read, placing the first word read into core address 20.

3. Upon completion of the I/O operation, if it is error free, generate the APKD Level in Central Control.

APKD to Processor #1 will insert the Initiate P1 operator into the T register. This operator will in turn initiate a fetch from core address 20 and start the program just loaded into core memory.

Initiate Test: This operator is used in conjunction with the Maintenance Test Routine. It is used only programmatically, with the Processor in control state, its operation parallels the Initiate P1 operator.

The prime difference between the two operators is that the Initiate Test operator has the capabilities of setting various control registers and flip-flops that the Initiate P1 operator cannot; for example, Y, Z, Q01F thru Q09F, J, MWOF, and MROF. With the ability to set these additional registers and flip-flops, the Initiate Test operator can initiate the Processor in the middle of any operator.

Because the J register, MWOF, and MROF are used by the Initiate Test operator, another register (TM register) is used to temporarily store the configuration of bits to be inserted into them. The TM register is an 8 bit register used in the following manner:

TM (4 thru 1) - Set by the Initiate Control Word bits 17 through 20, used as temporary storage for the J register bits.
TM5F - Set by bit 21 of the Initiate Control Word, contains the status of the Normal Control State Flip-flop (NCSF).
TM6F - Set by bit 22 of the Initiate Control Word, contains the status of the Clock Count Control Flip-flop (CCCF).
TM7F - Set by bit 44 of the Initiate Control Word, contains the status of MWOF.
TM8F - Set by bit 46 of the Initiate Control Word, contains the status of MROF.

The Clock Control Flip-flop (CCCF), is used in conjunction with SECL/FETCH. Its purpose is to define the method of testing to be done, that is, only a one clock pulse test or a several clock pulse test. If CCCF is set upon entry to a test procedure

then the test will encompass only one clock pulse. If CCCF is reset upon entry to a test procedure then the test will encompass several clock pulses, in fact, it may encompass several operators. To terminate this type of a test the Store For Test operator must be used programmatically. If CCCF is set upon entry to the test procedure then the Store For Test operator will be hardware generated.

DETAILED DESCRIPTION

For the detailed description of the Initiate P1 and Test operators refer to the flow charts, page 3.07.0.

The following description will concern itself with the programmatic execution of the Initiate P1 and the Initiate Test operators. Because the operators are similar in their operations, they will be discussed as one with the differences noted.

J = 0

If the operator is executed with the Processor in normal state (NCSF), the operator will function as a NO-OP; terminate the operator as indicated by the action EXIT. If the Processor is in control state then the actions indicated below the double line will place the top word of the MCP stack into the B register. Fetch is inhibited as this operator will initiate a fetch from a location other than presently specified by the C register. Change control by setting the J register to 3.

J = 3

Unconditionally mark the A and B registers empty (AROF to 0 and BROF to 0). They will be set, if necessary, by the control words that indicate their status upon initiation of a program.

EEZL

With EEZL true there is no memory access presently taking place and an Initiate Control Word is located in the B register. Transfer the address portion of the B register (15 through 1) to the S register. This sets the S register to the address of the top word in the object program stack, which at this time contains an Interrupt Return Control Word. The mode of operation being initiated is set into the Character Word Mode Flip-flop (B16F to CWMF). Set the E register to 3 to access the Interrupt Return Control Word addressed by the S register; change control to J = 4.

T10F

If the tenth bit of the T register is set, the operator in the T register is the Initiate For Test. Transfer the indicated bits to the TM register, the Z register, the Y register, and the Q register. The TM register will temporarily store the contents of the J register, NCSF, MROF, MWOF, and CCCF that are required upon entry to the test cycle being initiated. The Q register information will be set into the nine low order bits of the Q register.

J = 4

At this J register setting the Interrupt Return Control Word is distributed from the B register.

MROF

At MROF time of the memory read access to obtain the Interrupt Return Control Word, the S register is decremented by one (S-1) to point to the Interrupt Control Word in the stack.

EEZL

Upon completion of the memory access the B register contains the Interrupt Return Control Word; distribute this control word in the following manner:

1. Obtain the address of the first syllable to be executed in the program being initiated (B [15 through 1] to C and B [38 through 37] to L).
2. Set the dial registers to their required configurations (B [33 through 31] to K, B [36 through 34] to G, B [41 through 39] to V, and B [44 through 42] to H).
3. Set the F register to the top Return Control Word or top Mark Stack Control Word in the object program stack (B [30 through 16] to F).

Set the E register to 3 to obtain the Interrupt Control Word from the stack and change control to J = 5 in preparation for the distribution of the Interrupt Control Word.

Also set E16F to 1 to start a fetch; the C register now contains the proper address for the fetch.

T1OF + CWMF

If the operator is the Initiate For Test operator, or if we are initiating a program in character mode, obtain the correct status of BROF from bit 46 of the Interrupt Return Control Word.

J = 5

MROF

At MROF time of the memory access to obtain the Interrupt Control Word from the stack, decrement the S register to point to the next lower word in the stack. This next word in the stack will be an Interrupt Loop Control Word for character mode entry or a data word for word mode entry.

EEZL

When EEZL is true then the B register contains the Interrupt Control Word; distribute this control word in the following manner:

1. Set the Variant Flip-flop and the Sublevel Flip-flop to their required status as designed by bits 25 and 31 of the B register.

2. Set the Mark Stack Flip-flop as indicated by bit 32 of the B register; if MSFF is turned on then the F register is presently addressing a Mark Stack Control Word in the stack, if MSFF is left off then the F register is presently addressing a Return Control Word in the stack.

3. From bits 42 through 34 of the B register set the R register to its proper configuration. If entering a character mode program then the R register is being set to a tally amount; if entering a word mode program then the R register is set to the base address of the PRT of the object program.

CWMF + T1OF

If the Processor is initiating a program in character mode or if the operator in the T register is the Initiate For Test operator, then obtain the configuration for the M register from the 15 low order bits of the Interrupt Control Word and the N register from the B register bits 19 through 16. Set the E register to 3 in order to access the Interrupt Loop Control Word and change control to J = 6.

$\overline{\text{CWMF}} \cdot \overline{\text{T1OF}}$

If the program being initiated is not in character mode and the operator is the Initiate P1 operator then the operator is complete. Place the Processor into normal state (NCSF to 1) and terminate the operator with the Syllable Execute Complete Level (EXIT).

J = 6

At MROF time of the memory access to obtain the Interrupt Loop Control Word decrement the S register to point to the next lower word in the stack.

EEZL

With the memory access completed, the Interrupt Loop Control Word is in the B register; obtain the status of AROF from bit 46 (B46F to AROF) and enable the actions below the double line to take place.

Transfer the 39 low order bits of the B register to the X register. If entering a character mode program then the destination string address will now be placed into the X register bits 30 through 16. Change control by setting the J register equal to seven.

BROF + T1OF

If BROF had been set by the Interrupt Return Control Word or if this is the Initiate For Test operator then set E to 3 to load the B register with a data word.

J = 7

If a memory cycle had been done to load the B register with a data word, count down the S register at MROF time of the memory cycle.

EEZL • $\overline{\text{AROF}}$ • $\overline{\text{T1OF}}$

With no memory access taking place (EEZL) the operation of the operator is allowed to continue. If the A register is marked as being empty (would have been set by the Interrupt Loop Control Word at J = 6) and if the operator in the T register is the Initiate P1 ($\overline{\text{AROF}}$ • $\overline{\text{T1OF}}$) then there is no need to load the A register with a data word. Interchange the S register with bits 30 through 16 of the X register. This action places the destination string address of the character mode program into the S register and the address of the top word of the stack into the Loop Control Word in the X register.

Place the Processor into normal state (NCSF to 1) and terminate the Initiate P1 operator with the Syllable Execute Complete Level.

EEZL • (AROF + T1OF)

If at J = 6 AROF had been set by the Interrupt Loop Control Word or the operator in the T register is the Initiate For Test (AROF + T1OF), set the E register to 2 to load a data word into the A register and change control by setting the J register to 8.

J = 8

At MROF time of the memory access to load a data word into the A register decrement the S register to point to the next lower word in the stack. Upon completion of this memory cycle the remaining actions at this J register can occur.

EEZL • CWMF

If the program being executed is in character mode then interchange the S register with bits 30 through 16 of the X register. This action places the character mode destination string address into the S register and the address of the top word of the stack into the Loop Control word in the X register.

EEZL • $\overline{\text{T1OF}}$

If the operator in the T register is the Initiate P1, place the Processor into normal state (NCSF to 1) and terminate the operator with the Syllable Execute Complete Level (EXIT).

EEZL • T1OF

With the Initiate Test operator in the T register the actions indicated below the double line are enabled. Set, from the TM register, the J register, NCSF, and CCCF. Reset TROF to mark the operator in the T register as invalid and to allow the SECL/FETCH flow to initiate the test operation.

TM6F • $\overline{\text{CWMF}}$

With TM6F in J the set state and the test procedure in word mode then decrement the S register one additional count. This is necessary as a part of the Maintenance Test Routine procedure.

IFTLD • $\overline{\text{CCCF}}$

This portion of the flow is utilized only for the Initiate For Test operator. The term IFTLD will be true with the Initiate For Test (5111) code in the T register. After TROF is reset at J = 8, if CCCF is also reset, then at the time that operator "X" (the operator under test) is transferred from the P register to the T register TM6F is set to 1. TM6F is used in the SECL/FETCH operation to allow for a count up for fetch if the L register was equal to three. This logic is necessary as the termination of the Initiate Test operator does not use the normal Syllable Execute Complete Level to generate the count up for fetch.

The remainder of the Initiate P1 flow has to do with a hardware generated Initiate P1 operator. The operator will be forced into the T register if this is Processor #1 and the LOAD Button on the console is depressed or if this is Processor #2 being initiated by Processor #1. In either case the Initiate P1 operator is forced into the T register via the level APKD from Central Control. With APKD true to the Processor from Central Control the following actions will occur:

APKD

T to INIL, TROF to 1

With the level APKD and $\overline{\text{TROF}}$, indicating that the T register is empty, set the initiate P1 operator into the T register and mark the T register as valid by setting TROF to 1. $\overline{\text{TROF}}$ is used here in case Processor #1 should attempt to initiate Processor #2 when Processor #2 is already busy.

J to 2

If this is Processor #2, as indicated by the level $\overline{\text{PK1L}}$, then change control to J = 1, this Initiate P1 operator was generated via the LOAD Button.

J = 2

To get to this J register setting this must be an initiation of Processor #2; set M04F to 1 to address core address 10, set E to 5 to place the Initiate Control Word into the B register from core address 10, and change control to J = 3 to continue the initiation of the Processor in the same manner as the programmatic execution of the Initiate P1 operator.

J = 1

Entering this J register setting is due to depression of the LOAD Button on the operator console; set C05F to place a 20 into the C register and set E16F to 1 to initiate a memory fetch access. The word placed into the P register with this fetch access will be the first word read from either the Card Reader or the Drum Memory, depending upon the state of the Card Load/Drum Load switch on the operator console. The Syllable Execute Complete Level is true (EXIT) to terminate the operator in a normal manner. Under this condition, the Processor is initiated in control state.

5.6 INITIATE P2
INITIATE I/O

PTOL - 4211
IOOL - 4411

PURPOSE

The purpose of these operators is to allow the MCP to initiate an object program in normal state on the Processor designated as Processor #2, or to initiate an I/O operation, depending upon the syllable code in the T register. Both of these functions can be accomplished only with the Processor in Control State; if the Processor is in Normal State then the operators are treated as NO-OPs.

SUMMARY OF OPERATION

Initiate P2: Prior to the execution of the Initiate P2 syllable (4211), the top word of the stack must contain the Initiate Control Word. The Initiate Control Word would have been programmatically obtained from R + 10 of the program to be initiated. The Initiate P2 operator will store this word into absolute core address 10, then delete this control word from the top of the stack.

After storing this control word the Processor will generate and send two levels to Central Control, Commence Timing Level (UCMTD) and Commence I/O Level (UCMID). UCMTD (CMTL in Central Control) going true will set the Commence Timing Flip-flop (CMTF) in Central Control. CMTF is combined with UCMID (CMIS in Central Control) to initiate either Processor #2 or an I/O operation. UCMID (CMIS in Central Control) is anded with CMTL, if the Initiate P2 operator is in the T register, and term UCMID will be false, developing the condition of CMTF • CMIS/ which will generate APKD to Processor #2 to initiate the second Processor.

Initiate P2 = CMTL • $\overline{\text{UCMIL}}$

Once CMTF is set in Central Control, the Initiate P2 operator in Processor #1 is allowed to terminated.

Initiate I/O: The operation of the Initiate I/O operator is identical to the Initiate P2 operator. Only now the top word in the stack will be the address of an I/O descriptor; this also is stored in core address 10. With the Initiate I/O operator the term UCMIL (CMIS) will be true, (ANDED with CMTL), to Initiate an I/O unit.

Initiate I/O = CMTL • UCMIL

At this time the level APKD in Central Control will not be developed, instead, ADNS (Admit Descriptor Now) is generated and sent to the I/O channel.

DETAILED DESCRIPTION

JOOL

S - 1

The S register is decremented if MROF goes true while at J = 0 of this operator.

This action is a part of stack adjustment and will occur only if it was found necessary to push up the stack into the A register.

EXIT

If either of these operators is executed on a Processor in normal state (NCSF) then terminate the operator and allow no subsequent actions to occur. Either operator will be a NO-OP in normal state.

If the Processor is in control state ($\overline{NCSF}$) then allow the actions indicated below the double line to occur.

MOLF to 1

By setting bit four of the M register an address of 10 is placed into M. This address is used to access cell 10 in order to store an Initiate Control Word (if the operator is the Initiate P2) or the address of the I/O descriptor (if the operator is the Initiate I/O) from the top of the stack.

E to 12

With AROF on and EEZL true, the top word of the stack is located in the A register and there is no memory access taking place due to stack adjustment; set the E register to 12 to store the word in the A register into core address 10 using the M register as the addressing register.

E to 13, BROF to 0

In this case the top word of the stack is located in the B register ($\overline{AROF}$ • BROF); set the E register to store this word into core address 10, mark the B register as empty to delete this word from the stack.

E to 2, AROF to 1

If both the A and B registers are empty ($\overline{AROF} \bullet \overline{BROF}$) then set E to 2 to load the A register; mark the A register as being occupied. This action is a normal stack push-up.

J to 1

If either the A or B register is occupied and there is no memory access taking place (due to stack adjustment) then change control to J = 1.

J01L

AROF to 0

Unconditionally mark the A register as empty to delete the top word from the stack. This word was just stored into core address 10.

CMTL to 1

When the memory access to store the top word of the stack into core address 10 is completed (EEZL), allow the level CMTL in Central Control to go true. CMTL will set CMTF which will, depending upon the operator in the T register, initiate either Processor #2 or an I/O unit.

UCMTD = J01L • P1OLD • EEZL generates CMTL in Central Control
UCMID = CAGT52 • T09F • CWMF/ generates CMIS in Central Control

UCMTD is the Commence Timing Level in the Processor which is true with no memory access (EEZL), the Initiate P2 or Initiate I/O operator in the T register (P1OL), and the J register equal to 1. UCMID is the Initiate I/O operator code (CAGT52 • T09F) and the Processor in word mode (CWMF/). This level will be true only if the operator in the T register is the Initiate I/O, its purpose is to distinguish between the two operators.

J to 2

Once the Commence Timing Flip-flop (CMTF) is set in Central Control the Processor can complete the operation of this operator. Change control by setting the J register to 2.

J02L

Unconditionally terminate the operator with the Syllable Execute Complete Level.

5.7 HALT P2 SYLLABLE

HP2L - 2211

PURPOSE

The function of this operator is to halt processing on the Processor designated by D & D as number 2. This operator may only be used in Control State. If the Processor is already halted, the syllable becomes a NO-OP.

SUMMARY OF OPERATION

There are no prerequisites required of the Stack prior to the execution of this syllable. This operator may only be initiated on the Processor that is running in control state, (which must be the Processor designated as Processor #1) and will be a NO-OP if executed in Normal State. When this syllable is executed, a level will be gated to Central Control to set the Halt Processor 2 Flip-flop (HP2F). With this flip-flop set, Processor #2 will, at the completion of the current syllable being executed, do an automatic Store For Interrupt. This is accomplished through setting the "T" register of Processor #2 to the syllable code of SFIL (3111), and then executing that syllable. At the completion of the store operation, the Processor is left idle. This idle condition also marks the Processor not busy so that it may be reinitiated at a later time by an Initiate P2 syllable.

DETAILED DESCRIPTION OF SYLLABLE

JOOL

If Processor #2 has been halted and not reinitiated prior to the execution of this syllable (HP2F being set) or this Processor is in Normal State, SECL is immediately developed and the syllable is terminated.

SECL Q21IO1 = (JOOLT21 • HP2LD1 • DHP2X) + (HP2LD1 • J48LD1), where the latter logic is to terminate the syllable if this Processor is in Normal State. If the Processor is in Control State, then the level UHP2D1 (UHP2L) is developed and gated to Central Control where on the next clock pulse it will set HP2F in C.C.

UHP211 = JOOLT21 • HP2LD1 • NCSFS1/
HP2F = PAHP2L • PA1L (Processor "A" is Processor #1
+PBHP2L • PB1L (Processor "B" is Processor #1

HP2F in Central Control will gate the level IO4L true in the Processor designated as #2. The logic for this level is as follows:

IO4LC1D = DHP2X • DPK1X/ • 939LS1

where DHP2X is HP2F from Central Control.

DPK1X/ indicates this is the Processor that is designated as number 2, and

939LS1 = US13S1/ • IFTLD1/ • CCCFF/

At the completion of the syllable being executed on Processor #2, IO4L will cause the "T" register of Processor #2 to be set to the SFIL syllable code (3111) through the SECL/FETCH flow chart (3.13.0 - Branch on Interrupt). The subsequent operation of that Processor is to execute the storing of the various control words and then idle due to PROF/, NCSF/, TROF/, and E16F/ at the termination of the SFIL syllable. With this combination of flip-flop settings, the Processor is marked as not busy, and therefore, is available to be reinitiated by Processor #1 at some later time.

The clock pulse in Processor #1 that finds HP2F set, will then generate SECL true and allow the syllable to terminate.

5.8 READ TIMER OPERATOR

RDTL - 0411

PURPOSE

The purpose of the read timer operator is to interrogate the real time clock and to set the A register to the count contained in the counter flip-flops.

SUMMARY OF OPERATION

When a program is entered into the system, it contains an estimated running time. This estimate may be for program running time, I/O running time, or both. The MCP keeps track of the actual running times and compares the time against the estimated running time. The real time clock flip-flops, located in Central Control, count from 0 through 63 at a rate of 60 counts per second. Each time the counter recycles a time interval interrupt occurs. This allows the MCP to record the running time in increments of 64/60 seconds. To record the actual time the MCP must interrogate the state of counter flip-flops every time a program (or I/O operation) is started or terminated. This is accomplished by the Read Timer Operator which stores the configuration of the real time clock into A [6 ⇒ 1] and the status of the Timer Interrupt bit into A07F.

DETAILED DESCRIPTION

J = 0

EXIT

If the Processor is in the Normal State (NCSF), treat this operator as a NO-OP and allow the syllable execute complete level. With the Processor in Control State, as indicated by the Normal Control State flip-flop being off (NCSF'), allow the operation to continue as indicated below the double line.

E to 11, S + 1, J to 1

If both A and B registers are occupied, then initiate a stack adjustment to push down the stack to empty the A register; change control to J = 1 to complete the stack adjustment.

A to 0, J to 2

If the A or the B register is unoccupied, clear the A register and set the J register to two to complete the operation.

A to B, BROF to 1

This action is a portion of stack adjustment that occurs if the A register is full and the B register is empty. Transfer the contents of the A register to the B register and set BROF to mark the B register full.

J01L

With no memory access in process, E register is equal to zero (EEZL), and, as entry into this state was only achieved if both A and B had been occupied and a store B was initiated, the contents of A are transferred to B. The A register is cleared and the J count is advanced by one to change control to J02L.

J = 2

Transfer the contents of the real time clock flip-flops in Central Control, to the least significant 6 bits of the A register, where it can now be used as the MCP specifies. Also transfer the status of CCI03F to A07F.

Mark the A register as occupied and terminate the operation by enabling the syllable execute complete level.

5.9 LITERAL SYLLABLE OPERATOR

LTSL - XXX0
XXX4

PURPOSE

To set a literal value of 0000 thru 1777 at the top of the stack.

SUMMARY OF OPERATION

Upon entry into the operator, the conditions of the A and B register are checked. If both A and B are loaded, B is placed in the stack, A is placed in B, and the literal placed in the least significant bits of the A register. If the A register is loaded and B empty, A is placed in B and the literal in the least significant bits of A. If A is empty, the literal is placed in the least significant bits of A by transferring the 12 MSD bits of the T register to the 10 LSD bits of the A register.

DETAILED DESCRIPTION

J = 0

S + 1, E to 11, BROF to 0

If both the A and B registers are occupied (AROF • BROF), increment the stack address plus one (S + 1). Initiate a store B memory access by setting the E register to 11, and mark the B register as empty (BROF to 0).

A to B, BROF to 1

If the A register is loaded and the B register empty (AROF • BROF') and no memory access is in process (EEZL), transfer the contents of A to B and mark B as occupied.

If either the A register or the B register are empty (AROF' + BROF') and no memory access is in process (EEZL), clear the A register bits 11 through 48 and transfer T [12 through 3] into A [10 through 1], then terminate the operation (EXIT).

Mark the A register as occupied (AROF = 1) if either A or B are empty.

5.10 OPERAND CALL
DESCRIPTOR CALL

OCSL - XXX2 - XXX6
DCSL - XXX3 - XXX7

PURPOSE

To locate, via R, F, or C relative addressing, a word in core memory and place this word at the top of the stack. If the operator is the Operand Call the word placed on top of the stack is an operand or control word. If the operator is the Descriptor Call then the word placed on top of the stack is a data descriptor. If either operator encounters a program descriptor then entry is made to a subroutine.

SUMMARY OF OPERATION

The Operand Call and Descriptor Call operators are identified by the two low order bits of the operator in the T register. If T02F is set then the operator is either the OCSL or DCSL. T01F is used to distinguish between the two operators. With T01F • T02F the operator is the Descriptor Call, with T01F' • T02F the operator is the Operand Call.

The remaining bits of the operator in the T register will indicate the type and amount of relative addressing. Primarily, the three high order bits of the T register (T12F, T11F, and T10F) along with SALF (Sub Level Flip-flop) and MSFF (Mark Stack Flip-flop) are used to indicate the type of relative addressing. The remaining bits of the T register (3 ⇒ 9) are used to give the amount of relative addressing. The three high order bits in T register, depending upon their status, can also be used with T03F ⇒ T09F to give the amount of relative addressing. For a description of the various possible combinations of SALF, MSFF, T12F, T11F, and T10F, refer to Figure 5.10-1. This table indicates, for each of the possible conditions, the base register, the index sign (+ or -), the index bits, and the maximum amounts of relative addressing.

There are several possible combinations of Operand/Descriptor Call operations which may occur. These possible operations depend upon the word accessed by relative addressing. The following is a brief description of each of these operations:

OPERAND CALL

1. If the word accessed is an operand (A48F') or a control word (A48F • A47F • A45F') then this word is placed in the A register and the operation is terminated.

2. If the word accessed is a data descriptor (A48F • A47F') then the presence bit (A46F) is checked. If A46F is off then set the presence bit interrupt and terminate the operation. If A46F is on, then access the word addressed by the 15 low order bits of the descriptor. If this second word accessed is an operand, then place it in the A register and terminate the operation, but if it is another descriptor, set the flag bit interrupt and then terminate the operation.

NOTE

Any reference made to the setting of any interrupt require that the Processor be in Normal State. If in Control State no interrupt is set but the operation will be terminated.

DESCRIPTOR CALL

1. If the word accessed is an operand or control word then place in the A register (top of the stack) a present data descriptor with an address pointing to the operand or control word accessed.

2. If the word accessed is a data descriptor place it atop the stack (A register) and terminate the operation. If the presence bit is off, set the presence bit interrupt.

OCSL/DCSL - PROGRAM DESCRIPTORS

Accessing of a program descriptor by either the Operand Call or the Descriptor Call will result in the same course of action. The only difference is in the Return Control Word, bit 46 will be set if the operator is the Descriptor Call Syllable.

When the program descriptor is accessed, the presence bit of the program descriptor is checked. If it is off then the presence bit interrupt is set and the operation is terminated. If the presence bit of the program descriptor is on then one of three actions will take place:

1. If a Program descriptor is referenced and it requires parameters which have not been supplied (MSFF' • A43F), or character mode is called for but parameters are not specified (A44F • A43F') then terminate the operation with the program descriptor in the A register.

2. If a Program descriptor is referenced and parameters are required (A43F) and are present (MSFF is set) then a Return Control word is stored, and the program is branched to the subroutine called for by the program descriptor.

3. If a Program descriptor is referenced and parameters are not required and entry is to Word Mode, then a Mark Stack Control Word and a Return Control Word are stored and the program is branched to the subroutine called for by the program descriptor. This is referred to as the spontaneous entry to a subroutine.

WORD INDEXING

Any time a data descriptor is accessed, regardless of the operator, a check is made of the word count field of the descriptor (bits 31 ⇒ 40). If the word count equals zero then the subsequent operation of the operator is as previously described. If, however,

the word count field is unequal to zero, then the address portion of the data descriptor is indexed by the next word down in the stack. If the value of this word is equal to or greater than the word count of the descriptor then no indexing takes place; the invalid index interrupt is set if the Processor is in normal s state and the operation is terminated.

DETAILED DESCRIPTION

The detailed description of the OCSL and DCSL syllable makes use of the block diagram in Figures 5.10-2 through 5.10-5 and the system flows in Figures 5.10-6 through 5.10-9. The numbers at the block diagram boxes indicate the system flow chart location that does the indicated actions. For example, in Figure 5.10-2, just below the word start, is the box with J0-J1 indicated. The actions described in this box take place at J = 0 and J = 1 of the OCSL/DCSL operator flow. If, as in the next box down, the numbers read as J2-1 then the system flow location is at J = 2, line 1 as shown in Figure 5.10-7.

The relative addressing at the start of the Operand/Descriptor Call (refer to Figures 5.10-2 and 5.10-3) is accomplished at J0 and J1. Referring to these J counts, Figure 5.10-6, all of the actions labeled A occur only if stack adjustment is required. The stack adjustment is a push down to empty the A register. The remaining actions at J = 0 and J = 1 are to accomplish the relative addressing as indicated by the table in Figure 5.10-1.

FLOW CHART INDEX	SALF	T12F	T11F	T10F	MSFF	BASE	INDEX SIGN	INDEX BITS	ADDRESSABLE AREA SIZE IN DECIMAL
K	OFF	-	-	-	-	R	+	T (12 ⇒ 3)	(1,024)
K	ON	OFF	-	-	-	R	+	T (11 ⇒ 3)	(512)
E-G	ON	ON	OFF	-	OFF	F	+	T (10 ⇒ 3)	(256)
H	ON	ON	OFF	-	ON	(R+7)*	+	T (10 ⇒ 3)	(256)
C	ON	ON	ON	OFF	-	C	+	T (09 ⇒ 1)	(128)
E-F	ON	ON	ON	ON	OFF	F	-	T (09 ⇒ 3)	(128)
F-H	ON	ON	ON	ON	ON	(R+7)*	-	T (09 ⇒ 3)	(128)

- Irrelevant setting
* Relative addressing using as the base, bits 16 thru 30 of the word Stored in the programs PRT at R+7.

FIGURE 5.10-1 RELATIVE ADDRESSING TABLE

The Flow Chart Index column in this table makes reference to the appropriate lines on the flow chart as J = 0.

The description of the remainder of the operator flow is via the operational block diagram in Figures 5.10-2 through 5.10-5 with references to the operator flows in Figures 5.10-6 through 5.10-9.

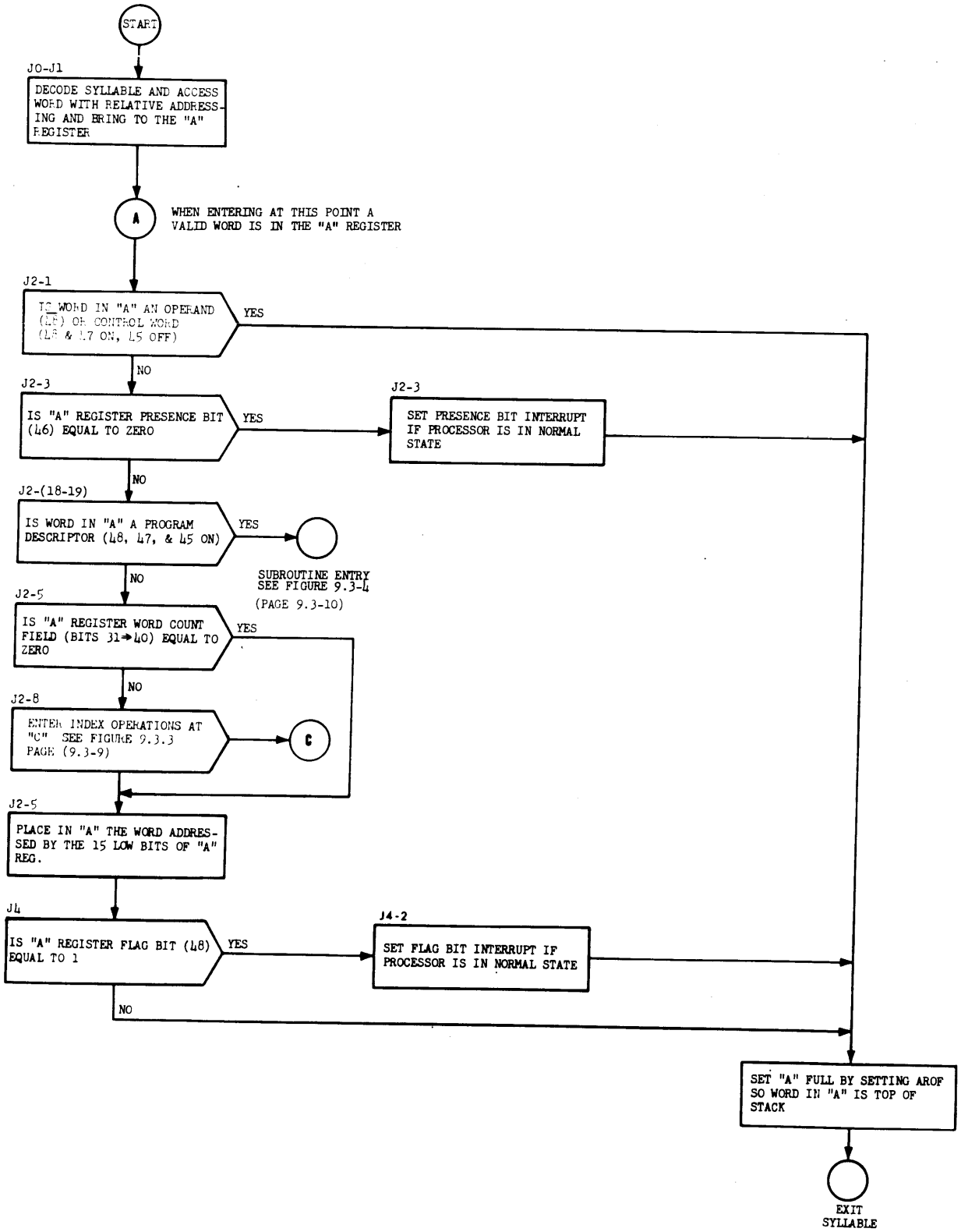


FIGURE 5.10-2 OPERAND CALL BLOCK DIAGRAM

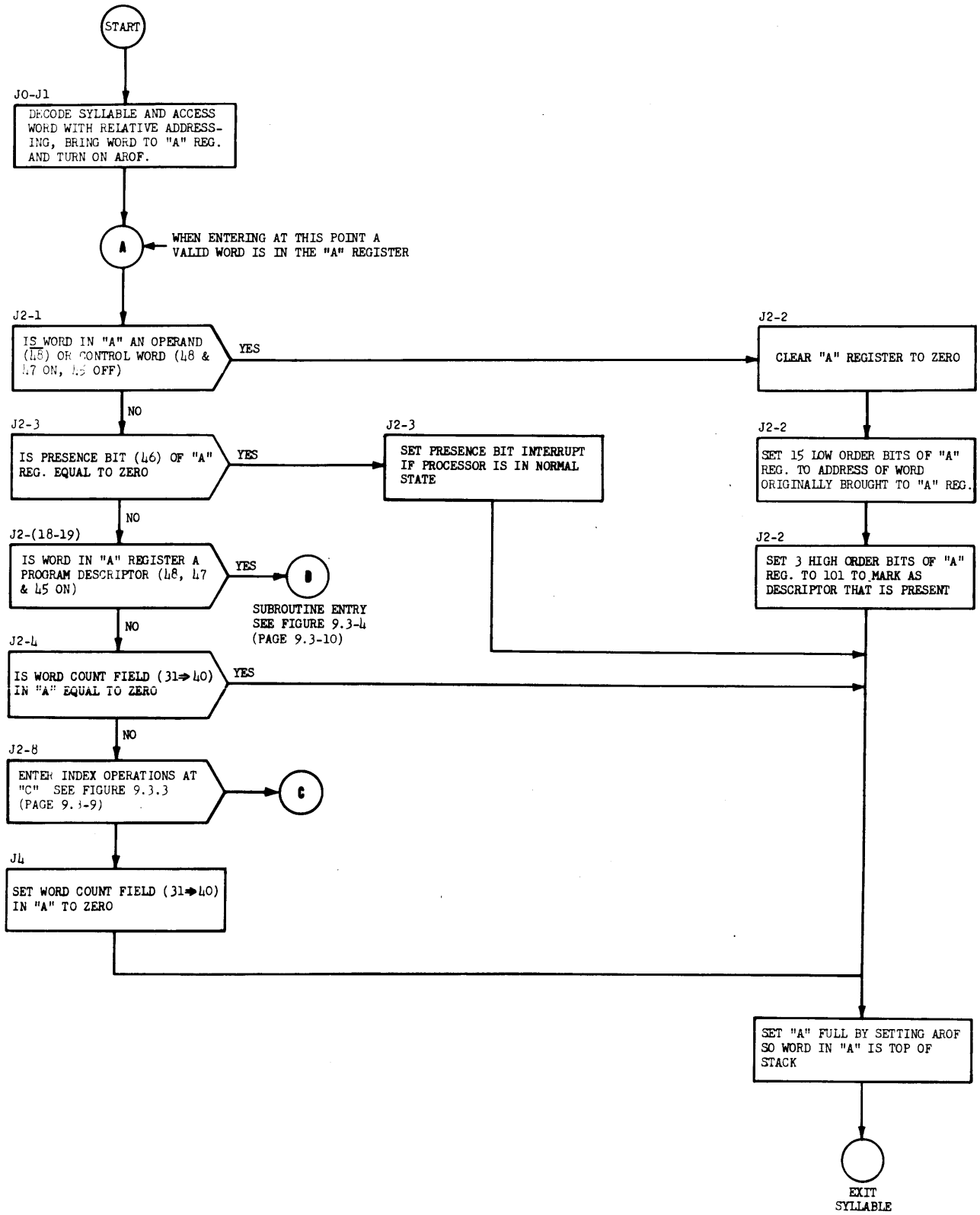


FIGURE 5.10-3 DESCRIPTOR CALL BLOCK DIAGRAM

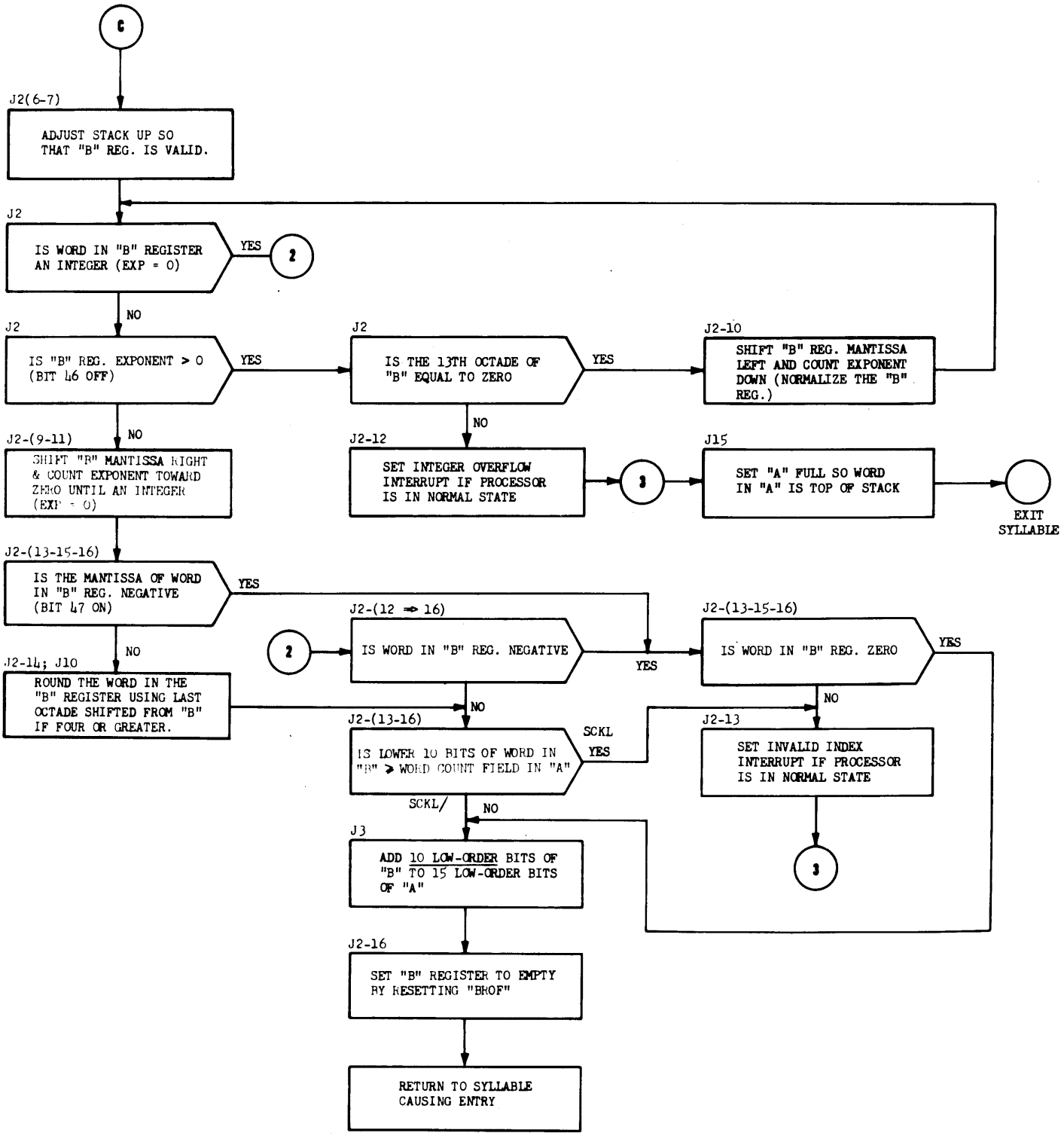


FIGURE 5.10-4 INDEX OPERATIONS - OPERAND AND DESCRIPTOR CALL

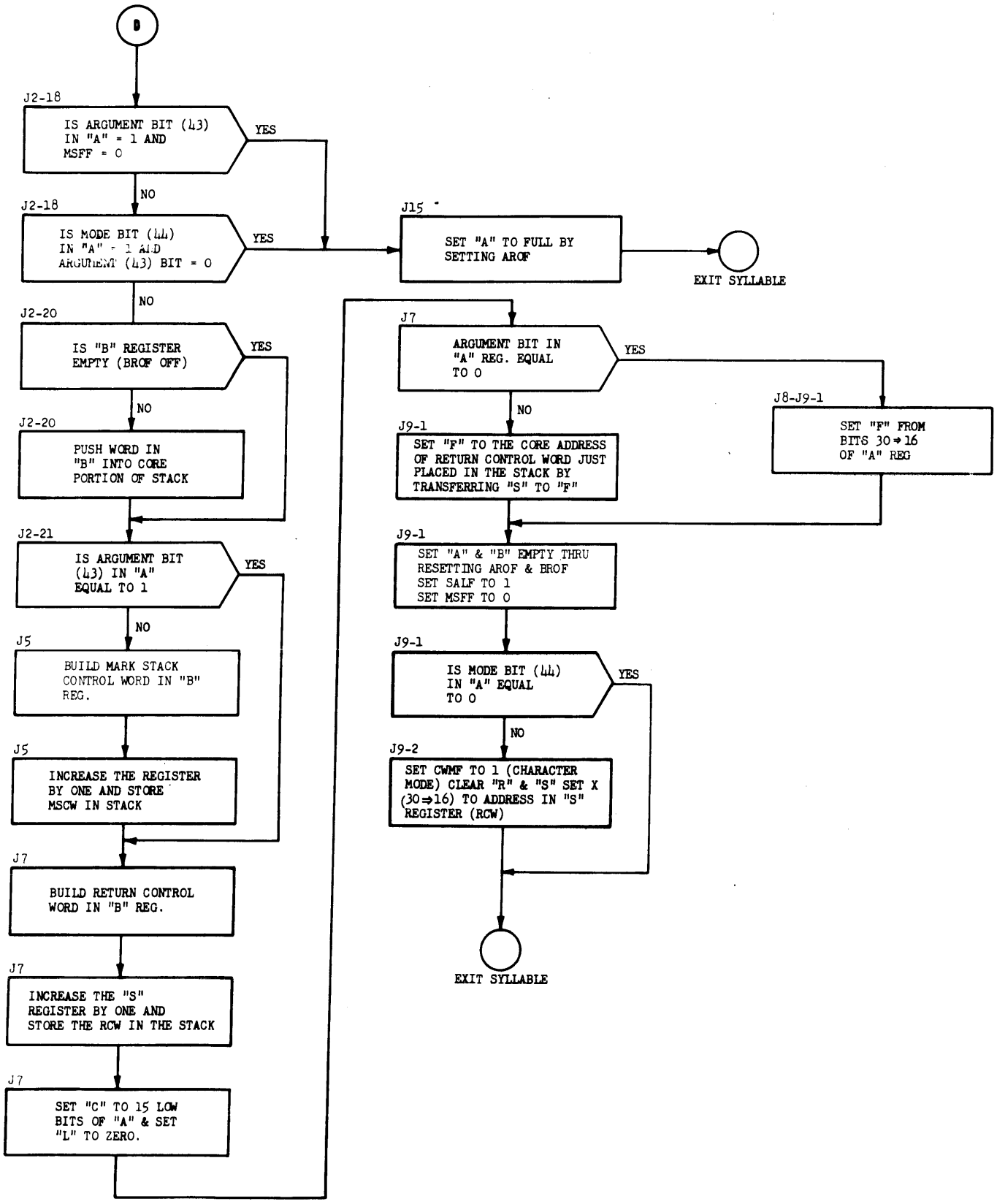


FIGURE 5.10-5 SUBROUTINE ENTRY - OPERAND OR DESCRIPTOR CALL

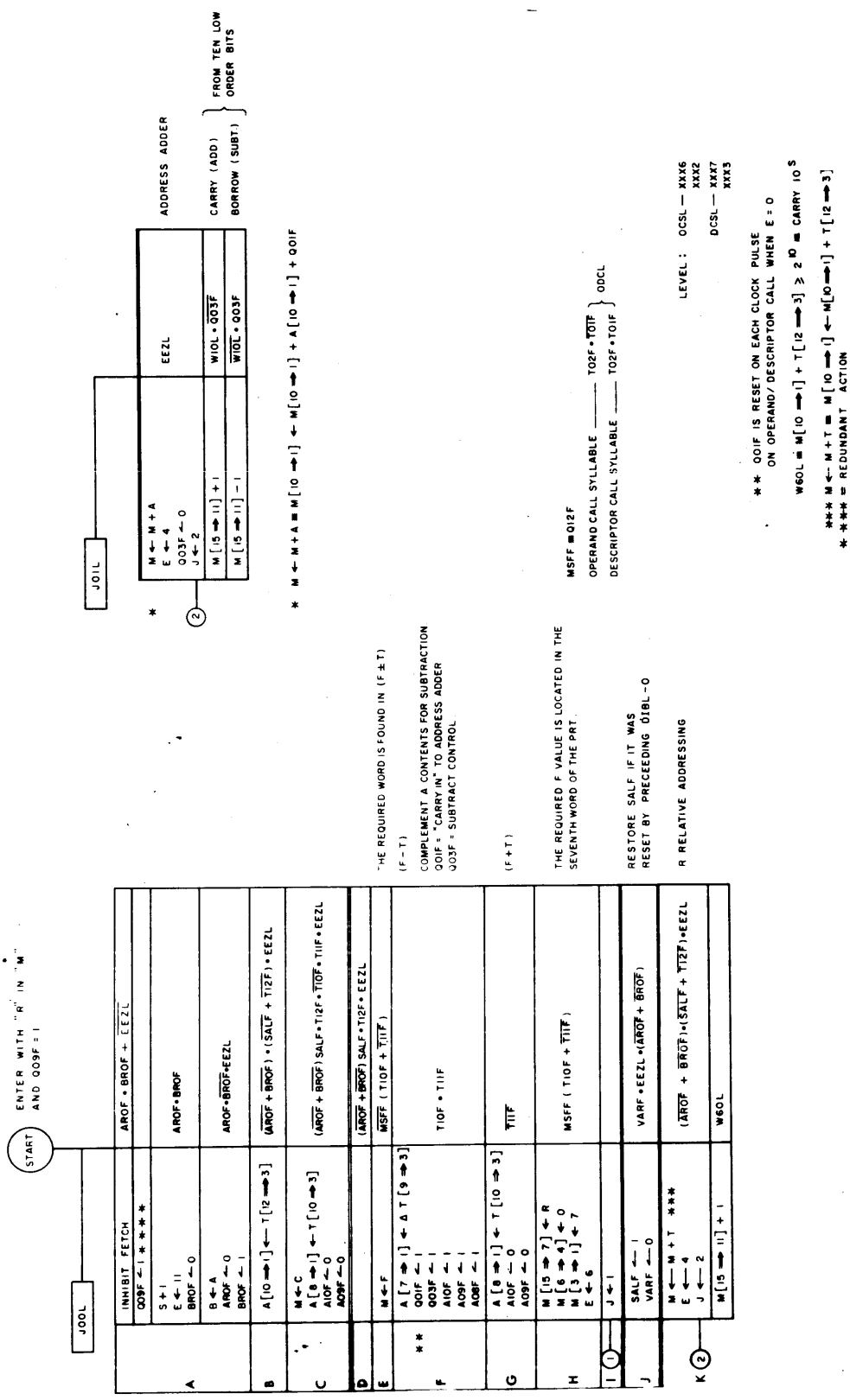


FIGURE 5.10-6 OCSL/DCSL FLOW (1 of 4)

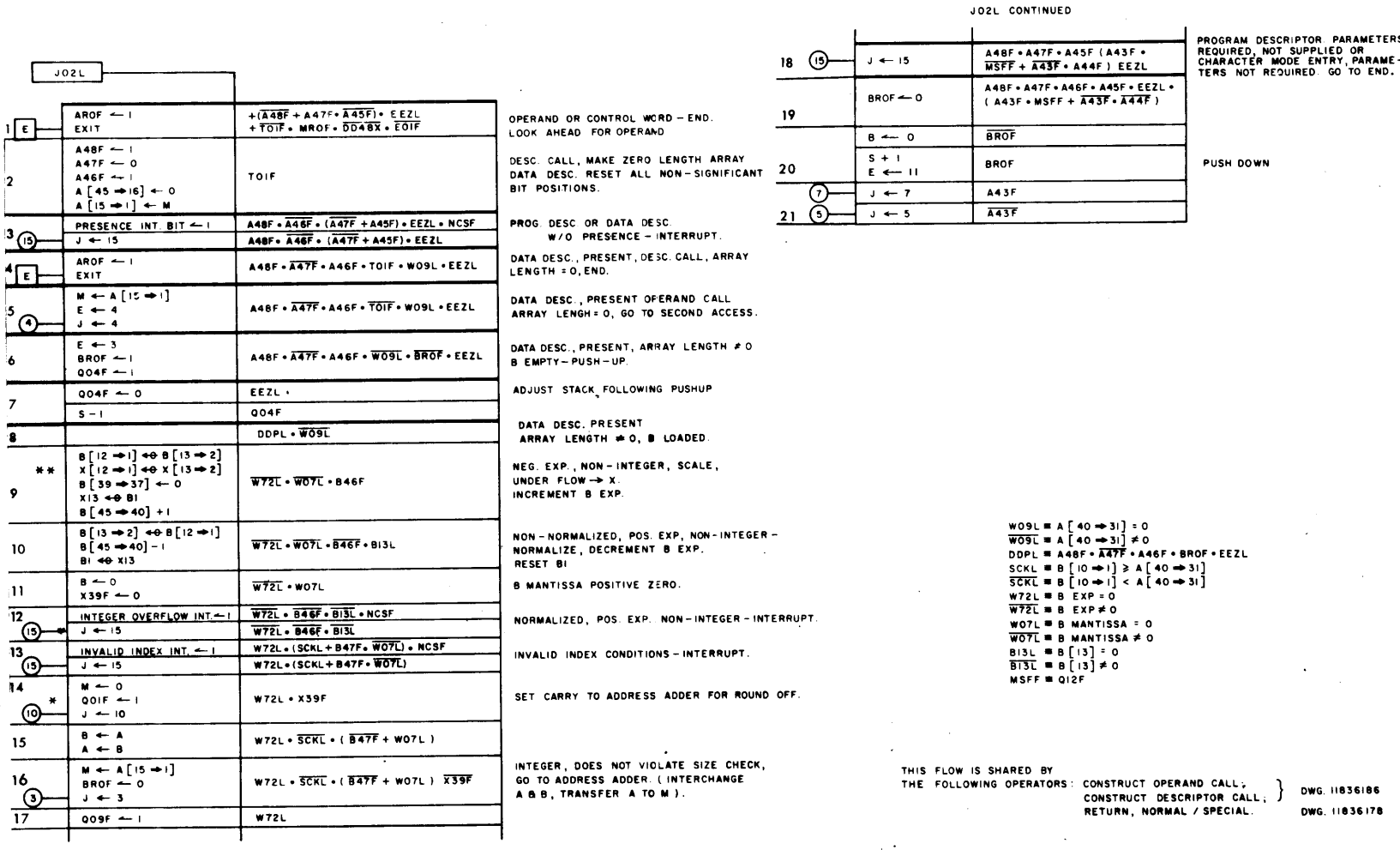


FIGURE 5.10-7 OCSL/DCSL FLOW (2 of 4)

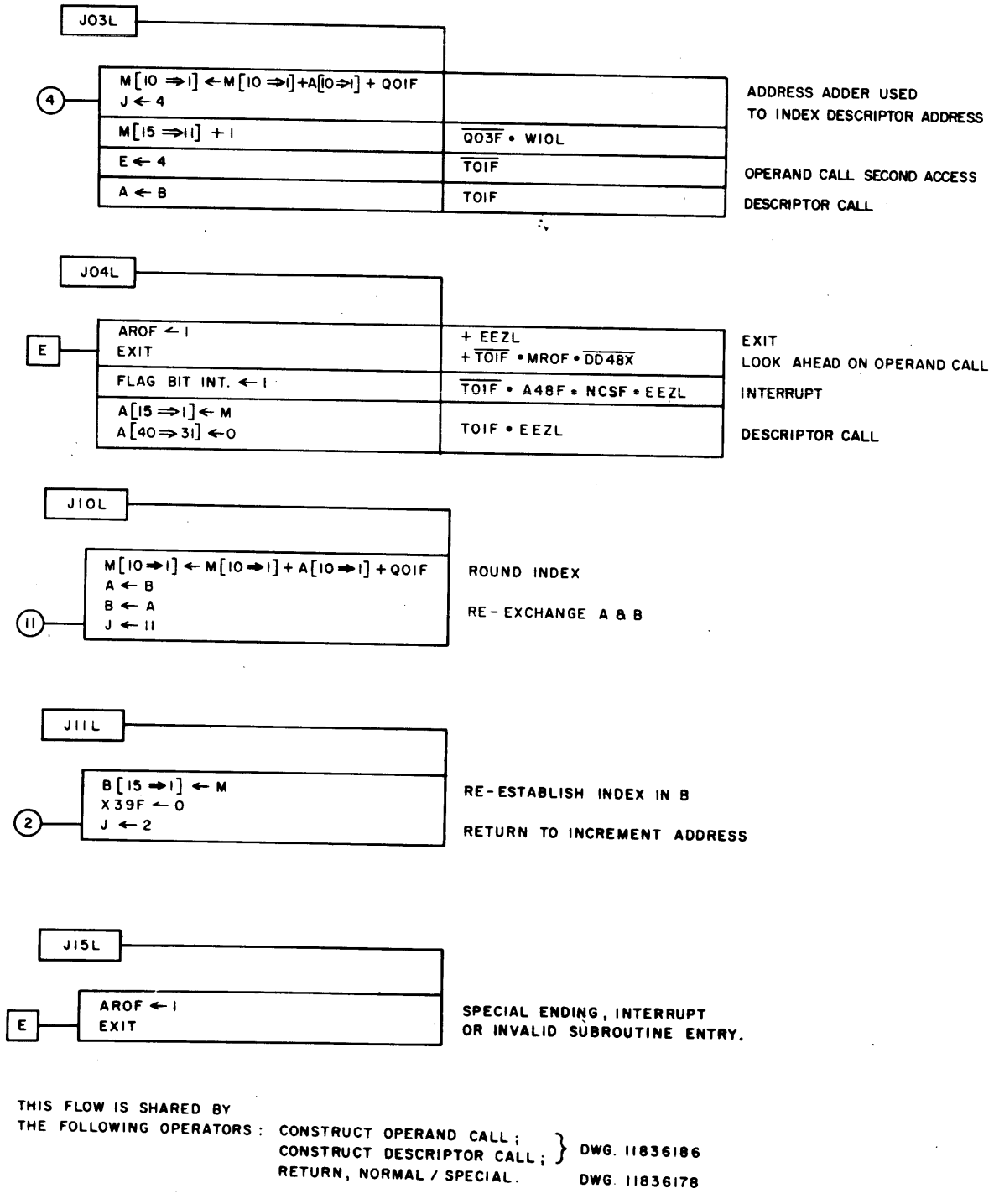


FIGURE 5.10-8 OCSL/DCSL FLOW (3 of 4)

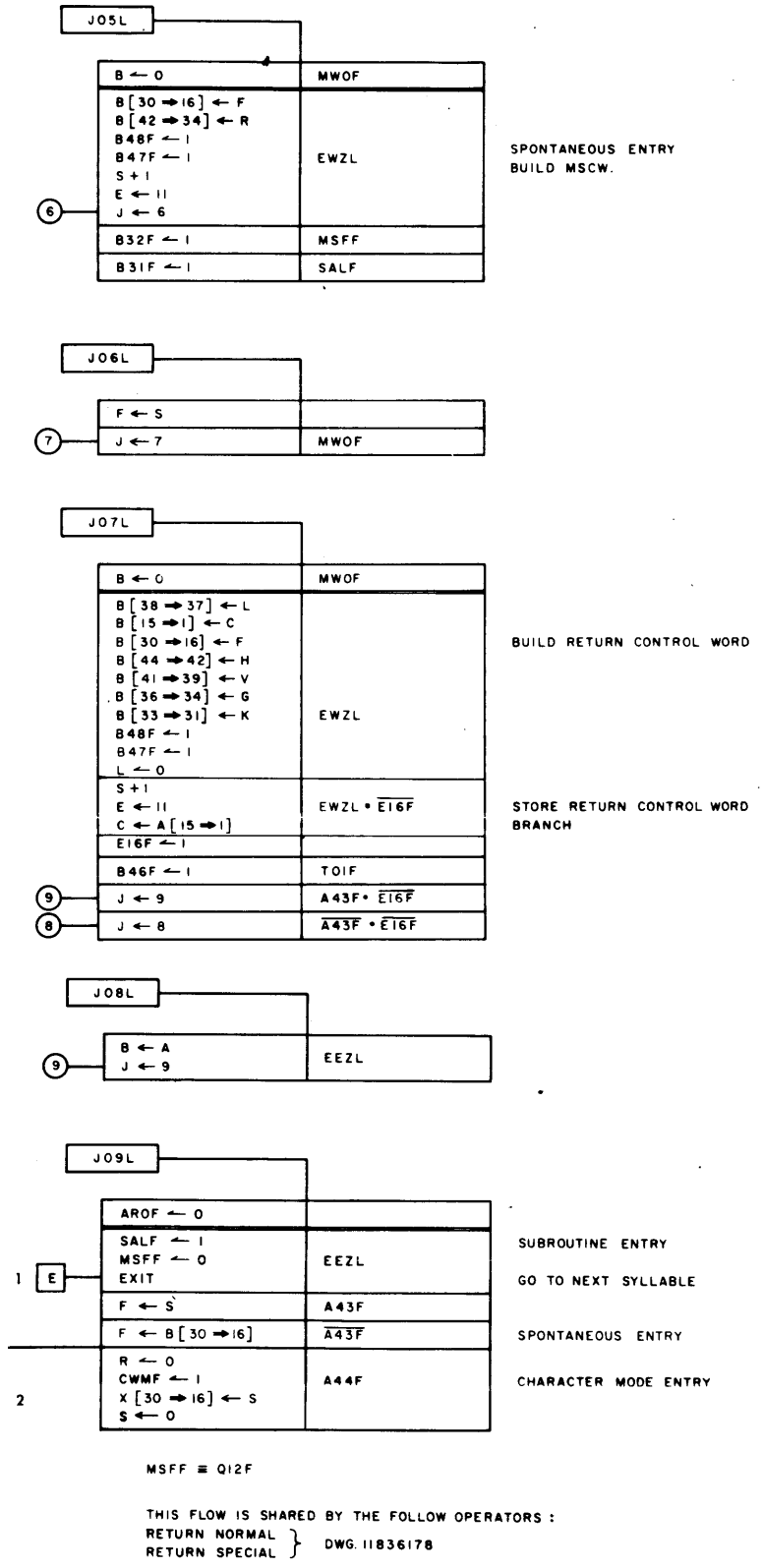


FIGURE 5.10-9 OCSL/DCSL FLOW (4 of 4)

5.11 CONDITIONAL STOP OPERATOR

PURPOSE

To stop clock pulses to a Processor, if maintenance test switch 14 is on. US14X is the Stop Operator switch.

SUMMARY

If CPHL is true and the Stop operator test switch is on, this Processor's clock pulses are inhibited. The Processor will halt until the stop operator test switch is placed in the off position. The next clock pulse will cause an exit of this operator.

DETAILED DESCRIPTION

Stop Clock

The Processor's clock pulses are inhibited at Central Control by the Inhibit Master Clock level (IMCL) going true. This level does not inhibit the master clock. It only inhibits this Processor's clock B.O. and Line Driver. The level IMCL is true when there is a CHPL syllable in the T register and US14X is on.

Exit

When US14X is placed in the off position, IMCL goes false and allows clock pulses to the Processor. The first clock pulses will allow the SECL operation.

5.12 SYLLABLE INTERFACE SECL/FETCH

Although it is not an operator in itself, the SECL/Fetch logic is common to all operators executed in the Processor. It serves to properly terminate the execution of each operator and to obtain the next operator to be executed. If the Processor is in normal state and an interrupt occurs the SECL/Fetch logic will initiate a store for interrupt operation.

Also included in the SECL/FETCH flow chart, flow 3.13.0, is the logic for MTR (Maintenance Test Routine). This logic is used to initiate a test and to terminate a test. The MTR portion of this flow will be described in Section 5.14.

SUMMARY OF OPERATION

The SECL/Fetch logic is concerned with that portion of a program referred to as the Program Segment String, which is that part of a program that contains the program words for the program. Each of these program words is divided into four parts, each part being one operator code of 12 bits. The subsequent action of the fetch logic is to access the Program Segment String, place a program word into the P register, and eventually into the T register for Processor execution. To do this, the Processor uses several registers and flip-flops.

1. C register - used as an address register to address the Program Segment String stored in core memory.

2. P register - used to store a program word of the Program Segment String. This register is a 48 bit register containing four 12 bit operators numbered from zero to three.

3. T register - a 12 bit register that contains the operator being executed by the Processor.

4. L register - a 2 bit register that functions as a binary counter, counting from zero to three and back to zero. Its purpose is to designate which of the four operators in the P register is to be placed into the T register.

5. TROF and PROF - the T Register Occupied Flip-flop and the P Register Occupied Flip-flop are used to indicate when the T register contents are valid and the P register contents are valid.

6. E16F - when set, this flip-flop indicates that a memory fetch access is requested by the Processor to reload the P register. This flip-flop can be set by the SECL/Fetch logic when L equals three or by some of the operators in the Processor (branch operators, return operators, etc.).

7. E17F - is set after E16F is set. This flip-flop actually initiates a memory fetch access. With both E16F and E17F set, the memory read will use the C register as the addressing register and the P register as the information register.

8. MRAF - Memory Read Access Flip-flop is used for a fetch access to indicate the termination of the memory read operation. It is set by timing from the Memory Module being accessed.

Referring to Figure 5.12-1, one word of the program Segment String, addressed by the C register, is placed into the P register. One of the operators will be transferred from the P register to the T register for execution by the Processor. Which operator is transferred from the P register is indicated by the L register.

When the L register equals three, the last operator of P is set into the T register, and the P register can be loaded with the next word of the Program Segment String. This is accomplished by setting E16F and E17F to initiate a fetch access. The completion of the memory fetch access is indicated by MRAF, which results in the next program word being placed into the P register.

The memory fetch access will occur simultaneously with the execution of the operator in the T register. Usually, the fetch access will be completed prior to the termination of the operator, resulting in no time loss due to a fetch access. However, in some cases the operator in the T register may be terminated prior to the completion of a memory fetch access. If this does occur then the Processor is idled, by resetting TROF, to wait for the completion of the fetch access. At the completion of the fetch access, with L equal to zero, the first syllable of the new program word is set directly into the T register as well as the P register.

MEMORY ACCESS PRIORITY

A memory fetch access is requested by setting E16F to one. With E16F set, the memory access for a fetch is initiated by setting E17F to one. However, due to the fetch access occurring simultaneously with the execution of an operator, possibly requiring a memory access, a conflict of memory accesses could exist. Priority is given to the memory access required for the operator over the fetch access. E17F cannot be set if any memory accesses is about to be initiated by the Processor,

$$E17F = + E16F \cdot SENL \cdot CCCF/ \cdot EWZL$$
$$+ E16F \cdot SENL \cdot CCCF/ \cdot MROF$$
$$+ E16F \cdot SENL \cdot CCCF/ \cdot MTOD$$

indicated by the term SENL. SENL will be true only if all of the inputs to set the E register bits 1 through 8 are false. With SENL true and a fetch access requested, E16F • SENL, and the operation is not MTR, (CCCF/), E17F can be set if one of three conditions exist:

1. No memory access is presently in progress by the Processor as indicated by EWZL being true. EWZL is true if the E register (1 through 8) is reset and if MWOF is off.

2. If a memory access is presently being executed, EWZL is false; the fetch access will wait for either MROF or EWZL to be true. MROF will be true at the termination of a memory read access; if the memory access is a write then EWZL going true will signify the end of the write access.

3. If the Core Memory on the system is the 4 μs memory (B 461) and a memory write access is in progress, the fetch access does not wait for EWZL to go true, instead E17F would be set with MTOD, Memory Time Zero Driver from Central Control. This condition would not exist for the 6 μs memory (B 460).

Once E17F is set, the memory access for a fetch is identical to any other memory read access, with one exception: Normally a memory read access uses MROF to indicate the termination of the access, instead, the fetch access used MRAF.

INTERRUPT CONDITIONS

If the Processor is executing a program in control state an interrupt will have no effect upon the operation of the program. Clearing of and handling of the interrupt is done programmatically within the control state program by executing the Interrogate Interrupt Operator. If the Processor is in normal state the Interrogate Interrupt Operator is a NO-OP, therefore, the Processor must be put in control state in order to interrogate the interrupt that exists.

With the Processor in normal state, at the termination of each operator the Interrupt Address Register (IAR) in Central Control is examined. If no interrupts exist, the IAR will be equal to zero; the fetch operation will be allowed to proceed as normal. If an interrupt does exist the IAR will contain a branching address for the Interrogate Interrupt Operator. This interrupt condition will, with the Processor in normal state, disrupt the actions of the fetch logic and place the Store For Interrupt operator into the T register. This operator will eventually place the Processor into normal state and the Interrogate Interrupt Operator in the T register.

Syllable Execute Level

The Syllable Execute Complete Level (SECL) is true at the termination of each operator in the Processor. On each of the operator flow charts SECL is indicated by the action EXIT. SECL will be true for one micro-second, and with the last clock pulse of the operator in the T register it will initiate the fetch action to place the next operator into the T register and perform the necessary housekeeping to properly terminate the operator, see Figure 5.12-2. The action of housekeeping is necessary to clear the Processor control registers that must be equal to zero when initiating the next operator. The registers that are cleared by SECL are the J register, Q register, (bits 1 through 9 only), Y register, Z register, N register, X register and the M register. The N, X and M registers are cleared only if the Processor is in word mode, if the Processor is in character mode then these registers are not affected by SECL as they contain data pertinent to the execution of the next operator in sequence.

DETAILED DESCRIPTION

For the following description of SECL/Fetch refer to the Processor flow charts, Page 3.13.0.

Operator Execution

Initial starting of the Processor is normally done via the LOAD Button on the operators console. This button will, refer to Figure 5.12-1, generate a master clear pulse to the system and initiate an I/O read operation. The clear pulse will result in all flip-flops in the system being reset allowing the initiation of the system with the Processor idle and no interrupts or error conditions in existence. The I/O operation will be a card or drum read, depending upon the state of the Card/Drum Load Select Switch on the operator console. The I/O read will read the first word from the peripheral unit into core address 20.

When the I/O operation is completed the result field of the D register in the I/O channel is examined for an error condition. If an error does exist then the system is placed in an idle condition; only manual intervention can restart the system. If no error condition exists upon completion of the I/O read, then the level APKD is developed in Central Control and sent to the Processor designated as Processor #1. APKD will place the Initiate P1 operator into the T register and set TROF. The setting of TROF will remove the Processor from an idle condition and allow it to execute the Initiate P1 operator. This operator will initiate a fetch access from core address 20 and then terminate (EXIT).

Once the Initiate P1 operator is in the T register and TROF is set, it is treated as any other operator, that is, the Processor SECL/Fetch logic is not concerned with what the operator in the T register is. During the execution of the operator, Figure 5.12-3, a constant check is made to see if E16F is set. If it is set by the operator then the branch to 5 is taken to initiate a fetch access of memory. Also the execution of the operator in the T register will continue until the action EXIT is encountered. When the action of EXIT is true then the branch to 2 is taken.

SYLLABLE EXECUTE COMPLETE (SECL/FETCH)

For the following description refer to Figure 5.12-2 and the flow chart page 3.13.0.

The normal entry to this flow diagram is at 2. This entry is due to an operator action of EXIT which in turn makes the Syllable Execute Complete Level (SECL) true. SECL will perform the necessary housekeeping by clearing the J register, Q register bits 1 through 8, Y register, and the Z register. If in word mode then the SECL term will also clear the N and X registers.

At this time the next operator to be executed must be put into the T register or the Processor must be idled. If the Processor is in normal state and IO4L is true, the Processor is idled by resetting TROF, the normal action of fetching is inhibited

TROF ⟵ 0 = + SECL • NCSF • IO4L
IO4L = (Interrupt + HP2F • PK1L/)

by taking the branch to 7. IO4L will be true if an interrupt exists or if the Processor is Processor #2 and the Halt Processor 2 Flip-flop in Central Control is set. The branch to 7 will result in the placing the Store For Interrupt operator into the T register.

If the Processor is in control state or no interrupt exists then it must be determined if the P register contains a valid program word. If the P register is marked empty (PROF/) or if a fetch access has been requested and not completed (E16F) then the Processor will be idled and the branch to 6 taken to wait for

$$TROF \longleftarrow 0 = + SECL \bullet (E16F + PROF/)$$

the completion of a fetch access.

If none of the above conditions exists to idle the Processor then the Branch to 3 is taken to place the next operator in sequence into the T register. The transfer is from the P register to the T register with the level PTTL. PTTL is a term used

$$T \longleftarrow P\ [1] = PTTL$$
$$PTTL = I1OL \bullet PROF \bullet E16F/ \bullet (SECL + TROF/)$$

only on the SECL/Fetch flow chart, that is, it does not exist in either the D.A. Schematics or the Logic Books. PTTL is equivalent to the logic indicated above. It states that this is Processor #1 in control state or that no interrupt condition

$$I1OL = (\overline{I04L} + NCSF/ \bullet PK1L)$$

exists (I1OL), the P register contents are valid (PROF), no fetch access of memory is taking place (E16F/), and either this is the termination of an operator or the T register contents are invalid (SECL + TROF/).

If the Processor had been idled due to an incomplete fetch access (resetting TROF and taking the branch to 6) entry at 3 would not be at SECL time. Instead the entry at 3 would be from a memory fetch access with TROF/. If the L register is unequal to zero then the next operator to go into the T register will be with PTTL, enabled by TROF/. If the L register is equal to zero then the next operator is placed into the T register with MTTL (MIR to T Transfer Level). The level MTTL will place syllable

$$T \longleftarrow MIR\ (0) = MTTL$$
$$MTTL = MRAF \bullet TROF/ \bullet LEZL \bullet I1OL \bullet IFTL/$$

zero of the program word being set into the P register into the T register also. This would normally occur if the Processor had been idled due to an incomplete fetch at SECL time of the previous operator.

If the operator being set into the T register is an Operand/Descriptor Call, Load (not the LOAD Button), or a store operator then relative addressing will take place during the execution of that operator. In preparation for the possibility of the relative addressing going R relative, the contents of the R register are set into the nine high order bits of the M register, the six low order bits of M are cleared, and Q09F is set to one to allow the parallel adder to function as an address adder. These actions will occur if one of the following equations are true:

$$+ MRAL \bullet SECL \bullet I1OL$$
$$+ MRAL \bullet TROF \bullet I1OL \bullet JOOL$$
$$+ MTTL \bullet JOOL \bullet (MRBL + MRCL)$$

The first of these three conditions is used during a normal fetch with the PTTL transfer. The second condition is used only during the execution of MTR and will be explained in Subject 5.13. The third condition is used when the transfer into the T register is with MTTL.

The enabling level for the first two conditions is MRAL. MRAL will be true if P05L is true (P register is occupied, the Processor is in word mode or character mode with the Exit Character Mode operator in the T register, and no fetch access requested) and the operator being set into the T register is one of the three operators that

-0- MRAL

-1- + P05L • P04L • LEZL
+ P05L • P03L • LE1L
+ P05L • P02L • LE2L
+ P05L • P01L • LE3L

-0/- P05L

-I- + PROF/
+ CWMF/ • RECL/
+ E16F

require relative addressing. P01L, P02L, P03L and P04L determine the type of operators in the P register by examining the second, third and fifth bits of each operator.

-0/- P01L

-I- + P05F/ P02F/
+ P03F P02F/

For example, P01L will be true if both of the input OR terms are false. Both terms will be false if the operator in P is an Operand/Descriptor Call (P02F/ false) or if the operator is the Load or one of the Store operators (P05F/ and P03F both false). With P01L true and the L register equal to three (LE3L) then the action of transferring the R register contents to the M register will occur.

If MRAL is false at SECL time, refer to SECL Housekeeping , then the entire M register is cleared and Q09F is reset due to no R to M transfer taking place.

Regardless of the type of operator being set into the T register the next sequence of operation in Figure 5.12-2 is to increase the count in the L register. The L register indicated the syllable being set into the T register and must now be incremented to point to the next operator in sequence. The count of the L register occurs each time a syllable is set into the T register, counted by PTTL or MTTL.

If the L register, prior to counting, is unequal to three the branch to 1 is taken to execute the operator in the T register. If the L register had been equal to a value of three then the last operator had been transferred from the P register; mark the P register as being empty and take both the 1 and 4 branches. The branch to 1 will again be for the execution of the operator; the branch to 4 will be to initiate a fetch access to reload the P register. Both of these actions will occur at the same time.

FETCH MEMORY ACCESS

For the following description refer to the flow diagram on Figure 5.12-3.

Entry to 4 is from the action of SECL and L equal to three, resetting the P Register Occupied Flip-flop (PROF) to prepare to request a fetch access. The action of initiating the fetch access at this time coincides with the first clock pulse of the operator previously set into the T register. At this time the two levels ICFL/ and ACFL are interrogated. ICFL/ (Inhibit Count Up For Fetch Level) will be true if the operator in the T register does not contain the action of Inhibit Fetch. This level will allow the fetch access to be requested at J equal zero of the operator in the T register by setting E16F. The C register is incremented by one (C + 1) to address the next program word in core memory. If ICFL/ is false then the fetch access is inhibited by the operator and the fetch access is not requested. Examples of operators that can Inhibit Fetch are all branching operators and any operator code that can change the configuration in the C register.

If, for example, the operator in the T register is a conditional branch operator that had executed the action of an Inhibit Fetch and this operator subsequently does not branch, the action of Enable Fetch or Allow Fetch will be executed. Either of these actions will make the term ACFL true (Allow Count Up For Fetch). ACFL, along with PROF/ will initiate a fetch access request by setting E16F and incrementing the C register by one. If both ICFL/ and ACFL are false no fetch access is initiated. A check of E16F is made, if set then a fetch access was requested by the operator in the T register.

Once the fetch access has been requested the access will be initiated by setting E17F if no memory access is about to be requested by the operator in the T register (SENL) and no memory access is in progress or a memory access is being completed (EWZL + MROF + MTOL).

With E17F set the P register is cleared in preparation for receiving the new program word from memory. This clearing of the P register is necessary as the transfer from MIR to P is a single ended transfer. At this point an entry can be made at 6 from Figure 5.12-2 where the Processor was idled due to PROF/ or E16F.

The operation of the fetch access now waits for the completion of the memory access, as indicated by MRAF. With MRAF on the new program word is set into the P register and the fetch access is terminated by resetting E16F and E17F. If TROF is on at this time then the Processor is presently executing an operator; take the branch to 1 to complete the execution of this operator. However, if TROF is off then take the branch to 3 to place the next operator into the T register from MIR (MTTL), and subsequently to the operator execution.

If the processor had been placed into an idle state due to an interrupt (see Figure 5.12-2) then entry to Figure 5.12-3 will be at 7. The normal action here is to place the Store For Interrupt operator into the T register, set Q07F to indicate to the SFIL operator that this was a hardware initiated Store For Interrupt, and reset Q09F. The next sequence of operation is to take the branch to 1 to execute the Store For Interrupt operator.

MISCELLANEOUS

Because they do not directly affect the operation of the SECL/Fetch operation some items are covered as miscellaneous items. For these items and their actions refer to the SECL/Fetch flow chart, Page 3.13.0.

Invalid Address

Any memory access with an invalid address is immediately withdrawn. The invalid address is indicated by the term 938L. This term is equivalent to addressing a nonexistent Memory Module or addressing the control state portion of memory with the Processor in normal state.

A nonexistent Memory Module can be a Memory Module that does not exist on the system or a Memory Module that is in local. In either case, if the three high order bits of the addressing register designate such a Memory Module, 938L will be true.

The addressing of the control state portion of memory when in normal state is indicated by the six high order bits of the addressing register. If these six bits are all reset then the portion of memory being addressed is 000 through 777 or the first 1024 cells of core memory. This portion of core is reserved for the MCP only and any object program (normal state program) accessing this portion of core will enable 938L.

938L and a memory access other than a fetch access (938L • E17F/) will clear the E register bits 1 through 8 to withdraw the memory access request. If the requested memory access is for a fetch access (938L • E17F) then the fetch access is withdrawn by resetting E16F and E17F with PROF being set to one. For either access the invalid address interrupt is set in the Processor I02F of the Processor interrupt register. If the Processor was in normal state the SECL/Fetch flow would allow the termination of the operator in the T register and the perform the Store For Interrupt. The only portion of the operation execution that will be inhibited is any memory access to write data into memory.

Stop Clock

If a memory address error or a memory parity error should occur while the Processor is running in control state the appropriate Processor interrupt will be set (I01F for parity and I02F for address error) and the Processor clock will be stopped with the level UIMC going true. The reason for stopping the clock is that in control state the error occurred during the running of the MCP and cannot be corrected programmatically.

Unit Busy

The unit busy line to Central Control (UBSS1) will be true if the P register is occupied, the T register is occupied, a fetch access has been requested or the Processor is in normal state. If none of the above conditions are true then the Processor is idle with no activity taking place.

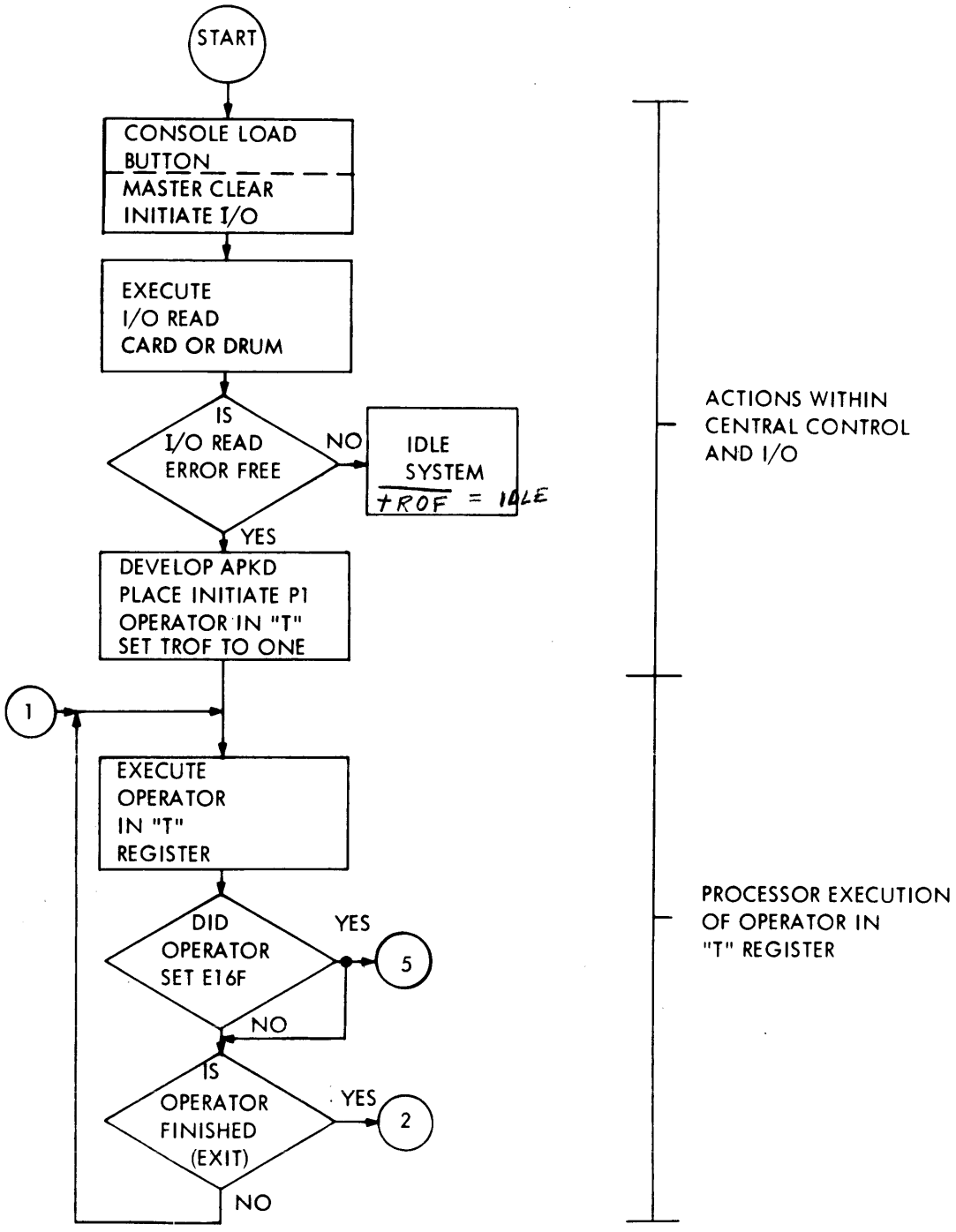


FIGURE 5.12-1 PROCESSOR INITIATION AND SYLLABLE EXECUTION

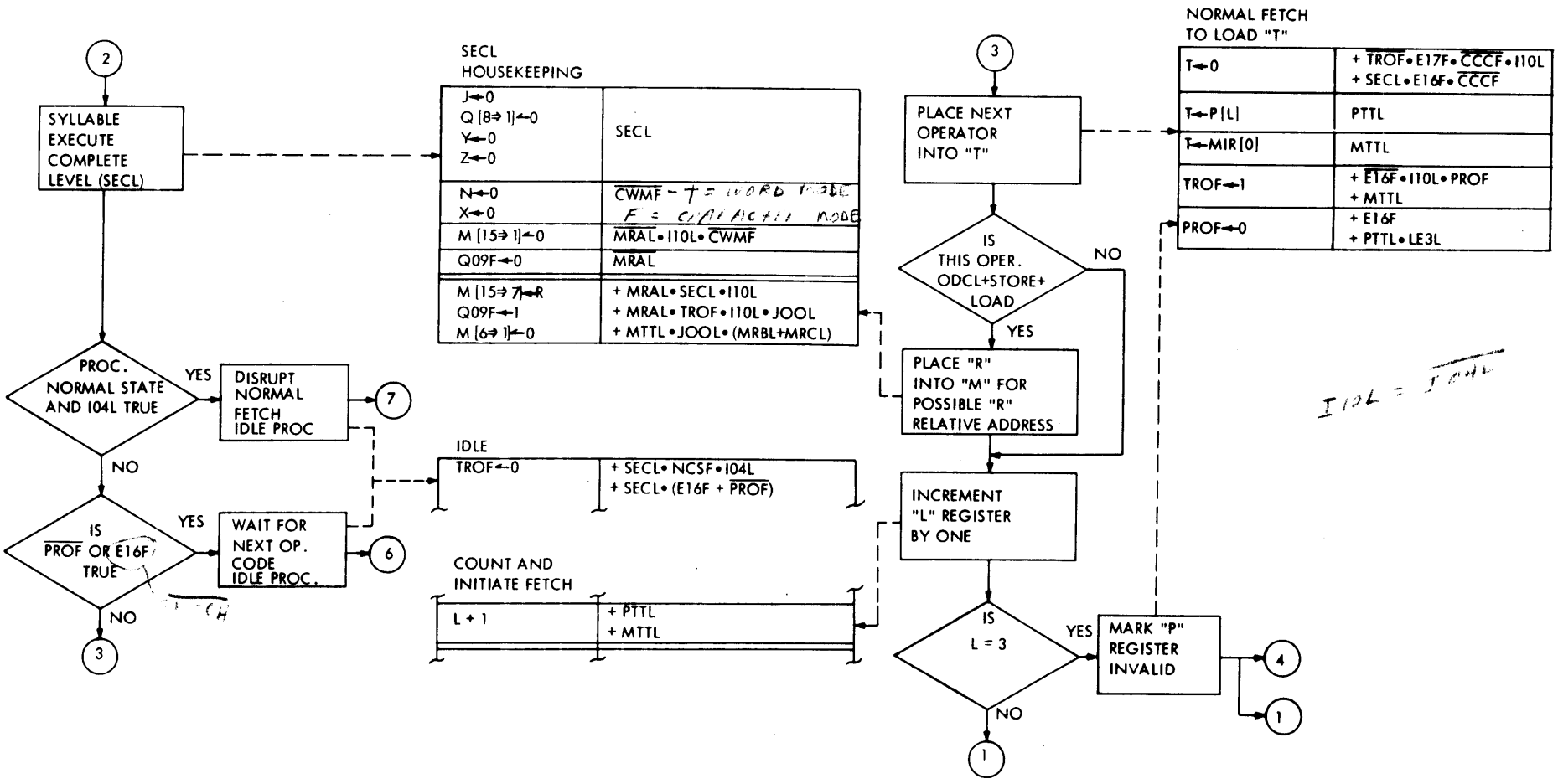


FIGURE 5.12-2 SYLLABLE EXECUTE COMPLETE

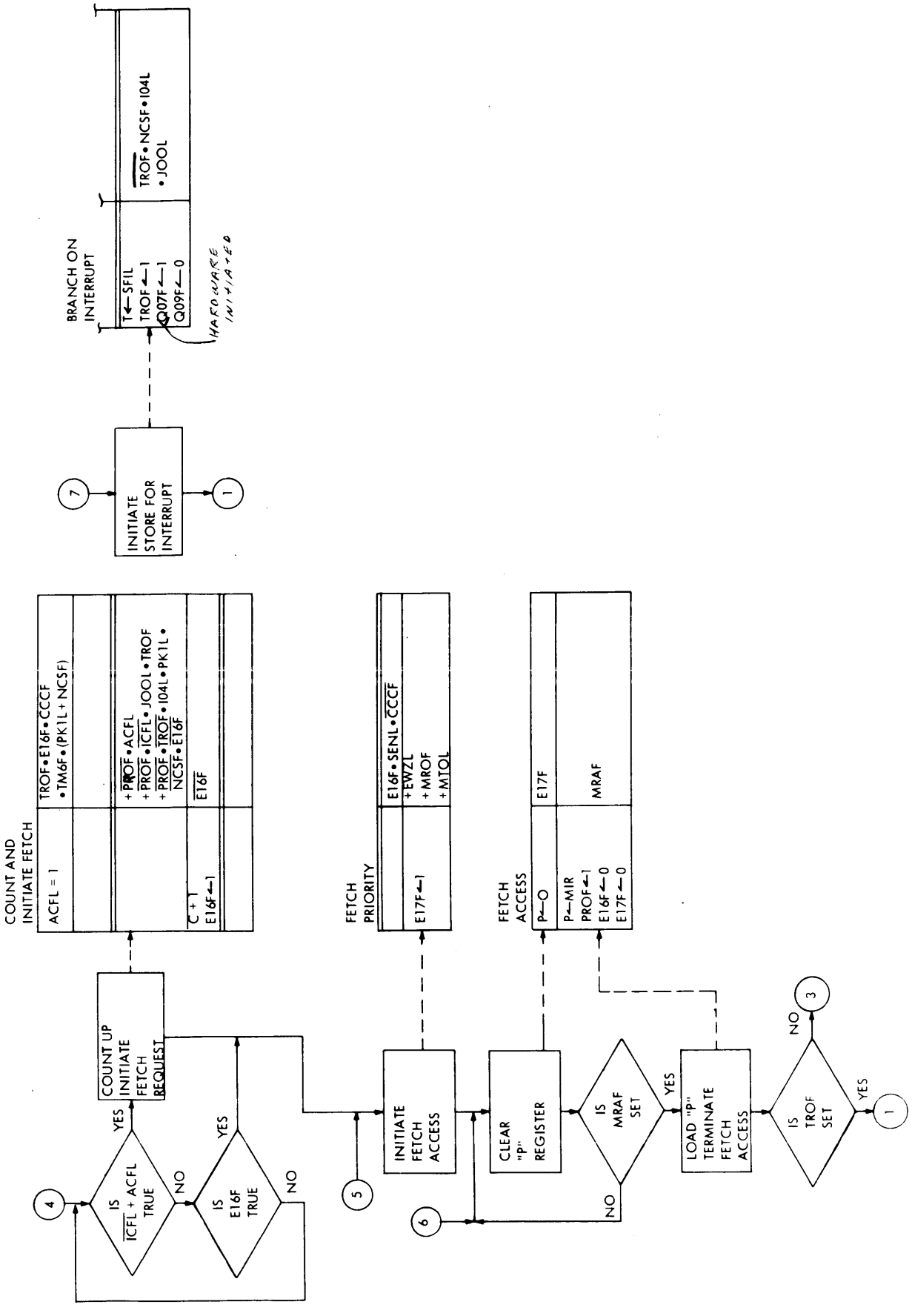


FIGURE 5.12-3 FETCH ACCESS

5.13 MTR LOGIC

The maintenance test logic in the B 5500 Processor is used in conjunction with the B 5500 Processor Maintenance Test Routine (MTR). This routine makes use of the Initiate For Test (IFTL) and Store For Test (SFTL) operators to programmatically simulate a pulse by pulse check of the Processor logics. The maintenance test logic provides the ability to preset all Processor registers and flip-flops (with the exception of the E register) prior to making a test. The test itself can be as short as one clock pulse in length or can be programmatically terminated.

TEST CONTROL WORDS

The two test operators, IFTL and SFTL, are described in Subjects 5.4 and 5.5 of this manual. Their operation is very similar to the Initiate P1 and Store for Interrupt operators in that use is made of the same type control words. The control words must contain the additional information to enable the Initiate For Test operator to set up all Processor flip-flops. The additional information contained in the control words for testing purposes is:

Initiate Control Word (INCW)

Bits 17 ⇒ 21 of the INCW are placed into TM (5 through 1). These bits are used to indicate the test status of the J register and NCSF.

Bit 22 is placed into TM6F, in turn indicating the test status of CCCF.

Bits 23 ⇒ 28 are used to give the test status of the Z register.

Bits 29 ⇒ 34 are used to give the test status of the Y register.

Bits 35 ⇒ 43 are placed into Q01F through Q09F.

Bit 44 is placed into TM7F. This bit indicates the test status of MWOF.

Bit 46 is placed into TM8F. This bit indicates the test status of MROF.

Interrupt Return Control Word (IRCW)
Bit 46 is used to indicate the test status of BROF.

Interrupt Control Word (I.C.W.)
Bits 15 ⇒ 1 are used to indicate the test status of the M register.

Bits 16 ⇒ 19 are used to indicate the test status of the N register.

Some of the above information is common to the Initiate For Test and an Initiate P1 into character mode. The Initiate For Test and Store For Test operators follow the same sequence of an entry to or exit from a character mode program.

TEMPORARY STORAGE REGISTER

The Temporary Storage Register (TM) is an eight bit register used to hold those test conditions which cannot be set up during the execution of IFTL and SFTL operators. These conditions are the test J register setting, MROF, MWOF, NCSF, and CCCF (Clock Count Control Flip-flop). The TM register bits and their associated usage are:

TM1F ⇒ TM4F - contains the J register configuration required at test time.

TM5F - contains the test status of NCSF. This flip-flop is normally set by an Initiate P1 operator, the Initiate For Test operator derives the status of NCSF from bit 21 of the Initiate Control Word.

TM6F - contains the test status of CCCF.

TM7F - contains the test status of MWOF.

TM8F - contains the test status of MROF.

The TM register contents are transferred to the appropriate registers upon completion of the Initiate For Test operator, just prior to the execution of the Test.

CLOCK COUNT CONTROL FLIP-FLOP

The Clock Count Control Flip-flop (CCCF) is used to distinguish between the two basic types of tests. The first type of test is entered with CCCF = 1. Entry to the test can be at any J count (0 through 15), regardless of the operator code in the T register. The operator under test (Operator X) can be any configuration of bits in the T register. Operator X can be an illegal operator, or an operator that does not use the value in the J register. The flip-flop (CCCF) will force an exit from the Test, regardless of the Test operator and J count, and will place the SFTL operator into the T register.

The second type of test is entered with CCCF = 0. Entry to the test can be at any J count (0 ⇒ 15), but the combination of J count and Operator X must provide a path to SECL. The termination of this type of a test is with either a Store For Interrupt or a Store For Test operator.

The status of CCCF is obtained from TM6F, which in turn is set by bit 22 of the Initiate Control Word.

TEST OPERATION

A detailed description of the test operators, IFTL and SFTL, is given in Subject 5.4 and 5.5 of this manual. The following description will primarily be concerned with the actions occurring during the actual test, after the last clock pulse of the IFTL operator. This write up will make reference to the SECL/FETCH flow chart, page 3.13.0.

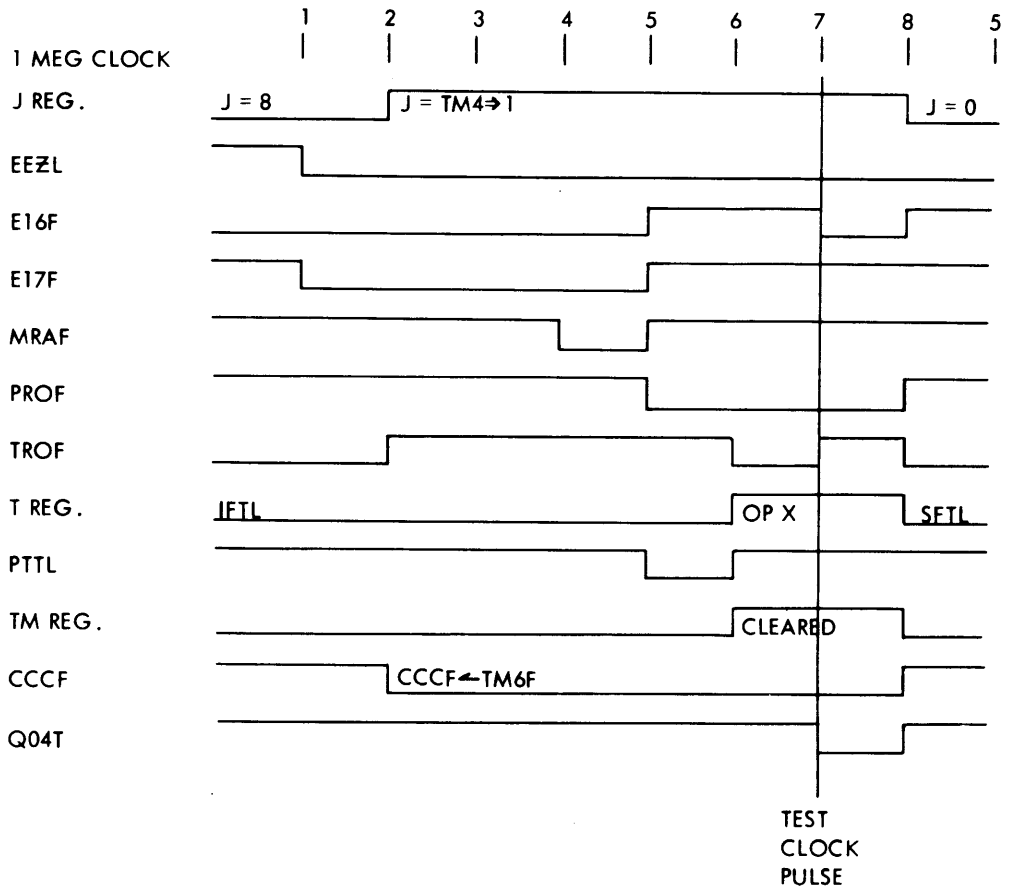


FIGURE 5.13-1 TEST - NO MEMORY ACCESS

Entry to a test with CCCF = 1 is basically a 1 clock pulse test. Referencing the SECL/FETCH flow, 3.13.0, and Figure 5.13-1, the pulse numbered one in Figure 5.13-1 is the clock pulse of the IFTL operator that coincides with the termination of the memory cycle to load the A register. The following pulse, pulse number 2, is the final clock pulse of the IFTL operator. This clock pulse will transfer TM4 ⇒ 1 to the J register and TM6F to CCCF. The Processor is now placed in an idle state by the reset of TROF.

A fetch access was initiated by Pulse #1 to load into the P register a program word that contains Operator X. The termination of the fetch access is indicated by MRAF at pulse five. This pulse will place the new program word into the P register, set PROF, and reset E16F and E17F. The P to T Transfer Level (PTTL) is then enabled to allow pulse six to transfer Operator X to the T register and set TROF. This same pulse, with TROF/ and PROF will clear the TM register and transfer TM7F and TM8F to MWOF and MROF, refer to SECL/FETCH flow under Special Housekeeping.

The Processor is now completely set up for the test pulse, pulse number 7. This pulse will perform the actions in the Processor as specified by the conditions set into the Processor flip-flops and also initiate the termination of the test by re-setting TROF. With TROF = 0, the End Test Control Driver (Q04T) will go true. This level will, at pulse eight, transfer the contents of the J register to TM4 ⇒ 1, place the SFTL operator into the T register, and reset CCCF. The next clock pulse, pulse S, is the first clock pulse of the Store For Test operator.

All of the actions of terminating the test and forcing the SFTL operator were enabled due to CCCF being equal to one. If the test had been entered with CCCF = 0, then the reset of TROF at pulse seven would not have occurred, the subsequent action of the Processor would be determined by Operator X, eventually leading to a SECL of Operator X and a programmatic termination of the test.

If at test time Operator X initiates a memory access, and CCCF = 1, then the test will be more than a one clock pulse test. Referring to Figure 5.13-2, if the memory initiated is a memory write then the test is a two clock pulse test. On the SECL/FETCH flow chart, TROF is reset by CCCF • E08F • MTOL, a memory write access at memory time zero.

If the initiated memory access is a read access, then the test will be a three pulse test, see Figure 5.13-3. TROF will be reset on the SECL/FETCH flow by the logic CCCF • EEZL/ • MWOF.

The term Q04T in either case will not be enabled until EEZL is true at the completion of the memory access.

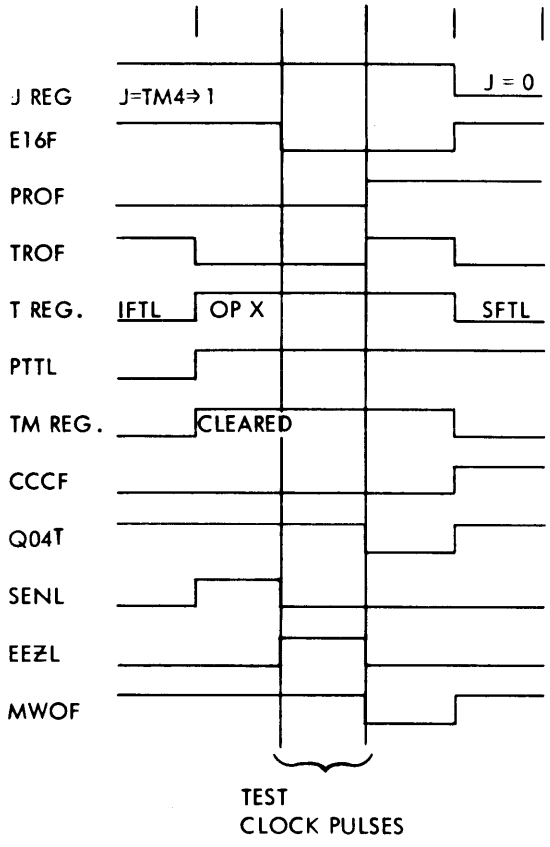


FIGURE 5.13-2 TEST AND MEMORY WRITE

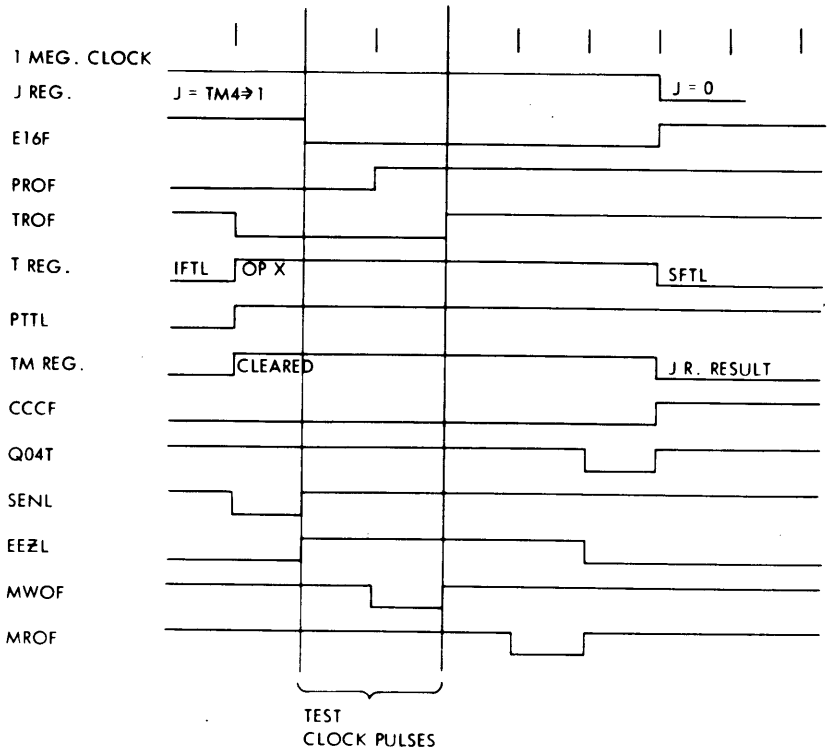


FIGURE 5.13-3 TEST & MEMORY READ

1024408-07