TM-3870

REFERENCE MANUAL FOR THE PHOENIX DIGITAL COMPUTER

15 NOVEMBER 1963

THE MITRE CORPORATION

Technical Memorandum TM- 3870

Sheet 1 of 85 Sheets

Contract No. AF 19(628)-2390

SUBJECT: REFERENCE MANUAL FOR THE PHOENIX DIGITAL COMPUTER (Project 508.0)

TO: Distribution List

FROM: D-13 Staff

DEPT.: D-13

DATE: 15 November 1963

APPROVED: H. K. Rising

THE MITRE CORPORATION

Bedford, Massachusetts

ISSUED AT:

ABSTRACT

This document describes the structure and operation of PHOENIX (nee FIFI), a digital computer being designed and built by MITRE Department D-13 for SDL.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

PREFACE

This manual is intended as a reference manual for the PHOENIX digital computer; it is not a primer, nor a system manual. Logically, the structure of the manual is characterized by the programming terms: "multiple pass" and "recursive."

An attempt has been made to exactly specify the functional characteristics of PHOENIX, and to provide enough information to allow even pathological situations to be analyzed.

The format is chosen to facilitate corrections and additions, which will be issued as needed.

Suggestions for improving this manual will be appreciated.

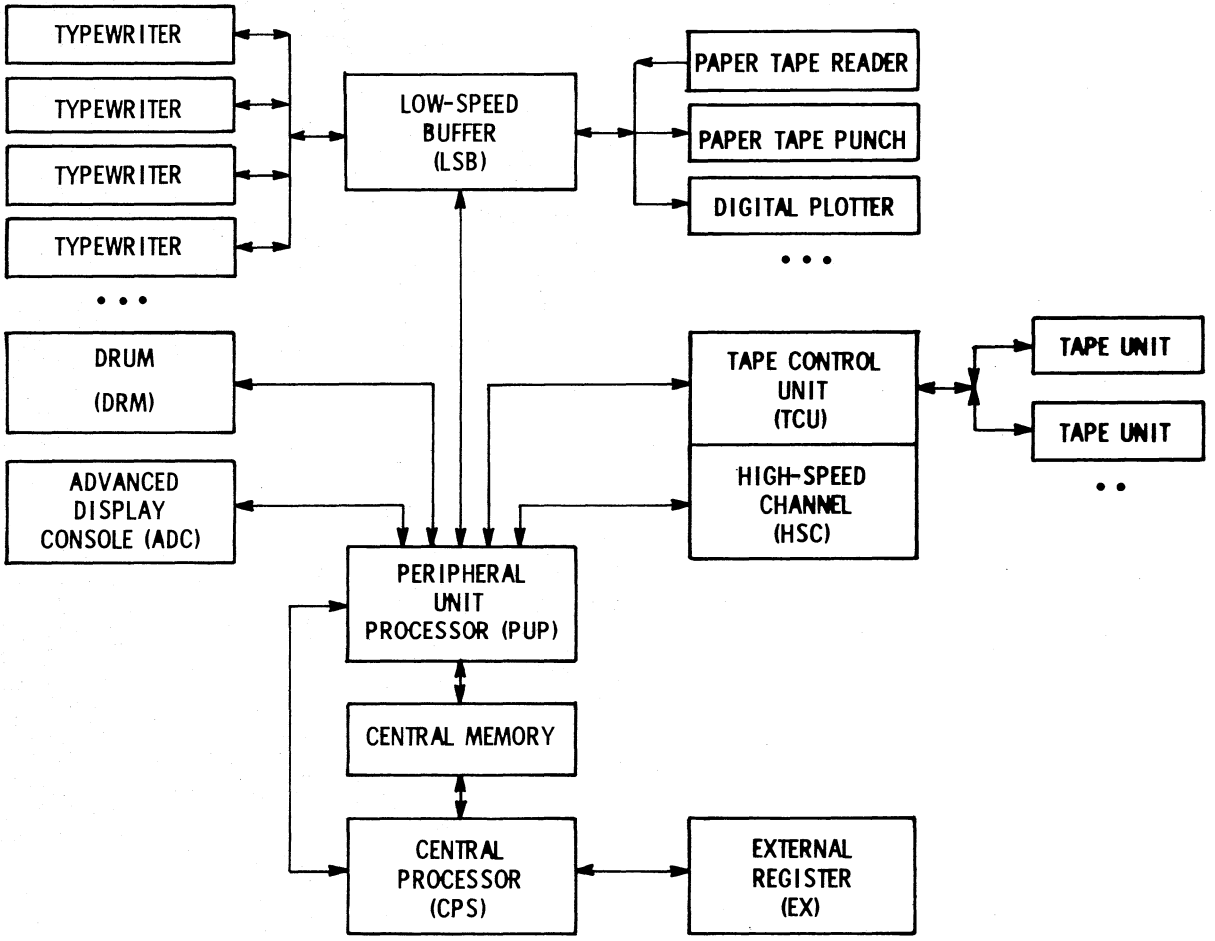


FIG 0.0

ORGANIZATION OF THE PHOENIX COMPUTER

0.0 INTRODUCTION

PHOENIX is a binary single address digital computer which performs instructions at an average rate of 500,000 instructions per second. The overall organization of the machine is shown in Figure 0.0.

0.1 Memory

The main memory consists of 4 memory units (A, B, C, and D) of 16,384 26-bit words each. Each word has, in addition to its normal 24 data bits, an associated metabit which is not affected by normal manipulation, and a parity bit used for internal error checking. These 4 memories are independently accessible, providing for the overlap of instructions, data, and I/O references (cf. 3.0). Memory is assigned to user programs in segments consisting of integral multiples of 4096 words whose base or starting address is effectively 0. The relocation and limiting of the space available to user programs is handled automatically by means of the boundary and relocation registers.

0.2 Central Processor (CPS)

The central processor has a repertoire of 54 basic instructions, several of which sub-divide into whole classes of subsidiary instructions. The entire main memory is addressable by these instructions. Address modification and indexing may be performed. There are 5 index registers per se; in addition, the program counter and the "address" portion of the accumulator are addressable as index registers; also there is an index whose contents are always zero, making a total of 8 "index" registers.

The arithmetic unit performs 1's complement, fixed-point operations.

A flexible interrupt system provides for interrupting the main stream of instructions to honor a variety of I/O demand and error conditions.

To facilitate time sharing, facilities are built into the machine which automatically relocate the user program, limit the amount of memory available to the user program, and prevent the user program from directly affecting I/O operations.

The external register (EX) provides a convenient and flexible channel for non-overlapped input/output operations between CPS and non-standard external devices.

The basic machine cycle is 1.6 microseconds.

0.3 Input-Output

I/O operations are performed in parallel with program operations by PUP (Peripheral Unit Processor) which mediates data transfers between one of 5 channels and main memory.

PUP operations are available to facilitate searching for records and sorting records, on the basis of key words or portions of a word.

The ADC (Advanced Display Console) is a flexible visual display device with internal storage and response capabilities.

The LSB (Low Speed Buffer) controls the transmission of I/O information between various low-speed devices and an external memory. Thirty-two devices, including 16 typewriters, may be connected to LSB. Initially, there will be 4 typewriters, a paper tape reader, a paper tape punch, and a digital incremental recorder.

The TCU (Tape Control Unit) controls I/O transmission to and from a maximum of 8 magnetic tape drives (TU). Four CDC 607 interfaces are provided which will handle data at rates up to 30,000 24-bit words per second. Four IBM 729-IV interfaces are provided which will handle data at rates up to 15,625 24-bit words per second. Initially, there will be two CDC 607 tape drives.

The HSC (High Speed Channel) provides for transmission of information on a character basis between external devices and PHOENIX at variable rates up to 333,000 6-bit characters per second (equivalent to 83,250 24-bit words per second).

The DRM (Drum) has a capacity of 278,528 words (34 sectors of 8,192 words each), which can be transmitted to and from CPS memory at the rate of 491,520 words per second.

BLANK PAGE

CHAPTER I --CENTRAL PROCESSOR

1 NOMENCLATURE

1.1 Quantities, Bits

Let x be an n-bit quantity; that is, an integer in the range 0 to 2^n-1 inclusive. It is well-known that there exist unique integers x_k ($k = 0,\ldots, n-1$), each $x_k = 0$ or 1, such that $x = \sum_{k=0}^{n-1} x_k \times 2^{n-1-k}$. We call this x_k the k-th bit of x.

1.2 Bytes

If x is an n-bit quantity, and i and j are integers such that $0 \leq i \leq j \leq n-1$, then we define the byte $x_{i:j} \equiv \sum_{k=i}^{j} x_k \times 2^{j-k}$.

We also define:

$x_s \equiv x_0$ (sign)

$x_a \equiv x_{8:23}$ (address)

$x_t \equiv x_{0:2}$ (tag)

$x_p \equiv x_{3:7}$ (opcode)

$x_m \equiv x_{24}$ (metabit)

$x_c \equiv x_{n-7:n-1}$ (count)

1.3 Registers and their contents

We consider the computer to be a collection of registers, each of which contains a quantity. We describe the operation of the computer in terms of how the contents of these registers change with time. Given a register α, we use the symbol $[\,\alpha\,]$ to denote the contents of α. We speak of α as an n-bit register if $[\,\alpha\,]$ is at most an n-bit quantity.

1.3.1 Memory registers

A certain set of registers (CPS memory registers) are named by quantities. Given a register α, where α is a quantity, then by $[\ \alpha\ ]$ we mean $[\ \alpha\ ]_{0:23}$.

1.3.2 Live registers

Registers which are not CPS memory registers are termed "live registers" and are referred to by symbolic names, rather than by quantities.

1.4 Replacement

"$x \rightarrow [\ \alpha\ ]_{i:j}$" means: "The quantity x replaces the contents of register α, bits i to j, inclusive, the remainder of α being left unchanged." More precisely, if α is an n-bit register, then $[\ \alpha\ ]$ changes to:

$$\sum_{k=0}^{i-1} [\alpha]_k \times 2^{n-1-k} + \sum_{k=i}^{j} x_{k-1} \times 2^{n-1-k} + \sum_{k=j+1}^{n-1} [\alpha]_k \times 2^{n-1-k}.$$

1.5 Operations

Let x and y be quantities of exactly n bits each. We define:

1.5.1 $x +_n y = x + y$ if $x + y < 2^n$ else

$x +_n y = x + y - 2^n$

1.5.2 $x \oplus_n y = x + y$ if $x + y < 2^n$ else

$x \oplus_n y = x + y - (2^n - 1)$

1.5.3 $x \vee y \equiv \sum_{k=0}^{n-1} \max(x_k, y_k) \times 2^{n-1-k}$

1.5.4 $x \wedge y \equiv \sum_{k=0}^{n-1} \min(x_k, y_k) \times 2^{n-1-k}$

1.5.5 $x \veebar y \equiv \sum_{k=0}^{n-1} (x_k +_1 y_k) \times 2^{n-1-k}$

1.5.6 $x -_n y = x +_n (2^n - y)$

1.5.7 $x \ominus_n y = x \oplus_n (2^n - y - 1)$

1.5.8 $\bar{x} = 2^n - 1 - x$

1.5.9 abs x = x if $x < 2^{n-1}$ else

abs x = $\bar{x}$

1.6 Metabits

Words in CPS memory, LSB memory, and drum memory have a 25th bit called the metabit in addition to the normal 24 bits of the word. The metabit is a flag and is not used in normal computations or logical manipulations. The metabit of word W is designated by the notation W_m. A metabit may be set to 0 or 1 by an I/O operation and the instructions MMZ and MMN. A metabit may be sensed by the user-controlled metabit trap, the instructions SMZ and SMN, or by some I/O operations.

1.6.1 Special I/O metabit handling

Data coming from HSC or ADC always has a metabit of 0. When data is sent to a device connected to one of these channels, the metabit is ignored. Data sent to the TCU also has its metabits ignored. However, on an input from the TCU, the first word of every record has a metabit of 1 associated with it.

1.7 Miscellaneous notation conventions

1.7.1 Where there is no possibility of ambiguity, we will elide the subscript specifying the number of bits involved in an arithmetic operation. If this subscript is unspecified, the size of the registers involved will be assumed. For example, we will write " [CAR] + [RL]" for "$[CAR] +_{16} [RL]$" since both CAR and RL are 16-bit registers.

1.7.2 Given a register α, "step α" means "$[\alpha] + 1 \Rightarrow [\alpha]$". (Cf. 2.22)

1.7.3 The term "minus zero", also noted "-0", designates the n-bit quantity 2^n-1.

2.0 LIVE REGISTERS

The operations of PHOENIX CPS (central processor) can be described in terms of its memory and the registers described in the following sections.

2.1 AC Accumulator (24 bits)

The AC is used in most instructions to hold operands and receive the results of arithmetic and logical operations. $AC_a \equiv XR6$

2.2 ACT Address Counter (16 bits)

The ACT is used during I/O operations to determine CPS storage locations for I/O data. [ACT] can be read into [BR] by a PER when [PR]=0.

2.3 BD Boundary Register (16 bits)

This register determines the limits of memory space available to a user program when the computer is in protected status. The BD may be changed only by an LRB instruction. $[BD]_{4:15} \equiv 0$.

2.4 BR B-register (24 bits)

The BR is used as an operand register for several arithmetic and logical operations. It also receives the results of several arithmetic and logical operations. The BR is also used to hold a mask for some operations.

2.5 CA Combined Arithmetic Register (47 bits)

The CA has no existence independent of the AC and BR. It is used in some shifts and cycles, divide, multiply, and the double-length arithmetic operations.

$CA_{0:23} \equiv AC_{0:23}$

$CA_{24:46} \equiv BR_{1:23}$

2.6 CAR Central Address Register (16 bits)

The CAR is used in computing effective addresses. It is not directly addressable by the programmer.

2.7 CCAR Cycle and Shift Central Address Register (7 bits)

The CCAR is used for computing the effective address E for shifts and cycles. It is not directly addressable by the programmer.

2.8 CCI Clear CAR Inhibit (1 bit)

If [CCI] = 0, the CAR and CCAR are cleared after the current operation. If [CCI] = 1, the CAR and CCAR are not cleared. CCI is not addressable by the programmer.

2.9 CK Clock (24 bits)

The [CK] may be changed only by an SKM instruction while the CPS is in unprotected status. The [CK] is decremented at intervals of 5 μsec, ± .005 μsec. When its contents are -0, the clock interrupt bit is set. More precisely, every 5 μsec, the following sequence of events takes place:

1) If [CK] = 1, then -0 → [CK] and 1 → $[NR]_{21}$ and go to 4
2) If [CK] = 0, then 2^{24} -2 → [CK] and go to 4
3) [CK] - 1 → [CK]
4) Done.

2.10 EN Enable Register (24 bits)

The enable register acts as a mask for the NR (interrupt register). The [EN] may be changed only by an SKM instruction while the CPS is in unprotected status. $[EN]_{20} \equiv 1$ and $[EN]_{22} \equiv 1$.

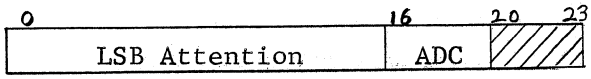
2.11 EX External Register (24 bits)

EX allows a program to communicate directly with devices external to PHOENIX without the mediation of PUP. The [EX] may be sensed or modified by a SKM instruction if either [PR] = 0, or [EXM] = 1. The sensing of [EX] automatically clears it. An external device suitably connected to EX can set or sense [EX] or any subset of its bits. The external interrupt bit $[NR]_{13}$ can be set by a pulse generated by an external device. It is intended that this interrupt will be used to request that a program perform an action involving EX.

2.12 EXM EX Mode Control Register (1 bit)

This register controls the EX with respect to a SKM instruction while the CPS is in protected status. If [EXM] = 0, then [EX] cannot be sensed or changed by a SKM instruction in protected status. If [EXM] = 1, then [EX] can be sensed or changed by a SKM instruction in protected status. [EXM] may be modified only by certain PER instructions which, themselves, are operative only in unprotected status.

2.13 LA LSB Attention and ADC Register (24 bits)



$[LA]_{0:15}$ are the attention status indicators for the 16 typewriter lines to the LSB. $[LA]_{16:19}$ are status bits for ADC.

Each bit is set by a pulse from LSB or ADC which also sets one of 3 indicators in NR:

LA bit	Meaning	Interrupt bit
0:15	Typewriter 0:15 attention	16
16	Not Storable	8
17	Memory overflow	8
18	Manual Input	11
19	Display Area Change	11

[LA] may be changed by SKM in unprotected status

2.14 LB LSB Buffer Status Register (24 bits)

$[LB]_i$ = 1 when the LSB memory area assigned to LSB input device i has been filled. $[LB]_{0:15}$ are for the 16 typewriters; $[LB]_{16}$ is for the paper tape reader.

2.15 LE LSB Device Error Register (24 bits)

$[LE]_i$ = 1 when a device error has occurred during a data transmission between LSB memory and device i, or during other LSB operations involving device i.

$[LE]_{0:15}$	typewriters
$[LE]_{16}$	paper tape reader
$[LE]_{17}$	paper tape punch
$[LE]_{18}$	digital incremental recorder

[LE] may be changed through the use of SKM.

2.16 LI LSB Input Status Register (24 bits)

$[LI]_i$ = 1 when LSB device i is in input status, i.e., when it is entering information into LSB memory. $[LI]_{0:15}$ are for the 16 typewriters; $[LI]_{16}$ is for the paper tape reader.

2.17 LO LSB Output Status Register (24 bits)

$[LO]_i$ = 1 when LSB device i is in output status, i.e., when it is receiving information from LSB memory. $[LO]_{0:15}$ are for the 16 typewriters; $[LO]_{17}$ is for the paper tape punch; $[LO]_{18}$ is for the digital incremental recorder.

2.18 MC Miscellaneous CPS indicators (24 bits)

$MC_3 \equiv$ VF (overflow indicator)

$MC_4 \equiv$ VFP (permanent overflow indicator)

$[MC]_5$ is the metabit trap enable bit.

[MC] may be changed through the use of SKM.

2.19 MP Miscellaneous PUP Indicators (24 bits)

Bit	Use
0	--
1	PI (PUP idle indicator)
2	TI (TCU idle indicator)
3	--
4	--
5	--
6	IOR parity error
7	PBR parity error
8	KEY parity error
9	MASK parity error
10	DSA parity error
11	APR parity error
12	Unexpected channel demand
13	Unexpected channel answer
14	PUP memory request error
15	ADC noisy
16	LSB noisy
17	TCU noisy
18	DRM noisy
19	--
20	--
21	Step control error
22	Time pulse distribution error
23	Drum compare error

Note that bits 6:23 represent conditions which may either cause an interrupt or halt the machine entirely. If $[EN]_1 = 1$ then these conditions cause an interrupt, otherwise the machine halts. $[MP]_{6:23}$ are set to zero before each I/O operation. These bits do not normally concern the programmer.

$[MP]_{1:2}$ may be set to 1 through the use of SKM in unprotected status.

2.20 MTC Metabit Trap Condition Register (1 bit)

The [MTC] = 1 if the conditions for a metabit trap obtain (cf. 6.2.1), otherwise [MTC] = 0. [MTC] is tested during each normal machine instruction cycle. [MTC] is not directly addressable by the programmer.

2.21 NR Interrupt Register (24 bits)

Various events cause bits to be set in NR.

Bit	Event
0	--
1	I/O faults
2	--
3	Magnetic tape read/write error
4	End-of-file reading tape, or End-of-tape writing tape
5	--
6	Tape unit or LSB device inoperative
7	TCU operation complete
8	LSB device busy, tape busy, or ADC I/O exception
9	PUP operation complete
10	Tape unit rewind complete
11	ADC attention
12	TCU/HSC special device attention
13	External device attention
14	--
15	LSB device error
16	Typewriter attention
17	Device to LSB transfer complete
18	--
19	LSB to device transfer complete
20	Out-of-bounds BCH
21	Clock counted down to -0
22	Out-of-bounds memory reference
23	Unused (reserved for trace interrupt)

[NR] may be changed through the use of SKM in unprotected status.

2.22 PC Program Counter (16 bits)

In the normal case of instruction flow [PC] is the location of the next instruction to be performed. For details, see section 4.

PC ≡ XR7

2.23 PCC Program Counter Control (1 bit)

The state of PCC determines whether the location of the next instruction is taken from PC ([PCC] = 1) or from CAR ([PCC] = 0).

2.24 PI PUP Idle Indicator (1 bit)

$PI \equiv MP_1$

[PI] = 1 indicates that PUP is idle (not busy). PUP may be forced to become idle by using SKM in unprotected status to put $1 \Rightarrow [MP]_1$.

2.25 PR Protected Status Indicator (1 bit)

PHOENIX operates in two statuses: protected and unprotected. The word "protected" is intended to signify that the machine (and the data of other users) is protected from the consequences of errors perpetrated by the currently operating program.

[PR] = 1 when the computer status is protected; [PR] = 0 when the status is unprotected.

$0 \Rightarrow [PR]$ when an interrupt takes place, while $1 \Rightarrow [PR]$ when the instruction RIS is performed when [PR] = 0.

The state of PR cannot otherwise be changed.

2.26 RL Relocation Register (16 bits)

[RL] is used automatically to relocate CPS memory references in protected status. For details, see section 4.

RL can be changed using an LRB instruction in unprotected status. It cannot otherwise be changed, and its contents cannot be fetched.

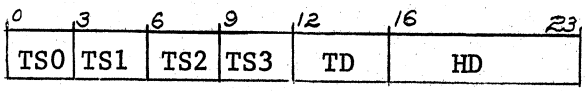
An LRB affects $[RL]_{0:3}$. $[RL]_{4:15}$ are always zero.

2.27 TI TCU Idle Indicator (1 bit)

$TI \equiv MP_2$

[TI] = 1 indicates that TCU is idle (not busy). TCU may be forced to become idle by using SKM in unprotected status to put $1 \rightarrow [MP]_2$.

2.28 TS Tape and Special Device Status Indicators (24 bits)



TS consists of 6 groups of indicators.

Each of the first four groups contains a 3-bit byte which specifies the status of the four CDC tapes which may be connected to TCU. For the i-th byte (i = 0,1,2,3), these 3 bits have the following significance:

Bit	Significance when bit is 1
0	CDC tape unit i not busy and ready for reading
1	CDC tape unit i not busy and ready for writing
2	CDC tape unit i at load point

Each of the remaining 12 indicators is set by a "special device attention" pulse. $TS_{12:15}$ correspond to the four IBM 729 tape unit lines to the TCU, while $TS_{16:23}$ correspond to the eight special device lines to the HSC. Any such "special device attention" pulse will also set $[NR]_{12}$.

$[TS]_{12:23}$ may be changed through the use of SKM in unprotected mode.

2.29 VF Overflow Indicator (1 bit)

$VF \equiv MC_3$

VF is set whenever one of the instructions ADD, DAD, DSU, SHA, SHB, SHC, or SUB is performed. If an overflow condition results, then $1 \rightarrow [VF]$, otherwise $0 \rightarrow [VF]$.

An SKM may be used to set or clear $[VF] \equiv [MC]_3$.

RIS performed in unprotected status will set $[4]_3 \rightarrow [VF]$.

2.30 VFP Permanent Overflow Indicator (1 bit)

$VFP \equiv MC_4$

VFP is set to one whenever one of the instructions ADD, DAD, DSU, SHA, SHB, SHC, or SUB is performed, if an overflow condition results. However it is not cleared if there is no overflow condition.

An SKM may be used to set or clear $[VF] \equiv [MC]_4$.

RIS performed in unprotected status will set $[4]_4 \rightarrow [VF]$.

2.31 WC Word Count Register (16 bits)

The WC is used to keep track of the number of words transmitted by an I/O operation. [WC] may be tested by an SKM, but not altered.

2.32 XR0-7 Index Registers (16 bits)

There are 8 index registers. Index registers 0, 6, and 7 have no independent existence:

$[XR0] \equiv 0$

$XR6 \equiv AC_a$

$XR7 \equiv PC$

XR5 has some special uses when the instructions BCH and CBX are executed.

BLANK PAGE

3.0 TIMING

The following gives a detailed description of central processor timing within PHOENIX.

3.1 Definitions

The basic unit of instruction timing is a machine cycle which is equivalent to 1.6 microseconds. We will use the term "cycle" within section 3 with the meaning: "machine cycle".

Concurrent operation of the central processor (CPS) and the input/output processor (PUP) is possible since CPS memory is divided into four separately addressable units. This also permits overlap operation of two successive instructions in certain cases which are described later. The low-order two bits of an address define the memory unit as follows:

2 LOW-ORDER ADDRESS BITS	MEMORY UNIT
00	A
01	B
10	C
11	D

Memory unit conflict is defined as an attempt to address a memory unit concurrently from more than one address source (such as the program counter (PC) and the I/O address counter (ACT) both addressing memory unit A). A priority is established in the computer to solve such memory conflicts. The descending order of priority is as follows:

1) I/O memory request.

2) CPS memory request for data.

3) CPS memory request for instruction.

The memory conflict is resolved by delaying the lower priority request for one or more cycles. Overlap of a CPS memory request for an instruction with a CPS memory request for data is conditional upon: 1) no memory conflict in the instruction request, and 2) no memory conflict in the data request.

3.2 Discussion of the Timing table

Table 3.4 defines the timing of each instruction. The following sub-sections define the assumptions used to produce this table and discuss timing peculiarities referred to by this table. The contents of Table 3.4 are also included within Table A-2 in the Appendix.

3.2.1 The minimum time to execute an instruction as given in Table 3.4 assumes "best case" conditions.

3.2.2 The average time to execute an instruction assumes:

1) No I/O processor memory conflict.

2) A probability of 0.25 that there will be a data-request/instruction-request memory conflict, where appropriate.

3) A probability of 0.0 that there will be an index register conflict such as noted in sub-sections 3.3.2 and 3.3.3.

4) No metabit traps.

5) Average data conditions as defined in sub-sections 3.3.1 thru 3.3.10.

3.2.3 The maximum time to execute an instruction assumes:

1) No I/O processor memory conflict.

2) Worst case data conditions.

3) Index register conflict.

3.2.4 'Machine Cycles" if Aborted refers to those instructions which are "no operation" instructions under certain conditions or to the DVD instruction which is aborted if the data are improper.

3.3 Timing variations of instructions

Within the conditions specified for Table 3.4, some instructions have timing variations due to index register conflicts, data-characteristics, etc. These instructions are discussed in the following sub-sections.

3.3.1 AND, DAC, DAM, DBR, DPX, EOR, LAC, LBR, MMN, MMZ, and VOR

Overlap conditions:

No memory conflict.

Timing:

Two cycles unless overlap, then 1 cycle.

Data-condition assumptions made for "average":

None.

3.3.2 ADD, DAD, DSU, and SUB

Overlap conditions:

1) No memory conflict.

2) Next tag $\neq 6$ ($\equiv$ AC).*

Timing:

1) ADD, SUB: 2 cycles unless overlap, then 1.

2) DAD, DSU: 3 cycles unless overlap, then 2.

Data-condition assumptions made for "average":

None.

* By "next tag" we mean bits 0:2 of the next instruction.

3.3.3 CBX

Overlap conditions:

Next tag ≠ 5.

Timing:

NUMBER OF SHIFTS	CYCLES; NO OVERLAP	CYCLES; OVERLAP
0-4	2	1
5-19	3	2
20-35	4	3
36-50	5	4
51-63	6	5

Data-condition assumptions made for "average":

Shift of 11.

3.3.4 CYA, CYC

Overlap conditions:

Next tag ≠ 6 (≡AC).

Timing:

Two cycles unless overlap, then 1.

Data-condition assumptions made for "average":

None.

3.3.5 CYB

Overlap conditions:

Always overlap next instruction.

Timing:

Actually 2 cycles, but always overlap, thus 1 cycle.

Data-condition assumptions made for "average".

None.

3.3.6 DVD

Overlap conditions:

No overlap possible.

Timing:

If ZQ = the number of "zeros" in abs(quotient),

then timing is:

15 + int* {-0.45 + 0.23 × ZQ} cycles.

Data-condition assumptions made for "average":

ZQ = 11.

3.3.7 MPY

Overlap conditions:

No overlap possible.

Timing:

If NOA = number of "ones" in abs [AC] at the

start of the instruction, and [PROD] = product,

then the timing is:

5 + int $\{0.69 + 0.19 \times NOA + 0.09 \times [PROD]_s\}$ cycles.

Data-condition assumptions made for "average":

NOA = 11.

---

*"Int" is defined as "largest integer not greater than". For example: int {0.9} = 0; int {1.1} = 1; int {-0.4} = -1.

3.3.8 NAL and NCL

Overlap conditions:

No overlap possible.

Timing:

NUMBER OF SHIFTS	CYCLES
0	1
1-2	2
3-10	3
11-18	4
19-26	5
27-34	6
35-42	7
42-46	8

Note: NAL limited to a maximum of 23 shifts.

Data-condition assumptions made for "average":

1) Shift 11 for NAL.

2) Shift 23 for NCL.

3.3.9 SHA and SHC

Overlap conditions:

Next tag ≠ 6 (≡AC).

Timing:

NUMBER OF SHIFTS	CYCLES; NO OVERLAP	CYCLES; OVERLAP
0-4	2	1
5-19	3	2
20-35	4	3
36-50	5	4
51-63	6	5

Data-condition assumptions made for "average":

1) Shift 11 for SHA.

2) Shift 23 for SHC.

3.3.10 SHB

Overlap conditions:

Always overlap next instruction.

Timing:

NUMBER OF SHIFTS	CYCLES
0-4	1
5-19	2
20-35	3
36-50	4
51-63	5

Data-condition assumptions made for "average":

Shift 11.

3.4 Instruction Timing

Table 3.4 is based on certain considerations, assumptions, and definitions developed in earlier portions of section 3.

Instruction	Normal Machine Cycles Min.	Ave.	Max.	Machine Cycles If Aborted	Section Reference
ADD	1	1.25	2		3.3.2
ADE	1	1	1		
ADX	1	1	1		
AND	1	1.25	2		3.3.1
ASB	1	1	1		
BAS	1	1	1		
BCH	1	1	1		
BSA	1	1	1		
BXA	1	1	1		
BXS	1	1	1		
CBX	1	2	6		3.3.3
CMA	2	2	2		
CMX	1	1	1		
CYA	1	1	2		3.3.4
CYB	1	1	1		3.3.5
CYC	1	1	2		3.3.4
DAC	1	1.25	2		3.3.1
DAD	2	2.25	3		3.3.2
DAM	1	1.25	2		3.3.1
DBR	1	1.25	2		3.3.1
DFR	1	1	1		
DPX	1	1.25	2		3.3.1
DSU	2	2.25	3		3.3.2
DVD	14	17	19	3	3.3.6
EOR	1	1.25	2		3.3.1
LAC	1	1.25	2		3.3.1
LBR	1	1.25	2		3.3.1
LDE	2	2	2		
LRB	1	1	1	1	
LXI	1	1	1		
LXR	2	2	2		
MMN	1	1.25	2		3.3.1
MMZ	1	1.25	2		3.3.1
MPY	5	7	10		3.3.7
NAL	1	4	5	1	3.3.8
NAR	1	1	1	1	
NCL	1	5	8	1	3.3.8
NCR	1	1	1	1	
NVX	1	1	1		
PAX	1	1	1		
PER	2	2	2	2	
PXA	1	1	1		
RIS	3	3	3	1	
SAS	2	2	2		
SHA	1	2	6		3.3.9
SHB	1	2	5		3.3.10
SHC	1	3	6		3.3.9
SIO	1	1	1	1	
SKM	2	2	2		
SMN	2	2	2		
SMZ	2	2	2		
SUB	1	1.25	2		3.3.2
TRP	2	2	2		
VOR	1	1.25	2		3.3.1

Table 3.4 Instruction Timing

3.5 Other timing considerations

We next consider certain timing details not as directly associated with normal instruction timing as the factors discussed in the preceding section.

3.5.1 Metabit trap timing

When metabit trap conditions obtain (cf. 6.2.1), a sequence of operations is initiated (cf. 6.2.2) which may be considered (for purposes of timing) as equivalent to an instruction with the following characteristics:

Overlap conditions:

No memory conflict (always* true).

Timing:

Two cycles unless overlap, then 1 cycle; but overlap is always possible, thus 1 cycle.

3.5.2 Interrupt timing

3.5.2.1 Normal Case

When interrupt conditions obtain (cf. 6.1.1), and the current instruction is not BCH, a sequence of operations is initiated (cf. 6.1.2) which may be considered (for purposes of timing) as equivalent to an instruction with the following characteristics:

Overlap conditions:

No memory conflict (always true).

Timing:

Three cycles unless overlap, then 2 cycles; but overlap always possible, thus 2 cycles.

---

* We still follow the convention, at this point, of ignoring I/O.

3.5.2.2 Pathological case (BCH)

If interrupt conditions obtain and if the current instruction is a BCH, then the timing given in 3.5.2.1 above is increased by one cycle.

BLANK PAGE

4.0 THE INSTRUCTION CYCLE

The CPS operates by performing a sequence of instructions.

4.1 Elementary Instruction Cycle

One may consider the (normal) basic instruction cycle to be subdivided into the following steps:

4.1.1 Step 1

Bring the word in memory location [PC] into the CPS, and treat it as an instruction I.

4.1.2 Step 2

Step PC.

4.1.3 Step 3

Compute the effective address E of I (cf. 8.0).

4.1.4 Step 4

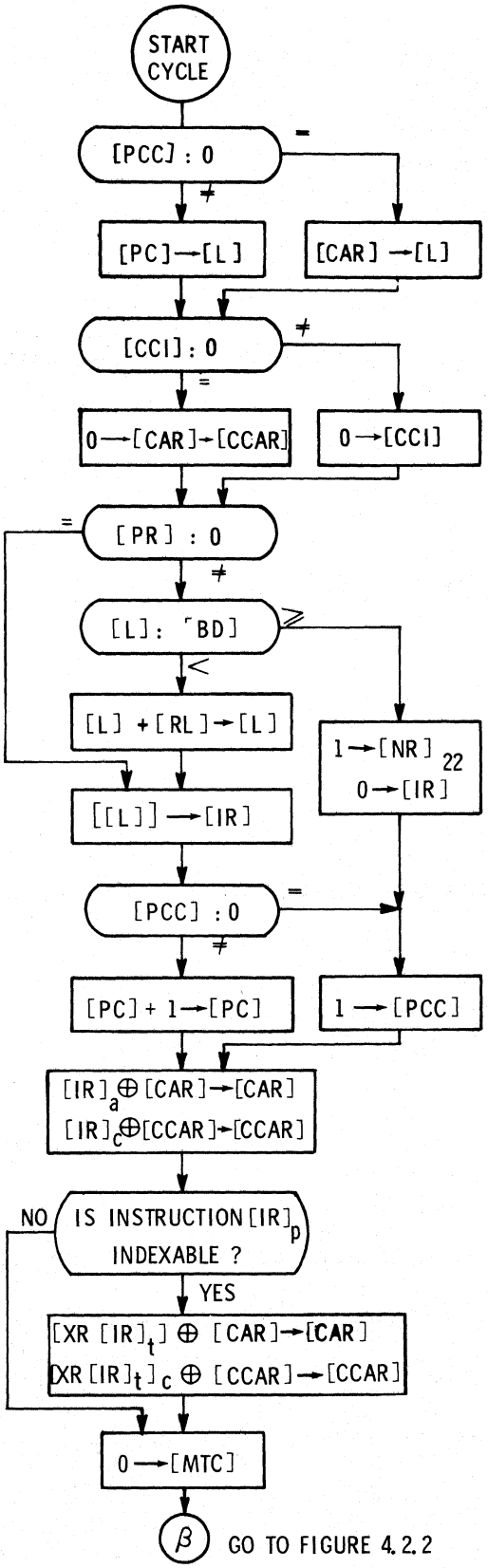
Perform the instruction I as described in 9.0.

4.1.5 Step 5

Go to step 1.

4.2 Detailed Instruction Cycle

More precisely, the functional behavior during an instruction cycle is specified by Figures 4.2.1 and 4.2.2. Figure 4.2.3 represents only those portions of the instruction cycle which are of interest to a user programmer (i.e., a programmer who writes programs for use in protected status).



[PCC]= 0 ONLY AFTER A DFR.
[L] WILL BE USED IN THE INSTRUCTION FETCH; NORMALLY, [L] = [PC] AT THIS POINT.

[CCI] ≠0 ONLY AFTER LDE & ADE.

NORMALLY, 0 → [CAR] → [CCAR].

IF UNPROTECTED (PR] = 0), THEN SKIP BOUND TEST AND INSTRUCTION RELOCATION.
IF PROTECTED ([PR] = 1), THEN TEST INSTRUCTION ADDRESS AGAINST BOUND.

NORMALLY RELOCATE BY [RL].
IF OUT OF BOUNDS, SET INTERRUPT BIT, FETCH 0 AS INSTRUCTION.

NORMAL INSTRUCTION FETCH.

NORMALLY STEP PC; DO NOT IF LAST INSTRUCTION WAS DFR.

NORMALLY [CAR] = [CCAR] = 0 BEFORE THIS ADDITION.

INDEXABILITY OR NON-INDEXABILITY IS AN INTRINSIC PROPERTY OF EACH INDIVIDUAL INSTRUCTION.

PERFORM THE INDEXING OPERATION.

RESET THE METABIT TRAP CONDITION INDICATOR.

FIG 4.2.1

FUNCTIONAL FLOW CHART OF INSTRUCTION CYCLE, PART I

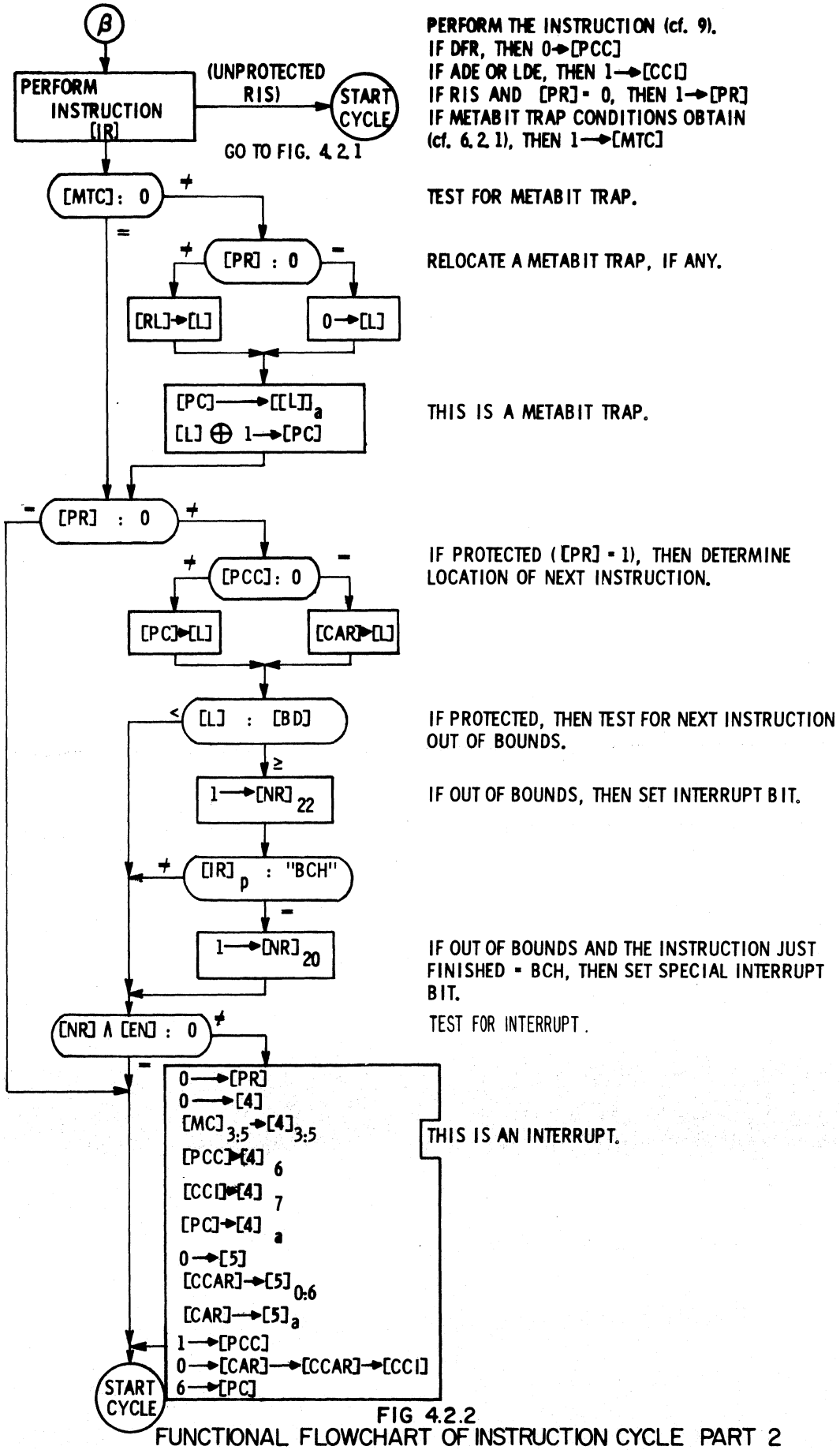
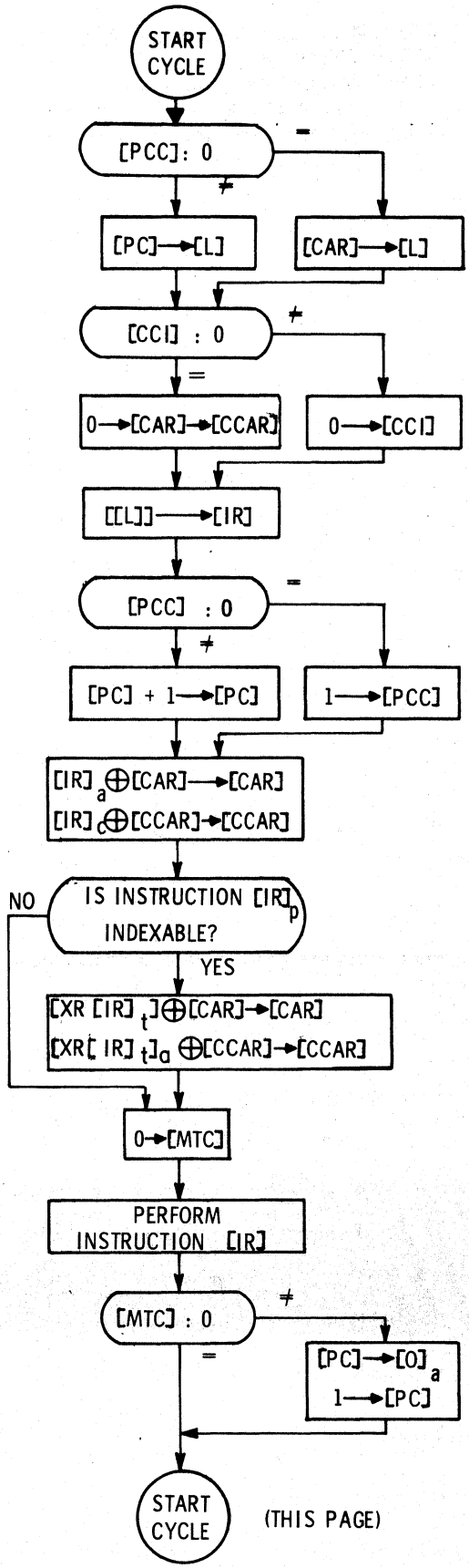


FIG 4.2.2
FUNCTIONAL FLOWCHART OF INSTRUCTION CYCLE PART 2

IB-12,388



ASSUMPTIONS:

1.) INTERRUPTS AND RELOCATION IGNORED.
2.) BOUNDARY TESTS IGNORED.

IF PREVIOUS INSTRUCTION WAS A DFR, THEN [PCC] = 0, ELSE [PCC] = 1.

[L] WILL BE USED IN THE INSTRUCTION FETCH.

[CCI] ≠ 0 ONLY IF PREVIOUS INSTRUCTION WAS A LDE OR ADE, ELSE [CCI] = 0.

NORMALLY 0→[CAR]→[CCAR] ; IF OTHERWISE, THIS IS THE INDIRECT ADDRESSING FUNCTION.

FETCH THE INSTRUCTION

NORMALLY STEP PC; IF THE LAST INSTRUCTION WAS A DFR, DO NOT STEP PC BUT RESET PCC.

BEGIN BUILDING EFFECTIVE ADDRESS, E; OF THE INSTRUCTION.

COMPLETE THE EFFECTIVE ADDRESS OF THE INSTRUCTION BY INDEXING, IF NECESSARY.

CLEAR METABIT TRAP CONDITION REGISTER.

PERFORM THE INSTRUCTION (cf. 9).
MTC WILL BE SET TO 1 IF THE METABIT TRAP CONDITIONS OBTAIN (cf. 6.2.1).

TEST FOR METABIT TRAP.

METABIT TRAP.

LOOP.

IB-12,381

FIG 4.2.3
SIMPLIFIED FUNCTIONAL INSTRUCTION CYCLE FOR THE "USER" PROGRAMMER

5.0 PROTECTED STATUS

To permit effective sharing among different user programs the computer normally operates in protected status. When an interrupt occurs, ordinary processing of instructions is suspended and control is transferred to a special program which operates in unprotected status. A special instruction, RIS, returns the program to protected status.

5.1 Protected Status Operations

When the computer is operating in protected status ([PR]=1), the processing of instructions and data described elsewhere is modified in the following ways:

5.1.1 Protected Instructions

The instructions SIO, RIS, LRB, and PER act as NOP (No operation).

5.1.2 Protected Registers

The SKM instruction will not alter the contents of CK, EN, NR, WC, MP, TS, LI, LO, LB, LA, LE, or, if [EXM]=0, EX.

5.1.3 Relocation

All CPS memory references are relocated by [RL]; that is when a memory fetch from, or store into, location L is to be performed, the actual memory location involved is not L, but L + [RL].

5.1.4 Boundary Protection

All CPS memory references are checked to see if they are in bounds, as follows:

5.1.4.1 Data Fetch

If an instruction requests a memory fetch from location L, and $L \geq [BD]$, then $1 \rightarrow [NR]_{22}$, and a word of all zeros is fetched.

5.1.4.2 Out of Bounds Instruction

If at the completion of the current instruction, [PCC] = 1 and [PC] ≥ [BD] or [PCC] = 0 and [CAR] ≥ [BD], then $1 \rightarrow [NR]_{22}$. If, in addition, the current instruction is a BCH, then $1 \rightarrow [NR]_{20}$ as well. (Note that $[EN]_{20,22} \equiv 1$.)

5.1.4.3 Data Store

If an instruction requests a memory store into location L, and L ≥ [BD], then $1 \rightarrow [NR]_{22}$; [L] remain unaffected.

6.0 INTERRUPTS AND TRAPS

Both interrupts and traps are breaks in the normal sequence of program operations. Each interrupt or trap saves [PC] in CPS memory, and then alters [PC] so as to change normal sequencing; the contents of certain other registers are also saved when an interrupt occurs (cf.6.1.2). An interrupt places the CPS in unprotected status (and resets certain registers), while a trap leaves the CPS status unchanged.

6.1 Interrupts

6.1.1 Interrupt Conditions

An interrupt will occur when all of the following conditions are satisfied.

6.1.1.1 CPS is in protected status, ([PR] = 1).

6.1.1.2 $[NR] \wedge [EN] \neq 0$. [EN] acts as a mask which allows the selection of which interrupt conditions will cause an interrupt to take place.

6.1.1.3 The current instruction has been completed.

6.1.2 Interrupt Events

When all interrupt conditions are satisfied, the following sequence of events takes place:

6.1.2.1 If a metabit trap condition has arisen as a result of performing the previous instruction, then the metabit trap takes place (cf. 6.2.1, 6.2.2).

6.1.2.2 CPS is put into unprotected status (0 → [PR])

6.1.2.3 $0 \rightarrow [4]_{0:24}$

6.1.2.4 $[MC]_{3:5} \rightarrow [4]_{3:5}$

6.1.2.5 $[PCC] \rightarrow [4]_{6}$

6.1.2.6 $[CCI] \rightarrow [4]_{7}$

6.1.2.7 $[PC] \rightarrow [4]_{a}$

6.1.2.8 $0 \rightarrow [5]_{0:24}$

6.1.2.9 $[CCAR] \rightarrow [5]_{0:6}$

6.1.2.10 $[CAR] \rightarrow [5]_{a}$

6.1.2.11 $1 \rightarrow [PCC]$

6.1.2.12 $0 \rightarrow [CAR] \rightarrow [CCAR] \rightarrow [CCI]$

6.1.2.13 $6 \rightarrow [PC]$

6.1.2.14 CPS resumes normal operation *

6.1.3 Interrupt Register (NR)

Various events taking place in the CPS or peripheral equipment cause bits to be set in NR. These events and their associated bits are listed in 2.21.

6.2 Traps

There are 2 types of traps: that initiated by the instruction TRP, and the metabit trap. The operation of TRP is described in Section 9.50.

---

* An interrupt cannot cause a metabit trap.

Both the TRP and the metabit trap are under the control of the user program.

6.2.1 Metabit Trap Conditions

A metabit trap will occur when all of the following conditions are satisfied:

6.2.1.1 $[MC]_5 = 1$. This bit may be set to zero or one using SKM.

6.2.1.2 The current instruction is not DFR, LDE, MMN, MMZ, RIS, SMN, or SMZ.

6.2.1.3 The current instruction has made a data reference (either fetch or store) to a memory location whose contents have metabit one.

6.2.1.4 The current instruction has been completed.

6.2.2 Metabit Trap Operations

When all metabit trap conditions are satisfied, the following sequence of events takes place:

6.2.2.1 $[PC] \rightarrow [0]_a$ *

6.2.2.2 $1 \rightarrow [PC]$

6.2.2.3 Normal operation is resumed, unless an interrupt condition obtains (cf. 6.1.2.1).

---

* The location here referred to is 0 if status is unprotected, but [RL] if status is protected (cf. 5.1.3). $[0]_m$ unchanged. A metabit trap cannot cause a metabit trap.

7.0 INSTRUCTION WORD FORMATS

7.1 Terminology

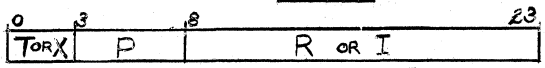
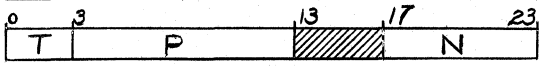
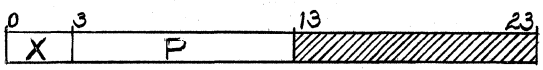
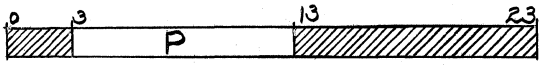
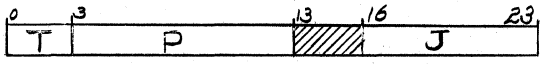
SYMBOL	FIELD SIZE (bits)	MEANING
T	3	The instruction is "indexable"*, and the [XRT] are used in the indexing operation.
X	3	The instruction is not "indexable", but XRX is somehow associated with the instruction.
P	5 or 10	The operation code specifying the instruction.
R	16	A number used in forming an effective address in CAR, used (as a Remote operand) by the instruction.
I	16	A number used in forming an immediate operand in CAR, used by the instruction.
N	7	A number used in forming an immediate operand in CCAR, used by the instruction.
J	8	A number which, along with bits 8:15 of the instruction, is used in forming a 16-bit number, E, in CAR. In the case of a LRB, the result is used as an immediate operand. A PER utilizes $E_{8:15}$ to supplement P during the instruction decoding.

---

* By "indexable", we mean that [XRT] and $[XRT]_c$ are added to [CAR] and [CCAR], respectively, during the execution of an instruction cycle (cf. 4.0).

7.2 Instruction Formats

Each PHOENIX instruction is one of six "types". Associated with each instruction type is a "format":

TYPE	FORMAT	NOTES
1		LAC, BAS,...
2		shifts, cycles
3		PXA, MMZ,...
4		RIS, ASB,...
5		LRB, PER only
6		TRP only

8.0 EFFECTIVE ADDRESSES

Referring to Figures 4.2.1 and 4.2.2, we see that the following sequence of operations has been performed on [CAR] and [CCAR] at the point within the instruction cycle at which the instruction is executed:

1) $0 \rightarrow [CAR] \rightarrow [CCAR]$ except after LDE or ADE

2) $[CAR] \oplus [IR]^{*}_{a} \rightarrow [CAR]$

   $[CCAR] \oplus [IR]_{c} \rightarrow [CCAR]$

3) If the instruction $[IR]_{p}$ is "indexable", (tag "T" in Section 9.0), then

   $[CAR] \oplus [XR[IR]_{t}] \rightarrow [CAR]$

   $[CCAR] \oplus [XR[IR]_{t}]_{c} \rightarrow [CCAR]$

From the state of [CAR] and [CCAR] at this point, we define the effective address E below and also show the modifications of E performed for relocation purposes (cf. 5.0).

INSTRUCTION TYPE	OPERAND TYPE	E ([PR] = 0)	$E_{modified}$ ([PR] = 1)
1	R	[CAR]	[CAR] + [RL]
	I	[CAR]	[CAR]
2	N	[CCAR]	[CCAR]
3	----	----	----
4	----	----	----
5	J	[CAR]	[CAR]
6	----	----	----

* IR is the "Instruction Register".

BLANK PAGE

9.0 DESCRIPTION OF PHOENIX INSTRUCTIONS

The following sections functionally describe the operations performed by the several PHOENIX instructions.

The operations are those referred to as "Step 4" in Section 4.1.4, and as "perform instruction $[IR]_p$" in Figures 4.2.2 and 4.2.3.

The particular effective address (cf. 8.0) used by each instruction is determined for each instruction by its "Type", as follows:

Type	Effective Address (E)
1	[CAR]
2	[CCAR]
3	none
4	none
5	[CAR]
6	none

Indexability or non-indexability is an intrinsic property of each instruction, and is indicated in the descriptions by the symbols used to represent the instruction operands. (cf. 7.0) The symbol "T" indicates that an instruction is indexable.

The notation and nomenclature is that described in Sections 1.0, 2.0, and 7.0.

9.1 ADD R,T Add Type 1

1) $0 \rightarrow [VF]$

2) $Y = [AC] \oplus [E]$

3) If $[AC]_s = [E]_s \neq Y_s$, then $1 \rightarrow [VF]$ and

$1 \rightarrow [VFP]$

4) $Y \rightarrow [AC]$

9.2 ADE I,T Add to Effective Address Type 1

1 → [CCI]

The effect of this instruction is to produce the CAR and CCAR modifications characteristic of a Type 1 indexable instruction (cf. 8.0, 8.1.2).

9.3 ADX I,X Add to XR Type 1

[XRX] ⊕ E → [XRX]

9.4 AND R,T And Type 1

$$[AC] \wedge [E] \rightarrow [AC]$$

9.5 ASB AC Sign to BR Type 4

$[AC]_s \rightarrow [BR]_i$, i = 0:23

9.6 BAS BR Sign to AC Sign Type 4

$[BR]_S \rightarrow [AC]_S$

9.7 BCH I,T Branch Type 1

1) [PC] → [XR5]

2) E → [PC]

9.8 BSA BR Sign to AC Type 4

$[BR]_s \rightarrow [AC]_i$, i = 0:23

9.9 BXA I,X Branch on XR and Add Type 1

1) If X = 6 or X = 7, then go to 5

2) If [XRX] = 177777_8, then go to 5

3) [XRX] $\oplus$ 1 → [XRX]

4) E → [PC]

5) Done.

9.10 BXS I,X Branch on XR and Subtract Type 1

1) If X = 6 or X = 7, then go to 5

2) If [XRX] = 177777_8, then go to 5

3) [XRX] ⊖ 1 → [XRX]

4) E → [PC]

5) Done.

9.11 CBX N,T Cycle BR and Shift Into XR5 Type 2

(Refer to 9.14)

1) $0 \rightarrow [XR5]$

2) If $E_s = 1$, then go to 8

3) If E = 0, then go to 14

4) $[XR5]_{i+1} \rightarrow [XR5]_i$, i = 0:14

5) $[BR]_s \rightarrow [XR5]_{15}$

6) $[BR]_{i+1} \rightarrow [BR]_i$, i = 0:22; $[BR]_s \rightarrow [BR]_{23}$

7) E - 1 → E; go to 3

8) $\overline{E} \rightarrow E$

9) If E = 0, then go to 14

10) $[XR5]_{i+1} \rightarrow [XR5]_i$, i = 0:14

11) $[BR]_{23} \rightarrow [XR5]_{15}$

12) $[BR]_i \rightarrow [BR]_{i+1}$, i = 0:22; $[BR]_{23} \rightarrow [BR]_s$

13) E - 1 → E; go to 9

14) Done.

9.12 CMA R,T Compare AC Type 1

{Step the [PC] 0, 1, or 2 times according as [AC] <, =, or > than [E]; and [AC] → [BR].}

1) [AC] ⊖ [E] → [BR]
2) If [BR] = ± 0, then step PC, and go to 6
3) If $[AC]_s$ = 0 and $[E]_s$ = 1, then step PC twice, and go to 6
4) If $[AC]_s$ = 1 and $[E]_s$ = 0, then go to 6
5) If $[BR]_s$ = 0, then step PC twice
6) [AC] → [BR]

9.13 CMX I,X Compare XR with I Type 1

Step [PC] 0, 1, or 2 times according as [XRX] <, =, or > E.

The 2^{16} values $0,\cdots,2^{16}-1$ which [XRX] or E may take are distinct, and increasing in the order given.

9.14 Definition: Cycle the n-bit register α E times, where E is a 7-bit signed (1's complement) quantity. If $E_s = 1$, cycle right; if $E_s = 0$, cycle left. Note that E ≡ [CCAR].

1) If $E_s = 1$, then go to 5

2) If E = 0, then go to 9

3) $[\alpha]_{i+1} \rightarrow [\alpha]_i$, i = 0:n-2; $[\alpha]_s \rightarrow [\alpha]_{n-1}$

4) E - 1 → E; go to 2

5) $\overline{E} \rightarrow E$

6) If E = 0, then go to 9

7) $[\alpha]_i \rightarrow [\alpha]_{i+1}$, i = 0:n-2; $[\alpha]_{n-1} \rightarrow [\alpha]_s$

8) E - 1 → E; go to 6

9) Done.

9.14.1 CYA N,T Cycle the AC Type 2

Cycle the AC E times (cf. 9.14).

9.14.2 CYB N,T Cycle the BR Type 2

Cycle the BR E times (cf. 9.14).

9.14.3 CYC N,T Cycle Combined AC and BR Type 2

1) Consider the 48-bit register CL:

$CL_{0:23} \equiv AC_{0:23}$; $CL_{24:47} \equiv BR_{0:23}$

2) Cycle CL E times (cf. 9.14).

9.15 DAC R,T Deposit AC Type 1

1) $[AC]_{0:23} \rightarrow [E]_{0:23}$

2) $[E]_m$ unchanged

9.16 DAD R,T Double Precision Add Type 1

1) Let $Y_i = [E]_i$, i = 0:23;

   $Y_{i+23} = [E \oplus 1]_i$, i = 1:23

2) $0 \rightarrow [VF]$

3) $Z = Y \oplus [CA]$

4) If $Y_s = [CA]_s \neq Z_s$, then $1 \rightarrow [VF]$ and $1 \rightarrow [VFP]$

5) $Z \rightarrow [CA]$

6) $[CA]_s \rightarrow [BR]_s$

9.17 DAM R,T Deposit AC, Masked Type 1

$([AC] \wedge [BR]) \vee ([E] \wedge \overline{[BR]}) \rightarrow [E]$

1) If $[BR]_i = 0$, then $[E]_i$ unchanged, i = 0:23

2) If $[BR]_i = 1$, then $[AC]_i \rightarrow [E]_i$, i = 0:23

3) $[E]_m$ unchanged

9.18 DBR R,T Deposit BR Type 1

1) $[BR]_{0:23} \rightarrow [E]_{0:23}$

2) $[E]_m$ unchanged

9.19 DFR I,T Defer Execution Type 1

0 → [PCC]

The effect of this instruction is to lend CPS control to the instruction [E].

9.20 DPX R,X Deposit XR Type 1

1) $[XRX] \rightarrow [E]_a$

2) $[E]_{0:7}$ and $[E]_m$ unchanged

9.21 DSU R,T Double Precision Subtract Type 1

1) Let $Y_i = \overline{[E]}_i$, i = 0:23;

$Y_{i+23} = \overline{[E \oplus 1]}_i$, i = 1:23

2) Proceed as in steps 2:6 of DAD (cf. 9.16).

9.22 DVD R,T Divide and Skip Type 1

[CA] is divided by [E] to form a quotient, [AC], and a remainder, [BR], as follows:

1) $SQ = [CA]_s \oplus [E]_s$; $SR = [CA]_s$

2) If abs[E] = 0, or
   abs[AC] $\geq$ abs[E] $\times 2^{23}$,
   then $[AC]_s \rightarrow [BR]_s$ and go to 8

3) Step PC

4) Compute the (unique) integers Q and R such that $0 \leq R < \text{abs}[E]$, and
   abs[CA] = (Q $\times$ abs[E]) + R*

5) $Q \rightarrow [AC]$; $R \rightarrow [BR]$

6) If SQ = 1, then $\overline{[AC]} \rightarrow [AC]$

7) If SR = 1, then $\overline{[BR]} \rightarrow [BR]$

8) Done.

---

* Note that $0 \leq R < \text{abs}[E] \leq 2^{23}-1$, and $Q < 2^{23}$, since abs[E] > 0 and Q $\times$ abs[E] = abs[CA] - R $\leq$ abs[CA] $< \text{abs}[E] \times 2^{23}$.

9.23 EOR R,T Exclusive "Or" Type 1

[AC] ∀ [E] → [AC]

9.24 LAC R,T Load AC Type 1

[E] → [AC]

9.25 LBR R,T Load BR Type 1

[E] → [BR]

9.26 LDE R,T Load Effective Address Type 1

1) $[E]_a \oplus [XR[E]_t] \rightarrow [CAR]$

2) $1 \rightarrow [CCI]$

9.27 LRB J,T Load RL and BD Type 5

1) $E \equiv [CAR]$

2) If $[PR] = 1$, then go to 5

3) $E_{8:11} \rightarrow [RL]_{0:3}$; note that $[RL]_{4:15} \equiv 0$

4) $E_{12:15} \rightarrow [BD]_{0:3}$; note that $[BD]_{4:15} \equiv 0$

5) Done.

LXI

9.28 LXI I,X Load XR Immediate Type 1

E ⇒ [XRX]

9.29 LXR, R,X Load XR Remote Type 1

$[E]_a \rightarrow [XRX]$

9.30 MMN X Make Metabit One Type 3

1) $1 \rightarrow [[XRX]]_m$

2) $0 \rightarrow [MTC]$

3) $[[XRX]]_{0:23}$ unchanged

9.31 MMZ X Make Metabit Zero Type 3

1) $0 \rightarrow [[XRX]]_m$

2) $0 \rightarrow [MTC]$

3) $[[XRX]]_{0:23}$ unchanged

9.32 MPY R,T Multiply Type 1

[AC] is multiplied by [E] to form a 47-bit product [CA] as follows:

1) $P_s = [AC]_s \forall [E]$

2) $Y = \text{abs}[AC] \times \text{abs}[E]$ *

3) $Y \rightarrow [CA]$

4) If $P_s = 1$, then $\overline{[CA]} \rightarrow [CA]$

5) $[CA]_s \rightarrow [BR]_s$

---

* abs[AC] × abs[E] is a 47-bit quantity Y; in fact, $0 \leq Y \leq 2^{46}-2^{24}+1$.

9.33 NAL X Normalize AC Left Type 3

1) If X = 6 or X = 7, then go to 8

2) $0 \rightarrow k$

3) If $[AC]_s \neq [AC]_1$, then go to 7

4) Shift AC left 1, i.e. $[AC]_i \rightarrow [AC]_{i-1}$,
$i = 2:23$; $[AC]_s \rightarrow [AC]_{23}$

5) $k + 1 \rightarrow k$

6) If $k < 23$, then go to 3

7) $[XRX] \ominus k \rightarrow [XRX]$

8) Done.

9.34 NAR X Normalize AC Right Type 3

1) If X = 6 or X = 7, then go to 6

2) If [VF] = 0, then go to 6

3) Shift AC right 1, i.e. $[AC]_i \rightarrow [AC]_{i+1}$, i = 0:22

4) $\overline{[AC]_s} \rightarrow [AC]_s$

5) $[XRX] \oplus 1 \rightarrow [XRX]$

6) Done.

9.35 NCL X Normalize CA Left Type 3

1) If X = 6 or X = 7, then go to 9

2) $0 \rightarrow k$

3) If $[CA]_s \neq [CA]_1$, then go to 7

4) Shift CA left 1, i.e. $[CA]_i \rightarrow [CA]_{i-1}$, $i = 2{:}46$; $[CA]_s \rightarrow [CA]_{46}$

5) $k + 1 \rightarrow k$

6) If $k < 46$, then go to 3

7) $[XRX] \ominus k \rightarrow [XRX]$

8) $[AC]_s \rightarrow [BR]_s$

9) Done.

9.36 NCR X Normalize CA Right Type 3

1) If X = 6 or X = 7, then go to 7

2) If [VF] = 0, then go to 6

3) Shift CA right 1, i.e. $[CA]_i \rightarrow [CA]_{i+1}$, i = 0:45

4) $\overline{[CA]_s} \rightarrow [CA]_s$

5) $[XRX] \oplus 1 \rightarrow [XRX]$

6) $[AC]_s \rightarrow [BR]_s$

7) Done.

9.37 NOP No Operation

A variety of "instructions" when performed have no effect other than those common to all instructions. All of the presently unused op-codes have this property at the present time, but one should not suppose that this will necessarily remain the case.

If one requires a NOP, there are a variety of instructions that will always have this effect, e.g.:

```
LXI I,0          (Type 1)
```

9.38 NVX X Invert XR Type 3

1) If X = 7, then go to 3

2) $\overline{[XRX]} \rightarrow [XRX]$

3) Done.

9.39 PAX X Put AC Into XR Type 3

$[AC]_a \rightarrow [XRX]$

9.40 PER J,T Operate Type 5

This class of "operate" instruction is primarily designed to facilitate maintenance of the machine; all PER instructions are equivalent to "NOP" when [PR] = 1.

9.41 PXA X Put XR Into AC Type 3

$[XRX] \rightarrow [AC]_a$

9.42 RIS Restore Interrupt Status Type 4

1) If [PR] = 1, then go to 10

2) $[4]_{3:5} \rightarrow [MC]_{3:5}$

3) $[4]_{6} \rightarrow [PCC]$

4) $[4]_{7} \rightarrow [CCI]$

5) $[4]_{a} \rightarrow [PC]$

6) $[5]_{0:6} \rightarrow [CCAR]$

7) $[5]_{a} \rightarrow [CAR]$

8) $1 \rightarrow [PR]$

9) Done.*

10) Done.

---

* If control reaches 9, machine control will by-pass that portion of the instruction cycle dealing with metabit traps and interrupts; thus, an interrupt will never take place immediately upon the completion of an "unprotected" RIS. If the conditions for an interrupt obtain, the interrupt will not occur until after the completion of the next instruction.

Note: From Figures 4.2.1 and 4.2.2 we see that an RIS will never cause a metabit trap.

9.43 SAS R,T Skip if AC Same, Masked Type 1

If ([AC] ∧ [BR]) = ([E] ∧ [BR]), then step PC.

9.44 Definition: Shift the n-bit register α E times, where E is the 7-bit signed (1's complement) quantity [CCAR]. If $E_s = 0$, shift left; if $E_s = 1$, shift right. Note that $E \equiv$ [CCAR].

1) $0 \rightarrow$ [VF]
2) If $E_s = 1$, then go to 7
3) If $E = 0$, then go to 11
4) If $[\alpha]_s \neq [\alpha]_1$, then $1 \rightarrow$ [VF] and $1 \rightarrow$ [VFP]
5) $[\alpha]_{i+1} \rightarrow [\alpha]_i$, i = 1:n-2; $[\alpha]_s \rightarrow [\alpha]_{n-1}$
6) $E - 1 \rightarrow E$; go to 3
7) $\overline{E} \rightarrow E$
8) If $E = 0$, then go to 11
9) $[\alpha]_i \rightarrow [\alpha]_{i+1}$, i = 0:n-2
10) $E - 1 \rightarrow E$; go to 8
11) Done.

9.44.1 SHA N,T Shift AC Type 2

Shift AC E times (cf. 9.44).

9.44.2 SHB N,T Shift BR Type 2

Shift BR E times (cf. 9.44).

9.44.3 SHC N,T Shift CA Type 2

Shift CA E times (cf. 9.44).

9.45 SIO I,T Start I/O Operation Type 1

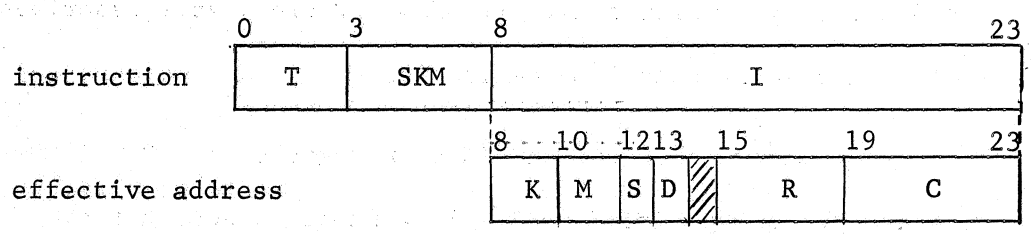
This instruction is a NOP in protected status ([PR] = 1). For its action in unprotected status, see Chapter II.

9.46 SKM I,T Skip -Make Type 1

9.46.1 Description. Every computer has a more or less heterogeneous collection of instructions used to perform conditional skips or branches and for manipulating the contents of special registers. In PHOENIX, these operations are performed by a single instruction, SKM, whose operation is controlled by subfields of its effective address. This instruction fetches the contents of a live register, prepares conditionally to skip the instruction following the SKM, conditionally modifies the word obtained, and stores the modified word in a live register. For example, the following can be done using SKM instructions:

1) Load AC from CK
2) Store AC into MC
3) Skip if AC is positive or -0
4) Skip if BR_{16} is zero
5) Set MC_4 to one
6) Complement BR if it is negative

The 16-bit effective address E of an SKM instruction is divided into six subfields: K(skip), M(make), S(source), D(destination), R(register), and C(condition).



The contents of these subfields determine the operation of SKM as follows:

1) Fetch the contents W of the source register specified by S and R.

2) Test W as specified by K and C to determine whether or not to skip.

3) Modify W as specified by M and C.

4) Store the new W in the destination register specified by D and R.

5) Skip if so indicated by the test performed in step 2.

If S=0 then the source register is AC. If S=1 then the source is specified by R. Each of the 16 values that R can take on specifies a different live register as source. In one case, W is not simply the contents of the R-th register. Table 9.46.3 describes precisely how each register acts when used as a source, in protected status and in unprotected status.

K and C together determine the condition which, if satisfied, results in a skip. Table 9.46.2 describes under what conditions a skip occurs for each C-K combination.

M and C together determine how W is modified. The possibilities are: setting a bit of W to one or zero, complementing a bit of W, and conditionally complementing all of W. Table 9.46.2 describes the modification caused by each C-M combination.

If D=0 then the destination register is AC. If D=1 then the destination is specified by R. In general, W does not simply replace the contents of the destination register. Table 9.46.3 describes precisely how each register acts when used as a destination, in protected status and in unprotected status.

Table 9.46.2 shows when a skip occurs for each C-K (condition-skip) condition, and also shows the modification performed on W for each C-M (condition-make) combination. In the table,"¢W" is used as an abbreviation for "complement W".

C	K	a skip occurs:	M	modification of W:
0:23	0	never	0	none
	1	if $W_c=0$	1	$1 \rightarrow W_c$
	2	if $W^c=1$	2	$\underline{0} \rightarrow W^c$
	3	always	3	$W_c \rightarrow \bar{W}_c$
24	0	never	0	none
	1	if W≠0	1	if W≠0 then ¢W
	2	if W=0	2	if W=0 then ¢W
	3	always	3	¢W
25	0	never	0	none
	1	if W≠-0	1	if W≠-0 then ¢W
	2	if W=-0	2	if W=-0 then ¢W
	3	always	3	¢W
26	0	never	0	none
	1	if $W_s=0$	1	if $W_s=0$ then ¢W
	2	if $W_s=1$	2	if $W_s=1$ then ¢W
	3	always	3	¢W
27	0	never	0	none
	1	if W≠±0	1	if W≠±0 then ¢W
	2	if W=±0	2	if W=±0 then ¢W
	3	always	3	¢W
28	0	never	0	none
	1	if W>+0	1	if W>+0 then ¢W
	2	if W≤+0	2	if W≤+0 then ¢W
	3	always	3	¢W
29	0	never	0	none
	1	if W<-0	1	if W<-0 then ¢W
	2	if W≥-0	2	if W≥-0 then ¢W
	3	always	3	¢W
30:31	0	never	0	none
	1	always	1	¢W
	2	never	2	none
	3	always	3	¢W

TABLE 9.46.2

Table 9.46.3 determines the W that is fetched from each possible source, and what modifications are made to each possible destination, in protected status and in unprotected status.

R	Register	Source, Protected	Destination, Protected	Source, Unprotected	Destination, Unprotected
-	AC, Accumulator	[AC]→W	W→[AC]	[AC]→W	W→[AC]
0	BR, B-register	[BR]→W	W→[BR]	[BR]→W	W→[BR]
1	CK, Clock	[CK]→W	[CK] unchanged	[CK]→W	W→[CK]
2	MC, Misc. CPS	[MC]→W	W→[MC]	[MC]→W	W→[MC]
3	EN, Enable	[EN]→W	[EN] unchanged	[EN]→W	W→[EN], $1\rightarrow[EN]_{20}\rightarrow[EN]_{22}$
4	NR, Interrupt	[NR]→W	[NR] unchanged	[NR]∧[EN]→W, [NR]∧$\overline{[EN]}$→[NR]	W∨[NR]→[NR]
5	WC, Word Count	$0\rightarrow W_{0:7}$, [WC]→W_a	[WC] unchanged	$0\rightarrow W_{0:7}$, [WC]→W_a	[WC] unchanged
6	MP, Misc. PUP	[MP]→W	[MP] unchanged	[MP]→W	$W_{1:2}\vee[MP]_{1:2}\rightarrow[MP]_{1:2}$
7	TS, Tape Status	[TS]→W	[TS] unchanged	[TS]→W	$W_{12:23}\vee[TS]_{12:23}\rightarrow[TS]_{12:23}$
8	LI, LSB Input Status	[LI]→W	[LI] unchanged	[LI]→W	[LI] unchanged
9	LO, LSB Output Status	[LO]→W	[LO] unchanged	[LO]→W	[LO] unchanged
10	LB, LSB Buffer Status	[LB]→W	[LB] unchanged	[LB]→W	[LB] unchanged
11	LA, LSB Attention	[LA]→W	[LA] unchanged	[LA]→W	W∨ [LA]→[LA]
12	LE, LSB Error	[LE]→W	[LE] unchanged	[LE]→W	W∨[LE]→[LE]
13	Spare	--	--	--	--
14	Zero	0→W	no action	0→W	no action
15	EX, External Register	if [EXM]=1 then [EX]→W and 0→[EX], otherwise 0→W and [EX] unchanged	if [EXM]=1 then W→[EX], otherwise [EX] unchanged	[EX]→W and 0→[EX]	W→[EX]

Table 9.46.3

9.47 SMN X Skip If Metabit One Type 3

1) If $[[XRX]]_m = 1$, then step PC

2) $0 \rightarrow [MTC]$

9.48 SMZ X Skip If Metabit Zero Type 3

1) If $[[XRX]]_m = 0$, then step PC

2) $0 \rightarrow [MTC]$

9.49 SUB R,T Subtract Type 1

1) $0 \rightarrow [VF]$

2) $Y = [AC] \ominus [E]$

3) If $[AC]_s = \overline{[E]}_s \neq Y_s$, then $1 \rightarrow [VF]$ and $1 \rightarrow [VFP]$

4) $Y \rightarrow [AC]$

9.50 TRP Trap Type 6

1) $[PC] \rightarrow [2]_a$

2) $3 \rightarrow [PC]$

3) $[2]_{0:7}$ and $[2]_m$ unchanged

Note that if the CPS is in protected status, [PC] are stored in memory location 2 + [RL].

Note also that a metabit trap is possible as a result of a TRP (if $[2]_m = 1$).

9.51 **VOR** R,T Inclusive "Or" Type 1

$[AC] \vee [E] \rightarrow [AC]$

BLANK PAGE

10.0 PHOENIX INITIALIZATION

The following sections describe the basic machine initialization from the programming point-of-view.

10.1 Master Reset Condition

We define a machine status which we call "master reset condition":

1) $0 \rightarrow$ [all live registers except those listed below]

2) $-0 \rightarrow [WC]$

3) $1 \rightarrow [PI] \rightarrow [EN]_{20} \rightarrow [EN]_{22}$

4) TCU and DRM circuits reset

10.2 "Load from Drum" Initialization

A "Load from Drum" initialization consists of the following steps:

1) Master reset

2) Transfer 119 words from the drum (starting location 8) to central memory (starting location 8) using standard DRM transfer operation with operation as if $CW2_4$ and $CW3_{0:3}$ were zero, (cf. 17.7, 17.8).

3) Wait for PUP operation complete signal ($[NR]_9 = 1$)

4) $16_{10} \rightarrow [PC]$

5) Place CPS into normal operation, (with $[PR] = 0$)

10.3 "Load from Tape" Initialization

A "Load from Tape" initialization consists of the following steps:

1) Master reset

2) Transfer one file up to a maximum of 119 words from CDC tape unit 0 into central memory (starting address 8) using a standard TCU operation as if $CW2_{11:19}$ and $CW3_{0:3}$ were zero (cf. 17.5,17.8) ($CW2_{18:19} = 0$ yields 556 bits/inch).

3) Wait for PUP operation complete signal. ($[NR]_9 = 1$)

(Normal "tape operation complete" interrupt bits also set.)

4) $16_{10} \Rightarrow [PC]$.

5) Place CPS into normal operation; (with [PR] = 0)

CHAPTER II -- I/O

11.0 DEFINITIONS

I/O equipment, be it PUP, a channel, or a device, is operative if it is capable of carrying out its intended functions under program control, otherwise it is inoperative; it is idle if it is operative but not carrying out an operation; it is busy, if it is actually carrying out an operation.

12.0 INTRODUCTION

An I/O operation begins with the execution of an SIO instruction. (If the CPS is in protected status or PUP is busy, the SIO will be treated as a NOP.) The SIO instructs PUP to begin the operation. PUP in turn instructs a channel which may be a device (ADC or DRM), or which may instruct a device (LSB, TCU, or HSC).

If the operation involves the transmission of data between a device and central memory, the path is: device → channel → PUP → central memory in the case of an input operation; central memory ⇒ PUP channel → device in the case of an output operation.

Execution of the SIO instruction will cause PUP to become busy immediately. If the selected channel is busy, PUP will wait until it becomes idle, at which time PUP will instruct the channel, causing the channel to become busy. If the device is inoperative or busy, an appropriate interrupt bit will be set and the I/O operation will be terminated, idling both the channel and PUP. If the device is idle, I/O operation will begin.

Whenever PUP, a device, or a channel completes its task in an I/O operation, it disconnects, i.e., it becomes idle and sets an appropriate interrupt bit.

If an error condition arises during an I/O operation, an appropriate interrupt bit is set.

The execution of an operation involves PUP only as long as information is flowing between channel and memory. This flow can cease, PUP becoming idle, before the I/O operation is complete. In addition, a channel (TCU) may become idle before an I/O operation is complete, (i.e., Rewind or Rewind and Unload).

13.0 I/O OPERATIONS

There are four modes of I/O operation: NORMAL, SORT, SEARCH, and MULTIPLE SEARCH. NORMAL operations further subdivide into three classes: READ, WRITE, and NON-DATA. The SORT, SEARCH, and MULTIPLE SEARCH modes are only effective as READ operations.

NORMAL operations require 3 control words to specify them, which we refer to as CW1, CW2, and CW3. SORT requires 4 control words: CW1, ACW2, CW2, and CW3. SEARCH and MULTIPLE SEARCH require 5 control words: CW1, ACW1, ACW2, CW2, and CW3.

CW1 specifies mode, channel, and word count, as shown in 17.2.

CW2 specifies information which differs from channel to channel, as shown in 17.3 - 17.7.

CW3 specifies central memory starting address, behavior of address counter, and behavior of data metabits.

ACW1 is the key for search operations.

ACW2 is the mask for sort and search operations.

I/O operations are described in detail in 17.1 and by annotated flow charts 17.9 - 17.16. A sketch of these operations is given below. Details concerning the operations of each channel are given in 14 and 15. Interrupts are discussed in 16.

A Normal data operation moves data words between central memory and a channel. Locations of data in central memory are specified by the starting address, together with the address counter control bits. The number of words transmitted is limited by the word count. Modification of data metabits is specified by the metabit control bits.

A Normal non-data operation causes the channel to perform an operation involving the transfer of no information other than that contained in the control words.

A Sort operation moves data records from a channel into central memory. A data word begins a record if its metabit is one, as received from the channel. The first word of each record is compared with the mask (ACW2) to determine how the record is to be treated. If the result of anding the word and mask together is zero, then the record is loaded into locations determined by the address counter, (generally, into increasing locations in memory). If the result is non-zero, then the record is loaded into decreasing locations in memory determined by the word counter. The operation ceases when the two counters have the same contents.

A Search or Multiple Search operation examines the first word of each data record input from the channel, to determine if it matches the key (ACW1). A data word begins a record if its metabit is one, as received from the channel. The data word and key are compared only for those bit positions of the words at which the mask (ACW2) has a one. If a match occurs, the matching data word and succeeding data words of that record are put in central memory as in a normal read. A non-multiple search operation will be terminated if the first word of a new record is encountered. During a Multiple Search, each record encountered is

processed; it is transmitted if its first word matches. Indicators are provided to tell whether a match was ever found, and whether at least one whole record was input.

Either type of search will terminate if the total number of words received (not necessarily transmitted to central memory) becomes equal to the specified word count.

14.0 CHANNEL AND DEVICE OPERATION

14.1 ADC (Advanced Display Console)

See Working Paper W-5272, "Proposed Display Console for FIFI", for details.

14.2 LSB (Low Speed Buffer)

See Chapter III, and Working Paper W-6323.

For LSB, I/O operations do not directly affect I/O devices, but rather affect the contents of LSB memory or I/O status bits. The LSB is itself only transiently "busy". Its devices communicate with LSB memory, and their busy/idle condition is only remotely related to PUP-LSB operations.

14.3 TCU (Tape Control Unit)

If the selected device (tape unit) is inoperative or busy, the TCU will disconnect immediately.

The conditions for TCU and devices to become idle is discussed in 15. Whenever TCU becomes idle after having been busy, the "TCU operation complete" bit is set in the interrupt register: $1 \rightarrow [NR]_7$.

14.4 HSC (High Speed Channel)

The high speed channel shares hardware with the TCU. Therefore, both the TCU and the HSC are either both busy or both idle at any given time.

14.5 DRM (Drum)

The drum is busy as long as PUP is busy with the operation. The drum is capable of storing 278,528 words. The words are stored in 34 drum fields, each of which is divided into 8,192 angular positions. The initial drum field address is specified by bits 5:10 of IOCW2 and the initial angular

position address is specified by bits 11:23 of IOCW2. After every word transfer, the angular position address is incremented by 1, with carry added to the drum field address.

The angular velocity of the drum is 3600 rpm giving an average access time of 8.333 ms and a word transfer time of 2.08 μs.

14.6 Spare Channels

PUP interfaces are provided for two additional channels. These channels are expected to accept and transmit standard 26-bit PHOENIX words. All control functions are provided except decoders which determine the class of I/O operation from IOCW2.

15.0 TCU AND TAPE UNITS

The format of information on tapes is that of IBM standard binary tapes except that the density of recorded information is either 556 characters/inch (low density) or 800 characters/inch (high density).

Ten operations can be performed on a TCU device (tape unit):

15.1 Read File

Data words are read from the selected tape and transmitted to PUP until either: 1) The number of words specified by the word count have been transmitted, or 2) An EOF (end-of-file record) has been read.

In case 1, TCU continues to move the tape until an EOF has been read.

After 2 has been satisfied, TCU and the device become idle and the following interrupt bits are set:

15.1.1 $1 \rightarrow [NR]_3$ if a redundancy error occurred during the operation.

15.1.2 $1 \rightarrow [NR]_4$

15.1.3 $1 \rightarrow [NR]_7$

15.2 Read Record

Data words are read from the selected tape and transmitted to PUP until either: 1) The number of words specified by the word count have been transmitted, or 2) An EOF has been passed, or 3) An inter-record gap is entered.

In case 1, TCU continues to move the tape until either 2 or 3 is satisfied.

When 2 or 3 is satisfied, TCU and the device become idle, and the following interrupt bits are set:

15.2.1 $1 \rightarrow [NR]_3$ if a redundancy error occurred during the operation.

15.2.2 $1 \rightarrow [NR]_4$ if an EOF was read.

15.2.3 $1 \rightarrow [NR]_7$ in any case.

15.3 Backspace Record

The selected tape is moved backward until either: 1) An inter-record gap is entered, or 2) The load point is reached. At this time TCU and the device become idle and $1 \rightarrow [NR]_7$.

15.4 Backspace File

The selected tape is moved backward until either: 1) An EOF record is passed over, or 2) The load point is reached. At this time TCU and the device become idle and $1 \rightarrow [NR]_7$.

15.5 Rewind

The selected tape is set in backward motion, TCU becomes idle, and $1 \rightarrow [NR]_7$. When the tape reaches its load point, it stops moving, becomes idle, and $1 \rightarrow [NR]_{10}$ (CDC tapes only). At this time, the tape status indicator register will indicate that the unit is idle and at load point (cf. 2.28) (CDC tapes only).

15.6 Rewind and Unload

Rewind and Unload is the same as Rewind, except that when the load point is reached, the device becomes inoperative (tape unloaded) rather than idle.

15.7 Write End-of-File

An end-of-file record is written on the selected tape, and checked for redundancy error, after which TCU and the device become idle, and the following interrupt bits are set:

15.7.1 $1 \Rightarrow [NR]_3$ if a redundancy error occurred during the operation.

15.7.2 $1 \Rightarrow [NR]_4$ if an end-of-tape condition occurred during the operation.

15.7.3 $1 \Rightarrow [NR]_7$

15.8 Write Record

Data words are obtained from PUP and written on the selected tape until PUP disconnects (13.1.13). The tape continues in motion until the longitudinal redundancy check bits are written, and the written information is checked. At this time the TCU and device become idle and the following interrupt bits are set:

15.8.1 $1 \Rightarrow [NR]_3$ if a redundancy error occurred during the operation.

15.8.2 $1 \Rightarrow [NR]_4$ if an end-of-tape condition occurred during the operation.

15.8.3 $1 \Rightarrow [NR]_7$

15.9 Blank Tape and Write

An empty space is written on the selected tape. The length of this space depends on a machine adjustment; it is nominally 2.25 inches. Then a record is written exactly as by a Write Record (15.9).

15.10 No Operation

The TCU and selected tape become idle immediately. $1 \Rightarrow [NR]_7$.

16.0 I/O INTERRUPTS

16.1 I/O Faults, $[NR]_1$

If any of the fault conditions noted by $[MP]_{6:23}$ (cf. 2.19) occur, then $1 \Rightarrow [NR]_1$. If $[EN]_1=0$ and $[NR]_1=1$, then the machine halts.

16.2 Magnetic Tape Read/Write Error, $[NR]_3$

Set to one if a redundancy error is detected during one of the following TCU operations: Read File, Read Record, Write End-of-File, Write Record, Blank Tape and Write.

16.3 EOT Writing Tape/EOF Reading Tape, $[NR]_4$

Set to one if either:

16.3.1 An end-of-tape condition occurs during a TCU operation Write End-of-File, Write Record, Blank Tape and Write; or

16.3.2 An end-of-file record is read during a TCU operation Read File, Read Record.

16.4 Tape Unit or LSB Device Inoperative, $[NR]_6$

Set to one if either:

16.4.1 One of the following LSB operations is attempted when the selected device is inoperative: Unload Input Buffer, I/O Status Command, Load Output Buffer; or

16.4.2 Any TCU operation is attempted when the selected device is inoperative. Note that a tape unit which has a file-protected tape mounted on it is inoperative for a Write operation, although it may be operative for other operations.

16.5 TCU Operation Complete, $[NR]_7$

Set to one when TCU becomes idle after having been busy.

16.6 LSB Device Busy, Tape Unit Busy, or ADC I/O Exception, $[NR]_8$

Set to one whenever:

16.6.1 One of the following operations is attempted when the selected device is busy: Unload Input Buffer*, Load Output Buffer.

16.6.2 Any TCU operation is attempted when the selected unit is rewinding.

16.6.3 An attempt is made to store into ADC memory in an improper manner or an ADC memory overflow condition is detected.

16.7 PUP Operation Complete, $[NR]_9$

Set to one when PUP becomes idle after having been busy.

16.8 Tape Unit Rewind Complete, $[NR]_{10}$

Set to one when any rewinding CDC tape unit reaches its load point.

16.9 ADC Attention, $[NR]_{11}$

Set to one when the ADC operator makes a manual input or when the operator makes a display area change.

16.10 TCU/HSC Special Device Attention, $[NR]_{12}$

Set to one whenever a "special device attention" pulse is received from one of the four IBM 729 tape unit lines or one of the eight special device lines to the HSC (cf. 2.28).

---

* There is one condition under which an Unload Input Buffer can be performed for a busy device without setting $[NR]_8$: For a typewriter which is in input status and single character mode (cf. Chapter III).

16.11 LSB Device Error, $[NR]_{15}$

Set to one when an error occurs during transmission of a character between LSB and a device, or if a typewriter's input and output status indicators are both set to "one", or if an attempt is made to transfer information between a device and one of the first 32 locations in LSB memory (control word locations).

16.12 Typewriter Attention Indicator, $[NR]_{16}$

Set to one when a Beginning-of-Message character is typed on a typewriter whose attention indicator is a zero while either its BS indicator is a one or its IS indicator a zero.

16.13 Device to LSB Transfer Complete, $[NR]_{17}$

Set to one when any transfer of data from an LSB input device (typewriter or paper tape reader) to LSB memory is completed.

16.14 LSB to Device Transfer Complete, $[NR]_{19}$

Set to one when any transfer of data from LSB memory to an LSB output device (typewriter, paper tape punch, or digital plotter) is completed.

17.0 I/O CONTROL WORDS AND FLOWCHARTS

The annotated flowcharts 17.9 through 17.16 describe the PUP operations involved in performing an I/O operation, from the time the location of the control words is obtained from CPS to the time PUP disconnects. The operations described include Write, Normal Read, Search Read (Simple and Multiple), and Sort Read.

To make the flowcharts more compact, functional notations are used to specify how the metabits of data words are modified during transmission, and how the contents of the Address Counter are modified after a data word is transmitted.

BLANK PAGE

17.1 Control of Address Counter and Data Metabits, and WC

Definition of $\mu(W)$. During data transmission the data word W will be modified to $\mu(W)$ in a way determined by two bits in CW3.

1) $\mu(W)_{0:23} = W_{0:23}$

2) If $CW3_2 = 0$ then $\mu(W)_m = W_m$

3) If $CW3_2 = 1$ then $\mu(W)_m = CW3_3$

Definition of $\alpha(ACT)$. After each data word is transmitted during an I/O operation, [ACT] is changed to $\alpha(ACT)$ in a way determined by two bits in CW3.

1) If $CW3_1 = 1$ then $\alpha(ACT) = [ACT]$

2) If $CW3_1 = 0$ and $CW3_0 = 0$ then $\alpha(ACT) = [ACT]+1$

3) If $CW3_1 = 0$ and $CW3_0 = 1$ then if operation is Sort, then $\alpha(ACT) = [ACT]+1$, otherwise, $\alpha(ACT) = [ACT]-1$.

Word Count (WC) Arithmetic

Note that arithmetic operations performed on [WC] in flowcharts 17.9 through 17.16 are 2's complement operations. The phrase "-0" or "minus zero" is used with the meaning defined in 1.7.3. If [WC]=1, then [WC]-1 = 0. If [WC]=0, then [WC]-1 = "-0" = 177777_8. If [WC]=-0, then [WC]-1 = 177776_8.

CW1 SPECIFIES THE MODE OF OPERATION (NORMAL, SEARCH, OR SORT), THE CHANNEL, AND THE WORD COUNT.

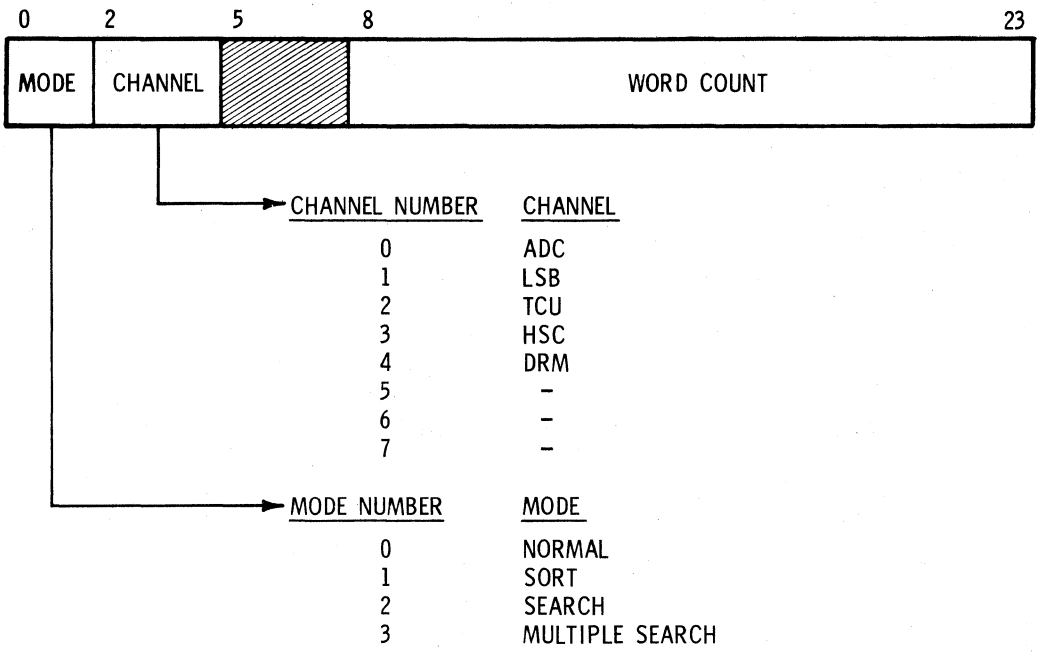
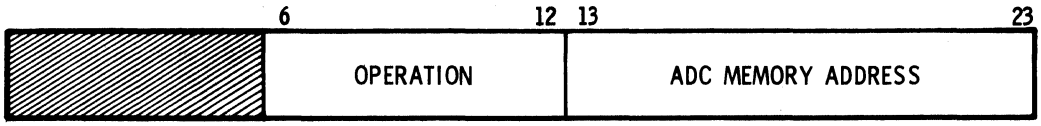


FIG 17.2

CW1 I/O CONTROL WORD 1

CW2 (ADC) SPECIFIES THE OPERATION TO BE PERFORMED, AND THE STARTING LOCATION IN ADC MEMORY.

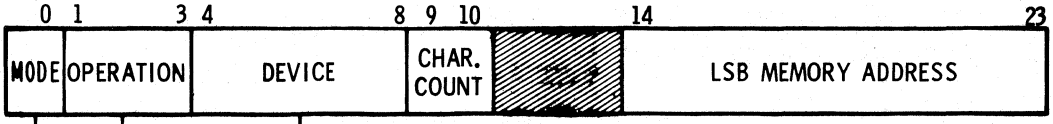


OPERATION CODE (OCTAL)	OPERATION	TYPE
0	IDLE	NON-DATA
1	READ	READ
2	WRITE	WRITE
4	NO STORE	WRITE
10	DISPLAY	NON-DATA
20	MI READ	READ
100	WRITE (FILTER)	WRITE

FIG 17.3

I/O CONTROL WORD 2 FOR ADC

CW2 (LSB) SPECIFIES THE OPERATION TO BE PERFORMED, THE DEVICE SELECTED, THE LSB MEMORY ADDRESS, THE INITIAL VALUE OF CHARACTER COUNT, AND THE MODE.



DEVICE CODE	DEVICE
0:15	TYPEWRITER 0:15
16	PAPER TAPE READER
17	PAPER TAPE PUNCH
18	DIGITAL INCREMENTAL READER
19-31	UNASSIGNED

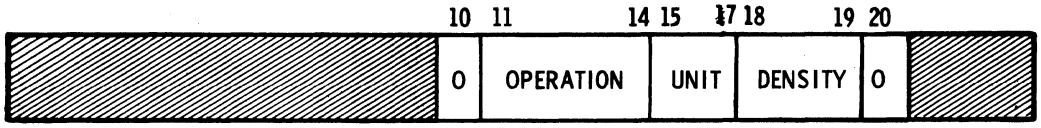
OP CODE	OPERATION	TYPE
0	NO OPERATION	NON-DATA
1	READ MEMORY	READ
2	READ INPUT BUFFER	READ
3	UNLOAD INPUT BUFFER	READ
4	SET INPUT BUFFER	NON-DATA
5	STATUS COMMAND	NON-DATA
6	LOAD MEMORY	WRITE
7	LOAD OUTPUT BUFFER	WRITE

MODE BIT	MEANING (TYPEWRITER INPUT ONLY)
0	MESSAGE MODE
1	SINGLE-CHARACTER MODE

FIG 17.4

I/O CONTROL WORD 2 FOR LSB

CW2 (TCU) SPECIFIES THE OPERATION TO BE PERFORMED, THE UNIT TO BE SELECTED, AND THE DENSITY OF RECORDED INFORMATION.



OPERATION CODE (OCTAL)	OPERATION	TYPE
0	READ FILE	READ
1	READ RECORD	READ
2	—	
3	BACKSPACE RECORD	NON-DATA
4	BACKSPACE FILE	NON-DATA
5	REWIND	NON-DATA
6	REWIND AND UNLOAD	NON-DATA
7	WRITE END OF FILE	NON-DATA
10	—	
11	NOP	NON-DATA
12	WRITE RECORD	WRITE
13	BLANK TAPE AND WRITE RECORD	WRITE
14-17	—	

UNIT-CODE	UNIT
0:3	CDC MAGNETIC TAPE UNIT, TYPE 607, NUMBER 0:3
4:7	IBM MAGNETIC TAPE UNIT, TYPE 729-VI, NUMBER 0:3

DENSITY CODE	DENSITY
0	556 BITS/INCH
1	—
2	800 BITS/INCH
3	—

FIG 17.5

I/O CONTROL WORD 2 FOR TCU

CW2 (HSC) SPECIFIES THE OPERATION TO BE PERFORMED, AND THE DEVICE SELECTED.

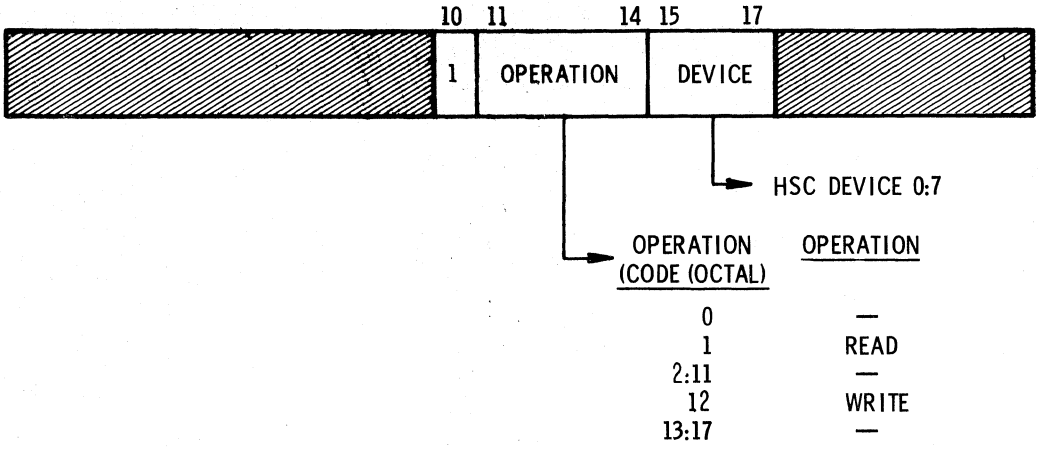


FIG 17.6

I/O CONTROL WORD 2 FOR HSC

CW2 (DRM) SPECIFIES THE TYPE OF OPERATION (READ OR WRITE), AND THE DRUM STARTING ADDRESS.

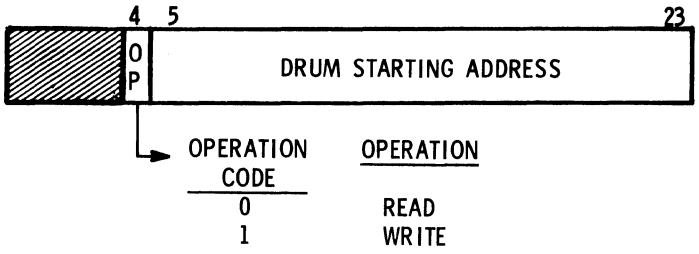


FIG 17.7

I/O CONTROL WORD 2 FOR DRUM

CW3 SPECIFIES THE CENTRAL MEMORY STARTING ADDRESS FOR THE I/O OPERATION, WHETHER THE ACT (ADDRESS COUNTER) WILL BE INCREMENTED, DECREMENTED, OR LOCKED DURING DATA TRANSFER, AND HOW DATA METABITS ARE MODIFIED.

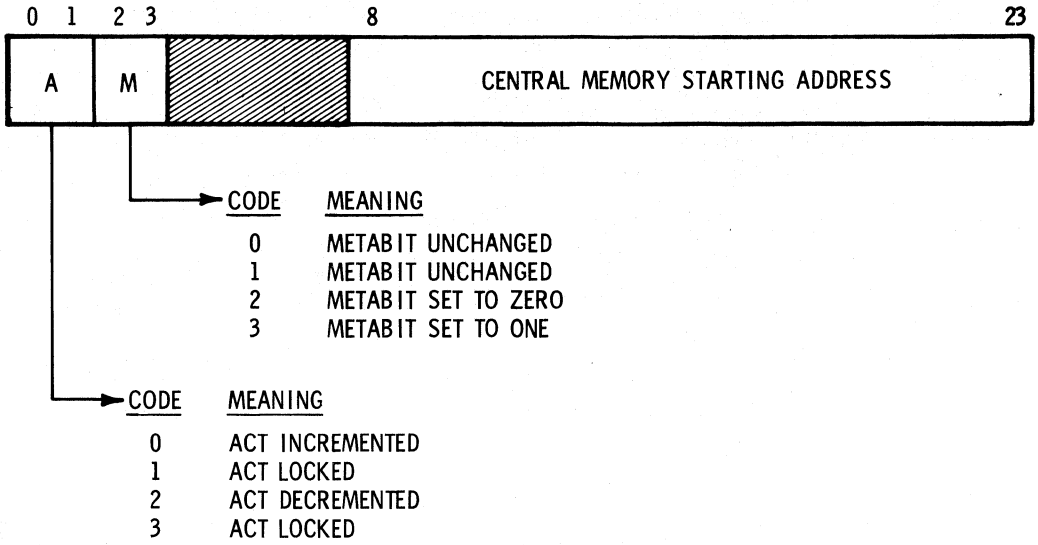
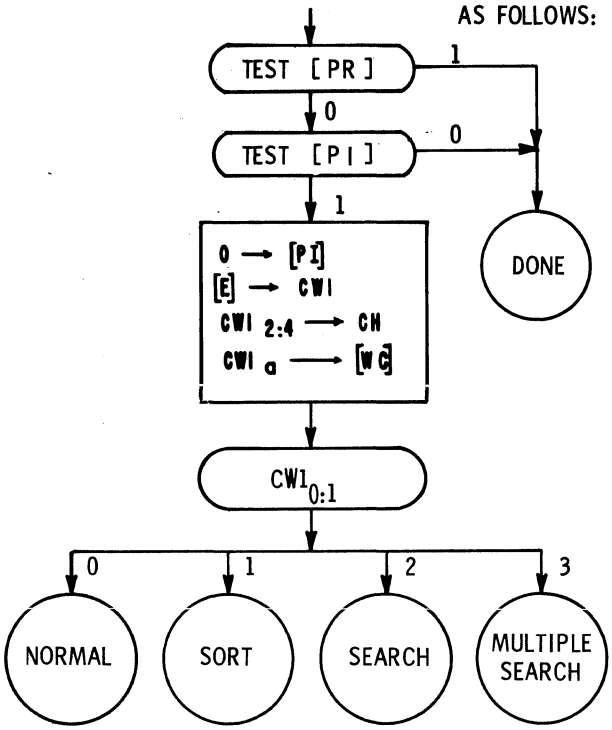


FIG 17.8

CW3 I/O CONTROL WORD 3

WHEN AN SIO INSTRUCTION IS PERFORMED, THE CPS TRANSMITS THE EFFECTIVE ADDRESS E OF THE INSTRUCTION TO PUP, WHICH PROCEEDS AS FOLLOWS:



IF CPS IN PROTECTED STATUS OR PUP NOT IDLE THEN TREAT SIO AS A NOP.

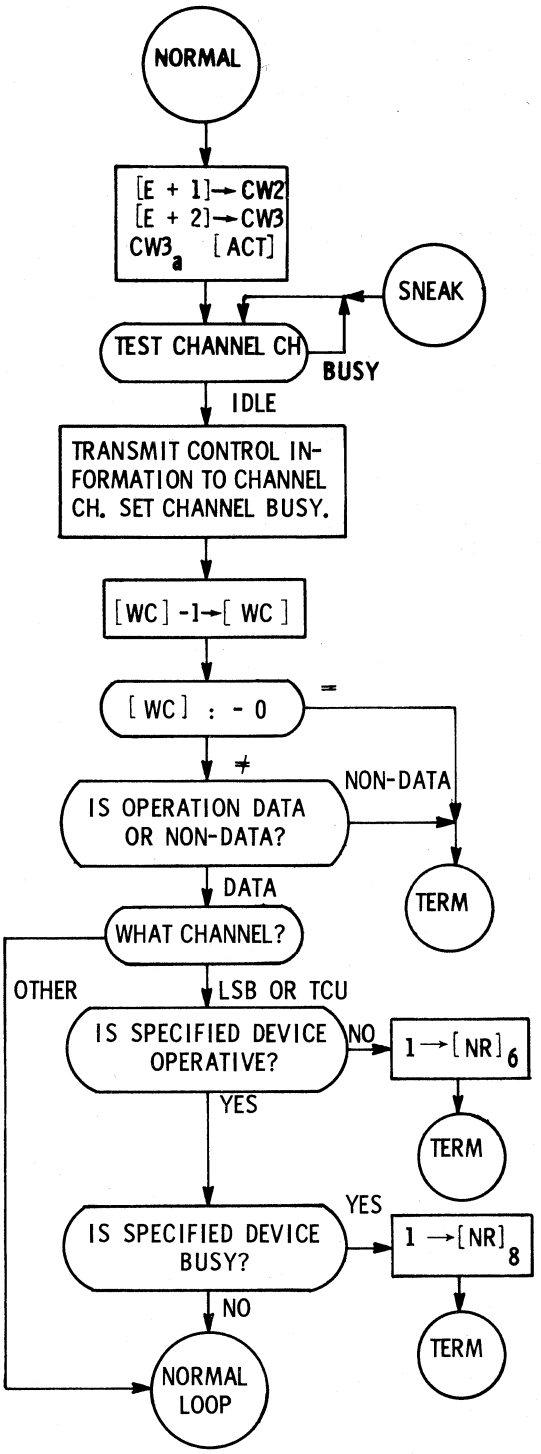
PUP BECOMES BUSY.
FETCH FIRST CONTROL WORD, AND EXTRACT CHANNEL NUMBER AND WORD COUNT.
EXTRACT BITS WHICH DETERMINE METABIT MODIFICATION.

TEST WHETHER THE OPERATION IS NORMAL, SEARCH, OR SORT.

SUBSEQUENT OPERATIONS ARE DESCRIBED ON THE FOLLOWING FLOWCHARTS.

SIO
FIG 17.9

A WRITE OR NORMAL READ PROCEEDS AS FOLLOWS:



FETCH CONTROL WORD 2. DETERMINE IF OPERATION IS READ OR WRITE, OR NON-DATA.
FETCH CONTROL WORD 3.
EXTRACT BITS WHICH CONTROL BEHAVIOR OR ADDRESS COUNTER.
EXTRACT CENTRAL MEMORY STARTING ADDRESS.

WAIT UNTIL SELECTED CHANNEL IS IDLE.

TRANSMIT CONTROL INFORMATION TO SELECTED CHANNEL. IT BECOMES BUSY.

DECREMENT WORD COUNT. (NOTE 2's COMP. ARITH.)

IF WORD COUNT IS -0, OR THE OPERATION IS NON-DATA, THEN GO TERMINATE THE OPERATION.

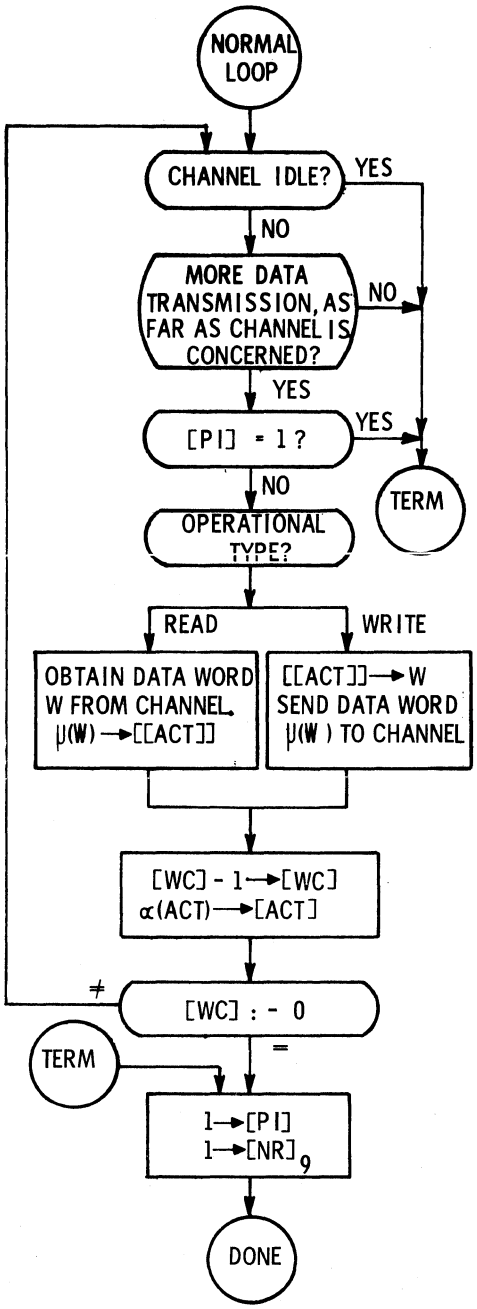
IF CHANNEL IS LSB OR TCU AND SPECIFIED DEVICE IS INOPERATIVE, THEN SET "DEVICE INOPERATIVE" INTERRUPT, AND GO TERMINATE THE OPERATION.

IF CHANNEL IS LSB OR TCU AND SPECIFIED DEVICE IS BUSY, THEN SET "DEVICE BUSY" INTERRUPT, AND GO TERMINATE THE OPERATION.

SUBSEQUENT OPERATIONS ARE DESCRIBED ON THE NEXT FLOWCHART.

FIG 17.10

NORMAL (START)



IF CHANNEL HAS BECOME IDLE, OR HAS INSTRUCTED PUP TO TERMINATE DATA TRANSMISSION, OR CPS HAS INSTRUCTED PUP TO BECOME IDLE, THEN GO TERMINATE THE OPERATION.

PUP TRANSMITS A WORD FROM CHANNEL TO CENTRAL MEMORY OR VICE-VERSA WITH APPROPRIATE METABIT MODIFICATION.

DECREMENT WORD COUNT. CHANGE CONTENTS OF ADDRESS COUNTER.

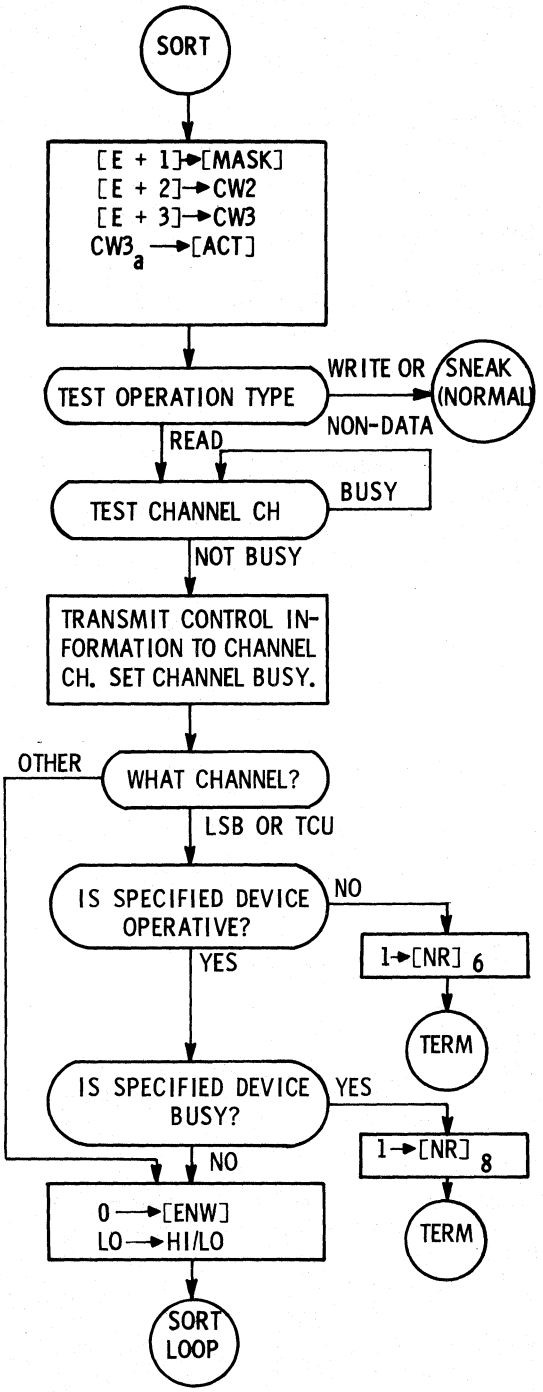
IF WORD COUNT IS NOT ZERO, THEN LOOP.

TERMINATE THE OPERATION: PUP DISCONNECTS, BECOMING IDLE AND SETTING "PUP" OPERATION COMPLETE" INTERRUPT.

PUP IS NOW IDLE, AND READY TO PROCESS A NEW SIO. THE CHANNEL AND DEVICE MAY NOT YET BE IDLE.

FIG 17.11

NORMAL (LOOP)



FETCH MASK.
FETCH CONTROL WORD 2, AND DETERMINE IF OPERATION IS READ, WRITE, OR NON-DATA.
FETCH CONTROL WORD 3. EXTRACT BITS WHICH CONTROL BEHAVIOR OF ADDRESS COUNTER, AND CENTRAL MEMORY STARTING ADDRESS.

IF OPERATION IS WRITE OR NON-DATA, THEN TREAT AS A NORMAL I/O OPERATION.

WAIT UNTIL CHANNEL IS IDLE.

TRANSMIT CONTROL INFORMATION TO CHANNEL. IT BECOMES BUSY.

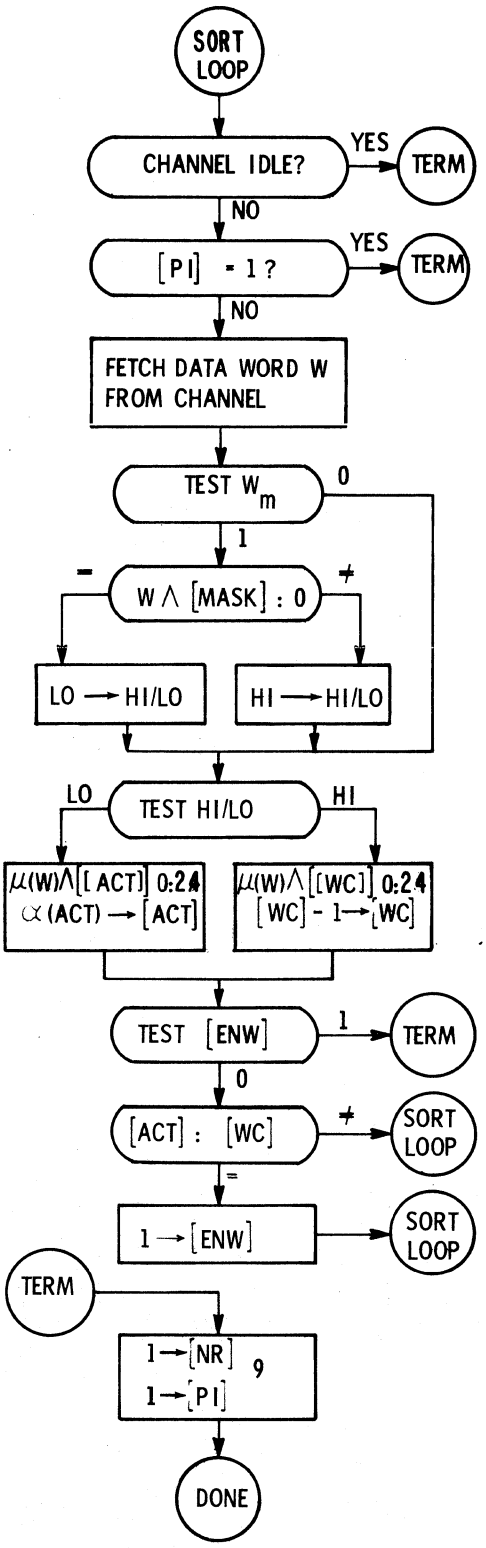
IF CHANNEL IS LSB OR TCU, AND SPECIFIED DEVICE IS INOPERATIVE, THEN SET "DEVICE INOPERATIVE" INTERRUPT, AND GO TERMINATE THE OPERATION.

IF CHANNEL IS LSB OR TCU AND SPECIFIED DEVICE IS BUSY, THEN SET "DEVICE BUSY" INTERRUPT, AND GO TERMINATE THE OPERATION.

INITIALIZE END-CONDITION TEST AND HIGH-LOW RECORD TEST. (ENW = "END NEXT WORD")

SUBSEQUENT SORT OPERATIONS ARE DESCRIBED ON THE FOLLOWING FLOWCHART.

FIG 17.12
SORT (START)



IF THE CHANNEL HAS BECOME IDLE, OR HAS INSTRUCTED PUP TO BECOME IDLE, OR CPS HAS INSTRUCTED PUP TO BECOME IDLE, THEN GO TERMINATE THE OPERATION.

PUP FETCHES A WORD W FROM THE CHANNEL.

IF THE DATA WORD W HAS METABIT ONE, THEN W BEGINS A NEW RECORD.

AND W WITH MASK TO DETERMINE IF RECORD IS TO BE STORED LOW OR HIGH.

REMEMBER WHERE RECORD IS TO BE STORED.

IF RECORD IS LOW, THEN STORE W IN LOCATION SPECIFIED BY ACT, AND CHANGE ACT.
IF RECORD IS HIGH, THEN STORE W IN LOCATION SPECIFIED BY WC, AND DECREMENT WC.
IN BOTH CASES, THE METABIT OF THE STORED DATA WORD MAY BE CHANGED.
IF W IS THE LAST WORD TO BE TRANSMITTED, THEN TERMINATE THE OPERATION.

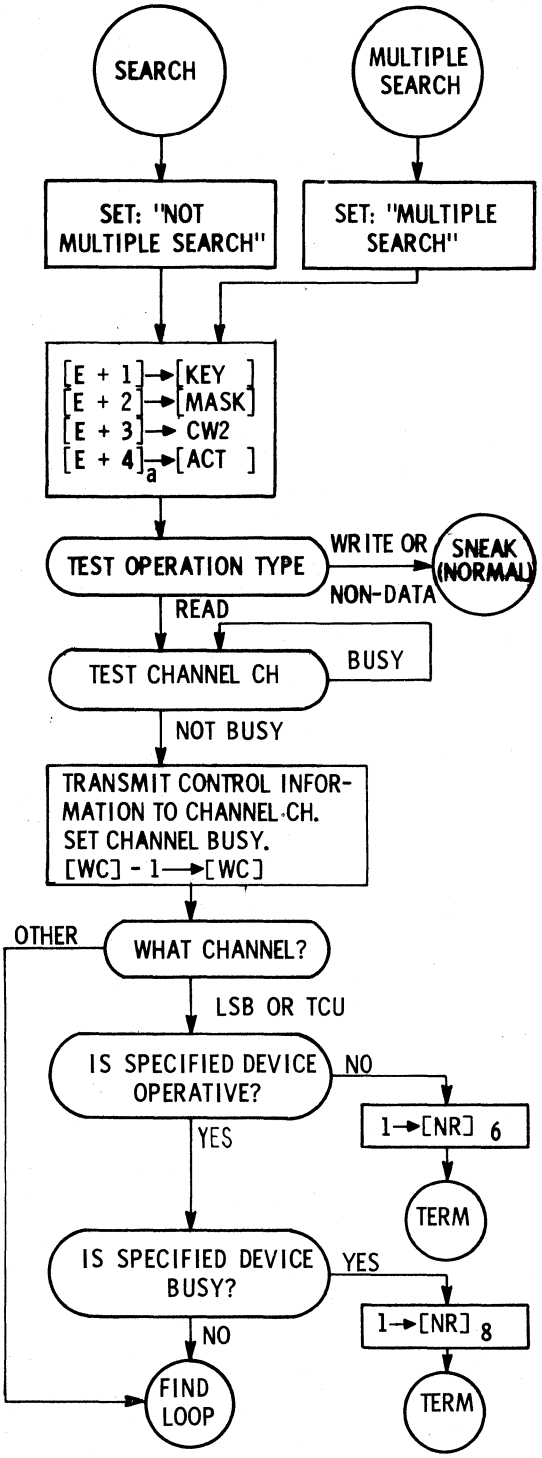
COMPARE ACT AND WC TO DETERMINE IF THE NEXT WORD TO BE TRANSMITTED IS THE LAST.
REPEAT THE SORT LOOP.

TERMINATE THE OPERATION: PUP DISCONNECTS, BECOMING IDLE AND SETTING "PUP OPERATION COMPLETE" INTERRUPT.

PUP IS NOW IDLE, AND READY TO PROCESS A NEW SIO. THE CHANNEL AND DEVICE MAY NOT YET BE IDLE.

FIG 17.13

SORT (LOOP)



SET FOR TYPE OF SEARCH MODE.

LOAD KEY AND MASK REGISTERS FROM ALTERNATE CONTROL WORDS 1 AND 2. FETCH CONTROL WORD 2. DETERMINE WHETHER OPERATION IS READ, WRITE, OR NON-DATA. FETCH CONTROL WORD 3. EXTRACT BITS WHICH CONTROL BEHAVIOR OF ADDRESS COUNTER. EXTRACT CENTRAL MEMORY STARTING ADDRESS, AND LOAD IT INTO ADDRESS COUNTER.

IF OPERATION IS WRITE OR NON-DATA, THEN TREAT IT AS A NORMAL I/O OPERATION.

WAIT UNTIL CHANNEL IS IDLE.

TRANSMIT CONTROL INFORMATION TO CHANNEL. IT BECOMES BUSY. DECREMENT WORD COUNT.

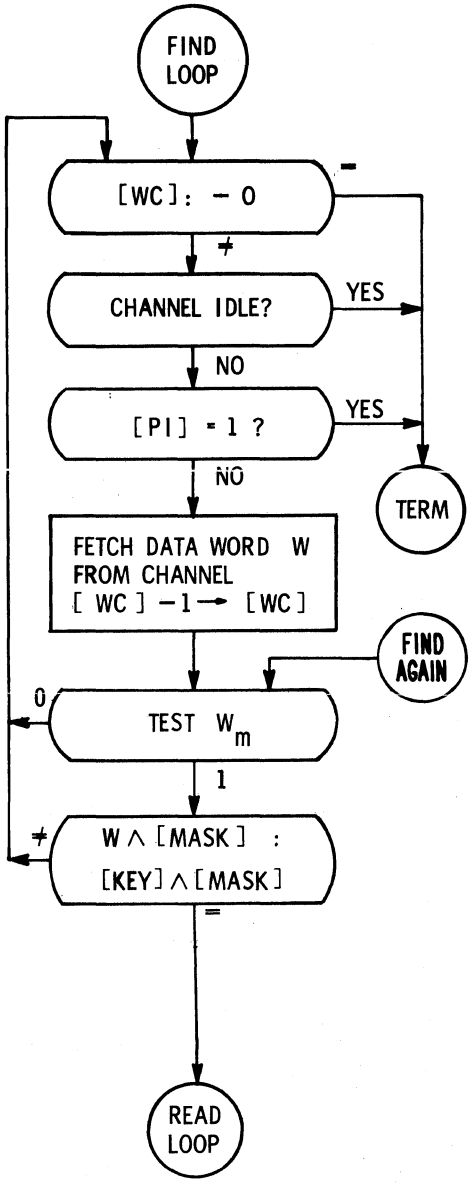
IF CHANNEL IS LSB OR TCU, AND SPECIFIED DEVICE IS INOPERATIVE, THEN SET "DEVICE INOPERATIVE" INTERRUPT, AND GO TERMINATE THE OPERATION.

IF CHANNEL IS LSB OR TCU AND SPECIFIED DEVICE IS BUSY, THEN SET "DEVICE BUSY" INTERRUPT, AND GO TERMINATE THE OPERATION.

SUBSEQUENT SEARCH OPERATIONS ARE DESCRIBED ON THE FOLLOWING TWO FLOWCHARTS.

FIG 17.14

SEARCH (START)



IF EITHER:

1) OUT OF WORD COUNT, OR

2) THE CHANNEL HAS BECOME IDLE, OR

3) THE CHANNEL OR CPS HAS INSTRUCTED PUP TO BECOME IDLE,
THEN, GO TERMINATE THE OPERATION.

FETCH A DATA WORD W FROM THE CHANNEL. DECREMENT WORD COUNT.

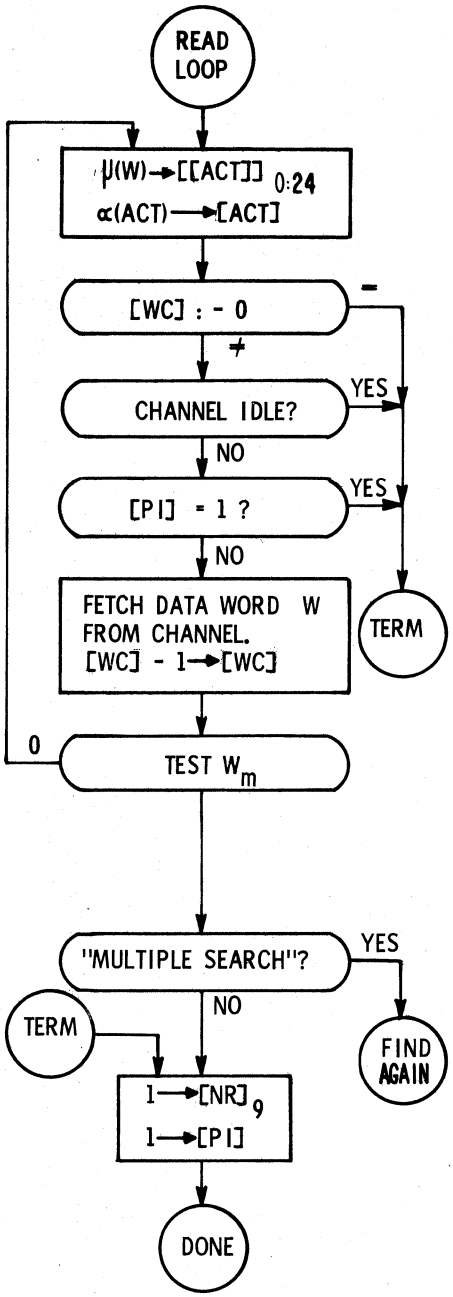
IF THE DATA WORD W HAS A METABIT ONE, THEN W BEGINS A NEW RECORD.

COMPARE W WITH [KEY] THROUGH THE [MASK] TO DETERMINE IF THIS IS THE RECORD BEING SEARCHED FOR.

SUBSEQUENT SEARCH OPERATIONS ARE DESCRIBED ON THE FOLLOWING FLOWCHART.

FIG 17.15

SEARCH (FIND LOOP)



STORE W, WITH METABIT APPROPRIATELY MODIFIED, IN LOCATION SPECIFIED BY ACT. CHANGE ACT.

IF WORD COUNT IS MINUS ZERO, THEN GO TERMINATE THE OPERATION.

IF CHANNEL HAS BECOME IDLE, THEN GO TERMINATE THE OPERATION.

IF CHANNEL HAS INSTRUCTED PUP TO TERMINATE DATA TRANSMISSION, OR CPS HAS INSTRUCTED PUP TO BECOME IDLE, THEN GO TERMINATE THE OPERATION.

FETCH DATA WORD W FROM CHANNEL. DECREMENT WORD COUNT.

IF METABIT OF W IS ZERO, THEN LOOP.

METABIT OF W IS NOT ZERO, INDICATING END OF RECORD.

IF "MULTIPLE SEARCH", GO SEARCH FOR NEXT RECORD.

TERMINATE THE OPERATION: PUP DISCONNECTS, BECOMING IDLE AND SETTING "PUP OPERATION COMPLETE" INTERRUPT.

PUP IS NOW IDLE, AND READY TO PROCESS A NEW SIO. THE CHANNEL AND DEVICE MAY NOT YET BE IDLE.

FIG 17.16

SEARCH (READ LOOP)

CHAPTER III -- THE LOW SPEED BUFFER

18.0 GENERAL DESCRIPTION

The Low Speed Buffer (LSB) has 1024 words of core memory, control elements, and the capacity for a maximum of 32 input or output devices. Presently existing are a paper tape reader, paper tape punch, digital plotter and sixteen typewriters, (each both an input and an output device). Device codes assigned to present devices are shown in Section 17.4.

The LSB core memory may be considered a 1024 word auxiliary memory or an input/output buffer for the various devices, depending upon the operation performed. As an I/O buffer for typewriters and paper tapes, each word is considered to be composed of three 8-bit characters. For the digital plotter each word contains six 4-bit commands.

The first thirty-two LSB memory positions will be used as control words -- one for each device -- specifying the buffer area for that device, and controlling the transmission of data between buffer and device, (Cf. Device Control Words).

19.0 STATUS INDICATORS

Each device has an input and/or output status indicator. This is on only when data is to be transmitted to or from the device. The LSB directs this transmission (to or from LSB memory) with the device control word. Each device's input status (I/S) or output status (O/S) indicator must be turned on in order to initiate an input or output transfer. It may be turned on or off (initiating or halting a transfer) with a special LSB operation, (Cf. I/O Status Command).

The digital plotter and paper tape punch each have an O/S indicator which is on when data or commands are being sent from the LSB to the device. When the tape punch or plotter output is complete, the O/S

indicator is turned off and the "LSB to device transfer complete" interrupt bit $[NR]_{19}$ is set.

The tape reader has an I/S indicator which is on when data are being read into LSB memory. When it goes off, the "device to LSB transfer complete" interrupt bit $[NR]_{17}$ is set.

The paper tape reader has a Buffer Status (B/S) indicator. When the LSB memory area assigned to the unit is filled, this indicator is turned on and the I/S is turned off. No more data can be read from the unit until the B/S is turned off and I/S is turned on. B/S can be turned off by the commands Set Input Buffer or Unload Input Buffer. I/S can be turned on by Unload Input Buffer or the status command (Cf. Operations).

Each typewriter has three indicators, an I/S indicator, an O/S indicator and a buffer status (B/S) indicator.

The I/S indicator is turned on by a status command and is turned off by performing a status command, or a Set Input Buffer command, or by hitting the END MESSAGE key. Only when the I/S indicator is on may information be typed into the buffer. If the I/S indicator is off, the operator may request the computer to turn it on by striking the Begin Message Key.

The B/S (buffer status) indicator is turned on by either of two actions: (1) hitting the typewriter's END MESSAGE key or (2) typing a character into the third character position of the last word in the LSB memory area assigned as that typewriter's buffer. When the character generated by actions (1) or (2) above is stored in the LSB memory, the "device to LSB transfer complete" bit is set. In the case of action (1), the I/S indicator is also cleared at that time. The B/S indicator is turned off when that typewriter's buffer area is unloaded into CPS

memory by an "Unload Input Buffer" command or when the buffer area is re-established by a "Set Input Buffer" command (Cf. Operations).

When the O/S indicator is turned on, it indicates that a message in LSB memory is to be typed out. The LSB transmits output characters to the typewriter from the typewriter buffer area until the last character has been typed out. The typewriter's O/S indicator is then turned off and the "LSB to device transfer complete" bit is set.

The possible combinations of the three status indicators' conditions are described in Table 19.1.

TABLE 19.1: TYPEWRITER STATUS

Condition of:			State	
I/S	B/S	O/S	Number	Explanation
off	off	off	0	Idle
off	off	on	1	Typing out
off	on	off	2	End Message key hit, buffer not unloaded
off	on	on	3	Same as 2, but typing out begun (not normally done)
on	off	off	4	Typing in
on	off	on	5	Error - see note
on	on	off	6	Typing in, buffer filled, not unloaded. (End Message Key may have been hit but End Message character not yet stored in memory. I/S is cleared when End Message character is stored in memory.)
on	on	on	7	Error - see note

NOTE: It is not permissible for the I/S and O/S indicators to be on simultaneously. If this occurs, the device's error flip-flop is set and the device error interrupt bit $[NR]_{15}$ is set.

19.2 Attention Status

Each typewriter has one additional indicator, the Attention Status (A/S) indicator. This is turned on and a program interrupt signal ($[NR]_{16}$) is simultaneously generated if the typist strikes the Begin Message key and the A/S and I/S indicators are off or the A/S indicator is off and the B/S indicator is on. In both cases the typist is unable to type in until appropriate action is taken by the CPS. In case 1, the I/S indicator must be turned on by the CPS. This will require prior termination of an output message if the O/S indicator was on when the begin message key was struck. In case 2, the CPS must execute an "Unload Input Buffer" command to clear the B/S indicator and empty the buffer to make room for additional inputs from the keyboard.

In both of the above cases, once the A/S indicator is set, repeated operation of the begin message key will have no further effect until the A/S indicator is cleared by the CPS through a skip-make instruction.

There is one additional condition for setting the A/S indicator. If the high order bit of a typewriter input control word (Cf. 21) is set, the LSB will examine the A/S indicator each time a character is received from the typewriter. If the A/S indicator is in the zero state, the LSB will set it and simultaneously generate a program interrupt signal (device to LSB transfer complete). If the A/S indicator is already set, it is left unaltered and the program interrupt signal is not generated. In both cases, the incoming character is stored in the LSB memory in the normal manner.

20.0 OUTPUT DEVICE CONTROL WORD

When a device O/S indicator is on, and the device is ready to receive data from the buffer, the LSB looks at the device control word. This is stored in LSB memory (at the location equal to its device number) in the following format:

RC		Character Count	Number of Words	Word Address
0		2 3	4 13	14 23

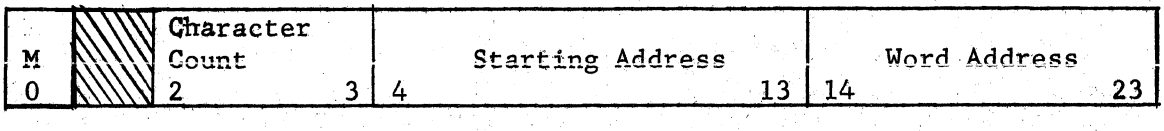
For the tape punch or the typewriters, the word address specifies the LSB memory location of the word from which the next character is to be taken, and the Character Count (a number from Ø to 2) specifies the character within that word. After the character has been sent to the unit, the Character Count is stepped up by one. If it goes from 2 to Ø, the Word Address is stepped up by one and the Number of Words is stepped down by one. Hence, the Number of Words gives the number of words yet to be written including the one from which characters are being written.

When the last character is transmitted, the character-count and number-of-words fields are both stepped to zero. When the device next signals the LSB that it is ready for another character, the LSB upon noting zeros in the character-count and number-of-words fields turns off the devices O/S indicator and generates a program interrupt signal, "LSB to device transfer complete".

The bit RC is used in controlling outputs to the digital plotter. RC specifies which half of the 8 bit character specified by CC is to be sent to the plotter; "0"=upper half, "1"=lower half. RC is complemented after each transmission. CC is incremented when RC goes from one to zero.

21.0 INPUT DEVICE CONTROL WORD

When a device's I/S indicator is on and the device is ready to transmit data to the buffer, the LSB looks at the device's control word, which is in the following format (in LSB memory):



The Starting Address remains unchanged during any LSB input operation. It can be set initially by a load memory on a set Input Buffer operation, and is used to specify the beginning of the LSB memory area assigned as that device's input buffer area.

The Character Count and Word Address specify where the next character from the unit is to be stored. They are stepped the same as in the Output Control Word.

As the characters are being read into an LSB memory position, they are positioned as described below:

X	Y	Z	
CØ	Ø	Ø	Before CØ is read in
CØ	C1	Ø	After CØ is read in
CØ	C1	C2	After C1 is read in
			After C2 is read in

The most significant bit, M of the input control word is a "mode" bit applicable to typewriter inputs. If M is a one, the LSB examines the specified typewriters A/S indicator each time a character is received. If the A/S indicator is a zero, the LSB sets the A/S indicator and bit number 17 of the interrupt register. If either M is a zero or the appropriate A/S indicator is already a one, these actions are not taken.

The input operation is terminated by the presence of a 1 Tag bit in LSB memory. Every LSB memory word has an extra bit position called the Tag. This bit is unaffected by the transmission of data from its buffer memory word or into its buffer memory word from an input device. However, it may be modified when the CPS writes into that LSB memory position. Normally, it is set only by a set Input Buffer operation. (Cf. Operations.) A "1" in the Tag of an LSB memory position indicates that it is the last word of the LSB memory area designated as a device's input buffer. Hence, in an input operation when the third character from the tape reader or typewriter is read into an LSB memory position having a 1 as its Tag bit, the Current Address is stepped up 1 (Character Count set to zero) and the transmission of data into the LSB is terminated in the following manner:

(a) For the tape reader, the B/S indicator is turned on and the I/S indicator is turned off.

(b) For a typewriter, the I/S indicator is left on and the B/S indicator is turned on, locking the keyboard. (However, the Begin Message key is not locked.) The B/S indicator is turned off when the buffer area is unloaded into CPS memory with an Unload Input Buffer operation or upon execution of a Set Input Buffer operation.

A typewriter input operation may also be terminated by striking the End Message key. This enters an end of message character into the buffer (in location indicated by the control word), steps the control word, and turns the I/S indicator off and the B/S indicator on.

22.0 INTERRUPTS

The following CPS interrupt bits may be set by an LSB condition in the manner indicated:

(1) Device Inoperative $[NR]_6$: This is set upon receipt of certain input-output commands specifying a device which is not in operating condition (e.g., power off, tape broken, unit disconnected). The Device Inoperative interrupt precedes the PUP Operation Complete interrupt in all cases.

The following LSB operations are capable of initiating a device inoperative interrupt:

a. Unload Input Buffer

b. Status Command

c. Load Output Buffer

(2) Device Busy $[NR]_8$: This is set upon receipt of certain input or output commands specifying a device that is already busy transmitting or receiving data. The operations and conditions which will cause a device busy interrupt are as follows:

a. Unload Input Buffer

1. Typewriters:

(a) The O/S indicator is set or

(b) The I/S indicator is set, the control word mode bit, M is a zero, and the B/S indicator is a zero.

2. Other Devices:

(a) The I/S or O/S indicator is set.

b. Load output buffer

b. Load Output Buffer

The devices I/S or O/S indicator is set. In all cases, the device busy interrupt precedes the PUP Operation complete interrupt.

(3) Device Error $[NR]_{15}$:

Various parity and other checks are made on data being transmitted between the LSB and the low speed IO devices. If any of these checks are not satisfied, the devices error flip-flop is set and the device error interrupt bit, $[NR]_{15}$ is set. In the case of the typewriters, these bits are also set if the devices I/S and O/S indicators are ever in the one state simultaneously.

(4) Typewriter Attention $[NR]_{16}$:

If a typewriter's Begin Message key is struck and the A/S indicator is off and either the B/S indicator is on or the I/S indicator is off, the typewriter's Attention Status indicator is set and this interrupt bit is set.

(5) Device to LSB Transfer Complete $[NR]_{17}$:

This interrupt bit is set when the paper tape reader's I/S indicator goes off or when the LSB receives a character from a typewriter which either fills the buffer or is an End Message character.

The device to LSB transfer complete bit is set also upon receipt of any input character from a typewriter if the mode bit "M" of the input control word is a one and the A/S indicator is a zero. Under these conditions, the A/S indicator is also set.

(6) LSB to Device Transfer Complete $[NR]_{19}$:

When an output operation is complete, i.e., when an O/s indicator goes off, this bit is set.

23.0 LSB OPERATIONS (Cf. Table 23.8)

23.1 Read Memory (Op Code 1)

This operation reads from LSB memory, beginning at the Starting Address in $IOCW2_{14:23}$ into CPS memory. The number of words to be ready is specified by $IOCW1_{8:23}$. Tag bits or device buffer areas do not affect the operation.

23.2 Read Input Buffer (Op Code 2)

This operation is used to read the portion of the device's buffer area that has been already filled by the input device. It does not disturb the device to LSB transfer. The specified device's control word is first retrieved from the LSB memory and sent to the CPS. Following this, the remainder of the operation is performed by looking at the device's input control word and transmitting words from LSB memory to the CPS beginning at the LSB memory location specified by the Starting Address up to the location given by the Word Address. If the character count is zero, the word at the Word Address location is not transmitted. If the character count is not zero, the word at the Word Address location is sent to the CPS. The number of words transmitted to CPS memory is limited by the Word Count ($IOCW1_{8:23}$) if this is smaller than the number of words read into the device's buffer area. If the Word Count is greater than the number of words read in so far, it does not affect the operation.

23.3 Unload Input Buffer (Op Code 3)

This operation causes the readout of the specified input device's LSB memory area into CPS memory. Initially, the device's control word is retrieved from LSB memory and sent to the CPS. If the device is the paper tape reader, the sign bit of control word 2 is a zero, and the device is operative, the I/S indicator is turned on to reinitiate inputs from the reader. Following this, the device status is examined. If the device is already in operation or is inoperative, a device busy or device inoperative interrupt will occur (cf. Interrupts).

If neither interrupt occurs, the device's B/S indicator and A/S indicator (in the case of typewriters) is turned off, and the buffer area will be read into CPS memory in the same way as in the Read Input Buffer operation. After the data transmission to the CPS is complete, the Word Address (in the device's input control word) is set equal to the Starting Address and the Character Count is set to $IOCW2_{4:5}$ modulo 3.

23.4 Set Input Buffer (Op Code 4):

This operation effectively sets up an input buffer in LSB memory. It first clears the I/S and B/S indicators for that device (generating no interrupts) and loads its input control word, setting the Starting Address as indicated by $IOCW1_{8:23}$, the Word Address equal to the Starting Address, the Character Count equal to $IOCW2_{4:5}$ modulo 3, and the mode bit to $IOCW2_{0}$.

It next clears the assigned LSB memory area, i.e., it clears LSB memory to all zeros, including the Tag bit, beginning at the Starting Address location and continuing until location equal to the Starting Address plus n-2 where $n = [IOCW2_{14:23}]$. It clears the location given

by Starting Address plus n-1 to zeros and sets the Tag bit to 1. Hence, all this sets up a cleared area in LSB memory (whose number of words is indicated by I/O Control Word 2) with all Tag bits cleared except for the last word's which is a 1.

Note that a Set Input Buffer operation with n = 0 just affects the input control word and the I/S and B/S indicators. The remaining memory area is unaffected.

23.5 I/O Status Command (Op Code 5)

This operation may be used to do any of the following:

(1) Determine if a device is operative

(2) Set or clear the I/S indicator

(3) Set or clear the O/S indicator

(4) Set the ILZ (Inhibit Lead Zeros) indicator of the Paper Tape Reader.

The device code, $IOCW2_{6-10}$, specifies the device to be selected. The specified device is examined and if found to be inoperative, a device inoperative program interrupt signal is generated.

If $OICW2_{22}$ is a one and the device code specifies device 16 (paper tape reader), the "Inhibited Lead Zeros" indicator is set. If this indicator is set, characters received from the paper tape reader are ignored until the first non-zero character is detected. Upon receipt of a non-zero character, the ILZ indicator is reset and all characters thereafter, including the initial non-zero character, are accepted and stored in the LSB memory.

If $IOCW2_{23}$ is a one or if $IOCW2_{0}$ is a one and the device is inoperative the Status Command is terminated at this point. If not, the operation continues as follows.

$IOCW2_{21}$ specifies whether the I/S or O/S indicator is to be affected: $\emptyset$ = I/S indicator, 1 = O/S indicator; $IOCW2_{\emptyset}$ tells whether the specified status indicator is to be turned on or off: $\emptyset$ = off, 1 = on. The specified bit is modified as indicated and the operation ends.

Turning an O/S indicator off will generate an LSB to device transfer complete interrupt. A status command to turn the paper tape reader's I/S indicator off initially sets the B/S indicator. The next character received from the paper tape reader will then cause the I/S indicator to be cleared and a device to LSB transfer complete interrupt to be generated. Turning a typewriter's I/S indicator off does not generate an interrupt signal. Note that a command to turn an indicator on or off does nothing if the indicator is already in the specified state.

The status command should be used with caution in modifying typewriter status bits. Due to timing peculiarities of the typewriter, the following dangers exist:

(1) Turning off the I/S indicator can cause undetected loss of the last character typed in by the operator.

(2) Turning on the O/S indicator soon after turning off the I/S indicator can cause the first few characters going out to be garbled. It is conceivable for example that a carriage return may have been in progress when I/S was turned off. This operation may last up to a second, thus, causing the first several outgoing characters to be printed during fly-back of the carriage.

(3) Turning off the O/S indicator immediately after a character printing cycle has been initiated may cause the character to be lost or to be printed incorrectly.

23.6 Load Memory (Op Code 6)

This operation writes from CPS memory into LSB memory. It is the exact converse of the Read Memory operation. Note that tag bits will be recorded as received from PUP.

23.7 Load O/P Buffer (Op Code 7)

This operation loads the indicated number of words into LSB memory beginning at the starting address. The number of words and the starting address are loaded into the output device's control words and the character count is set to $IOCW_{4:5}$ modulo 3. Note that Tag bits will be recorded as received from PUP.

If bit S of IOCW2 is a zero, the appropriate O/S indicator will be turned on, causing the data to be written out. A one in bit S suppresses the actual output to the device.

If the device's I/S or O/S indicator was on initially or if the device is inoperative, the operation will not be performed (no data will be sent to the LSB memory) and an LSB device busy or device inoperative will occur.

Roland Silver

Roland Silver
For D-13 Staff

/nb

Attachments:
Appendix A
Appendix B (Distribution List)

LSB Operation	$IOCW1_{8:23}$	$IOCW2_{0}$	$IOCW2_{1:3}$	$IOCW2_{4:8}$	$IOCW2_{9:10}$	$IOCW2_{14:23}$	$IOCW3_{8:23}$
Read Memory	Number of words read into CPS memory		001			Address in LSB memory of first word sent to CPS	CPS memory location into which first word from LSB is stored
Read Input Buffer	Maximum number of words sent to CPS memory		010	Device Address			CPS memory location into which first word from LSB is stored
Unload Input Buffer	Maximum number of words sent to CPS memory	For paper tape reader, 0 = Set I/S to turn reader on. 1 = Leave reader off.	011	Device Address	New character count for device control word		CPS memory location into which first word from LSB is stored
Set Input Buffer	Address in LSB memory of first word of input buffer	Copy into "mode" bit position of device control word	100	Device Address	Character count for device control word	Length of input Buffer	
Status Command		0 = turn indicator off 1 = turn indicator on	101	Device Address		$IOCW2_{21}$: 0 = IS; 1 = OS. $IOCW2_{22}$: 1 = Set ILZ if tape reader specified, otherwise unused $IOCW2_{23}$: 1 = modify status bit; 0 = do not modify status bit	
Load Memory	Number of words loaded into LSB memory		110			LSB memory location into which first word is loaded	CPS memory location of first word sent to LSB memory
Load Output Buffer	Number of words loaded into LSB memory and number of words sent to output device by LSB	0 = Set O/S 1 = Do not set OS (suppress output)	111	Device Address	Character count for device control word	LSB memory location into which first word is loaded	CPS memory location of first word sent to LSB memory

TABLE 23.8 LSB OPERATIONS

APPENDIX A -- INSTRUCTION SUMMARY

The following tables present, in a compact form, a summary of the PHOENIX instructions.

Table A-1

Table A-1 presents the instructions in groups according to certain of their characteristics. A given instruction may be a member of several groups.

Table A-2

In Table A-2, a miscellaneous assortment of information regarding the PHOENIX instructions is given. The nomenclature and notation is that developed in Chapter I.

The following references may prove useful in the use of Table A-2:

ITEM	SECTION
1) Basic notation	1
2) Live registers	2
a) AC	2.1
b) BR	2.4
c) CA	2.5
d) XR	2.32
e) VF, VFP	2,29, 2.30, 9.46
3) Notation for arguments	7.1
4) Instruction formats	7.2
5) Metabit trap	6.2
6) Interrupts	6.1
7) Protected status ([PR] =1)	5.0
8) Timing	3.0
(Note particularly the assumptions made for "average" normalize instructions, 3.3.8)	

Octal Listing of Instructions

Octal	Mnemonic	Type	Octal	Mnemonic	Type
00000000	*		00200000	LXI	1
00004000	RIS	4	00400000	LXR	1
00010000	*		00600000	DPX	1
00014000	*		01000000	*	
00020000	NAL	3	01200000	TRP	6
00024000	NCL	3	01400000	BXS	1
00030000	SMZ	3	01600000	BXA	1
00034000	SMN	3	02000000	BCH	1
00040000	CYA	2	02200000	CMX	1
00044000	CYB	2	02400000	SIO	1
00050000	CYC	2	02600000	DFR	1
00054000	*		03000000	ADE	1
00060000	SHA	2	03200000	LDE	1
00064000	SHB	2	03400000	DBR	1
00070000	SHC	2	03600000	DAC	1
00074000	*		04000000	EOR	1
00100000	LRB	5	04200000	VOR	1
00104000	NVX	3	04400000	AND	1
00110000	PAX	3	04600000	LAC	1
00114000	PXA	3	05000000	LBR	1
00120000	CBX	2	05200000	ADX	1
00124000	*		05400000	SUB	1
00130000	MMZ	3	05600000	ADD	1
00134000	MMN	3	06000000	MPY	1
00140000	NAR	3	06200000	DVD	1
00144000	NCR	3	06400000	CMA	1
00150000	BAS	4	06600000	SAS	1
00154000	BSA	4	07000000	SKM	1
00160000	ASB	4	07200000	DAM	1
00164000	*		07400000	DSU	1
00170000	*		07600000	DAD	1
00174000	PER	5			

* Unused at present;
functionally a "no-operation".

Instructions By Type

Type 1			Type 2	Type 3	Type 4
ADD	DAD	LDE	CBX	MMN	ASB
ADE	DAM	LXI	CYA	MMZ	BAS
ADX	DBR	LXR	CYB	NAL	BSA
AND	DFR	MPY	CYC	NAR	RIS
BCH	DPX	SAS	SHA	NCL	**Type 5**
BXA	DSU	SIO	SHB	NCR	
BXS	DVD	SKM	SHC	NVX	LRB
CMA	EOR	SUB		PAX	PER
DAC	LAC	VOR		PXA	**Type 6**
	LBR			SMN	
				SMZ	TRP

TABLE A-1 (1st page)

Arithmetic Instructions		Instructions Which May Affect [AC] and/or [BR] But Which Do Not Reference Central Memory		Instructions Which May Affect [AC] and/or [BR] and Which Reference Central Memory	
ADD	NCL	ASB	NAR	ADD	LAC
DAD	NCR	BAS	NCL	AND	LBR
DSU	SHA	BSA	NCR	DAD	MPY
DVD	SHB	CBX	PXA	DSU	SUB
MPY	SHC	CYA	SHA	DVD	VOR
NAL	SUB	CYB	SHB	EOR	
NAR		CYC	SHC		
		NAL	SKM		

Instructions Which May Change Central Memory Locations		Instructions Which May Affect the Program Control ([PC])		Bit Manipulation Instructions	
DAC		ADX	LXI	Boolean:	
DAM		BCH	LXR	AND	
DBR		BXA	PAX	EOR	
DPX		BXS	PER	VOR	
MMN		CMA	SAS	Others:	
MMZ		CMX	SKM	ASB	NCL
TRP		DFR	SMN	BAS	NCR
		DVD	SMZ	BSA	SAS
		LRB	TRP	CBX	SHA
				DAM	SHB
Also:		Also:		MMN	SHC
Metabit Trap		Metabit Trap		MMZ	SKM
Interrupt		Interrupt		NAL	SMN
I/O Operation				NAR	SMZ

Instructions Which May Change [XRs]	Instructions Which "Save" [XRs]	Program Protection and Relocation
ADX	DPX	LRB
BCH	PXA	PER
BXA		RIS
BXS	Instructions Which May Modify Effective Addresses	Also: Interrupt
CBX		
LXI	ADE	Metabit Instructions
LXR	LDE	
NAL		MMN
NAR	I/O Instruction	MMZ
NCL		SMN
NCR	SIO	SMZ
NVX		
PAX		

TABLE A-1 (cont'd.)

Instruction Name	Mnemonic OPC	Mnemonic Arg	Type	May Change [PC]	May Affect VF and VFP	mb TRAP	NOP if [PR] = 1	Timing Min	Timing Avg	Timing Max	Octal	Ref. Sect.	Comments
Add	ADD	R,T	1		√	√		1	1.25	2	05600000	9.1	
Add to Effective Address	ADE	I,T	1					1	1	1	03000000	9.2	1 → [CCI]
Add to XR	ADX	I,X	1	√				1	1	1	05200000	9.3	
And	AND	R,T	1			√		1	1.25	2	04400000	9.4	
AC Sign to BR	ASB		4					1	1	1	00160000	9.5	Sign to all of BR
BR Sign to AC Sign	BAS		4					1	1	1	00150000	9.6	
Branch	BCH	I,T	1	√				1	1	1	02000000	9.7	[PC] → XR5
BR Sign to AC	BSA		4					1	1	1	00154000	9.8	Sign to all of AC
Branch on XR and Add	BXA	I,X	1					1	1	1	01600000	9.9	Test on 177777_8; NOP if XR6 or XR7
Branch on XR and Subtract	BXS	I,X	1					1	1	1	01400000	9.10	Test on 177777_8; NOP if XR6 or XR7
Cycle BR and Shift into XR5	CBX	N,T	2					1	2	6	00074000	9.11	Inverts bit order (into XR5) if n<0
Compare AC	CMA	R,T	1	+0,1,2				2	2	2	06400000	9.12	[AC]<,=,>[E]; [AC]→[BR]
Compare XR with I	CMX	I,X	1	+0,1,2				1	1	1	02200000	9.13	[XRX]<,=,>E
Cycle the AC	CYA	N,T	2					1	1	2	00040000	9.14.1	
Cycle the BR	CYB	N,T	2					1	1	1	00044000	9.14.2	
Cycle Combined AC and BR	CYC	N,T	2					1	1	2	00050000	9.14.3	
Deposit AC	DAC	R,T	1			√		1	1.25	2	03600000	9.15	
Double Precision Add	DAD	R,T	1		√	√		2	2.25	3	07600000	9.16	
Deposit AC, Masked	DAM	R,T	1			√		1	1.25	2	07200000	9.17	
Deposit BR	DBR	R,T	1			√		1	1.25	2	03400000	9.18	
Defer Execution	DFR	I,T	1					1	1	1	02600000	9.19	
Deposit XR	DPX	R,X	1			√	*	1	1.25	2	00600000	9.20	
Double Precision Subtract	DSU	R,T	1		√	√		2	2.25	3	07400000	9.21	
Divide and Skip	DVD	R,T	1	+0,1		√		14	17	19	06200000	9.22	Normally Skip. Quotient→[AC], Remainder→[BR] Signs at end: AC: Algebraic BR: As Dividend
Exclusive "Or"	EOR	R,T	1			√		1	1.25	2	04000000	9.23	
Load AC	LAC	R,T	1			√		1	1.25	2	04600000	9.24	
Load BR	LBR	R,T	1			√		1	1.25	2	05000000	9.25	
Load Effective Address	LDE	R,T	1			√		2	2	2	03200000	9.26	1 → [CCI]
Load RL and BD	LRB	J,T	5			√	√	1	1	1	00100000	9.27	
Load XR Immediate	LXI	I,X	1	√				1	1	1	00200000	9.28	
Load XR Remote	LXR	R,X	1	√		√		2	2	2	00400000	9.29	

TABLE A-2 (1st page)

Instruction Name	Mnemonic OPC	Arg	Type	May Change [PC]	May Affect VF and VFP	mb TRAP	NOP if [PR] = 1	Timing Min	Avg	Max	Octal	Ref. Sect.	Comments
Make Metabit One	MMN	X	3					1	1.25	2	00134000	9.30	
Make Metabit Zero	MMZ	X	3					1	1.25	2	00130000	9.31	
Multiply	MPY	R,T	1			√		5	7	10	06000000	9.32	
Normalize AC Left	NAL	X	3					1	4	5	00020000	9.33	[XRX] ⊖ k → [XRX]; NOP if XR6 or XR7
Normalize AC Right	NAR	X	3					1	1	1	00120000	9.34	[XRX] ⊕ 1 → [XRX]; NOP if [VF] = 0 or XR6 or XR7
Normalize CA Left	NCL	X	3					1	5	8	00024000	9.35	[XRX] ⊖ k → [XRX]; NOP if XR6 or XR7
Normalize CA Right	NCR	X	3					1	1	1	00124000	9.36	[XRX] ⊕ 1 → [XRX]; NOP if [VF] = 0 or X = 6 or X = 7
Invert XR	NVX	X	3					1	1	1	00010000	9.38	NOP if XR7
Put AC Into XR	PAX	X	3	√				1	1	1	00110000	9.39	
Operate	PER	J,T	5	+0,1			√	2	2	2	00174000	9.40	
Put XR Into AC	PXA	X	3					1	1	1	00114000	9.41	
Restore Interrupt Status	RIS		4	√	*		√	3	3	3	00004000	9.42	* See 9.42
Skip if AC Same, Masked	SAS	R,T	1	+0,1		√		2	2	2	06600000	9.43	Skip if same, masked.
Shift AC	SHA	N,T	2		√			1	2	6	00060000	9.44.1	
Shift BR	SHB	N,T	2		√			1	2	5	00064000	9.44.2	
Shift CA	SHC	N,T	2		√			1	3	6	00070000	9.44.3	
Start I/O Operation	SIO	I,T	1				√	1	1	1	02400000	9.45	
Skip-Make	SKM	I,T	1	+0,1	*		*	2	2	2	07000000	9.46	K,M,S,D,R,C; * see 9.46
Skip if Metabit One	SMN	X	3	+0,1				2	2	2	00034000	9.47	
Skip if Metabit Zero	SMZ	X	3	+0,1				2	2	2	00030000	9.48	
Subtract	SUB	R,T	1		√	√		1	1.25	2	05400000	9.49	
Trap	TRP		6	3				2	2	2	01200000	9.50	[PC] → [2]; 3 → [PC]
Inclusive "Or"	VOR	R,T	1			√		1	1.25	2	04200000	9.51	
Unused Codes	-	-	-	-	-	-	*	1	1	1	-	-	* Always NOP, see 9.37
Interrupt				6								6.1	Store into 4 and 5; branch to 6.
Metabit Trap				1								6.2	[PC] → [0]; 1 → [PC]

TABLE A-2 (cont'd.)

APPENDIX B

DISTRIBUTION LIST

MITRE

Codes: E, H, I

Department D-13

All Staff
D-13 File
R. J. Beller
E. B. Griswold
P. Heckel
C. S. Wells (100 copies)

Lawrence Radiation Laboratory
Livermore, California

Mr. Norman Hardy
Mr. George Michael

Massachusetts Institute of Technology, Technology Square
Cambridge, Mass. - Project MAC

Prof. Fernando J. Corbato
Prof. Robert M. Fano
Mr. Richard Mills
Prof. Marvin L. Minsky

Stanford University, Computation Center, Palo Alto, California

Prof. John McCarthy

ESD

ESAT
ESR
ESRC
ESRD
ESRH
ESSB
ESSL
ESSQ
ESSTT
Det. 8
Major S. Jeffrey

DCA

Dr. Frank L. Hassler

SACSO

Colonel Lewis E. Sutton

Advanced Research Project Agency
Pentagon Building, Washington, D.C.

Dr. J. C. R. Licklider
Ass't. Dir. for Behavioral Sciences