UNIVAC

DATA PROCESSING DIVISION

9200/9300

SYSTEMS

PRELIMINARY HARDWARE

REFERENCE MANUAL

For Internal Use Only

UP-4122

This document is provisional in nature and is intended as a vehicle for meeting immediate needs with regard to system familiarization and orientation. UNIVAC ® Division of Sperry Rand Corporation reserves the right to change and/or modify such information contained herein as may be required by subsequent system developments.

PRINTED IN U.S.A.

UNIVAC 9200/9300 HARDWARE REFERENCE MANUAL

TABLE OF CONTENTS

	Page
Organization of the UNIVAC 9200/9300	1
Program Instructions - General	8
Binary Instructions	11
Logical Instructions	13
Decimal Instructions	18
Branch Instructions	20
Expanded Instruction Repertoire (Multiply, Divide, Edit)	22
State Control Instructions	25
Input/Output Control	27
General Purpose Input/Output Channel	30
Basic Peripherals	36
Peripheral Control	40
Control Panel	48
Operation Description	55
Physical Characteristics	56
Instruction Repertoire Summary	57
Multiply, Divide and Edit Timing	Appendix A

ORGANIZATION OF THE UNIVAC 9200/9300

The Central Processor Unit consists of four operational sections which are described below.

MAIN STORAGE:

The main storage of the UNIVAC 9200 is characterized by 1.2 microsecond cycle time while the UNIVAC 9300 has a 0.6 microsecond cycle time. Each addressable position of main storage consists of a nine bit unit called a byte. The bit combination within a byte can represent alphabetic, numeric (decimal), binary or logical data. The main storage is used to store data received from input peripherals, results of processed data, data to be distributed to output peripherals, program instructions and control information.

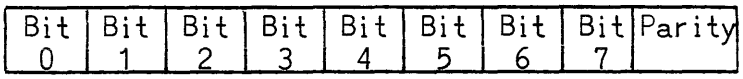


Figure 1. Byte Representation

Storage Boundaries:

Locations in main storage memory are numbered consecutively from 0 through available memory. The maximum address of the UNIVAC 9200 is 16,383 and for the UNIVAC 9300, 32,767. Bytes may be addressed separately or in groups. The address of a group of bytes is the left most byte of the group. Two consecutive bytes constitute a halfword if the most significant byte starts at an even address such as 0, 2, 4, etc.

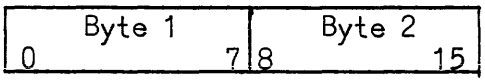


Figure 2. Halfword Representation

Instructions use two halfword (4 bytes) or three halfword (6 bytes) representations and are restricted to halfword boundaries. For instructions which specify a fixed length field, the second operand is restricted to halfword boundaries. If the field lengths are variable, boundary restriotions do not apply. All fixed or variable length operands are addressed at the most significant byte position regardless of instruction processing, i.e., right to left or left to right. For instructions with variable length operands, the length is specified as a binary number which is 1 less than the count; thus a length count of 1 is expressed by zero.

NOTE: Although the hardware length is expressed as one less than the actual count, the Assembler allows the programmer to specify the exact field length he desires.

Parity Checking:

Each 9-bit byte of memory uses the ninth bit to store odd parity. Odd parity is derived by summing the number of 1 bits within the other 8-bits. If the sum is even, the parity bit value is made a 1. If the sum is already odd, the parity bit value is made a 0. This parity check bit is checked as data are read out of memory and is regenerated when data are altered as result of processing. Parity is also generated when data are introduced into the processor from peripherals or other sources that do not already have a parity bit included in their data format. In other parts of this description of the central processing unit the parity bit will not be mentioned except as the source of a possible error condition. If a processor parity error that is detected during the transfer of data to and from the peripherals occurs, it is considered a peripheral parity error and will be covered in the input/output error section.

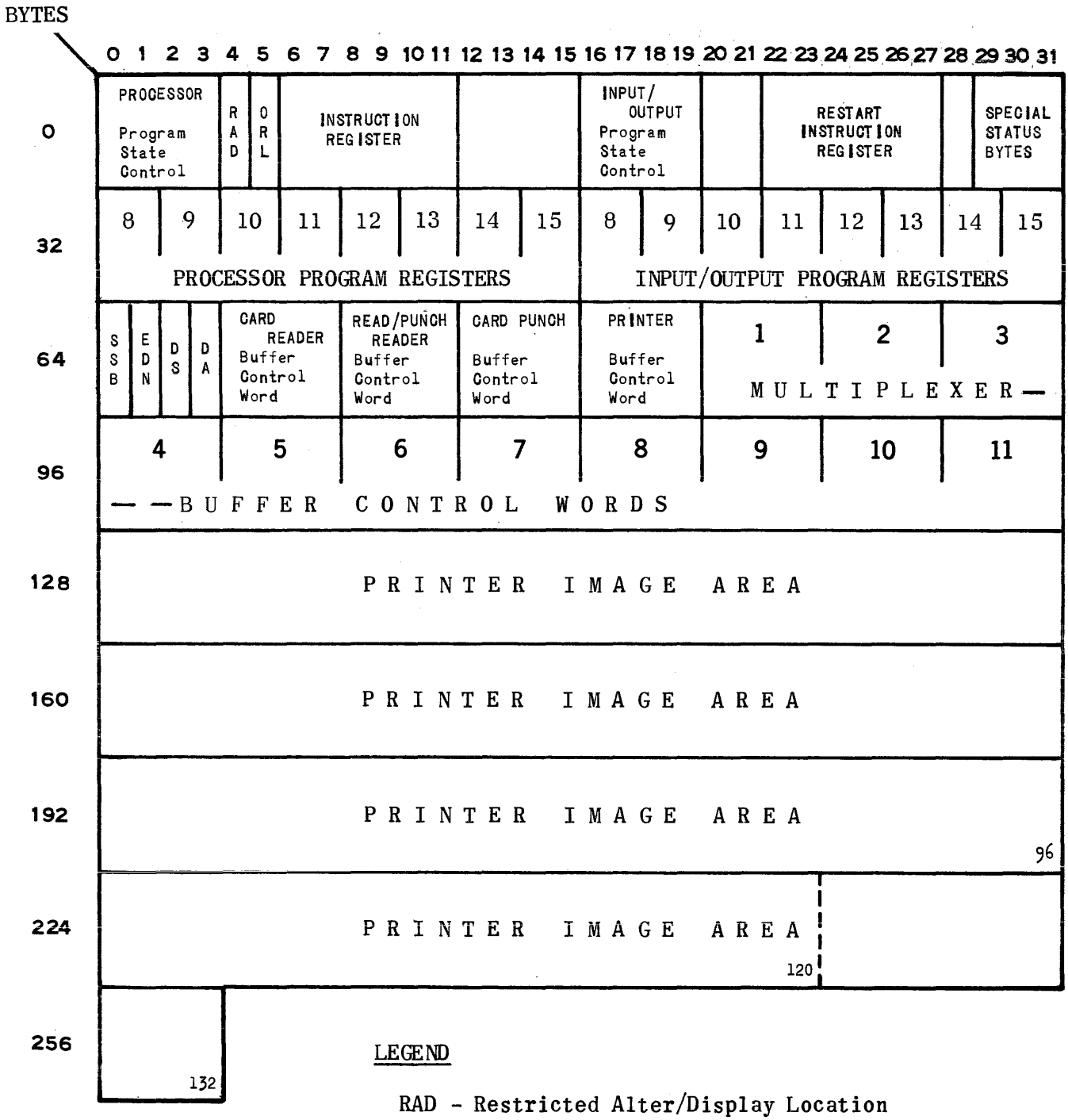
MEMORY ORGANIZATION:

The following memory map (Figure 4) shows the specifically assigned memory areas of the first 260 bytes of memory.

Processor State Control:

Associated with the UNIVAC 9200/9300 are two types of instructions, an instruction set for use by the programmer and a state control instruction set. State control instructions are concerned with changing the Program State Control (PSC). The PSC words have the following organization (Figure 5).

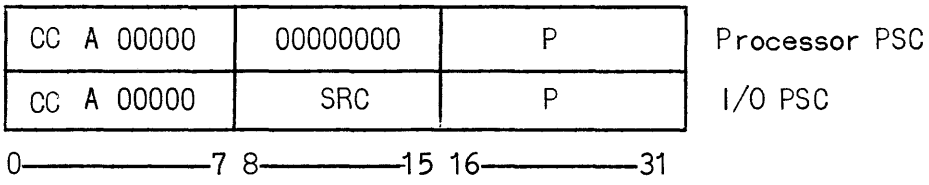
There are two program states: Processor and I/O. In the Processor PSC, interrupt is never inhibited; in the I/O PSC, interrupt is always inhibited. Also, in the I/O state restricted memory locations 0 through 63 can be accessed without a processor abnormal error being generated. There are two banks of General Purpose registers, one associated with each PSC. Depending upon which PSC the program is operating within, designations refer to only one of the two sets of registers.



LEGEND

RAD - Restricted Alter/Display Location
ORL - Operator Request Location
SSB - Special Status Byte
EDN - Error Device Number
DS -- Device Status
DA -- Device Address

Figure 4



CC = Condition Codes
A = ASCII control code
1 for ASCII
0 for EBCDIC
00000 = Unspecified
SRC = Supervisor Request Call - (I/O PSC only)
P = Program address

Figure 5

CONTROL SECTION:

The control section controls the sequencing of instructions and interprets and controls the execution of each individual instruction. The cycling of main storage is initiated by this section. The control section also handles all interrupt handling and error checking. The control section also maintains the program address location counter and handles the different processor mode of operation.

ARITHMETIC SECTION:

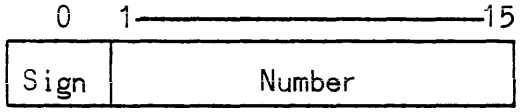
The arithmetic section performs all data manipulations including logical and numerical arithmetic, data comparison and shifting. This section also performs single or double indexing of operand addresses. The adder in this section performs arithmetic in two's complement form.

Sixteen arithmetic registers each two bytes wide are used for arithmetic and indexing. The registers are divided into two sets of eight registers. One set is associated with the Processor Program State Control, and the other associated with the Input/Output Program State Control. The arithmetic registers are specified by address 8 through 15. Addresses 0_8 through 7_8 are illegal.

There are three classes of arithmetic operation in the UNIVAC 9200/9300.

Fixed-Point Arithmetic:

A fixed-point arithmetic operand is a 16 bit binary value (halfword). The most significant bit represents the sign, and the least significant 15 bits contain the binary number. Signed binary arithmetic is only performed within the boundaries of the arithmetic registers.



Fixed Point Representation

Decimal Arithmetic:

Decimal numbers are variable in length and exist in two formats, unpacked decimal numbers and packed decimal numbers. The unpacked decimal format is divided between zones and digits within a byte. The zone is represented in the most significant 4 bits, the digit in the least significant 4 bits.

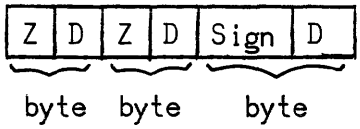


Figure 6. Unpacked Decimal Format

Where a decimal number field is represented in unpacked format, the sign of the number is in the zone position of the least significant digit.

The packed decimal format contains decimal digits in the 4 most significant and the 4 least significant bits of each byte. The least significant byte of the field contains the sign in the 4 least significant bits.

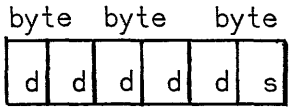


Figure 7. Packed Decimal Format

Sign Bits:

The binary value of the sign bits will be interpreted as follows:

0000 through 1000 - negative number
1001 - negative number
1010 - positive number - generated by instructions in ASCII mode
1011 - negative number - generated by instructions in ASCII mode
1100 - positive number - generated by instructions in EBCDIC mode
1101 - negative number - generated by instructions in EBCDIC mode
1110 - positive number
1111 - positive number

Logical Operations:

Comparing, translating, editing, bit setting and testing are provided by the arithmetic section. Logical operations may be performed on fixed length and variable length operands. Logical operations on fixed length operands are performed in the registers and logical operations on variable length operands are performed in main storage memory. The determination of where the logical operation is performed is made by the instruction format.

INPUT/OUTPUT:

The input/output section initiates, directs and monitors the transfer of data between main storage and the peripheral devices. The UNIVAC 9200/9300 interfaces directly with the bar printer, serial reader and serial punch. It is possible to interface with other devices through a General Purpose Channel. (See state control instruction set section and input/output section.)

The speed of the processor allows memory time-sharing. Memory time-sharing makes it possible for programs to share memory access with many peripherals without distorting the sequence of instructions while data are transferred to and from various peripheral devices. Memory sharing makes it possible for all peripherals to perform at or near their rates speeds without undue processing delay.

Input/Output Interrupt:

An I/O interrupt is permitted only at the end of a program instruction in the Processor Program State Control. As each instruction ends in the Proc. PSC, the peripheral interrupt request line is examined. If the request line reveals a pending interrupt request, interrupt is granted. When interrupt is granted, program control is shifted to the I/O Program State Control (I/O PSC). Also, the device address and status are stored in fixed locations in memory. For a more detailed explanation of PSC operations see sections on state control instructions and input/output control.

ERROR CHECKING:

The processor will detect certain programming errors and parity errors and stop.

Programming Errors:

If the program attempts a read or write function at an address beyond the capacity of a particular processing unit, a processor abnormal will result. While operating in the Processor Program State Control mode, addresses to the restricted memory area (0-63) will cause an address error. Any write into the print buffer during a print scan will also cause an address error.

Parity Errors:

The processor will detect and stop the program on parity errors as described in the section on parity checking.

Divide Check:

A divide check error will result if a quotient digit greater than nine (9) is formed during a divide order.

PROGRAM INSTRUCTIONS

The instructions process fixed length binary numbers, variable length decimal numbers, fixed and variable length logical data, and direct peripheral devices. To perform these functions, three instruction formats are used: RX (register to storage and storage to register), SI (instruction to storage), and SS (storage to storage).

INSTRUCTION FORMATS:

RX Instruction Format:

The RX instructions consist of 4 bytes. They all reference one of the operating halfword registers. These registers may be used to load, store, compare, add, subtract, or branch; and they also may be used as index registers for the RX and other instructions. The instruction format is shown below. Bits 12 through 15 are not used and will be all zeros.

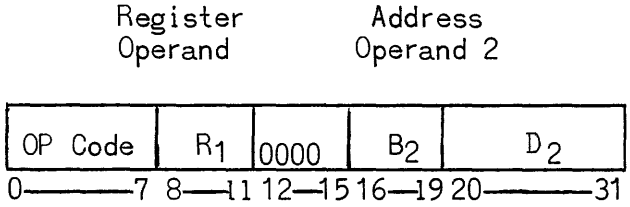


Figure 8. RX Instruction Format

SI Instruction Format:

The SI instructions consist of 4 bytes. These instructions contain one operand from the instruction (immediate data) and another operand from memory and are used for testing or storing immediate data. The instruction format is as follows:

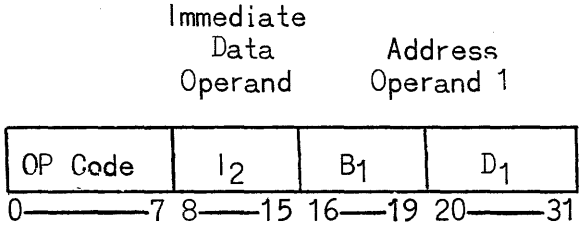


Figure 9. SI Instruction Format

SS Instruction Format:

The SS instructions consist of 6 bytes. These instructions control the transfer of data from storage to storage and are used for decimal operations, translate operations, transfer operations, logical operations, and pack-unpack operations. These instructions have variable length operands, the length is specified as a binary number which is 1 less than the actual count; thus, a length count of 1 is expressed by four or eight zeros. The instruction format is as follows:

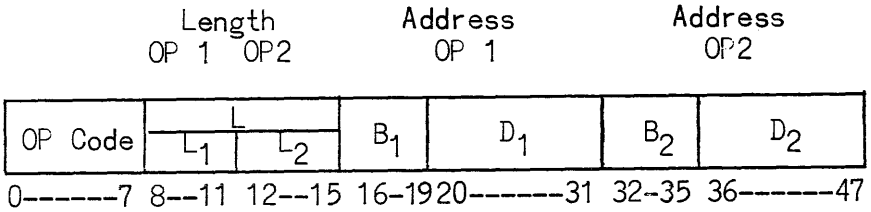


Figure 10. SS Instruction Format

OPERAND ADDRESSING:

An operand may be obtained from memory either by direct addressing or by indexing.

Instructions specify direct addressing by means of an 0 in the most significant bit of the B-field. When the most significant bit of the B-field contains an 0, the three remaining bits of the field are used to extend the D-field by two bits to make a combined direct address instruction field length of 15 bits. A direct address instruction appears as follows:

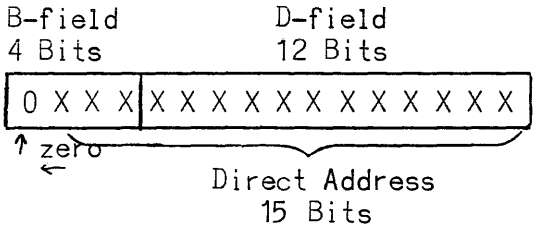
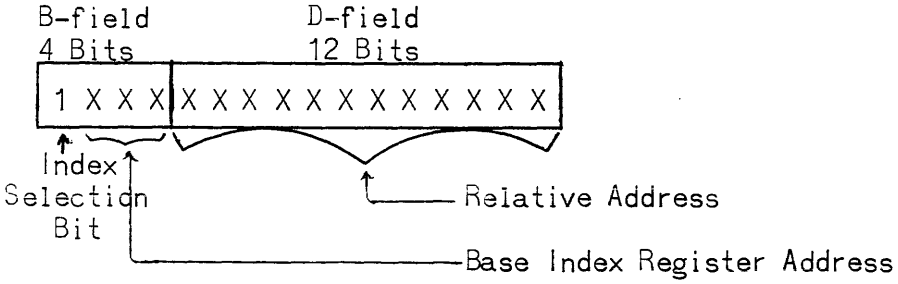


Figure 11. Direct Address Format

Instructions specify indexing by means of a 1 bit in the most significant bit of the B field. When the most significant bit of the B-field is 1, the B-field is used to choose one of the eight binary arithmetic registers. The address portion of any indexed address instruction appears as follows:



The specified register supplies the base address which is added to the 12-bit value contained in the D-field on the instruction. The result of this addition now becomes the effective address of the instruction. Any carry beyond 15 bits that may have occurred will be ignored.

The following table shows the range of valid addresses that apply for either direct or indexed addressing. Unlisted addresses will unconditionally cause an error, but only when the improper access is made. Therefore, certain instructions which do not have any operand cycles, may have non-valid addresses without giving an error. For example, the direct address, or the calculated indexed address at a Halt and Proceed instruction, which becomes the display information, may have any desired binary value without causing an error condition.

0	----	63	Restricted memory
64	----	8191	Permissible
8192	----	12287	Permissible if the memory is 12K, 16K(or 32K on 9300)
12288	----	16383	Permissible if the memory is 16K(or 32K on 9300)
16384	----	32767	Permissible if the memory is 32K

The addressing of restricted memory under Processor PSC Control results in an error stop. Restricted memory may, however, be addressed under I/O PSC Control without an error stop. I/O devices may read out data from restricted memory without an error stop, however, only during initial load may an I/O device write into restricted memory.

The memory locations 128 to 259 inclusive are reserved for the print buffer. Any attempt to write into the first 128 of these locations during a print operation (excluding paper spacing) will also result in an addressing error.

OP CODES:

The OP Code designations are expressed by two hexadecimal notations for the 8 bit code as shown in Figure 12.

Binary	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Figure 12

BINARY INSTRUCTIONS:

The binary instructions that control the use of the arithmetic registers are:

1. Store halfword
2. Load halfword
3. Add halfword
4. Subtract halfword
5. Compare halfword

The following table shows some of the binary values that can be expressed in the halfword registers or their correspondent halfwords in memory.

Binary Number	Decimal Equivalent	Bit Representation
$2^{15}-1$	32767	0111 1111 1111 1111
2^{8}	256	0000 0001 0000 0000
2^{0}	1	0000 0000 0000 0001
0	0	0000 0000 0000 0000
-2^{0}	-1	1111 1111 1111 1111
-2^{1}	-2	1111 1111 1111 1110
-2^{8}	-256	1111 1111 0000 0000
$-2^{15}+1$	-32767	1000 0000 0000 0001
-2^{15}	-32768	1000 0000 0000 0000

Binary numbers in the registers are expressed as positive and negative numbers. A zero in the most significant bit indicates a positive number. A one in the most significant bit indicates a negative number. Negative numbers are expressed by the 2's complement value of the number. Thus, a plus 1 as shown in the table has a 1 in the least significant bit and the remaining bits are all 0's. For a minus 1, all 16 bits are 1's.

CONDITION INDICATORS:

Associated with the Program State Control are condition codes. These condition codes are used to indicate the result of the last instruction performed if they were employed by the instruction. There are four mutually exclusive settings of these codes. The codes are stored in the Program State Control words as follows:

CC 0 = 00
CC 1 = 01
CC 2 = 10
CC 3 = 11

The following table shows the results of addition, subtraction, and comparison.

	Condition Indicator Setting			
	0	1	2	3
Compare Halfword	equal	low	high	---
Subtract Halfword	zero	<zero	>zero	overflow
Add Halfword	zero	<zero	>zero	overflow

The binary instructions all use the RX format, which is as follows:

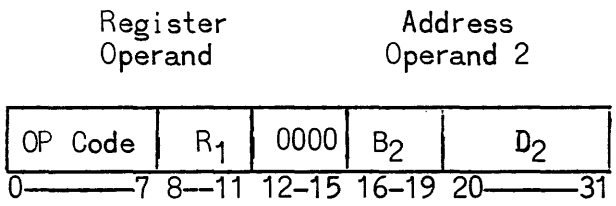


Figure 13. RX Instruction Format

In this format, R_1 specifies the arithmetic register containing operand 1, and B_2-D_2 specifies the second operand address either indexed or direct with an implied length of a halfword.

Store Halfword: STH OP Code 40

This instruction stores the contents of the specified arithmetic register at the specified memory address.

Load Halfword: LH OP Code 48

This instruction loads the arithmetic register with the halfword specified by the operand 2 address.

Add Halfword: AH OP Code AA

The halfword operand in memory is added to the contents of the specified register. Several examples of binary arithmetic and their resulting condition codes are as follows:

				CC	
R	OP1	0000000101010101	+341	0	(Zero)
B-D	OP2	1111111010101011	+-341		
	R	0000000000000000	+ 0		
R	OP1	1111111010101011	-341	1	(Positive)
B-D	OP2	0000000101010101	-+341		
	R	1111110101010110	-682		
R	OP1	0000000101010101	+341	2	(Negative)
B-D	OP2	0000000101010101	++341		
	R	0000001010101010	+682		
R	OP1	0111111111111111	+32767	3	(Overflow)
B-D	OP2	0000000000000001	+ 1		
	R	1000000000000000	-32768		
		carry			

Subtract Halfword: SH OP Code AB

This instruction is performed in the same manner as the add halfword instruction except that the 2's complement of the halfword operand 2 field is added to the specified register. The preceding examples show some results of binary addition and subtraction.

Compare Halfword: CH OP Code 49

This instruction algebraically compares the contents of the designated register with the designated halfword in memory. The results of the comparison are shown in the condition code table.

Binary Overflow:

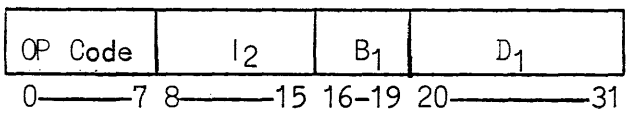
Binary overflow occurs when the sign is changed erroneously from plus to minus or minus to plus, which happens when an attempt is made to develop a larger negative or positive number than can be expressed in 15 bits.

LOGICAL INSTRUCTIONS:

The following instructions deal with the logical manipulation of data and are not necessarily related in function to binary or decimal arithmetic.

Logical Instructions Using the SI Format:

The SI format is employed in certain logical instructions.



In these instructions the I portion of the instruction is used either for an immediate second operand or it is ignored. The B and D-fields are normally used to specify the first operand address. The instructions using SI format are:

1. Halt and Proceed
2. Test Under Mask
3. AND
4. OR
5. Compare Immediate
6. Move Immediate
7. Add Immediate

Halt and Proceed HPR OP Code A9

This instruction causes the processor to stop and an indication to be set on the control panel. The B_1 and D_1 part of the instruction can be used to display the reason for the halt. If bit 16 = 0, bits 17 through 31 of the instruction word will be displayed. If bit 16 = 1, normal indexing will result and the value of the indexed address will be displayed. Depressing start will cause the next instruction in sequence to be executed.

Test Under Mask TM OP Code 91

This instruction uses the (I_2) immediate operand bits that are 1's to test for equivalent 1's in the byte of memory specified by B_1 and D_1. If all the 1 bits in the mask are matched against 0 bits in the byte tested, the condition code will be 0. If some of the 1 bits in the mask have corresponding 1 bits in the byte tested, the condition code is 1. If all the 1 bits in the mask have corresponding 1 bits in the byte tested, the condition code is 3. If the mask is 0, the resultant condition code setting will also be 0.

AND NI OP Code 94

This instruction forms a logical product (AND) of the I_2 bits of the instruction with one byte of memory specified by the B-D field. If the corresponding bits of I_2 and the one byte of memory are both 1's the result will be 1, otherwise the result will be zero. If all the bits of the memory byte are 0's at the end of the instruction, the condition code will be set to 0. If all or some of the bits of the memory byte are 1's, the condition code is set to 1.

OR OI OP Code 96

This instruction forms the logical sum (OR) of the I_2 bits of the instruction with one byte of memory specified by the B-D field. All 1 bits of the I_2 field are superimposed onto the memory byte. All 0 bits of the I_2 field leave the corresponding bits in the memory byte unchanged. If all bits of the memory byte are 0 at the end of the instruction, the condition code is set to 0. If all or some of the bits of the memory byte are 1, the condition code is set to 1.

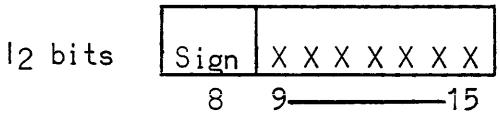
Compare Immediate CLI OP Code 95

This instruction compares the contents of the I_2 field of the instruction to the B-D byte of memory. A full 8 bits are compared without regard to sign. If the I_2 value is equal to the compared byte of memory, the condition code is set to 0. If the byte of memory is less than the I_2 value, the condition indicator is set to 1. If the byte of memory is greater than the I_2 value, the condition code is set to 2.

Move Immediate MVI OP Code 92

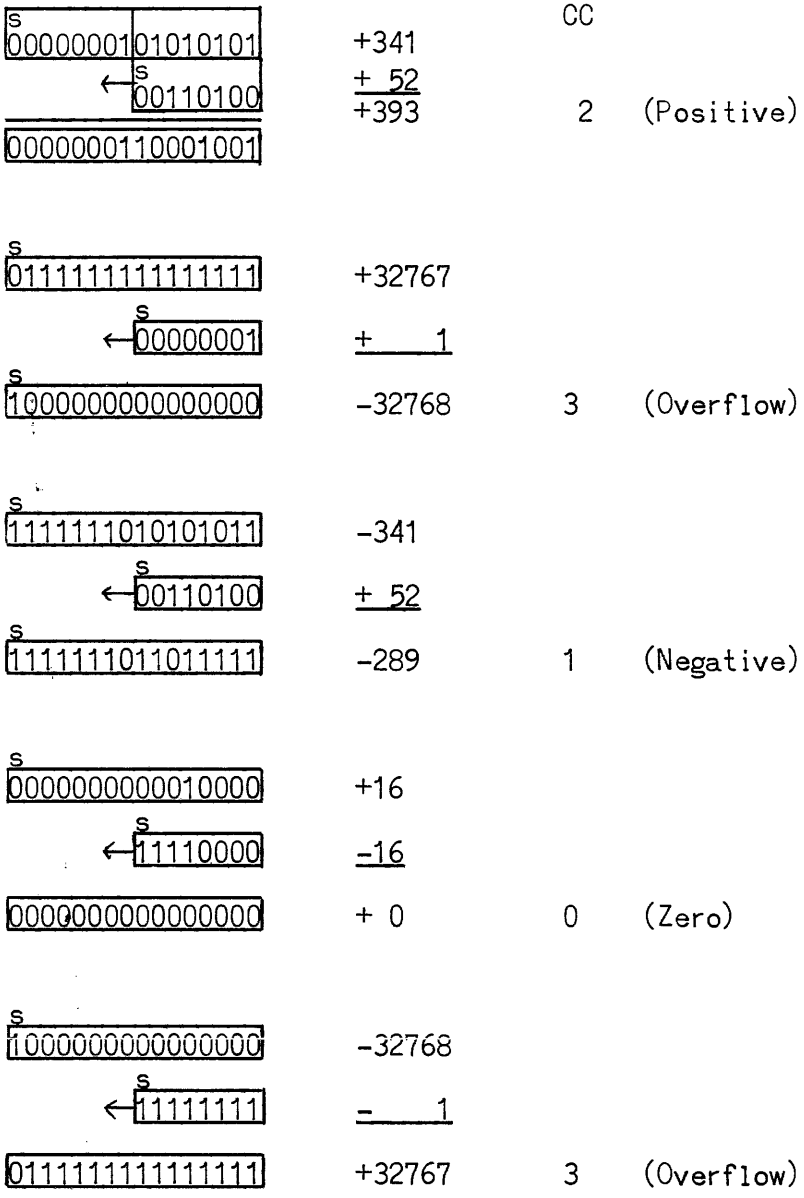
This instruction stores the I_2 value of the instruction at the byte of memory specified by the B-D field.

Add Immediate AI OP Code A6



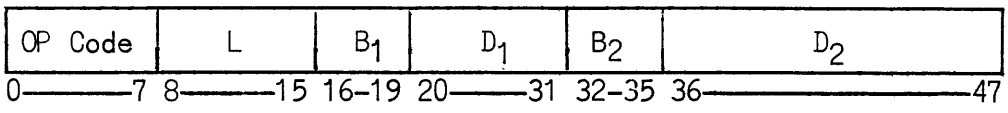
Sign bit 0 is Positive
Sign bit 1 is Negative

This instruction adds the I_2 bits of the instruction to the halfword in memory specified by B_1-D_1. The sign of the I_2 value is extended from the first byte of the halfword in memory through the second byte of the halfword. The condition code will be set to 0 at the end of the instruction if the answer is zero; to 1 if the answer is minus; to 2 if the answer is plus; and to 3 if an overflow condition causes the sign to be erroneously changed. The following table shows the results of some sample examples of binary addition via this instruction.



Logical Instructions Using the SS Format:

These instructions use the SS format to manipulate data logically in one area of storage against data in another area of storage or to transfer data from one storage area to another storage area. The L field in these instructions is 8 bits in length and governs both operands. The L field can specify from 1 to 256 characters.



The instructions which are covered under this category are:

1. Move Characters
2. Compare Logical
3. Translate
4. AND
5. OR
6. Move Numeric

Move Characters: MVC OP Code D2

In this instruction, L characters can be transferred from the memory address specified by B_2-D_2 to the memory address specified by B_1-D_1. Characters are unaltered in the transfer and are not checked for any validity except for parity. The instruction is processed from left to right.

Compare Logical: CLC OP Code D5

In this instruction, L characters of operand 1 and operand 2 are compared with each other. The results of the comparison are shown by the condition indicator. If both operands are equal, the condition indicator is set to 0. If operand 1 is less than operand 2, the condition code is set to 1. If operand 1 is greater than operand 2, the condition code is set to 2. The instruction is processed from left to right and is terminated early upon the first inequality.

Translate: TR OP Code DC

This instruction translates any 8-bit code to any other 8-bit code. Each 8-bit code of the desired field is translated in sequence from left to right. Each 8-bit code of the first operand is used as an argument to increment the address specified by B_2-D_2 so that the derived address can be used to obtain from memory the translated code value which replaces the character to be translated. The translation table, unless restricted by program operation (operand 1 character values), is 256 characters in capacity. The operation proceeds until L + 1 characters have been translated.

AND: NC OP Code D4

This instruction forms a logical product (AND) with L + 1 characters of operand 2 and operand 1. If the corresponding bits of both operands are 1, the result is 1; otherwise, the result is 0. If all the bits of the logical product in operand 1 are 0, the condition code is set to 0. If all or some of the bits of the logical product in operand 1 are 1, the condition code is set to 1. This instruction proceeds from left to right.

OR: OC OP Code D6

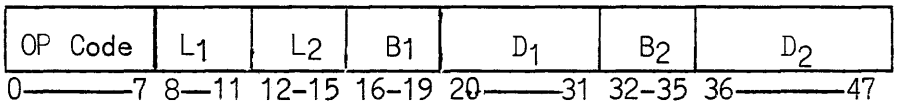
This instruction forms a logical sum (OR) with L characters of operand 2 and operand 1. All 1 bits of operand 2 are superimposed on operand 1. All 0 bits of operand 2 leave the corresponding bits of operand 1 unchanged. If all the bits of operand 1 are 0's at the end of the instruction, the condition code is set to 0. If all or some of the bits in operand 1 are 1's, the condition code is set to 1. This instruction proceeds from left to right.

Move Numeric: MVN OP Code D1

In this instruction, L + 1 numeric-digits minus the zone digits are transferred from the memory address specified by B_2-D_2 to the memory address specified by B_1-D_1. The zone portions of the B_1-D_1 field are unaltered as the numeric portions are replaced. The two operand characters are not checked for validity except for parity. The instruction is processed from left to right.

DECIMAL INSTRUCTIONS:

The following instructions are primarily concerned with decimal calculations or decimal digit manipulations. These instructions all use the SS format.



The following is a list of decimal instructions:

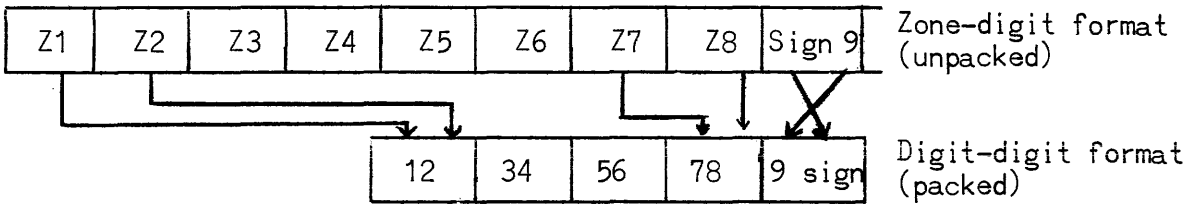
1. Pack
2. Unpack
3. Compare Decimal
4. Add Decimal
5. Subtract Decimal
6. Move with Offset
7. Zero and Add

The SS format used in decimal manipulations splits the L field into four bits for each operand. Thus, the maximum number of bytes that can be handled in any decimal instruction is 16.

The add decimal and subtract decimal instructions should be executed using operands that are in the packed format only. They will interpret bits 4 through 7 of the least significant byte of the operand fields as sign codes when performing the arithmetic. The sign code which is inserted into the result in operand 1 will be determined by the sign of the answer and the ASCII Control Code bit in the controlling Program State Control word.

Pack: PACK F2

This instruction packs decimal data which are in the unpacked zone-digit data format to the packed digit-digit data format.



This instruction proceeds from right to left. The first byte of operand 2 is assumed to have the sign of the field to be packed. As the diagram shows, the sign and digit positions are reversed when operand 2 is transferred to the operand 1 location. All other bytes of operand 1 receive two digits from operand 2. If operand 2 is exhausted before operand 1, 0's are assumed for the remaining positions of operand 2. If operand 1 is exhausted before operand 2, the remainder of operand 2 is ignored.

Unpack: UNPK OP Code F3

This instruction is the reverse of the pack instruction. It also proceeds from right to left. The first byte of operand 2 is assumed to contain the sign. This instruction reverses the sign and digit positions when storing the first byte of operand 2 in operand 1. All remaining digits of operand 2 are placed in operand 1 sequentially in the numeric position of operand 1 with 1111 (if binary coded decimal) or 0101 (if ASCII code) placed in the zone portion of operand 1. If operand 1 is too short to contain all the unpacked digits from operand 2, the remaining digits are ignored. If operand 2 is exhausted before operand 1 is filled, 0's are assumed for the remainder of the operand 2 digits to complete the operation.

Compare Decimal: CP OP Code F9

This instruction algebraically compares $L_1 + 1$ bytes of operand 1 with L2 + 1 bytes of operand 2; L_1 must always be equal to or greater than L_2. If L_1 is larger than L_2, 0's are assumed to extend operand 2. The instruction proceeds from right to left, and both operands remain unchanged when the instruction ends. The comparison sets the condition code to 0 for equality, to 1 if operand 1 is low, and to 2 if operand 1 is high.

Add Decimal: AP OP Code FA

This instruction adds decimally $L_2 + 1$ bytes of operand 2 to $L_1 + 1$ bytes of operand 1; L_1 must always be equal to or greater than L_2. Addition is performed algebraically. The condition code will be set to 0 at the end of the instruction if the answer is zero; to 1 if the answer is minus; to 2 if the answer is plus; and to 3 if an overflow beyond the field results from the addition. The sign in the case of overflow will reflect the sign of the true answer even if the answer is zero; otherwise, on a zero answer the sign is always plus. The instructions proceed from right to left, and in the case where operand 2 is shorter than operand 1, 0's are assumed for the remaining digits of operand 2.

Subtract Decimal: SP OP Code FB

This instruction is performed in the same manner as add decimal except that the sign of operand 2 is reversed.

Move with Offset: MVO OP Code F1

This instruction proceeds from right to left. Its function is to transfer $L_2 + 1$ bytes of operand 2 to $L_1 + 1$ bytes of operand 1 and to shift the digits left by four bits. It also preserves the first four least significant bits of the least significant byte of operand 1. This instruction is used to position for later packed addition and subtraction. This instruction can effectively shift data either right or left. If L_2 is less than L_1, 0's will be assumed for operand 2 and will fill the remaining positions of operand 1. If L_2 is greater than L_1, the remaining digits will be ignored. This operation does not affect the condition code, and no check is made for validity of the digits shifted.

Zero and ADD: ZAP OP Code F8

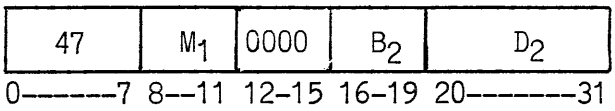
This instruction is essentially a move instruction proceeding from right to left; L_1 must be equal to or greater than L_2. The instruction clears operand 1 to 0's before adding operand 2. The condition code will reflect the sign and the value of operand 2; however, a -0 cannot result from this operation.

BRANCH OPERATIONS:

The following instructions are primarily concerned with changing the program sequence or branching from one sequence of operations to another. These instructions use the RX format. The following are the branch instructions:

1. Branch on Condition
2. Branch and Link

Branch on Condition: BC OP Code 47

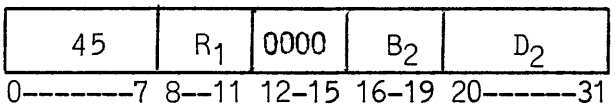


M_1 CC Test

M_1	Condition Indicator Setting
8	0
9	1
10	2
11	3

This instruction uses the four M_1 bits to test none, some, or all of the possible states of the condition indicator. Each bit in the M_1 portion tests the presence of one of the possible four condition settings. If a condition setting is met by the test, a branch will occur. The branch address is determined by a binary value of B_2-D_2 either as a direct address or an indexed address. If M_1 is all 0's, the result is a no-op or skip. If the mask is all 1's, the resultant instruction is an unconditional branch. The instruction does not change the condition indicator setting.

Branch and Link: BAL OP Code 45



This instruction is an unconditional branch instruction. It stores the program address, which points at the next instruction in logical sequence, in the register specified by R_1, and branches to the address specified by B_2-D_2.

SS FORMAT INSTRUCTIONS

EXPANDED INSTRUCTION REPERTOIRE:

The Multiply, Divide and Edit instructions which are described in the three succeeding sections are available as a factory or field installable feature.

Multiply Decimal: MP OP Code FC

This instruction multiplies OP1 by OP2 placing the product in OP1. $L_1 + 1$ specifies the size of the product and $L_2 + 1$ is the size of the multiplier. The size of the multiplier originally in OP1 equals L_1 minus L_2. If more than L_1-L_2 digits are in the multiplicand, the most significant digits will be ignored.

The length of the operands are expressed in numbers of bytes of packed decimal data. The condition codes will not be changed by this instruction. The sign of the product is determined by the normal rules of arithmetic.

Programming Restrictions:

A. The multiplier and multiplicand field may not overlap.

B. OP1 must contain at least $L_2 + 1$ leading decimal zeros prior to multiplying.

C. The sign position of OP1 (multiplier) must contain a value greater than 9.

In addition to obtaining an incorrect result, memory fields beyond OP1 may be altered if the above restrictions are not met. No error indicator will be set.

Divide Decimal: DV OP Code FD

The divide instruction divides the $L_1 + 1$ bytes of OP1 by the $L_2 + 1$ bytes of OP2, developing the Quotient right justified in the L_1 minus L_2 most significant bytes of OP1. The remainder is right justified in the $L_2 + 1$ least significant bytes of OP1.

The operand field lengths are specified in numbers of bytes of packed decimal data. The sign of the Quotient is determined by the normal rules of arithmetic. The sign of the remainder is the same as the original OP1 sign (dividend).

Programming Restrictions:

A. L_1 must be greater than L_2.

B. The sign of OP1 (dividend) must contain a value greater than 9.

C. The divisor and dividend fields may not overlap.

In addition to an incorrect result, memory fields beyond OP1 may be altered or a Divide Check error may occur, if the above restrictions are not met.

Divide Check Error:

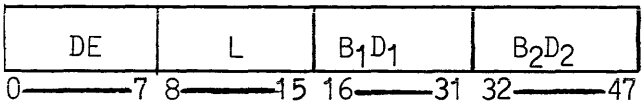
If a Quotient digit greater than nine is formed, a divide check error will occur and the processor will stop immediately.

A quotient digit greater than 9 will occur if the absolute value of OP2 is not greater than the absolute value of the L_2 + 1 most significant bytes of OP1; thus, it follows that OP2 must be non-zero and OP1 must contain at least one most significant zero.

Edit: ED OP Code DE

The edit instruction proceeds from left to right unpacking and expanding OP2 into the L_1 + 1 bytes of OP1. Editing is controlled by a pattern of bytes originally in OP1. The value inserted in the zone position will F_{16} in EBCDIC mode or 5_{16} in ASCII mode.

The instruction format is as follows:



The data from the most significant byte of OP1 is retained and used as a FILL character for the editing that follows. Zero suppression and character insertion are controlled by the Significant (S) Indicator associated with the Edit instruction. Characters in OP2 are unpacked into OP1 or suppressed whenever a Digit Select Byte (DSB) is encountered in any position of the OP1 edit pattern. The DSB is a hexadecimal 20 and operates as described below.

Zero Suppression:

The S Indicator will be set (turned on) whenever a non-zero digit is encountered in OP2. If the S Indicator is set the DSB in OP1 will be replaced by the corresponding digit, including zeros, in OP2. If the S Indicator is not set the DSB will be replaced by the FILL character when the corresponding digit in OP2 is a zero. The S Indicator is in the reset (off) condition when editing begins. Once the S Indicator is set it remains set.

If leading zeros must be retained the S Indicator can be set by a Significant Start Bit (SSB), hexadecimal 21, placed in the edit pattern in the position that is to contain the first leading zero. The S Indicator may be reset (turned off) by a Field Separator Byte (FSB), hexadecimal 22, anywhere in the OP1 pattern. The FSB will itself be replaced by a FILL character.

Character Insertion:

Any character other than DSB, FSB or SSB encountered in OP1 will be retained in OP1 if the S Indicator is set. OP2 will not be accessed. If the S Indicator is reset the values will be replaced by FILL characters.

Any sign accessed in OP2 will be ignored, a positive sign will, however, reset the S Indicator. Therefore, OP2 must contain a number of valid digits (zeros included) equal to the sum of the DSB's and SSB's in OP1. A sign is examined at the time when a digit immediately to its left is accessed. The value of the sign position must exceed hexadecimal 9.

Condition codes are set as follows:

CC set to 00: OP2 is zero
CC set to 01: OP2 is negative or unsigned
CC set to 10: OP2 is positive

In the case of multiple field editing, CC will not reflect any part of OP2 which precedes the last FSB in the OP1 pattern.

STATE CONTROL INSTRUCTIONS

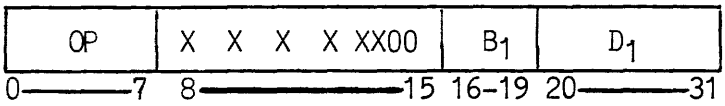
State control instructions for the UNIVAC 9200/9300 are those which alter the Program State Control (PSC).

The privileged instructions use the SI format and are:

1. Load State
2. Store State
3. Supervisor Request Call

PSC OPERATIONS:

Load State: LPSC OP Code A8



This instruction replaces or modifies the Processor or I/O Program State Control Word and directs the instruction sequence to proceed under control of either of the PSC words. The PSC load must have a halfword boundary specified for B_1-D_1. This instruction is also used to restrict the operator Alter and Display functions to memory location 4.

Bits 8 and 9 control the modification of a PSC word as follows:

Bit 8	Bit 9	Modification
0	0	PSC remains unchanged
0	1	Load full PSC word start at B_1-D_1 address (halfword boundary)
1	0	Set ASCII bit to 0 (off); leave remainder of PSC word unchanged
1	1	Set ASCII bit to 1 (on); leave remainder of PSC word unchanged

Bit 10 specifies which PSC word to be acted upon as follows:

bit 10 = 1 Modify or load I/O PSC word.
bit 10 = 0 Modify or load processor PSC word.

Bit 11 specifies the PSC word which will control the next instruction as follows:

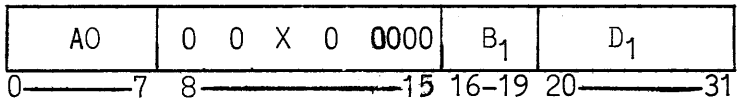
bit 11 = 1 I/O PSC word has control.
bit 11 = 0 Processor PSC word has control.

Bits 12 and 13 control the Alter and Display functions as follows:

bit 12 = 1 Alter and Display will be restricted to location 4.
bit 13 = 1 Remove restriction.

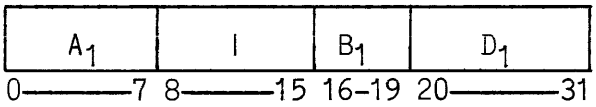
If both bits are zero, the alter state is unchanged. If both are ones, the result is unpredictable.

Store State: SPSC OP Code A0



This instruction will store the PSC word in the memory position designated by B_1-D_1. The PSC that is stored is determined by bit 10 where a 0 specifies processor PSC and a 1 specifies I/O PSC. For storing PSC, the storage area specified by B-D must have a halfword boundary.

Supervisor Request Call: SRC OP Code A1



This instruction stores the I bits in the SRC portion of the I/O PSC and sets interrupt request. Fields B_1-D_1 in this instruction are ignored. Note that the interrupt will be granted immediately when operating in the processor PSC but will be stored if in the I/O PSC. The acceptance of this interrupt request will not cause a channel number or any status information to be stored.

INPUT/OUTPUT CONTROL

There are various sets of memory locations assigned for use with input/output control units. Each set consists of an area of memory (called a Buffer Control Word) which is used for storing data storage addresses and character counts, where applicable, and other details pertaining to the specific input/output function performed.

INPUT/OUTPUT PROGRAM PROCEDURE:

Input/output control requires the following software steps:

1. Load the proper Buffer Control Word with the necessary information required by the control unit, provided the unit is not busy.

2. Issue an input/output instruction which specifies the device address and the function to be performed.

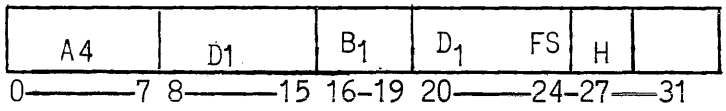
3. Test the Condition Indicator to determine if the instruction was accepted.

4. Test the status of the device when the operation is completed (normally indicated by the generation of an interrupt) to determine if the operation was successful.

Input/output instructions all use the SI format in which the I portion is used to specify the Device Address. The B_1-D_1 portion of the instruction functions in the normal manner, either allowing indexing or direct addressing.

INPUT/OUTPUT INSTRUCTION:

Execute I/O: XIOF OP Code A4



This instruction is used to initiate a function on the device specified by the Device Address (DA). The Buffer Control Word associated with the device must have been loaded with the information required by the particular device. The Function Specification (FS) defines the type of operation to be initiated. Some devices require part of the FS to be placed in the Buffer Control Word. The H bit of the FS(27) is reserved to inhibit the generation of all interrupt requests when the operation ends.

The execution of an XIOF instruction sets the condition codes as follows for the basic peripherals (card reader, printer and punch):

CC = 00 Function accepted

CC = 10 Function rejected

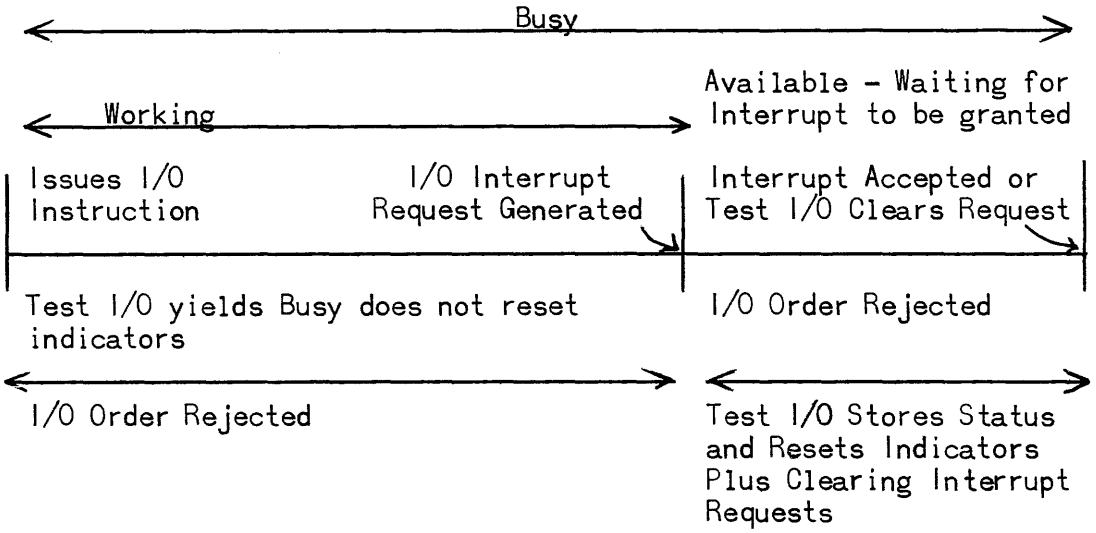
CC = 11 Function rejected - non valid device number

A device is "BUSY" from the time of acceptance of one I/O instruction until an Interrupt Request is granted or is reset by Test I/O. If an XIOF is not accepted, a TIO will store the status to indicate why.

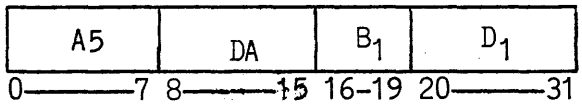
When an interrupt is granted, the indicators of the device are reset, and the status of the device is stored in memory location DS. The device number is stored in memory location DA.

On a Test I/O instruction (TIO) the status byte is stored in the memory location specified by B_1-D_1.

Timing Diagram of a General Peripheral Cycle



Test I/O Status TIO OP Code A5



This instruction tests the peripheral specified by the Device Address. The status of the addressed unit is stored at the address specified by B_1-D_1. This instruction clears the device status storage for the device if it is not working. It should be noted that interrupt request is part of the device status and will be cleared by this instruction. If the device is working, then status will not be reset. The instruction sets the following condition codes on basic peripherals:

CC = 00 Zero Status - device available.
CC = 01 Valid Status - interrupt was pending; device now available.
CC = 10 Busy Status - device not available.
CC = 11 Zero Status - non valid device number.

GENERAL PURPOSE INPUT/OUTPUT CHANNEL

The General Purpose Input/Output Channel feature provides an interface to which devices other than the basic reader, printer and punch may be attached. The channel allows attachment of eight control units, each capable of controlling eight devices. The channel operates at 100,000 bytes per second in a multiplex mode. Each control unit receives input or output servicing for one character time.

The channel accepts I/O instructions from the CPU and generates the initial selection sequence required to control and test control units and their associated devices. The channel will generate the necessary sequences to respond to the various requests of the CPU. The UNIVAC 9200/9300 memory, registers and arithmetic circuits are shared by the General Purpose I/O Channel.

CHANNEL ADDRESSING:

Execute I/O and Test I/O operations:

When execute or test I/O instructions are issued to devices other than the basic peripherals (Device Address $\neq$ 1, 2 or 3) and the channel feature is present, the channel will attempt to execute the Initial Selection Sequence or I/O Command. If the addressed device is off-line or does not exist the channel will reject the command and produce an error condition code.

Subchannels:

A subchannel is required to sustain a data transfer operation concurrently with other data transfer operations. The subchannel is part of the General Purpose Input/Output Multiplexer Channel. The Multiplexer Channel is capable of handling data transfer operations from several concurrently operating subchannels so long as the maximum system data rate is not exceeded and no device is monopolizing the interface by operating in burst mode.

Devices operating in burst mode monopolize the channel facilities by holding the channel active for the duration of an entire Input/Output operation. Once the channel has assumed control over the time shared Central Processor, the UNIVAC 9200/9300 may not be accessed, until the channel is freed. Basic peripherals may transfer data in burst mode but the UNIVAC 9200/9300 processor is effectively locked out during this interval.

BUFFER CONTROL WORDS:

Each Subchannel requires a four-byte Buffer Control Word (BCW) in the main memory. The BCW's contain initial and working data counts, data addresses and control bits. The program must load the proper BCW with the proper initial conditions before issuing any EXECUTE I/O order to any subchannel. Note also that program modifications of the

BCW of a working Subchannel is a delicate matter. Eleven BCW's have been reserved for the channel (Memory locations 84 - 127_{10}). The channel therefore may have eight Subchannels. Note that the channel may also use the BCW's allotted to Basic peripherals if these are not present in some systems.

Device, Control Unit & Subchannel Numbering:

Control Units may be assigned any appropriate address or group of addresses at installation time. The channel recognizes two Device Address formats, one for shared subchannels and one for non-shared subchannels. Devices that share a Control Unit and are not able to transfer data concurrently may share a subchannel. A subchannel may be shared by up to eight devices. A shared Control Unit with more than eight devices would require at least two subchannels.

Devices operating through Non-shared Subchannels have addresses with the binary format:

0000SSSS where SSSS is the subchannel address.

Devices sharing subchannels have addresses with the binary format:

1SSSSdddd where SSSS is the Subchannel number and dddd is a particular device number.

BCW Location:

The location of the most-significant Byte (BC00) of Subchannel N is memory location 64_{10} + 4N, where N = SSSS (000 0000 01SS SS00 in binary).

Status Byte Format:

During Initial Selection, occurring on EXECUTE I/O or TEST I/O, and on termination of I/O operations, Control Units present a STATUS BYTE with the following format:

Bit 0 - Attention
1 - Status Modifier
2 - Control Unit End
3 - Busy
4 - Channel End
5 - Device End
6 - Unit Check
7 - Exception

This status byte is stored in program-specified locations on Test I/O instruction and in location 66_{10} when the channel is allowed to interrupt the program.

Condition Code Summary:

When an EXECUTE I/O or TEST I/O Instruction is issued via the channel the result of the operation is summarized in the Condition Code placed in the appropriate Program State Control area.

CC Code	EXECUTE I/O	TEST I/O
00	Command Accepted No channel error STATUS Byte all zero except for end bits. (Status = 0000,XX00)	Device Available No Channel Error Status Byte = 0000,0000
10	Device, Control Unit or Channel Busy Command Rejected No Channel Error STATUS, if any, all zeros except Busy or Busy & Status Modifier Bits (Status - 0X01,0000)	Device, Control Unit, or Channel Busy. No Channel error. Status, if any, = 0X01,0000
01	STATUS PENDING Command Rejected No Channel Error Status other than Busy or End bits. Status ≠ 0X01,0000 or 0000,XX00	NON-ZERO STATUS STORED and Cleared. No Channel Error Status other than Busy. Status ≠ 0X0X,0000
11	Device Control Unit or Channel NOT OPERATIONAL Probe return or Channel-detected Error	Device, Control, or Channel NOT operational. PROBE RETURN or Channel-detected Error.

Note that the channel may be momentarily busy on rare occasions, having just committed itself to a Service Request sequence as the I/O instruction was being staticized.

Note also that a programming error, namely addressing a non-existent device, can generate the 11 condition code.

Buffer Control Word Formats:

When a Control Unit initiates a sequence in order to request or present data or to present a status byte, the Control Unit presents a Device Address along with appropriate control signals. This address is placed in the channel's Device Address Register where it is used to determine the location of the proper BCW. The action taken by the channel depends upon the contents of the BCW. The normal BCW format is shown below:

	WMT	BYTE COUNT (13 bits)	0	DATA ADDRESS (15 bits)	
	BC00	BC01	BC10	BC11	
Loc.	64+4N	64+4N+1	64+4N+2	64+4N+3	

Basic Format: WM ≠ 11

W - Data Direction Bit

W = 1 for Write (output) or "Buffered" Control operations

W = 0 for Read: (input) operations.

M - Addressing Mode Bit

M = 0 for forward (A + 1 → A) Sequence.

M = 1 for backward (A-1 → A) Sequence.

T - Termination Bit

If T = 1, no data will be transferred, the BCW will not be modified by the channel and the Terminate (Function Out) response will be given to Data Requests.

The Channel will set T = 1 after the transfer of a byte of data causes the Byte Count to go to zero. The channel will <u>not</u> erase a T bit.

Byte Count: This field is decremented by the channel whenever a byte of data is transferred. An initial count of zero gives a block length of 8192 bytes if T = 0.

Note that a Control Unit may terminate an operation before the count becomes zero. On termination this field indicates the difference, if any, between the initial Byte Count and the number of bytes actually transferred.

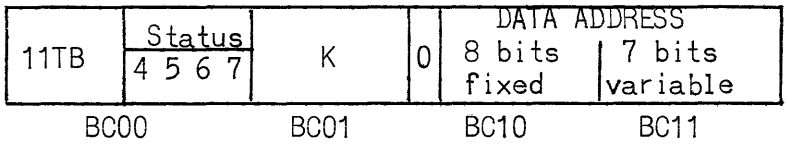
Data Address: This field is fetched by the channel and used as the address for the current byte of data. The address is modified in the BCW under control of the M bit in preparation for the next byte. On termination this field indicates where the next byte would have gone had the operation continued (last address ± 1).

Note the necessity for agreement between the W and M bits and the I/O command initiated via the subchannel.

ALTERNATE BCW FORMAT:

This alternate Format (LT Format) is provided so that the system can handle devices that transfer a continuous stream of data at relatively low speeds, more specifically Communications Line terminals for remote inquiry stations, computer-to-computer transmissions, etc. The main variation is the fixed-length wrap-around buffer addressing sequence which could be called a limited form of data chaining. This operation is defined by program and not by any device characteristic. The LT format, described below, is specified when the W and M bits are both ones, which would specify "write backwards" in the normal format.

LT BCW Format



T - Terminate Bit
If T = 1 no data will be transferred, the Data Address will not be modified, the channel will give the Terminate response to Data Requests. The channel will set T = 1 when "Wrap-around error" occurs (see B Bit below). The channel will not erase a T bit.

B - Buffer End Bit
When the address modification generates a carry from the 2^5 bit position of the data address (when the address is modified to an integral multiple of 64_{10}) the channel sets the B bit to one and generates on LT Summary Interrupt Request. The B bit alerts the program that a 64 byte buffer segment has ended. The program is expected to remove the B bit when the buffer segment is ready for use by the channel again. If the channel finds a B bit remaining in the BCW when the End of Buffer Segment occurs again, the channel sets the T Bit to one so that the data will not be overlaid. This is the "Wrap-Around error" situation. The channel will not erase a B bit.

Data Address: This field contains the address of the next data byte to be transferred. The address modification in the LT format is always A + 1 → A, Mod 128. This sequence, with the B-bit, gives the effect of alternating the use of two adjacent 64-byte buffer areas.

Status Field: When a device operating in the LT mode initiates a sequence to present status, bits 4 - 7 of the Status Byte are merged (OR function) into this field. If the CPU allows the Interrupt the entire status byte is also placed in the Interrupt Entry area and the LT summary

Interrupt Request is reset. If the Interrupt is not allowed the LT Summary Interrupt Request is set.

K Field - Address trap: If a device operating in the LT mode attempts to present status and bits 4 and 5 in the status field of the BCW were previously both zero, the 8 least significant bits of the Data Address are transferred to the K field. If either bit 4 or bit 5, in the BCW was previously a one the transfer does not occur.

Data Direction Control: No control bits are provided in this format for data direction. In the LT format odd-numbered devices are assumed to be executing Input operations, even-numbered devices are assumed to be executing Output operations. The data direction is controlled by the least-significant bit of the device address transmitted at the beginning of the Control Unit initiated sequence.

BAR PRINTER, 250 LPM:

The line printer operates at 250 lines per minute with a 63-character font. The paper-skipping speed is 25 inches per second. Paper spacing is controlled by program with a paper loop. Line spacing is six to the inch. Horizontal spacing is ten characters to the inch.
The printer is assigned a reserved area of memory for storage of data to be printed, and it will print only from this area. The printer will use only the low-order six bits of each byte and will scan for printing the 96 (9200 only), 120, or 132 characters within its reserved memory area.

Options to this basic printer are:

1. Expansion of the number of print positions from 96 to 120 or 132.

2. A replaceable character font of 48 printable characters which will allow 16 numeric and special characters to be printed at 500 LPM on the basic 250 LPM printer. This font will require program translation.

CARD READER, (400 CPM-COLUMN)

The card reader included with this system operates at 400 cards per minute. The unit has four functional stations: input hopper, read wait, read, and stacker. The hopper capacity is 1200 cards, and the stacker capacity is 1500 cards. Reading is accomplished on a demand cycle with only one card committed at a time. A full 80 columns will be read during each read cycle at all times unless the card is a stub card.

Data may be presented to the processor in two formats: compressed mode and image mode.

Options to the basic reader are:

1. 51 column stub card feature

2. 66 column stub card feature

CARD PUNCH 75 to 200 CPM:

The card punch provides the means of punching cards in compressed or binary image mode in a serial or column-by-column manner. The unit has six functional stations: input hopper, ready, wait, punch, and output stackers. The input hopper has a capacity of 1200 cards, and the output and reject stackers have a capacity of 850 and 850 cards, respectively. The speed varies from 75 to 200 cards per minute depending upon location of the last column punched. Punching always starts at the first column. Punched data is (echo) checked.

The card punch will be able to handle cards scored for subsequent separation into stub cards.

A pre-punch read station is available as an option to permit reading and punching along the same card path.

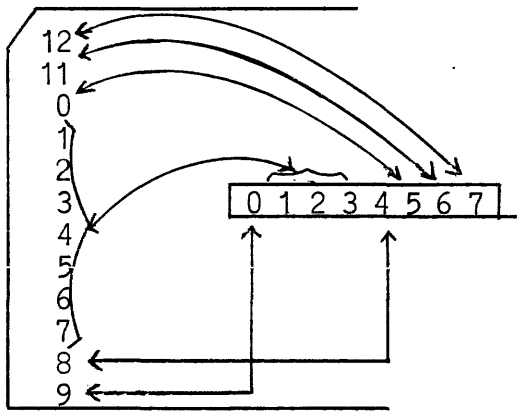
Stacker select is performed automatically by the hardware for error cards; however, program-controlled stacker select is available as an option.

The card punch may be omitted from this system.

Card Code Translation (80 Column Card):

When an 80 column card is read or punched with hardware compression, the hardware compression is to or from an 8-bit code in memory. This 8-bit code is not one of the internal processing codes and requires program translation to be processed after a card is read into memory and a program retranslation to generate the 8-bit code for punching a card. The hardware translation of row punches in the 80 column card to or from an 8-bit code are shown in the following table.

Row Punched holes



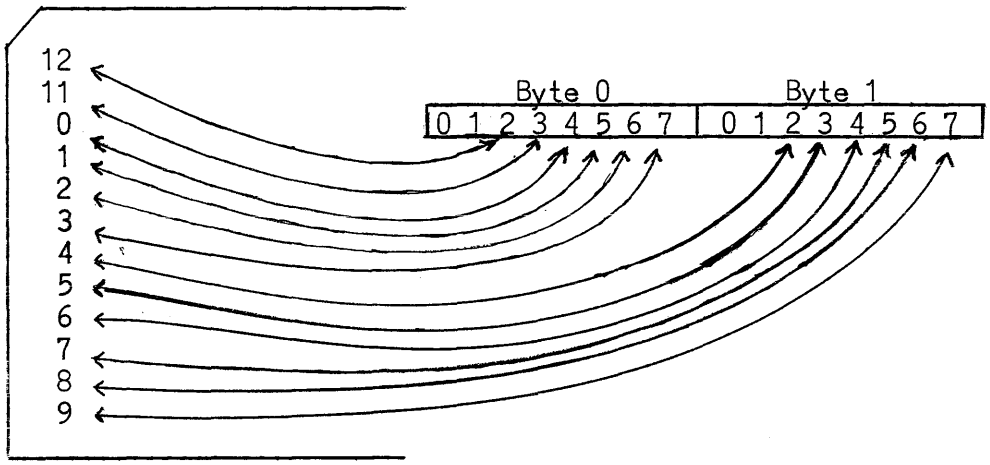
Punches in rows 1 through 7 generate the following bit combinations

	Bits 123
No Punch	= 000
1	" = 011
2	" = 101
3	" = 001
4	" = 010
5	" = 100
6	" = 111
7	" = 110

If more than one row punch from 1 through 7 is present on a card, the combined codes will be OR'ed together; thus, a 2 and a 5 punch will appear to have been a 2 punch.

Card Code Image Mode:

When an 80 column card is read or punched in the image mode, the twelve possible punched holes of a column are represented by 1's in two bytes of memory. As shown in the diagram, bit positions 0 to 1 are unused in the image mode. When punching the image mode one's in bit position 0 to 1 will be ignored. When reading or punching in the image mode the 80 columns of the card occupy 160 bytes of memory. These 160 bytes are consecutively occupied by 2 bytes for each consecutive column in a card starting at column 1. Bits 0 and 1 of the data bytes will be cleared to zeros on an image read.



Internal Codes:

There are 256 possible code combinations that can be represented by any eight bit byte of memory. As described in a previous subsection, card code translation is to or from a compressed code. This compressed code generated by the card reader must then be translated to the desire internal code for processing and retranslated from the internal code to the compressed code for punching. There are two sets of internal codes which are subsets of the 256 possible code combinations which will be used to select the two bar printer graphics. Figures 1 and 2, following, contain the 63 character print set and the 48 character print set.

Bit Designation

4567	23			
	00	01	10	11
0000	NP	&	-	0
0001	A	J	/	1
0010	B	K	S	2
0011	C	L	T	3
0100	D	M	U	4
0101	E	N	V	5
0110	F	O	W	6
0111	G	P	X	7
1000	H	Q	Y	8
1001	I	R	Z	9
1010	¢	!	\	:
1011	.	$	,	#
1100	<	*	%	@
1101	(	)	_	'
1110	+	;	>	=
1111	|	¬	?	"

Figure 14

63 Character Graphic Set assigned to the six bit printer synchronizer code.

4567	23			
	00	01	10	11
0000	A	NP	Q	0
0001	B		R	1
0010	C		S	2
0011	D		T	3
0100	E		U	4
0101	F		V	5
0110	G		W	6
0111	H		X	7
1000	I		Y	8
1001	J		Z	9
1010	K		+	.
1011	L		&	$
1100	M		%	*
1101	N		#	-
1110	O		'	/
1111	P	NP	;	,

Figure 15

48 Character Graphic Set assigned to the six bit printer synchronizer code.

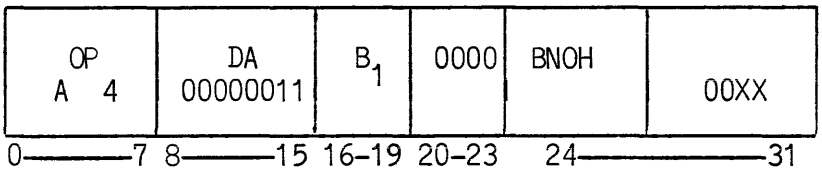
INDIVIDUAL PERIPHERAL CONTROL REQUIREMENTS

Each peripheral unit associated with the processor, whether attached as a basic unit or by way of the general purpose channel, has unique control requirements.

PRINTER:

The printer's sequence of the operation is to print and then advance paper. To allow the maximum amount of time to prepare the next line of data and to store it in the specified print area, interrupt is generated and the next XIOF can be issued before the paper advance operation is completed from the last XIOF instruction.

The printer Execute I/O instruction has the following format:



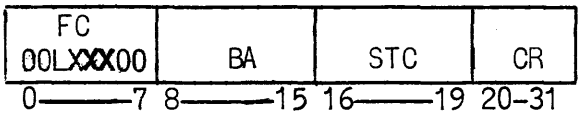
Print BNOX0001
Control BNOX0011

B = 0 Standard 63 Character Bar
B = 1 Optional 48 Character Bar
N = 1 Print Numeric if 48 Character Bar option is activated
H = 1 Inhibit all interrupts

There are only two valid print instructions, print and control, the Bar selection is a modifier to these codes and is only effective if the Bar Print option has been included as part of the active system. Any other codes will cause invalid operations. Print may be given with or without paper advance. Control is used for paper feeding without printing.

Processing will continue during all I/O instructions. If the H bit is set to one all interrupts from the device will be inhibited. The TIO instruction should then be used to determine the status of the device.

The buffer control word for the printer contains the following data:



BA = Base Address
STC = Starting Code
CR = Code Register
FC = Forms Control

Software does not load CR, STC, or BA because these three bytes are under complete hardware control. If they are inadvertently changed by a program, a loss of printer control will probably result.

The Forms Control byte is loaded by the program once it has been determined either by a TIO or an interrupt that it is permissible. The Forms Control byte will not be changed by the execution of a printer function. The four bits which designate the desired forms action as follows:

L X X X	
0 0 0 0	Space 0 lines
0 0 0 1	Space 1 line
0 0 1 0	Space 2 lines
1 X X X	Select any of 7 paper loop channel controls by matching holes in the paper loop to the 1 bits in the X bits.

There are two paper loop conventions:

X X X	
1 1 1	for home paper
0 0 1	for form overflow

If a hole combination is sought under paper loop control that is not punched on the tape, a runaway paper condition will result.

The status byte contains information pertaining to the result of the last issued order or the next to last issued order. The status indications are as follows:

1. All zeros = No indicators set; function performed as specified.

2. Bit 7 set to 1 = Paper Low* - as a result of paper spacing. Until the paper condition is corrected, this indicator will occur for each XIOF. Paper low will be indicated when the bottom edge of the form is 15 1/3" $\pm$ 1/3" from print line.

3. Bit 6 set to 1 = Form overflow* - 001 sensed at paper loop station, while single or double spacing.

4. Bit 5 set to 1 = Interrupt request pending. This status bit is only set if the TIO function clears a pending interrupt before it is accepted. This status bit does not indicate an error.

5. Bit 4 set to 1 = Instruction does not agree with bar switch setting.

6. Bit 3 set to 1 = Data Parity or Control Parity error on last XIOF instruction - stops printer immediately.

7. Bit 2 set to 1 = Memory overload occurred on last XIOF instruction - stops printer immediately.

8. Bit 1 set to 1 = Paper Runaway* - Forms Control lost. No further orders will be accepted without operator intervention.

9. Bit 0 set to 1 = Abnormal or Not Ready

* These conditions are recognized following the normal interrupt request, therefore the previous function will be properly completed except in the case of Paper Runaway where paper has been spaced improperly. If another XIOF has already been accepted it will be aborted and an interrupt will be generated. If the next XIOF is not issued until after detection of the condition, the order will be accepted, then aborted and an interrupt request will be generated. Any error that happens before paper is advanced will void paper advancing.

Interrupt requests will occur at the following times:

1. End of print before associated paper feed is started.

2. Immediately following an accepted paper feed order before paper advancing has begun, unless a previously initiated paper feed order is in progress. In this case the interrupt is delayed until the previously initiated paper feed order has been completed.

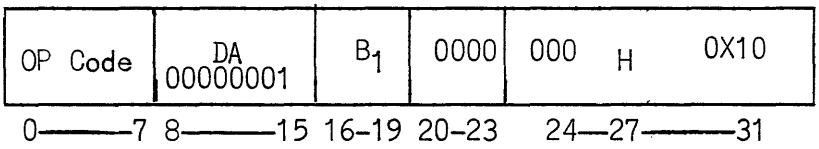
3. Upon the abortion of an order due to detection of Paper low, Forms overflow, or Forms Runaway as a result of a preceeding order.

4. Upon termination of an operation due to any other error condition.

CARD READER INSTRUCTION CONTROLS:

The card reader test I/O instruction performs normally as described in the general section on the TIO instruction.

The Execute I/O instruction for the card reader has the following format:

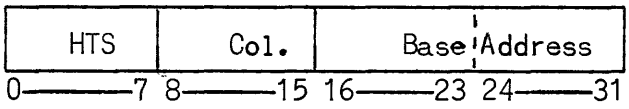


000H0010 Read Translate Mode
000H0110 Read Image Mode
H = 1 inhibit all interrupts

These two combinations of bits, in the direct B_1-D_1 field or the indexed B_1-D_1, are the only permissible combinations of reader XIOF instructions. Any other combination may cause an error.

Processing will continue during all I/O instructions. If the IH bit is set to one all interrupts from device will be inhibited. The TIO instructions should then be used to determine the status of the device.

The buffer control word for the card reader contains the following data:



HTS = Hardware temporary storage reserved for the reader. This Byte should not be loaded by the program.

Col. = The number of columns to be read. This must always be 80. At the end of a card operation this count will be decremented to zero.

Base Address = The starting address where the card will be read into. This must always be an even address. At the end of a card operation this address will point to the Byte immediately following the last Byte of information read from the card.

The status byte contains information pertaining to the result of the last issued order or the next to last issued order. The status indications are as follows:

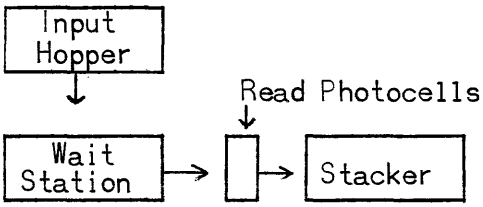
1. All zeros = No indicators set, function performed as specified.

2. Bit 5 set to 1 = Interrupt request pending. This status bit is only set if the TIO function clears a pending interrupt before it is accepted. This status bit does not indicate an error.

3. Bit 1 set to 1 = Misfeed, Not Ready, Offline, Hopper empty or Stacker full -- These conditions are sampled only at initiation time of the XIOF instruction. If any one of these conditions exist, the XIOF instruction will be rejected and the indicator bit set. Test I/O instruction will store this indicator only if it follows an XIOF instruction which was rejected because of one of these conditions.

4. Bit 0 set to 1 = Stacker jam, control parity or photocell check; instruction may or may not have been accepted and card been fed.

Card Reader Feed Path:



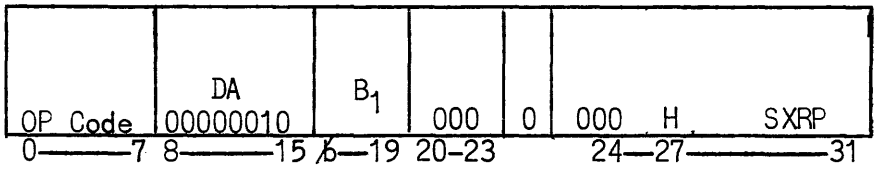
Card must be manually fed into the wait station after hopper is loaded. After this is done, each read feeds a card from the wait station past the photocells and into the stacker, while feeding the next card into the wait station.

CARD PUNCH:

The Card Punch in this section will include the controls required for the card reader option that may be incorporated to form a card read punch.

The Card Punch Test I/O instruction performs normally as described in the general section on the TIO instructions.

The Execute I/O instruction for the card punch and reader option has the following format:



H if 1 = inhibit all interrupts
P if 1 = Punch a card
R if 1 = Read a card
X if 0 = R and/or P a card in compressed mode
X if 1 = R and/or P a card in image mode
S if 1 = Select Stacker. Effective only if the program stacker select feature is installed.

Either the R or P bit must be 1. All other bits shown as 0's must be 0's, or an error may result.

Col. = The number of columns to be read. This must always be 80. At the end of a card operation this count will be decremented to zero.

Base Address = The starting address where the card will be read into. This must always be an even address. At the end of a card operation this address will point to the Byte immediately following the last Byte of information read from the card.

HTS = Hardware temporary storage reserved for the reader. This Byte should not be loaded by the program.

The status byte contains information pertaining to the result of the last issued order or to the next to the last issued order. The status indications are as follows:

1. All zeros = No indicators set; function performed as specified.

2. Bit 6 set to 1 = Hopper Empty or Stacker Full -- When this status bit is set the last XIOF function was terminated before it was executed. To recover from this early termination reinitiate the XIOF order once the condition has been relieved.

3. Bit 5 set to 1 = Interrupt request pending. This status bit is only set if the TIO function clears a pending interrupt before it is accepted. This status bit does not indicate an error.

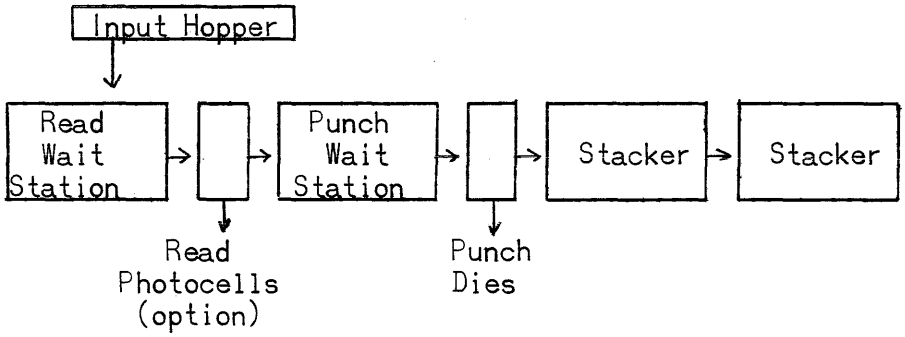
4. Bit 4 set to 1 = Photocell Check Error -- This is a check on the read Photocells as well as possible indication of a card jam. This error indication will be registered by its status bit at termination of an XIOF. The last accepted XIOF function should be assumed in error.

5. Bit 3 set to 1 = Data Parity or Control Parity Error -- Card at read station or card at punch station may be in error. Immediate interrupt upon recognition of error and XIOF function terminated. Card passing through punch station will automatically be selected to error stacker. This status bit indicates that the last XIOF instruction was terminated, probably before completion.

6. Bit 2 set to 1 = Punch Check error. Interrupt after card has been punched. Card being punched in error will automatically be selected to the error stacker. The status bit being set indicates that the last card punched was in error.

7. Bit 0 set to 1 = Stacker Jam, punch entry or exit check, interlocks, and any other condition that may necessitate manual intervention. This status bit may indicate serious or minor malfunction. Recovery procedure will be up to the operator.

Card Punch Feed Path:



Two cards must be manually fed to fill the wait stations. The first card will not be read if the read option is present. After this is dones, a XIOF instruction to read, punch, or read and punch will cause the following action:

1. Move a card from the punch wait station past the punch dies (punching it if specified) into one of the output stackers.

2. Move a card from the read wait station past the read photocells (reading it if specified) into the punch wait station.

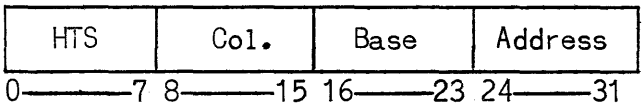
3. Move a card from the hopper into the read wait station.

Stacker selection is given for the card in the punch wait station in the same instruction that caused it to move from the punch wait station.

NOTE: The second stacker is considered an error stacker and is selected by the hardware on punch errors. The optional feature makes this stacker selectable by program. However, errors will always cause this stacker to be selected regardless of program choice.

Processing will continue during all I/O instructions. If the IH bit is set to one all interrupts from the device will be inhibited. The TIO instruction should then be used to determine the status of the device.

The buffer control word for the card punch contains the following data:

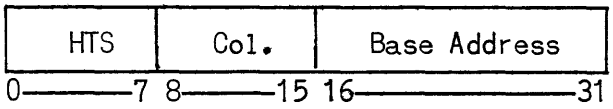


HTS = Hardware temporary storage reserved for the punch. This Byte should not be loaded by the program.

Col. = The number of columns to be punched. This must always be an even zero number. At the end of a card operation this count will be decremented to zero.

Base Address = The starting address where the card will be punched from. This must always be an even address. At the end of a card operation this address will point to the Byte immediately following the last Byte of information punched.

The buffer control for the punch reader option contains the following data:



CONTROL PANEL

GENERAL DESCRIPTION:

The Control panel provides the means to operate and control the system. By switches and lights, the operator or service engineer performs the following functions:

1. Turns power on or off.
2. Starts or stops processing.
3. Displays or alters the content of any memory position.
4. Displays all major registers, flip-flops and status indicators.
5. Operates the processor in run, cycle, or instruction modes.
6. Initializes processor, general purpose channel (GPC), and all basic peripherals.
7. Controls the bar printer, card punch and card reader manual functions.
8. Provides for initial loading of programs.

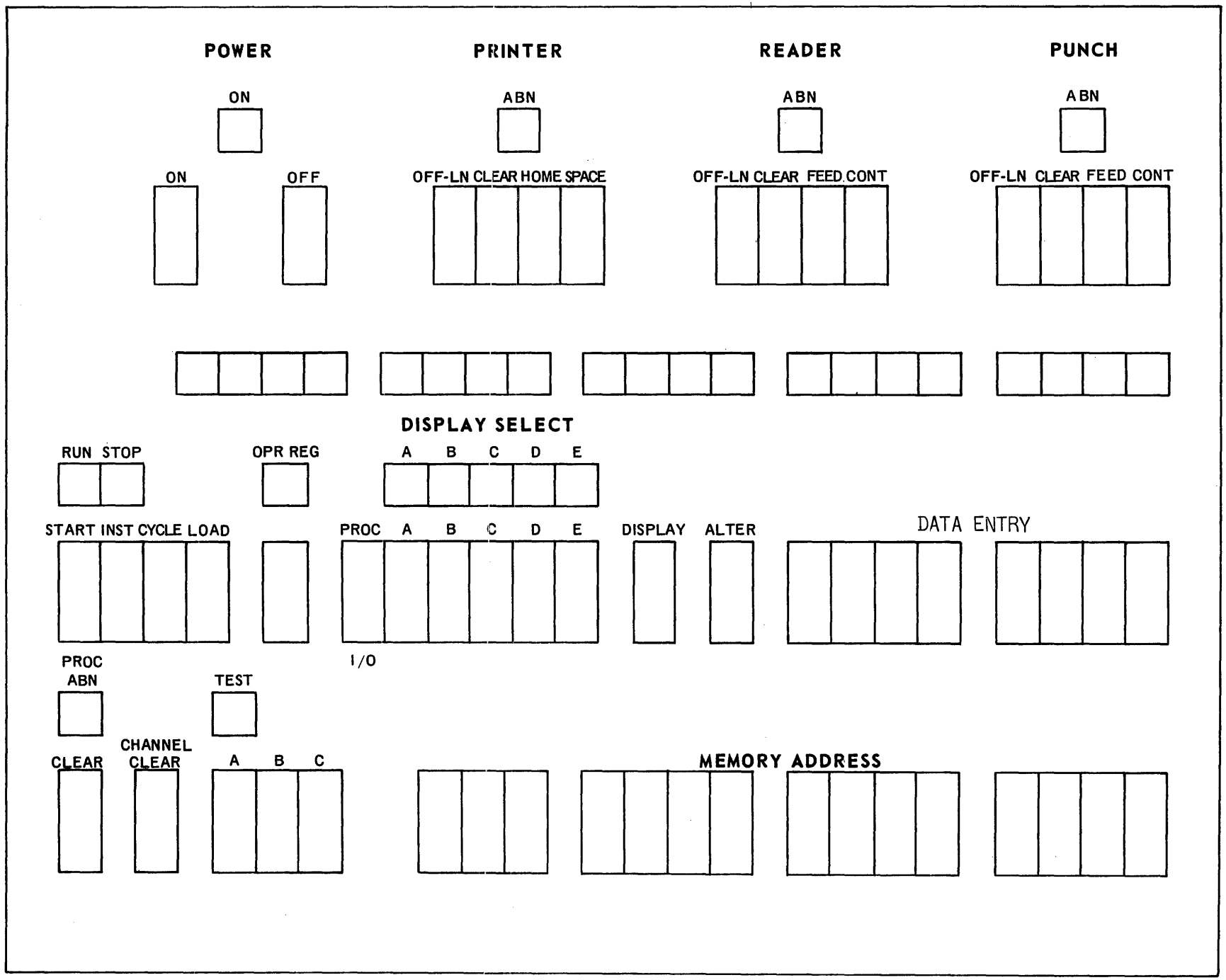
The following description of the individual switches and lights should be read with the attached control panel layout exposed.

Display Lights:

There are 20 white display lights. These lights are used to display the contents of memory and the contents of various registers. They are also used to display the settings of various flip-flops within the processor and the basic peripheral controls. When displaying registers and memory the lights are lit when the designated bit value is 1 and are out if the designated bit value is 0. The flip-flop status indicators are lit when a particular condition is detected.

Display Selection Switches and Lights:

There are 6 display selection switches: a two-positional master switch and 5 mutually exclusive momentary switches with associated indicating selection lights. The combination of these 6 switches yields 16 independent display settings for the 20 display lights. The master switch is labeled PROC (Processor) and I/O (Peripheral Control Units); the five momentary switches are labeled A, B, C, D and E respectively.


POWER
ON
ON
OFF
PRINTER
ABN
OFF-LN CLEAR HOME SPACE
READER
ABN
OFF-LN CLEAR FEED CONT
PUNCH
ABN
OFF-LN CLEAR FEED CONT
DISPLAY SELECT
RUN STOP
OPR REG
A B C D E
START INST CYCLE LOAD
PROC A B C D E
DISPLAY
ALTER
DATA ENTRY
I/O
PROC ABN
TEST
CLEAR
CHANNEL CLEAR
A B C
MEMORY ADDRESS

The display selection switches, and display lights can be used while operating in various modes of the processor, but while in specific modes only selective settings and lights yield valid information. When the processor is in either the "run" mode or the "instruction" mode, the four most significant display lights can be used to monitor processor and peripheral control unit errors. This is accomplished by setting the display selection switches to PROC (D) or the I/O (A, B, C, D or E). When operating in these two modes, the remaining 16 lights of the display are used independently of the display selection switches to indicate any of the following:

1. The most significant 2 bytes of each instruction before it is executed (instruction mode),

2. the address portion (15 bits) of the halt and proceed instruction which has stopped the processor (run or instruction mode), and

3. a byte from memory through the operation of the memory display switch (run or instruction mode).

When the processor is in cycle mode, all 8 settings of the display selection switches are valid, and each selects 20 signals to be displayed.

Memory Address Switches:

There are 15 address two positional switches. The upper position denotes a 1 and the low position denotes a 0. Via these switches, an address from 0 through 16383 (32767 on 9300) can be represented.

Data Entry Switches:

There are 8 data entry switches which are used to enter data into the processor's memory. The upper position denotes a 1 and the lower position denotes a 0. Via these switches all possible values of a byte of data can be represented. These switches are also used during initial load.

DISPLAY AND ALTER SWITCHES:

Display Switch:

This momentary switch when depressed causes the contents of a byte of memory specified by the memory address switches to be displayed. The byte is displayed in the least significant eight display lights on the right side of the control panel. A single byte of memory can be displayed while the processor is running if the following conditions are understood:

1. The display switch must be continually depressed during the time it is desired to view the display.

2. The display switch while depressed will block the automatic display of the 14 bits of the halt and proceed instruction.

This switch is normally used when the processor and peripherals are stopped and the processor is not in cycle mode. Under these conditions the 16 right-most display lights can be used to display byte values of memory. To display a byte of memory the address switches must be set to the desired memory address for each byte. After the initial display, each succeeding depression of the display switch will cause the byte values displayed in the eight right-most lights to be displayed in the next set of lights with the new value displayed in the first set of eight lights. The previous value in the second set of eight lights will be lost. The displaying of memory does not change any values in memory.

The memory display logic will work in cycle mode except for the following condition.

Alter Switch:

This momentary switch when depressed causes the value set up in the data entry switches to be stored in the memory at the address specified by the memory address switches. This may be accomplished while running, stopped, or a cycle at a time, similar to the display operation. No display occurs.

MODE SWITCHES:

There are three two-position mode switches; instruction, cycle and load. Combinations of these switches may be used.

Instruction Switch:

The instruction switch has two functions. The depression of the instruction switch while processing will cause processing to stop at the end of instruction staticize. The depression of the instruction switch will also allow the operator to operate the processor one instruction at a time in conjunction with the start switch, stopping at the end of staticize of each instruction. The instruction staticized is displayed via the sixteen right-most display lights.

Cycle Switch:

The cycle switch when depressed is used in conjunction with the start switch to single cycle the action of the processor. This control panel function is primarily used by service personnel for debugging the processor. The complete description of all the aspects of the cycle switch will be described in the service manual. This switch should not be depressed while peripherals are running since it freezes both processor and control units after the current cycle and may cause abnormal results in the peripherals.

Load Switch:

The load switch is used in conjunction with the data entry switches and the start switch. With the load switch depressed and the device address set up in the data entry switches the depression of the start switch will cause an initial load function from the desired peripheral. In the basic system this will be the read card function either in the card reader or the reader in the card read-punch. This switch must be reset before using the start switch following initial load. After an initial load operation, the device selected will have an interrupt pending.

If initial load is to be done from magnetic tape, the General Purpose Channel Clear operation will set the control unit to read 9 channel if the selected tape unit is 9 channel. If the selected tape unit is 7 channel, the control unit will be set to read odd parity, and if the tape unit includes the data conversion option, the control unit will be set to use the option.

After initial load, the first instruction executed is as indicated in the I/O PSC.

START SWITCH:

The start switch is used with or without the mode switches. The actions with the mode switches have been explained under mode switches. With all the mode switches in the off position, depression of the start switch will start the processor in the run mode under control of the governing Program State Control.

RUN AND STOP LIGHT:

The run light will be on and the stop light will be off if the processor is operating free of any control panel intervention as well as program stops or abnormal conditions. The run and the stop light will both be on if the program has been stopped via the depression of the instruction mode switch. The run light will be off and stop light will be on if the program has caused processing to stop via the halt and proceed instruction.

The run and stop light will be off if power is off or the processor is in a stall condition. If a processor abnormal condition stops the processor, the stop light and the processor error lights will be on; the run lamp may be on or off.

PROCESSOR ERROR LIGHT:

The processor error light is lit when a parity error is detected, an illegal memory address is detected or a divide check occurs or an abnormal condition is detected by the hardware protection circuits.

PROCESSOR CLEAR SWITCH:

The processor clear switch is a momentary switch that upon depression clears all the processor and peripheral indicators and sets the processor and peripherals to a ready state so that initialization of program start can take place normally. Depressing this switch also provides a lamp test function by lighting all lamps except the four display selection indicators.

1. Tells processor to find next instruction in 22-25. The instruction must be a branch.

2. Puts the processor in I/O state.

3. Causes the destination of alters to be determined by the memory address switches.

GENERAL PURPOSE CHANNEL CLEAR (GPC):

This momentary switch is only active if the general purpose channel option has been activated. Its function is to clear all GPC flip-flops to a ready state.

POWER:

Power is turned on by momentary switch and off by another momentary switch. The power light being lit indicates that power is on and the processor is available. Power should normally only be turned off when the processor and all peripherals are stopped. Turning power on clears processor and peripherals.

BASIC PERIPHERAL CONTROLS:

Off Line Switches:

The printer, the card reader and the card punch control units each have a two positional off line switch. These switches functionally disconnect their associated peripheral from the central processor so that they can be serviced independently from the rest of the system.

Clear Switches:

The printer, the card reader and the card punch control units each have a momentary clear switch. The clear switch clears all abnormal indicators and sets each peripheral individually to a ready state.

Abnormal Lights:

The printer, the card reader and the card punch control units each have an abnormal light. These lights go on if these peripherals stop as a result of some abnormal condition. The specific cause of the error can be displayed by the Display lights. Depression of the clear switch associated with the peripheral will reset the status flip-flops and turn off the abnormal light if the condition has been corrected.

Home Paper Switch:

This momentary switch for the printer causes the paper to be advanced to the home paper position, under paper loop control. If the paper loop is absent, paper will space one position. If the paper loop does not have the home paper punch combination punched, a runaway paper condition will result.

Space Paper Switch:

This momentary switch associated with the printer causes the paper to space at the rate of five lines per second.

Feed Switches:

These momentary switches associated with the card reader and card punch cause cards to be advanced one station.

The card reader and the card punch control units each have a two positional continuous switch. These switches enable the continuous feeding of cards if the unit is off line.

OPERATION DESCRIPTION

Some basic control panel operational procedures that are required by the system are as follows:

INITIAL LOAD:

1. Ready the peripherals for use.

   a. Clearing peripherals of abnormal indications.
   b. Feeding 1 card into the card reader.
   c. Feeding 2 cards into the card punch, first card blank.

2. Depress the processor clear switch.

3. Set the device address of the loading peripheral into the data entry switches.

4. Depress the load switch.

5. Depress the start switch.

6. Reset the load switch.

Processor clear forces the program to be governed by the I/O Program State Control. The data segment is read into memory starting at address 0. The generated peripheral function is read with lockout.

CONTROL PANEL RESPONSE TO HALT AND PROCEED:

The processor has been stopped via the program halt and proceed instruction. This will be indicated by the lights on the processor. Specifically the run and processor error light will be off and the stop light will be on.

1. Examine the three light conditions run, stop and error.

2. Record the light settings in right-most 16 display lights. The lights will represent the value of the bits in the B_1-D_1 field of the halt and proceed instruction.

3. Follow the operating specification procedure for halt and display stops:

   Example: Card Reader Hopper empty:

   a. Load cards into hopper
   b. Feed a card
   c. Clear Card reader
   d. Depress processor start

If operator was in the process of manually displaying the contents of memory and the program stops as a result of a halt and proceed instruction, the B_1-D_1 values will not be displayed in the display lights. These values can be displayed in the lights by setting up address 8 and 9 consecutively and depressing the display switch.

PHYSICAL CHARACTERISTICS

DIMENSIONS - LAYOUT:

The basic system shown in Figure 16 consists of a processor, bar printer, card reader, and column punch.

WEIGHT:

The total estimated weight of the basic system is approximately 1400 lbs. Addition of the optional serial punch will increase the weight to approximately 1800 lbs.

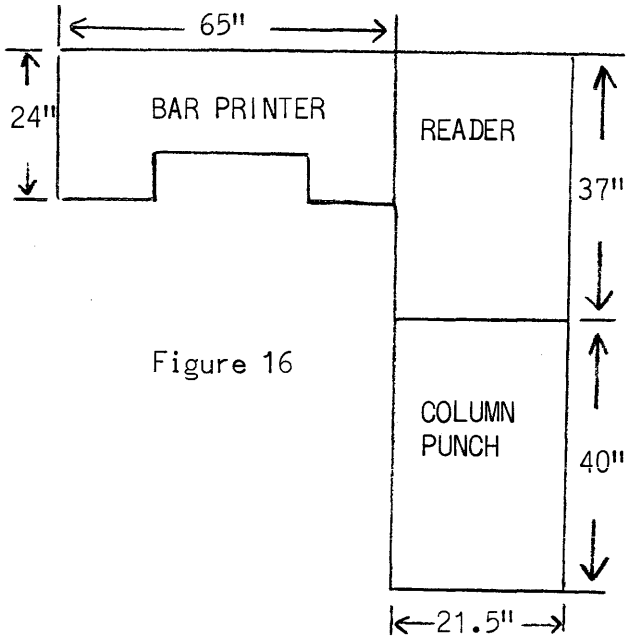


Figure 16

Note: All dimensions shown are approximate.

UNIVAC 9300 INSTRUCTION REPERTOIRE SUMMARY

Operation	OP Code	Instruction	Mnemonic	Format	Instruction Times (Note 1) (in microseconds)
Binary	40	Store Halfword	STH	RX	20.4
	48	Load Halfword	LH		20.4
	49	Compare Halfword	CH		20.4
	A6	Add Immediate	AI	SI	19.2
	AA	Add Halfword	AH	RX	20.4
	AB	Subtract Halfword	SH		20.4
Logical	91	Test Under Mask	TM	SI	No match or mask if zero 16.8 Partial or full mask 19.2
	92	Move Immediate	MVI		16.8
	94	AND Immediate	NI		16.8
	95	Compare Immediate	CLI		16.8
	96	OR Immediate	OI		16.8
	A9	Halt and Proceed	HPR		14.4
	D1	Move Numeric	MVN	SS	16.8 + 8.4(N) (Note 2)
	D2	Move Characters	MVC		16.8 + 8.4(N)
	D4	AND Characters	NC		16.8 + 8.4(N)
	D5	Compare Logical	CLC		25.2 + 8.4(N_E) (Note 3)
	D6	OR Characters	OC		16.8 + 8.4(N)
	DC	Translate	TR		16.8 + 14.4(N)
	DE	Edit	ED		See Appendix
Decimal	F1	Move with Offset	MVO	SS	25.2+3.6(N_2)+6(N_1)
	F2	Pack	PACK		25.2+3.6(N_2)+4.8(N_1)
	F3	Unpack	UNPK		21.6+7.2(N_2)+4.8(N_1)
	F8	Zero and Add	ZAP		26.4+3.6(N_2)+4.8(N_1)
	F9	Compare Decimal	CP		26.4+3.6(N_2)+4.8(N_1)
	FA	Add Decimal	AP		26.4+3.6(N_2)+4.8(N_1)

INSTRUCTION REPERTOIRE SUMMARY (Con't)

Operation	OPCode	Instruction	Mnemonic	Format	Instruction Times (Note 1) (in microseconds)
Decimal (continued)	FB	Subtract Decimal	SP		$26.4+3.6(N_2)+4.8(N_1)$
	FC	Multiply Decimal	MP		See Appendix
	FD	Divide Decimal	DP		See Appendix
Branch	45	Branch and Link	BAL	RX	18 No Branch 15.6
	47	Branch on Condition	BC		Branch 18
Privileged	A0	Store State	SPSC	SI	24 Load entire stateword 24
	A8	Load State	LPSC		Other than load entire 18 stateword
Special	A1	Supervisor Call	SRC	SI	12
	A4	Execute I/O	XIOF	SI	Integrated I/O units 18 General-Purpose Channel
	A5	Test I/O	TIO		22.8

NOTES: 1. Timing for all instructions assumes no indexing. Add 3.6 microseconds for each indexing operation.

2. N, N_1, N_2 = The number of bytes specified in the respective fields ($L + 1$, $L_1 = 1$, or $L_2 = 1$).

3. N_E = The number of most significant bytes that will compare identically between OP1 and OP2 in the CLC instruction.

Table 1. Summary of 9300 System Instruction Repertoire

APPENDIX A

Multiply, Divide and Edit Timing

Divide

$$56 + 6N_1 + 8N_2 + (\Sigma OQD + 1) + \Sigma\overline{EQD}\ (22 + 14N_2)$$

Where:

ΣOQD = sum of odd quotient digits

$\Sigma\overline{EQD}$ = sum of 10's compliment of even quotient digits

Edit

$$30 + 6S_2 + 14_N = 6\ (N_{DS} + N_{SS})$$

Where:

S_2 = number of signs in OP2

N_{DS} = number of Digit Select Bytes

N_{SS} = number of Significant Start Bytes

If operands of either Divide or Edit instruction are indexed add 6 for each operand indexed.

Multiply

Tables or graphs showing typical times will be published later.

UNIVAC

DIVISION OF SPERRY RAND CORPORATION

UP-4122