



## 1401 DATA FLOW

New, more efficient programming techniques for the IBM 1401 Data Processing System are being developed as the programming knowledge and the experience level on the system increases. This manual presents a semi-detailed data flow explanation of every 1401 operation. This approach should assist both the new, and the more experienced, programmer with his 1401 programming effort.

Each operation is presented in the form of a data-flow diagram. The path that the data takes during an operation is graphically presented along with a written explanation of the steps involved. The internal-parity and validity-checking operations are also presented, along with a list of the console lights that will be ON in the event of a parity or validity check condition.

**Data Flow**

The data flow of an IBM 1401 Data Processing System is schematically shown in Figure 1. The flow paths used are initially specified by the stored program instructions. These instructions tell the system what areas to read out of, and what areas to read into. The internal circuitry of the system then carries out the specified data movement. The various component areas of the system are:

**CORE STORAGE**

The IBM 1401 Data Processing System uses magnetic-core storage for storing instructions and data. All the data in core storage is readily available, and the design of the core-storage area makes each position individually *addressable*. All data received from input units is stored in the core storage, and all data sent to the output units is sent from core storage.

Information is always read out of core storage during the early half of a cycle. The read-out is actually accomplished by setting all the cores, at the specified location, to zero. A core originally set at *one* will, when flipped from one to zero during read-out, induce a volt-

age on one of the wires running through the center of the core. This voltage is recognized as a bit.

Information is always read into core storage during the late half of a cycle. If the information that was read out of a storage location is to be retained in the same location, it is read back into that location from one of the registers during the late half of the read-out cycle. The 1401 system also makes use of this core-storage operation to perform the system's arithmetic operations. Two areas are alternately read out a position at a time, added together, and the sum stored in the last position read out. This is called *add-to-storage logic*, and it eliminates the need for special-purpose accumulators or counters. Because any group of storage positions can be used as an accumulating field, the capacity for arithmetic functions is not limited by a predetermined number of counter positions.

**DATA LINES AND INHIBIT DRIVE**

The data-flow paths shown in Figure 1 as single lines are actually eight lines (4 digit, 2 zone, 1 word mark, and 1 check). The lines leading to the inhibit drive are called inhibit lines because they inhibit, or prevent, the setting of cores unless *activated* by the presence of a bit of information. Information being sent into core storage goes through the inhibit-drive area, while information being sent from core storage goes through the B-register.

**B-REGISTER**

Each character leaving 1401 core storage enters the B-register and is stored in an 8-bit code (BCD code, word mark, check bit). The register is reset and filled with a character from core storage during the read-out portion of every storage cycle. The character can be entered back into core storage from the output of the B-register during the storage read-in portion of a storage cycle. This is necessary when an instruction is being read and will be needed another time, because the cores of a position are all set to zero when that position is read out.

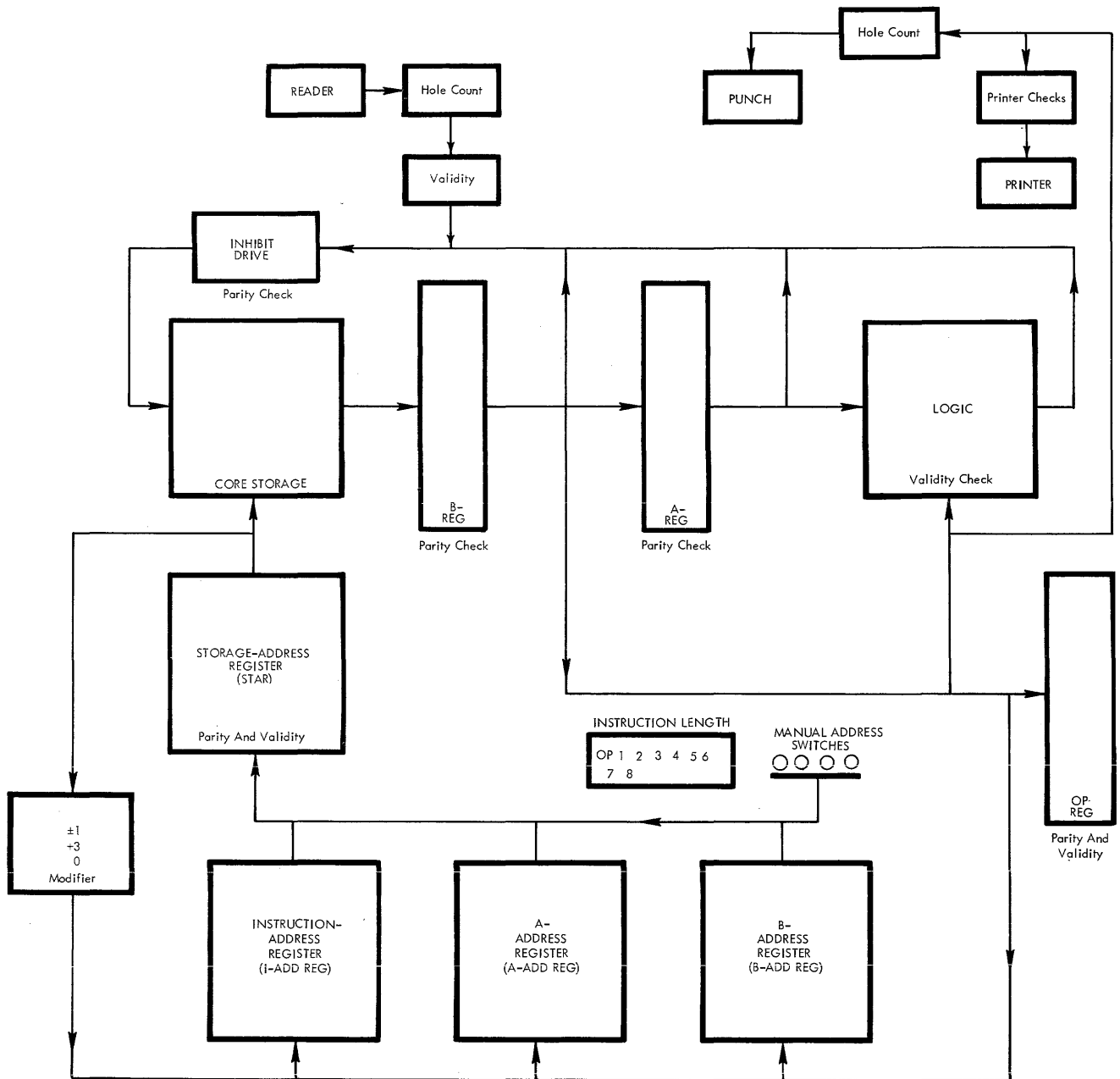


Figure 1. Data Flow and Checking Features

#### A-REGISTER

The A-register is reset and filled with the character from the B-register during each cycle that involves an A-address, and during all instruction cycles, except the first and last cycle of each instruction. The character is stored in an 8-bit code. The character can be entered back into core storage from the output of the A-register during the storage read-in portion of a storage cycle.

#### OP-REGISTER

The Op- (operation) register is reset and filled with a 7-bit character output from the B-register (word mark is dropped), whenever the character is an operation-code character. The Op-register stores the operation code of the instruction in process for the duration of the operation.

#### LOGIC AREA

The logic area is made up of the circuitry that executes the adding, subtracting, and comparing of the A- and B-register outputs. Depending on the operation, the resultant logic-area output may be entered back into storage and/or may indicate the next step to be taken.

#### I-ADDRESS REGISTER

The I-(Instruction) address register (I-Add. Reg.) is a 3-position register, and always contains the core storage location of the next instruction character to be read out. (This 3-digit core-storage location is converted to its 4- or 5-digit number when it is displayed in the storage-address register.) The location number is increased by one as the instruction is read out of core storage, lower-order core-storage position to higher-order core-storage position.

#### A-ADDRESS REGISTER

The A-address register (A-Add. Reg.) is a 3-position register, and normally contains the core-storage location specified in the A-address portion of an instruction. (This 3-digit core-storage location is converted to its 4- or 5-digit number when it is displayed in the storage-address register.) Normally, this core-storage location is the units position of the A-field. As the instruction is executed, the number in this register is decreased by one during each storage cycle that involves the A-address. During several operations, the A-Add. Reg. operation differs. These differences are discussed as they are encountered. *Note:* If the A-address portion of the instruction does not contain a 1401 storage address (%Ux, for example), the numeric portion of the A-Add. Reg. contents is not disturbed as the instruction is executed.

#### B-ADDRESS REGISTER

The B-address register (B-Add. Reg.) is a 3-position

register, and normally contains the core-storage location specified in the B-address portion of an instruction. (This 3-digit core-storage location is converted to its 4- or 5-digit number when it is displayed in the storage-address register.) Normally, this core-storage location is the units position of the B-field. As the instruction is executed, the number in this register is decreased by one during each storage cycle that involves the B-address. During several operations, the B-Add. Reg. operation differs. These differences are discussed as they are encountered.

#### STORAGE-ADDRESS REGISTER

The storage-address register (STAR) is a 3-position register, and contains the address of the core-storage location that is being read out and/or read into on any particular storage cycle. (This 3-digit core-storage location is converted to its 4- or 5-digit number when it is displayed.) This address is received from one of the address registers. As the STAR addresses core storage, the address is also modified and read back into the appropriate address register.

#### ADDRESS-MODIFICATION AREA

Because each character in core storage has a different address, the circuitry that specifies the address must be constantly changing. Instructions are placed in storage with the Op code occupying the lowest-numbered location and the rest of the instruction occupying the adjacent higher-numbered locations. To read out the Op code and then the rest of the instruction, in sequence, the lowest-numbered location must be addressed first, and then each succeeding location must be addressed. Therefore, during the read out of instructions, the address must be modified +1 each time so that the adjacent higher-numbered location is read out.

The data fields, however, are placed in core storage the opposite way. The units position of the field occupies the highest-numbered core-storage location, and the rest of the field occupies the adjacent lower-numbered core-storage locations. To perform the arithmetic functions correctly, the units digit of a field must be worked on first, followed by the tens digit, etc. Therefore, the address must now be modified by -1 each time so that the adjacent lower-numbered location is read out.

Operations involving the printer require that the address must be increased by three each storage cycle, and still other operations require no address modification.

#### INSTRUCTION-LENGTH LIGHTS

These lights indicate which position of an instruction is being read out of core storage.

MANUAL-ADDRESS SWITCHES

The four manual-address switches select the address entered in the storage-address register. These switches are effective only with these selected positions of the mode switch:

1. Character display
2. Alter
3. Address stop
4. Storage print-out
5. Storage scan.

INPUT-OUTPUT UNITS

The various input units send data into the 1401 system where each character is converted to its BCD form and then stored in a specified core-storage location. Core

storage supplies each character for the various output units that can be attached to a 1401 system. The character is converted from its BCD form to a form that is acceptable by the output unit.

PARITY AND VALIDITY CHECKING

The internal self-checking features within the process unit consist of parity and validity checking. Each character is checked, at various locations in the process unit, to be sure it has an odd number of bits, and that it is a valid 1401 character. An even number of bits initiates a parity-check condition, and an incorrect bit configuration initiates a validity-check condition. Refer to Figure 2 for a list of the process-unit check conditions.

Additional checking of the input-output units is done. Refer to the appropriate sections for more detail.

UNIT	TYPE OF CHECK	MACHINE STOPS CHECK STOP SWITCH ON CE PANEL ON	STORAGE-ADDRESS REGISTER (STAR) CONTAINS	LIGHTS ON WHEN STOPPED	RESET BY	REMARKS
A-REG	PARITY	END OF NEXT CYCLE		PROCESS A-REGISTER CHECK RESET	CHECK RESET KEY	A-REGISTER CONTENTS IN ERROR STILL ON DISPLAY ORDINARILY
B-REG	PARITY	END OF CYCLE IN WHICH CHECK IS DETECTED	ADDRESS LOCATION THAT WAS READ INTO THE B-REGISTER	PROCESS B-REGISTER CHECK RESET	CHECK RESET KEY	B-REGISTER CONTENTS IN ERROR STILL ON DISPLAY IN B-REGISTER
LOGIC	VALIDITY	END OF FOLLOWING CYCLE	NEXT A-ADDRESS	PROCESS LOGIC CHECK RESET	CHECK RESET KEY	B-ADDRESS REGISTER INDICATES ONE LESS THAN THE LOCATION THAT THE RESULT READ INTO EXCEPT: 1. WHEN THE CHECK IS DETECTED IN THE LAST CYCLE OF THE FIRST FORWARD SCAN DURING A RECOMPLEMENT OPERATION. INDICATED LOCATION IS THE ERROR LOCATION 2. DURING A REVERSE SCAN OPERATION INDICATED LOCATION IS ONE MORE THAN THE LOCATION THAT THE RESULT READ INTO  THE BIT COMBINATION THAT CAUSES THE CHECK IS IN THE STORAGE UNIT AND IS NOT ON DISPLAY IN THE LOGIC AREA.
INHIBIT DRIVE	PARITY	END OF FOLLOWING CYCLE	DEPENDS UPON THE OPERATION BEING PERFORMED AND THE PHASE (I OR E) THE SYSTEM IS IN	PROCESS STORAGE CHECK RESET	CHECK RESET KEY	A CHECK INDICATES THAT AN EVEN-BIT COMBINATION WAS READ INTO STORAGE
OP-REGISTER	PARITY AND VALIDITY	END OF CYCLE IN WHICH CHECK IS DETECTED	DEPENDS UPON THE OPERATION BEING PERFORMED AND THE PHASE (I OR E) THE SYSTEM IS IN	PROCESS OP REG CHECK RESET	CHECK RESET KEY	CHECK WILL NOT BE DETECTED DURING AN I-OP CYCLE
STORAGE-ADDRESS REGISTER	PARITY AND VALIDITY	END OF CYCLE IN WHICH CHECK IS DETECTED	BIT COMBINATION THAT CAUSED THE ERROR	PROCESS STORAGE-ADDRESS CHECK RESET	CHECK RESET KEY	THE CHECK IS MADE AFTER THE FULL ADDRESS HAS BEEN ENTERED
	WRAP-AROUND		DEPENDS UPON THE OPERATION BEING PERFORMED AND THE MODIFICATION	PROCESS STORAGE-ADDRESS CHECK RESET	CHECK RESET KEY	CAN BE MODIFIED BY +1 OR BY -1
						NOTE: IF ANY OF THE ABOVE CHECKS OCCUR DURING AN INPUT-OUTPUT OPERATION, THAT OPERATION IS COMPLETED BEFORE THE SYSTEM STOPS.

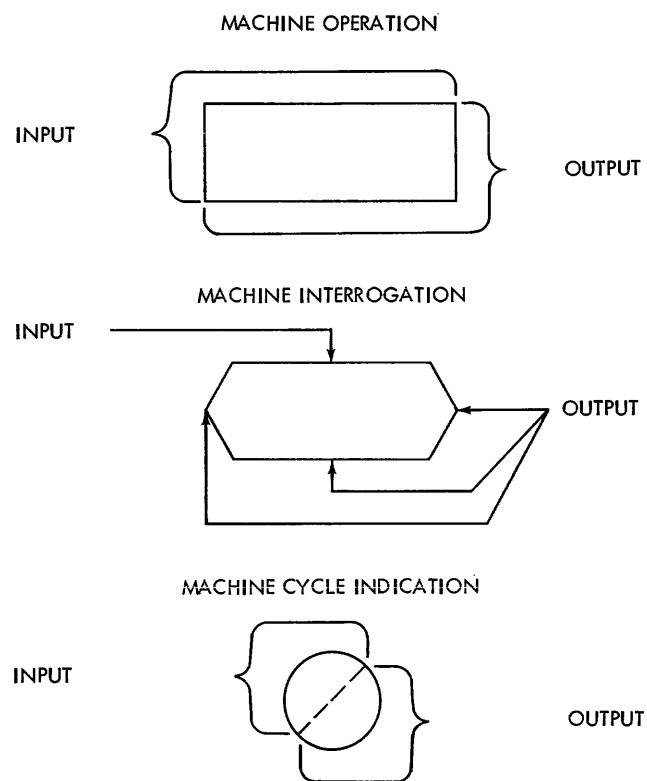
Figure 2. Process Unit Check Conditions

## Data Flow Diagrams

The positive logic approach is used in the data-flow diagrams. For example, all register modification is shown. If register modification is bypassed for any reason, it will not be shown (rather than show it as not happening). If a latch or trigger is shown turned on, it remains on until shown turned off or until it is automatically reset on during the next I-Op cycle. For example, if all A-cycles are eliminated, it will remain that way until the next I-Op cycle, or until the system is instructed to start an A-cycle.

## Abbreviations and Symbols

The abbreviations and symbols used in the data-flow diagrams are shown in Figure 3.



A-REG	A-register
A-ADD REG	A-address register
B-REG	B-register
B-ADD REG	B-address register
C B	Circuit breaker(s)
CHAR	Character; all bits, including zone bits, digit bits, word mark bit, and check bit.
CHK	Check

C L	Console Light(s)
DIGIT	The 1, 2, 4, and 8 bits and C-bit as required.
E-PHASE	The machine cycles required to execute an instruction.
HUND	Hundred
I-ADD REG	Instruction-address register
I-OP	The portion of the I-phase when the operation code is handled by the system.
I-PHASE	The machine cycles required to read out an instruction from core storage.
NSI	Next Sequential Instruction
PAR	Parity
POS	Position
R B	Read back into core storage from the B-register. The entire character is placed in the core storage location specified by the Storage Address Register. This is the location it was originally read out of.
R S	Reverse Scan. Data is read out of core storage starting at the high-order position (lower address) and ending at the low-order position (higher address). This is the reverse of a forward scan operation where data is read out of core storage starting at the low-order position and ending at the high-order position. When a reverse scan is initiated after a forward scan, the high-order position is re-addressed by keeping the previous core storage location in the Storage Address Register.
STAR	Storage Address Register
S F	Standard Form -- applies to sign indication. Any field is considered plus if it has any zone bit combination other than a B-bit alone. Standard form for a plus sign is an A- and a B-bit combination.
THOU	Thousand
TRIG	Trigger
VAL	Validity
WM	Word Mark
WO	Without
Z S	Zero Suppress

Figure 3. Abbreviations and Symbols

## Instruction Reading (I-Phase)

All operations executed by an IBM 1401 Data Processing System are initiated by a stored-program instruction.

The instruction is read first, and then the operation specified by the instruction is executed. The system operating time used to read one complete instruction from core storage is called the instruction (I) phase of the instruction. The time used to execute the specified operation is called the execute (E) phase.

### I-Phase

The I-phase is divided into 11.5  $\mu$ s storage cycles, called I-cycles. A total of nine I-cycles (I-Op, I-1, through I-8) is indicated, but the exact number of I-cycles taken during any I-phase depends on the instruction length. Each instruction character reads out of storage on a separate I-cycle. An additional I-cycle is needed to recognize the end of the instruction (the first word mark encountered after an I-Op cycle).

Two exceptions to this rule are the SET WORD MARK (two addresses) and the UNCONDITIONAL BRANCH instructions. The I-phase portion of the operation is automatically ended after the I-7 cycle on the SET WORD MARK operation and after the I-4 cycle on the unconditional branch operation.

As each character is read out of core storage and placed in its proper register, it is also transferred back into core storage for later use.

### Active Components

There are four components that both receive and transmit data during an I-phase. These components are the I-address register, the storage-address register, the core-storage area, and the B-register.

#### I-ADDRESS REGISTER

During I-phase, the I-address register (I-Add. Reg.) specifies the core-storage position that is read out next, with one exception. This is during branch operations when the A-Add. Reg. specifies the core-storage position that reads out next. This address is transferred to the storage-address register, which does the actual core-storage addressing. The address in the I-Add. Reg. is modified by +1 so that another instruction character can be read out on the following storage cycle.

#### STORAGE-ADDRESS REGISTER

The storage-address register (STAR) sets up the core-storage selection circuitry so that the requested core-storage position contents read out to the B-register.

#### CORE STORAGE

The core-storage area reads out the contents of the specified core-storage position to the B-register.

#### B-REGISTER

The B-register accepts the core-storage read-out and, depending on the I-cycle involved, transmits it to other

registers. It also sends the instruction character back into core storage where it is stored in the same position from which it came.

There are other registers that receive certain instruction-word characters. For the most part, however, these are stored in the registers for use during the E-phase. These registers and their contents are pointed out when they receive the data.

The various I-cycle operations are shown as they appear during a single-cycle operation.

### I-Op Cycle (Figure 4)

The I-phase is started when the last execute cycle completion signals the system to end the E-phase and start the I-phase. The first I-phase trigger (I-Op) is turned on and the OP instruction-length light on the 1401 console panel turns on.

The controlling circuitry now determines the starting address of the instruction that will be read out of core storage during this I-phase. If the next sequential instruction (NSI) in the stored program is used, the I-Add. Reg. already contains the core-storage address of its first character. This address was established during the previous I-phase. The address in the I-Add. Reg. is transferred to the STAR and the I-Add. Reg. key-light on the 1401 console panel turns on. The STAR lights on the console panel display the core-storage position being addressed (in BCD bit form).

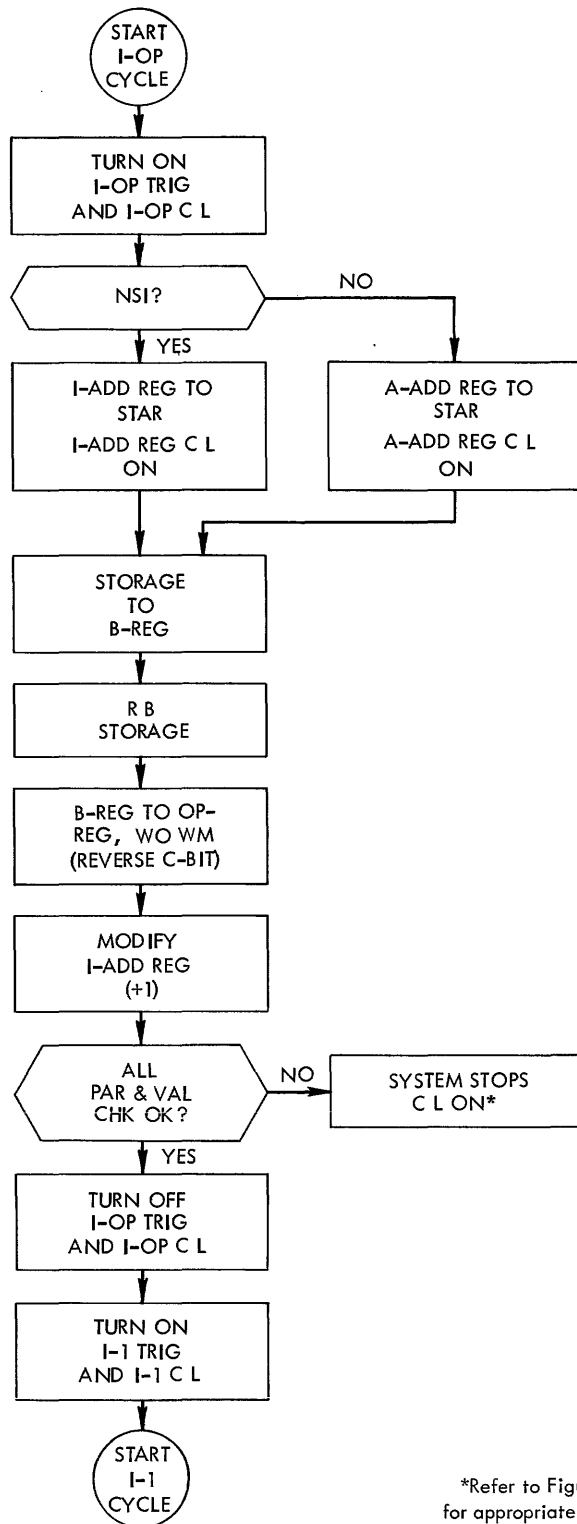
*NOTE: If the previous instruction was a branch instruction, and a branch was initiated, the core-storage address in the A-Add. Reg. is used. The A-Add. Reg. obtained this address (the I-address of a branch instruction) during the previous I-phase when the branch instruction was read out of core storage.*

The STAR activates the lines that cause the specified core-storage position to read out to the B-register where the contents of that position are displayed in BCD form. This core-storage position contains the operation code of the instruction. From the B-register, the Op-code character:

1. is transferred back into core storage where it is stored in the same position it came from
2. is transferred into the Op-register. During the transfer, the character is stripped of the word mark, and a C-bit is added or deleted (depends on original bit configuration) for parity purposes.

The core-storage position specified in the I-Add. Reg. is changed by adding one to it. This is accomplished by reading out the contents of the STAR and adding one to it before reading it back into the I-Add. Reg. The I-Add. Reg. now contains the address of the character that follows the Op-code character.

The parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. (An Op-register par-



\*Refer to Figure 2 for appropriate lights

Figure 4. I-Op Cycle

ity or validity check will not show up during an I-Op cycle, however.)

If no check condition occurs, the I-Op trigger is turned off. This turns off the op instruction length light on the console panel and turns on the I-1 trigger. The

I-1 instruction length light on the console panel is then turned on.

### I-1 Cycle (Figure 5)

The I-1 trigger and its associated I-1 instruction-length light turn on when the I-Op trigger turns off. The I-Add. Reg. already contains the core-storage location of the second instruction character. (Actually, the STAR already contains the core-storage location of the second instruction character. This address was transferred from the I-Add. Reg. to the STAR in the last part of the previous I-cycle, but this is not evident during a single-cycle operation.) This address is transferred to the STAR, which then addresses core storage. The character is read out of storage and into the B-register. From the B-register it is read back into core storage, and is also checked to see if the character has a word mark associated with it (signifying another Op code).

### I-CHARACTER INSTRUCTION

Normally, the second character of an instruction is the hundreds-thousands position of an A-field address. If the character does have a word mark, it means that this character is the Op code for the next instruction. Only one character has been read out of core storage (Op code) so the previous addresses in the A- and B-address registers will be used in the execution of this instruction if this is a *chaining* type of operation.

The previous Op code (read out during the I-Op cycle) is checked to see if it is one of the Op codes that has to set up the alteration of some of the normal E-phase operations before the actual instruction execution begins. If E-phase alteration is not needed, the Op-register and all other appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-phase ends and the E-phase begins.

### 2-CHARACTER INSTRUCTION

If this second character does not have a word mark, it means that the instruction is at least two characters long. It may be only two characters long, but this cannot be established until I-2 cycle time. Because of this possibility, the character is transferred from the B-register to the A-register. The A-register normally stores the operation modifier character and, in a 2-character instruction, the second character would be the modifier.

### OTHER LENGTH INSTRUCTION

Because this might be a longer-length instruction, the character must be stored in the hundreds-thousands position of the A- and B-Add. Reg. However, for certain operations, the character should be placed in the hundreds-thousands position of the A-Add. Reg. only.

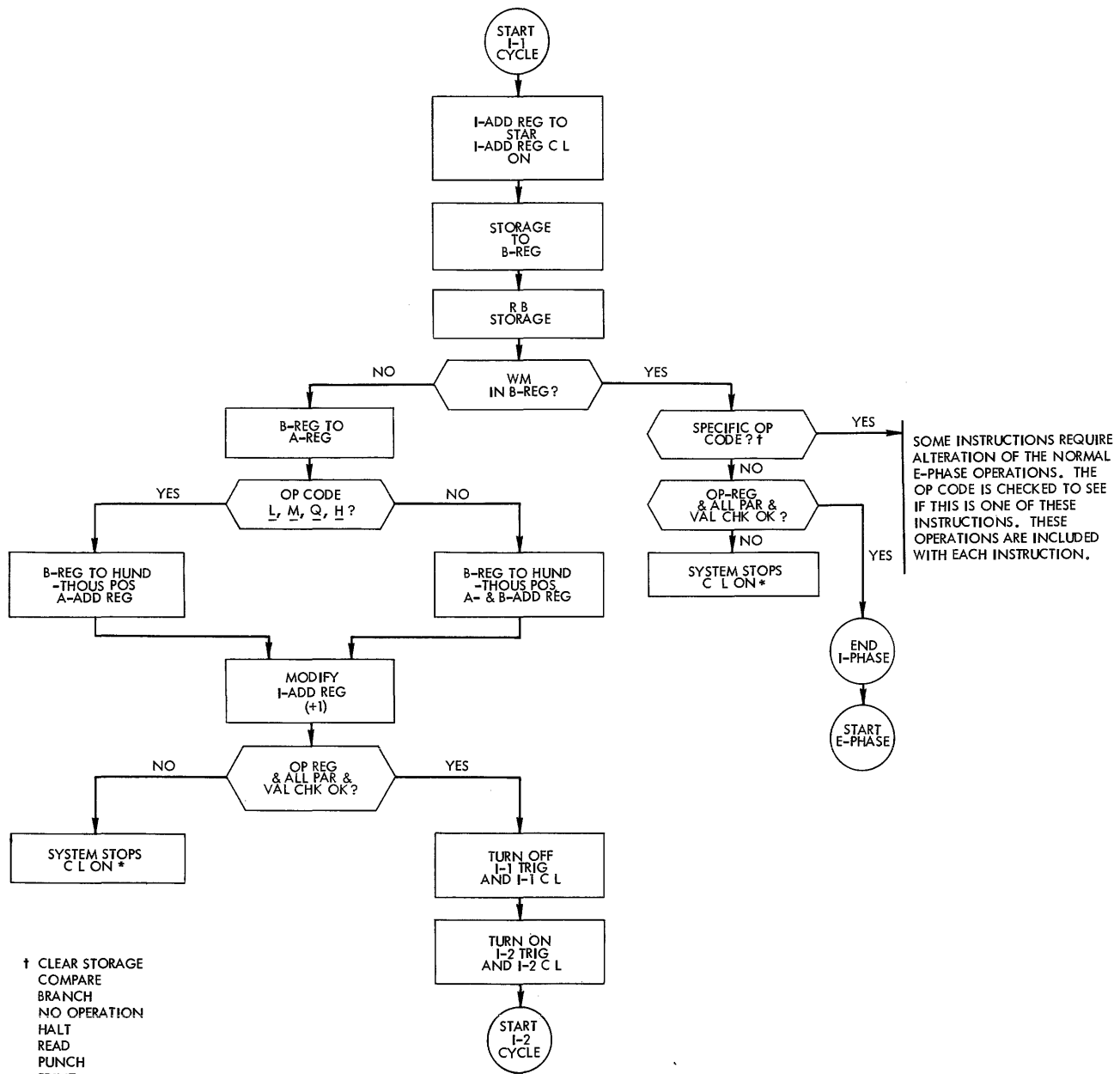


Figure 5. I-1 Cycle

The Op code is checked, and if it is an L, M, Q, or H Op code, the character enters the A-Add. Reg. only. If it is not one of these codes, it is placed in the hundreds-thousands position of the A-Add. Reg. and B-Add. Reg.

The address in the I-Add. Reg. is increased by one, and the Op-register and all other appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-1 trigger and its associated console light are

turned off. This turns on the I-2 trigger and its associated console light.

### I-2 Cycle (Figure 6)

The I-2 trigger and its associated instruction-length light are turned on when the I-1 trigger is turned off. The I-Add. Reg. already contains the core-storage location of the third instruction character. (Actually, the STAR already contains the core-storage location of the third instruction character. This address was transferred from the I-Add. Reg. to the STAR in the last part



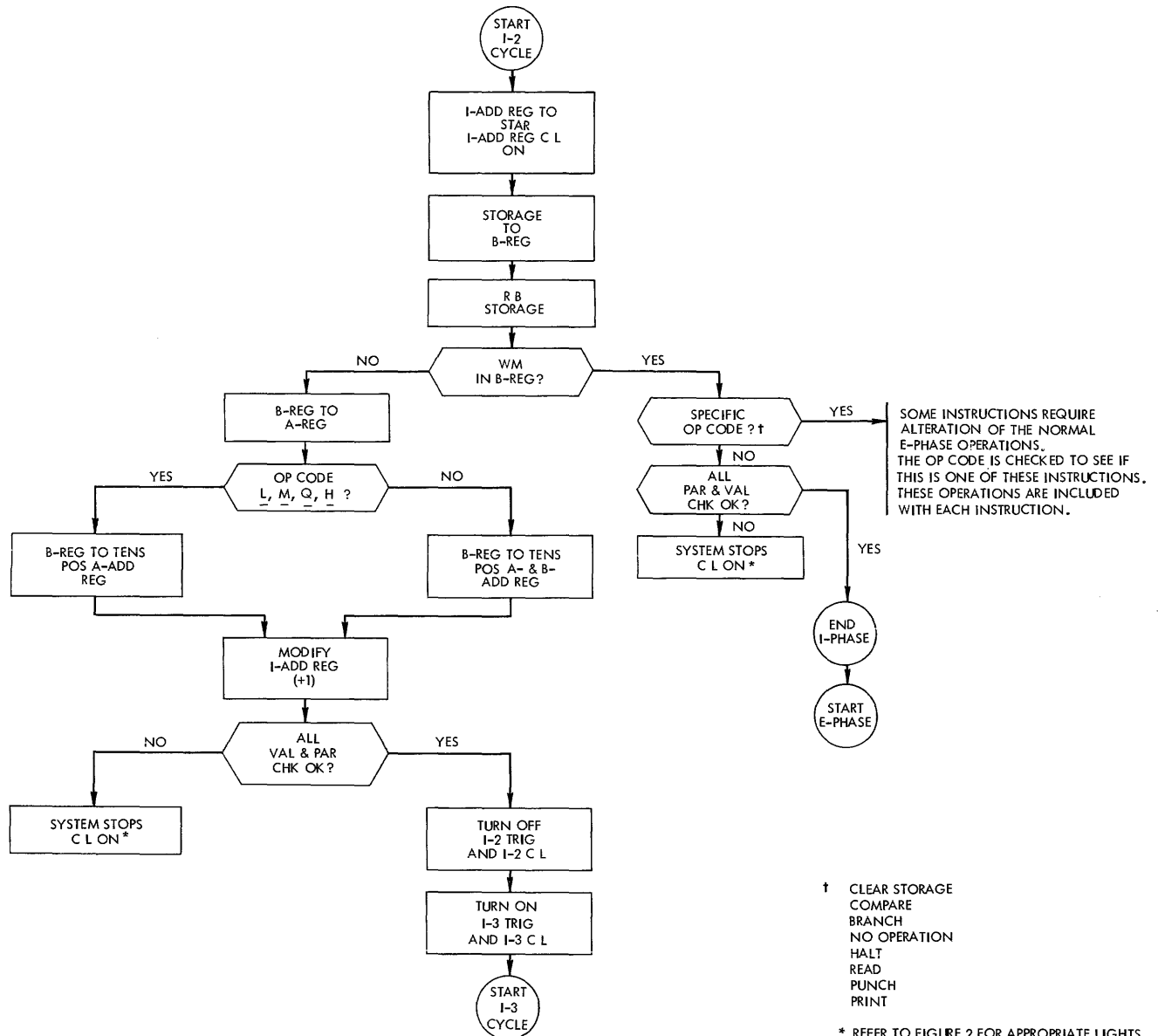


Figure 6. I-2 Cycle

of the previous I-cycle, but this is not evident during a single-cycle operation.) This address is transferred to the STAR which then addresses core storage. The character is read out of storage and into the B-register. From the B-register it is read back into core storage and is also checked to see if the character has a word mark associated with it (signifying another Op code).

#### 2-CHARACTER INSTRUCTION

Normally, the third character of an instruction is the tens-position character of an A-field address. If the character does have a word mark, it means that this character is the Op code for the next instruction. The two characters read out of core storage make up the complete instruction.

The previous Op code (read out during the I-Op cycle) is checked to see if it is one of the Op codes that has to set up the alteration of the normal E-phase operations before the actual instruction execution begins. If E-phase alteration is not needed, the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-phase ends and the E-phase begins.

#### OTHER LENGTH INSTRUCTION

If this third character does not have a word mark, it means that the instruction is at least four characters long (Op code plus one 3-character address). It may be longer, but this cannot be established now.

The character is transferred from the B-register to the A-register, where it replaces the character that was stored there during the I-1 cycle. This character could not be an operation modifier character, but it is still stored in the A-register.

Also, the character must be stored in the tens position of the A- and B-Add. Reg. However, for certain operations, the character should be placed in the tens position of the A-Add. Reg. only. The Op code is checked, and if it is an L, M, Q, or H Op code, the character enters the A-Add. Reg. only. If it is not one of these codes, it is placed in the tens position of the A-Add. Reg. and B-Add. Reg.

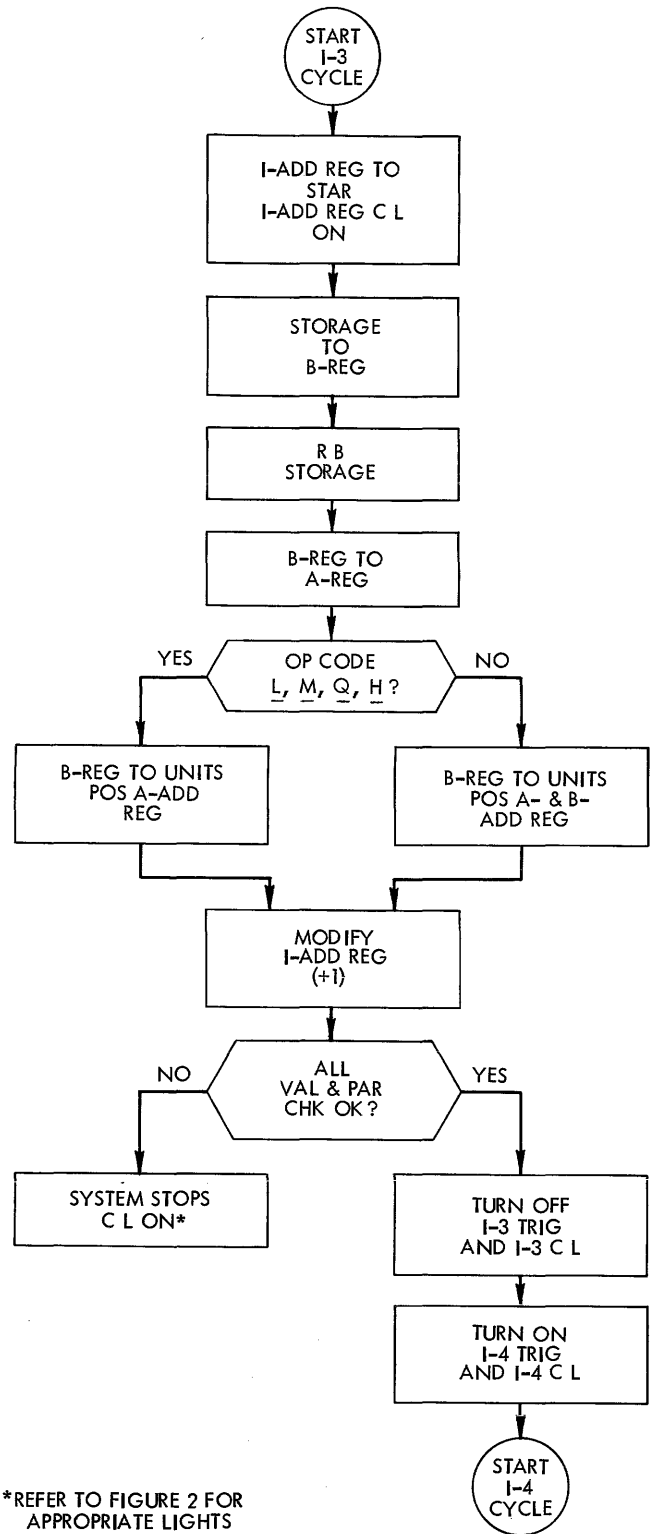
The address in the I-Add. Reg. is increased by one, and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-2 trigger and its associated console light are turned off. This turns on the I-3 trigger and its associated console light.

**I-3 Cycle (Figure 7)**

The I-3 trigger and its associated instruction-length light are turned on when the I-2 trigger is turned off. The I-Add. Reg. already contains the core-storage location of the fourth instruction character. (Actually, the STAR already contains the core-storage location of the fourth instruction character. This address was transferred from the I-Add. Reg. to the STAR in the last part of the previous I-cycle, but this is not evident during a single-cycle operation.) This address is transferred to the STAR, which then addresses core storage. The character is read out of storage and into the B-register. From the B-register it is read back into core storage and read into the A-register.

It is possible that this may be the last character of the instruction, but this cannot be established before the next I-cycle. Regardless of the final length, this fourth character must be stored in the units position of the A-Add. Reg. and B-Add. Reg. However, the same conditions present during the I-1 and I-2 cycles are still valid here. The Op code is checked, and if it is an L, M, Q, or H Op code, the character enters the units position of the A-Add. Reg. only. If it is not one of these Op codes, it is placed in the units position of the A-Add. Reg. and B-Add. Reg.

The address in the I-Add. Reg. is increased by one, and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-3 trigger and its associated console light are turned off. This turns on the I-4 trigger and its associated console light.



\*REFER TO FIGURE 2 FOR APPROPRIATE LIGHTS

Figure 7. I-3 Cycle

**I-4 Cycle (Figure 8)**

The I-4 trigger and its associated instruction-length light are turned on when the I-3 trigger is turned off. The I-Add. Reg. already contains the core-storage location of the fifth instruction character. (Actually, the

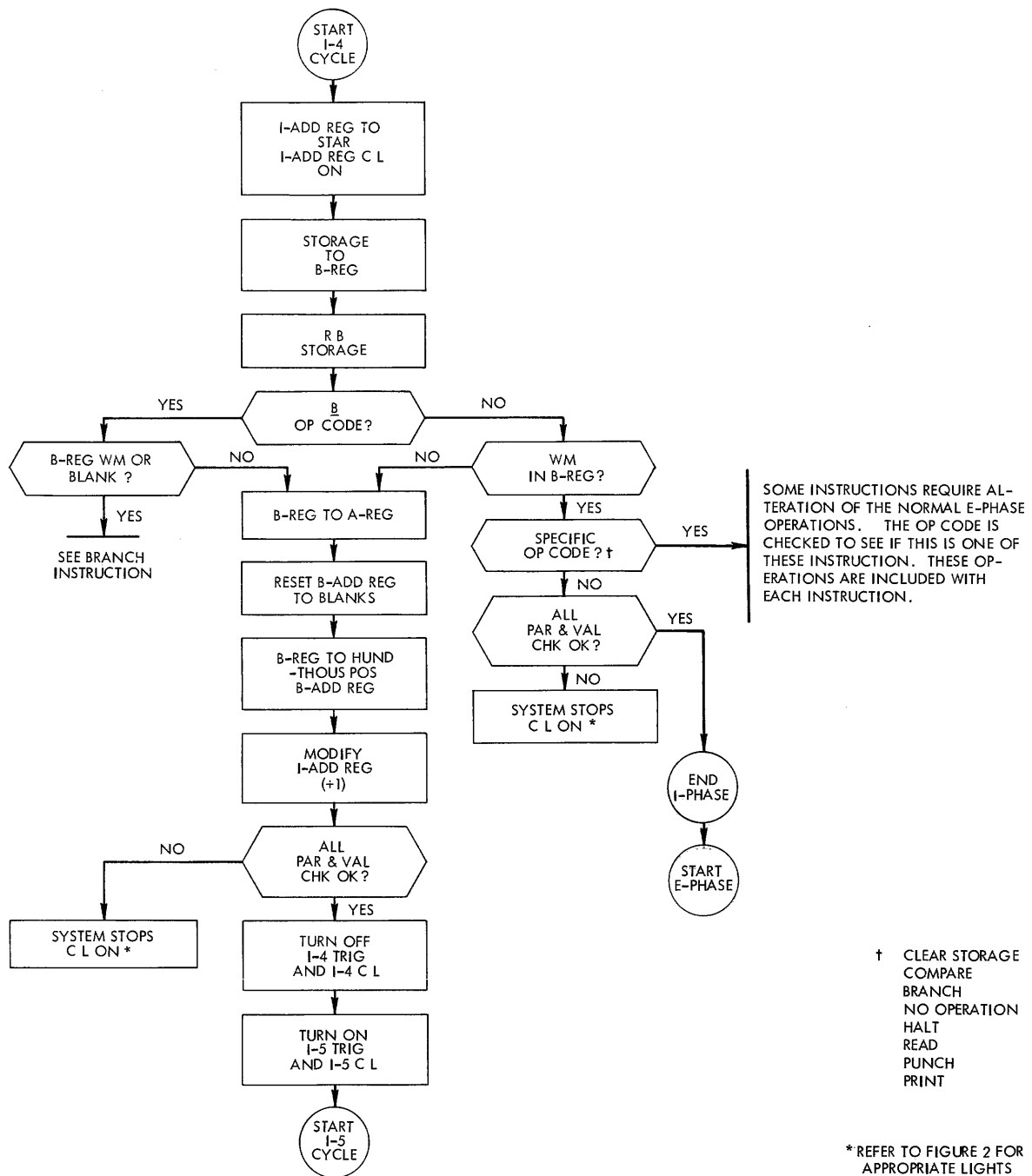


Figure 8. I-4 Cycle

STAR already contains the core-storage location of the fifth instruction character. This address was transferred from the I-Add. Reg. to the STAR in the last part of the previous I-cycle, but this is not evident during a single-cycle operation.) This address is transferred to the STAR, which then addresses core storage. The character is read out of storage and into the B-register. From the B-register it is read back into core storage.

#### BRANCH INSTRUCTION

The Op code stored in the Op-register is checked to see if it is a B. If it is a B, then the B-register contents are checked to see if it is a blank character (C-bit only), or whether the character has a word mark associated with it. If either of these conditions are present, it signifies a BRANCH instruction (unconditional). Refer to the BRANCH instruction for the remainder of the operation.

If it is still a B Op code, but the B-register contains

something else, it signifies a BRANCH IF INDICATOR ON instruction or a BRANCH IF CHARACTER EQUAL instruction. The B-register then sends this fifth instruction character (the operation-modifier character) to the A-register and the hundreds-thousands position of the B-Add. Reg. (B-Add. Reg. previously reset to blanks).

The address in the I-Add. Reg. is increased by one, and the appropriate parity and validity checks are made. Any check condition stops the system, and turns on the appropriate check lights on the console panel.

If no check condition occurs, the I-4 trigger and its associated console light are turned off. This turns on the I-5 trigger and its associated console light.

OTHER INSTRUCTION

If the Op code stored in the Op-register is not a B, then the character in the B-register is checked for an associated word mark. If there is no word mark, then the B-register contents are sent to the A-register and the hundreds-thousands position of the B-Add. Reg. (B-

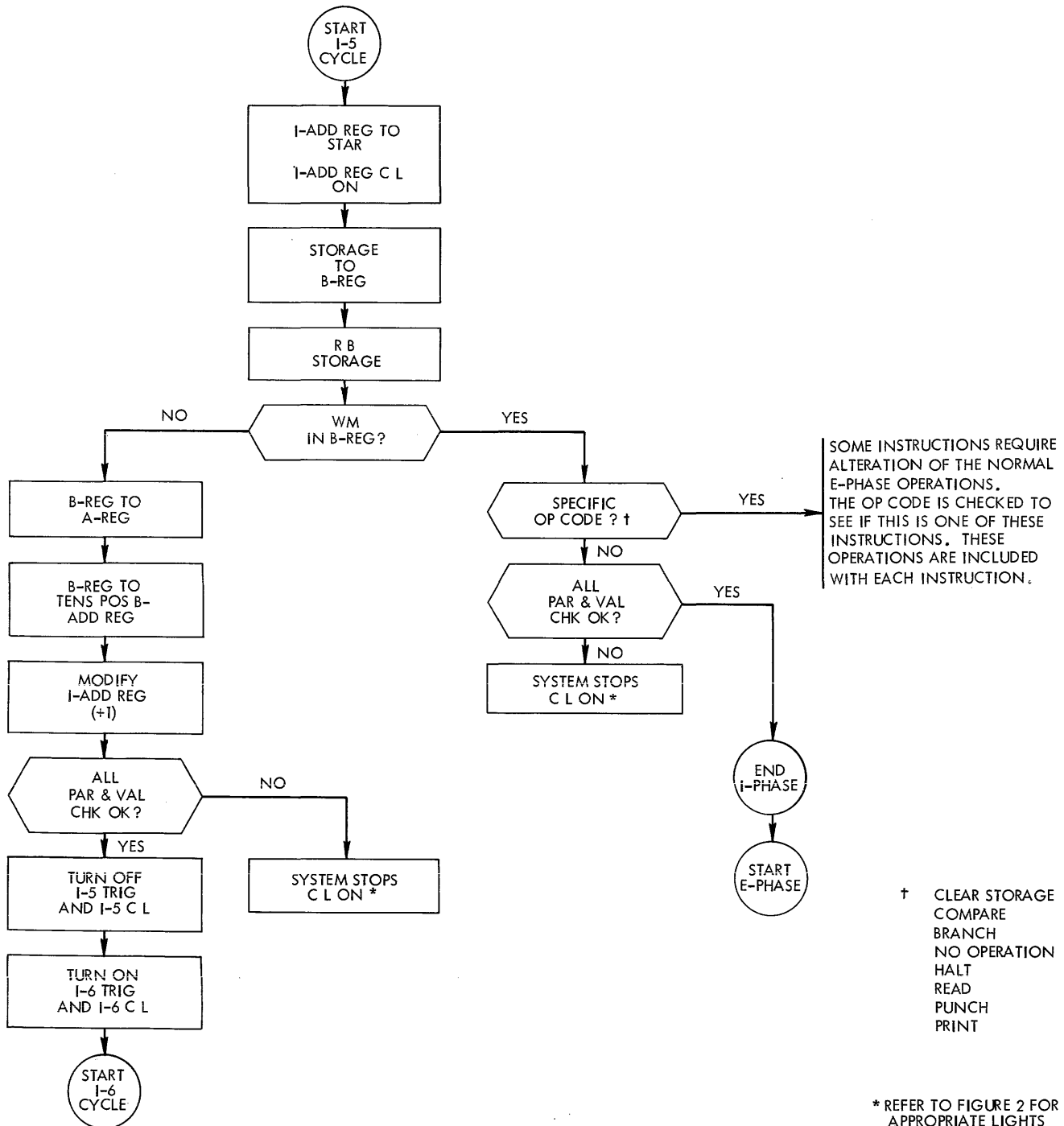


Figure 9. I-5 Cycle

Add. Reg. previously reset to blanks). The I-Add. Reg. modification, the parity and validity checking, and the I-cycle progression previously described now occur.

If there is a word mark in the B-register (signifying a new Op code), then the Op code read out during the I-Op cycle is checked to see if it is one of the Op codes that has to set up the alteration of the normal E-phase operations before the actual instruction execution begins. If E-phase alteration is not needed, the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-phase ends and the E-phase begins.

#### I-5 Cycle (Figure 9)

The I-5 trigger and its associated instruction-length light are turned on when the I-4 trigger is turned off. The I-Add. Reg. already contains the core-storage location of the sixth instruction character. (Actually, the STAR already contains the core-storage location of the sixth instruction character. This address was transferred from the I-Add. Reg. to the STAR in the last part of the previous I-cycle, but this is not evident during a single-cycle operation.) This address is transferred to the STAR, which then addresses core storage. The character is read out of storage and into the B-register. From the B-register it is read back into core storage, and the character is also checked to see if it has a word mark associated with it (signifying another Op code).

#### 5-CHARACTER INSTRUCTION

If the character has a word mark associated with it, then the five preceding characters constitute a complete instruction (Op code, a 3-character address, and an operation-modifier character). The Op code read out during the I-Op cycle is checked to see if it is one of the Op codes that has to set up the alteration of the normal E-phase operations before the actual instruction execution begins. If E-phase alteration is not needed, the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-phase ends and the E-phase begins.

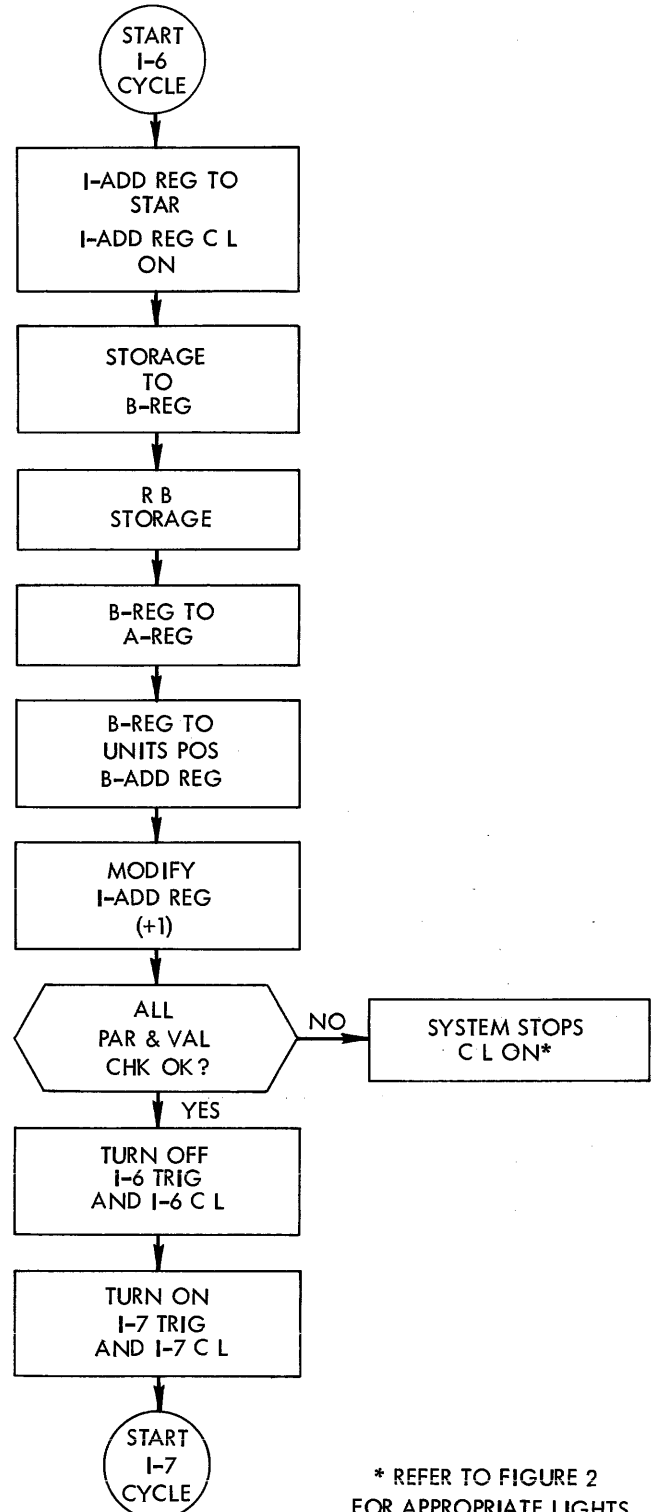
#### OTHER INSTRUCTION

If the character does not have a word mark associated with it, then the B-register contents are sent to the A-register and the tens position of the B-Add. Reg.

The address in the I-Add. Reg. is increased by one, and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-5 trigger and its associated console light are turned off. This turns on the I-6 trigger and its associated console light.

#### I-6 Cycle (Figure 10)

The I-6 trigger and its associated instruction-length light are turned on when the I-5 trigger is turned off. The I-Add. Reg. already contains the core-storage location of the seventh instruction character. (Actually, the STAR already contains the core-storage location of



\* REFER TO FIGURE 2 FOR APPROPRIATE LIGHTS

Figure 10. I-6 Cycle

the seventh instruction character. This address was transferred from the I-Add. Reg. to the STAR in the last part of the previous I-cycle, but this is not evident during a single-cycle operation.) This address is transferred to the STAR, which then addresses core storage. The character is read out of storage and into the B-register. From the B-register it is read back into core storage, and read into the A-register, and read into the

units position of the B-Add. Reg.

The address in the I-Add. Reg. is increased by one, and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-6 trigger and its associated console light are turned off. This turns on the I-7 trigger and its associated console light.

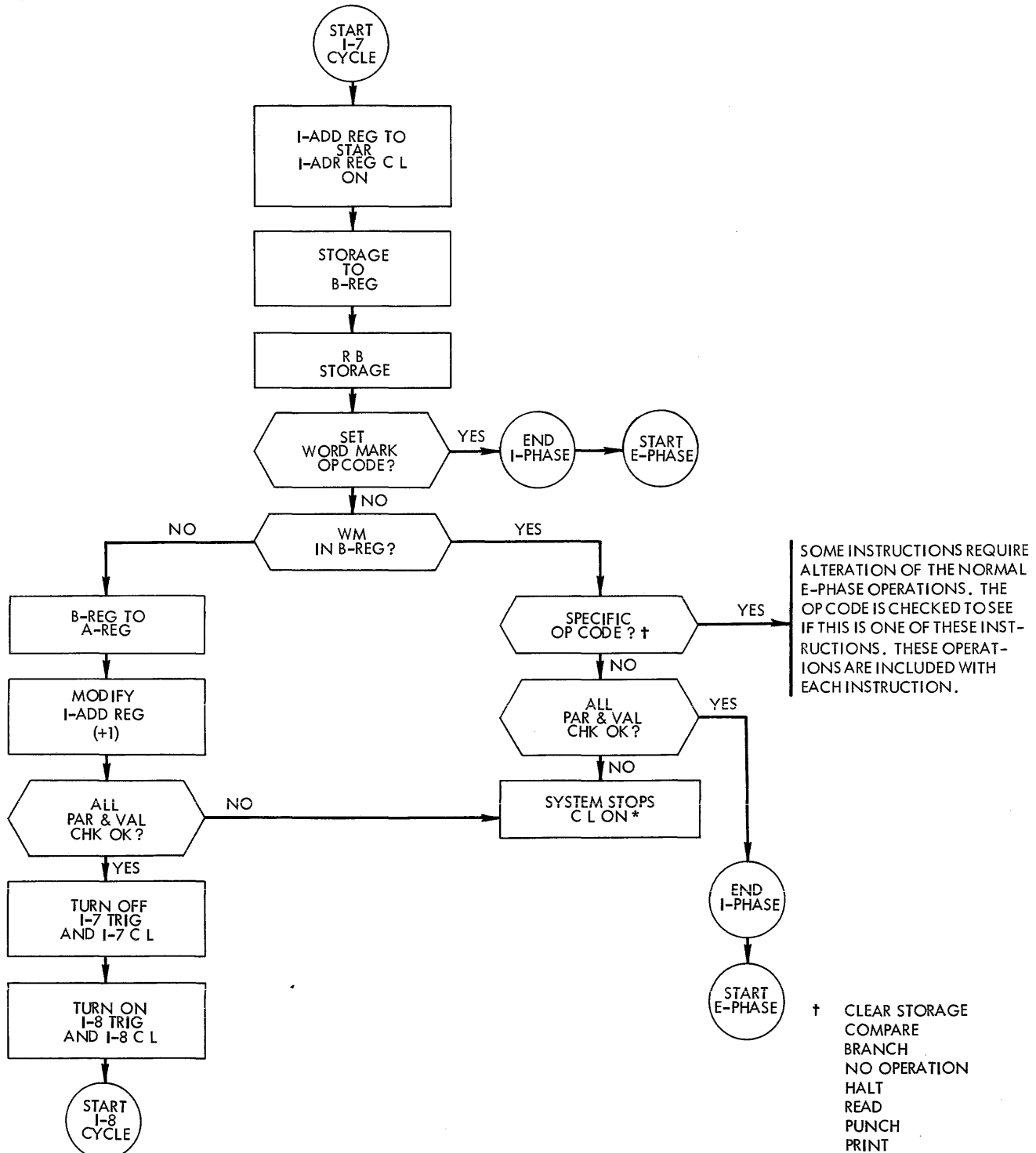


Figure 11. I-7 Cycle

### I-7 Cycle (Figure 11)

The I-7 trigger and its associated instruction-length light are turned on when the I-6 trigger is turned off. The I-Add. Reg. already contains the core-storage location of the eighth instruction character. (Actually, the STAR already contains the core-storage location of the eighth instruction character. The address was transferred from the I-Add. Reg. to the STAR in the last part of the previous I-cycle, but this is not evident during a single-cycle operation.) This address is transferred to the STAR, which then addresses core storage. The character is read out of storage and into the B-register. From the B-register it is read back into core storage.

#### SET WORD MARK INSTRUCTION

The Op code stored in the Op-register is checked to see if it is a *set word mark* Op code. If it is, the I-phase ends and the E-phase begins. This is done so that word marks can be set in specified core-storage locations during the program-loading routine.

If the Op code stored in the Op-register is not a *set word mark* Op code, then the B-register character is checked to see if it has a word mark associated with it (signifying another Op code).

#### 7-CHARACTER INSTRUCTION

If the character has a word mark associated with it, then the seven preceding characters constitute a complete instruction (Op code and two 3-character addresses). The Op code read out during the I-Op cycle is checked to see if it is one of the Op codes that has to set up alteration of some of the normal E-phase operations before the actual instruction execution begins. If E-phase alteration is not needed, the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check conditions occur, the I-phase ends and the E-phase begins.

#### OTHER INSTRUCTION

If the character does not have a word mark associated with it, then the character must be an operation-modifier character, and the B-register contents are sent to the A-register.

The address in the I-Add. Reg. is increased by one, and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-7 trigger and its associated console light are turned off. This turns on the I-8 trigger and its associated console light.

### I-8 Cycle (Figure 12)

The I-8 trigger and its associated instruction-length light are turned on when the I-7 trigger is turned off.

The I-Add. Reg. already contains the core-storage location of the next character, which should be the Op-code character of the next instruction. (Actually, the STAR already contains this core-storage location. The address was transferred from the I-Add. Reg. to the STAR in the last part of the previous I-cycle, but this is not evident during a single-cycle operation.) This address is transferred to the STAR, which then addresses core storage. The character is read out of storage and into the B-register. From the B-register it is read back into core storage. The character is also checked to see if it has a word mark associated with it (signifying another Op code).

#### 8-CHARACTER INSTRUCTION

If the character has a word mark associated with it, then the eight preceding characters institute a complete instruction (Op code, two 3-character addresses, and an operation-modifier character). The Op code read out during the I-Op cycle is checked to see if it is one of the Op codes that has to set up alteration of some of the normal E-phase operations before the actual instruction execution begins. If E-phase alteration is not needed, the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check conditions occur, the I-phase ends and the E-phase begins.

#### PROGRAMMING ERROR

If the character does not have a word mark associated with it, then the B-register contents are sent to the A-register. The address in the I-Add. Reg. is increased by one, and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, then the I-8 cycle is repeated until a character with a word mark is read out of core storage. This condition signifies poor programming technique because the program has gaps between sequential instructions.

### **Instruction Execution (E-Phase)**

Normally, the word mark associated with the Op code of the next instruction signals the end of the I-phase. Ending the I-phase automatically starts the E-(execute) phase.

The E-phase is the system operating time necessary to perform the operation specified by the instruction read out during I-phase. The E-phase is made up of A-cycles and/or B-cycles. During an A-cycle, one position of data from the previously-specified A-field is read out of core storage. During a B-cycle, one position of data from the previously-specified B-field is read out of core storage.

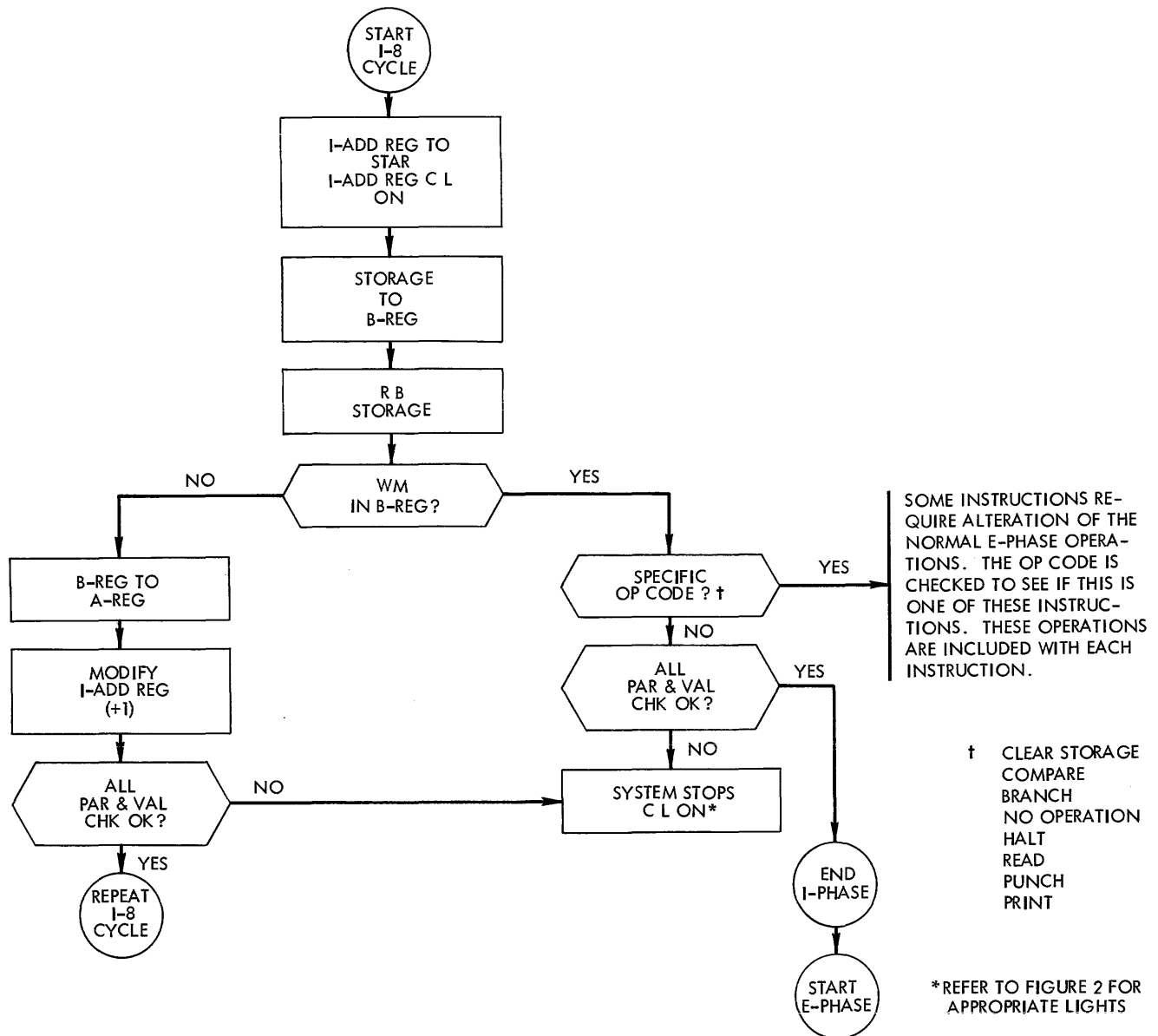


Figure 12. I-8 Cycle

### A-Cycle (Figure 13)

The first cycle during any E-phase is usually an A-cycle (some instructions have no A-cycles). The A-cycle illustrated in Figure 13 is an A-cycle operation that is common to many 1401 instructions. When discussing instructions that use the common A-cycle, reference is made to Figure 13. A-cycles that are different are covered in that instruction writeup.

The A-Add. Reg. contains the core-storage location specified in the A-address portion of the previously-read instruction. This core-storage location is, normally, the units position of the A-field. The address in the A-Add. Reg. is transferred to the STAR, and the A-Add. Reg. key-light on the 1401 console panel is turned on. The STAR lights on the console panel display the core-

storage position being addressed (in BCD bit form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register where the contents of that position are displayed in BCD form. From the B-register, it is normally read back into the core-storage position it came from, and it is also transferred to the A-register where it is displayed in BCD form.

The core-storage position specified in the A-Add. Reg. is changed by subtracting one from it. This is done because the units position of a data field occupies the highest-numbered core-storage location and the rest of the field occupies the adjacent lower-numbered core-storage locations. To perform the arithmetic functions correctly, the units digit of a field must be worked on



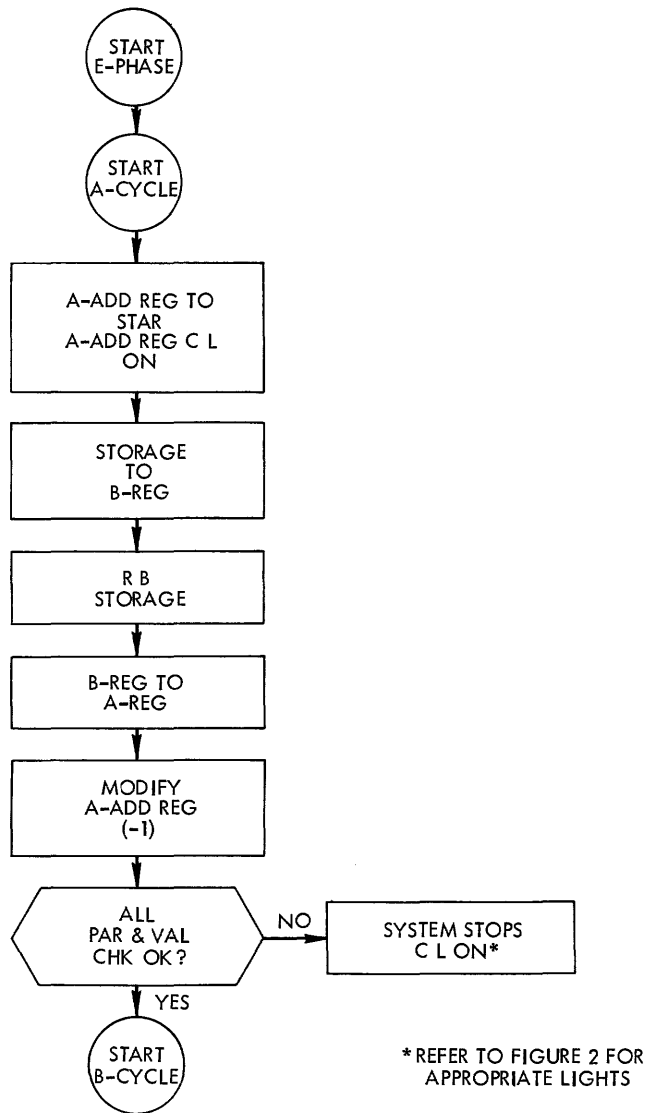


Figure 13. A-Cycle

first, followed by the tens digit, etc. Therefore, the address in the A-Add. Reg. must be modified by  $-1$  each time so that the adjacent lower-numbered location is read out. This  $-1$  modification is accomplished by reading out the contents of the STAR and subtracting one from it before reading it back into the A-Add. Reg.

The parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel.

If no check condition occurs, the A-cycle ends and the B-cycle begins (the STORE ADDRESS REGISTER instructions are exceptions).

### B-Cycle

It is during the B-cycle that the instruction execution usually takes place. In an add operation, for instance, the actual addition of the A- and B-field characters takes place during the B-cycle and the result is stored

in the B-field position. Therefore, all instructions discussed in this publication normally show the system's method of operation during a B-cycle. However, when the A-cycles differ from the common A-cycle, this difference is discussed along with any other forms of instruction execution that may be used.

### E-Phase Termination

The E-phase portion of an instruction is usually ended when a word mark is sensed in the B-register. The word mark signifies that the end of that word in core storage has been reached. However, there are operations that stop only when an A-field word mark is sensed. Other operations stop only when a B-field word mark is sensed, and still other operations stop when either an A-field or a B-field word mark is sensed. These different conditions are all included in the system's internal circuitry and the actual operation termination is done automatically.

## Logic Operations

### Branch (B III)

The BRANCH (unconditional) instruction (B III) always causes the program to branch to the specified I-address. This address contains the Op code of some instruction. This branch operation is used to interrupt normal program sequence, and to continue the program at some other desired point, without testing for any specific conditions.

#### I-Phase Operation (Figures 8 and 14)

During I-4 time of an I-phase, the Op-register is checked to see if it contains a B Op code. When a B Op code is established, the B-register is checked to see if it contains a blank character or a word mark (the Op-code position of the next sequential instruction). When one of these two conditions is established, the circuitry to eliminate the E-phase portion of this instruction is activated.

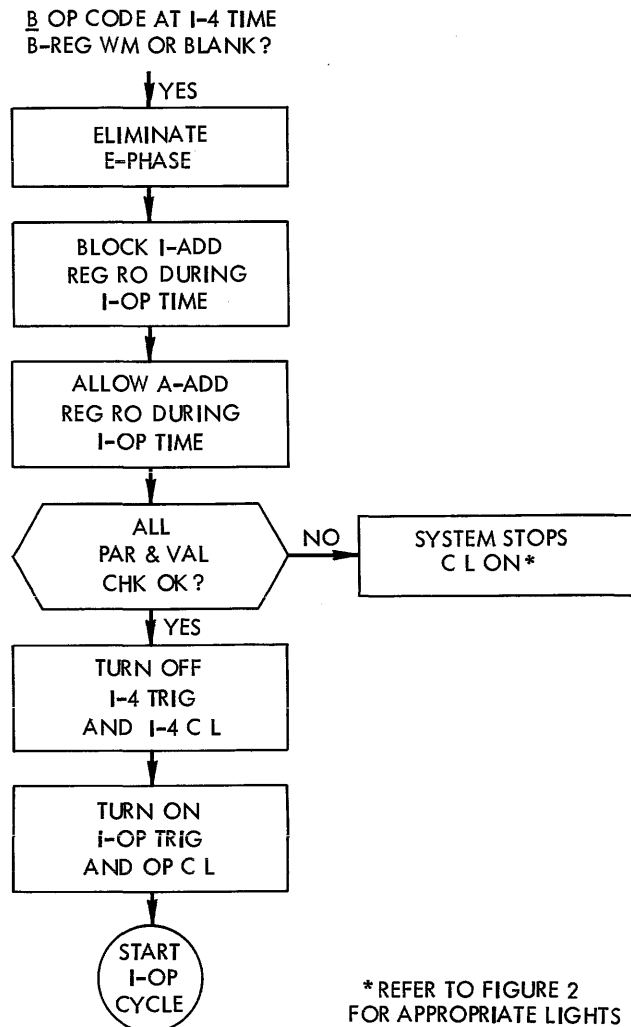


Figure 14. Branch Instruction I-Phase Operation

Also, circuitry is activated that blocks the I-Add. Reg. read-out, and allows the A-Add. Reg. read-out during the next I-Op cycle. (The A-Add. Reg. contains the I-address as specified by the instruction.)

The appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-4 trigger and its associated console light are turned off. This turns on the I-Op trigger and its associated console light. During the I-Op cycle, the core-storage position specified by the I-address (in the A-Add. Reg.) is read out of core storage, followed by a normal I-phase operation.

### Branch if Indicator On (B III d)

The BRANCH IF INDICATOR ON instruction (B III d) causes the program to branch to the specified I-address if the specified indicator, when tested, is on. If the indicator is off, the next sequential instruction is read.

#### I-Phase Operation (Figures 9 and 15)

During I-5 time of an I-phase, the B-register is checked to see if it contains a word mark. If the B-register does contain a word mark (the Op-code position of the next sequential instruction), the Op-register is checked to see if it contains a B.

When the presence of a B Op code is established, the circuitry to eliminate the E-phase portion of this instruction is activated.

The indicator specified by the operation-modifier character (d-character) is tested to see if it is on. If the indicator is on, circuitry is activated that blocks the I-Add. Reg. read-out, and allows the A-Add. Reg. read-out during the next I-Op cycle. (The A-Add. Reg. contains the I-address as specified by the instruction.) If the indicator is not on, the normal I-Add. Reg. read-out is active during the next I-cycle.

The appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-4 trigger and its associated console light are turned off. This turns on the I-Op trigger and its associated console light. During the I-Op cycle, the core-storage position specified by either the I-address (in the A-Add. Reg.) or the I-Add. Reg. is read out of core storage.

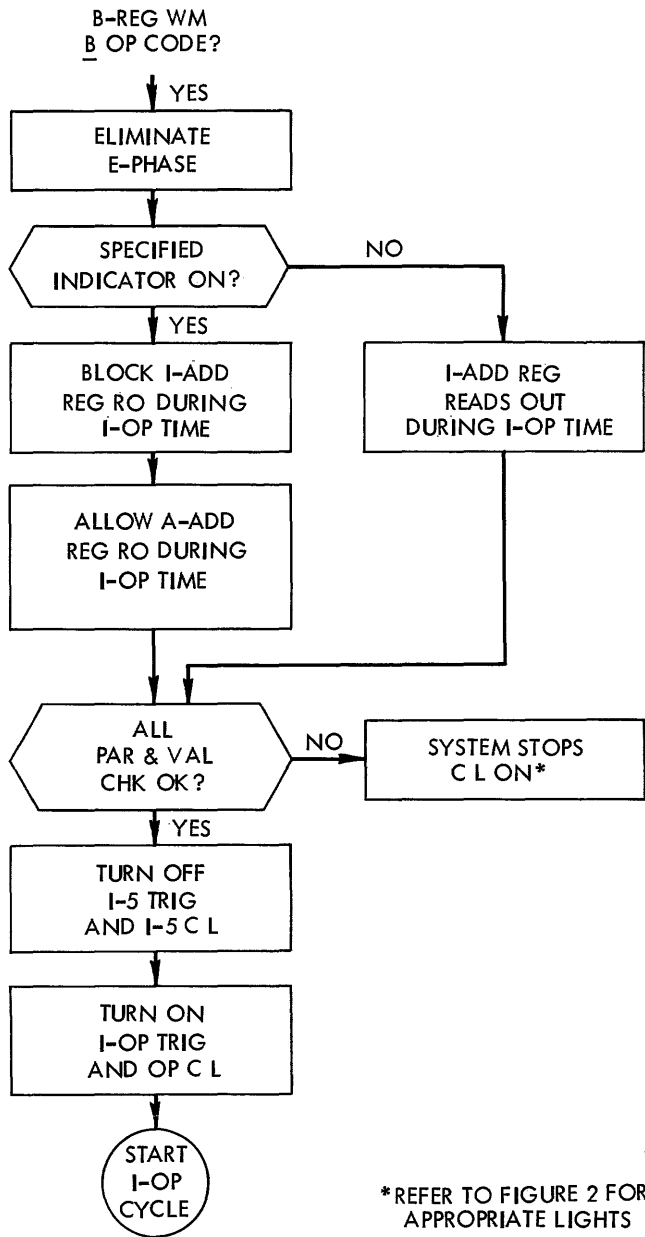


Figure 15. Branch if Indicator On Instruction: I-Phase Operation

**Branch if Character Equal (B III BBB d)**  
**Branch if Word Mark and/or Zone (V III BBB d)**

Both the BRANCH IF CHARACTER EQUAL instruction (B III BBB d) and the BRANCH IF WORD MARK AND/OR ZONE instruction (V III BBB d) cause the program to branch to the specified I-address if the condition specified by the operation-modifier character (d-character) is met by the character located in the specified B-address. The BRANCH IF CHARACTER EQUAL instruction tests the character at the B-address for the same bit configuration as the d-character, and branches if the bit configuration is

d-character	Condition
1	Word mark
2	No zone (No-A, No-B-bit)
B	12-zone (AB-bits)
K	11-zone (B, No-A-bit)
S	Zero-zone (A, No-B-bit)
3	Either a word mark, or no zone
C	Either a word mark, or 12-zone
L	Either a word mark, or 11-zone
T	Either a word mark, or zero-zone

Figure 16. Branch if Word Mark and/or Zone d-Characters and Conditions

the same. The BRANCH IF WORD MARK AND/OR ZONE instruction tests the character at the B-address for the condition specified by the d-character, and branches if the condition is met. The d-characters and the conditions they represent are shown in Figure 16.

**I-Phase Set-Up Operations (Figure 17)**

Some instructions require alterations of the normal E-phase operations for correct execution of the instruction. These alterations are set up before the I-phase ends. During certain I-cycles the Op code is checked to see if it is one of the Op codes that has to set up the alteration of the normal E-phase operations before the actual instruction execution begins. As soon as the word mark is sensed in the B-register, either a B or V Op code sets up the circuitry to eliminate A-cycles and execute one B-cycle. The appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-phase ends and the E-phase begins.

**Single B-Cycle (Figure 17)**

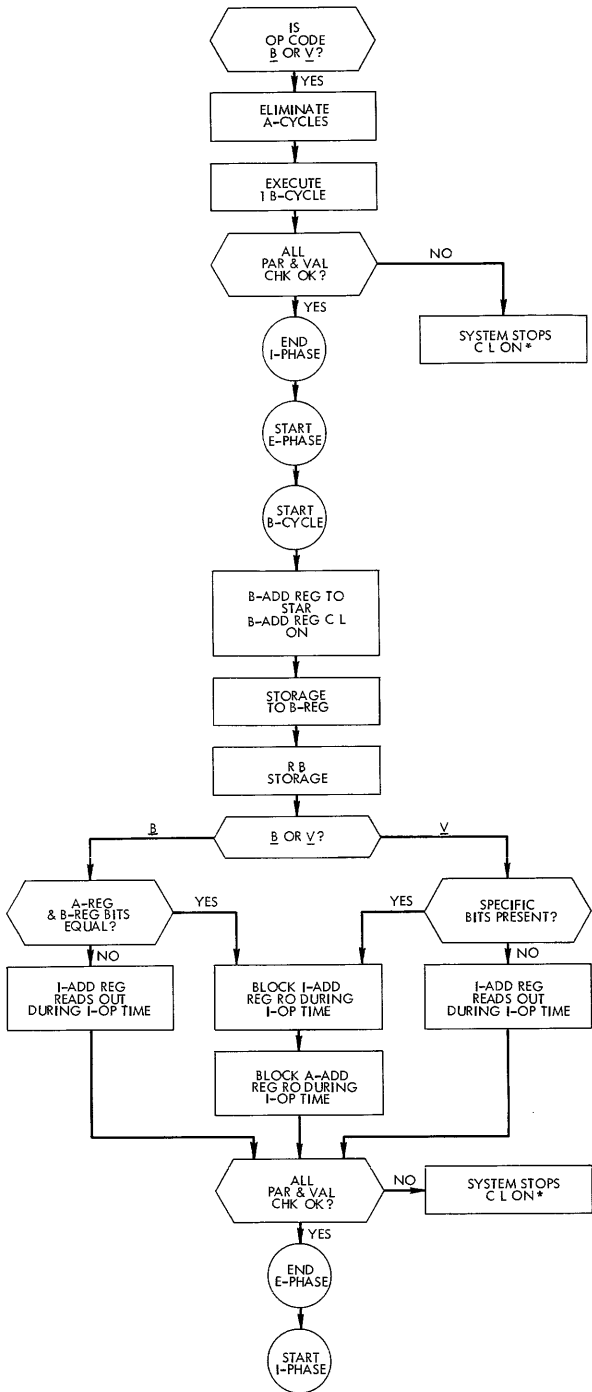
The core-storage position containing the character that is checked by the d-character during this single B-cycle was previously specified by the instruction, and the address was placed in the B-Add. Reg. during I-phase. The address in the B-Add. Reg. is transferred to the STAR and the B-Add. Reg. key-light on the console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.

The character in the B-register is read back into the core-storage position it came from. The Op code is then checked to see if it is a B or a V Op code.

**B OP CODE**

If it is a B Op code, then the A-register contents (the d-character) are compared against the B-register con-



\* REFER TO FIGURE 2 FOR APPROPRIATE LIGHTS

Figure 17. Branch if Character Equal and Branch if Word Mark and/or Zone Instructions: I- and E-Phase Operations

tents (the B-field character). If the bit configurations are the same, the branch to the specified I-address occurs. The circuitry is activated that blocks the I-Add. Reg. read-out, and allows the A-Add. Reg. read-out during the next I-Op cycle. (The A-Add. Reg. contains the I-address as specified by the instruction.) If the bit

configurations are not the same, the normal I-Add. Reg. read-out is active during the next I-Op cycle.

#### V OP CODE

If the Op code is a V Op code, then the B-register contents (the B-field character) is checked to see if the bits specified by the A-register contents (the d-character) are present. If the specified bits are present, the branch to the specified I-address occurs. The circuitry is activated that blocks the I-Add. Reg. read-out, and allows the A-Add. Reg. read-out during the next I-Op cycle. (The A-Add. Reg. contains the I-address as specified by the instruction.)

If the specified bits are not present, the normal I-Add. Reg. read-out is active during the next I-Op cycle.

The appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the E-phase ends and I-phase begins.

#### No Operation (N)

An instruction with an N Op code performs no operation. It can be substituted for the operation code of any instruction to make that instruction ineffective. It is commonly used in program modification to cause the program to skip over a specific instruction. This instruction skipping is accomplished by eliminating the E-phase. The instruction reading continues until the word mark of the next instruction is sensed. If characters without word marks follow an N Op code, these characters enter the A- and B-address registers.

#### I-Phase Operation (Figure 18)

During I-phase, a word mark in the B-register signals the end of I-phase and the beginning of E-phase. During certain I-cycles, the Op code is checked to see if any special operations must be performed before I-phase ends.

If the Op code is an N, circuitry is set up to eliminate the E-phase portion of the operations. Then the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-trigger and the associated console light that are on are turned off. The I-Op trigger and its associated console light are turned on and an I-Op cycle begins.

#### Compare (C AAA BBB)

The COMPARE instruction compares the data in the A-field to an equal number of characters in the B-field.

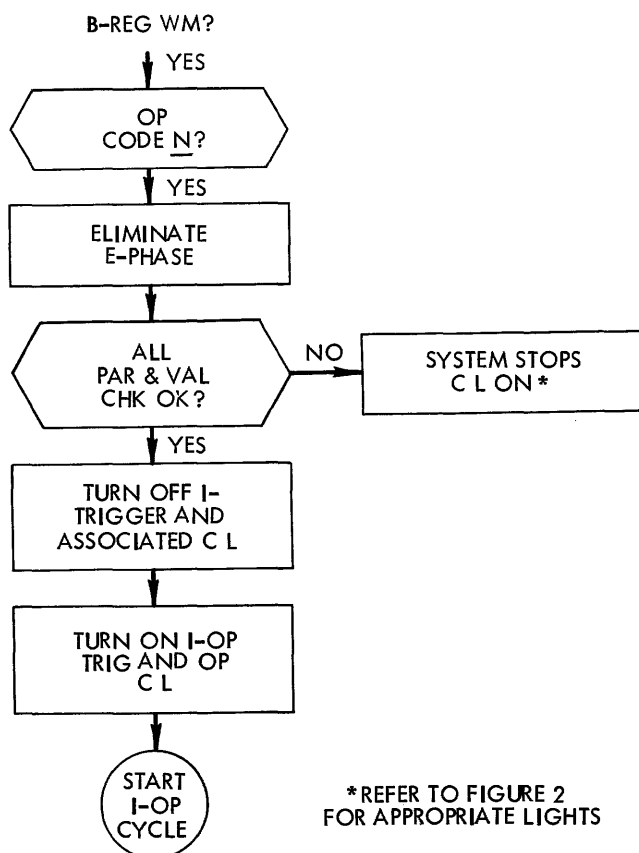


Figure 18. No Operation Instruction: I-Phase Operation

The bit configuration of each character in the two fields is compared and the comparison lets the equal-compare latch stay on, or it turns the latch off. The latch setting can be tested by a **BRANCH IF INDICATOR ON** instruction. This latch is initially set ON during I-2 time (not shown in Figure 6).

### I-Phase Set-Up Operations (Figure 19)

Some instructions require alterations of the normal E-phase operations for correct execution of the instruction. These alterations are set up before the I-phase ends. During certain I-cycles the Op code is checked to see if it is one of the Op codes that has to set up the alteration of the normal E-phase operations before the actual instruction execution begins. As soon as a word mark is sensed in the B-register, the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-phase ends and the E-phase begins. The first cycle executed during the E-phase is an A-cycle. The common A-cycle previously described (see Figure 13 and the accompanying text) is used during the A-cycle portion of the E-phase. As soon as the A-cycle is successfully completed, the B-cycle begins.

### B-Cycle (Figure 19)

The core-storage position containing the B-field character compared by the A-field character during this B-cycle was previously specified by the instruction, and the address was placed in the B-Add. Reg. during I-phase. The address in the B-Add. Reg. is transferred to the STAR and the B-Add. Reg. key-light on the console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.

The character in the B-register is read back into the core-storage position from which it came. The address in the B-Add. Reg. is decreased by one, and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the A-field character bit configuration is compared against the B-field character bit configuration.

If the character comparison results in an unequal condition, the equal-compare latch, which was turned on during the I-phase, is turned off. The latch remains on if the characters are equal.

The B-register is then checked to see if it contains a word mark. If a word mark is present, the E-phase ends and the I-phase for the next instruction begins.

If no word mark is present in the B-register, then the A-register is checked to see if it contains a word mark. If no word mark is present in the A-register, then the compare operation is continued and another A-cycle is started. If a word mark is present in the A-register, it signifies the presence of a B-field that is longer than the A-field and it results in an unequal condition. The equal-compare latch is turned off, followed by the ending of the E-phase and the starting of the I-phase for the next instruction.

### Halt (●)

### Halt and Branch (● III)

Both the **HALT** instruction (●) and the **HALT AND BRANCH** instruction (● III) cause a system stop, and turn on the stop key-light on the 1401 console panel.

If the instruction is a **HALT** instruction only, operating the start key causes the program to start at the next instruction in sequence.

If the instruction is a **HALT AND BRANCH** instruction, operating the start key causes the program to start at the specified I-address.

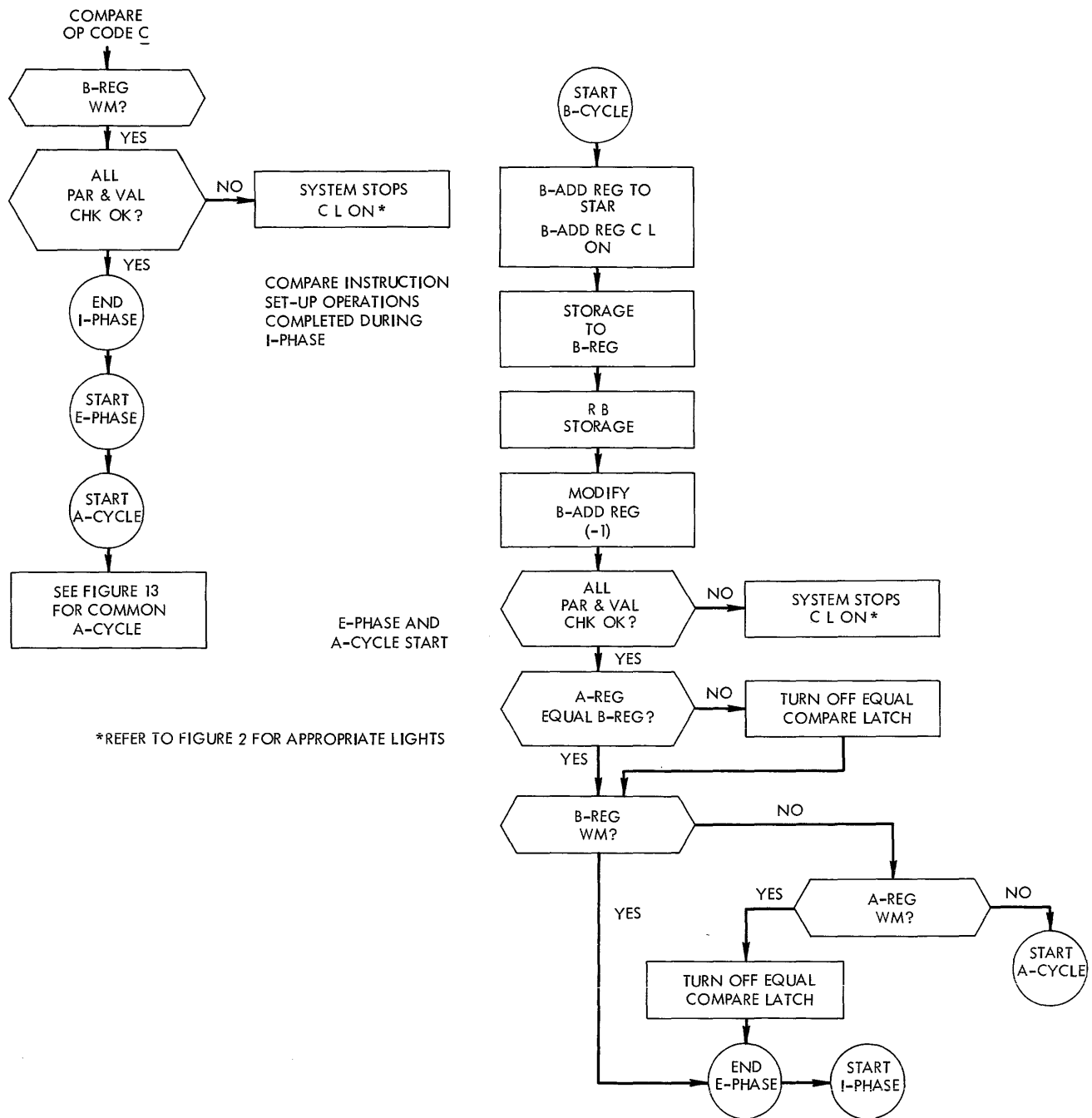


Figure 19. Compare Instruction: I- and E-Phase Operation

### I-Phase Operation (Figure 20)

During the I-phase cycle that has a word mark in the B-register, the Op-register is checked to see if it contains a decimal. When the presence of a decimal Op code is established, the circuitry that eliminates the E-phase portion of this instruction is activated.

The present I-phase cycle must be determined. If the I-phase cycle is I-4, circuitry is activated that blocks the I-Add. Reg. read-out, and allows the A-Add. Reg. read-out during the next I-Op cycle. (The A-Add. Reg. contains the I-address as specified by the instruction.)

If the I-phase cycle is not I-4, the normal I-Add. Reg. read-out is active during the next I-cycle.

The appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the circuitry that stops the system is activated and the stop key-light on the 1401 console panel turns on.

Operating the start key on the console panel turns off the stop key-light and turns on the I-Op trigger and its associated console light. Depending on the operation,

the core-storage position specified by either the I-address (in the A-Add. Reg.), or the I-Add. Reg. will be read out of core storage during this I-Op cycle.

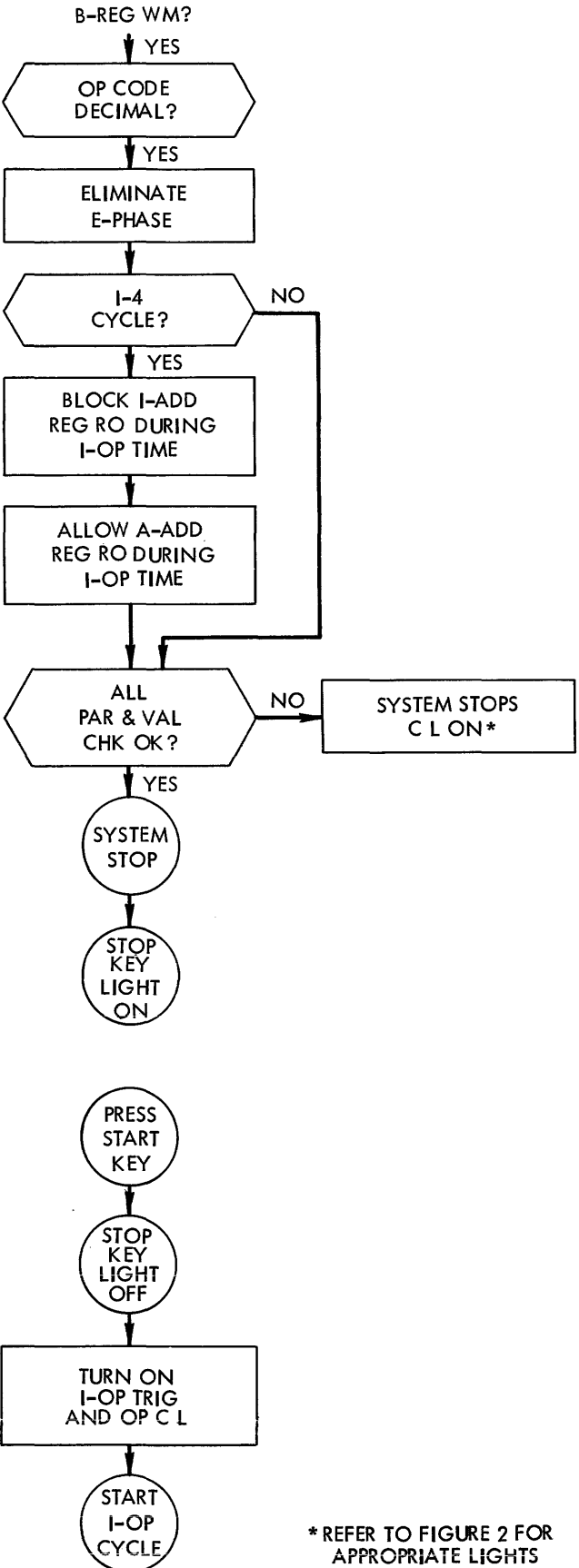


Figure 20. Halt and Branch Instructions: I-Phase Operation

## Arithmetic Operations

The add, subtract, zero and add, and zero and subtract operation codes are used to perform the system's arithmetic operations. The use of add-to-storage logic in the IBM 1401 eliminates the need for special-purpose accumulators or counters in the system. Because any group of storage positions can be used as an accumulating field, the capacity for arithmetic functions is not limited by a predetermined number of counter positions.

SIGN	BCD CODE BIT CONFIGURATION	CARD CODE CONFIGURATION
Plus Plus Minus Plus	No A- or B-bit A- and B-bits B-bit only A-bit only	No-Zone 12-Zone 11-Zone 0-Zone

Figure 21. Sign Bit Equivalents

All arithmetic operations are performed under complete algebraic sign control. Figure 21 shows the four possible combinations of zone bits and the values of the signs they represent. When the system signs a field, it is done in the standard form. A positive factor is indicated with an A- and a B-bit, and a negative factor is indicated with a B-bit. The sign of the resultant field is determined by the type of operation and the signs and values of the data fields as shown in Figure 22.

TYPE OF OPER.	A-FLD. SIGN	B-FLD. SIGN	TYPE OF ADD CYCLE	SIGN OF RESULT
A D +	+	+	True add	+
		-	Compl. add	Sign of greater value (Standard Form)
	-	+	Compl. add	
		-	True add	-
S U B T R A C T	+	-	True add	-
		+	Compl. add	Sign of greater value (Standard Form)
	-	-	Compl. add	
		+	True add	+

Figure 22. Types of Add Cycles and Sign of Result for Add and Subtract Operation

### Add (A AAA BBB) and Subtract (S AAA BBB)

The add and subtract operations in the IBM 1401 Data Processing System are performed by using one of the two types of add operations incorporated in the system:

1. true add
2. complement add

The type of add operation that will be taken is determined during the first E-phase B-cycle.

#### First A-Cycle

The first cycle during any E-phase is always an A-cycle. The A-Add. Reg. contains the core-storage location specified in the A-address portion of the previously-read instruction. In this instance, it is the units position of the A-field. The address in the A-Add. Reg. is transferred to the STAR, and the A-Add. Reg. key-light on the 1401 console panel turns on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form. The B-register contents are then read back into the core-storage position they came from, and are also transferred to the A-register. The address in the STAR is transferred back into the A-Add. Reg. without modification. The A-cycle ends and the B-cycle begins.

#### First B-Cycle (Figure 23)

The core-storage position that is the units position of the B-field was previously specified by the instruction, and the address was placed in the B-Add. Reg. during I-phase. The address in the B-Add. Reg. is transferred to the STAR and the B-Add. Reg. key-light on the console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.

The digit portion of the B-register contents is read back into the core-storage position from which it came. Also, a C-bit is added or removed to maintain odd-bit parity. The address in the STAR is transferred back into the B-Add. Reg. without modification, and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs the A-register sign is compared against the B-register sign. The result of this comparison, in conjunction with the specified operation, determines the type of add operation performed, and determines the initial sign of the result field.

#### EQUAL SIGNS

If the A-register sign is equal to the B-register sign, and the Op Code is A (add), then the B-register zone bits are read back into the core-storage position they came from, and a C-bit is added or removed to main-



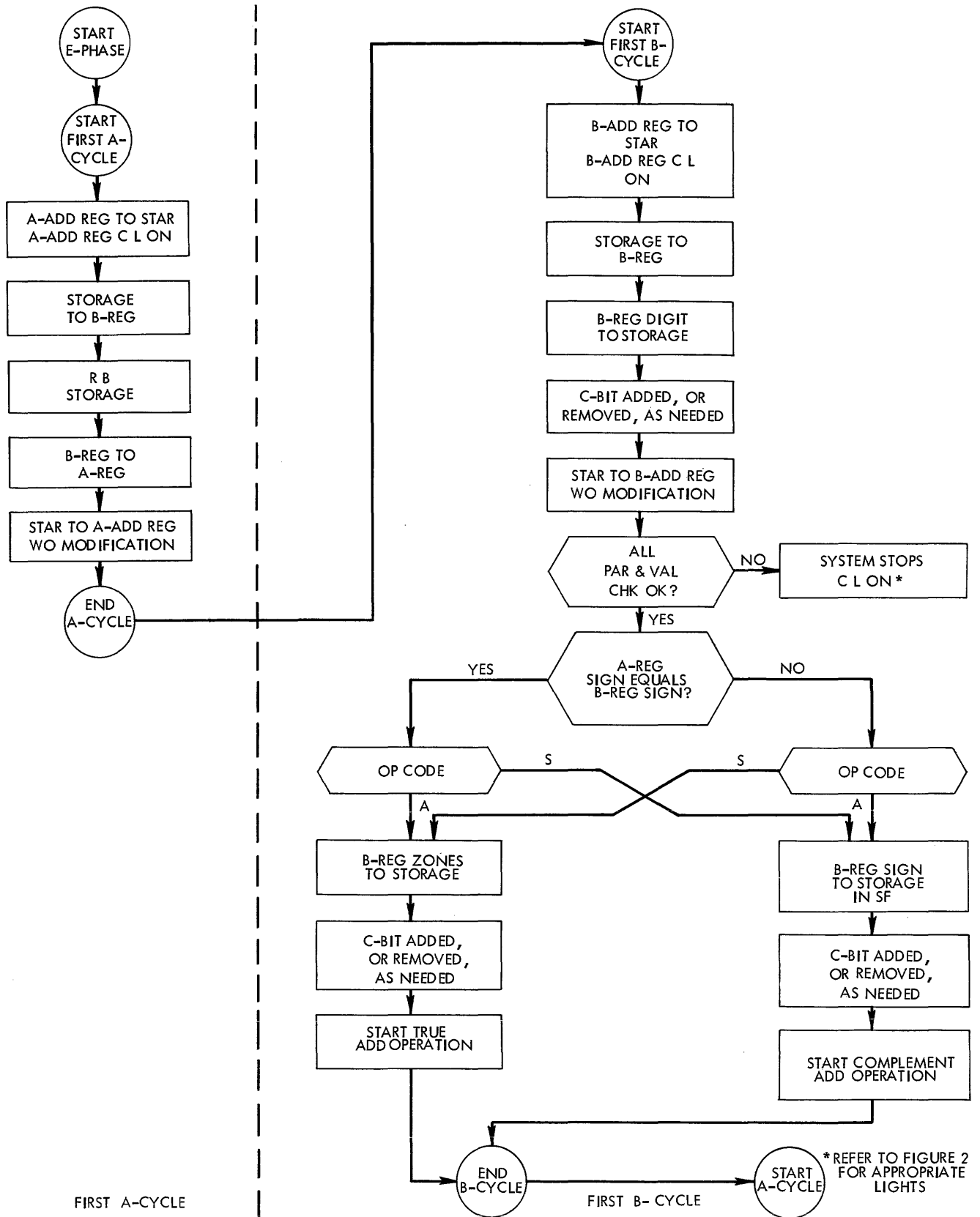


Figure 23. First A- and B-Cycle During an Add or Subtract Operation

tain odd-bit parity. In this instance, the sign returned to storage is the sign of the resultant B-field (refer to Figure 22). The circuitry used to execute true-add operations is also activated, and the first A-cycle of the true-add operation starts when the B-cycle ends.

If the A-register sign is equal to the B-register sign, and the Op code is S (subtract), then the B-register sign is read back in standard form into the core-storage position it came from and a C-bit is added or removed to maintain odd-bit parity. Depending on the value of the fields involved, this sign may, or may not, be the sign of the resultant B-field (refer to Figure 22). The circuitry used for executing complement-add operations is also activated, and the first A-cycle of the complement-add operation starts when the B-cycle ends.

#### UNEQUAL SIGNS

If the A-register sign is not equal to the B-register sign, and the Op code is S (subtract), then the B-register zone bits are read back into the core-storage position they came from, and a C-bit is added or removed to maintain odd-bit parity. In this instance, the sign returned to storage is the sign of the resultant B-field (refer to Figure 22). The circuitry used for executing true-add operations is also activated and the first A-cycle of the true-add operation starts when the B-cycle ends.

If the A-register sign is not equal to the B-register sign, and the Op code is A (add), then the B-register sign is read back in standard form into the core-storage position it came from and a C-bit is added or removed to maintain odd-bit parity. Depending on the value of the fields involved, this sign may, or may not, be the sign of the resultant B-field (refer to Figure 22). The circuitry used to execute complement-add operations is also activated, and the first A-cycle of the complement-add operation starts when the B-cycle ends.

No addition takes place during the first A- and B-cycles of an add operation (true or complement). These cycles are used to determine the type of add operation performed, and to activate the necessary circuitry.

#### True-Add Operation

The decision to execute a true-add operation is arrived at in the system by an A- and B-register sign comparison in conjunction with the specific operation (addition or subtraction) to be performed. This decision is made during the first B-cycle of an add or subtract operation. The various groups of conditions that can result in a true-add operation are shown in Figure 24. Notice that the original sign of the B-field is always the sign of the resultant B-field.

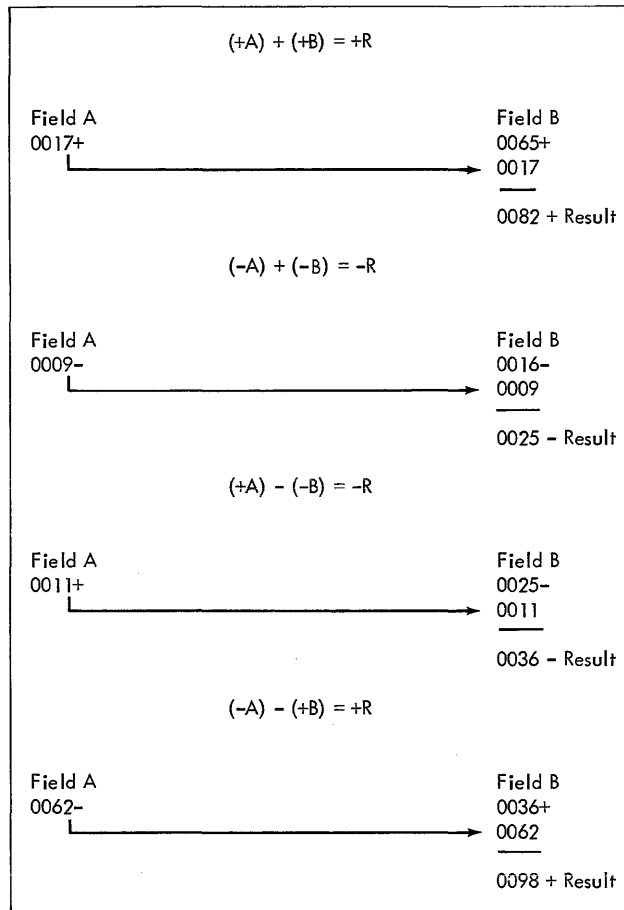


Figure 24. True-Add Cycle Examples

#### A-CYCLE

The common A-cycle previously described (see Figure 13 and the accompanying text) is used during the A-cycle portion of all true-add operations. As soon as the A-cycle is successfully completed, the B-cycle begins.

#### B-CYCLE (FIGURE 25)

The core-storage position that receives the result of the A- and B-field digit addition during the B-cycle was previously specified by the instruction, and the address was placed in the B-Add. Reg. during I-phase. This is the units position of the B-field, and is the same position previously read out during the first B-cycle when the type of add operation was decided. During that B-cycle, there was no B-Add. Reg. modification so the B-Add. Reg. still contains the original address (see Figure 23). (This was also true for the A-Add. Reg. during its first cycle.) The address in the B-Add. Reg. is transferred to the STAR, and the B-Add. Reg. key-light on the 1401 console panel turns on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).

The operation is checked to see if this B-cycle is the first true-add B-cycle. If it is the first true-add B-cycle,

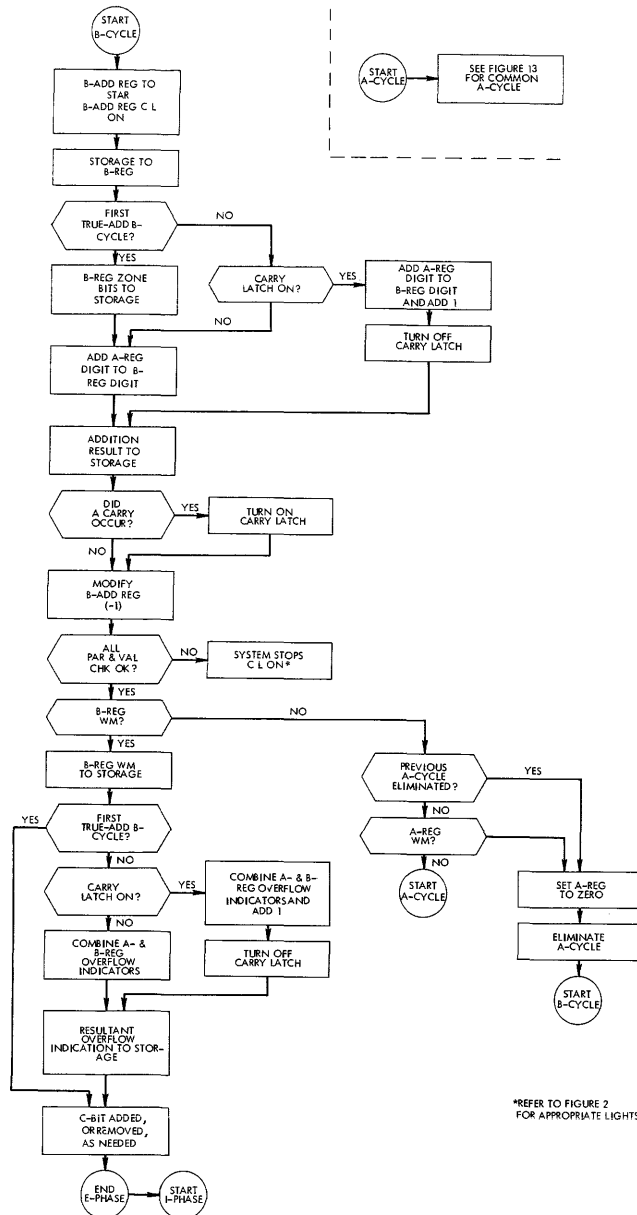


Figure 25. True-Add B-Cycle Operation

the zone portion of the B-register is read back into the core-storage position from which it came. The digit portion of the A-register is added to the digit portion of the B-register next, and then the addition result is read back into the specified core-storage location (in this case, the units position of the B-field).

If this B-cycle is not the first true-add B-cycle, then the carry latch is checked to see if it was turned on during the previous B-cycle. If the carry latch is not on, then the digit portion of the A-register is added to the digit portion of the B-register, and the addition result is read back into the specified core-storage location. If the carry latch is on, then the digit portion of the A-register is added to the digit portion of the B-register and this result is increased by one (the carry). The carry

latch is turned off, and the addition result is then read back into the specified core-storage location.

A check is made to see if a carry resulted from this addition. If no carry occurred, the address in the B-Add. Reg. is decreased by one. If a carry occurred, then the carry latch is turned on and the B-Add. Reg. is decreased by one.

The appropriate parity and validity checks are made next. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the B-register is checked to see if it contains a word mark.

If the B-register contains a word mark, it is transferred to the specified core-storage position. A check is made to see if this B-cycle is the first true-add B-cycle. If it is the first true-add B-cycle, then a C-bit is added or removed to maintain odd-bit parity in that core-storage position. Then, the E-phase ends and the I-phase of the next instruction begins. If this B-cycle is not the first true-add B-cycle, then the carry latch is checked to see if it is on. If the carry is not on, the overflow indications (zone bits) in the A-register are added to the overflow indications (zone bits) in the B-register. The resultant addition of overflow indications are transferred into the specified core-storage location, and a C-bit is added or removed to maintain odd-bit parity in that core-storage position. (Refer to Figure 26 for zone-bit configurations of overflow indications.) The E-phase ends and the I-phase of the next instruction begins.

OVERFLOW INDICATION	ZONE-BIT CONFIGURATION
First overflow	A-bit
Second overflow	B-bit
Third overflow	A- and B-bits
Fourth overflow	No A- or B-bits
Overflow Indications 1-4 Repeated for Subsequent Overflows	

Figure 26. Zone-Bit Configuration of Overflow Indications

If the carry latch is on, the overflow indications in the A-register are added to the overflow indications in the B-register, and this result is increased by one (the carry). The carry latch is turned off and the resultant addition of overflow indications are transferred into the specified core-storage location. A C-bit is added or removed to maintain odd-bit parity in that core-storage position. The E-phase ends and the I-phase of the next instruction begins.

If the B-register does not contain a word mark, then a check is made to see if the previous A-cycle was eliminated. The A-cycle would have been eliminated if an A-field word mark had been encountered before a B-field word mark. (If the A-field is the same length as the B-field, an A-field word mark is not needed.) If the pre-

vious A-cycle was eliminated, then a zero is generated and transferred into the A-register. Then, the circuitry is set up to eliminate the next A-cycle and another B-cycle starts.

If the previous A-cycle was not eliminated, then the A-register is checked to see if it contains a word mark. If the A-register does not contain a word mark, then the next A-cycle begins. If the A-register does contain a word mark, it signifies that the end of the A-field has been reached before the end of the B-field. A zero is generated and transferred into the A-register. Then, the circuitry is set up to eliminate the next A-cycle and another B-cycle starts.

### Complement-Add Operation

The decision to execute a complement-add operation is arrived at in the system by an A- and B-register sign comparison in conjunction with the specific operation (addition or subtraction) to be performed. This decision

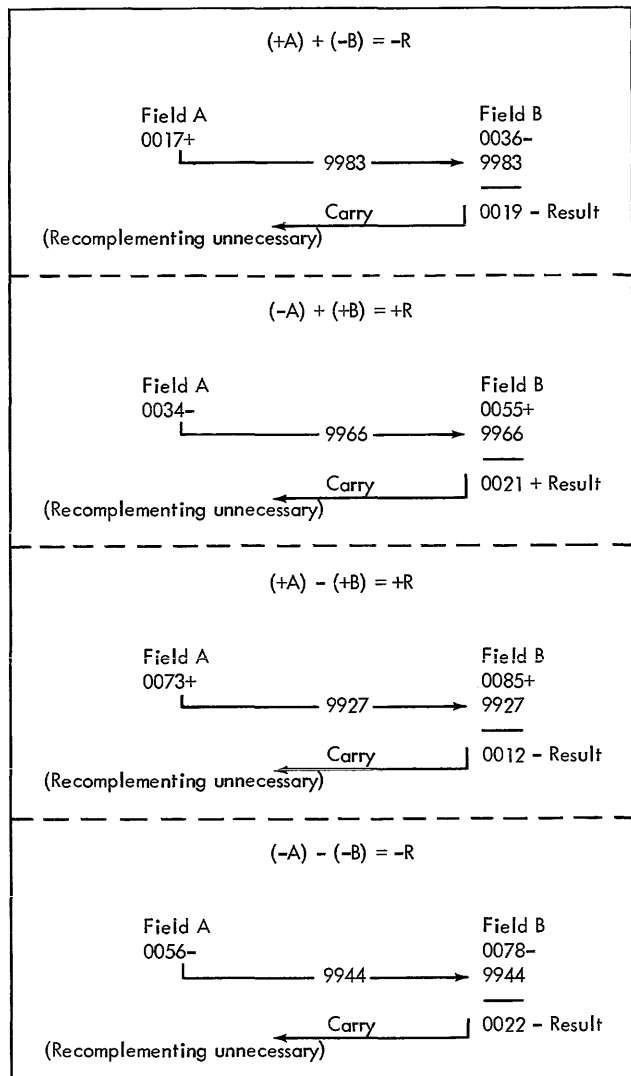


Figure 27. Complement-Add Cycle Examples

is made during the first B-cycle of an add or subtract operation. The various groups of conditions that can result in a complement-add operation are shown in Figures 27 and 28.

The system converts the A-field factor to its nines complement figure and adds it to the B-field factor (plus one initial carry). When the addition is over, the system initiates a carry test to determine whether a carry occurred from the high-order position of the B-field. The presence of a carry indicates that the result in the B-field is a true figure (Figure 27). The original sign of the B-field is the sign of the result (in standard form).

If there was no carry from the high-order position of the B-field, then the result in the B-field is not a true figure (Figure 28). A re-complement cycle is performed

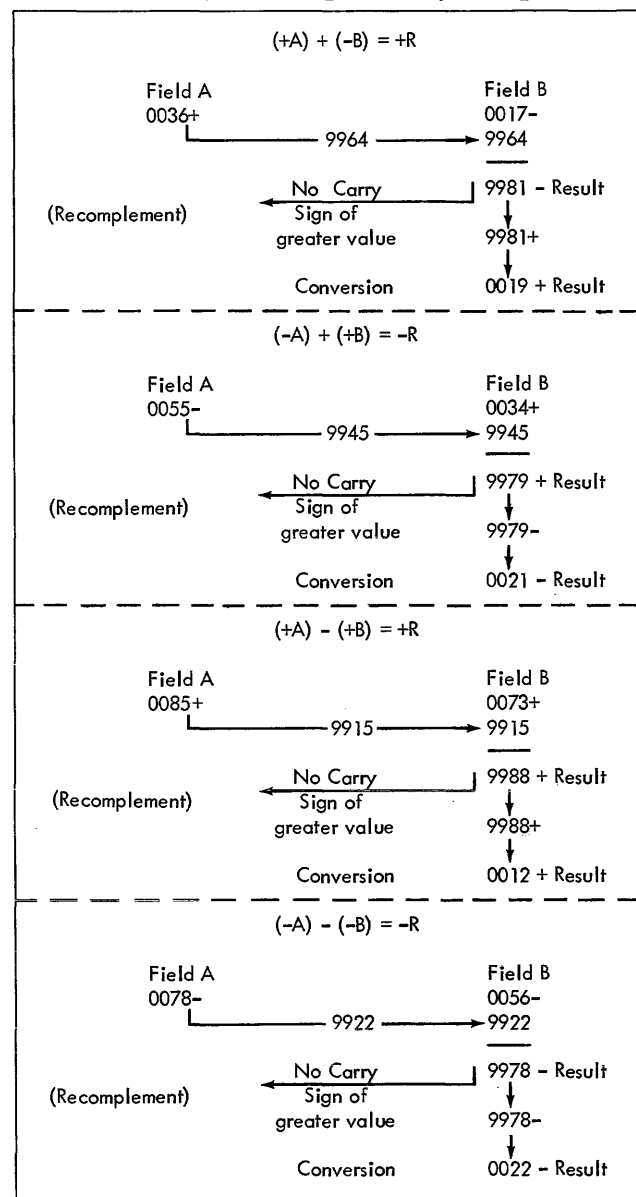


Figure 28. Complement-Add (with Re-complementing) Cycle Examples

to convert the result to a true figure. In an add or subtract operation that results in a negative figure, the sign of the result is also changed during the recomplement operation. The system generates this new sign automatically.

#### A-CYCLE

The common A-cycle previously described (see Figure 13 and the accompanying text) is used during the A-cycle portion of all complement-add operations. As soon as the A-cycle is successfully completed, the B-cycle begins.

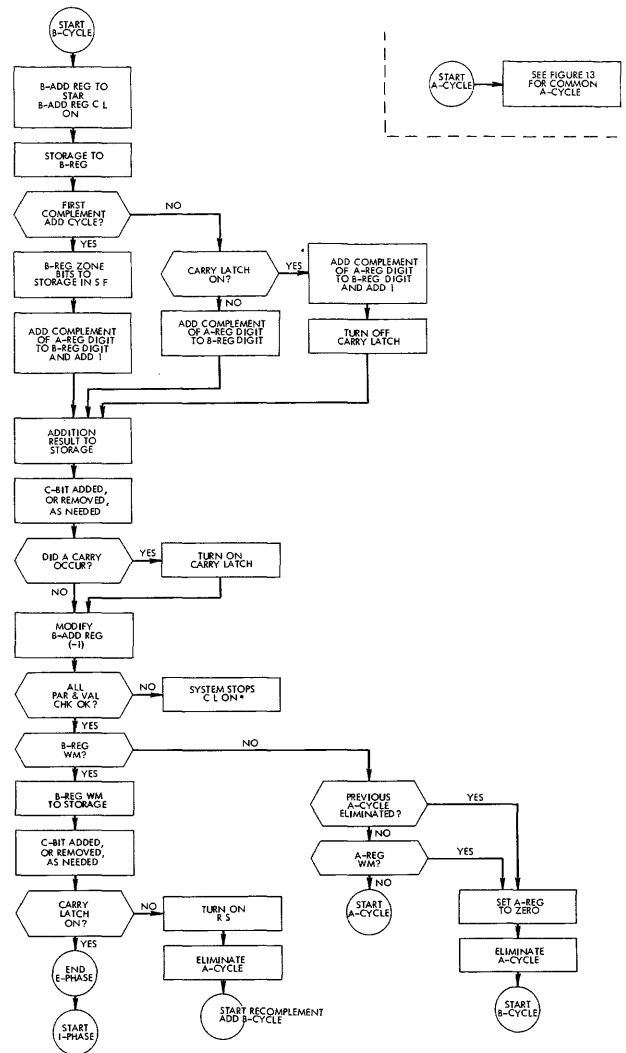
#### B-CYCLE (FIGURE 29)

The core-storage position that receives the result of the A- and B-field digit addition during the B-cycle was previously specified by the instruction, and the address was placed in the B-Add. Reg. during I-phase. This is the units position of the B-field and is the same position that was previously read out during the first B-cycle when the type of add operation was decided. During that B-cycle, there was no B-Add. Reg. modification, so the B-Add. Reg. still contains the original address (see Figure 23). (This was also true for the A-Add. Reg. during its first cycle.) The address in the B-Add. Reg. is transferred to the STAR, and the B-Add. Reg. key-light on the 1401 console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).

The operation is checked to see if this B-cycle is the first complement-add B-cycle. If it is the first complement-add B-cycle, the zone portion of the B-register is read back into the core-storage position from which it came (in standard form). The digit portion of the A-register is complement added to the digit portion of the B-register, and this result is increased by one (initial carry). The addition result is then read back into the specified core-storage position (in this case, the units position of the B-field).

If this B-cycle is not the first complement-add B-cycle, then the carry latch is checked to see if it was turned on during the previous B-cycle. If the carry latch is not on, then the digit portion of the A-register is complement added to the digit portion of the B-register, and the addition result is read back into the specified core-storage position. If the carry latch is on, then the digit portion of the A-register is complement added to the digit portion of the B-register and this result is increased by one (the carry). The carry latch is turned off, and the addition result is read back into the specified core-storage position.

A C-bit is added or removed to maintain odd-bit parity in the core-storage position that received the addition result.



\*REFER TO FIGURE 2 FOR APPROPRIATE LIGHTS

Figure 29. Complement Add B-Cycle Operation

A check is made to see if a carry resulted from this addition. If a carry occurred, then the carry latch is turned on. The address in the B-Add. Reg. is decreased by one.

The appropriate parity and validity checks are made next. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the B-register is checked to see if it contains a word mark.

If the B-register contains a word mark, it is transferred to the specified core-storage position, and a C-bit is added or removed to maintain odd-bit parity in that core-storage position. The carry latch is checked to see if it is on. If it is on, it indicates that the result in the B-field is a true figure. The E-phase ends and the I-phase of the next instruction begins. If the carry latch is not on, it indicates that the result in the B-field is *not* a true figure. The reverse-scan circuitry and the cir-

circuitry to eliminate A-cycles are turned on, and a re-complement-add B-cycle begins. If the B-register does not contain a word mark, then a check is made to see if the previous A-cycle was eliminated. The A-cycle would have been eliminated if an A-field word mark had been encountered before a B-field word mark. (If the A-field is the same length as the B-field, an A-field word mark is not needed.) If the previous A-cycle was eliminated, then a zero is generated and transferred into the A-register. Then, the circuitry is set up to eliminate the next A-cycle and another B-cycle starts.

If the previous A-cycle was not eliminated, then the A-register is checked to see if it contains a word mark. If the A-register does not contain a word mark, then the next A-cycle begins. If the A-register does contain a word mark, it signifies that the end of the A-field has been reached before the end of the B-field. A zero is generated and transferred into the A-register. Then the circuitry is set up to eliminate the next A-cycle and another B-cycle starts.

### Recomplement Operation

At the end of a complement-add operation, a no-carry indication from the high-order position of the B-field indicates that the B-field figure is not a true figure and a recomplement operation must be performed. During the recomplement operation, the B-field sign is also reversed.

The conditions used to signal a recomplement operation are shown in Figure 28. The circuitry during a reverse-scan operation is activated and the A-cycles are eliminated.

#### REVERSE-SCAN OPERATION (FIGURE 30)

The reverse-scan operation is made up of a series of repetitive B-cycles that continue until the B-field units position is reached. The first reverse-scan B-cycle during a recomplement operation reads out the same core-storage position that was involved in the last complement-add B-cycle. The address of this core-storage position is no longer in the B-Add. Reg., because the B-Add. Reg. address was already modified. The STAR still does have this address, which is used to activate the lines that cause the specified core-storage position to read out to the B-register, where the contents are displayed in BCD form. During all other reverse scan cycles, the B-Add. Reg. address is transferred to the STAR, and the B-Add. Reg. key-light on the console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form). Then the STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.

The B-register is checked to see if it contains a B-bit. The sign in the units position of the B-field always contains a B-bit because the B-field sign was stored in its standard form during a complement-add operation. Because the zone bits were removed from every B-field position, except the units position, the B-bit can be used to signal the end of a reverse-scan operation.

If the B-register does not contain a B-bit (this signifies that the B-field units position has not been reached), the character in the B-register is read back into the core-storage position from which it came. The address in the B-Add. Reg. is increased by one and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the circuitry to eliminate A-cycles is activated and another B-cycle begins.

When a B-bit is found in the B-register, the B-register character is read into the A-register, but not back into the core-storage position from which it came. A zero is generated instead, and it replaces the B-register character in the specified core-storage position. The A-register input is then blocked so that it cannot accept a character during the next cycle.

The reverse-scan circuitry is turned off, and the carry latch is turned on because an additional *one* must be added during the first recomplement-add (forward-scan) cycle to obtain the correct total. Sign control is turned on, and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the circuitry that eliminates A-cycles is activated and another B-cycle begins.

This B-cycle operation during the last reverse-scan cycle is actually the first B-cycle of the two B-cycles needed to recomplement the units position of the field. The B-field recomplementing is done one position at a time, and each position requires two B-cycles. During the first recomplement B-cycle, the core-storage position is read out to the A- and B-registers. The B-register output to core storage is blocked and a zero is inserted in the specified core-storage position. During the second recomplement B-cycle, the zero is read back into the B-register and the A-register digit is complement added to the B-register digit. The resultant addition is stored in the specified core-storage position.

#### FIRST FORWARD SCAN B-CYCLE (FIGURE 30)

The first forward-scan cycle (second recomplement cycle for the B-field units position) reads out the same core-storage position that was involved in the last reverse-scan B-cycle (the B-field units position). The address of this core-storage position is no longer in the B-Add. Reg., because the B-Add. Reg. address was al-

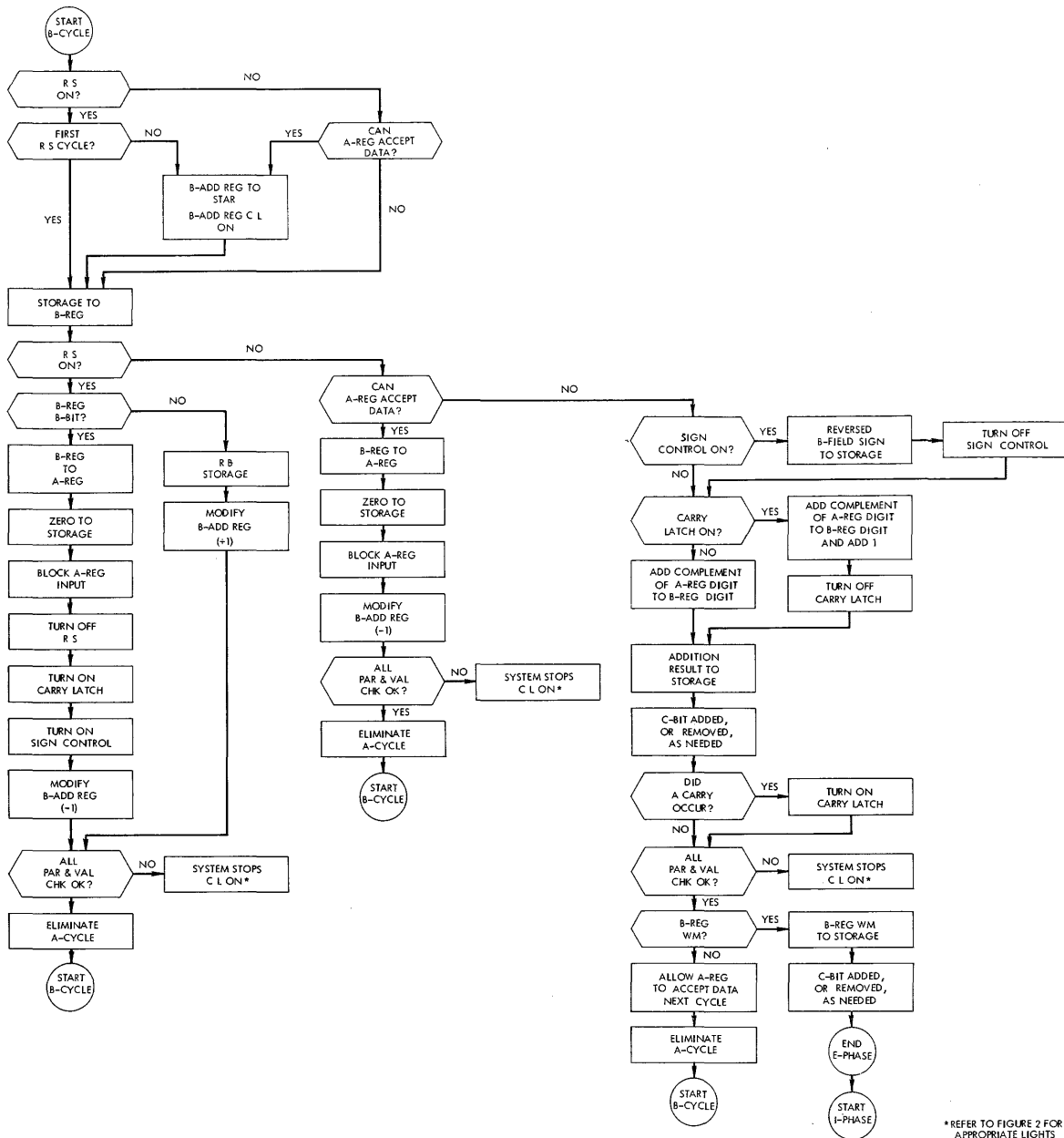


Figure 30. Recomplement B-Cycles Operation

ready modified. The STAR still does have this address, so it is used to activate the lines that cause the specified core-storage position to read out to the B-register, where the contents are displayed in BCD form (a zero). This is possible, as shown in Figure 30, because this B-cycle is not a reverse-scan cycle, and the A-register cannot accept data.

Sign control is checked to see if it is on. It is on when the units position of the field is addressed. It is off during the rest of the operation. When it is on, it activates the circuitry that reverses the B-register sign, and sends it back to the specified core-storage position. Then the sign control is turned off.

The carry latch is checked next to see if it is on. It is

always on when the units position of the field is addressed. At other times, it may be on or off, depending on the previous addition result. If the carry latch is on, the digit portion of the A-register is complement added to the digit portion of the B-register, and this result is increased by one (the carry). The carry latch is turned off and the addition result is read back into the specified core-storage position. If the carry latch is not on, the digit portion of the A-register is complement added to the digit portion of the B-register, and the addition result is read back into the specified core-storage position.

A C-bit is added or removed to maintain odd-bit parity in the core-storage position that received the addition result.

A check is made to see if a carry resulted from this addition. If a carry did occur, the carry latch is set on and the appropriate parity and validity checks are made. If no carry occurred, the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the B-register is checked to see if it contains a word mark.

If the B-register contains a word mark, it is transferred to the specified core-storage position, and a C-bit is added or removed to maintain odd-bit parity in that core-storage position. The E-phase ends and the I-phase of the next instruction begins.

If the B-register does not contain a word mark, the A-register is conditioned so that it will be able to accept a character during the next cycle. This will be a B-cycle, as all recomplement forward-scan B-cycles are, because the circuitry to eliminate A-cycles is activated at this time.

#### OTHER FORWARD SCAN B-CYCLES

The two recomplement B-cycles per field position forward-scan operation continue until a B-register word mark is sensed during the second recomplement B-cycle operation on that particular field position.

#### 1-Field Operation

Both the add and the subtract operation can be performed with only the A-field specified.

#### ADD OPERATION (A AAA)

An add operation with only the A-field specified (A AAA) adds the A-field data to itself. The add operation is always a true-add operation, and the operation results in an amount that is double the original A-field amount. The resultant A-field sign is the same as the original A-field sign and has the same form. Any overflow bits present are also affected, and the resultant A-field overflow indication is double the original indication.

The effective instruction is A AAA AAA, and the operation is basically the same as a A AAA BBB instruction. Alternate A- and B-cycles are used to accomplish the operation, and each A-field position is addressed twice because the A-field address is in both the A-Add. Reg. and the B-Add. Reg. The high-order position of the A-field must have a word mark.

#### SUBTRACT OPERATION (S AAA)

A subtract operation with only the A-field specified (S AAA) subtracts the A-field from itself. The subtract operation is always a complement-add (no recomplement) operation, and the operation results in an A-field with an amount of zero. The resultant A-field sign is the same as the original A-field sign, but it is in standard

form. Zone bits in all the A-field positions, except the units position, are dropped.

The effective instruction is S AAA AAA and the operation is basically the same as a S AAA BBB instruction. Alternate A- and B-cycles are used to accomplish the operation, and each A-field position is addressed twice since the A-field address is in both the A-Add. Reg. and the B-Add. Reg. The high-order position of the A-field must have a word mark.

#### Zero and Add (? AAA BBB) and Zero and Subtract (! AAA BBB)

Both the ZERO AND ADD instruction and the ZERO AND SUBTRACT instruction replace the data in the B-field with the data from the A-field. The data from the A-field moves directly from the A-register into storage. Zone bits are stripped from all positions except the units position, and blanks in the A-field are stored as blanks in the B-field. The only difference between the two instructions is the treatment of the sign.

#### A-Cycle

The common A-cycle previously described (see Figure 13 and the accompanying text) is used during the A-cycle portion of the E-phase. As soon as the A-cycle is successfully completed, the B-cycle begins.

#### B-Cycle (Figure 31)

The core-storage position that receives the A-register digit during the B-cycle was previously specified by the instruction, and the address was placed in the B-Add. Reg. during I-phase. The address in the B-Add. Reg. is transferred to the STAR, and the B-Add. Reg. key-light on the 1401 console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register when the contents of that position are displayed in BCD form.

The A-register digit contents replace the B-register contents in the specified core-storage position. If this is *not* the first B-cycle, and a C-bit is needed to obtain odd-bit parity, the C-bit is generated and placed in the same core-storage position.

If this B-cycle is the first B-cycle, then the operation code is checked to see if it is a ? (ZERO AND ADD Op code) or a ! (ZERO AND SUBTRACT Op code). If it is a ZERO AND ADD Op code, the A-register sign is transferred to the specified core-storage position in its standard form. If it is a ZERO AND SUBTRACT Op code, the reverse of the A-register sign is transferred to the specified core-storage position in its standard form. If a C-bit is



needed to obtain odd-bit parity, it is also generated and placed in the same core-storage position.

The address in the B-Add. Reg. is decreased by one, and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the B-register is checked to see if it contains a word mark.

If the B-register contains a word mark, it is transferred to the specified core-storage position and a C-bit is added or removed to maintain odd-bit parity. The

previous A-cycle was eliminated, a zero is generated and transferred into the A-register. Then, the circuitry is set up to eliminate the next A-cycle and another B-cycle starts.

If the previous A-cycle was not eliminated, the A-register is checked to see if it contains a word mark. If the A-register does not contain a word mark, the next A-cycle begins. If the A-register does contain a word mark, it signifies that the end of the A-field has been reached before the end of the B-field. A zero is generated and transferred into the A-register. This zero replaces the B-register contents in core-storage during the next B-cycle. Circuitry is set up to eliminate the next A-cycle and another B-cycle begins.

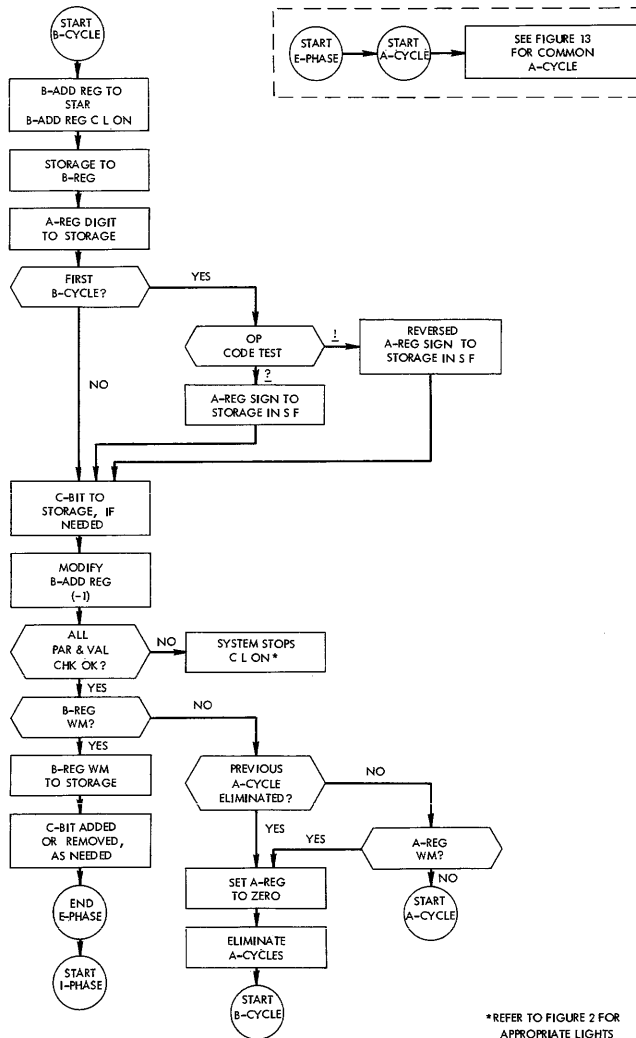


Figure 31. Zero and Add and Zero and Subtract Instructions: E-Phase Operation

B-register word mark also terminates the operation by ending the E-phase and starting the I-phase of the next instruction.

If the B-register does not contain a word mark, then a check is made to see if the previous A-cycle was eliminated. The A-cycle would have been eliminated if an A-field word mark had been encountered before a B-field word mark. (If the A-field is the same length as the B-field, an A-field word mark is not needed.) If the

### I-Field Operation

Both the zero and add and the zero and subtract operations can be performed with only the A-address specified. However, during I-phase, the A-address enters both the A-Add. Reg. and the B-Add. Reg.

A zero and add operation, with only the A-address specified, can accomplish two things:

1. The plus sign in its standard form can be inserted in the units position of the field.
2. Zone bits are stripped from every position in the field, except the units position.

Both of these functions, however, can usually be accomplished in other ways in less processing time.

A zero and subtract operation, with only the A-address specified, can accomplish two things:

1. The sign of the specified field is changed.
2. Zone bits are stripped from every position in the field, except the units position.

The second function, however, can usually be accomplished in other ways in less processing time.

## Clear, Move, Load, and Word Mark Operations

### Clear Storage (/ AAA)

The CLEAR STORAGE instruction clears all data and all word marks from a specified core-storage area. The high-numbered core-storage position of the cleared area is the position specified by the A-address in the CLEAR STORAGE instruction. The low-numbered core-storage position of the cleared area is the nearest hundreds position *below* the core-storage position specified by the A-address. For example, if the specified core-storage position is 691, then positions 691 *down* through 600 are cleared.

### I-Phase Set-Up Operations (Figure 32)

Some instructions require alterations of the normal E-

phase operations for correct execution of the instruction. These alterations are set up before the I-phase ends. During certain I-cycles the Op code is checked to see if it is one of the Op codes that has to set up the alteration of the normal E-phase operations before the actual instruction execution begins. During I-4 time, a CLEAR STORAGE Op code and a B-Reg. word mark set up circuitry that eliminates all A-cycles during the next E-phase. The appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the I-phase ends and the E-phase begins.

### B-Cycle (Figure 32)

The execution of a CLEAR STORAGE instruction consists of successive B-cycles until the operation is ended. The B-Add. Reg. received the A-address during the I-phase portion of the operation. The A-Add. Reg. also contains

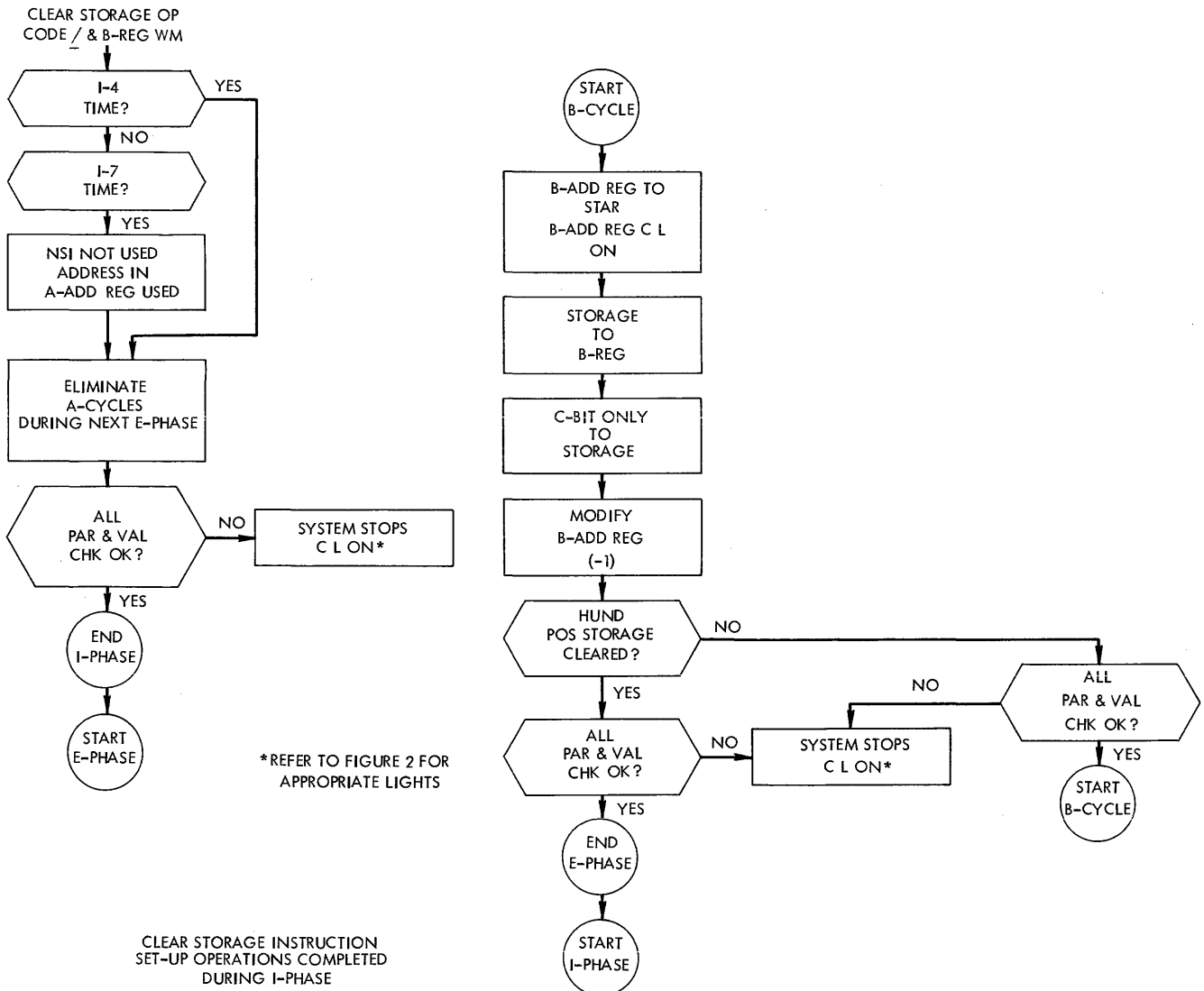


Figure 32. Clear Storage and Clear Storage and Branch Instructions: I- and E-Phase Operations

the A-address, but the A-Add. Reg. contents are ignored. (A-Add. Reg. contents are read out only during A-cycles.)

The address in the B-Add. Reg. is transferred to the STAR, and the B-Add. Reg. key-light on the console panel displays the core-storage position being addressed (in BCD form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.

The clear storage operation does not let any of the B-register contents read back into core storage however. This results in a completely blank core-storage position. A C-bit is generated and placed in the blank core-storage position so that the position has odd-bit parity.

The address in the B-Add. Reg. is decreased by one, and a check is made to see if the core-storage position just cleared was the hundreds position of the core-storage block being cleared.

#### CONTINUE OPERATION

If the hundreds position of the core-storage block being cleared has not been reached, the appropriate parity and validity checks are made before another B-cycle is started. Any check condition, except a B-Reg. parity-check condition, stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, another B-cycle is started.

#### TERMINATE OPERATION

If the hundreds position of the core-storage block being cleared has been reached, the appropriate parity and validity checks are made before the operation is ended. Any check condition, except a B-Reg. parity-check condition, stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the E-phase ends and the I-phase for the next instruction begins.

### **Clear Storage and Branch (/ III BBB)**

The CLEAR STORAGE AND BRANCH instruction clears all data and all word marks from a specified core-storage area in the same manner as a CLEAR STORAGE instruction. The difference between the two instructions is that the CLEAR STORAGE AND BRANCH instruction sets up circuitry so that the address of the next instruction is taken from the A-Add. Reg. instead of the I-Add. Reg. During I-7 time, a CLEAR STORAGE Op code and a B-Reg. word mark set up the circuitry needed to read the address in the A-Add. Reg. into the STAR, instead of the address in the I-Add. Reg. The actual transfer takes place during the I-phase that follows the CLEAR STORAGE

AND BRANCH E-phase. Once the A-Add. Reg. address is in the STAR, the address is modified by +1 and transferred into the I-Add. Reg. The remaining I-cycle addressing is done in the normal way.

At I-7 time, the CLEAR STORAGE Op code and B-Reg. word mark also set up the circuitry that ends the I-phase, starts the E-phase, and eliminates all A-cycles during the E-phase.

### **Set Word Mark (, AAA BBB) (, AAA)**

The SET WORD MARK instruction sets word marks in the core-storage positions specified in the instruction. The data already in these positions is undisturbed.

#### **A-Cycle (Figure 33)**

The core-storage position that receives the word mark during the A-cycle was previously specified by the instruction, and the address was placed in the A-Add. Reg. during I-phase. The address in the A-Add. Reg. is transferred to the STAR, and the A-Add. Reg. key-light on the 1401 console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.

An Op-code test establishes the operation as a set word mark operation. A set word mark operation allows any A-, B-, 1-, 2-, 4-, or 8-bit in the B-register to read back into the core-storage position from which they came. (The B-register contents also go to the A-register, but serve no purpose during this operation.) A word mark is also generated and placed in the core-storage position with the bits just read back.

If a C-bit is needed to obtain odd-bit parity, it is also generated and placed in the same core-storage position.

The address in the A-Add. Reg. is decreased by one, and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the A-cycle ends and the B-cycle begins. (See Figure 34 and the accompanying text for the B-cycle operation.)

### **Clear Word Mark (≡ AAA BBB) (≡ AAA)**

The CLEAR WORD MARK instruction clears the word marks from the core-storage position specified in the instruction. The data already in these positions is undisturbed.

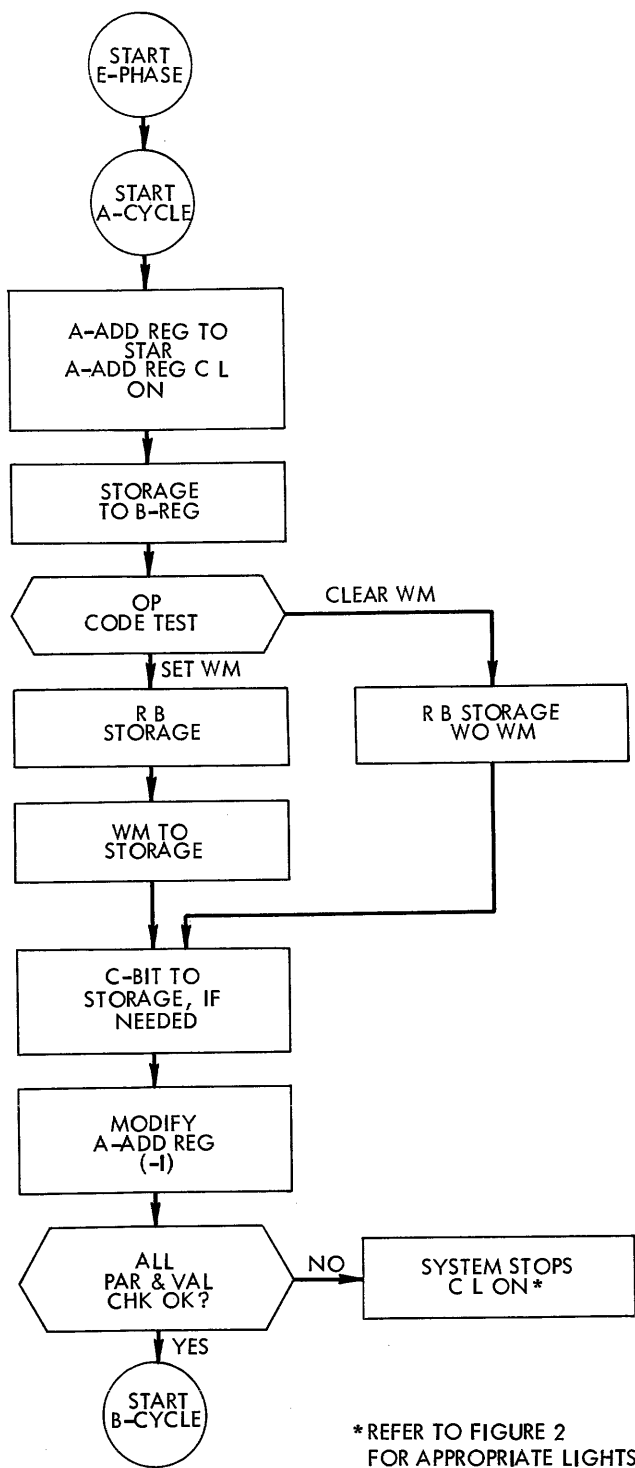


Figure 33. Set Word Mark and Clear Word Mark Instructions: A-Cycle Operation

**A-Cycle (See Figure 33)**

The core-storage position that loses its word mark during A-cycle was previously specified by the instruction, and the address was placed in the A-Add. Reg. during I-phase. The address in the A-Add. Reg. is transferred to the STAR and the A-Add. Reg. key-light on the 1401

console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.

An Op code test establishes the operation as a clear word mark operation. A clear word mark operation allows any A-, B-, 1-, 2-, 4-, or 8-bit in the B-register to read back into the core-storage position from which they came. In this manner, the word mark, if previously present, is eliminated.

If a C-bit is needed to obtain odd-bit parity, it is also generated and placed in the core-storage position with the bits just read back.

The address in the A-Add. Reg. is decreased by one and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the A-cycle ends and the B-cycle begins. (See Figure 34 and the accompanying text for the B-cycle operation.)

**Set Word Mark B-Cycle (See Figure 34)**

**TWO FIELDS**

The two core-storage positions that receive the word marks were previously specified by the instruction, and their addresses were placed in the A-Add. Reg. and the B-Add. Reg. during I-phase. The address in the A-Add. Reg. was used during the A-cycle. During the B-cycle, the B-Add. Reg. specifies the core-storage position that will receive the word mark. The address in the B-Add. Reg. is transferred to the STAR and the B-Add. Reg. key-light on the 1401 console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.

The Op code test done during the A-cycle established the operation as a set word mark operation. A set word mark operation allows any A-, B-, 1-, 2-, 4-, or 8-bit in the B-register to read back into the core-storage position from which they came. A word mark is also generated and placed in the core-storage position with the bits just read back.

If a C-bit is needed to obtain odd-bit parity, it is also generated and placed in the same core-storage position.

The address in the B-Add. Reg. is decreased by one and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel.

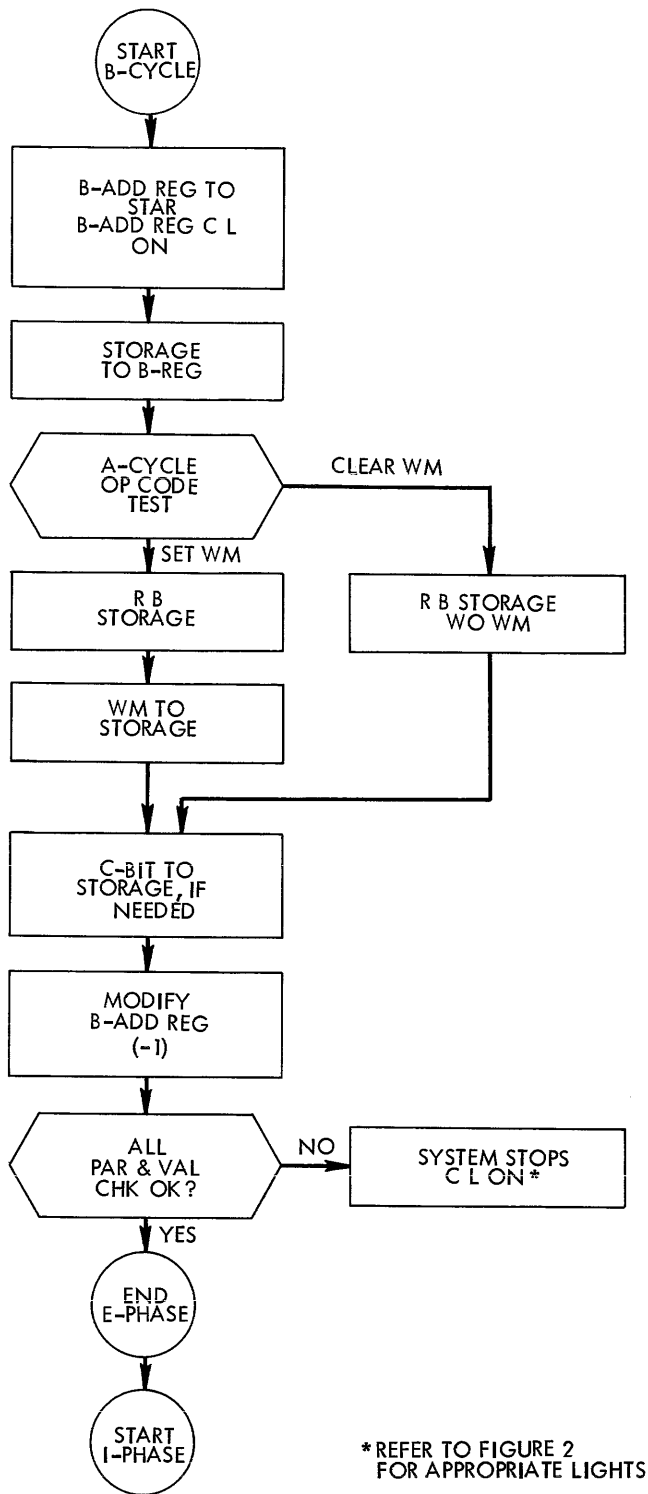


Figure 34. Set Word Mark and Clear Word Mark Instructions: B-Cycle Operation

If no check condition occurs, the E-phase ends and the I-phase of the next instruction begins.

#### ONE FIELD

The core-storage position that receives the word mark was previously specified by the instruction, and the ad-

dress was placed in *both the A-Add. Reg. and the B-Add. Reg. during the I-phase*. The B-cycle occurs, and the core-storage position that received the word mark during the A-cycle receives a second word mark during the B-cycle. The B-cycle operation for a 1-field instruction is exactly the same as for a 2-field instruction.

#### Clear Word Mark B-Cycle (See Figure 34)

##### 2-FIELD OPERATION

The two core-storage positions that lose their word marks were previously specified by the instruction, and their addresses were placed in the A-Add. Reg. and the B-Add. Reg. during I-phase. The address in the A-Add. Reg. was used during the A-cycle. During the B-cycle, the B-Add. Reg. specifies the core-storage position that loses its word mark. The address in the B-Add. Reg. is transferred to the STAR and the B-Add. Reg. key-light on the 1401 console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.

The Op-code test, done during the A-cycle, established the operation as a clear word mark operation. A clear word mark operation allows any A-, B-, 1-, 2-, 4-, or 8-bit in the B-register to read back into the core-storage position from which they came. In this manner, the word mark, if previously present, is eliminated.

If a C-bit is needed to obtain odd-bit parity, it is also generated and placed in the core-storage position with the bits just read back.

The address in the B-Add. Reg. is decreased by one and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the E-phase ends and the I-phase of the next instruction begins.

##### 1-FIELD OPERATION

The core-storage position that loses its word mark was previously specified by the instruction, and the address was placed in *both the A-Add. Reg. and the B-Add. Reg. during I-phase*. The B-cycle occurs, and the core-storage position that lost its word mark during the A-cycle attempts to lose a word mark again during the B-cycle. The B-cycle operation for a 1-field instruction is exactly the same as for a 2-field instruction.

#### Move (M AAA BBB)

The MOVE instruction moves the data in the A-field to the B-field. The word-mark status of both fields and the

data in the A-field remain unchanged during the move operation.

This data movement is done by executing alternate A- and B-cycles. The A-cycle occurs first and brings the character from the specified core-storage location to the A-register. The following B-cycle brings a character out of the other specified core-storage location and replaces it with the character in the A-register. This new character is then stored in the core-storage location specified by the B-Add. Reg.

#### **A-Cycle**

The common A-cycle previously described (see Figure 13 and the accompanying text) is used during the A-cycle portion of the E-phase. As soon as the A-cycle is successfully completed, the B-cycle begins.

#### **B-Cycle (Figure 35)**

The core-storage position that receives the A-cycle character during the B-cycle was previously specified by the instruction, and the address was placed in the B-Add. Reg. during I-phase. The address in the B-Add. Reg. is transferred to the STAR and the B-Add. Reg. key-light on the console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.

The MOVE operation lets the A-register character, minus the word-mark bit, replace the B-register character in the specified core-storage position.

Both the A- and the B-register characters are checked to see if either one contains a word mark. If a word mark is present in either register, the operation is terminated at the end of the B-cycle. If no word mark is present, it means that more A- and B-cycles must be taken until a word mark is encountered.

#### **CONTINUE OPERATION**

If more A- and B-cycles must be taken, the address in the B-Add. Reg. is decreased by one and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the B-cycle ends and the A-cycle begins. Alternate A- and B-cycles take place until an A- or B-register word mark is encountered during a B-cycle. This condition sets up the circuitry that terminates the operation.

#### **TERMINATE OPERATION**

If the word mark encountered is in the B-register, it is sent into core storage and stored with the last character moved. (The core-storage position is still being specified by the B-Add. Reg.)

If a C-bit is needed to obtain odd-bit parity, it is also generated and placed in the same core-storage position.

The address in the B-Add. Reg. is decreased by one and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the E-phase ends and the I-phase for the next instruction begins.

#### **1-Field Operation**

It is possible to execute a MOVE instruction with only the A-address specified (M AAA). The B-address that is used is the resultant B-address from the last operation. During the I-phase of a move operation, the B-Add. Reg. read-in is blocked during I-cycles 1, 2, and 3. This format is particularly useful when sequential storing of scattered fields is wanted. When this instruction is used, the scattered fields in an A-area are sequentially stored in the B-area. The A-address specifies the starting position of an A-field, while the B-address is automatically taken from the B-Add. Reg. The A- and B-cycle execution for a 1-field instruction is exactly the same as for a 2-field instruction.

*Note.* This instruction cannot be used when the preceding operation was a branch operation because the B-Add. Reg. is reset to blanks during a branch operation.

#### **Move Characters and Suppress Zeros (Z AAA BBB)**

The MOVE CHARACTERS AND SUPPRESS ZEROS instruction moves the data in the A-field to the B-field, followed by a zero-suppression operation. The zero-suppression operation starts at the high-order position of the data field and ends at the low-order position of the data field. Operating on data in this manner is called *reverse scanning*. During the zero-suppression operation, any zeros, commas, or hyphens are changed to blanks if encountered before the first significant digit. The first significant digit encountered stops the zero-suppression operation. However, any valid 1401 characters (other than a zero, comma, hyphen, blank, or significant digit), encountered after the first significant digit, start zero suppression again.

The MOVE CHARACTERS AND SUPPRESS ZEROS operation is made up of two distinct operations. During the first operation, the data is moved from the A-field to the B-field. During the second operation, the B-field reverse-scan operation and its associated zero-suppression operation take place.

#### **Data-Movement Operation**

This data movement is done by executing alternate A- and B-cycles. The A-cycle occurs first and brings the character from the specified core-storage location to the

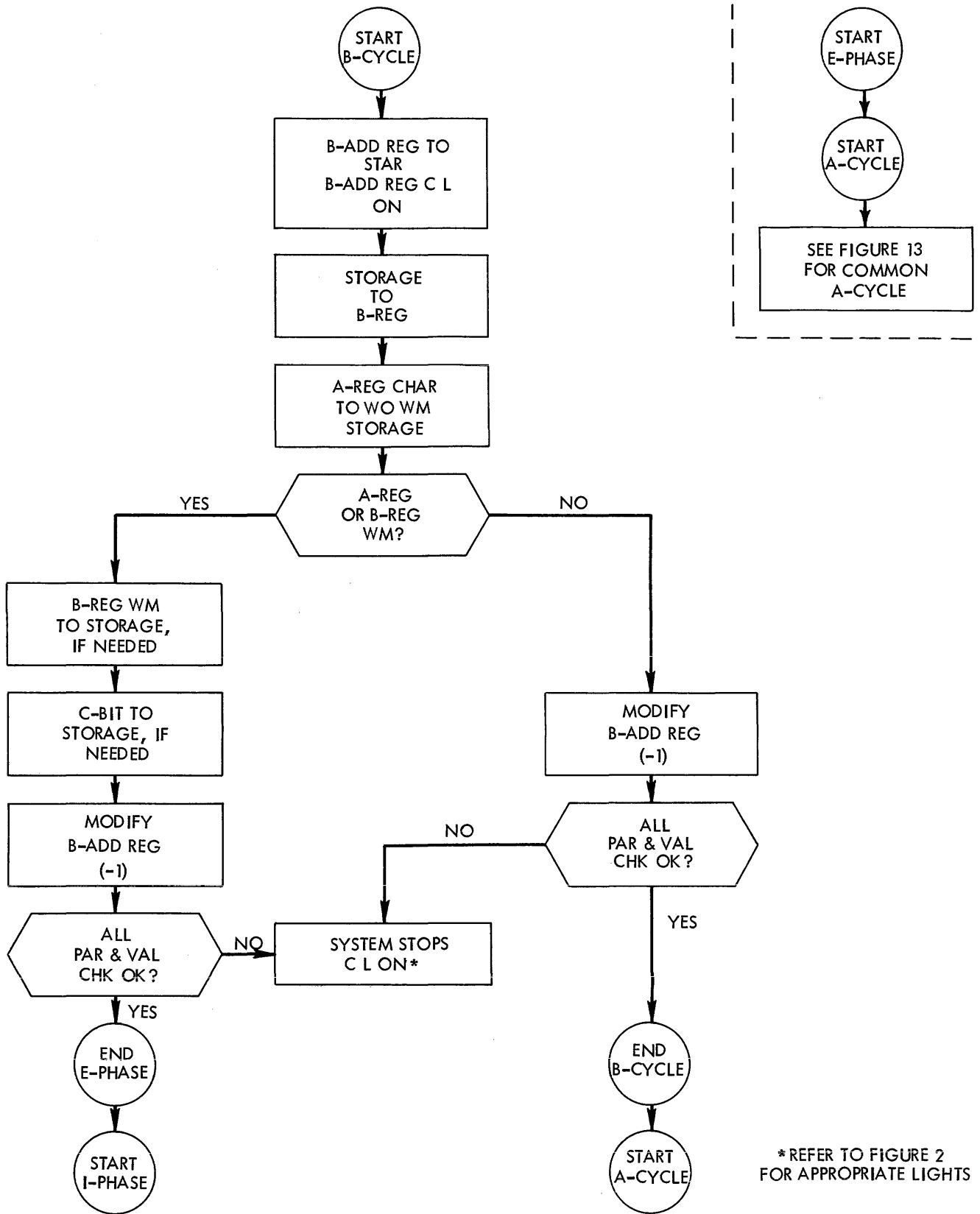


Figure 35. Move Instruction: B-Cycle Operation

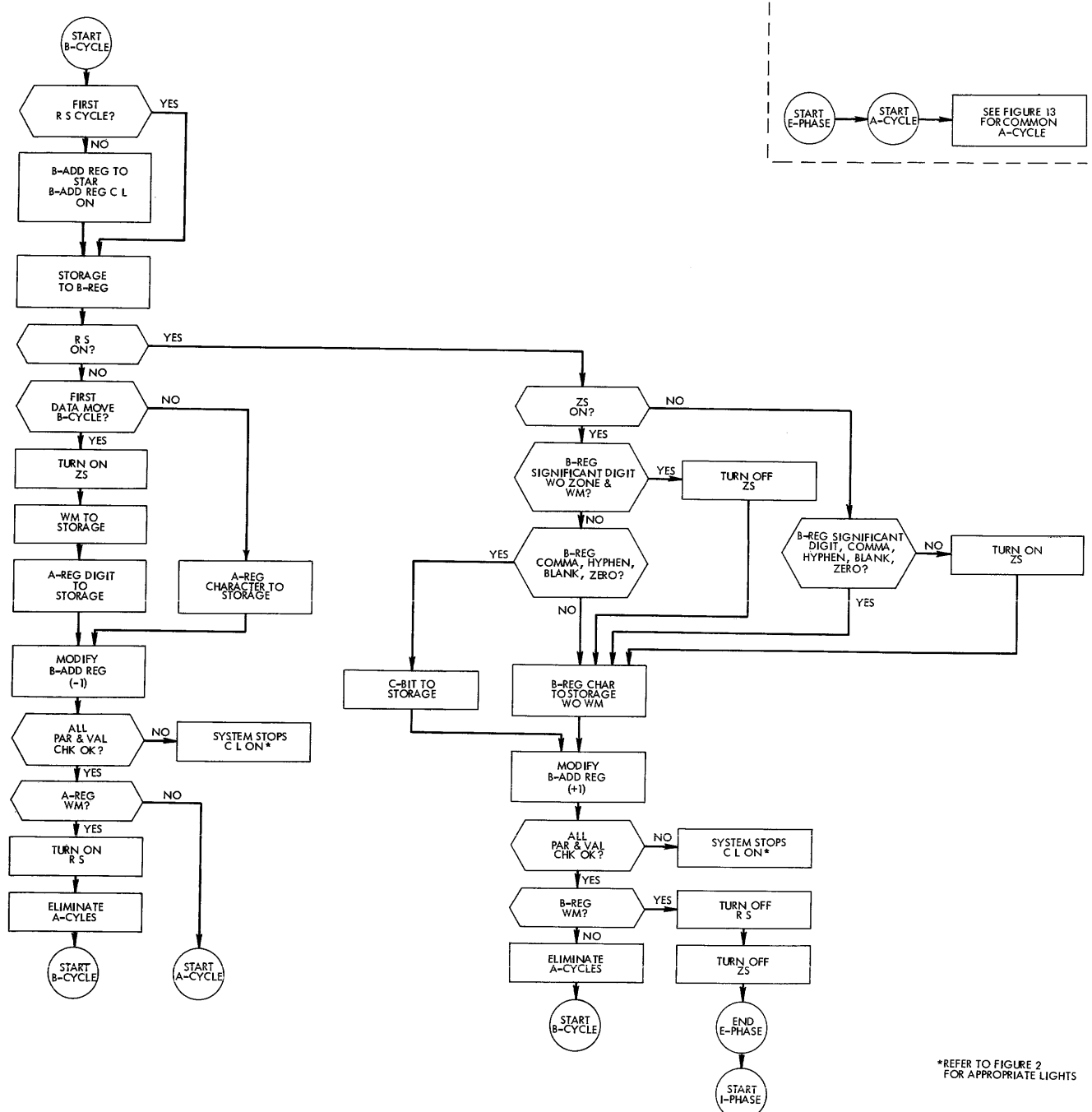
A-register. The following B-cycle brings a character out of the other specified core-storage location and replaces it with the character in the A-register. This new character is then stored in the core-storage location specified by the B-Add. Reg.

**A-CYCLE**

The common A-cycle previously described (see Figure 13 and the accompanying text) is used during the A-cycle portion of the E-phase. As soon as the A-cycle is successfully completed, the B-cycle begins.

**B-CYCLE (FIGURE 36)**

The core-storage position that receives the character in the A-register during the B-cycle was previously specified by the instruction and the address was placed in the B-Add. Reg. during I-phase. Because this is the data movement operation, and not the first reverse-scan cycle of a zero suppression operation, the address in the B-Add. Reg. is transferred to the STAR and the B-Add. Reg. light on the console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).



\*REFER TO FIGURE 2 FOR APPROPRIATE LIGHTS

Figure 36. Move Characters and Zero Suppress Instruction: B-Cycle Operation



The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.

During the first data movement B-cycle, a word mark is placed in the core-storage position specified by the B-Add. Reg. Also, circuitry is set up so that a reverse scan and its associated zero-suppression operation follow the data-movement operation. During this first data movement B-cycle, only the A-register digit bits and any associated C-bit are transferred into the core-storage position specified by the B-Add. Reg. On subsequent data movement B-cycles, the entire A-register character is transferred.

The address in the B-Add. Reg. is decreased by one and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the A-register is checked to see if it contains a word mark. A word mark in the high-order position of the A-field stops the data-movement operation. If no word mark is present, the B-cycle ends and another A-cycle begins.

A word mark in the A-register signals the end of the data-movement operation and the beginning of the reverse-scan operation. This reverse scan operates on the B-field from its high-order position to its low-order position. In addition to setting up the reverse-scanning circuitry, a word mark in the A-register also sets up the circuitry to eliminate all A-cycles. This is done because the field to be operated on is located in the B-field area.

### **Reverse-Scan Operation**

The reverse-scan operation is made up of a series of B-cycles, and is terminated when the word mark previously set in the field's low-order position is sensed. The first reverse-scan B-cycle reads out the same core-storage position that was involved in the last data-movement B-cycle.

The address of this core-storage position is no longer in the B-Add. Reg., however, because the B-Add. Reg. address was already modified. The STAR still has this address, so it is used to activate the lines that cause the specified core-storage position to read out to the B-register, where the contents are displayed in BCD form.

During a reverse-scan operation, zero suppression may be active or inactive. During the first reverse-scan B-cycle, zero suppression is always on.

The character read back into core storage depends on the B-register contents. If the B-register contains a significant digit, the zero-suppression circuitry is turned off and the significant digit in the B-register is read back into the core-storage position from which it came. If zero suppression is ON and the B-register does not contain a significant digit, but does contain a comma,

hyphen, blank, or zero, then this character is kept from entering core storage. A C-bit is generated and read into storage in place of the character. If zero suppression is OFF and the B-register does not contain a significant digit, a comma, hyphen, blank, or zero, then the character is read back into the core-storage position it came from, and zero suppression is turned on again.

When zero suppression is inactive, the B-register character is read back into the core-storage position from which it came. Some characters are also capable of activating the zero-suppression circuitry after it has been turned off. Any valid 1401 character, other than a zero, comma, hyphen, blank, or significant digit, can activate the zero-suppression circuitry after it has been turned off.

After the B-register character is read back into core storage, or is suppressed, the address in the B-Add. Reg. is increased by one and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the B-register is checked to see if a word mark is present. There is only one word mark present in the B-field and it is in the units position of the field. If the word mark is not present, the circuitry to eliminate the A-cycles is activated and another B-cycle is started. If the word mark is present, the reverse-scan circuitry, and the zero-suppression circuitry if on, are turned off. The E-phase is ended and the I-phase for the next instruction begins.

### **Move Digit (D AAA BBB)**

The MOVE DIGIT instruction moves the numerical portion (8-, 4-, 2-, 1-bits) of the single character in the specified A-address position to the specified B-address position. These addresses were placed in their respective address registers during the I-phase. The zone portions (A-, B-bits) and any word marks are undisturbed at both addresses.

### **A-Cycle**

The common A-cycle previously described (see Figure 13 and the accompanying text) is used during the A-cycle portion of the E-phase. As soon as the A-cycle is successfully completed, the B-cycle begins.

### **B-Cycle (Figure 37)**

The core-storage position that receives the numerical portion of the A-address character during the B-cycle was previously specified by the instruction, and the address was placed in the B-Add. Reg. during I-phase. The address in the B-Add. Reg. is transferred to the STAR and the B-Add. Reg. key-light on the 1401 console panel is turned on. The STAR lights on the console

panel display the core-storage position being addressed (in BCD form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.

An Op-code test establishes the operation as a move digit operation. A move digit operation allows any A-, B-, and word-mark bits in the B-register to read back into the core-storage position from which they came. In addition, any 8-, 4-, 2-, and 1-bits in the A-register are read into the same core-storage position. (These digits bits, along with the rest of the character, were placed in the A-register during the A-cycle.)

If a C-bit is needed to obtain odd-bit parity, it is also generated and placed in the same core-storage position.

The address in the B-Add. Reg. is decreased by one and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the E-phase ends and the I-phase for the next instruction begins.

### Move Zone (Y AAA BBB)

The MOVE ZONE instruction moves the zone portion (A-, B-bits) of the single character in the specified A-address position to the specified B-address position. These addresses were placed in their respective address registers during the I-phase. The digit portions (8-, 4-, 2-, 1-bits) are undisturbed at both addresses.

### A-Cycle

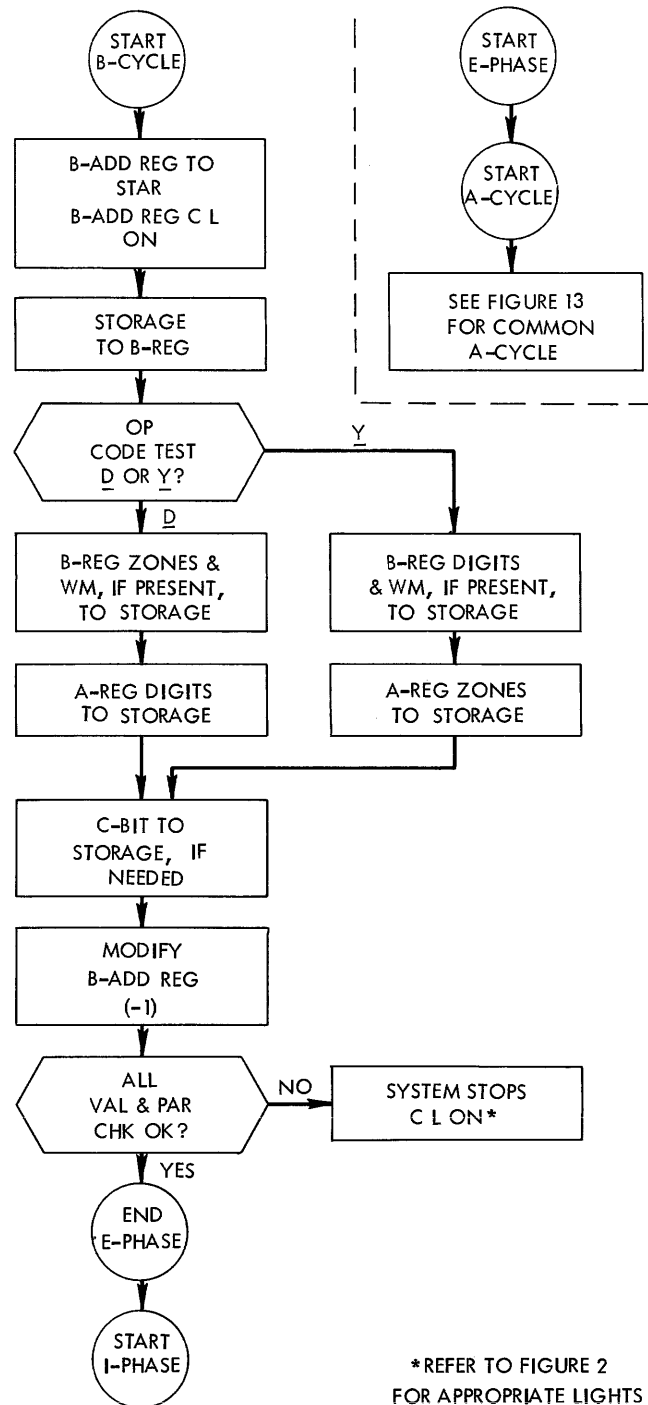
The common A-cycle previously described (see Figure 13 and the accompanying text) is used during the A-cycle portion of the E-phase. As soon as the A-cycle is successfully completed, the B-cycle begins.

### B-Cycle (Figure 37)

The core-storage position that receives the zone portion of the A-address character during the B-cycle was previously specified by the instruction, and the address was placed in the B-Add. Reg. during I-phase. The address in the B-Add. Reg. is transferred to the STAR and the B-Add. Reg. key-light on the 1401 console panel is turned on. The STAR lights on the console panel display the core-storage positions being addressed (in BCD form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.

An Op-code test establishes the operation as a move zone operation. A move zone operation allows any 8-,



\*REFER TO FIGURE 2 FOR APPROPRIATE LIGHTS

Figure 37. Move Digit and Move Zone Instructions: B-Cycle Operation

4-, 2-, or 1-bit (and word mark, if present) in the B-register to read back into the core-storage position from which they came. Also, any A- or B-bits in the A-register are read into the same core-storage position. (These bits, along with the rest of the character, were placed in the A-register during the A-cycle.)

If a C-bit is needed to obtain odd-parity, it is also generated and placed in the same core-storage position. The address in the B-Add. Reg. is decreased by one

and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the E-phase ends and the I-phase for the next instruction begins.

### Load (L AAA BBB)

The LOAD instruction moves the data and word mark from the A-field to the B-field. All data and word marks in the affected B-field are cleared.

This data movement is done by executing alternate A- and B-cycles. The A-cycle occurs first and brings the character from the specified core-storage location to the A-register. The following B-cycle brings a character out of the other specified core-storage location and replaces it with the character in the A-register. This new character is stored in the core-storage location specified by the B-Add. Reg.

#### A-Cycle

The common A-cycle previously described (see Figure 13 and the accompanying text) is used during the A-cycle portion of the E-phase. As soon as the A-cycle is successfully completed, the B-cycle begins.

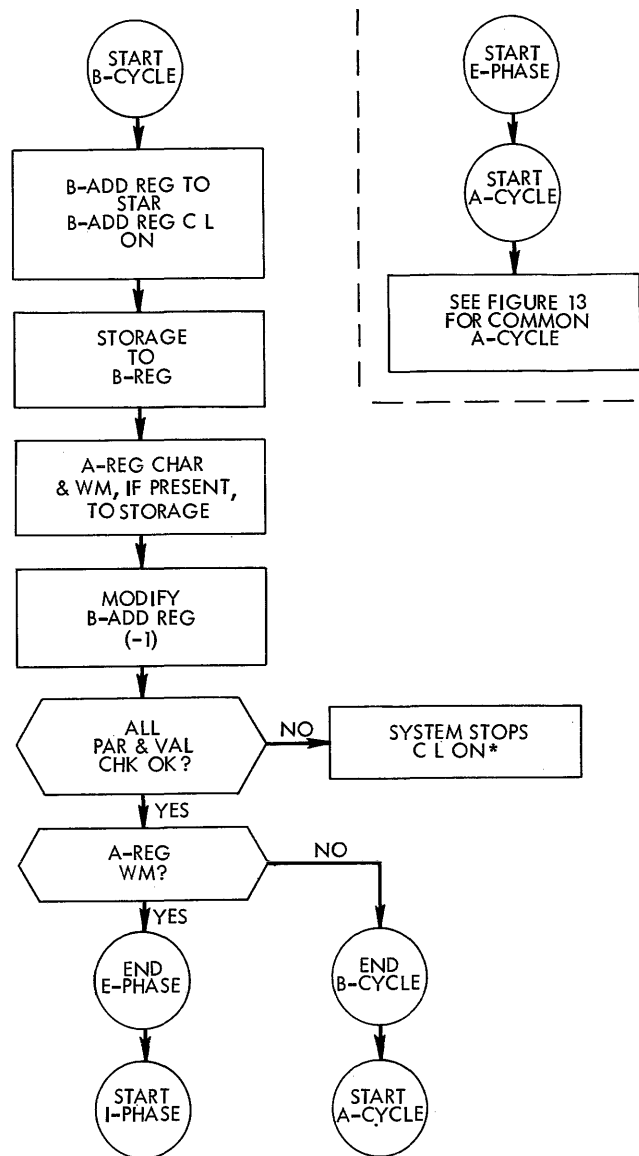
#### B-Cycle (Figure 38)

The core-storage position that receives the A-cycle data during the B-cycle was previously specified by the instruction and the address was placed in the B-Add. Reg. during I-phase. The address in the B-Add. Reg. is transferred to the STAR and the B-Add. Reg. key-light on the console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).

The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.

The load operation lets the A-register character (and word mark, if present) replace the B-register character (and word mark, if present) in the core-storage position specified by the B-Add. Reg. The address in the B-Add. Reg. is decreased by one, and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the console panel. If no check condition occurs, the A-register character is checked to see if it contains a word mark. If a word mark is present in the A-register, it signifies the end of the data field, and the operation is terminated by ending the E-phase and starting the I-phase of the next instruction.

If there is no word mark present, then more alternate A- and B-cycles must be taken until an A-register word



\* REFER TO FIGURE 2 FOR APPROPRIATE LIGHTS

Figure 38. Load Instruction: B-Cycle Operation

mark is encountered. Therefore, the B-cycle ends, and the next A-cycle begins.

#### I-Field Operation

It is possible to execute a LOAD instruction with only the A-address specified (L AAA). The B-address used is the resultant B-address from the last operation. During the I-phase of a load operation, the B-Add. Reg. read-in is blocked during I-cycles 1, 2, and 3. This format is particularly useful when sequential storing of scattered fields is wanted. When this instruction is used, the scattered fields in the A-area are sequentially stored in the B-area. The A-address specifies the starting position of the A-field, while the B-address is automatically taken

from the B-Add. Reg. The A- and B-cycle execution for a 1-field instruction is exactly the same as for a 2-field instruction.

*Note:* This instruction cannot be used when the preceding operation was a branch operation, because the B-Add. Reg. is reset to blanks during a branch operation.

### Move Characters and Edit (E AAA BBB)

A MOVE CHARACTERS AND EDIT instruction moves the A-field data into the B-field area. During the move, however, the A-field data is modified by the contents of the B-field, which is the edit control field. This edit control field specifies the punctuation of the edited data, the insertion of identifying symbols, and indicates where zero suppression will occur.

The MOVE CHARACTERS AND EDIT operation may be made up of two distinct operations, depending on whether zero suppression is specified. During the first operation, the data in the A-field is moved, and modified under control of the B-field, into the B-field. During the second operation, a B-field reverse scan and its associated zero-suppression operation takes place, if it is specified during the first operation.

#### Data Movement and Modification Operation

The data movement and modification is accomplished by executing A- and B-cycles as specified by the edit control word. The first cycle of an E-phase is always an A-cycle, followed by a B-cycle. From then on, the specific cycle execution is under control of the edit control word. All numeric, alphabetic, and special characters can be used in the edit control word. Some of these characters have special uses, and each of these uses is covered in the text.

The various operations that take place during the E-phase of an edit operation are illustrated in Figures 40 and 41. A sample edit operation is performed (refer to edit operation example, Figure 39), and the various operations performed are listed by number with any necessary clarification also included. (The various interrogations performed are listed.)

A-Field (Data Word)	(-) 01405
B-Field Before Edit (Control Word)	\$b,b0b.bb&CR*
B-Field After Modification	\$b,014.05bCR*
B-Field After Zero Suppression	\$bbb14.05bCR*

Figure 39. Move Characters and Edit Operation Example

#### ALL A-CYCLES

The common A-cycle previously described (see Figure 13 and the accompanying text) is used during all the A-cycle portions of the E-phase. As soon as the A-cycle is successfully completed, the B-cycle begins.

#### FORWARD SCAN OPERATION (SEE FIGURE 40)

*A-Cycle 1.* Common A-cycle transfers a five and a minus sign from the specified core-storage position into the A-register.

*B-Cycle 1 (Figure 40).* CONDITIONS: A-register contains a five and a minus sign after a common A-cycle completion.

B-cycle 1 executes these operations (refer to Figure 40).

- 1
- 2 This is a B-cycle executed during a forward-scan cycle.
- 3 Specified core-storage position contains an asterisk (\*), which is read into the B-register.
- 4 The \* is part of the identifying symbols used with the number, so it is placed back in the storage position from which it came.
- 5
- 6 The next character in the B-field must be brought out so it can control the A-register contents, if necessary.
- 7

B-cycle 2 executes these operations (refer to Figure 40).

- 1
- 2 This is a B-cycle executed during a forward-scan cycle.
- 3 Specified core-storage position contains an R, which is read into the B-register.
- 4 Because the A-register contains a minus sign, the R portion of a CR notation is placed back in the storage position from which it came.
- 5
- 6 The next character in the B-field must be brought out so it can control the A-register contents, if necessary.
- 7

B-cycle 3 executes these operations (refer to Figure 40).

- 1
- 2 This is a B-cycle executed during a forward-scan cycle.
- 3 Specified core-storage position contains a C, which is read into the B-register.
- 4 Because the A-register contains a minus sign, the C portion of a CR notation is placed back in the storage position from which it came.
- 5
- 6 The next character in the B-field must be brought out so it can control the A-register contents, if necessary.
- 7

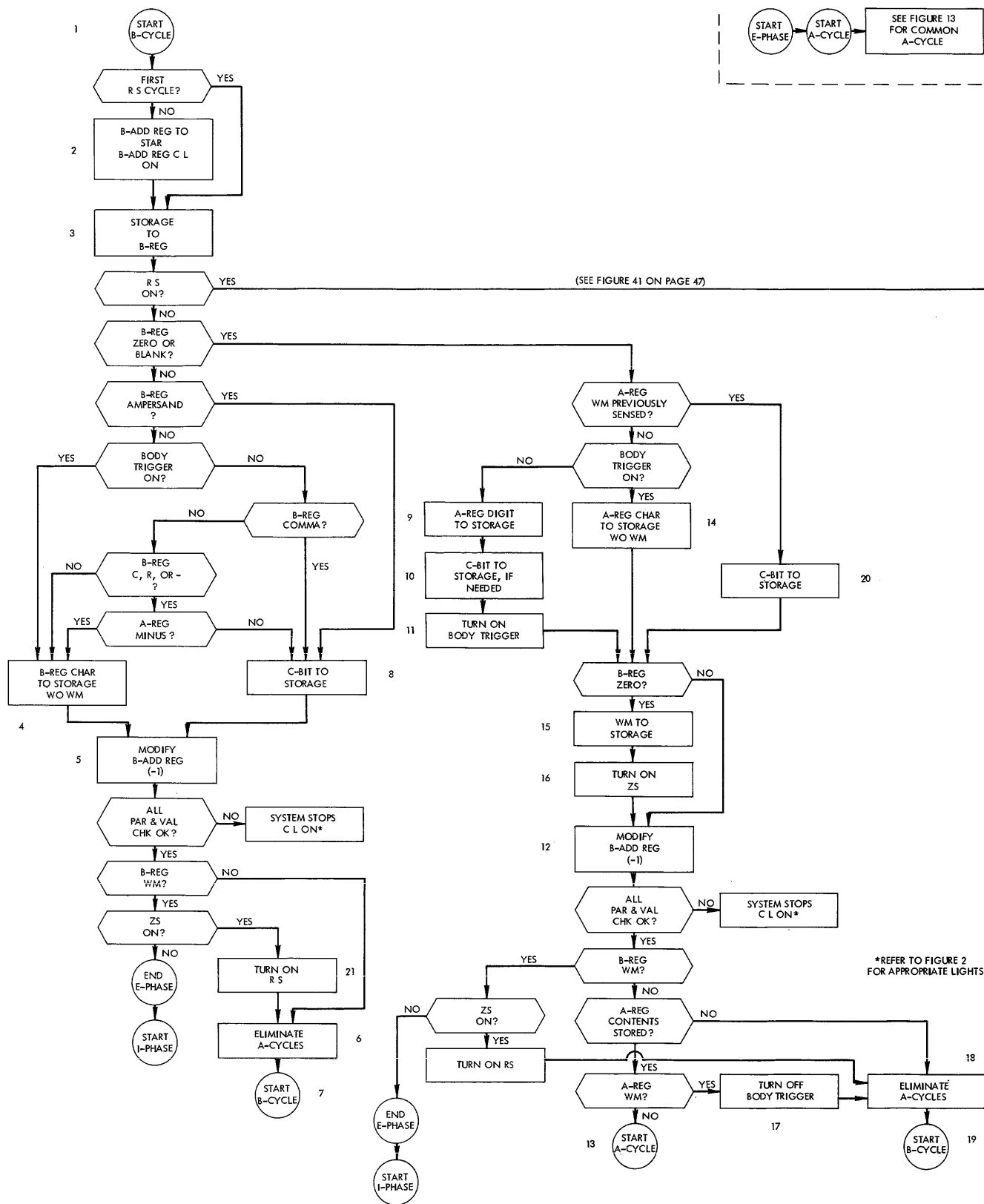


Figure 40. Move Characters and Edit Instruction: Forward-Scan Operation

B-cycle 4 executes these operations (refer to Figure 40). CONDITIONS: A-register still contains a five and a minus sign.

- 1
- 2 This is a B-cycle executed during a forward-scan cycle.
- 3 Specified core-storage position contains an ampersand (&), which is read into the B-register.
- 8 An ampersand sensed in the B-register results in a blank space in the edited B-field. A C-bit replaces the ampersand in the specified core-storage position.
- 5
- 6 The next character in the B-field must be brought out so it can control the A-register contents, if necessary.
- 7

B-cycle 5 executes these operations (refer to Figure 40). CONDITIONS: A-register still contains a five and a minus sign.

- 1
- 2 This is a B-cycle executed during a forward-scan cycle.
- 3 Specified core-storage position contains a blank, which is read into the B-register.
- 9 The five in the A-register enters the specified core-storage position. The zone portion of the A-register does not enter core-storage during the first cycle that transfers data from the A-register into core storage.
- 10 A C-bit is added to the transferred digit if it is needed to obtain odd-bit parity.
- 11 The body trigger is set ON. There are two parts to every edit control word: the *body* portion (which punctuates the A-field), and the *status* portion (which contains the dollar sign, sign-symbols, and class of total asterisks). A blank or zero in the B-register, coupled with the body trigger turned OFF, signifies the first body position of the edit control word. The body trigger remains ON to the end of the A-field, if sensed.
- 12
- 13

*A-Cycle 2.* CONDITIONS: Body trigger ON. Common A-cycle transfers a zero from the specified core-storage position into the A-register.

B-cycle 6 executes these operations (refer to Figure 40). CONDITIONS: A-register contains a zero, and the body trigger is ON.

- 1
- 2 This is a B-cycle executed during a forward-scan cycle.
- 3 Specified core-storage position contains a blank, which is read into the B-register.
- 14 Character in the A-register (a zero) replaces the B-register blank in the specified core-storage position.
- 12
- 13

*A-Cycle 3.* CONDITIONS: Body trigger is ON. Common A-cycle transfers a four from the specified core-storage position into the A-register.

B-cycle 7 executes these operations (refer to Figure 40). CONDITIONS: A-register contains a four, and the body trigger is ON.

- 1
- 2 This is a B-cycle executed during a forward-scan cycle.
- 3 Specified core-storage position contains a decimal (.), which is read into the B-register.
- 4 The decimal in the B-register is placed back in the storage position from which it came.
- 5
- 6 The next character in the B-field must be brought out so it can control the A-register contents, if necessary.
- 7

B-cycle 8 executes these operations (refer to Figure 40). CONDITIONS: A-register still contains a four, and the body trigger is ON.

- 1
- 2 This is a B-cycle executed during a forward-scan cycle.
- 3 Specified core-storage position contains a blank, which is read into the B-register.
- 14 Character in the A-register (a four) replaces the B-register blank in the specified core-storage position.

12

13

*A-Cycle 4.* CONDITION: Body trigger is ON. Common A-cycle transfers a one from the specified core-storage position into the A-register.

B-cycle 9 executes these operations (refer to Figure 40). CONDITIONS: A-register still contains a one and the body trigger is ON.

- 1
- 2 This is a B-cycle executed during a forward-scan cycle.
- 3 Specified core-storage position contains a zero, which is read into the B-register.
- 14 Character in the A-register (a one) replaces the B-register zero in the specified core-storage position.
- 15 A B-register zero condition also generates a word mark that is placed in the specified core-storage position with the A-register character that replaced the zero. This word mark ends the reverse scan and its associated zero-suppression operation.
- 16 A zero in the B-register also sets up the circuitry necessary for executing a zero-suppression operation after the data movement and modification operation ends.

12

13

*A-Cycle 5.* CONDITIONS: Zero suppression and the body trigger are ON. Common A-cycle transfers a zero and a word mark from the specified core-storage position into the A-register.

B-cycle 10 executes these operations (refer to Figure 40). CONDITIONS: A-register contains a zero and a word mark; zero suppression and the body trigger are ON.

- 1
- 2 This is a B-cycle executed during a forward-scan cycle.

- 3 Specified core-storage position contains a blank, which is read into the B-register.
- 14 Character in the A-register (a zero) replaces the B-register blank in the specified core-storage position (no word mark transferred).
- 12
- 17 An A-register word mark indicates the end of the data (A) field. This indication is used to turn off the body trigger.
- 18 The B-field forward-scan operation must continue until a B-register word mark is sensed.

19 B-cycle 11 executes these operations (refer to Figure 40). CONDITIONS: A-register still contains a zero and a word mark; zero suppression is ON, but body trigger is OFF.

- 1 This is a B-cycle executed during a forward-scan cycle.
- 2 Specified core-storage position contains a comma, which is read into the B-register.
- 8 With the body trigger OFF, a C-bit is generated and it replaces the comma in the specified core-storage position.
- 5
- 6 The B-field forward scan operation must continue until a B-register word mark is sensed.

7 B-cycle 12 executes these operations (refer to Figure 40). CONDITIONS: A-register still contains a zero and a word mark; zero suppression is ON, body trigger is OFF.

- 1 This is a B-cycle executed during a forward-scan cycle.
- 2 Specified core-storage position contains a blank, which is read into the B-register.
- 20 A C-bit, denoting a blank position, is generated and placed in the specified core-storage position. (This same thing would happen if the B-register contained a zero.)
- 12
- 18 The B-field forward-scan operation must continue until a B-register word mark is sensed.

19 B-cycle 13 executes these operations (refer to Figure 40). CONDITIONS: A-register still contains a zero and a word mark; zero suppression is ON, body trigger is OFF.

- 1 This is a B-cycle executed during a forward-scan cycle.
- 2 Specified core-storage position contains a dollar sign (\$) and a word mark, which are read into the B-register.
- 4 The dollar sign in the B-register is placed back in the core-storage position from which it came (no word mark transferred).
- 5
- 21 With zero suppression ON, a B-register word mark signals the reverse-scan operation and its associated zero-suppression operation.
- 6
- 7

END OPERATION WITHOUT ZERO SUPPRESSION

With no zero-suppression specified, a B-register word mark ends the E-phase and begins the I-phase of the next instruction.

Reverse-Scan Operation (Figure 41)

The reverse-scan operation, and its associated zero-suppression operation, is made up of a series of B-cycles. The operation is terminated when the word mark set in the edited word is sensed. The first reverse-

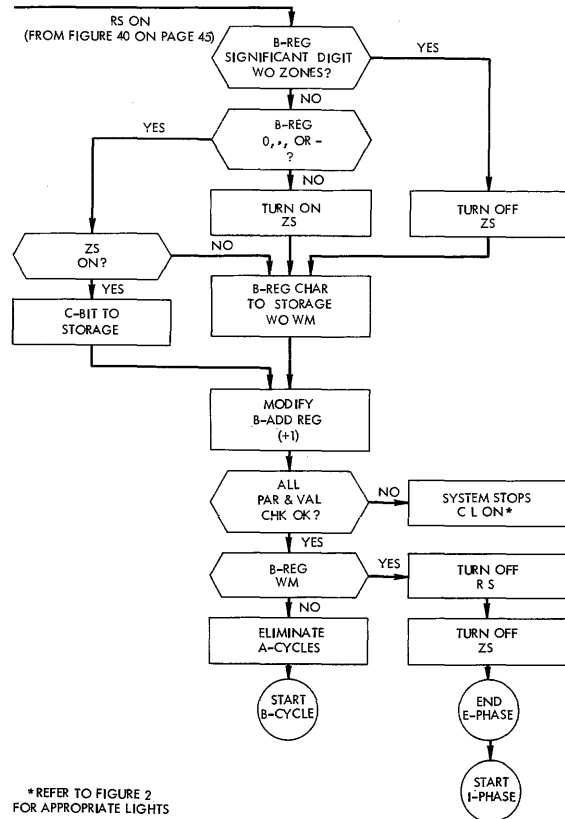


Figure 41. Move Characters and Edit Instruction: Reverse-Scan Operation

scan B-cycle reads out the same core-storage position that was involved in the last data movement and modification B-cycle. The address of this core-storage position is no longer in the B-Add. Reg., however, because the B-Add. Reg. address was already modified. The STAR does have this address, so it is used to activate the lines that cause the specified core-storage position to read out to the B-register, where the contents are displayed in BCD form.

During a reverse-scan operation, zero suppression may be active or inactive. During the first reverse-scan B-cycle, zero suppression is always ON.

The character read back into core storage depends on the B-register contents. If the B-register contains a significant digit, the zero-suppression circuitry is turned off, and the significant digit in the B-register is read

back into the core-storage position from which it came.

If zero suppression is on, and the B-register does not contain a significant digit (but does contain a zero, decimal, or hyphen) then this character is kept from entering core storage. A C-bit is generated and read into storage in place of the character. If zero suppression is off, then the character is read back into the core-storage position from which it came.

If zero suppression is off and the B-register does not contain a significant digit, zero, decimal, or hyphen, then the character is read back into the core-storage position it came from, and zero suppression is turned on again.

After the B-register character is read back into core storage, or is suppressed, the address in the B-Add. Reg. is increased by one and the appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check light on the console panel. If no check condition occurs, the B-register is checked to see if a word mark is present. There is only one word mark present in the edited word, and that was set to mark the limit of zero suppression. If the word mark is not present, the circuitry to eliminate the A-cycles is activated and another B-cycle is started. If the word mark is present, the reverse-scan circuit, and the zero-suppression circuitry if on, are turned off. The E-phase ends and the I-phase for the next instruction begins.



## Input-Output Operations

This section describes the data movement between the 1401 and the various input-output units that can be attached to an IBM 1401 Data Processing System.

### Card Read Operations

It is possible to attach the IBM 1402 Card Read-Punch to the 1401. The read section has a maximum rated reading speed of 800 cards per minute. The actual card speed realized is governed by the stored-program instructions.

#### Read (I) or Read and Branch (I III)

The stored-program instruction that initiates the data transfer between the 1402 and the 1401 is the READ (I) instruction. To execute an instruction other than the instruction that is next in sequence to the READ instruction, a READ AND BRANCH instruction is given. The regular read operation takes place, but the address of the

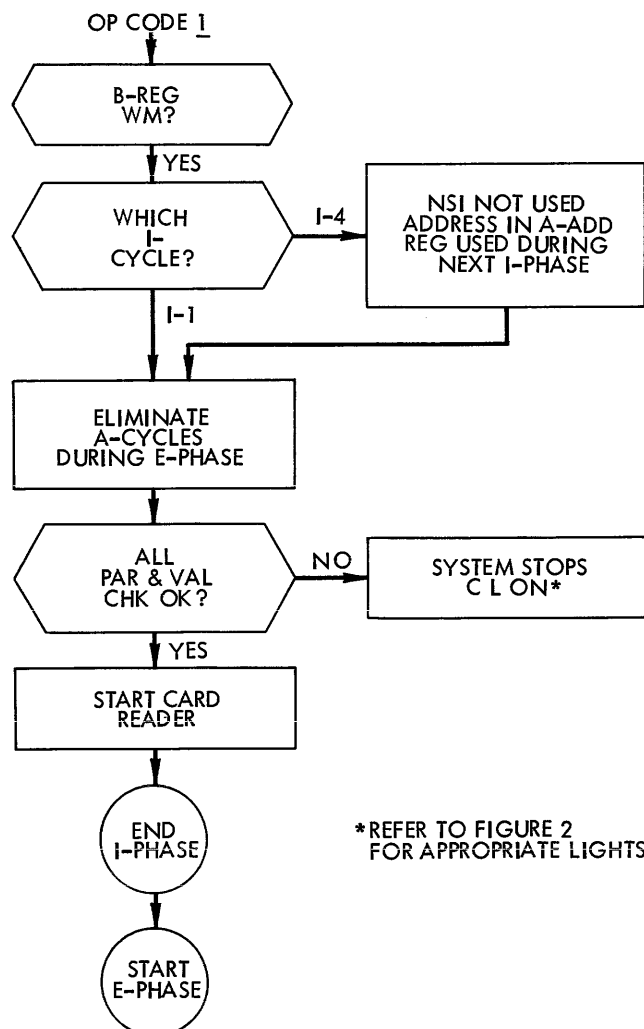


Figure 42. Read or Read and Branch Instruction: I-Phase Set-up Operations

next instruction is taken from the A-Add. Reg. instead of the I-Add. Reg. The circuitry that accomplishes this is set up before the I-phase of the read operation ends.

#### I-PHASE SET-UP OPERATIONS (FIGURE 42)

Some instructions require alterations of the normal E-phase operations for correct execution of the instruction. These alterations are set up before the I-phase ends. During certain I-cycles the Op code is checked to see if it is one of the Op codes that has to set up the alteration of the normal E-phase operations before the actual instruction execution begins.

When the B-register word mark is sensed during the I-phase of a read operation, the specific I-cycle must be established. If the I-cycle is I-1, the instruction is a READ instruction and the next sequential instruction is executed after the read operation E-phase. If the I-cycle is I-4, the address of the next instruction is the address in the A-Add. Reg. The address in the I-Add. Reg. is ignored.

The circuitry is also set-up to eliminate A-cycles during the E-phase portion of the operation. The appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check light on the console panel. If no check condition occurs, the card reader is started. The I-phase ends and the E-phase begins.

#### E-PHASE OPERATION (FIGURE 43: 1 AND 2 OF TWO)

Once the card reader is started, it remains operating long enough to feed one card past the second read brushes and another card past the first read brushes. The first read brushes read the card and establish a hole count for checking purposes. The second read brushes read the card, prove the hole count established by the first read brushes, and direct the data into storage.

The hole-count check and the data transfer are accomplished by using two separate 80-core extensions of the main core-storage area. These extensions of the core-storage area are called row-bit cores. The first set of 80 brushes are connected to one 80-core extension, called the read one row-bit core area. The second set of 80 brushes are connected to the other 80-core extension, called the read two row-bit core area.

The card moves through the 1402 read section 9-edge first, face down. Therefore, nine is the first digit read by the first and second sets of brushes. The digit nine is followed by the digit eight, and so forth, through the 12-zone portion of the card. Between each digit reading time, the read two row-bit core information is transferred into the read area of core storage (positions 1 through 80) in BCD form.

The various operations that take place during the E-phase of a read operation are numbered in the text and

illustrated in Figure 43 (1 and 2 of two). A card read and data transfer operation is performed, and the sequence of operations is listed by number with any necessary clarification included. (The various interrogations performed are not listed.)

### 9-Digit Time.

- 1 The card reader contains a set of circuit breakers that emit an impulse for each digit time of a card (9 through 12). Before the setting of its read two row-bit core, the set of circuit breakers must emit an impulse for that particular digit time.
- 2 Nine is the first digit read. With the CBs emitting an impulse at 9-digit time, every card column that contains a 9-punch will *flip* its associated row-bit core. (To flip a row-bit core is to change its status—from on to off, or vice-versa. The status change is the information that is transferred to core storage.)
- 3 The 9-time CB impulse breaks. No other digit time CB impulses make until 8-digit time.
- 4 During the read-scan operation, the status information of the read two row-bit cores must be transferred into the read areas of core storage. Every read two row-bit core that changed status during 9-digit time must be recorded by inserting a nine in the associated read area core-storage position. This operation must be completed before 8-digit time.

Each read scan is made up of a series of B-cycles. Depending on the read scan, there are either two or three B-cycles necessary to set up the information transfer from the read two row-bit cores into core storage. The objective of each B-cycle is described as it occurs.

- 5 The address in the STAR is set to 000, and the B-Add. Reg. key-light on the console panel is turned on.
- 6 If the read-scan operation being performed is a 9-digit time read-scan operation, B-cycle A is started. During B-cycle A, a zero is inserted in core-storage position 000.
- 7 The contents of core-storage position 000 are read into the B-register, where they are displayed in BCD form.
- 8 A zero is generated and read back into core-storage position 000, replacing the B-register contents.
- 9 A C-bit is added to maintain odd-bit parity.
- 10 Another B-cycle (B) is started. The objective of this B-cycle is to place the digit that corresponds to the card-reading time in core-storage position 000. This is done by adding nine to the previous contents of core-storage position 000. During 9-digit time, the result is nine. During 8-digit time, the result is eight. (Previous contents of location 000 is nine; add nine to this and the result (units portion only) is eight.)
- 11 The zero just inserted into core-storage position 000 is read into the B-register, where it is displayed in BCD form.
- 12 The A-register is set to blank.
- 13 The digit portion of the A-register (a blank) is complement-added (a nine) to the digit portion of the B-register (a zero).
- 14 The addition result (9) is read back into the specified core-storage position (000).
- 15 A C-bit is added or removed to maintain odd-bit parity.
- 16 Another B-cycle (C) is started. The objective of B-cycle C is to transfer the digit located in core-storage position 000 into the A-register, and then get ready to translate the read two row-bit information and enter it into core-storage positions 001-080.

- 17 The addition result (9) just inserted into core-storage position 000 is read into the B-register, where it is displayed in BCD form.
- 18 The B-register contents are read back into the core-storage position from which they came (000).
- 19 At this time, the B-register contents are also transferred into the A-register.
- 20 The address in the B-Add. Reg. is increased by one (001).
- 21 Another B-cycle (D) is started. The objective of B-cycle D is to transfer the BCD-translated read two row-bit information into the specified core-storage position.
- 22 The address in the B-Add. Reg. is transferred to the STAR and the B-Add. Reg. key-light on the console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).

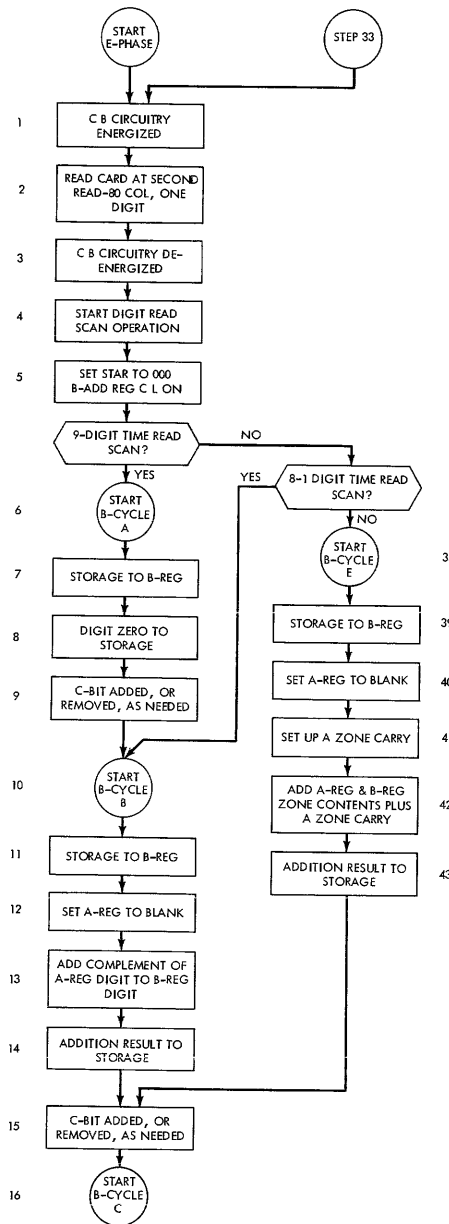


Figure 43 (1 of 2). Read or Read and Branch Instruction: E-Phase Operation

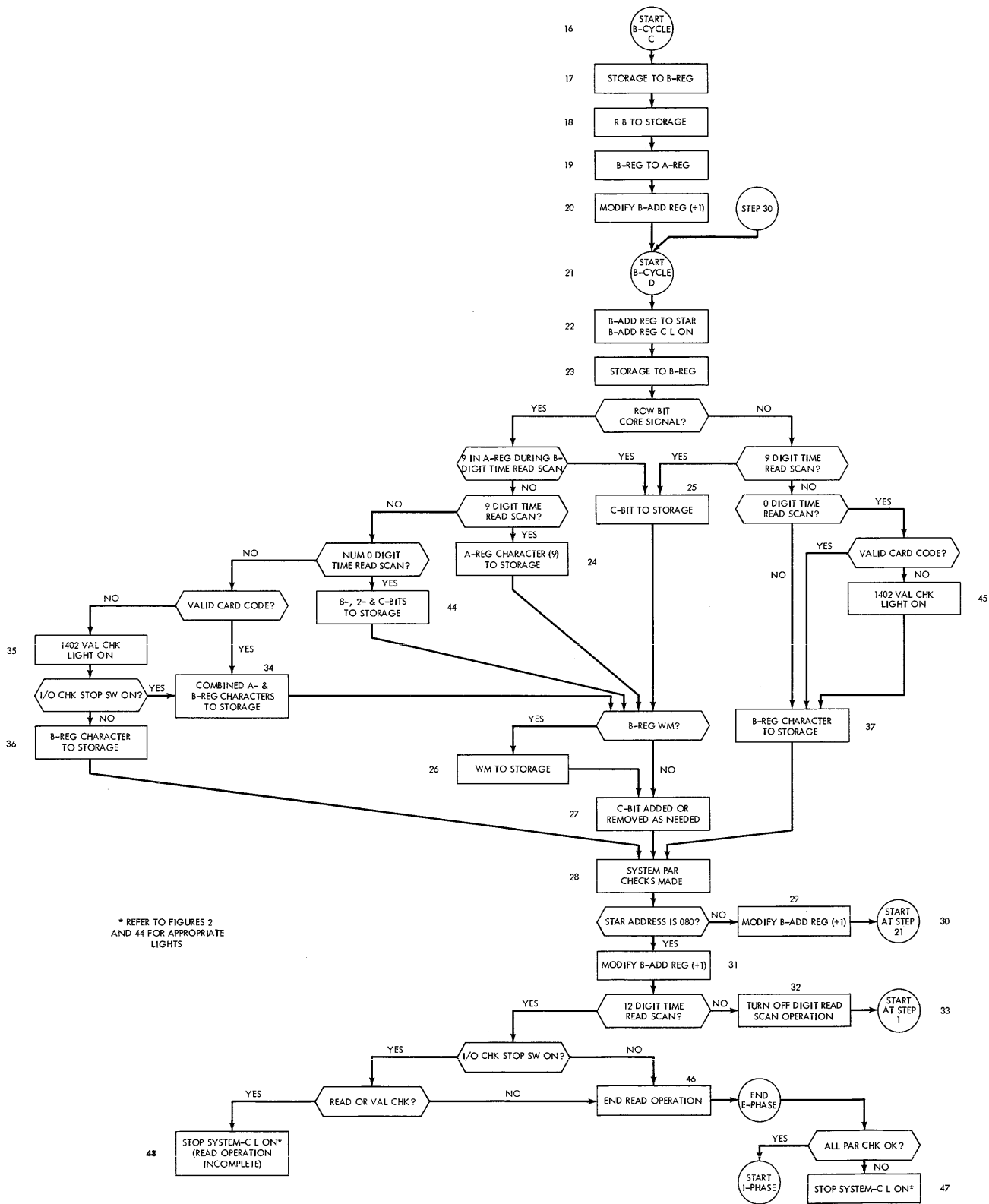


Figure 43 (2 of 2). Read or Read and Branch Instruction: E-Phase Operation

- 23 The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.  
As each specified core-storage position is addressed, the corresponding read two row-bit core position is also addressed.
- 24 If the addressed row-bit core changed status, during 9-digit time, then the A-register contents (9) are transferred into the specified core-storage position. With the exception of a word mark, the B-register contents are ignored.
- 25 If the addressed row-bit core did not change status, a C-bit replaces the B-register contents in the specified core-storage position. The read area of core storage is cleared of all previous information, except word marks, during 9-digit time read-scan operation.
- 26 If the B-register contains a word mark, it is transferred to the specified core-storage position.
- 27 A C-bit is added or removed, as needed.
- 28 The appropriate parity checks are made and the indicators, if any, are set on. The system will not stop, however, until all 12 scans have been made.
- 29 If the STAR address is not 080, then the address in the B-Add. Reg. is increased by one.
- 30 This B-cycle D ends and another B-cycle D begins. All 80 read two row-bit cores are tested sequentially.
- 31 As soon as all the read two row-bit cores are tested (STAR address is 080), the address in the B-Add. Reg. is increased by one.
- 32 The 9-digit read-scan operation ends.
- 33 The next digit read-scan operation can now take place.

### 8- Through 1-Digit Scan Time.

- 1-5 These steps are exactly the same as during 9-digit scan time.
- 10 A B-cycle (B) is started. The B-cycle A is used only during 9-digit time.
- 11-15 The B-cycle B operates the same as during 9-digit scan time. During this B-cycle, the previous contents of core-storage position 000 are brought out to the B-register and nine is added to it, resulting in a character that is one less than the previous character.  
The new character is read back into core-storage position 000.
- 16-20 The B-cycle C operates the same as during 9-digit scan time.
- 21-23 Steps 21, 22, and 23 of B-cycle D operate the same as during 9-digit scan time.
- 25 If the addressed row-bit core changes status during 8-digit time, and the B-register contains a nine, this MLP coding is rejected and a C-bit is placed in the specified core-storage position.
- 34 If the addressed row-bit core changes status, and the valid card code is not an MLP code, the A- and B-register contents are combined and placed in the specified core-storage position.
- 35 If the addressed row-bit core changes status, and the card code is not an MLP code, but is invalid, then the 1402 validity-check light is turned on.  
Under these conditions, the character sent back to core storage depends on the I/O check-stop switch setting. If the switch is set ON, step 34 occurs. If the switch is set OFF, step 36 occurs.
- 36 The B-register contents are transferred back into the core-storage position from which they came.

52

- 37 If the addressed row-bit core does not change status during 8- through 1-digit time, then the B-register contents are transferred back into the core-storage position from which they came.
- 28-33 These steps are exactly the same as during 9-digit scan time.

### 0-, 11-, and 12-Digit Scan Time.

- 1-5 These steps are exactly the same as during 9-digit scan time.
- 38 If it is 0-, 11-, or 12-digit scan time, a B-cycle E is started. The objective of this B-cycle is to place the BCD equivalent of a 0-, 11-, or 12-digit in core-storage position 000.
- 39 The contents of core-storage position 000 are moved into the B-register, where they are displayed in BCD form.
- 40 The A-register is set to blank (a C-bit).
- 41 The circuitry to set up a zone carry is activated.
- 42 The A-register zone contents (C-bit ignored) is added to the B-register zone contents plus a carry.
- 43 The addition result is stored in core-storage position 000.  
During 0-, 11- or 12-digit time, this addition operation results in the correct BCD equivalent.
- 15-23 These steps operate the same as during 9-digit scan time.
- 44 If the addressed row-bit core changes status during 0-digit time, and the B-register does not contain any other digit (signifying a numeric zero rather than a zone zero), an 8-, 2-, and C-bit combination (a numeric zero) are transferred into the specified core-storage position. (A zone zero would be added to the B-register contents in step 34.)
- 37 If the addressed row-bit core does not change status during 0-, 11-, or 12-digit time, and the card code is valid (11- and 12-digit time only), the B-register contents are transferred back into the core-storage position from which they came. An invalid card code initiates step 45.
- 45 If an invalid card code is detected during 11- or 12-digit time (B-register contains an 8-bit, a 2-bit, no 4-bit, and no 1-bit), the 1402 validity check light is turned on.  
Steps 34, 35, 36, 26-33 are exactly the same as during 8- through 1-digit scan time.
- 46 A special CB in the 1402 read section signals 12-digit time. When the STAR address is 080 at 12-digit time, and the I/O check-stop switch is OFF, the read operation ends. The E-phase ends and the I-phase of the next instruction begins unless a parity check occurred during the operation.  
If the I/O check-stop switch was ON, but no validity checks occurred during the operation, the read operation also ends.
- 47 If a parity check did occur during the operation, the system stops and the appropriate lights on the 1402 and the 1401 console panel, if not already on, are turned on.
- 48 If the I/O check-stop switch was ON, and either a hole-count or a validity check is detected during the operation, the system stops and the appropriate lights on the 1402 and the 1401 console panel, if not already on, are turned on. Under these circumstances, a non-process run-out will run the cards out. The cards can then be placed back in the read hopper, and the operation can be repeated by pressing the 1402 start key.

### Card Read Check Conditions (Figure 44)

These are certain check conditions that can occur during a 1402 read operation. These check conditions are:

1. Hole-count check
2. A-Register contents check
3. Validity
4. Feeding failure.

#### HOLE-COUNT CHECK

The read hole-count check detects reading errors that occur in the read section of the IBM 1402 Card Read-Punch.

The read hole-count check is a comparison of the punched holes read, in each card column, at the first set of read brushes against the punched holes read from each card column, of the same card, at the second set of brushes. An unequal comparison stops the system and turns on the read check light on the 1402 and the read light on the 1401 console panel.

The hole-count check is performed by two sets of two check planes. Each plane contains 80 cores. The check for any one card makes use of one set of these check planes.

The hole-count check for any one card starts with the card being read at the first reading station. The punched holes in the card are recorded by the read one row-bit cores. The information is then transferred from the read one row-bit cores into the assigned set of two check planes. Each plane accepts certain information, and the two planes together check for every probable reading error.

The punched holes in the card are then read at the second reading station and are recorded by the read two row-bit cores. As this information is transferred from the read two row-bit cores into the read area of

core storage, it is also transferred into the previously-specified set of check planes. Each plane accepts the same information that it accepted from the read one row-bit cores.

All cores in a specified check plane are at 0 when a check operation starts. Any check-plane core that is *flipped* a certain number of times by the read one row-bit core output, should be flipped the same number of times by the read two row-bit core output. Because of this, all cores in the specified check planes should be set to 0 at the end of the card-feed cycle. If any of the cores are set to one, it means that the card was incorrectly read at one of the reading stations.

#### A-REGISTER CONTENTS CHECK

During 12-digit time, the A-register is checked to see if it contains a B- and an A-bit (BCD bit code for 12). If the A-register does not contain these bits, the system stops. The 1402 read-check light and the read light on the 1401 console panel are turned on.

#### VALIDITY

As the information from the card reader is transferred into core storage and the characters are formed, each character is checked to see if it is a valid character.

At certain card-read times, when punched holes are sensed, the B-register contents are checked. Depending on the B-register contents, the character can be valid or invalid. An invalid character stops the system at the end of that read-feed cycle, provided the I/O check-stop switch is on. The validity light on the 1402 and the reader light on the 1401 console panel are turned on.

If the I/O check-stop switch is ON, the invalid character may also cause a parity error. In this instance, the storage light, the check-reset key-light, the process

TYPE OF CHECK	MACHINE STOPS (I/O CHECK STOP SWITCH ON CONSOLE PANEL ON)	LIGHTS ON WHEN STOPPED	RESET BY	REMARKS
Hole-Count Check A-Register Contents Check	At end of feed cycle	Read Check (1402) Read (1401)	Check Reset Key (1402)	Cards must be run out before the check reset key is effective.
Validity	At end of feed cycle	Validity (1402) Read (1401)  If the invalid combination causes incorrect parity Storage (1401) Process (1401) Check Reset (1401)	Check Reset Key (1402)  Check Reset Key (1401)	Cards must be run out before the check reset key is effective.
Feeding Failure	At end of feed cycle	Reader Stop (1402) Read (1401)	Check Reset Key (1402)	Cards must be run out before the check reset key is effective.

Figure 44. Card Read Check Conditions

light, and, in some instances, the B-Reg. light on the 1401 console panel are turned on.

#### FEEDING FAILURE

Card feeding failures are checked in the 1402. Any type of feeding failure in the read section stops card feeding at the end of that cycle, and turns ON the reader stop light on the 1402 and the reader light on the 1401 console panel.

### Card Punch Operations

The punch section of the IBM 1402 Card Read-Punch has a rated speed of 250 cards per minute. Cards feed through the punch section 12-edge first, face down, and pass a blank station, a punch station, and a punch-check station in that order.

#### Punch (4) or Punch and Branch (4 III)

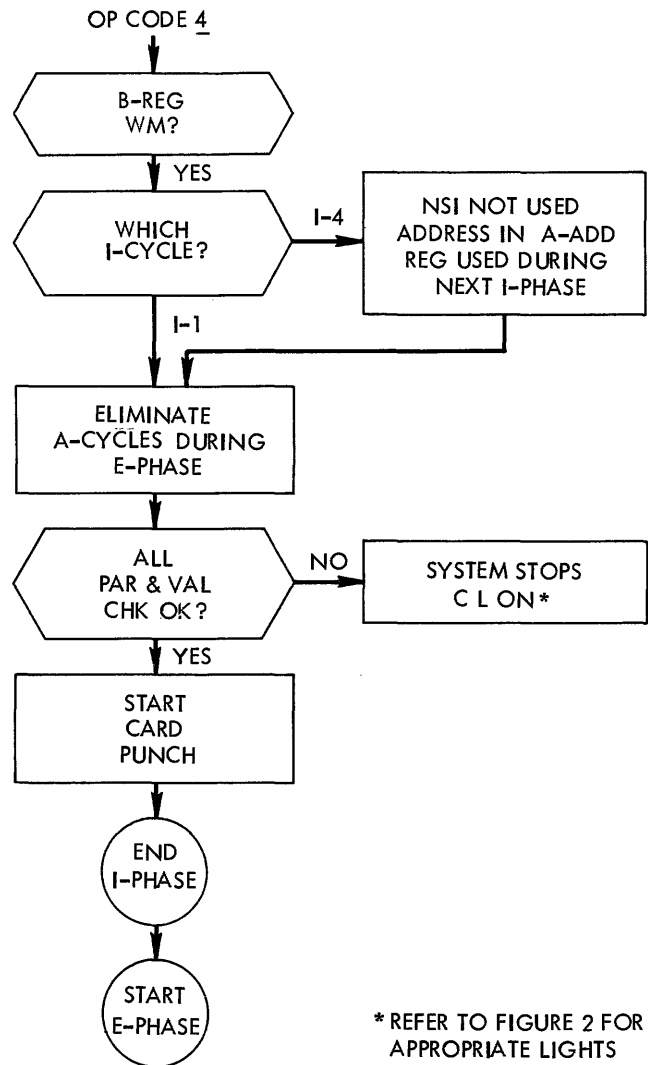
The stored program that initiates the data transfer between the 1401 and the punch section of the 1402 is the PUNCH (4) instruction. To execute an instruction other than the instruction that is next in sequence to the PUNCH instruction, a PUNCH AND BRANCH (4 III) instruction is given. The regular punch operation takes place, but the address of the next instruction is taken from the A-Add. Reg. instead of the I-Add. Reg. The circuitry that accomplishes this is set up before the I-phase of the punch operation ends.

#### I-PHASE SET-UP OPERATIONS (FIGURE 45)

Some instructions require alterations of the normal E-phase operations for correct execution of the instruction. These alterations are set up before the I-phase ends. During certain I-cycles the Op code is checked to see if it is one of the Op codes that has to set up the alteration of the normal E-phase operations before the actual instruction execution begins.

When the B-register word mark is sensed during the I-phase of a punch operation, the specific I-cycle must be established. If the I-cycle is I-1, then the instruction is a PUNCH instruction and the next sequential instruction will be executed after the punch operation E-phase. If the I-cycle is I-4, then the address of the next instruction will be the address in the A-Add. Reg. The address in the I-Add. Reg. will be ignored.

The circuitry is also set up to eliminate A-cycles during the E-phase portion of the operation. The appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check light on the console panel. If no check condition occurs, the card punch is started. The I-phase ends and the E-phase begins.



\* REFER TO FIGURE 2 FOR APPROPRIATE LIGHTS

Figure 45. Punch or Punch and Branch Instruction: I-Phase Set-up Operations

#### E-PHASE OPERATION (FIGURE 46: 1 AND 2 OF TWO)

Once the card punch is started, it remains operating long enough to feed a card, previously located between the blank station and the punch station, past the punch station.

The card passes through the punch station 12-edge first. Therefore, the first holes punched will be the 12-zone holes. Previous to the punching of the 12-zone holes, and prior to the punching of all the other holes, certain preparatory operations must be performed. These operations are numbered in the text and are illustrated in Figure 46 (1 and 2 of two). A card-punch operation is performed and the sequence of operations are listed by number with any necessary clarification. (The various interrogations performed are not listed.)

#### 12-Digit Time.

- 1 The card-punch feed mechanism is energized and starts moving the card through the punch station. However, all necessary set-up operations are completed before the card is in position for 12-digit punching.

- 2 The punch-scan circuitry is energized. This, in turn, energizes the other controlling circuitry.
- 3 The address in the STAR is set to 100.
- 4 When the punch-scan operation being performed is a 12-digit time punch-scan operation, a B-cycle A is executed. The objective of B-cycle A is to place a digit zero in the core-storage position 100.
- 5 The contents of core-storage position 100 are read into the B-register, where they are displayed in BCD form.
- 6 A zero is generated and read back into core-storage position 100, replacing the B-register contents.
- 7 A C-bit is added to maintain odd-bit parity.
- 8 Another B-cycle (B) is started. The objective of this B-cycle is to place the BCD equivalent of a 12-digit in core-storage position 100.
- 9 The contents of core-storage position 100 (a zero) are moved into the B-register, where they are displayed in BCD form.
- 10 A B-bit is generated and placed in the A-register.
- 11 The circuitry needed to set up a zone carry condition is activated.
- 12 The A-register contents (a B-bit) is added to the B-register contents (a zero) plus a carry.
- 13 The addition result (a B- and an A-bit) is stored in core-storage position 100.
- 14 A C-bit is added to maintain odd-bit parity.
- 15 Another B-cycle (C) is started. The objective of this B-cycle is to transfer the contents of core-storage position 100 into the A-register and then get ready to transfer the 12-digit information located in the core-storage punch area to the card punch.
- 16 The contents of core-storage position 100 (C-, B-, and an A-bit) are moved into the B-register, where they are displayed in BCD form.
- 17 The B-register contents are read back into the core-storage position from which they came (100).
- 18 At this time, the B-register contents are also transferred into the A-register.
- 19 The address in the B-Add. Reg. is increased by one (101).
- 20 Another B-cycle (D) is started. The objective of this B-cycle is to check each punch-area core position for a BCD-coded 12-zone, and then transferring the information (when one is found) so that a 12-zone digit can be punched in the corresponding card column.
- 21 The address in the B-Add. Reg. is transferred to the STAR and the B-Add. Reg. key-light on the console panel is turned on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).
- 22 The STAR activates the lines that cause the specified core-storage position to read out to the B-register, where the contents of that position are displayed in BCD form.
- 23 The B-register contents are read back into the core-storage position from which they came. This is done so that the character will be available during the following punch-scan operations.
- 24 The A-register contents (only the A- and the B-bits) are compared against the B-register contents. If the B-register contains an A- and a B-bit, the punch position in the card punch corresponding to the core-storage punch-area position is set up so that a 12-zone hole is punched in that column. An A- and a B-bit in core-storage position 108 (eighth position of the punch area) sets up the punching of a 12-zone hole in column 8 of the card. (No action is taken when the comparison is unequal.)
- 25 If the STAR address is not 180, the address in the B-Add. Reg. is increased by one.
- 26 The appropriate parity checks are made. Any check condition turns on the appropriate check lights on the 1402 and on the 1401 console panel.
- 27 This B-cycle D ends and another B-cycle D begins. All 80 positions of the punch area (core-storage positions 101 through 180) are tested sequentially.
- 28 As soon as all 80 positions of the punch area have been tested (STAR address is 180), the address in the B-Add. Reg. is increased by one.
- 29 The appropriate parity checks are made. Any check condition turns on the appropriate check lights on the 1402 and on the 1401 console panel.
- 30 The 12-digit time punch-scan operation ends.
- 31 The 12-zone holes are now punched in the card. Every punch position that was set up to punch will now have a 12-zone hole in that card column.
- 32 The punch-position circuitry and the punch mechanics are restored to their normal state.
- 33 The 11-digit punch-scan operation can now take place.
- 11-Digit Time.*
- 2 The punch-scan circuitry for an 11-digit punch-scan operation is energized.
- 3 The address in the STAR is set to 100.
- 8 Because this is an 11-digit punch-scan operation, a B-cycle B is started. The objective of this B-cycle is to place the BCD equivalent of an 11-digit in core-storage position 100.
- 9 The contents of core-storage position 100 (a C-, B-, and an A-bit) are moved into the B-register, where they are displayed in BCD form.
- 10 A B-bit is generated and placed in the A-register.
- 11 The circuitry needed to set up a zone-carry condition is activated.
- 12 The A-register zone contents (a B-bit) are added to the B-register zone contents (an A- and a B-bit; C-bit is ignored) plus a carry.
- 13 The addition result (a B-bit) is stored in core-storage position 100.
- 14 No C-bit is needed to maintain odd-bit parity.
- 15-33 The rest of the 11-digit punch-scan operation occurs exactly as described in the 12-digit punch-scan operation.
- 0-Digit Time.*
- 2 The punch-scan circuitry for a 0-digit punch-scan operation is energized.
- 3 The address in the STAR is set to 100.
- 8 Because this is a 0-digit punch-scan operation, a B-cycle B is started. The objective of this B-cycle is to place the BCD equivalent of the 0-digit in core-storage position 100.
- 9 The contents of core-storage position 100 (a B-bit) are moved into the B-register, where they are displayed in BCD form.

- 10 A B-bit is generated and placed in the A-register.
- 11 The circuitry needed to set up a zone-carry condition is activated.
- 12 The A-register zone contents (a B-bit) are added to the B-register zone contents (a B-bit) plus a carry.
- 13 The addition result (an A-bit) is stored in core-storage position 100.
- 14 No C-bit is needed to maintain odd-bit parity.
- 15-33 The rest of the 0-digit punch-scan operation occurs exactly as described in the 12-digit punch-scan operation. (A numeric zero with a BCD code of an 8- and a 2-bit is forced to compare as equal with the A-bit in the A-register.)

**1- Through 9-Digit Time.**

- 2 The punch-scan circuitry for the specific digit punch-scan operation is energized.
- 3 The address in the STAR is set to 100.
- 34 During 1- through 9-digit time, the contents of core-storage position 100 (an A-bit during 1-digit time) are moved into the B-register, where they are displayed in BCD form.
- 35 The A-register is set to blank (a C-bit).
- 36 The circuitry to set up a digit carry is activated.
- 37 During 1-digit time, the A-register numeric contents (blank; C-bit is ignored) is added to the B-register numeric contents (blank; A-bit is ignored) plus a digit carry.
- 38 The addition result (a 1-bit during 1-digit time) is read back into core-storage position 100. During each following digit time, an additional one is added.
- 14 During 1-digit time, a C-bit is not needed.
- 15-32 The rest of digits 1 through 9 punch-scan operations occur exactly as described in the 12-digit punch-scan operation.

*Punch Scan Following 9-Digit Time Punch Scan.* An additional punch scan is executed following the 9-digit time punch scan. This additional punch scan is used to complete the checking operation on the card that has just been read by the punch-check brushes.

This punch scan occurs exactly as described in the 1- through 9-digit time punch-scan operation. The result of steps 37, 38, 14-18 place a zero (8-, 2-, and C-bit) in the A-register. This zero blocks the further setting up of the punching circuitry.

- 28 As soon as the STAR address is 180, the address in B-Add. Reg. is increased by one.
- 29 The appropriate parity checks are made. Any check condition turns on the appropriate check lights on the console panel.
- 39 If no parity check condition occurred during the punch operation, and this is the punch-scan following the 9-digit time punch-scan, the punch operation ends. The E-phase ends and the I-phase of the next instruction begins.
- 40 If a parity check condition occurred during the punch operation, and this is the punch-scan following the 9-digit time punch-scan, the punch-check light on the 1402 and the punch light on the 1401 console panel are turned on. (Any B-register parity check occurring during the punch operation forces a punch-check condition.)

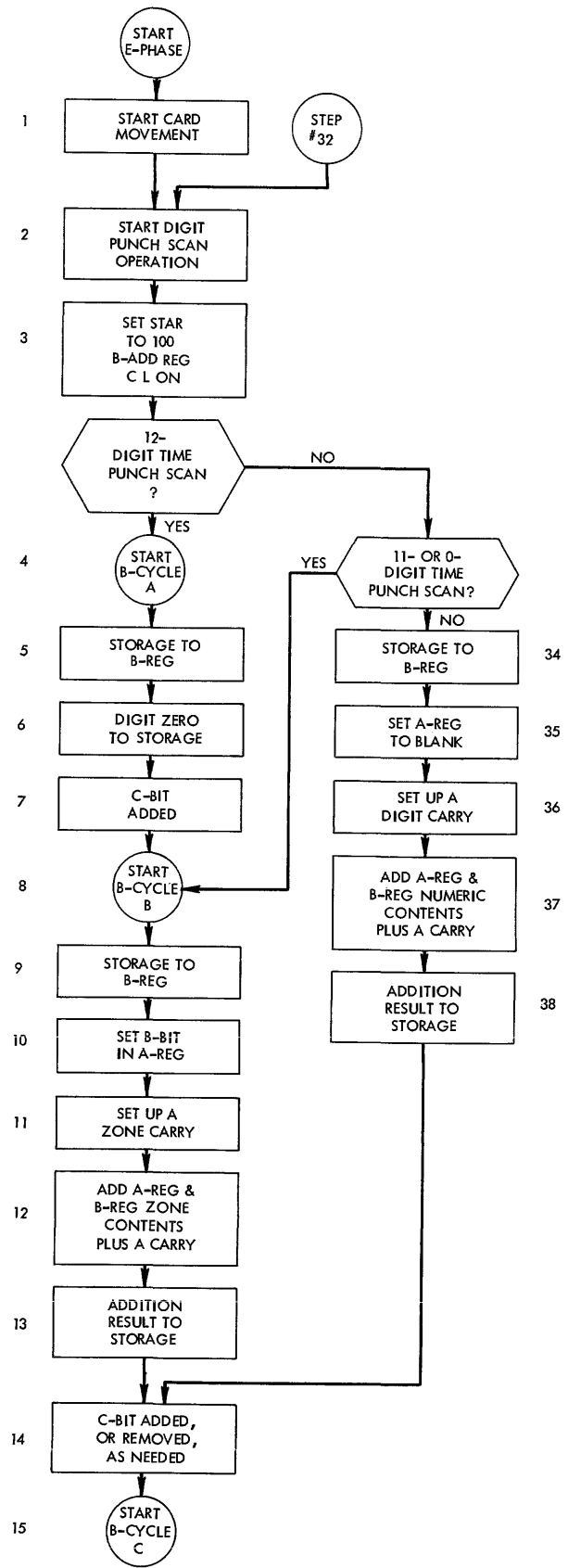
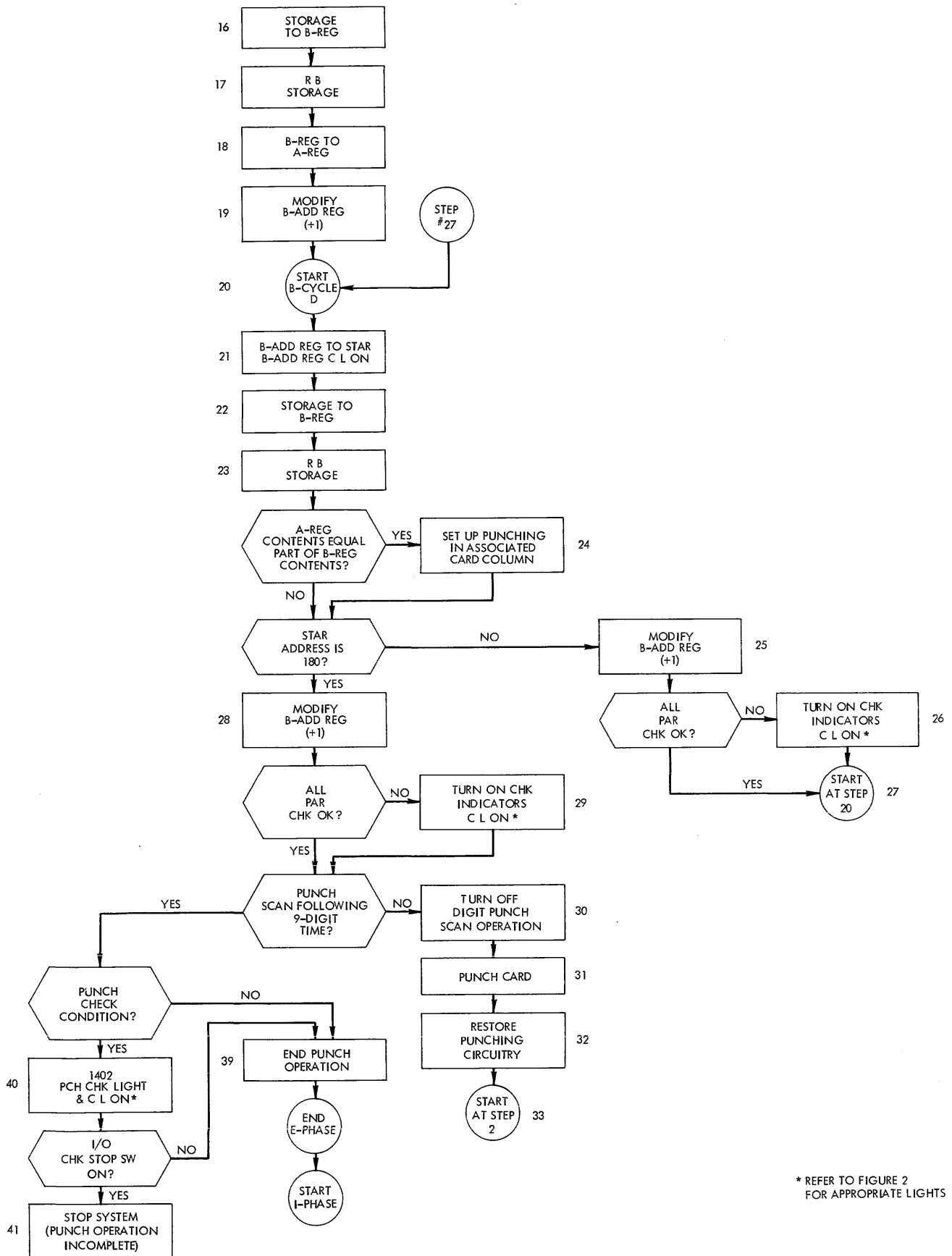


Figure 46 (1 of 2). Punch or Punch and Branch Instruction: E-Phase Operation





\* REFER TO FIGURE 2 FOR APPROPRIATE LIGHTS

Figure 46 (2 of 2). Punch or Punch and Branch Instruction: E-Phase Operation

At this point, the completion of the operation depends on the I/O check-stop switch setting. If the I/O check-stop switch is OFF, the punch operation ends (step 38). The E-phase ends and the I-phase of the next instruction begins. If the I/O check stop switch is ON, step 41 is initiated.

41 With the I/O check-stop switch set ON, the system stops (punch operation incomplete). The card that was punched during this operation can be removed from the 1402 punch section by a non-process run-out operation. The card can be repunched by placing it, and the other cards, back in the punch hopper and pressing the check-reset key and the start key.

No provision is made, however, for repunching or rechecking the card that was punched during the preceding punch operation and checked during this punch operation.

### Card Punch Check Conditions (Figure 47)

There are certain check conditions that can occur during a 1402 punch operation. These check conditions are:

1. Hole-count check
2. A-register contents check
3. Punch parity
4. Feeding failure

#### HOLE-COUNT CHECK

The punch hole-count check detects punching errors that occur in the punch section of the IBM 1402 Card Read-Punch.

The punch hole-count check is a comparison of the holes set up to punch in each card column at the punch station against the punched holes read from each card column, of the same card, at the punch-check station. An unequal comparison stops the system and turns on the punch-check light on the 1402 and the punch light on the 1401 console panel.

The hole-count check is performed by two sets of two check planes. Each plane contains 80 cores. The check for any one card makes use of one set of these check planes.

The hole-count check for any one card starts with the 12-digit time punch-scan at the punch station, and is based on the number of holes that should be punched for each character stored in the core-storage punch area. The check cores, corresponding to each card column that should be punched with a zone punch, are *flipped* during the 12-digit punch-scan operation. The check cores, corresponding to each card column that should be punched with a digit (1-9) punch, are *flipped* during the 11-digit punch-scan operation. The check cores, corresponding to each card column that should be punched with a 3-hole character, are *flipped* during the 0-digit punch-scan operation. This hole-count check does not depend on the functioning of the punch operation comparison circuits in any way, and the check is completed at the end of the 0-digit punch-scan operation.

The holes punched in the card at the punching station are now read by a set of brushes at the punch-check station. This information is sent to the previously-specified set of check planes, and each plane accepts the same information that it accepted when the card was at the punch station.

All cores in a specified check plane are at zero when a check operation starts. Any check-plane core that is *flipped* a certain number of times when the card is at the punch station, should be *flipped* the same number of times when the card is at the punch-check station. Because of this, all cores in the specified check planes should be set to zero at the end of a card-punch cycle. If any of the cores are set to one, it means that the card was incorrectly punched at the punch station.

#### A-REGISTER CONTENTS CHECK

During the punch-scan that follows the 9-digit punch-scan, the A-register is checked to see if it contains an 8-, and a 2-bit (BCD bit code for a zero; C-bit is not

TYPE OF CHECK	MACHINE STOPS (I/O CHECK STOP SWITCH ON CONSOLE PANEL ON)	LIGHTS ON WHEN STOPPED	RESET BY	REMARKS
Hole-Count Check A-Register Contents Check	At end of feed cycle	Punch Check (1402) Punch (1401)	Check Reset Key (1402)	Card in the NP stacker is in error. Cards must be run out before the check reset key is effective.
Parity	At end of feed cycle	Punch (1401) Process (1401) B-Reg (1401) Check Reset (1401)	Check Reset Key (1401)	Card in the NP stacker may be in error. Next card in the punch feed (card just punched) may be in error. Cards must be run out before the check reset key is effective.
Feeding Failure	At end of feed cycle	Punch Stop (1402) Punch (1401)	Check Reset Key (1402)	Cards must be run out before the check reset key is effective.

Figure 47. Card Punch Check Conditions

checked). If the A-register does not contain these bits, the system stops and the 1402 punch-check light and the 1401 console panel punch light are turned on.

**PUNCH PARITY**

The information to be punched is parity-checked in the B-register. An even-bit configuration turns on the punch light, the process light, the B-register light, and the check-reset key-light on the 1401 console panel. Also, the 1402 is stopped at the end of that punch cycle.

**FEEDING FAILURE**

Card feeding failures are checked in the 1402. Any type of feeding failure in the punch section stops card feeding at the end of that cycle, and turns on the punch-stop light on the 1402 and the punch light on the 1401 console panel.

**Printer Operations**

It is possible to attach either model of the IBM 1403 Printer to the 1401. Model 1 has 100 print positions and model 2 has 132 print positions. Either model can print up to 600 lines per minute. The actual printing speed realized is governed by the stored-program instructions.

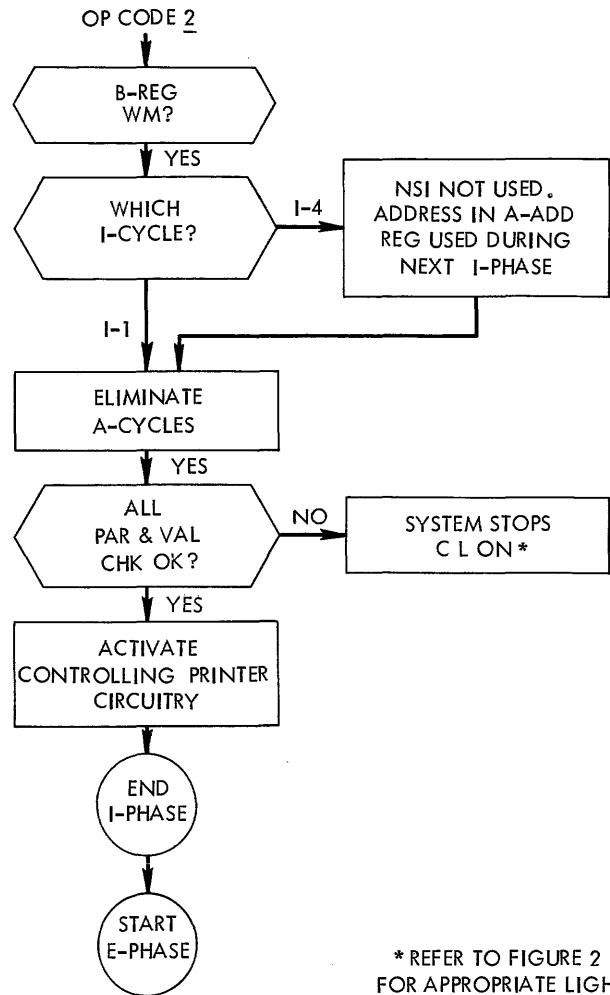
**Print (2) or Print and Branch (2 III)**

The stored program that initiates the data transfer between the 1401 and the 1403 is the PRINT (2) instruction. To execute an instruction other than the instruction that is next in sequence to the PRINT instruction, a PRINT AND BRANCH (2 III) instruction is given. The regular print operation takes place, but the address of the next instruction is taken from the A-Add. Reg. instead of the I-Add. Reg. The circuitry that accomplishes this is set up before the I-phase of the print operation ends.

**I-PHASE SET-UP OPERATIONS (FIGURE 48)**

Some instructions require alterations of the normal E-phase operations for correct execution of the instruction. These alterations are set up before the I-phase ends. During certain I-cycles the Op code is checked to see if it is one of the Op codes that has to set up the alteration of the normal E-phase operations before the actual instruction execution begins.

When the B-register word mark is sensed during the I-phase of a print operation, the specific I-cycle must be established. If the I-cycle is I-1, the instruction is a PRINT instruction and the next sequential instruction is executed after the print operation E-phase. If the I-cycle is I-4, the address of the next instruction is the address in the A-Add. Reg. The address in the I-Add. Reg. is ignored.



\* REFER TO FIGURE 2 FOR APPROPRIATE LIGHTS

Figure 48. Print or Print and Branch Instruction. I-Phase Set-up Operations

The circuitry is also set-up to eliminate A-cycles during the E-phase portion of the operation. The appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check light on the console panel. If no check condition occurs, some of the printer's controlling circuitry is activated. The I-phase ends and the E-phase begins.

**E-PHASE OPERATION**

During E-phase, each one of the 132 print positions (Model 2) is given an option to print each one of the 48 characters that make up the alphabet on the chain. This is accomplished by executing a series of 48 print-scan operations. During each scan operation, each print position is given an option to print a character. As soon as one print-scan ends, another print-scan begins. During each print-scan, each print position is given an option to print some other character. These print-scans continue until each print position has been given the option to print each one of the 48 characters that make up the alphabet on the chain.

Each print-scan is divided into three minor divisions, called print subscans. During print subscan (PSS) 1, the print positions that have an option to print are position one, followed by every third position (4, 7, — 127, 130). During PSS 2, the print positions that have an option to print are position two, followed by every third position (5, 8, — 128, 131). During PSS 3, the print positions that have an option to print are position three, followed by every third position (6, 9, — 129, 132). This sequence of three print subscans is repeated 48 times.

The mechanical units required to do the actual printing are the hammer unit and a chain, or type, array. There is one hammer for each print position, and the printing is accomplished when the hammer forces the paper and the ribbon against the type array.

The type array spans the entire length of the print line and contains five identical alphabets of 48 characters each, including numbers and special characters. Within each alphabet, the characters are grouped in sequence by a modified BCD character bit value. This arrangement simplifies character identification as the type becomes aligned with the various print positions.

The operations that take place during print-scan 1 and subsequent print-scans are numbered in the text and are illustrated in Figure 49. (The various interrogations performed are not listed.)

*Print-Scan 1.* During print-scan 1, every print position has an option to print one character. Print subscan 1 is executed first, followed by print subscans 2 and 3.

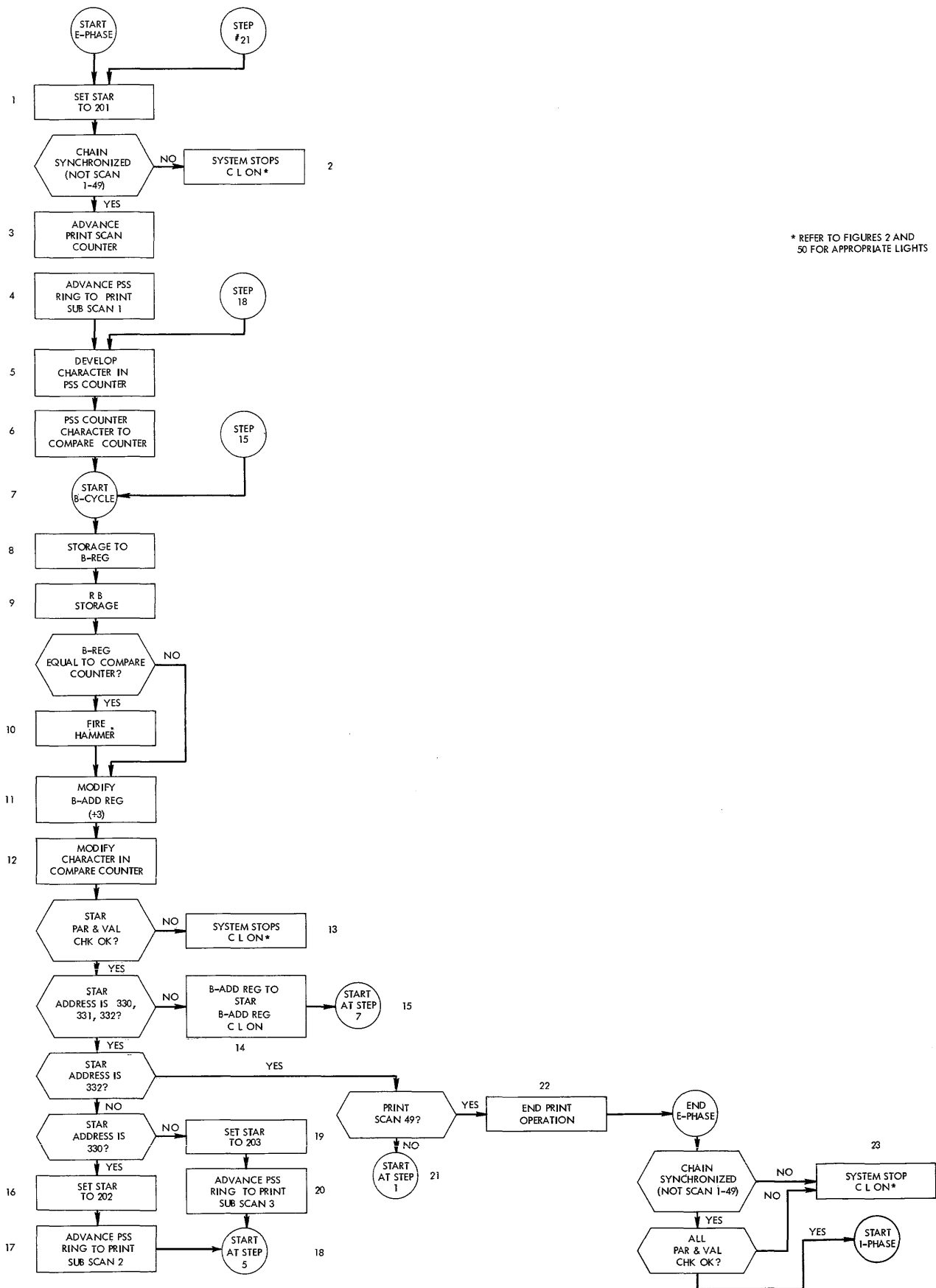
#### *Print Subscan 1*

- 1 The address in the STAR is set to 201. This is the first position of the specified printer output area in core storage. This output area starts at core-storage position 201, and extends 100 positions to core-storage position 300 or 132 positions to core-storage position 332.
- 2 The chain is checked to see if it is in synchronism with the 1401 and all the controlling circuitry. If it is not in synchronism, the system stops and the appropriate check lights turn on.
- 3 If the chain is in synchronism, the total in the print-scan counter is increased by one. The print-scan counter is used to keep count of the print-scans. There will be 49 print-scans. The additional print-scan is necessary to complete checks made during the printing operation.
- 4 The print subscan (PSS) ring is advanced to print subscan 1. The PSS ring is a closed 3-position ring that specifies which one of the three print subscans is active. Print-scan 1 always starts out with PSS 1.
- 5 The PSS counter generates the character that will be opposite print-position 1 when that print position has an option to print.
- 6 The PSS counter character (actually the modified BCD bit value of the character) is transferred to the compare counter.
- 7 A B-cycle is started. The objective of this and the following B-cycles is to read out the printer output core-storage positions for printing.

- 8 The STAR activates the lines that cause the specified core-storage position (201 set up in step 1) to read out to the B-register, where the contents of that position are displayed in BCD form.
- 9 The B-register contents are read back into the core-storage position from which they came. This is done so that the character is present when the right print option occurs.
- 10 If the B-register character is the same as the compare-counter character, the hammer opposite print-position 1 is activated, and a character is printed in print-position 1. If the characters are not equal, no printing takes place.
- 11 A STAR to B-Add. Reg. address transfer takes place. The address in the STAR is modified by plus three and transferred into the B-Add. Reg.
- 12 The compare-counter character is modified so that it represents the character that can be printed at print-position 4. The alphabets on the type array are sequentially arranged so that a compare-counter modification of plus two results in the character that can be printed at print-position 4.
- 13 A parity and validity check of the STAR is made. Any check condition stops the system and turns on the appropriate check lights on the 1401 console panel.
- 14 If no check condition occurs, the STAR is checked to see if the address in it is 330, 331, or 332. If the STAR address is not 330, 331, or 332, this B-cycle ends. The address in the B-Add. Reg. is transferred to the STAR and the B-Add. Reg. key-light on the console remains on. The STAR lights on the console panel display the core-storage position being addressed (in BCD form).
- 15 Another B-cycle is started. During these 11.5  $\mu$ s B-cycles, every third core-storage position in the printer output area is compared against the character positioned to print at the corresponding print position. If core-storage position 207 contains the character K, and a K is positioned to print in print-position 7 (when core-storage position 207 is addressed), the K will print. The B-cycles continue until the address in the STAR is 330. This signifies that all the print positions controlled by PSS 1 have had one option to print and that PSS 2 will be next.
- 16 The address in the STAR is set to 202. During print subscan 2, core-storage position 202 and every third position following it in the printer output area are read out to the B-register.
- 17 The PSS ring is advanced to PSS 2.

#### *Print Subscan 2*

- 18 Print subscan 2 starts (start at step 5).
- 5 The PSS counter generates the character that will be opposite print-position 2 when that print position has an option to print.
- 6–15 During PSS 2, steps 6 through 15 occur exactly as described in PSS 1. The PSS 2 B-cycles continue until the address in the STAR is 331. This signifies that all the print positions controlled by PSS 2 have had an option to print and that PSS 3 will be next.
- 19 The address in the STAR is set to 203. During print subscan 3, core-storage position 203 and every third position following it in the printer output area will be read out to the B-register.
- 20 The PSS ring is advanced to PSS 3.



\* REFER TO FIGURES 2 AND 50 FOR APPROPRIATE LIGHTS

Figure 49. Print or Print and Branch Instruction: E-Phase Operation

### Print Subscan 3

- 18 Print subscan 3 starts (start at step 5).
- 5 The PSS counter generates the character that will be opposite print-position 3 when that print position has an option to print.
- 6-15 During PSS 3, steps 6 through 15 occur exactly as described in PSS 1.
- 21 The PSS 3 B-cycles continue until the address in the STAR is 332. This signifies that all the print positions controlled by PSS 3 have had an option to print, and that this print-scan operation should end. If this is not print-scan 49, then another print-scan is initiated.

*Print-Scan 2 Through 48.* Print-scans 2 through 48 and their associated print subscans are executed exactly as described in print-scan 1.

*Print-Scan 49.* Print-scan 49, steps 1 through 20, occur exactly as described in print scan 1.

- 22 When the address in the STAR becomes 332 during print-scan 49, the print operation and the E-phase end.
- 23 The chain is checked to see if it is in synchronism with the 1401 and all the controlling circuitry. If it is not in synchronism, the system stops and the appropriate check lights turn on.  
The appropriate parity and validity checks are made. Any check condition stops the system and turns on the appropriate check lights on the 1403 and on the 1401 console panel.  
If no check condition occurs, the I-phase of the next instruction begins.

### Print Word Marks

When a lozenge (◻) is used as an operation modifier character (d-character) in any type of print instruction, only word marks are printed. The word marks associated with the core-storage positions in the printer output area print as ones in the corresponding print positions. This type of print operation is executed exactly the same as a normal print operation.

### Printer Check Conditions (Figure 50)

There are certain check conditions that can occur during a 1403 print operation. These check conditions are:

1. Print parity
2. Type synchronization
3. Hammer fire—print compare
4. Print-line complete

#### PRINT PARITY

The information to be printed is parity-checked in the B-register. An even-bit configuration turns on the process light, the B-register light, and the check-reset key-light on the 1401 console panel. Also, the 1403 is stopped at the end of the print operation if the check-stop switch on the auxiliary console is set ON.

#### TYPE SYNCHRONIZATION

This check operation insures that the 1401 is in synchronism with the 1403 chain. Circuitry is activated

when a character 1 on the print chain is in position to be printed in print-position one. This circuitry is checked against the PSS ring and other controlling circuitry to be certain that everything is in synchronism. Any discrepancy during a print operation stops the machine at the end of the print operation and turns on the SYNC CHECK light on the 1403 and the printer light on the 1401 console panel. If the discrepancy is detected before the first print subscan, the machine stops before the printing starts. This check operation is effective only during an actual print operation.

#### HAMMER FIRE—PRINT COMPARE

This circuit is designed to detect two conditions:

1. Hammer fails to fire after being energized.
2. Hammer fires without being energized.

Either of these conditions stops the machine at the end of that print operation if the check-stop switch on the auxiliary console is set ON. The printer light on the 1401 console panel and the print-check light on the 1403 are turned on.

The two conditions are detected by using two additional core-storage areas of 100 or 132 positions (depends on the 1403 model). One area is called the hammer-fire area and the other is called the print-compare area. The individual cores are addressed at the same time the corresponding cores in the printer output area are addressed.

The hammer-fire core is set to 1 each time the corresponding core-storage position in the printer output area is addressed. If the hammer fires the core, it is set back to zero.

The print-compare core setting depends on the result of the comparison between the B-register character and the compare-counter character. If there is no equal comparison, the print-compare core that corresponds to the addressed printer output core-storage position is set to zero. If there is an equal comparison, the core is set to 1.

The individual core settings are checked during the following print-scan as each printer output area core-storage position is addressed. If the core settings are both the same, an error is indicated. A zero setting in both cores indicates that the hammer received an erroneous fire impulse. A 1 setting in both cores indicates that the hammer failed to fire.

Also, either one of these conditions sets a core in a separate core-storage area called print-error storage. The core that is set corresponds to the addressed core-storage position being printed. In effect, this core stores the core-storage position that caused the check condition, and indicates that location on a storage-scan operation.

TYPE OF CHECK	MACHINE STOPS (I/O CHECK STOP SWITCH ON CONSOLE PANEL ON)	LIGHTS ON WHEN STOPPED	RESET BY	REMARKS
Parity	At end of print operation	Process (1401) B-Reg (1401) Check Reset (1401)	Check Reset Key (1401)	
Type Synchronization	Before printing takes place or at end of print operation	Printer (1401) Sync Check (1403)	Check Reset Key (1403) or I/O Check Reset Switch (1401 Aux Cons)	If discrepancy is detected prior to the first print sub scan, the machine stops before the printing starts.
Hammer Fire- Print Compare	At end of print operation	Printer (1401) Print Check (1403)	Check Reset Key (1403) or I/O Check Reset Switch (1401 Aux Cons)	Sets a print error storage core.
Print-Line Complete	At end of print operation	Printer (1401) Print Check (1403)	Check Reset Key (1403) or I/O Check Reset Switch (1401 Aux Cons)	Sets a print error storage core.

Figure 50. Printer Check Conditions

#### PRINT-LINE COMPLETE

Another separate core-storage area, called print-line complete, is provided to determine whether each printer output area core-storage position that contained printable information actually had an opportunity to print. A printer output area core-storage position that contains information to be printed, but does not get an opportunity to print it (no B-register—compare-counter equal comparison), initiates a check condition. This check condition stops the machine at the end of the print operation if the check-stop switch on the auxiliary console is set ON and turns on the 1401 console panel printer light and the 1403 print check light.

Each individual print-line-complete core corresponds to a core-storage position in the printer output area, and is set to zero at the beginning of a print operation. During the print operation the cores are set to 1:

1. when the corresponding core-storage position does not have a printable character (This is accomplished during the first print-scan and the 1 setting is regenerated until test time, print-scan 49.)
2. when the corresponding core-storage position that contains a printable character has an opportunity to print (a B-register—compare-counter equal comparison).

Any core that is not set to 1 by print-scan 49 test time indicates that the corresponding core-storage position had a printable character, but did not have an opportunity to print it.

Also, this condition sets a core in a separate core-storage area called print-error storage. The core that is set corresponds to the addressed core-storage position that did not have an opportunity to print its character.

#### Print-Error Storage Area

The print-error storage cores *store* the printer output core-storage position address that caused a check condition when it printed. These cores correspond to, and are addressed simultaneously with, the printer output area core-storage positions. The print-error storage cores can be set to 1 during any print-scan, except print-scan 1. The check conditions that can set the core to 1 are:

1. sensing a hammer-fire core setting at 1 and the corresponding print-compare core, also set at 1
2. sensing a hammer-fire core setting at 0 and the corresponding print-compare core, also set at 0
3. sensing a print-line-complete check condition
4. sensing a print-compare condition in combination with a 1 from the corresponding print-line-complete core.

The print-error storage cores are reset to zero during print-scan 1, and they can be set to 1 during any of the remaining 48 print-scans. Once the core is set to 1, the 1 setting is regenerated for the remaining print-scans. It is not reset to 0 until the first print-scan of the next print operation, or until a storage-scan operation is executed.

During a storage-scan operation, any core that is set to 1 stops the system at that location. This provides the location of the core-storage position that caused a check condition when it printed.

# Index

1-Character Instruction	7	I-8 Cycle	15
2-Character Instruction	7, 9	I-Address Register (I-Add. Reg.)	3
5-Character Instruction	13	I-Cycles	6
7-Character Instruction	15	I-Op Cycle	6
8-Character Instruction	15	I-(Instruction) Phase	6
		Inhibit Drive	1
		Input-Output Operations	49
		Input-Output Units	4
		Instruction Execution (E-Phase)	15
		Instruction Reading (I-Phase)	5
		Instruction-Length Lights	3
A-Address Register (A-Add. Reg.)	3	Load ( <u>L</u> AAA BBB)	43
A-Cycle	16	Logic Area	3
A-Register	3	Logic Operations	18
Abbreviations	5		
Active Components, I-Phase	6	Magnetic-Core Storage	1
Add ( <u>A</u> AAA BBB)	24	Manual-Address Switches	4
Add-to-Storage Logic	1	MLP Code	52
Add, Single Field ( <u>A</u> AAA)	32	Mode Switch	4
Address-Modification Area	3	Move ( <u>M</u> AAA BBB)	37
Addressable Core Storage	1	Move Characters and Edit ( <u>E</u> AAA BBB)	44
Arithmetic Operations	24	Move Characters and Suppress Zeros ( <u>Z</u> AAA BBB)	38
		Move Digit ( <u>D</u> AAA BBB)	41
		Move Zone ( <u>Y</u> AAA BBB)	42
B-Address Register (B-Add. Reg.)	3	Next Sequential Instruction (NSI)	6
B-Cycle	17	No Operation ( <u>N</u> )	20
B-Register	1	NSI (Next Sequential Instruction)	6
Branch ( <u>B</u> III)	18		
Branch if Character Equal ( <u>B</u> III BBB d)	19	Op-Register	3
Branch if Indicator On ( <u>B</u> III d)	18	Operation Modifier Character (d-character)	10
Branch if Word Mark and/or Zone ( <u>V</u> III BBB d)	19		
Branch Instruction	11	Parity and Validity Checking	4
		Positive Logic	5
		Print ( <u>2</u> )	59
Card Read Check Conditions	53	Print and Branch ( <u>2</u> III)	59
Card Read Operations	49	Print Scans	59
Card Punch Check Conditions	58	Print Subscans	60
Card Punch Operations	54	Print Word Marks	62
Chaining Operation	7	Print-Error Storage Area	63
Clear Storage ( <u>L</u> AAA)	34	Printer Check Conditions	62
Clear Storage and Branch ( <u>L</u> III BBB)	35	Printer Operations	59
Clear Word Mark ( <u>L</u> AAA BBB) ( <u>L</u> AAA)	35	Punch ( <u>4</u> )	54
Clear, Move, Load, and Word Mark Operations	34	Punch and Branch ( <u>4</u> III)	54
Compare ( <u>C</u> AAA BBB)	20	Programming Error	15
Complement-Add Operation	28		
Core Storage	1	Read ( <u>1</u> )	49
		Read and Branch ( <u>1</u> III)	49
		Recomplement Operation	30
Data Flow	1	Set Word Mark ( <u>2</u> AAA BBB) ( <u>2</u> AAA)	15, 35
Data-Flow Diagrams	5	Storage Address Register (STAR)	3
Data Lines	1	Subtract ( <u>S</u> AAA BBB)	24
		Subtract, Single Field ( <u>S</u> AAA)	32
		Symbols	5
E-(Execute) Phase	15	True-Add Operation	26
E-Phase Termination	17		
		Zero and Add ( <u>?</u> AAA BBB)	32
		Zero and Subtract ( <u>!</u> AAA BBB)	32
Halt ( <u>a</u> )	21		
Halt and Branch ( <u>a</u> III)	21		
I-1 Cycle	7		
I-2 Cycle	8		
I-3 Cycle	10		
I-4 Cycle	10		
I-5 Cycle	13		
I-6 Cycle	13		
I-7 Cycle	15		

**IBM**

International Business Machines Corporation  
Data Processing Division  
112 East Post Road, White Plains, New York

Printed in U. S. A.

G24-1477-0

620575CVO