FST-1 Computer Systems Manual

STOCK NO.: 67095031
RELEASED: October 1970

FAIRCHILD
SYSTEMS TECHNOLOGY

Warranty

Seller warrants equipment of its manufacture against defective materials or workmanship for a period of one year from date on which Seller determines the installation to be complete. The liability of the Seller under this warranty is limited, at Sellers option, solely to repair, replacement with equivalent Fairchild equipment, or an appropriate credit adjustment not to exceed the original equipment sales price, of equipment returned to the Seller provided that (a) Seller is promptly notified in writing by Buyer upon discovery of defects, (b) the defective equipment is returned to Seller, transportation charges prepaid by Buyer, and (c) Seller's examination of such equipment disclosed to its satisfaction that defects were not caused by negligence, misuse, improper installation, accident, or unauthorized repair or alteration by the Buyer. This warranty does not include mechanical parts failing from normal usage nor does it cover limited life electrical components which deteriorate with age such as vacuum tubes, choppers, lamps, etc. In the case of accessories, ie, card punches, typewriters, etc., not manufactured by Seller, but which are furnished with the Seller's equipment. Seller's liability is limited to whatever warranty is extended by the manufacturers thereof and transferable to the Buyer. This warranty is expressed in lieu of all other warranties, expressed or implied, including the implied warranty of fitness for a particular purpose, and of all other obligations or liabilities on the Seller's part, and it neither assumes nor authorizes any other person to assume for the Seller any other liabilities. This warranty should not be confused with or construed to imply free preventative or remedial maintenance, calibration or other service required for normal operation of the equipment.

List of Effective Pages

FST-1 SYSTEMS MANUAL

The total number of pages of this publication is 68, consisting of the following:

Page No.	Issue
Title	Original
i through ix	Original
1-1 through 1-42	Original
2-1 through 2-5	Original
3-1 through 3-5	Original
4-1 through 4-9	Original
5-1 through 5-6	Original
6-1 through 6-8	Original
A1 through A2	Original
B1 through B2	Original

TABLE OF CONTENTS

Page

SECTION I FST-1 SYSTEM DESCRIPTION, FORMATS AND INSTRUCTIONS (Continued)

SECTION II CONTROL PANEL

Page

SECTION III CONSOLE TYPEWRITER

SECTION IV CARD READER

SECTION V DISC FILE

SECTION V DISC FILE (Continued) Page

FIGURES

TABLES

SECTION I
FST-1 SYSTEM DESCRIPTION, FORMATS AND INSTRUCTIONS

1.1 INTRODUCTION

This section discusses the Central Processor Unit, its organization, reserved memory, instruction repertoire, and the accumulator and memory interface system.

Note: Octal notation is used as a convenient shorthand when discussing the binary contents of computer words, registers, etc. Thus operation codes and addresses should always be read as octal values.

1.2 CENTRAL PROCESSOR UNIT

The FST-1 Central Processor Unit (CPU) is a high-speed general purpose digital computer, with the following characteristics:

- 24-bit word length.
- Two's complement binary arithmetic.
- 1.75μsec complete cycle time.
- Eight hardware index registers.
- Indirect addressing with most instructions.
- Basic core memory of 4096 words, expandable in modules of 4096 words, up to 16,384 words, all directly addressable.
- Two memory buses for simultaneous access to two memory banks in systems having 8192 or more words of core.
- Direct Memory Access (DMA) on both memory buses.
- The standard DMA interface unit allows data to be either stored or retrieved at a rate of 571,428 words per second per memory bus, or 1,142,856 words per second total.
- The basic system has a capacity of up to 16 DMA channels on each memory bus.
- Sixteen external interrupt channels are also available on the basic system.
- Standard input/output equipment includes a console typewriter with attached paper tape reader and punch, card reader, disk, line printer and magnetic tape.

1.3 SYSTEM ORGANIZATION

The FST-1 System is "memory oriented." That is, the memory is the central part of the system, with all other system components interfaced as peripherals to memory. The memory system has two independent memory buses, with each bus having its own priority system. The CPU and all peripherals are assigned access priority when they are interfaced to the memory system.

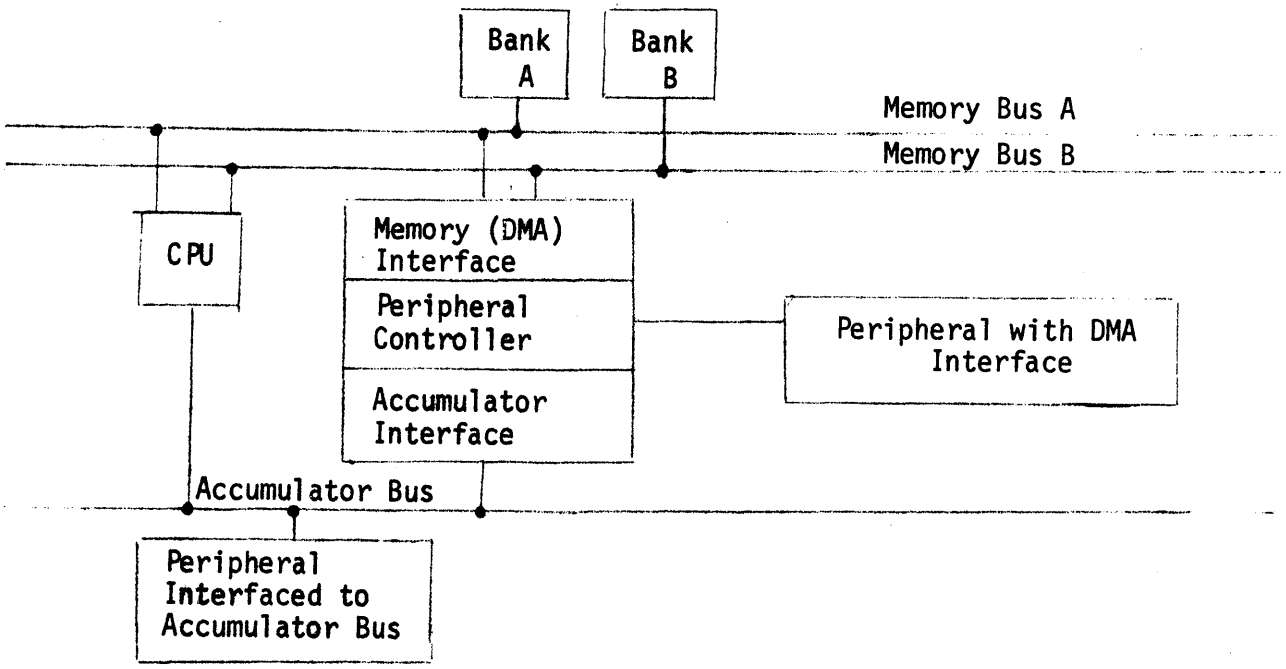


Figure 1.3.1

FST-1 System Organization with one Peripheral Interfaced to the Accumulator Bus and another with DMA Interface

Peripherals may be interfaced to either or both memory buses. The size of core memory can be 4096, 8192, 12,288, or 16,384 words. Any or all of the 4K blocks of memory can be tied to either bus. The CPU has the lowest memory access priority.

The accumulator bus provides a data and control path between the CPU and peripheral controllers. This bus is used by the CPU to select peripheral units and to read either data or status from the peripherals. In addition, it is used to transmit data and control to the peripheral controllers. The accumulator bus is also used when a peripheral requests an interrupt. The bus is used to establish priority between the interrupting devices and to direct the transfer of program control.

1.4 CENTRAL PROCESSOR UNIT ORGANIZATION

1.4.1 Introduction

This section describes the word and instruction formats and the use of the principal registers in the FST-1.

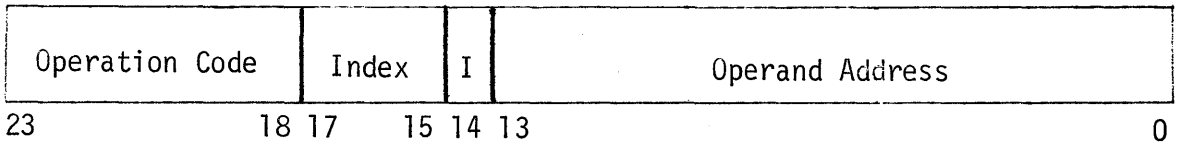
1.4.2 Word Format

The FST-1 CPU word is 24 bits. The bit positions are numbered from right to left, beginning with 0. Bit 23 of data words is the sign bit. Negative numbers are stored in two's complement form.

1.4.3 Instruction Formats

Standard Instruction Word

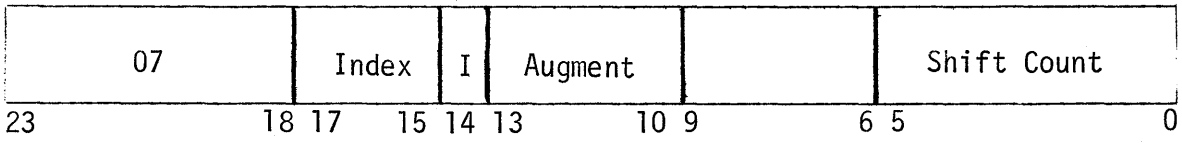
The standard instruction word format consists of a six (6) bit operation code field, a three (3) bit index field, a one (1) bit field used to specify direct (0) or indirect (1) addressing and a fourteen (14) bit address field.



Augmented Instructions

Some instructions do not require a fourteen bit address field, as used in the standard instruction word format. These instructions have operation code 07 and use bits 10 through 13 to 'augment' the 07 operation code. (See also Section 1.7.9.) An example of an augmented operation code is shown below:

Augmented Instruction Word:



1.4.4 Registers

The main registers of the FST-1 CPU are described below:

The Command Register, CR, is a 24-bit register which holds the current instruction for execution.

The A Register or Accumulator (24 bits) is the main arithmetic register.

The E Register is a 24-bit extension of the accumulator. It is used during double precision instructions, such as double-precision additions, multiplications, etc.

The B Register is the memory buffer register (24 bits). Operands are stored in B while the CPU executes arithmetic or data transfer instructions.

The P Register (14 bits) is the program counter. P contains the memory address of the instruction which follows the instruction currently in the command register.

The X Registers (14 bits each) are eight addressable index registers. Registers X1 through X7 may be used for instruction indexing, while X0 may not. X0 is used as the comparison index for both the ADD-TO-INDEX instructions or as a simple counter.

The W Register (24 bits) is the console switch register. It is used for manually loading the accumulator, command register and P counter. The W register can also be loaded into the accumulator under program control.

The R Register. This 6 bit register is the interrupt address director register. It holds one of 63 indirect addresses, to which an external interrupt can cause a program control transfer.

The CPU Registers are interfaced to the memory system buses with the memory interface logic. They are also interfaced to the peripheral controllers on the accumulator bus through the accumulator interface logic. The main data paths are illustrated in Figure 1.4.4.1.

1.4.5 Reserved Memory

There are 64 reserved memory locations in the FST-1 memory system. Memory address zero (00000_8) is reserved as a return address storage location for the BSZ instruction. Memory locations 1 through 63 (00001_8 - 00077_8) are reserved for indirect address storage for the external interrupt system. If the external interrupt system is not in use, these memory locations are available for normal program use.

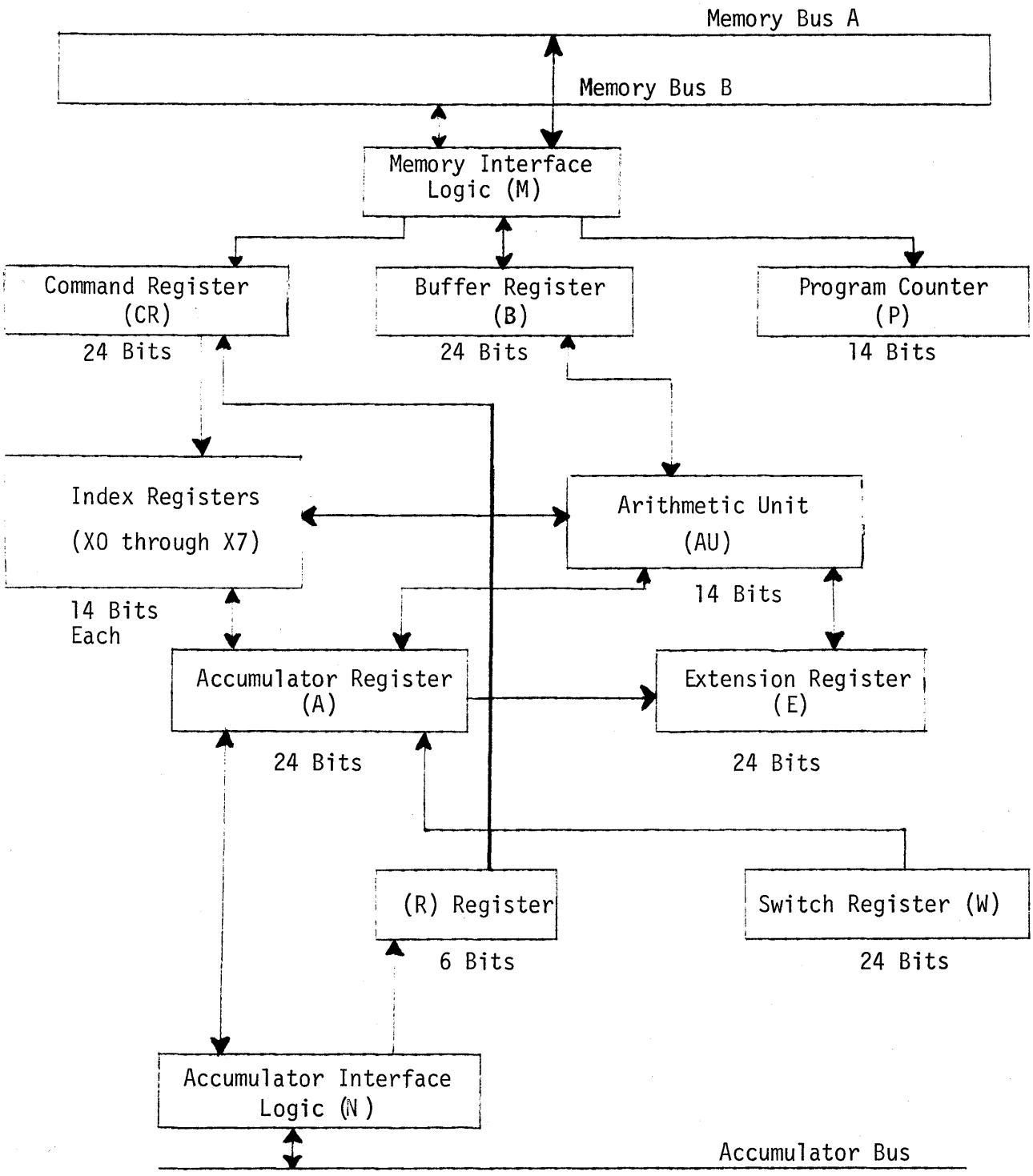


Figure 1.4.4.1

FST-1 Main Data Paths

1.5 ADDRESS MODIFICATION

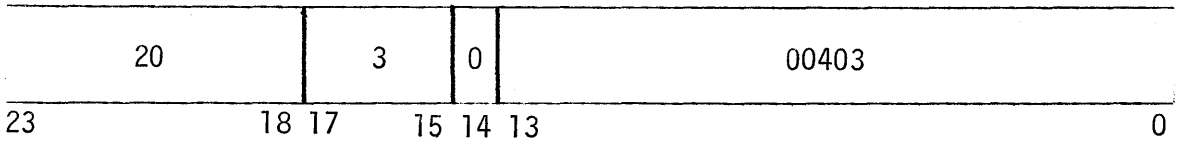
1.5.1 Introduction

FST-1 instructions can be address modified by means of either indexing or indirect addressing.

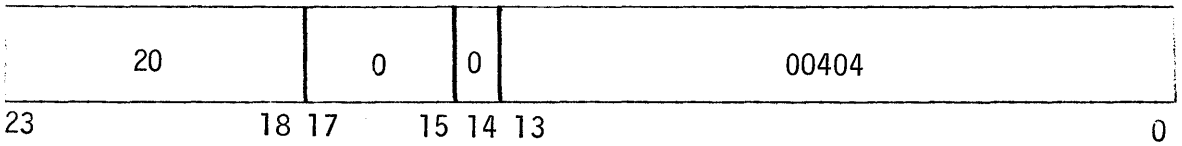
1.5.2 Indexing

Address modification by indexing is possible for all memory reference instructions, except those instructions which use bits 15, 16 and 17 for a purpose other than indexing, i.e., BAT, BOI, and BOS. Prior to the execution of an indexable instruction, the contents of the index register specified by bits 15, 16 and 17 of the instruction word are added to the operand address field of the instruction word. The resultant sum replaces the original operand in the command register.

For example, if index register 3 contains 00001 and the command register contains the following instruction:



the 00403 and the 00001 are added together, the sum replacing the 00403, so that the instruction in the command register changes to:

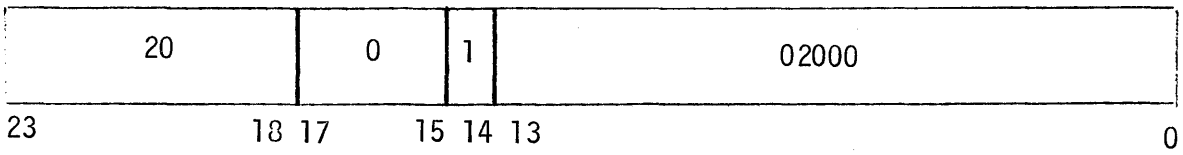


The instruction shown immediately above is the instruction actually executed. Note that the index address has been changed to 0. Index address 0 is interpreted as "no indexing" rather than specifying index register 0. It follows then that INDEX REGISTER 0 CANNOT BE SPECIFIED FOR ADDRESS MODIFICATION.

1.5.3 Indirect Address Modification

The second technique available in the FST-1 for instruction address modification is indirect address modification. Indirect address modification applies to all memory reference instructions except BAT, BOI, BOS and SPU. Indirect address modification occurs when bit 14 of an instruction word is set.

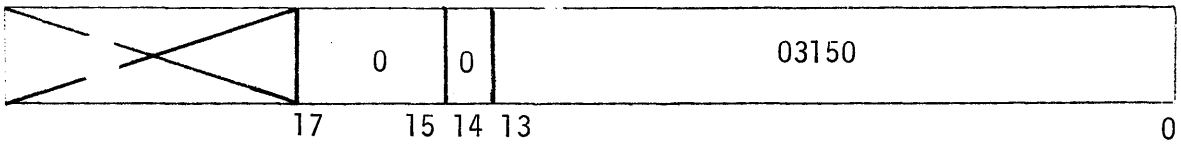
For example:



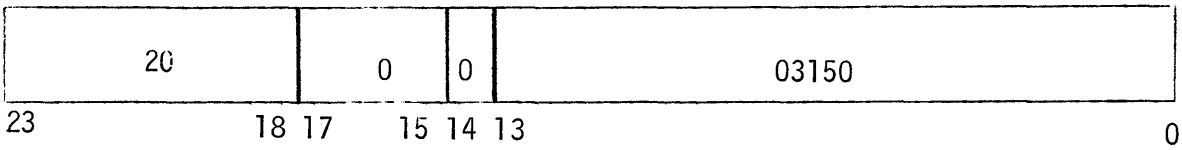
is a typical instruction specifying indirect address modification. The indirect address modification operation occurs as follows:

The instruction operand address is used to fetch a new operand address, indirect flag and index address bits from memory.

For example, if memory location 02000 contained



then the above ADD instruction would be equivalent to:



The FST-1 always performs indexing prior to indirect address modification. It is possible to index an instruction, fetch an indirect address word, which, when bits 0-17 are substituted in the instruction, may create another indexing operation and another indirect address modification. This in turn may cause another index operation and another indirect address modification, and so on. There is no theoretical limit to the number of indirect address cycles.

1.6 PERIPHERAL INTERFACE ORGANIZATION

1.6.1 Introduction

Peripherals are controlled by the FST-1 CPU via the Accumulator Bus. The accumulator bus is a programmed Input/Output channel. The Select Peripheral Unit (SPU) instruction commands CPU-peripheral communication via the accumulator bus. The accumulator bus is time-shared during the execution of a Select Peripheral Unit instruction to provide a peripheral unit address and command and a 24-bit data transfer either to or from the addressed peripheral. Peripheral status is also transmitted via the accumulator bus during every Select Peripheral Unit instruction execution.

Special synchronization signals allow for the data lines of the accumulator bus to be used as program interruption requests. Sixteen levels of priority interrupt request are available in the FST-1 system. Honored interruptions require the requestor to submit a six-bit interrupt identification code via the accumulator bus. This code is interpreted as an address to one of the 63 low-core memory addresses (location 00000_8 excluded) for purposes of interruption servicing.

1.6.2 Program Interruption

The program interruption facilities of the FST-1 computing system provide for the diversion of the CPU from its normal tasks to the processing of a subsidiary task in response to an external request. Sixteen fixed-priority levels of interrupt are available. Each individual request may be enabled or disabled through the Select Peripheral Unit (SPU) command, or, the entire structure of interrupts may be enabled or disabled via the Set State (SST) or Reset State (RST) instructions, respectively.

Interruption of the normal program is allowed at the conclusion of one program step, while its successor is being fetched from memory. At this time, the presence of an interrupt request causes the command register to be loaded with an indirect Branch Store Memory (BSM) command rather than a new instruction word. The Program Counter is inhibited so as to preserve the location of the supplanted instruction. The BSM command word also obtains a six-bit address referencing one of the addresses 01_8 through 77_8 from the interrupt requestor.

These 63 memory locations constitute a branch table for entry to subroutines designed for processing the various interrupt conditions. This table contains appropriate addresses supplied by the interrupt requestor for the prevailing interrupt condition or conditions.

The address stored in the branch table is used in the normal way by the BSM command; (indexing and further indirection may occur). After tracing through all indirections, the final address receives the contents of the indicators GT, EQ, LT, BE & OV and the contents of the program counter, which locates the return point to the interrupted program. Program control resumes at the next sequential address following the return address.

The requestor of an accepted interrupt maintains its interrupt request until receipt of an acknowledgement. However, at the recognition of the interrupt by the CPU, the Interrupt-Enable flip-flop is cleared, which in turn removes the interrupt synchronizing signal from all peripherals. Therefore, interrupt requests are masked until the Interrupt Enable flip-flop is again set. A Set-State instruction is required for this purpose.

Each interrupt routine must provide for a completion SPU instruction to return the interrupt requestor to its normal state. Furthermore, an SST instruction is needed to re-enable the interrupt system.

Return to the main program via an indirect unconditional branch (BRU*) on the return address, placed at the beginning of the subroutine by the interrupt induced BSM command, will restore the indicators GT, EQ, LT, BE, and OV to their original state.

1.7 INSTRUCTION REPERTOIRE

1.7.1 Introduction

This section discusses the instruction repertoire of the FST-1. It consists of nine instruction groups, totaling 48 instructions. Accompanying each instruction are examples coded as they would be for the FST-1 assembler.

1.7.2 Abbreviations

The appendix contains a list of abbreviations which are used in the description of machine instructions. Any abbreviation which is enclosed in parentheses is a reference to the contents of that particular register or memory location. For example, (M) is a reference to the contents of memory location M; (A) refers to the contents of the accumulator, etc. (A)→M is read, "The contents of A go to memory location M." (M) + (A)→A is read, "The contents of memory location M, plus the contents of A go to A." In the following instruction descriptions, M_e is used to refer to the effective memory address, i.e., after both indexing and indirection.

Subscripts are used to reference individual bits or groups of bits in the registers. For example, A_0 represents the "0" bit of the accumulator; A_{0-7} represents the least significant eight bits of the accumulator, etc.

1.7.3 Instruction Format(s)

An instruction word generally has four parts: an Operation Code, an Index Address, an Indirect Address Indicator, and an Operand Address. If an insturction can be indexed, the instruction format will include an X, (which designates the index to be used). If an instruction can be indirect address modified, the format will include an I. If the instruction can reference memory, then the M in the instruction format is the memory location of the operand.

1.7.4 Assembler Formats

Each of the instruction descriptions which follow is illustrated with an example of the appropriate FST-1 symbolic assembly code. The FST-1 Assembler manual should be consulted for the details of instruction, formats, conventions, etc. The discussion presented here will clarify the instruction descriptions.

All instructions have an Opcode and most of them also have an Operand, although some others, (such as TCA), do not. "Opcode" is the mnemonic name for the command or Operation Code and "Operand" is the symbolic address of the operand, etc. . .

There are two basic operand formats: indexable and non-indexable, viz:

(1) Indexable instruction: Symbolic address, index

Examples:

```
STA    BUFFER,5
LDA    TABLE (no index specified)
```

(2) Non-indexable instruction:

(a) index register, operand address
(b) indicator value, operand address
(c) state flip-flop, operand address

Examples:

```
BOS    5,L1
BOS    PASSL,L232
BOI    3,LEQ
LDX    6,5
LDX    PNTRX,TABLE-2
```

Indirect addressing is noted by an asterisk (*) immediately following the opcode mnemonic, viz:

```
LDE* TEMP1,3
MUL* FACTOR
```

1.7.5 Cycles Required

In each of the instruction descriptions which follow, the number of machine cycles required to execute the instruction is given - exclusive of indexing and indirection. A memory cycle is 1.75 microseconds in duration.

1.7.6 Arithmetic Instructions

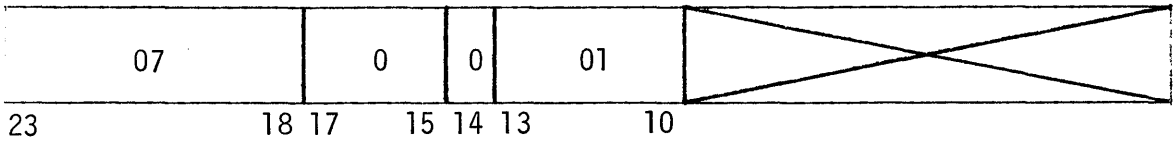
This section contains the following 10 arithmetic instructions: TCA, DTC, ADD, SUB, DADD, DSUB, MUL, DIV, AOM and SOM.

TCA TWO'S COMPLEMENT A REGISTER

Definition: Two's Complement of (A)→A

Cycles Required: 1

Instruction Format (Augmented):



Description: The contents of the accumulator are two's complemented and placed in the accumulator.

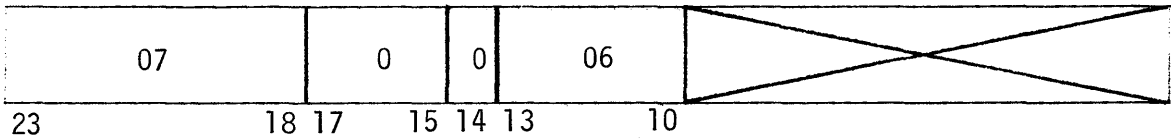
Assembler Format: TCA

DTC DOUBLE TWO'S COMPLEMENT

Definition: Two's Complement of (A and E)→A and E

Cycles Required: 2

Instruction Format (Augmented):



Description: The contents of A and E are two's complemented and the result is placed in A and E.

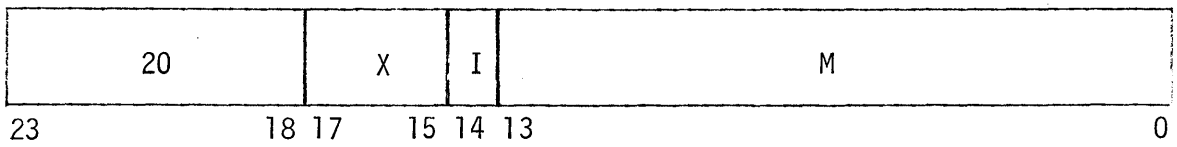
Assembler Format: DTC

ADD ADDITION

Defintion: (A) + (M_e)→A

Cycles Required: 2

Instruction Format:



Description: The contents of memory location, M_e, are added algebraically to the contents of the accumulator, with the sum being stored in the accumulator. The contents of memory are not changed. An overflow from the accumulator will set the overflow flag OV, indicating the result is incorrect.

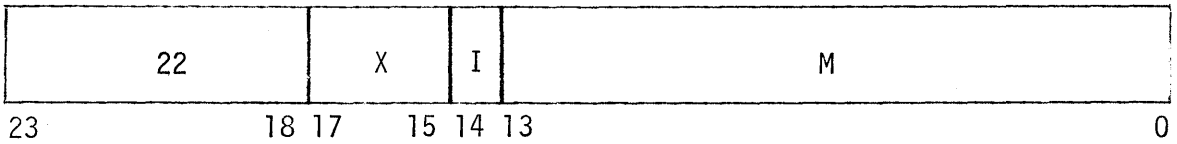
Assembler Format: ADD* TABLE+1,2

SUB SUBTRACTION

Definition: (A) - (M_e) →A

Cycles Required: 2

Instruction Format:



Description: The contents of memory location, M_e, are subtracted algebraically from the contents of the accumulator, with the difference being stored in the accumulator. The contents of memory are not changed. An overflow from the accumulator will set the overflow flag OV, indicating the result is incorrect.

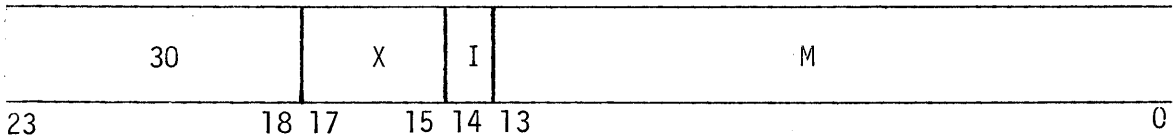
Assembler Format: SUB* TABLE+1,7

DADD DOUBLE ADDITION

Definition: (A and E) + (M_e and M_e + 1)→ A and E, where M_e is an even numbered address.

Cycles Required: 4

Instruction Format:



Description: The contents of memory locations, M_e and M_e + 1 are added algebraically to the contents of A and E. Bits 23 of (M_e) and (A) are the operand signs. The sum is stored in A and E as a 47 bit signed number with A containing the most significant half of the sum. The sign of the sum is stored in A_{23}. Two's complement is used for negative numbers. The contents of memory are unchanged by the operation. An overflow will set the overflow flag OV, indicating the result is incorrect.

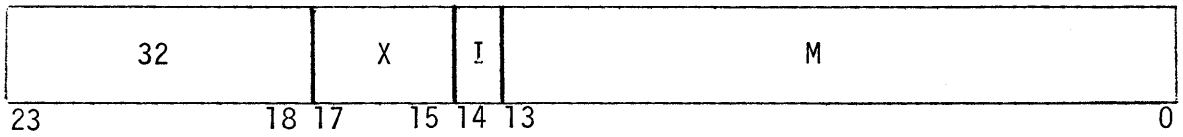
Assembler Format: DADD* TABLE+1

DSUB DOUBLE SUBTRACTION

Definition: (A and E) - (M_e and M_e + 1) →A and E, where M_e is an even numbered address.

Cycles Required: 4

Instruction Format:



Description: The contents of M_e and M_e + 1 are subtracted algebraically from the contents of A and E. Bits 23 of (M_e) and (A) are the signs of the operands. The difference is stored in A and E as a 47 bit signed number, with A containing the most significant half. The sign of the difference is stored in A_{23}. Two's complement notation is used for negative numbers. The contents of memory are unchanged by the operation. An overflow will set the overflow flag OV, indicating the result is incorrect.

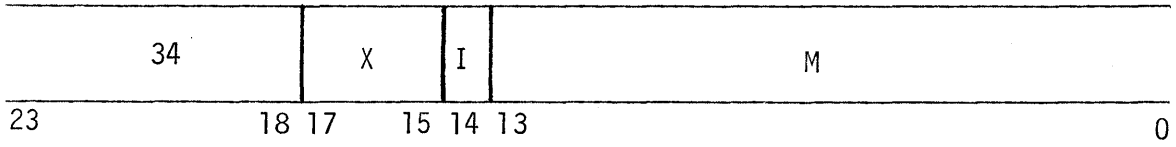
Assembler Format: DSUB* TABLE+1

MUL MULTIPLY

Definition: $(M_e) \times (A) \rightarrow A$ and E

Cycles Required: 25

Instruction Format:



Description: The contents of memory location, M_e, are multiplied by the contents of the accumulator. The product is stored in A and E, with A containing the most significant half. A and M_e are assumed to be positive numbers. The contents of memory are not changed.

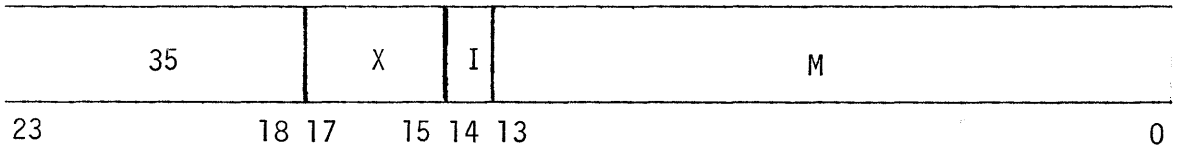
Assembler Format: MUL* ALPHA

DIV DIVISION

Definition: (A and E)/$(M_e) \rightarrow$ E
Remainder $\rightarrow$ A

Cycles Required: 25

Instruction Format:



Description: The contents of A and E are divided by the contents of memory location M_e. The quotient is left in E and the remainder in A. The original contents of A, E and M_e are assumed to be positive. The contents of memory are not changed. A divide overflow will occur if $(A) \geq (M_e)$. For this condition, the divide is terminated and the overflow flip flop is set. In the event of an overflow, A and E remain shifted left one place.

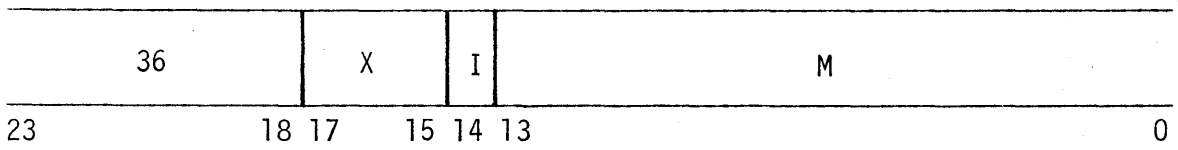
Assembler Format: DIV* ROGER,2

AOM ADD ONE TO MEMORY

Definition: $(M_e) + 1 \rightarrow M_e$

Cycles Required: 4

Instruction Format:



Description: The contents of memory location, M_e, are incremented by one (1). An overflow condition will cause the OV flag to be set. In the event of an overflow, the result of the operation is incorrect.

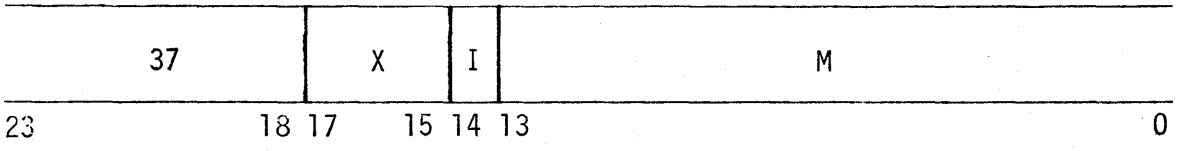
Assembler Format: AOM* BETA

SOM SUBTRACT ONE FROM MEMORY

Definition: $(M_e) - 1 \rightarrow M_e$

Cycles Required: 4

Instruction Format:



Description: The contents of memory location, M_e, are decremented by one (1). An overflow condition will cause the OV flag to be set. In the event of an overflow, the result of the operation is incorrect.

Assembler Format: SOM TEM1

1.7.7 Data Transfer Instructions

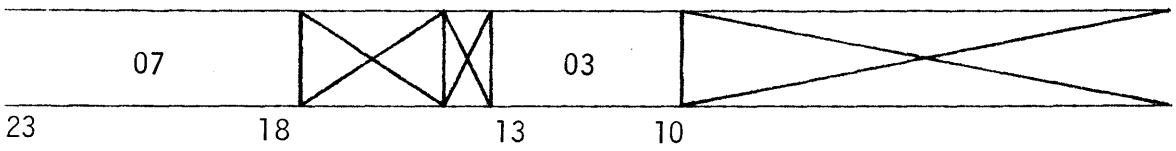
This section contains the following 8 data transfer instructions: RSR, EXC, STA, STE, LDA, LDE, DLD and DST.

RSR READ SWITCH REGISTER

Definition: (W)→ A

Cycles Required: 1

Instruction Format (Augmented):



Description: The contents of the console switch register, W, are loaded in the A register.

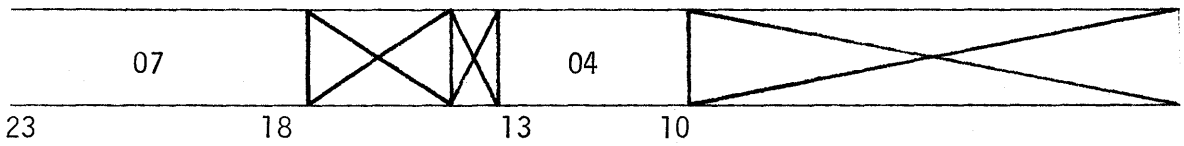
Assembler Format: RSR

EXC EXCHANGE A AND E

Definition: (A)→ E, (E)→ A

Cycles Required: 1

Instruction Format (Augmented):



Description: The contents of the A register and the contents of the E register are exchanged.

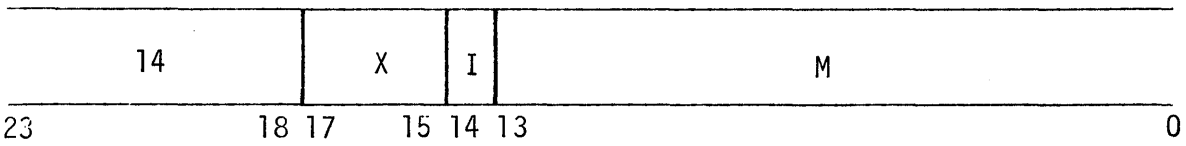
Assembler Format: EXC

STA STORE A

Definition: $(A) \rightarrow M_e$

Cycles Required: 2

Instruction Format:



Description: The contents of the A register are stored in memory location, M_e. The contents of A are not changed.

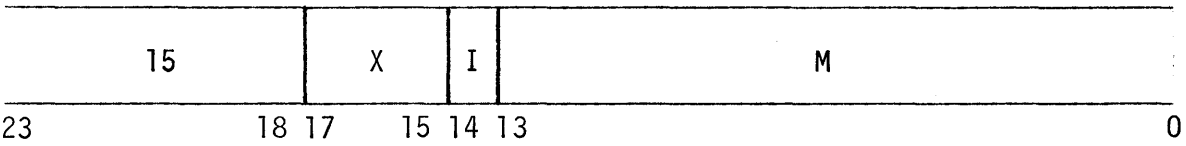
Assembler Format: STA* ALP1,2

STE STORE E

Definition: $(E) \rightarrow M_e$

Cycles Required: 2

Instruction Format:



Description: The contents of the E register are stored in memory location M_e. The contents of E are not changed.

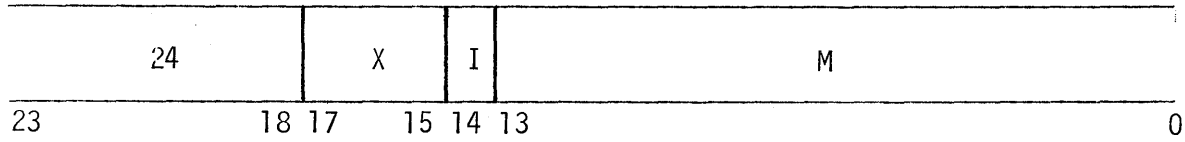
Assembler Format: STE* TEMP5

LDA LOAD A

Definition: $(M_e) \rightarrow A$

Cycles Required: 2

Instruction Format:



Description: The contents of memory location, M_e, are copied into the accumulator A. The contents of memory are not changed.

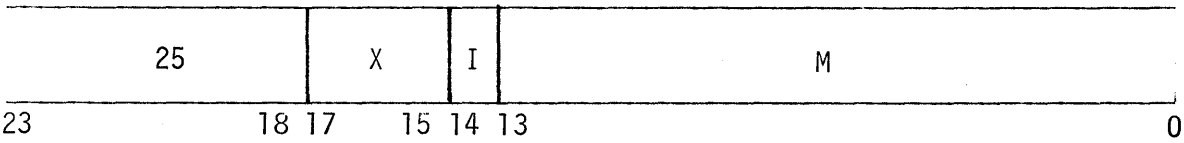
Assembler Format: LDA* TABLE,XPNTR

LDE LOAD E (The Accumulator Extension)

Definition: $(M_e) \rightarrow E$

Cycles Required: 2

Instruction Format:



Description: The contents of memory location, M_e, are copied into the accumulator extension E. The contents of memory are not changed.

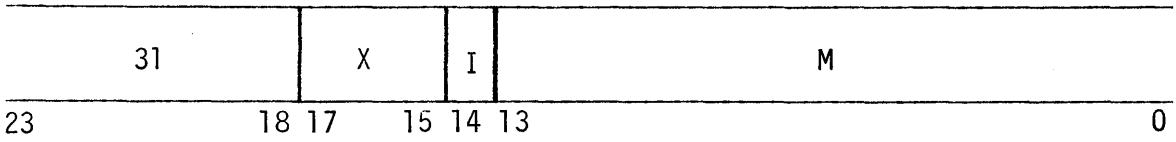
Assembler Format: LDE* TABLE+1

DLD DOUBLE E LOAD A AND E

Definition: (M_e and $M_e + 1$) → A and E, where M_e is an even numbered address.

Cycles Required: 3

Instruction Format:



Description: The contents of memory location, M_e and $M_e + 1$, are loaded into A and E respectively. The contents of memory are not changed.

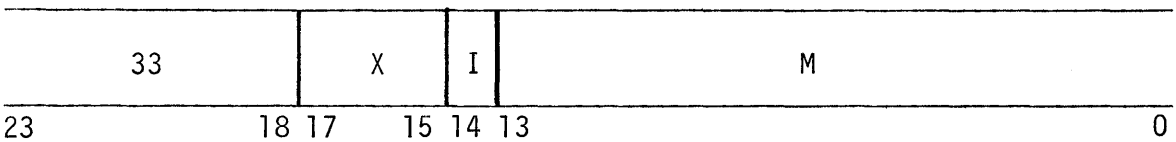
Assembler Format: DLD* TABLE+5,2

DST DOUBLE STORE

Definition: (A and E) → M_e and $M_e + 1$, where M_e is an even numbered address.

Cycles Required: 3

Instruction Format:



Description: The contents of the A and E registers are stored in memory locations, M_e and $M_e + 1$ respectively. The contents of A and E are not changed.

Assembler Format: DST TEMP2+2

1.7.8 Index Instructions

This section contains the following 5 instructions: LDX, LXA, ATX, STX and LAX.

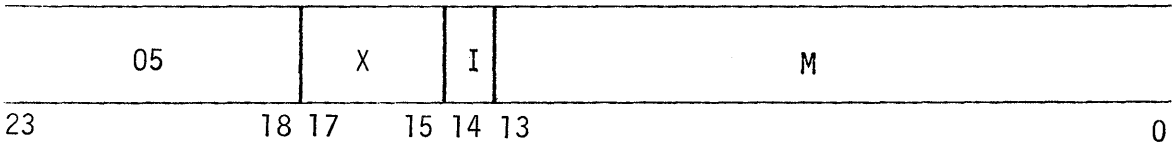
Note on Index Register Usage: It is standard practice to use the index registers in the FST-1 in adjacent pairs, viz: X7 with X6, X5 with X4, X3 with X2 and X1 with X0. When so used, the odd index is the active, working index while the even index is the limit index for comparison purposes.

LDX LOAD INDEX

Definition: $M_e \rightarrow X_n$

Cycles Required: 1

Instruction Format:



Description: The effective address, M_e is loaded into the addressed index register. Index address modification does not occur, but a special form of indirect addressing does take place: bits 14-0 are replaced in the command register from memory address M, bits 17-15 being obtained from the current instruction word and <u>not</u> from memory address M. (See also ATX and STX)

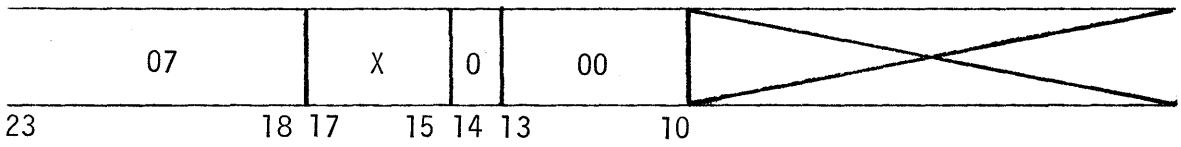
Assembler Format: LDX* 7,200

LXA LOAD INDEX FROM A

Definition: $(A_{0-13}) \rightarrow X_n$

Cycles Required: 1

Instruction Format (Augmented):



Description: The addressed index register, X_n, is loaded from the contents of the accumulator bits, A_0 through A_{13}. Index address modification does not occur.

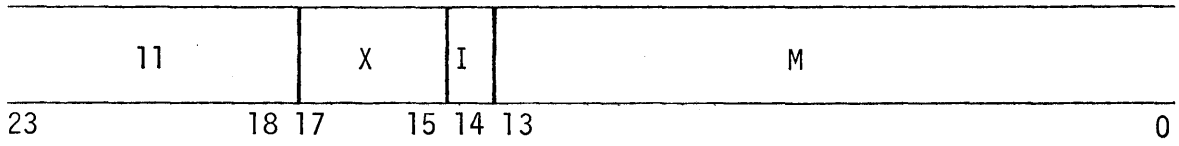
Assembler Format: LXA INDX3

ATX ADD TO INDEX

Definition: $M_e + (X) \rightarrow X$

Cycles Required: 2

Instruction Format:



Description: The contents of the addressed index register are added to the effective address (after indirect address modification) and the sum is placed back in the addressed index register. Then the GT, EQ, and LT indicators are set by comparing X_n (the addressed index register) to X_{n-1} ('n' must be odd). Index address modification does not occur.

Note that only bits 14-0 are replaced in the command register from memory address M under indirection. (See also LDX and STX).

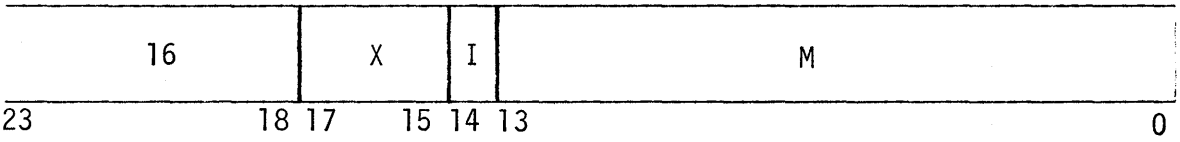
Assembler Format: ATX* 5,TABLE1

STX STORE

Definition: $(X_n) \rightarrow M_e$

Cycles Required: 2

Instruction Format:



Description: The contents of the addressed index register X are stored in memory location M_e. The contents of X are unchanged. Bits 14-23 of M_e are zeroed. Index address modification does not occur, but a special form of indirect addressing does take place: bits 14-0 are replaced in the command register from memory address M, bits 17-15 being obtained from the current instruction word and not from memory address M. (See also ATX and LDX).

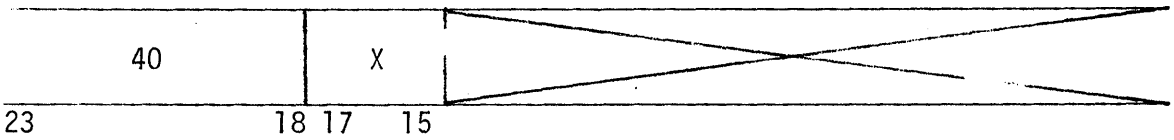
Assembler Format: STX 5,TEMP1

LAX LOAD A FROM INDEX

Definition: $(X_n) \rightarrow A$

Cycles Required: 1

Instruction Format:



Description: The contents of the specified index register are transferred to A. Bits 23-14 of A are zeroed.

Assembler Format: LAX 2

1.7.9 Shift Instructions

This section consists of the following 9 augmented instructions: DSN, SR, LS, SA, SL, DSR, LDS, DSA and DSL. The execution time depends upon the number of bit positions to be shifted.

The following table illustrates the execution time versus the number of shifts:

2 cycles for $J_e \leq 9$
3 cycles for $9 < J_e \leq 14$
4 cycles for $14 < J_e \leq 19$
5 cycles for $19 < J_e \leq 24$
6 cycles for $24 < J_e \leq 29$
7 cycles for $29 < J_e \leq 34$
8 cycles for $34 < J_e \leq 39$
9 cycles for $44 < J_e \leq 49$

Expressed as a formula:

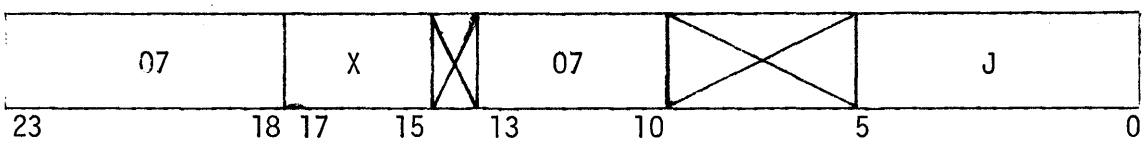
$$T = 2 + \underbrace{\lfloor (J_e - 9)/5 \rfloor}_{\text{integer}} \quad \text{cycles.}$$

DSN DOUBLE SHIFT NORMALIZE

Definition: Normalize A and E

Cycles Required: $2 + [(J-9)/5]$ (integer)

Instruction Format:



Description:

The contents of A and E are shifted left J_e bit positions, or until the information in bit position A_{23} differs from that in A_{22}. E_{23} shifts into A_0 and zeros are entered into E_0. At the termination of the shifting, the contents of the shift counter are stored in Index register zero. DSN may use indexing; the contents of X are added to J to obtain the modified shift count, J_e.

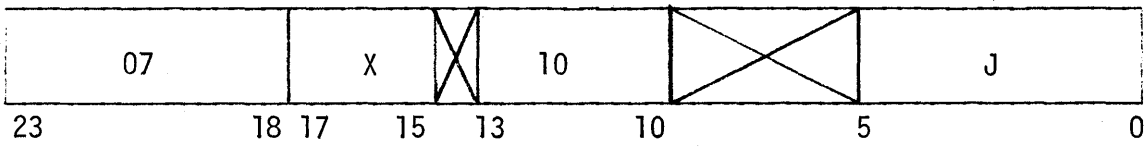
Assembler format: DSN 10

SR SHIFT RIGHT

Definition: Shift (A) Right Arithmetical

Cycles Required: $2 + [(J-9)/5]$ (integer)

Instruction Format:



Description:

The contents of the A register are shifted right J_e bit positions. The sign bit, bit 23, of the A register is copied into bit position 22 as the register is shifted. Bits shifted from A_0 are lost. SR may use indexing; the contents of X_n are added to J to obtain the modified shift count, J_e.

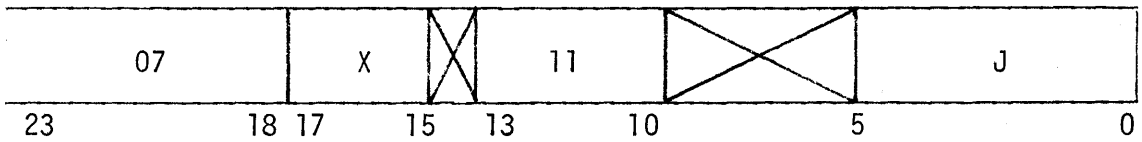
Assembler format: SR 5

LS LOGICAL SHIFT

Definition: Shift (A) Right Logical

Cycles Required: $2 + [(J-9)/5]_{\text{integer}}$

Instruction Format:



Description:

The contents of the A register are shifted right J_e bit positions, zeros being entered into A from the left (A_{23}). LS may use indexing; the contents of X are added to J to obtain the modified shift count, J_e.

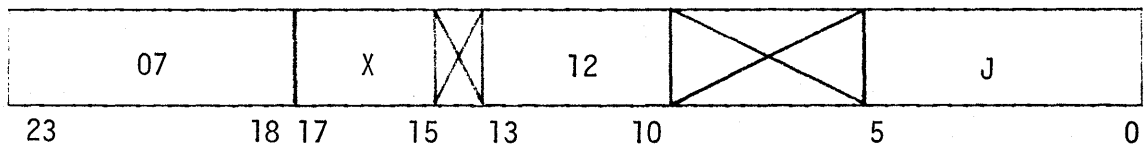
Assembler Format: LS 5

SA SHIFT AROUND

Definition: Shift (A) Left Around

Cycles Required: $2+ [(J-9)/5]_{\text{integer}}$

Instruction Format:



Description:

The contents of the A register are shifted left around J_e bit positions, with A_{23} shifting into A_0. SA may use indexing; the contents of the X are added to J to obtain the modified shift count, J_e.

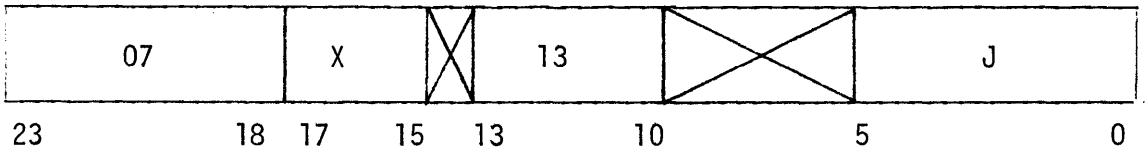
Assembler format: SA 4,5

SL SHIFT LEFT

Definition: Shift (A) Left End Off

Cycles Required: $2 + [(J-9)/5]_{integer}$

Instruction Format:



Description:

The contents of the A register are shifted left J_e bit positions, with zeros being entered into A_0. SL may use indexing; the contents of X are added to J to obtain the modified shift count, J_e.

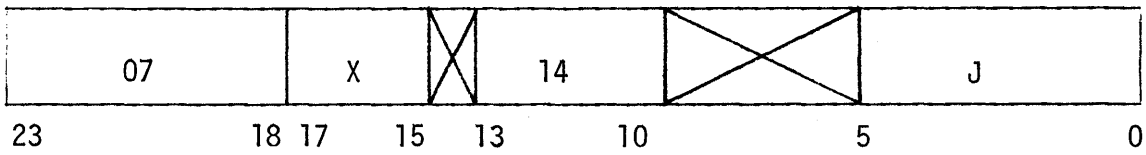
Assembler format: SL 0,3

DSR DOUBLE SHIFT RIGHT

Definition: Shift A and E Right Arithmetical

Cycles Required: $2 + [(J-9)/5]_{integer}$

Instruction Format:



Description:

The contents of the A and E registers are shifted right (A_0 shifting into E_{23}) J_e bit positions. The sign of A (A_{23}) does not change during this shift operation and is repeatedly copied into A_{22} during the shift. DSR may use indexing; the contents of X are added to J to obtain the modified shift count, J_e.

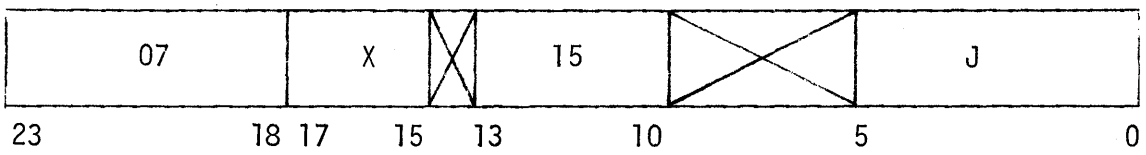
Assembler format: DSR 25

LDS LOGICAL DOUBLE SHIFT

Definition: Logical Shift A and E Right

Cycles Required: $2 + [(J-9)/5]_{\text{integer}}$

Instruction Format:



Description:

The contents of A and E are shifted right J_e bit positions. Zeros are entered into A_{23}. In addition, A_0 is shifted into E_{23}, while bits shifted out of E_0 are lost. LDS may use indexing; the contents of X_n are added to J to obtain the modified shift count, J_e.

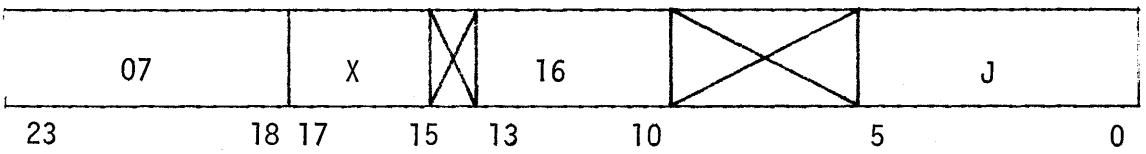
Assembler format: LDS 0,5

DSA DOUBLE SHIFT AROUND

Definition: Shift A and E Around Left

Cycles Required: $2+[(J-9)/5]_{\text{integer}}$

Instruction Format:



Description:

The contents of A and E are shifted left around J_e positions, (A_{23} going to E_0 and E_{23} going to A_0). DSA may use indexing; the contents of X are added to J to obtain the modified shift count, J_e.

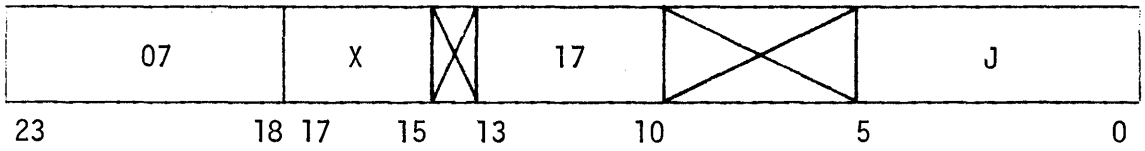
Assembler format: DSA 24

DSL DOUBLE SHIFT LEFT

Definition: Shift A and E Left

Cycles Required: $2 + [(J-9)/5]$ integer

Instruction Format:



Description:

The contents of A and E are shifted left J_e bit positions; E_{23} is shifted into A_0. Zeros are entered into E_0 and the bits shifted out of A_{23} are lost. DSL may use indexing; the contents of X are added to J to obtain the modified shift count, J_e.

Assembler format: DSL 20

1.7.10 Logical Instructions

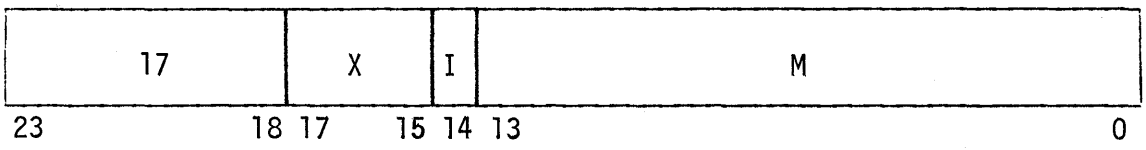
This section contains the following 4 logical instructions; RUM, EOR, AND and OR.

RUM REPLACE UNDER MASK

Definition: $(M_e) \wedge (E) \vee (A) \wedge (\overline{E}) \rightarrow A$ on a bit by bit basis.

Cycles Required: 2

Instruction Format:



Description:

The contents of M_e are masked into A under the control of E. For each "1" bit in E, the corresponding bit in A is replaced by the corresponding bit in M_e. Neither (M_e) nor (E) change.

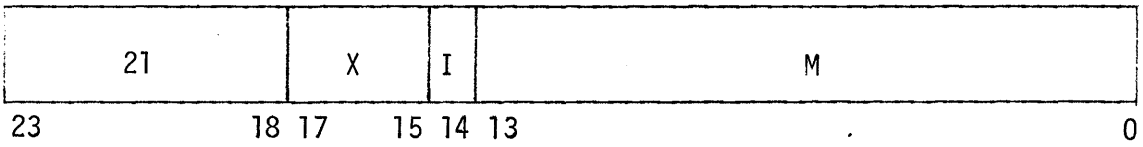
Assembler format: RUM* 2020

EOR EXCLUSIVE OR

Definition: $(A) \forall (M_e) \rightarrow A$

Cycles Required: 2

Opcode Format:



Description:

The contents of M_e are "Exclusively ORed," with the contents of A on a bit by bit basis, and the results stored in A.

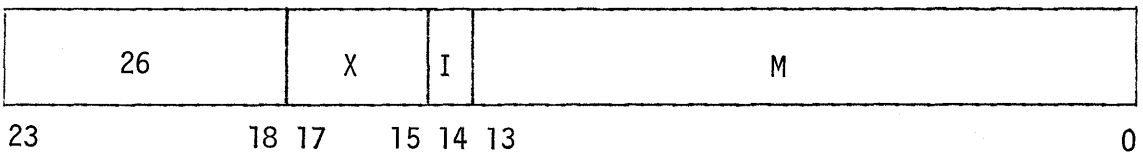
Assembler format: EOR TEMP1+1

AND LOGICAL AND

Definition: $(A) \wedge (M_e) \rightarrow A$

Cycles Required: 2

Instruction Format:



Description:

The contents of M_e and the A register are "ANDed" on a bit by bit basis and the results stored in A.

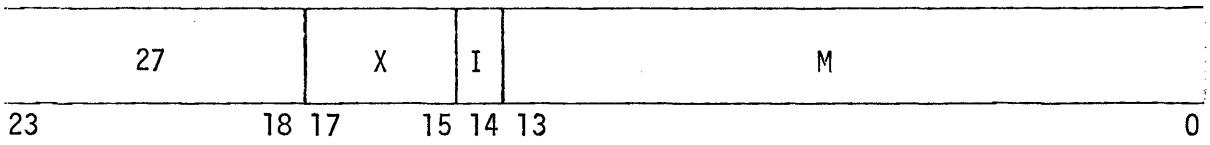
Assembler format: AND TEMP1

OR LOGICAL OR (INCLUSIVE OR)

Definition: (A) (M) → A

Cycles Required: 2

Instruction Format:



Description:

The contents of M_e and the A register are "ORed" on a bit by bit basis and the results stored in A.

Assembler Format: OR* 1000

1.7.11 State Control Instructions

This section consists of 2 instructions: "SET STATE" and "RESET STATE", both of which are augmented instructions. The state flip flops affected by these instructions are defined by C_e, the least significant 10 bits of the instruction, modified by the contents of X. The ten state flip flops which are affected by these instructions are: SW0, SW1, SW2, SW3, SW4, SW5, SW6, and SW7, the interrupt enable flip flop IE, and the overflow indicator OV.

The individual controls for these indicators are the set state and reset state instruction bits, 0_0 through 0_9, respectively.

If the effective address of the set state (or reset state) instruction has a logical one in the least significant bit, bit 0_0, SW0 will be set (or reset) by the instruction.

If 0_0 is a logical zero, SW0 will not be changed. SW1 will be set (reset) if a logical one exists in bit 0_1 of the effective address of the instruction.

Any number of the state flip flops can be set (or reset) with one instruction execution.

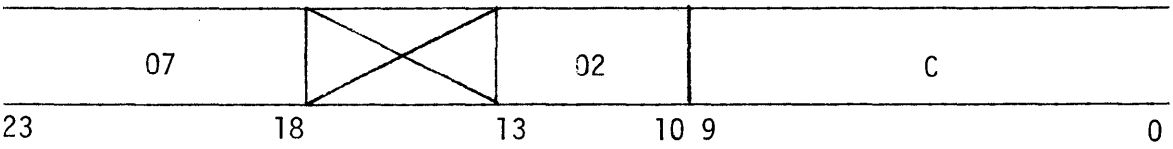
Operand Address Bit	State Flip Flop Affected
0_0	SW0
0_1	SW1
0_2	SW2
0_3	SW3
0_4	SW4
0_5	SW5
0_6	SW6
0_7	SW7
0_8	IE
0_9	OV

SST SET STATE

Definition: Set States Defined by C

Cycles Required: 1

Instruction Format:



Description:

Execution of the SET STATE instruction will cause any of ten state flip flops to be set.

Assembler Format: SST, SW0, SW1, SW4, OV

Note: A special assembler mnemonic exists for setting bit 0_8 (IE). This is IEN for Interrupt Enable.

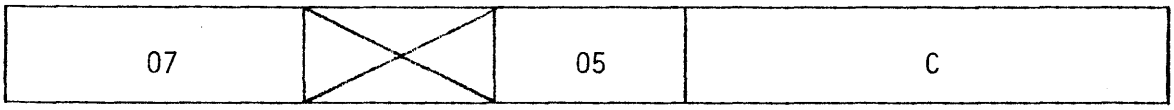
Assembler Format: IEN

RST RESET STATE

Definition: Reset States Defined by C

Cycles Required: 1

Instruction Format:



Description:

The execution of the Reset State instruction will cause the state flip flops addressed to be reset.

Assembler Format: RST, SW2, SW3, IE, OV

Note: A special assembler mnemonic exists for resetting bit 0_8 (IE). This is IDA for Interrupt Disable.

Assembler Format: IDA

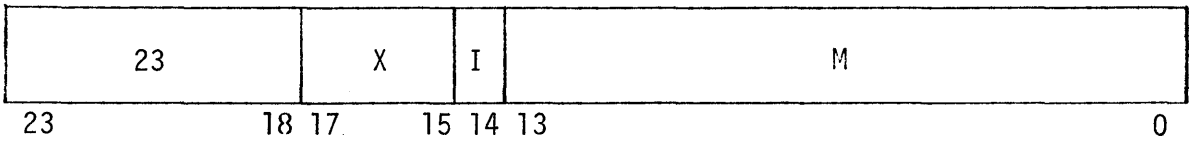
1.7.12 Compare Instruction (1 instruction).

CAM COMPARE A WITH MEMORY

Definition: The contents of A, (A), are compared with the contents of M_e, (M_e). The indicators GT, EQ, LT and BE are set accordingly.

Cycles Required: 2

Instruction Format:



Description:

The contents of A are compared with the contents of memory location M_e. The greater than (GT), equal (EQ), less than (LT), or bit equal (BE) indicators are set in accordance with the outcome of the comparison as described below:

1. If (A) > (M_e) the GT indicator is set,
2. If (A) = (M_e) the EQ indicator is set,
3. If (A) < (M_e) the LT indicator is set,
4. BE is set if a logical one exists in any corresponding bit positions of both A and M_e. For example, if the fifth bit of A is a one, and the fifth bit position of M_e is also a one, BE will be set when the comparison is complete.

The contents of M_e are not changed.

Assembler Format: CAM TEMP5

1.7.13 Transfer of Control Instructions

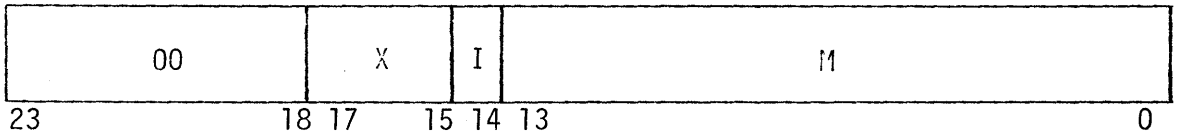
This section consists of 7 instructions which effect transfer of control (or branching). They are: BAH, BRU, BAT, BOI, BOS, BSM and BSZ.

BAH BRANCH AND HALT

Definition: Branch to M_e and Halt

Cycles Required: 1

Instruction Format:



Description:

Program Control is transferred to M_e, after which program execution is halted. The next instruction, which will be executed if the start switch is actuated, is displayed in the command register indicators.

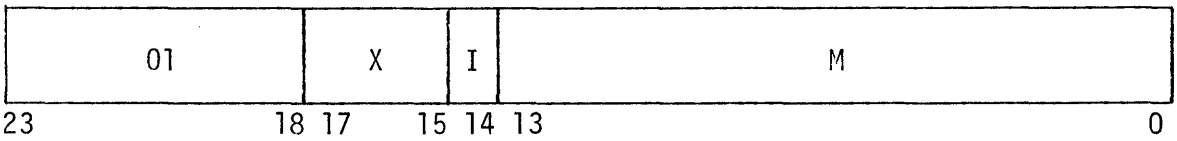
Assembler Format: BAH START2

BRU BRANCH UNCONDITIONALLY

Definition: Branch Unconditionally to M_e

Cycles Required: 1

Instruction Format:



Description:

The BRU instruction will transfer program control unconditionally to M_e. BRU can be indexed and indirect address modified.

An Indirect Address modification of BRU will set the five indicators OV, GT, LT, EQ, and BE from bit positions 23, 22, 21, 20 and 19 of the memory location containing the effective address word. For example, if bit position 23 of the memory location containing the M_e for BRU contains a one-bit, OV will be set during execution of the BRU instruction. Bit 22, containing a one, will cause GT to be set, etc.

An indirect BRU is generally used as a return branch for either a BSM (branch, store return at location M") or a BSZ ("branch store return at location zero") instruction. Note that this restores the five indicators to the states which existed when either a BSM or BSZ instruction was executed.

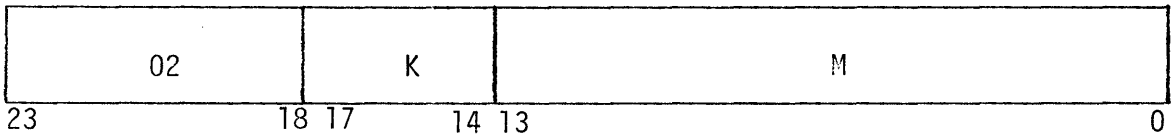
Assembler Format: BRU* START+5

BAT BRANCH A REGISTER TEST

Definition: Branch to M on A Register Test

Cycles Required: 1

Instruction Format:



Description:

The BAT instruction will transfer program control to M, dependent upon the contents of the accumulator. The accumulator contents are tested for positive, zero, negative or odd states.

BAT can neither be indexed nor address modified. The K-field (bits 17-14) specifies the state of A to be tested. If bit 17 is a one, program control will be transferred to M, providing the contents of A are positive (A_{23}=0). If bit 16 is a one, program control will be transferred to M, providing the contents of A are zero, etc. Combinations of states are allowed. For example, if both bits 17 and 14 are ones, program control will be transferred to M if A is positive or if A is odd (A_0=1).

Note: Zero is an exclusive state and is neither positive nor negative.

Assembler Format: BAT K,TEST2

Note: Seven special assembler mnemonics exist to aid the programmer. These are: BP, BPZ, BZ, BNZ, BN, BNEZ and BO for K = 10, 14, 4, 6, 2, 12 and 1, respectively, (i.e. Branch Positive, Positive or Zero, Zero, Negative or Zero, Negative, Not Equal to Zero and Odd, respectively).

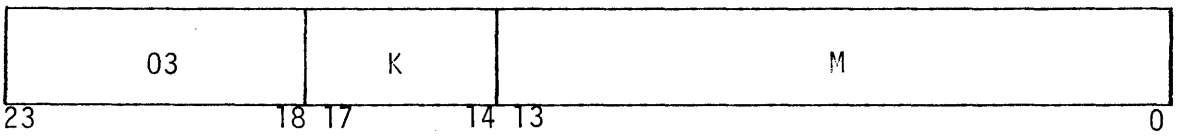
Assembler Format: BPZ TEST2

BOI BRANCH ON INDICATOR

Definition: Branch to M if tested Indicator(s) set

Cycles Required: 1

Instruction Format:



Description:

The BOI instruction will transfer program control to M, dependent upon the state of the four indicators GT, EQ, LT, or BE. BOI can neither be indexed nor indirect address modified. The value, K, is defined by bits 17-14 of the BOI instruction word. Bit 17 tests the state of the GT indicator, while bits 16, 15 and 14, respectively, test the states of the EQ, LT and BE indicators. If one or more of the tests is true, program control will be transferred to M. For example, if bits 17 or 16 are set in the BOI instruction word, then program control will be transferred to M, if either GT or EQ is set.

Assembler Format: BOI K,TEST2

Note: Seven special assembler mnemonics exist to aid the programmer. These are: BG, BGE, BE, BLE, BL, BNE and BBC for K = 10, 14, 4, 6, 2, 12 and 1, respectively, (i.e. Branch Greater, Greater or Equal, Equal, Less Than or Equal, Less Than, Not Equal and Bit Compare, respectively).

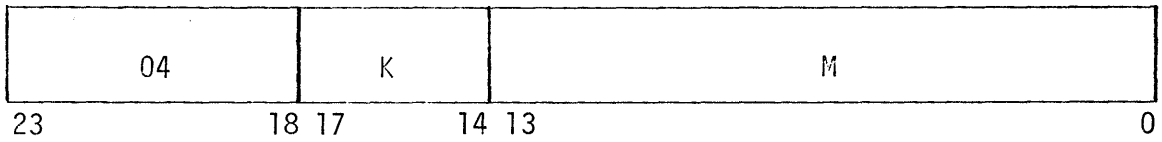
Assembler Format: BGE TEST2

BOS BRANCH ON STATE

Definition: Branch to M if State K Set

Cycles Required: 1

Instruction Format:



Description:

Program control is transferred to M, providing the switch or indicator defined by K is set. BOS can neither be indexed nor indirect address modified. Bits 17, 16, 15 and 14 of the BOS instruction word are decoded into sixteen values of K. The table below defines the appropriate switch or indicator tested for each value of K (expressed octally):

K_8	State Tested	
0	Switch Flip Flop	0
1	Switch Flip Flop	1
2	Switch Flip Flop	2
3	Switch Flip Flop	3
4	Switch Flip Flop	4
5	Switch Flip Flop	5
6	Switch Flip Flop	6
7	Switch Flip Flop	7
10	Interrupt Enable	IE
11	Overflow Indicator	OV (Note: after testing, OV is reset.)
12	Console Switch	CS0
13	Console Switch	CS1
14	Console Switch	CS2
15	Console Switch	CS3
16	Console Switch	CS4
17	Console Switch	CS5

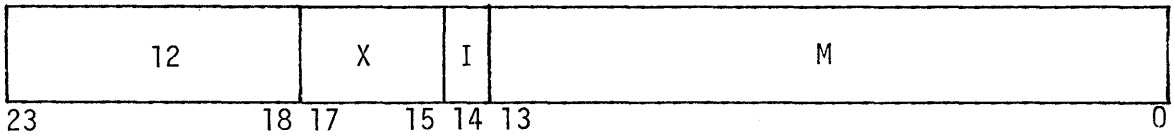
Assembler format: BOS K,ALPHA

BSM BRANCH STORE RETURN AT M

Definition: Branch to $M_e + 1$, Store Return at M_e

Cycles Required: 2

Instruction Format:



Description:

Program control is unconditionally transferred to $M_e + 1$. The contents of the Program Counter, (current program address + 1) are stored in M_e, bits 0 - 13. The states of the five indicators OV, BT, EQ, LT, and BE, are stored in memory location M_e in bit positions 23, 22, 21, 20, and 19, respectively. These states are restored to the indicators when an indirect BRU instruction is used as a subroutine exit (see BRU description).

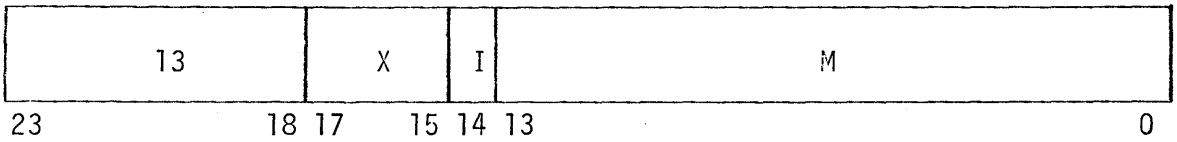
Assembler format: BSM PRTCH

BSZ BRANCH STORE RETURN AT ZERO

Definition: Branch to M_e, Store Return at Memory Location Zero

Cycles Required: 2

Instruction Format:



Description:

Program control is unconditionally transferred to M_e. The contents of the Program Counter, (current program address + 1) are stored in memory location zero, bits 0 - 13.

The states of the five indicators OV, GT, EQ, LT, and BE, are stored in memory location zero, in bit positions 23, 22, 21, 20 and 19, respectively. These states are restored to the indicators when an indirect BRU instruction is used as a subroutine exit (see BRU description).

Assembler format: BSZ* TEST3

1.7.14 Input/Output Instructions

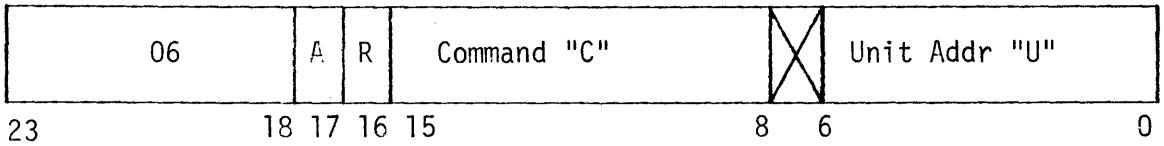
This section consists of 1 multifunction instruction: SPU.

SPU SELECT PERIPHERAL UNIT

Definition: Select Peripheral Unit "U"

Cycles Required: 1

Instruction Format:



The Select Peripheral Unit is a multifunction instruction. These functions are:

(1) the addressing of a peripheral unit for selection;

(2) the transfer of a command to the addressed unit;

(3) the transfer of up to 24 bits of information in either direction between the addressed unit and the CPU's accumulator;

(4) the transfer of the unit's status to the CPU.

"U" defines the unit to be selected by the SPU command. The seven bits in this field allow the selection of up to 128 unique units.

"C" defines the command to the addressed peripheral unit. During the SPU execution, this command field is gated to the peripheral unit, where it is decoded and used to initiate a peripheral operation. (For a description of the commands for each peripheral unit, refer to the sections describing the particular peripheral unit.)

The "A" and "R" bits define a transfer between the addressed peripheral and the CPU accumulator. If the A bit is a "1", there will be an information transfer. If R = 0, the transfer will be from the CPU accumulator to the peripheral unit; if R = 1, the transfer will be from the peripheral unit to the CPU accumulator. If the A bit is a "0", no transfer will occur. During each SPU execution, the addressed peripheral will send status to the CPU. This status is stored in the GT, EQ, LT and BE indicators. Refer to the section devoted to peripheral controllers for interpretation of indicators following an SPU command.

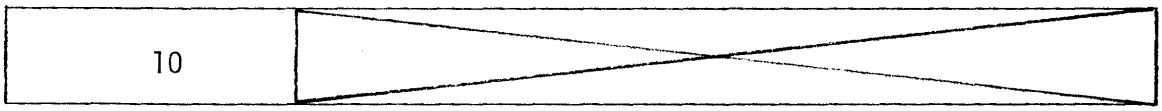
1.7.15 No Operation Instruction

NOP NO OPERATION

Definition: No operation

Cycles Required: 1

Instruction Format:



Description:

No Operation of any kind will occur on the instruction. Its main use is in debugging so as to provide spare instruction slots in assembly programs.

Assembler format: NOP

SECTION II

CONTROL PANEL

2.1 INTRODUCTION

This section describes the manual controls and displays of the FST-1 control panel. A photograph of the control panel is shown in Figure 2.1.1. These controls and displays provide for data entry, single step control and monitoring of the principal components of the FST-1 CPU.

2.2 CONTROL PANEL SWITCHES

2.2.1 Sense Switches

These six switches, located on the upper right corner of the control panel are used to control the response of the Branch On State (BOS) instruction.

2.2.2 Switch Register (W_{23}-W_{00})

These 24 switches, located across the center of the control panel allow manual entry of data to the A Register, the Program Counter, and the Command Register, using the LDA, LDP and LDC switches, respectively, (described below). The setting of these switches may also be loaded into the A register by execution of the Read Switch Register (RSR) instruction.

2.2.3 Register Display Select Switches

These 12 push buttons, located at the left below the switch register, select one of 12 registers for display in the indicator lights immediately above the Switch Register. They are labeled A (accumulator), E (extension), C (command), B (buffer), and X_0 through X_7 (indexes 0 through 7).

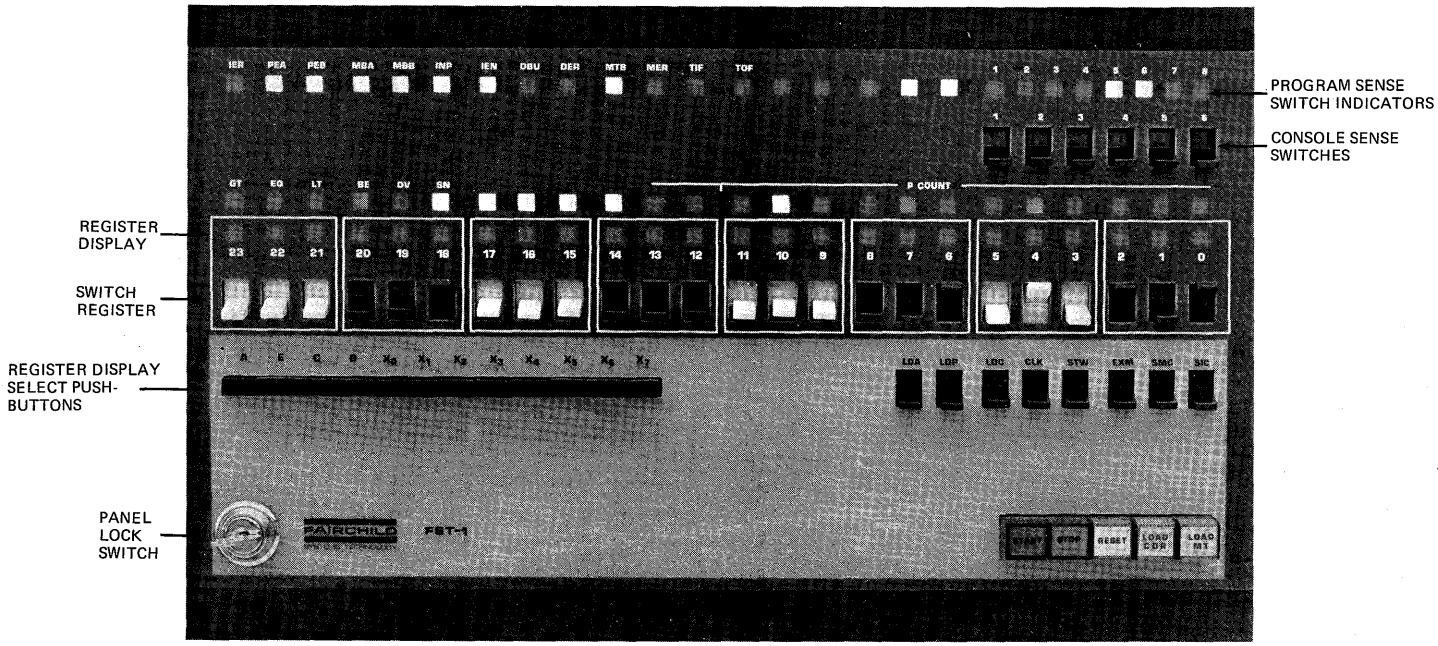
2.2.4 Manual Command Control

The following eight switches, located at the right below the switch register, are enabled only when the keyswitch (lower left corner) is turned clockwise.

2.2.4.1 LDA (Load A)

Loads the A register from the switch register.

Figure 2.1.1. FST-1 Computer Control Panel



2.2.4.2 LDP (Load P)

Loads the P counter from the switch (W_0-W_{14} only).

2.2.4.3 LDC (Load C)

Loads the Command Register from the switch register.

2.2.4.4 CLK (Command Lock)

Locks the current command in the command register. The effective memory address is formed by joining bits 12 and 13 from the command register with bits 0 thru 11 from the Program counter. The P counter advances by one after each execution of the command.

2.2.4.5 STW (Store Switch Register)

Loads the B register from the switch register and initiates a write memory cycle to store the contents of B at the memory location specified by the contents of the Program Counter. The P-counter advances by one for each actuation of the STW switch.

2.2.4.6 EXM (Examine Memory)

Initiates a read memory cycle from the memory location specified by the contents of the Program Counter and loads the read data into the B register. The P-counter advances by one for each actuation of the EXM switch.

2.2.4.7 SMC (Single Memory Cycle)

Causes a Halt of the CPU following each memory cycle.

2.2.4.8 SIC (Single Instruction Cycle)

Causes a Halt of the CPU following the execution of each instruction.

2.2.5. Operational Controls

The following five push buttons, located at the lower right corner of the control panel are the operational controls for the FST-1 CPU.

2.2.5.1 Start

Causes execution to commence with the instruction which occupies the Command Register and to continue with the instruction at the location specified by the contents of the Program Counter.

2.2.5.2 Stop

Causes execution to halt with the next instruction in the Command Register and the address of the next instruction plus one in the Program Counter.

2.2.5.3 Reset

Sets the Program Counter to 00100_8 and prepares the CPU for an instruction fetch. The CPU must be halted for the reset push button to be effective.

2.2.5.4 Load CDR

Causes card reader to read a single card and store the resulting data in 40 consecutive memory locations beginning with address 00100_8.

2.2.5.5 Load MT

Pushing this button causes one record of magnetic tape to be read and stored in 40 consecutive memory locations beginning with address 00100_8.

2.3 CONTROL PANEL DISPLAYS

2.3.1 Status Indicators

Thirteen indicators along the upper left edge indicate system status. They are:

IER	Instruction ERror
PEA	Parity Error A memory bank
PEB	Parity Error B memory bank
MBA	Memory Busy A bank
MBB	Memory Busy B bank
INP	INput Pending
IEN	Interrupt ENable
DBU	Disc BUsy
DER	Disc ERror
MTB	Magnetic Tape Busy
MER	Magnetic Tape ERror
TIF	Time of Instruction Fetch
TOF	Time of Operand Fetch

2.3.2 State Flip-Flop Indicators

At the upper right, the condition of eight state flip-flops is displayed. They are controlled by program using the Set STate (SST) and Reset STate (RST) instructions.

2.3.3 Condition Indicators

Upper center on the left are six condition indicators:

GT	Greater Than
EQ	EQual
LT	Less Than
BE	Bit Equal
OV	OVerflow
SN	Sign

2.3.4 Program Counter

The contents of the Program Counter is displayed at the right center.

2.3.5 Register Indicators

Immediately above the switch register is the selectable register display. Any one of the registers A (accumulator), E (extension), C (command), B (buffer), or X_0 through X_7 (index) may be selected for display by depressing the appropriate pushbutton at the lower left of the control panel. These pushbuttons are mechanically interlocked such that depressing one releases all others.

SECTION III

CONSOLE TYPEWRITER

3.1 INTRODUCTION

A console typewriter is provided as standard equipment with the FST-1. This unit has a keyboard, a printer, and may optionally have a paper tape punch and paper tape reader. The typewriter can be used either off-line or on-line.

The characteristics of the typewriter are given below;

Input/Output Speed	10 characters per second
Code	ASCII
Printable Characters	63
Characters per Line	73

3.2 SYSTEM CONFIGURATION

The typewriter is interfaced to the FST-1 on the Accumulator Bus. The typewriter controller contains two character buffers, one for input and one for output. With both the input and output buffered, input and output data transfers can be performed simultaneously at the maximum mechanism speed.

The typewriter controller utilizes the FST-1 interrupt system. When enabled, either a reader or a printer interrupt is generated at the completion of a reader or printer operation. Either of these interrupts will cause program control to be transferred to an appropriate service program in memory.

3.3 CONSOLE TYPEWRITER CONTROL

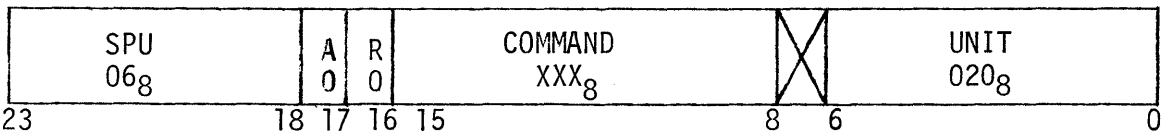
3.3.1 Reader Commands

Seven commands are available for testing and control of the paper tape reader. These commands are variations of the Select Peripheral Unit (SPU) instruction. They differ only in the command code. The unit address remains constant. Each paper tape reader command causes the general status of the controller to be signalled. General status is stored in the CPU indicators GT, EQ, LT, BE, where program tests may be conducted on the status. These indicators convey the following information when set:

GT	Idle
EQ	Idle with error
LT	Busy
BE	Not available

SPU SELECT PERIPHERAL UNIT

Instruction Format:



Note: Commands which the paper tape reader controller is unable to accept are ignored.

STST STATUS TEST

Description: Command Code 000_8

The Status Test command is a null command used to obtain the reader status without changing the condition of the reader controller.

Assembler Format: STST 20B

RDS READ STATUS

Description: Command Code 023_8

The Read Status instruction execution returns controller status to the CPU Accumulator (A_{0-3}), as indicated below:

Bit 0 set: Read Error;
Bit 1 set: Read Interrupt In Process;
Bit 2 set: Read Interrupt Enabled;
Bit 3 set: SPU Command Code Error.

Assembler Format: RDS 20B (Read Status, Reader)

FEED CHARACTER FEED

Description: Command Code 041_8

The Character Feed instruction execution transfers the ASCII code for the character presently in the read station of the paper tape reader to the Input buffer. It also advances the paper tape one character position.

Assembler Format: FEED 20B (Character Feed)

RD READ INPUT BUFFER

Description: Command Code 003_8

The Read Input Buffer instruction execution transfers the contents of the Reader's Character Buffer into the CPU's Accumulator (A_{0-7}), providing:

(a) the Reader Controller is in the Idle or Idle Error State; and
(b) the Reader Buffer is Full.

Assembler Format: RD 20B (Read Input Buffer)

PON READER INTERRUPT ENABLE

Description: Command Code 026_8

The Reader Interrupt Enable instruction sets the reader interrupt control, enabling the Reader Controller to interrupt the CPU at the completion of an operation.

Assembler Format: PON 20B (Priority Interrupt ON)

POFF READER INTERRUPT DISABLE

Description: Command Code 022_8

The Reader Interrupt Disable instruction resets the reader interrupt control, preventing the Reader Controller from interrupting the CPU.

Assembler Format: POFF 20B (Priority Interrupt OFF)

PCOMP READER INTERRUPT COMPLETE

Description: Command Code 002_8

The Reader Interrupt Complete instruction execution resets the Interrupt In Process control in the Reader Controller. Whenever the interrupt system is in use, the instruction must be executed at the termination of the interrupt service routine.

Assembler Format: PCOMP 20B (Priority Interrupt Complete)

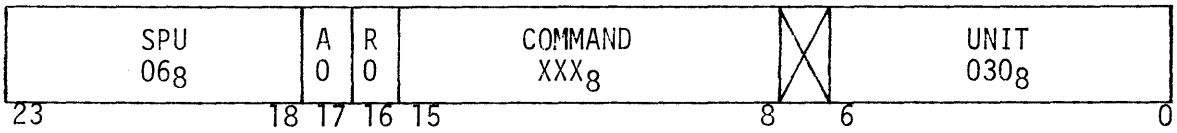
3.3.2 Printer (Punch) Commands

Six commands are provided for the testing and control of the printer and paper tape punch. The paper tape must be engaged manually to respond to these commands. The unit address for the printer (punch) is 030_8; it remains constant for all commands for the printer (punch). Each printer (punch)

command causes the general status of the controller to be signalled to the CPU, where general status is stored in the indicators GT, EQ, LT, BE. Program tests may be performed on these indicators to interpret device status. These indicators convey the following information when set:

GT	Idle
EQ	Idle with error
LT	Busy
BE	Not Available

Instruction Format:



Note: Commands which cannot be accepted by the printer (punch) controller are ignored.

STST STATUS TEST

Description: Command Code 000_8

The Status Test command is a null command used to obtain the printer status without changing the condition of the printer controller.

Assembler Format: STST 30B

RDS READ STATUS

Description: A=1, R=1, Command Code 023_8

The Read Status instruction execution returns controller status to the CPU Accumulator (A_{0-3}), as indicated below:

Bit	0	set: Print Error
Bit	1	set: Printer Interrupt In Process
Bit	2	set: Printer Interrupt Enabled
Bit	3	set: Printer Command Code Error

Assembler Format: RDS 30B (Read Status Printer)

WRIT SELECT AND PRINT

Description: Command Code 007_8

The Select and Print (Write) instruction execution transfers the contents of the CPU Accumulator (A_{0-7}) to the Printer Buffer. The contents of the buffer is then transferred to the teletype, causing a character to be printed, providing:

(a) the Printer Buffer is empty,
(b) the Printer Controller is in the Idle or Idle With Error state.

If turned on, the punch will also record the character.

Assembler Format: WRITE 30B (Select and Print)

PON PRINTER INTERRUPT ENABLE

Description: Command Code 026_8

The Printer Interrupt Enable instruction sets the printer interrupt control, enabling the Printer Controller to interrupt the CPU at the completion of an operation.

Assembler Format: PON 30B (Priority Interrupt On)

POFF PRINTER INTERRUPT DISABLE

Description: Command Code 022_8

The Printer Interrupt Disable instruction resets the printer interrupt control, preventing the Printer Controller from interrupting the CPU.

Assembler Format: POFF 30B (Priority Interrupt Off)

PCOMP PRINTER INTERRUPT COMPLETE

Description: Command Code 002_8

The Printer Interrupt Complete instruction execution resets the Interrupt In Process control in the Printer Controller. Whenever the interrupt system is in use, the instruction must be executed at the termination of the interrupt service routine.

Assembler Format: PCOMP 30B (Priority Interrupt Complete)

SECTION IV

CARD READER

4.1 INTRODUCTION

The FST-1 Card Reader System uses a 300 card per minute mechanism which reads cards serially by columns. This unit is interfaced to the FST-1 system with a controller which provides full word buffering between the card reader and memory.

There are eight commands which provide the programmer with the capability of testing the status of the Card Reader System (CRS), reading cards in either a binary or BCD mode, and for controlling card reader interrupts.

4.2 SYSTEM CONFIGURATION

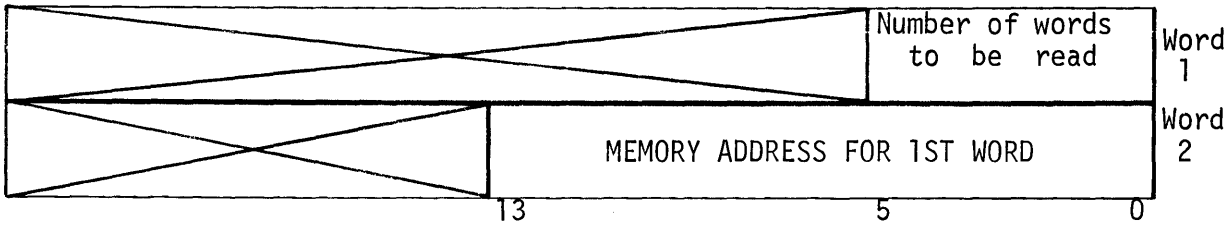
The CRS is interfaced to the FST-1 DMA (Direct Memory Access) System and the Accumulator Bus System. Commands are directed to the CRS with the SPU (Select Peripheral Unit) instruction. The CRS's unit address is 40_8. Data transfers from the CRS to memory are automatically handled by the DMA System. The CRS also uses the DMA System to access Data Control Words (DCW's) stored in memory.

When the CRS Interrupt System is enabled, it will automatically generate an interrupt at the end of a Card Reader operation. This interrupt will cause program control to be transferred to a card reader service program which is stored in memory.

4.3 CARD READER INSTRUCTIONS

4.3.1 Initialization

The CRS (Card Reader System) reads a variable number of words from a card and transfers these words into sequential memory locations using the DMA system. The Programmer must specify both the number of words to be transferred and the memory address where the 1st word read is to be stored. These parameters are passed to the CRS through two words stored in memory. Before initiating a read operation in the CRS, the programmer must store these words in memory. The format is as follows:



Card Reader Data Control Words

The 1st DCW specifies the total number of words to be transferred from the CRS to memory. The 2nd DCW specifies the memory address of the 1st word to be read by the CRS.

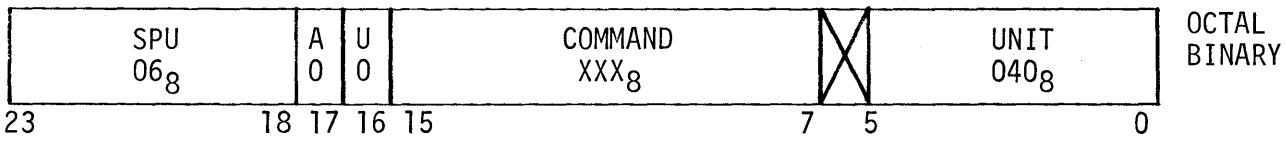
The location of the 1st DCW is transferred to the CRS during the execution of the Read Card instruction. The programmer must load the address of the 1st DCW into the accumulator, prior to the execution of the Read Card command. The accumulator-to-peripheral transfer is specified in the instruction if bit-17 is one and bit-16 is zero.

4.3.2 Card Reader Status Transfers

The card reader returns its General Status to the CPU, as described below in the Status Test instruction description, for all Card Reader SPU instructions except the Error Test instruction. If the card reader cannot accept a command, the command is ignored, but the General Status, which indicates why the command was ignored, is returned to the CPU. After executing a Card Reader instruction, a branch on the indicators GT, EQ, LT, or BE will determine if the SPU command was accepted by the CRC and, if not, why.

4.3.3 Card Reader Instructions, Formats and Descriptions

The eight forms of the SPU instructions recognized by the CRC are illustrated and described below. The instruction format for each follows:



STST STATUS TEST

Description: Command Code 000

The Card Reader Status Test instruction tests the General Status of the CRS without disturbing the state of the controller. The status of the controller is set into the GT, EQ, LT, and BE indicators, with the following meaning:

GT Card Reader is Idle with no Error;
EQ Card Reader is Idle with an Error;
LT Card Reader Busy;
BE Card Reader Not Available.

Assembler Format: STST 40B (Status Test Unit 40)

RD READ CARD BINARY

Description: Command Code 003_8

The Read Card Binary instruction sets the CRS to the read binary mode. The CRS accepts the address of the DCW from the CPU accumulator and reads the number of words specified in the 1st DCW into consecutive memory locations, starting with the memory address specified in the 2nd DCW. In this mode, two card columns are packed into each word. The maximum number of words which can be read per Read Card Binary instruction is 40. Each instruction reads one card.

Assembler Format: RD 40B (Read Unit 40)

ARD READ CARD BCD

Description: Command Code 007_8

The Read Card BCD instruction sets the CRS to the read Binary Coded Decimal mode. The CRS accepts the address of the DCW from the CPU accumulator and reads the number of words specified in the 1st DCW into consecutive memory locations, starting with the memory address specified in the 2nd DCW. In this mode the CRC packs 4 BCD characters into each word which is transferred to memory. The maximum number of words which can be read with this instruction is 20. Each instruction reads one card.

Assembler Format: ARD 40B (Alternate Read Unit 40)

PON CARD READER INTERRUPT ENABLE

Description: Command Code 026_8

The Card Reader Interrupt Enable instruction sets the card reader's interrupt control, allowing the CRC to interrupt the CPU at the termination of the card reader's read operation.

Assembler Format: PON 40B (Priority Interrupt ON)

POFF CARD READER INTERRUPT DISABLE

Description: Command Code 022_8

The Card Reader Interrupt Disable instruction resets the card reader's interrupt control, preventing the CRC from interrupting the CPU.

Assembler Format: POFF 40B (Priority Interrupt OFF)

PCOMP CARD READER INTERRUPT COMPLETE

Description: Command Code 002_8

The Card Reader Interrupt Complete instruction resets the interrupt in process control in the CRC. Whenever the interrupt system is in use, this instruction must be executed at the termination of an interrupt service routine.

Assembler Format: PCOMP 40B (Priority Interrupt Complete)

ETST ERROR TEST

Description: Command Code 020_8

The ERROR TEST instruction will read the CRC error status into the CPU GT, EQ, LT, and BE indicators, with the following meaning:

Indicator State				Meaning
GT	EQ	LT	BE	
0	0	0	0	No Error.
X	X	X	1	Data Control Block Error - Word Count too large/Memory Address out of Range.
X	X	1	X	Data Overflow - CRC cannot obtain access to Memory.
X	1	X	X	Card Reader Device Error - Validity Error, Card Jam, etc.
1	X	X	X	End of file.

"X" indicates that the other conditions may or may not be present. The ERROR TEST instruction neither interrupts nor initiates any CRC operation.

Assembler Format: ETST 40B (Error Test Unit 40)

RDS READ STATUS

Description: A=1, R=1, Command Code 023_8

The Read Status instruction will transfer the Card Reader Status to the CPU accumulator. The error status indicators in the CRS are reset unless the CRS is Busy (LT indicator set after SPU).

The following table shows the meaning of the accumulator bits when they are set to the "1" state by the Read Status Command.

Assembler Format: RDS 40B (Read Status for Unit 40)

Accumulator Bit Position		Meaning when bit set to the "1" state with the Read Status Instruction
23		Read Status Command accepted by CRC. Accumulator contains Valid Status.
22	]	
21	|	
20	|	
19	|	Residual count from the word counter at the time the Read Status Instruction was accepted.
18	|	
17	]	
16	+	End of File
15		Memory protect switch on.
14	+	Feed Error.
13	+	Excess Word Count.
12	+	Device Logic Error.
11	+	Device Read Error
10	+	Invalid Control Code
9		Controller off-line or test switch on.
8	+	Device Card Read Error.
7	+	Data Overflow.
6	+	Memory Address Overflow.
5		DMA Interface Memory Overflow Error.
4		Interrupt Pending.
3		BCD Mode.
2		Interrupt Enabled.
1		Interrupt in Process.
0		Device Ready.

+ Resetable Indicators. These indicators are reset whenever the CR accepts a Read-Card-Binary, Read-Card-BCD, or a Read-Status instruction while not busy.

CARD READER CONTROLLER

CODE CONVERSIONS

OCTAL TRASCII	7-BIT ASCII	ALPHA	HOLLERITH	SYSTEM CHARACTER	029 EQUIVALENT
00	040	20	NO PUNCH	SPACE	
01	041	00	11-8-2	!	
02	042	17	8-7	"	
03	043	13	8-3	#	
04	044	53	11-8-3	$	
05	045	34	0-8-4	%	
06	046	60	12	&	
07	047	15	8-5	'	
10	050	75	12-8-5	(	
11	051	55	11-8-5	)	
12	052	54	11-8-4	*	
13	053	76	12-8-6	+	
14	054	33	0-8-3	,	
15	055	40	11	- (Minus)	
16	056	73	12-8-3	.	
17	057	21	0-1	/	
20	060	12	0	0	
21	061	01	1	1	
22	062	02	2	2	
23	063	03	3	3	
24	064	04	4	4	
25	065	05	5	5	
26	066	06	6	6	
27	067	07	7	7	
30	070	10	8	8	
31	071	11	9	9	
32	072	32	0-8-2	:	0-8-2
33	073	56	11-8-6	;	
34	074	72	12-0	<	12-0
35	075	16	8-6	=	
36	076	52	11-0	>	11-0
37	077	37	0-8-7	?	
40	100	14	8-4	@	
41	101	61	12-1	A	
42	102	62	12-2	B	
43	103	63	12-3	C	

(Continued on next page)

OCTAL TRASCII	7-B ASCII	ALPHA	HOLLERITH	SYSTEM CHARACTER	029 EQUIVALENT
44	104	64	12-4	D	
45	105	65	12-5	E	
46	106	66	12-6	F	
47	107	67	12-7	G	
50	110	70	12-8	H	
51	111	71	12-9	I	
52	112	41	11-1	J	
53	113	42	11-2	K	
54	114	43	11-3	L	
55	115	44	11-4	M	
56	116	45	11-5	N	
57	117	46	11-6	O	
60	120	47	11-7	P	
61	121	50	11-8	Q	
62	122	51	11-9	R	
63	123	22	0-2	S	
64	124	23	0-3	T	
65	125	24	0-4	U	
66	126	25	0-5	V	
67	127	26	0-6	W	
70	130	27	0-7	X	
71	131	30	0-8	Y	
72	132	31	0-9	Z	
73	133	74	12-8-4	[	<
74	134	57	11-8-7	\	¬
75	135	36	0-8-6	]	>
76	136	77	12-8-7	↑	|
77	137	35	0-8-5	←	_

All other hole punch combinations are illegal in the Read Alpha mode and will be flagged by "Validity Error".

Card Reader Controller
Code Conversions
(Continued from previous page)

CARD READER CONTROLLER

CARD TO WORD COUNT

CONVERSION TABLE

This table is provided as an aid for use of the multiple card read capability of the CRC.

Number of Cards	Octal Word Count Binary	Alpha
1	50	24
2	120	50
3	170	74
4	240	120
5	310	144
6	360	170
7	430	214
8	500	240
9	550	264
10	620	310
11	670	334
12	740	360
13	1010	404
14	1060	430
15	1130	454
16	1200	500
17	1250	524
18	1320	550
19	1370	574
20	1440	620
30	2260	1130
40	3100	1440
50	3720	1750
100	7640	3720
200	17500	7640
300	27340	13560
400	37200	17500

COMPUTATION EXAMPLE: NUMBER OF CARDS TO BE READ IN BINARY IS 225_{10}.

OCTAL WORD COUNT IS

	CARDS	WC	
	200	17500	
	20	1440	
	5	310	
DECIMAL SUM	225	21450	OCTAL SUM
	CARD		WORDS

21450_8 IS WORD COUNT

SECTION V

DISC FILE

5.0 DISC FILE

5.1 INTRODUCTION

The Disc File subsystem, Model 2060, uses a high-speed, head-per-track, disc unit. The organization of this system provides the FST Central Processor with a data base of 768,000 words at a transfer rate of 113,000 words per second.

The Disc File format provides 200 useable tracks, 80 segments per track with 48 words per segment. The Disc Controller provides one six bit parity check per segment.

Data access of the Disc/Disc Controller requires a BCD track and segment address word. Track addresses are BCD '000' to '199'. Segment addresses are BCD '00' to '79'.

5.2 SYSTEM CONFIGURATION

The Disc Control Unity (DCU) is interfaced to the FST-1 Central Processor (CPU) through the Common Peripheral Interface (CPI). The CPI provides the DCU with direct access to memory, access to the accumulator bus and required interrupt features.

The DCU provides a six bit character format to the Disc, decomposing and reconstructing 24 bit words. Variable length word files up to the memory size of the FST-1 Central Processor can be handled by the disc subsystem. The DCU is capable of interrupting the Central Processor upon normal completion of commands or upon error terminations.

5.3 DISC SUBSYSTEM INSTRUCTIONS

5.3.1 Initialization

The programmer must specify three data control words for use by the DCU:

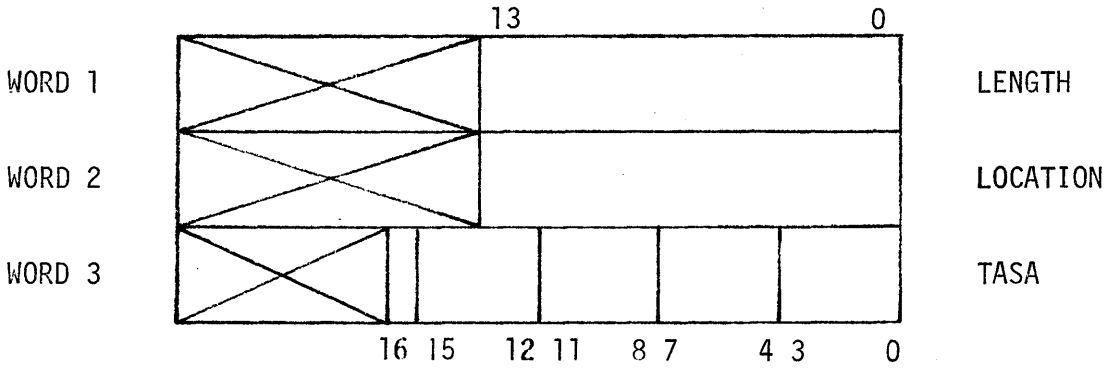


Figure 5.3.1.1

Word 1, File Length: specifies the number of words in the file. This number may be any value up to 16, 383. A value of n=0 will terminate before operation. Values greater than 16,383 will be truncated at 14 bits and used as a valid file length.

Word 2, Location: specifies the memory address of the first word for which the file is a source or destination.

Word 3, TASA: contains the TRACK and SEGMENT address. The segment address is two BCD characters right justified in WORD 3. The range of the segment address is 00 to 79. Values greater than 79 will cause a "DCB ERROR" and will terminate the DCU operation.

The track address is two BCD characters plus one bit. The track address is in bit positions 8 through 16 of WORD 3. The range of the track address is 000 through 199.

Each TASA character is checked and if it exceeds a BCD "9", a "DCB Error" will be noted.

The address of the first data control word must be in the CPU accumulator when a command is issued.

5.3.2 General Status

The Disc Control Unit, DCU, accepts ten commands which provide control for Read, Write, Parity Check, and various status and control functions. The DCU controls the Disc Unit in such a way that the programmer may consider it as a continuous word-string file.

The DCU returns general status through the CPU indicators for all commands with the exception of the ERROR-TEST instruction. The meaning of these conditions is as follows:

GT	Idle
EQ	Idle Error
LT	DCU Busy
BE	Disc Not Ready

Program tests may be performed on the CPU indicators to interpret device general status.

5.3.3 Instruction Format

Instructions for the DCU are Select Peripheral Unit type with a unit code of 70_8. The general format is:

STST STATUS TEST

Description: Command Code 000_8

The Status Test command is a null command used to obtain the DCU status without changing the DCU controller condition.

Assembler Format: STST 70B

ETST ERROR TEST

Description: Command Code 020_8

The Error Test command provides broad error status for the DCU. The ETST command uses the CPU indicators to store error information. This is the only command which does not return general status. The meaning of the indicators is as follows:

GT	Track Address Overflow
EQ	Parity Error
LT	Data Overflow
BE	Data Control Word Error

RDST READ STATUS

Description: Command Code 023_8

Execution of the RDST instruction transfers the contents of the DCU status register to the CPU accumulator, clears the DCU error flip-flops and returns the DCU to the Idle state. General status is stored in the CPU indicators. The meaning of the CPU accumulator bits is summarized in the following table:

Bit	0	Disc Not Ready
	1	DCU Parity Error
	2	DCU Interrupt Control active/enabled
	3	DCU Interrupt not manually inhibited
	4	DCU/CPI Memory Interface not manually inhibited
	5	DCU NORMAL/80 segments/track are addressable
	6	Data Overflow/memory not available when required
	7	Track Address Overflow
	8	Data Control Block Error (TASA or Memory Address out of bounds)
	9	Interrupt operation active
	10	Segment Not Found, Address Search not successful
	11	Disc Write not manually inhibited
	12	Disc Write not inhibited
	13	DCU has one or more error states

Status bit 5 reflects the condition of the manual switch which allows access to the "maintenance" segment (segment 80) on all tracks.

RD READ DISC

Description: Command Code 003_8

The number of words specified by the first data control word is transferred to memory beginning at the location specified by the second data control word. A parity check is made on all segments read. The source of data read is specified by the track and segment address.

Assembler Format: RD 70B

ARD ALTERNATE READ

Description: Command Code 007_8

This command performs the same function as RD except that data transfers to memory are inhibited. Parity is checked on all segments read.

Assembler Format: ARD 70B

PON DCU INTERRUPT ENABLE

Description: Command Code 026_8

This instruction enables the Interrupt Control of the DCU.

Assembler Format: PON 70B

POFF DCU INTERRUPT DISABLE

Description: Command Code 022_8

This instruction resets the DCU Interrupt Control, preventing an interrupt to the CPU.

Assembler Format: POFF 70B

PCOMP DCU INTERRUPT COMPLETE

Description: Command Code 002_8

This instruction resets the Interrupt In Process Control in the DCU/CPI.

Whenever the interrupt system is used, this instruction must be executed at the termination of an interrupt service routine.

Assembler Format: PCOMP 70B

APPENDIX OF DISC COMMANDS

06 00 00 70	TEST NO-OP, STATUS TEST
06 01 00 70	ERROR TEST
06 00 10 70	INTERRUPT ROUTINE COMPLETE
06 01 10 70	INTERRUPT DISABLE
06 01 30 70	INTERRUPT ENABLE
06 61 14 70	READ STATUS
06 61 34 70	READ ALTERNATE STATUS, READ TRACK ADDRESS/ SEGMENT ADDRESS
06 40 14 70	READ DISC BINARY
06 40 34 70	PARITY CHECK - READ DISC ALTERNATE
06 42 14 70	WRITE DISC BINARY

SECTION VI

MAGNETIC TAPE

6.1 INTRODUCTION

The magnetic tape subsystem consists of a tape control unit (TCU) and up to three tape transports. Each transport is connected to the TCU by its own interface cable.

Recording format is 9-track, 800 BPI, NRZI, IBM compatible. Data rate is 6,400, words/sec. Tape speed is 24 IPS (refer to USAS 3.22 for a format description).

6.2 SYSTEM CONFIGURATION

The Magnetic Tape Subsystem uses the Common Peripheral Interface (CPI) for interfacing with the CPU. It receives SPU instructions via the accumulator bus. Data is transferred by direct-memory-access (DMA).

The tape subsystem is assigned the following addresses and interrupt levels:

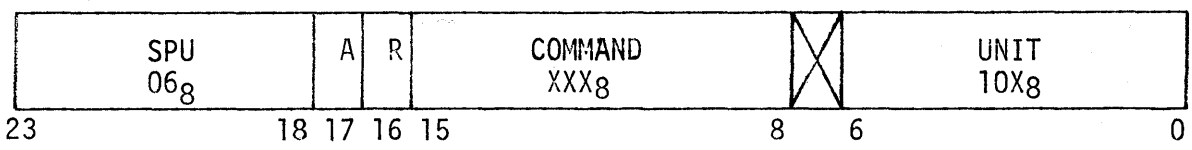
Device code:	$10X_8$ (X=0, 1, 2 for one of up to three tape transports per TCU)
Interrupt address:	10_8
Interrupt priority:	7
Memory priority:	6

6.3 TAPE COMMAND CODES

The Tape Control Unit accepts eighteen forms of the SPU instruction. Table 1 summarizes all commands executed by the TCU.

All commands requiring data transfer operations use a Data Control Block (DCB). Refer to Table 2 for DCB and Data Format.

Instruction Format:

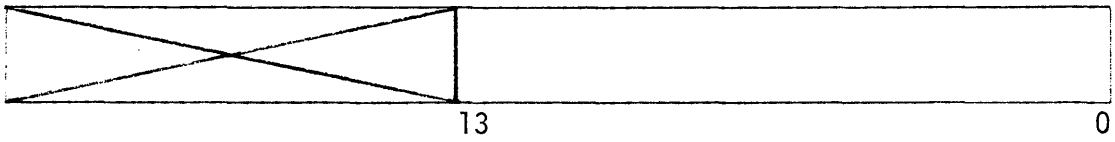


TCU COMMAND CODES

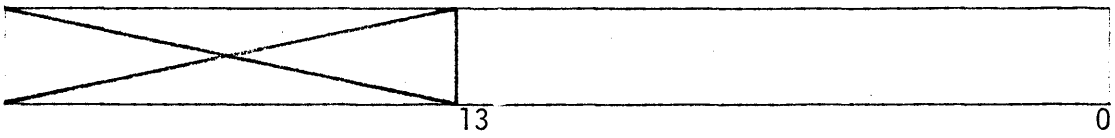
	COMMAND	Octal Code Generated by Assembler	17	16	15	14	13	12	11	10	9	8	7
Write Command	Write record	0642150X	1	0	0	0	1	0	X	X	1	1	X
	Skip and write record	0646150X	1	0	0	1	1	0	X	X	1	1	X
	Write tape mark	0606150X	0	0	0	1	1	0	X	X	1	1	X
Read Command	Read record	0650150X	1	0	1	0	0	0	X	X	1	1	X
	Alternate read record	0640150X	1	0	0	0	0	0	X	X	1	1	X
	Advance one record	0600150X	0	0	0	0	0	0	X	X	1	1	X
	Go back one record	0601150X	0	0	0	0	0	1	X	X	1	1	X
	Advance to tape mark	0604150X	0	0	0	1	0	0	X	X	1	1	X
	Go back to tape mark	0605150X	0	0	0	1	0	1	X	X	1	1	X
Logic Command	Rewind	0600050X	X	X	X	X	X	X	X	X	0	1	0
	Read status	0661150X	1	1	0	0	0	1	0	0	1	1	0
	Read excess word count	0661170X	1	1	0	0	0	1	0	0	1	1	1
	Status test	0600010X	0	0	0	0	0	0	0	0	0	0	0
	Error test	0601010X	0	0	0	0	0	1	0	0	0	0	0
	Enable interrupt	0601310X	0	0	0	0	0	1	0	1	1	0	0
	Disable interrupt	0601110X	0	0	0	0	0	1	0	0	1	0	0
	Interrupt completed	0600110X	0	0	0	0	0	0	0	0	1	0	0

TABLE 1

1. DCB address Word

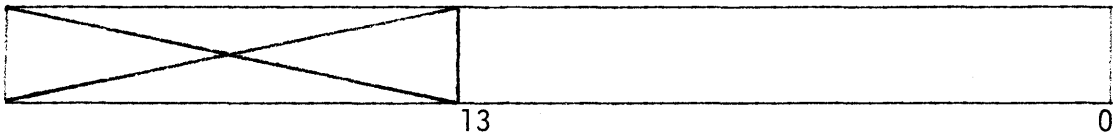


2. Word Count



The word count is stored at "DCB address". If the word count on "Read" is greater than the record length, the complete record will be read and stored in memory. "DCB error" and "Excess word count" (bit 13) will be flagged.

3. Start Address



The start address is stored at "DCB address" = 1. If the start address and word count are such that memory overflow occurs, enough data will be transferred between tape and memory to fill or read memory locations "start address" through "highest memory address". "DCB error" (bit 11) will be set.

4. Data, 9 track

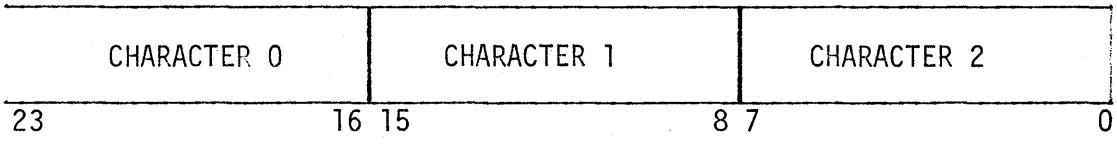


TABLE 2

The TCU, SPU has a device address field of three bits of which only three codes are used:

0 = TRANSPORT-0
1 = TRANSPORT-1
2 = TRANSPORT-2

Note that the TCU decodes bit 7 as a part of the command field.

STST STATUS TEST

Description: Assembler Octal - 0600010X

The Status Test command returns general status of the TCU into the indicators of the CPU.

General Status

BE	Device Not Available
LT	Device Busy
EQ	Device Idle with Error
GT	Device Idle with no Error

All SPU commands return general status except ETST. SPU commands will return status even when the TCU is Busy.

RDS READ STATUS

Description: Assembler Octal - 0661150X

The RDS instruction reads the TCU status register into the accumulator.

Status Register

Bit	Definition
0	Device Ready
1	Interrupt In Process
2	Interrupt Enabled
3	Interrupt Pending
4	Rewinding
5	No Write Enable Ring
6	Memory Protect Switch on
7	BOT
8	Low Density
9	Tape Mark has passed
10	Data Overflow
11	DCB Error
12	Rewind ended

Bit	Definition
13	Word Count > Record Length
14	Word Count < Record Length
15	Longitudinal Parity Error
16	Vertical Parity Error
17	EOT has passed
18	
19	
20	
21	
22	
23	Error Status set

REWC READ EXCESS WORD COUNT

Definition: Assembler Octal 0661170X

The TCU transfers the contents of the memory address register into the accumulator. This contains the last memory location accessed by the TCU.

ETST ERROR TEST

Definition: Assembler Octal 0601010X

The TCU returns error status to the CPU indicators.

	Error Test
BE	DCB Error
LT	Data Overflow
EQ	Parity Error
GT	Write: Write Enable Ring Not Present Read: Word Count Not Equal to Record Length

PON INTERRUPT ENABLE

Definition: Assembler Octal 0601310X

This enables the TCU to request an interrupt of the CPU. An interrupt will then occur after all instructions which require some time to be processed by the TCU.

POFF INTERRUPT DISABLE

Definition: Assembler Octal 0601110X

Disables TCU interrupt request feature.

PCOMP PRIORITY ROUTINE COMPLETE

Definition: Assembler Octal 0600110X

This instruction resets the TCU Interrupt In Process flag in preparation for future interrupt requests.

WRIT WRITE RECORD

Definition: Assembler Octal 0642150X

The TCU loads the word count register and memory-address register from the DCB. It starts the tape transport, and, if the tape is at the BOT marker, writes a 4 inch gap. It reads from consecutive memory locations and writes on tape. The tape is read while being written and the characters checked for parity.

Minimum record length is 12 characters. When the count reaches zero, the TCU writes end-of-record characters (CRC and LRC), puts in a gap and stops the transport.

If enabled, an interrupt will be issued to the CPU on completion of this command.

If the write enable ring is not in the tape reel, the transport is not started and status register bit 23 is set.

If a data overflow occurs, the TCU ends the record, sets bit 10 of the status register, writes a gap and halts.

If memory overflow occurs, the TCU ends the record and sets bit 11 of the status register.

If read-after-write check has a vertical parity (column) error, the TCU sets bit 16 of the status register. The record is completed.

SKWR SKIP AND WRITE RECORD

Definition: Assembler Octal 0646150X

Identical to "Write Record" with the exception that the TCU writes 4 inches of blank tape before it starts writing the record. If enabled, an interrupt will be issued to the CPU on completion of this command.

WRITM WRITE TAPE MARK

Definition: Assembler Octal 0606150X

The TCU starts the transport, writes a 4 inch gap followed by a tape mark (character 310_8) and a normal record gap. The TCU sets status bit 9, stops the transport, and, if enabled, generates an interrupt.

RDT READ RECORD TAPE

Definition: Assembler Octal 0650150X

The TCU loads the word-count register and MAR from the DCB. It starts the tape transport and enables reading after start-up distance. If the tape was at BOT, after two inches of tape travel the TCU reads tape, checks parity, assembles characters into words and writes them into consecutive memory locations. At the end of the record, it checks longitudinal parity, stops the transport, and, if enabled, generates an interrupt.

Minimum record length is 9 characters.

If the record consists of a tape mark, status register bit 9 is set.

If data overflow occurs, the TCU stops transmission to memory and sets bit 10 of the status register. The TCU moves tape to the record gap and halts.

If memory overflow occurs, the TCU stops transmission to memory, sets bit 11 of the status register, moves tape to the record gap and halts.

If a parity error occurs, the TCU replaces the character in error with all ones, sets bit 16 of the status register and continues.

If the word count is less than the record length, the TCU stops transmission to memory, sets bit 14 of the status register and continues for parity check.

If the record is less than the word count, the TCU sets bit 13 of the status register, halts tape, and, if the character count MODULO 3 was non-zero, the last word is padded with all ones.

ART ALTERNATE READ RECORD TAPE

Definition: Assembler Octal 0640150X

Identical to "Read Record" except that data is not transmitted to memory. The word count is not checked. If enabled, an interrupt will be issued to the CPU on completion of this command.

RSKIPF READ, SKIP FORWARD, ONE RECORD

Definition: Assembler Octal 0600150X

The TCU moves tape in the forward direction, advances to the end of the record and stops in the record gap. If enabled, an interrupt will be issued to the CPU on completion of this command.

RSKIPB READ, SKIP BACKWARD, ONE RECORD

Definition: Assembler Octal 0601150X

The TCU moves tape in the reverse direction to the beginning of record and halts in the record gap. If enabled, an interrupt will be issued to the CPU on completion of this command.

FSKIPF FILE SKIP FORWARD

Definition: Assembler Octal 064150X

The TCU moves tape forward until a tape mark record or End of Tape (EOT) is detected. The tape is stopped after the tape mark or EOT.

If a tape mark is detected, bit 9 of the status register is set.

If EOT is detected, bit 17 of the status register is set. If enabled, an interrupt will be issued to the CPU on completion of this command.

FSKIPB FILE SKIP BACKWARD

Definition: Assembler Octal 0605150X

The TCU moves tape backwards until a tape mark record or Beginning of Tape (BOT) is detected.

The tape is stopped before the tape mark. If a tape mark is detected, bit 9 of the status register is detected.

If BOT is detected, bit 7 of the status register will be true.

If enabled, an interrupt will be issued to the CPU on completion of this command.

REWIND REWIND TAPE

Definition: Assembler Octal 0600050X

The TCU initiates a rewind operation on the selected tape drive. During a rewind operation, the TCU is busy for six memory cycles.

When the selected drive detects a BOT marker the tape will stop. The TCU will set bit 12 of the status register, REWIND END.

If enabled, an interrupt will be issued to the CPU on completion of this command.

APPENDIX A

FST-1 ABBREVIATIONS

ABS	Accumulator Bus System
AIS	Accumulator Interface System
AR	(Memory) Address Register
A-Register (A)	Accumulator (24 bits)
ARL	Access Request Line
AU	Arithmetic Unit
BE	Bit Equal Indicator
B-Register (B)	Memory Buffer Register (24 vits)
CP	Card Punch
CR	Card Reader
CR	Command Register (24 bits)
CRC	Card Reader Controller
CRS	Card Reader System
DCB	Data Control Block
DCW	Data Control Word
DMA	Direct Memory Access
EQ	'Equal' Indicator
E-Register (E)	Extension Register (24 bits)
FCS	Fairchild Computer System
GT	'Greater Than' Indicator
I	Indirect Address Indicator
IAM	Indirect Address Modification
IE	Interrupt Enable Flip-Flop
IP	Interrupt Priority
IR	Interrupt Required Flip-Flop
J	Shift Constant
K	Value Substituted for I and X on Conditional Branches
LT	'Less Than' Indicator
M	Memory Location
MAG	Memory Access Gained Flip-Flop
MAR	Memory Address Register

APPENDIX A
FST-1 ABBREVIATIONS (CONT'D)

MIS	Memory Interface System
O	Operand Address
OC	Instruction Operation Code
OV	'Overflow' Indicator
P-Register (P)	Program Counter (14 bits)
R-Register (R)	Interrupt Address Director Register (6 bits)
S_N	Nth Output of the AU
TOL	Tape Object Loader
Ua	"A" Input to AU
Ub	"B" input to AU
W-Register (W)	Console Switch Register (24 bits)
X	Index Address
X-Register	Index Registers X0----X7 (14 bits)

APPENDIX B

FST-1 INSTRUCTION MNEMONICS

		Reference Page Number
ADD	Add	1-12
AND	Logical And	1-28
AOM	Add One to Memory	1-15
ATX	Add to Index	1-21
AUG	Augmented Instruction	1-3
BAH	Branch and Halt	1-34
BAT	Branch on A-Register Test	1-36
BBC	Branch Bit Compare	1-37
BE	Branch if Equal	1-37
BG	Branch if Greater	1-37
BGE	Branch if Greater or Equal	1-37
BL	Branch if Less Than	1-37
BLE	Branch if Less Than or Equal	1-37
BN	Branch if Negative	1-36
BNE	Branch Not Equal	1-37
BNEZ	Branch if Not Equal to Zero	1-36
BNZ	Branch if Negative or Zero	1-36
BO	Branch if Odd	1-36
BOI	Branch on Indicator	1-37
BOS	Branch on State	1-38
BP	Branch if Positive	1-36
BPZ	Branch if Positive or Zero	1-36
BRU	Branch Unconditional	1-35
BSM	Branch Store Return at M	1-39
BSZ	Branch Store Return at Zero	1-40
BZ	Branch if Zero	1-36
CAM	Compare A to Memory	1-33
DADD	Double Add	1-13
DIV	Divide	1-14
DLD	Double Load	1-19
DSA	Double Shift Around	1-26
DSL	Double Shift Left	1-27
DSN	Double Shift Normalize Left Shift A and E	1-23
DSR	Double Shift Right	1-25

FST-1 INSTRUCTION MNEMONICS (CONT'D)

		Reference Page Number
DST	Double Store	1-19
DSUB	Double Subtract	1-13
DTC	Double Two's Complement	1-11
EOR	Exclusive Or	1-28
EXC	Exchange A and E	1-16
IEN	Interrupt Enable	1-31
IDA	Interrupt Disable	1-32
LAX	Load A from X	1-22
LDA	Load A	1-18
LDE	Load E	1-18
LDS	Logical Double Shift	1-26
LDX	Load Index	1-20
LS	Logical Shift	1-24
LXA	Load Index from A	1-20
MUL	Multiply	1-14
NOP	No Operation	1-42
OR	Inclusive Or	1-29
RSR	Read Switch Register	1-16
RST	Reset State	1-32
RUM	Replace Under Mask	1-27
SA	Shift Around	1-24
SL	Shift Left	1-25
SOM	Subtract One from Memory	1-15
SPU	Select Peripheral Unit	1-41
SR	Shift Right	1-23
SST	Set State	1-31
STA	Store A	1-17
STE	Store E	1-17
STX	Store Index	1-21
SUB	Subtract	1-12
TCA	Two's Complement	1-11

COMMENT SHEET

FROM NAME: _____

BUSINESS
ADDRESS: _____

_____

_____

BUSINESS
PHONE: _____

COMMENTS: (Please describe errors, suggested additions or deletions and reference page number, paragraph number, or drawing number.)

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FORM 700622

FOLD ON DOTTED LINES AND STAPLE

Cut Along Dotted Line

Fold Along Dotted Line

FIRST CLASS
PERMIT NO. 329
Sunnyvale,
California

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

Postage Will Be Paid By

FAIRCHILD SYSTEMS TECHNOLOGY
974 East Arques Avenue
Sunnyvale, California 94086

ATTEN: SYSTEMS TECHNICAL PUBLICATION DEPT.

Fold Along Dotted Line

Cut Along Dotted Line

STAPLE

STAPLE