WESTINGHOUSE DPS-2402 COMPUTER

MACHINE REFERENCE MANUAL

Westinghouse Defense and Space Center
Surface Division
P. O. Box 1897
Baltimore 3, Maryland 21203

No. DPS-2402-6

PREFACE

This manual describes the operation of the Westinghouse DPS-2402 Computer. This manual may serve as a programming guide and teaching aid as well as a system reference manual.

The Westinghouse DPS-2402 assembly program is supplied with every system along with a comprehensive software package including a Fortran Compiler, bootstrap loader, a relocatable loader, input-output conversion routines, and a library of numerical and utility routines. The software is described in the Westinghouse DPS-2402 Programmers Reference Manual.

The first section of this manual gives the system specification of the Westinghouse DPS-2402. Thereafter, the description is carried to the detailed level required for confident programming of the basic computer. Programming required to use peripheral equipment is described in the reference manual for each type of peripheral equipment.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (CONTINUED)

LIST OF TABLES

LIST OF FIGURES

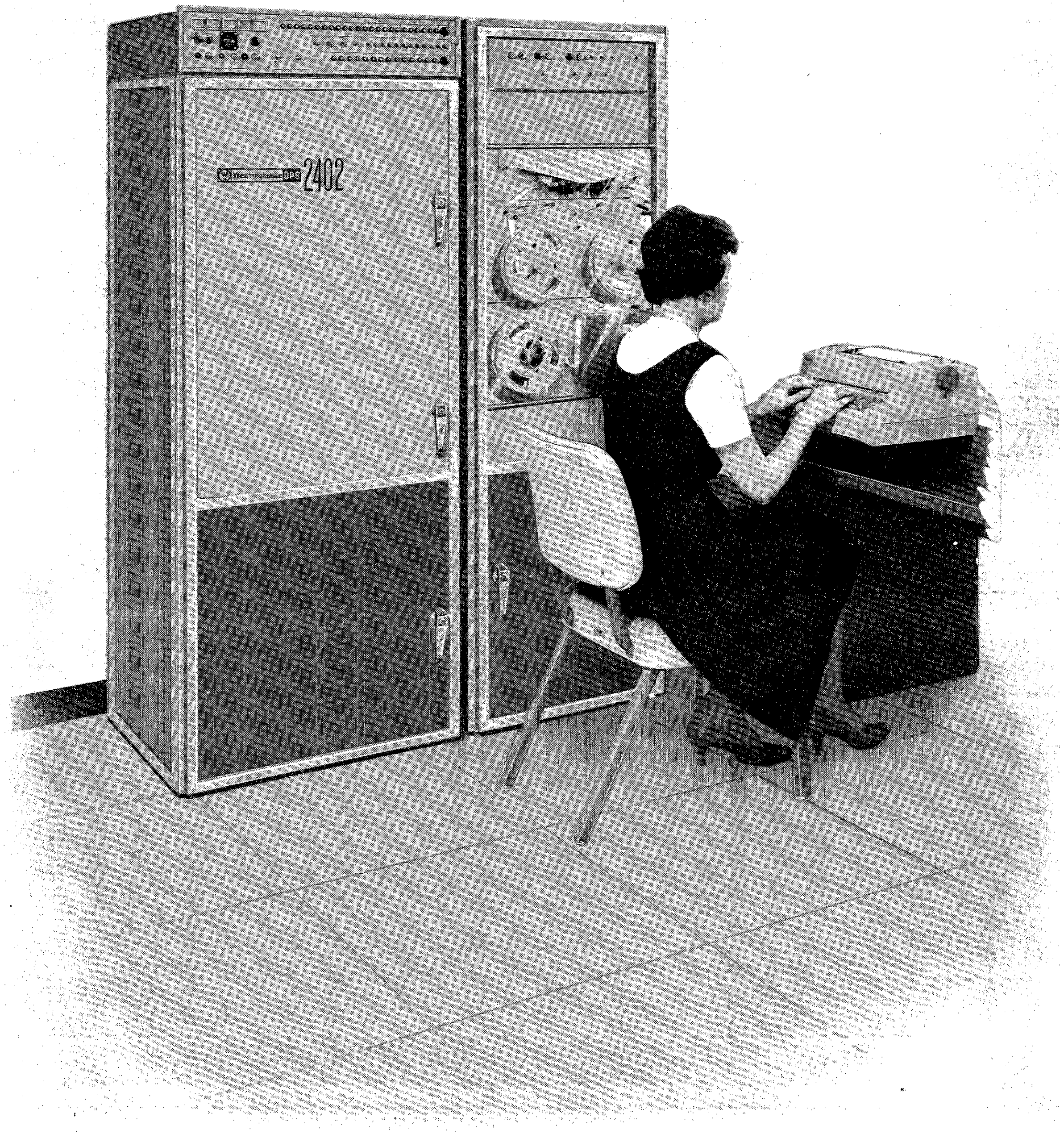


Figure 1. Basic Westinghouse DPS-2402 Computer

SYSTEM SPECIFICATIONS

GENERAL

The following is a summary of the technical characteristics of the Westinghouse DPS-2402 Computer:

Type: General-purpose, stored program integrated circuit, medium scale, binary, parallel.

Memory Section:

1.0 microsecond access time

2.0 microsecond read-write cycle time

24-bit word length, parallel transfer (plus 1 parity bit)

4,096, 8,192 or 16,384 words expandable to 32,768 words.

Coincident current mode.

Arithmetic Section:

Organization: 24 bit, parallel, 2's complement, binary, fixed-point

Registers - Two arithmetic (A and Q)

- One transient (D)

- One shift count (N)

Instruction Execution Time (Including procurement of instruction and operand)

Add Time	4.0 microseconds
Multiply Time	12.0 microseconds
Divide Time	22.0 microseconds

Control

Section: Word length: 24 bits

Single address instructions: one instruction per word.

Addressable index registers: 3 (B-boxes)

Parallel operation

Multiple-level indirect address capability with indexing of primary and indirect addresses.

38 basic instructions, or a total of 128 instructions by using modifiers.

Input/Output

Section: Eight (8) channels (expandable to 16)

Any channel may be used in fully buffered, express, or external function mode.

Both input and output buffers may be enabled on any channel simultaneously.

Buffer mode operation, once initiated proceeds without program intervention.

Internal and external interrupts on each channel.

Real-time clock, set by program, with interrupt when recycling to zero count.

Physical

Characteristics: Size: 72 x 28.75 x 25 inches

Weight: 750 pounds

Power Consumption: 1200 watts 120 volt, single phase, 60 cycle.

Environmental Requirements: Non-volatile core memory with power failure lockup which saves the contents of operational registers. The mechanical design of the DPS-2402 was performed using the requirements of the following specifications as a guide.

General:	MIL-E-16400E
Vibration:	MIL-STD-167, Type I
Shock:	MIL-S-901 or MIL-E-5400, 3.2.21.6
RF Interference:	MIL-I-16910A
Temperature:	0°-50°C Room Ambient
Humidity:	To 95%

MACHINE STRUCTURE

Figure 1 is a photograph of the Basic Westinghouse DPS-2402 Computer System and Figure 2 is a functional system block diagram. The DPS Computer is organized into four main units, the arithmetic, control, input-output, and the memory units. These are discussed in the following paragraphs.

1. Arithmetic Unit

The arithmetic unit contains the working registers, the high-speed adder, and the logical gating necessary for high-speed execution of the arithmetic and logical commands. The registers required by the unit include two addressable registers (A and Q), one transient register (D), and the shift count register (N).

2. Control Unit

The control unit provides the logic necessary to interpret and direct the execution of commands through cycles of operation. It includes two working registers (I and P), the core memory parity check logic, the function code and designator translators, and the sequence generators. The fault and and parity interrupts are processed by the control unit.

3. Input-Output

The input-output unit consists of a data exchange register (X), a register for updating the buffer control word (Z), input amplifiers and level converters, output drivers, priority gating, buffer control, external function control, and interrupt control. The functions of these circuits are briefly described in the following paragraphs.

4. Memory Unit

The DPS-2402 Computer contains a magnetic core memory operating in a coincident-current mode for storage. The memory has a capacity of 16,384 words of 24 bits plus parity each and a full cycle time of 2 microseconds.

SYMBOLISM FOR REGISTERS

A	Accumulator (A-Register)
(A)	Content of the accumulator.
B_b	Specified Index Register (location) i.e., B1, B2, or B3.
I	Instruction Register
P	Program Register
Q	Multiplier-Quotient Register
S	Storage Address Register
M	Memory Regeneration Register
D	Arithmetic Transient Register
N	Shift Count Register
X	Input-Output Transfer Register
Z	Buffer Control & Real Time Clock Register

REGISTER DESCRIPTIONS

1. A-Register (24 Bits)

This accumulator, referred to as the A-register, is the principle arithmetic register. Inputs to this register are derived from the adder selection gates which provide parallel addition and shifting capability.

With the exception of multiplication, all arithmetic operations call for one operand to be in this register prior to execution of that instruction.

After addition or subtraction, the A-register contains the sum or difference; after a multiplication, the most significant half of the product remains in the accumulator. Additionally, this register contains both the remainder after the execution of a divide and the number on which logical operations are performed.

The contents of the accumulator may be shifted either left or right, closed or open, as described by the shift operations.

2. Q-Register (24 Bits)

This register serves as the multiplier-quotient register. Prior to multiplication, the register contains the multiplier. During multiplication, the multiplier is shifted right, two positions at a time. The three least-significant bits are examined during each shift to determine if the multiplicand should be applied to the partial product. At the same time, the significant part of the partial product is shifted right into the Q-register. At the completion of the high-speed multiplication process, facilitated by the two-bit look-ahead feature, the least significant half of the product is found in the Q-register.

The Q-register contains the least significant half of the dividend prior to the division process. This register is used to assemble and hold the quotient, and the sign of the quotient is found in the most significant bit position.

Shifting of the Q-register contents is similar to that of the A-register. There are cases in which the A and Q register are shifted as a single 48-bit register.

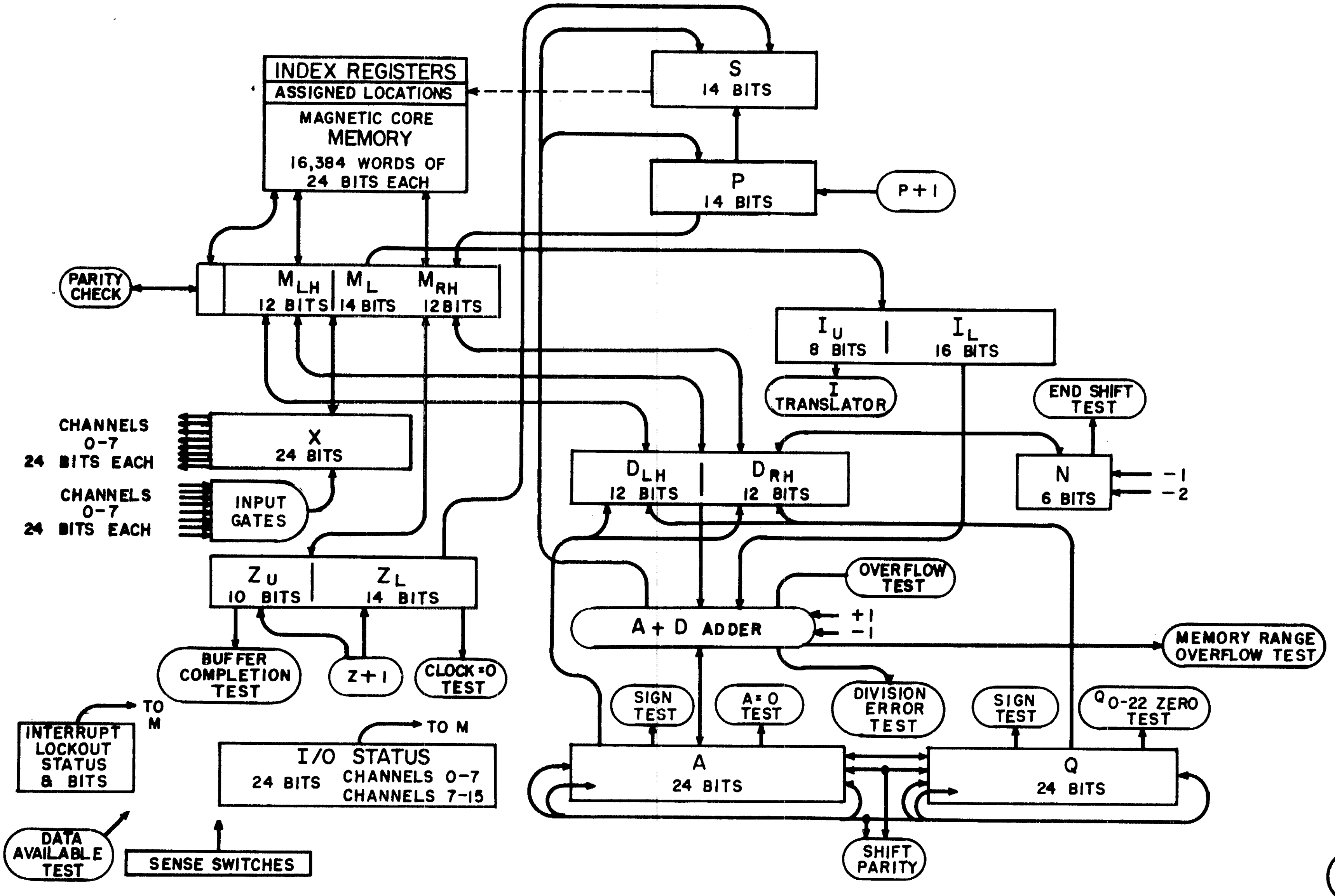


Figure 2. Westinghouse DPS-2402 Computer,
System Block Diagram

3. D-Register (24-Bits)

The D-register is an intermediate register which contains the operand from memory (Y) while the sum, difference, product, or quotient is being formed. This register also contains the B-modifier while it is being added to the address.

A second function of this register is that it serves to transfer data for instruction words to and from the memory unit and all of the arithmetic and control unit registers.

4. N-Register (6 Bits)

This register is implemented in the form of a counter and is used to control the shifting during multiplication, division, and shift command execution.

To readily facilitate a programmed floating point, a "scale factor" instruction is included in the repertoire. After execution of this instruction, the N-register contains the number of positions shifted in the scaling operation. The N-register is not addressable; therefore, the scale factor count is stored in location Y.

5. I-Register (24 Bits)

The I-register stores the current instruction word and thereby serves as the source to the device which controls the function currently being executed. This register acquires the instructions from memory through the M-register. In addition to serving as an input to the function code and designator translators, this register also inputs to the adder circuit to provide modification of the operand address.

6. P-Register (14 Bits)

The address of the next instruction is stored in the P-register. This register is loaded through the adder from the I-register and also is incremented by 1, automatically, as the instructions are sequentially executed. Execution of a skip instruction causes the P-register to be incremented by additional 1. The parallel loading of this register is accomplished by the execution of a jump instruction. A capability of storing the contents of the P-register also is included as a means of implementing the jump and set return instruction.

7. S-Register (14 Bits)

The S-register holds the storage address during memory references. The address is received from the input-output or control units at the beginning of a storage access period. The contents of the S-register then are decoded to operate the memory selection system.

8. M-Register (25 Bits)

The M-register serves as an operand buffer for storage references. At the beginning of each access period, the M-register is cleared. During the read-access period, the contents of the desired memory location are sensed by the memory sense circuits and loaded into the M-register. The contents of M then are written back into memory; the M-register controls the inhibit circuits to restore the information in the memory location. During the write access period, the contents of the desired memory location are sensed by the memory sense circuits, but not loaded into the M-register. The M-register is loaded from the input-output or control units; then the contents of M are written into the desired memory location. Twenty-four bits are transmitted between M and D or Z. The 25th bit is a parity check bit.

9. X-Register (24 or 30 bits)

The X-register is an exchange register which functions to transfer data and commands between the computer and peripheral equipment. For usage on a 30 bit channel, this register transfers data in two 15 bit sectors to or from two consecutive core locations. The transfer is made between the least significant 15 bits and location Y and the most significant sector and Y + 1.

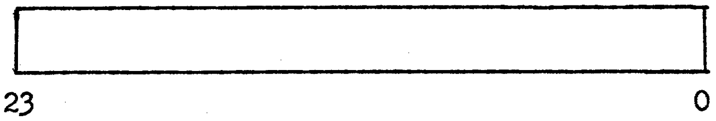
10. Z-Register (24 Bits)

The Z-register is a 24-bit register which functions to increment a count and an address in buffer mode control and the contents of the real time clock location.

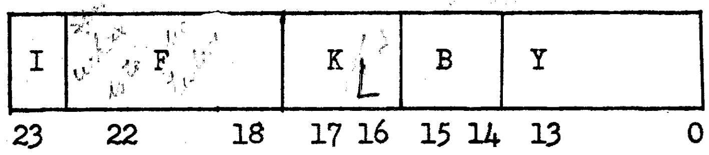
WORD FORMATS

1. Instruction

The bit positions of the DPS-2402 Computer word are numbered 0 to 23 from right to left.



Each instruction is contained in a 24--bit word. The basic format has an I, F, K, B, and Y field, as shown below.



The I field is used to specify indirect addressing on most instructions (see page 21).

The F field is used to specify the basic function or operation to be performed.

The K field contains a designator which further qualifies the function.

The B field is used to specify address modification by an index register.

The Y field is used to specify the base address of the operand to which the function applies.

2. K Designator

The K designators are used to specify the type of operand. Four options are available. The options apply after B and I fields have had their effect.

For K = 0 the Y as a 24-bit word is used. The mnemonic in the option field is a blank.

For K = 1 the right half of Y is used as the operand. The mnemonic in the option field is RH. When the fetch is made, bits 0 to 10 of Y become bits 0 to 10 of the operand, and bit 11 of Y is replicated through bits 11 to 23 of operand. When a store is performed, bit 11 of the operand becomes bit 11 of Y, bits 0 to 10 of the operand become bits 0 to 10 of Y and the left half of Y is not changed. Note that with 2's complement arithmetic, if a number does not overflow the half-word size, bit 11 of the operand will be the same as bit 23, the sign.

For K = 2 the left half of Y is used as the operand. The mnemonic in the option field is LH. When the fetch is made bits of 12 to 22 of Y become bits 0 to 10 of the operand and bit 23 of Y is replicated through bits 11 to 23 of the operand. When a store is performed bit 11 of the operand becomes bit 23 of Y, bits 0 to 10 of the operand become bits 12 to 22 of Y and the right half of Y is not changed.

For K = 3 the low 14 bits of the instruction word (i.e. the Y field) are used as the operand. The mnemonic in the option field is Y. No fetch is made, the bits of Y become bits 0 to 13 of the operand, and zero is replicated through bits 14 to 23 of the operand. When a store is performed Y is used as an address and bits 0 to 13 of the operand replace bits 0 to 13 of Y, leaving bits 14 to 23 of Y unchanged.

3. L Designator

Some skip and jump instructions use an L designator in place of the K designator. In this case, the operand always refers to a full word. The F designator specifies what is to be tested, and the L designator specifies the case for the abnormal sequence.

For L = 0 the skip or jump occurs when the sign of the operand is plus, (or zero). The mnemonic in the option field is P.

For L = 1 the skip or jump occurs when the sign of the operand is minus. The mnemonic in the option field is M.

For L = 2 the skip or jump occurs when the operand is zero. The mnemonic in the option field is Z.

For L = 3 the skip or jump occurs when the operand is non-zero. The mnemonic in the option field is N.

Special word formats for shift and input-output are described concurrent with the instructions. Other instructions required the I and/or K field for special designations are described concurrent with the function description. The M designator implies interpretation of bits 16 and 17 depends on the particular instruction.

4. B Designator

Three of the memory locations are used as index registers. These locations: 00001, 00002, and 00003 may be addressed normally, as well as being used as index registers.

Instructions whose function specifies an operation on the index registers are not susceptible to relative addressing. All other instructions with a B field are performed as follows:

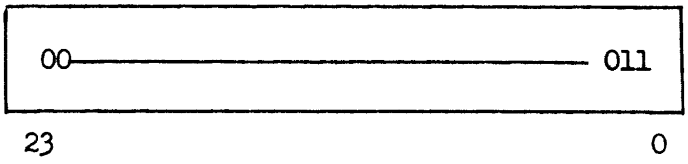
If B = 0 no indexing occurs.

If B = 1, the bits 0 to 13 of location 1 are fetched from memory and added to bits 0 to 13 of the instruction. A carry-to-bit 14 sets the memory range overflow indicator. Bits 0 to 13 of the sum are used as the instruction address. Relative addressing is specified in the assembly langauge by appending a comma and the index register number to the address expression.

If B = 2 or B = 3 the operation is similar to the case B = 1, using locations 2 and 3, respectively. Relative addressing is specified by append ", 2" or ", 3" to the address expression, Indexing requires one additional memory reference and is affected before indirect addressing. If an indirect address specifies an index register, one additional memory reference is required, and indexing is affected, before consideration is given the K option.

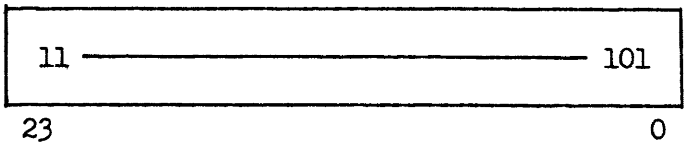
5. Numeric

Numeric data are stored in 2's complement form. This applies to both full and half words. Full-word positive numbers have a zero in bit position 23, while negative numbers have a 1 in this bit position (e.g., $+3_{10}$ appears in storage as



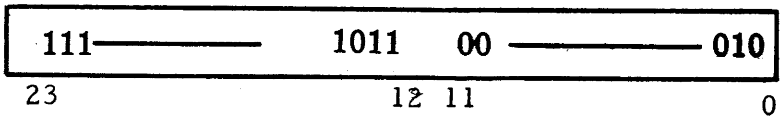
;

-3_{10} appears in storage as



).

Half-word numbers similarly have the sign bit as the left most bit of the appropriate half. For example, with the left half -5 and the right half +2 the word in storage appears as



The range of full-word numbers as integer and fraction is given below. Note that with 2's complement notation, zero is unique as a positive number and a negative sign (1 in bit position 23) with 23 trailing zeros representing the largest negative number.

Integer		Binary	Fractional	
8388607	$2^{23}-1$	011 —— 11 23 0	$1.0-2^{-23}$	.99999988
1	1	00 —— 01	2^{-23}	.00000012
0	0	00 —— 00	0	.0
-1	-1	11 —— 11	-2^{-23}	-.00000012
-8388608	-2^{23}	10 —— 00	-1.0	-1.0

INTERRUPT DEFINITION AND SOURCES

The term interrupt refers to the process of causing the computer to execute an instruction out of the programmed sequence. In the interrupt mode, the instruction address is not specified by the P-Register but is instead fetched from a fixed location corresponding to the interrupt source. The instruction at this location in most cases would be a Jump and Set Return (see page 62) and therefore causes normal execution of the proper subroutine. The interrupt may be ignored by placing a No Operation instruction (see page 64) in the fixed location. It should be noted that when an instruction is acquired from a fixed location, the normal incrementing of the P-Register does not take place.

An interrupt is processed after the completion of the current instruction if it is the highest priority interrupt waiting to be processed. Interrupt priorities are listed on page 84-86 and are briefly described below.

1. Fault Interrupt

If an instruction is executed which has bits 18 to 23 all zero (e.g., a FAULT instruction), and the FAULT REACTION switch is in the INTERRUPT position, the instruction in location 00020 is executed in interrupt mode.

2. Input/Output Parity Error Interrupt

If the computer detects a parity error while reading from memory in the process of input or output, buffered or express, and the I/O PARITY REACTION switch is in the INTERRUPT position, the instruction in location 00021 is executed in interrupt mode.

3. Program Parity Error Interrupt

If the computer detects a parity error while reading instructions or operands from memory and the PROGRAM PARITY ERROR REACTION switch is in the INTERRUPT position, the instruction in location 00022 is executed in interrupt mode.

4. Manual Interrupt

If the computer is running and the MANUAL INTERRUPT switch on the OPERATION panel is depressed, the instruction in location 00023 is executed in the interrupt mode.

5. Power Failure Interrupt

Loss of power is sensed 100 microseconds before voltages drop below safe operating levels. Upon sensing imminent power failure, the instruction in location 00026 is executed in interrupt mode.

6. Real Time Clock Interrupt

This interrupt is described in the Real Time Clock Section, this page.

7. Input/Output Channel Interrupts.

The input/output control provides for both internal (buffer completed) interrupt and external interrupt processing on all channels. See the section entitled, "Input/Output Characteristics" for details and priorities.

REAL TIME CLOCK

When the computer is running and the REAL TIME CLOCK switch is in the ON position, the contents of location 00024 is automatically incremented every 5 milliseconds. The incrementing process does not interrupt the execution of a program, but does require three memory cycles for each update. Various other clock rates to meet customer requirements can be supplied as an option.

An interrupt will occur immediately after the clock has been updated from all 1's to all 0's. This interrupt forces the instruction in location 00025 to be executed in interrupt mode. Since a full 24-bit word is used for the clock location, its 0 to 0 period is slightly less than 24 hours.

The programmer may use this interrupt feature to generate any accurate real time interval by pre-setting the contents of location 00024.

ASSIGNED CORE MEMORY LOCATION

OCTAL	
00000	Master Clear - Start Location
00001	Index Register #1
00002	Index Register #2
00003	Index Register #3
00004) to) 00017)	Reserved for Index Register Expansion
00020	Fault Interrupt Location
00021	Parity Interrupt Location, I/O
00022	Parity Interrupt Location, Program
00023	Manual Interrupt Location
00024	Real Time Clock
00025	Clock = 0 Interrupt Location
00026	Power Failure Interrupt Location
0030	Input Buffer Status, Channels 0 - 7
0031	Output Buffer Status, Channels 0 - 7
0032	Input Buffer Status, Channels 8 - 15
0033	Output Buffer Status, Channels 8 - 15
0040 to 0057	Input Buffer Control Word for Channels 0 through 15
0060 to 0077	Output Buffer Control Word for Channels 0 through 15

OCTAL	
0100 to 0117	Input Buffer Complete Interrupt Locations for Channels 0 through 15
0120 to 0137	Output Buffer Complete Interrupt Locations for Channels 0 through 15.
0140 to 157	External Interrupt Status Word Address for Channels 0 through 15
160 to 177	Input External Interrupt Locations for Channels 0 through 15
0200 to 0217	Output External Interrupt Locations for Channels 0 through 15

Sense Switch Assignments

Bit	Manual Switch	Input Data Available	Output Data Lines Available	Special
	GP = 0	GP = 1	GP = 2	GP = 3
0	Switch 0	Channel 0	Channel 0	Arithmetic Overflow
1	Switch 1 (S1)	Channel 1	Channel 1	Division Error
2	Switch 2 (S2)	Channel 2	Channel 2	Shift Parity
3	Switch 3	Channel 3	Channel 3	Sign A
4	Switch 4	Channel 4	Channel 4	Sign of Q
5	Switch 5	Channel 5	Channel 5	Contents of A = 0
6	Switch 6	Channel 6	Channel 6	Memory Range Overflow
7	Switch 7	Channel 7	Channel 7	Master Lockout
8	Switch 8	Channel 8	Channel 8	Program Flag Light (set)
9	Switch 9	Channel 9	Channel 9	Not Assigned
10	Switch 10	Channel 10	Channel 10	
11	Switch 11	Channel 11	Channel 11	
12	Clock - ON, OFF	Channel 12	Channel 12	
13	Parity Reaction	Channel 13	Channel 13	
14	Parity Reaction I/O	Channel 14	Channel 14	
15	Fault Reaction	Channel 15	Channel 15	

Machine Instruction Mnemonics in Alphabetical Order

Assembler Mnemonic	Function Code	Definitions	Page
ADD	24	Add	33
✓ADD1	1 03	Add one to storage	53
✓AND	10	Logical AND	38
BIMT	1 05	Input buffer mode terminate	71
BIMTI	1 05	Input buffer mode terminate and interrupt	72
BIN	1 05	Establish buffer input mode	68
BBIN	1 05	Establish block buffer input mode	68
BINI	1 05	Establish buffer input mode with interrupt	69
BBINI	1 05	Establish block buffer input mode with interrupt	69
BOMT	1 05	Output buffer mode terminate	71
BOMTI	1 05	Output buffer mode terminate and interrupt	71
BOT	1 05	Establish buffer output mode	69
BBOT	1 05	Establish block buffer output mode	70
✓ BOTI	1 05	Establish buffer output mode with interrupt	70
BBOTI	1 05	Establish block buffer output mode with interrupt	70
CIR	1 04	Clear interrupt request	73
CMP	16	Compare	53
✓CYS	17	Compare with Y and skip	54
DIV	27	Divide	35
✓ELS	1 04	Enter lockout status	75
✓ENA	21	Enter the A register	28
✓ENB	22	Enter the index register	30
✓ENQ	20	Enter the Q register	29
EOR	13	Exclusive OR	39
✓EXF	1 06	External function	66
✓EXI	1 00	Express input	66
✓EXO	1 01	Express output	66
FAULT	0 00	Fault	63
HEY	0 07	Turn on program flag-light	59

Figure 3. Westinghouse DPS-2402 Instruction Repertoire

Bit Number						Function Mnemonic			Assembly Language Mnemonics			
23	22	16	17 16	15 14	13 0		PAGE		DESIGNATOR FOR	BITS 16, 17		
0	0	0				FAULT	63	FAULT	FAULT			
1	0	0	CHANNEL		Y	EXI	66	(CHANNEL) → Y & input ack.	EXI			
0	0	1	S		S N	SHF	41	Shift	SCA SCAQ RBAQ TLY SHR SHR Q SHR AQ SHR C SHR QC SHR AQC LGR LGR Q LGR AQ LGR C LGR QC LGR AQC SHL SHL Q SHL AQ SHL C SHL QC SHL AQC LGL LGL Q LGL AQ LGL C LGL QC LGL AQC			
1	0	1	CHANNEL		Y	EXO	66	(Y) → CHANNEL & output ack.	EXO			
0	0	2	NOT ASSIGNED									
1	0	2	NOT ASSIGNED									
0	0	3	L	B	Y	JDB	50	(B) - 1 → B, cond. jump	0 JDB P	1 JDB M	2 JDB Z	3 JDB N
1	0	3	K	B	Y	RAO	53	(Y) + 1 → Y	0 ADD1	1 ADD1 RH	2 ADD1 LH	3 ADD1 Y
0	0	4	L	B	Y	SSK	55	(Y), cond. skip	0 SSK P	1 SSK M	2 SSK Z	3 SSK N
1	0	4	M	B	Y	CIS	72	Change interrupt status	0 CIR	1 SAL	2 ELS	3 SLS
0	0	5	L	B	Y	SSH	55	(Y) shifted left 1, cond. skip	0 SSH P	1 SSH M	2 SSH Z	3 SSH N
1	0	5	CHANNEL			CBM	68		BIN, BINI, BOT, BOTI, BOMT, BOMTI, BIMT, BIMTI, SOBS, SIBS, BBOT, BBOTI, BBIN, BBINI			
0	0	6	L	B	Y	RSO	52	(Y) - 1 → Y, cond. skip	0 SUB1 P	1 SUB1 M	2 SUB1 Z	3 SUB1 N
1	0	6	CHANNEL		Y	EXF	66	(Y) → CHANNEL & EXP signal	EXF			
0	0	7	GP	BITS		SSW	56	Senseline Test	0 SSW	1 IDA	2 ODA	3 SOF, SDE, SOP, SAM, SQM, SAZ, SMO SMS, HEY
1	0	7	M	B	Y	NXN	65 32 63		0 NEG	1 XAQ	2 JHT	3 NOP
I	1	0	K	B	Y (Y)	AND AND	38	(A) ^ (Y) → A	0 AND	1 AND RH	2 AND LH	3 AND Y *
I	1	1	K	B	Y (Y)	IOR IOR	38	(A) v (Y) → A	0 IOR	1 IOR RH	2 IOR LH	3 IOR Y *
I	1	2	K	B	Y (Y)	SCL SCL	39	Clear (A) for 1's of (Y)	0 SCL	1 SCL RH	2 SCL LH	3 SCL Y *
I	1	3	K	B	Y (Y)	EOR EOR	39	(A) ⊕ (Y) → A	0 EOR	1 EOR RH	2 EOR LH	3 EOR Y *
I	1	4	K	B	Y (Y)	SBT SBT	40	(Y) → A for 1's of (Q)	0 SBT	1 SBT RH	2 SBT LH	3 SBT Y *
I	1	5	L	B	Y (Y)	SCP SCP	56	(A) - [(Q) ^ (Y)], cond skip	0 SCP P	1 SCP M	2 SCP Z	3 SCP N
I	1	6	K	B	Y (Y)	CMP CMP	53	if (A) < (Y) skip	0 CMP	1 CMP RH	2 CMP LH	3 CMP Y *
I	1	7	L	B	Y (Y)	CYS CYS	54	(A) - (Y), cond. skip	0 CYS P	1 CYS M	2 CYS Z	3 CYS N
I	2	0	K	B	Y (Y)	ENQ ENQ	29	Enter Q, (Y) → (Q)	0 ENQ	1 ENQ RH	2 ENQ LH	3 ENQ Y *
I	2	1	K	B	Y (Y)	ENA ENA	28	Enter A, (Y) → A	0 ENA	1 ENA RH	2 ENA LH	3 ENA Y *
I	2	2	K	B‡	Y (Y)	ENB ENB	30	Enter B, (Y) → B	0 ENB	1 ENB RH	2 ENB LH	3 ENB Y *
I	2	3	K	B	Y (Y)	RAD RAD	37	(A) + (Y) → Y	0 RAD	1 RAD RH	2 RAD LH	3 RAD Y
I	2	4	K	B	Y (Y)	ADD ADD	33	(A) + (Y) → A	0 ADD	1 ADD RH	2 ADD LH	3 ADD Y *
I	2	5	K	B	Y (Y)	SUB SUB	34	(A) - (Y) → A	0 SUB	1 SUB RH	2 SUB LH	3 SUB Y *
I	2	6	K	B	Y (Y)	MUL MUL	34	(Q) x (Y) → AQ	0 MUL	1 MUL RH	2 MUL LH	3 MUL Y *
I	2	7	K	B	Y (Y)	DIV DIV	35	(AQ) ÷ (Y) → Q, A	0 DIV	1 DIV RH	2 DIV LH	3 DIV Y *
I	3	0	K	B	Y (Y)	STQ STQ	31	Store Q, (Q) → Y	0 STQ	1 STQ RH	2 STQ LH	3 STQ Y
I	3	1	K	B	Y (Y)	STA STA	30	Store A, (A) → Y	0 STA	1 STA RH	2 STA LH	3 STA Y
I	3	2	K	B‡	Y (Y)	STB STB	32 64	Store B, (B) → Y Store Zero → Y	0 STB B=0 STZ	1 STB RH B=0 STZ RH	2 STB LH B=0 STZ LH	3 STB Y B=0 STZ Y
I	3	3	K	B	Y (Y)	RSB RSB	37	(A) - (Y) → Y	0 RSB	1 RSB RH	2 RSB LH	3 RSB Y
I	3	4	K	B	(Y)	XEQ INOP	65 63	Execute (Y)	0 INOP	1 NOT ASSIGNED	2 NOT ASSIGNED	3 XEQ
I	3	5	M	B	Y (Y)	JPA JPA	60	(A), cond. jump	0 JMP AP	1 JMP AM	2 JMP AZ	3 JMP QP
I	3	6	M	B	Y (Y)	JPM JPM	61	misc. cond. jump	0 JMP	1 JMP S1	2 JMP S2	3 JMP CML Not Indexable
I	3	7	M	B	Y (Y)	JSR JSR	62	(P) → Y, jump to (Y) + 1	0 JSR	1 JSR AM	2 JSR AZ	3 JSR SML Not Indexable

L designator condition

P: L = 0 ( ) = + M: L = 1 ( ) = - Z: L = 2 ( ) = 0 N: L = 3 ( ) ≠ 0

I designator — To specify indirect addressing in the assembly language enclose address field in parenthesis

I = 0 no direct addressing I = 1 indirect addressing

K designator

blank:	K = 0	USE FULL WORD
RH:	K = 1	USE RIGHT HALF
LH:	K = 2	USE LEFT HALF
Y:	K = 3	USE Y FIELD

B designator

blank:	B = 0	NO INDEXING
,1:	B = 1	Y + (B1) → Y
,2:	B = 2	Y + (B2) → Y
,3:	B = 3	Y + (B3) → Y

S designator

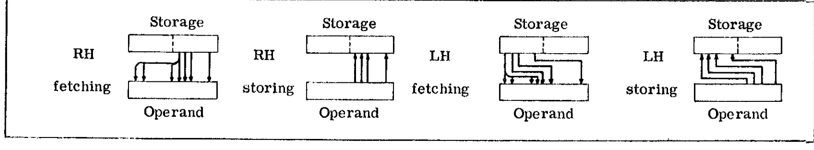
b_{17}	b_{16}		b_{15}		b_{14}		b_{11}	
0	0	see SCA and RBAQ	1	circular	0	logical	0	left
1	0	use A	0	linear	1	arithmetic	1	right
0	1	use Q						
1	1	use AQ	N is 6 bit shift count.					

GP DESIGNATOR

SEE OPTIONS TO RIGHT

M DESIGNATOR

Interpretation dependent on particular instruction.



* The low order 14 bits of the instruction with 10 leading zeros are used as the operand providing B = 0. If B ≠ 0, the low order 14 bits of the designated index register are used as the operand.

+ The low order 14 bits of the location called out by the address are used as the operand. The high order 10 bits are zero after fetching and are not disturbed while storing.

‡ The B field designates the index register to which the function applies unless indirect addressing is specified. Indirect addressing uses the B field to designate indexing and uses the B field of the indirect address word as part of the function.

Machine Instruction Mnemonics in Alphabetical Order (Continued)

Assembler Mnemonic	Function Code	Definitions	Page
IDA	0 07	Input data available	57
INOP	34	Indirect no operation	64
IOR	11	Inclusive OR	38
✓JDB	0 03	Jump and decrement an index register	50
✓JHT	1 07	Jump and halt	63
✓JMP	35	Jump conditioned on arithmetic registers	60
✓JMP	36	Jump	61
✓JSR	37	Jump and set return	62
✓LGL	0 01	Logical left shift	46
LGR	0 01	Logical right shift	45
MUL	26	Multiply	34
NEG	1 07	Negate the A register	65
✓NOP	1 07	No operation	64
ODA	0 07	Output data available	57
RAD	23	Replace add	37
RBAQ	0 01	Reverse bits of the A and Q registers	50
RSB	33	Replace subtract	37
SAL	1 04	Set additional lockouts	73
✓SAM	0 07	Skip on the A register minus	58
SAZ	0 07	Skip on the A register equal zero	59
SBT	14	Substitute bits	40
SCA	0 01	Scale the A register	49
SCAQ	0 01	Scale the AQ registers	49

Machine Instruction Mnemonics in Alphabetical Order (Continued)

Assembler Mnemonic	Function Code		Definitions	Page
SCL		12	Selective clear	39
SCP		15	Selective compare	56
SDE	0	07	Skip on division error	58
SHL	0	01	Shift left	41
SHR	0	01	Shift right	41
SIBS	1	05	Store input buffer status	73
SLS	1	04	Store lockout status	73
SMO	0	07	Skip on memory overflow	59
SMS	0	07	Skip if master lockout not set	59
SOBS	1	05	Store output buffer status	72
SOF	0	07	Skip on overflow	58
SOP	0	07	Skip on odd shift parity	58
SQM	0	07	Skip on the Q register minus	59
SSH	0	05	Storage skip and shift	55
✓SSK	0	04	Storage skip	55
✓ SSW	0	07	Sense switch skip	56
✓STA		31	Store the A register	30
✓STB		32	Store an index register	32
✓STQ		30	Store the Q register	31
✓STZ		32	Store zero	64
✓ SUB		25	Subtract	34
SUB1	0	06	Subtract one from storage	52
TLY (Opt.)	0	01	Tally 1's in the A register	50
XAQ	1	07	Exchange A and Q register contents	32
XEQ		34	Execute (Y)	65

used to supply additional
[illegible]

Pseudo Instruction Mnemonics in Alphabetical Order

Assembler Mnemonic	Function Code		Definition	Page
ABS			Begin absolute mode of assembly	See DPS-2402 Programmers Reference Manual
ARG	1	07	Argument in calling sequence	
BCI			Binary coded information	
✓BCW			Buffer control word	
✓BSS			Block starting symbol	
BTS			Block terminating symbol	
CALL	0	37	Call subroutines	
COMON			Common storage allocation	
DEC			Decimal information	
✓END			End of this assembly	
ENTRY	1	07	Entry to a subroutine	
EXIT	1	35	Exit from a subroutine	
JOB			Entry to main program	
LIBENT			Library entry name	
✓OCT			Octal information	
ORG			Set instruction sequence to new origin	
REL			Begin relocatable mode of assembly	
REP			Repeat assembly of lines	
SET			Set the address associated with a symbol	
TITLE			Page title for this assembly	

DETAILED DESCRIPTION OF INSTRUCTIONS

GENERAL

Instructions are presented in 8 categories: data transfer, arithmetic, logical, shift, increment, conditional skip, jump, and input-output. The assembler mnemonic, descriptive name, and field designation are presented for each instruction. Timing is given as the number of memory accesses to fetch the instruction and operand and to perform the instruction without indexing or indirect addressing. One memory access is added for indirect addressing and one memory access for each address modification by an index register. The first instruction in a category is described in more detail to clarify options and mnemonics. A composite list of all instructions appears in figure 3.

Multiple-level Indirect Addressing

Execution of an indirect instruction is identical to the execution of its direct equivalent once the effective address has been attained. The indirect cycle can run indefinitely and is stopped by the acquisition of any direct instruction code. The indirect address at every level is modified by the contents of B specified at that level. Indirect addressing is specified in the assembly language by enclosing the address field in parenthesis. This has the conventional meaning for "( )" since the address of the operand is the contents of the location called out by the address expression. Timing: 1 cycle per level.

Address Indexing

For those instructions which allow indexing, the specification of a B modifier will cause the contents of the designated location to be added to the direct or indirect Y field and the sum serves as the operand address or the new indirect address. Each indirect address level can indicate a different or no B modifier.

Timing: 1 cycle for direct addressing
1 cycle per level for indirect addressing

Data Transfer Instructions (ENA, ENQ, ENB, STA, STQ, STB, XAQ)

ENA Enter A

I	21	K	B	Y
23	22 18	17 16	15 14	13 0

When no option is specified K = 0.

The contents of location Y replace the contents of the A register. The contents of location Y are unchanged. Indexing and indirect addressing apply in all combinations. No other registers are changed and the next instruction is taken in sequence. Timing: 2 cycles.

ENA RH Enter A from Right Half

I	21	1	B	Y
23	22 18	17 16	15 14	13 0

The contents of bits 0 to 10 of location Y replace the contents of bits 0 to 10 of A. The contents of bit 11 (sign) of location Y is replicated in bits 11 to 23 of A. Thus the sign and integer magnitude of the right half are represented in 2's complement form as a full 24-bit word in the A register. The contents of location Y are unchanged. Indexing and indirect addressing apply in all combinations. The RH option applies to the effective address after indexing and indirect addressing. No other registers are changed and the next instruction is taken in sequence. Timing: 2 cycles.

ENA LH Enter A from Left Half

I	21	2	B	Y
23	22 18	17 16	15 14	13 0

The contents of bits 12 to 22 of location Y replace the contents of bits 0 to 10 of A. The contents of bits 23 (sign) of location Y is replicated in bits 11 to 23 of A. See ENA RH for further details.

ENA Y Enter A with Bits of Y

I	21	3	B	Y
23	22 18	17 16	15 14	13 0

Bits 0 to 13 of this instruction (i.e., the Y field) with 10 leading zero bits replace the contents of the A register. Thus the address field of the ENA instruction is expanded into a full word positive integer of the same magnitude in A. If an index register is specified, bits 0 to 13 of the index register with 10 leading zeros replace the contents of the A register. If indirect addressing is specified the net result is to place the effective address in bits 0 to 13 of A and zero bits 14 to 23 of A. No other registers are changed and the next instruction is taken in sequence.

Timing: 2 cycles.

ENQ Enter Q

I	20	K	B	Y
23	22 18	17 16	15 14	13 0

The contents of location Y replace the contents of the Q register. The contents of Y are unchanged. Indexing and indirect addressing apply in all combinations. No other registers are changed and the next instruction is taken in sequence. Timing: 2 cycles. The options RH, LH, and Y apply as described under ENA.

ENB Enter B

I	22	K	B	Y
23	22 18	17 16	15 14	13 0

The contents of location Y replace the contents of the memory location specified by the B field. The Y field is not modified by the specified index register. If indirect addressing is called for, it proceeds as usual with Y being modified by the index register designated in the instruction. The index register designated in the contents of location Y is then loaded with the contents of the unindexed indirect address. Note: location 00000 may be loaded even though it is not considered an index register. No other registers are changed, and the next instruction is taken in sequence. Memory range overflow is possible if B $\neq$ 0 and indirect addressing is used. Timing: 3 cycles.

The options RH and LH apply as described under ENA. When the Y option is used, bits 0 to 13 of this instruction replace 0 to 13 of the specified index register. Bits 14 to 23 of the index register are set to zero.

STA Store A

I	31	K	B	Y
23	22 18	17 16	15 14	13 0

When no option is specified K = 0. The contents of A replace the contents of location Y. The A register is unchanged. Indexing and indirect addressing apply in all combinations. No registers are changed, and the next instruction is taken in sequence. Timing: 2 cycles.

STA RH Store A in Right Half

I	31	1	B	Y
23	22 18	17 16	15 14	13 0

The contents of bits 0 to 10 of A replace the contents of bits 0 to 10 of location Y. The contents of bit 11 (the same as bit 23, the sign if the half word has not overflowed) of A replace the contents of bit 11 of location Y. Bits 12 to 23 of location Y are unchanged. See STA for further details.

STA LH Store A in Left Half

I	31	2	B	Y
23	22 18	17 16	15 14	13 0

The contents of bits 0 to 10 of A replace the contents of bits 12 to 22 of location Y. The contents of bit 11 of the A register replace the contents of bit 23 of location Y. Bits 0 to 11 of location Y are unchanged. See STA for further details.

STA Y Store A in Address

I	31	3	B	Y
23	22 18	17 16	15 14	13 0

The contents of bits 0 to 13 of the A register replace the contents of bits 0 to 13 of location Y. Bits 14 to 23 of location Y are unchanged. See STA for further details.

STQ Store Q

I	30	K	B	Y
23	22 18	17 16	15 14	13 0

Contents of Q replace the contents of location Y. The Q register is unchanged. Indexing and indirect addressing apply in all combinations. No registers are changed and the next instruction is taken in sequence. Timing: 2 cycles.
The options RH, LH and Y apply as described in STA.

STB Store B

I	32	K	B	Y
23	22 18	17 16	15 14	13 0

The contents of the specified index register replace the contents of location Y. The Y field is not modified by the specified index register. If indirect addressing is called for, it proceeds as usual with Y being modified by index register designated. When an effective address has been reached the contents of index register specified are then stored at the unindexed address. The case where B = 0 is described under STZ, the store zero instruction. No registers are changed and the next instruction is taken in sequence. Memory range overflow is possible if B ≠ 0 and indirect addressing is used. Timing: 3 cycles. The options RH, LH, and Y apply as described under STA.

XAQ Exchange A and Q and Jump

1	07	1	B	Y
23	22 18	17 16	15 14	13 0

The contents of the A and Q registers are interchanged. No memory locations are changed. The next instruction is taken from location Y. Indexing applies to modify the location from which the next instruction is taken. Indirect addressing is not available. Timing: 2 cycles.

In the assembly language, if the address field is blank an address will be filled in to be the next location in sequence.

Arithmetic Instructions (ADD, SUB, MUL, DIV, RAD, RSB)

ADD

Add

I	24	K	B	Y
23	22 18	17 16	15 14	13 0

When no option is specified K = 0. the contents of location Y are added to the contents of the A register. Both numbers are treated as full word 2's complement numbers. The result is a full word 2's complement number in the A register. The Q register and the contents of location Y remain unchanged. Indexing and indirect addressing apply in all combinations. Overflow is possible. The next instruction is taken in sequence. Timing: 2 cycles.

ADD RH

Add Right Half

I	24	1	B	Y
23	22 18	17 16	15 14	13 0

The contents of bits 0 to 10 of location Y are used as bits 0 to 10 of the operand. The contents of bit 11 (sign) of location Y are replicated in bits 11 to 23 of the operand. The resulting full word 2's complement operand is added to the contents of the A register. (See ADD for further details.)

ADD LH

Add Left Half

I	24	2	B	Y
23	22 18	17 16	15 14	13 0

The contents of bits 12 to 22 of location Y are used as bits 0 to 10 of the operand. The contents of bit 23 (sign) of location Y is replicated in bits 11 to 23 of the operand. The resulting full word 2's complement operand is added to the contents of the A register. (See ADD for further details).

ADD Y   Add Bits of Y

I	24	3	B	Y
23	22 18	17 16	15 14	13 0

Bits 0 to 13 of this instruction (i.e., the Y field) with 10 leading zero bits are used as a full word 2's complement operand providing B = 0. If indexing is specified and indirect addressing is not used, bits 0 to 13 of the specified index registers are used as the operand. (See ADD for further details.)

SUB   Subtract

I	25	K	B	Y
23	22 18	17 16	15 14	13 0

The contents of location Y are subtracted from the contents of the A register. Both numbers are treated as full word 2's complement numbers. The result is a full word 2's complement number. The Q register and the contents of location Y remain unchanged. Indexing and indirect addressing apply in all combinations. Overflow is possible. The next input is taken in sequence. Timing: 2 cycles.

The options RH, LH, and Y apply as described under ADD RH, ADD LH, and ADD Y, respectively, with the operand being subtracted from the contents of the A register.

MUL   Multiply

I	26	K	B	Y
23	22 18	17 16	15 14	13 0

The contents of location Y are multiplied by the contents of the Q register. Before multiplication the A register is cleared. The operand and contents of the Q register are treated as full word 2's complement numbers. The high order part of the product is retained in the A register, and the low order part of the product is retained in the Q register. Bit 23 of the A register is the sign of the product as is bit 23 of the Q register. The only exceptions to this rule are when the contents of A are negative and the magnitude of A is zero (and thus positive), or the contents of Q are negative and the magnitude of A is zero. (and thus positive). Therefore, the Q register may be stored as the signed product of two integers with less than 23 significant bits, or the A register may be stored as the signed product of two fractions with less than 23 significant bits without loss of accuracy, etc. In general, the binary point of the product is n bits to the left of bit 0 where n is computed as the sum of the number of bits the binary points are to the left of bit 0 in the multiplier and multiplicand. Indexing and indirect addressing apply in all combinations. The overflow indicator is not changed by this instruction. The next instruction is taken in sequence. Timing: 6 cycles.

The options RH, LH and Y apply as described under ADD RH, ADD LH, and ADD Y, respectively, with the operand being multiplied by the contents of the Q register.

DIV Divide

I	27	K	B	Y
23	22 18	17 16	15 14	13 0

The contents of the A and Q register, treated as a 46-bit dividend; are divided by the contents of location Y. It is assumed A and Q are full word 2's complement numbers with the same sign. The operand is also treated as a full word 2's complement number. The signed quotient is retained in the Q register in 2's complement form and the remainder is retained in the A register in 2's complement form with the same sign as Q. The only exceptions to this rule are when the quotient is negative and the remainder is zero, in which case, the sign of the A register is plus. The division error indicator is turned on if significant bits of the quotient would be lost (i.e., if the magnitude in the A register is greater than or equal the magnitude in location Y). If division error is detected, the A and Q registers remain unchanged. The Q register may be stored as the integer quotient of two integers or the fractional quotient of two integers or the fractional quotient of two fractions, etc. In general, the binary point for the quotient is n bits to the left of bit 0 when n is computed as the number of bits the binary point of the dividend is to the left of bit 0 minus the number of bits the binary point of the divisor is to the left of bit 0. The binary point of the remainder corresponds to the binary point of the dividend.

Indexing and indirect addressing apply in all combinations. The overflow indicator is not changed by this instruction. The next instruction is taken in sequence. Timing: 11 cycles.

The options RH, LH, and Y apply as described under ADD RH, ADD LH, and ADD Y, respectively, with the operand being the divisor.

RAD Replace ADD

I	23	K	B	Y
23	22 18	17 16	15 14	13 0

Same as ADD except the sum replaces the contents of location Y and the contents of the A register are unchanged. Timing: 3 cycles.

The options RH and LH apply as described for ADD RH and ADD LH with the same bits being replaced by the sum as are used as an operand. When the Y option is used, bits 0 to 13 of location Y are fetched as an operand. Bits 0 to 13 of the sum of the operand and the A register replace bits 0 to 13 of the location from which the operand was fetched. Use of index register applies to all options.

RSB Replace Subtract

I	33	K	B	Y
23	22 18	17 16	15 14	13 0

Same as SUB, except the difference replaces the contents of location Y, and the contents of the A register are unchanged. Timing: 3 cycles.

The options RH, LH, AND Y apply as described for RAD.

Logical Instructions (AND, IOR, EOR, SCL, SBT)

In combination and with appropriate masks, the following instructions can be used to obtain all 16 logical functions (bit by bit) of two words.

AND Logical AND

I	10	K	B	Y
23	22 18	17 16	15 14	13 0

The contents of location Y and the contents of the A register are examined bit by bit for all 24 bits. Whenever both are 1 in a bit position, a 1 is placed in that position of the A register; otherwise a 0 is placed in the A register for that position. The contents of the Q register and location Y are unchanged. Indexing and indirect addressing apply in all combinations. The next instruction is taken in sequence. Timing: 2 cycles.

The options RH, LH, and Y apply as described under ADD RH, ADD LH, and ADD Y respectively for fetching the operand.

IOR Logical Inclusive OR

I	11	K	B	Y
23	22 18	17 16	15 14	13 0

The contents of location Y and the contents of the A register are examined bit by bit for all 24 bits. Whenever either or both are 1 in a bit position a 1 is placed in that position of the A register, otherwise a 0 is placed in the A register for that position. The contents of the Q register and location Y are unchanged. Indexing and indirect addressing apply in all combinations. The next instruction is taken in sequence. Timing: 2 cycles.

The options RH, LH, and Y apply as described under ADD RH, ADD LH, and ADD Y, respectively, for fetching the operand.

EOR Logical Exclusive OR

I	13	K	B	Y
23	22 18	17 16	15 14	13 0

The contents of location Y and the contents of the A register are examined bit by bit for all 24 bits. Whenever both are 0 or both are 1 in a bit position a 0 replaces the contents of that position in the A register, otherwise a 1 replaces the contents of that position in the A register. The contents of the Q register and location Y are unchanged. Indexing and indirect addressing in any combination is allowed. The next instruction is taken in sequence. Timing: 2 cycles.

The options RH, LH, and Y apply as described under ADD RH, ADD LH, and ADD Y, respectively, for fetching the operand.

SCL Selective Clear

I	12	K	B	Y
23	22 18	17 16	15 14	13 0

The contents of location Y and the contents of the A register are examined bit by bit for all 24 bits. Whenever there is a 1 in a bit position of the contents of location Y a zero is placed in the corresponding position of the A register; otherwise the A register is not changed for that position. The contents of the Q register and location Y are unchanged.

Indexing and indirect addressing apply in all combinations. The next instructions are taken in sequence. Timing: 2 cycles.

The options RH, LH, and Y apply as described under ADD RH, ADD LH, and ADD Y, respectively, for fetching the operand.

SBT Substitute.

I	14	K	B	Y
23	22 18	17 16	15 14	13 0

The Q register is used as a 24-bit mask. For each bit position in the Q register which is a 1, the contents of the corresponding bit of location Y replace the bit of the A register for that position. The A register is unchanged in those positions where the Q register has zeros. The contents of the Q register and location Y are unchanged. Indexing and indirect addressing in any combination is allowed. The next instruction is taken in sequence. Timing: 2 cycles.

The options RH, LH, and Y apply as described under ADD RH, ADD LH, and ADD Y, respectively, for fetching the operand.

Shift Instructions (SHR, SHL,LGR, LGL, SCA, SCAQ, RBAQ, TLY)

The basic format for the shift instruction is

0	01	S	▨	S	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The five bits of S are used to specify the 24 possible shift instructions and four special instructions. For shift instructions, the 6 bits of N specify the number of places to shift. The S designator in position 11 controls whether the shift is right or left. Thus a right-half store into a shift instruction can set the count and direction of a shift. Neither indirect addressing nor indexing is allowed. Timing is proportional to the number of positions shifted; 1.3 + N/6 cycles.

SHR Shift Right

0	01	1001	▨	1	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the A register are shifted right the number of positions specified by N. Bits shifted out of bit position 0 are lost. Bit 23 (the sign) of A is not shifted, and positions vacated on the left are filled from bit 23. Thus when the contents of the A register are considered as a 2's complement number, the effect of the shift is to divide by 2^N while retaining the 2's complement form. The Q register is unchanged. The next instruction is taken in sequence. Timing: 1 to 5 cycles.

SHL Shift Left

0	01	1001	▨	0	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the A register are shifted left the number of positions specified by N. Bits shifted out of position 22 are lost and Bit 23 (the sign) of A is not shifted. Positions vacated on the right are filled with zeros. The overflow indicator is turned on if a significant bit leaves position 22 when the contents of the A register are considered at a 2's complement number. The effect of the shift is to multiply by 2^N while retaining the 2's complement form. The Q register is unchanged. The next instruction is taken in sequence. Timing: 1 to 5 cycles.

SHR Q Shift Q Right

0	01	0101	▨	1	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the Q register are shifted right the number of positions specified by N. Bits shifted out of bit position 0 are lost. Bit 23 (the sign) of Q is not shifted, and positions vacated on the left are filled from bit 23. Thus, when the contents of the Q register are considered as a 2's complement number, the effect of the shift is to divide by 2^N while retaining the 2's complement form. The A register is unchanged. The next instruction is taken in sequence. Timing: 1 to 5 cycles.

SHL Q Shift Q Left

0	01	0101	▨	0	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the Q register are shifted left the number of positions specified by N. Bits shifted out of bit position 22 are lost. Bit 23 (the sign) of Q is not shifted. Positions vacated on the right are filled with zeros. The overflow indicator is turned on if a significant bit leaves position 22 when the contents of the Q register are considered as a 2's complement number. The effect of the shift is to multiply by 2^N while retaining the 2's complement form. Timing: 1 to 5 cycles.

SHR AQ Shift A and Q Right

0	01	1101	▨	1	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the A and Q register are shifted right together the number of positions specified by N. Bits shifted out of position 0 to the Q register are lost. Bits shifted out of position 0 to the A register enter position 22 of the Q register. Bit 23 of the A and Q registers are not shifted. Positions vacated on the left of the A register are filled from bit 23 of the A register. Thus when the contents of the AQ register are considered as a double length 2's complement number, the effect of the shift is to divide the double length number by 2^N while retaining the 2's complement form. The next instruction is taken in sequence. Timing: 1 to 9 cycles.

SHL AQ   Shift A and Q Left

0	01	1101	▨	0	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the A and Q register are shifted left together the number of positions specified by N. Bits shifted out of position 22 of the A register are lost. Bits shifted out of position 22 of the Q register enter position 0 of the A register. Positions vacated at the right of the Q register are filled with zeros.

The overflow indicator is turned on if a significant bit leaves position 22 when the contents of the AQ register are considered as a double length 2's complement number. The effect of the shift is to multiply the double length number by 2^N while retaining the 2's complement form. The next instruction is taken in sequence. Timing: 1 to 9 cycles.

SHR C Shift Right Closed

0	01	1011	▨	1	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the A register are shifted right the number of positions specified by N. Bits shifted out of position 0 enter position 22. Position 23 is not shifted. The Q register is unchanged and no overflow is possible. The next instruction is taken in sequence. Timing: 1 to 5 cycles.

SHL C Shift Left Closed

0	01	1011	▨	0	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the A register are shifted left the number of positions specified by N. Bits shifted out of position 22 enter position 0. Position 23 is not shifted. The Q register is unchanged and no overflow is possible. The next instruction is taken in sequence. Timing: 1 to 5 cycles.

SHR QC Shift Q Right Closed

0	01	0111	▨	1	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the Q register are shifted right the number of positions specified by N. Bits shifted out of position 0 enter position 22. Position 23 is not shifted. The A register is unchanged and no overflow is possible. The next instruction is taken in sequence. Timing: 1 to 5 cycles.

SHL QC Shift Q Left Closed

0	01	0111	▨	0	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the Q register are shifted left the number of positions specified by N. Bits shifted out of position 22 enter position 0. Position 23 is not shifted. The A register is unchanged and no overflow is possible. The next instruction is taken in sequence. Timing: 1 to 5 cycles.

SHR AQC

Shift A and Q Right Closed

0	01	1111	▨	1	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the A and Q are shifted right together the number of positions specified by N. Bits shifted out of position 0 of the A register enter position 22 and Q register. Bits shifted out of position 0 of the Q register enter position 22 of the A register. Position 23 of A and Q registers is not shifted. No overflow is possible. The next instruction is taken in sequence. Timing: 1 to 9 cycles.

SHL AQC

Shift A and Q Left Closed

0	01	1111	▨	0	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the A and Q registers are shifted left together the number of positions specified by N. Bits shifted out of position 22 of the A register enter position 0 of the Q register. Bits shifted out of position 22 of the Q register enter position 0 of the A register. Position 23 of A and Q registers is not shifted. No overflow is possible. The next instruction is taken in sequence. Timing: 1 to 9 cycles.

LGR

Logical Right Shift

0	01	1000	▨	1	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the A register are shifted right the number of positions indicated by N. Bits shifted out of position 0 are lost. Positions vacated on the left are filled with zeros. The Q register is not changed. Parity is accumulated. No overflow is possible. The next instruction is taken in sequence. Timing: 1 to 5 cycles.

LGL Logical Left Shift

0	01	1000	▨	0	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the A register are shifted left the number of positions indicated by N. Bits shifted out of position 23 are lost. Positions vacated on the right are filled with zeros. The Q register is not changed. Parity is accumulated. No overflow is possible. The next instruction is taken in sequence. Timing: 1 to 5 cycles.

LGR Q Logical Right Shift Q

0	01	0100	▨	1	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the Q register are shifted right the number of positions indicated by N. Bits shifted out of position 0 are lost. Positions vacated on the left are filled with zeros. The A register is not changed. Parity is accumulated. No overflow is possible. The next instruction is taken in sequence. Timing: 1 to 5 cycles.

LGL Q Logical Left Shift Q

0	01	0100	▨	0	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the Q register are shifted left the number of positions indicated by N. Bits shifted out of position 23 are lost. Positions vacated on the right are filled with zeros. The A register is not changed. Parity is accumulated. No overflow is possible. The next instruction is taken in sequence. Timing: 1 to 5 cycles.

LGR AQ

Logical Right Shift
A and Q

0	01	1100	▨	1	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the A and Q registers are shifted right the number of positions indicated by N. Bits shifted out of position 0 of the A register enter position 23 of the Q register. Bits shifted out of position 0 of the Q register are lost. Positions vacated on the left are filled with zeros. Parity is accumulated. No overflow is possible. The next instruction is taken in sequence. Timing: 1 to 9 cycles.

LGL AQ

Logical Left Shift
A and Q

0	01	1100	▨	0	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

The contents of the A and Q registers are shifted left the number of positions indicated by N. Bits shifted out of position 23 of the Q register enter position 0 of the A register. Bits shifted out of position 23 of the A register are lost. Positions vacated on the right are filled with zeros. Parity is accumulated. No overflow is possible. The next instruction is taken in sequence. Timing: 1 to 9 cycles.

LGR C

Logical Right Shift
Closed

0	01	1010	▨	1	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

Same as LGR except bits shifted out of position 0 of the A register enter position 23 of the A register and parity is not accumulated.

LGL C

Logical Left Shift
Closed

0	01	1010	▨	0	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

Same as LGL except bits shifted out of position 23 of the A register enter position 0 of the A register and parity is not accumulated.

LGR QC

Logical Right Shift
Q Closed

0	01	0110	▨	1	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

Same as LGR Q except bits shifted out of position 0 of the Q register enter position 23 of the Q register and parity is not accumulated.

LGL QC

Logical Left Shift
Q Closed

0	01	0110	▨	0	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

Same as LGL Q except bits shifted out of position 23 of the Q register enter position 0 of the register and parity is not accumulated.

LGR AQC

Logical Right Shift
A and Q closed

0	01	1110	▨	1	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

Same as LGR AQ except bits shifted out of position 0 of the Q register enter position 23 of the A register and parity is not accumulated.

LGL AQC

Logical Left Shift
A and Q Closed

0	01	1110	▨	0	▨	N
23	22 18	17 14	13 12	11	10 6	5 0

Same as LGL AQ except bits shifted out of position 23 of the A register enter position 0 of the Q register and parity is not accumulated.

SCA

Scale A Register

0	01	00	00	Y
23	22 18	17 16	15 14	13 0

The contents of the A register are shifted left until the most significant bit is in position 22. Positions vacated on the right are filled with zeros. The sign (position 23) is not shifted. One is added to the shift counter for each shift, and the final contents of the shift counter are stored in location Y as a right-half integer. Scaling the number zero results in a shift count of 63 being stored. Scaling a number which is already in normalized form results in a shift count of zero being stored. To restore a scaled number, it is entered into the A register and a SHR instruction is executed with the scale count in its N field. The Q register is not changed. No overflow is possible. Parity is not accumulated. The next instruction is taken in sequence. Timing: 1 to 7 cycles (except scaling zero: 13 cycles).

SCAQ

Scale A and Q
Registers

0	01	00	01	Y
23	22 18	17 16	15 14	13 0

Same as SCA except A and Q are both shifted left with bits shifted out of position 22 of the Q register entering position

0 of the A register. Position 23 of the Q register is not shifted except if bits 0 to 22 of the Q register would result in all zeros in which case bit 23 of the Q register is forced to zero. The inverse operation is SHR AQ. Timing: 1 to 11 cycles (except scaling zero: 13 cycles).

RBAQ

Reverse Bits in A and in Q

0	01	00	10	▨
23	22 18	17 16	15 14	13 0

The bits in the A register and Q register are reversed such that the contents of position 0 and position 23 are interchanged, the contents of position 1 and position 22 are interchanged, etc. No overflow is possible. Parity is not accumulated. The next instruction is taken in sequence. Timing: 6 cycles.

TLY
(Optional)

Tally 1's in the A Register

0	01	00	11	Y
23	22 18	17 16	15 14	13 0

The number of 1's in the A register are counted and the result in stored location Y as a right half integer. The contents of the A and Q registers are unchanged by this instruction. The status of the overflow other indicators is not changed. The next instruction is taken in sequence. Timing: 11 cycles.

Increment or Decrement Instructions (JDB, SUB1, ADD1)

JDB

The basic format for the jump and decrement index register instruction is

0	03	L	B	Y
23	22 18	17 16	15 14	13 0

An integer 1 is subtracted from the contents of the designated index register. The result replaces the contents of that index register. The result also is used as an operand for a conditional jump according to the L designator as described below. Y is neither indexable nor indirectly addressable. The contents of register, indicators, and other memory locations remain unchanged. Timing: 3 cycles.

JDB P

0	03	0	B	Y
23	22 18	17 16	15 14	13 0

If the resulting operand has a sign which is plus, the next instruction is taken from location Y, otherwise the next instruction is taken in sequence.

JDB M

0	03	1	B	Y
23	22 18	17 16	15 14	13 0

If the resulting operand has a sign which is minus, the next instruction is taken from location Y; otherwise the next instruction is taken in sequence.

JDB Z

0	03	2	B	Y
23	22 18	17 16	15 14	13 0

If the resulting operand is zero, the next instruction is taken from location Y; otherwise the next instruction is taken in sequence.

JDB N

0	03	3	B	Y
23	22 18	17 16	15 14	13 0

If the resulting operand is non-zero, the next instruction is taken from location Y; otherwise the next instruction is taken in sequence.

SUB1 Subtract 1 from Storage and Skip

0	06	L	B	Y
23	22 18	17 16	15 14	13 0

An integer 1 is subtracted from the contents of location Y as a full word. The result is used for an operand and placed back in the same location. Indexing applies. Indirect addressing does not apply. Overflow is possible. The contents of registers and other memory locations are not affected. Timing: 3 cycles.

SUB1 P

0	06	0	B	Y
23	22 18	17 16	15 14	13 0

If the resulting operand has a sign which is plus, the next instruction is skipped and execution proceeds from the following instruction; otherwise, the next instruction is taken in sequence.

SUB1 M

0	06	1	B	Y
23	22 18	17 16	15 14	13 0

If the resulting operand has a sign which is minus, the next instruction is skipped and execution proceeds from the following instruction; otherwise, the next instruction is taken in sequence.

SUB1 Z

0	06	2	B	Y
23	22 18	17 16	15 14	13 0

If the resulting operand is zero, the next instruction is skipped and execution proceeds from the following instruction; otherwise the next instruction is taken in sequence.

SUB1 N

0	06	3	B	Y
23	22 18	17 16	15 14	13 0

If the resulting operand is non-zero, the next instruction is skipped and execution proceeds from the following instruction; otherwise the next instruction is taken in sequence.

ADD1 Add 1 to Storage

1	03	K	B	Y
23	22 18	17 16	15 14	13 0

An integer 1 is added to the contents of location Y. The result is placed back in the same location. Indexing applies. Indirect addressing does not apply. Overflow is possible. The contents of registers and other memory locations are not affected. Timing: 3 cycles.

The options RH, LH, and Y apply as described under STA RH, STA LH, and STA Y with the same portion fetched and incremented as is stored back.

Skip Instructions (CMP, CYS, SSK, SSH, SCP, SSW, IDA, ODA, SOF, SDE, SOP, SAM, SQM, SAZ, SMO, SMS, HEY)

CMP Compare

I	16	K	B	Y
23	22 18	17 16	15 14	13 0

The contents of location Y are compared with the contents of the A register. If the contents of the A register are algebraically smaller, the next instruction in sequence is skipped and the following instruction is executed; otherwise the next instruction in sequence is executed. Indexing and indirect addressing apply in all combinations. The contents of location Y and the A register remain unchanged. Timing: 2 cycles.

The options RH, LH, and Y apply as described under ADD RH, ADD LH, and ADD Y, respectively.

CYS Compare with Y and Skip

I	17	L	B	Y
23	22 18	17 16	15 14	13 0

The contents of location Y are subtracted from the contents of the A register to form an operand. If the skip condition specified by the L designator is met, the next instruction in sequence is skipped and the following instruction is executed; otherwise the next instruction in sequence is executed. Indexing and indirect addressing apply in all combinations. The contents of location Y and the A register remain unchanged. Timing: 3 cycles.

The options are as follows:

CYS P

I	17	0	B	Y
23	22 18	17 16	15 14	13 0

Skip if the sign of the operand is plus.

CYS M

I	17	1	B	Y
23	22 18	17 16	15 14	13 0

Skip if the sign of the operand is <u>minus.</u>

CYS Z

I	17	2	B	Y
23	22 18	17 16	15 14	13 0

Skip if the operand is <u>zero.</u>

CYS N

I	17	3	B	Y
23	22 18	17 16	15 14	13 0

Skip if the operand is <u>non-zero.</u>

SSK Storage Skip

0	04	L	B	Y
23	22 18	17 16	15 14	13 0

The contents of location Y are examined as an operand. If the condition specified by the L designator is met, the next instruction in sequence is skipped and the following instruction is executed; otherwise the next instruction in sequence is executed. Indexing applies but indirect addressing is not available. The contents of location Y are unchanged. Timing: 2 cycles.

The P, M, Z, and N options apply as described under CYS P, CYS M, CYS Z, and CYS N, respectively.

SSH Storage Shift Skip

0	05	L	B	Y
23	22 18	17 16	15 14	13 0

Same as SSK except the contents of location Y are shifted logically left closed one position then replaced. The test is made on the resulting shifted operand.

SCP Selective Compare

I	15	L	B	Y
23	22 18	17 16	15 14	13 0

The Q register is assumed to contain a mask. The logical AND of the contents of the Q register and the contents of location Y are formed. The result is subtracted from the contents of the A register to form an operand. Use of index register and indirect addressing apply in all combinations. From this point the execution is the same as for CYS. Timing: 2 cycles.

SSW Sense Switch

0	07	0	SWITCHES
23	22 18	17 16	15 0

The settings of the 16 sense switches on the programmers panel are sensed. The bit-by-bit AND of the sense switches and the SWITCHES bits are formed, and the result is combined by a logical OR to form the skip condition. If the skip condition is 1, the next instruction in sequence is skipped, and execution proceeds with the following instruction; otherwise the next instruction is executed in sequence. Neither indexing nor indirect addressing apply.

A switch in the up position is "ON" and has value 1. A switch in the down position is "OFF" and has value 0. The skip occurs when any of the switches are "ON" that are specified by 1's in the SWITCH field. No registers, indicators, or memory locations are changed by this instruction.

In the assembly language, when designating one switch the option field contains the decimal designation, 0 to 15. If more than one switch is to be tested, an octal qualifier followed by octal digits is placed in the address field. Timing: 1 cycle.

IDA Input Data Available

0	07	1	CHANNELS
23	22 18	17 16	15 0

The next instruction is skipped and execution proceeds from the following instruction if there is input data available on any channel where there is a 1 in the CHANNELS field; otherwise the next instruction is taken in sequence. No registers, indicators, or memory locations are changed by this instruction. Timing: 1 cycle.

Note: The results of this instruction are unreliable if a buffer input mode is established on any of the channels being tested. In the assembly language, when designating one channel, the option field contains the decimal channel number to be tested. When designating more than one channel, an octal qualifier followed by octal digits are placed in the address field.

ODA Output **Data Lines Available**

0	07	2	CHANNELS
23	22 18	17 16	15 0

The next instruction is skipped and execution proceeds from the following instruction if the output lines are available on any channel where there is a 1 in the CHANNELS field.

See description of IDA instruction for further details.

Indexing and indirect addressing do not apply to the following instructions. In the assembly language, an entry in the address field is flagged as an error. Timing: 1 cycle.

SOF

Skip on Overflow

0	07	3	0	01
23	22 18	17 16	15	0

Skip the next instruction if the overflow indicator is "ON." The execution of this instruction turns the indicator "OFF."

SDE

Skip on Division Error

0	07	3	0	010
23	22 18	17 16	15	10

Skip the next instruction if the division error indicator is "ON." The execution of this instruction turns the indicator "OFF."

SOP

Skip on Odd Parity Developed by Shifting

0	07	3	0	0100
23	22 18	17 16	15	2 0

Skip the next instruction if the parity developed by open logical shifting is odd. The execution of this instruction sets the indicator to even.

SAM

Skip if Sign of A is Minus

0	07	3	0 ——	01000
23	22 18	17 16	15	3 0

Skip the next instruction if the sign bit of the A register is minus. The contents of the A register are not changed.

SQM Skip if Sign of Q is Minus

0	07	3	0 ——— 010 — 0
23	22 18	17 16	15 4 0

Skip the next instruction if the sign bit of the Q register is minus. The contents of the Q register are unchanged.

SAZ Skip if Contents of A is Zero

0	07	3	0 ——— 010 ——— 0
23	22 18	17 16	15 5 0

Skip the next instruction if the contents of the A register is zero. The contents of the A register are unchanged.

SMO Skip if Memory Range Overflow

0	07	3	00 ——— 010 ——— 0
23	22 18	17 16	15 6 0

Skip the next instruction if the memory range overflow indicator is "on." The execution of this instruction turns the indicator "off."

SMS Skip if Master Lockout not Set

0	07	3	0 ——— 010 ——— 0
23	22 18	17 16	15 7 0

Skip the next instruction if the master lockout is not in effect. The execution of this instruction does not change the state of the master lockout.

HEY Turn on Program Flag Light

0	07	3	0 ——— 010 ——— 0
23	22 18	17 16	15 8 0

The Program Flag light on the operation panel is turned on to gain the operators attention. The light can only be turned off by depressing the light.

Jump Instructions (JMP, JSR)

JMP AP Jump on A Plus

I	35	0	B	Y
23	22 18	17 16	15 14	13 0

If the sign of the A register is plus, the next instruction is taken from location Y and execution proceeds from there; otherwise the next instruction is executed in sequence. Indexing and indirect addressing apply in all combinations. Timing: 1 cycle.

JMP AM Jump on A Minus

I	35	1	B	Y
23	22 18	17 16	15 14	13 0

If the sign of the A register is minus the next instruction is taken from location Y and execution proceeds from there; otherwise the next instruction is executed in sequence. Indexing and indirect addressing apply in all combinations. Timing: 1 cycle.

JMP AZ Jump on A Zero

I	35	2	B	Y
23	22 18	17 16	15 14	13 0

If the contents of the A register is zero the next instruction is taken from location Y and execution proceeds from there; otherwise the next instruction is executed in sequence. Indexing and indirect addressing apply in all combinations. Timing: 1 cycle.

JMP QP Jump on Q Positive

I	35	3	B	Y
23	22 18	17 16	15 14	13 0

If the sign of the Q register is plus, the next instruction is taken from location Y and execution proceeds from there. Otherwise the next instruction is executed in sequence. Timing: 1 cycle.

JMP S1 Jump on Switch 1 on

I	36	1	B	Y
23	22 18	17 16	15 14	13 0

If sense switch 1 is up or in the 'on' position the next instruction is taken from location Y and execution proceeds from there; otherwise the next instruction is executed in sequence. Indexing and indirect addressing apply in all combinations. Timing: 1 cycle.

JMP S2 Jump on Switch 2 on

I	36	2	B	Y
23	22 18	17 16	15 14	13 0

If sense switch 2 is up or in the "on" position the next instruction is taken from location Y and execution proceeds from there; otherwise the next instruction is executed in sequence. Indexing and indirect addressing apply in all combinations. Timing: 1 cycle.

JMP Jump

I	36	0	B	Y
23	22 18	17 16	15 14	13 0

Unconditional jump. The next instruction is taken from location Y and execution proceeds from there. Indexing and indirect addressing apply in all combinations. Timing: 1 cycle.

JMP CML

Jump and Clear
Master Lockout

I	36	3	B	Y
23	22 18	17 16	15 14	13 0

The master lockout is reset. See Input-Output Instructions for further details. The next instruction is taken from location Y and execution proceeds from there. The B field is ignored for this option. Indirect addressing applies. Timing: 1 cycle.

JSR

Jump and Set
Return

I	37	0	B	Y
23	22 18	17 16	15 14	13 0

The location of this instruction plus 1 is stored in bits 0 to 13 of location Y. The remainder of that location is not changed. The next instruction is taken from location Y plus 1 and execution proceeds from there. Use of index register and indirect addressing apply in all combinations. No registers, indicators, or other memory locations are changed. Timing: 2 cycles.

JSR AM

Jump and Set
Return if A
Minus

I	37	1	B	Y
23	22 18	17 16	15 14	13 0

If the sign of the A register is minus, execution proceeds the same as JSR; otherwise the next instruction is executed in sequence. Timing: 1 or 2 cycles.

JSR AZ Jump and Set Return if A Zero

I	37	2	B	Y
23	22 18	17 16	15 14	13 0

If the contents of the A register are zero, execution proceeds the same as JSR; otherwise the next instruction is executed in sequence. Timing: 1 or 2 cycles.

JSR SML Jump and Set Return Set Master Lockout

I	37	3	B	Y
23	22 18	17 16	15 14	13 0

The master lockout is turned on and execution proceeds the same as JSR. The B field is ignored for this option. Timing: 2 cycles.

Misc. Instructions (FAULT, JHT, NOP, INOP, STZ, NEG, XEQ)

FAULT Fault

0	00	▨
23	22 18	17 0

The sequence clock and real time clock are turned off. All computation and input-output processing is suspended and the FAULT indicator is lighted. The machine must be manually reset to reinstate operation. If sense switch 15 is up the machine does not stop and a FAULT interrupt occurs. See page 16 on interrupts.

JHT Jump and Halt

1	07	2	B	Y
23	22 18	17 16	15 14	13 0

The sequence clock and real time clock are turned off. All computation and input-output processing is suspended and the

RUN indicator is turned off. Computation resumes with a jump to location Y when the RUN button is depressed. Relative addressing via indexing is applicable. The jump address is displayed in the P register when the stop occurs.

NOP No Operation

1	07	3	B	Y
23	22 18	17 16	15 14	13 0

No registers, or memory locations are changed. The computer takes the instruction in sequence. The B and Y fields may be used but do not affect the execution of the instruction.

Timing: 1 cycle.

INOP Indirect NO Operation

I	34	0	B	Y
23	22 18	17 16	15 14	13 0

This instruction is intended for use in connection with indefinite indirect addressing operations. When executed as a direct instruction, no registers or memory locations are changed, however, a normal operand acquisition cycle is executed . If an index register is specified, an additional cycle time is lost.

Timing: 2 cycles.

STZ Store Zero

I	32	K	00	Y
23	22 18	17 16	15 14	13 0

A full word of zeros is stored at location Y. Indirect addressing is applicable but no indexing is allowed.

Timing: 2 cycles.

The options RH, LH, and Y apply as described under STA RH, STA LH, and STA Y, respectively, to specify the portion of a word to be zeroed.

NEG Negate and Jump

1	07	0	B	Y
23	22 18	17 16	15 14	13 0

The contents of the A register are negated (i.e., replaced by its 2's complement). Since 0 and the maximum negative number, -2^{23}, are their own 2's complement, the A register is unchanged for these. No overflow is possible. Indirect addressing is not applicable. Indexing is applicable. The next instruction is taken from location Y. Timing: 2 cycles.

XEQ Execute

I	34	3	B	Y
23	22 18	17 16	15 14	13 0

The instruction at location Y is executed. The program then returns to the next instruction in normal sequence. Indexing and indirect addressing apply.

Timing: 1 cycle + normal timing of instruction executed.

Input-Output Instructions (EXF, EXI, EXO, BCW, BIN, BBIN, BINI, BBINI, BOT, BBOT, BOTI, BBOTI, BOMT, BOMTI, BIMT, BIMTI, SOBS, SIBS, CIR, SAL, SLS, ELS).

Details of input-output channels are contained in the next section. The mnemonics and operation of input-output instructions are given below. The timing indicated is for the instruction execution and does not include input-output data transfer unless specifically stated. Timing requirements for input-output devices are covered in the next section.

EXF External Function

1	06	Channel	Y
23	22 18	17 14	13 0

The contents of location Y are placed on the 24 output lines of the indicated channel in parallel. At the same time the external function control signal is sent to the indicated channel. Neither indirect addressing nor indexing is allowed. This instruction may be executed while a buffer is established. Timing: 3 cycles. In the assembly language the channel is designated by a decimal number, 0 to 15 in the option field.

EXI Express Input

1	00	channel	Y
23	22 18	17 14	13 0

The 24 input lines of the indicated channel are sensed in parallel and the resulting word is stored in location Y. Upon completion of the sensing, an input acknowledge signal is sent to the indicated channel. Timing: 3 cycles.

Note: Depending on the input-output device there may be restrictions on the use of this instruction. See Input-Output Instructions and EXF instruction for further description.

EXO Express Output

1	01	channel	Y
23	22 18	17 14	13 0

The contents of location Y are placed on the 24 output lines of the indicated channel in parallel. At the same time, the output acknowledge signal is sent to the indicated channel. Timing: 3 cycles.

Note: EXF, EXI, and EXO are 4 cycle instructions when channels are specified which transfer more than 24 bits. In this case the transferred word is broken into two equal sectors. The least significant bits are located in Y and the most significant bits are located in Y + 1.

BCW Buffer Control Word

Count	Y

23 13 0

This is not a machine instruction. The word generated is in the format required for the buffer control word locations during buffered input and output. The count (decimal integer) is placed in the options field for assembly language coding and specifies the number of words to be transmitted. The assembler places the 2's complement of the count in bits 23 to 14 thus the maximum count is 1024. Theaddress calls out the first location used for input-output. The successive locations involved in input-output proceed from the address in the Y field to the address in the Y field plus the count. The buffer will be terminated automatically after the count reaches zero. Bits 23 to 14 are zero and the Y field calls out the last location Plus 1 used for input-output upon buffer termination via interrupt. A word of this type must be physically placed after an instruction which establishes a buffer mode. When the buffer control word is used for channels transferring more than 24 bits, the input-output proceeds from the address in the Y field to the address in the Y field plus twice the count. In this case the transferred word is broken into two equal sectors. The least significant bits are located in Y and the most significant bits are located in Y + 1.

Note: Depending on the input-output device there may be restrictions on the use of this instruction. See Operating System Reference Manual for further details.

BIN

Establish buffer input mode

1	05	Channel	1	0	1	0	0	▨
23	22 18	17 14	13	12	11	10	9	8 0

The contents of the next sequential location, the buffer control word (BCW), is placed in the buffer control word location as indicated in Assigned Core Memory Locations. The designated channel is then established in buffer input mode. No interrupt will occur when the buffer count has reached zero but further input request signals will be ignored. Neither indirect addressing nor indexing is applicable. No registers, indicators, or memory locations are changed, and no information is transmitted to or received from the indicated channel. The next location in sequence is not used as an instruction, and execution proceeds with the instruction in the following location. In the assembly language the channel, 0 to 15, is specified in the option field. Timing: 3 cycles.

BBIN

Establish Block Buffer Input Mode

1	05	Channel	1	0	1	0	1	▨
23	22 18	17 14	13	12	11	10	9	8 0

Same as BIN instruction, except when request is received, the entire buffer count is buffered in as a block of data before performing the next instruction.

BINI

Establish Buffer Input Mode, Interrupt

1	05	Channel	1	1	1	0	0	▨
23	22 18	17 14	13	12	11	10	9	8 0

Same as BIN except an interrupt occurs forcing the instruction in the Buffer Finished Interrupt location of the designated channel to be executed when the buffer control word count has reached zero.

BBINI

Establish Block Buffer Input Mode With Interrupt

1	05	Channel	1	1	1	0	1	▨
23	22 18	17 14	13	12	11	10	9	8 0

Same as BBIN except on completion of the buffer an interrupt occurs forcing the instruction in the buffer finished interrupt location of the designated channel to occur.

BOT

Establish Buffer Output Mode

1	05	Channel	0	0	1	0	0	▨
23	22 18	17 14	13	12	11	10	9	8 0

The contents of the next sequential location, the buffer control word (BCW), is placed in the buffer control word location as indicated in Assigned Core Memory Locations. The designated channel is then established in buffer output mode. No interrupt will occur when the buffer count has reached zero, but further output request signals will be ignored. Neither indirect addressing nor indexing is applicable. No registers, indicators, or memory locations are changed, and no information is transmitted or received from the indicated channel. The next location in sequence is not used as an instruction and execution proceeds with the instruction in the following location. In the assembly language the channel, 0 to 15, is specified in the option field. Timing: 3 cycles.

BBOT

Establish Block Buffer Output Mode

1	05	Channel	0	0	1	0	1	▨
23	22 18	17 14	13	12	11	10	9	8 0

Same as BOT, except the buffer is established as a single block of data and when request is received entire block of data determined by buffer count in BCW is outputted before the next instruction is executed.

BOTI

Establish Buffer ~~Output Mode~~ Interrupt

1	05	Channel	0	1	1	0	0	▨
23	22 18	17 14	13	12	11	10	9	8 0

Same as BOT, except an interrupt occurs forcing the instruction in the buffer finished interrupt location to be executed.

BBOTI

Establish Block Buffer Output Mode With Interrupt

1	05	Channel	0	1	1	0	1	▨
23	22 18	17 14	13	12	11	10	9	8 0

Same as BBOT, except on completion of the buffer an interrupt occurs forcing the instruction in the buffer finished interrupt location of the designated channel to occur.

BOMT Output Buffer Mode Termination

1	05	Channel	0	0	0	0		▨
23	22 18	17 14	13	12	11	10	9	8 0

The output buffer on the designated channel is terminated without an interrupt occurring. The next instruction is skipped and execution proceeds from the following location. The buffer control word in the fixed location is not changed. Timing: 2 cycles.

BOMTI Output Buffer Mode Terminate and Interrupt

1	05	Channel	0	1	0	0		▨
23	22 18	17 14	13	12	11	10	9	8 0

The output buffer mode on the designated channel is terminated and an interrupt results. A return after the interrupt will skip the next instruction. The buffer control word in the fixed location is not changed. Timing: 2 cycles.

BIMT Input Buffer Mode Termination

1	05	Channel	1	0	0	0		▨
23	22 18	17 14	13	12	11	10	9	8 0

The input buffer on the designated channel is terminated without an interrupt occurring. The next instruction is skipped and execution proceeds from the following location. The buffer control word in the fixed location is not changed. Timing: 2 cycles.

BIMTI Input Buffer Mode Terminate and Interrupt

1	05	Channel	1	1	0	0		▨
23	22 18	17 14	13	12	11	10	9	8 0

The input buffer on the designated channel is terminated and an interrupt results. A return after the interrupt will skip the next instruction. The buffer control word in the fixed location is not changed. Timing: 2 cycles.

SOBS Store Output Buffer Status

1	05	Channel	0	0	0	1		▨
23	22 18	17 14	13	12	11	10	9	8 0

The status of eight output channels is stored in a fixed location as indicated in Figure 5. If any output channel 0 to 7 is specified the status of all is stored in location 00031. If any output channel 8 to 15 is specified the status of all is stored in location 00033. No registers, indicators, or other memory locations are changed. Timing: 2 cycles.

SIBS Store Input Buffer Status

1	05	Channel	1	0	0	1	0	▨
23	22 18	17 14	13	12	11	10	9	8 0

The status of eight input channels is stored in a fixed location as indicated in Figure 5. If any input channel 0 to 7 is specified the status of all is stored in fixed location 00030. If any input channel 8 to 15 is specified the status of all is stored in location 00032. See SOBS for further description.

CIR Clear Interrupt Request

1	04	0	B	Y
23	22 18	17 16	15 14	13 0

The interrupt request state is cleared on all channels indicated by a 1 in the contents of location Y. See Input-Output Instructions, for further description. Indexing is allowed but indirect addressing is not applicable. Timing: 2 cycles.

SAL Set Additional Lockouts

1	04	1	B	Y
23	22 18	17 16	15 14	13 0

For each bit in location Y that is a 1 set the lockout state. Previously locked out channels remain locked out. See CIR instruction for further description.

SLS Store Lockout Status

1	04	3	B	Y
23	22 18	17 16	15 14	13 0

The state of the lockouts on all channels is stored in location Y. Bit positions of channels in lockout status are

	CH 8			CH 9					CH 15	
Fixed Locations 00032 or 00033	BB	IR	BA	BB	IR	BA	----- ----- ----- -----	BB	IR	BA
	23	22	21	20	19	18		2	1	0

Bit 23 - indicates block mode selected on CH 8.
Bit 22 - indicates interrupt is selected on CH 8.
Bit 21 - indicates buffer is established on CH 8.
Bit 20 - indicates block mode selected on CH 9.
Bit 19 - indicates interrupt is selected on CH 9.
Bit 18 - indicates buffer is established on CH 9.

This form continues in groups of three bits per channel storing the status of channels 8 through 15.

	CH 0			CH 1					CH 7	
Fixed Locations 00030 or 00031	BB	IR	BA	BB	IR	BA	----- ----- ----- -----	BB	IR	BA
	23	22	21	20	19	18		2	1	0

Bit 23 - indicates block mode is selected on CH 0.
Bit 22 - indicates interrupt is selected on CH 0.
Bit 21 - indicates buffer is established on CH 0.
Bit 20 - indicates block mode is selected on CH 1.
Bit 19 - indicates interrupt is selected on CH 1.
Bit 18 - indicates buffer is established on CH 1.

This form continues in groups of three bits per channel storing the status of Channels 0 through 7.

Figure 4. Buffer Status Word

set to 1 and all other bits are set to 0. See CIR instruction for further description.

ELS Enter Lockout Status

1	04	2	B	Y
23	22 18	17 16	15 14	13 0

The channels corresponding to the bits of location Y are set to lockout status for 1's and reset for 0's. See CIR instruction for further description.

INPUT/OUTPUT CHARACTERISTICS

PHYSICAL INTERFACE

A basic machine has eight input and eight output channels with the capability of expansion to 16 input and 16 output channels.

Each channel has two cable connectors associated with it. The input connector consists of 24 incoming data lines, an input request line, an input acknowledge line, and an interrupt line. The output connector consists of 24 outgoing data lines, an output request line, an output acknowledge line, an external function line, and an interrupt line.

CHANNEL STATES

Each channel has a set of states which operate as described below:

The input request state is set by a continuous signal on the input request line supplied by a peripheral device. This state can be tested by a conditional skip instruction, input data available, IDA. If an input buffer is established the input request is serviced and acknowledged in accordance to its priority and the request line must be dropped for a minimum of 1 microsecond in order to establish a new request state.

The output request state is set by a continuous signal on the output request line supplied by a peripheral device. This state can be tested by a conditional skip instruction, output data available, ODA. If an output buffer is established the servicing is as described for input request.

The input external interrupt state is set by a continuous signal on the input interrupt line supplied by the peripheral device. When the interrupt is serviced the 24-bit status word is stored in an address

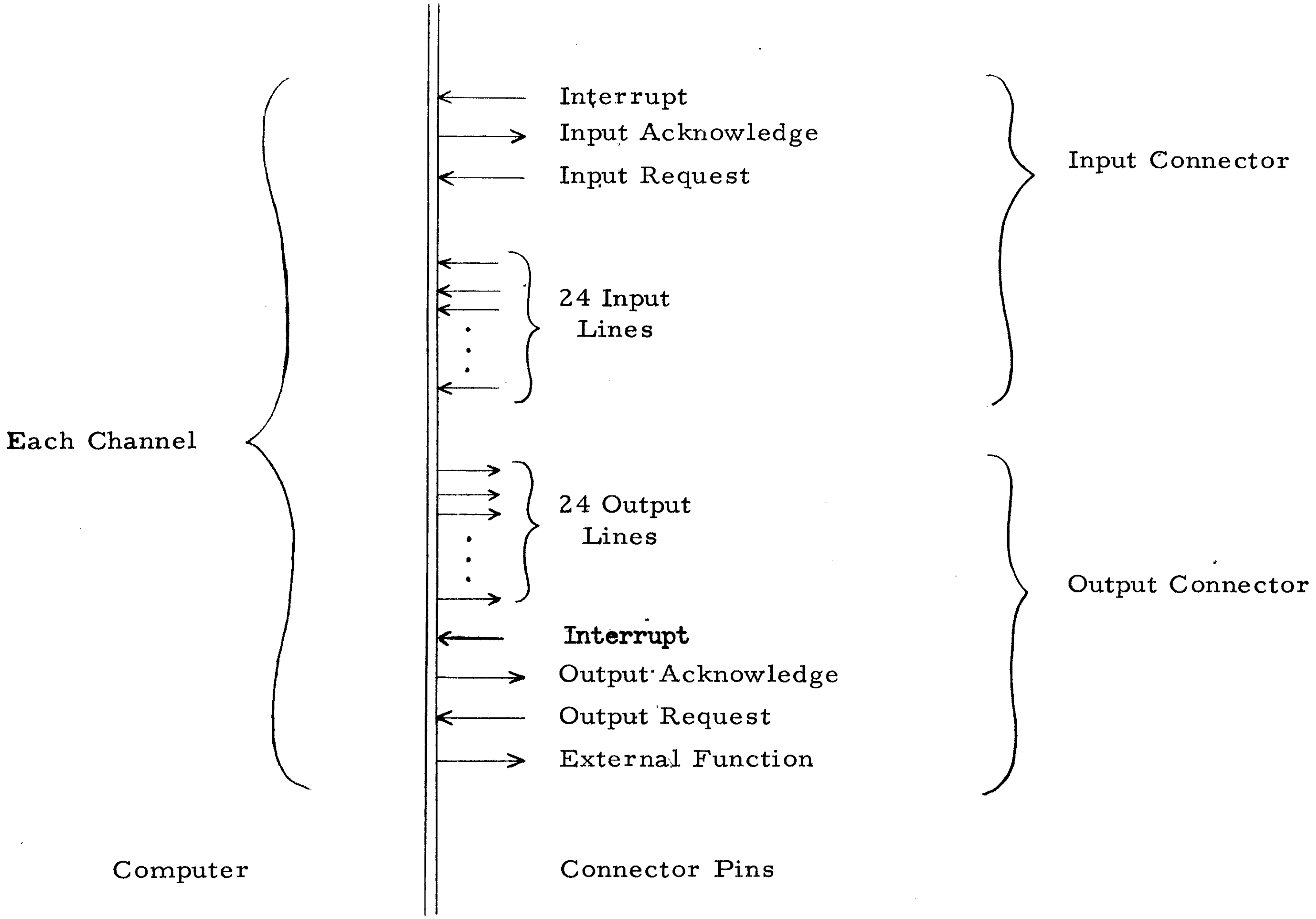


Figure 5. Channel Interface

specified by the contents of fixed locations 00140 to 00157. After the status word is stored the computer executes an instruction at location 00160 to 00177 according to channel and sends an input acknowledge. The interrupt state cannot be set again until the request line has dropped for a minimum of 1 microsecond.

The output external interrupt state is set by a continuous signal on the output interrupt line supplied by the peripheral device. When the interrupt is serviced the program is forced to execute an instruction at location 00200 to 00217 according to channel and sends an output acknowledge. The interrupt state cannot be set again until the request line has dropped for a minimum of 1 microsecond.

If the external interrupts on any given channel have been locked out, and it is desirable to clear the interrupt states without servicing them, a CIR instruction may be executed.

A channel is set to the input buffer enabled state by execution of any establish buffer input mode instructions, BIN, BINI, BBIN, or BBINI. This state is cleared when the buffer control word count is exhausted, and an input internal interrupt state is set if previously specified by BINI or BBINI. This internal interrupt forces the program to execute an instruction at location 00100 to 00117 according to channel. The input buffer active state can also be cleared with or without interrupt by execution of the proper terminate buffer input mode instruction BIMT or BIMTI.

The output buffer enabled state is identical to the input buffer state except the interrupt locations are from 0120 to 137, and the instructions involved are BOT, BOTI, BOMT, BOMTI, BBOT, and BBOTI.

Both the input and output buffer enabled states may exist on any channel simultaneously.

An interrupt lockout state is established, modified or stored by execution of the instructions: enter lockout status, ELS, set additional lockouts, SAL, and store lockout status, SLS.

ACKNOWLEDGE SIGNALS

An input acknowledge signal is generated when an input external interrupt is processed, when input data is stored by an express input instruction, and when input data is stored by a channel in input buffer mode.

An output acknowledge signal is generated when an output interrupt is processed, when output data is put on the output lines by an express output instruction, and when output data is put on the line by a channel in output buffer mode.

EXTERNAL FUNCTION SIGNALS

An external function, EXF, instruction is provided which places a signal on the external function line of a channel and places a 24-bit word from memory on the channel output lines. Interpretation of this word varies with the type of peripheral device. For a channel with many peripheral devices, part of the word would be interpreted by a multiplexer, appropriate routing paths established, and the remainder of the word used to establish the desired state of a particular device.

EXPRESS MODE

Upon execution of an express output instruction, EXO, a word from memory is set on the designated channel's output data lines. The output acknowledge line is signalled concurrent with data for 2 microseconds. The peripheral device must be ready to accept the output in that 2 microsecond period. If there is a possibility the channel is not ready to accept the output, the output request state may be tested by an output data available, ODA, instruction.

Upon execution of an express input instruction, EXI, a word from the designated channel's input data lines is stored in memory. The input acknowledge line is signaled for 2 microseconds immediately after data transfer has been completed. The channel is assumed ready to supply the input data. If there is a possibility the channel is not ready, the input request state may be tested by an input data available, IDA, instruction.

BUFFER MODE

When a buffer mode is established, a buffer control word, BCW, containing a base address and 2's complement of a count is loaded into a fixed location 00040 to 00077 according to channel. See page 67 for BCW word format, and page 18 for fixed location assignments.

Once an input buffer is established, the occurrence of an input request from a peripheral device causes the following sequence of events: The input request state is set. When processing of the current instruction and higher priority input-output functions are completed a memory access is made to fetch the buffer control word. The input data word from the channel is stored in memory at the address designated by the buffer control word and one is added to the count and one is added to the address of the buffer word. This new buffer control word is stored in its original location and the input acknowledge signal is sent. The input data must be held on the input lines until the acknowledge is received, or the information will not be transferred accurately.

Once an output buffer is established, the occurrence of an output request from a peripheral device causes the following sequence of events: The output request state is set. When processing of the current instruction and higher priority input-output functions are completed, a memory access is made to fetch the buffer control word. A word called out by the buffer control word address field is fetched from memory and placed on the output data lines of the channel. One is added to the count and one is added to the address of the buffer control word. The result is stored back in the same buffer control word location. The output acknowledge signal is sent concurrently with the output for 2 microseconds.

Upon incrementing either the input or output the buffer control word count from all 1's to all 0's and if the buffer was established to interrupt upon completion, the internal interrupt state is set and an instruction from the prescribed fixed location is executed.

The time sequences for signals and data flow are shown below in figure 7 for the case where a sequence of ADD instructions are being executed by a program and only one channel is being used.

BLOCK BUFFER

When a BLOCK BUFFER INPUT MODE is established, the occurrance of an input request from a peripheral device causes the following sequence of events:

The input request state is set, when processing of the current instruction and higher priority input-output functions are completed a memory access is made to fetch the buffer control word. The input data word from the channel is stored in memory at the address designated by the buffer control word. The buffer control word count and address are both incremented by one. The acknowledge signal is sent. The next input data word from the channel is stored in memory at the address called out by the buffer control word. The buffer control word count and address are both incremented by one. The acknowledge signal is sent. This cycle continues to repeat itself until the buffer control word count is cycled from all "1" to all "0". If buffer with interrupt was established the next instruction is taken from the prescribed fixed location. If no interrupt mode was established the next instruction in sequence is executed.

When a Block Buffer Output Mode is established the sequence of events are the same as input mode except data is transferred from the memory to the channel output data lines. This data is on the lines in concurrence with the acknowledge signal.

HARDWARE INTERRUPT AND BUFFER PRIORITIES

Interrupt and request lines are sampled during instruction cycles and at the completion of the instruction are serviced. Since it is possible for two or more interrupts or request signals to arrive during the execution of a single instruction, hardware priorities are necessary to allow only one interrupt or request to be serviced at a time. All interrupts released from master lockout or channel lockouts are considered to have arrived simultaneously and are also subject to these priorities. The table below specifies the order of the hardware priorities. It should be noted that instructions are available to lock out an interrupt signal so that the order of processing interrupt subroutines need not be the same order as for hardware priorities. NOTE: Following an interrupt or request being processed or serviced, one normal instruction will be executed to obtain the next sampling. This feature prevents a run-away peripheral device from locking out an I/O control program. Also, no processing or servicing takes place immediately after a JMP CML or JSR SML instruction.

	FETCH ADD INST	FETCH OPERAND	FETCH BCW	STORE INPUT WORD	STORE BCW AND ACKNOWLEDGE	FETCH ADD INST	FETCH OPERAND
Input Request Signal Arrives				Increment Count and Address	Reset Request Signal	Ready for next Input Signal Request.	

NORMAL INPUT BUFFER

FETCH ADD INST	FETCH OPERAND	FETCH BCW	STORE INPUT WORD	ACKNOWLEDGEMENT	STORE INPUT WORD	ACKNOWLEDGEMENT	FETCH ADD INST
Input Request Signal Arrives			Increment Count and Address		Increment Count and Address		2 microseconds per block.

BLOCK INPUT BUFFER

NOTE 1: Store, acknowledge cycle repeat until count cycles from all "1" to all "0". If interrupt on completing buffer is selected, interrupt location is fetched instead of ADD instruction.

NOTE 2: Output Buffers have same timing as input buffers except store input word cycle is fetch data cycle with data being held on output data lines during entire acknowledge time.

Figure ~~6~~ 7. Time Sequences for Signals and Data

TABLE OF PRIORITIES FOR INTERRUPTS
AND INPUT-OUTPUT FUNCTIONS

Priority Order	DESCRIPTION
1	Program Parity Error Interrupt
2	Input-Output Parity Error Interrupt
3	Fault Interrupt
4	Power Failure Interrupt
5	Real Time Clock Accumulation
6	Real Time Clock = $2^{24} - 1$ Interrupt
7	Manual Interrupt
8	Channel 0 - Input External Interrupt
	- Output External Interrupt
	- Input Buffer
	- Output Buffer
	- Input Internal Interrupt (Buffer Complete)
	- Output Internal Interrupt (Buffer Complete)
9	Channel 1 - Input External Interrupt
	- Output External Interrupt
	- Input Buffer
	- Output Buffer
	- Input Internal Interrupt (Buffer Complete)
	- Output Internal Interrupt (Buffer Complete)

Priority Order	DESCRIPTION
10	Channel 2 - Input External Interrupt
	- Output External Interrupt
	- Input Buffer
	- Output Buffer
	- Input Internal Interrupt (Buffer Complete)
	- Output Internal Interrupt (Buffer Complete)
11	Channel 3 - Input External Interrupt
	- Output External Interrupt
	- Input Buffer
	- Output Buffer
	- Input Internal Interrupt (Buffer Complete)
	- Output Internal Interrupt (Buffer Complete)
12	Channel 4 - Input External Interrupt
	- Output External Interrupt
	- Input Buffer
	- Output Buffer
	- Input Internal Interrupt (Buffer Complete)
	- Output Internal Interrupt (Buffer Complete)
13	Channel 5 - Input External Interrupt
	- Output External Interrupt
	- Input Buffer
	- Output Buffer
	- Input Internal Interrupt (Buffer Complete)
	- Output Internal Interrupt (Buffer Complete)

Priority Order		DESCRIPTION
14	Channel 6	- Input External Interrupt
		- Output External Interrupt
		- Input Buffer
		- Output Buffer
		- Input Internal Interrupt (Buffer Complete)
		- Output Internal Interrupt (Buffer Complete)
15	Channel 7	- Input External Interrupt
		- Output External Interrupt
		- Input Buffer
		- Output Buffer
		- Input Internal Interrupt (Buffer Complete)
		- Output Internal Interrupt (Buffer Complete)

INTERRUPT LOCKOUTS

If an interrupt lockout is established and an interrupt signal arrives, the interrupt signal is ignored. When the interrupt lockout is removed the interrupt will be processed according to the priority specified in Hardware Interrupt priorities. Unless the interrupt has been previously cleared by CIR instruction or the request for interrupt has been removed.

1. Channel Interrupt Lockouts

Each channel has an interrupt lockout which, when set, prevents processing of input and output buffer finished interrupts and external interrupts. The lockout status on all channels can be set by the enter lockout status instruction, ELS. If some channels may already be locked out, a set additional lockouts, SAL, instruction will place more channels in lockout status without affecting those already locked out. The status of the lockouts on all channels may be determined by storing the lockout status, SLS instruction, and examining the lockout bits.

If an interrupt has been locked out and no processing of the interrupt is desired, the clear interrupt request, CIR, instruction can be used.

2. Master Interrupt Lockout

When the master interrupt lockout is activated, all channel external and internal interrupts are prevented while all other interrupts are allowed. The real time clock will still be incremented and buffer input and output will continue.

The master lockout is activated by execution of the jump and set return, set master lockout instruction, JSR SML. The master lockout is cleared by the jump and clear master lockout instruction, JMP CML. When the jump and set return, set master lockout instruction is used at the beginning of a bookkeeping operation it is not possible for any channel interrupts to take place before the necessary bookkeeping is completed. Then, by ending the bookkeeping with a jump and clear master lockout, interrupts may be processed. During the bookkeeping, lockouts may be changed on any channels to allow any desired order of priority for processing interrupt subroutines.

GENERAL INPUT-OUTPUT TIMING CONSIDERATIONS

A peripheral equipment should normally be designed to accept an external function, i.e., control signal at any time, however, data transfers must be geared to the rate of the external device. Synchronization of data transfers is automatic when accomplished in the buffer mode; however, synchronization is a responsbility of the programmer when data transfers are made under program control. To assist the programmer in this respect, two instructions (IDA and ODA) are available.

Prior to executing input or output instructions or establishing a buffer mode, an external function code should be outputted to the peripheral device to set it in the proper state for data transfers.

External function code definition and other related information can be found in the reference manual describing the particular peripheral equipment involved.

When high bit-transfer rate input-output devices are being used, a number of hardware timing characteristics influence the coding of an input-output executive. The tables below give worst case figures on various characteristics.

The maximum input-output bit transfer rate is 6 x 10^6 bits per second. This may be obtained by the block buffer mode of operation with a series of express input instructions; one 24 bit data word is input per instruction executed. Fetching the EXI instruction requires one 2.0 microsecond memory cycle. Transfer of data requires one cycle, and acknowledge requires one cycle. Thus, every 6.0 microseconds, 24 bits of data can be input. Similarly, a sequence of express output, EXO, instructions can output a 24 bit word every 6.0 microseconds.

Operating under express input or output, the computer expects the peripheral equipment to be ready for the input or output and does not wait for request signals. This express function is useful where very high speed peripheral equipment is being used and where blocks of data are relatively small since one instruction per input-output word is needed. A slower form of express input-output uses the following sequence to bring in a block of N words into locations L, L + 1, ..., L + N-1.

```
ENB             = N, 1
EXI             L               4 microseconds
ADD 1           *-1             6 microseconds
JDB N           *-2, 1          6 microseconds
```

This, however, is not recommended since the bit transfer rate of 1.5 million bits per second is one half the maximum buffered input-output rate. Further, express input-output does not allow the computer to wait for peripheral device to request an input-output transfer.

Buffered input-output makes use of circuitry within the computer to increment the buffer address, to increment the count of the number of data words transferred, and to automatically terminate the buffer when the count is exhausted. The current buffer address and count are stored in core as a buffer control word. The maximum bit transfer rate of 3.0 million bits per second occurs under the following conditions:

Either instruction sequence is executed,

```
BINI                                      BOTI
BCW                                       BCW
JMP *                                     JMP *
```

timing for the sequence of operations is

JMP	Fetch BCW	Store/Fetch I/O Data & Increment BCW	Store BCW, Test Counted Acknowledge	JMP*	Fetch BCW
	2 µsec	2 µsec	2 µsec	2 µsec	

8 µsec per 24 bit word

A faster block buffer mode is also available. In the block buffer mode the Z Register holds the buffer control word (BCW), performs the incrementing, and the next instruction is not called up. Thus, 4 microseconds are eliminated from each I/O transfer in the block buffer mode of operation.

For two or more channels in normal buffer operation, the sum of the transfer rated can be as high as 3.0 megabits per second. If each channel gets a turn the timing would be shown below:

Channel 1 BCW and I/O	1 cycle instruction	Channel 2 BCW and I/O	1 cycle instruction	Channel 1 BCW and I/O	2 cycle instruction
6 μsec	2 μsec	6 μsec	2 νsec	6 μsec	4 μsec

16 μsec per 24 bit word on Channel 1

In general the bit transfer rates could be:

1 channel active 3.0 megabits per seconds

2 channels active 1.5 megabits per second per channel

3 channels active 1.0 megabits per second per channel

- - - -

8 channels active .375 megabits per second per channel

These figures assume each peripheral device requests input or output once during the time all channels are serviced.

The maximum delay (between waiting BCW and I/O phases) for various representative instructions is:

JMP	2 μsec
XAQ	2 μsec
SHR	2 μsec (i.e., I/O overlaps shifting)
ADD ( )	4 μsec
ADD ( )	6 μsec (with indirect addressing)
ADD -,1	6 μsec (with indexing)
ADD (-,1)	8 μsec (with indirect and indexing)
ADD (-,1), 2	10 μsec (with indirect and 2 indexes)
MUL	4 μsec (i.e., I/O overlaps multiply)
DIV	4 μsec (i.e., I/O overlaps divide)
SCA	2+2 μsec (i.e., I/O can take place in middle)
TLY	2+2 μsec (i.e., I/O can take place in middle)
RAD (-,1), 2	12 μsec (maximum)

Note: The real time clock automatically takes 6 usec every 5000 usec. Assuming the real time clock is off and no interrupts occur the minimum transfer time is 12 + 6 = 18 μsec per 24 bit word = 1.33 megabits per second total I/O transfer rate.

Allowing the real time clock there could be a maximum delay between BCW and I/O phases of 24 microseconds.

Allowing an interrupt with a JSR SML instruction being executed in interrupt mode gives the following maximum delay:

Channel 2 BCW & I/O	Instruction	Channel 1 interrupt JSR SML	Instruction	Channel 2 BCW & I/O
6 µsec	12 µsec	10 µsec	12 µsec	6 µsec

40 µsec (non-typical, near maximum)

This situation can only occur with two or more channels in operation simultaneously where worst case non-interrupt timing is:

Channel 2 BCW and I/O	Instruction	Channel 1 BCW and I/O	Instruction	Channel 2 BCW and I/O	Instruction
	12 µsec	6 µsec	12 µsec	6 µsec	

36 µsec (non-typical, near maximum)

The 2 channel worst case delay of buffered information with clock incrementation, interrupt and all maximum length instructions but no other internal interrupts is:

Channel 2 BCW and I/O	Instruction	Clock increment	Instruction
6 µsec	12 µsec	6µsec	12 µsec

Channel 1 interrupt JSR SML	Instruction	Channel 2 BCW and I/O	Instruction
10 µsec	12 µsec	6 µsec	

58 µsec (maximum)

Block Buffer Mode

As described in the preceding section the standard buffered channel transfers 24 bit words at a maximum rate of 125 KC, yielding a bit rate of 3.0 MC. This word transfer rate can be doubled to 250 KC by the Block Buffer Mode of operation. In this mode the peripheral equipment must be capable of sending or receiving data at this rate utilizing the acknowledge signal from the computer as information as to the time to put the next data word on the line.

The Block Buffer Mode is a programmer's option on any channel. This feature causes a buffer, once initiated with the first request, to continue until completed, locking out all other operations. The buffer complete interrupt capability is identical to that described for the standard Buffer.

PANELS

OPERATION PANEL

The 2402 Computer has three front panels. The operation panel at the top of the cabinet is always visible. The maintenance panel below the operation panel has a removable cover, as does the fuse panel near the bottom of the computer. A complete set of displays and controls are available on these panels to allow convenient operation and maintenance.

WARNING:

The pushbutton-lights, switches, and knobs on the panel are always active; therefore, any action taken at the panel applies immediately, even if the computer is running. It is suggested a good operating procedure is to always stop the machine before using any control except sense switches.

1. Operation Panel

A picture of the Operation Panel is shown in Figure 7.

FAULT

○

The FAULT LIGHT is turned on when the program executes an instruction with 6 or more high-order bits which are zero.

The FAULT light is turned off by depressing the light.

PARITY
ERROR

○

The PARITY ERROR light is turned on when a word read from core storage has an incorrect parity.

The PARITY ERROR light is turned off by depressing the light.

TEST
MODE

○

The TEST MODE light remains on as long as certain switches on the maintenance panel are in test, rather than normal, status.

PROGRAM
FLAG

○

The PROGRAM FLAG light is turned on by executing the Program Flag, HEY, instruction.

The PROGRAM FLAG light is turned off by depressing the light.

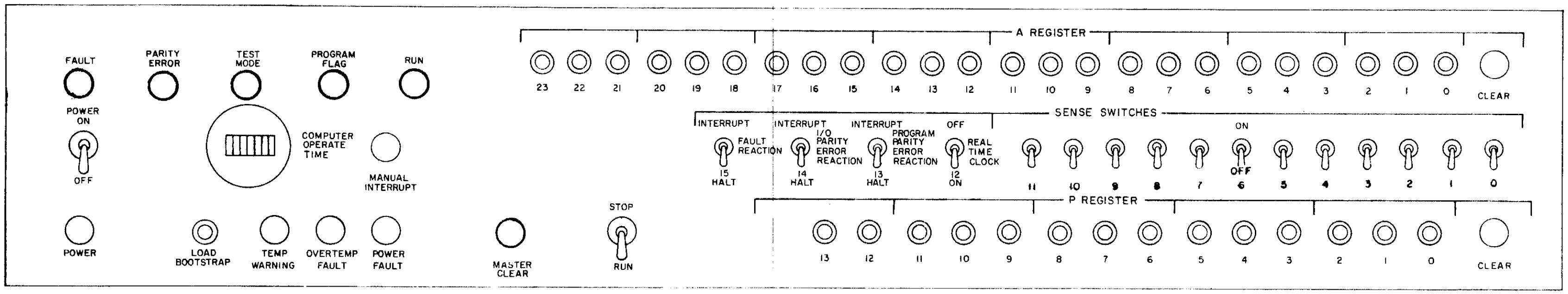
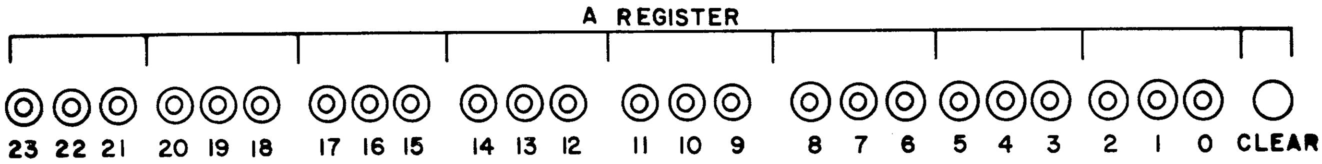


Figure 7: Operation Panel

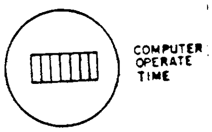
The RUN light is turned on when the RUN switch is depressed to the RUN position.



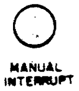
The A REGISTER is continuously displayed by 24 pushbutton lights. Depress-a pushbutton causes that bit of the A REGISTER to be set to a 1 and turns the light ON. The A REGISTER may be cleared at any time by depressing the CLEAR pushbutton to the right of the register display.

POWER
ON
OFF

The POWER switch is a two-position toggle switch. This switch controls the distribution of power to the cabinet. Up is the ON position. Down is the OFF position.

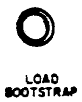


The COMPUTER OPERATE TIME meter reads the number of hours the d-c power has been on.



Depressing the MANUAL INTERRUPT pushbutton while the computer is running causes the instruction in location 00023 to be processed in interrupt mode. Depressing the MANUAL INTERRUPT pushbutton when the computer is not running may cause an interrupt to occur when the RUN switch is depressed.

The POWER light is on whenever the power switch is on.



the LOAD BOOTSTRAP switch is a spring-loaded pushbutton indicator switch. The MASTER CLEAR switch must be depressed first; then depressing this switch causes the bootstrap sequence to be given for channel 0. The POWER and COMPUTER switches must be in the ON position. Then, after depressing the LOAD BOOTSTRAP switch, the RUN switch must be depressed.

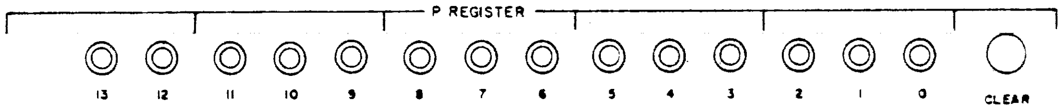
The TEMP WARNING light is ON whenever cabinet temperature is near the operating limits.

The OVERTEMP FAULT light indicates power has been shut off due to cabinet temperature exceeding operating limits.

The POWER FAULT light indicate power from the source has exceeded the allowable operating tolerances or a d-c power failure.

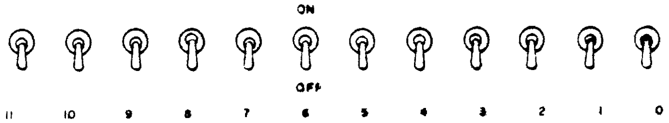
The MASTER CLEAR switch is a spring-loaded pushbutton switch When depressed, all registers are cleared and most of the flip-flops are set to the normal or inactive state. Core memory is not cleared or changed.

The RUN switch is a three-position spring-loaded to neutral toggle switch. When power is ON, depressing the RUN switch causes the computer to begin execution at the location indicated in the P-REGISTER. Raising the STOP switch causes the computer to stop at the completion of processing the current instruction.

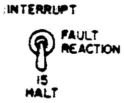


The P REGISTER is continuously displayed by 14 combination pushbutton lights. Depressing a pushbutton causes that bit of the P REGISTER to be set to a 1 and turns the light ON. The entire P REGISTER is cleared by depressing the CLEAR pushbutton to the right of the register display.

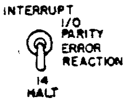
SENSE SWITCHES



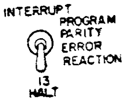
The SENSE SWITCHES are two-position toggle switches. A switch in the UP position is ON and will cause a skip when tested by a sense switch, SSW, instruction. A switch in the DOWN position is OFF and does not cause a skip when tested.



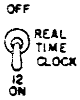
SENSE SWITCH No. 15 is wired as a FAULT REACTION switch. If the switch is down, in HALT position, the computer stops when a FAULT instruction is executed. If the switch is up, in INTERRUPT position, the computer does not stop, and the instruction at location 00020 is processed in interrupt mode.



THE SENSE SWITCH No. 14 is wired as an I/O PARITY ERROR REACTION switch. If the switch is down, in HALT position, the computer stops when a PARITY ERROR occurs while reading from core storage for input-output. If the switch is up, in INTERRUPT position, the computer does not stop, and the instruction at location 00021 is processed in interrupt mode.



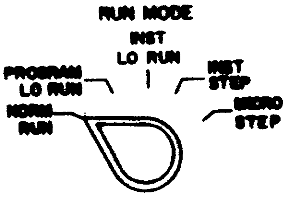
SENSE SWITCH No. 13 is wired as a PROGRAM PARITY ERROR REACTION switch. If the switch is down, in HALT position, the computer stops when a PARITY ERROR occurs while reading from core storage (except during input-output). If the switch is up, in INTERRUPT position, the computer does not stop, and the instruction at location 00022 is processed in interrupt mode.



SENSE SWITCH No. 12 is wired as a REAL TIME CLOCK switch. If the switch is ON (down), the core location 00024 is incremented by one each 5 milleseconds. If the switch is OFF (up), no incrementation takes place.

2. **Maintenance** Panel

Figure 8 shows the Westinghouse DPS-2402 Computer maintenance panel.



The RUN MODE selector switch allows the selection of five modes of execution. This allows maintenance personnel to perform most malfunction locations from the panel with the aid of diagnostic programs delivered with the aid of diagnostic programs delivered with the computer. The RUN MODE selector should not be turned while the computer is running (Depress STOP switch on operation panel before turning selector). The TEST MODE light is ON is the selector is in other than NORMAL RUN position.

1. The NORMAL RUN position is for normal operation.

2. The PROGRAM LO RUN position allows normal operation except when a halt instruction, JHT, is executed and the computer stops for a period of time whichis controlled by the LO RUN FREQUENCY CONTROL.

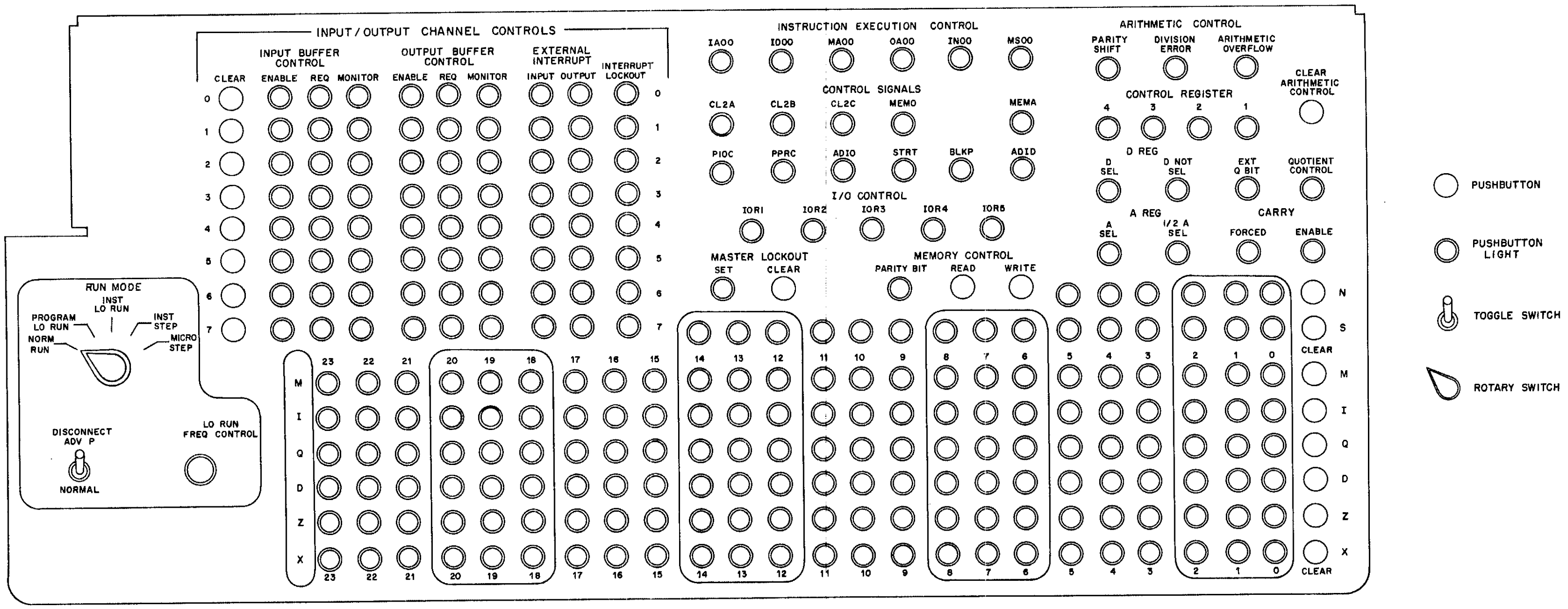


Figure 8. Maintenance Panel

3. The INST LO RUN position allows normal operation, except when, after the completion of the execution of each instruction, the computer stops for a period of time controlled by the LO RUN FREQUENCY CONTROL.

4. The INST STEP position allows normal execution of one instruction for each depression of the RUN switch on the operation panel.

5. The MICRO STEP position allows execution of one micro step for each depression of the RUN switch on the operation panel. All instructions except those with a store cycle can be micro-stepped while maintaining the correct execution.

The LO RUN FREQ CONTROL allows the time the computer stops in RUN MODEs 2 and 3 to be varied from 0.1 second to 2 seconds. Counter-clockwise rotation increases the delay.

The DISCONNECT ADV P switch is a two-position toggle switch. In the NORMAL position, the P-register is incremented during the execution of instrucitons, other than jump or skip types. In the DISCONNECT ADV P position, the advance of the P-register is suppressed and thus the same instruction is repeatedly executed for instructions other than jump or skip types. When in the DISCONNECT ADV P position, the TEST MODE light on the operation panel is ON.

The MASTER LOCKOUT status can be set by depressing the pushbutton-light labeled SET and can be cleared by depressing the pushbutton labled CLEAR. The SET pushbutton-light is ON when MASTER LOCKOUTS is in effect.

INSTRUCTION EXECUTION CONTROL

The IAOO light indicates instruction acquisition is in progress.

The IDOO light indicates indirect addressing is in progress.

The MAOO light indicates address modification is in progress.

The OAOO light indicates operand acquisition is in progress.

The INOO light indicates intermediate execution is in progress (not required by most instructions).

The MSOO light indicates a memory store is in progress.

CONTROL SIGNALS

The 3.0-megacycle clock is divided by 6 with a 3 stage counter to yield a 2 microsecond time interval.
CL2A comes on at the beginning of each 2-microsecond cycle and remains on for 1 microsecond.
CL2B comes on 2/3 microsecond after CL2A, and
CL2C comes on 2/3 microsecond after CL2B.

The MEMO light is ON if the MEMORY RANGE OVERFLOW INDICATOR is set.

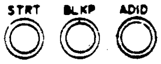
The MEMA light is ON if memory access is available to input-output.

The PIOC light is ON while in interrupt mode due to input-output parity error.

The PPRC light is ON while in interrupt mode due to the program parity error.

The ADIO light is ON while transferring I_{0-13} to S or P registers.

The BLKP light is ON during the IA cycle while in interrupt mode to suppress the advancing of the P-register.



The STRT light is ON when memory is either available for input-output or available for fetching the next instruction.

The ADID light is ON when indexing.

I/O CONTROL

IOR1

○

IORI - light will be on during the fetching of a buffer control word or the Real Time Clock, when calling up the fixed location for storing the Interrupt Status Word while servicing an input external interrupt; when selecting the channel during either an EXI, EXO, or EXF instruction.

IOR2

○

IOR2 - light will be on when inputting or outputting the first half(most significant) of a 30 bit word during an input buffer or output buffer; when inputting the first half of the Interrupt Status Word on a 30 bit channel; when inputting or outputting the first half of a 30 bit word during an EXO, EXI or EXF instruction.

IOR3

○

IOR3 - light will be on when inputting or outputting the data during an input or output buffer on a 24 bit channel; when inputting the Interrupt Status Word on a 24 bit channel; when inputting or outputting data on a 24 bit channel during either on EXO, EXI or EXF instruction; when inputting or outputting the second half (least significant) of a 30 bit word during an input or output buffer; when inputting the second half of a 30 bit word during an EXO, EX* or EXF instruction, and second half of interrupt status word on 30 bit channel and while incrementing the real time clock.

IOR4

○

IOR4 - light will be on when storing the Buffer Control Word or the Real Time Clock.

IOR5

○

IOR5 - The IOR5 light will be on during the Acknowledge for a 30 bit channel.

READ

○

Depressing the READ pushbutton causes the contents of the location specified by the S-register to be fetched into the M-register.

WRITE

○

PARITY BIT

○

Depressing the WRITE pushbutton causes the contents of the M-register to be stored at the location called out by the contents of the S-register. The PARITY BIT pushbutton light in ON when the last stored word has an even number of ONES.

I/O CHANNEL Control

CLEAR

○

CLEAR CHANNEL PUSHBUTTON - depressing a clear channel pushbutton resets all channel control flip-flops and all indicator lights on the channel are turned off.

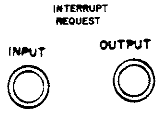
ENABLE

○

BUFFER ENABLE - depressing the pushbutton light will cause the indicator light to come on and set the enable buffer flip-flop. The light indicates a buffer is established on the channel.

DATA REQUEST - depressing the pushbutton light will cause the indicator light to come on and will set the data request flip-flop. The light indicates that there is data available on the line for the input channel and the peripheral device is ready for data on the output channel.

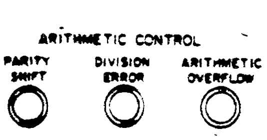
MONITOR - depressing the pushbutton light will cause the indicator light to come on and will set the monitor flip-flop. The light indicates that an interrupt will occur when the buffer is terminated.



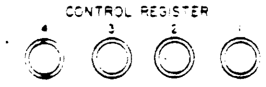
INTERRUPT REQUEST - depressing the pushbutton light will cause the indicator lights to come on and will set the interrupt request state. The light indicates that the peripheral device wants to interrupt the computer. If it is an input interrupt request the input data lines contain an interrupt status word.

INTERRUPT LOCKOUT - depressing the pushbutton light will cause the indicator light to come on and will set the interrupt lockout flip-flop. The light indicates that no interrupts can occur on that channel.

ARITHMETIC CONTROL



The PARITY SHIFT light is ON whenever the PARITY SHIFT indicator is 1.
The DIVISION ERROR light is ON whenever the DIVISION ERROR indicator is set.
The ARITHMETIC OVERFLOW light is ON whenever the ARITHMETIC OVERFLOW indicator is set.



The four pushbutton-lights of the ARITHMETIC CONTROL register are ON as described below:

(1 is ON, 0 is OFF)

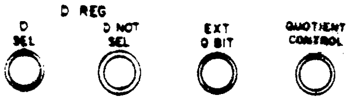
PHASE	4321	
A	0000	During all one-cycle manipulations
B	1000	During scale A
L	0100	During division
P	1100	During division
C	0010	During shifting
E	1010	During scale AQ
M	0110	During division
N	1110	During division
D	0001	During division and negate A
Q	1001	During division
R	0101	During division
S	1101	During division
F	0011	During reverse bits of A and Q
K	1011	During multiply
G	0111	During reverse bits of A and Q
J	1111	During reverse bits of A and Q

CLEAR
ARITHMETIC
CONTROL

○

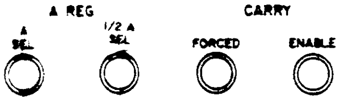
Depressing the CLEAR ARITHMETIC CONTROL pushbutton causes all Arithmetic control Flip Flops to be reset.

D REG



The D SEL pushbutton light is ON when the D-register is selected to be gated into the adder. The EXT Q BIT pushbutton light is the extra bit of the Q-register used in multiplication and division. The QUOTIENT CONTROL pushbutton light indicates that the partial remainder has equalled zero during the division process.

A REG

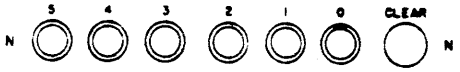


The A SEL pushbutton light is ON when the A-register is selected to be gated into the adder.

The $\frac{1}{2}$A SEL pushbutton light is ON when the contents of the A-register shifted right one bit are gated into the adder.

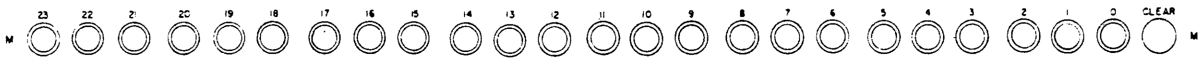
The FORCED pushbutton-light is ON when a carry is forced into the low-order stage of the adder.

The ENABLE pushbutton-light is ON during arithmetic operations requiring use of the adder carry circuits.

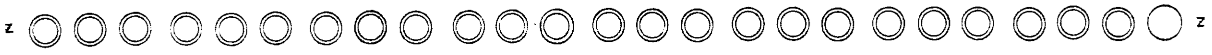


The N register is continuously displayed by 6 pushbutton-lights. Depressing a pushbutton causes that bit of the N register to be set to a 1 and turns the light ON. The register may be cleared to all zeros by depressing the CLEAR pushbutton.

The Q register is continuously displayed by 24 pushbutton-lights. Depressing a pushbutton causes that bit of the Q register to be set to a 1 and turns the light ON. The register may be cleared to all zeros by depressing the CLEAR pushbutton.

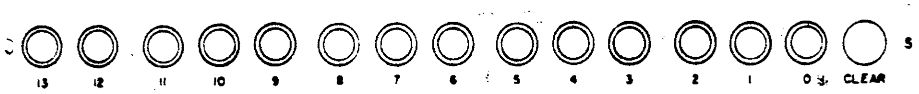


The M register is continuously displayed by 24 pushbutton-lights. Depressing a pushbutton causes that bit of the register to be set to a 1 and turns the light ON. The register may be cleared to all zeros by depressing the CLEAR pushbutton.

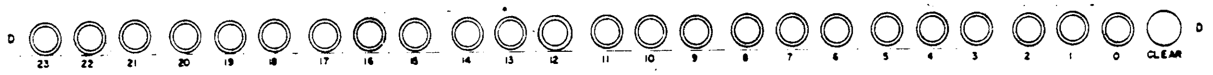


The Z register is continuously displayed by 24 pushbutton-lights. Depressing a pushbutton causes that bit of the Z register to be set to a 1 and turns the light ON. The register may be cleared to all zeros by depressing the CLEAR pushbutton.

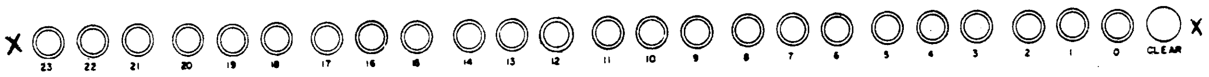
The I register is continuously displayed by 24 pushbutton-lights. Depressing a pushbutton causes that bit of the I register to be set to a 1 and turns the light ON. The register may be cleared to all zeros by depressing the CLEAR pushbutton.



The S register is continuously displayed by 14 pushbutton-lights. Depressing a pushbutton causes that bit of the S register to be set to a 1 and turns the light ON. The register may be cleared to all zeros by depressing the CLEAR pushbutton.



The D register is continuously displayed by 24 pushbutton-lights. Depressing a pushbutton causes that bit of the D register to be set to a 1 and turns the light ON. The register may be cleared to all zeros by depressing the CLEAR pushbutton.



The X register is continuously displayed by 24 pushbutton-lights (30 indicators, when 30 bit channels are implemented). Depressing the pushbutton caused that bit of the X register to be set to a 1 and turns the light ON. The register may be cleared to all zeros by depressing the CLEAR pushbutton.

OPERATION

START-UP PROCEDURE

The following procedure should be used to turn on the computer:

1. Place the COMPUTER POWER switch on the Operating Panel to the "ON" position.
2. If the program to be run is in memory, set the starting address into the P register. If the program is not in memory, it may be loaded by use of a load routine or the bootstrap load sequence. The bootstrap load sequence is

TABLE OF POWERS OF TWO AND RECIPROCALS

2^m	m	2^{-m}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5
2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25
4 398 046 511 104	42	0.000 000 000 000 227 373 675 443 232 059 478 759 765 625
8 796 093 022 208	43	0.000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
17 592 186 044 416	44	0.000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
35 184 372 088 832	45	0.000 000 000 000 028 421 709 430 404 007 434 844 920 703 125
70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 460 351 562 5
140 737 488 355 328	47	0.000 000 000 000 007 105 427 357 601 001 358 711 230 175 781 25
281 474 976 710 656	48	0.000 000 000 000 003 552 713 678 800 500 679 355 615 087 890 625
562 949 953 421 312	49	0.000 000 000 000 001 776 356 839 400 250 339 677 807 543 945 312 5

referenced by depressing the LOAD BOOTSTRAP switch on the Operating Panel. (For details see Bootstrap Load Sequence below.)

3. Depress the RUN switch.

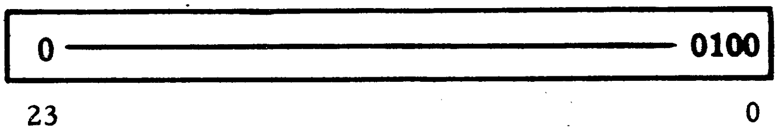
INSTRUCTION EXECUTION SEQUENCE

The computer is now in the run condition and the green RUN indicator illuminated. The instructions are executed consecutively. If an instruction is not a jump or skip type, the P register is incremented by one for each instruction. Thus, the normal arrangement of instructions is placed in ascending sequence in core storage.

BOOTSTRAP LOAD SEQUENCE

The COMPUTE POWER switches must be in the ON position. No error or warning indicators should be on. Status of the machine is cleared via depressing the MASTER CLEAR switch. The power must be ON for the input device on channel zero, and the device should be readied for use manually. The LOAD BOOTSTRAP switch is depressed and released; then the RUN switch is depressed to initiate execution of the following bootstrap load sequence:

1. An unconditional jump to 00000$_8$ is stored in location 00000$_8$.
2. A buffer control word is stored in location 00040. This word has a count of 1023 and an address of 00041$_8$.
3. The external function signal is applied to channel 0 and the word



is applied to the output lines of channel 0.

4. A buffer input mode with interrupt upon termination is established for channel 0.

5. Execution is started at location 0 upon depressing the RUN switch.

Execution proceeds via an interrupt to locating 00060 or 00120, depending on the type of bootstrap program being loaded.