Part 2: Analysis of a Typical Small Computer

Chapter 11

General Description of the PDC 808

The Computer Automation PDC 808 is one of the easiest small machines to understand. Therefore, it serves as an admirable vehicle for acquiring an understanding of the fundamental concepts behind all computers. The PDC 808 was chosen for several reasons, not the least of which was CAI's willingness to divulge what is normally held as closely guarded proprietary information. The material that follows is doubly valuable since the logic diagrams are practically impossible to come by without actually going to work for a computer company. This also holds true for the reference and maintenance data which applies in whole or in part to all general purpose computers.

BLOCK DIAGRAM

Terminology used in the PDC 808 requires some amplification. Earlier the various necessary computer registers were discussed in some detail. It was also mentioned that few if any machines use the same names for these registers and the PDC 808 is no exception to this.

One of the first and most important registers named was the accumulator which holds the results of all arithmetic operations. In the PDC 808, the accumulator register is called the A register. It is fed by the A bus and it feeds the S bus. The next register of importance is the instruction register and in the PDC 808 it is called the I register.

The program counter or instruction location counter is made up of two parts in the 808. Since the word length is only eight bits, the addresses that can be referenced by one word are 000 to 377 octal or 000 to 255 decimal. For this reason, the storage in the 808 is broken up into pages. With a 4K memory module, there are 16 decimal or 20 octal pages. For this reason, the L counter specifies a word in a page and the P counter specifies the page currently being referenced.

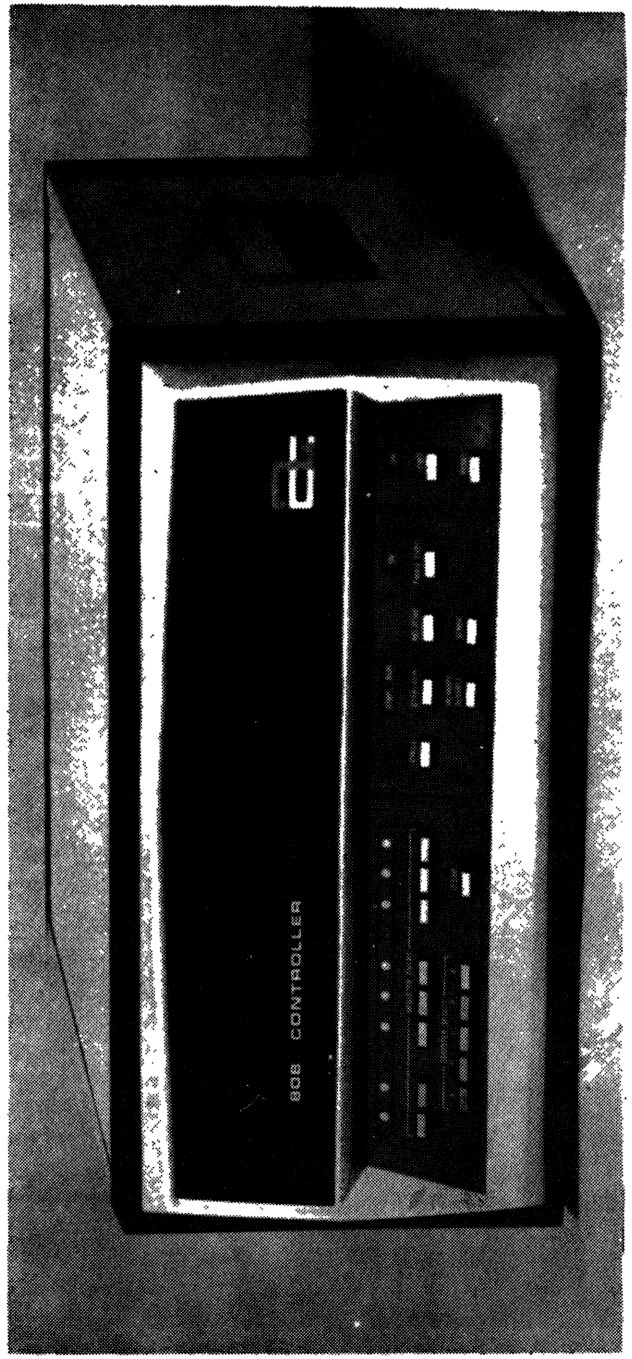


Fig. 11-1. Front view of the PDC 808.

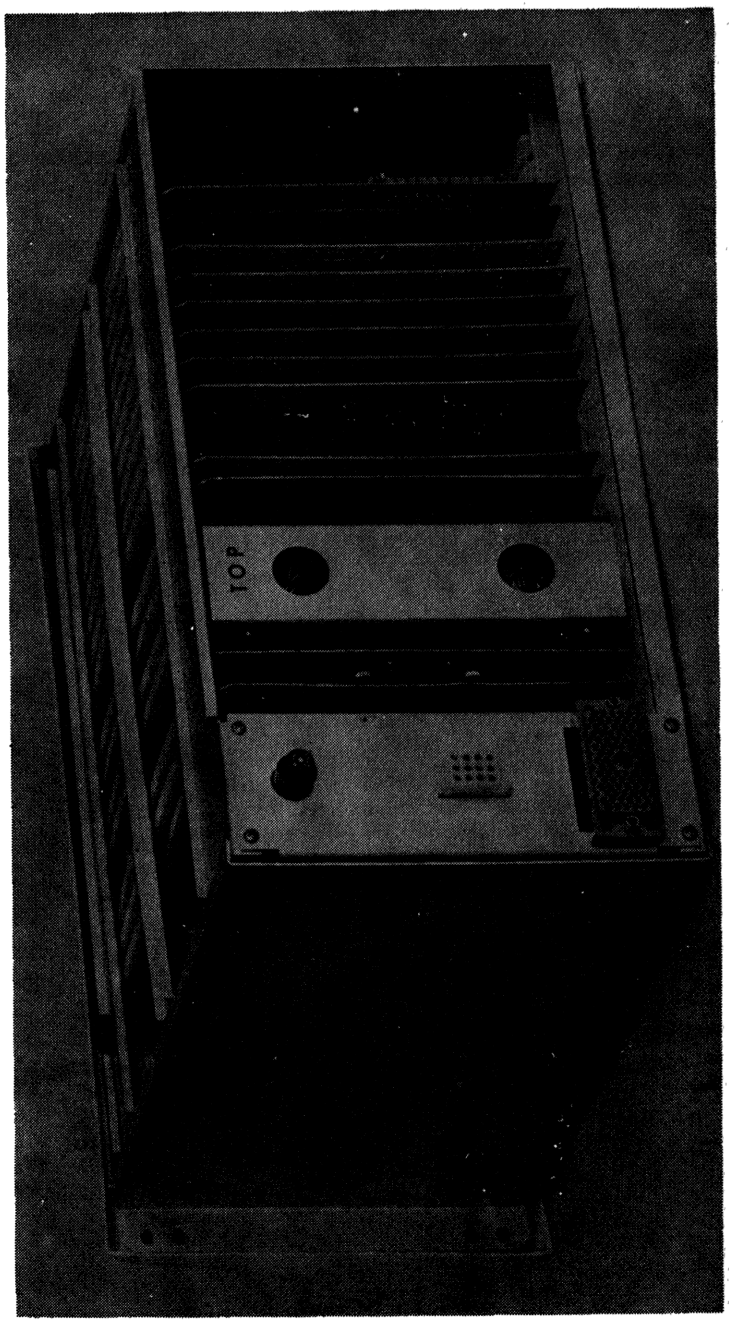


Fig. 11-2. Rear view of the typical PDC 208 and 808.

The M register in the 808 corresponds to the address register or memory address register and is used to store the location within a page during a memory reference instruction. The ME register holds the page information. Finally, the W register in the PDC 808 corresponds to the memory data register in our original discussion.

The PDC 808 and 208 computers (see Figs. 11-1 and 11-2) are digital stored-program units that utilize integrated circuits and a 3D core memory for simplicity and reliability. Parallel organization and over 70 instructions provide general-purpose computer power and flexibility.

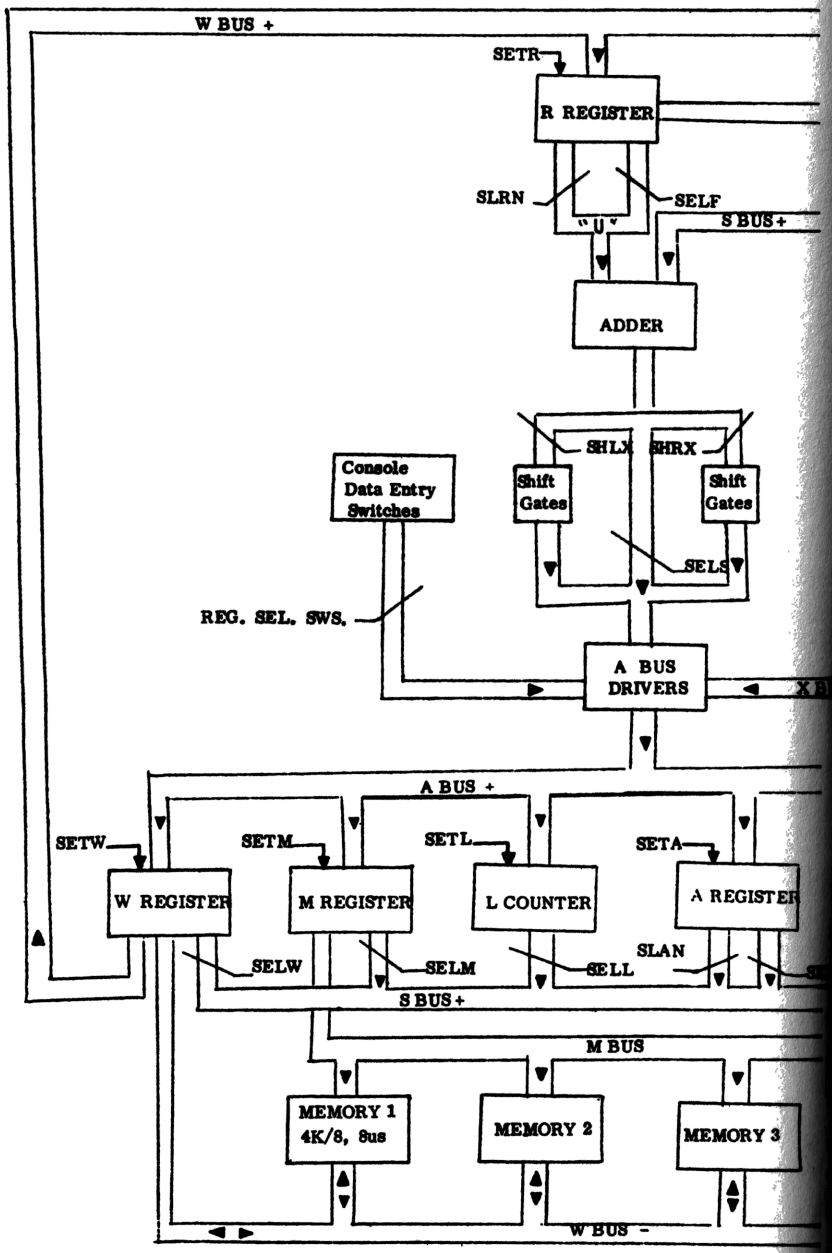
The 208 and 808 computers contain eight hardware registers, the adder unit and the control section. Refer to Fig. 11-3. The adder is an 8-bit, parallel-add, serial-carry unit utilizing complex function TTL integrated circuits. Two 8-bit words are presented to the adder unit via the S and U buses. The sum appears on the A bus, which distributes it to the W, M, L, P and A registers. The sum is then strobed into the desired register by a set or load pulse. Control of the adder can be shifted left or right one place by selection gates between the adder and the A bus.

There are seven 8-bit registers and one 6-bit register in the basic processor. Information is transmitted within the controller to and from registers via several 8-bit transmission buses. One bit of a typical bus structure is shown in Fig. 11-4. The information on the buses is under the control of the processor control logic. By placing the contents of one register on a bus, it may be transferred to another bus by routing it through the various buses and-or adder unit until it appears as an input to one or more registers and then strobing it into the desired register.

The W register is an 8-bit register that interfaces the processor to the data circuits in memory. Data read from or stored in memory is held in the W register during the memory cycle. The W register is always cleared at the beginning of a memory cycle. If data is to be stored in memory, it is strobed into W halfway through the cycle. If data is being read from memory, it is placed in W approximately one fourth of the way through the memory cycle and held there during the restore portion (last half) of the cycle.

The M register is an 8-bit register that interfaces the processor to the lower eight bits of the memory address circuits. Address information is stored in the M register at the beginning of the memory cycle where it is held throughout the cycle.

The ME register is an extension to the M register which provides 16K addressing capability. The timing of the ME



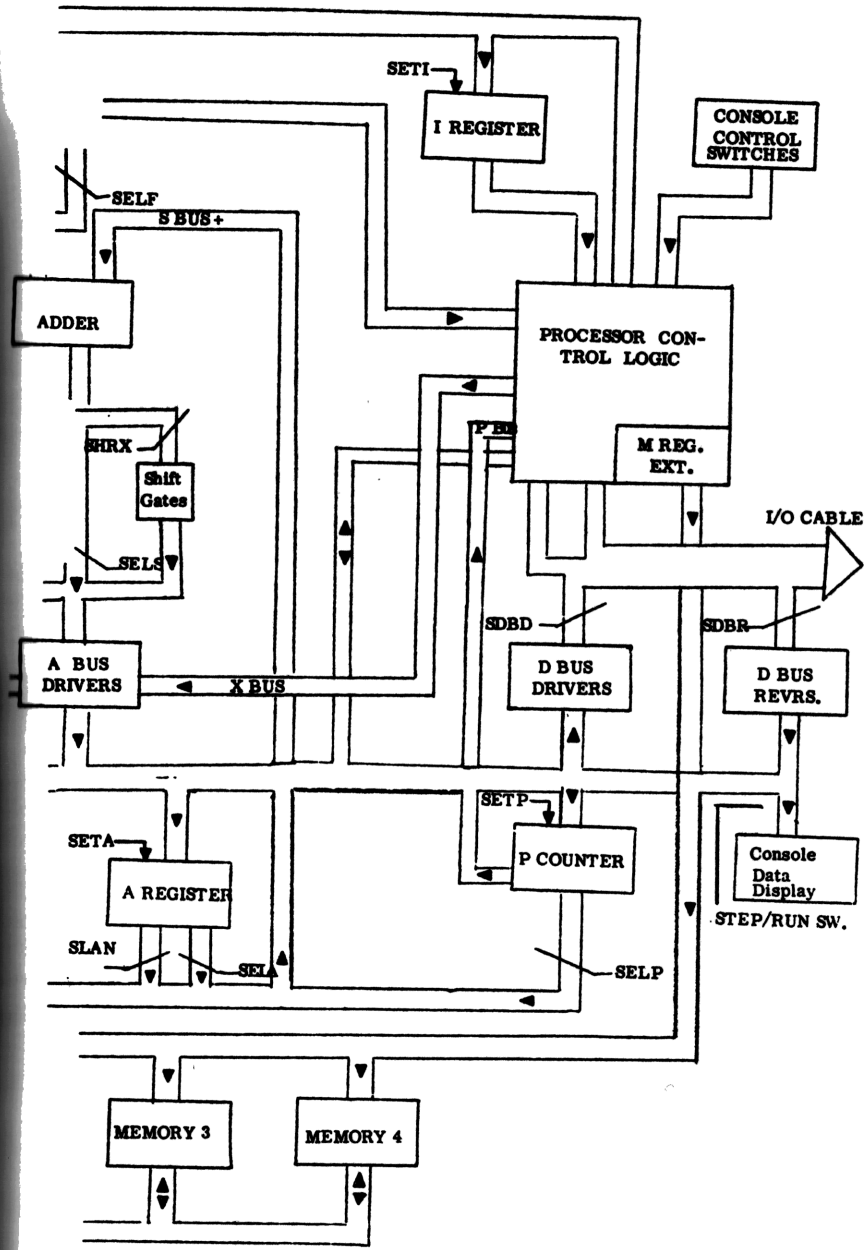


Fig. 11-3. Block diagram of the organizational registers.

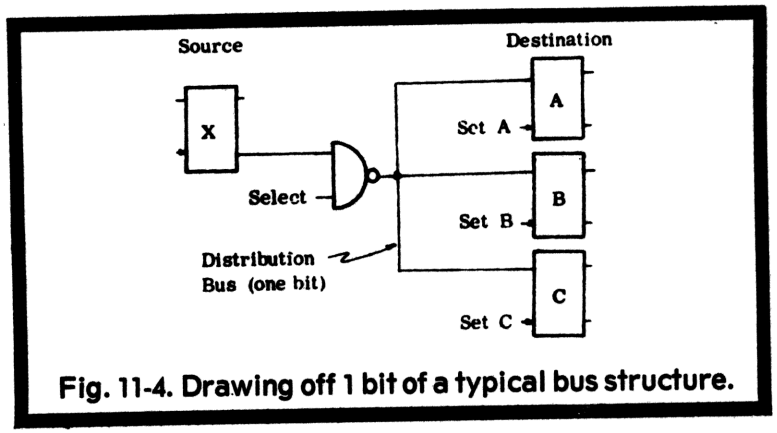


Fig. 11-4. Drawing off 1 bit of a typical bus structure.

register is identical with that of the M register. The ME register contains the page number of the memory page being accessed.

The L counter is an 8-bit register that is used as the lower portion of a 16-bit program counter. It is used to hold the word location of the next instruction word in the program.

The P counter in the 808 is an 8-bit register that is used as the upper portion of a 16-bit program counter. It is used to hold the page number of the next instruction; that is, the P counter defines the page and the L counter the word within that page where the next instruction word in the program is to be obtained. The two counters serve as a program sequencer. Notice that the upper two bits of the P counter are never used in a 16K-word machine. Thus, only 14 bits, eight of L and six of P, are used and loaded into the M and ME registers. In the 208, the P counter is a ripple counter that serves the same function.

The A register is an 8-bit hardware accumulator for arithmetic operations and serves as a word buffer register for programmed data transfers to or from I-O devices.

The I register is an 8-bit register that holds the first word of an instruction.

The R register is an 8-bit register that holds the second word of a double-word instruction or the operand for memory reference instructions. It is a time-shared register. For instance, during the execution of add, the second instruction word is held in R until the operand is obtained from memory. The operand then replaces the A field of the instruction during the actual add operation.

The basic computer memory is a conventional 4-wire, 3-D core memory described earlier. The basic configuration is 4096 words of eight bits and expansion is in 4096-word blocks up to 16K words. A power fail-safe option is available to prevent

loss of memory as power collapses due to a primary power failure or a power turn-off. If low power is detected, an interrupt occurs which allows entry into a subroutine to effect an orderly halt to operations.

MEMORY ADDRESSING

The memory is a random access type requiring 14 bits of address (for 16K). The 14-bit address is supplied to the memory from two registers in the processor—the 8-bit M register and the 6-bit ME (M extension) register. Both registers hold the address for the duration of the memory cycle.

The memory is divided into pages of 256 words each. The L counter is used to specify one of the 256 words within a page, and the P counter specifies the page. The L and P counters together constitute the program counter, or sequencer, and specify the memory address of the next instruction in the program.

In addition to obtaining instructions from memory, operands (data) can be retrieved or stored in memory by the memory reference instructions and the block I-O instructions. The memory address of the operands must be specified by the instruction (direct addressing) or by referring to the contents of another pair of locations in memory (indirect addressing) or by a combination of both methods (block transfer uses a pointer in memory and part of the instruction word).

Direct addressing allows direct access to 512 words of memory: 256 in "page zero" (the first 256 memory words) and 256 in the "current page" (the page in which the instruction is located). Thus if the operating program is in page 4, all the words in page 0 and page 4 may be accessed directly.

To reach operands outside the current page or page 0, indirect addressing is used. The operand address is stored in two words located in page 0 or the current page. The instruction then specifies the first of these locations and also specifies indirect addressing. The contents of the two locations are then used to recall or store the operand. The first location specifies the address within the page and the second location specifies the page.

Multi-level indirect addressing is accomplished by setting the most significant bit of the indirect page address. The computer, after retrieving both indirect pointers (for L and P), examines MSB of the second pointer and uses the contents as a new pointer of an address if the MSB is a 1. If the MSB is a 0, the contents are used as the address of the operand.

The block I-O instructions can access a 2K-word block of memory (the second 2K of the first 4K memory). A combination of direct and indirect addressing is employed. Two pointers (one for input and one for output) are located in page 0. The contents of one of the pointers and three bits of the instruction word are used to form an 11-bit effective address, which is used to retrieve or store data with a block transfer instruction. (A 12th bit is added to the address to force the address into the upper 2K of the first 4K memory.)

When studying the instruction SET for any processor it is well to remember a few things. Each instruction is nothing more than a binary number stored in memory that will be called to the instruction register and decoded to cause the computer to perform some specified operation.

Looking at an instruction format, then (such as the one that follows), is nothing more than looking at the contents of the I register. In the case of memory reference instructions (instructions that have an address associated with them), there is an associated A or address field contained in the R register. In the condensed instruction list, on the other hand, these instructions are listed in octal! For this reason the following material contains both the binary format as it would appear in the I register and the octal listing as well.

Taking as an example, the add instruction; if the instruction is a direct page 0 instruction bit, position 0 would have a 0 in it and bit position 1 (A) would have a 0 so the octal notation would be 300. If bit position 0 (P) contained a 1, indicating the current page, the instruction would become 301; page 0, indirect, 302, and current page, indirect, 303.

Chapter 12

Processor Instructions

There are four classes or groupings of instructions defined by the first two bits of each instruction. Two classes are double-word instructions: memory reference and I-O instructions, and two memory cycles are required to establish the complete instruction. Skips, register change, and control instructions (register change and control instructions are in the same class) are all single-word instructions.

MEMORY REFERENCE

The format of the memory reference instructions is shown in Fig. 12-1. The first instruction word contains the class, op code, addressing mode, and page information. The second word contains an 8-bit address—the A field—which defines a location within a page.

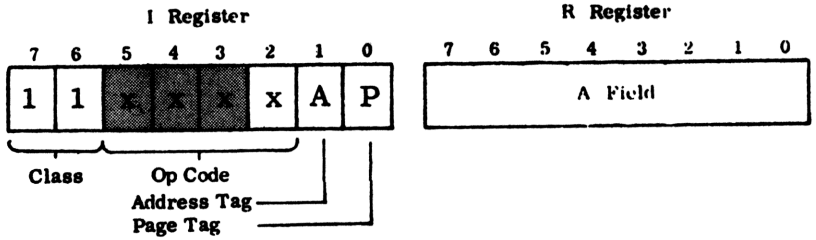


Fig. 12-1. Memory reference instruction format.

ADD

The instruction in Fig. 12-2 adds the contents of the effective memory location to the contents of the A register.

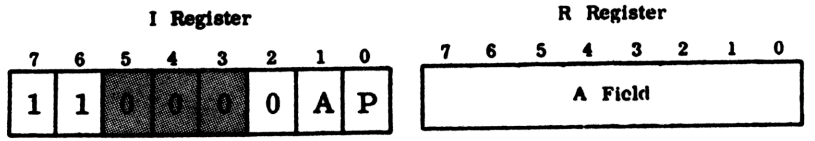


Fig. 12-2. Memory add instructions.

IOR: Inclusive OR

The instruction in Fig. 12-3 inclusively ORs the contents of the effective memory location to the contents of the A register.

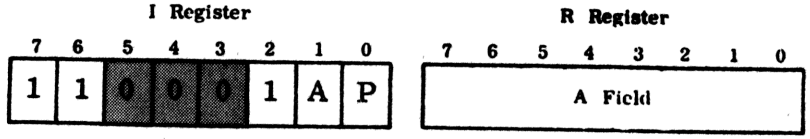


Fig. 12-3. Memory inclusive OR instructions.

SUB: Subtract

Fig. 12-4's instruction subtracts the contents of the effective memory location from the contents of the A register.

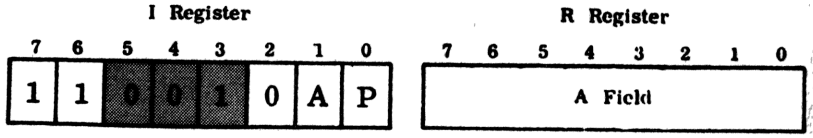


Fig. 12-4. Memory subtract instructions.

LDA: Load A

The instruction in Fig. 12-5 loads the contents of the effective memory location into the A register.

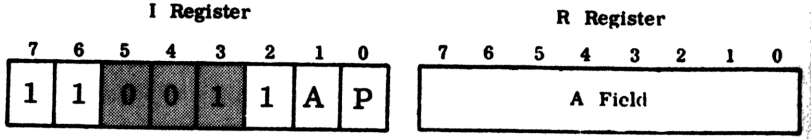


Fig. 12-5. Load A memory instructions.

STA: Store A

The contents of the A register is stored in the effective memory location with the instruction in Fig. 12-6.

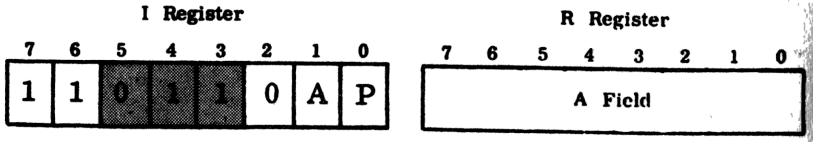


Fig. 12-6. Store A memory instructions.

ANA: AND to A

The instruction in Fig. 12-7 logically ANDs the contents of the effective memory location and the contents of the A register. Result appears in the A register.

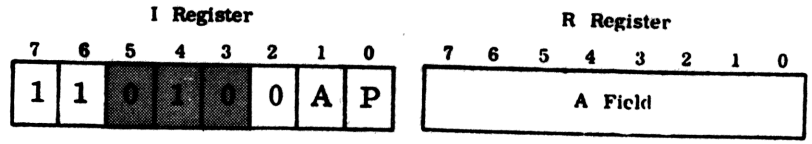


Fig. 12-7. AND to A memory instructions.

IMA: Interchange A with Memory

Fig. 12-8's instruction stores the contents of the A register in the effective memory location and loads the contents of the effective memory location into the A register.

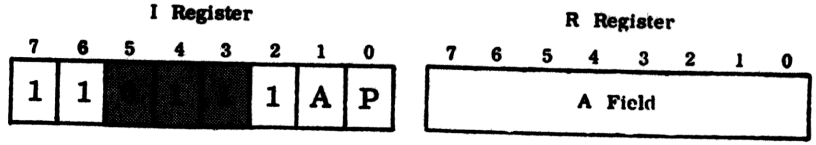


Fig. 12-8. Interchange with A memory instructions.

JMP: Jump

The instruction in Fig. 12-9 places the jump address in the program counters for a direct jump to page zero, the P counter is reset and the A field transferred to L. A jump in the current page leaves P unchanged. An indirect jump causes the two indirect address words to be placed in the L and P counters.

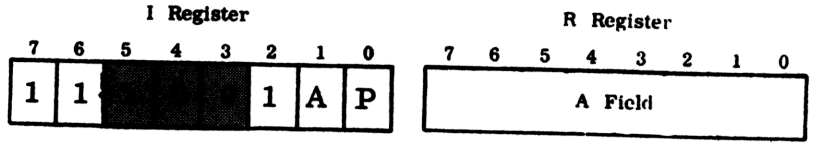


Fig. 12-9. Jump memory instructions.

IMS: Increment Memory & Skip-On-Zero Result

The contents of the effective memory location is incremented by one count and replaced with Fig. 12-10's instruction. If incrementing causes the result to become zero, a skip occurs. Overflow is set if the result of the incrementation

is 200_8. Note: Neither skip nor OV occurs if IMS is executed as an interrupt instruction.

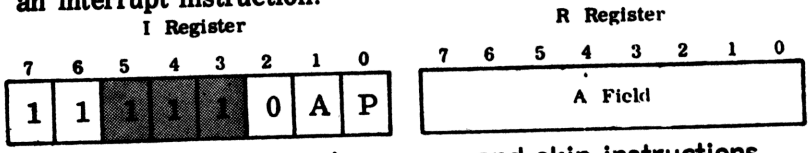


Fig. 12-10. Increment memory and skip instructions.

JST: Jump & Store, L & P Counters

With the instruction in Fig. 12-11 the contents of the L and P counters are stored in memory at the effective jump address and jump address plus one, and the counters are altered to contain jump address plus two.

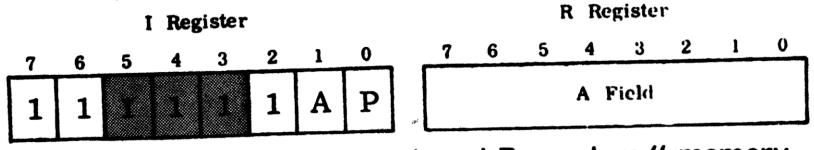


Fig. 12-11. "Jump and store, L and P counters," memory instructions.

EOR: Exclusive OR

The instruction appearing in Fig. 12-12 exclusively ORs the contents of the effective memory location to the contents of the A register in the 208.

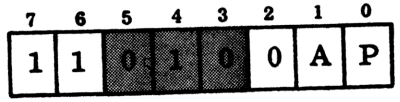


Fig. 12-12. "Exclusive OR" memory instructions, PDC208.

ADM: Add Magnitude

In the 208, Fig. 12-13's instruction adds the contents of the effective memory location to the contents of the A register, treating each value as an 8-bit unsigned integer.

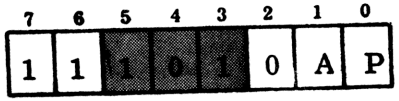


Fig. 12-13. "Add magnitude" memory instructions, PDC208.

CMS: Compare Memory and Skip If Not Equal

The instruction in Fig. 12-14 (in the 208) compares the contents of the effective memory location to the contents of the A register and skips if they are not equal. Otherwise, the next instruction is executed.

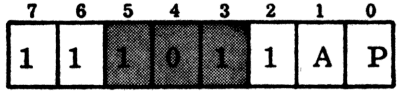


Fig. 12-14. "Compare memory and skip if not equal" memory instructions, PDC208.

SKIP INSTRUCTIONS

Instructions that test conditions within the computer and take action depending upon the results of the test fall in the skip instruction class. The term skip is derived from the action taken if a test condition is satisfied; the computer skips over the next two words in the program.

Skip is a single-word, micro-programmed instruction. The first two bits specify the skip class and the six remaining bits are used to specify the test conditions. Notice that only five of the six available bits are used, allowing 30 usable combinations for skip conditions, 15 in each of two groups.

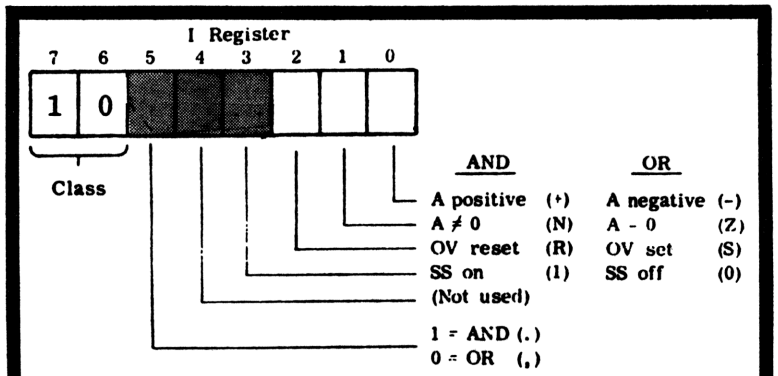


Fig. 12-15. Skip instruction format.

AND SKIP GROUP

The 15 instructions in this group are combinations of the four conditions in Fig. 12-16. If more than one condition is

specified (by placing a 1 in the condition bit) the AND of all specified conditions must be met to cause a skip. Only a few of the combinations in this group will be presented here.

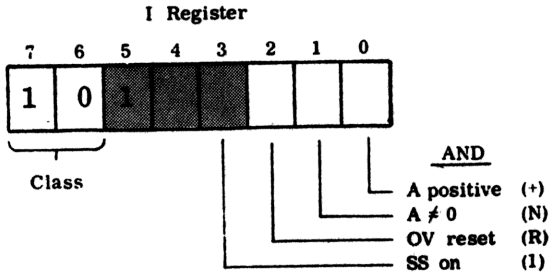


Fig. 12-16. AND skip instruction format.

IFT +: Skip If A Positive

With the conditions specified in Fig. 12-17 a skip occurs if the contents of the A register is equal to or greater than zero Otherwise, the next instruction in sequence is executed.

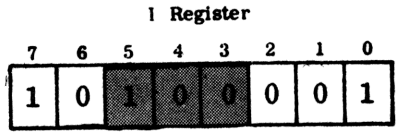


Fig. 12-17. "Skip if a positive" instruction.

IFT N: Skip If A Non-Zero

With the instructions in Fig. 12-18 a skip occurs if the contents of the A register is not zero. Otherwise, the next instruction in sequence is executed.

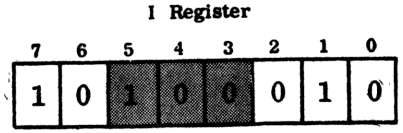


Fig. 12-18. "Skip if a non-zero" instruction.

IFT R: Skip If OV Reset

With the instruction in Fig. 12-19 a skip occurs if OV is not on. Otherwise, the next instruction sequence is executed. OV is reset.

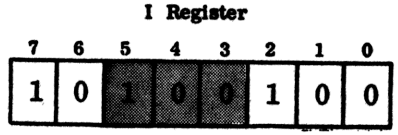


Fig. 12-19. "Skip if OV reset" instruction.

IFT 1: Skip If Sense Switch Is On

With the instruction appearing in Fig. 12-20 a skip occurs if the console sense switch is on. Otherwise, the next instruction in sequence is executed.

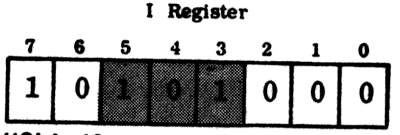


Fig. 12-20. "Skip if sense switch is on" instruction.

IFT + N: Skip If A Greater Than Zero

With Fig. 12-21's instruction a skip occurs if the contents of A register is greater than zero. Otherwise, the next instruction in sequence is executed.

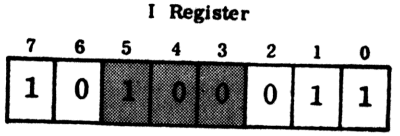


Fig. 12-21. "Skip if a greater than zero" instruction.

OR SKIP GROUP

The 15 instructions in this group are combinations of the four conditions in Fig. 12-22. If more than one condition is specified, a skip will occur if any of the specified contitions are met.

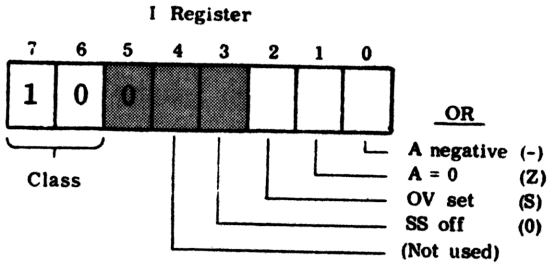


Fig. 12-22. OR skip group instruction format.

IFT: Skip If A Negative

The instruction in Fig. 12-23 causes a skip to occur if the contents of the A register is less than zero. Otherwise, the next instruction in sequence is executed.

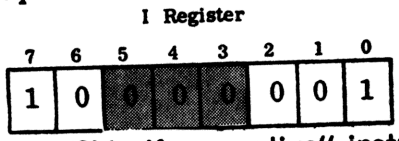


Fig. 12-23. "Skip if a negative" instruction.

IFT Z: Skip If A Zero

With Fig. 12-24's instruction a skip occurs if the contents of the A register is zero. Otherwise, the next instruction in sequence is executed.

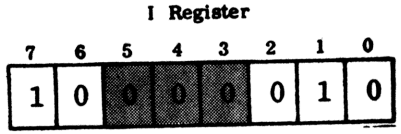


Fig. 12-24. "Skip if a zero" instruction.

IFT S: Skip If OV Set

If the instruction in Fig. 12-25 is applied a skip occurs if OV is set. (The OV will be reset at the end of the instruction). Otherwise, the next instruction in sequence is executed.

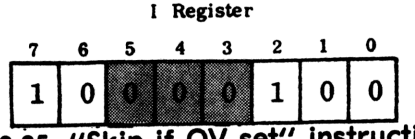


Fig. 12-25. "Skip if OV set" instruction.

IFT O: Skip If Sense Switch Off

The instruction in Fig. 12-26 tells the computer to skip if the console sense switch is off (not depressed). Otherwise, the next instruction in sequence is executed.

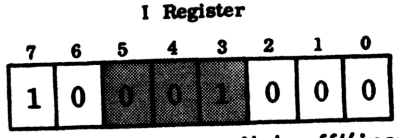


Fig. 12-26. "Skip if sense switch off" instruction.

IFT-, Z: Skip If A Equal To Or Less Than Zero

With the instruction in Fig. 12-27 skip occurs if the content of A is zero or negative. Otherwise, the next instruction sequence is executed.

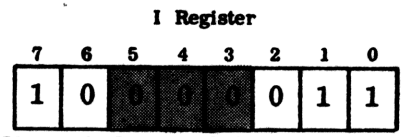


Fig. 12-27. "Skip if A equal to or less than zero" instruction.

REGISTER CHANGE

The register change class contains the control, shift, and register change instructions. All are single-word instructions.

A Register Change (ARC)

ARC Z: Clear A Register

Fig. 12-28's instruction clears the A register.

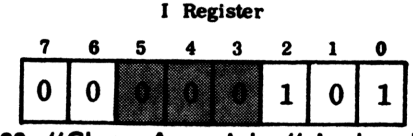


Fig. 12-28. "Clear A register" instruction.

ARC I: Increment A Register

The combination in Fig. 12-29 adds one (least significant bit) to the contents of A. OV is set if the contents go from 0, 1111111 to 1, 0000000.

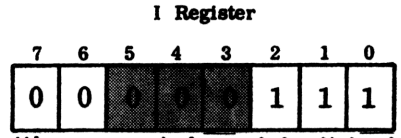


Fig. 12-29. "Increment A register" instruction.

ARC D: Decrement A Register

The combination in Fig. 12-30 subtracts one (least significant bit) from the contents of A. OV is set if the contents go from 1,0000000 to 0, 1111111.

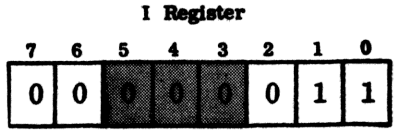


Fig. 12-30. "Decrement A register" instruction.

ARC C: 1s Complement A Register

With the instruction in Fig. 12-31 the computer performs the 1s complement (bit-by-bit inversion) of the contents of the A register.

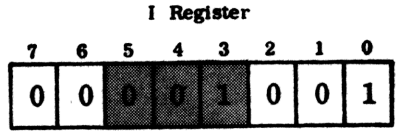


Fig. 12-31. "1s complement A register" instruction.

ARC N: 2s Complement A Register

With the instruction in Fig. 12-32 the computer performs the 2s complement of (negation of) the contents of the A register.

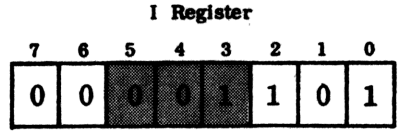


Fig. 12-32. "2s complement A register" instruction.

ARC -1: Transfer -1 To A

The instruction in Fig. 12-33 replaces the contents of the A register with 1, 1111111.

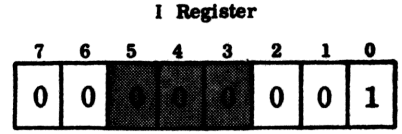


Fig. 12-33. "Transfer -1 to A" instruction.

ARC 1: Transfer +1 To A

The instruction in Fig. 12-34 replaces the contents of the A register with 0,0000001.

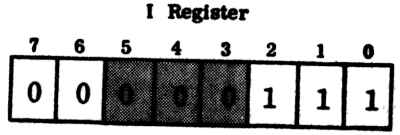


Fig. 12-34. "Transfer +1 to A" instruction.

Overflow Register Change (ORC)

ORC S: Set Overflow

Fig. 12-35's instruction sets the overflow flip-flop. If already set, there is no change.

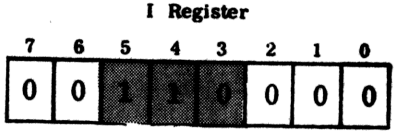


Fig. 12-35. "Set overflow" instruction.

ORC R: Reset Overflow

The instruction in Fig. 12-36 resets the overflow flip-flop.

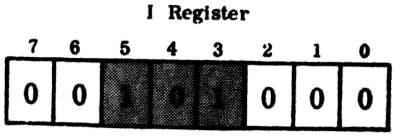


Fig. 12-36. "Reset overflow" instruction.

ORC C: Complement Overflow

The combination in Fig. 12-37 complements the content of the overflow flip-flop.

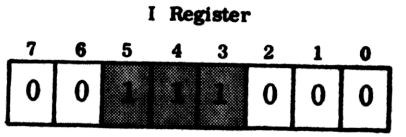


Fig. 12-37. "Complement overflow" instruction.

SHIFT INSTRUCTIONS SHF

There are three types of shift instructions: logical, rotate, and arithmetic. Each type affects only the A register and in some cases the overflow flip-flop.

Logical Shift

The logical shift instruction causes the contents of the A register to be shifted one place to the left or right. Zeros are shifted into one end of the register and the contents are shifted out and lost. Thus, if eight consecutive, logical shift left instructions were executed, the A register would contain all zeros. Note: The overflow flip-flop is reset on a logical shift instruction at the beginning of the operation and may be set at the end of the instruction.

Rotate

The rotate instruction, sometimes referred to as logical rotate, preserves the contents of the A register as it is shifted. This is accomplished by an "end around shift," but with the overflow flip-flop in the loop as shown in Fig. 12-38.

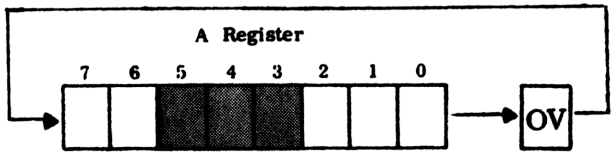


Fig. 12-38. "Rotate" or "end around shift" instruction.

For a rotate right (left), the content of the overflow flip-flop is shifted into bit 7 (bit 0) of the A register and bit 0 (bit 7) of the A register is shifted into OV. This essentially creates a 9-bit register. Thus, 9 consecutive rotate right (left) instructions leave the A register and the overflow flip-flop unchanged. Shifting A through the overflow flip-flop facilitates testing bits of A. For instance, rotate right and then skip on overflow reset tests bit 0 of the A register.

Arithmetic Shift

The arithmetic shift differs from the logical or rotate shifts in that the A register is assumed to contain a signed number. The sign bit is A7. Since shifting left multiplies by 2 and shifting right divides by 2, the sign of a number must not change. Thus an arithmetic shift can affect only the magnitude position of the number and cannot change the sign bit. The 208-808 uses 2s complement arithmetic. If two numbers of equal magnitude but opposite signs are added together, the result is zero. Not surprising. Consider two binary numbers of the same magnitude but opposite sign:

$$+14_{10} = + \ 001\ 110$$

$$-14_{10} = - \ ?$$

Using 2s complement representation and substituting 0 for + and 1 for -

$$+14_{10} = 0{,}001\ 110$$

$$-14_{10} = 1{,}110\ 010$$

Where the sign bit has been separated from the magnitude with a comma. Notice that if the two numbers are added together, the result is all zeros with a carry.

Now if +14 is divided by 2, by performing an arithmetic shift right, the result is

$$+7_{10} = 0,\ 000\ 111$$

Dividing -14 by 2 produces

$$-7_{10} = 1,\ 111\ 001$$

Which is obtained by shifting -14_{10} right one place, propagating the sign bit and not changing the sign bit. Thus arithmetic shift right causes the sign bit to be shifted while remaining unchanged. An arithmetic shift left performs a logical shift left except the sign bit is unchanged. All shifts require 1 cycle.

SHF R: Logical Shift Right

The instruction in Fig. 12-39 resets OV, shifts the contents of the A register right one place. Zero is shifted into the sign bit, and the LSB of A is shifted into OV.

Fig. 12-39. "Logical shift right" instruction.

SHF L: Logical Shift Left

The Fig. 12-40 instruction resets OV and shifts the contents of the A register left one place. Zero is shifted into the LSB of A and the sign bit of A is shifted into OV.

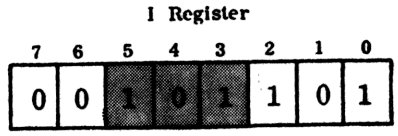


Fig. 12-40. "Logical shift left" instruction.

SHF): Rotate Right

Fig. 12-41's instruction shifts the contents of the A register right one place through OV. The LSB of A is shifted into OV and the previous contents of OV is shifted into A7.

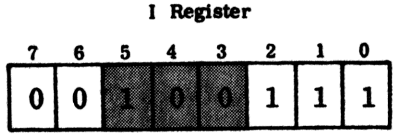


Fig. 12-41. "Rotate right" instruction.

SHF (: Rotate Left

The combination in Fig. 12-42 shifts the contents of the A register left one place through OV. The sign bit is shifted into OV. The previous contents of OV is shifted into the LSB of A.

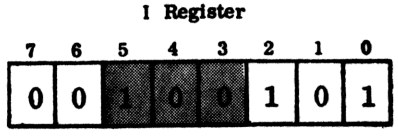


Fig. 12-42. "Rotate left" instruction.

SHF >: Arithmetic Shift Right

Fig. 12-43's combination shifts the lower seven bits of the A register right one place. The sign bit remains unchanged and is also shifted into A6. The LSB of A is lost.

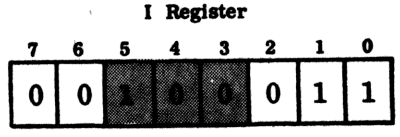


Fig. 12-43. "Arithmetic shift right" instruction.

SHF <: Arithmetic Shift Left

The instruction in Fig. 12-44 shifts the lower seven bits of the A register left one place. The sign bit remains unchanged. A6 is lost and zero is shifted into the LSB of A.

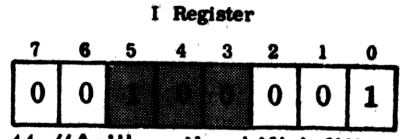


Fig. 12-44. "Arithmetic shift left" instruction.

Control

NOP: No Operation

The instruction in Fig. 12-45 causes an 8-microsecond pause in program for the 808 and a 2.67-microsecond pause for the 208

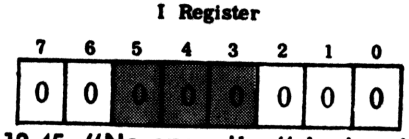


Fig. 12-45. "No operation" instruction.

HLT: Halt

The combination in 12-46 halts the computer.

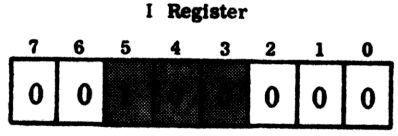


Fig. 12-46. "Halt" instruction.

EIN: Enable Interrupts

Fig. 12-47's combination enables the computer to respond to a request for interrupt.

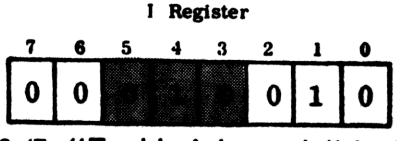


Fig. 12-47. "Enable interrupts" instruction.

DIN: Disable Interrupts

The instruction in Fig. 12-48 inhibits the computer from accepting a request for interrupt.

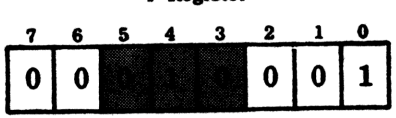


Fig. 12-48. "Disable interrupts" instruction.

Chapter 13

Input-Output Section

The 208 and 808 computers are designed for communications, control and monitoring applications. Thus, the I-O sections of the unit have capabilities usually found only in larger computers. The 13 I-O instructions provide the power and flexibility to allow the 208 and 808 to handle tasks other machines in their class cannot. Data transfers to and from the unit are 8-bits parallel. The I-O bus utilizes the party-line technique to simplify the interface system. For real time applications, the 208 and 808 have two fully implemented priority interrupt lines plus a third interrupt request line that can be used to implement additional priority interrupts.

INPUT—OUTPUT (I-O) INSTRUCTIONS

The I-O instructions allow the 208 and 808 to communicate with the "outside world"— peripheral device controllers such as the teletype controller and special logic generated by Computer Automation or the user Communication is in the form of commands which can trigger flip-flops or relays, sense signals which test the state of a flip-flop, relay, or incoming line, and data transfer operations which transfer parallel 8-bit words into or out of the computer.

All peripheral logic is connected in parallel on the I-O cable which contains the data bus, control lines, and the device address bus. Selection of the device (peripheral logic group) with which communication is desired is accomplished by assigning each device a number or address and applying this number to the device address bus in the I-O instruction. There are up to 32 device addresses available.

INSTRUCTION FORMAT

I-O instructions are double-word instructions. The first word specifies the class (I-O in this case), the operation code, and special conditions available by micro-coding. The second word specifies the device address (5 bits) of the device with

which communication is to take place and a 3-bit function code. The 3-bit function code allows up to eight functions to be created within the same device address for a given op code. For instance, at device 4, eight different flip-flops may be tested (sensed) by executing the "sense device 4" instruction eight times with the eight different function codes.

The I-O instruction involves micro-coding. The 3-bit op code defines the type of instruction — sense, external control, or transfer. The next three bits are used to place conditions on the op code (see Fig. 13-1).

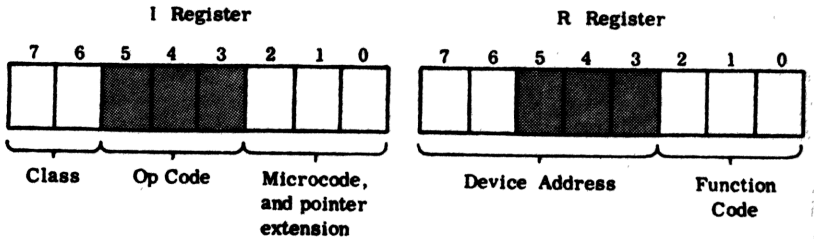


Fig. 13-1. I-O instruction format.

SENSE INSTRUCTIONS

The sense and skip instruction allows the 208 and 808 to test the state of a specified function in an I-O device. Up to eight different conditions may be tested in each device. The second word of the instruction specifies the device address and function code. The micro-code in the first word of the instruction is used to specify whether the skip is to occur on a sense response from the device being tested or on no response.

SSR: Sense and Skip on Response

The function specified in Fig. 13-2 is tested in the device specified and a skip will occur if a response is obtained. If no response occurs, the next instruction in sequence is executed.

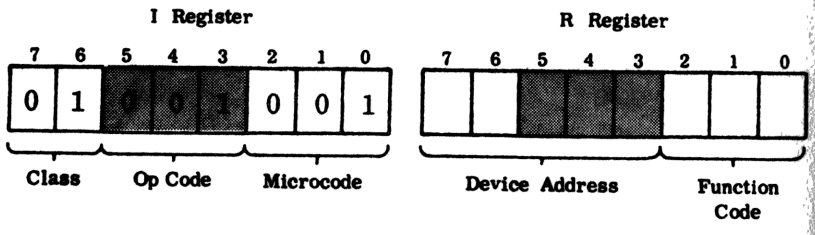


Fig. 13-2. "Sense and skip on response" instruction.

SSN: Sense and Skip on No Response

The function specified in Fig. 13-3 is tested in the device specified and a skip will occur if a response is not obtained. If a response occurs, the next instruction in sequence is executed.

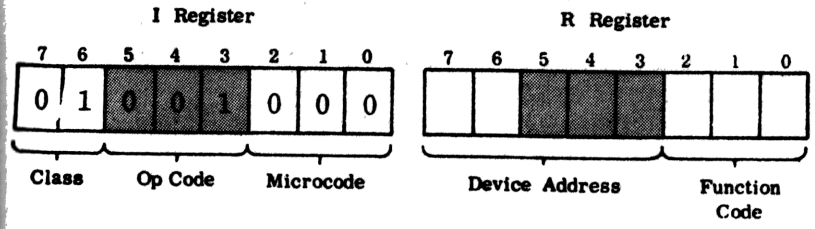


Fig. 13-3. "Sense and skip on no response" instruction.

EXTERNAL CONTROL INSTRUCTION

The external control (EXT) instruction allows the 208 and 808 to pulse a specified flip-flop, relay, or external line in a device controller or special logic connected to the I-O bus. Instructions such as start tape, rewind, clear, stop reader, are examples of uses of EXC instructions. Up to eight control functions may be implemented for each device.

EXC: External Control

The control function specified in Fig. 13-4 by the function code is executed.

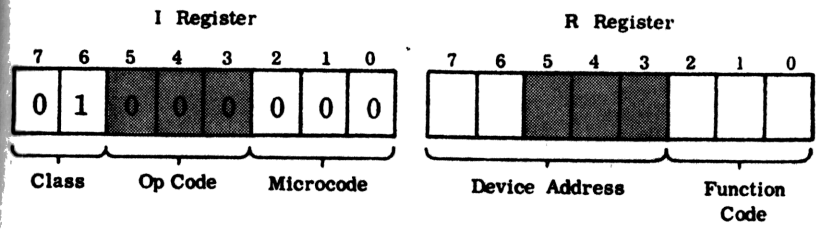


Fig. 13-4. "External control" instruction.

INPUT TO A INSTRUCTIONS

There are four I-O instructions that can cause data to be transferred from I-O devices to the A register. Two are unconditional transfers and two are transfers that are conditioned on a sense response. The latter two instructions are

actually combinations of the sense and skip instruction with the input to A instruction. All transfers are 8-bit parallel.

INA: Input To A Register (Unconditionally)

With the code in Fig. 13-5 a data word is transferred from the device specified to the A register. The function code can be used to designate the source of data within the device if multiple sources exist, as in the case of a 16-bit converter output. Up to eight sources may be specified by the function code, allowing a 64-bit input data word to be handled in eight successive inputs, greatingly simplifying the device logic.

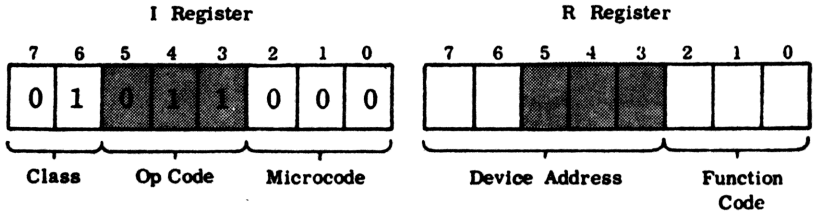


Fig. 13-5. "Input to A register (unconditionally)" instruction.

MIA: Masked Input To A Register (Unconditionally)

The masked input instruction in Fig. 13-6 causes the input data word from the specified device to be ANDed with the previous contents of the A register and the results placed in the A register. The source of the data word within the device may be specified by the function code if multiple sources exist.

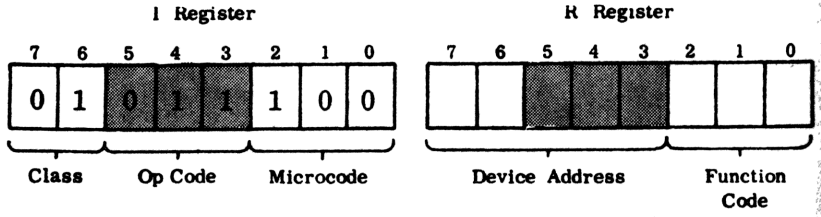


Fig. 13-6. "Masked input to A register (unconditionally)" instruction.

IAS: Input To A And Skip On Response

The IAS instruction in Fig. 13-7 first senses the function specified by the function code within the device specified. If a response is obtained, the data word is transferred to the A register and a skip occurs. If no response is obtained, the data word is not transferred, the A register is unchanged, and the next instruction in sequence is executed.

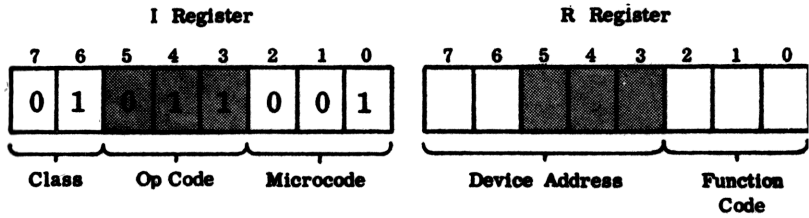


Fig. 13-7. "Input to A and skip on response" instruction.

MIS: Masked Input To A And Skip On Response

Fig. 13-8's instruction is a combination of the sense and the masked input to A. The instruction first senses the condition specified by the function code within the device specified. If a sense response is obtained, a masked input to A register and a skip occur. If no sense response is obtained, the A register is left unaltered and the next instruction in sequence is executed.

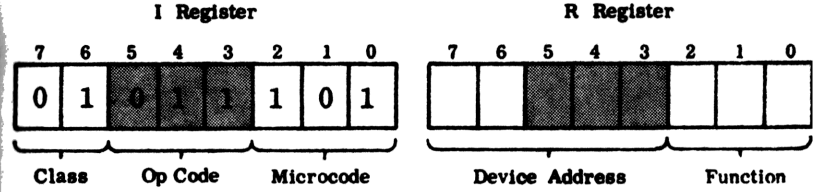


Fig. 13-8. "Masked input to A and skip on response" instruction.

OUTPUT FROM A INSTRUCTIONS

There are four I-O instructions that can cause data to be transferred from the A register to a device on the I-O bus. Two of the instructions are unconditional transfers and two are conditioned on a sense response. The latter two are actually combinations of the sense instructions with the output A instruction.

OTA: Output A Register (Unconditionally)

The OTA instruction in Fig. 13-9 transfers the contents of the A register to the device specified. The A register is not altered. The function code can be used to steer the data into

multiple registers if required. For instance, if a 32-bit register is to be loaded by OTAs, the register can have four segments and each segment can be specified by the function code.

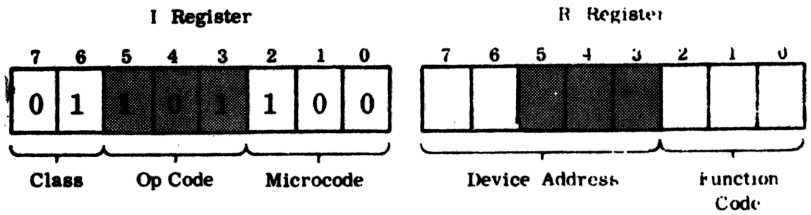


Fig. 13-9. "Output A register" instruction.

OTZ: Output Zero (Unconditionally)

The OTZ instruction in Fig. 13-10 is similar to the OTA except the contents of the A register is not selected onto the I-O bus, resulting in a transfer of all zeros in the data word. The contents of A is not altered.

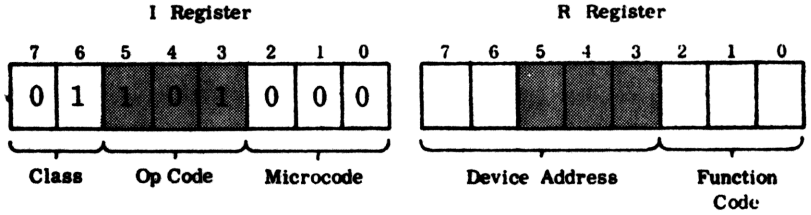


Fig. 13-10. "Output zero (unconditionally)" instruction.

OAS: Output A And Skip On Response

The OAS instruction in Fig. 13-11 is a combination of the OTA and the sense and skip instructions. The condition specified by the function code is first sensed and if a response is obtained, the contents of A is transferred to the I-O device and a skip occurs. If no sense response occurs, the transfer is not made and the next instruction in sequence is executed.

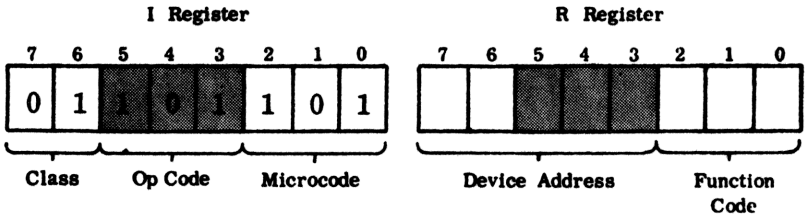


Fig. 13-11. "Output A and skip on response" instruction.

OZS: Output Zero And Skip On Response

The OZS instruction in Fig. 13-12 is a combination of the OTZ and the sense and skip instructions. The function code specifies the condition to be sensed. If no sense response is obtained, the A register is unaffected and the next instruction in sequence is executed.

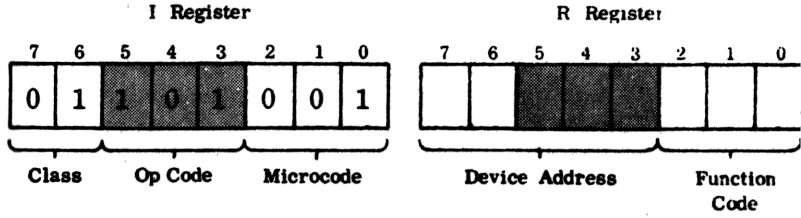


Fig. 13-12. "Output zero and skip on response" instruction.

BLOCK TRANSFER INSTRUCTIONS

The block input-output instructions (Fig. 13-13) are powerful I-O instructions that allow access to memory by peripheral devices without going through the A register. Automatic indexing of an address pointer reduces overhead time and allows the block I-O instruction to be used as single-execute interrupt instructions. Four memory cycles are required for each transfer: two cycles to retrieve the double-word instruction, one cycle to index the pointer and form the operand address, and one cycle to affect the transfer between memory and the I-O device.

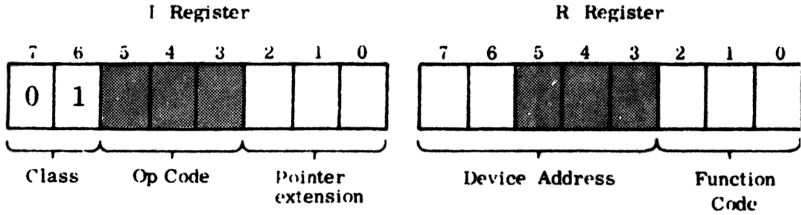


Fig. 13-13. Block I-O instruction format.

The A register is not altered by the block I-O instructions. The second 2K of memory provides the working area for the block transfers. Each block can be up to 256 words and up to eight blocks can be handled by the block instruction.

The second word of the instruction defines the peripheral and specifies a function code if needed. The first word of the instruction contains three bits that are used as an extension to

the input pointer to provide 11 variable address bits. A 12th bit is a hardwired to force the block buffer areas to be in the second 2K of memory, preventing accidental storage in page 0.

The pointer (Fig. 13-14) is first indexed (incremented) and replaced and the indexed value is then used in the effective address. Thus, the pointer must be preset to one less than the starting address. For example, if the first word in the block is to be stored in page 12, location zero, the extension in the instruction word is 2, and the pointer is preset to 377. After the 256th word has been transferred, the pointer will again contain 377. The extension may now be incremented to 3 and the process restarted to transfer the second block in page 13 starting with word 000. The input pointer is stored at location 6, page 0, and the output pointer is stored at location 10, page 0.

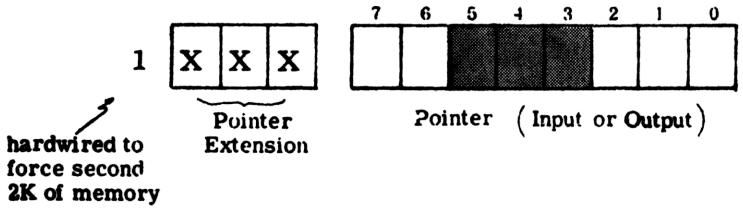


Fig. 13-14. Block pointer format.

The use of the block I-O instructions as interrupt instructions provide an extremely flexible and powerful data handling capability. By using two standard priority interrupt lines in the 208 and 808, blocks may be chained together to provide up to 2K-word records to be transferred to or from memory automatically without intervention of the main operating program.

Using block input (or output) as an interrupt instruction, a word may be transferred under interrupt control without disturbing the operating program except to cause the program to be delayed four cycles while the block input instruction is being executed. This requires one interrupt line. The second interrupt line is used for end-of-block as indicated by an echo pulse formed when the pointer goes to 377 after being indexed. The echo pulse is used to set a flag in the peripheral controller which causes the end-of-block interrupt. A subroutine is entered via the jump and save instruction to update the pointer extension and re-enable the first interrupt line (which is disabled by the echo pulse) and the next block transfer begins.

INB: Input Block To Memory

The instruction in Fig. 13-15 transfers a word from the specified device to the memory at the location specified by the input pointers +1 and the pointer extension. Up to 256 words may be transferred as a block (the INB instruction is executed 256 times). The input pointer is location 6, page zero.

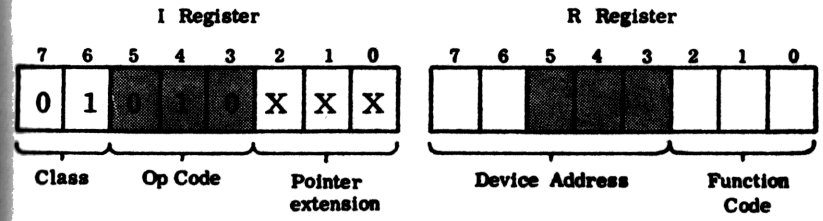


Fig. 13-15. "Input block to memory" instruction.

OTB: Output Block From Memory

The instruction in Fig. 13-16 transfers a word from memory at the address specified by the output pointer +1 and the pointer extension to the specified device. Up to 256 words may be transferred as a block. The output pointer is location 10_8, page zero.

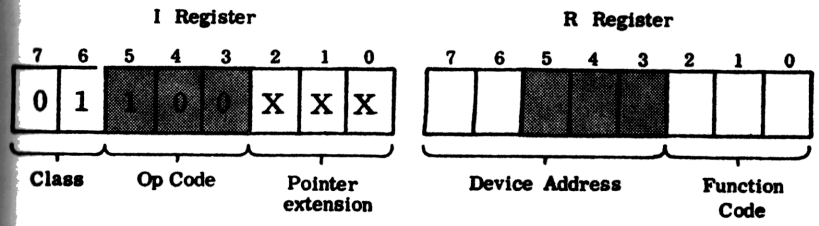


Fig. 13-16. "Output block from memory" instruction.

PRIORITY INTERRUPT SYSTEM

The 208 and 808 priority interrupt system makes the computer an extremely flexible and powerful real time systems component. The interrupt system provides the ability to obtain fast response to external stimuli, to transfer blocks of data without attention of the operating program, to generate timing intervals and-or time-of-day clock (with real time clock option), and to react to low power line conditions (using power fail-restart option).

When an interrupt request is received by the computer, the computer responds to the interrupt at the end of the current instruction being processed (provided interrupts are enabled). Control is transferred to the memory location associated with the particular interrupt line requesting the interrupt. There are two reserved memory locations in page 0 for each interrupt line in the system. These two memory locations contain the instruction to be executed by the interrupt. For single-instruction interrupts, the interrupt instruction is one of the 208 and 808 instructions that does not change the program counter or the A register. A single-instruction interrupt can be serviced without altering the operating program except to delay it for the amount of time it takes to execute the interrupt instruction. The normal interrupt instruction is "jump and save" which transfers control to a subroutine designated by the jump and save instruction. The contents of the L and the P counters are stored in the first two memory locations of the subroutine and are used later for a return jump back to the operating program.

Control

The PDC 208 and 808 can be inhibited from responding to any interrupt request by an interrupt mask. Two control instructions allow the interrupts to be disabled or enabled under program control. In addition, if a jump and save instruction is used as an interrupt instruction, the interrupt system is automatically disabled.

Single-instruction interrupts do not require control of the interrupt mask by the program; for example, if teletype buffer ready flip-flop is the stimuli causing an interrupt and the interrupt instruction is block output instruction. This removes the interrupt condition by the time the interrupt instruction is over. If the interrupt instruction had been a jump and save, the interrupt request would remain and would cause another interrupt before the transfer to teletype had occurred, with the result that the return address is within the interrupt subroutine and the computer cannot recover from the loop. Thus, it is necessary to disarm the interrupt system until one of two things occurs, either (a) the line in the teletype is disarmed or (b) the transfer to the teletype occurs, which satisfies the interrupt condition.

CONTROL CONSOLE

The computer console consists of an 8-bit register display, eight data switches, five register select switches, and miscellaneous control switches and indicators (Fig. 11-1).

Register display: An 8-bit register display is provided to allow viewing the contents of a selected register. Contents of the A, I, L, P, and R may be displayed while the computer is halted. In the run mode the contents of the A bus is displayed.

Data entry switches: Eight data entry switches are provided for entering data into a selected register. The switches are momentary-action types. Data is entered immediately upon depression if the computer is halted and in the step mode. In the run mode, the switches are disabled.

Register select switches: Five interlocked, alternate action switches are provided for selection of the register to be connected to the display and the data switches. The switches are wired on a priority basis so that if two switches are depressed at one time, the switch on the left takes priority, disabling the one on the right. This prevents the contents of the two registers from being ORed together and placed in both registers.

Clear switch: A momentary-action switch is provided for clearing a selected register. The switch is disabled in the run mode.

Cycle switch: A momentary action switch is provided for cycling the computer in the step mode (depressed position), and for initiating the run mode if the step-run switch is in the run (released) position.

Step-run switch: An alternate action switch is provided for controlling the two operating modes of the computer. When the switch is released, the computer enters the run mode when the cycle switch is depressed. If the computer is in the run mode, depressing the step-run switch causes the computer to enter the step mode; i.e., the computer will halt at the end of the next instruction.

Step indicator: An indicator is provided to indicate that the computer is in the run mode. The step and run indicators are driven from opposite sides of the run flip-flop in the computer.

Overflow (OV) indicator: An indicator is provided to indicate the state of the overflow flip-flop. The indicator lights when the OV flip-flop is set.

Relative switch: An alternate action switch is provided to allow addressing relative to the program counter when manually storing or examining a program in the core memory. The relative switch is for use with the load A and

store A instructions in the step mode only. When it is depressed, the instruction in the instruction register is not allowed to change, and the A field in the R register is ignored. Instead, the lower eight bits of the operand address are obtained from the P counter if the current address is specified; otherwise, the operand address is in page zero.

Manual execute: An alternate action switch is provided to allow the operator to insert an instruction into the I and R registers and manually execute it by depressing the cycle switch. Normal operation of the computer requires retrieving an instruction when the cycle switch is depressed, since the instruction in the instruction register has already been executed before the computer halts. The manual execute switch forces the computer to execute the instruction in the instruction register. The program counter is not updated, which means the manual execute switch allows instructions to be executed manually without disturbing program location.

Sense switch: An alternate action switch is provided to allow the operator to interface to an operating program. The state of the sense can be sensed under program control, allowing branching conditions to be executed, conditioned on the state of the sense switch. The sense switch may be changed in the run mode.

Memory disable: An alternate action switch is provided to allow the operator to disable memory. When the memory disable is depressed, the current generators in the memory core stack drive circuitry are disabled, inhibiting the memory from operating. This allows the contents of the memory to be protected from transients when power on the controller is turned on or off manually. If the controller is equipped with the optional power fail restart option, the memory disable switch need not be used except in instances where the power fail subroutine is not in memory.

Reset switch: A momentary action switch is provided to allow the computer system to be initialized when power is turned on. The reset switch initializes the computer processor flip-flops, placing the computer in the stop mode, and provides a pulse on the I-O cable which is used by peripheral logic for initialization. The reset switch may also be used as a panic button to break a runaway loop.

Power switch: An alternate switch is provided for controlling AC power to the computer power supply. Power is on when the power switch is depressed.

Power indicator: An indicator is provided to show when the power switch is depressed. Power for the indicator lamp is obtained through the circuitbreaker on the back of the computer.

PERIPHERAL EQUIPMENT DESCRIPTION

Described in this section are some of the I-O devices that are available with the PDC-208 and 808.

33 Teletype Option

The 33 teletype option provides the PDC-208 and 808 system with four I-O features in one package: keyboard input, page printer, paper tape reader, and paper tape punch. The peripheral device is a Model 33TC send-receive set (manufactured by Teletype Corp.) operated in the duplex mode. A peripheral interface to connect the computer to the teletype is contained in one option logic board and is mounted in the basic PDC-208 and 808 chassis. The interface contains a buffer-shift register that performs parallel-to-serial conversion when feeding from the computer to the teletype and serial-to-parallel conversion when feeding from the teletype to the computer.

High-Speed Reader Option

The high-speed paper tape reader option (PTR) consists of a 300 character per second optical reader and the interface logic between the reader and the PDC-208 and 808. The reader is an undirectional, 8-level unit that reads continuously or steps one character at a time under control of the interface. Each word (8-bit character) is read in parallel and transferred to a buffer register in the interface where it is held until it is transferred to the computer. The reader mounts in a standard 19-inch equipment rack, requiring 7 inches of rack height. The interface is contained on one logic board and is mounted within the basic PDC-208 or 808 chassis.

High-Speed Punch Option

The high-speed paper tape punch option (PTP) consists of a 60 character per second punch and the interface logic between the punch and the PDC-208 or 808. The punch is an 8-level unit that can punch paper or mylar tapes. The punch interface contains an 8-bit buffer register to hold information being punched and the control and interface logic required to control

the punch and communicate with the computer. The punch mounts in a standard 19-inch equipment rack and requires 10½ inches of rack height. The interface is contained in a logic board which mounts in the basic PDC 208 or 808 chassis.

MAINFRAME OPTIONS

This section describes some of the options available with the PDC-208 and 808 processor.

Power Fail-Restart Option

The power fail-restart (PFR) option allows the computer to be operated from an unreliable AC source. A low-power condition or a temporary power outage is detected in time to allow the operating program to prepare for the power loss. When power returns to normal the computer is automatically restarted. Unattended operation is possible at remote sites.

The PFR option consists of a voltage detector, interrupt register, and priority interrupt logic on an option module in the PDC-208 or 808 chassis. Two interrupts are provided with the option. One interrupt flags the low-power condition, and the other interrupt restarts the computer one second after power resumes.

Operation

The PFR option monitors the 115v AC line voltage and closes a switch when the voltage drops below a preset value (typically 100v). The switch closing sets a flip-flop which requests an interrupt. The interrupt location should contain a jump and save to the power fail subroutine. Approximately two milliseconds after low power is detected, inhibiting memory start pulses shunt the current sources in the memory drive electronics preventing spurious memory cycles as the power collapses. When power returns, the detector output (indicating power is normal) is delayed one second and then used to create the restart interrupt. During the one second interval between power up and the restart of the computer, the computer system is initialized. After the one second delay, the computer is forced to location zero in page 0 to obtain the restart interrupt instruction (jump to restart subroutine).

Real Time Clock Option

The real time clock (RTC) is a PDC-208 or 808 processor option that provides a means of determining elapsed time and-

or creating a time-of-day clock with software. The RTC derives time pulses from the 60 Hz primary input (120 PDS) to the PDC-208 or 808. These time pulses are then used to generate an interrupt (clock interrupt) every 8.33 milliseconds to increment a counter in memory.

A second interrupt (sync interrupt) is generated by the RTC when the counter in memory reaches a count of zero (overflow) while being incremented by the clock interrupt instruction. The sync interrupt is used to enter a time-keeping subroutine. The two interrupts required by the clock and sync pulses are contained within the RTC. Interrupt addresses for each line are also generated by the RTC logic, providing a complete stand alone option.

Operation

The instruction set of the basic PDC-208 or 808 system is increased by the RTC. The number of interrupts is also increased by two. Each instruction, including the suggested interrupt instructions, are described in Chapter 18.

Priority

Priority of the RTC interrupts, relative to other interrupts in a PDC-208 or 808 system, is determined by the wiring of the PROT line. Priority in the 808 is IL1X, IL2X, PF, RTC then others. In the 208 it is PF, IL1X, IL2X, RTC then others.

RESERVED MEMORY LOCATIONS

Since each interrupt line has two memory locations associated with it, these locations must not be used for other purposes. As an aid to the programmer, all reserved locations in the basic PDC-208 and 808 system plus mainframe options are listed. In larger systems additional locations may be required. Programming codes for I-O devices appear in Chapter 18.

Chapter 14

I-O Interface Reference

The PDC-208 and 808 use a minimum amount of interface logic. Standard DTL circuits are used throughout the I-O, providing savings in parts cost and power.

PARTY LINE I-O BUS

The PDC-208 and 808 incorporate a "party-line" I-O bus to communicate with peripheral logic. Each peripheral logic group or "device" is given an address which is used in all communications between the processor and that device. A block diagram of a PDC-208 or 808 system with three devices is shown in Fig. 14-1. Notice that while the I-O cable is physically connected in a serial fashion; the devices are electrically connected in parallel.

A termination "shoe" containing line-terminating resistors is connected to the end of the I-O cable at the last device. The resistor is required to maintain the integrity of the transmission line by terminating the line in its characteristic impedance (or approximation thereof) at the receiving end to minimize reflections. Since the data lines are bilateral; i.e., they can be driven from either end, a termination is required at each end. The unilateral lines require a terminating resistor only on the receiving end. Fig. 14-2 shows the circuit schematics of the two types of lines. All lines are twisted pairs.

The length of the I-O cable should be kept as short as possible to minimize cross talk between lines. When possible peripheral logic should be located adjacent to the computer. The maximum length allowable varies with the application, but in general it should be kept under ten feet.

The I-O cable consists of a number of lines grouped according to functions. Each functional grouping is called a bus.

Data bus—eight bidirectional lines that transmit data between devices and the computer. Each of the eight lines is terminated on each end.

Peripheral address bus—five lines that transmit the device address to devices during the execution of an I-O instruction.

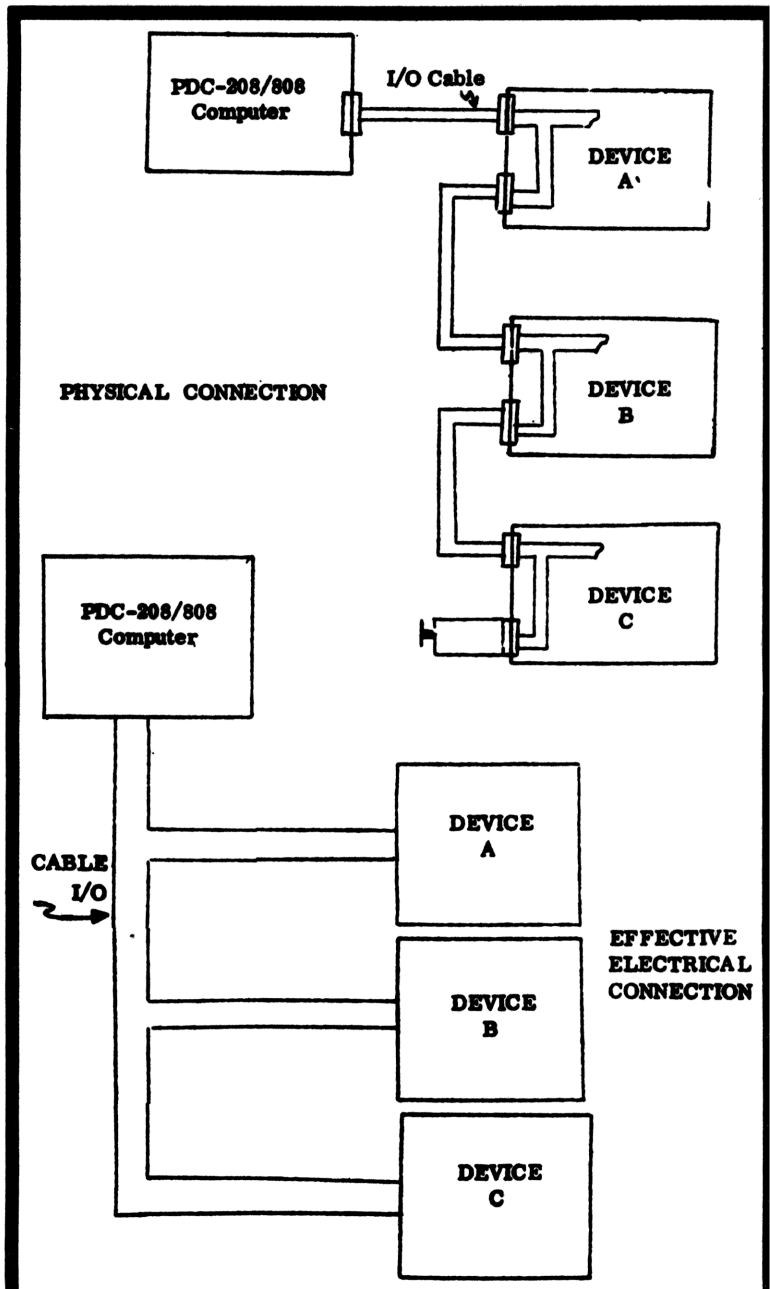


Fig. 14-1. Block diagram of a PDC 208 or 808 system.

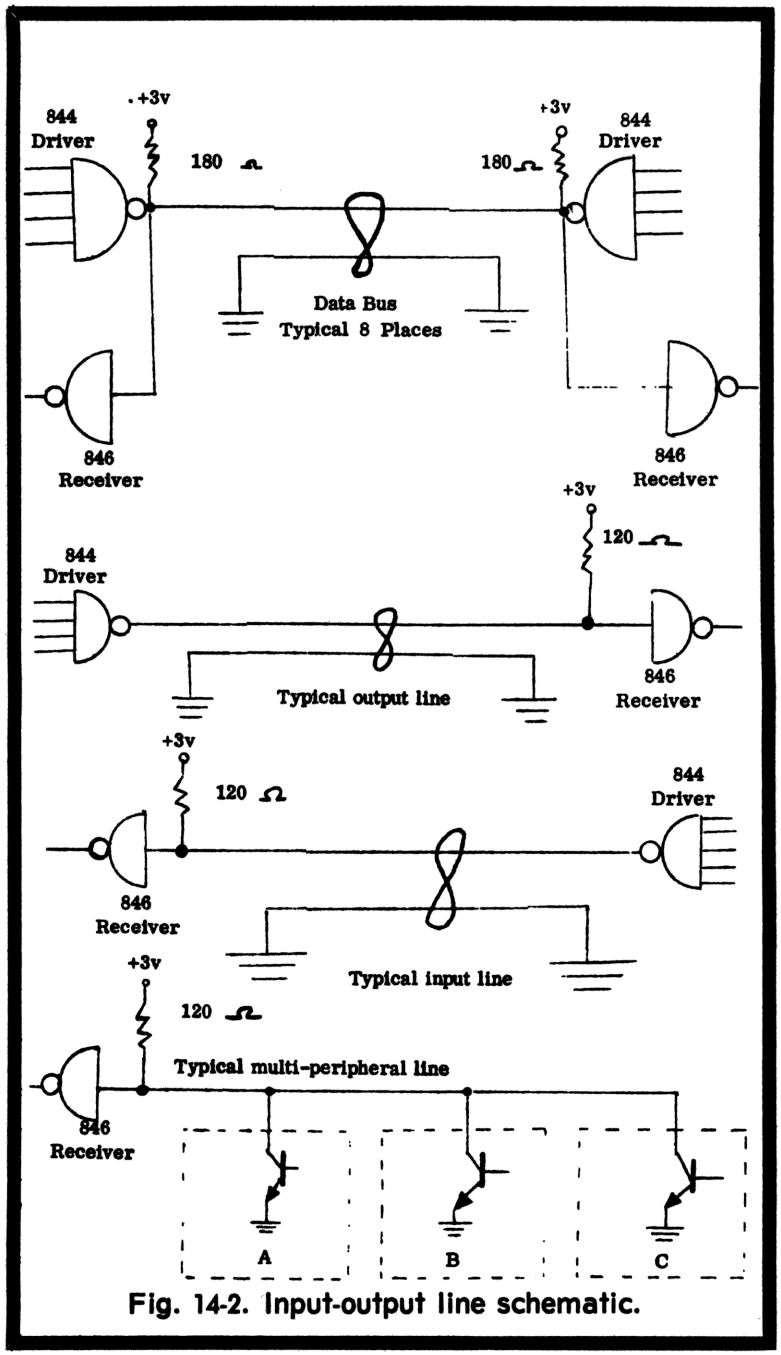


Fig. 14-2. Input-output line schematic.

Function bus—three lines that transmit the function code to the devices during an I-O instruction.

Control bus—fourteen lines that transmit control signals between the computer and I-O devices. Each signal is described in the function index in Chapter 15.

Miscellaneous lines—the power lines for the termination resistors and several spare lines that are not used for PROT wiring and other systems requirements. **Note:** The signals appearing on the I-O cable are "ground true" logic; i.e., the false level is the high level. This permits the parallel connection of devices using the "wired OR" technique and also conserves power since the line terminations do not dissipate power except when the true (ground) signal exists.

The PDC-208 and 808 employ a 2 to 2.75 ms I-O period. Consequently, considerable time is available in implementing the signals necessary to interface with peripheral logic, eliminating the need for fast or special circuits. The shortest pulse width used is the 500 ns PLSE signal which is used as a strobe. All other signals are at least two microseconds in duration, providing plenty of time for decoding and gating.

OUTPUT COMMAND

The output instructions—OTA, OTZ, OAS, OTB—all employ the same peripheral logic and timing. Data is placed on the data bus for two microseconds and at the same time the device address and function code is placed on the peripheral address bus and function bus, respectively. The OUTX control line is true for the two microseconds that data is present, indicating an output instruction is being executed.

At the end of the 2-microsecond OUTX signal, the PLSE signal occurs if the output transfer is to be made. In the case of a sense and output combination, the PLSE signal will not

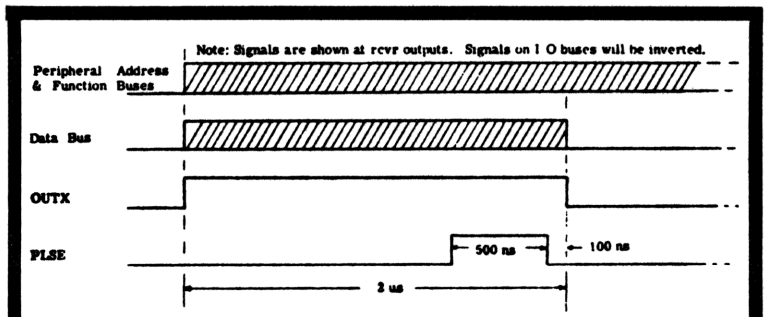


Fig. 14-3. Output timing diagram.

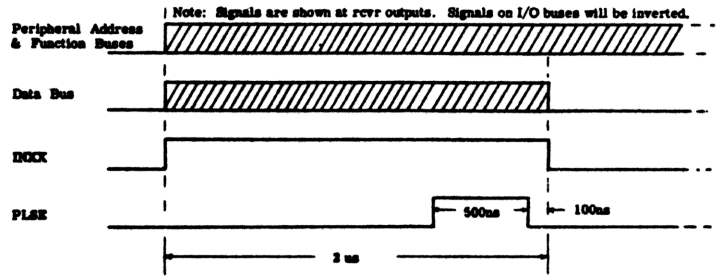


Fig. 14-4. Input timing diagram.

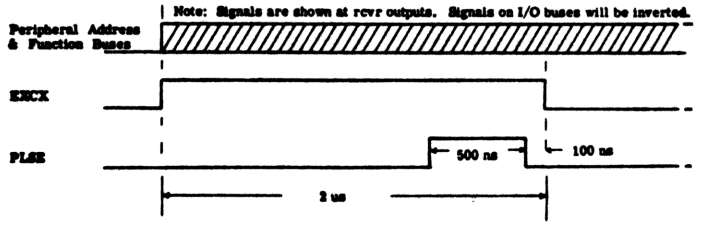


Fig. 14-5. External control timing diagram.

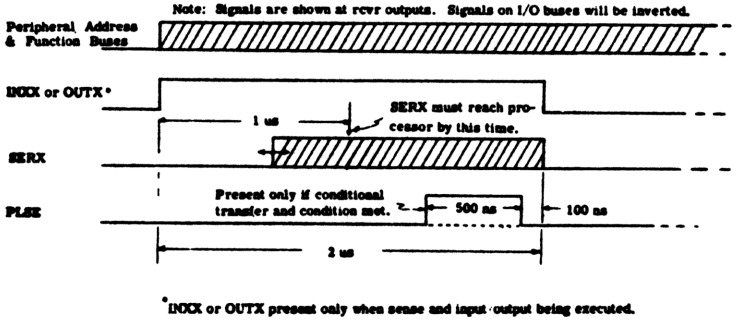


Fig. 14-6. Sense timing diagram.

occur unless a sense response is obtained by the processor. The composite timing of an output command is shown in Fig. 14-3.

INPUT COMMAND

The input instructions—INA, MIA, IAS, MIS—employ the same peripheral logic and timing. Data is placed on the data bus by the device which is addressed during the two microseconds the INXX signal is true. At the end of this time, the PLSE signal occurs if the input transfer is to be made. In the case of the sense and input instructions, the PLSE will not occur if a sense response is not obtained by the processor. The absence of the PLSE signal indicates the transfer was not made. The composite timing of an input command is shown in Fig. 14-4.

EXTERNAL CONTROL COMMAND

The EXC instruction timing is similar to the transfer instructions except the data bus is not active. The EXCX control signal is true for two microseconds. The PLSE signal occurs at the end of the EXCX signal and is used to generate set pulses which are gated by the function codes generated by decoding the function bus. Fig. 14-5 shows the timing of the EXC command.

SENSE COMMAND

The sense instructions do not use a sense control line, since the final gating for sense is done inside the processor. This allows the sense instruction to be combined with the input and output instructions, creating powerful I-O instructions that conserve memory and shorten program execution times.

When a peripheral device is addressed by the processor, the function code is used to gate the condition of a specified function onto the sense response line, SERX. If a true response exists, the SERX line in the control bus is driven to ground. Thus the sensing is accomplished during all I-O instructions, but using the results of the sense is determined by the instruction being executed. The processor must receive the sense response one microsecond after the beginning of the instruction to allow time for decision making within the processor. Fig. 14-6 shows the time for sense.

INTERRUPT TIMING

The use of the standard interrupts in the basic PDC-208 or 808 enables the computer to perform as a real time systems

component responding to external stimuli and to allow considerable time savings in executing I-O programs that transfer data to or from peripheral devices. A typical interrupt sequence using one of the two fully implemented interrupt lines is as follows:

A. The buffer ready status flip-flop in a device turns on, indicating that the device desires data to be transferred from the computer and drives the IL1X line in the control bus to the ground (true) state, requesting an interrupt.

B. The processor finishes the current instruction and if interrupts are enabled; i.e., the processor is allowed to respond to interrupts, the request for interrupt is acknowledged and the instruction located in the two reserved memory locations associated with IL1X is executed. The IUAX signal becomes true as soon as the request is acknowledged and stays on for the time required to execute the interrupt instruction, typically two to four cycles.

C. If the interrupt instruction is a JST, the interrupt enable flip-flop in the processor is turned off, preventing further interrupts from being acknowledged until the enable flip-flop is again turned on by the program.

D. A subroutine is entered (if JST is used) which transfers the desired data to the interrupting device, satisfying the request and resetting the buffer ready flip-flop which removes the interrupt request. Interrupts can now be enabled and a return jump back to the operating program executed.

E. If the interrupt is a single-execute instruction such as block output, the request for data is satisfied by the execution of the interrupt instruction and the program continues as soon as IUAX goes off, creating only a 4-cycle pause in the program.

The use of the IURX line is different from that of the IL1X and IL2X lines in that the interrupt address must be specified by the device requesting the interrupt, whereas the IL1X and IL2X lines have associated address generators within the computer. To obtain the interrupt address from the device, the processor sends an interrupt address request signal IARX to the devices and the device requesting sends back the address the processor is to use to find the interrupt instruction. The IARX signal is two microseconds long. The PLSE signal occurs at the end of the IARX signal to notify the device that the processor has received the address. The timing of the interrupt sequence using the IURX line in this fashion is shown in Fig. 14-7.

Another method of using the IURX line is the scanning technique. Several devices are allowed to generate an interrupt request at will and a subroutine is entered which polls each device capable of having generated a request. The polling is done by executing sense instructions until the requesting device is located. In this manner, priorities may be changed at will by the programmer by simply changing the order in which the devices are sensed. The technique is slower, however, than a fully automatic interrupt sequence but is considerably less expensive. If a device requests an interrupt using IURX and does not respond with an address during IARX, the processor accesses memory locations 20 and 21 for the interrupt instruction (which in almost all cases must be a JST).

All PDC-208 and 808 standard peripheral interfaces have two interrupt line drives implemented. One driver is normally connected to IL1X and is used to generate an interrupt on buffer ready. The other driver is normally connected to IL2X and is used to generate an interrupt on end-of-block when performing block transfers under interrupt control. Each driver has a mask associated with it which may be controlled by the operating program.

Several peripheral devices may be connected to the IL1X and IL2X lines, provided only one device is allowed to be operating under interrupt control at a given time. Since the IL1X and IL2X lines are fully implemented and, therefore,

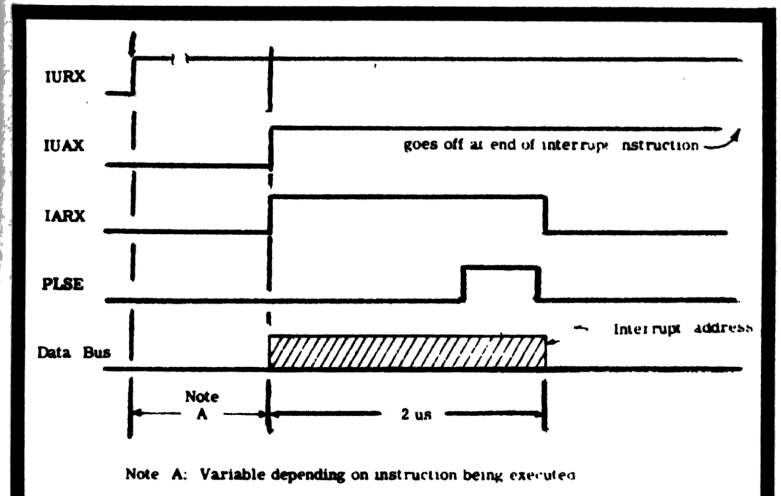


Fig. 14-7. Interrupt timing diagram.

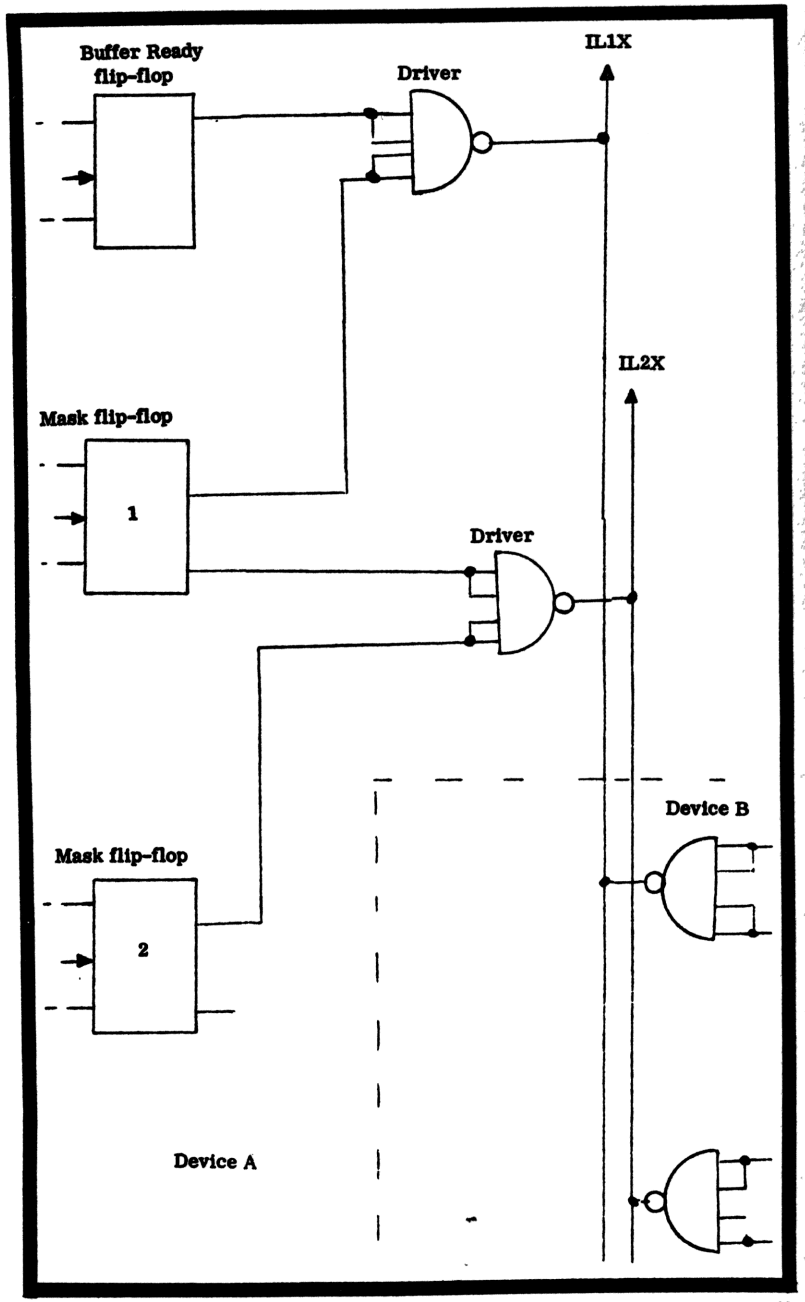


Fig. 14-8. Diagram of the standard interrupts in peripheral devices.

general purpose, they may also be used by customer interface logic without the need to generate priority and address logic outside the PDC-208 or 808.

Fig. 14-8 shows the logic in a standard PDC-208 or 808 peripheral computer that transfers data to or from the process or (TTY, high-speed reader, etc.). The buffer ready flip-flop is set when the device is ready to send or receive data. If the Mask FF 1 is on, an interrupt is generated on IL1X which is used to cause data to be transferred. If the block transfer instructions are used, an end-of-block echo pulse occurs when the pointer overflows which resets mask FF 1. This creates an interrupt on IL2X (if mask FF 2 has been previously set) which is used to signal the end of the block being transferred. Notice that each interrupt line driver can be disabled by the mask flip-flops, allowing complete system flexibility in the use of the interrupts.

Part 3: Maintenance

Chapter 15
Introduction To Maintenance

The logic diagrams in this section quite frequently, especially in the case of MSI devices, list only the pin numbers associated with a simple block. The diagrams included here (Figs. 15-1 through 15-12) allow the association of these inputs with specific circuit functions.

The logic diagrams are followed by a complete **Function Index List**. Every computer has a similar list which simply names the various signals used in the system. For a synchronous computer, such as the PDC 808, the logical place to start is with the clock signals since they, ultimately, control all internal operations.

Wherever possible, waveforms have been photographed to show the deviations that might be expected from the "idealized" waveforms presented in the text. In many instances the waveforms are practically unrecognizable when compared to the "idealized" waveforms.

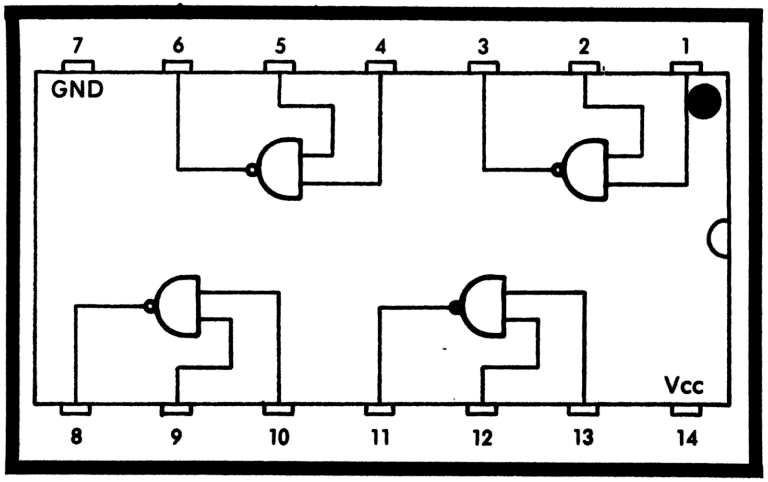


Fig. 15-1. Diagram of the MC846 and SN15846 quad 2-input NAND gate.

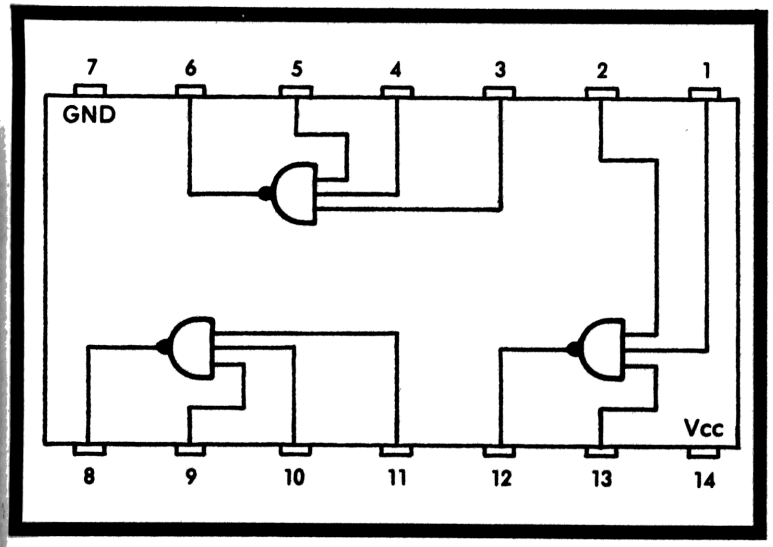


Fig. 15-2. Diagram of the MC862 and SN15862 triple 3-input NAND gate.

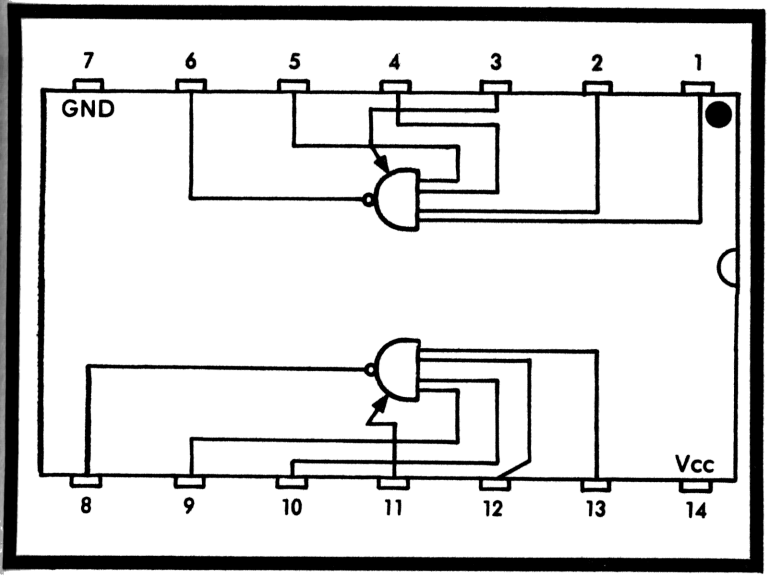


Fig. 15-3. Diagram of the MC830 and SN15830 expandable dual 4-input gate; the MC832 and SN15832 expandable dual 4-input buffer, and the MC844 expandable dual 4-input power gate (line, relay, or lamp driver).

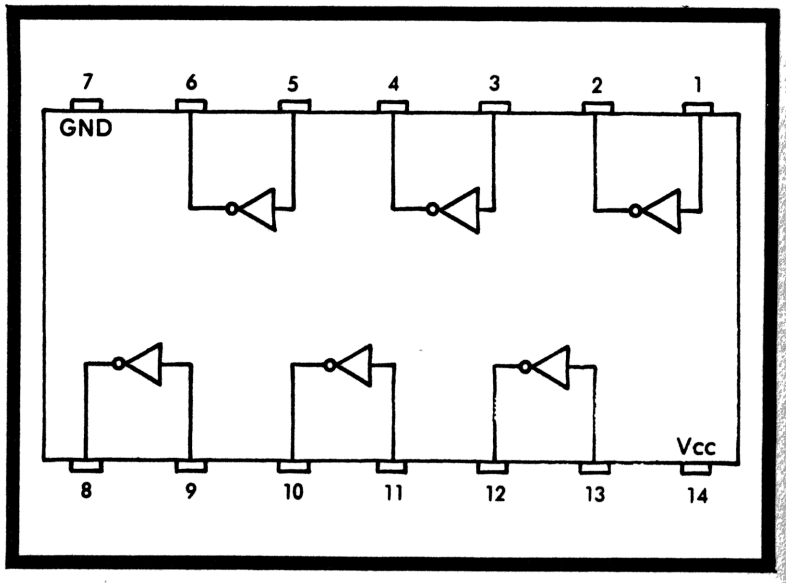


Fig. 15-4. Diagram of the MC836 and SN15836 hex inverter.

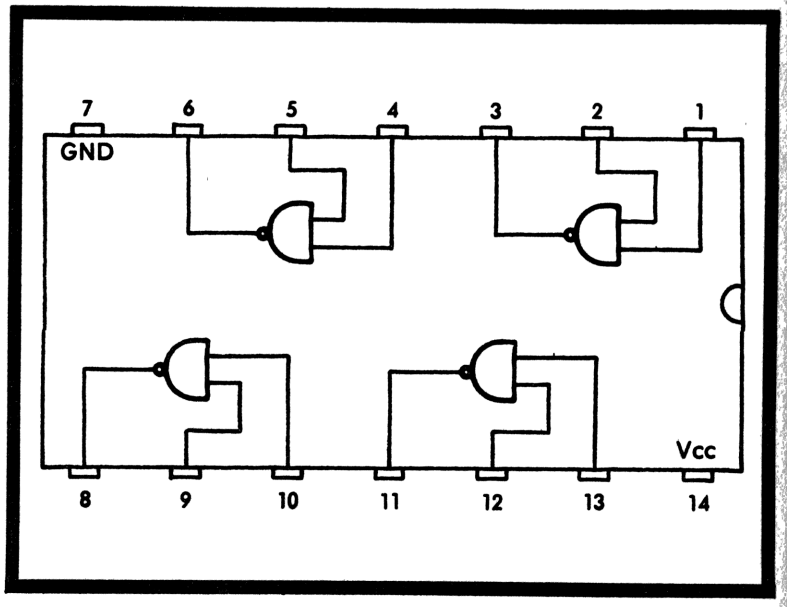


Fig. 15-5. Diagram of the SN7400 quad 2-input NAND gate.

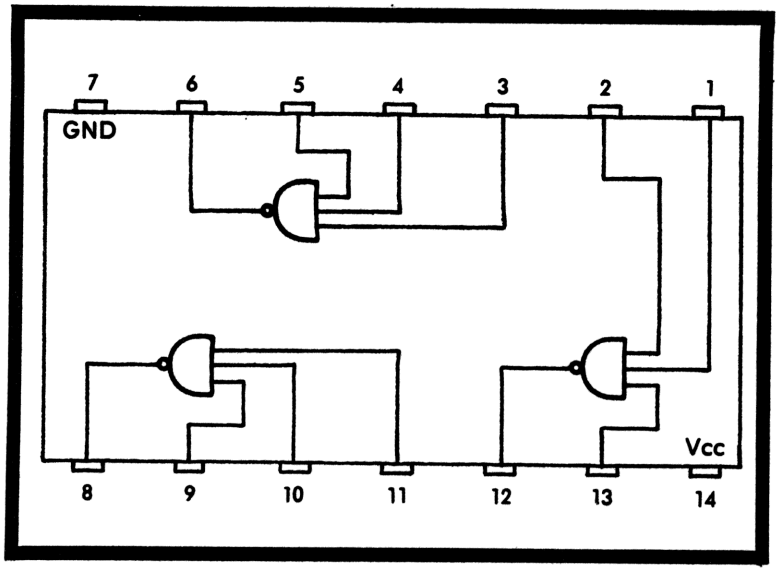


Fig. 15-6. Diagram of the SN7410 triple 3-input NAND gate.

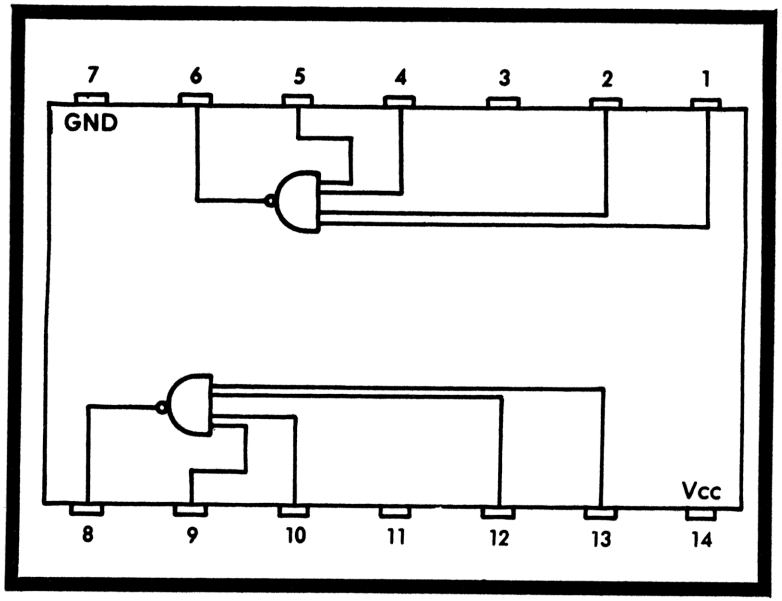


Fig. 15-7. Diagram of the SN7420 dual 4-input NAND gate and the SN7440 dual 4-input NAND buffer.

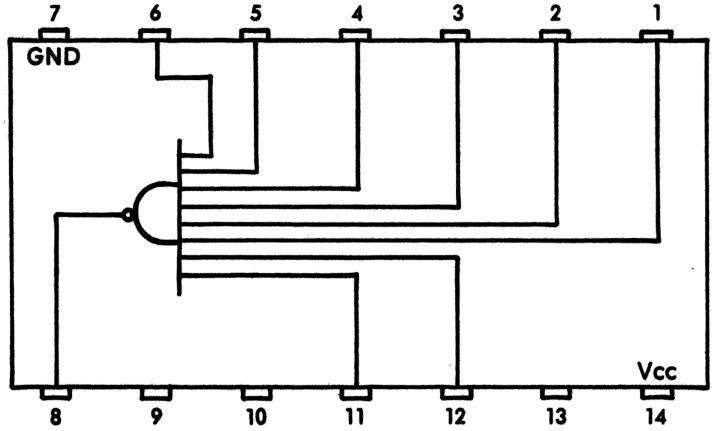


Fig. 15-8. Diagram of the SN7430 single 8-input NAND gate.

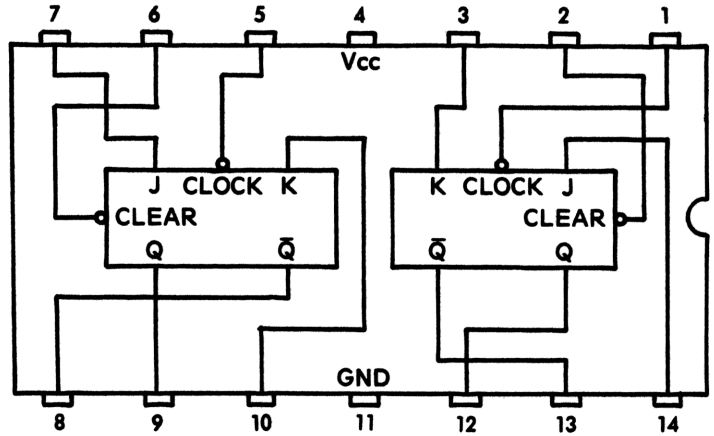


Fig. 15-9. Diagram of the SN7473 dual J-K flip-flop.

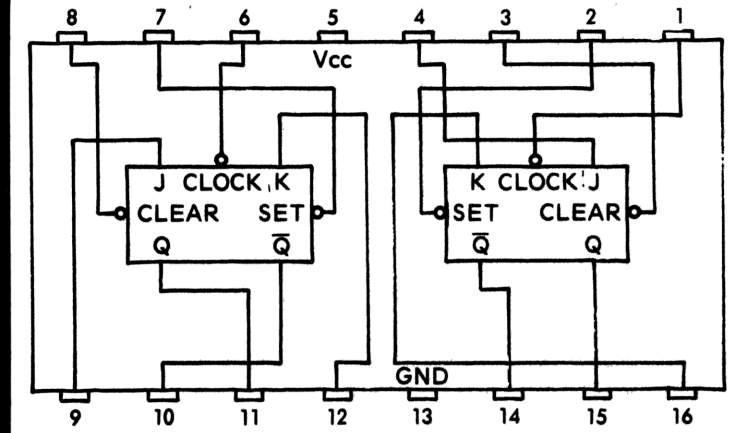


Fig. 15-10. Diagram of the SN7476 dual J-K flip-flop with direct set capability.

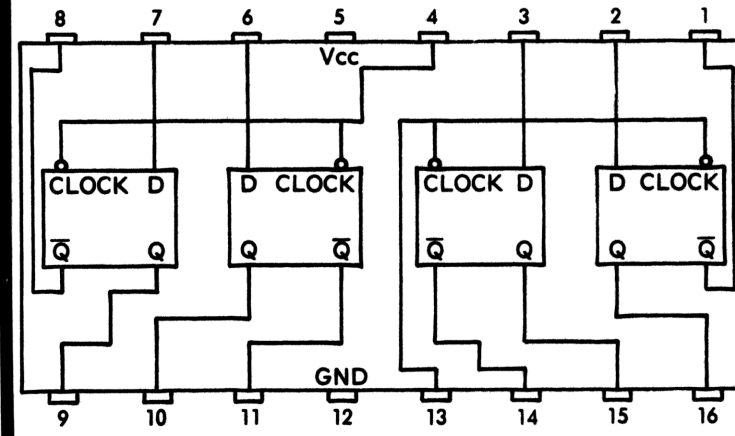


Fig. 15-11. Diagram of the SN7475 4-bit bistable latch.

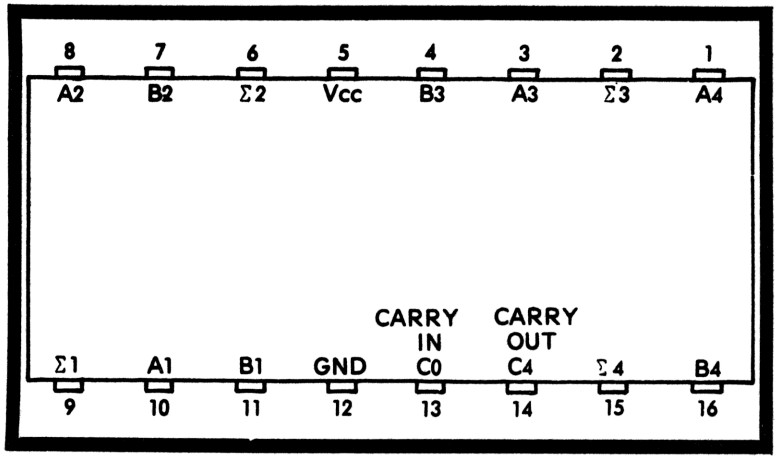


Fig. 15-12. Diagram of the SN 7483 4-bit binary full adder. Since each bit requires nine gates to illustrate it, the gate symbols have been omitted (Σ is the Greek sigma for sum.)

FUNCTION INDEX LIST

AAXX—1-MHz square wave.
AB00 to AB07—A bus. Distributes output of the adder to A, P, L, M, and W registers, console display, and D bus drivers.
ACY1—Address cycle 1. Signal time when the word being read from the memory is the least significant eight bits of an indirect address or is the pointer for a block I-O instruction.
ACY2—Address cycle 2. Signal true when the word being read from the memory is the most significant portion (page number) of an indirect address.
ACYE—Address cycle enable. Enables the ACY1 flip-flop when the next memory cycle is to be an address cycle.
AR00 to AR07—A register outputs.
ASHX—Arithmetic shift.
AXXX—500-kHz square wave.
BXXX—250-kHz square wave interrupted when the computer is in the step mode.
CARY—Carry from the adder.
CINP—Carry in pulse. Inputs a carry to the least significant bit of the adder.
CLRW—Clear W register. Occurs at the beginning of each memory cycle.
CLRX—Clear. Signal generated in the step mode when the clear switch on the console is depressed.

CXXX—125-kHz square wave interrupted when the computer is in the step mode.
CYCX—Cycle. Pulse of about 100 ns duration generated when the cycle switch on the console is depressed.
DB00 to DB07—D bus. Bilateral data bus in the I-O cable used for parallel transfer of data between the computer and I-O devices.
DJ2X—Direct jump. Signal true during ICY2 of a direct jump instruction.
DS00 to DS07— Data switch inputs to the A bus drivers.
DSSL—Data switch select. Signal is true in the step mode if any data entry switch is depressed.
ECHO—A pulse that occurs during execution of block I-O if the pointer overflows. Used to reset the interrupt enable flip-flop on the IL1X line in a peripheral.
EXCX—External control. Signal in the I-O cable that is true during execution of an external control instruction. 2-microsecond pulse width.
FEIX—"Fetch" instruction. Signal that indicates the next memory cycle is to be an ICY1 instruction cycle.
FJIX—"Fetch" jump instruction. Signal indicates that an indirect jump is in progress and the next memory cycle is to be an ICY1 cycle.
G0XX—Signal that is true for register change instructions, shift instructions and operate instructions. G0XX equals (IR06-)(IR07-)
G1XX—Signal that is true for I-O instructions. G1XX equals (IR06)(IR07-)
G2XX—Signal that is true for skip instructions. G2XX equals (IR06-)(IR07)
G3XX—Signal that is true for memory reference instructions. G3XX equals (IR06)(IR07)
H0XX to H7XX—Signals generated from decoding IR03 to IR04.
IARX—Interrupt address request. Signal on the I-O bus that requests interrupt address from the interrupting device if the device is using the IURX request line. 2 microseconds duration.
ICY1—Instruction cycle 1. True when the word being read from the memory is the first word (or only word) of an instruction and is to be placed in the I register. True for one memory cycle.
ICY2—Instruction cycle 2. True when the word being read from memory is the second word of a 2-word instruction, and is to be placed in the R register.
ICYX—Instruction cycle. True for either an ICY1 or ICY2 memory cycle.

+INH—Inhibit voltage supply to inhibit drivers in memory.
IMXX—Interrupt mask. Flip-flop that enables or disables interrupt requests to the IUAX flip-flop.
INXX—Input. Control signal on the I-O bus that indicates an input instruction is being executed.
IOCL—I-O clock. 1-MHz pulse train from the processor. Used in interrupt generation and for other clocking functions in peripheral logic. IOCL is interrupted when IUAX is on.
IORX—Inclusive OR. Signal that indicates that an inclusive OR instruction is being executed.
IR00 to IR07—I register stages.
IUAX—Interrupt acknowledge. Signal goes true at the end of the current instruction if an interrupt request is present and enabled. Stays true until interrupt instruction is executed.
JMPS—Jump and save. Signal that is true when a jump and save instruction is in the I register.
JMPX—Jump. Signal that is true when a jump instruction is in the I register.
LR00 to LR08—Location register. Contains the eight least significant bits of the program location, which specifies the location of a word within a page.
LSHL—Logical shift left.
LSHR—Logical shift right.
MDSA—Memory disable switch A.
MDSB—Memory disable switch B.
MROC—Memory reference operand cycle.
MR08 to MR13—M extension register. Upper six bits of memory address.
MSAX—Manual select A register. True in the step mode if A is selected for display.
MSCX—Memory start clock. Pulse that occurs every 8 microseconds when the computer is in the run mode.
MSIX—Manual select I register. True in the step mode if I is selected for display.
MSLX—Manual select L register. True in the step mode if L is selected for display.
MSPX—Manual select P register. True in the step mode if P is selected for display.
MSRX—Manual select R register. True in the step mode if R is selected for display.
MSTA—Manual set A register. Generates set pulses to A in the step mode when A is selected and a data switch is depressed.
MSTI—Manual set I register. Generates set pulses to I in the step mode when I is selected and a data switch is depressed.
MSTL—Manual set L register. Generates set pulses to L in the step mode when L is selected and a data switch is depressed.

MSTP—Manual set P register. Generates set pulses to P in the step mode when P is selected and a data switch is depressed.
MSTR—Manual set R register. Generates set pulses to R in the step mode when R is selected and a data switch is depressed.
MSTW—Manual set W register. Set pulses to the W register generated when MSTI or MSTR is generated.
OCY1—Operand cycle 1. Signal that is true when an operand is being read from or stored into the memory, or when the contents of the L register is being stored on a jump and save instruction.
OCY2—Operand cycle 2. Signal that is true when the contents of P is being stored in the memory during a jump and save instruction.
OCYE—Operand cycle enable. Signal that enables the OCY1 flip-flop indicating the next memory cycle is to be an operand cycle.
OSCX—Oscillator. 2-MHz pulse train derived from the basic oscillator in the processor.
OUTX—Output. Control signal on the I-O bus to indicate an output command is being executed. Two microseconds duration.
OVRE—Overflow reset enable. Signal generated during shifts that causes overflow to be reset.
OVSE—Overflow set enable. Signal generated during shifts that causes OV to be set.
OVXX—Overflow flip-flop. Set on arithmetic operations when the sign bit changes illegally, and on shifts or register change instructions.
PC00 to PC01—P counter (P register).
PH1X—Phase 1. 2-microsecond timing pulse that occurs each cycle.
PH2X—Phase 2. 2-microsecond timing pulse that occurs each cycle.
PH3X—Phase 3. 2-microsecond timing pulse that occurs each cycle.
PH4X—Phase 4. 2-microsecond timing pulse that occurs each cycle.
PH23—Phase 2 or 3.
PLSE—Pulse. Strobe signal (500 microseconds) on the I-O bus that is used by peripheral logic to form I-O command signals. True for input, output, and external control commands and during IARX.
PROT—Priority out. Signal from processor to I-O indicating that IL1X and IL2X are not active.
RC1X—Equal to (IR04)(IR05-). Used in register change logic

to indicate set or complement overflow.
RC3X—Register change phase 3. Signal that indicates phase 3 of a register change instruction.
RDXX—Read drive timing signal in memory.
REGS—Register selected. Signal generated in the step mode when any of the registers are selected for display.
RSXX—Read switch timing signal in the memory.
RSTX—Reset. Initializes control and peripheral logic. Caused by depressing the reset button on the console.
RUNX—Run. Signal is true when the controller is in the run mode; i.e., not stopped.
SB00 to SB07—S bus. Positive bus that routes information from A, P, L, M, and W registers to the adder.
SCMX—Sense condition met. Flip-flop that is set on a sense instruction if the sense response is obtained one microsecond after the beginning of the sense command.
SDBR—Select D bus receivers. Enables the contents of the D bus to be placed on the S bus.
SELA—Select A register. Signal that enables the contents of the A register to be placed on the S bus.
SELL—Select L register. Signal that enables the contents of the L register to be placed on the S bus.
SELP—Select P register. Signal enables the contents of the P register to be placed on the S bus.
SELR—Select R register. Enables the contents of the R register to be presented to the adder. If SELR and SLRN are both true simultaneously, zeros are presented to the adder on the U bus.
SELS—Select sum. Signal that enables the output of the adder to be placed on the A bus without shifting.
SETA—Set A register.
SETL—Set L register.
SETP—Set P register.
SETR—Set R register.
SETW—Set W register.
SFTX—Shift.
SHLI—Shift left input to least significant bit of A.
SHLX—Shift left.
SHR7—Shift right 7. Input to the most significant bit of A on a logical right shift.
SHRX—Shift right.
SKEX—Skip enable.
SKIP—Skip flip-flop. When true, indicates a skip is being performed and the L counter is being incremented twice. If L overflows, then P is incremented. Skip is true for 1.5 microseconds.

SLA1—Signal that is used to select the A register during the execution of an output A command or a masked input to A command.
SLAN—Select A Not. Signal that enables the 1s complement of the A register to be placed on the S bus.
SLL1—Signal that causes select L to be true while the skip flip-flop is set and L is being incremented twice. Also used to create two set L pulses.
SLL2—Signal that is used to cause select L during phase 1 of either ICY1 or ICY2 cycles for updating the L counter.
SLLM—Select lower M register. Signal that enables the M register contents to be placed on the S bus.
SLP1—Signal true during second half of phase 1 of each instruction cycle. Used to update the P register.
SLP2—Signal true during skip. Used to increment P if L overflows when incremented on a skip.
SLR1—Signal that is true during phase 4 of ICX2 during a memory reference instruction (except a direct jump). Used to form the address for the operand or address cycle.
SLUM—Select the upper M register signal that selects the contents of the M extension register on the X bus and thence onto the A bus.
SLRN-Select R Not. Enables the 1s complement of the R register to be presented to the adder.
SLW1—Signal true during phase 2 and 3 of the operand cycle of an increment and replace instruction. Allows W to be incremented during phase 2 and the results tested during phase 3.
SLW3—Signal true during ACX2 phase 3 of an indirect jump. Allows W to be transferred to P.
SLW4—Signal that is true during operand cycle phase 3 of an output A instruction. Allows W to be placed on the D bus.
S0XX to S7XX—Sum. Output of adder.
SSEX—Sense strobe enable. Generated during RDXX pulse.
STA2—Signal that is true if a sense condition is met during an I-O instruction that uses a sense test.
STA3—Signal that occurs during an input to A instruction. Used to create a set A pulse.
STA5—Signal that is true during the execution phase if an I-O instruction. (G1XX • ICY2 • PH3X)
STL1—Signal that is true during phase 4 of OCY2. This occurs only during the execution of a jump and save instruction where the contents of M plus 1 is being transferred to the L register.
STL2—Signal that is true during phase 3 of ICY2 during a direct jump. Used to cause the contents of the R register to be

transferred to the L register.
STL3—Signal that occurs during JMPX : ACY1 and phase 3 which is used to transfer the first word of an indirect address from the W register to the L register or an indirect jump.
STLM—Set lower M. Sets lower eight bits of the memory address.
STUM—Set upper M. Set the pulse to the M extension (ME) register.
STW1—Signal true during I-O operand cycle on an input to memory. (block input).
SUBX—Subtract. Signal true during operand cycle phase 3 of a subtract instruction.
T1XX—Timing pulse that occurs every two microseconds in the run mode.
T2XX—Timing pulse that occurs at the end of phase 2.
T3XX—Timing pulse that occurs at the end of phase 3.
T4DX—Delayed T4XX.
T4XX—Timing pulse that occurs at the end of phase 4.
T0XX—Timing pulse that occurs every microsecond in the run mode.
VXDX—Voltage for X drive. Analog voltage that supplies drivers in the X side of memory electronics.
VYDX—Voltage for Y drive. Analog voltage that supplies drivers in the Y side of memory electronics.
WDXX—Write drive timing signal in memory.
WEXX—Signal true when the memory cycle is to result in a store cycle. Enables the set W signal to be generated with T2XX.
WR00 to WR07—Word register. Word buffer register between the processor and memory.
WSXX—Write switch timing signal in memory.
XB01 to XB03—X bus. Used as an auxiliary input to the A bus. Pointer addresses and the interrupt addresses for IL1X and IL2X are transferred to L through the X bus.
XB00 to XB07—X bus. Auxiliary input to the A bus through the A bus driver. Allows information to be placed on the A bus without going through adder.
XIMX—Transfer I to M extension. The three bits of the pointer extension in a block I-O instruction are transferred to ME.
XPMX—Transfer P to M extension. P is normally transferred to ME at T4XX.

Chapter 16

System & Functional Components

The PDC 808, as with many small processors, is constructed on cards which plug into printed-circuit jacks wired on what is called a "back-plane." The pins on the printed-circuit jacks are numbered. Also, the connections (if any) to these pins are numbered and the numbers are shown in parentheses on the diagrams.

The logic diagrams that follow are the diagrams for the various registers. The upper left rectangle in Fig. 16-1 contains the designations PC01, 8, and 7475. The PC01 says that this latch is the bit 01 latch of the program counter register. The 8 is the IC number, again referencing the layout drawings not included in this book, and the 7475 is the type designator which can be referenced back to the diagrams shown in Chapter 15. The numbers on the input and output lines refer to the pin numbers on those same IC diagrams so that pin 2, referring back to Fig. 15-11 is the D or data input to that particular position.

For the logic gates, the number just outside and below each gate (upper left-hand corner for example) is the device number, in this case a Motorola Type 832 DTL device. The large rectangular block on the right-hand side is the adder for the two bits in Fig. 16-1 and the two bits in Fig. 16-2.

The DS01- signal mentioned earlier illustrates one of the problems in troubleshooting computers. This signal comes from some source but it does not indicate what that source is, so it becomes necessary to check the function index list.

PROCESSOR

The processor portion of the computer contains all the active registers, the adder unit, the control logic and the signal buses. There are eight registers in the processor (Figs. 16-1 through 16-4):

A register: The A register (AR) is the accumulator and the input-output buffer register. It is loaded through select gates. The contents of AR can be shifted, complemented,

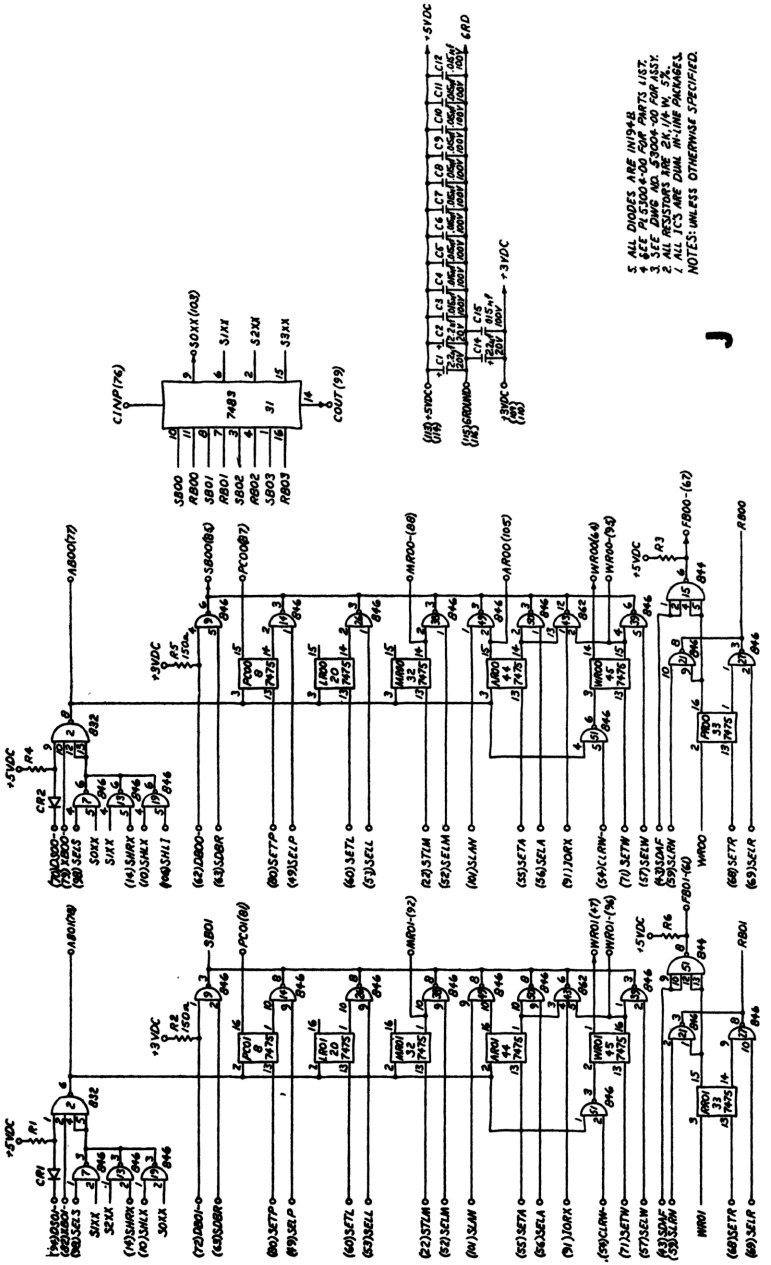


Fig. 16-1. Processor schematic, 1st and 2nd bits.

stored in memory, added, subtracted (memory subtracted from AR), cleared, outputed to the I-O bus, ORed with memory, ANDed with memory, and interchanged with memory. The register is eight bits long, constructed with "quad latch" integrated circuits and controlled by the following control signals: SELA, SLAN and SETA.

I register: The I register (IR) is the 8-bit instruction register that holds the single-word instructions being executed and holds the first word of double-word instructions. (The second word is held in the R register). The IR receives the instruction word from the W register, which is directly connected to IR and RR. The IR is loaded at T2 time during the instruction "fetch" cycle ICY1. The control signal associated with IR is: SETL.

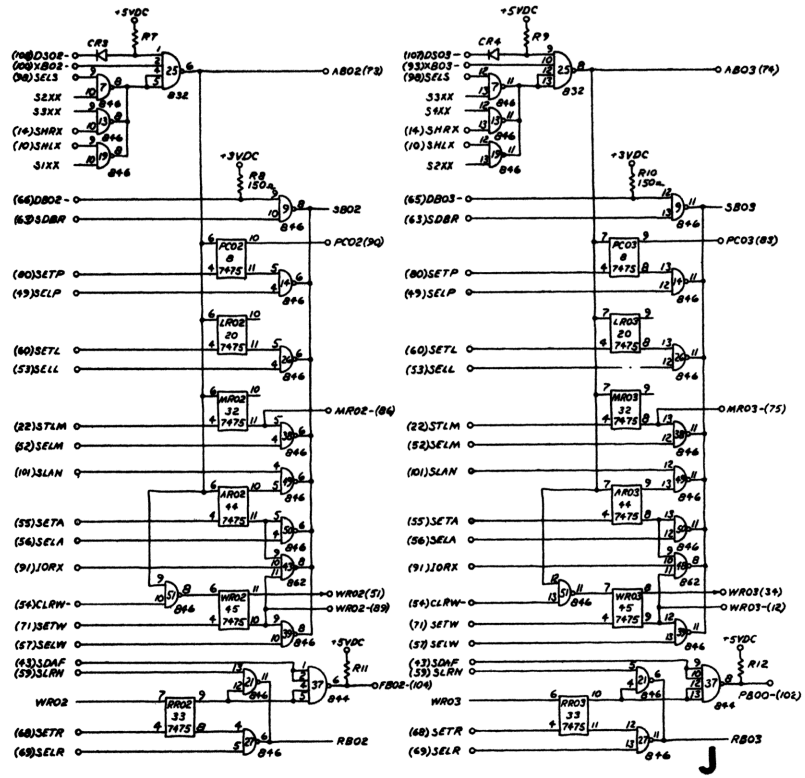


Fig. 16-2. Processor schematic, 3rd and 4th bits.

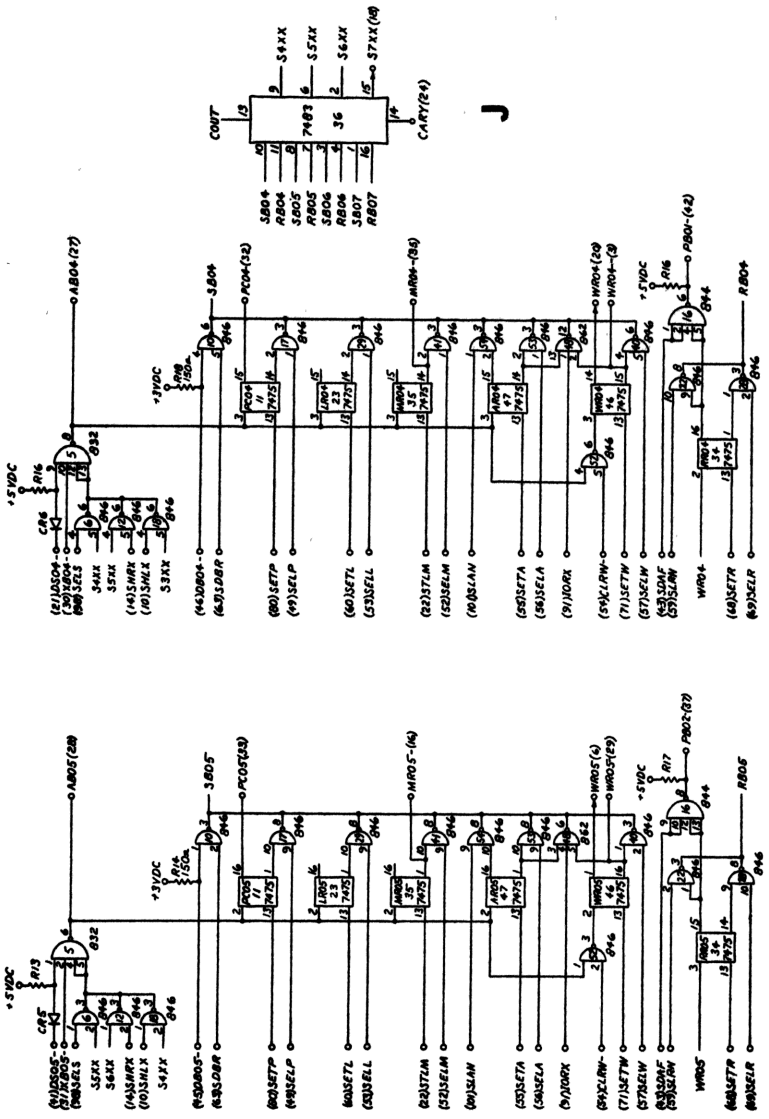


Fig. 16-3. Processor schematic, 5th and 6th bits.

R register: The R register (RR) is used as the operand address register for most memory reference instructions and also is used as an extension to the IR for double-word instructions. In that sense, it is a time-shared register. For instance, on a load A instruction, the second word of the instruction is held in RR until the operand from memory is obtained. The operand then replaces the instruction word in RR. For memory reference instructions, the contents of RR is the A field (the 8-bit operand address in the instructions) until the operand is obtained.

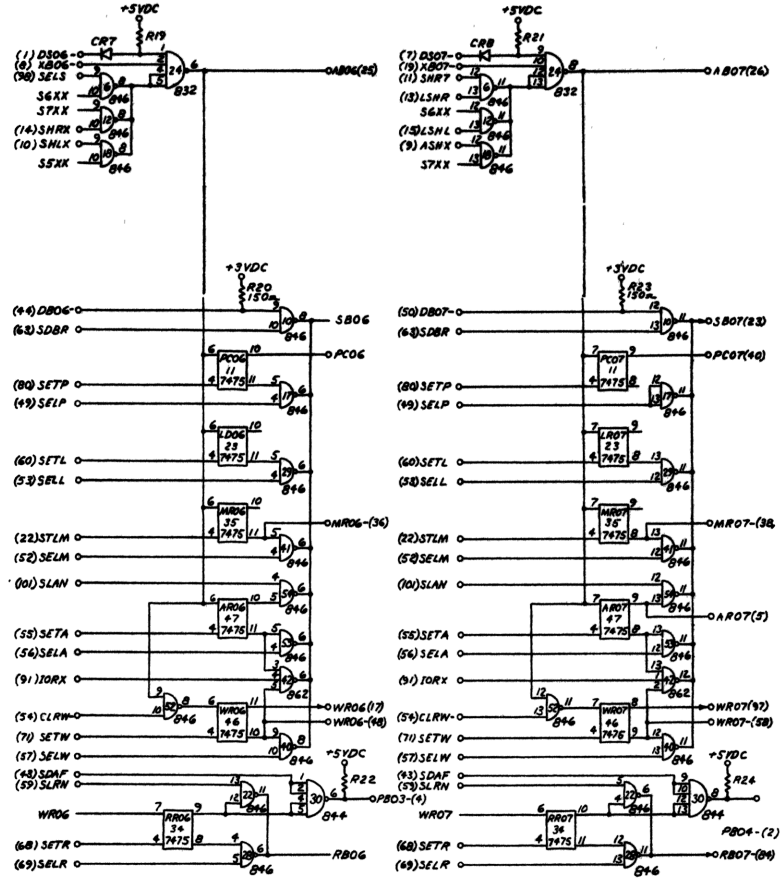


Fig. 16-4. Processor schematic, 7th and 8th bits.

For I-O instructions, RR contains the device address and function code portion of the instructions. The contents of RR is always seen on the P bus and the F bus in the I-O cable. The control signals associated with RR are: SELR, SLRN and SETR.

W register: The W register (WR) is the memory data word register that interfaces the processor to the memory. Any data to or from memory passes through the WR (except in the case of direct memory access). The WR is eight bits long (quad latch circuits) and is unconditionally cleared at the beginning of each memory cycle. The data from memory is read into the WR by "collector setting." That is, the reset side of the W flip-flop is pulled to ground by the sense amplifier if a 1 is detected in memory. This causes the set side of the flip-flop to go true, entering a 1 in the stage.

If data is to be stored into memory, it is transferred to W (replacing the data just read out of memory) just prior to the restore cycle. Thus the memory thinks it is always operating in a read-restore mode, but may be restoring completely new information. The WR is controlled by the following control signals: SELW, CLRW and SETW.

L register: The L register (LR) is the first half of the program sequencer. It holds the lower eight bits of the memory address where the next instruction word is to be obtained. Since LR is eight bits long and can specify 256 locations, it can be thought of as specifying the location within a 256-word page or sector of memory. For that reason it is sometimes referred to as the location counter.

The LR is incremented during the first phase of each instruction cycle, except for interrupts. The contents of LR is transferred to the M register just prior to starting an instruction cycle. LR is loaded from the A bus and the contents of LR is placed on the S bus through select gates. The signals associated with LR are: SELL and SETL.

P register: The P register (PR) is a 7-bit register that forms the upper half of the program sequencer. It is basically an extension to the LR. It holds the higher order portion of the memory address where the next instruction is obtained. Since LR specifies the word within a page of memory, the P register can be thought of as specifying which page. Thus it is sometimes referred to as the page counter.

The PR is incremented on an instruction cycle if the LR overflowed when it was incremented. Thus in very simplified form, the carry out of LR carries into PR. LR and PR constitute what is normally called the program counter in a computer. The contents of PR is normally transferred to the ME register just prior to an instruction cycle. There is a direct connection from the PR to the input gates to the ME register. The control signals associated with PR are: SELP, XPMX and SETP.

M register: The M register (MR) is an 8-bit register that holds the lower portion of the memory address while the memory cycle is in progress. It can thus be thought of as part of the memory address register. The MR is loaded from the A bus and the contents of MR can be selected onto the S bus through select gates.

The output of MR is wired directly to the memory modules. Only one side of each MR flip-flop is used since it is inverted in the memory addressing circuits to provide double-rail signals. The control signals associated with MR are: SELM and SETM.

M register extension: The M register extension (ME) is a 6-bit register that holds the upper portion of the memory address during a memory cycle. The ME and M registers together constitute a memory address register of 14-bits sufficient for 16,384 words of memory.

The ME receives information from PR, WR and IR, and is set to zero when addressing page 0. The output of each stage of ME is directly ORed to the memory similar to the M register. The control signals associated with ME are: STUM, XIMX, XPMX and RJIX.

Adder unit: The adder unit (AU) consists of two "quad adder" chips, each with four adder stages complete with serial carry and some associated gating. The two 8 bit words presented to the AU come from the S bus and the R bus. The R bus is the gated output of the R register.

An input carry allows adding 1 to the sum of R and S. The adder output is an 8-bit word which is the sum of R and S and a carry (if any). The output is gated through one level which allows shifting left or right if desired.

Since almost all instructions involve the AU, and not all instructions need additions, it is worthwhile to notice how data is handled for different cases.

Case 1. Normal addition of accumulator and operand: The contents of AR is selected onto the S bus and the contents of RR is selected onto the R bus. The carry into the adder is made zero. The sum is R and S which is selected onto the A bus through the SELS gate.

Case 2. Subtraction of R from A: AR is selected onto the S bus. The 1s complement of R is selected onto the R bus. By inputting a carry the 2s complement of R is created. The result is A - R.

Case 3. Transfer of A to W: A is selected onto the S bus. Nothing is selected onto the R bus, creating all 1s. Carry in is true. The result is A, which is strobed into WR.

Case 4. Increment A: The contents of A is placed on the S bus. Both R and $\overline{R}$ are selected to insure the R bus is all zeros. Carry in is true. The result is A +1.

Case 5. Decrement A: The contents of A is placed on the S bus. Neither R nor $\overline{R}$ is selected, leaving the R bus all 1s (-1). Carry in is false. The results is A-1.

Case 6. Shift A: The contents of A is placed on the S bus. Carry in is true. Neither R nor $\overline{R}$ is selected, leaving the R bus all 1s. SELS is false. SHR or SHL is true, depending on the desired direction of shift. Thus the contents of the A bus is equal to the S bus except shifted one place.

Transmission Buses

S bus: The S bus routes the output of the A, L, P, M, and W registers and the D bus to the adder unit. It is all ones when nothing is selected onto it. The A bus distributes the output of the adder unit (after the two levels of gating) to the A, L, P, M, and W registers and also to the D bus drivers and the console light drivers. It is ground when nothing is selected onto it.

R bus: The R bus routes the output of the R register to the adder unit. It is all 1s except when R or $\overline{R}$ is selected onto it.

The M and W buses are described in the memory section and the D bus is described in the I-O sections. (Fig. 16-5 shows the processor timing circuits.)

CONTROL

The control logic of the processor provides the timing, decoding, and sequencing required for the instructions. The basic timing element is the cycle, the time required for the memory to execute a full read-restore cycle. The cycle is broken into four equal phases of 2.0 microseconds. The sub-phase timing pulses are derived from the count-down timing

chain driven by the 2.0-MHz crystal oscillator. Fig. 16-14 shows the primary timing signals in the processor and Figs. 16-6 and 16-7 the processor control circuits.

There are six types of cycles in three groups. The first we consider are the instruction cycles, ICY1 and ICY2. When the memory cycle is being used to retrieve an instruction, the cycle is an ICY1 or ICY2, depending on whether the first or second word of an instruction is being recalled, respectively.

ICY1 is set at the same time the memory cycle is started to retrieve the first instruction word. Halfway through the cycle, the instruction word is transferred from WR to IR (at T2). If the first word indicates that the instruction is a double-word, then ICY2 is set at the same time the next memory cycle is started. Halfway through ICY2, WR is transferred to RR, completing the instruction formation.

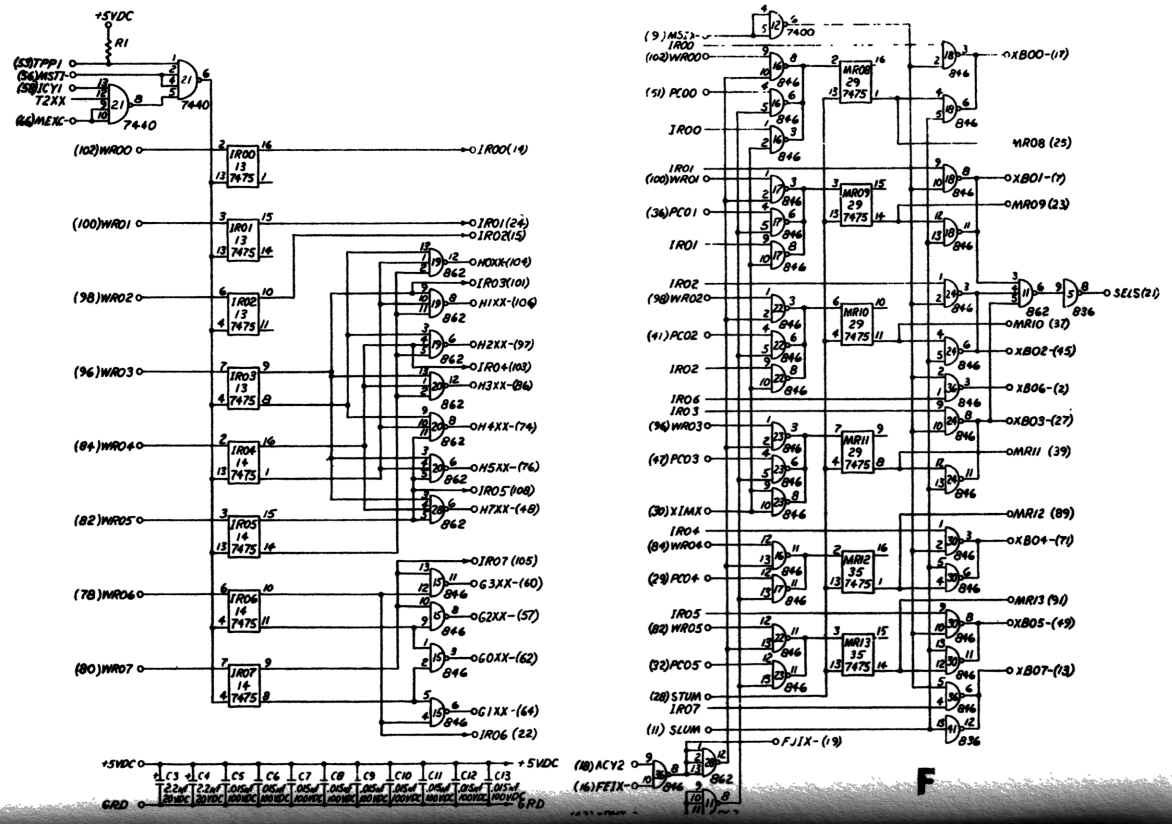
The second type of cycles are the operand cycles, OCY1 and OCY2. When the memory is referenced for data (data is stored into memory or retrieved from memory) the memory cycle is referred to as an operand cycle. For example, during the execution of an add instruction, the operand in memory must be obtained and added to the contents of the accumulator. When the memory cycle is started to recall the operand, the OCY1 flip-flop is set. Halfway through the cycle, the operand is available in the W register instructions during block I-O instructions. OCY2 occurs only during the jump and save instruction, when the contents of the P register is stored in memory.

The final type to be considered are the address cycles, ACY1 and ACY2. When the indirect addressing mode is used with memory reference instruction, the indirect address is obtained from two memory locations. ACY1 is set at the beginning of the first indirect cycle to indicate that the information being obtained is the first part of an address. ACY1 occurs during the block I-O instructions also. The input or output pointer is read from memory, incremented, and replaced during ACY1. ACY2 can occur only during memory reference instructions.

Instruction Addressing

In the 208 and 808 an instruction can make reference to two types of addresses: direct address and indirect address.

Direct Address: The direct address is the address specified by the second word of a double-word instruction. The A field specifies the location within a page and the page is



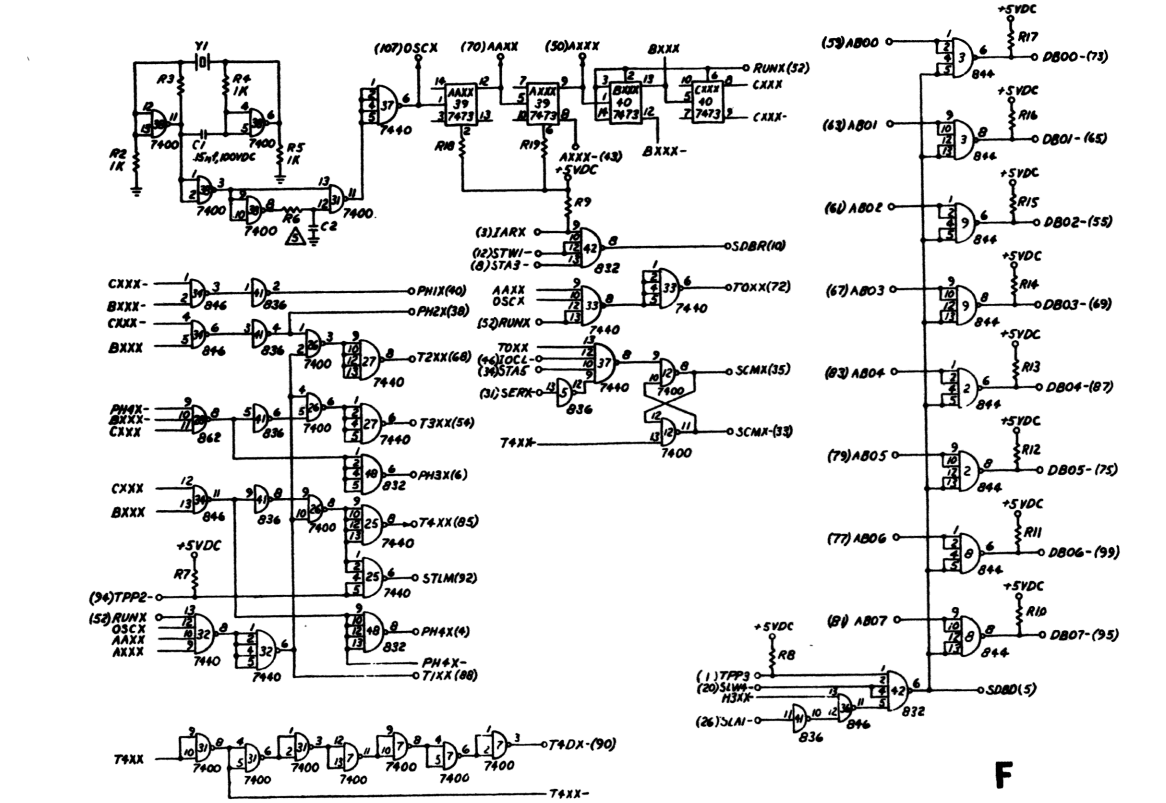
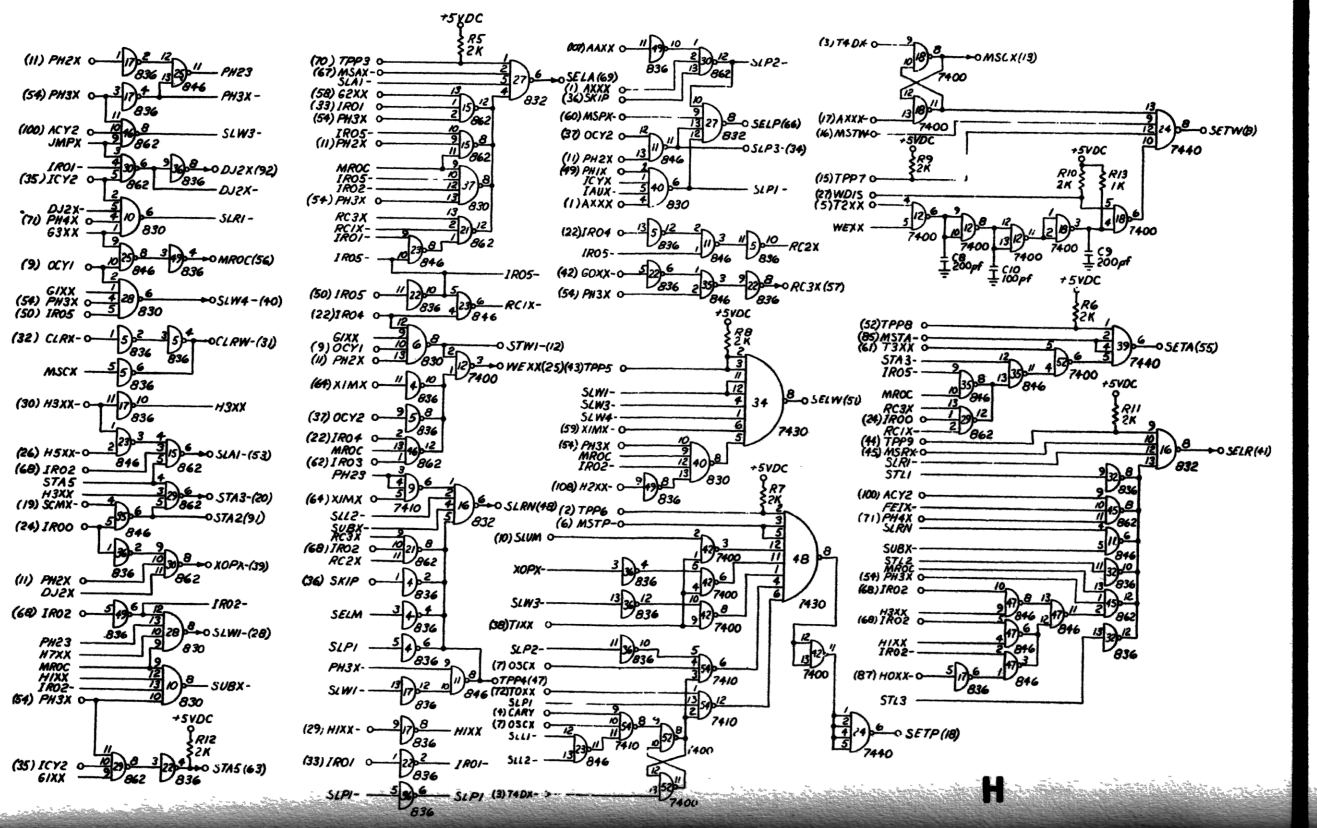


Fig. 16-5. Processor timing circuits.



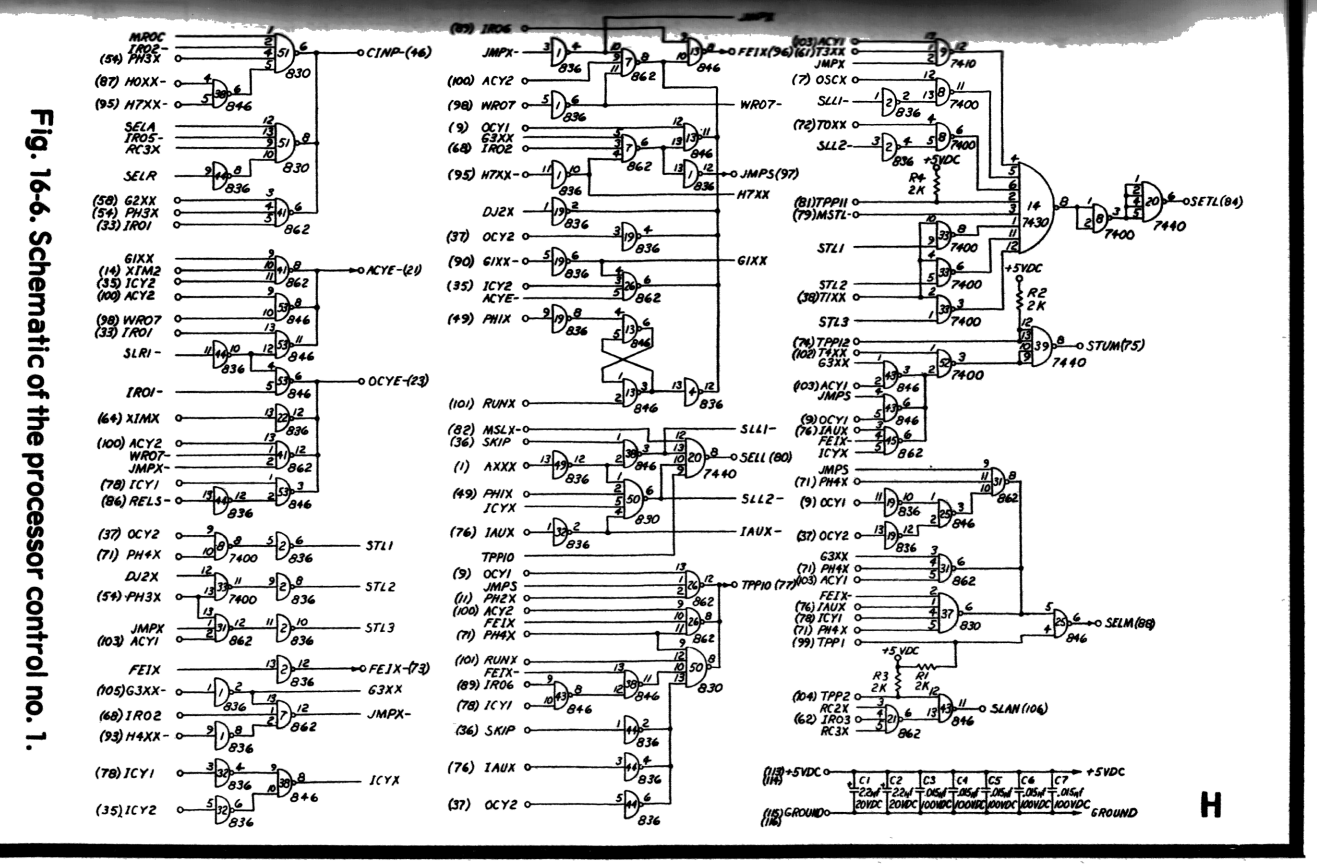


Fig. 16-6. Schematic of the processor control no. 1.

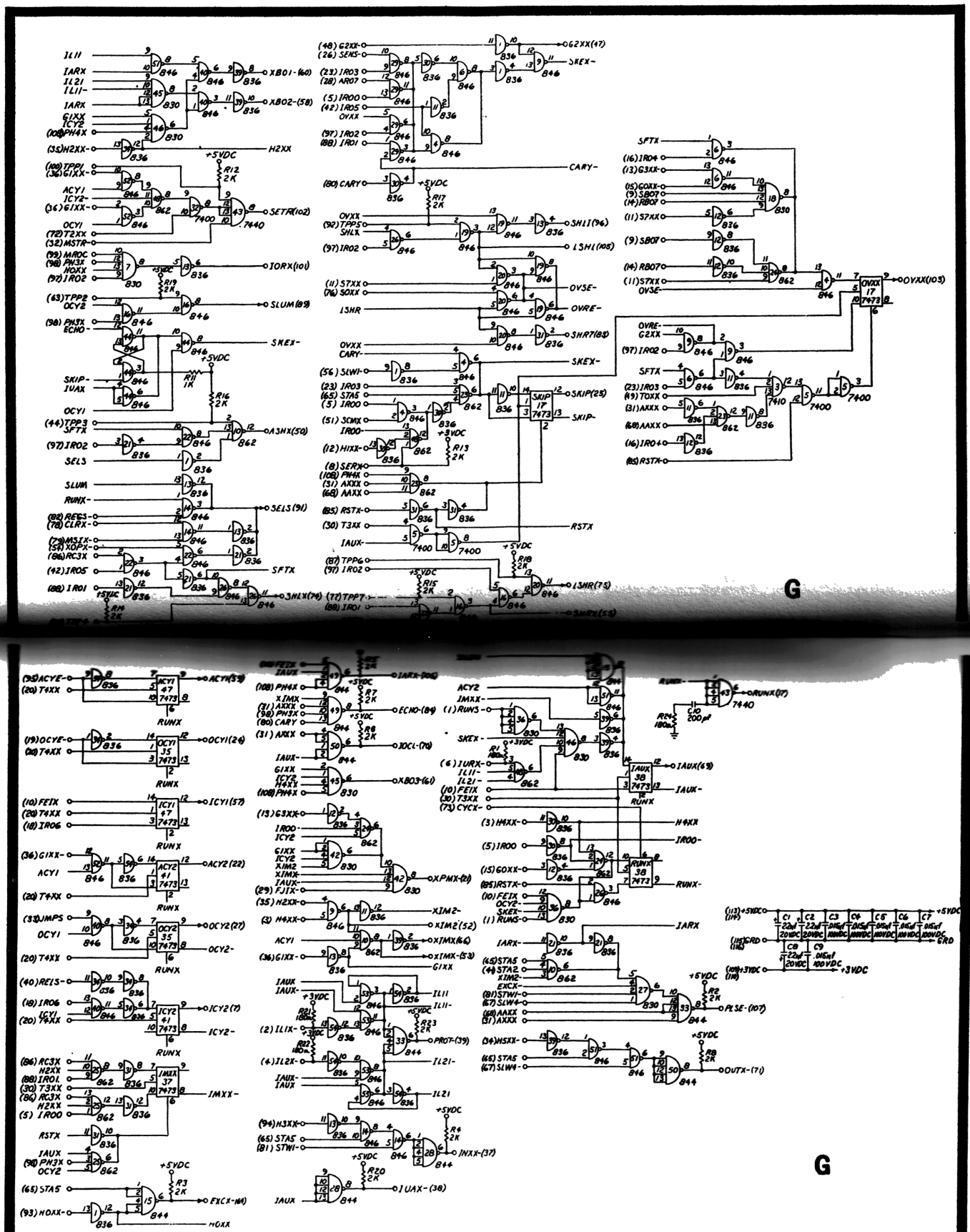


Fig. 16-7. Schematic of the processor control no. 2.

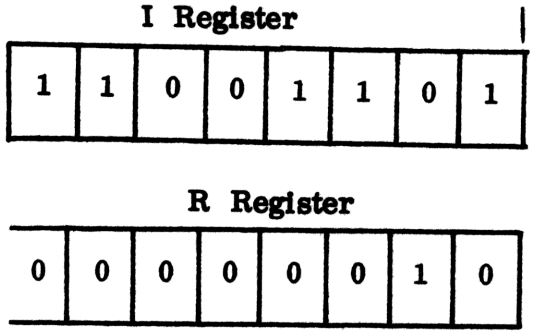


Fig. 16-8. Example of direct addressing.

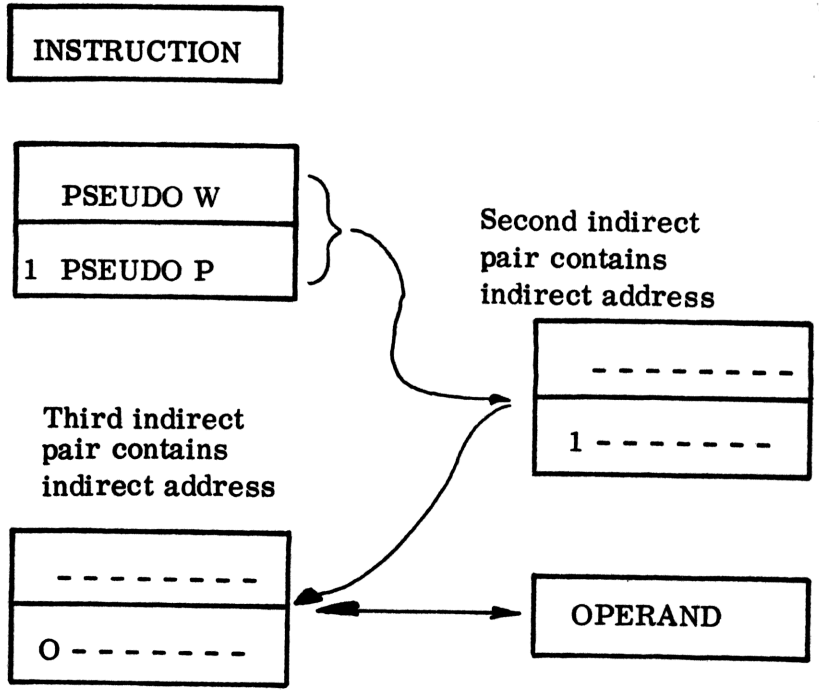


Fig. 16-9. Example of multi-level indirect addressing.

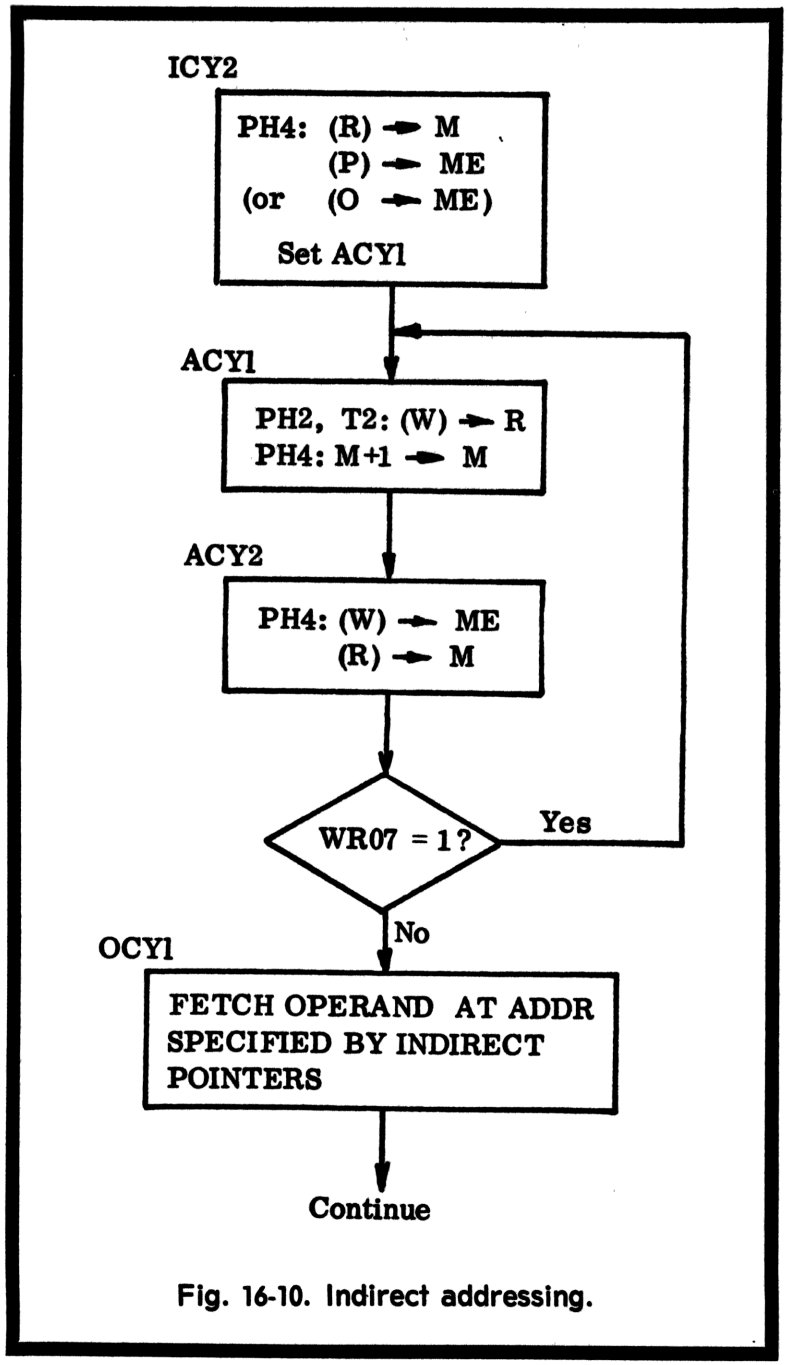


Fig. 16-10. Indirect addressing.

specified by a page tag. The page is either the zero page (P equals 0) or the current page (P equals 1). The address tag specifies either direct (A equals 0) or indirect (A equals 1) addressing. Direct addressing allows 512 memory locations to be reached directly. For example, the LDA instruction shown in Fig. 16-8 loads the contents of memory location 002 into the A register.

Indirect addressing (refer to Fig. 16-10). To reach locations outside this 512 word area, indirect addressing is used. This is accomplished by storing in two consecutive memory locations the location and page of the operand. The contents of the first word defines the location within a page and the contents of the second word defines which page. By setting the indirect bit in a memory reference instruction, the computer will access the location specified by the A field and page designator. The contents of that location is placed in the R register and the next memory location is accessed to obtain the page number. This is transferred to ME while R is transferred to M and the operand cycle started.

Multi-level indirect addressing: In some cases, it is not the operand address that is stored in the two memory words by the address of the operand address. To determine the difference, the computer examines the most significant bit of the second word (the page number). If that bit is a 1, then the contents of the two memory words is an address of an address rather than an address of an operand, and another level of indirect addressing is started in which another pair of memory locations is examined. See Fig. 16-9. The procedure for obtaining the indirect addresses is as follows:

1. During PH4 of ICY2, the contents of R (the A field) is transferred to M while ME is set to zero or is loaded with the contents of P. At T4, ACY1 is set and the memory cycles.

2. The first indirect word is transferred from W to R at T2 where it is held temporarily. The previous contents of R is lost.

3. During PH4 of ACY1, the contents of M is incremented to allow accessing memory for the next indirect pointer. M is not supposed to overflow since there is no provision for incrementing ME; thus the two indirect locations must always be in the same page. At the end of PH4, after M has been routed through the adder and incremented, ACY2 is set with PH4 and the memory cycled to get the second indirect pointer.

4. During PH4 of ACY2, the contents of R is transferred to M and the contents of W is transferred to ME. If WRO7 is true, indicating that another level of indirect addressing is required, then ACY1 is enabled. Otherwise, OCY1 is enabled. At T4, the memory is cycled again to retrieve operand or another indirect address.

MEMORY

Inhibit Driver: There are two inputs to the inhibit driver: Data input which is positive true and comes from the W register in the processor, and the WZSX strobe which comes from the inhibit strobe circuit. The data input acts as an enable to the driver to allow the driver to respond to the WZSX strobe signal. (See Fig. 16-11.)

Driver switch circuits: The driver-switch strobe circuits are used to strobe the switches and drivers in the XY drive electronics in the memory. The operation of the strobe circuit is identical to the inhibit strobe circuit except that +12v is used. There are four strobe circuits:

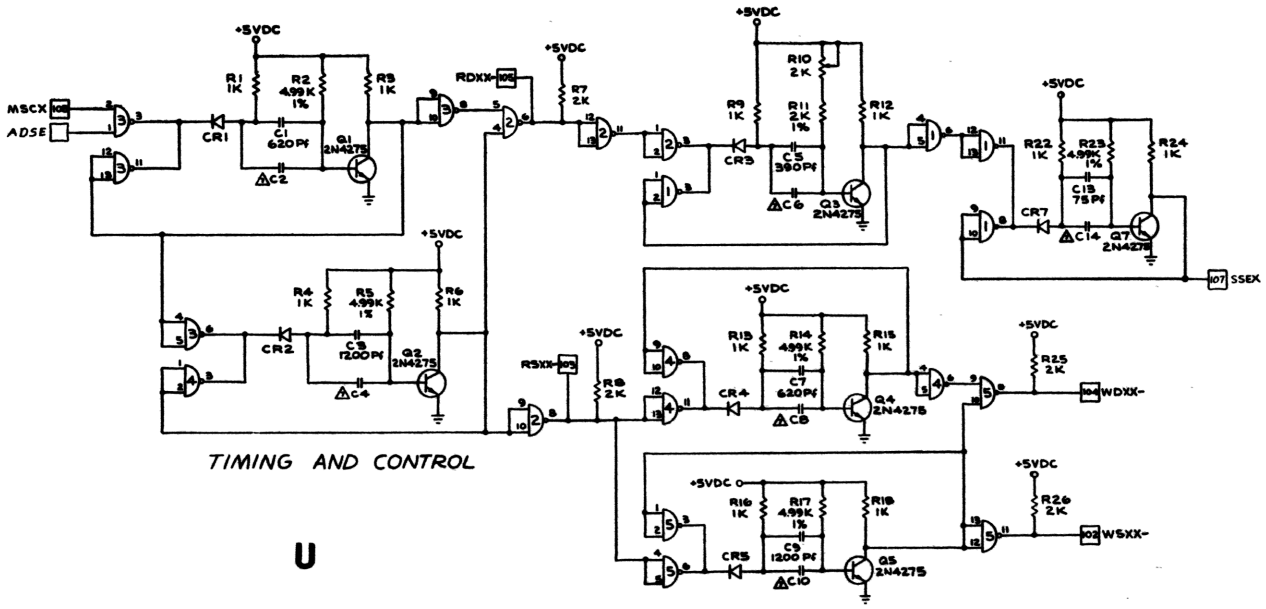
Read switch strobe, RSSX
Read driver strobe, RDSX
Write switch strobe, WSSX
Write driver strobe, WDSX

The strobe circuits pulse the X and Y drivers and switches during the read-restore memory cycle.

Fig. 16-12 shows the typical driver-switch circuit. It is identical for X and Y. The driver portion supplies current from a positive source (Y drive) and the switch sinks the current to -12v. The driver and switch in the same circuit are never turned on at the same time. Instead the driver on one side of the stack drives into a switch on the other side of the stack.

The operation of driving the stack is as follows: During read time, one driver-switch pair on each side of the stack is enabled. The driver on the left side is pulsed by RDSX and the switch on the right side is pulsed by RSSX. Thus current flows from the driver through the stack to the switch. During the write time, the circuits reverse functions to drive current back through the same stack line in the opposite direction. The driver is pulsed by WDSX and the switch is pulsed by WSSX. The outputs of the timing circuitry go to the following places:

A. RDXX-, WDXX-, RSXX-, WSXX-, go to the data board (Fig. 16-24) where they are amplified, inverted and routed to the driver-switch board.



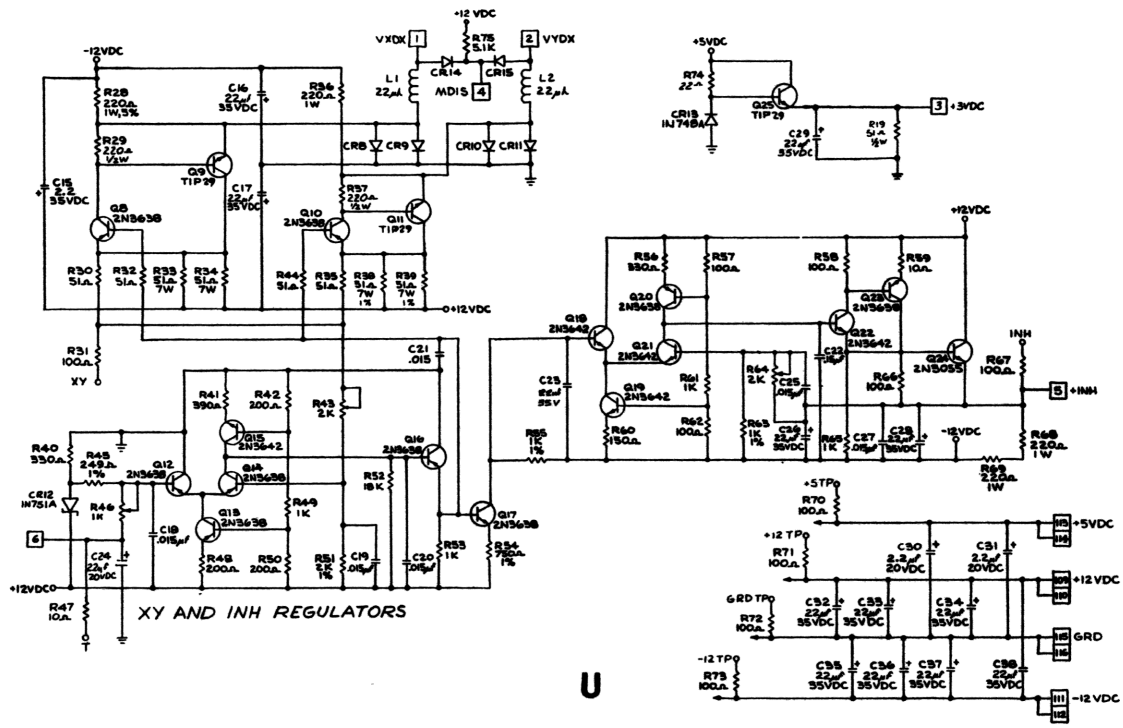
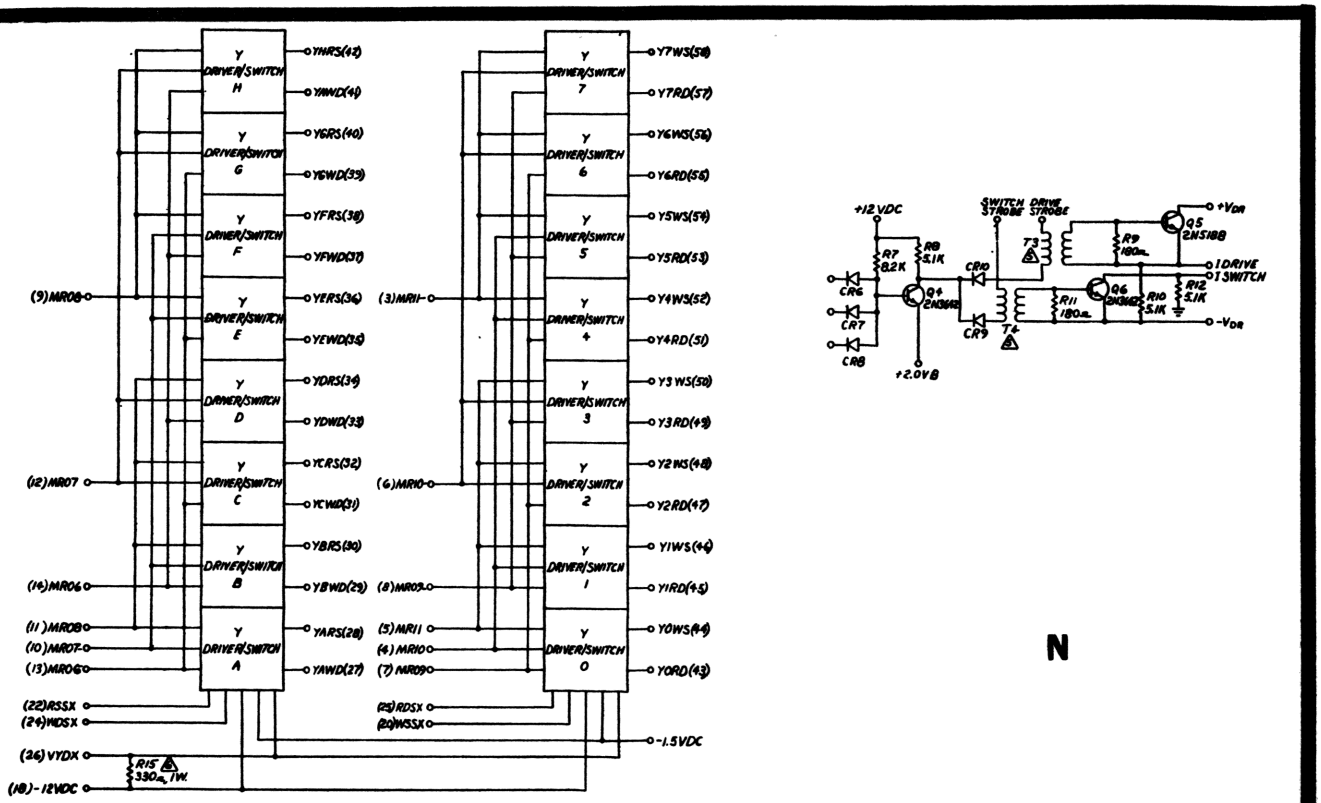
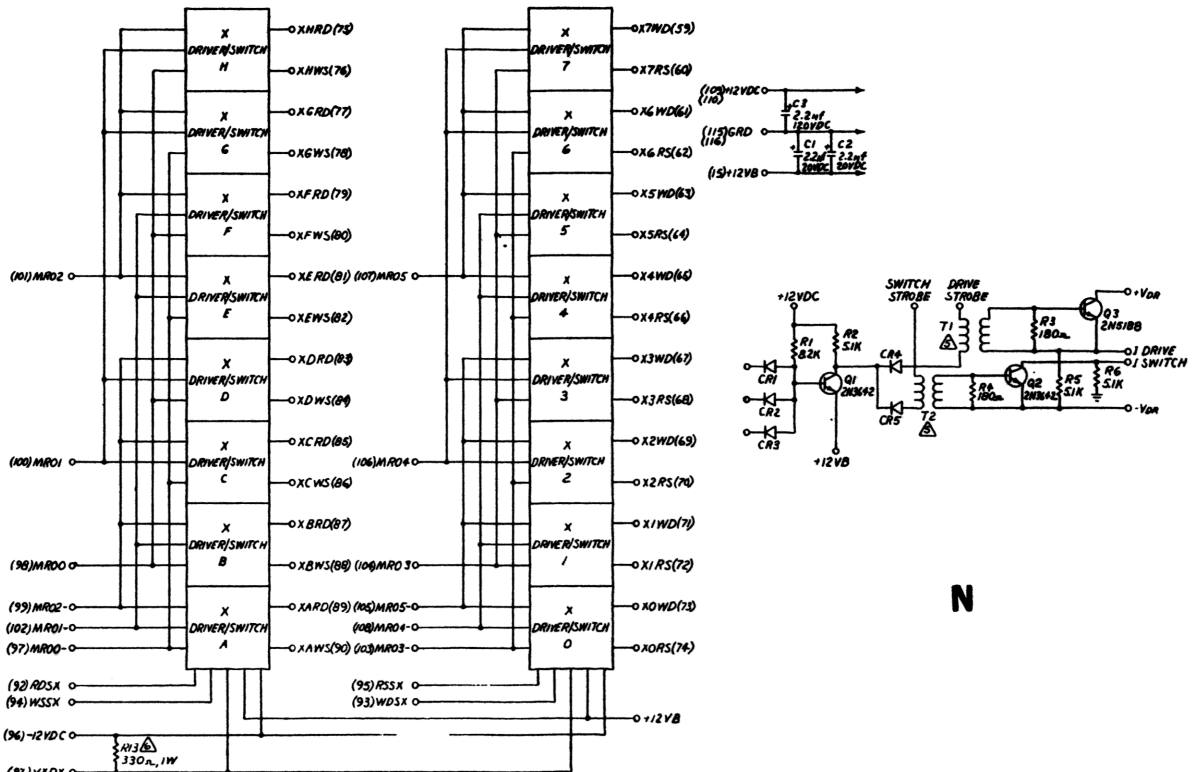


Fig. 16-11. Memory regulator schematic.



N



N

Fig. 16-12. Schematic of the driver-switch circuits.

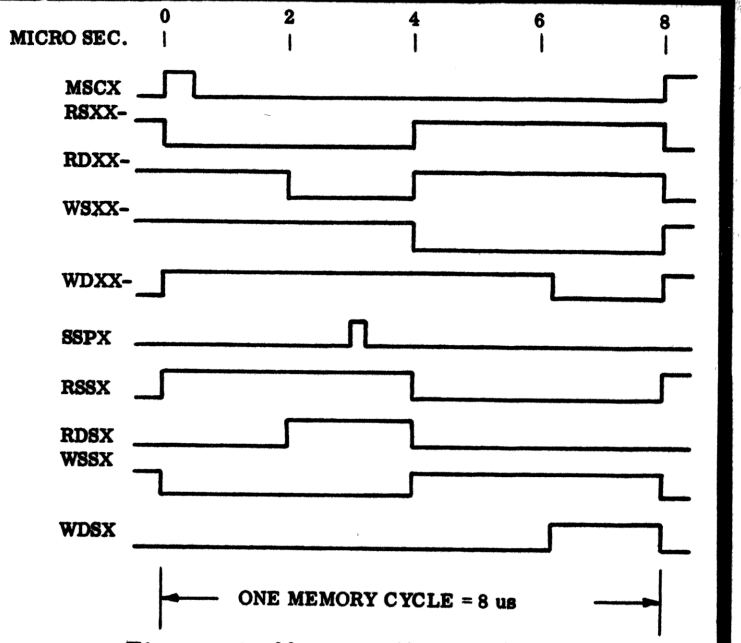


Fig. 16-13. Memory timing diagram.

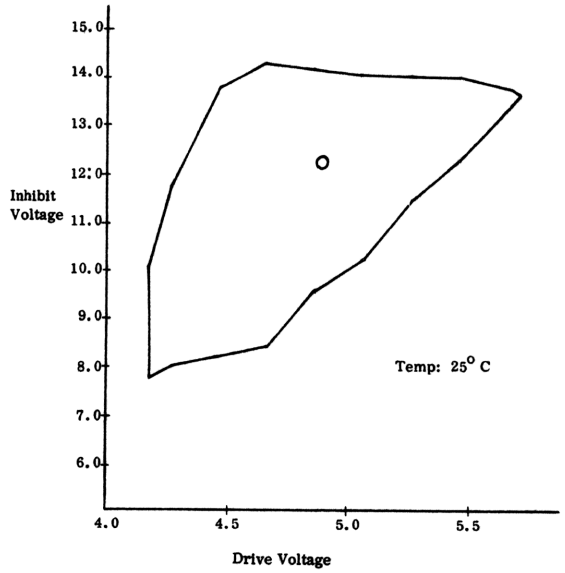


Fig. 16-14. Typical "shmoo" diagram.

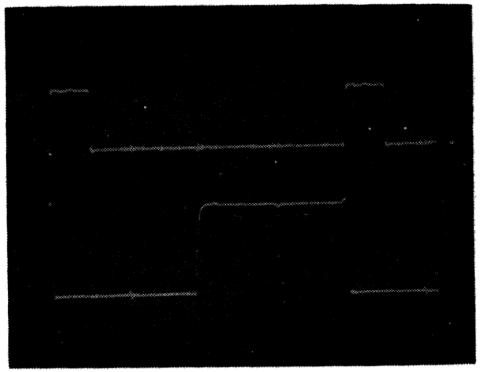


Fig. 16-15. RSXX pulse viewed with a 1 microsec per div. sweep, 2v per div. Top pulse is the master clock start (MSCX).

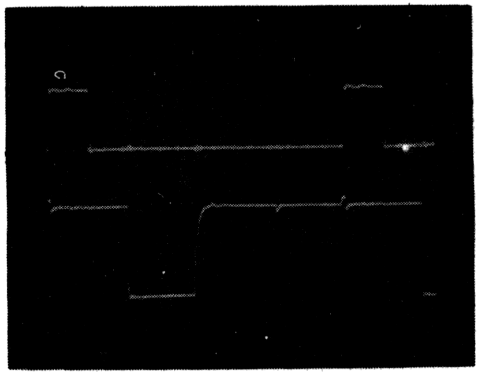


Fig. 16-16. RDXX pulse (bottom) and MSCX. 1 microsec per div. and 2v.

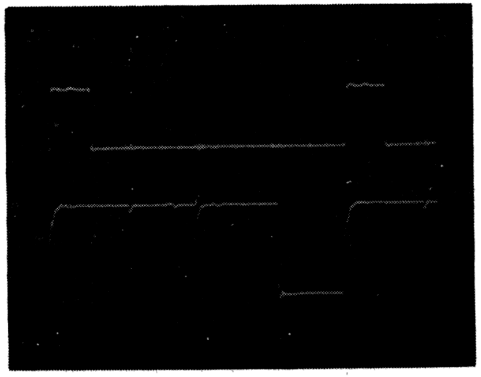


Fig. 16-17. WDXX pulse (bottom) and MSCX. 1 microsec per div. and 2v.

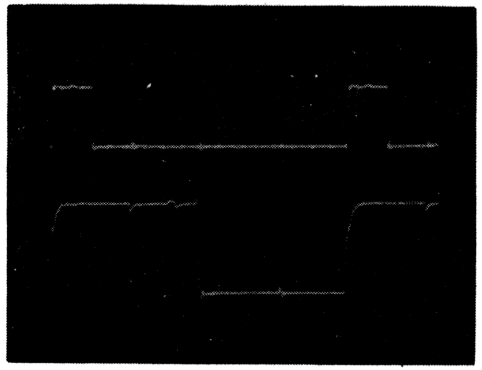


Fig. 16-18. WSXX pulse (bottom) and MSCX. 1 microsec per div. and 2v.

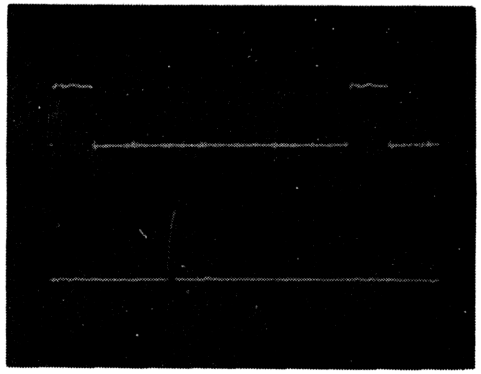


Fig. 16-19. SSPX pulse (bottom) and MSCX. 1 microsec per div. and 2v.

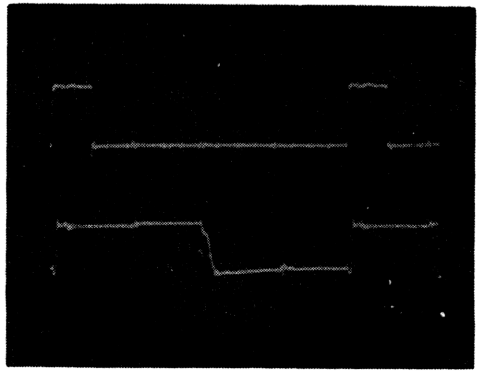


Fig. 16-20. RSSX pulse (bottom) and MSCX. 1 microsec div. and 2v.

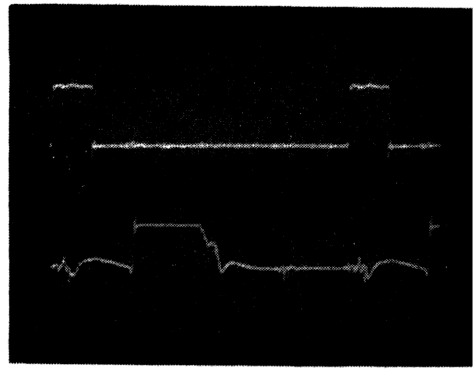


Fig. 16-21. RDSX pulse (bottom) and MSCX. 1 microsec per div. and 2v.

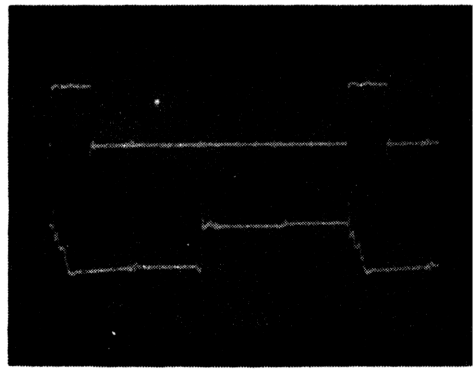


Fig. 16-22. WSSX pulse (bottom) and MSCX. 1 microsec per div. and 2v.

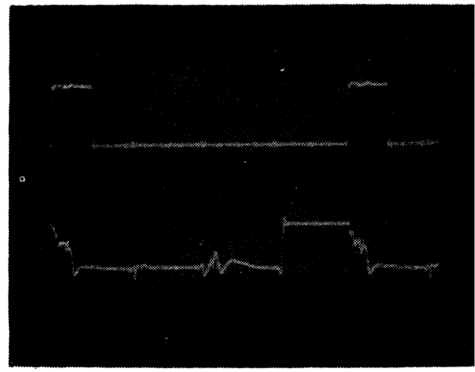


Fig. 16-23. WDSX pulse (bottom) and MSCX. 1 microsec per div. and 2v.

B. SSEX is the sense amplifier strobe and is routed to the data board.

C. ISSX is not used.

The 3-volt clamp voltage for the I-O cable is also generated on the regulator board by a zener diode and transistor. The timing chain for the memory is a series of one-shots formed by delay networks with feedback. The timing sequence is shown in Fig. 16-13. A typical shmoo curve is shown in Fig. 16-14. Figs. 16-15 through 16-23 show the various timing signals (bottom waveforms) and the master clock start (top waveform). All waveforms were taken on a Hewlett-Packard Model 180 oscilloscope.

The D-S circuits are enabled by address lines coming from the processor. The 3-diode AND gate enables Q1 which when conducting provides a path for current through the pulse transformers T1 and T2 (Fig. 16-12).

There are 64 X drive lines and 64 Y drive lines in a 4096-bit memory plane. The decoding diodes on the stack connect these 64 lines to eight drivers and eight switches on each side of the place. Thus an 8x8 matrix is created. Each driver can drive into eight switches, and each switch can service eight drivers.

Memory regulator board: The regulator board (Fig. 16-11) contains circuitry to regulate the XY drive and inhibit current to the memory, and contains the timing circuitry to control the current pulses, sense amplifier strobe, and inhibit strobes. The current regulator circuits are a controlled differential amplifier whose output can be adjusted by R46 and can also be changed by the resistance of thermistor RT1. The thermistor resistance decreases with temperature, causing the output currents to the memory to decrease. The rate of decrease of current per unit decrease of RT1's resistance is controlled by R46. The larger the value of R46, the less effect RT1 has. The output current level for the X and Y drive currents is determined by R43. The current range is 170 to 220 ma under normal operating conditions. The inhibit current regulator is slaved to the XY regulator. The offset is determined by R64. After the ratio of XY to inhibit currents has been set by R64, the inhibit current will track XY as the temperature changes.

Input-output devices

The following I-O devices are available on the 808 and 208:

1. 33 teletype option
2. High-speed reader option
3. High-speed punch option

Teletype interface: The teletype interface performs parallel-to-serial conversion when outputting to the TTY and serial-to-parallel conversion when inputting from the TTY. All data transmission between the computer and the TTY interface is parallel. The interface consists of the following major portions (Fig. 16-25):

A. **Buffer-shift register:** This register is loaded in parallel by the computer and the contents can be transferred in parallel to the computer. After data is loaded into the register by the computer, it is serially shifted out to the TTY. When the TTY is transmitting data, it is serially shifted into the register. Notice that the register is used for both input and output.

B. **Divide by 11 counter:** This counter is a 4-stage ripple counter with feedback to cause divide-by-11. It is used to measure the 100 ms period required for each character transmitted to or from the TTY. The counter stages are labeled BC01 through BC04 (Fig. 16-26). The input to the counter is from a flip-flop which divides the oscillator output by 2.

C. **Oscillator:** A gated multivibrator is used to synchronize the shifting operations. The oscillator is turned on by the write flip-flop, which is automatically set by an output command and by the read flip-flop, which is set by a space from the TTY if the read select flip-flop (RSEL) is on. The oscillator is turned off at the end of the 100 millisecond period by BC04, resetting either write or read (depending upon which is on). The positive-going edges of the oscillator output (Q2) are used to form narrow pulses which trigger flip-flop 24. The first pulse is not allowed to trigger 24-1. The flip-flop 24-5 disables 24-1 until the first pulse has occurred. This is required since the output of the oscillator goes high as soon as the oscillator is enabled.

D. **Shift pulse generator:** When outputting data to the TTY, shift pulses are required at the beginning of each bit time. When inputting from the TTY, shift pulses are required in the middle of each bit time. The phase shift is accomplished by using the set output of flip-flop 24-1 to form the read shift and the reset output to form the write shift. In both cases the positive-going edge of the flip-flop output is used to form a narrow (200 ns) pulse, which is gated with write or RSEL.

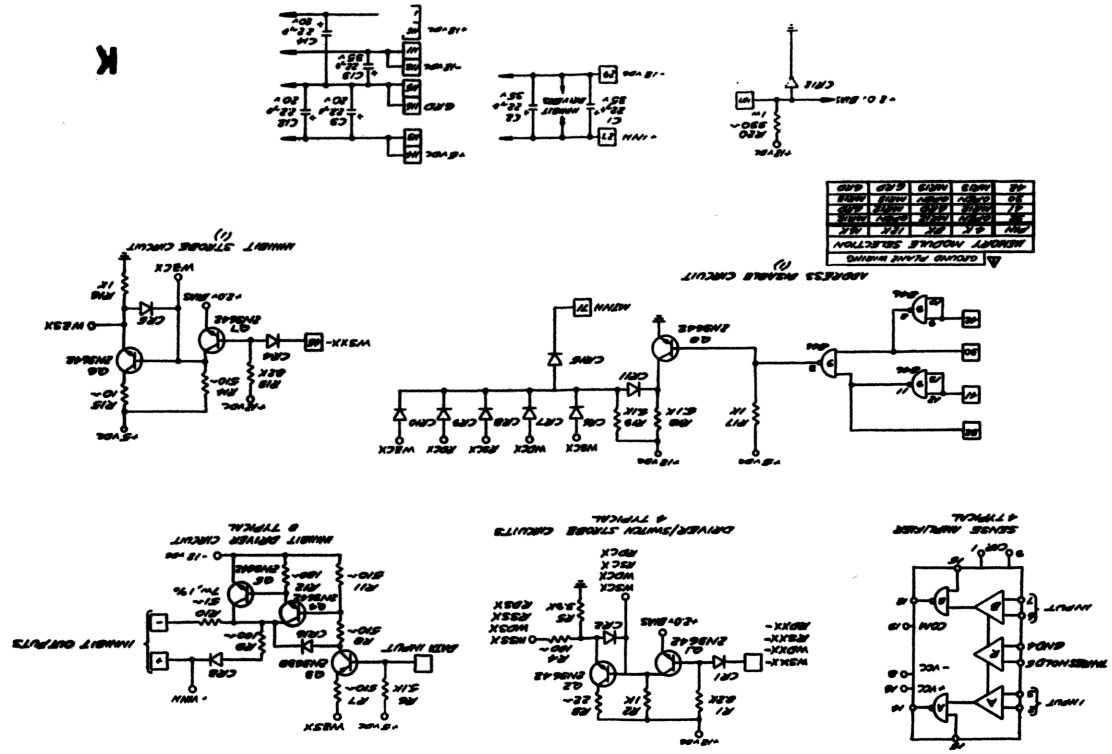


Fig. 16-24. Schematic memory data board.

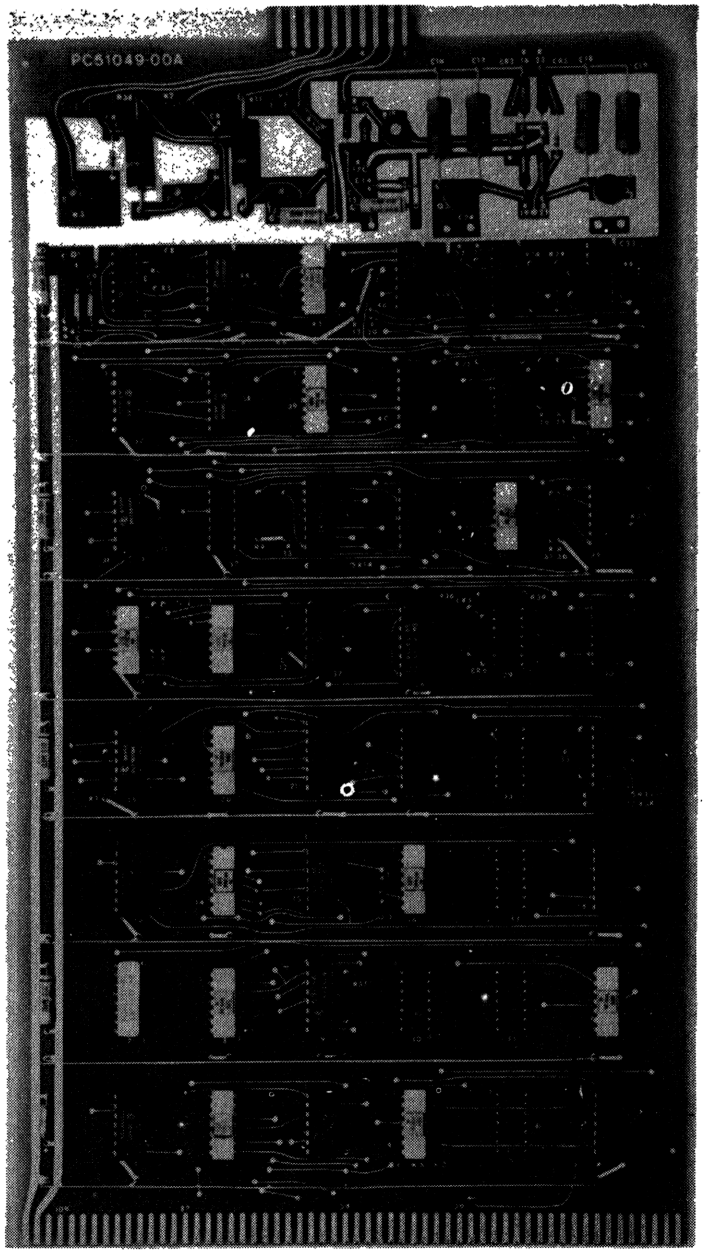


Fig. 16-25. Teletype controller circuit board.

E. Control flip-flops: There are six flip-flops used for control and status:

Buffer ready (BUFR) is used to indicate when the interface is ready to receive or transmit data to the computer. It is reset by an input or output command and by external control commands 71, 72, and 73. It is set by the external control command 74 (initialize) and by BC04.

Read Select (RSEL) is set when the computer selects keyboard, step read on continuous read. When set, RSEL allows data to be shifted from the TTY into the shift register. It is reset by the initialize commande (74) and by an output to the interface (which implies write).

Write (WRTE) is set on an output command and is reset by BC04 at the end of the 100 ms period. When set, it enables the oscillator and gates the oscillator to the proper path for generation of write shift pulses. It is also reset by the initialize command.

Read is set by a space received from the TTY (if RSEL is on). (The first bit of the TTY is always a space which is used as a synchronizing bit.) When set it enables the oscillator. It is reset by BC04 at the end of the 100 ms period.

Step is set by the EXC 72 command and is reset about 20 ms later by feedback from a delay network. When set, it causes relay K3 to close, completing the current loop in the reader solenoid circuit, causing the TTY reader to read the character under the read station and advance the tape one position. (The switch on the TTY reader must be in the run position for this sequence, since K3 and the switch are in series with the solenoid.

Continuous (CONT) is a flip-flop that is set to cause the TTY reader to run continuously. It is set by EXC 73 and reset by EXC 77 and the initialize command.

F. Sense logic: Three functions in the TTY interface may be sensed by the computer: buffer ready, word transfer mask on, and oscillator enable off.

G. Interrupt logic: The TTY interface is equipped with drivers and masks for interrupt lines 1 and 2. Interrupt 1 is created by BUFR if the word transfer mask (WXMX) is set. Interrupt 2 is created by WXMX being off and the block transfer mask (BXM) being on. Both WXMX and BXM are reset by initialize and by EXC 77. In addition, WXMX is reset by the end-of-block echo when block I-O transfers are used.

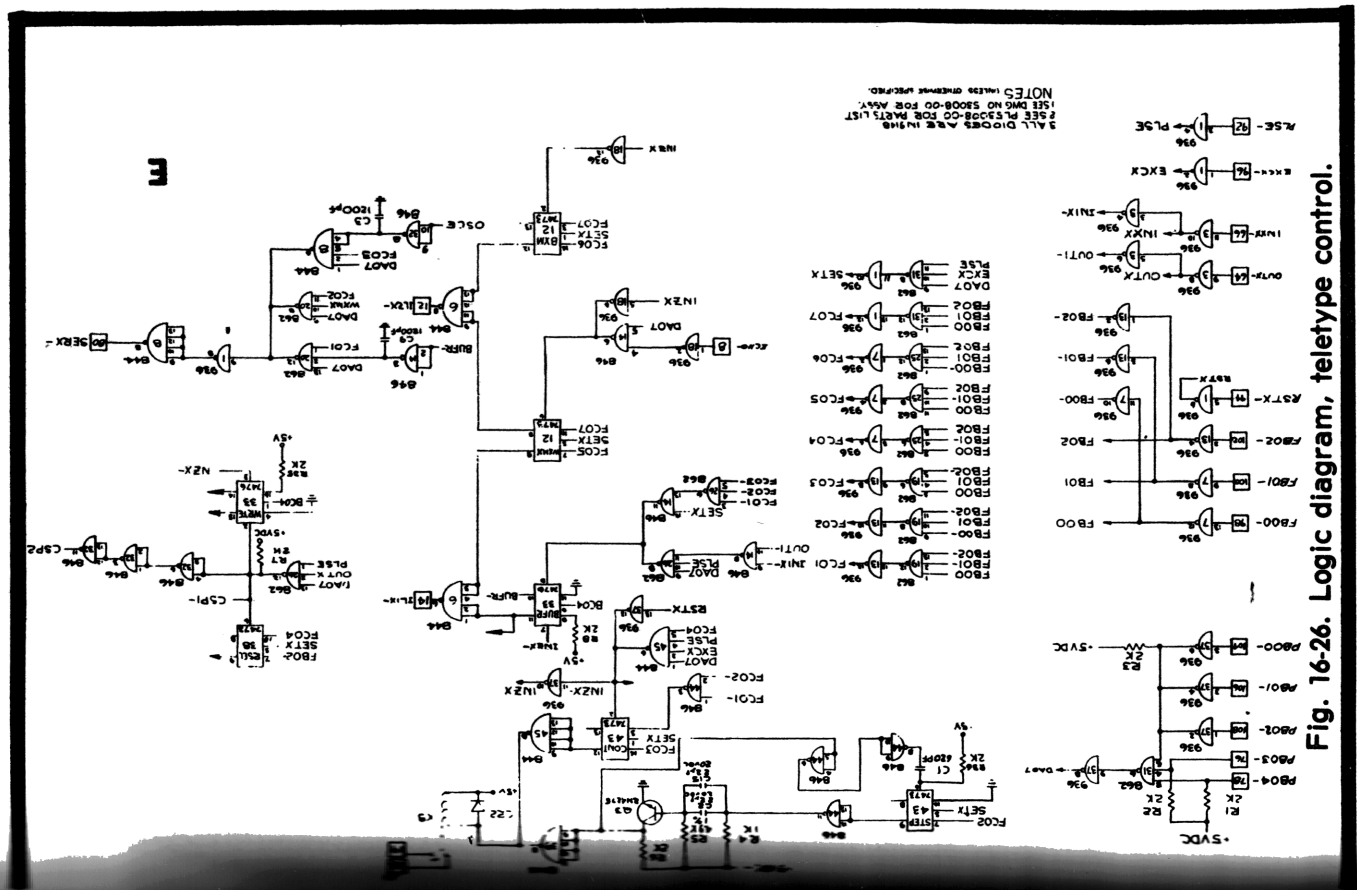
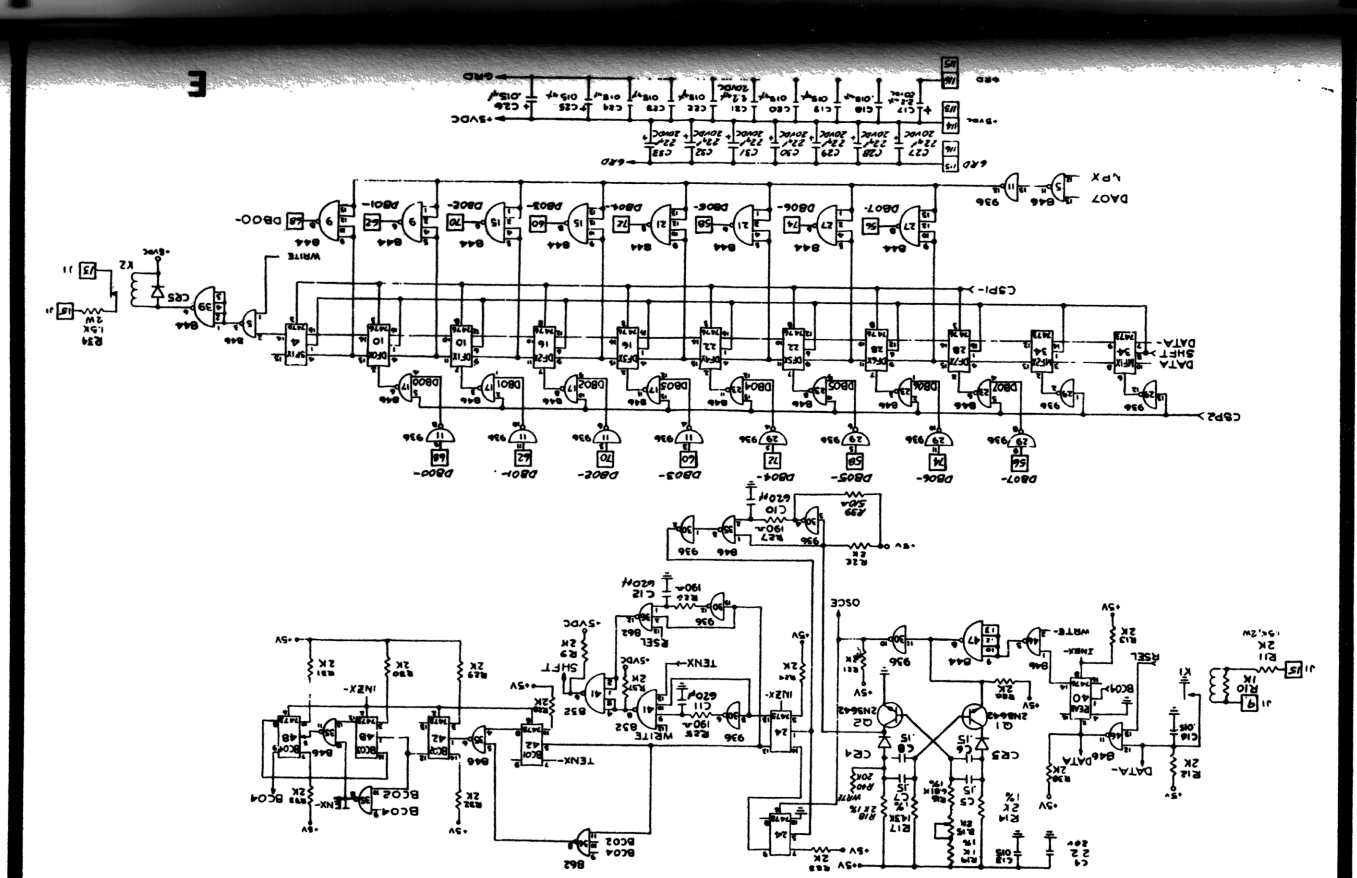


Fig. 16-26. Logic diagram, teletype control.



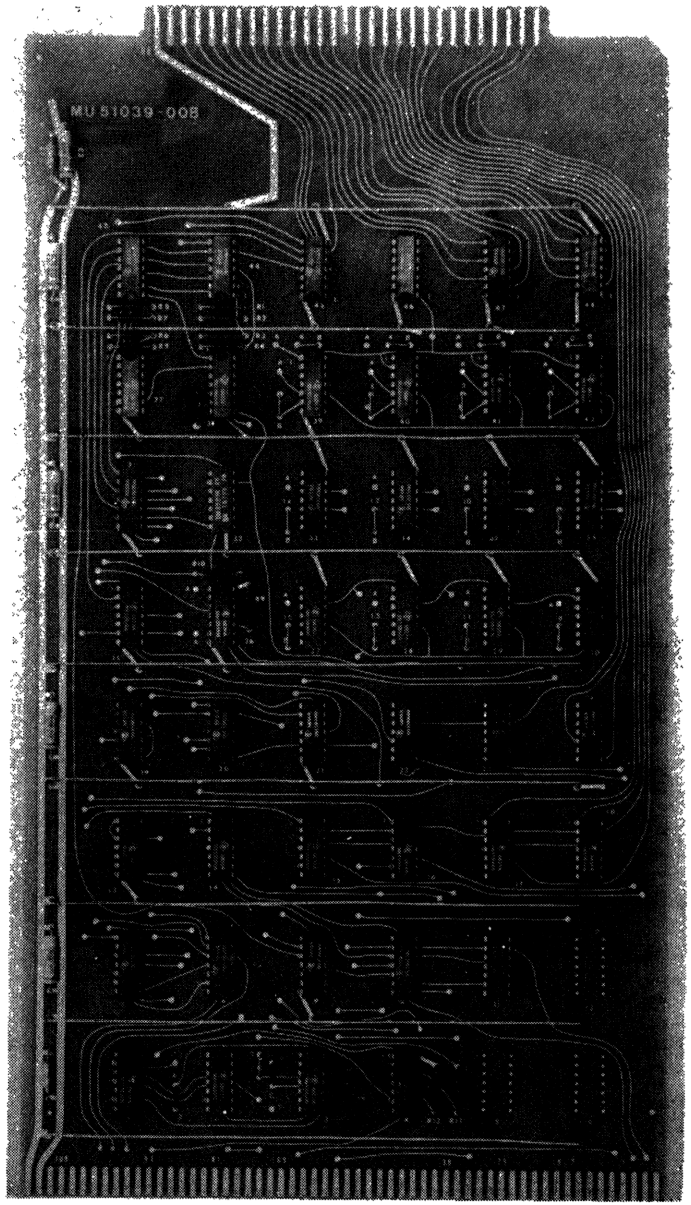


Fig. 16-27. Digital buffered input module 1.

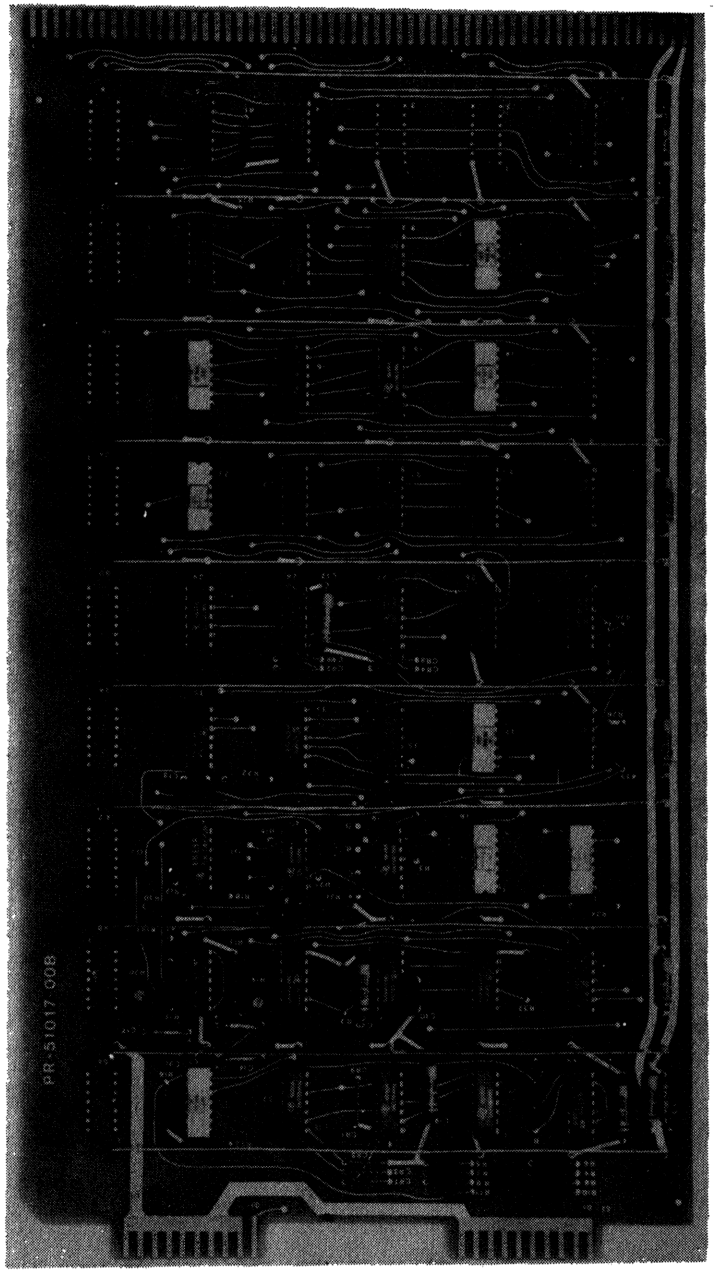


Fig. 16-28. Paper tape reader—paper tape punch module.

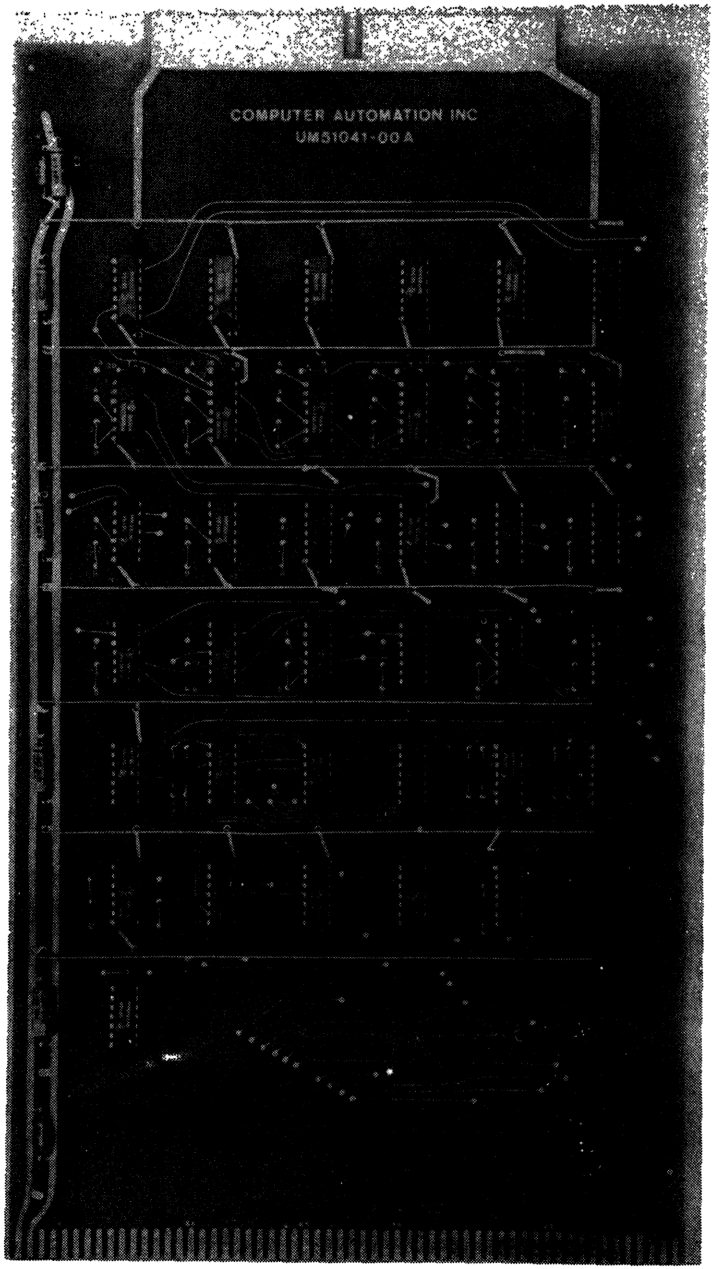


Fig. 16-29. Digital buffered input module 3.

H. I-O bus interface logic. The I-O interface circuits consist of line receives for data, device address, function code and control lines; and drivers for data and sense response.

J. TTY relay interface: Three reed relays are used to connect the interface logic to the TTY. K1 is in the TTY transmit loop. The relay contacts are closed while the loop has the 20 ma DC current flowing. K2 is in the TTY receive loop. It is kept closed in the idle (marking) state to maintain the 20 ma current in the loop. The open circuit voltage in these two loops is approximately 50v DC obtained from a power supply in the TTY. K3 completes a 120v AC circuit to the reader trip solenoid. It is normally open and is closed briefly for step read or closed continuously for continuous read.

Oscillator adjustment procedure

A. Set up the scope with internal sync, and observe BC04. (Connect scope to logic ground and not to chassis ground. Use an AC line adapter to plug into a 3-pronged 115v AC outlet.)

B. Temporarily ground OSC enable (OSCE-) at pac 47 pin 8. This causes the oscillator to run continuously.

C. Adjust pot R15 until the BC04 repetition rate is 99 ms.

D. Remove the ground from 47 pin 8.

E. Put the TTY in continuous read by executing EXC 73 command with a tape loop in the reader.

F. Observe OSCE- on the scope. Sync should be on the positive-going edge. A pulse of approximately 1 to 2 ms should occur every 100 ms.

G. Adjust R15 until the observed pulse is 500 microseconds, within 100 microseconds.

H. Seal the pot and remove the scope.

The high-speed reader and punch specifications are described in Chapter 14. Figs. 16-27 through 16-29 show several of the modules used in the interface.

SPECIAL CIRCUITS

Overflow Logic

The overflow flip-flop (OV) is used as an indication of an illegal arithmetic operation and as a link flip-flop for rotate shifts. For arithmetic operations, OV is set if two positive numbers result in a negative result or if two negative numbers result in a positive result. For instance, incrementing the number 01111111 will cause it to go negative: 10000000. This will cause OV to set.

For shift operations, OV is used to link the two ends of the A register on the rotate instructions. Rotate left causes AR07 to shift into OV and OV to shift into AROO. Nine shifts cause the OV and A register to return to the original condition. Rotate right is similar except the direction is reversed. OV can be set and reset under program control for use as a flag or indicator in a program. It is also reset by the reset switch on the console.

Interrupt Logic

The 808 and 208 interrupt logic consists of several major sections:

Three interrupt input lines: IL1X, IL2X and IURX.

Interrupt acknowledge flip-flop, IUAX, and associated gating.

Priority generator, the output of which is the priority out signal, PROT.

Interrupt address generator for IL1X and IL2X which generates the address associated with these two interrupt lines.

Interrupt mask flip-flop, IMXX, and associated gating. IMXX when set enables the 808 and 208 to respond to interrupts. Otherwise, interrupt requests are ignored.

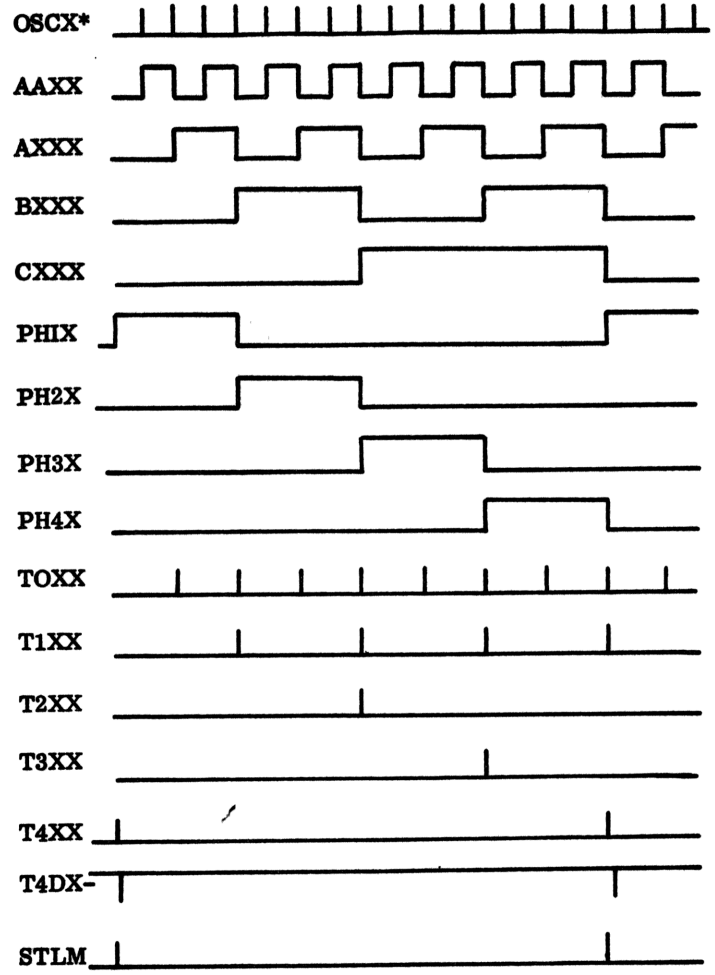
Interrupt address request generator, IARX, which requests the interrupt address to be accessed for the interrupt instruction.

Interrupt clock generator, IOCL, which provides a clock signal to external interrupt logic.

The interrupt logic provides the capability of responding automatically to external stimuli, making the computer a powerful and flexible component in real time control applications. The operation of a typical interrupt sequence is as follows:

A. The interrupt system is assumed to be enabled; i.e., IMXX is true. A request for interrupt is generated by external logic driving one of the interrupt lines in the I-O bus to ground.

B. At the end of the current instruction, IUAX is set (if a skip is not to be accomplished, and the instruction is not a jump and save). Notice that the end of the current instruction is T3 of the last cycle in the instruction.



* One oscillator pulse should be 40-60 ns wide at the 1 volt point.

Fig. 16-30. Processor timing waveform.

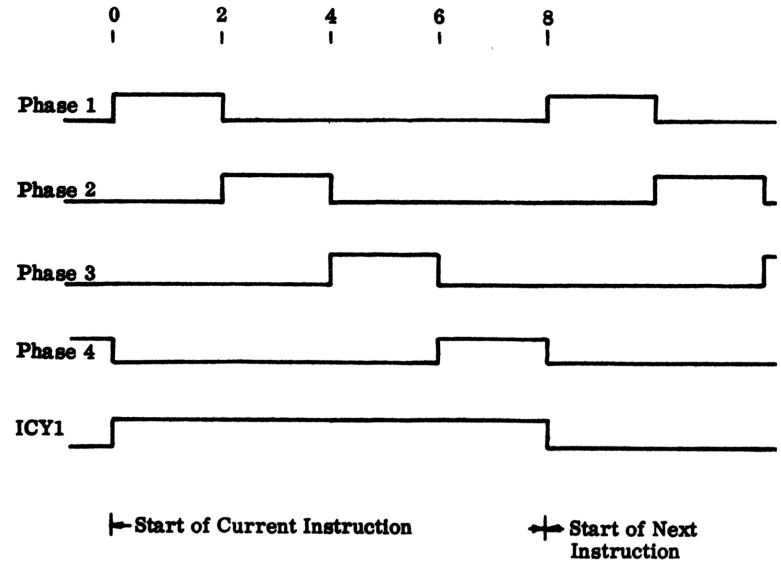


Fig. 16-31. Typical single-word instruction timing diagram.

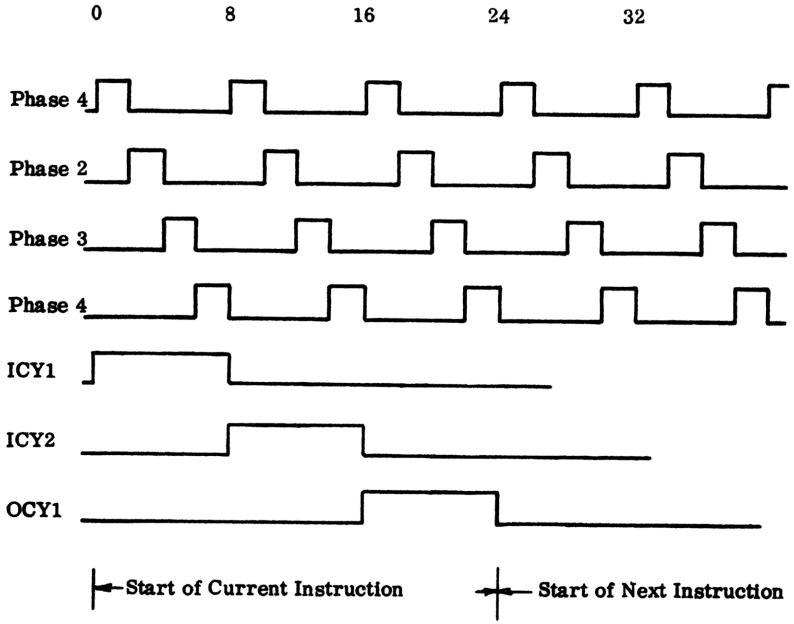


Fig. 16-32. Typical double-word instruction timing diagram.

C. Normally, PH4 of the last cycle of an instruction is used to generate the address to accept the next instruction; i.e., L → M and P → ME or 0 → ME. When IUAX comes on at the beginning of PH4, the interrupt address is sent to M and zero is sent to ME. This is accomplished by gating the interrupt address onto the X bus if IL1 or IL2 is requesting an interrupt. If IURX is the line causing the interrupt, then the interrupt address is obtained from the data bus and gated to the M register during PH4.

D. The instruction located at the interrupt address is established. If it is a double-word instruction, M is incremented to accept the second word. Notice that L and P are not used to establish the instruction nor are L and P incremented during ICY1 and ICY2 when IUAX is on.

E. The interrupt instruction is executed. It may be a single-word instruction or may be a double-word instruction four or five cycles long. In any case, IUAX remains on during the entire instruction and is not turned off until the end of the instruction, and at T3.

F. The next instruction in the program as specified by L and P is executed. This means that even with a continuous interrupt request, every other instruction executed will be from the program. This prevents interrupts from "hanging up" the processor.

If the interrupt instruction is a jump and save, the mask flip-flop IMXX is automatically reset. This allows the programmer to enter a subroutine and disable the device, creating the interrupt before allowing more interrupts to be acknowledged. If this were not done, the sequence would be jump and save, next instruction in interrupt subroutine, and another jump and save. This causes the save locations to point back to the interrupt subroutine instead of the main program and the program hangs up in the interrupt subroutine.

Priority among interrupts is established by the PROT line. IL1X has the highest priority and IL2X is next. If neither IL1X or IL2X are requesting, then priority is passed out to the "down stream" devices, allowing them to request an interrupt. By serially wiring PROT to the down stream devices, relative priorities may be established.

TIMING

The processor timing essentially consists of a crystal oscillator, a timing chain, and decoding logic. The crystal oscillator inputs timing pulses to the timing chain which is tapped at several points to generate various square waves. These are used to generate the four phases from which the timing pulses occur. Refer to Fig. 16-30 for the basic processor timing wave form. The oscillator schematic is in Fig. 16-5. OSCX is a 2-MHz pulse train derived from the basic oscillator. The pulse is actually 450 nanoseconds wide and taken from the timing chain. Also generated on the timing are:

AAXX, a 500 nanosecond square wave

AXXX, a 1 microsecond square wave

BXXX, a 2 microsecond square wave

CXXX, a 4 microsecond square wave

Timing resulting from the above square waves is under control of RUNX, meaning when the signal is true the computer is in the run mode; i.e., not stopped.

One processor cycle is 8 microseconds in duration. Each processor cycle is divided into four phases which are a result of square waves generated on the timing chain. The four phases are each 2 microseconds in duration. A single-word instruction takes 8 microseconds; i.e., four phases.

Phase 1 is used to increment the program sequencer, which consists of the L and P registers. During phase 2 different events occur, depending on the instruction being executed. Each instruction is described in detail by events that occur in each phase. In general, phase 3 is an execution cycle. Any variation in the instructions are described. Phase 4 is generally used as the address phase during which the address of the operand or the next instruction is transferred to M and ME.

Timing pulses, designated TXXX, are generated either continuously (TOXX, T1XX) or at the end of phase 2, 3, and 4. The timing pulses allow events within the computer to occur. TOXX is a 45 nanosecond wide timing pulse that occurs every microsecond in the run mode. T1XX is a 45 nanosecond wide timing pulse that occurs every 2 microseconds in the run mode. T2XX is a timing pulse that occurs at the end of phase 2. T3XX is a timing pulse that occurs at the end of phase 3. T4XX is a timing pulse that occurs at the end of phase 4. T4DX is a

timing pulse delayed approximately 25 nanoseconds after T4XX. Single-word instructions require only one word in memory and the cycle sequence consists only of ICY1. Fig. 16-31 shows the timing diagram of a typical single-word instruction.

Double-word instructions such as I-O and block transfer instructions use two words in memory and require two memory cycles to assemble the complete instruction. A typical timing diagram of a double word address is shown in Fig. 16-32. ICY1 is true when the first word of the instruction is being read from memory to be placed in the I register. Instruction cycle 2 (ICY2) is true when the second word of the double-word instruction is being read from memory to be placed in the R register. Operand type instructions require an additional memory cycle, during which Operand cycle 1 (OCY1) is true. This is true when an operand is being read from or stored into memory, or when the contents of the L register are being stored on a jump and save instruction. Operand cycle 2 (OCY2) is true when the contents of the P register is being stored in memory during a jump and save instruction. Figs. 16-33 through 16-45 show various timing signals observed with a Hewlett-Packard Model 180 oscilloscope. The oscillator or clock waveform appears at the top in each case.

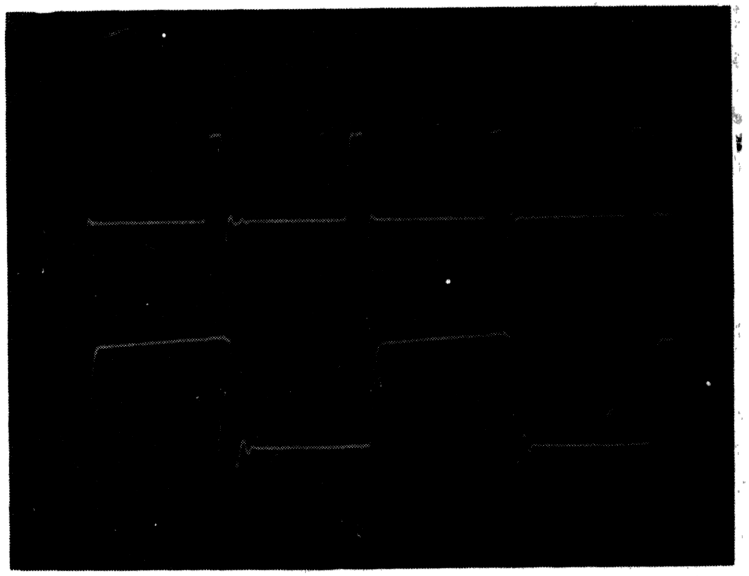


Fig. 16-33. AAXX waveform (bottom) and OSCX (top). 0.2 microseconds per div. and 2v.

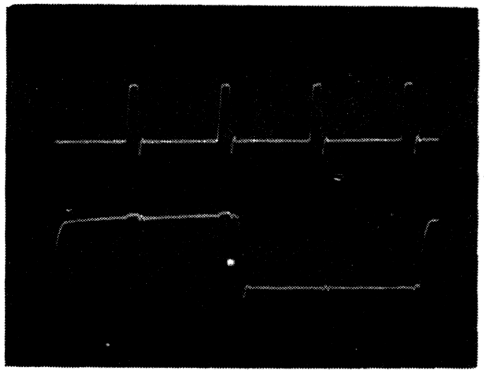


Fig. 16-34. AXXX waveform (bottom) and OSCX. 0.2 microseconds per div. and 2v.

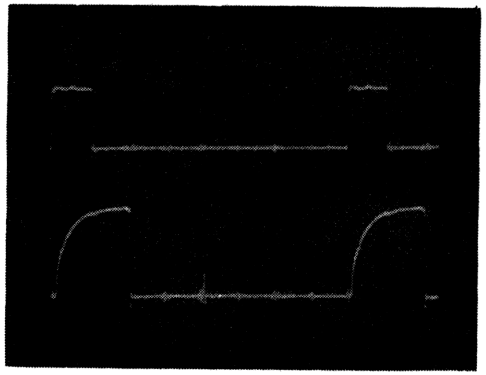


Fig. 16-35. PH1X waveform (bottom) and MSCX. 1 microsecond per div. and 2v.

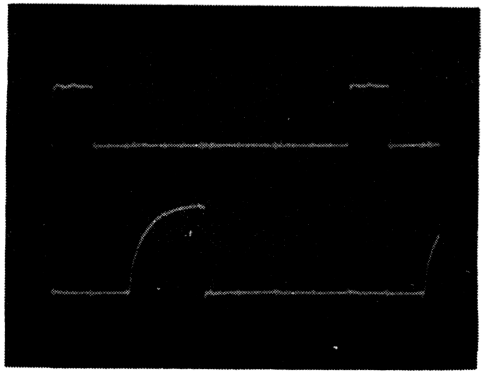


Fig. 16-36. PH2X waveform (bottom) and MSCX. 1 microsecond per div. and 2v.

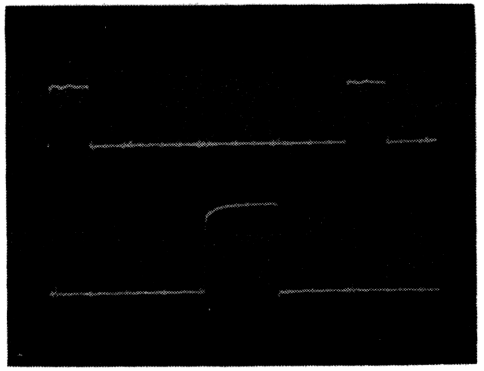


Fig. 16-37. PH3X waveform (bottom) and MSCX. 1 microsecond per div. and 2v.

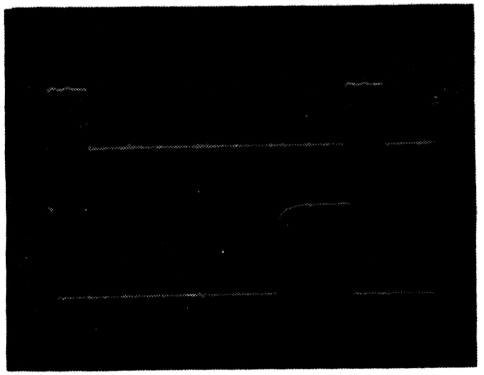


Fig. 16-38. PH4X waveform (bottom) and MSCX. 1 microsecond per div. and 2v.

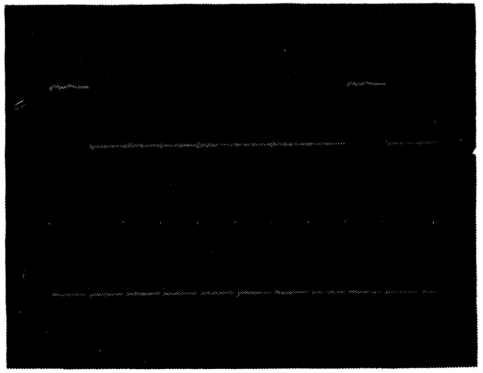


Fig. 16-39. T0XX waveform (bottom) and MSCX. 1 microsecond per div. and 2v.

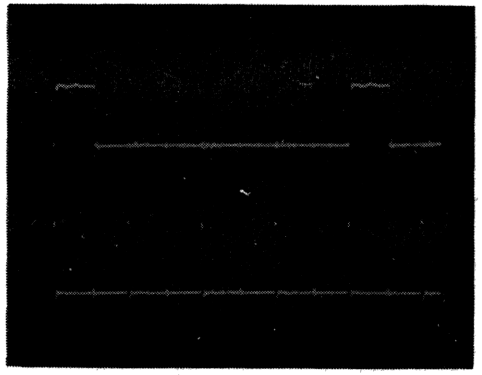


Fig. 16-40. T1XX waveform (bottom) and MSCX. 1 microsecond per div. and 2v.

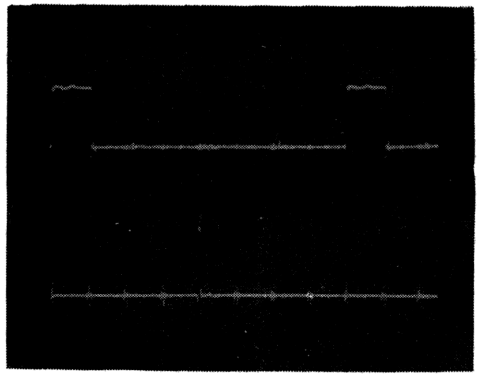


Fig. 16-41. T2XX waveform (bottom) and MSCX. 1 microsecond per div. and 2v.

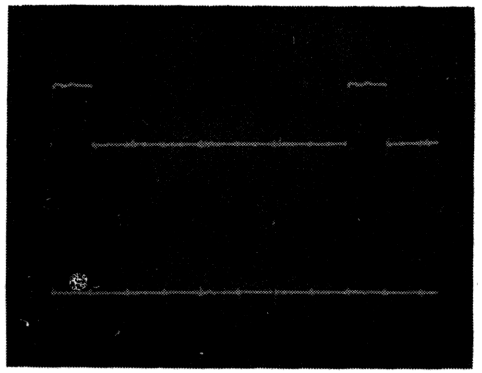


Fig. 16-42. T3XX waveform (bottom) and MSCX. 1 microsecond per div. and 2v.

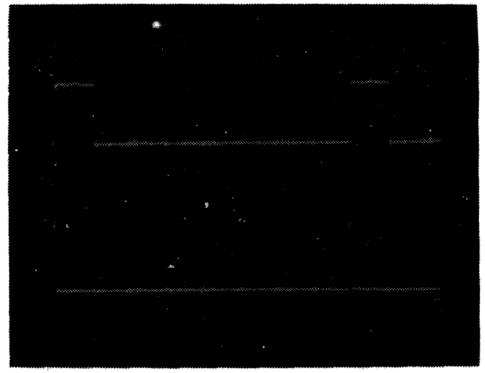


Fig. 16-43. T4XX waveform (bottom) and MSCX.

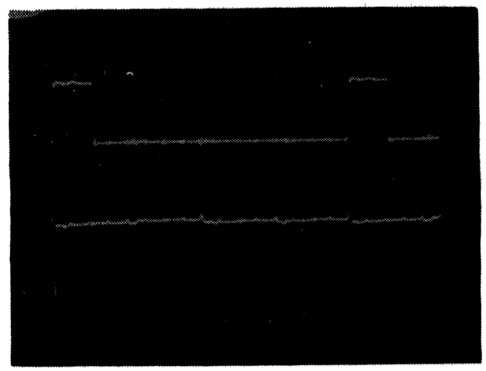


Fig. 16-44. T4DX- waveform (bottom) and MSCX. 1 microsecond per div. and 2v.

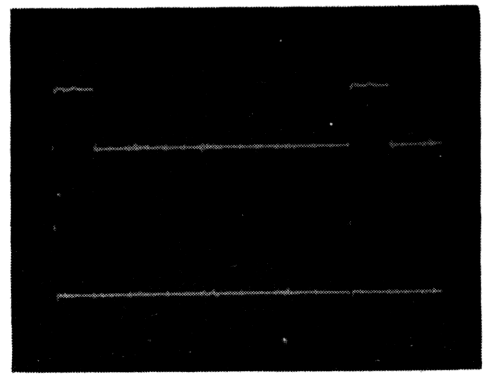


Fig. 16-45. STLM waveform (bottom) and MSCX. 1 microsecond per div. and 2v.

Chapter 17

Processor Operation

It is the purpose of this section to describe all the processor instructions in detail. There is a short introduction for each class, and in some cases, for each group of instructions within a class that differ in some respects. In addition, a flow-chart is included for each class, and in some instances, for each instruction where it is deemed necessary.

REGISTER CHANGE

The register change class contains the control, shift and register change instructions. All the instructions in this group are single-word.

CONTROL

NOP: No Operation

The execution of this instruction causes an 8-microsecond pause in the program currently being executed. See Figs. 17-1 and 17-2. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of Phase 1. SELL, SELR, SLRN, CINP, SELS, SET1, CLRW, SETW occur. If L overflowed, the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: Approximately halfway through phase 2, memory data is set into the W register. Contents of W is transferred to the I register at the end of the phase. SETI occurs.

Phase 3: No action.

Phase 4: Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SELL, CINP, SELS, STLM, XPMX, and STUM occur.

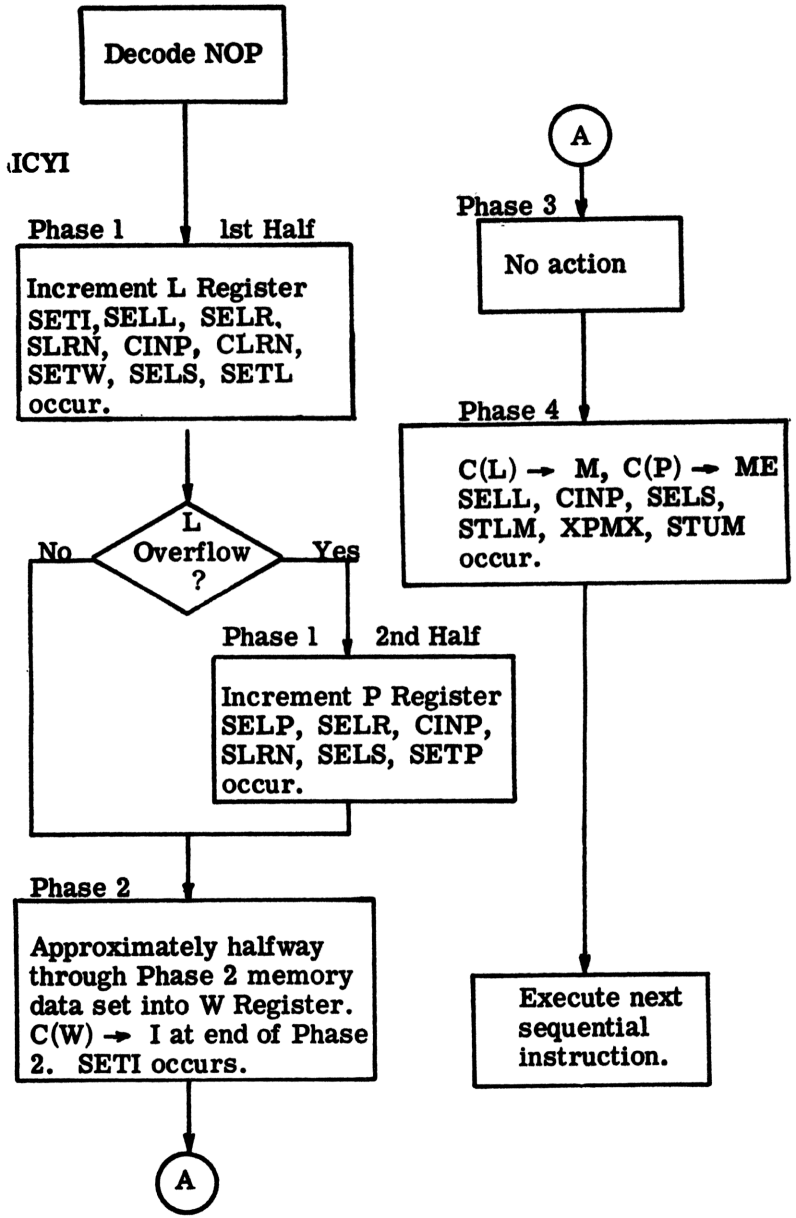


Fig. 17-1. NOP flow chart.

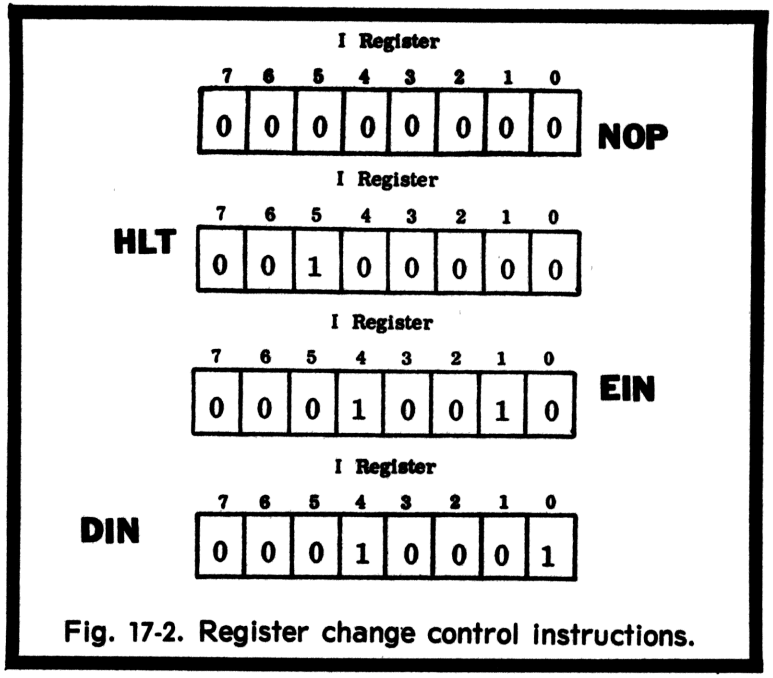


Fig. 17-2. Register change control instructions.

HLT: Halt

This instruction halts the computer. See Fig. 17-8. The timing is 1.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: Processor halts at the end of phase 3 and does not enter phase 4. RUNX flip-flop turned off.
Phase 4: Same as NOP, phase 4.

EIN: Enable Interrupts

This instruction enables the computer to respond to a request for interrupt. The interrupt mask is affected. The timing is 1.

Cycle Sequence:

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.

Phase 3: Set IMXX.
Phase 4: Same as NOP, phase 4.

DIN: Disable Interrupts

The instruction inhibits the computer from accepting a request for interrupt. The interrupt mask is affected. The timing is 1.

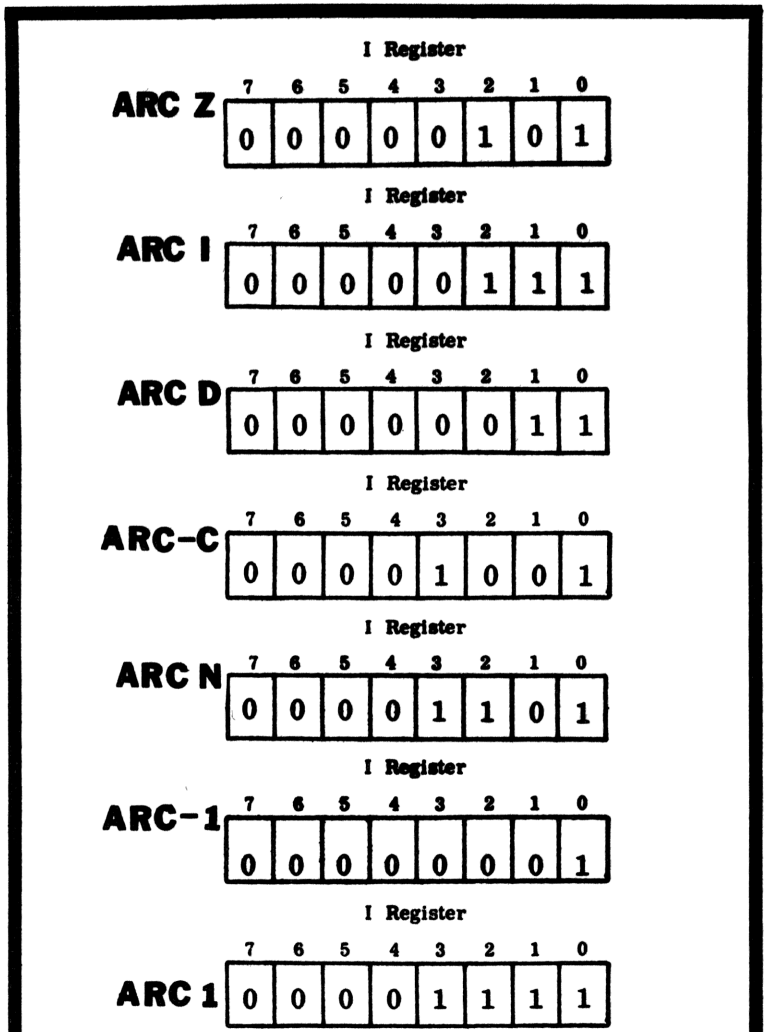


Fig. 17-3. A register change instructions.

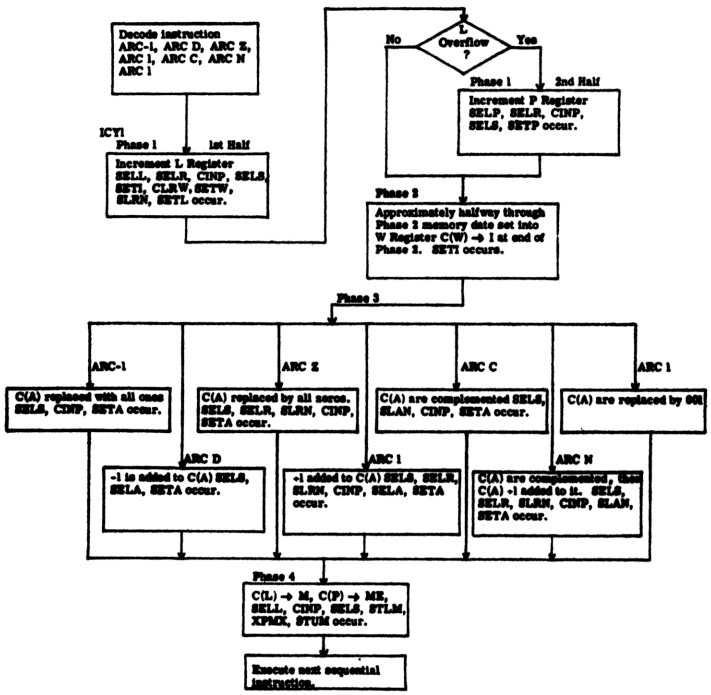


Fig. 17-4. A register change flow chart.

Cycle Sequence:

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3:Reset IMXX.
Phase 4: Same as NOP, phase 4.

A Register Change

ARC Z: Clear A Register

ARC causes the C(A) to be replaced by zeros, affecting the A register. See Figs. 17-3 and 17-4. The timing is 1.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.

Phase 3: 000 entered into the A register. SELS, SELR, SLRN, CINP, and SETA occur.
Phase 4: Same as NOP, phase 4.

ARC I Increment A Register

The ARC I adds one (least significant bit) to the contents of A. OV is set if the contents goes from 0,1111111 to 1,0000000. The timing is 1.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: +1 is added to the A register. SELS, SELR, SLRN, CINP, SELA and SETA occur.
Phase 4: Same as NOP, phase 4.

ARC D Decrement A Register

ARC D subtracts one (least significant bit) from the contents of A. OV is set if the contents goes from 1,0000000 to 0,1111111. The timing is 1.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: -1 is added to the A register. SELS, SELA and SETA occur.
Phase 4: Same as NOP, phase 4.

ARC C 1s Complement A Register

ARC C performs the 1s complement (bit-by-bit inversion) of the contents of the A register. The timing is 1.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: Contents of the A register is complemented. SELS, SLAN, CINP and SETA occur.
Phase 4: Same as NOP, phase 4.

ARC N 2s Complement A Register

ARC N performs the 2s complement of (negative) the contents of the A register. The timing is 1.

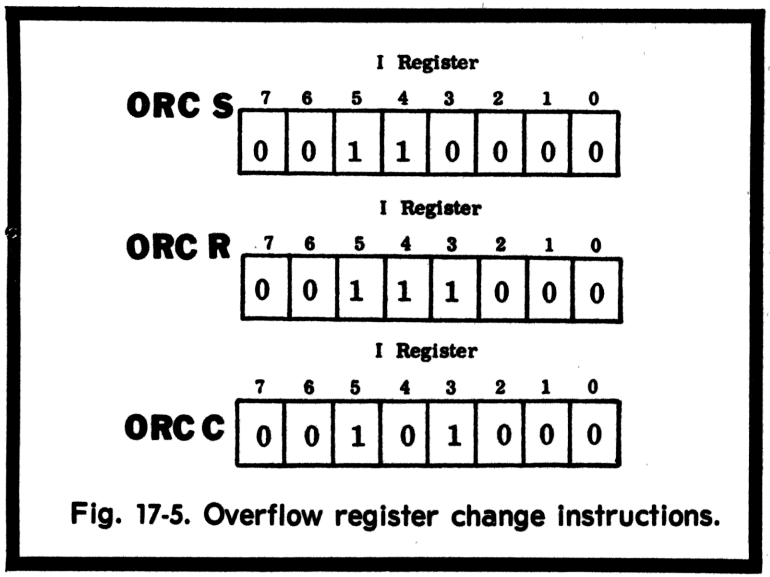


Fig. 17-5. Overflow register change instructions.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: Contents of A register is complemented and A + 1 added to it. SELS, SELR, SLRN, CINP, SLAN and SETA occur.
Phase 4: Same as NOP, phase 4.

ARC-1 Transfer -1 to A

ARC-1 replaces the contents of the A register with 1,1111111. The timing is 1.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: 377 set into the A register. SELS, CINP, and SETA occur.
Phase 4: Same as NOP, phase 4.

ARC 1 Transfer +1 to A

ARC 1 replaces the contents of the A register with 0,0000001. The timing is 1.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: 001 is entered into the A register. SLAN, SELA, SLRN, SELR, CINP, SELS, and SETA occur.
Phase 4: Same as NOP, phase 4.

Overflow Register Change (ORC)

ORC S Set Overflow

ORC S sets the overflow flip-flop. If already set there is no change. The timing is 1. See Figs. 17-5 and 17-6.

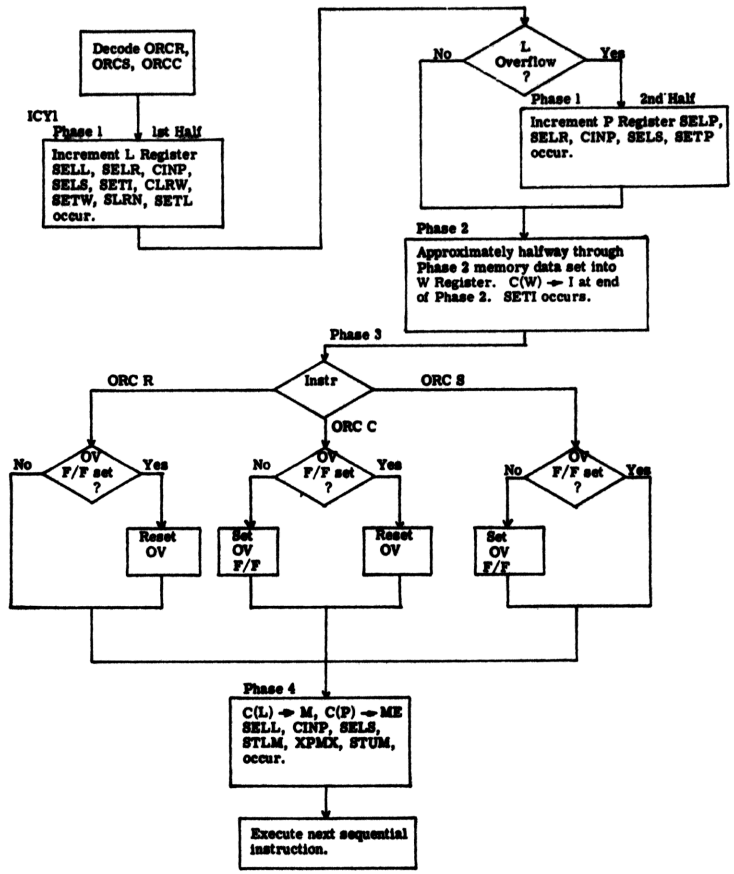


Fig. 17-6. Overflow register change flow chart.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: Overflow flip-flop set at T3XX, the time OVSE occurs.
Phase 4: Same as NOP, phase 4.

ORC R Reset Overflow

ORC R resets the overflow flip-flop. The timing is 1.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: Overflow flip-flop reset. OVRE occurs.
Phase 4: Same as NOP, phase 4.

ORC C Complement Overflow

ORC C complements the content of the overflow flip-flop. The timing is 1.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: Complements the overflow flip-flop at T3XX.
Phase 4: Same as NOP, phase 4.

LOGICAL SHIFT

The logical shift instruction causes the contents of the A register to be shifted one place to the left or right. Zeros are shifted into one end of the register and the contents is shifted out and lost. Thus, if eight consecutive logical shift left instructions were executed, the A register would contain all zeros. Note: The overflow flip-flop is reset on a logical shift instruction at the beginning of the operation and may be set at the end of the instruction.

SHF R Logical Shift Right

SHF R resets OV, shifts the contents of the A register right one place. Zero is shifted into the sign bit and the LSB of A shifted into OV. The timing is 1. See Figs. 17-7 and 17-8.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: Contents of the A register is shifted right one position. AROO shifts into the overflow flip-flop. Zero shifts into ARO7. SELA, CINP, SETA, LSHR, SHRX occur.
Phase 4: Same as NOP, phase 4.

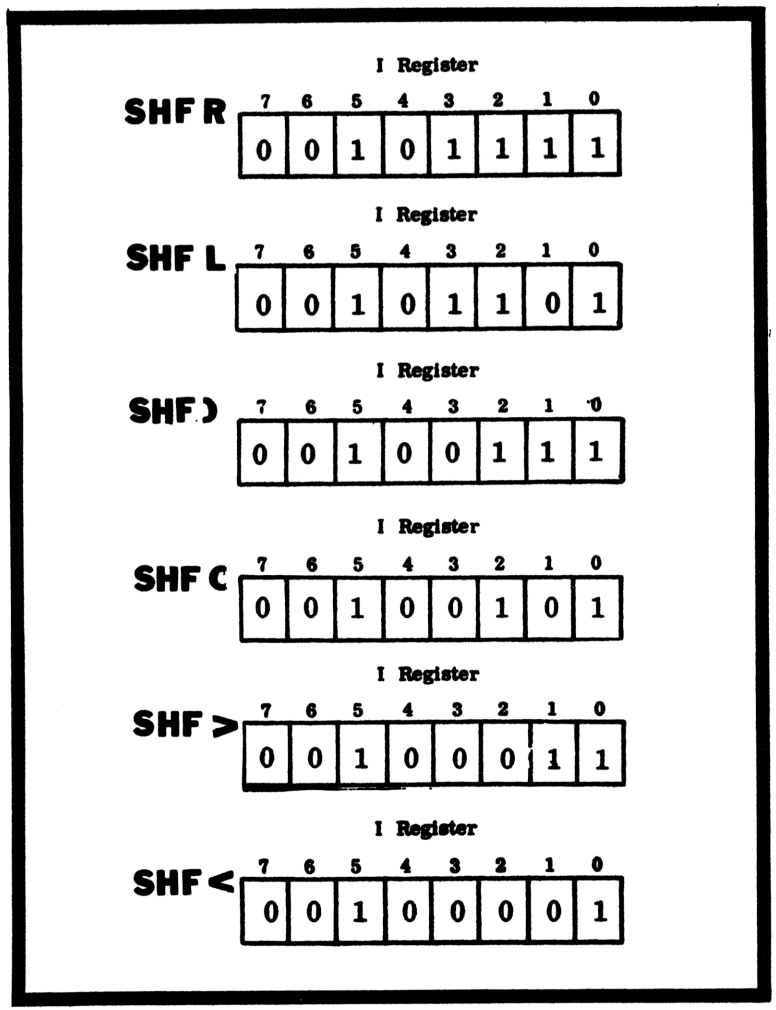


Fig. 17-7. Logical shift, rotate, and arithmetic shift instructions.

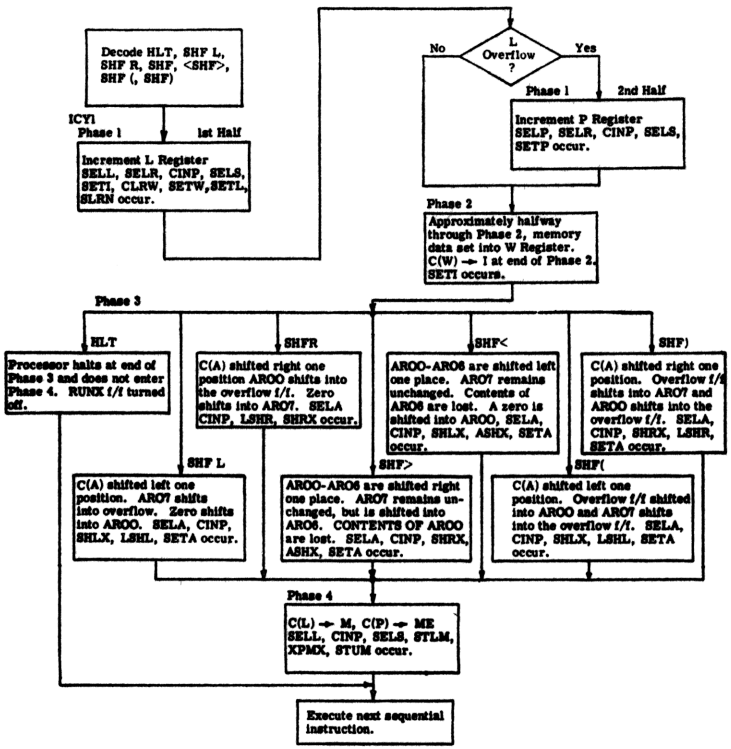


Fig. 17-8. Logical shift, rotate and arithmetic shift flow chart.

SHF L Logical Shift Left

SHF L resets OV, shifts the contents of the A register left one place. Zero is shifted into the LSB of A and the sign bit of A is shifted into OV. The timing is 1.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: Contents of the A register is shifted left one position. AR07 shifts into overflow. Zero shifts into AR00. SELA, CINP, SHLX, LSHL, SETA occur.
Phase 4: Same as NOP, phase 4.

ROTATE

The rotate instruction, sometimes referred to as logical rotate, preserves the contents of the A register as it is shifted. For a rotate right (left), the content of the overflow flip-flop is shifted into bit 7 (bit 0) of the A register, and bit 0 (bit 7) of the A register is shifted into OV. This essentially creates a 9-bit register. Thus, nine consecutive rotate right (left) instructions leave the A register and the overflow flip-flop is unchanged. Shifting A through the overflow flip-flop facilitates testing bits of A. For instance, rotate right and then skip on overflow reset tests bit 0 of the A register. Rotate left restores A to its original state.

SHF) Rotate Right

SHF) shifts the contents of the A register right one place through OV. Previous contents of OV is shifted into A7. The timing is 1.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: Contents of the A register is shifted right one position. AROO shifts into the overflow flip-flop. The previous contents of the overflow flip-flop is shifted into AR07. SELA, CINP, SHRX, LSHR, SETA occur.
Phase 4: Same as NOP, phase 4.

SHF (Rotate Left

SHF (shifts the contents of the A register left one place through OV. Sign bit is shifted into OV. Previous contents of OV is shifted into the LSB of A. The timing is 1.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: Contents of the A register is shifted left one position. AR07 shifts into the overflow flip-flop and the previous contents of the overflow flip-flop is shifted into AROO. SELA, CINP, SHLX, LSHL, SETA occur.
Phase 4: Same as NOP, phase 4.

ARITHMETIC SHIFT

The arithmetic shift differs from the logical or rotate shifts in that the A register is assumed to contain a signed number. The sign bit is A7. Since shifting left multiples by 2 and shifting right divides by 2, the sign of a number must not change. Thus an arithmetic shift can affect only the magnitude position of the number and cannot change the sign bit. The PDC-808 uses 2s complement arithmetic. The arithmetic shift right causes the sign bit to be shifted while remaining unchanged. An arithmetic shift left performs a logical shift left, except the sign bit is unchanged.

SHF > Arithmetic Shift Right

SHF > shifts the lower seven bits of the A register right one place. Sign bit remains unchanged and is also shifted into A6. The LSB of A is lost. The timing is 1.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: The lower seven bits of the A register are shifted right one place. ARO7, the sign bit, remains unchanged but is shifted into ARO6. The contents of bit position AROO is lost, since the contents of bit position O7 replaced it. SELA, CINP, SHRX, ASHX, SETA occur.
Phase 4: Same as NOP, phase 4.

SHF < Arithmetic Shift Left

SHF < shifts the lower seven bits of the A register left one place. Sign bit remains unchanged. A6 is lost and zero is shifted into the LSB of A. The timing is 1.

Cycle Sequence: ICY1

Phase 1: Same as NOP, phase 1.
Phase 2: Same as NOP, phase 2.
Phase 3: The lower seven bits of the A register are shifted left one place. ARO7 the sign bit, remains unchanged; position ARO6 is replaced by the contents of bit position ARO5. Bits ARO4-ARO2 are treated in a similar manner. A zero is shifted into AROO. SELA, CINP, SHLX, ASHX, SETA occur.

Phase 4: Same as NOP, phase 4.

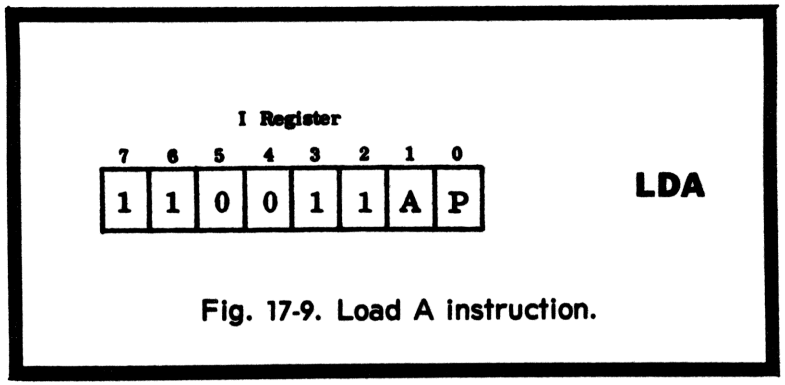


Fig. 17-9. Load A instruction.

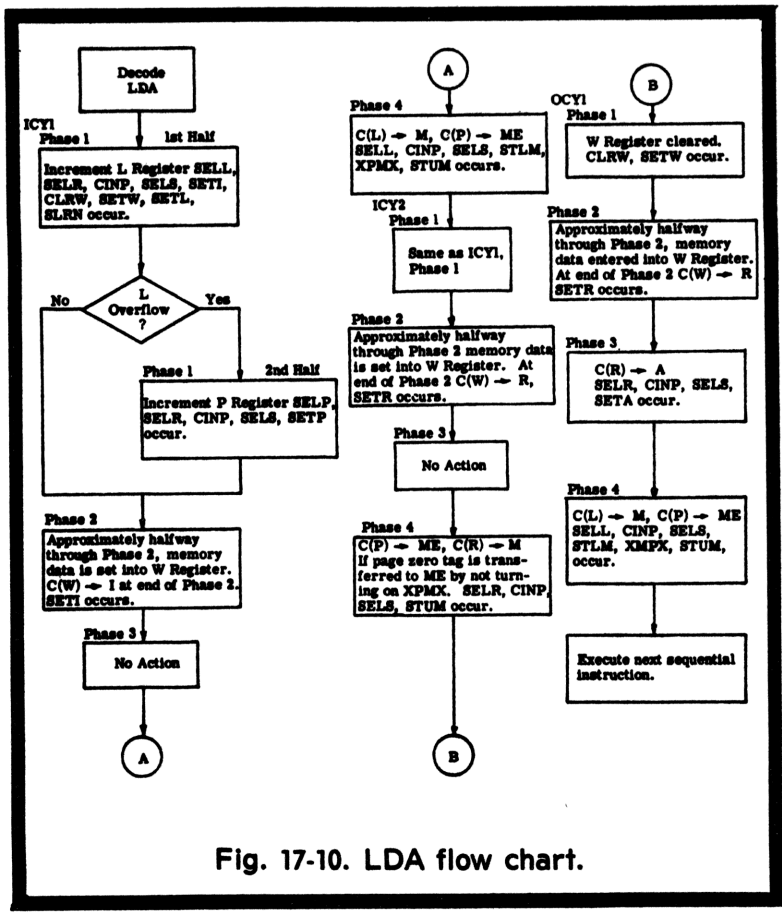


Fig. 17-10. LDA flow chart.

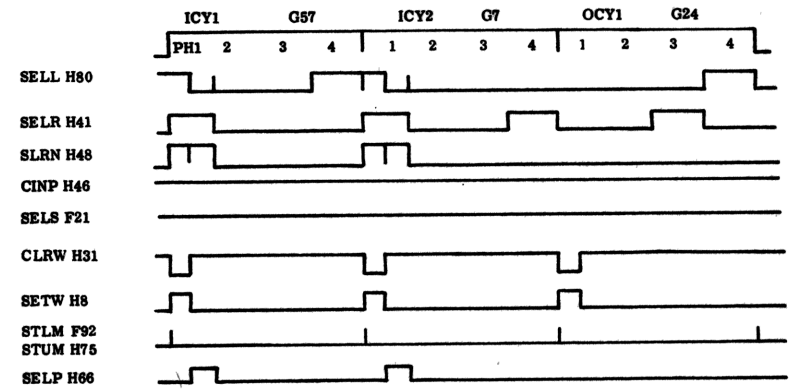


Fig. 17-11. LDA timing diagram.

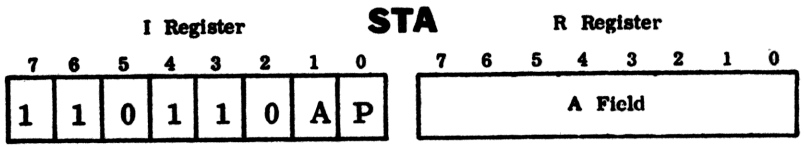


Fig. 17-12. Store A instruction.

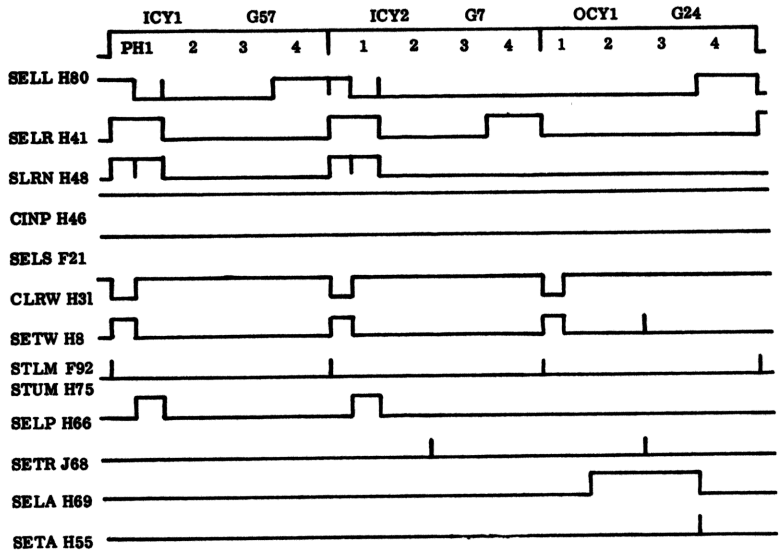


Fig. 17-13. STA timing diagram.

MEMORY REFERENCE

LDA LOAD A

LDA loads the contents of the effective memory location into the A register. The timing is 3 plus 2 for each indirect level. See Figs. 17-9, 17-10 and 17-11.

Cycle Sequence: ICY1, ICY2, OCY1

ICY1 cycle: Same as STA ICY1 cycle.
ICY2 cycle: Same as STA ICY2 cycle.
OCY1 cycle: Phase 1: Same as phase 1, OCY1 cycle, STA instruction. Phase 2: Approximately halfway through phase 2, memory data is entered into the W register. At the end of phase 2 the contents of the W register is transferred into the R register. SETR occurs. Phase 3: Contents of the R register is transferred to the A register. SELR, CINP, SELS and SETA occur. Phase 4: Same as NOP, phase 4.

STA Store A

STA stores the contents of the A register in the effective memory location. The timing is 3 plus 2 for each indirect level. See Figs. 17-11, 17-12, 17-13.

Cycle Sequence: ICY1, ICY2 OCY1

ICY1 cycle: Same as NOP.
ICY2 cycle: Phase 1: Same as NOP, phase 1. Phase 2: Approximately halfway through phase 2 the memory data is entered into the W register. At the end of phase 2 the contents of W is transferred to the R register. SETR occurs. Phase 3: No action. Phase 4: Contents of the P register is transferred to the M extension register and the contents of the R register is transferred to the M register. If page zero tag is used zero is transferred to the M extension register; by not turning on XPMX, SELR, CINP, SELS, STLM, and STUM occur.

OCY1 cycle: Phase 1: W register cleared. CLRW and SETW occur. Phase 2: Approximately halfway through phase 2 the memory data is entered into the W register. At the end of phase 2 the contents of the A register is transferred into the W register. SELA, CINP, SELS and SETW occur. Phase 3: No action. Phase 4: Same as NOP, phase 4.

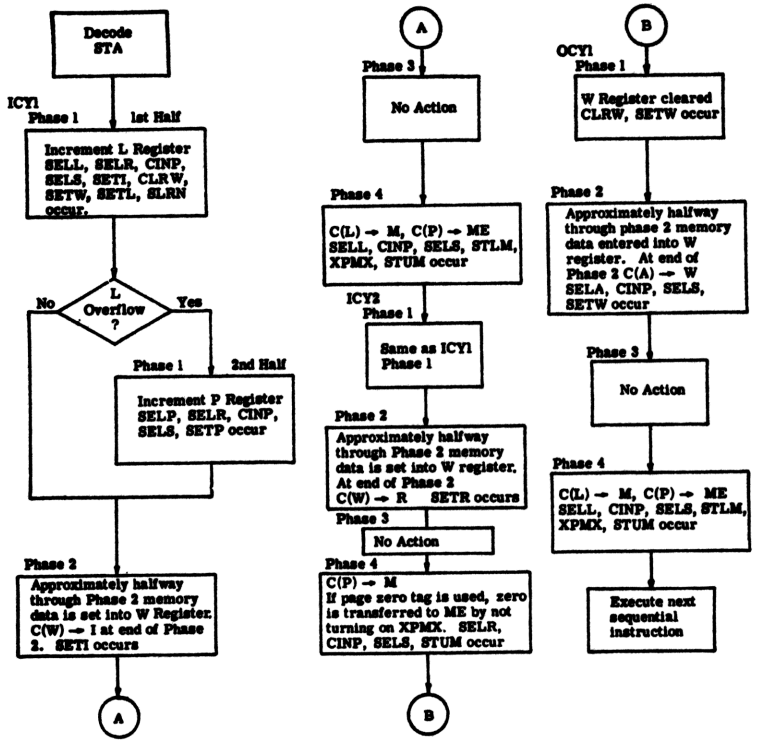


Fig. 17-14. STA flow chart.

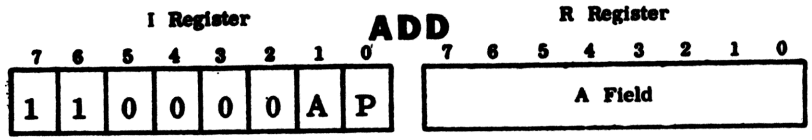


Fig. 17-15. ADD instruction.

ADD

ADD adds the contents of the effective memory location to the contents of the A register. The timing is 3 plus 2 each indirect level. See Fig. 17-15.

Cycle Sequence: ICY1, ICY2, OCY1

ICY1 cycle: Same as STA, 331.
ICY2 cycle: Same as STA, 331.
OCY1 cycle: Phase 1: Same as phase 1, OCY1 cycle STA instruction. Phase 2: Same as phase 2, OCY1 cycle, LDA in-

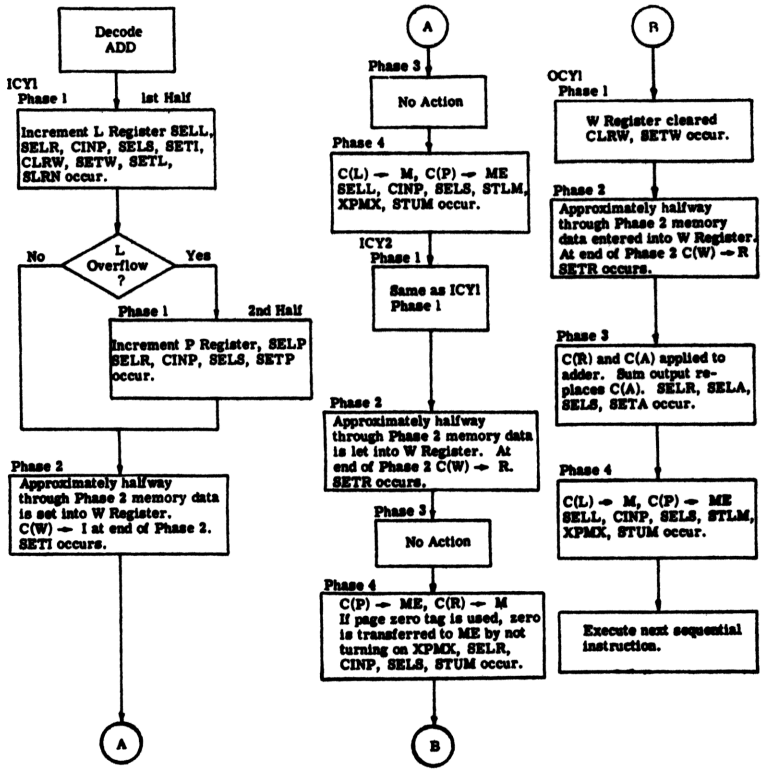


Fig. 17-16. ADD flow chart.

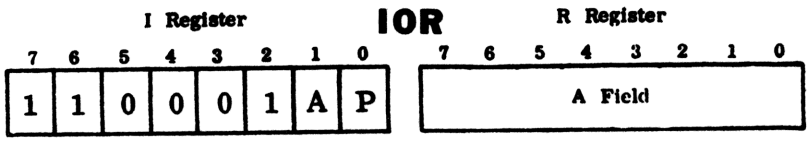


Fig. 17-17. IOR instruction.

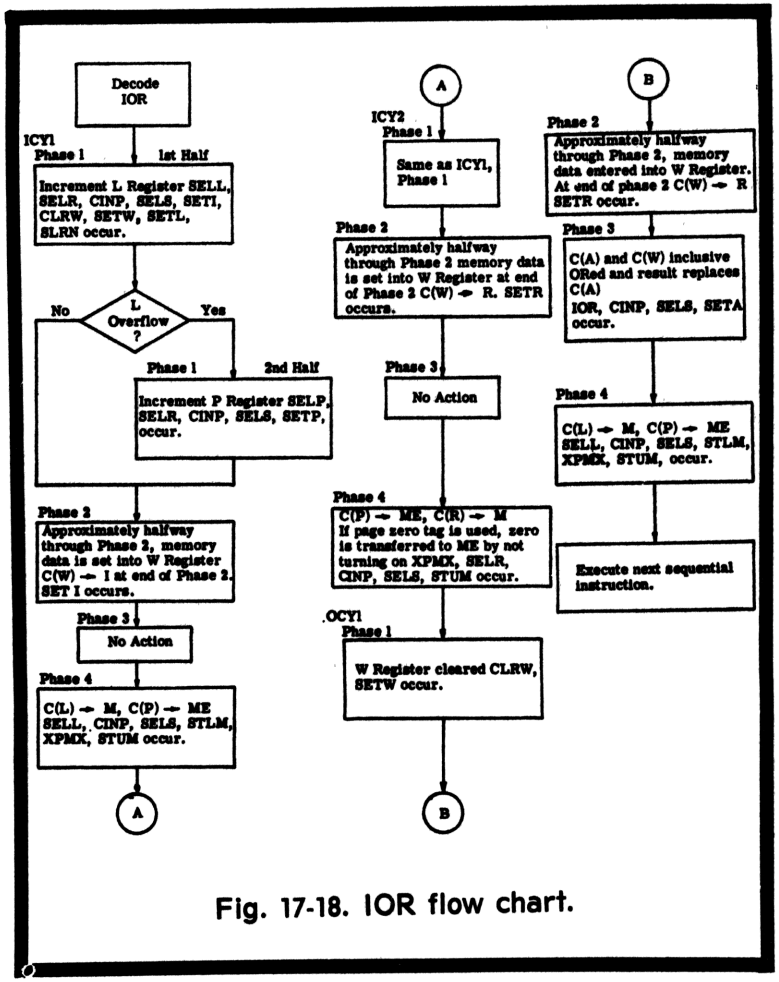


Fig. 17-18. IOR flow chart.

struction. Phase 3: Contents of the R and A registers are applied to the adder. Sum output entered into the A register. SELR, SELA, SELS and SETA occur. Phase 4: Same as phase 4, OCY1, STA instruction.

IOR Inclusive OR

IOR inclusively ORs the contents of the effective memory location to the contents of the A register. The timing is 3 plus 2 for each indirect level. See Figs. 17-17 and 17-18.

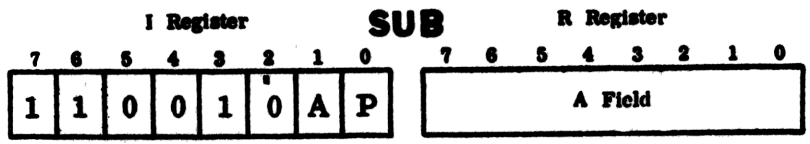


Fig. 17-19. SUB instruction.

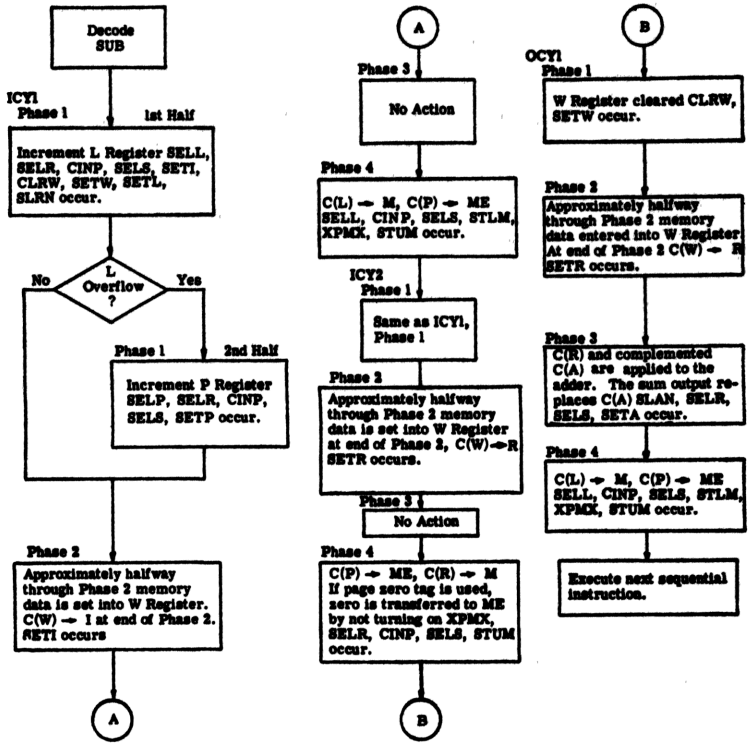


Fig. 17-20. SUB flow chart.

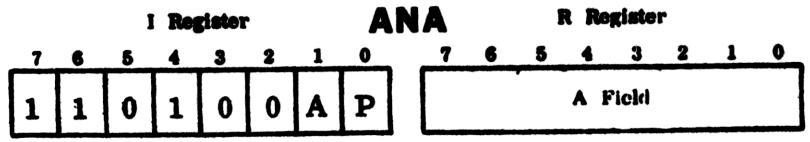


Fig. 17-21. ANA instruction.

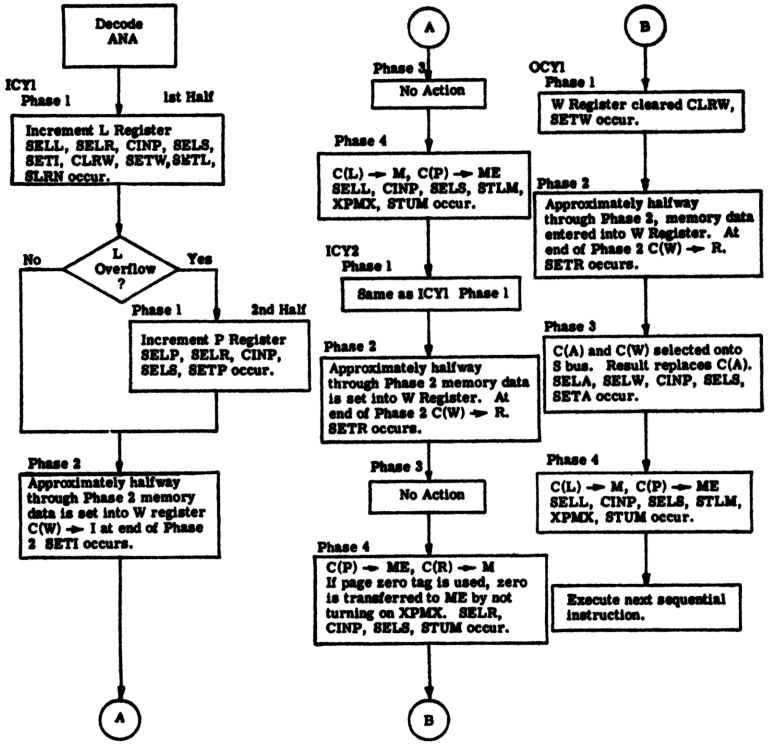


Fig. 17-22. ANA flow chart.

Cycle Sequence: ICY1, ICY2, OCY1

ICY1 cycle: Same as STA.
ICY2 cycle: Same as STA.

OCY1 cycle: Phase 1: Same as phase 1, OCY1, STA instruction. Phase 2: Same as phase 2, OCY1, LDA instruction. Phase 3: Contents of the A and W registers inclusive ORed result entered in the A register. IOR, CINP, SELS and SETA occur. Phase 4: Same as phase 4, OCY1 cycle, STA instruction.

SUB Subtract

SUB subtracts the contents of the effective memory location from the contents of the A register. The timing is 3 plus 2 for each indirect level. See Figs. 17-19 and 17-20.

Cycle Sequence: ICY1, ICY2, OCY1

ICY1 cycle: Same as STA.
ICY2 cycle: Same as STA.

OCY1 cycle: Phase 1: Same as phase 1, OCY1, STA instruction. Phase 2: Same as phase 2, OCY1 cycle, LDA instruction. Phase 3: Contents of the R register and the complemented contents of the A register are applied to the adder. The sum output is entered into the A register. SLAN, SELR, SELS and SETA occur. Phase 4: Same as phase 4, OCY1 cycle, STA instruction.

ANA AND to A

ANA ANDs the contents of the memory location specified by the A field and the C(A). Result is in A. The timing is 3 plus 2 for each indirect level. See Figs. 17-21 and 17-22.

Cycle Sequence: ICY1, ICY2, OCY1

ICY1 cycle: Same as STA.
ICY2 cycle: Same as STA.

OCY1 cycle: Phase 1: Same as phase 1, OCY1 cycle, STA instruction. Phase 2: Same as phase 2, OCY1 cycle, LDA instruction. Phase 3: Contents of the A and W registers are selected onto the S bus. Result is entered into the A register. SELA, SELW, CINP, SELS AND SETA occur. Phase 4: Same as phase 4, OCY1 cycle, STA instruction.

IMA Interchange A with Memory

IMA stores the contents of the A register in the effective memory location and loads the contents of the effective memory location into the A register. The timing is 3 plus 2 for each indirect level. See Figs. 17-23 and 17-24.

Cycle Sequence: ICY1, ICY2, OCY1

ICY1 cycle: Same as STA.
ICY2 cycle: Same as STA.

OCY1 cycle: Phase 1: Same as STA. Phase 2: Approximately halfway through phase 2 the memory data is entered in the W register. At the end of phase 2 the contents of the W register is transferred to the R register and the contents of the A register is transferred to the W register. SETW is delayed slightly from SETR to allow the transfer to take place. SELA, CINP, SELS, SETR and SETW occur. Phase 3: Contents of the R register is transferred to the A register. SELR, CINP, SELS and SETA occur. Phase 4: Same as phase 4, OCY1 cycle, STA instruction.

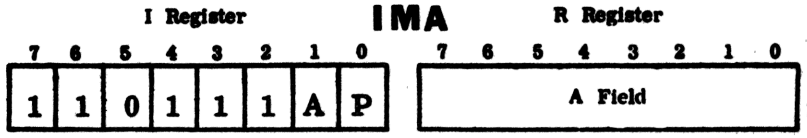


Fig. 17-23. IMA instruction.

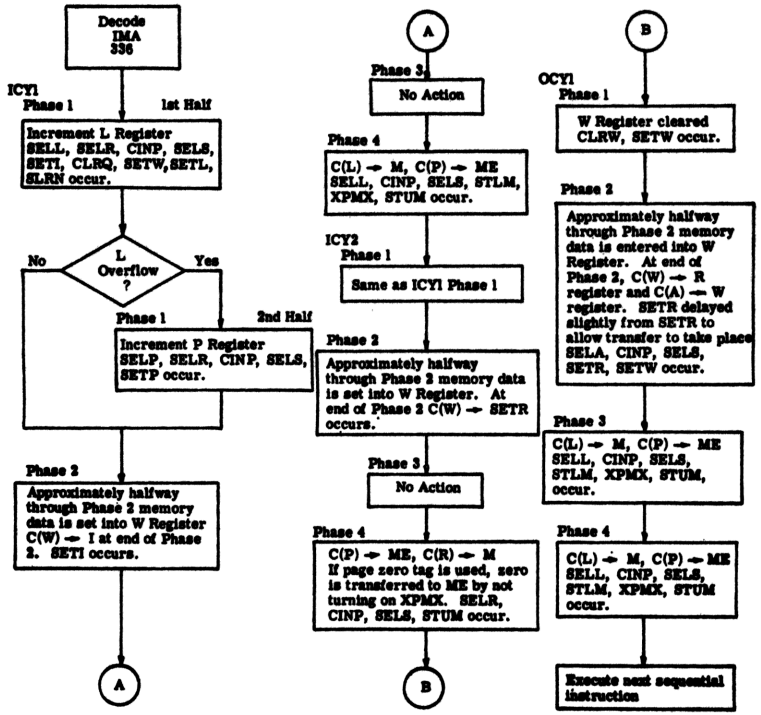


Fig. 17-24. IMA flow chart.

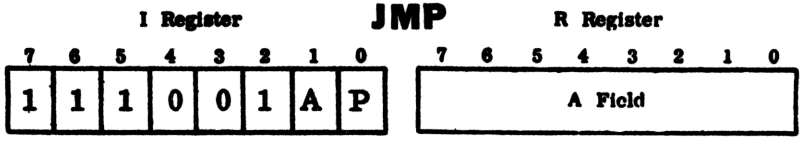


Fig. 17-25. JMP instruction.

JMP: Jump

The jump address is placed in the program counters for a direct jump to page zero; the P counter is reset and the A field transferred to L. A jump in the current page leaves P unchanged; an indirect jump causes the two indirect address words to be placed in the L and P counters. The timing is 2 plus 2 for each indirect level. See Figs. 17-25 and 17-26.

Cycle Sequence: ICY1, ICY2

ICY1 cycle: Same as STA.

ICY2 cycle: Phase 1. Same as phase 1, ICY2 cycle, STA

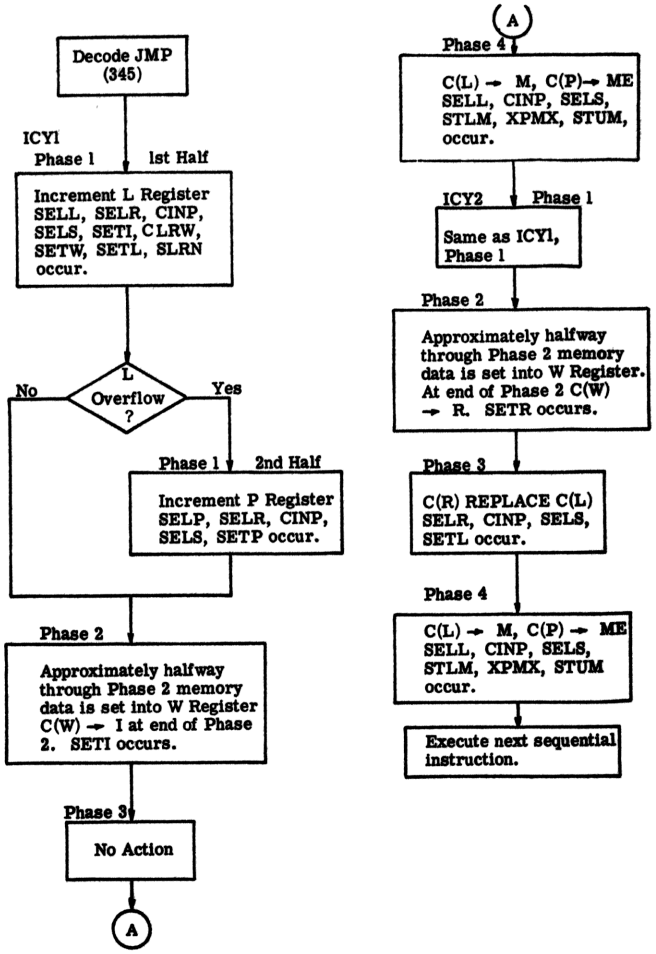


Fig. 17-26. JMP flow chart.

instruction. Phase 2. Same as phase 2, ICY2 cycle, STA instruction. Phase 3. Transfers the contents of the R register to the L register. SELR, CINP, SELS and SETL occur. Phase 4. Same as phase 4, ICY1 cycle, NOP instruction.

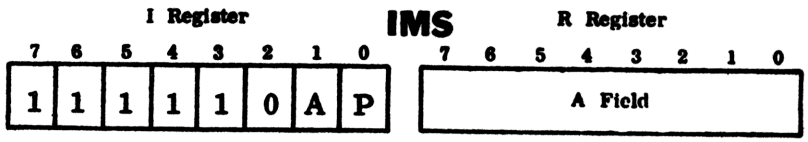


Fig. 17-27. IMS instruction.

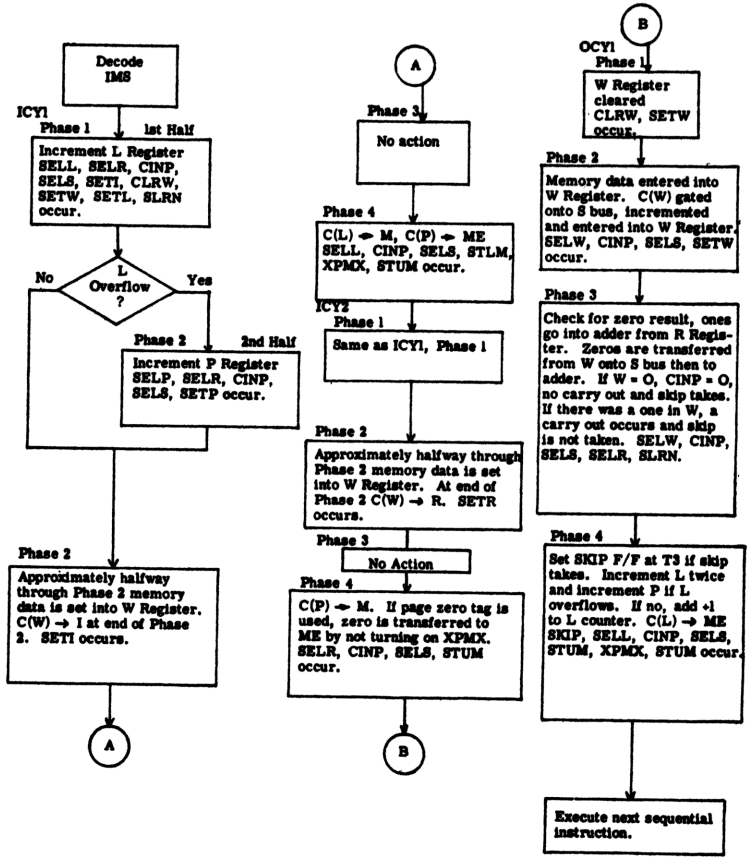


Fig. 17-28. IMS flow chart.

IMS: Increment Memory and Skip on Zero Result

The contents of the effective memory location is incremented by one count and replaced. If incrementing causes the result to become zero, a skip occurs. Overflow is set if the result of the incrementation is 200_8. Note: Neither skip nor OV occurs if IMS is executed as an interrupt instruction. The timing is 3 plus 2 per indirect level. See Figs. 17-27 and 17-28.

Cycle Sequence: ICY1, ICY2, OCY1

ICY1 cycle: Same as STA.
ICY2 cycle: Same as STA.
OCY2 cycle: Phase 1: Same as phase 1, OCY1 cycle, STA instruction. Phase 2. Memory data is entered into the W register. The W register's contents is gated onto the S bus, incremented and entered into the W register. SELW, CINP, SELS and SETW occur. Phase 3. Want to check for zero result, all 1s go into the adder from the R register. All zeros are transferred from W onto the S bus to the adder. If W was all zeros, CINP is 0 (no carry in) and there is no carry out. If there was a 1 in W, a carry out occurs and a skip is not taken. SELW, CINP-, SELS, SELR-, SLRN-, set skip F-F with T3. Phase 4: Same as skip, phase 54.

JST: Jump and Store L and P Counters

The contents of the L and P counters are stored in memory at the effective jump address and jump address plus one, and the counters are altered to contain jump address plus two. The timing is 4 plus 2 for each indirect level. See Figs. 17-29 and 17-30.

Cycle Sequence: ICY1, ICY2, OCY1, OCY2

ICY1 cycle: Same as STA.
ICY2 cycle: Same as STA.
OCY1 cycle: Phase 1: Same as Phase 1, OCY1, STA instruction. Phase 2. Approximately halfway through phase 2 the memory data is entered into the W register. At the end of Phase 2 the contents of W is transferred into the R register. After the transfer of W to R takes place the contents of L, which has been gated onto the A bus, is transferred into W. SELL, CINP, SELS, SETR and SETW occur. Phase 3: No Action. Phase 4: The contents of M is selected onto the S bus, incremented and placed back into the M register at T4 time. SELM, SELR, SLRN, CINP, SELS, SETM occur.

OCY2 cycle: Phase 1: Same as phase 1, OCY1 cycle, STA instruction. Phase 2: Approximately halfway through phase 2 the memory data is entered into the W register. At the end of Phase 2 the contents of the P register, which has been selected onto the A bus, is entered into the W register. SELP, CINP,

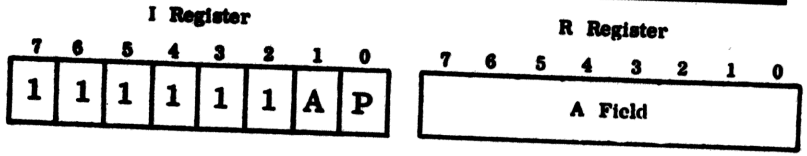


Fig. 17-29. JST instruction.

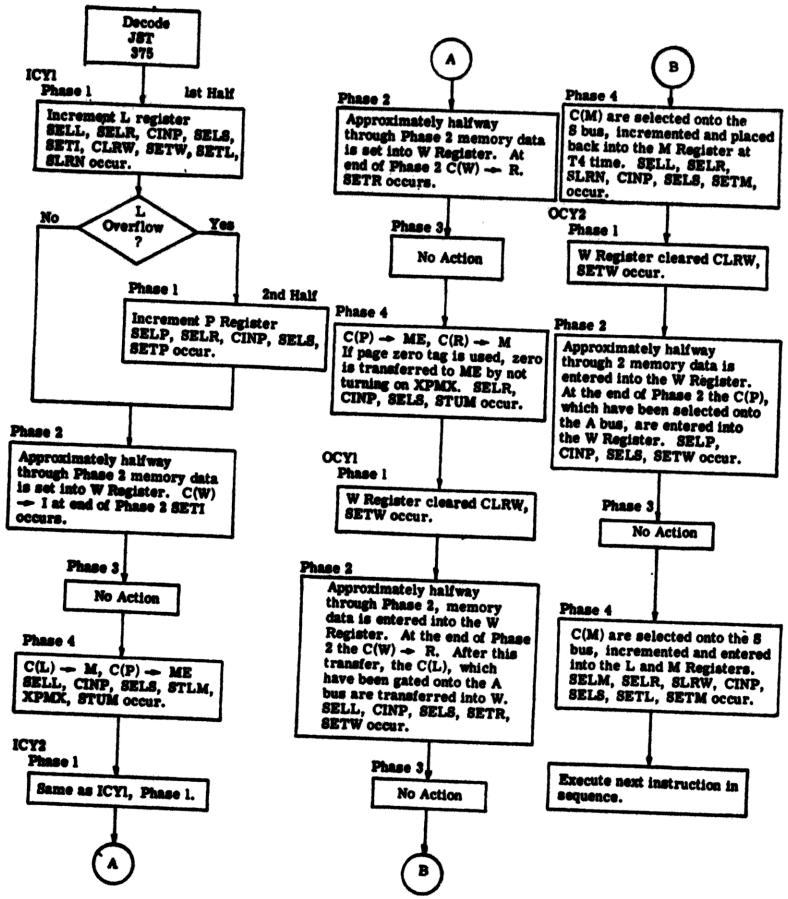


Fig. 17-30. JST flow chart.

SELS and SETW occur. Phase 3: No Action. Phase 4: Contents of the M register is selected onto the S bus, incremented and entered into the L and M registers. SELM, SELR, SLRN, CINP, SELS, SETL and SETM occur.

SKIP INSTRUCTIONS

The skip instructions allow the computer to make tests and take action based on the results of the tests. For this reason, they are referred to as IF TESTS, (IFT). "Skip if A equal to zero" tests the contents of the A Register and if the contents equals zero a skip occurs. Skips can also occur during the execution of instructions other than skips. Increment memory and skip on zero result, and I-O instructions involving sense can cause skips. When a skip occurs, the L register (counter) is incremented twice prior to accepting the next instruction. This causes two words to be skipped.

The sequence is as follows: During phase 3, the condition is tested. If the condition for a skip exists, the skip flip-flop is enabled and is set with T3 if IYAX is not on (a skip cannot be allowed during interrupts). Thus skip comes on during PH4X, the phase during which the address for the next memory cycle is usually formed, and stays on for 1.5 of the 2.0 microseconds of the phase. During this 1.5 microseconds, L is incremented twice, and P is incremented if L overflowed. During the last .5 microseconds of the phase, L and P are used to generate the memory address for the next instruction. NOTE: When manually stepping through a program, the computer normally stops when phase 4 comes on. If the instruction is a skip and the skip condition is met, the computer is not allowed to halt during phase 4, but instead made to continue and execute the next instruction.

Skip Logic

The skip flip-flop when set causes the L counter to be incremented twice, and if it overflows, the P counter is incremented. This causes two instruction locations to be skipped. A skip is not allowed if interrupt acknowledge is on since this would allow an interrupt to alter the L and P counters, causing havoc with the operating program.

Skip is set with T3 (if IUAX off) and is reset 1.5 microseconds later. Thus it is on for three-fourths of PH4. The L and P counters are modified during this time and the remaining 500 microseconds of PH4 are used to get the new address to the memory registers M and ME.

Normally the computer stops at the end of PH3 when going into the step mode. However, if a skip was about to be executed and the computer was allowed to stop at Te, skip would be set and when operation was resumed, even though L and P had been manually changed, the skip would cause L and P to be incremented. This is undesirable, hence the computer is not allowed to stop if skip is about to be set. This means that when manually stepping through a program containing skips, the computer will not stop at the end of a skip instruction if the skip condition is met, but will go ahead and make the skip, pick up the next instruction and execute it.

There are three ways the skip flip-flop can be set. By skip instructions, by increment memory and skip on zero result, and by block I-O executed under program control. In the first two cases, the skip condition is tested at the end of the instruction. In the case of block I-O, however, the skip condition is tested in the middle of the instruction (during ACY1) and the result of the test must be temporarily stored until the end of the instruction at which time skip is set. The storage device is a pair of gates cross coupled as a DC flip-flop and set by echo (if IUAX off). During OCY1, skip is enabled by this flip-flop if it has been set by echo. The format of the skip instruction appears in Chapter 12.

IFT-: Skip if A Negative

A skip will occur if the contents of the A register is less than zero. Otherwise, the next instruction in sequence is executed. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. C (A) less than zero?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflows. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT Z: Skip if A Zero

A skip occurs if the contents of the A register is zero. Otherwise, the next instruction in sequence is executed. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELA and SETP occur.

Phase 2: No action.

Phase 3: Test condition. C (A) equal 0?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflows. If no, add +1 to the L counter and execute the next instruction in sequence. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT-, Z: Skip if A Equal to or less than Zero

A skip occurs if the contents of A is zero or negative. Otherwise, the next instruction sequence is executed. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

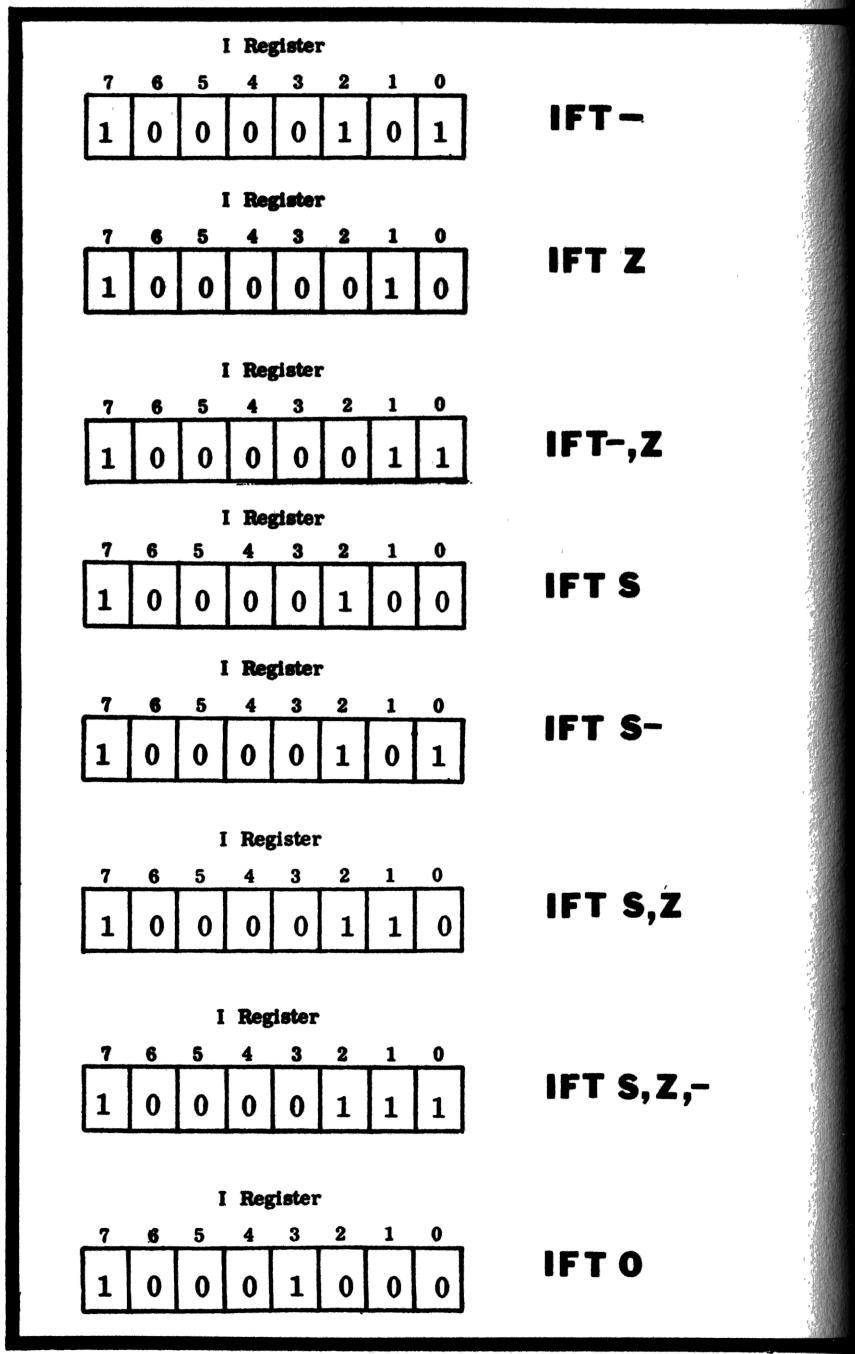
Phase 2: No action.

Phase 3: Test condition. C (A) $\overset{L}{\equiv}$ 0?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflows. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT S: Skip if OV Set

A skip occurs if the OV is set. Otherwise, execute the next sequential instruction. The timing is 1.



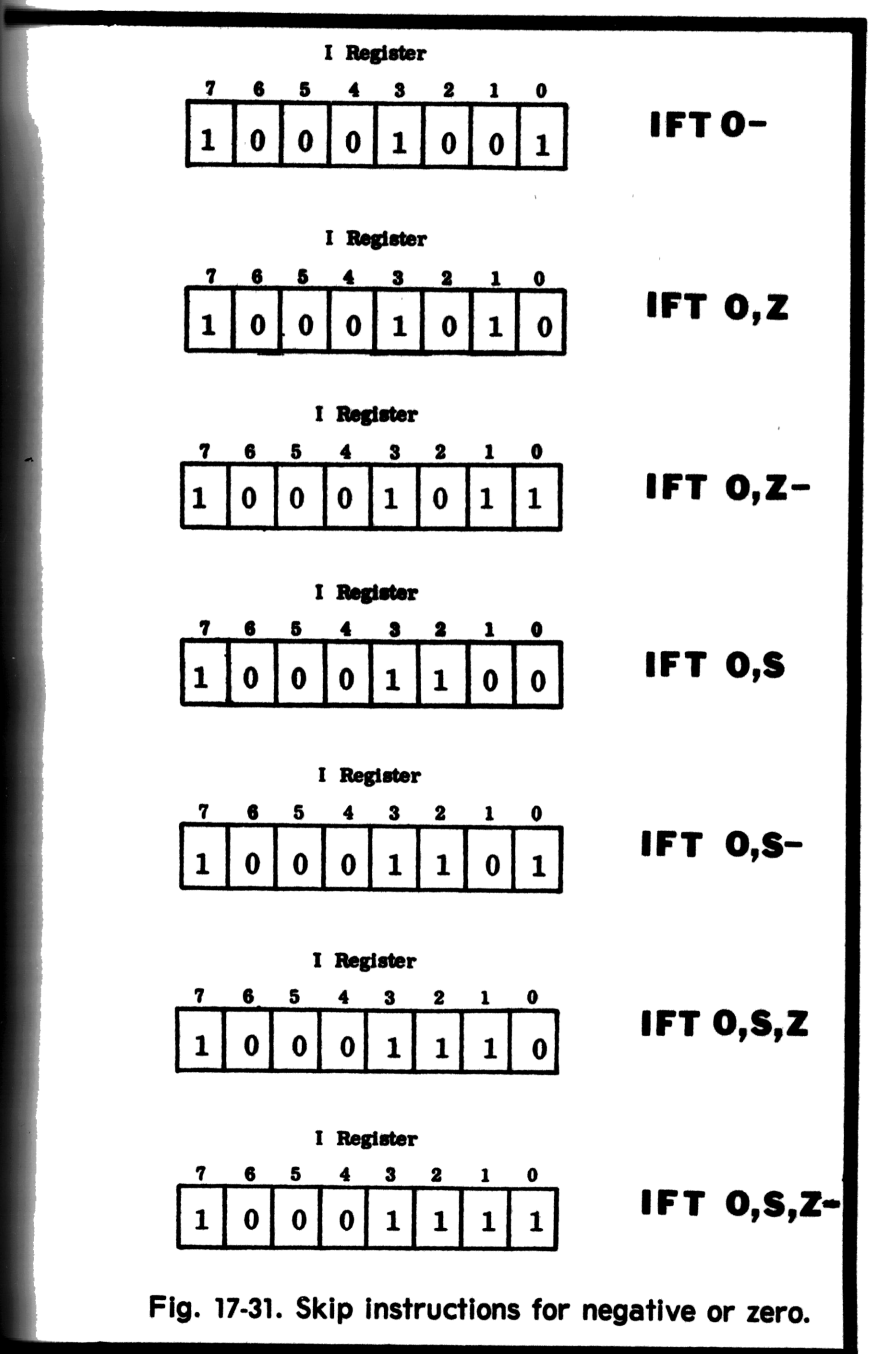


Fig. 17-31. Skip instructions for negative or zero.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is OV set?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflows. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT S-: Skip if OV Set Or A Negative

A skip occurs if the OV F-F is set or the C(A) is negative. Otherwise, execute the next sequential instruction. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the OV F-F set or are C(A) negative?

Phase 4: If either condition exists, set the skip F-F at T3. Increment L twice and increment P if L overflows. If neither condition exists, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT S, Z: Skip if OV Set Or A Zero

A skip occurs if the OV F-F is set or C(A) are 0. Otherwise, execute the next sequential instruction. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is OV F-F set or are C(A) equal to 0?

Phase 4: If either condition exists, set the skip F-F at T3. Increment L twice and increment P if L overflows. If neither condition exists, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT S, Z,-: Skip if OV Set or A Zero

A skip occurs if either OV F-F is set or C(A) are 0. Otherwise, execute the next sequential instruction. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

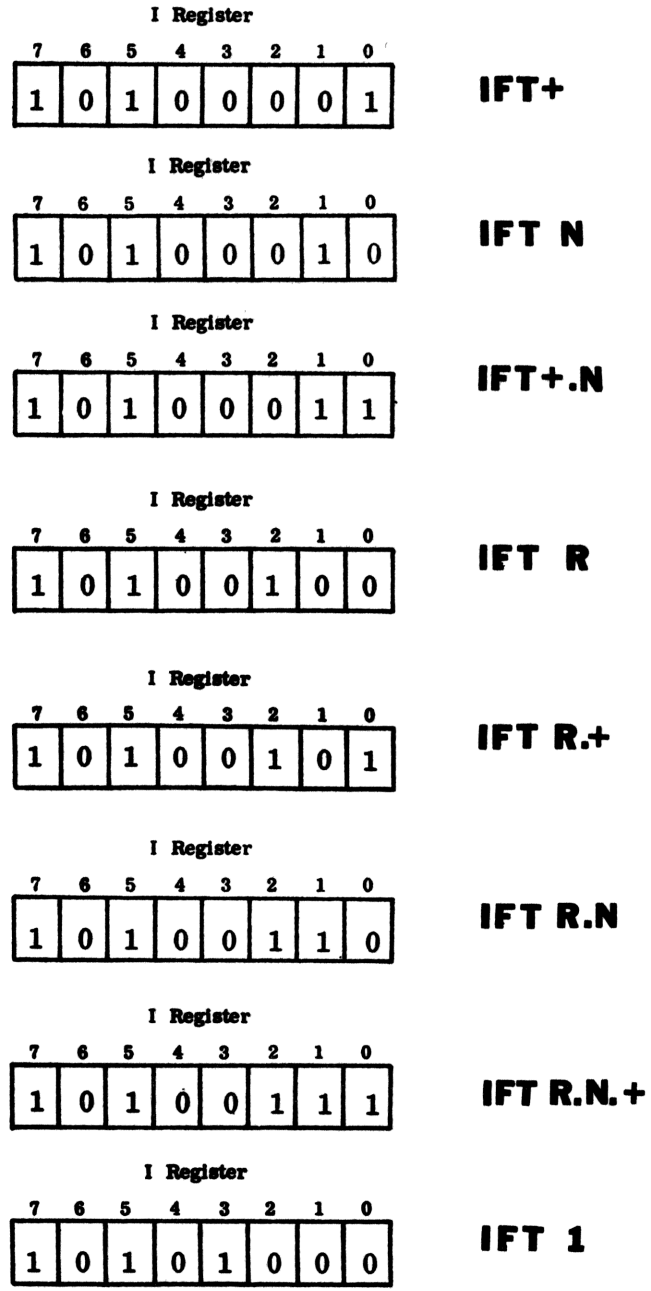
Phase 2: No action.

Phase 3: Test condition. Is either OV F-F set or are C(A) equal to 0?

Phase 4: If either condition exists, set the skip F-F at T3. Increment L twice and increment P if L overflows. If neither condition exists, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT 0: Skip if Sense Switch Off

A skip occurs if the console sense switch is off (not depressed). Otherwise, the next instruction in sequence is executed. The timing is 1.



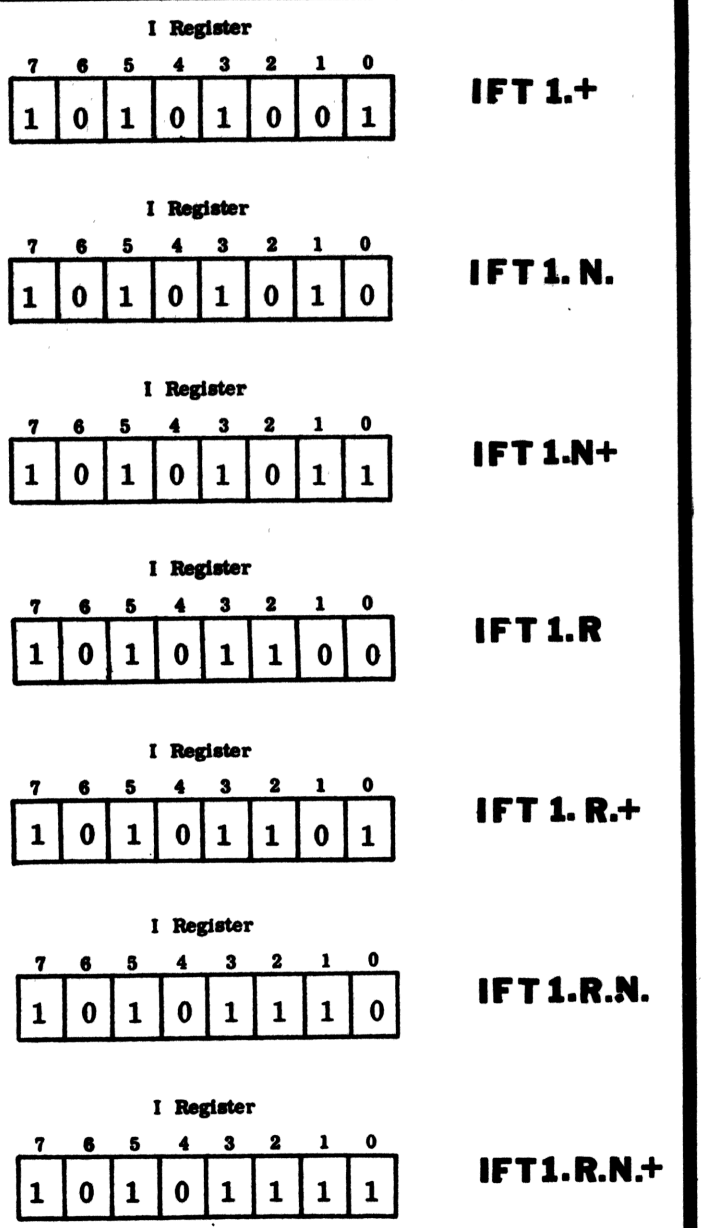


Fig. 17-32. Skip instructions for positive or non-zero.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the sense switch off?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflows. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT O- Skip if Sense Switch Reset or A Negative

A skip occurs if either the console sense switch is reset or C(A) are negative. Otherwise, execute the next sequential instruction. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is either console switch reset or are C(A) negative?

Phase 4: If either condition exists, set the skip F-F at T3. Increment L twice and increment P if L overflows. If neither condition exists, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT O, Z: Skip if Sense Switch Reset or A Zero

A skip occurs if the console sense switch is reset or C(A) are zero. Otherwise, execute the next sequential instruction. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the sense switch off or are C(A) equal to 0?

....Phase 4: If either condition exists, set the skip F-F at T3. Increment L twice and increment P if L overflows. If neither condition exists, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT O, Z- : Skip if Sense Switch Reset or A Zero or Negative

A skip occurs if the console sense switch is off or C(A) are 0 or C(A) are negative. Otherwise, execute the next instruction in sequence. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the sense switch reset or are C(A) equal to 0 or C(A) negative?

Phase 4: If any one of the conditions are met set the skip F-F at T3. Increment L twice and increment P if L overflows. If none of the conditions exist, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT O, S: Skip if Sense Switch Reset or OV Set

A skip occurs if either the console sense switch is reset or the OV F-F is set. Otherwise, execute the next sequential instruction. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the sense switch off or the OV F-F set?

Phase 4: If either condition exists, set the skip F-F at T3. Increment L twice and increment P if L overflows. If neither condition exists, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT O, S-,: Skip if Sense Switch Reset or OV Set

A skip occurs if either the console sense switch is reset or the OV F-F is set. Otherwise, execute the next sequential instruction. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of Phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the sense switch off or the OV F-F set?

Phase 4: If either condition exists, set the skip F-F at T3. Increment L twice and increment P if L overflows. If neither condition exists, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT O, S, Z: Skip if Sense Switch Reset or OV Set or A Zero

A skip occurs if any one of the three conditions is met. They are: 1. Sense switch off; 2. OV F-F set; 3. C(A) are 0. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test conditions. Sense switch off, or OV F-F set or C(A) equal to 0?

Phase 4: If any one of the conditions is met, set the skip F-F at T3. Increment L twice and increment P if L overflows. If none of the conditions exist, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT O, S, Z-: Skip if Sense Switch Reset or OV Set or A Zero or Negative

A skip occurs if any one of the following four conditions is met: 1. Sense switch off; 2. OV F-F set; 3. C(A) equal 0; 4. C(A) are negative. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test conditions. Are any of the conditions listed above met?

Phase 4: If any one of the conditions is met, set the skip F-F at T3. Increment L twice and increment P if L overflows. If none of the conditions exist, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

AND SKIP GROUP

The 15 instructions in this group are a combination of the four conditions. If more than one condition is specified (by placing a 1 in the condition bit) the AND of all specified con-

ditions must be met to cause a skip. For the skip instruction format, see Chapter 12.

IFT: Skip if A Positive

A skip occurs if the C(A) are positive. Otherwise, execute the next instruction in sequence. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS, and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Are the C(A) positive?

Phase 4: If yes set skip F-F at T3. Increment L twice and increment P if L overflowed. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, SINP, SELS, STLM, XPMX and STUM occur.

IFT N: Skip if A Non-Zero

A skip occurs if the contents of the A register is not zero. Otherwise, the next instruction in sequence is executed. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Does A contain any 1s?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflowed. If A contains all zeros, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT +.N: Skip if A Greater than Zero

A skip occurs if C(A) are greater than zero. Otherwise, the next instruction in sequence is executed. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is A positive and does it contain any 1s?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflowed. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT R: Skip if OV Reset

A skip occurs if OV is not on. Otherwise, the next instruction sequence is executed. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3. Test condition. Is the OV F-F set?

Phase 4: If the OV F-F is not set then set the skip F-F at T3. Increment L twice and increment P if L overflowed. If OV F-F is set, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT R.+: Skip if OV Reset and A Positive

A skip occurs if the OV F-F is reset and A is positive. Otherwise, execute the next instruction in sequence. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Are the C(A) greater than zero?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflowed. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT R.N: Skip if OV Reset and A Positive and Non-Zero

A skip occurs if the OV F-F is reset and C(A) are positive and greater than 1. Otherwise, execute the next sequential instruction. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the OV F-F reset and are C(A) positive and greater than 1?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflowed. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT R.N.+ Skip if OV Reset and A Positive and Non-Zero

A skip occurs if the OV F-F is reset and C(A) are positive and greater than zero. Otherwise, execute the next sequential instruction. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the OV F-F reset and are C(A) positive and greater than zero?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflows. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT 1: Skip if Sense Switch On

A skip occurs if the console sense switch is on. Otherwise. the next instruction in sequence is executed. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS, and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the sense switch on?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflowed. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT 1. +: Skip if Sense Switch Set and A Positive

A skip occurs if the console sense switch is on and C(A) are positive. Otherwise, execute the next instruction in sequence. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS, and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the sense switch on and are C(A) positive?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflows. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT 1.N: Skip if Sense Switch Set and A Non-Zero

A skip occurs if the console sense switch is on and the C(A) are greater than zero. Otherwise, execute the next sequential instruction. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the sense switch on and are C(A) more than zero?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflows. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT 1. N+: Skip if Sense Switch Set and A Positive and Non-Zero

A skip occurs if the console switch is set and C(A) are positive and non-zero. Otherwise, execute the next sequential instruction. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the sense switch set and are C(A) positive and greater than zero.

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflows. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT 1. R: Skip if Sense Switch Set and OV Reset

A skip occurs if the console sense switch is set and the OV F-F is reset. Otherwise, execute the next instruction in sequence. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the sense switch on and the OV F-F reset?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflows. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT 1. R.+: Skip if Sense Switch Set and OV Reset and A Positive

A skip occurs if the console sense switch is set and the OV F-F is reset and the C(A) are positive. Otherwise, execute the next instruction in sequence. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS, and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the sense switch on and the OV F-F reset and are C(A) positive?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflows. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT 1. R.N: Skip if Sense Switch Set and OV Reset and A Non-Zero

A skip occurs if the console sense switch is on and the OV F-F is reset and C(A) are non-zero. Otherwise, execute the next sequential instruction. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS, and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the sense switch on and the OV F-F reset and are C(A) non-zero?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflows. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

IFT 1. R.N.+: Skip if Sense Switch Set and OV reset and A Non-Zero And A Positive

A skip occurs if the sense switch is on, the OV F-F is reset and C(A) are non-zero and positive. Otherwise, execute the next instruction in sequence. The timing is 1.

Cycle Sequence: ICY1

Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

Phase 2: No action.

Phase 3: Test condition. Is the sense switch on and the OV F-F reset and are C(A) non-zero and positive?

Phase 4: If yes, set the skip F-F at T3. Increment L twice and increment P if L overflowed. If no, add +1 to the L counter and execute the next sequential instruction. Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SKIP, SELL, CINP, SELS, STLM, XPMX and STUM occur.

Figs. 17-33 through 17-44 are waveforms of various processor instructions. In all cases the ICY1 waveform appears at the top.

Fig. 17-33. ICY2 waveform (bottom) and ICY1. 5 microsecond sweep, 2v div.

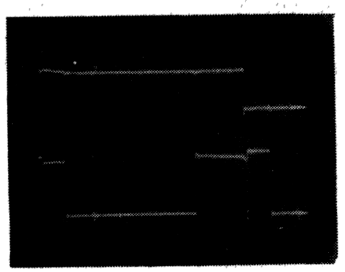


Fig. 17-34. SELL waveform (bottom) and ICY1. 1 microsecond sweep, 2v div.

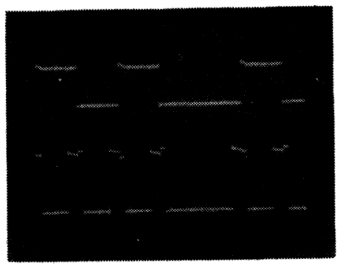


Fig. 17-35. SELL waveform (bottom) and ICY1. 5 microsecond sweep, 2v div.

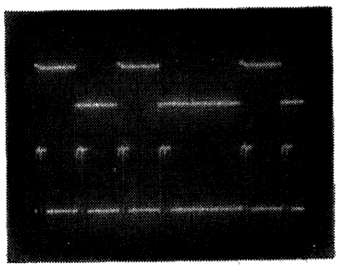


Fig. 17-36. SLRN waveform (bottom) and ICY1. 5 microsecond sweep, 2v div.

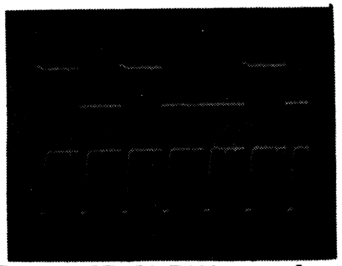


Fig. 17-37. CLRW waveform (bottom) and ICY1. 5 microsecond sweep, 2v div.

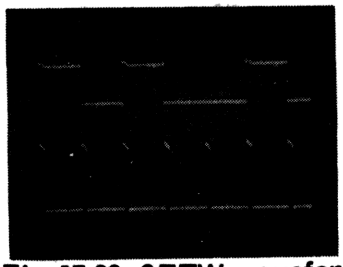


Fig. 17-38. SETW waveform (bottom) and ICY1. 5 microsecond sweep, 2v div.

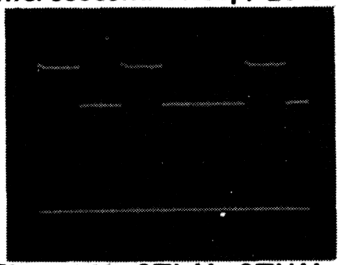


Fig. 17-39. STLM, STUM waveform (bottom) and ICY1.

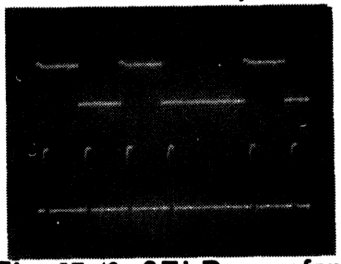


Fig. 17-40. SELP waveform (bottom) and ICY1. 5 microsecond sweep, 2v div.

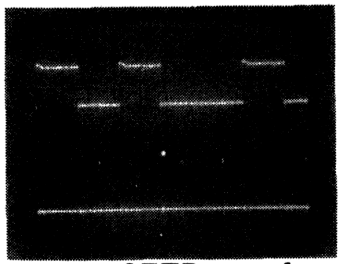


Fig. 17-41. SETR waveform (bottom) and ICY1. 5 microsecond sweep, 2v div.

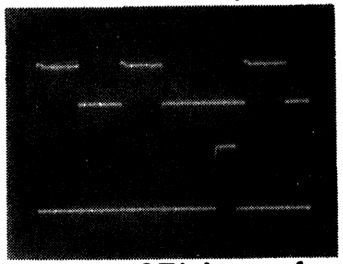


Fig. 17-42. SELA waveform (bottom) and ICY1. 5 microsecond sweep, 2v div.

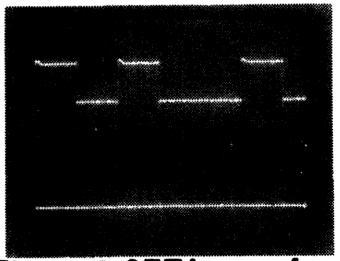


Fig. 17-43. SETA waveform (bottom) and ICY1. 5 microsecond sweep, 2v div.

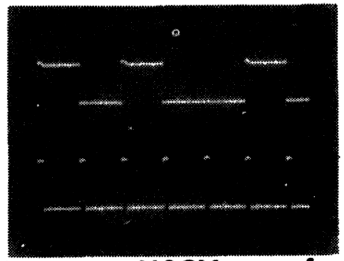


Fig. 17-44. MSCX waveform (bottom) and ICY1. 5 microsecond sweep, 2v div.

Chapter 18

Input-Output Operation

There are six types of I-O instructions: sense, external control, input to A, output from A, input block to memory, and output block from memory. By combining the sense and the input or output instructions two additional types are created: sense and input conditionally, and sense and output conditionally. Each I-O instruction may refer to one of 32 different device numbers or addresses. (The useful combination of I-O instructions in the 208 and 808 systems total over 1800 instructions.)

I-O instructions are double-word instructions as shown in Fig. 18-1. The first word specifies the class (I-O in this case), the operation code and special conditions available by microcoding. The second word specifies the device address (5 bits) of the device with which communication is to take place, and a 3-bit function code. The 3-bit function code allows up to eight functions to be created within the same flip-flops and may be tested (sensed) by executing the "sense device 4" instruction eight times with the eight different function codes.

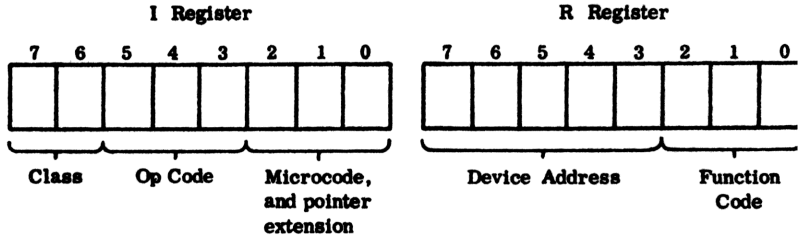


Fig. 18-1. Input-output instruction format.

DEVICE ADDRESSES

The following is a list of device address assignments for standard peripherals:

0		10	Real time clock
1	Computer automated tester	11	Magnetic tape
2		12	Magnetic tape
3		13	
4		14	
5	Card reader	15	
6	Paper tape reader-punch	16	
7	Teletype	17	Disc controller

20 to 27 This block is reserved for customer-designed interfaces and specials designed by CAI for customers.

30 to 37 These are reserved for utility modules.

Teletype Instructions

The teletype option uses 07 as the octal device address. The instructions associated with the teletype option are listed below.

100 071 — Select Keyboard: This instruction resets the buffer ready flip-flop and puts the teletype interface in the read mode.

100 072 — Step Read: This command causes the character under the read station on the paper tape reader to be read and the tape advanced one character. The reader switch on the teletype must be in the run position. The buffer ready flip-flop is reset.

100 073 — Select Continuous Read: This command causes the paper tape reader to continuously read at the rate of 10 characters per second until the reader is stopped or the tape runs out. The reader switch must be in the run position. The buffer ready flip-flop is reset.

100 074 — Initialize the Teletype Interface: This command resets the control flip-flops, stops the oscillator and puts the interface in a static marking condition. The buffer ready flip-flop is reset.

100 075 — Set Word Xfer Mask: This command sets a mask flip-flop in the interface to enable an interrupt to be generated by buffer ready flip-flop. (The interrupt line is wired according to system requirements.)

100 076 — Set Block Xfer Mask: This command sets a mask flip-flop in the interface to allow an interrupt to be generated when the word Xfer mask is in the off state. The interrupt can be used to indicate "end-of-block."

100 077 — Reset Masks: This instruction disables both interrupt lines in the teletype interface by resetting the mask flip-flops.

111 071 — Sense Buffer Ready: This instruction senses the on state of the buffer ready flip-flop; i.e., a true response will occur if the flip-flop is set.

111 072 — Sense Word Xfer Mask Off: This instruction senses the word Xfer mask flip-flop and generates a true response if the flip-flop is in the off state.

111 073 — Sense TTY Not Busy: This instruction senses the state of the TTY controller and generates a true response if the TTY is not printing or reading a character.

15X 070 — Output A Register to Teletype: This instruction transfers the contents of the A register (or all zeros) to the teletype interface and causes the character to be printed. If the punch is on, the character will also be punched.

14X 070 — Output Memory to Teletype: This is a block output instruction that causes one word of a block to be transferred from the memory location specified by X (in the instruction) and the output pointer (which the instruction automatically interrogates and updates). The character transferred is printed or punched on the teletype.

13X 070 — Input From Teletype to the A Register: The character in the teletype interface buffer is transferred to the A register. The word may be ANDed with the previous contents of A (masking) or A may be cleared prior to transfer.

12X 070 — Input From Teletype to Memory: This is a block input instruction that inputs the word from the teletype interface buffer to memory. The memory location is specified by X and the input pointer.

Paper Tape Reader Instructions

The paper tape uses 06 as the octal device address. The instructions associated with the paper tape reader are listed below.

100 064 — Initialize the Reader Interface: This instruction resets the control flip-flops in the interface and makes the "buffer not ready."

100 061 — Stop Reader: This instruction causes the reader to stop.

100 065 — Set Word Xfer Mask (WXM): This instruction causes the WXM flip-flop to be set, enabling interrupts to be generated with the buffer ready. (The actual use of interrupts is determined by system wiring.)

100 063 — Run: This instruction causes the reader to continuously slew tape at 30 inches per second.

100 062 — Step Read: This instruction causes the reader to read one character and stop.

100 066 — Set Block Xfer Mask (BXM): This instruction causes the BXM flip-flop to be set, enabling an interrupt to be generated when WXM is off; i.e., at end-of-block.

100 067 — Reset Masks: This instruction resets both the WXM and BXM flip-flops which disable interrupts for the buffer ready and end-of-block.

111 061 — Sense Buffer Ready: This instruction tests the state of the buffer ready flip-flop and generates a true response if the flip-flop is set.

111-062 — Sense Word Xfer Mask: This instruction tests the state of the WXM flip-flop and generates a true response if the flip-flop is off.

13X 060 — Input Buffer to A Register: This instruction transfers the contents of the buffer to the A register. Depending upon the value of X, the A register is either cleared first, or the contents of A is ANDed with the incoming data.

12X 060 — Input Buffer to Memory: This is a block input instruction that inputs the word from the buffer to memory. The memory location is specified by X and the contents of the input pointer.

Paper Tape Punch Instructions

The paper tape punch uses 06 as the octal device address. The instructions associated with the paper tape punch are listed below.

100 064— Initialize: This instruction resets all control flip-flops in the punch interface and makes the buffer "not ready."

100 060 — Punch Contents of Buffer: This instruction is used when copying an existing paper tape.

100 065 — Set Word Xfer Mask (WXM): This instruction causes the WXM flip-flop to be set, enabling an interrupt to be generated with the buffer ready. (The actual use of interrupts is determined by system wiring.)

100 066 — Set Block Xfer Mask (BXM): This instruction causes the BXM flip-flop to be set, enabling an interrupt to be generated when WXM is off; i.e., end-of-block.

100 067 — Reset Masks: This instruction resets both the WXM and BXM flip-flops which disable the interrupts for the buffer ready and end-of-block.

111 061 — Sense Buffer Ready: This instruction tests the state of the buffer ready flip-flop and generates a true response if the flip-flop is set.

111 062 — Sense Word Xfer Mask: This instruction tests the state of the WXM flip-flop and generates a true response if the flip-flop is off.

15X 060 — Punch Contents of A Register (or punch all zeros): This instruction causes the contents of the A register to be transferred to the punch interface and punched.

14X 060 — Output Memory and Punch: This instruction is a block output that outputs the word in memory specified by X and the output pointer. The word is transferred to the interface buffer and punched.

SENSE INSTRUCTIONS

To enable the controller to test the state of a device and take appropriate action, the sense skip instruction is used. The function code determines the condition to be tested. By decoding the three function code bits into the eight possible states, up to eight conditions may be tested in a particular device.

During PH3X of ICY2, the device being queried sends back a sense response if the condition being tested is met. If the condition is not met, no response is sent back to the controller. There is no sense control line. Instead, a sense occurs on every I-O command, but the processor ignores the response except for cases where it is applicable. There are two sense skip instructions: **Sense and Skip on Response** tests one of eight possible conditions within a device address. If a sense response is obtained at the processor within 1.0 microsecond after the beginning of PH3X during ICY2, the SCMX flip-flop is set and a skip occurs. **Sense and Skip on No Response** is identical except a skip occurs in the absence of a sense response.

SSR: Sense and Skip on Response

The function specified in Fig. 18-2 is tested in the device specified and a skip occurs if a response is obtained. If no response occurs, the next instruction in sequence is executed.

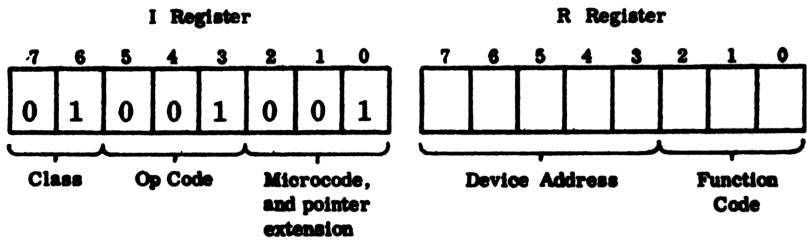


Fig. 18-2. Sense and skip on response instruction.

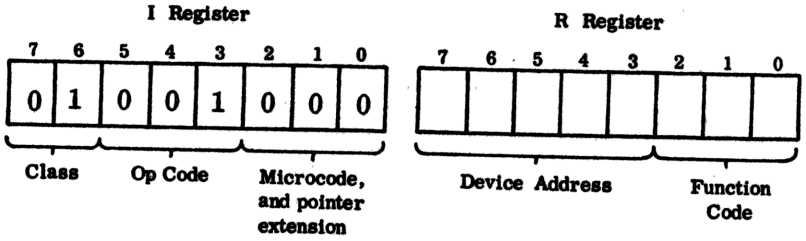


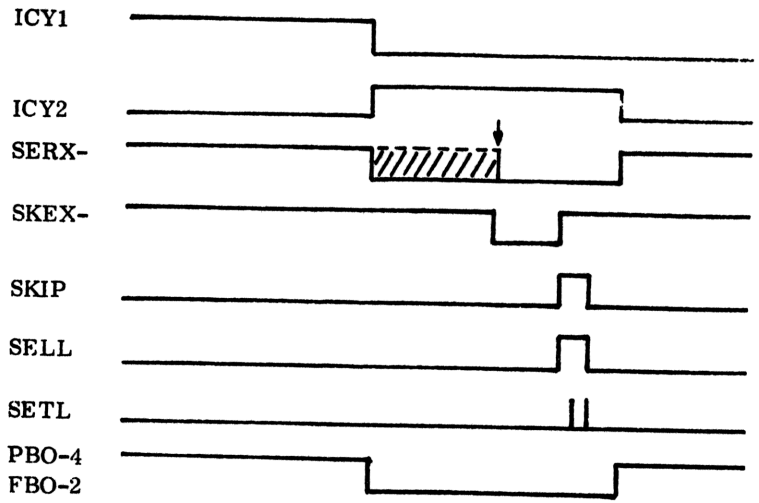
Fig. 18-3. Sense and skip on no response instruction.

Skip on response will occur if bit 0 of the I register is on. See Figs. 18-4 and 18-5. The timing is 2.

Cycle Sequence: ICY1, ICY2

ICY1 cycle: Phase 1: Same as NOP, phase 1. Phase 2: Same as NOP, phase 2. Phase 3: Same as NOP, phase 3. Phase 4: Same as NOP, phase 4.

ICY2 cycle: Phase 1: Same as NOP, phase 1. Phase 2: Same as NOP, phase 2. Phase 3: During ICY2, the device address (PB00-04) and the function code (FB00-02) are placed onto the peripheral address bus and function bus, respectively, for action in the I-O controller. Whether or not the



* Response must be at ground.

Fig. 18-4. SSR and SSN timing diagram.

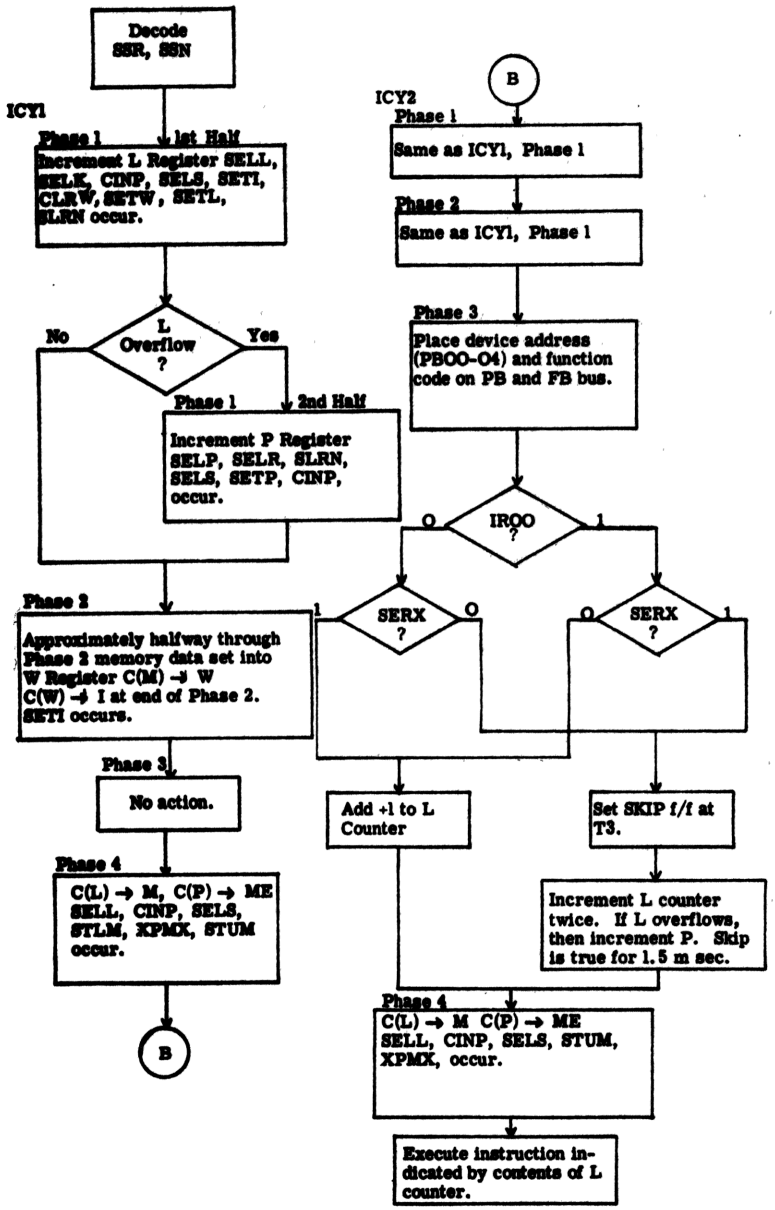


Fig. 18-5. SSR and SSN flow chart.

processor skips the next two words in the memory depends on the condition of the instruction register, bit 0, and the I-O signal (SERX-) from the I-O device being tested. A skip is taken if IR00 is a 1; i.e., SERX- is at ground (true). If the response is true the skip flip-flop is set at T3. PB0-4, FB0-2, SERX-, SKEX-, SKIP, SELL, SETL occur. Phase 4: Same as NOP, phase 4.

SSN: Sense and Skip on No Response

The function specified in Fig. 18-3 is tested in the device specified and a skip occurs if a response is not obtained. If a response occurs, the next instruction in sequence is executed. See Fig. 18-4. The timing is 2.

Cycle Sequence: ICY1, ICY2

ICY1 cycle: Phase 1: Same as NOP, phase 1. Phase 2: Same as NOP, phase 2. Phase 3: Same as NOP, phase 3. Phase 4: Same as NOP, phase 4.

ICY2 cycle: Phase 1: Same as NOP, phase 1. Phase 2: Same as NOP, phase 2. Phase 3: During ICY2, the device address (PB00-04) and the function code (FB00-02) are placed onto the peripheral address bus and function bus, respectively, for action in the I-O controller. In this case, the skip is taken; i.e., the L counter is incremented twice if IR00 is a 0 and there is no response. This is indicated by the signal SERX- being at 3v. If the skip flip-flop is not set, the next instruction in sequence is executed. SKEX-, SELL, SETL, SERX- occur. Phase 4: Same as NOP, phase 4.

EXTERNAL CONTROL INSTRUCTION

To enable the computer to control peripheral logic the external control (EXC) command is used. The EXC steers a pulse to one of eight destinations within a device address. The pulse may be used to set flip-flops, increment a counter, latch a relay, etc. The EXCX control line goes true for 2.0 microseconds (PH3X) during ICY2. The function code is decoded into eight codes to allow steering of the pulse signal, PLSE. Thus up to eight EXC functions may be implemented within a device address.

EXC: External Control

The control function specified by the function code in Fig. 18-6 is executed. The timing is 2. See Figs. 18-7 and 18-8.

Cycle Sequence: ICY1, ICY2

ICY1 cycle: Phase 1: Same as NOP, phase 1. Phase 2: Same as NOP, phase 2. Phase 3: Same as NOP, phase 3. Phase 4: Same as NOP, phase 4.

ICY2 cycle: Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS, SETB occur. Phase 2: Approximately halfway through phase 2 the memory data is set into the W register. C(W) are transferred to the I register at the end of the phase. SETI occurs. Phase 3: The device address (PB00-04) and the function code (FB00-02) are placed onto the peripheral address bus and function bus, respectively, for action in the I-O controller. During phase 3, two control signals are on the I-O control bus. They are the EXCX- and PLSE- signals. Phase 4: Same as NOP, phase 4.

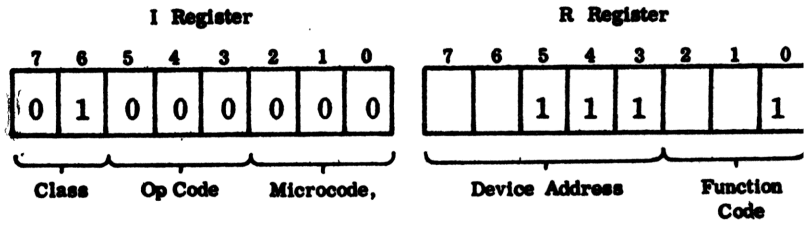


Fig. 18-6. External control instruction.

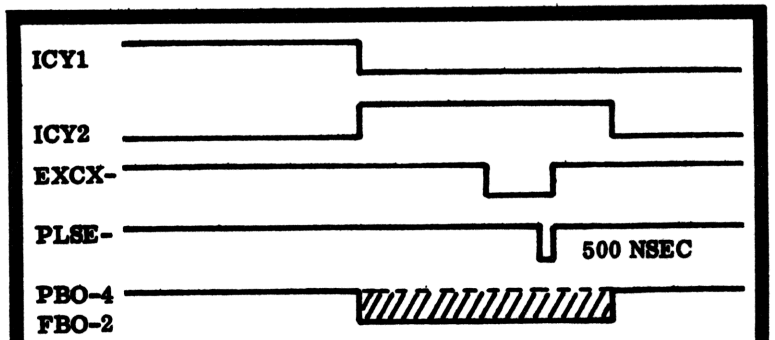


Fig. 18-7. Timing diagram for the EXC instruction.

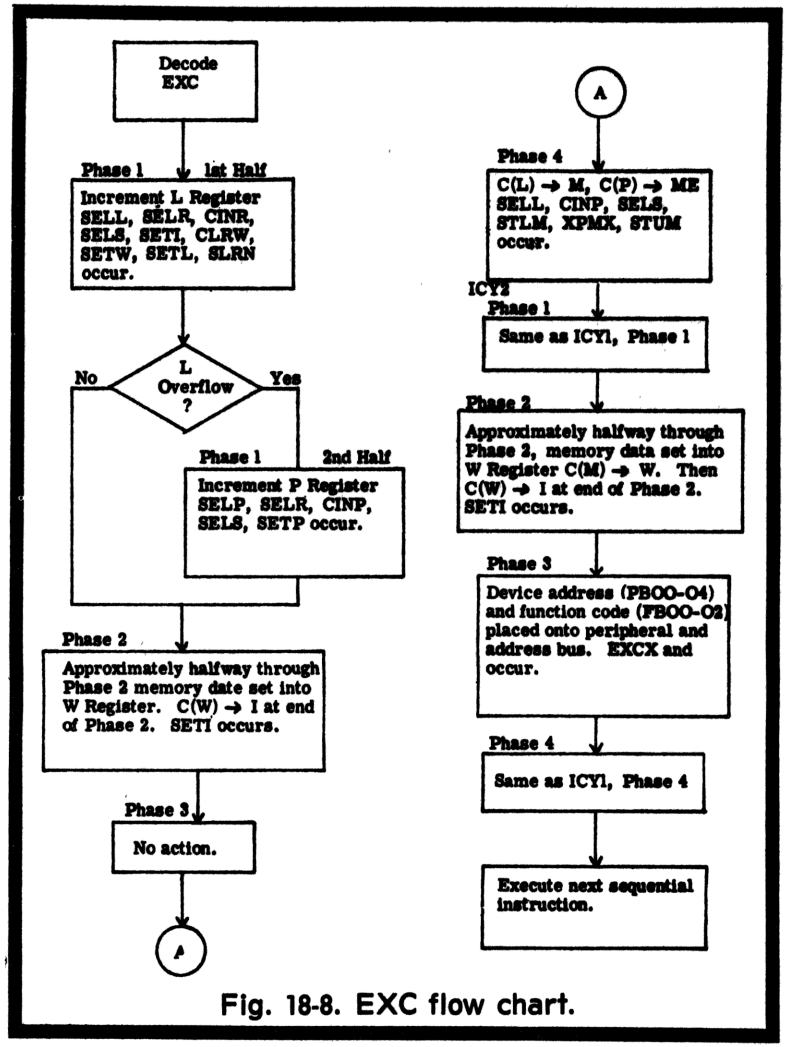


Fig. 18-8. EXC flow chart.

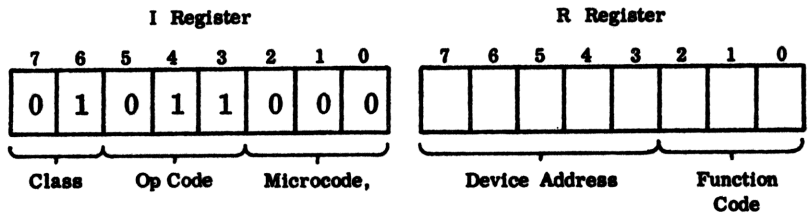


Fig. 18-9. Input to A register (unconditionally) instruction.

INPUT TO A REGISTER INSTRUCTIONS

There are four variations to the input to A instructions:

1. Input to A unconditionally causes the contents of the D bus to be transferred to the S bus, from the S bus to the adder (SELR and SLRN both off, CINP true) and from the adder to the A bus where it is strobed into A. PLSE occurs at the end of PH3 of ICY2 as in the case of output A unconditionally.

2. Input to A on sense response is identical to the unconditional input, except PLSE and SETA are gated by SCMX. If the sense response is not obtained halfway through PH3X, then PLSE and SETA are inhibited and a skip does not occur.

3. Masked input to A unconditionally is identical to the unconditional input except the contents of A is selected onto the S bus along with the contents of the D bus. Selecting A and D bus contents simultaneously causes the AND of the two, which is routed around to the A bus where it is strobed into A.

4. Masked input to A on sense response is identical to the unconditional masked input except PLSE and SETA are inhibited if the sense response is not received within 1.0 microsecond after the beginning of HH3X. If the response is received PLSE and SETA occur and a skip occurs.

The input to A instructions are used to transfer data from an output device to the processor A register. This is accomplished by generating the required signals which allows the device to place information on the data bus, DB00-07. This is accomplished by INXX, which indicates that an input instruction is being executed and the device addressed. The generation of these signals causes data to be placed on the D bus. The processor then selects data onto the S bus with SDBR, which enables the contents of the D bus to be placed on the S bus. This selected data is routed to the A bus and a SETA pulse causes the data to be entered into the A register. The SETA and PLSE- are generated under control of the sense response (SERX-) when executing an IAS or MIS instruction. SETA and PLSE- always occur on INA and MIA.

When conditions in the I-O device must be met, the signal PLSE- is an indication that they have been met. Instructions that are executed unconditionally also always generate PLSE-; i.e., the sense response is always true. After the conditions have been met, transfer of data occurs. The trailing edge of PLSE- is an indication that transfer has been completed. The conditions which give a sense response are: sense buffer ready, sense word transfer mask off, and sense device not busy.

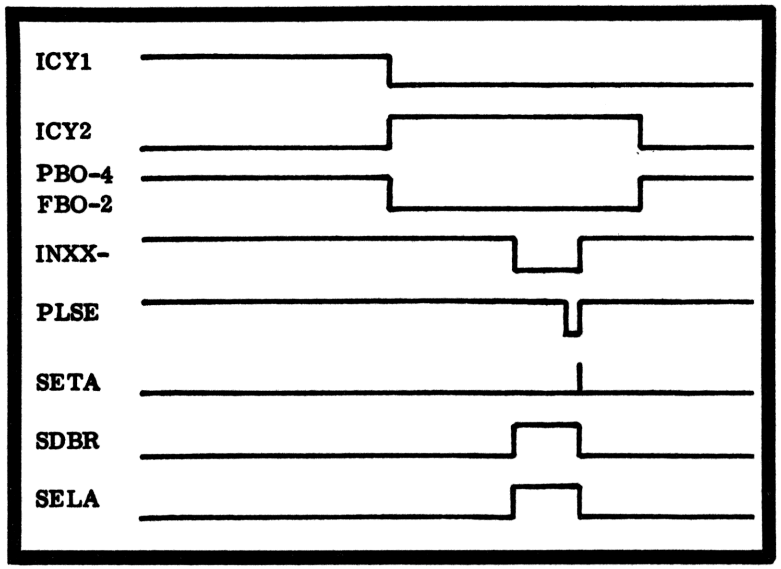


Fig. 18-10. Timing diagram of the INA, IAS, MIS, and MIA.

INA: Input to A Register (Unconditionally)

A data word will be transferred from the device specified to the A register. The function code can be used to designate the source of data within the device if multiple sources exist as in the case of a 16-bit converter output. Up to eight sources may be specified by the function code, allowing a 64-bit input data word to be handled in eight successive inputs, greatly simplifying the device logic. The timing is 2. See Figs. 18-10 and 18-11.

Cycle Sequence: ICY1, ICY2

ICY1 cycle: Phase 1: During the first half of phase 1, increment the L register. SELL, SELR, CINP, SELS, SETI, CLRW, SETW occur. If L overflows, increment the P register during the second half of phase 1. SELP, SELR, CINP, SELS, SETP occur. Phase 2: Approximately halfway through phase 2, memory data is set into the W register. C(W) go to the I register at the end of phase 2. SETI occurs. Phase 3: No action. Phase 4: C(L) go to M, C(P) to ME. SELL, CINP, SELS, STLM, XPMX, STUM occur.

ICY2 cycle: Phase 1: Same as ICY1, phase 1, except that the device address (DB00-05) and function code (FB00-02) are placed on the DB and FB bus, respectively. PB00-04, FB00-02 occur. Phase 2: Approximately halfway through phase 2, memory data is set into the W register. C(W) arrive at I at the end of phase 2. SETI occurs. Phase 3: The I-O device puts data on the I-O bus. Data then is loaded into A. INXX, PLSE, SDBR, SELA, SETA, occur! Phase 4: C(L) go to M and C(P) to ME. SELL, CINP, SELS, STLM, XPMX, STUM occur.

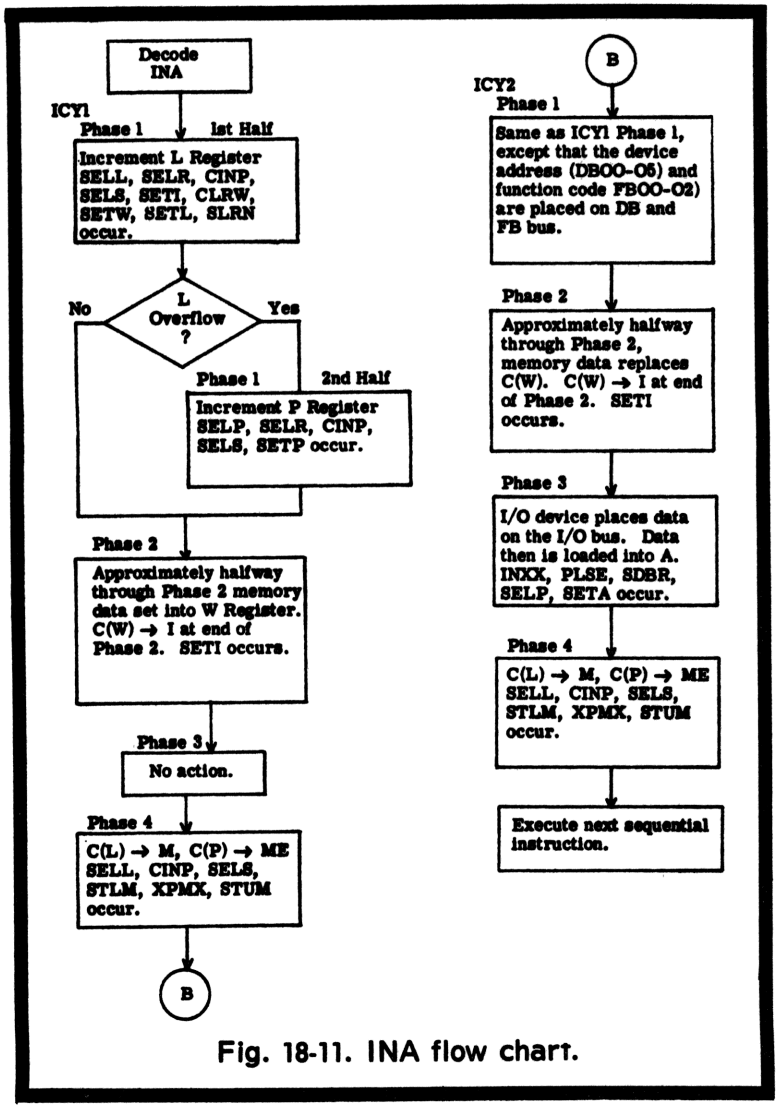


Fig. 18-11. INA flow chart.

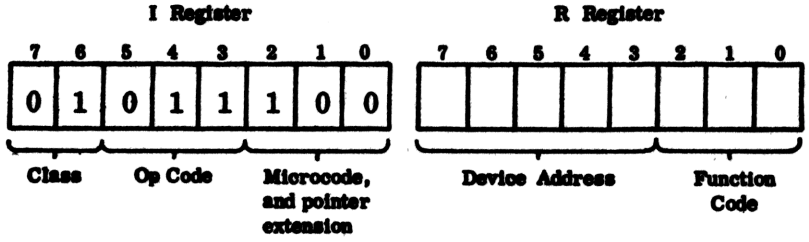


Fig. 18-12. Masked input to A register (unconditionally) instruction.

MIA: Masked Input to A Register (Uncondtionally)

The masked input instruction in Fig. 18-12 causes the input data word from the specified device to be ANDed with the previous contents of the A register and the results placed in the A register. The source of the data word within the device may be specified by the function code if multiple sources exist. The timing is 2. See Figs. 18-10 and 18-13.

Cycle Sequence: ICY1, ICY2

ICY1 cycle: Phase 1: During the first half of phase 1, increment the L register. SELL, SELR, CINP, SELS, SETI, CLRW, SETW occur. If L overflows, increment the P register during the second half of phase 1. SELP, SELR, CINP, SELS, SETP occur. Phase 2: Approximately halfway through phase 2, memory data is set into the W register. C(W) go to I at the end of phase 2. SETI occurs. Phase 3: No action. Phase 4: C(L) go to M, C(P) to ME. SELL, CINP, SELS, STLM, XPMX, STUM occur.

ICY2 cycle: Phase 1: Same as ICY1, phase 1, except that the device address (DB00-05) and function code (FB00-02) are placed on the DB and FB bus, respectively. PB00-04, FB00-02 occur. Phase 2: Approximately halfway through phase 2, memory data replaces C(W). C(W) go to I at the end of phase 2. SETI occurs. Phase 3: The I-O device places data on the I-O bus. Data is then ANDed with the previous contents of A. Results are placed back into A. SELA, SETA, STA3, INXX, PLSE, SDBR occur. Phase 4: C(L) go to M and C(P) to ME. SELL, CINP, SELS, STLM, XPMX, STUM occur.

IAS: Input to A and Skip on Response

The IAS instruction first senses the function specified by the function code within the device specified. If a response is

obtained; i.e., device is ready, the data word is transferred to the A register and a skip occurs. If no response is obtained, the data word is not transferred, the A register is unchanged, and the next instruction in sequence is executed. The timing is 2. See Figs. 18-10 and 18-15.

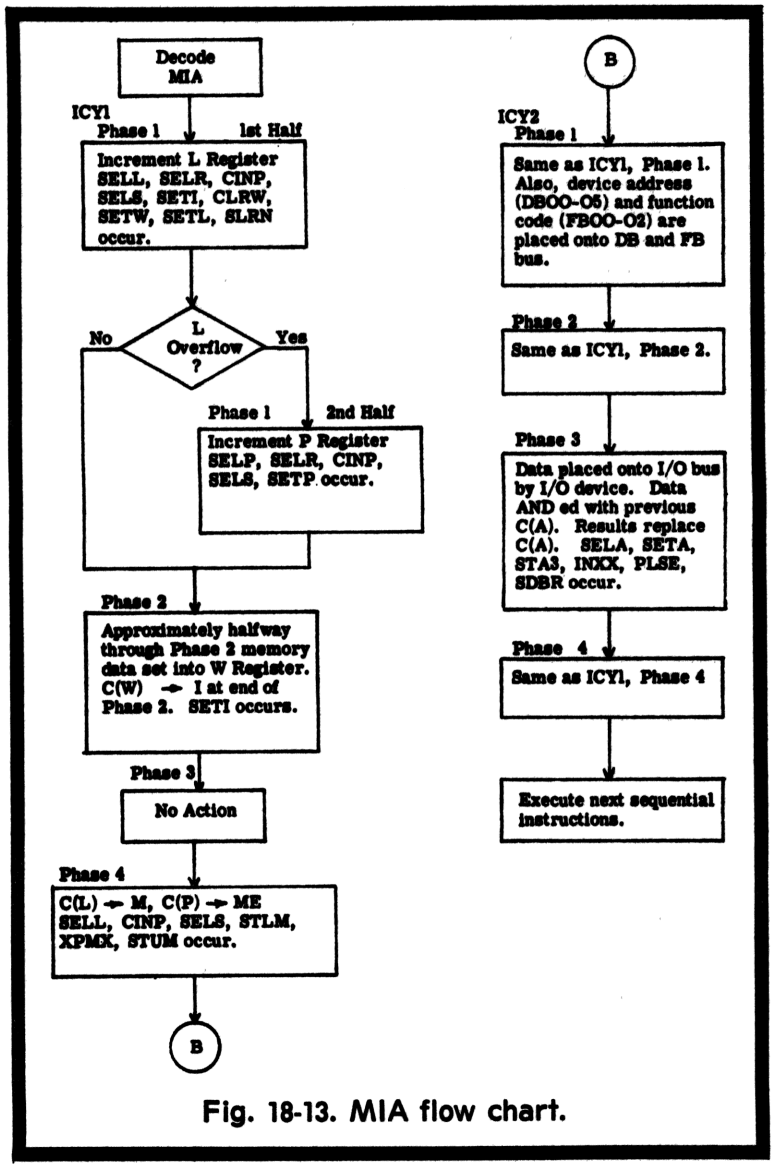


Fig. 18-13. MIA flow chart.

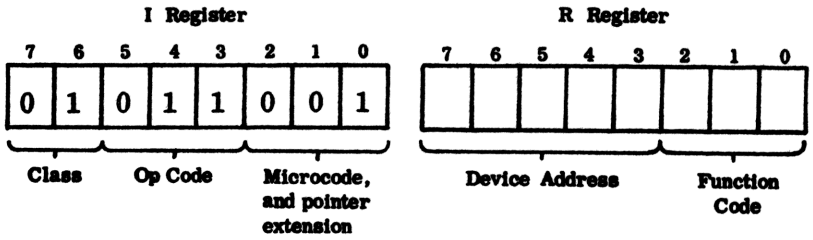


Fig. 18-14. Input to A and skip on response instruction.

Cycle Sequence: ICY1, ICY2

ICY1 cycle: Phase 1: During the first half of phase 1, increment the L register. SELL, SELR, CINP, SELS, SETI, CLRW, SETW occur. If L overflows, increment the P register during the second half of phase 1. SELP, SELR, CINP, SETP occur. Phase 2: Approximately halfway through phase 2, memory data is set into the W register. C(W) go to the I register at the end of phase 2. SETI occurs. Phase 3: No action. Phase 4: C(L) go to M, C(P) to ME. SELL, CINP, SELS, STLM, XPMX, STUM occur.

ICY2 cycle: Phase 1: Same as ICY1, phase 1, except that the device address (DB00-05) and function code (FB00-02) are placed on the DB and FB bus, respectively. PB00-04, FB00-02 occur. Phase 2: Approximately halfway through phase 2, memory data is set into the W register. C(W) go to I at the end of phase 2. SETI occurs. Phase 3: If the device addressed is ready, the condition tested is true; i.e., a response occurred. The data word is transferred to the A register. PLSE, SETA, SELA, SDBR, INXX occur. If the device addressed is not ready, no response occurs and the condition being tested is false. The data word is not transferred. INXX occurs. Phase 4: If the condition being tested is true the skip is taken; i.e., the L register is incremented twice prior to receiving the next instruction. Also, P is incremented if L overflowed. The skip F-F is set at T3XX if IYAX is not on. A skip is not allowed during interrupts. L and P are then used to generate the memory address for the next instruction. SKIP F-F is set at T3XX, SELL, SELP, SKIP, SETL, SETP occur. If the condition being tested is false a skip is not taken and the next instruction in sequence is executed.

MIS: Masked Input to A and Skip on Response

The instruction in Fig. 18-16 is a combination of the sense and the marked input to A. The instruction first senses the

condition specified by the function code within the device specified. If a sense response is obtained, a masked input to the A register and a skip occur. If no sense response is obtained, the A register is left unaltered and the next instruction in sequence is executed. The timing is 2. See Figs. 18-10 and 18-17.

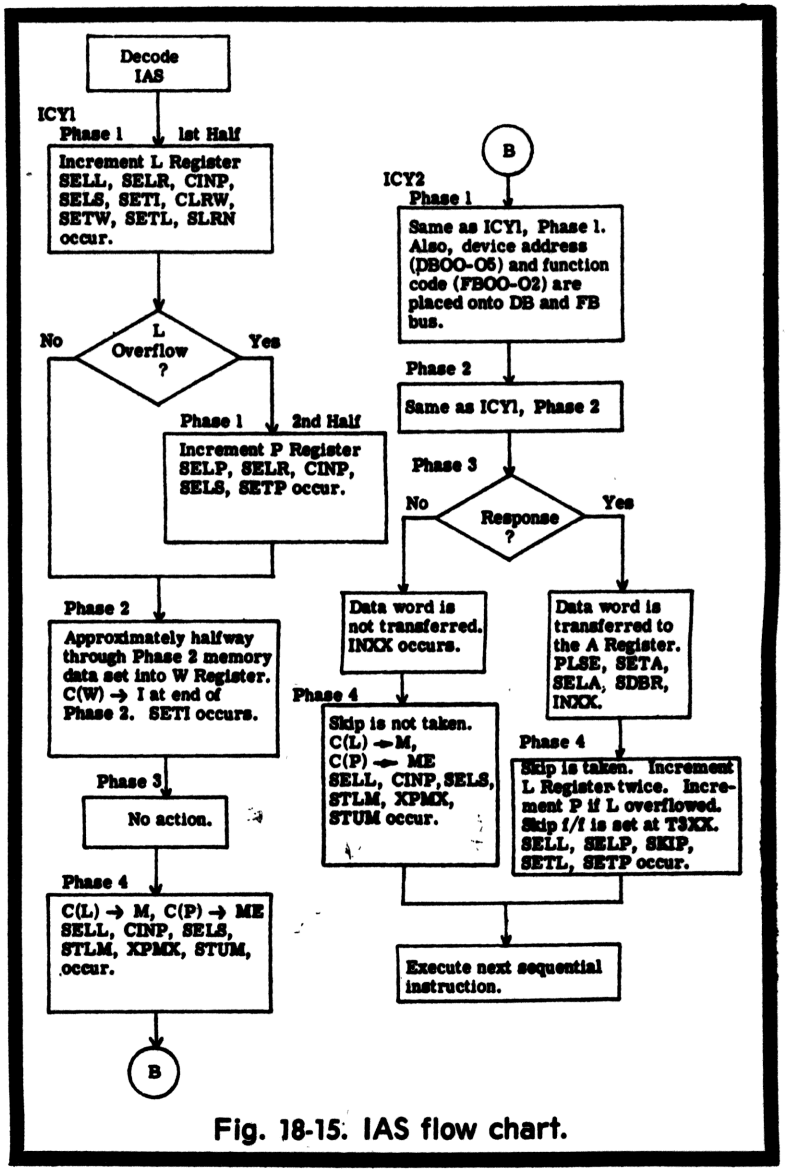


Fig. 18-15. IAS flow chart.

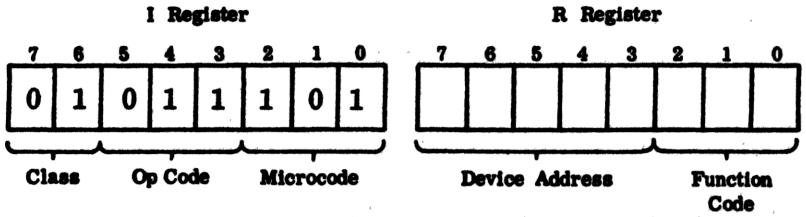


Fig. 18-16. Masked Input to A and skip on response instruction.

Cycle Sequence: ICY1, ICY2

ICY1 cycle: Phase 1: During the first half of phase 1, increment the L register. SELL, SELR, CINP, SELS, SETI, CLRW, SETW occur. If L overflows, increment the P register during the second half of phase 1. SELP, SELR, CINP, SELS, SETP occur. Phase 2: Approximately halfway through phase 2, the memory data is set into the W register. C(W) go to the I register at the end of phase 2. SETI occurs. Phase 3: No action. Phase 4: C(L) go to M, C(P) to ME. SELL, CINP, SELS, STLM, XPMX, STUM occur.

ICY2 cycle: Phase 1: Same as ICY1, phase 1, except that the device address (DB00-05) and function code (FB00-02) are placed on the DB and FB bus respectively. Phase 2: Approximately halfway through phase 2, memory data is set into the W register. C(W) go to I at the end of phase 2. SETI occurs. Phase 3: If a response is obtained; i.e., if a device is ready to transmit data, the condition is true. The input data word from a specified device is ANDed with the previous contents of the A register and the results placed back into A. PLSE, SDBR, SELA, SETA, INXX occur. If no response is obtained; i.e., the device is not ready to transmit data, the A register is not changed. INXX occurs. Phase 4: If the condition being tested is true the skip is taken; i.e., the L register is incremented twice prior to introducing the next instruction. Also, P is incremented if L overflowed. The skip F-F is set at T3XX if IYAX is not on. A skip is not allowed during interrupts. L and P are then used to generate the memory address for the next instruction. Skip F-F is set at T3XX. SELP, SELL, SKIP, SETL, SETP occur. If the condition being tested is false, a skip is not taken and the next instruction in sequence is executed.

OUTPUT FROM A REGISTER INSTRUCTIONS

There are four variations of the output A instructions: Output A unconditionally causes the contents of A to be

selected onto the S bus, through the adder (SELR and SCRN both off, CINP true), onto the A bus and onto the D bus. This happens during PH3 of ICY2. At the same time, the control line OUTX in the I-O cable goes true (2.0 microseconds). The strobe signal PLSE (500 ns) occurs during the last portion to OUTX, completing the instruction. Output A on sense response

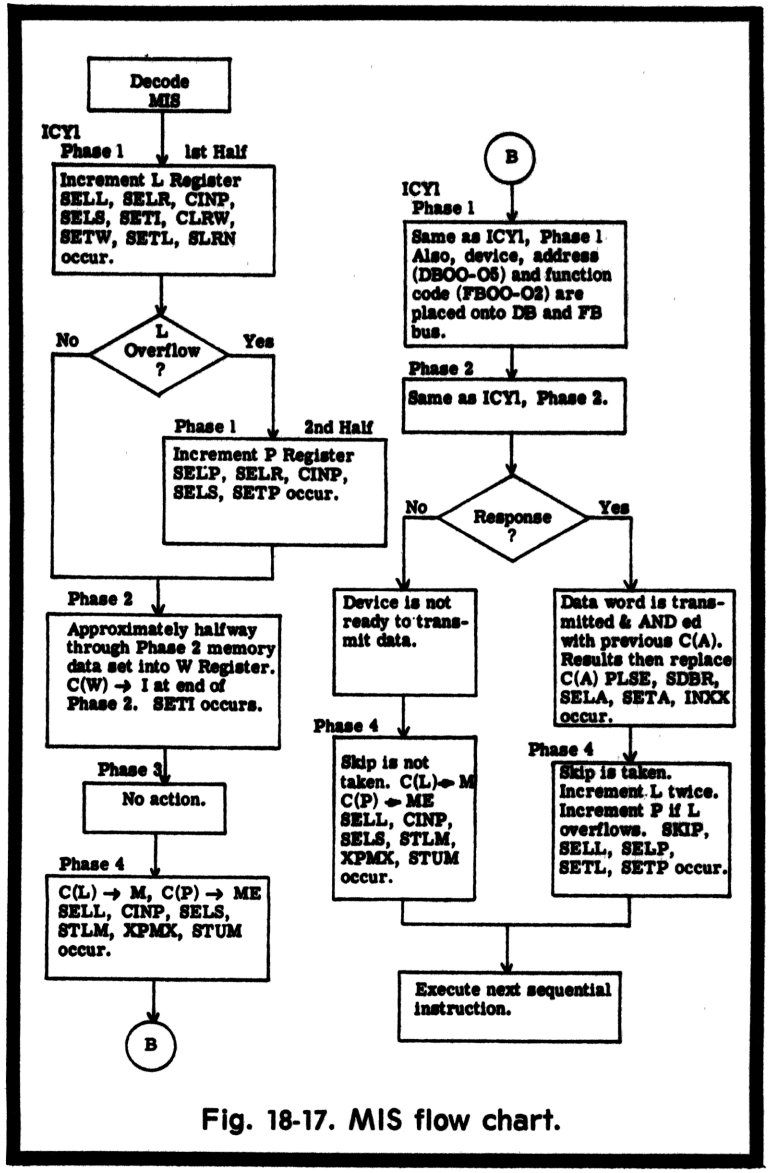


Fig. 18-17. MIS flow chart.

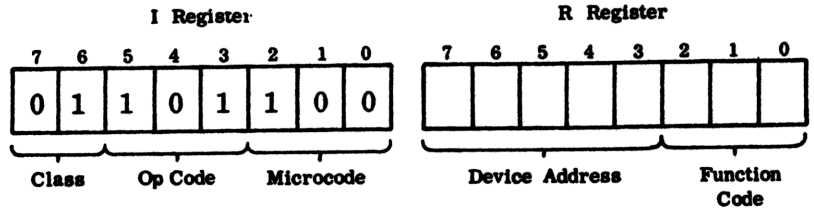


Fig. 18-18. Output A unconditional instruction.

is similar to the unconditional output except PLSE is gated with the sense condition met flip-flop, SCMX. If the sense response from the condition tested is not received within 1.0 microsecond after the beginning of PH3X, PLSE is inhibited and a skip does not occur. If the response is received in time, SCMX, is set, PLSE is allowed to occur and after the transfer, a skip occurs. Output zero unconditionally is identical with output A unconditionally, except the adder is not allowed to be gated onto the A bus.

The processor transfers the data from the A register to the S bus, through the adder to the A bus. The signal SLAI gates the data onto the D bus. The control signals OUTX- and PLSE- are also generated in the processor. These signals, when ANDed with the device address are normally used by the controller to set data into the A register.

OTA: Output A Unconditionally

The OTA instruction in Fig. 18-18 transfers the contents of the A register to the device specified. The A register is not altered. The function code can be used to steer the data into multiple registers if required. For instance, if a 32-bit register is to be loaded by OTAs the register can have four segments and each segment can be specified by the function code. The timing is 2. See Figs. 18-19 and 18-20.

Cycle Sequence: ICY1, ICY2

ICY1 cycle: Phase 1: W register is cleared. L register is incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETI occur. Phase 2: Approximately halfway through phase 2 memory data is set into the W register. The contents of W is transferred

to the I register at the end of the phase. SETI occurs. Phase 3: No action. Phase 4: The contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SELL, CINP, SELS, STLM, XPMX and STUM occur.

ICY2 cycle: Phase 1: Same as ICY1, phase 1 except that the device address (DB00-05) and function code (FB00-02) are placed on the DB and FB bus, respectively. Phase 2: Ap-

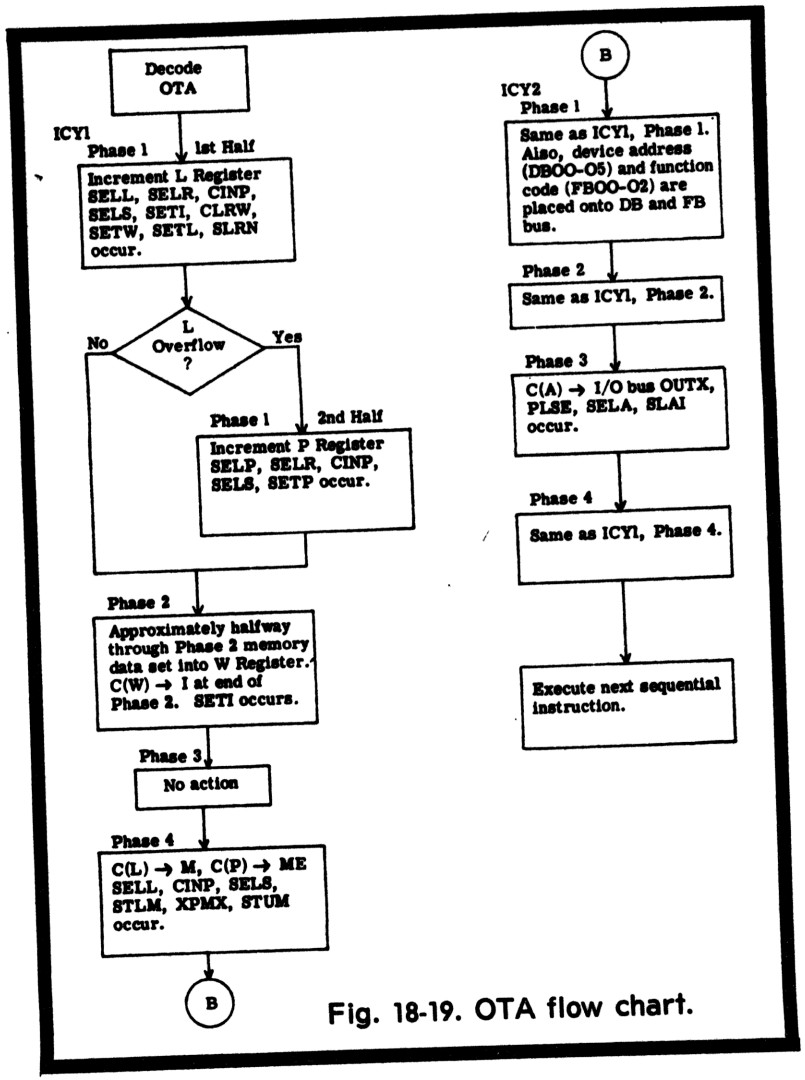


Fig. 18-19. OTA flow chart.

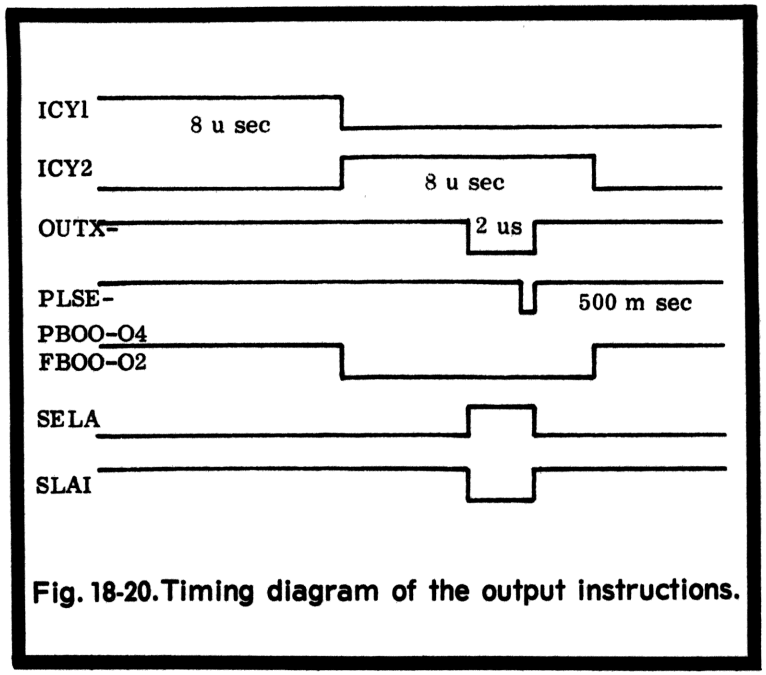


Fig. 18-20. Timing diagram of the output instructions.

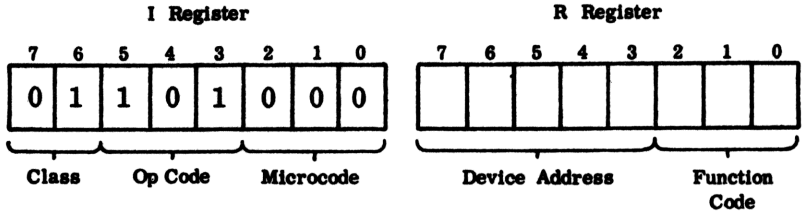


Fig. 18-21. Output zero (unconditionally) instruction.

proximately halfway through phase 2, memory data is set into the W register. C(W) go to I at the end of phase 2. SETI occurs. Phase 3: The signal OUTX, which is 2 microseconds in duration, occurs. This indicates that an output command is being executed. The data word in the A register is placed onto the S bus and routed through the adder. The A bus drivers place the word onto the A bus and the D bus drivers place the data word onto the D bus. The data word is then transmitted to the I-O device via the I-O cable. The signal PLSE- occurs unconditionally since the OTA instruction occurs unconditionally. OUTX-, SELA, SLAI, PLSE- occur. Phase 4: C(L) go to M, C(P) to ME. SELL, CINP, SELS, STLM, XPMX, STUM occur.

OTZ: Output Zero (Uncondtionally)

The OTZ instruction in Fig 18-21 is similar to the OTA, except the contents of the A register is not selected onto the I-O bus, resulting in a transfer of all zeros in the data word. The contents of A is not altered. The timing is 2. See Figs. 18-20 and 18-22.

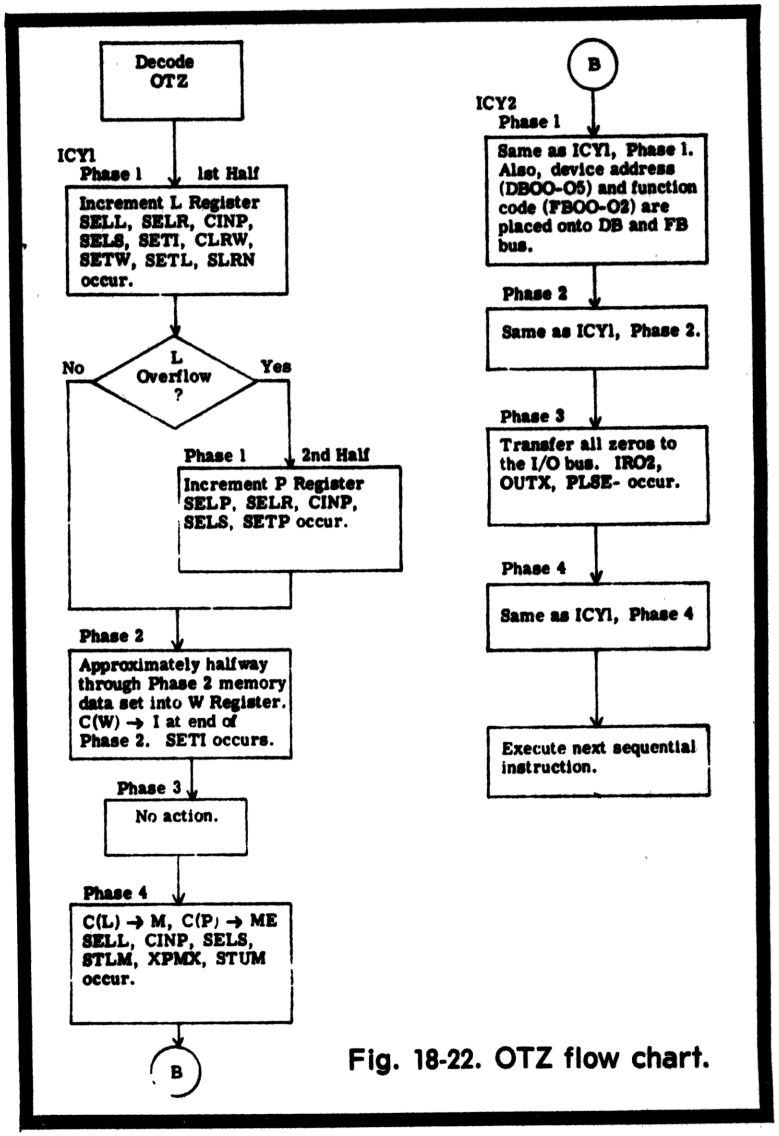


Fig. 18-22. OTZ flow chart.

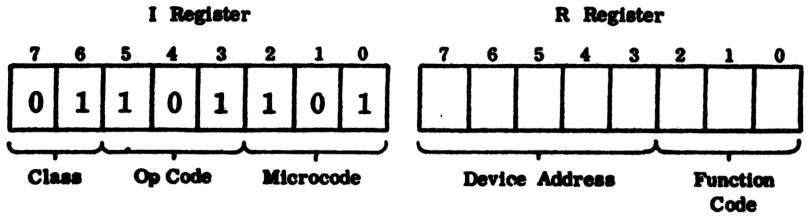


Fig. 18-23. Output A and skip on response instruction.

Cycle Sequence: ICY1, ICY2

ICY1 cycle: Phase 1: W register is cleared. L register is incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur. Phase 2: Approximately halfway through Phase 2 memory data is set into the W register. Contents of W is transferred to the I register at the end of the phase. SETI occurs. Phase 3: No action. Phase 4: Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SELL, CINP, SELS, STLM, XPMX and STUM occur.

ICY2 cycle: Phase 1: Same as ICY1, phase 1, except that the device address (DB00-05) and function code (FB00-02) are placed on the DB and FB bus, respectively. PB00-04, FB00-02 occur. Phase 2: Approximately halfway through phase 2, memory data is set into the W register. C(W) go to I at the end of phase 2. SETI occurs. Phase 3: The C(A) are not selected onto the I-O bus, resulting in a transfer of all zeros in the data word. This is accomplished by IR02 being at ground and SELA not being generated. OUTX-, PLSE-, IR02 occur. Phase 4: C(L) go to M, C(P) to ME. SELL, CINP, SELS, STLM, XPMX, STUM occur.

OAS: Output A and Skip on Response

The OAS instruction in Fig. 18-23 is a combination of the OTA and the sense and skip instructions. The condition specified by the function code is first sensed and if a response is obtained, the contents of A is transferred to the I-O device and a skip occurs. If no sense response occurs, the transfer is not made and the next instruction in sequence is executed. The timing is 2. See Figs. 18-20 and 18-24.

Cycle Sequence: ICY1, ICY2

ICY1 cycle: Phase 1: W register cleared. L register incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur.

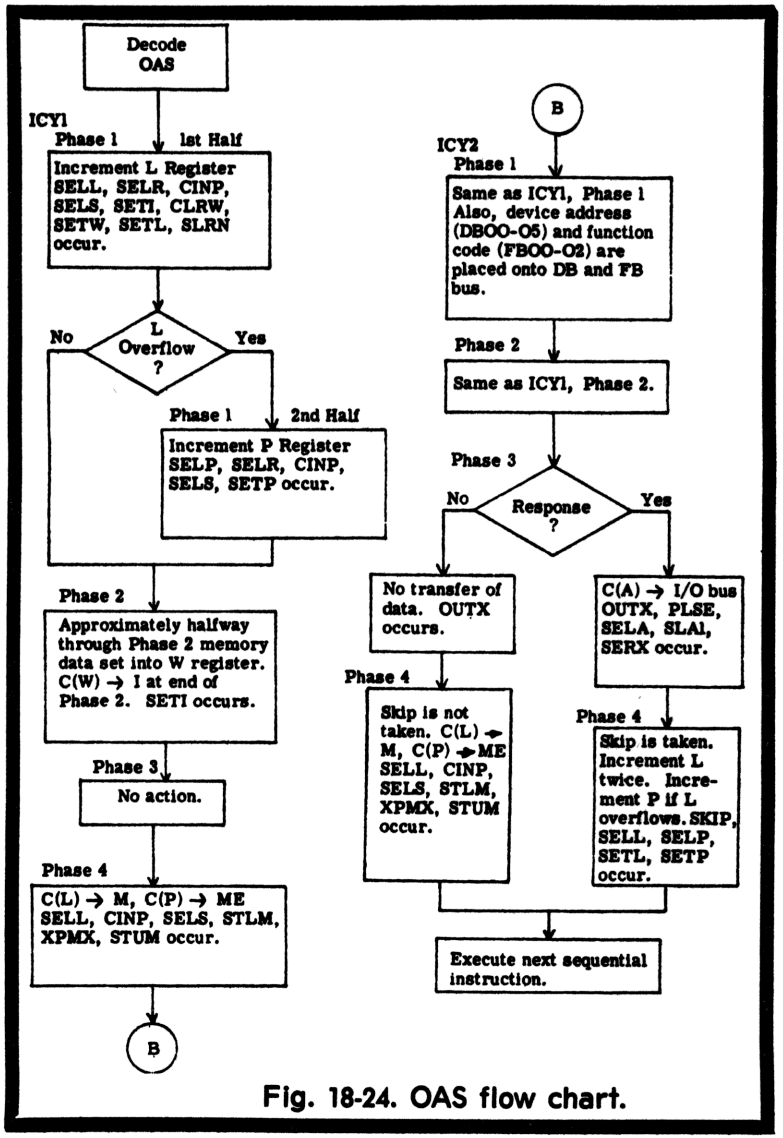


Fig. 18-24. OAS flow chart.

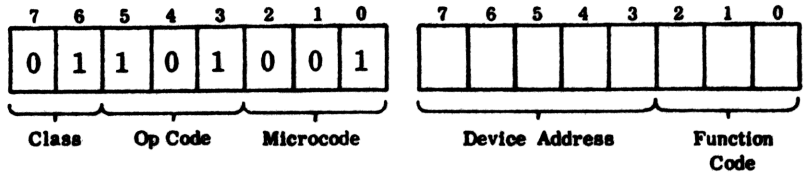


Fig. 18-25. Output zero and skip on response instruction.

Phase 2: Approximately halfway through phase 2 memory data is set into the W register. Contents of W is transferred to the I register at the end of the phase. SETI occurs. Phase 3: No action. Phase 4: Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SELL, CINP, SELS, STLM, XPMX and STUM occur.

ICY2 cycle: Phase 1: Same as ICY1, phase 1, except that the device address (DB00-05) and function code (FB00-02) are placed on the DB and FB bus, respectively. PB00-04, FB00-02 occur. Phase 2: Approximately halfway through phase 2, memory data is set into the W register. C(W) go to I at the end of phase 2. SETI occurs. Phase 3: If the condition being tested is true; i.e., a response occurred, the device is ready to receive data from A. Then SERX- is at ground. Transfer from the A to the I-O bus occurs in the following sequence: OUTX-, PLSE-, SELA, SLAI- and SERX-. If the condition being tested is false; i.e., no response occurred, the device is not ready to receive data, and SERX- is at +3v. The signal PLSE- is inhibited by STA2 being at ground. OUTX- occurs. Phase 4: If the condition being tested is true the skip is taken; i.e., the L register is incremented twice prior to receiving the next instruction. Also, P is incremented if L overflowed. The skip F-F is set at T3XX if IYAX is not on. A skip is not allowed during interrupts. L and P are then used to generate the memory address for the next instruction. Skip F-F is set at T3XX. SELL, SELP, SKIP, SETL, SETP occur. If the condition being tested is false a skip is not taken and the next instruction in sequence is executed.

OZS: Output Zero and Skip on Response

The OZS instruction in Fig. 18-24 is a combination of the OTZ and the sense and skip instructions. The function code specifies the condition to be sensed. If no sense response is obtained, the A register is unaffected and the next instruction in sequence is executed. The timing is 2. See Figs. 10-20 and 18-26.

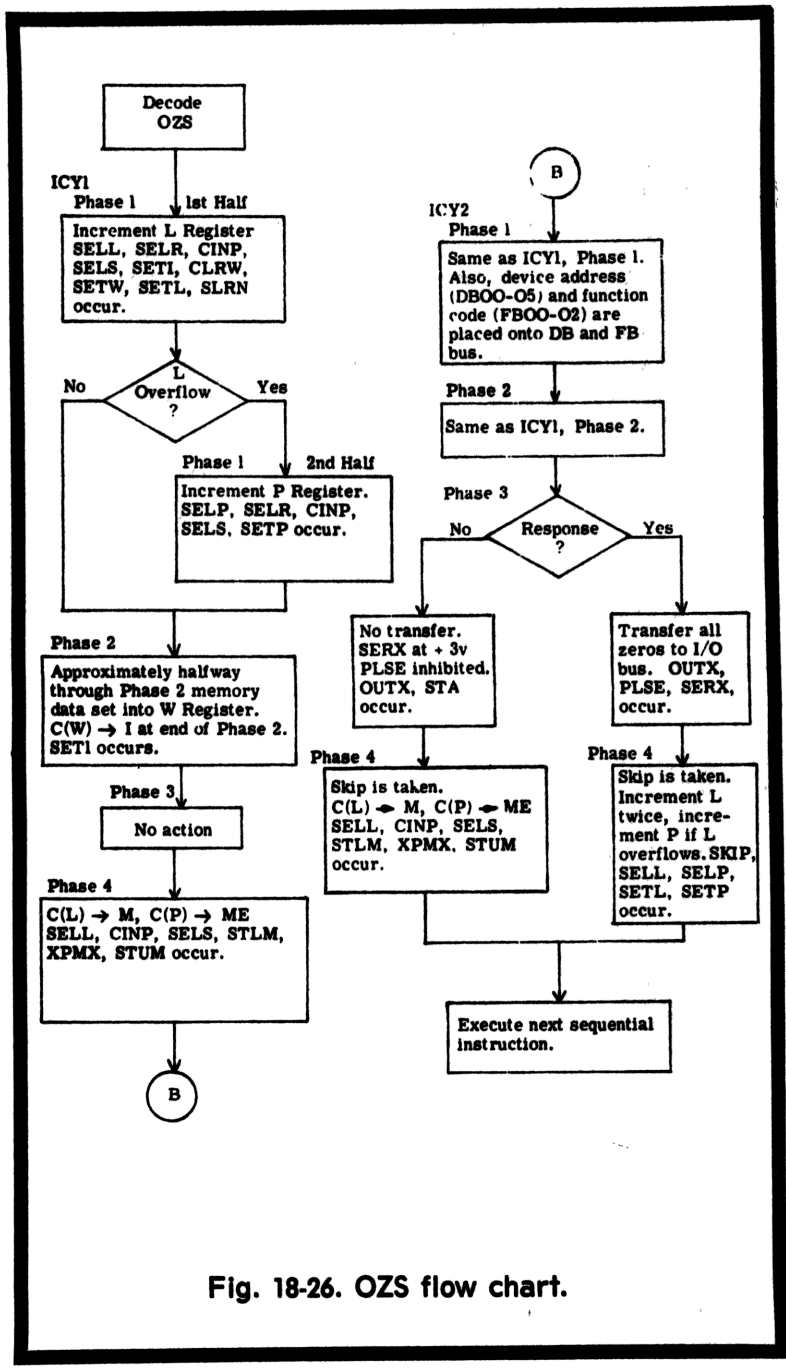


Fig. 18-26. OZS flow chart.

Cycle Sequence: ICY1, ICY2

ICY1 cycle: Phase 1: W register is cleared. L register is incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur. Phase 2: Approximately halfway through phase 2 memory data is set into the W register. Contents of W is transferred to the I register at the end of the phase. SETI occurs. Phase 3: No action. Phase 4: Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SELL, CINP, SELS, STLM, XPMX and STUM occur.

ICY2 cycle: Phase 1: Same as ICY1, phase 1, except that the device address (DB00-05) and function code (FB00-02) are placed on the DB and FB bus, respectively. PB00-04, FB00-02 occur. Phase 2: Approximately halfway through phase 2, memory data is set into the W register. C(W) go to I at the end of phase 2. SETI occurs. Phase 3: If the condition being sensed is true (SERX- is at ground), a transfer of all zeros is accomplished by IROZ being at ground and SELA not generated. OUTX-, PLSE-, SERX- occur. If the condition being tested is false (SERX is at +3v) and the signal PLSE is inhibited by STA2 being at ground. OUTX, STA- occur. Phase 4: If the condition being tested is true the skip is taken; i.e., the L register is incremented twice prior to receiving the next instruction. Also, P is incremented if L overflowed. The skip F-F is set at T3XX if IYAX is not on. A skip is not allowed during interrupts. L and P are then used to generate the memory address for the next instruction. Skip F-F is set at T3XX, SELL, SELP, SKIP, SETL, SETP occur. If the condition being tested is false a skip is not taken and the next instruction in sequence is executed.

BLOCK TRANSFER INSTRUCTIONS

The block input-output instructions allow data transfers to and from memory without disturbing the A register and, when used with interrupts, without disturbing the L and P counters. Each time the instruction is executed, one word is transferred. When the last word of the block is transferred, notification is made either by a skip or an echo pulse to the transferring peripheral.

INB: Input Block to Memory

The instruction in Fig. 18-27 transfers a word from the specified device to memory at the location specified by the input pointers +1 and the pointer extension. If INB is executed as an interrupt instruction, an echo pulse is generated when the end-of-block occurs which can be used to create an end-of-block interrupt. (No skip can occur during an interrupt instruction execution). If INB is executed in a program (i.e., not as an interrupt instruction), a skip will occur at end-of-block. The timing is 4. See Fig. 18-28.

Cycle Sequence: ICY1, ICY2, ACY1, OCY1

ICY1 cycle: Phase 1: W register is cleared. L register is incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur. Phase 2: Approximately halfway through phase 2 memory data is set into the W register. Contents of W is transferred to the I register at the end of the phase. SETI occurs. Phase 3: No action. Phase 4: Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SELL, CINP, SELS, STLM, XPMX and STUM occur.

ICY2 cycle: Phase 1: W register is cleared. L register is incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur. Phase 2: Approximately halfway through phase 2 memory data is set into the W register. Contents of W is transferred to the I register at the end of the phase. SETI occurs. Phase 3: No action. Phase 4: Form an address to receive the pointer; for the input pointer, location 6, page 0. Turn off the SELS gate;

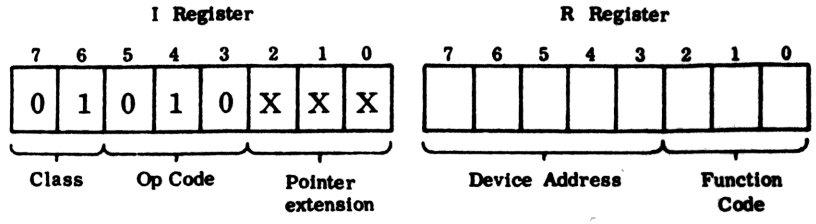


Fig. 18-27. Input block to memory instruction.

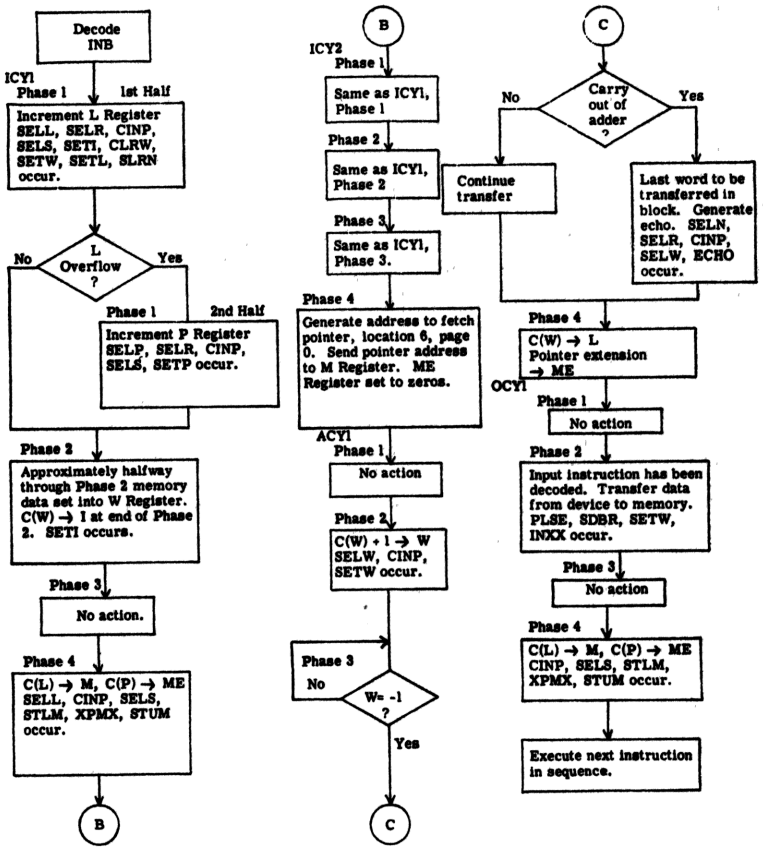


Fig. 18-28. INB flow chart.

send the pointer address to the M register. ME register is set to zero.

ACY1 cycle: Phase 1: No action. Phase 2: C(W) +1 go to W, index pointer. SELW, CINP, SETW occur. Phase 3: Check for W -1. No carry out indicates there still are words to be transferred. A carryout indicates that this is the last word to be transferred in the block. In this case, generate an ECHO. SELR, SELN, CINP, SELW, ECHO occur. Phase 4: C(W) go to L and the pointer extension to ME.

OCY1 cycle: Phase 1: No action. Phase 2: INXX signal indicates that an input instruction has been decoded. PLSE, SDBR, SETW occur. Phase 3: No action. Phase 4: C(L) go to M, C(P) to ME. SELL, CINP, SELS, STLM, XPMX, STUM occur.

OTB: Output Block from Memory

The instruction in Fig. 18-29 transfers a word from memory at the address specified by the output pointer +1 and the pointer extension to the specified device. If OTB is executed as an interrupt instruction, an echo pulse is generated when the end-of-block occurs, which can be used to create an end-of-block interrupt. (No skip can occur during an interrupt instruction execution). If OTB is executed in a program (i.e., not as an interrupt instruction), a skip will occur at end-of-block. The timing is 4. See Fig. 18-30.

Cycle Sequence: ICY1, ICY2, ACY1, OCY1

ICY1 cycle: Phase 1: W register is cleared. L register is incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur. Phase 2: Approximately halfway through phase 2 memory data is set into the W register. Contents of W is transferred to the I register at the end of the phase. SETI occurs. Phase 3: No action. Phase 4: Contents of the L register is transferred to the M register and the contents of the P register is transferred to the M extension register. SELL, CINP, SELS, STLM, XPMX and STUM occur.

ICY2 cycle: Phase 1: W register is cleared. L register is incremented during the first half of phase 1. SELL, SELR, SLRN, CINP, SELS, SETI, CLRW, SETW occur. If L overflowed the P register is incremented during the second half of phase 1. SELP, SELR, SLRN, CINP, SELS and SETP occur. Phase 2: Approximately halfway through phase 2 memory data is set into the W register. Contents of the W is transferred to the I register at the end of the phase. SETI occurs. Phase 3:

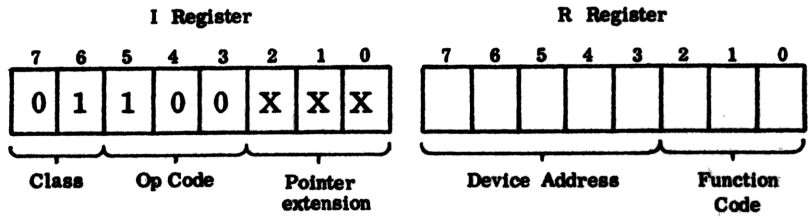


Fig. 18-29. Output block from memory instruction.

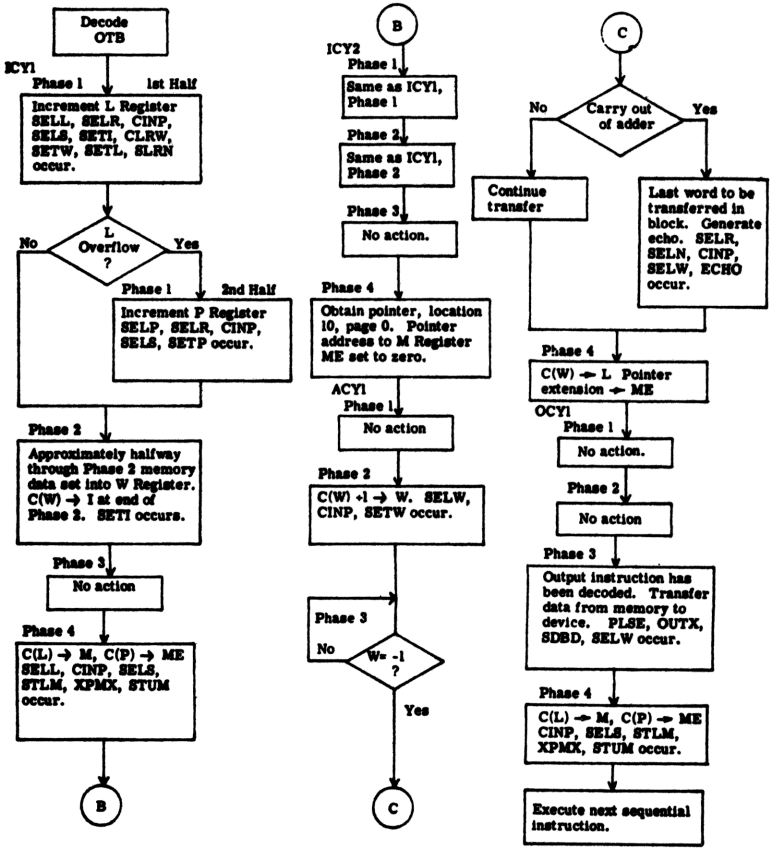


Fig. 18-30. OTB flow chart.

No action. Phase 4: Form the address to obtain the pointer. For the output pointer go to location 10, page 0. Turn off SELS gate. The pointer address goes to the M register. ME is set to zeros.

ACY1 cycle: Phase 1: No action. Phase 2: W +1 goes to W, index pointer. SELW, CINP, SETW occur. Phase 3: Check for

W equal to -1. No carry out indicates there are still words to be transferred. A carry out indicates that this is the last word to be transferred in the block. This causes an echo to be generated. SELR, SELN, CINP, SELW, ECHO occur. Phase 4: C(W) go to L and the pointer extension to ME.

OCY1 cycle: Phase 1: No action. Phase 2: No action. Phase 3: The signal OUTX indicates that an output instruction has been decoded. Phase 4: C(L) go to M, C(P) to ME. SELL, CINP, SELS, STLM, XMPX, STUM occur.

Chapter 19

Diagnostic Information

PROCESSOR MEMORY DIAGNOSTIC (PMD)

PMD tests every core of the memory to ensure that no bits are "picked up" or "dropped." Address logic is checked by storing every word address in itself. Words are read twice to check the read-restore memory cycle. Error diagnostics are typed on the teletype.

Memory Tests

The following three tests are run once for each cycle. Test 1 fills the memory with 0s then reads them back to check for "picked up" bits. Tests 2 fills the memory with all 1s then reads them back to check for "dropped" bits. Test 3 stores the address of each word of the memory in itself then reads them back to verify the memory address circuits.

Each of the three tests is run first for upper memory (page 10 and above) and then for lower memory (pages 0-7). Since this diagnostic destroys all of the memory a binary loader is stored at 350_8 to facilitate reloading the system's software.

Diagnostic Printouts

Error printouts appear in the following format, where all outputs are printed as four octal numbers; **ID, Location, Page** and Error bits.

Setup Procedures

1. Load the PMD in the memory using the object loader.
2. Place the step-run switch in the step mode.
3. Depress the A register select switch (A), depress the clear switch, and enter on the data switches one of the following numbers to indicate the size of the memory.
4. Set the sense switch to run the test continuously.
5. Place the step-run switch in the run mode.
6. Depress the cycle switch.
7. PMD will test the memory module indicated by the sense switch. To exercise the alternate memory module merely change the sense switch, the program will relocate itself and test the alternate memory bank.


```
0001  000 000                *
0002  000 000                *  7/30/68
0003  000 000                *
0004  000 000                *
0005  000 000                ***  PMD  PROCESSOR MEMORY DIAGNOSTIC
0006  000 000                *
0007  000 000                *    PMD TESTS MEMORY FOR 'PICKING'
0008  000 000                *    AND 'DROPPING' OF BITS BY FILLING
0009  000 000                *    CORE WITH ZEROES AND ONES AND
0010  000 000                *    THEN READING MEMORY BACK FOR
0011  000 000                *    VERIFICATION.  ADDRESS LOGIC IS
0012  000 000                *    CHECKED BY STORING THE ADDRESS OF
0013  000 000                *    EACH WORD IN ITSELF.  WORDS ARE
0014  000 000                *    READ TWICE TO CHECK THE READ/RESTORE
0015  000 000                *    MEMORY CYCLE.  EACH OF THE 3 TESTS
0016  000 000                *    IS RUN FIRST FOR UPPER MEMORY AND
0017  000 000                *    THEN FOR LOWER MEMORY (PAGES 0-10).
0018  000 000                *    SINCE THIS DIAGNOSTIC DESTROYS ALL
0019  000 000                *    OF MEMORY A BINARY LOADER IS STORED
0020  000 000                *    AT 0350 TO FACILITATE RE-LOADING
0021  000 000                *    THE SYSTEM SOFTWARE.
0022  000 000                *
0023  000 000                *    TO OPERATE
0024  000 000                *
0025  000 000                *    1) LOAD DIAGNOSTIC AT PAGE 0
0026  000 000                *
0027  000 000                *    2) SET SENSE SWITCH
0028  000 000                *       ON = TEST CONTINUOUSLY
0029  000 000                *       OFF= HALT AT END OF TEST
0030  000 000                *
0031  000 000                *    3) SET A REG TO CORE SIZE AND RUN
0032  000 000                *
0033  000 000                *         4K = 020
0034  000 000                *         8K = 040
0035  000 000                *        12K = 060
0036  000 000                *        16K = 100
0037  000 000                *
0038  000 000                *    ERROR DIAGNOSTIC MESSAGESARE
0039  000 000                *    PRINTED AS 4 OCTAL NUMBERS; ID,
0040  000 000                *    LOC, PAGE AND ERROR BITS.
0041  000 000                *
0042  000 000                *    ID = OAB  WHERE
0043  000 000                *    WHERE A = 0  ERROR AT READ TIME
0044  000 000                *          A = 1  ERROR AT STORE TIME
0045  000 000                *          B = 0  ZERO TEST
0046  000 000                *          B = 1  ONES TEST
0047  000 000                *          B = 2  ADDR TEST
0048  000 000                *
0049  000 000                ***  ITEMS AND POINTERS **
0050  000 000                *
0051  000 000  000 000           ORG  0
0052  000 000  000               DEC  0       ZERO TEST WORD
0053  001 000  377               DEC  -1      ONES TEST WORD
0054  002 000  000          LOC  PTR  **      ADDR TEST WORD AND
0055  003 000  000          PAG  PTR  **      STORAGE PTR
0056  004 000  020          MEM  PTR  @20     CORE SIZE (4K STANDARD)
0057  005 000  000          SAV  PTR  **      TEMP FOR TYP
0058  006 000  000 000      CHK  DPT  **      TEST WORD PTR
0059  010 000                *
0060  010 000                ***  MAIN PROGRAM  ***
0061  010 000                *
0062  010 000  315 004      PMD  LDA  MEM     GET CORE SIZE
0063  012 000  040          040  HLT          OPERATOR MAY CHANGE IT
0064  013 000  331 004           STA  MEM     SET CORE SIZE
0065  015 000  005          PM1  ARC  Z       TEST UPPER MEMORY
0066  016 000  331 007           STA  CHK+1   CHECK WORD IN PAGE 0
0067  020 000  017 000      07   EQU  *-1     (ADDR = 07)
0068  020 000  315 047           LDA  010
0069  022 000  331 323           STA  BOM     BEGIN TEST AT PAGE 10
0070  024 000  315 004           LDA  MEM
```

```
0071  026 000   331 322        STA  EOM    END TEST AT END OF MEMORY
0072  030 000   375 063  M3    JST  TST    TEST UPPER MEMORY
0073  032 000   331 323        STA  BOM    BEGIN TEST AT PAGE 0
0074  034 000   315 047        LDA  010
0075  036 000   331 322        STA  EOM    END TEST AT PAGE 10
0076  040 000   331 007        STA  CHK+1  CHECK WORD IN PAGE 10
0077  042 000   375 230  CTR   JST  MOV    MOVE PGM TO PAGE 10
0078  044 000   347 046       *JMP  *+1    GO TO PAGE 10
0079  046 000   050            PTR  PM2    PTR TO PAGE 10
0080  047 000   010       010  OCT  10
0081  050 000   375 063  PM2   JST  TST    TEST LOWER MEMORY
0082  052 000   375 230        JST  MOV    MOVE PGM TO PAGE 0
0083  054 000   344 056        JMP /PM3    GO TO PAGE 0
0084  056 000   250       PM3  IFT  1      REPEAT TEST?
0085  057 000   345 010        JMP  PMD    NO, GO HALT FOR OPERATOR
0086  061 000   345 015        JMP  PM1    YES REPEAT
0087  063 000   062 000  015   EQU  *-1    (ADDR = 015)
0088  063 000                 *
0089  063 000                 ***  TST  TEST SUBROUTINE
0090  063 000                 *
0091  063 000   000 000  TST   DPT  **
0092  065 000   005            ARC  Z      INITIALIZE
0093  066 000   331 002        STA  LOC    MEMORY WORD PTR
0094  070 000   331 006        STA  CHK    TEST WORD PTR
0095  072 000   001       TS1  ARC  -1
0096  073 000   331 231        STA  FLG    RESET STORE FLAG
0097  075 000   315 323  TS2   LDA  BOM    INITIALIZE
0098  077 000   331 003        STA  PAG    MEMORY WORD PTR
0099  101 000   315 231  TS3   LDA  FLG
0100  103 000   045            SHF  (      OV = STORE FLAG
0101  104 000   317 006       *LDA  CHK    GET TEST WORD
0102  106 000   244             IFT  R      STORING?
0103  107 000   333 002       *STA  LOC    YES, STORE TEST WORD
0104  111 000   317 002       *LDA  LOC    READ MEMORY
0105  113 000   331 230        STA  TMP    DEVELOP EXCLUSIVE OR
0106  115 000   307 006       *IOR  CHK    OF MEMORY AND TEST WORD
0107  117 000   335 230        IMA  TMP
0108  121 000   323 006       *ANA  CHK
0109  123 000   011             ARC  C
0110  124 000   321 230        ANA  TMP
0111  126 000   242             IFT  N      MEMORY = TEST WORD?
0112  127 000   345 170        JMP  TS4    YES, INCR MEM PTR
0113  131 000   331 230        STA  TMP    NO, SAVE ERROR BITS
0114  133 000   315 231        LDA  FLG    GET STORE/TEST FLAG
0115  135 000   202             IFT  Z      TESTING?
0116  136 000   315 047        LDA  010    NO, SET STORE IND
0117  140 000   301 006        ADD  CHK    ADD TEST NO.
0118  142 000   375 254  CNT   JST  TYP    TYPE TEST ID
0119  144 000   315 002        LDA  LOC
0120  146 000   375 254        JST  TYP    TYPE LOC
0121  150 000   315 003        LDA  PAG
0122  152 000   375 254        JST  TYP    TYPE PAGE
0123  154 000   315 230        LDA  TMP
0124  156 000   375 254        JST  TYP    TYPE ERROR BITS
0125  160 000   315 062        LDA  015
0126  162 000   375 312        JST  OTY    CR
0127  164 000   301 030        ADD  M3
0128  166 000   375 312        JST  OTY    LF
0129  170 000   371 002  TS4   IMS  LOC    BUMP PTR
0130  172 000   345 101        JMP  TS3
0131  174 000   371 003        IMS  PAG    BUMP PAGE
0132  176 000   315 003        LDA  PAG
0133  200 000   311 322        SUB  EOM
0134  202 000   202             IFT  Z      END OF MEMORY?
0135  203 000   345 101        JMP  TS3    NO, CONTINUE
0136  205 000   371 231        IMS  FLG    END OF TEST
0137  207 000   345 213        JMP  *+3
0138  211 000   345 075        JMP  TS2    NO, GO TEST
0139  213 000   371 006        IMS  CHK    YES, BUMP CHK WORD PTR
0140  215 000   371 042        IMS  CTR    3RD TEST?
```



```
0141  217 000   345 072        JMP  TS1    NO, NEXT TEST
0142  221 000   315 030        LDA  M3     YES RESET CTR
0143  223 000   331 042        STA  CTR
0144  225 000   005            ARC  Z
0145  226 000   347 063       *JMP  TST    RETURN
0146  230 000                 *
0147  230 000                 ***  MOV  MOVE PGM TO ANOTHER PAGE  ***
0148  230 000                 *
0149  230 000   000 000  MOV   DPT  **
0150  232 000   331 003        STA  PAG    SET DESTINATION PAGE
0151  234 000   001            ARC  -1
0152  235 000   331 243        STA  PTR    SET PTR
0153  237 000   005            ARC  Z
0154  240 000   371 243  MO1   IMS  PTR    BUMP PTR
0155  242 000   315 000        LDA  **     GET PGM WORD
0156  244 000   243 000  PTR   EQU  *-1
0157  244 000   333 002       *STA  LOC    MOVE TO OTHER PAGE
0158  246 000   371 002        IMS  LOC    END OF PAGE?
0159  250 000   345 240        JMP  MO1    NO LOOP
0160  252 000   347 230       *JMP  MOV    YES, RETURN
0161  254 000                 *
0162  254 000                 ***  TELETYPE OUTPUT SUBROUTINES  ***
0163  254 000                 *
0164  254 000                 *    TYP  TYPE OCTAL WORD AND SPACE
0165  254 000                 *
0166  254 000   000 000  TYP   DPT  **
0167  256 000   050            ORC  R      BIT 6 = 0
0168  257 000   045       TY1  SHF  (      NEXT 3 BITS
0169  260 000   045            SHF  (
0170  261 000   045            SHF  (
0171  262 000   331 005        STA  SAV    SAVE BITS
0172  264 000   321 017        ANA  07     KEEP LO 3 BITS
0173  266 000   301 324        ADD  060    MAKE ASCII
0174  270 000   375 312        JST  OTY    TYPE DIGIT
0175  272 000   315 005        LDA  SAV    GET BITS
0176  274 000   371 142        IMS  CNT    3RD DIGIT
0177  276 000   345 257        JMP  TY1    NO, LOOP
0178  300 000   315 030        LDA  M3     YES, RESET CTR
0179  302 000   331 142        STA  CNT
0180  304 000   315 012        LDA  040
0181  306 000   375 312        JST  OTY    TYPE SPACE
0182  310 000   347 254       *JMP  TYP
0183  312 000                 *
0184  312 000                 *    OTY  OUTPUT TO TELETYPE
0185  312 000                 *
0186  312 000   000 000  OTY   DPT  **
0187  314 000   155 073        OAS  073    OUTPUT IF BFR RDY
0188  316 000   345 314        JMP  *-3    BFR NOT RDY
0189  320 000   347 312       *JMP  OTY
0190  322 000                 *
0191  322 000                 *    ITEMS
0192  322 000                 *
0193  322 000   230 000  TMP   EQU  MOV    USE ENTRY AS TEMP STORE
0194  322 000   231 000  FLG   EQU  MOV+1  AND FLAG
0195  322 000   000       EOM  PTR  **     END PAGE OF TEST
0196  323 000   000       BOM  PTR  **     BEGINNING PAGE OF TEST
0197  324 000   060       060  OCT  60
0198  325 000                 *
0199  325 000                 *    BINARY LOADER
0200  325 000                 *
0201  350 000   350 000            ORG  0350
0202  350 000   331 002  PBL   STA  LOC    SAVE LOC PART OF ADR
0203  352 000   331 003        STA  PAG    SAVE PAGE PART OF ADR
0204  354 000   100 072  BL1   EXC  072    SEL STEP READ PAPER TAPE
0205  356 000   131 071        IAS  071
0206  360 000   345 356        JMP  *-3    TTY NOT READY
0207  362 000   333 002       *STA  LOC    PUT CHAR IN NEXT PROG WD
0208  364 000   371 002        IMS  LOC    GET ADR OF NEXT PROG WD
```

```
0209  366 000   345 354       JMP  BL1    GO GET NEXT CHAR
0210  370 000   371 003       IMS  PAG    END OF PAGE. INC PAGE NO.
0211  372 000   210           IFT  0      IS LOADING TO BE HALTED
0212  373 000   040           HLT         YES HALT
0213  374 000   000           NOP
0214  375 000   345 354       JMP  BL1    GO LOAD NEXT PAGE
0215  377 000   010 000       END  PMD    (LOAD & EXECUTE FOR 4K CO
```

WORST PATTERN DIAGNOSTIC(WPD)

WPD occupies the first 85 words in page 0 and fills the remainder of the memory with a worst-case pattern of 0s and 1s. This causes most of the cores in the memory to be magnetized in the same direction. This pattern is then read out and verified under the worst-case noise level for memory. Since all of the memory is destroyed this should either be the first or last diagnostic exercised, since all loaders and boot-straps must be reloaded.

Setup Procedures

1. Load the WPD in the memory using the object loader.
2. Place the step-run switch in the step mode.
3. Depress the A register select switch (A), depress the clear switch, and enter on the data switches one of the numbers below to indicate the size of the memory. If running a 4k test, leave the A register as set (2).

$$40_8 = 8k$$

$$60_8 = 12k$$

$$100_8 = 16k$$

4. Set the sense switch to run the test continuously.
5. Place the step-run switch in the run mode.
6. Depress the cycle switch.
7. The memory will be filled with the worst pattern and verified. Errors cause a halt at 125-0 with OV equal to 1. The A register contains a 1 in the bit that failed. The address of the bad word is stored at location 10 of page 0. Press the cycle switch to continue the test.

```
0001  000 000             *
0002  000 000             ***  WPD  WORST PATTERN MEMORY DIAGNOSTIC
0003  000 000             *
0004  000 000             *    WPD RESIDES IN PAGE 0 AND FILLS THE
0005  000 000             *    REMAINDER OF MEMORY WITH THE WORST
0006  000 000             *    CASE PATTERN OF ZEROES AND ONES
0007  000 000             *    TESTING MEMORY UNDER THE HIGHEST
0008  000 000             *    NOSE LEVEL.
0009  000 000             *
0010  000 000             *    TO OPERATE
```



```
0011  000 000             *
0012  000 000             *    1) LOAD DIAGNOSTIC AT PAGE 0
0013  000 000             *    2) SET SENSE SWITCH
0014  000 000             *       ON = TEST CONTINUOUSLY
0015  000 000             *       OFF= HALT AT END OF TEST
0016  000 000             *
0017  000 000             *    3) SET A REG TO CORE SIZE AND RUN
0018  000 000             *
0019  000 000             *         4K = 020
0020  000 000             *         8K = 040
0021  000 000             *        12K = 060
0022  000 000             *        16K = 100
0023  000 000             *
0024  000 000             *    SYSTEM HALTS
0025  000 000             *
0026  000 000             *       OV OFF = NORMAL END OF TEST
0027  000 000             *       OV ON =  ERROR HALT, WHERE
0028  000 000             *          A REG CONTAINS ERROR BITS
0029  000 000             *          LOC 10-0 = ADDR OF ERROR WD
0030  000 000             *          (PRESS RUN TO CONTINUE TEST)
0031  000 000             *
0032  000 000  000 000          ORG  0
0033  000 000  377              DEC  -1     WORST PATTERN TABLE
0034  001 000  000              DEC  0
0035  002 000  000              DEC  0
0036  003 000  377              DEC  -1
0037  004 000  000              DEC  0
0038  005 000  377              DEC  -1
0039  006 000  377              DEC  -1
0040  007 000  000              DEC  0
0041  010 000             *
0042  010 000             *    ITEMS AND POINTERS
0043  010 000             *
0044  010 000  130         LOC  PTR  FLG    MEMORY ADDR PTR (LOC)
0045  011 000  000         PAG  PTR  0
0046  012 000  020         CSZ  OCT  20     CORE SIZE (4K STANDARD)
0047  013 000             *
0048  013 000             *    ENTRY TO DIAGNOSTIC
0049  013 000             *
0050  013 000  315 012     WPD  LDA  CSZ    GET CORE SIZE
0051  015 000  050              ORC  R      SET END OF TEST FLAG
0052  016 000  040              HLT         LET OPERATOR CHANGE SIZE
0053  017 000  331 012         STA  CSZ    SET MEMORY SIZE
0054  021 000  005         WP1  ARC  Z
0055  022 000  331 130         STA  FLG    SET STORE FLAG
0056  024 000  331 011     WP2  STA  PAG    SET PAG PTR TO 0
0057  026 000  315 023         LDA  SLC
0058  030 000  331 010         STA  LOC    SET LOC PTR TO END OF PGM
0059  032 000  371 010     WP3  IMS  LOC    BUMP LOC PTR
0060  034 000  345 063         JMP  WP4
0061  036 000  371 011         IMS  PAG    BUMP PAGE PTR
0062  040 000  315 011         LDA  PAG
0063  042 000  311 012         SUB  CSZ
0064  044 000  202             IFT  Z      END OF CORE?
0065  045 000  345 063         JMP  WP4    NO, GO STORE/TEST
0066  047 000  315 130         LDA  FLG    GET STORE FLAG
0067  051 000  371 130         IMS  FLG    BUMP STORE FLAG
0068  053 000  242             IFT  N      END OF TEST?
0069  054 000  345 024         JMP  WP2    NO, GO TEST
0070  056 000  250             IFT  1      REPEAT REST?
0071  057 000  345 013         JMP  WPD    NO, GO HALT
0072  061 000  345 021         JMP  WP1    YES, REPEAT
0073  063 000  315 010     WP4  LDA  LOC    USE LOC TO GET TBL ADDR
0074  065 000  055             SHF  L      7 6543210Z
0075  066 000  055             SHF  L      6 543210ZZ
0076  067 000  041             SHF  <      6 53210ZZZ
0077  070 000  041             SHF  <      6 5210ZZZZ
0078  071 000  041             SHF  <      6 510ZZZZZ
0079  072 000  045             SHF  (      5 10ZZZZZ6
0080  073 000  045             SHF  (      1 0ZZZZZ65
```

```
0081  074 000   045          SHF  (      0 ZZZZZ651
0082  075 000   331 100      STA  *+2    SET TBLE ADDR IN LOAD INS
0083  077 000   315 000      LDA  **     GET WORST PATTERN
0084  101 000   202          IFT  Z
0085  102 000   070          ORC  C
0086  103 000   000          NOP
0087  104 000   315 130      LDA  FLG    GET CYCLE FLAG
0088  106 000   202          IFT  Z      STORING?
0089  107 000   345 116      JMP  WP5    NO, GO CHECK
0090  111 000   244          IFT  R      STORE ZEROES?
0091  112 000   001          ARC  -1     NO, SET ALL ONES
0092  113 000   060          ORC  S      SET ONE'S FLAG
0093  114 000   333 010      *STA LOC    STORE PATTERN
0094  116 000   317 010  WP5 *LDA LOC    GET MEM WORD
0095  120 000   244          IFT  R      SHOULD PATTERN BE ZEROES?
0096  121 000   011          ARC  C      NO, COMPLEMENT MEM WORD
0097  122 000   000          NOP
0098  123 000   202          IFT  Z      MEM WORD OK?
0099  124 000   060          ORC  S      NO, SET ERROR IND
0100  125 000   040          HLT         ERROR HALT
0101  126 000   345 032      JMP  WP3    YES, GO CHECK END OF TEST
0102  130 000   000      FLG OCT  0      STORE/CHECK CYCLE FLAG
0103  131 000   023 000  SLC EQU  WP2-1  (END OF PGM)
0104  131 000   013 000      END  WPD
```

PROCESSOR TELETYPE DIAGNOSTIC (PTD)

PTD tests all teletype I-O logic and checks the reader, punch, and printer for every character code ($0\text{-}377_8$). Also, the keyboard input is checked for any character the user may desire to test.

Sequence of Testing

First the initialize teletype command (EXC @ 74) is checked along with the different TTY sense instructions. Then if the sense switch is depressed, the diagnostic will output the leader and the CR-LF (carriage return and line feed). Leader and CR-LF will be punched out following each set of characters output. There are 64 characters output in each set.

The first set of characters output will be $0\text{-}77_8$. This will be followed by an input sequence which reads the characters ($0\text{-}77_8$) just output. Then these characters input are compared against those that were output. The comparison starts on the first character after the carriage return input. If any discrepancies exist, the processor will halt. At this point the A register will contain the input character which failed the test. (After any halt if the cycle switch is depressed twice, the diagnostic should proceed.) The sequence of output characters, output leader and input is repeated four times per pass with this diagnostic.

The following I-O instructions are used on these sequences:

Set	Character Codes	Output	Input
1	0-77 or 200-277	Output A reg.	Block-input
2	0-77 or 200-277	Block-output	Input-A reg. on sense
3	100-77 or 300-377	Output-A reg. on sense	Block-input (interrupt)
4	100-77 or 300-377	Block-output (interrupt)	Masked input

(Note: On every other pass the character codes are offset by 200_8.)

These sequences are run over and over until the sense switch is depressed. When the sense switch is depressed the diagnostic will not start another pass, but instead will "hang" in a loop waiting for an input from the keyboard. The keyboard test will just output each character that is input, timing out each input so that no garbage should be printed out.

Setup Procedures

1. Load PTD in the memory using the object loader.
2. Place the step-run switch in the step mode.
3. Make sure the relative and manual execute switches are released.
4. Depress the reset switch.
5. Depress the page register select switch (P), depress the clear switch, and enter on the data switches the page number in the memory where the teletype diagnostic starts (5).
6. Depress the location register select switch (L), depress the clear switch.
7. Depress the instruction register select switch (I), depress the clear switch.
8. Depress the A register select switch (A), depress the clear switch.
9. Punch the leader with the teletype on local and enter the tape (which is hanging from the punch) in the paper tape reader so that there is nothing but the leader between the reader and the punch. If the keyboard is to be tested, proceed to the next step. If the diagnostic is to be run continuously (I-O instruction diagnostic plus punch and reader diagnostic) depress the sense switch.
10. Place the TTY power on switch on the teletype in the line position.

11. Make sure the paper tape punch is on and the tape reader is on.

12. Place the step-run switch in the run mode.

13. Depress the cycle switch.

14. If the diagnostic halts, the A register will tell what input character failed. To determine what test failed and how, look at the attached listing. If it is desired to have the test run repeatedly, store a jump instruction (to the beginning of the test) in place of the two halts.

15. When the sense switch is depressed, the diagnostic will "hang," looking for an input from the keyboard.

16. If it is desired to continue after any halt, depress the cycle switch twice.

```
0001  000 000              *
0002  000 000              *  07/20/68
0003  000 000              *
0004  000 000              *
0005  000 000              *    PROCESSOR TELETYPE DIAGNOSTIC (PTD)
0006  000 000              *
0007  000 000              *
0008  000 005  000 005          ORG  0-5
0009  000 005  100 074  SAR    EXC  @74
0010  002 005  150 070         OTZ  @70     OUTPUT ZERO
0011  004 005  100 074         EXC  @74
0012  006 005  111 071         SSR  @71
0013  010 005  040             HLT          READY FAILED
0014  011 005  040             HLT
0015  012 005  110 073         SSN  @73     NOT BUSY FAILED
0016  014 005  345 020         JMP  *+3
0017  016 005  040             HLT
0018  017 005  040             HLT
0019  020 005  111 072         SSR  @72     INT #1 OFF FAILED
0020  022 005  345 026         JMP  *+3
0021  024 005  040             HLT
0022  025 005  040             HLT
0023  026 005  374 216         JST  /SUP
0024  030 005  005             ARC  Z
0025  031 005  330 176         STA  /INT
0026  033 005  331 234         STA  OCH
0027  035 005  374 140         JST  /LDR
0028  037 005  315 074  SPT    LDA  BIN     SET UP INPUT BLOCK
0029  041 005  330 002         STA  /2      INST FOR INTRPT #1
0030  043 005  250             IFT  1       IS KEYBOARD TO BE TESTED
0031  044 005  344 100         JMP  /KBD    YES
0032  046 005  315 234         LDA  OCH     SET UP OUTPUT WORD
0033  050 005  111 073  OT1    SSR  @73
0034  052 005  345 050         JMP  *-3     TEST OUTPUT
0035  054 005  154 070         OTA  @70     INSTRUCTION
0036  056 005  375 237         JST  INC
0037  060 005  345 050         JMP  OT1
0038  062 005  374 140         JST  /LDR    OUTPUT LEADER
0039  064 005  374 177         JST  /CKC
0040  066 005  100 073         EXC  @73
0041  070 005  111 071  RD1    SSR  @71
0042  072 005  345 070         JMP  RD1
0043  074 005  120 070  BIN    INB  @70-10  TEST BLOCK INPUT
0044  076 005  345 070         JMP  RD1
0045  100 005  100 074         EXC  @74     SHUT OFF CONTINUOUS READ
0046  102 005  375 264         JST  CAR     GO CHECK INPUT
0047  104 005  111 073  OT2    SSR  @73
0048  106 005  345 104         JMP  *-3
0049  110 005  140 070  BOT    OTB  @70-10  TEST BLOCK OUTPUT
0050  112 005  345 104         JMP  OT2
```



```
0051  114 005  375 355              JST  RSB
0052  116 005  100 072  RD2    EXC  @72     TEST STEP READER AND
0053  120 005  131 071         IAS  @71     INPUT AND SENSE INST'S
0054  122 005  345 120         JMP  *-3
0055  124 005  375 323         JST  RCK     CHECK INPUT
0056  126 005  345 116         JMP  RD2
0057  130 005  315 234         LDA  OCH
0058  132 005  155 073  OT3    OAS  @73     TEST OUTPUT AND SENSE
0059  134 005  345 132         JMP  *-3     INSTRUCTION
0060  136 005  375 237         JST  INC     INC OUTPUT WD
0061  140 005  345 132         JMP  OT3
0062  142 005  374 140         JST  /LDR    OUTPUT LEADER
0063  144 005  374 177         JST  /CKC
0064  146 005  100 075         EXC  @75     ENABLE MASK INT #1
0065  150 005  111 072         SSR  @72     TEST OKAY?
0066  152 005  040             HLT          YES
0067  153 005  040             HLT
0068  154 005  100 076         EXC  @76     ENABLE MASK INT #2
0069  156 005  022             EIN          ENABLE INTRPT'S
0070  157 005  100 073         EXC  @73     SEL CONT RD FOR INTRPT
0071  161 005  314 176         LDA  /INT    IS BLOCK INPUT FINISHED
0072  163 005  242             IFT  N
0073  164 005  345 161         JMP  *-4     NO
0074  166 005  315 110         LDA  BOT     YES, SET UP BLOCK OUTPUT
0075  170 005  330 002         STA  /2      INST
0076  172 005  375 264         JST  CAR     CHECK INPUT
0077  174 005  100 074         EXC  @74     TEST BLOCK OUTPUT INIT
0078  176 005  100 075         EXC  @75     TTY FOR INTRPT ENABLE
0079  200 005  100 076         EXC  @76     INT MASKS #1 & #2
0080  202 005  022             EIN
0081  203 005  314 176         LDA  /INT    IS BLOCK OUTPUT FINISHED
0082  205 005  242             IFT  N
0083  206 005  345 203         JMP  *-4     NO
0084  210 005  375 355         JST  RSB
0085  212 005  100 072  RD4    EXC  @72     TEST MASKED INPUT
0086  214 005  315 236         LDA  T77     CLEAR BIT 6
0087  216 005  111 071         SSR  @71
0088  220 005  345 216         JMP  *-3
0089  222 005  134 070         MIA  @70
0090  224 005  301 212         ADD  RD4     ADD BACK BIT 6
0091  226 005  375 323         JST  RCK     CHECK INPUT
0092  230 005  345 212         JMP  RD4
0093  232 005  345 037         JMP  SPT     START OVER
0094  234 005              *
0095  234 005              *    LITERALS AND POINTERS
0096  234 005              *
0097  234 005  000         OCH  PTR  **
0098  235 005  200         02H  OCT  200
0099  236 005  277         T77  OCT  277
0100  237 005              *
0101  237 005              *    INCREMENT OUTPUT WORD SUB
0102  237 005              *
0103  237 005  000 000     INC  DPT  **
0104  241 005  315 234         LDA  OCH
0105  243 005  007             ARC  I       INC I/O WORD
0106  244 005  331 234         STA  OCH
0107  246 005  321 263         ANA  077
0108  250 005  202             IFT  Z       IS THAT ALL FOR THIS TEST
0109  251 005  345 257         JMP  *+5     NO
0110  253 005  371 237         IMS  INC     YES
0111  255 005  371 237         IMS  INC
0112  257 005  315 234         LDA  OCH
0113  261 005  347 237        *JMP  INC
0114  263 005  077         077  OCT  77
0115  264 005              *    COMPARE READ
0116  264 005              *
0117  264 005  000 000     CAR  DPT  **
0118  266 005  374 216         JST  /SUP
0119  270 005  331 321         STA  BFP
0120  272 005  315 234         LDA  OCH
```

```
0121  274 005   311 212        SUB  RD4    RESET I/O WORD
0122  276 005   331 234        STA  OCH
0123  300 005   371 321  CR1   IMS  BFP
0124  302 005   313 321       *SUB  BFP    IS THIS CORRECT CHAR
0125  304 005   242            IFT  N
0126  305 005   345 313        JMP  *+5
0127  307 005   317 321       *LDA  BFP
0128  311 005   040            HLT          NO
0129  312 005   040            HLT
0130  313 005   375 237        JST  INC    YES, INC I/O WORD
0131  315 005   345 300        JMP  CR1
0132  317 005   347 264       *JMP  CAR    EXIT SUB
0133  321 005   277 010  BFP   DPT  @277-10
0134  323 005                  *
0135  323 005                  *    READ CHECK SUB
0136  323 005                  *
0137  323 005   000 000  RCK   DPT  **
0138  325 005   331 354        STA  CHR    SAVE CHAR
0139  327 005   315 234        LDA  OCH    NO, RESET I/O WORD
0140  331 005   311 354        SUB  CHR    IS THIS CORRECT CHAR
0141  333 005   242            IFT  N
0142  334 005   345 342        JMP  *+5    YES
0143  336 005   315 354        LDA  CHR    NO
0144  340 005   040            HLT
0145  341 005   040            HLT
0146  342 005   375 237        JST  INC    INC I/O WD
0147  344 005   347 323       *JMP  RCK    IMMED EXIT
0148  346 005   371 323        IMS  RCK
0149  350 005   371 323        IMS  RCK
0150  352 005   347 323       *JMP  RCK
0151  354 005   000       CHR  PTR  **
0152  355 005   000 000  RSB   DPT  **
0153  357 005   374 140        JST /LDR
0154  361 005   374 177        JST /CKC
0155  363 005   315 234        LDA  OCH
0156  365 005   311 212        SUB  RD4
0157  367 005   331 234        STA  OCH
0158  371 005   347 355       *JMP  RSB
0159  373 005                  *
0160  373 005                  *    KEYBOARD TEST
0161  373 005                  *
0162  100 000   100 000        ORG  @100
0163  100 000   100 071  KBD   EXC  @71     SELECT KEYBOARD
0164  102 000   131 071        IAS  @71
0165  104 000   345 102        JMP  *-3
0166  106 000   210            IFT  0
0167  107 000   347 121       *JMP  STP
0168  111 000   154 070        OTA  @70    OUTPUT CHARACTER JST INPU
0169  113 000   111 071        SSR  @71
0170  115 000   345 113        JMP  *-3
0171  117 000   345 100        JMP  KBD    GO CHECK IF THAT'S ALL
0172  121 000   037 005  STP   DPT  SPT
0173  123 000                  *
0174  123 000                  *    END OF BLOCK INTERRUPT SUB
0175  123 000                  *
0176  123 000   000 000  EOB   DPT  **
0177  125 000   021            DIN
0178  126 000   111 073        SSR  @73
0179  130 000   345 126        JMP  *-3
0180  132 000   371 176        IMS  INT    IND EOB INTERRUPT
0181  134 000   100 074        EXC  @74    INIT TTY
0182  136 000   347 123       *JMP  EOB
0183  140 000                  *
0184  140 000                  *    OUTPUT LEADER SUB
0185  140 000                  *
0186  140 000   000 000  LDR   DPT  **
0187  142 000   315 142        LDA  *-1
0188  144 000   151 073  LDI   OZS  @73
0189  146 000   345 144        JMP  *-3
0190  150 000   007            ARC  I
```



```
0191  151 000   202            IFT  Z
0192  152 000   345 144        JMP  LDI
0193  154 000   314 235        LDA /012
0194  156 000   155 073        OAS  @73
0195  160 000   345 156        JMP  *-3
0196  162 000   315 234        LDA  015
0197  164 000   155 073        OAS  @73
0198  166 000   345 164        JMP  *-3
0199  170 000   111 071        SSR  @71
0200  172 000   345 170  LD2   JMP  *-3
0201  174 000   347 140       *JMP  LDR
0202  176 000   000       INT  PTR  **
0203  177 000                  *
0204  177 000   000 000  CKC   DPT  **
0205  201 000   100 072        EXC  @72
0206  203 000   131 071        IAS  @71
0207  205 000   345 203        JMP  *-3
0208  207 000   311 234        SUB  015
0209  211 000   202            IFT  Z
0210  212 000   345 201        JMP  *-10
0211  214 000   347 177       *JMP  CKC
0212  216 000                  *
0213  216 000                  *
0214  216 000   000 000  SUP   DPT  **
0215  220 000   005            ARC  Z
0216  221 000   330 176        STA /INT
0217  223 000   315 233        LDA  N65
0218  225 000   330 006        STA /6
0219  227 000   330 010        STA /8
0220  231 000   347 216       *JMP  SUP
0221  233 000   277       N65  OCT  277
0222  234 000                  *
0223  234 000                  *
0224  234 000   015       015  OCT  15
0225  235 000   012       012  OCT  12
0226  236 000                  *
0227  236 000                  *
0228  002 000   002 000        ORG  2
0229  002 000   120 070        INB  @70-10
0230  004 000   375 123        JST  EOB
0231  006 000   277 000        DPT  @277-0
0232  010 000   277 000        DPT  @277-0
0233  012 000   000 000        END
```

PROCESSOR INSTRUCTION DIAGNOSTIC (PID)

PID tests all memory reference and register change instructions for all possible results and tests enough skip instructions to check all the skip logic. If any test on any instruction fails, the processor will halt.

General Processing

First of all, the jump (JMP) is tested to make sure a transfer takes place. Secondly, since the A register is set to zero upon starting the program, the IFT Z (skip if the A register is equal to zero) instruction is tested. Then the IFT N (skip if the A register is not equal to zero) and the IFT + (skip if the A register is positive) are tested respectively. Next the instructions tested to this point are used to test the LDA (load the A register from memory instruction). This sort of

procedure (i.e., use of any of the previously checked instructions used to check the next instruction) is used to check all the remaining instructions.

The skip on the setting of the sense switch is the last instruction tested in each pass through the diagnostic. This test assumes that the sense switch is depressed. Therefore, if the sense switch is not depressed, the diagnostic should halt at the end of one pass, and if depressed, the diagnostic should go through another pass.

Operating Procedures

When a test fails the processor halts. At each halt two locations are reserved. These locations are reserved so that the maintenance engineer can store a jump instruction (to the beginning of the test that failed) in these locations. This gives the engineer the opportunity to run continuously a test that fails, so that he may look at associated signals with a scope.

Setup Procedures

1. Load the PID in the memory using the object loader.
2. Place the step-run switch in the step mode.
3. Make sure the relative and manual execute switches are released.
4. Depress the reset switch.
5. Depress the page register select switch (P), depress the clear switch, and enter on the data switches the page number in the memory where the instruction diagnostic starts (1).
6. Depress the location register select switch (L), depress the clear switch.
7. Depress the instruction register select switch (I), depress the clear switch.
8. Depress the A register select switch (A), depress the clear switch.
9. If the diagnostic is to be run continuously depress the sense switch.
10. Place the step-run switch in the run mode.
11. Depress the cycle switch.
12. If the diagnostic halts other than on location 324_8 of page 0, an instruction failed a test. To determine what test failed and how, look at the attached listing. If it is desired to have the test run repeatedly, store a jump instruction (to the beginning of the test) in place of the two halts.
13. When the sense switch is depressed, the diagnostic will halt at location 324_8, page 0.
14. If it is desired to continue after any halt, depress the cycle switch twice.


```
0001  000 000                  *
0002  000 000                  *   INSTRUCTION DIAGNOSTIC
0003  000 000                  *
0004  000 000                  *   THIS PROGRAM TESTS ALL INSTRUCTIONS
0005  000 000                  *   WITH EXCEPTION OF THE I/O INSTRUCTIONS,
0006  000 000                  *   IT TESTS EACH INSTRUCTION FOR ALL
0007  000 000                  *   POSSIBLE EFFECTS.
0008  000 000                  *
0009  000 000                  *   IF AN INSTRUCTION FAILS ON A TEST,
0010  000 000                  *   THE PROCESSOR WILL HALT. AT EACH HALT
0011  000 000                  *   THE HALT IS REPEATED. THE REASON FOR
0012  000 000                  *   THIS IS TO ALLOW THE ENGINEER TO
0013  000 000                  *   REPLACE THE TWO HALTS WITH A JUMP
0014  000 000                  *   INSTRUCTION TO THE BEGINNING OF THE
0015  000 000                  *   TEST THAT FAILED.  THE ENGINEER CAN
0016  000 000                  *   THEN HOOK-UP A SCOPE AND CHECK SIGNALS.
0017  000 000                  *
0018  000 000                  *
0019  000 000                  *   TO OPERATE
0020  000 000                  *
0021  000 000                  *   1) SET SENSE SWITCH
0022  000 000                  *      OFF = HALT AT END OF TEST PASS
0023  000 000                  *      ON  = TEST CONTINUOUSLY
0024  000 000                  *
0025  000 000                  *   2) IF PROGRAM HALTS OTHER THAN AT THE
0026  000 000                  *      NORMAL HALT, LOOK AT INSTRUCTIONS
0027  000 000                  *      WHICH PRECEDE THE HALT ON THIS
0028  000 000                  *      LISTING TO DETERMINE INSTRUCTION
0029  000 000                  *      THAT FAILED.
0030  000 000                  *
0031  000 001  000 001           ORG  0-1
0032  000 001  345 003  BGN     JMP  *+2      TEST JMP INST
0033  002 001  040             HLT           FAILED
0034  003 001  000             NOP
0035  004 001  345 010         JMP  *+3
0036  006 001  040             HLT           JMP FAILED
0037  007 001  040             HLT           JMP FAILED
0038  010 001  345 015         JMP  *+4
0039  012 001  040             HLT           JMP FAILED
0040  013 001  040             HLT           JMP FAILED
0041  014 001  040             HLT           JMP FAILED
0042  015 001  202             IFT  Z        TEST IFT ZERO INST, A = 0
0043  016 001  040             HLT           IFT ZERO FAILED
0044  017 001  040             HLT
0045  020 001  242             IFT  N        TEST IFT NON-ZERO, A = 0
0046  021 001  345 025         JMP  *+3
0047  023 001  040             HLT
0048  024 001  201             IFT  -        TEST IFT MINUS, A = 0
0049  025 001  345 031         JMP  *+3
0050  027 001  040             HLT           IFT - FAILED
0051  030 001  040             HLT
0052  031 001  315 030         LDA  *-2      TEST LDA INST & IFT Z
0053  033 001  202             IFT  Z        WITH NON-ZERO VALUE
0054  034 001  345 040         JMP  *+3
0055  036 001  040             HLT           LDA FAILED OR IFT Z
0056  037 001  040             HLT
0057  040 001  242             IFT  N        TEST IFT NON-ZERO, A=NZ
0058  041 001  040             HLT           IFT N FAILED
0059  042 001  040             HLT
0060  043 001  201             IFT  -        TEST IFT MINUS, A=;
0061  044 001  345 050         JMP  *+3
0062  046 001  040             HLT           IFT - FAILED
0063  047 001  040             HLT
0064  050 001  315 373         LDA  N1       MINUS ONE TO A
0065  052 001  201             IFT  -        TEST IFT MINUS, A= -
0066  053 001  040             HLT           IFT - FAILED
0067  054 001  040             HLT
0068  055 001  202             IFT  Z        TEST IFT ZERO, A= -
0069  056 001  345 062         JMP  *+3
0070  060 001  040             HLT           IFT Z FAILED
```

```
0071  061 001  040          HLT
0072  062 001  242          IFT  N      TEST IFT NON-ZERO, A= -
0073  063 001  040          HLT         IFT N FAILED
0074  064 001  040          HLT
0075  065 001  005          ARC  Z      TEST ARC Z  INST
0076  066 001  202          IFT  Z      WITH A NON-ZERO
0077  067 001  040          HLT         ARC Z FAILED
0078  070 001  040          HLT
0079  071 001  315 364      LDA  ZER    TEST LDA INST
0080  073 001  242          IFT  N      WITH ZERO
0081  074 001  345 100      JMP  *+3
0082  076 001  040          HLT         LDA  FAILED
0083  077 001  040          HLT
0084  100 001  202          IFT  Z
0085  101 001  040          HLT         LDA FAILED
0086  102 001  040          HLT
0087  103 001  005          ARC  Z      TEST ARC Z INST
0088  104 001  202          IFT  Z      WITH A ALREADY ZERO
0089  105 001  040          HLT         ARC Z FAILED
0090  106 001  040          HLT
0091  107 001  315 365      LDA  C1     TEST SUB INST, A = 1
0092  111 001  311 365      SUB  C1
0093  113 001  202          IFT  Z
0094  114 001  040          HLT         SUB FAILED
0095  115 001  040          HLT
0096  116 001  315 366      LDA  EVN    TEST SUB INST, WITH A
0097  120 001  311 366      SUB  EVN    EVEN AND NEG
0098  122 001  202          IFT  Z
0099  123 001  040          HLT         SUB FAILED
0100  124 001  040          HLT
0101  125 001  315 367      LDA  ODD    TEST SUB WITH A =
0102  127 001  311 367      SUB  ODD    TO HIGH ODD NUM
0103  131 001  202          IFT  Z
0104  132 001  040          HLT         SUB FAILED
0105  133 001  040          HLT
0106  134 001  315 366      LDA  EVN    TEST STA INST WITH A
0107  136 001  331 370      STA  TMP    EVEN AND NEG
0108  140 001  311 366      SUB  EVN
0109  142 001  202          IFT  Z
0110  143 001  040          HLT         STA DESTROYED CONTENTS
0111  144 001  040          HLT         OF A-REG
0112  145 001  315 367      LDA  ODD    TEST STA INST WITH A ODD
0113  147 001  331 370      STA  TMP
0114  151 001  005          ARC  Z      CLEAR A
0115  152 001  315 370      LDA  TMP
0116  154 001  311 367      SUB  ODD
0117  156 001  202          IFT  Z
0118  157 001  040          HLT         STA DID STORE INTO MEMORY
0119  160 001  040          HLT
0120  161 001  005          ARC  Z      CLEAR A
0121  162 001  335 370      IMA  TMP    TEST IMA INST, A = 0
0122  164 001  311 367      SUB  ODD    AND M = ODD NUM
0123  166 001  202          IFT  Z
0124  167 001  040          HLT         IMA FAILED TO LOAD WITH
0125  170 001  040          HLT         CONTENTS OF MEMORY
0126  171 001  335 370      IMA  TMP
0127  173 001  202          IFT  Z
0128  174 001  040          HLT         IMA FAILED TO STORE ZERO
0129  175 001  040          HLT         INTO MEMORY
0130  176 001  315 366      LDA  EVN    TEST IMA INST WITH A = 0
0131  200 001  331 370      STA  TMP    AND M = EVEN AND NEG
0132  202 001  005          ARC  Z      CLEAR A
0133  203 001  335 370      IMA  TMP
0134  205 001  311 366      SUB  EVN
0135  207 001  202          IFT  Z
0136  210 001  040          HLT         IMA FAILED TO LOAD A WITH
0137  211 001  040          HLT         CONTENTS OF MEMORY
0138  212 001  335 370      IMA  TMP
0139  214 001  202          IFT  Z
0140  215 001  040          HLT         IMA FAILED TO STORE ZERO
```



```
0141  216 001  040          HLT         INTO MEMORY
0142  217 001  315 365      LDA  C1     TEST ADD INST, A = 1
0143  221 001  301 365      ADD  C1     ADDING POSITIVE NUM
0144  223 001  311 371      SUB  C2
0145  225 001  202          IFT  Z
0146  226 001  040          HLT         ADD FAILED
0147  227 001  040          HLT
0148  230 001  315 365      LDA  C1     TEST ADD INST, A = 1
0149  232 001  301 372      ADD  N2     ADDING NEG NUM, THEN POS
0150  234 001  301 365      ADD  C1     NUM
0151  236 001  202          IFT  Z
0152  237 001  040          HLT         ADD FAILED
0153  240 001  040          HLT
0154  241 001  315 373      LDA  N1     TEST ADD INST, A = -1
0155  243 001  301 371      ADD  C2     ADDING POS. NUM
0156  245 001  311 365      SUB  C1
0157  247 001  202          IFT  Z
0158  250 001  040          HLT         ADD FAILED
0159  251 001  040          HLT
0160  252 001  315 373      LDA  N1     TEST ADD INST, A = -1
0161  254 001  301 373      ADD  N1     ADDING NEG NUM, THEN
0162  256 001  301 371      ADD  C2     ADD POS NUM
0163  260 001  202          IFT  Z
0164  261 001  040          HLT         ADD FAILED
0165  262 001  040          HLT
0166  263 001  315 373      LDA  N1     TEST SUB & ADD SEQUENCE
0167  265 001  311 365      SUB  C1     SUB POS NUM, ADD POS NUM
0168  267 001  301 371      ADD  C2
0169  271 001  202          IFT  Z
0170  272 001  040          HLT         SUB & ADD SEQ FAILED
0171  273 001  040          HLT
0172  274 001  315 373      LDA  N1     TEST SUB -1 FROM -1
0173  276 001  311 373      SUB  N1
0174  300 001  202          IFT  Z
0175  301 001  040          HLT         SUB -1 FROM -1 FAILED
0176  302 001  040          HLT
0177  303 001  315 365      LDA  C1     TEST SUB & ADD SEQUENCE
0178  305 001  311 373      SUB  N1     SUB NEG NUM, ADD NEG NUM
0179  307 001  301 372      ADD  N2
0180  311 001  202          IFT  Z
0181  312 001  040          HLT         SUB & ADD SEQ FAILED
0182  313 001  040          HLT
0183  314 001  315 366      LDA  EVN    TEST ANA INST, A-REG EVEN
0184  316 001  321 366      ANA  EVN    MASK EVEN
0185  320 001  311 366      SUB  EVN
0186  322 001  202          IFT  Z
0187  323 001  040          HLT         ANA FAILED
0188  324 001  040          HLT
0189  325 001  315 367      LDA  ODD    TEST ANA INST, A-REG ODD
0190  327 001  321 367      ANA  ODD    MASK ODD
0191  331 001  311 367      SUB  ODD
0192  333 001  202          IFT  Z
0193  334 001  040          HLT         ANA FAILED
0194  335 001  040          HLT
0195  336 001  315 366      LDA  EVN    TEST ANA INST, A-REG EVEN
0196  340 001  321 367      ANA  ODD    MASK ODD
0197  342 001  202          IFT  Z
0198  343 001  040          HLT         ANA FAILED
0199  344 001  040          HLT
0200  345 001  317 374      *LDA ODP    TEST INDIRECT
0201  347 001  311 367      SUB  ODD
0202  351 001  202          IFT  Z
0203  352 001  040          HLT         INDIRECT DID NOT WORK
0204  353 001  040          HLT
0205  354 001  347 360      *JMP P2N    GO TO NEXT PAGE
0206  356 001  040          HLT         INDIRECT JUMP
0207  357 001  040          HLT         DID NOT WORK
0208  360 001  000 002  P2N DPT  NP2
```

```
0209  362 001  040          HLT         IND JUMP WAS
0210  363 001  040          HLT         DIRECT - FAILED
0211  364 001               *
0212  364 001               *   LITERALS AND STORAGE WORDS
0213  364 001               *
0214  364 001  000     ZER  OCT  0
0215  365 001  001     C1   OCT  1
0216  366 001  252     EVN  OCT  252
0217  367 001  125     ODD  OCT  125
0218  370 001  000     TMP  PTR  **
0219  371 001  002     C2   OCT  2
0220  372 001  376     N2   DEC  -2
0221  373 001  377     N1   DEC  -1
0222  374 001  363 002 ODP  DPT  OD1
0223  000 002  000 002       ORG  @0-2
0224  000 002  315 363 NP2  LDA  OD1
0225  002 002  305 363       IOR  OD1    TEST IOR INST, A-REG ODD
0226  004 002  311 363       SUB  OD1    AND OR CONSTANT ODD
0227  006 002  202           IFT  Z
0228  007 002  040           HLT         IOR FAILED
0229  010 002  040           HLT
0230  011 002  315 364       LDA  EV1    TEST IOR INST, A-REG EVEN
0231  013 002  305 364       IOR  EV1    AND OR CONSTANT EVEN
0232  015 002  311 364       SUB  EV1
0233  017 002  202           IFT  Z
0234  020 002  040           HLT         IOR FAILED
0235  021 002  040           HLT
0236  022 002  315 364       LDA  EV1    TEST IOR INST, A-REG EVEN
0237  024 002  305 363       IOR  OD1    AND OR CONSTANT ODD
0238  026 002  301 365       ADD  O1
0239  030 002  202           IFT  Z
0240  031 002  040           HLT         IOR FAILED
0241  032 002  040           HLT
0242  033 002  315 363       LDA  OD1    TEST IOR INST, A-REG ODD
0243  035 002  305 364       IOR  EV1    AND OR CONSTANT EVEN
0244  037 002  301 365       ADD  O1
0245  041 002  202           IFT  Z
0246  042 002  040           HLT         IOR FAILED
0247  043 002  040           HLT
0248  044 002  005           ARC  Z      TEST IMS INST, A = 0
0249  045 002  331 366       STA  TST    AND M = 0
0250  047 002  371 366       IMS  TST
0251  051 002  202           IFT  Z
0252  052 002  040           HLT         IMS INST HAD EFFECT ON
0253  053 002  040           HLT         A-REG
0254  054 002  315 366       LDA  TST
0255  056 002  311 365       SUB  O1
0256  060 002  202           IFT  Z
0257  061 002  040           HLT         IMS INST FAILED TO INC
0258  062 002  040           HLT
0259  063 002  315 367       LDA  C77    TEST IMS INST FOR SKIP
0260  065 002  331 366       STA  TST    ON ZERO RESULT, ) = 177
0261  067 002  371 366       IMS  TST
0262  071 002  345 075       JMP  *+3
0263  073 002  040           HLT         IMS SKIPPED WHEN GOING
0264  074 002  040           HLT         FROM MAX POS TO MAX NEG
0265  075 002  315 366       LDA  TST
0266  077 002  201           IFT  -
0267  100 002  040           HLT         IMS DID NOT INC FROM MAX
0268  101 002  040           HLT         POS TO NEG
0269  102 002  321 367       ANA  C77
0270  104 002  202           IFT  Z
0271  105 002  040           HLT         IMS DID NOT INC FROM MAX
0272  106 002  040           HLT         POS TO MAX NEG
0273  107 002  315 370       LDA  M1     TEST IMS INST FOR SKIP
0274  111 002  331 366       STA  TST    ON ZERO RESULT, ) = -1
0275  113 002  371 366       IMS  TST    A = -1
0276  115 002  040           HLT         IMS DID NOT SKIP ON
0277  116 002  040           HLT         ZERO RESULT
```



```
0278  117 002  242           IFT  N
0279  120 002  040           HLT         IMS INST HAD EFFECT ON
0280  121 002  040           HLT         A-REG
0281  122 002  315 366       LDA  TST
0282  124 002  202           IFT  Z
0283  125 002  040           HLT         IMS INST DID NOT INC
0284  126 002  040           HLT         MEMORY TO ZERO
0285  127 002  050           ORC  R      RESET OV
0286  130 002  244           IFT  R      IS OV RESET
0287  131 002  040           HLT         ORC OV OR IFT R
0288  132 002  040           HLT         FAILED
0289  133 002  204           IFT  S      IS OV SET
0290  134 002  345 140       JMP  *+3
0291  136 002  040           HLT         IFT S INST FAILED
0292  137 002  040           HLT
0293  140 002  070           ORC  C      COMPLEMENT OV   0 TO 1
0294  141 002  244           IFT  R      IS OV RESET
0295  142 002  345 146       JMP  *+3    NO;
0296  144 002  040           HLT         ORC C INST FAILED
0297  145 002  040           HLT
0298  146 002  070           ORC  C
0299  147 002  204           IFT  S      IS OV SET
0300  150 002  040           HLT         ORC C INST FAILED
0301  151 002  040           HLT
0302  152 002  060           ORC  S      TEST SET OV INST
0303  153 002  204           IFT  S
0304  154 002  040           HLT         ORC S FAILED
0305  155 002  040           HLT
0306  156 002  244           IFT  R      TEST ORC S FOR RESET OV
0307  157 002  040           HLT         ORC S DID NOT RESET OV
0308  160 002  040           HLT
0309  161 002  060           ORC  S      SET OV
0310  162 002  070           ORC  C      COMPLEMENT OV  1 TO 0
0311  163 002  244           IFT  R      IS OV RESET
0312  164 002  040           HLT         NO, ORC INST FAILED
0313  165 002  040           HLT
0314  166 002  315 367       LDA  C77    TEST SETTING OV WHEN
0315  170 002  301 365       ADD  O1     GOING FROM MAX POS TO
0316  172 002  204           IFT  S      MAX NEG
0317  173 002  040           HLT         OV WAS NOT SET - FAILED
0318  174 002  040           HLT
0319  175 002  050           ORC  R      RESET OV
0320  176 002  315 370       LDA  M1     TEST SETTING OF OV WHEN
0321  200 002  301 365       ADD  O1     GOING FROM MIN NEG
0322  202 002  244           IFT  R      TO ZERO
0323  203 002  040           HLT         OV WAS SET WHEN GOING
0324  204 002  040           HLT         FROM -1 TO ZERO - FAILED
0325  205 002  315 374       LDA  T25    TEST SETTING OF OV WHEN
0326  207 002  311 367       SUB  C77    GOING FROM NEG
0327  211 002  204           IFT  S      TO POS NUM
0328  212 002  040           HLT         FAILED - OV SET
0329  213 002  040           HLT
0330  214 002  050           ORC  R      RESET OV
0331  215 002  315 370       LDA  M1     TEST SETTING OF OV WHEN
0332  217 002  301 370       ADD  M1     GOING FROM MIN NEG THRU
0333  221 002  244           IFT  R      ZERO TO NEG
0334  222 002  040           HLT
0335  223 002  040           HLT
0336  224 002  315 364       LDA  EV1    TEST LOGICAL LEFT SHIFT
0337  226 002  055           SHF  L      A IS EVEN
0338  227 002  204           IFT  S
0339  230 002  040           HLT         FAILED BIT DID NOT GET
0340  231 002  040           HLT         SHIFTED INTO OV
0341  232 002  311 371       SUB  C24
0342  234 002  202           IFT  Z
0343  235 002  040           HLT         FAILED, WORD DID NOT GET
0344  236 002  040           HLT         SHIFTED CORRECTLY
0345  237 002  315 363       LDA  OD1    TEST LOGICAL LEFT SHIFT
0346  241 002  055           SHF  L      A IS ODD
0347  242 002  244           IFT  R
```

```
0348  243 002  040          HLT          FAILED - BIT WAS SHIFTED
0349  244 002  040          HLT          INTO OV
0350  245 002  311 364      SUB  EV1
0351  247 002  202          IFT  Z
0352  250 002  040          HLT          FAILED, WORD DID NOT
0353  251 002  040          HLT          GET SHIFTED CORRECTLY
0354  252 002  315 364      LDA  EV1     TEST LOGICAL RIGHT SHIFT
0355  254 002  057          SHF  R       A IS EVEN
0356  255 002  244          IFT  R
0357  256 002  040          HLT          FAILED, OV WAS SET
0358  257 002  040          HLT
0359  260 002  311 363      SUB  OD1
0360  262 002  202          IFT  Z
0361  263 002  040          HLT          FAILED, WORD DID NOT
0362  264 002  040          HLT          GET SHIFTED CORRECTLY
0363  265 002  315 363      LDA  OD1     TEST LOGIVAL RIGHT SHIFT
0364  267 002  057          SHF  R       A IS ODD
0365  270 002  204          IFT  S
0366  271 002  040          HLT          FAILED, OV DID NOT GET
0367  272 002  040          HLT          SET
0368  273 002  311 372      SUB  052
0369  275 002  202          IFT  Z
0370  276 002  040          HLT          FAILED, WORD DID NOT GET
0371  277 002  040          HLT          SHIFTED CORRECTJY
0372  300 002  005          ARC  Z       CLEAR A
0373  301 002  315 363      LDA  OD1     TEST ANA INST WITH A SET
0374  303 002  321 364      ANA  EV1     ODD AND MASK EVEN
0375  305 002  202          IFT  Z
0376  306 002  040          HLT          ANA INST FAILED
0377  307 002  040          HLT
0378  310 002  060          ORC  S       SET OV
0379  311 002  315 364      LDA  EV1     TEST ROTATE RIGHT SHIFT
0380  313 002  047          SHF  )       A IS EVEN
0381  314 002  244          IFT  R
0382  315 002  040          HLT          FAILED, OV WAS SET
0383  316 002  040          HLT
0384  317 002  047          SHF  )       TEST ROTATE RIGHT AGAIN
0385  320 002  204          IFT  S
0386  321 002  040          HLT          FAILED, OV WAS NOT SET
0387  322 002  040          HLT
0388  323 002  311 373      SUB  C52
0389  325 002  202          IFT  Z
0390  326 002  040          HLT          SHIFT ) FAILED, WORD NOT
0391  327 002  040          HLT          SHIFTED CORRECTLY
0392  330 002  050          ORC  R       RESET OV
0393  331 002  315 363      LDA  OD1     TEST ROTATE RIGHT SHIFT
0394  333 002  047          SHF  )       A IS ODD
0395  334 002  047          SHF  )
0396  335 002  311 374      SUB  T25
0397  337 002  202          IFT  Z
0398  340 002  040          HLT          FAILED, WORD NOT SHIFTED
0399  341 002  040          HLT          CORRECTLY
0400  342 002  060          ORC  S       SET OV
0401  343 002  315 363      LDA  OD1     TEST ROTATE LEFT SHIFT
0402  345 002  045          SHF  (       A IS ODD
0403  346 002  244          IFT  R
0404  347 002  040          HLT          FAILED, OV SET
0405  350 002  040          HLT
0406  351 002  045          SHF  (       SHIFT AGAIN
0407  352 002  311 375      SUB  C26
0408  354 002  202          IFT  Z
0409  355 002  040          HLT          FAILED, WORD NOT SHIFTED
0410  356 002  040          HLT          CORRECTLY
0411  357 002  347 361      *JMP P3N     GO TO NEXT PAGE
0412  361 002          *
0413  361 002          *   LITERALS AND STORAGE WORDS
0414  361 002          *
0415  361 002  000 003  P3N  DPT  NP3
0416  363 002  125      OD1  OCT  125
```



```
0417  364 002  252      EV1  OCT  252
0418  365 002  001      01   OCT  1
0419  366 002  000      TST  OCT  0
0420  367 002  177      C77  OCT  177
0421  370 002  377      M1   OCT  377
0422  371 002  124      C24  OCT  124
0423  372 002  052      052  OCT  52
0424  373 002  152      C52  OCT  152
0425  374 002  225      T25  OCT  225
0426  375 002  126      C26  OCT  126
0427  000 003  000 003       ORG  @0-3
0428  000 003  050      NP3  ORC  R       RESET OV
0429  001 003  315 322       LDA  VEN     TEST ROTATE LEFT SHIFT
0430  003 003  045          SHF  (       A= EVEN
0431  004 003  045          SHF  (
0432  005 003  311 324      SUB  T51
0433  007 003  202          IFT  Z
0434  010 003  040          HLT          SHF ( FAILED
0435  011 003  040          HLT
0436  012 003  315 326      LDA  DDO
0437  014 003  041          SHF  <       TEST ARITH SHIFT LEFT
0438  015 003  311 327      SUB  E52     A= ODD
0439  017 003  202          IFT  Z
0440  020 003  040          HLT          SHF < FAILED
0441  021 003  040          HLT
0442  022 003  315 322      LDA  VEN     TEST ARITH SHIFT LEFT
0443  024 003  041          SHF  <       A= EVEN
0444  025 003  311 330      SUB  TT
0445  027 003  202          IFT  Z
0446  030 003  040          HLT          SHF < FAILED
0447  031 003  040          HLT
0448  032 003  315 326      LDA  DDO
0449  034 003  043          SHF  >       TEST ARITH SHIFT RIGHT
0450  035 003  311 327      SUB  E52     A= ODD
0451  037 003  202          IFT  Z
0452  040 003  040          HLT          SHIFT > FAILED
0453  041 003  040          HLT
0454  042 003  244          IFT  R
0455  043 003  040          HLT          SHF > FAILED, OV WAS SET
0456  044 003  040          HLT
0457  045 003  315 322      LDA  VEN     TEST ARITH SHIFT RIGHT
0458  047 003  043          SHF  >       A= EVEN
0459  050 003  311 331      SUB  TT5
0460  052 003  202          IFT  Z
0461  053 003  040          HLT          SHF> FAILED
0462  054 003  040          HLT
0463  055 003  315 326      LDA  DDO     TEST ONES COMPLEMENT A
0464  057 003  011          ARC  C       A= EVEN
0465  060 003  311 322      SUB  VEN
0466  062 003  202          IFT  Z
0467  063 003  040          HLT          ARC C FAILED
0468  064 003  040          HLT
0469  065 003  315 322      LDA  VEN     TEST ONES COMPLEMENT A
0470  067 003  011          ARC  C       A= ODD
0471  070 003  311 326      SUB  DDO
0472  072 003  202          IFT  Z
0473  073 003  040          HLT          ARC C FAILED
0474  074 003  040          HLT
0475  075 003  315 335      LDA  L1      TEST ONES COMPLEMENT A
0476  077 003  011          ARC  C       A= -1
0477  100 003  202          IFT  Z
0478  101 003  040          HLT          ARC C FAILED
0479  102 003  040          HLT
0480  103 003  005          ARC  Z       CLEAR
0481  104 003  331 114      STA  HLD     JST RETURN ADR STORE
0482  106 003  331 115      STA  HLD+1   LOCATIONS
0483  110 003  375 114      JST  HLD     TEST JUMP & SAVE
0484  112 003  040      PTT  HLT          JST INST DID NOT TRANSFER
0485  113 003  040          HLT
```

```
0486  114 003  000 000  HLD  DPT  **
0487  116 003  315 114       LDA  HLD
0488  120 003  311 336       SUB  PPT
0489  122 003  202           IFT  Z
0490  123 003  040           HLT       LOC OF RETURN ADR
0491  124 003  040           HLT       IS INCORRECT
0492  125 003  315 115       LDA  HLD+1
0493  127 003  311 337       SUB  PPT+1
0494  131 003  202           IFT  Z
0495  132 003  040           HLT       PAGE OF RETURN ADR
0496  133 003  040           HLT       IS INCORRECT
0497  134 003  315 322       LDA  VEN  TEST NEGATE A, A= EVEN
0498  136 003  015           ARC  N
0499  137 003  301 322       ADD  VEN
0500  141 003  202           IFT  Z
0501  142 003  040           HLT       ARC N FAILED
0502  143 003  040           HLT
0503  144 003  315 326       LDA  DDO  TEST NEGATE A, A = ODD
0504  146 003  015           ARC  N
0505  147 003  301 326       ADD  DDO
0506  151 003  202           IFT  Z
0507  152 003  040           HLT       ARC N FAILED
0508  153 003  040           HLT
0509  154 003  315 322       LDA  VEN  TEST INCREMENT A
0510  156 003  007           ARC  I    A= EVEN
0511  157 003  311 323       SUB  VEN+1
0512  161 003  202           IFT  Z
0513  162 003  040           HLT       ARC I FAILED
0514  163 003  040           HLT
0515  164 003  315 335       LDA  L1   TEST INCREMENT A
0516  166 003  007           ARC  I    A= -1
0517  167 003  202           IFT  Z
0518  170 003  040           HLT       ARC I FAILED
0519  171 003  040           HLT
0520  172 003  315 332       LDA  K77  TEST ARC I
0521  174 003  007           ARC  I    FOR OV
0522  175 003  204           IFT  S
0523  176 003  040           HLT       ARC I FAILED
0524  177 003  040           HLT
0525  200 003  001           ARC  -1   TEST INCREMENT A
0526  201 003  007           ARC  I    A=-1
0527  202 003  202           IFT  Z
0528  203 003  040           HLT       ARC I FAILED
0529  204 003  040           HLT
0530  205 003  007           ARC  I    INC A & TEST DEC A
0531  206 003  003           ARC  D    A = 1
0532  207 003  202           IFT  Z
0533  210 003  040           HLT       ARC D FAILED
0534  211 003  040           HLT
0535  212 003  315 334       LDA  TOO  TEST ARC D
0536  214 003  003           ARC  D    FOR OV
0537  215 003  204           IFT  S
0538  216 003  040           HLT       ARC D FAILED
0539  217 003  040           HLT
0540  220 003  315 326       LDA  DDO  TEST DECREMENT A
0541  222 003  003           ARC  D    A= ODD
0542  223 003  311 325       SUB  DDO-1
0543  225 003  202           IFT  Z
0544  226 003  040           HLT       ARC D FAILED
0545  227 003  040           HLT
0546  230 003  017           ARC  1    TEST TRANS ONE TO A
0547  231 003  003           ARC  D    A=1
0548  232 003  202           IFT  Z
0549  233 003  040           HLT       ARC 1 FAILED
0550  234 003  040           HLT
0551  235 003  050           ORC  R
0552  236 003  315 332       LDA  K77  TEST IMS
0553  240 003  331 333       STA  TLY  FOR OV
0554  242 003  371 333       IMS  TLY
0555  244 003  204           IFT  S
```



```
0556  245 003  040           HLT       IMS FAILED
0557  246 003  040           HLT
0558  247 003  314 341       LDA /C31  TEST MR OF PAGE ZERO
0559  251 003  311 340       SUB  K31
0560  253 003  202           IFT  Z
0561  254 003  040           HLT       MEMORY REFERENCE OF
0562  255 003  040           HLT       PAGE ZERO FAILED
0563  256 003  377 341       *JST PJT  TEST INDIRECT JUMP & SAVE
0564  260 003  040      ISB  HLT       JST INDIRECT FAILED
0565  261 003  040           HLT
0566  262 003  317 343       *LDA IPE  TEST MULTI-LEVEL INDIRECT
0567  264 003  311 322       SUB  VEN  MEMORY REFERENCE
0568  266 003  202           IFT  Z
0569  267 003  040           HLT       MULTI-LEVEL FAILED
0570  270 003  040           HLT
0571  271 003  017           ARC  1
0572  272 003  377 345       *JST IPS  TEST MULTI-LEVEL IND JST
0573  274 003  040      BIS  HLT       TRANSFER FAILED
0574  275 003  040           HLT
0575  276 003  347 347       *JMP IPJ  TEST MULTI-LEVEL IND JUMP
0576  300 003  040           HLT
0577  301 003  040           HLT       TRANSFER FAILED
0578  302 003  374 342  BIJ  JST /PZS  TEST JST TO PAGE ZERO
0579  304 003  040      ONW  HLT       TRANSFER FAILED
0580  305 003  040           HLT
0581  306 003  001           ARC  -1
0582  307 003  376 337       *JST /IPZ TEST IND JST TO PAGE 0
0583  311 003  040      ONZ  HLT       TRANSFER FAILED
0584  312 003  040           HLT
0585  313 003  344 320       JMP /@320 TEST JUMP TO PAGE ZERO
0586  315 003  040           HLT       TRANSFER FAILED
0587  316 003  040           HLT
0588  320 003  320 003       ORG  @320-3
0589  320 003  040           HLT       TRANSFER TO PAGE ZERO
0590  321 003  040           HLT       FAILED
0591  322 003               *
0592  322 003               *   LITERALS AND STORAGE WORDS
0593  322 003               *
0594  322 003  252       VEN  OCT  252
0595  323 003  253            OCT  253
0596  324 003  251       T51  OCT  251
0597  325 003  124            OCT  124
0598  326 003  125       DDO  OCT  125
0599  327 003  052       E52  OCT  52
0600  330 003  324       TT   OCT  324
0601  331 003  325       TT5  OCT  325
0602  332 003  177       K77  OCT  177
0603  333 003  000       TLY  OCT  0
0604  334 003  200       TOO  OCT  200
0605  335 003  377       L1   DEC  -1
0606  336 003  112 003   PPT  DPT  PTT
0607  340 003  131       K31  OCT  131
0608  341 003  000 004   PJT  DPT  SBR
0609  343 003  102 004   IPE  DPT  NEV
0610  345 003  341 203   IPS *DPT  PJT
0611  347 003  351 203   IPJ *DPT  JIP
0612  351 003  103 004   JIP  DPT  PIJ
0613  000 004  000 004        ORG  0-4
0614  000 004  000 000   SBR  DPT  **      SUB FOR TEST IND JST'S
0615  002 004  242            IFT  N
0616  003 004  345 034        JMP  SB1
0617  005 004  241            IFT  +
0618  006 004  345 054        JMP  SB3
0619  010 004  315 000        LDA  SBR
0620  012 004  311 100        SUB  SIB
0621  014 004  202            IFT  Z
0622  015 004  040            HLT       MULTI-LEVEL IND JST DID
0623  016 004  040            HLT       NOT STORE COR LOC RET ADR
0624  017 004  315 001        LDA  SBR+1
```

```
0625  021 004   311 101        SUB  SIB+1
0626  023 004   202            IFT  Z
0627  024 004   040            HLT          MULTI-LEVEL IND JST DID
0628  025 004   040            HLT          NOT SAVE COR PAGE RET ADR
0629  026 004   371 000  SB2   IMS  SBR
0630  030 004   371 000        IMS  SBR
0631  032 004   347 000       *JMP  SBR
0632  034 004   315 000  SB1   LDA  SBR
0633  036 004   311 076        SUB  BSI
0634  040 004   202            IFT  Z
0635  041 004   040            HLT
0636  042 004   040            HLT
0637  043 004   315 001        LDA  SBR+1
0638  045 004   311 077        SUB  BSI+1
0639  047 004   242            IFT  N
0640  050 004   345 026        JMP  SB2
0641  052 004   040            HLT
0642  053 004   040            HLT
0643  054 004   315 000  SB3   LDA  SBR
0644  056 004   311 074        SUB  SIX
0645  060 004   202            IFT  Z
0646  061 004   040            HLT
0647  062 004   040            HLT
0648  063 004   315 001        LDA  SBR+1
0649  065 004   311 075        SUB  SIX+1
0650  067 004   242            IFT  N
0651  070 004   345 026        JMP  SB2
0652  072 004   040            HLT
0653  073 004   040            HLT
0654  074 004   311 003  SIX   DPT  ONZ
0655  076 004   260 003  BSI   DPT  ISB
0656  100 004   274 003  SIB   DPT  BIS
0657  102 004   252       NEV  OCT  252
0658  103 004   347 107  PIJ *JMP  JIB     RETURN FROM MULTI IND JMP
0659  105 004   040            HLT
0660  106 004   040            HLT
0661  107 004   302 003  JIB   DPT  BIJ
0662  320 000   320 000        ORG  @320-0
0663  320 000   210            IFT  0       TEST SENSE SWITCH OFF
0664  321 000   345 325        JMP  *+3
0665  323 000   040            HLT          IFT 0 FAILED OR SS
0666  324 000   040            HLT          HAS BEEN RELEASED
0667  325 000   250            IFT  1       TEST SENSE SWITCH ON
0668  326 000   040            HLT          IFT  1 FAILED
0669  327 000   040            HLT
0670  330 000   005            ARC  Z       CLEAR A AND
0671  331 000   347 335       *JMP  BNG     START OVER
0672  333 000   040            HLT
0673  334 000   040            HLT
0674  335 000   000 001  BNG   DPT  BGN
0675  337 000   000 004  IPZ   DPT  SBR
0676  341 000   131        C31  OCT  131
0677  342 000   000 000   PZS   DPT  **    SUB TO TEST JST TO PAGE 0
0678  344 000   315 342         LDA  PZS
0679  346 000   311 370         SUB  SIZ
0680  350 000   202             IFT  Z
0681  351 000   040             HLT        JST TO PAGE ZERO DID NOT
0682  352 000   040             HLT        STORE LOC RET ADR
0683  353 000   315 343         LDA  PZS+1
0684  355 000   311 371         SUB  SIZ+1
0685  357 000   202             IFT  Z
0686  360 000   040             HLT        JST TO PAGE ZERO DID NOT
0687  361 000   040             HLT        STORE PAGE RET ADR
0688  362 000   371 342         IMS  PZS
```



```
0689  364 000   371 342         IMS  PZS
0690  366 000   347 342        *JMP  PZS
0691  370 000   304 003  SIZ   DPT  ONW
A 0692  372 000   000 000         END
```

Index

A

Accumulator register, 151
Add, 386
Adder, 107, 148, 179
Adder feeder register, 151
Adder unit, 327
Addition, binary, 38
Addition, octal, 43
Addition-subtraction subsystems, 171
Addressing, 329
Add-subtract subsystems, 179
AND gate, 69, 74, 100
AND gate truth table, 73
AND skip instructions, 271
AND to A, 392
A register, 264, 321
A register change, 275, 374
Arithmetic shift, 278, 382
ASCII code, 45
Associative memory, 237
Asynchronous 4-bit counter, 140
Asynchronous operation, 119

B

Basic Laws and Common Identities, Boolean, 83
BCD counter, 146
BCD-to-decimal decoder, 148
Binary addition, 38
Binary coded decimal, 46
Binary coded hexidecimal, 46
Binary coded octal, 46
Binary ripple counters, 235
Binary system, 17
Binary-to-octal conversion, 26
Biquinary code, 49
Bistable latch, 117
Block diagram, 8
Block I-O instructions, 289
Block pointer instructions, 290
Block transfer instructions, 289, 446
Boolean function generator, 157
Boolean Laws, 74
Bootstrap load routine, 26
Bootstrap loader, 15
Buffer, 231

C

Carry gate, 151
Cascading counters, 217
CDC flip-flop, 119
Clear switch, 293
Clocked flip flop, 120
Clocked operation, 131
Code conversion, 191
Code table, 52
Coincidence gate, 165
Commutative and Associative Laws, Boolean, 83
Complement arithmetic, 39
Complement generator, 218
Complementary Laws, Boolean, 83
Conditional complementor, 179
Console, PDC208 and 808 control, 292
Control bus, 301
Control console, PDC208 and 808, 292
Control Data 1604 logic card, 97
Control instructions, 281
Counting circuits, 140
Cycle switch, 293
Conversion, number systems, 25
Counter, 123

D

Data bus, 298
Data, definition, 13
Data entry switches, 293
DCTL, 104
Decade decoder, 187
Decimal system, 17
Decimal-to-BCD encoder, 146
Decoder, 148
DeMorgan's Theorem, 86
Device addresses, 419
D flip-flop, 129
Digital multiplexers, 153
Diode gating circuits, 100
Diode transistor logic, 104
Direct address, 329
Direct-coupled transistor logic, 104
Distributive Law, 86
Double Negative Law, Boolean, 83
Double rank, 120
Double rank register, 126
Double-ward instructions, 365
Down counter, 140
Driver-switch, 339
DTL, 104
Dual rank, 120

E

Electrical memory, 8
Encoders, 146
End-around-carry, 42
Equality comparator, 165
Equations, logic, 78
Error-correcting code generator, 208
Even parity checker, 179
Even parity generator, 174
Excess 3 code, 50
Exclusive OR gate, 165
Expansion, multiplexer, 153
External control command, 303
External control instruction, 285, 425

F

Fast adder, 253
Flip-flops, 117
Four-bit counter, 140
Frequency multiplexer, 225
Frequency selector, 157
Full adder, 107, 151, 242
Function bus, 301
Function index, 314
Functional rank, 120

G

Gray code, 50
Gray to BCD code conversion, 191

H

Half address, 151
Hexidecimal conversions, 36
High-speed buffer, 231
High-speed equality comparator, 172
High-speed paper tape punch, 295
High-speed reader, 295
Hollerith code, 46

I

IC construction, 107
ICs, 103
Idempotent Law, Boolean, 83
Inclusive OR, 388
Increment memory and skip on zero result, 395
Identity Laws, Boolean, 83
Indirect addressing, 338
Inhibit driver, 339
Inhibit regulator, 348
Input command, 303
Input-output instructions, 283
Input to A instructions, 285
Input to A register, 429
Instruction addressing, 329
Integrated circuits, 103
Interchange A with memory, 392
Interrupt logic, 360
Interrupt timing, 303

Intersection Law, Boolean, 86
Inverter, 74, 97
I-O instructions, 283
I register, 264, 323

J

J-K flip-flop, 129
Jump, 393
Jump and store, 395

K

Karnough maps, 161
Keyboard, teletype, 61

L

Laws of Identity, Boolean, 83
L counter, 264
Left-right shift register, 204
Load, 385
Location counter, 326
Logic diagrams,
MC846, SN15846, 308
MC862, SN15862, 309
MC830, SN15830, 309
MC832, 15832, 309
MC844, 309
MC836, SN15836, 310
SN7400, 310
SN7410, 311
SN7420, 311
SN7440, 311
SN7430, 312
SN7473, 312
SN7476, 313
SN7475, 313
SN7483, 314
Logical equations, 78
Logical product, 78
Logical shift, 378
Logical shift instruction, 278
Logical sum, 78
L register, 326

M

Magnetic memory, 8
Magnitude and sign generator, 220
Mainframe options, 296
Manual execute, 294
Master rank, 120
Memory, 8, 339
Memory addressing, 265
Memory disable, 294
Memory element, 228
Memory reference, 385
Memory reference instructions, 267
Memory regulator, 348
ME register, 261
Modulus of the register, 41
Monostable multivibrator, 136
M register, 261, 327
M register extension, 327
MS1 gating arrays, 165
Multiple-emitter transistor, 109
Multiplexers, 182

N

NAND gate, 69, 104, 107
NAND gate latch, 117
N-bit parallel adder, 255
N-bit parallel subtractor, 257
N counter, 205
Negative logic, 73
NOR gate, 69, 103
NOR latch flip-flop, 97
NOR latch flip flop, 97
NOT gate, 69

O

Octal arithmetic, 43
Octal decoder, 187
Octal-to-binary conversion, 25
Octal-to-decimal conversion, 25
Octal system, 17
Odd parity checker, 179
Odd parity generator, 174
One-bit counter, 126
One-out-of-N decoding, 225
One-shot, 136
Optical reader, 295
OR gate, 69, 74, 102
OR gate truth table, 71
OR skip instructions, 273
Output command, 301
Output from A instructions, 287
Output from A register, 436
Overflow, 41
Overflow indicator, 293
Overflow logic, 359
Overflow register change, 277, 377

P

Page counter, 326
Paper tape code, 45
Paper tape punch, 56, 295
Paper tape punch instructions, 422
Paper tape reader, 61
Paper tape reader instructions, 421
Parallel adder, 152
Parallel adder-subtractor, 182
Parity tester, generator, 173
Party line I-O bus, 298
P counter, 264
Peripheral address bus, 298
Peripheral equipment, 295
Polynomial expansion, 24
Positional notation, 23
Positive logic, 73
Power fail-restart, 296
P register, 326
Printer, teletype, 64
Priority interrupt system, 291
Processor, 321
Processor instruction diagnostic, 463
Processor memory diagnostic, 452
Processor teletype diagnostic, 458
Product of sums, 78
Program, definition, 13
Programmable frequency divider, 225
Pull-up resistor determination, 167
Pulse stretcher, 136
Pulse width discriminator, 135
Punch, 295
Punch, paper tape, 56
Push down stack, 235

Q

Quad equality comparator, 165
Quad exclusive OR gate, 165
Quadlatch, 124

R

Racing, 126
R bus, 328
Reader, 295
Reader, paper tape, 61
Real time clock, 296
Real time operation, 13
Register change, 275, 370
Register display, 293
Register select switches, 293
Relative switch, 293
Reserved memory location, 297
Resistor transistor logic, 103
Right shift register, 198
Ring circuit, 199
Ring counter, 204
Ripple adder system, 247
Ripple counter, 140
Rotate, 381
Rotate instruction, 278
R register, 264, 325
RS flip-flop, 121
RST flip-flop, 128
RTL, 103

S

S bus, 328
Schmidt trigger, 136
Scratch pad memory, 157, 231
Sense command, 303
Sense instructions, 284, 423
Sense switch, 294
Serial adder, 151
Shift instructions, 277
Shift registers, 125, 137, 23
Single rank flip-flop, 131
Single rank register stage, 120
Single-word instructions, 36
Skip, 397
Skip instructions, 271
Source control, 13
Source data, 13
Source material translator, 14
Step indicator, 293

Step-run switch, 293
Storage rank, 120
Store, 385
Subtract, 390
Subtraction, binary, 44
Subtraction, octal, 43
Subtractor-adder, 182
Sum of products, 78
Switch-bounce eliminator, 131
Synchronous counter decoding, 222
Synchronous operation, 131
Synchronous up counter, 140
Synchronous up-down counters, 210

T

Teletype input-output, 295
Teletype instructions, 420
Teletype interface, 349
T flip-flop, 129
Three-bit counter, 146
Timing cycle, 328
Timing, processor, 364
Toggle flip-flop, 129
Transfer line, 120
Transistor-transistor logic, 105
Transmission buses, 328
Truth tables, 71
TTL IC layout, 109
TTL input transistor layout, 110
TTL logic, 105
TTL output circuits, 113
Two-input AND gate, 100

U

Up counter, 140
Up-down counter, 146, 210
Union Law, Boolean, 86
Unweighted codes, 49

V

Variable modulus counter, 225
Veitch diagrams, 90

W

Weighted codes, 49
Weighting, number systems, 24
Word length, 39
Worst-pattern diagnostic, 456
W register, 261, 326