IBM

Field Engineering
Theory of Operation

Restricted Distribution

This manual is intended for internal use only and may not be used by other than IBM personnel without IBM's written permission.

2025 Processing Unit

Y24-3527-0

RESTRICTED DISTRIBUTION: This publication is intended for internal use only and may not be used by other than IBM personnel without IBM's written permission.

First Edition (October 1968)

Significant changes or additions to the specifications contained in this publication are continually being made. Any such changes will be reported in subsequent revisions or FE Supplements.

This publication has been prepared by the IBM Systems Development Division, Product Publications, Dept. 171, P.O. Box 6, Endicott, N.Y. 13760.
A form has been provided at the back of this publication for readers' comments. If the form has been removed, comments may be addressed the same as above.

© Copyright International Business Machines Corporation 1968

BACKGROUND AND LEVEL OF READER

This manual is intended for use by IBM personnel who have a knowledge of System/360 operation.

PREREQUISITE MANUALS

The reader should be familiar with the following SRL publications.

- IBM System/360 Principles of Operation, Form A22-6821
- IBM System/360 Model 25 Functional Characteristics, Form A24-3510
- IBM System/360 Model 25, 1401/1460 and 1440 Compatibility Features, Form A24-3512.

COMPANION MANUAL

The following companion manual is necessary to study the IBM 2025.

- IBM 2025 Processing Unit, FE Maintenance Diagrams Manual, Form Y24-3529.

Diagrams in the FE Maintenance Diagrams Manual are referenced using the initials MDM followed by the diagram number; e.g., MDM x-xx.

SYSTEM MANUALS

The following manuals are necessary for overall study of the Model 25 System.

- IBM 2025 Processing Unit, System/360 Channels FETOM, Form Y24-3531.
- IBM 2025 Processing Unit, Integrated 2540 Attachment FETOM, Form Y24-3532.
- IBM 2025 Processing Unit, Integrated 1403 Attachment FETOM, Form Y24-3533.
- IBM 2025 Processing Unit, Integrated 2311 Attachment FETOM, Form Y24-3534.
- IBM 2025 Processing Unit, FE Maintenance Manual, Form Y24-3528.

FEATURES OF MANUAL

Material is arranged in the order of learning.

Contents

Lists the major headings of the manual.

Abbreviations

Alphabetic list of abbreviations, acronyms, and symbols used in the manual.

Introduction

High-level presentation of the major points needed to understand the system.

Functional Units

Describes parts of the machine that always act in much the same manner.

Principles of Operation

Contains the concepts and information needed to understand the microlistings.

Features

Describes special features, including 1400 Compatibility.

Power Supplies and Controls

Describes power sequencing, power distribution, etc.

Console and Maintenance Features

Describes all console keys, lights, and switches. The theory of operation for the Console Printer-Keyboard is included in this section, as well as the alter/display and logout facilities.

Appendixes

List speed and capacity of system units. Provide World Trade information.

Index

Detailed, cross-referenced, alphabetic listing of headings, key words, phrases, operations, etc.

PREFACE. iii

Background and Level of Reader iii

Prerequisite Manuals iii

Companion Manual iii

System Manuals iii

Features of Manual iii

CONTENTS v

ABBREVIATIONS. ix

CHAPTER 1. IBM SYSTEM/360 MODEL 25 . . . 1-1

System Control 1-1

Data Flow. 1-1
Main Storage. 1-3
Program Storage. 1-3
Control Storage. 1-3
Auxiliary Storage. 1-4
M-Register and M-Register Assembler . 1-4
Storage Address Assembler 1-5
W-Register. 1-5
C-Register. 1-5
Local Storage 1-5
Local-Storage Data Assembler . . . 1-5
Local-Storage Address Assembler. . 1-6
A/B Register Assembler. 1-6
Arithmetic and Logical Control Circuits 1-6
Integrated I/O and Channels 1-6
External Facilities 1-6

CHAPTER 2. FUNCTIONAL UNITS. 2-1

CPU Clock. 2-1
Clock Control 2-1

Core Storage 2-1
Core Array (8K Halfwords) 2-3
Common Sense/Inhibit. 2-5
Storage Clocks. 2-5
Storage Addressing. 2-5
Current Sources 2-7
X- and Y-Drive Gate and Selection . . 2-8
X-Decode Drive 2-8
Y-Decode Drive (Main Storage). . . 2-8
XY-Decode Drive (Auxiliary Storage). 2-9
Storage Data Assembler and Register . 2-9
Power Supply and Temperature Compensation 2-9
M-Register Assembler. 2-11
M-Register. 2-11
W-Register. 2-11
Address Match Circuits. 2-12
Traps and Priority. 2-12
Traps. 2-12
Priority 2-12

External Facilities. 2-14
MODE Register 2-17
MMSK Register 2-17
S-Register. 2-18
BC - External Facility. 2-20
Dynamic Condition (DYN) Register. . . 2-20

Control Register 2-21

Arithmetic and Logical Control Circuits. 2-22
AB-Register Assembler 2-23
A-Register 2-23
B-Register 2-23
ALU 2-24
A-Register Gating. 2-24
B-Register Gating. 2-24
True/Complement Controls 2-24
Adder. 2-24
ALU Controls 2-25
Carry Insert 2-25
Decimal Corrector. 2-26
ALU Check. 2-26
ALU Modifier 2-26

Local Storage (Figure 2-15). 2-28
Zone 0 2-30
Zone 1 2-30
Zone 4 2-30
Zone 5 2-31
Zone 6 2-31
Zone 7 2-31
Local Storage Address Assembler . . . 2-31
Local Storage Data Assembler. 2-32

CPU Checks 2-33
Hard-Stop Latch. 2-34
Machine-Check Latch. 2-34

CHAPTER 3. THEORY OF OPERATION 3-1

Microprogram Concepts. 3-1
Definitions 3-1
Microprogram Functions. 3-3
Microlistings 3-3
Microprogram, Machine Language Tie-In. 3-3
Control Storage Load (CSL). 3-5
CSL Using Integrated 2540. 3-5
CSL Using a Reader on the Standard Interface. 3-5
CSL Using Integrated 2560. 3-5
Set/Reset Word. 3-5
Bit Significance Fields (Figure 3-2). 3-5
Description. 3-5
Examples 3-6
Arithmetic Constant Word. 3-15

Bit Significance Fields (Figure 3-6). 3-15
Examples 3-17
Storage Word. 3-17
Bit Significance Fields (Figure 3-7). 3-19
Description. 3-20
Examples 3-20
Move/Arithmetic Word. 3-23
Bit Significance Fields (Figure 3-8). 3-23
Examples 3-23
Branch Unconditional Word 3-24
Bit Significance Fields (Figure 3-9). 3-25
Description. 3-25
Examples 3-25
Branch on Mask Word 3-26
Bit Significant Fields (Figure 3-10) 3-26
Description. 3-26
Examples 3-29
Branch on Condition Word. 3-30
Bit Significance Fields (Figure 3-12) 3-30
Description. 3-33
Examples 3-33

CHAPTER 4. FEATURES. 4-1

System/360 Model 25 1401/1460 and 1440 Compatibility Features. 4-1

Implementation 4-1

Compatibility Initialization 4-2

Op-Code Conversion and Recognition . . . 4-2

Character Configuration. 4-4

1401/1460 and 1440 Program Addressing. . 4-4

Address Conversion 4-5
Address Error Detection 4-7

Auxiliary Storage. 4-7
Auxiliary Storage Module 0. 4-10
Auxiliary Storage Module 1. 4-12
Auxiliary Storage Module 2. 4-12
Auxiliary Storage Module 3. 4-13
Auxiliary Storage Module 4. 4-13
Auxiliary Storage Module 5. 4-13
Auxiliary Storage Module 6 (Module 8 in 24K Systems). 4-13
Auxiliary Storage Module 7 (Module 9 in 24K Systems). 4-13

Local Storage. 4-13
Zone 0. 4-13
Zone 1. 4-14
Zone 4. 4-14
Zone 6. 4-14
Zone 7. 4-14

Operation. 4-14

CPU Operations 4-15

I-Cycles. 4-15
I-Cycles Objectives. 4-16
I8 Objectives. 4-16
I7 Objectives. 4-16
I5 Objectives. 4-16
I4 Objectives. 4-17
I2 Objectives. 4-17
I1 Objectives. 4-17
Indexing. 4-17

I/O Operations 4-17
1402 Read Operations. 4-18
Microroutine Objectives. 4-19
1402 Punch Operations 4-19
Microroutine Objectives. 4-19
1403 Printer Operations 4-20
Microroutine Objectives. 4-21
1443 Printer Operations 4-21
Microroutine Objectives. 4-21
1442 Card Read-Punch Operations . . . 4-22
Microroutine Objectives; Read Operations. 4-22
Microroutine Objectives, Punch Operations. 4-22
Disk Compatibility Operation. 4-23
Formats. 4-23
File Circuitry 4-24
Operational Objective 4-25
Seek Command 4-25
Read/Write Operations With Indelible Address 4-25
Wordmarks and Zone Bits in Control Fields. 4-26
Read/Write Operations, Sector Mode. 4-26
Alternate Track Operation. 4-28
Magnetic Tape Operations. 4-28
Microroutine Objectives. 4-29

Storage Protection 4-30
Storage Key 4-32
Protection Key. 4-34
Protection Exception. 4-34
Setting Up Storage Protection 4-35
Set Storage Key (Machine Language Instruction) 4-37
Insert Storage Key (Machine Language Instruction) 4-37
Displaying A Storage Protect Key. . . 4-37

Functional Units (Refer to Figure 4-13) 4-38
STP1 Map. 4-38

Theory of Operation (MDM 4-90) 4-38
STP1 Addressing Structure (Figure 4-21). 4-38
Loading STP1. 4-38
Set Storage Key Microword Cycle (Figure 4-24) 4-41
Loading Q- and FQ-Registers 4-43
Loading GQ- and HQ-Registers (Figure 4-29). 4-44
Main Storage Access (Figure 4-29) . . 4-44
Zero Test (Protection Key 0). 4-44
Parity Check (Figure 4-30). 4-44
Match Circuitry (STP0=STP1) 4-44
Insert Storage Key (CSTS Routine) . . 4-49

Insert Storage Key (Microword Statement) 4-49

Interval Timer 4-50
60-Cycle Operation. 4-51
50-Cycle Operation. 4-52

Direct Control Feature 4-52
Direct Control Bus-Out. 4-52
Timing Signal Bus-Out 4-52
Direct Control Bus-In 4-52
External Signal Bus-In. 4-53
Writeout. 4-54
Readout 4-54
Hold-In 4-54
Read-In 4-54
Application 4-54
Situation A. 4-55
Situation B. 4-55

External Interrupt 4-55

CHAPTER 5. POWER SUPPLIES AND CONTROL. . 5-1

Power-On Sequence. 5-2

Power-Off Sequence 5-2
Power-Off Key 5-2
Emergency Power Off (EPO) 5-3
Cvercurrent, Overvoltage Detection. . 5-4
Voltage Sensing System. 5-4
Thermal Sensing 5-5

Power Distribution 5-5

Marginal Checking. 5-5

CHAPTER 6. CONSOLE AND MAINTENANCE FEATURES. 6-1

System Control Panel 6-1

Indicators or Displays 6-1
Byte-0 and Byte-1 Indicators. 6-1
CPU Status Indicators 6-1
System Checks 6-3
CPU Checks. 6-4

Operator Control Section 6-5
Switches and Keys 6-5
Indicators. 6-6

Operator Intervention Section. 6-6

Customer Engineering Controls. 6-8
Mode Switch 6-8
Operating Mode Positions (MDM 4-4). 6-8
Single-Cycle Mode Positions. . . . 6-9
Alter/Display Functions. 6-9
Check Control Switch. 6-10
Diagnostic Control Switch (Figure 6-1) 6-10
Scan and Load Positions. 6-10
Metering Switch 6-11
Console Procedures. 6-12
Manual Display of Core Storage . . 6-12
Manual Store Operation (Core Storage). 6-12
Alter/Display Local Storage. . . . 6-13
ALU/External Display 6-13

Console Printer-Keyboard 6-13

Comprehensive Introduction 6-13
Input Data Code 6-13
Output Data Code--EBCDIC. 6-14
Tilt/Rotate Printing Code 6-16
Data Flow 6-18
Controls-Out (TA). 6-18
Controls-In (TU and TT). 6-18
Diagnostic Registers (TR and TD) . 6-18
Input Data (TI). 6-18
Output Data (TE) 6-18
Share Request Control. 6-18

Functional Units 6-18
Printer Keyboard. 6-18
PR-KB Auxiliary Storage Areas 6-19
Alter/Display Message Area. 6-19
Sense Byte (PR-KB). 6-19
PR-KB Unit Control Word (UCW) 6-19
Flags/Op Byte. 6-19
Channel Status Byte. 6-19
Unit Status Byte 6-21
PR-KB Translate Table (Keyboard Code to EBCDIC). 6-22
Translate Subroutine (Keyboard Code to EBCDIC) 6-22
Tilt/Rotate Translator. 6-23
External Fields--CPU to PR-KB 6-25
TA--Controls Out 6-25
TE--Data Out 6-26
External Fields--PR-KB to CPU 6-26
TI--Data In. 6-26
TU--Branch Conditions. 6-27
TT Branch Conditions 6-28
TR--Tilt/Rotate Diagnostic Register. 6-28
TD -- PR-KB Diagnostic Register (Figure 6-16) 6-29

Console Printer-Keyboard Theory of Operation 6-29
Request Key--Attention Status 6-29
I/O Instructions Routine. 6-30
Start I/O with Prior Status to Clear. 6-30
Start I/O--Printer Keyboard Not Busy 6-30
Sense Command 6-31
Intervention Required 6-31
No-Operation Command. 6-31
Read Command. 6-31
Initial Selection--Read Command. . 6-34
Keyboard Shift 6-35
Write Command 6-36
Initial Selection--Write Command and Write with ACR Command. . . . 6-38
PR-KB Request Routine 6-39
Active Bit = 0, Secondary Bit = 0 6-39
Active Bit = 0, Secondary Bit = 1 6-39

Active Bit = 1, Secondary Bit = 1 6-40
Active Bit = 1, Secondary Bit = 0 6-40
Read Share Request. 6-40
Read Operation Ending Procedure. 6-41
CR/LF Ending Routine 6-41
Function Cycle 6-41
Write Share Request. 6-42
Write Operation Ending Procedure 6-42
Case Shift During Write Operation 6-42
Common Interrupt Routine 6-43
Test I/O Command (PR-KB). 6-44
Active Bit = 1, Secondary Bit = 0 6-44
Active Bit = 1, Secondary Bit = 1 6-44
Active Bit = 0, Secondary Bit = 1 6-44
Active Bit = 0, Secondary Bit = 0 6-44
Halt I/O Command (PR-KB). 6-45
Active Bit = 0, Secondary Bit = 0 6-45
Active Bit = 1, Secondary Bit = 0 6-45
Secondary Bit = 1, CC Flag = 1 . . 6-45
Secondary Bit = 1, CC Flag = 0, Active Bit = 0. 6-45
Secondary Bit = 1, CC Flag = 0, Active Bit = 1. 6-45

Keys and Indicators (PR-KB). 6-45
Indicators (Figure 6-25). 6-45
Intervention Required (INTVTN REQD) 6-45
Proceed (PROCD). 6-45
Attention (ATTN) 6-46
Alter/Display Indicator on CPU Console (ALTER DPLY). 6-46
Keys (Figure 6-25). 6-47
Request. 6-47
Ready. 6-47
Not Ready. 6-47
Alternate Coding (ALTN CODING) . . 6-47
End-of-Block (EOB) 6-47
Cancel 6-47
PR-KB Alter/Display (on CPU Console). 6-47

PR-KB Alter/Display Facility 6-47

Description and Operation. 6-47
Setup 6-47
Addressing 6-48
Execution of Alter and Display . . 6-48
End Operation 6-48
Message Formats 6-49
Alter Storage. 6-49
Display Storage. 6-49
Errors. 6-49

PR-KB Alter/Display Microprogram Routine 6-50

Logout 6-50

Machine Check Logout 6-50

Channel Control Check Logout 6-50

Interface Control Check Logout 6-50
GS Channel External Conditions . . 6-51
GT Channel External Conditions . . 6-51
GD Channel External Conditions . . 6-51
Code Byte. 6-51

APPENDIX A. MACHINE CHARACTERISTICS. . . A-1
Power and Cooling Requirements . . A-1
Overall Dimensions and Weights . . A-1

APPENDIX B. SPECIAL CIRCUITS B-1

APPENDIX C. WORLD TRADE. C-1

INDEX. X-1

Note: This list does not include microprogramming mnemonics, which are defined in the sections on microprogramming in Chapter 3 or in the section where they are used.

ACR	Automatic Carrier Return
Addr, Adr	Address
ALU	Arithmetic and Logic Unit
AS	Auxiliary Storage
Asm, Assm	Assembler
Attn	Attention
Aux	Auxiliary
BCD	Binary Coded Decimal
bpi	bits per inch
BSM	Basic Storage Module
CAW	Channel Address Word
CC	Command Chain
CCW	Channel Command Word
CD	Chain Data
Chan	Channel
Char	Character
Chk	Check
Chnl	Channel
CKD	Count, Key, and Data
Comm	Communications
CPU	Central Processing Unit
CR/LF	Carriage Return/Line Feed
CSL	Control Storage Load
CSW	Channel Status Word
Ctr	Counter
Ctrl	Control
DAC	Disk Attachment Control
DCF	Disk Control Field
Diag	Diagnostic
Dply, Dsply	Display
EBCDIC	Extended Binary-Coded-Decimal Interchange Code
EOB	End of Block
EPO	Emergency Power Off
Ext	External
Hex	Hexadecimal
I/A	Indelible Address
I.B.	Interrupt Buffer
IC	Instruction Counter
IL	Incorrect Length
I/O	Input/Output
IPL	Initial Program Load
Insn	Instruction
Int	Interrupt
Intlk	Interlock
Intvtn	Intervention
KB	Keyboard
LC	Lowercase
LCW	Line Control Word
LS	Local Storage
Man	Manual
MAS	Microprogram Automation System
MDM	Maintenance Diagram Manual
Mem	Memory
Mod	Modifier
Mplx, MPX	Multiplexer
MS	Main Storage
ms	millisecond
No-Op	No Operation
ns	nanosecond
Op	Operation
Pch	Punch
PCI	Program Controlled Interrupt
PFR	Punch Feed Read
PLB	Print Line Buffer
PLBAR	Print Line Buffer Address Register
Prgm	Program
PR-KB	Printer-Keyboard
Procd	Proceed
Prot	Protect
Prt	Printer
PSW	Program Status Word
Pwr	Power
RBC	Read Back Check
Rdr	Reader
Reg	Register
Reqd	Required
Rst	Reset
RTH	Return to Home
Rtn	Return, Routine
SAR	Storage Address Register
SCR	Silicon Controlled Rectifier
SDB	Storage Data Bus
SDBO	Storage Data Bus-Out
Seq	Sequential
Serv	Service
SLI	Suppress Length Indication
SLT	Solid Logic Technology
SRP	Serial Reader Punch
Stor	Storage
STP	Storage Protect
Str	Straight
Sw	Switch
Sys	System
Temp	Temperature
T/R	Tilt/Rotate
UC	Uppercase
UCW	Unit Control Word
WLR	Wrong Length Record

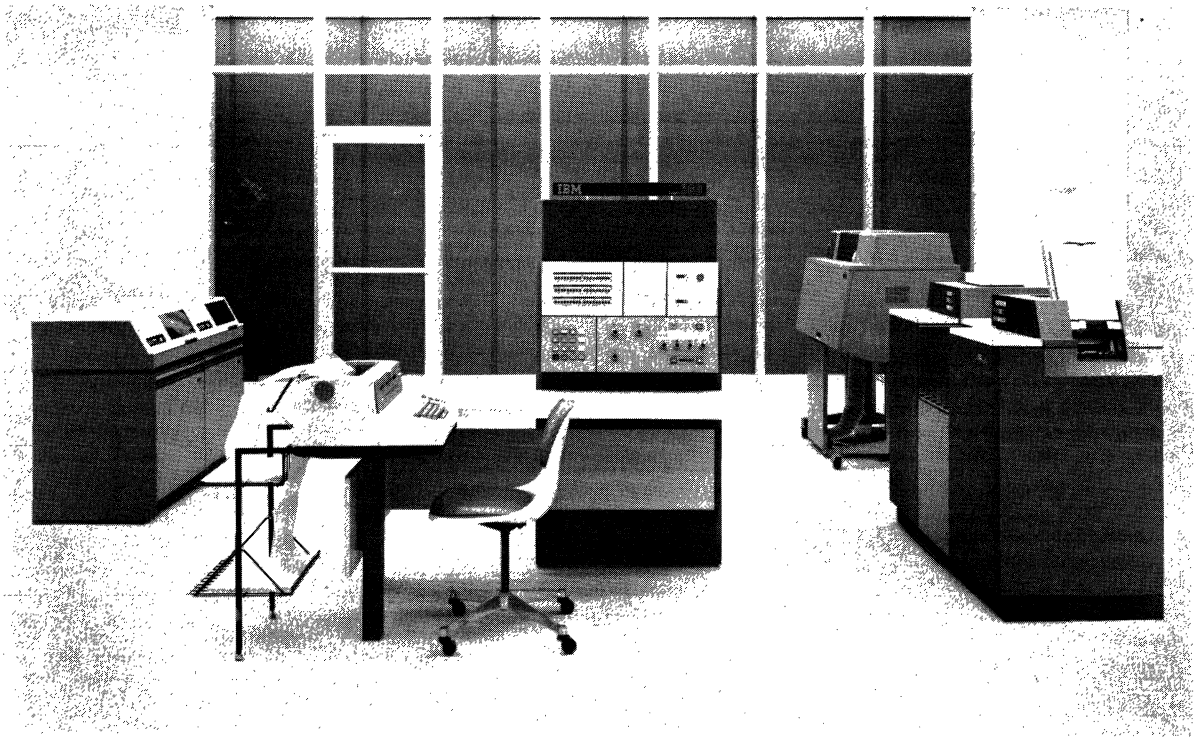


Figure 1-1. System/360 Model 25

The Model 25 (Figure 1-1) is a solid-state medium-speed data processing system. It includes the integrated attachment of a console printer-keyboard as a required feature, as well as the 2540 reader/punch, the 1403 printer (Model 2 or 7) and the 2311 disk drive. Two channel options also are available using the System/360 standard interface to attach additional I/O equipment.

The 2025 processing unit uses a core storage unit for program, control, and auxiliary storage and a 64-byte monolithic storage unit for local storage. The remaining major sections of the processor are the arithmetic and logic unit (ALU) and the clock and control circuits. Each of these units is discussed in detail under Data Flow and in Chapter 2.

SYSTEM CONTROL

- System control is provided by circuitry that decodes and interprets control words.
- The control words are contained in the control storage area of main storage.

Overall control of the Model 25 system is by microprogram routines. The microprogram routines used for system control are a series of steps referred to as control words. How these control-word routines are constructed and placed into control storage is explained in Chapter 3.

The microprogram routines, however, provide the control words necessary for the system to perform its basic functions. These functions include:

1. Entering machine-language instructions and information into the system through an input device.
2. Processing the information.
3. Formatting the information on an output device to produce a meaningful record.

Further control of these basic functions must be provided for the processing to result in useful information. This is achieved through the interaction of the microprograms and the machine language instructions. The machine language instructions define the operations to be performed while the microprograms actually perform the operation. For example, a simple machine language program function might be to read a card, add field 1 to field 2, and print the results. These three instructions to the processing unit are executed by a series of microprogram steps (control words).

In this example, several different areas of the system are involved:
1. The card reader that puts the data into the system
2. The storage areas that hold the data and the instructions during execution
3. The ALU, where the fields are added
4. The printer that performs the output functions.

Figure 1-2 shows how some of these areas are tied together to form a system.

DATA FLOW

- Control word decode determines actual data flow.

Actual data flow in the Model 25, for any given operation, is determined by the bit configuration of the control word being executed. A single control word is decoded to provide the gating and control necessary to perform a specific function. This could be a program storage access, an arithmetic operation, a movement of data to or from local storage, or in some cases, a combination of these.

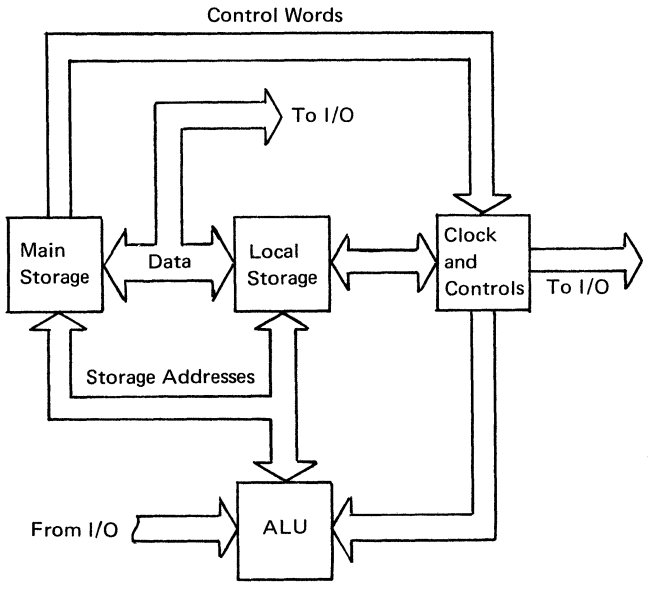


Figure 1-2. System Data Flow

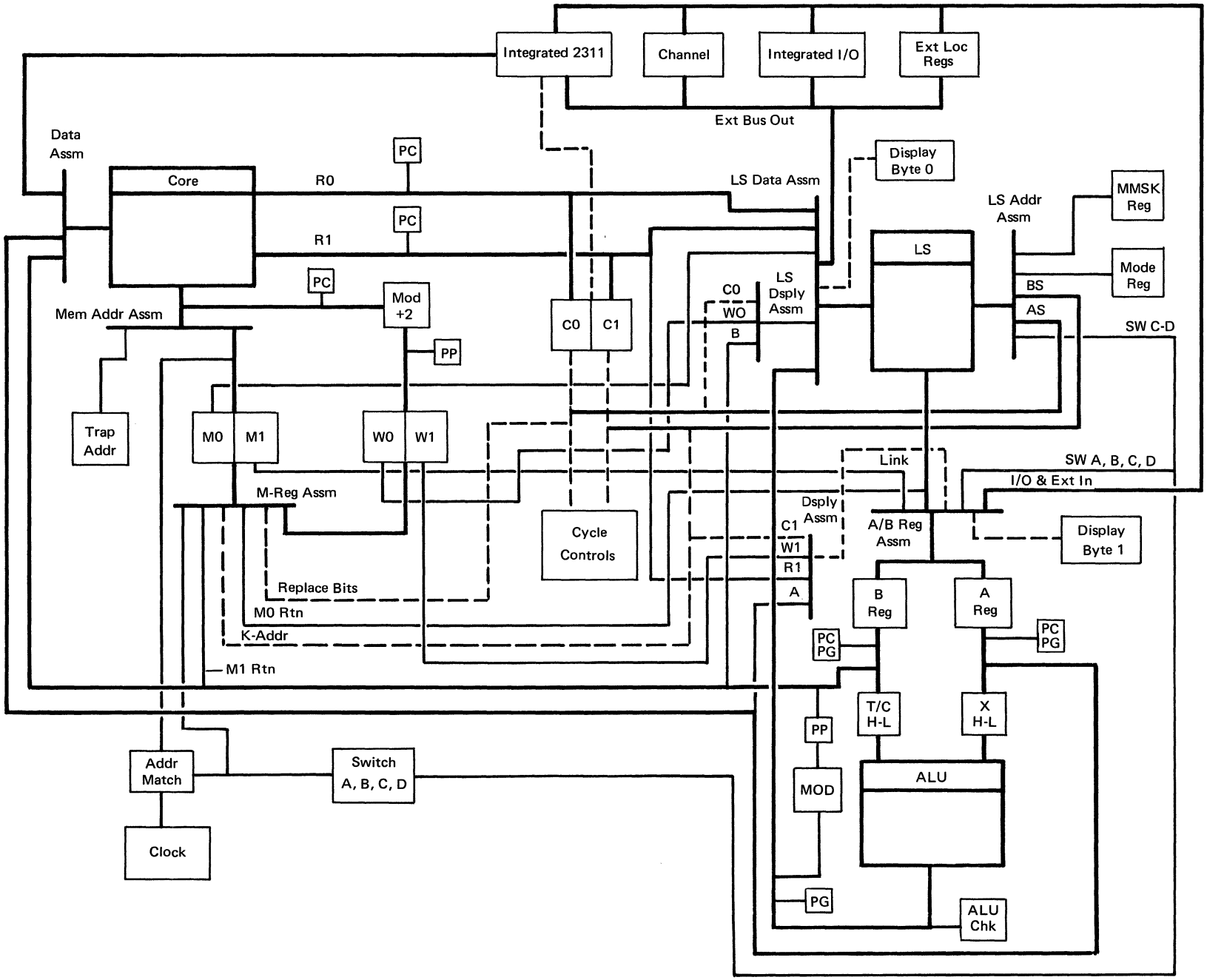


Figure 1-3. Model 25 Data Flow

Control words in the 2025 are one halfword in length (16 information bits and 2 parity bits). The words are arranged in sequence to form a microprogram routine. The routines are assembled from symbolic input statements written by a microprogrammer, and are loaded into the control-storage area of main storage prior to the execution of any machine language instruction. This control-storage load is ordinarily done when the system is installed, or when changing modes of operation (between System/360 and 1400 emulator modes, for example). A breakdown of the control words and a description of the control-storage load routine can be found in Chapter 3.

Figure 1-3 is a simplified data-flow diagram for the Model 25. The description of the functions performed by elements in the data flow is of a general nature because of the dependency on the control word decode for actual data patns.

MAIN STORAGE

- Model 25 uses an 18-bit 900-ns storage unit.
- Storage sizes are 16K, 24K, 32K, and 48K.

The main-storage array used by the Model 25 for program, control, and auxiliary storage is the 18-bit 900-ns M2-I. Three array sizes (16K, 24K, and 32K) are used in combination to provide the total main storage requirements of the system (Figure 1-4).

Main storage is divided into three distinct areas:

Program storage
Control storage
Auxiliary storage

Program Storage

This area of main storage contains the user's programs and is used for storing machine language instructions and data. Certain locations in program storage (hex addresses 0000 through 0087) are permanently assigned to ensure program compatibility within System/360, as described in the IBM System/360 Principles of Operation, Form A22-6821. The remaining locations of program storage can be used as desired.

Control Storage

This main storage area contains the microprogram routines necessary for control of system operations. Model 25 has 16,384 bytes of control storage. These locations are always the highest 16K block of byte addresses of the installed array(s).

System Size	Array Size	Program Storage		Control Storage		Auxiliary Storage	
		Bytes	Address Range Decimal/Hex	Bytes	Address Range Decimal/Hex	Bytes	Address Range Hex
16K	One 32K Composite Array (Note 1)	16,384	0-16,383 0000-3FFF	16,384	16,384-32,767 4000-7FFF	2,048	0X00-7XFF
24K	One 24K Array for Program Storage One 16K Array for Control Storage	24,576	0-24,575 0000-5FFF	16,384	32,768-49,151 8000-BFFF	2,560	0X00-5XFF & 8X00-BXFF
32K	One 32K Array for Program Storage One 16K Array for Control Storage	32,768	0-32,767 0000-7FFF	16,384	32,768-49,151 8000-BFFF	3,072	0X00-BXFF
48K	One 32K Array for Program Storage One 32K Composite Array (Note 1)	49,152	0-49,151 0000-BFFF	16,384	49,152-65,535 C000-FFFF	4,096	0X00-BXFF

Notes: 1. Program storage is always located in the low-order area of the composite core-storage array(s). For example, program storage for the 48K system is all of the low order 32K unit plus the low-order 16K bytes of the high-order 32K unit.

2. Control storage is always the high-order 16K bytes of the composite core-storage array(s).

Figure 1-4. Model 25 Core-Storage Allocations and Addressing Scheme

Auxiliary Storage

There are 1,024 bytes of auxiliary storage associated with each main storage array of 16K byte locations. (256 bytes are associated with each 1000 hexadecimal addresses.) Model 25 has a maximum of 4096 bytes in a 48K system (Figure 1-4).

This storage area is used for the sixteen general purpose registers, four floating-point registers, multiplexer channel UCW storage, I/O translation tables, multiply and divide tables, and other microprogramming requirements.

As with other System/360 models, only limited areas of auxiliary storage (general and floating-point registers) are normally accessed by the user's program. Other defined locations are used only by the microprogram routines. Undefined areas are available for expanding microprogram functions.

Information read from main storage is placed on the storage data-out lines (via the storage data register) and enters the LS data assembler or the control register (C-register). Control words are entered into the C-register and decoded at its output to bring up the gating and branching lines necessary to perform the functions indicated by the control word.

Data and information from main storage is sent to the LS data assembler. Entry into local storage or an external facility from the assembler is controlled by the decode of the control word being executed.

M-REGISTER AND M-REGISTER ASSEMBLER

Main storage is addressed from the M-register, a 16-bit register in the processing unit. The M-register is composed of two single-byte registers, M0 and M1, which are fed from the M-register assembler where the actual addresses are constructed.

Control-storage addresses require a slight change in the addressing scheme because of the location of control storage for the different program storage sizes. Regardless of storage size, control storage is the upper 16K block of byte addresses.

For branch unconditional, branch and link, and return control words, the two high-order bits of M0 are forced to a specific pattern, corresponding to the

total storage size, to address control storage.

Auxiliary storage addressing is similar to program storage addressing except that an additional signal (USE AUX STORE) is generated. Control-word bit structure is used to generate this signal.

STORAGE ADDRESS ASSEMBLER

The output of the M-register feeds the storage address assembler. This assembler, in turn, feeds the main storage addressing circuits. The assembled address is modified and then placed in the W-register. The updated address in the W-register provides a means of stepping to the address of the next sequential control word.

The storage address assembler is used also for address substitution in case of a trap (a forced address to execute a control-word routine other than the next sequential word). The contents of the assembler are replaced with the starting address of the trap routine, and the M-register is left unchanged. This allows the next control-word address to be saved. The trap routine stores the contents of the M-register so that the correct control-word sequence can be resumed when the trap routine is completed.

The forced trap address is modified +2 and placed into the W-register from the assembler. Normal control-word stepping occurs during execution of the trap routine.

W-REGISTER

The W-register is a 16-position register that holds an updated (+2) address of the control word presently being executed. It retains this address during program or auxiliary storage access cycles.

The output of the storage address assembler and M1 register bit 7 are presented to a modifier circuit for address modification of +2 and stored in the W-register. The modified output is then presented to the M-register assembler as the address of the next sequential control word, if the next sequential word is to be executed. In some branching cases (see Chapter 3), only part of the W-register output is gated to the M-register.

C-REGISTER

The C-register is a 16-bit register that is fed from the main storage data-out lines. It is used to hold the control word being executed. The C-register output decode circuitry interprets the control-word bits and conditions the gates, lines, and data paths that determine the operation of the CPU as required for that particular control word.

Cycle Controls: These circuits include the system clock, elements that decode the control words to direct the flow of information through the system, and elements that provide data paths for the information.

LOCAL STORAGE

- Local storage is provided by the Model B150 high-speed (180-nanosecond) monolithic stack.
- The unit contains sixty-four 1-byte locations on a single SLT card.
- Readout is non-destructive.

Local storage as used in the Model 25 is separated from main storage, and is composed of sixty-four 9-bit (including parity) bytes. (A tenth bit for each byte is available but not used.) These locations are used for intermediate storage for factors to be operated on by the Arithmetic and Logical Unit (ALU): the operation bytes, instruction counter, register operand, program mask, backup locations, data addresses for the channel and integrated I/O devices, and work areas.

Local-storage locations are loaded by control words so that required storage addresses and problem or microprogram factors are made available without readdressing main storage.

Data stored in local storage is used by control words to perform some problem-program function.

Data enters local storage from the LS data assembler and exits on the LS bus-out lines. Output is to the A/B register assembler or the M-register assembler, depending upon the control word being executed.

During the execution of microprogram (control program) steps, local storage can accept or read out a byte of information as many as four times within the primary system cycle of 900 nanoseconds.

Local-Storage Data Assembler

The LS data assembler is a group of gating circuits that can receive a byte of data from several sources. Among the sources are the storage data bus lines, the

M0-register, the ALU modifier, the W0-register, the B-register, and the Z-bus. The control word being executed determines which of these is used. (W or B inputs are used for manual display purposes.)

The output of the LS data assembler is presented to local storage or to external facilities.

Local-Storage Address Assembler

Local storage is addressed through the LS address assembler. The outputs of this assembler are the X- and Y-drive lines for local storage. Local-storage addresses are usually made up from the outputs of the C-register, combined with the MODE register. However, when a trap occurs, the MMSK register overrides the MODE register and combines with the AS- or BS-field to form the LS address. The switches on the console also can be used to address local storage.

A/B REGISTER ASSEMBLER

The A/B register assembler is a group of gating circuits that can receive data bytes from local storage, external facilities, and a number of other sources. Data selection and gating for the assembler is determined by the control word decode or the function being performed. The output of the assembler is to the A-register and B-register.

ARITHMETIC AND LOGICAL CONTROL CIRCUITS

All arithmetic and logical data manipulations are performed by these circuits. The A-register and B-register provide the input to the ALU; however, the control word decode determines the manner in which the data is presented and the operation performed.

Refer to Chapters 2 and 3 for further information concerning ALU operation for the various control words.

INTEGRATED I/O AND CHANNELS

The integrated attachments for the Model 25 are:

IBM 1052 Model 7 Console Printer-Keyboard (PR-KB)
IBM 1403 Printer Model 2 or 7
IBM 2311 Disk Storage Drive Model 1 (up to four)
IBM 2540 Card Read-Punch Model 1.

These units, together with the 2025 processing unit, make up a compact and versatile data processing system.

The PR-KB, 1403, and 2540 are permanently assigned to channel 0. The 2311s are permanently assigned to channel 1. In addition, the Model 25 can have a System/360 standard interface optional feature that may be either a byte-mode (multiplexer) channel assigned as channel 0 or a burst-mode (selector) channel assigned as channel 1. Circuitry for these I/O attachments and the channel feature is contained in the central processing unit.

EXTERNAL FACILITIES

The external facilities (registers and lines) are source and destination locations that are not within local storage. That includes all locations external to local storage that can be addressed by a control word. This does not include the A, B, C, M, or W registers, or any main storage location.

Some of the more prominent externals are the S-register, MMSK register, BC-Facility, DYN register, and the MODE register. Each of these is discussed in detail in Chapter 2.

CPU CLOCK

- The basic timing pulses for the IBM 2025 (except core storage) are generated by the CPU clock.
- A crystal oscillator drives a five-stage latch ring.

Five latches are connected to form an overlapped latch ring for creating the basic clock pulses (MDM 4-11). A free-running crystal oscillator provides the pulses that drive the latch ring circuit.

The latch ring is reset with the clock-5 latch on. For the clock to be started, the clock-start latch must be turned on. This allows clock-1 latch to turn on with the next oscillator pulse. The following (not) oscillator pulse ANDed with clock 1 turns clock-5 latch off. The next oscillator pulse ANDed with clock 1 and (not) clock 5 turns clock 2 on. This sequence continues through the latch ring. The result is five overlapping clock pulses that develop ten overlapping (T0 through T9) timing pulses (MDM 4-11). The complete clock cycle takes 900 nanoseconds.

The 180-nanosecond T0-through-T9 pulses are used to develop 90-nanosecond P-pulses (P1, P3 through P8). The P pulses are developed near the logic area that uses the actual pulse; this minimizes ringing and noise.

CLOCK CONTROL

- The clock operates as long as the clock-start latch is on or the inhibit clock stop line is active.
- The clock always stops with only clock-5 latch on.

For the clock to be started, the start interlock latch must be turned on (MDM 4-11). Once the start interlock latch turns on, the next (not) oscillator pulse turns the clock start latch on. This conditions the clock-1 latch to turn on with the following oscillator pulse. Once the clock start latch is set, the clock operates until the clock start latch is reset.

The clock start latch is reset by:

1. the machine reset switch line being active
2. the hard stop latch being on
3. the clock-stop pulse line being active.

The machine reset switch line is active during the power-on sequence or when the system reset, load, or control storage load key is pressed. The hard-stop latch is set by check control switch at the check-stop position and any error occurring, or by an incorrect checksum being recognized. The clock-stop pulse is activated by several conditions as shown on MDM 4-11.

When the machine reset switch line is active, the entire clock and the control latches are reset immediately. This turns the clock-5 latch and the clock-off latch on.

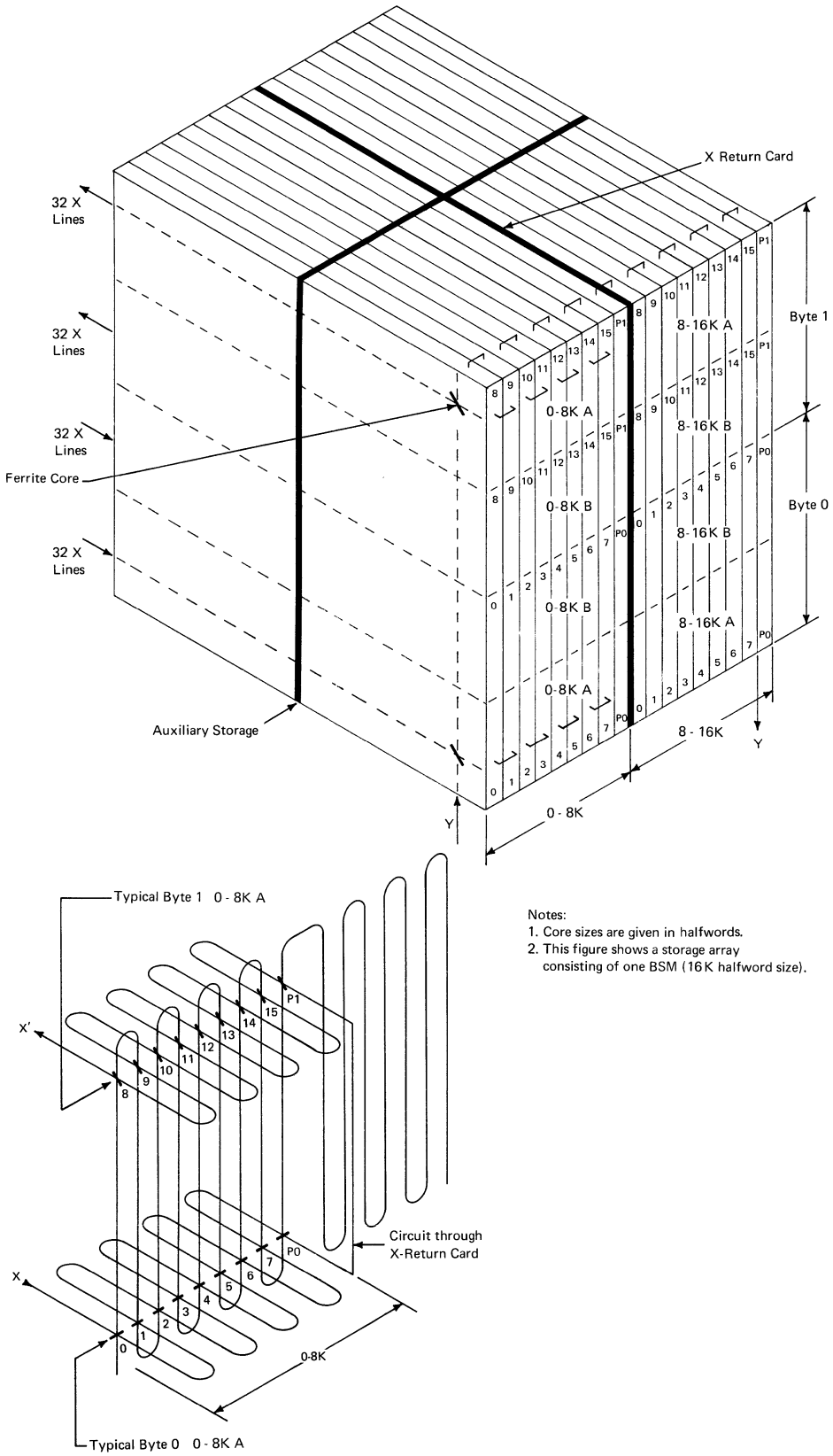
The clock-off latch is turned off the first time the clock-1 latch is turned on after the clock-start latch is set. When the clock is conditioned to stop at the end of a cycle, the clock-off latch turns on. This blocks the T8 and T9 lines from being active with the clock stopped (clock-5 latch on). Also the line clock-off is activated, which conditions many of the manual operations.

CORE STORAGE

The M2I Basic Storage Module (BSM) operating at 900 nanoseconds is used in the 2025. Depending on the amount of core storage installed, one or two BSMs are used. The BSM(s) is a separate unit installed within the 2025 and connected to the 2025 control and data lines via a core storage interface cable. The BSM is available in three sizes: 8K, 12K, or 16K halfwords. (The size of the BSM also depends on total storage capacity of the system.)

Note: In this publication, the 16K BSM is used to describe the storage theory, and all references to BSM are for 16K.

Unless otherwise indicated, all reference to core size is by addressable halfwords, both in this description and in the ALDs. Whenever core storage is read out, two bytes (18 bits: 16 data plus 2 parity) are read out and two bytes are written back. A read call cycle is divided into two portions: read and write. A write always occurs after a read in the 2025. Refer to MDM 3-3 for the overall data flow of the core storage unit and the associated 2025 functional units.



Notes:
1. Core sizes are given in halfwords.
2. This figure shows a storage array consisting of one BSM (16K halfword size).

Figure 2-1. Storage Array

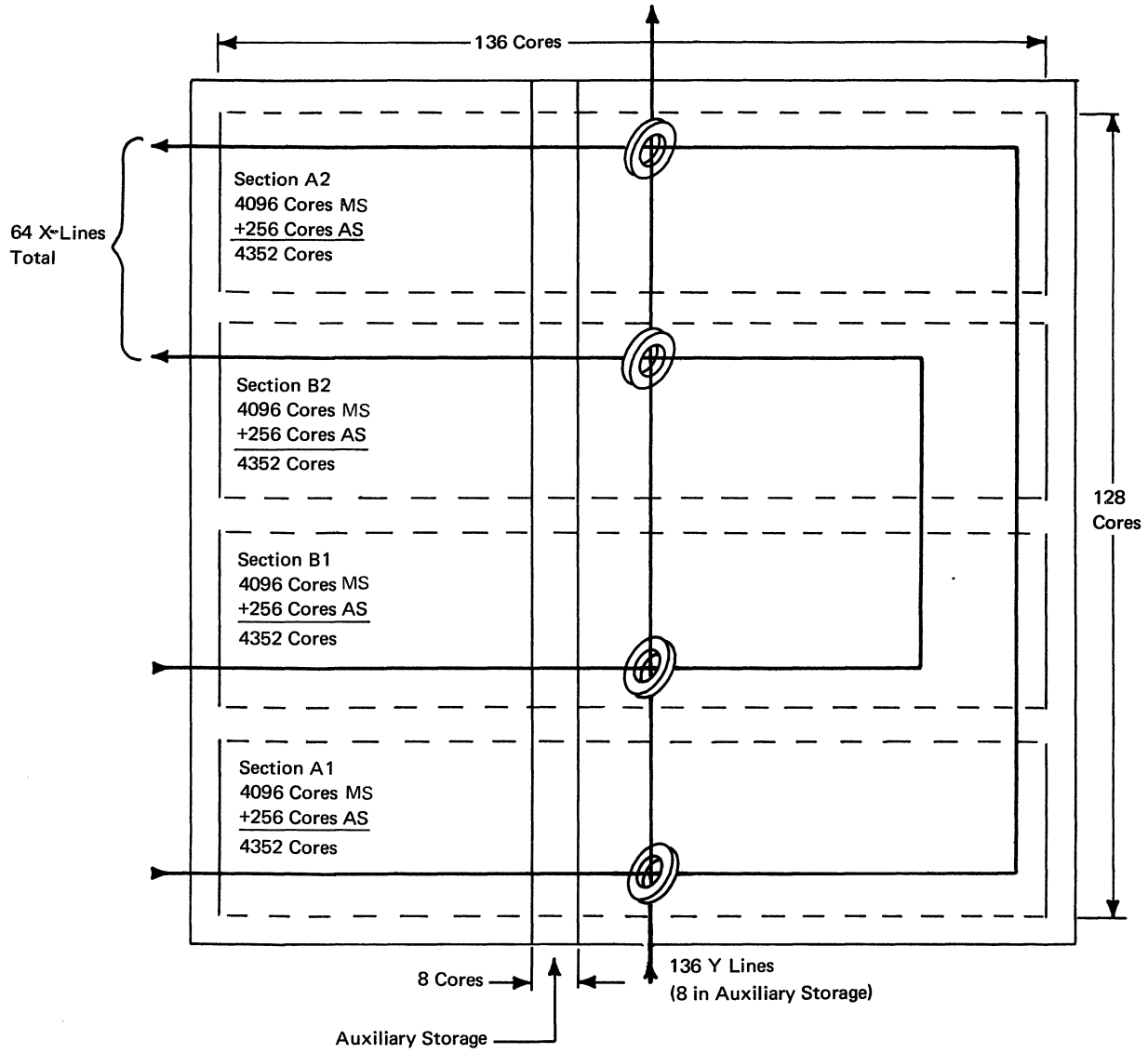


Figure 2-2. Core Plane

CORE ARRAY (8K HALFWORDS)

- An 8K core array consists of nine discrete core planes.
- Two 8K core arrays make up one 16K BSM.
- Three wires pass through each core.

The arrangement of one BSM is shown in Figure 2-1. An 8K array is composed of nine core planes (one-half of a 16K BSM). Each plane is constructed of a plastic frame approximately 6-1/2 inches square. The frame is crossed horizontally by 128 X-lines and vertically by 136 Y-lines, with a ferrite core located at each intersection of the lines. Of the 17,408 cores in each plane, 16,384 are used for main storage (control and program) and 1,024 for auxiliary storage. Each core plane consists of four sections A1, B1, B2, A2 (Figure 2-2).

Each X-line winding travels through one section of nine planes before crossing to another section of the same nine planes via an X-return card at the end of the array. The X-return card contains printed lines that carry current from the X-line winding in one section of the array to the X-line winding in another section. The winding pattern of the array is such that alternate X-line winding current is driven from

opposite ends of the core plane. An example of this can be seen in MDM 4-27, Part 4.

The Y-line winding starts at either the top (side C) or the bottom (side A) of the array and winds through each plane as shown in Figure 2-1 or Part 4 of MDM 4-27. Therefore, each Y-line winding crosses any X-line winding at 18 points: two points on any one plane. This provides a selection of 18 bits for any address.

A combined sense/inhibit line winding in each section of one plane is wound parallel to the X-line winding and passes through each core in that section.

Two 8K arrays are used to form one 16K BSM. Two, three, or four 8K arrays are possible, depending on the amount of core storage installed. In one 16K BSM, the two 8K arrays share a common set of Y-line windings. Each winding passes through all 18 planes. The 16K BSM uses two sets of X-line windings: one for the 0-8K halfword addressable array and the other for the 8-16K halfword addressable array (Figure 2-1 or MDM 4-27 Part 4).

In each 8K array the X-line windings are divided into two groups, each group containing 32 lines. Thirty-two of these lines pass through section A1 and A2, while the remaining 32 pass through sections B1 and B2. A core in section A1 or B1 is a byte-0 bit, while a core in section A2 or B2 is a byte-1 bit.

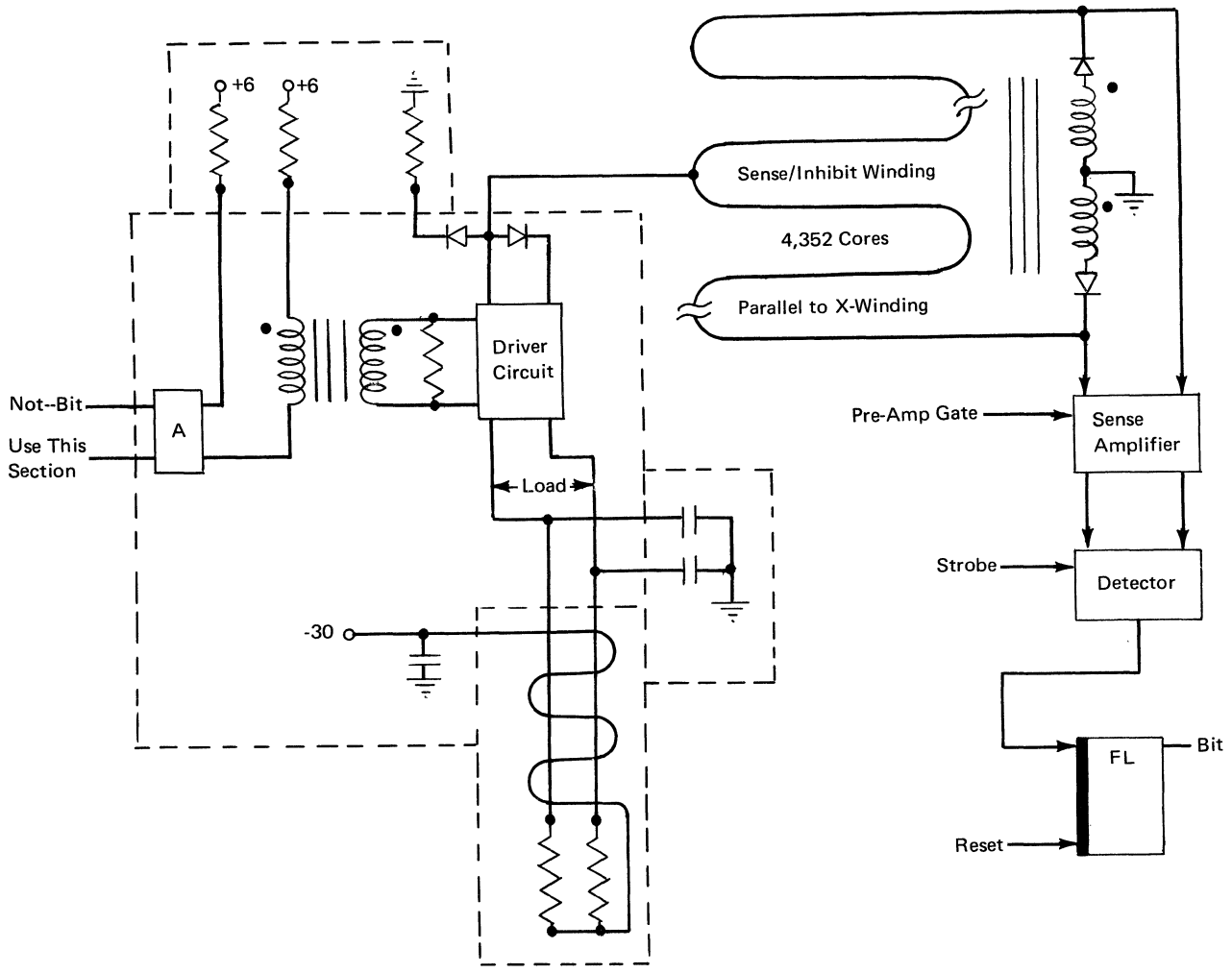


Figure 2-3. Sense and Inhibit Logic

COMMON SENSE/INHIBIT

- Use combination sense/inhibit winding.
- Each sense/inhibit winding goes through 4352 cores, parallel to X-line winding.
- The sense/inhibit windings sense pulses caused by cores changing status during the read portion of the cycle.
- The sense/inhibit windings are used to prevent cores from changing 0 status during the write portion of the cycle.

One 16K BSM contains eight sections of nine planes each, providing a total of 72 combined sense/inhibit windings. A sense/inhibit winding is parallel to the X-line winding, going through 4,352 cores in a single core plane. There are four sense/inhibit windings in each core plane. During the read portion of a storage cycle, a core that switches status (logical 1 to logical 0) induces a pulse onto the sense/inhibit winding. This pulse is amplified by a sense amplifier (Figure 2-3). If a core does not switch status (remains at a logical 0), no pulse is induced onto the sense/inhibit winding. A differential amplifier senses the change or difference in current on the sense/inhibit winding.

To prevent unwanted noise from being amplified in other storage sections, only the sense/inhibit windings output for the two sections of the BSM being addressed is gated to the sense amplifiers. If the byte being sensed (one section) is to be regenerated during the write portion of the storage cycle, a strobe pulse (generated from the storage clock) gates the sense amplifier output to the storage data register (MDM 4-27 Part 5). Sensing a logical 1 in a core causes the data-bit latch for that position to set, while a logical 0 does not set the data-bit latch.

During the write portion of the cycle, if a bit is to be stored in a core, the core is switched by the effect of the coincident X- and Y-line winding currents. Inhibit current is allowed for a bit when the corresponding data-bit latch in the storage data register is not set.

Because inhibit current flows in the opposite direction of the X-line winding write current, that position does not switch (stays a logical 0).

SAR bits 1 and 14 are used to select the group of sense/inhibit windings to be gated to the storage data register. SAR bits 1 and 14 also gate the inhibit drivers to the same sensed inhibit windings during the write portion of the storage cycle.

STORAGE CLOCKS

- Each BSM has a separate clock.
- The clock consists of two delay lines.
- Both clocks are started by a read call.

Each BSM in the 2025 has a separate clock, allowing the storage unit to operate independently of the 2025 once a read call starts a storage cycle (MDM 4-27 Part 1). Each clock is composed of two 250-nanosecond delay lines tied together. The delay lines are tapped at 25-nanosecond intervals. These outputs are ANDed and ORed together to form the necessary signals to read out and write into the core storage array.

If core-storage size requires two BSMs, both core storage clocks run at the same time. Only one BSM is allowed to read out or write in on any one storage cycle. This is accomplished by allowing only the current drivers in the selected BSM to drive the storage array. The desired BSM is selected by the high-order bit of the addressing network (storage address lines), which also blocks the control lines for the other BSM.

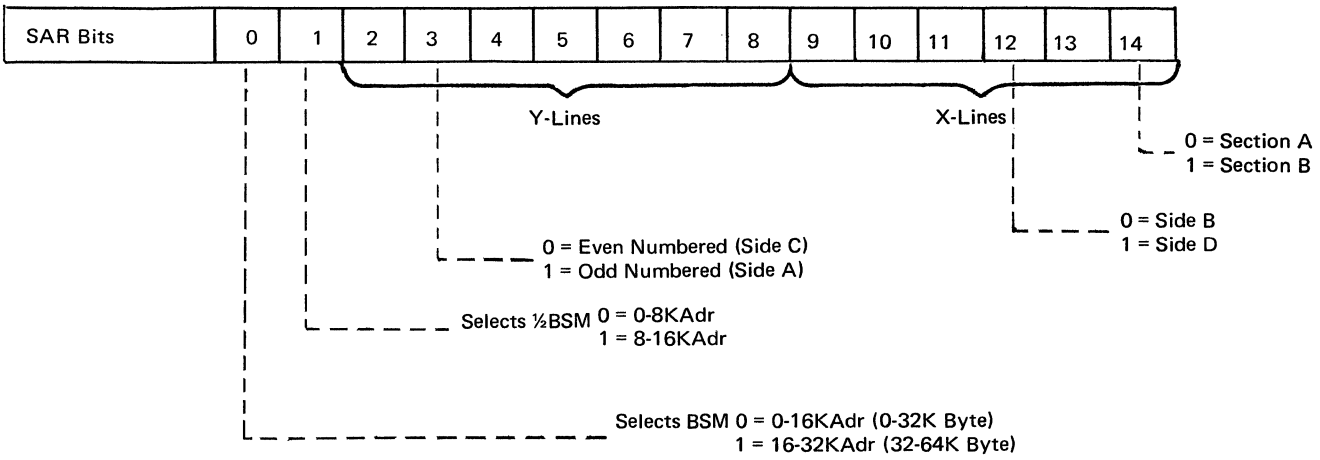
STORAGE ADDRESSING

- Fifteen address lines are decoded to select the desired address.
- Two bytes are read out for each address.

Fifteen storage address (SAR) lines are routed from the 2025 to each BSM. These lines are decoded to select the corrected BSM and the correct X- and Y-lines to read out and write into the cores selected for a given address (Figure 2-4). The SAR lines are set at P4 time of a 2025 clock cycle, and the 'read call' latch is set at T5 time (MDM 4-27 Part 1). This allows the address lines to be valid by 10 nanoseconds after 'read call' initiates a storage clock cycle.

The decode of the SAR lines for the X- and Y-lines is shown on MDM 4-27 Parts 2 and 3. The actual line selected for any given address is shown on MDM 4-27 Part 4.

If auxiliary storage is selected, the 'main aux control' latch (MDM 4-27 Part 2) is turned off. Normal X-line decode is done while the Y-line decode is limited to SAR bits 2, 3, and 8, as shown on MDM 4-27 Parts 3 and 4. SAR bit 0 selects the BSM to be used.



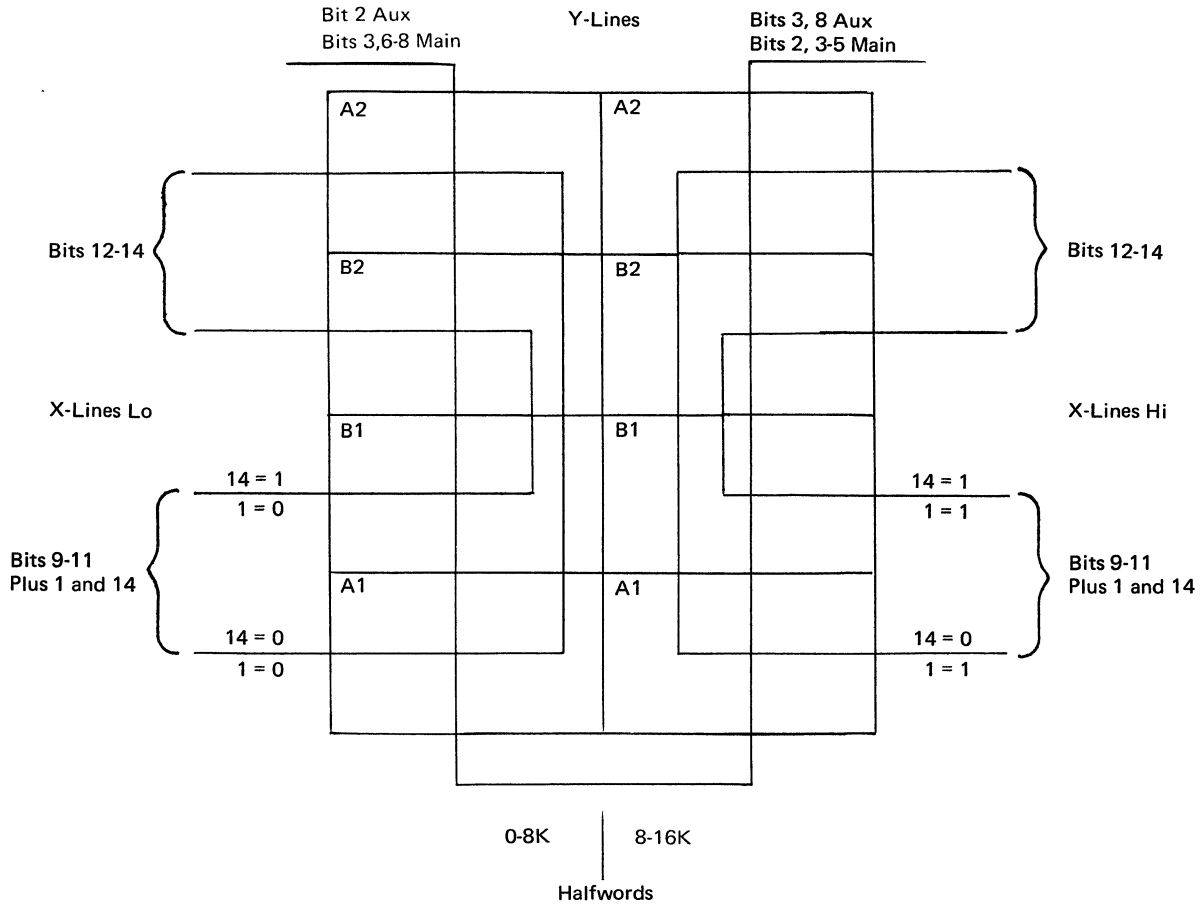


Figure 2-4. SAR Lines

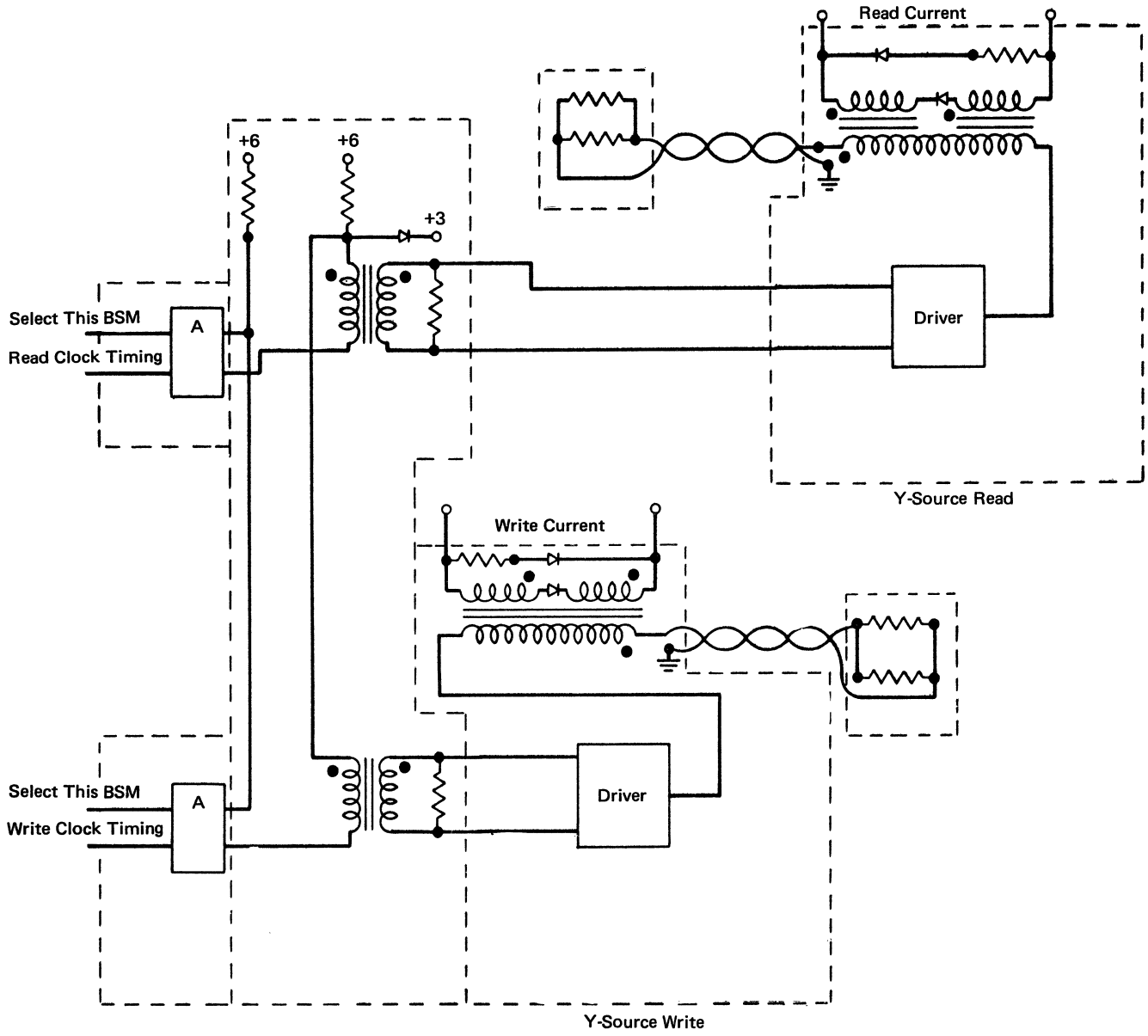


Figure 2-5. Current Sources

CURRENT SOURCES

- Current sources supply drive current to the X- and Y-line windings.
- The drive current comes from the secondary winding of a transformer.
- The primary windings of the source transformer are driven by transistors.
- Selection of current source allows current to flow only in the selected BSM.

Each current source consists of a transformer secondary winding (Figure 2-5). The primary winding of each transformer is driven by a transistor that is signaled to turn on by a time pulse developed from the storage clock. Each BSM has its own source drivers. During the read portion of the storage cycle, the 'Y-read current on' signal provides the timing for the Y-read current source. On the selected BSM, the Y-source circuit is turned on causing current to flow in the Y-source read transformer primary winding. This current causes the transformer secondary current to flow. By this time, the selection circuitry couples the source transformer to a Y-line, and current flows through the winding. During the write portion of the storage cycle, 'Y-write current source' is turned on, causing current flow in the

opposite direction in the same Y-line winding.

The same action occurs in the X-read source and X-write source circuits.

X- AND Y-DRIVE GATE AND SELECTION

- The gate and selection system directs drive current to a single X-line winding and a single Y-line winding.
- The gate and selection logic consists of control drivers, address decodes, and gates.

The gate and selection system routes drive current from a current source to a single X-line winding and a single Y-line winding. The gate and selection system acts as a switch at each end of the drive lines. Thus the current source supplies the operating current, and the gate and selection circuitry routes the current to the appropriate drive line. The logic to select one halfword in either core array of a BSM is shown on MDM 4-26.

X-Decode Drive

A 16K BSM has a total of 128 X-line windings (64 in each 8K array). For one of these 128 lines to be selected, 24 decode drivers are used. These decode drivers are divided into three groups of eight. SAR bits 12, 13, and 14 select one driver from one group. The selected driver in turn selects one end of 16 X-lines through diodes: 8 lines are selected for the low-order 8K array, and 8 lines are selected for high-order 8K array.

The other end of one of the 16 X-lines is selected as follows. SAR bit 1 selects either the high- or low-order 8K array. SAR bits 9, 10, and 11 are decoded to select one end of 8 X-lines through diodes. Only one wire is common to any two decode drivers.

After the X-line is selected, the X-current source is activated, driving current through the selected decode drivers and selected X-line.

SAR bit 1 selects the 8K array (0-8K or 8-16K), SAR bit 12 selects the side the line enters and leaves the array, and SAR bit 14 selects the sections of an array (A1, A2, or B1, B2) the line is in. Current is supplied by the X-read or X-write current source.

X-Read Decode Drive: The following action takes place for one X-drive line to be selected and driven during the read portion of a storage cycle.

1. One of the eight X-read decode drivers selected by SAR bits 12, 13, and 14 decode is active, feeding 16 diodes.
2. One of the sixteen X-read decode drivers (8 for low-order 8K array and 8 for high-order 8K array) is selected by SAR bits 1, 9, 10, and 11.
3. One of the sixteen lines (8 for each 8K array) activated by the driver in step 1 is common with one of the eight lines activated by the driver in step 2 allowing current to flow through only one line.

X-Write Decode Drive: The address decode is common to both the read and write drivers. The same steps are necessary during the write portion of the storage cycle as during the read portion, except that the current flow in the selected winding is in the opposite direction.

Y-Decode Drive (Main Storage)

The decode of SAR bits 3, 6, 7, and 8 selects one of sixteen Y-decode drivers at one end of the Y-line windings; the decode of SAR bits 2, 3, 4, and 5 selects one of sixteen Y-decode drivers at the other end of the windings. Each Y-decode driver drives 8 lines. Only one of the 8 lines is common to any two decodes.

SAR bit 3 selects the odd or even numbered Y-lines; even-numbered lines enter side C and odd-numbered lines enter side A. Each Y-line winding passes through both arrays of a BSM.

Y-Read Decode Drive: The following action takes place for one Y-drive line to be selected and driven during the read portion of a storage cycle.

1. One of the sixteen Y-read decode drivers selected by SAR bits 3, 6, 7, and 8 decode is active, feeding 8 diodes.
2. One of the sixteen Y-read decode drivers selected by SAR bits 2, 3, 4, and 5 decode is active, feeding 8 diodes.
3. One of the eight lines activated by the driver in step 1 is common with one of eight lines activated by the driver in step 2, allowing current to flow through only one line.

Y-Write Decode Drive: The address decode is common to both the read and write drivers. The same steps are necessary during the write portion of the storage cycle as during the read portion, except that the current flow in the selected winding is in the opposite direction.

XY-Decode Drive (Auxiliary Storage)

The X-decode drivers for main storage also are used for auxiliary storage. There are eight Y-line windings, which can be selected only when auxiliary storage is addressed. Four of the Y-decode drivers used for main storage also are used for auxiliary storage (MDM 4-27 Part 3). Each of these drivers drives two Y-drive windings for auxiliary storage in addition to the eight Y-line windings for main storage. The other end of the eight lines for auxiliary storage is selected by one of two drivers selected by SAR bit 2. Operation of the drivers during the read and write portions of a storage cycle is the same as for the main storage X- and Y-decode drivers.

STORAGE DATA ASSEMBLER AND REGISTER

- The storage-data register contains the output of storage and the input to storage.
- The storage-data assembler selects the data to be stored in storage.

The storage-data register is made up of eighteen latches, and is divided in half for control purposes. Bits 0 through 7 and associated parity bit (P0) make up byte 0 of the addressable halfword. Bits 8 through 15 and associated parity bit (P1) make up byte 1 of the addressable halfword (MDM 4-27 Part 5).

The reset for each half of the storage-data register occurs while the data is being read out of storage. (See timing chart on MDM 4-27 Part 1.) If the data read out of storage is not replaced by new data, a strobe pulse is generated. The strobe pulse gates the selected sense/inhibit winding pulses to the Storage Data register. A pulse on the winding of a given position at strobe time sets the data-bit latch for that position in the storage-data register. If a pulse is not on the winding at that time, the data-bit latch remains off (reset).

If the data being accessed is to be altered (this can happen only for a store operation of a storage word or a manual store operation), the data-bit latches for that byte(s) are held reset until after strobe time. The storage word operation that accesses storage can replace either a byte or halfword depending on the controls set up by the storage word decode. The replacement data is selected in the storage-data assemblers. The assembler for byte 1 receives data from the A-register or from the file data-in bus. The assembler for byte 0 receives data from the B-register or from the file data-in bus. The A-register input is used for all operations except a file share operation. During a file share cycle, the file data-in bus (bits P, 0-7) is routed to both byte 0 and byte 1 assemblers, but only one byte is gated to the storage-data register. This byte is selected by the setting of the M-1 register bit-7 latch.

The condition (1 or 0) of diagnostic register (DR) bit 2 is routed to both assemblers and is used for diagnostic purposes. When the bit is on, the parity bit for each byte is forced on, if gated to the storage-data register.

POWER SUPPLY AND TEMPERATURE COMPENSATION

- The four voltages required for operation of core storage are -3, +3, +6, -30.
- The logic boards are cooled by fans.
- The core array is heated by a heater element and fan.
- The core array heater element is controlled by a heater controller card.

A BSM requires four voltages for operating the logic and drive circuitry. The voltages and the 2025 power supply from which they originate are:

1. -3 Power Supply 6
2. +3 Power Supply 9
3. +6 Power Supply 10
4. -30 Power Supply 12.

A -18 volt supply is generated internally in the memory from the -30 volt supply. This special voltage supplies bias current for the sense amplifier.

Also supplied to each BSM is a 208-volt ac line that supplies voltage for the two logic cooling fans and the core array heater and fan.

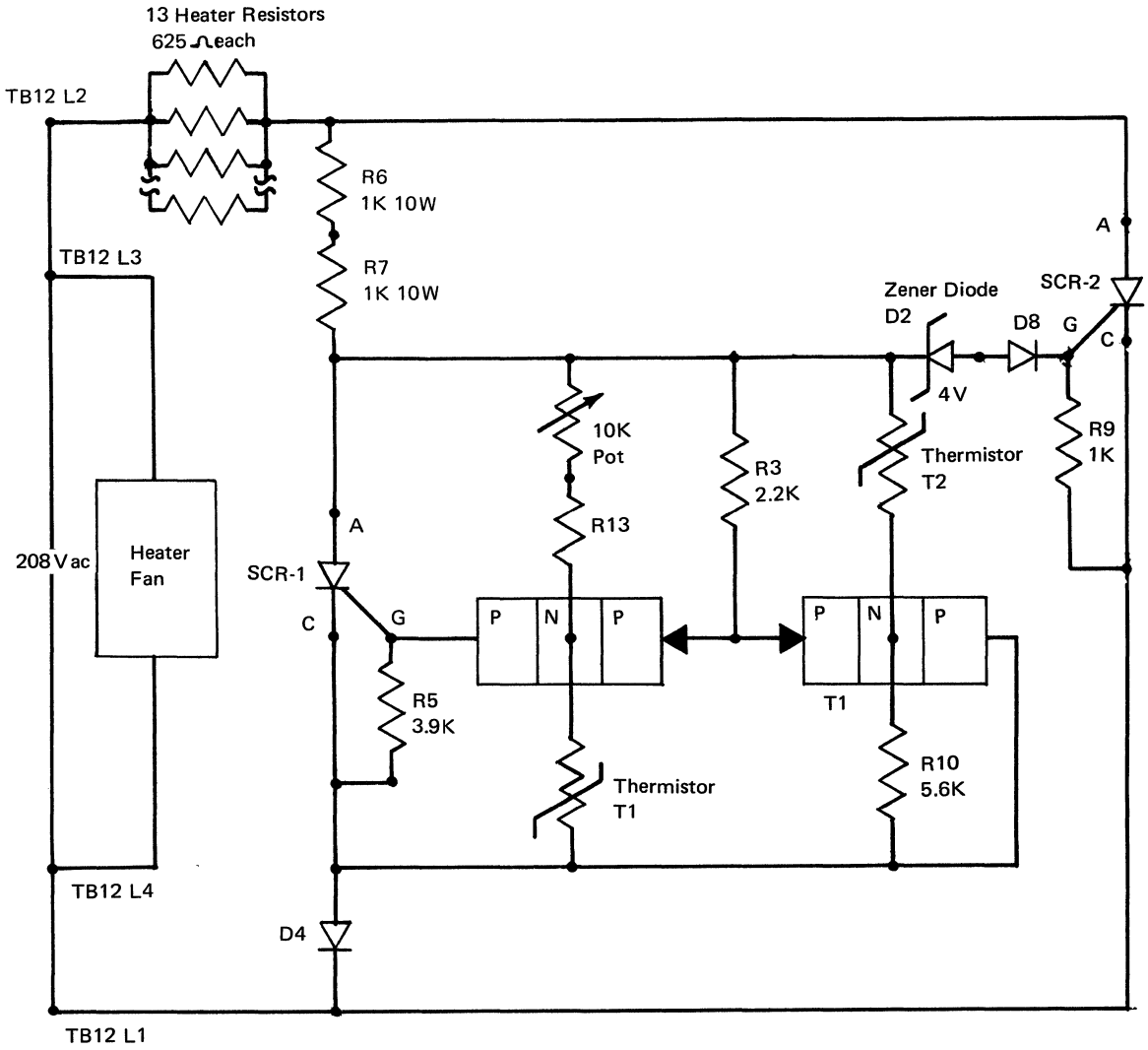


Figure 2-6. Heater Control

The heater used for a BSM unit in the 2025 is a low-noise thermistor-controlled heater. The heater element is made up of thirteen resistors connected in parallel. Circuit switching is accomplished by two SCRs controlled by a thermistor. The heater maintains array temperature at the optimum operating point.

The heater controlled circuit operates as follows (Figure 2-6). When the array is cold and power is turned on, 208V ac is applied to the input terminals. On the first positive half cycle, current flows through the heaters, through R6, R7, D2, D8, and SCR-2 (gate to cathode), to the other side of the line. When the potential on the gate of SCR-2 reaches about 6 volts, the SCR fires and the voltage drop in the circuit is across the heater resistors. As the applied voltage swings back through the zero point, SCR-2 cuts off for the negative half-cycle.

This sequence is repeated for each positive half cycle until the array warms up and the thermistors T1 and T2 begin to change value. As the temperature rises, thermistor T1 decreases resistance allowing transistor T2 to conduct. When transistor T2 fires, a voltage drop occurs across R5 gating SCR-1 on. This SCR fires at about 1.5 volts (again on the positive half cycle) and draws current through R6 and R7. This creates enough of a voltage drop across R6 and R7 to effectively turn off the heaters. The sequence is repeated for each positive half cycle until the array begins to cool.

As the array cools, the resistance change of thermistor T2 changes base voltage allowing transistor T1 to conduct. When transistor T1 is conducting, the gate voltage is removed from SCR-1 and it can no longer conduct. SCR-2 again gates as the voltage level rises to about 6 volts, allowing it to fire. The cycle repeats itself as the array warms up and cools off.

The zener diode (D2), rated at about four volts, effectively blocks the gate-on of SCR-2 until gate voltage rises above this level. For this reason, SCR-1 always fires first even though both may be gated. Once SCR-1 fires, there is not enough voltage drop across the gate circuit of the SCR-2 to fire SCR-2.

The -24V ac line passes through the BSM in the thermal circuits. There is one over-temperature sensor over the core array and one in the heater section. Should one of the sensor contacts open due to a temperature over 120F, a power-off sequence occurs.

There is an under-temperature light on the console that should be on until array temperature is above 98F. If both BSMs are installed, the light is on until the temperature in both units is above 98F.

M-REGISTER ASSEMBLER

- The M-register assembler is made up of the M0- and M1-register assemblers and the M-register gating circuits.
- The M-register assembler receives data from:
  Control register
  A-register
  W-register
  Local storage
  Console switches
  Storage data-out bus

The M-register assembler combines the various input data and control lines to form an address to be gated to the M0- and M1-register for addressing core storage. The program, control, and auxiliary storage areas are addressed through this assembler.

Microprogram trap addresses bypass the M-register circuits, and address a particular control storage area through forced addressing. Trap address formation is covered under Trap Priority Circuits in this chapter.

The M-register assembler gating circuitry generates the gating lines necessary to set the M0- and M1-register assembler with the proper data. The gating circuits are controlled by the type of control word in operation and by system status conditions (MDM 4-20).

The M0-register assembler receives data from:
Control register
W0-register
Local storage
Storage data-out bit 5
Forced bits
Switches A and B

The M1-register assembler receives data from:
Control register
W1-register
A-register
Switches C and D
Storage data-out bit P1

The various combinations of input data and gating lines are shown in MDM 4-20. The use of the M-register assembler for a particular operation is shown in the operational diagram that covers that operation. See examples on MDM 5-1 through 5-13.

M-REGISTER

- Made up of the M0- and M1-registers.
- Gated to the storage-address assembler, address-match circuits, local-storage assembler, and AB-register assembler.

The M-register provides the normal addressing for the 2025 core storage. Addresses formed in the M-register assembler are set into the M-register. The M-register is latched until the next M-register set is allowed.

The M0-register output is gated to the storage-address assembler, address-match circuits, and the local-storage assembler.

The M1-register output is gated to the storage-address assembler, address-match circuits, and the AB-register assembler.

Refer to MDM 4-21 for details of gating.

W-REGISTER

- Made up of the W0- and W1-registers.
- Receives address bits from the storage address modifier.
- Gated to the M-register for control-word addressing.

The main function of the W-register is to receive the next sequential control word address from the storage-address modifier,

and gate this address to the M-register to address the next control word.

The storage address modifier generally updates the control storage address by 2. The update is inhibited during a manual core storage display and a file-share cycle.

For details of W-Register setting and gating, refer to MDM 4-24.

ADDRESS MATCH CIRCUITS

The address match circuits compare the M0- and M1-register outputs with the setting of address switch A, B, C, and D. If there is a match, the following conditions can occur.

1. The CPU clock is stopped if:
   a. the mode switch is set to MS ADR STOP, '(not) use auxiliary storage' line active, and address match pulse generated.
   b. the mode switch is set to AS ADR STOP, 'use auxiliary storage' line active, and the address match pulse generated.
2. Set the Soft Stop latch if the mode switch is set to SAR DELAYED STOP, '(not) use auxiliary storage' line active, and 'address match' latch set.

The address-match indicator turns on when an address match occurs. Refer to MDM 4-21.

TRAPS AND PRIORITY

- Traps bypass the normal core storage addressing circuits.
- A trap address is the starting address of a trap handling microroutine that operates on the particular trap allowed.
- Traps arranged at different priority levels.

Traps

Traps are nonprogrammed nonscheduled breaks in the normal operation of the system. They cause microprogram handling of errors, I/O requests, and program loading.

When a microprogram trap occurs, the control storage address of the microroutine designated to handle that trap is forced into the storage address assembler, and the control word located at that forced address is read out and executed.

Traps are not allowed during:
- the first cycle of a file share
- the first cycle of a storage word
- the return function of the set/reset word
- a branch-on-mask word.

Traps are also inhibited when a trap with a higher priority is being operated on. Refer to MDM 4-22 and 4-23 for details of trap controls and addressing.

Priority Level	Trap Addr	Names
4	(0010)	Control Storage Load
4	(0240)	System Reset or Load (IPL)
*	(0220)	Machine Check
*	(0280)	CE TRAP - With the diagnostic control switch in the TRAP position and a positive pulse (or no wiring) to the IN hub of the CE panel, the 'machine check' latch is set and a trap is taken to 0280.
*	(0210)	Storage wrap or storage protect violation
3	(0170)	Channel High Priority
2	(0140)	File Chaining
1	(0180)	Channel Low Priority
1	(01B0)	2540 Reader
1	(0110)	2540 Punch
1	(01E0)	Communications Bit Service

* These trap routines assume the priority of the routine that was in progress when the trap occurred.

Priority

Model 25 uses a system of priorities allowing computing and input/output operations to proceed with a minimum of interference. Devices with higher data rates are given a higher position in the priority hierarchy so that the possibility of data overrun is reduced.

Six levels of operational priority are used in Model 25. As shown in the following descriptions of the various priority levels, any microprogram trap routine generally can be interrupted by a trap having a higher priority. Exceptions are noted in the level descriptions. The basic priority structure is shown in the following.

Microprogram-trap priority level 4 (highest level)
2311 disk data transfer (cycle steal)
Microprogram-trap priority level 3
Microprogram-trap priority level 2
Microprogram-trap priority level 1
Priority level 0 (lowest level).

At least two microprogram words of a trap routine must be allowed before any other trap is allowed. This permits

storage of backup addresses so that a preempted routine can be resumed later.

At the completion of any trap, system control is returned for at least one cycle to the routine that was preempted if any other priority level is tested.

Priority Level 4: System reset, initial program load (IPL), and control storage load (CSL) are assigned level 4 in the priority structure. This is the highest level: it is allowed to interrupt any other routine. When the system-reset key is pressed, the operation in progress is terminated; data overruns result if any I/O operation has not been completed.

Even though the operation being performed is halted, the full core-storage read/write cycle is completed so that the halfword being operated on is successfully stored.

For an IPL, input information is read at level 0 (except for file-share cycles) even though the calling routine is operating at level 4.

For a CSL, input information is read at level 4 and not at the normal level for the device. This is because a special (resident) microprogram is used to read the input records required for primary system initialization. In this case, any error conditions are handled at level 4.

2311 Disk Data Transfer (Cycle Steal): Data transfers between the processing unit and the integrated 2311 disk attachment are handled by using as many as two of every seven storage cycles. When the disk attachment control (DAC) requests service, the operation in progress is preempted at the end of the microprogram step in progress. Two special file-share storage control words are forced into the C-register for an input from the 2311. Two other file-share words are forced when an output-to-file operation is required. The address of the next sequential microprogram step is retained in the W-register so that the preempted routine can be resumed after the two (read or write file) storage words have been executed. The first of the file-share words is used to update the affected core-storage address. This address is stored in the 2311 area of local storage to be used for the 2311 next-byte transfer. The actual data byte is also handled during this cycle. The second forced file-share word is used to decrement the count for the file-data transfer.

For an input-from-file operation, the data byte enters core storage directly rather than via the normal A/B-register paths. For an output-to-file operation, the data byte is gated from core storage through the normal I/O Adapter-Out paths.

This method of handling data to and from the integrated 2311 devices is sufficient for the 156 KB (156,000 bytes/second) data rate of the 2311. It is because of this high data rate that disk information is processed directly into and out of core storage rather than by a microprogram-trap routine.

Although the file-share function is not actually a trap, it is included in the priority hierarchy between level 3 and level 4. Levels 0, 1, 2, and 3 are subject to a reduction of available cycles (to five out of every seven) each time data service is requested for the integrated 2311 attachment.

Priority Level 3: This priority level is used to handle high-speed I/O devices attached to the selector channel, such as magnetic tape units.

A maximum of eight cycles (up to ten cycles for the last byte) is allowed for the transfer of data.

Channel status is handled at priority level 1.

In addition to channel operations, a machine check, storage wrap, or storage-protection violation that occurs at level 3 is handled at level 3.

Priority Level 2: This priority level is reserved for chaining operations for the integrated 2311 disk attachment. A timer buffer (67-microsecond interval between 2311 chaining functions) is provided to further reduce the possibility of overruns for lower-priority devices.

A machine check or storage wrap that occurs at this level is handled at level 2. (Storage protection violation errors cannot occur for 2311 chaining functions.)

Operations at level 2 can be interrupted by a trap request at a higher level, and can interrupt operations of a lower priority.

Priority Level 1: The following microprogram functions are handled at priority level 1.

1. Channel low priority
   a. Status transfers
   b. Data chaining
   c. Command chaining
   d. Multiplexer channel data
   e. Selector channel data (buffered devices only).
2. 2540 core-storage buffer update
   a. Card-image transfer and checking,

reader
   b. Card-image transfer and checking, punch
      Note: References to the communications adapter apply only when this special feature is installed.
3. Communications channel bit service
   a. Character assembly
   b. Character disassembly.
4. Communication channel character service
   a. Read/store character
   b. Translate
   c. UCW update.

Priority Level 0: Priority level 0 handles operations performed at the lowest level. These functions are not critically time dependent. The operations handled at level 0 are:

1. 2540 reader status or chaining (including translation on reader run-in)
2. 2540 punch status or chaining
3. 1052 data, status, or chaining
4. 1403 data, status, or chaining
5. System/360 external interruption
6. System/360 I/O interruption for channel 1 (selector channel or integrated 2311 files)
7. System/360 I/O interruption for channel 0 (multiplexer channel of integrated 1052, 1403, or 2540)
8. Control-panel functions:
   Instruction step
   Set IC
   Stop key
   Alter/display on 1052
9. CPU instruction processing.

EXTERNAL FACILITIES

- Used mainly for transferring data and status information between I/O units and the microprograms.

- Some external facilities are assigned CPU status and information handling operations.

External facilities are various combinations of latches, control lines, or other status conditions. Most external facilities are gated on the external bus-in to the AB-assembler. Data being gated to most external facilities is placed on the external bus-out to the circuitry involved.

Some of the CPU externals are set and/or sampled directly, without being gated to any common bus. Externals are addressed dynamically by control words and the MODE or MMSK register decodes. Externals can be addressed manually through console switch settings.

The externals assigned for each mode of operation are shown in Figure 2-7.

AS Field Decodes	CPU Mode		2311 Disk Mode		2540 Punch Mode		1403 Printer Mode	
	MODE REG BITS 2,3,4 = 000		MODE REG BITS 2,3,4 = 001		MODE REG BITS 2,3,4 = 010		MODE REG BITS 2,3,4 = 011	
	Ext - CPU	CPU - Ext	Ext - CPU	CPU - Ext	Ext - CPU	CPU - Ext	Ext - CPU	CPU - Ext
0000	Console Addr Switch 0 (SWAB)		Tag Register In (TGRI)					
0001	Console Addr Switch 1 (SWCD)		File Bus In (FBI)					
0010	Storage Protect 0 (STP0)	Storage Protect 0 (STP0)	2311 Storage Protect Key (STP0)	2311 Storage Protect Key (STP0)	Storage Protect 0 (STP0)	Storage Protect 0 (STP0)	Storage Protect 0 (STP0)	Storage Protect 0 (STP0)
0011	Storage Protect 1 (STP1)	Storage Protect 1 (STP1)	File Out Bus Diag (FOB)	File 1400 Emulator Bus-Out (FEBO)	Storage Protect 1 (STP1)	Storage Protect 1 (STP1)	Storage Protect 1 (STP1)	Storage Protect 1 (STP1)
0100*	Dynamic Condition Reg (DYN)	System Mask (SM)	Dynamic Condition Reg (DYN)		Dynamic Condition Reg (DYN)		Dynamic Condition Reg (DYN)	
0101*	Status Register (S)		Disk Attach Status In (DASI)		Status Register (S)		Status Register (S)	
0110*	MMSK Register (MMSK)		MMSK Register (MMSK)		MMSK Register (MMSK)		MMSK Register (MMSK)	
0111*	Branch Conditions (BA)	Direct Control-1 (JO)	Branch Conditions (BA)		Branch Conditions (BA)		Branch Conditions (BA)	
1000	Direct Control In (JI)		Counter 1 High In Diag (CHI)		Diagnostic R/P Conditions 1 (RPD1)			
1001	External Interruption (XINT)	Direct Control-2 (Timing) (JA)	Counter 1 Low In Diag (CLI)	Module Select Reg (MS)	Diagnostic R/P Conditions 2 (RPD2)			Print Char Counter Length (PCCL)
1010	Timer Count (TIM)		Terminating Conditions (TC)		Reader/Punch Data In 2 (RP2)		1403 PLBAR Data In (PRT)	
1011			Serializer/ Deserializer In Diag (SDI)	Tag Register Out (TGRO)	Reader/Punch Data In 1 (RP1)		1403 PLB Data In (PRI)	1403 PLB Data Out (PRO)
1100*	Diagnostic Register (DR)		File Gated Attention (FGA)					
1101*			File Flags In (FFI)	File Flags Out (FFO)	Reader Branch Conditions (RS)			1403 Carriage Data Out (PRC)
1110*	Error Reg (MC)		Disk Status In (DS)	File Bus Out (FBO)	Reader/Punch Branch Cond (RPS)		Sense/Status Conditions (PRS)	
1111*	Soft Stop Branch Cond (BB)	MW Bits (2) of AMWP Bits (MW)	File Op Register (FOP)	File Op Register (FOP)	Punch Branch Conditions (PS)	Punch Data Out (PO)	1403 Diag Conditions (PRD)	1403 PLBAR Data Out (PR)

* These eight Ext-to-CPU external facilities can be tested using the Branch on Condition or Branch on Mask words.

Note: External facilities that can be addressed by the Set/Reset word are given in Chapter 3 under Set/Reset word description.

Figure 2-7. External Assignments (Part 1 of 2)

Figure 2-7. External Assignments (Part 2 of 2)

AS Field Decodes	1052 Mode		Communications Mode		2540 Reader Mode		Channel Mode	
	MODE REG BITS 2,3,4 = 100		MODE REG BITS 2,3,4 = 101		MODE REG BITS 2,3,4 = 110		MODE REG BITS 2,3,4 = 111	
	Ext - CPU	CPU - Ext	Ext - CPU	CPU - Ext	Ext - CPU	CPU - Ext	Ext - CPU	CPU - Ext
0000								
0001								
0010	Storage Protect 0 (STP0)	Storage Protect 0 (STP0)	Storage Protect 0 (STP0)	Storage Protect 0 (STP0)	Storage Protect 0 (STP0)	Storage Protect 0 (STP0)	Storage Protect 0 (STP0)	Storage Protect 0 (STP0)
0011	Storage Protect 1 (STP1)	Storage Protect 1 (STP1)	Storage Protect 1 (STP1)	Storage Protect 1 (STP1)	Storage Protect 1 (STP1)	Storage Protect 1 (STP1)	Storage Protect 1 (STP1)	Storage Protect 1 (STP1)
0100*	Dynamic Cond Reg (DYN)		Dynamic Condition Reg (DYN)		Dynamic Condition Reg (DYN)		Dynamic Cond Reg (DYN)	
0101*	Status Register (S)		Status Register (S)	Communications Adapt Diag Reg (CADR)	Status Register (S)		Status Register (S)	
0110*	MMSK Register (MMSK)		MMSK Register (MMSK)		MMSK Register (MMSK)		MMSK Register (MMSK)	
0111*	Branch Conditions (BA)		Branch Conditions (BA)	Communications Parity Check (PARCK)	Branch Conditons (BA)		Branch Conditions (BA)	
1000					Diagnostic R/P Cond 1 (RPD1)			
1001				Line Adapter Diag Reg (LADR)	Diagnostic R/P Cond 2 (RPD2)			
1010	1052 Data In (TI)		Data In (DAIN)		Reader-Punch Data In 2 (RP2)			
1011	Tilt/Rotate Register (TR)		Line Address In (LAIN)	Data Out (DAOUT)	Reader-Punch Data In 1 (RP1)			
1100*			Line Adapter Conditions (LACON)				Channel Branch Cond (GS)	
1101*	PR-KB Diag Br Conditions (TD)		Line Adapter Status (LASTAT)	Dial Out (DILOUT)	Reader Branch Conditions (RS)		Channel Branch Cond (GT)	
1110*	PR-KB Branch Conditons (TT)		Dial In (DILIN)		Reader/Punch Branch Cond (RPS)		Channel Diag Reg (GD)	
1111*	PR-KB Branch Conditons (TU)	PR-KB Data Out (TE)	General Status (GSTAT)	Line Address Out (LAOUT)	Punch Branch Conditions (PS)	Punch Data Out (PO)	Channel Bus In (GB/IN)	Channel Bus Out (GB/OUT)

* These eight Ext-to-CPU external facilities may be tested using the Branch on Condition or Branch on Mask words

Note: External facilities that can be addressed by the Set/Reset word are given in Chapter 3 under the Set/Reset word description.

MODE Bits

0 - 1400 Emulator Mode

1

2 }

3 } External Gating Controls

4 }

5 }

6 } Local Storage Zones

7 }

External Gating Control	
MODE Bits 2 3 4	Mode
0 0 0	CPU Mode
0 0 1	2311 Disk Mode
0 1 0	2540 Punch Mode
0 1 1	1403 Mode
1 0 0	1052 Mode
1 0 1	Communications
1 1 0	2540 Reader Mode
1 1 1	Channel Mode

Local Storage Zones	
MODE Bits 5 6 7	Zone
0 0 0	Zone 0 CPU
0 0 1	Zone 1 - 2311
0 1 0	Not Available
0 1 1	Not Available
1 0 0	Zone 4 Backups
1 0 1	Zone 5 Communications
1 1 0	Zone 6 (2540)
1 1 1	Zone 7 Channel

Figure 2-8. Mode Register Controls

MODE REGISTER

- Specifies the external gating controls and the local storage zone to be used during control word execution (except during I/O Traps).
- Specifies the emulator mode in operation.
- Set or reset by the Set/Reset control word.

The status of the MODE register determines the particular group of external facilities and local storage registers that can be addressed (Figure 2-8). The MODE register is always in control of this addressing capability except when an I/O trap is being handled.

During an I/O trap routine, the MMSK register overrides the MODE register addressing controls. The MODE register status is not disrupted during the I/O trap; control of addressing is turned back to the MODE register when the I/O trap operation is complete and the MMSK latch is reset.

MMSK REGISTER

- Designates trap priority.
- Specifies local storage and external facility addressing.
- Set and reset by the Set/Reset control word.

The MMSK register is the trap-microprogram priority register. The latches of the MMSK register are set or reset by the Set/Reset control word. The setting of one of the MMSK latches is normally done at the start of a trap microroutine; the reset is done when the trap microroutine is completed.

Setting an MMSK latch establishes a trap priority level. The level of priority depends on which latch is set. Once an MMSK latch is set (trap priority established), only traps of a higher priority level can trap in on the routine in progress. Refer to Traps and Priority for further discussion of priority and a chart of priority level.

Local storage zones and external gating controls are forced by the I/O trap MMSK latches. The MMSK latches override the MODE register status.

Example of MMSK register handling are covered in the Set/Reset control word description in Chapter 3.

The MMSK function assignments are as follows.

MMSK Latch: 0
Significance: Channel High Priority
Forces LS zone 111 and external gating control 111.

MMSK Latch: 1
Significance: 2311 Disk Control Priority
Forces LS zone 001 and external gating control 001.

MMSK Latch: 2
Significance: Channel Low Priority
Forces LS zone 111 and external gating control 111.

MMSK Latch: 3
Significance: 2540 Reader Priority
Forces LS zone 110 and external gating control 110.

MMSK Latch: 4
Significance: 2540 Punch Priority
Forces LS zone 110 and external gating control 010.

MMSK Latch: 5
Significance: Communications Channel Bit Service Priority
Forces LS zone 101 and external gating control 101.

MMSK Latch: 6
Significance: Communications Channel Character Service Priority
Forces LS zone 101 and external gating control 101.

MMSK Latch: 7
Significance: Level #1 Priority Hold
Existing LS zone and external gating control are used.

MMSK Latch: 8
Significance: System Reset, IPL, and CSI Priority
Existing LS zone and external gating control are used.

MMSK Latch: 9
Significance: Machine Check Priority
Existing LS zone and external gating control are used.

Note: The MMSK register does not change the MODE register status, but does override its addressing control. When the MMSK register is reset, the MODE register takes control of LS and external addressing again.

Refer to MDM 4-15 for details of setting and resetting the MMSK latches.

S-REGISTER

- An external register made up of 8 latches.
- Holds status and conditions to be used by the microprograms and ALU circuits.

Latches 0, 3, 4, 5, 6, and 7 can be set by the set function of the Set/Reset control word. All latches of the S-register can be reset by the reset function of the Set/Reset word. Refer to MDM 4-16.

The S-register latches have the following significance.

S0---The S0 latch output controls the true/complement gates for the B-register data input to the ALU circuits. The S0 latch is used for true/complement control only for a Move/Arithmetic word that specifies a decimal plus/minus function (@) or a binary plus/minus function(+-). If either of these functions is specified, and S0=1, the contents of the B-register is inverted and gated to the ALU circuits. If S0=0, the B-register is gated to the ALU circuits in true form.
The S0 latch is set or reset only by the Set/Reset control word.

S1---The S1 latch is set in two different ways. If a doublebyte modify operation is being performed (storage control word--word type 2) and the Z-bus and modifier are not equal to zero, the S1 latch is set on. The Z-bus output is the low-order byte of the halfword being modified, the modifier output is the high-order byte of the halfword being modified.
If an arithmetic operation (word type 3) is being performed with the A-register input being the destination of the result, and the Z-bus is not equal to zero, the S1 latch is set on.
The setting of the S1 latch is inhibited during I/O trap operations.
The S1 latch cannot be set, only reset, by the Set/Reset control word.

S2---The S2 latch is set on if a binary plus/minus or decimal plus/minus operation (word type 3) is being performed with the A-register input being the destination of the result, and the Z-bus is not equal to zero. The S2 latch is also set on if a logical operation of word type 3 is being performed with the A-register input being the destination of the result, and the Z-bus is not equal to zero.
The S2 latch cannot be set, only reset, by the Set/Reset control word.
The setting of the S2 latch is inhibited during I/O trap operations.

S3---The S3 latch can be set on during a Move/Arithmetic control word if the save-carry indication is part of the control word decode, and a carry from bit 0 of the adder circuits is detected. If the carry from bit 0 did not occur, the S3 latch is reset.
The S3 latch can be set or reset by the Set/Reset control word.
Refer to the Move/Arithmetic word

description in Chapter 3 for examples using the S3 latch status.

S4---The S4 latch is set on if an invalid decimal digit is detected in either the A- or B-registers during a decimal add function of the Move/Arithmetic word.

The S4 latch also can be set or reset by the Set/Reset control word.

S5---The S5 latch is used for general purposes and may be set or reset by the Set/Reset control word.

S6---The S6 latch is normally on, and is turned off to indicate that the operation just completed was performed by an Execute machine-language instruction.

The machine reset function turns this latch on. The Set/Reset control word can set or reset by the S6 latch.

S7---The S7 latch is set or reset by the Set/Reset control word only.

The S7 latch set on indicates a channel-0 interruption.

When S7 is named as a branch test source by either the Branch on Condition or Branch on Mask control word, the 'S7 branch condition' line is actually tested. The 'S7 branch condition' line is the result of an OR of the exceptional conditions that cause an interrupt (Figure 2-9).

For the S7 latch to be tested exclusively, the BA (Branch Conditions CPU) bit-0 position must be named as a branch test source (Figure 2-10). This test indicates the channel-0 interruption status.

Bits 0-6 of the S-register may be displayed directly by addressing the S-register. The true status of S7, however, is seen by displaying the BA-facility and observing the bit 0 indicator.

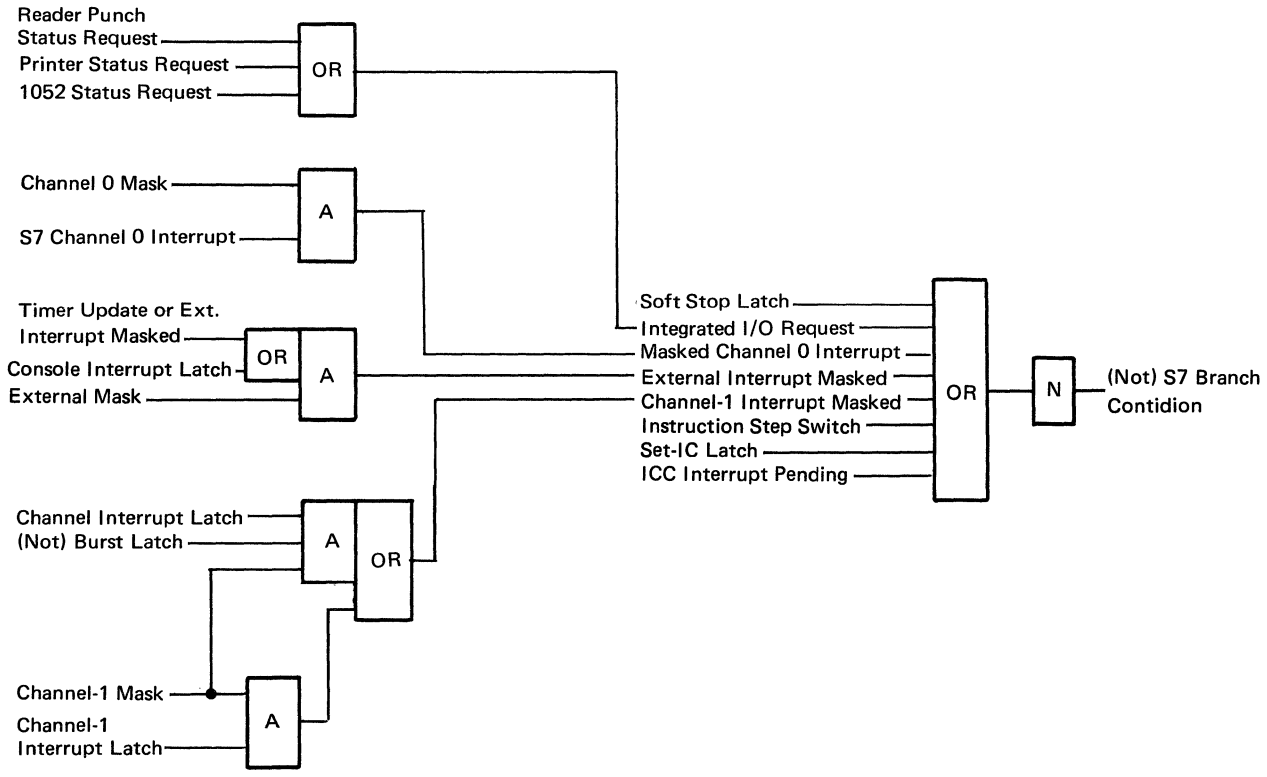


Figure 2-9. S7 Branch Conditicns

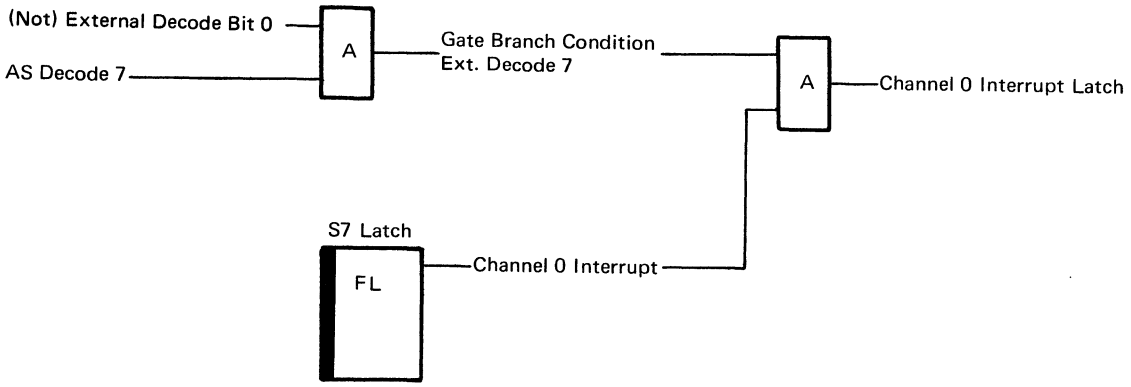


Figure 2-10. Exclusive Test of S7 Latch

BC - EXTERNAL FACILITY

The BC-mnemonic refers to certain latches and lines in the system that can be addressed by the Set/Reset control word.

When the BC facility is named in a Set/Reset control word statement, the following latches and lines are affected.

Set Function of the Set/Reset Word

K-High bit 0	Set Instruction Step Latch (if in instruction step mode)
K-High bit 1	Set Timer Interrupt
K-Low bit 3	Set Logout Latch

Other K-bits do not apply for a set function to the BC-external facility.

Reset Function of the Set/Reset word

K-High bit 0	Set Soft-Stop Latch
K-High bit 1	Reset CSL Latch
K-High bit 2	Reset System Reset Latch
K-High bit 3	Reset IC Latch
K-Low bit 0	Reset File Error Line Reset File STP Violation Latch and the File Storage Wrap Latch
K-Low Bit 1	Reset Console Interrupt Latch
K-Low bit 2	Reset IPL Latch
K-Low bit 3	Reset Logout Latch

DYNAMIC CONDITION (DYN) REGISTER

- An external facility that allows microprogram and console access of the DC-register latches, storage-wrap latch, hold-in latch, and disable position of the check control switch (Figure 2-11).
- Output lines are gated on the external bus-in to the AB-assembler.

The DYN register bits have the following significance.

Bit 0 DYN bit 0 is actually the DC-register 0-bit developed by ANDing the DC-register bit-6 and bit-7 latch outputs. DYN bit 0=1 signifies that the Z-bus is equal to zero as the result of an arithmetic control word operation, or that the status of both the modifier and the Z-bus are zero as the result of a storage control word modify operation.

Bit 1 DYN bit 1 is the sample of the 'storage wrap' latch. DYN bit 1=1 signifies that a storage-wrap violation has occurred.

Bit 2 DYN bit 2 is the sample of the DC-register bit-2 latch output. The DC-register bit-2 latch is set to flag an adder overflow during an arithmetic control word operation.

Bit 3 DYN bit 3 is the sample of the off side of the DC-register bit-3 latch output. DYN bit 3=1 signifies an adder carry condition for any arithmetic operation.
DYN bit 3=1 signifies a modifier carry out for a storage control word plus modify operation.

Bit 4 DYN bit 4 is the sample of the hold-in latch of the Direct Control feature.

Bit 5 DYN bit 5 is the sample of the check control switch Disable position.

Bit 6 DYN bit 6 is the sample of the DC-register bit-6 latch. DYN bit 6=1 specifies that the high 4 bits of the Z-bus are zero as a result of an arithmetic control word operation, or that the modifier output is zero as a result of a storage control word operation.

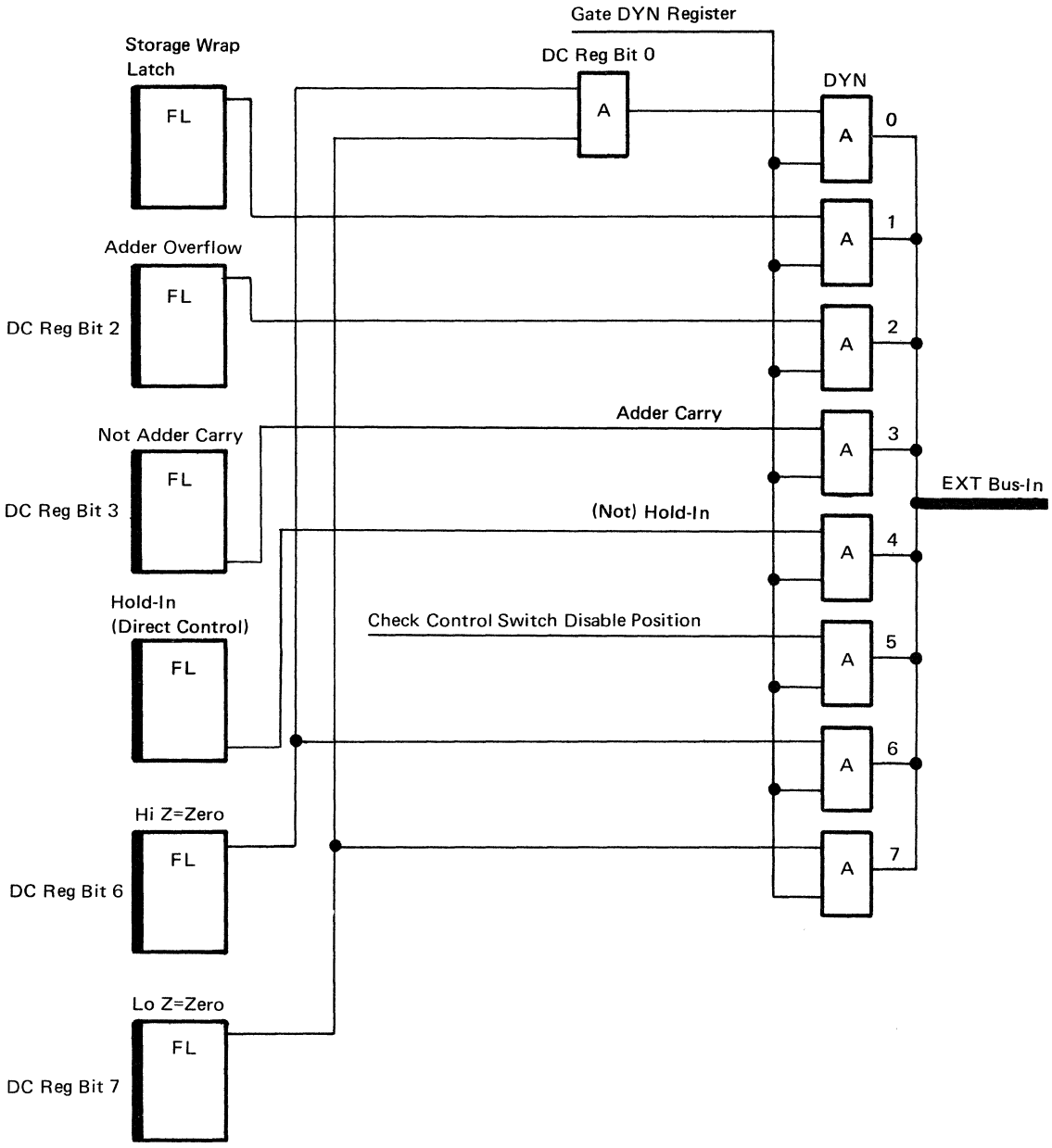


Figure 2-11. DYN-Register Gating

Bit 7 DYN bit 7 is the sample of the DC-register bit-7 latch. DYN bit 7=1 specifies that the low 4 bits of the Z-bus are zero as the result of an arithmetic control word operation, or that the entire Z-bus is zero as the result of a storage control word operation.

CONTROL REGISTER

- Receives control words on the Storage Data bus-out from core storage.
- Sixteen latches make up the control register.

The control register receives and holds bit-significant control words from core storage. The output lines of the control register are used in various combinations to control the data flow of the system

during the cycle(s) that each control word is in operation.

The control register is reset and set during T0 time of any cycle, if the control cycle latch is active. See MDM 4-12.

Bit 5 of the control register is forced to 1 for control word types 4, 5, 6 and 7. The true status of the control word 5 bit is sampled from the storage data bus-out. Therefore, the control register contents may not be the same as the hexadecimal word in the listing. Refer to the descriptions of control words: Branch Unconditional, Branch on Mask, and Branch on Condition, in Chapter 3 for details of bit-5 handling.

If the IBM 2311 Disk Storage Drive Model 1 is part of the system configuration, the control register can be forced to certain bit configurations by the file-share requests. This control register handling is covered in the IBM 2025 Processing Unit, Integrated 2311 Attachment, Form Y24-3534.

ARITHMETIC AND LOGICAL CONTROL CIRCUITS

- Comprised of the:
  AB-Register Assembler
  A-Register
  B-Register
  Arithmetic Logical Unit (ALU)
  Modifier

- Provides arithmetic and logical operation facilities, and data distribution gating.

The arithmetic and logical control circuits (Figure 2-12) are the central point of the Model 25 data flow. These circuits are used in the majority of operations for data or address handling.

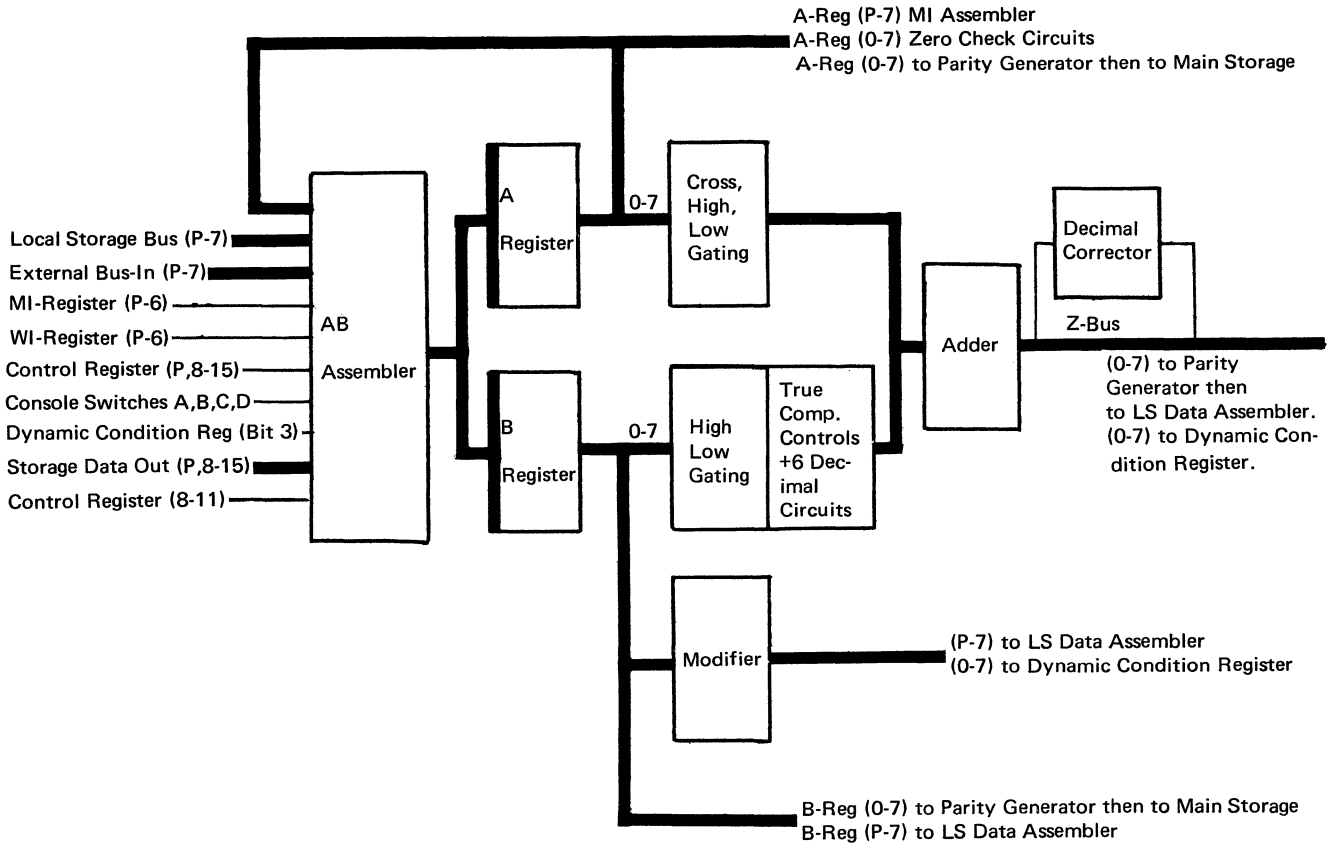


Figure 2-12. Arithmetic and Logic Control Circuits

AB-REGISTER ASSEMBLER

- Receives data from the:
  Local Storage
  External Facilities
  M1-Register
  W1-Register
  Control Register Bits 8 through 15
  Console Switches A, B, C, and D
  Dynamic Condition Register Bit 3
  A-Register
  Storage Data Bus-Out (Bits P, 8-15).

- Gates data to the A- and B-registers and to display circuits.

The local storage data is gated through the AB-assembler to the A- and B-registers whenever local storage is accessed by a 'read local storage' operation. The data is then gated to main storage, adder, local storage assembler, or to the M1-assembler. This further gating depends on the type of control word being executed.

The external facility data is gated from the external bus-in through the AB-assembler to the A- and B-registers. This gate is brought up by the AS-decode of the control word and the mode register setting for external gating.

The M1-register contents are gated to the AB-assembler whenever an Unconditional Branch word or a Set/Reset word with Link function specified is executed.

The W1-register is gated through the AB-assembler to the display circuits when a 'display W1' is initiated.

Bits 8-15 of the control register are gated through the AB-assembler to the display circuits when a 'display control register' is initiated.

Bits 8-11 of the control register are gated to the AB-assembler when an Arithmetic Constant control word is being operated on. These bits are the K-field of that control word, and are gated to both the high and low 4 bits of the B-register for further gating.

Refer to description of Arithmetic Constant control word in Chapter 3 for examples of this gating.

The console switches A, B, C and D are gated to the AB-assembler for any of the following conditions.

1. CPU mode, Move/Arithmetic control word with external to CPU gate, AS-field decode 0 (switches A and B) or AS-field decode 1 (switches C and D).
2. CPU mode, first cycle of storage control word with external data register indicated, AS-field decode 1 (switches C and D). Switches A and B cannot be accessed by this word.
3. Diagnostic switch set to either Scan Storage, Single Address, Load Program Store, Test Pattern, or Load Storage, gates switches A and B or C and D to the AB-assembler.
4. Mode switch set to MOD/LS SW position and store switch activated gate switches A and B to the AB-assembler.
5. An 'alter core storage' operation gates switches A, B, C, and D to the AB-assembler. Switch data is gated to the storage-data assembler.

The status of bit 3 of the dynamic condition register (DYN) is gated to the bit-7 position of the AB-assembler during an Unconditional Branch control word, or a Link function of a Set/Reset control word. The status of DYN bit 3 indicates the adder-carry status from a previous operation and is stored into local storage along with the backup address.

Refer to description of Set/Reset and Branch Unconditional control words in Chapter 3 for examples of DYN bit-3 handling.

The A-register is gated to the AB-assembler when a 'display BA' operation is performed. The A-register is displayed on the output of the AB-assembler. The B-register is gated to, and displayed on, the output of the LS data assembler.

The storage data-out bus (bits 8-15 plus parity) is gated to the AB-assembler when a 'display main storage' operation is performed.

A-Register

The A-register is made up of 9 polarity-hold latches (parity bit plus bits 0-7). The A-register holds data until either an A-register set pulse is generated or a machine reset occurs.

The A-register is gated to the:
1. AB-register assembler for display
2. M1-register assembler for first cycle of a storage control word (low-order byte of indirect address)
3. Storage data assembler for storage operations (odd byte)
4. ALU circuits (all operations), further gated
5. Zero-check circuits (all operations)
6. Invalid decimal digit circuits (all operations), further gated.

B-Register

The B-register is made up of 9 polarity-hold latches (parity bit plus bits

0-7). The B-register holds data until either a B-register set pulse is generated or a machine reset occurs.

The B-register is gated to the:

1. Local-storage data assembler for display of B-register contents or a manual local-storage store operation
2. Storage-data assembler for storage operations (even byte)
3. ALU Modifier for an address or count update function of a storage control word. The high byte of the address or count is the B-register output, and is incremented or decremented by 1 depending on the update function and the ALU carry.
4. ALU circuits (all operations), further gated
5. Invalid decimal digit circuits (all operations), further gated.

ALU

- Performs arithmetic and logical operations on the data gated from the A- and B-Registers.

- Circuits that make up the ALU are:
  A-Register gating
  B-Register gating
  True/Complement controls
  Adder
  ALU Controls
  Carry Insert
  Decimal Corrector
  ALU Check

A-Register Gating

The A-register contents can be gated to the adder in the following manner.

1. A-register straight (gate bits 0-7 of the A-register to bits 0-7 of the adder).
2. A-register high (gate high 4 bits of the A-register to high 4 bits of the adder, block the low 4 bits).
3. A-register low (gate low 4 bits of the A-register to the low 4 bits of the adder, block the high 4 bits).
4. A-register crossed low (gate the high 4 bits of the A-register to the low 4 bit input to the adder, block the high 4 bits).
5. A-register crossed high (gate the low 4 bits of the A-register to the high 4 bits of the adder, block the low 4 bits).
6. A-register crossed (gate the low 4 bits of the A-register to the high 4 bits of the adder, gate the high 4 bits of the A-register to the low 4 bits of the adder).
7. Block A-register input to the adder.
8. The low-order 4 bits of the A-register gating circuits are gated to the M1-register for Branch on Mask control word operations. Refer to the Branch on Mask control word description in Chapter 3 for examples of this operation.

   The low-order 4 bits of the A-register gating circuits are also gated to the branch condition detect circuits for Branch on Condition control word operation. Refer to the Branch on Condition control word description in Chapter 3 for examples of this operation.

B-Register Gating

The B-register contents can be gated to the adder in the following manner.

1. B-register straight (gate bits 0-7 of the B-register to bits 0-7 of the true/complement controls).
2. B-register high (gate bits 0-3 of the B-register to bits 0-3 of the true/complement controls, block bits 4-7 of the B-register).
3. B-register low (gate bits 4-7 of the B-register to bits 4-7 of the true/complement controls, block bits 0-3 of the B-register).

The B-register gating circuits also generate a line, B=1, for addressing and count updating for storage and Branch Unconditional control words. In combination with the carry insert line, this line effectively increments or decrements the A-register input to the adder by 2.

True/Complement Controls

The true/complement controls gate the B-register contents to the adder in the proper format. If the true control is active and the operation is not a decimal add, the B-register contents are gated undisturbed to the adder. If the true control is active and the operation is a decimal add, a value of 6 is added to each 4-bit group of the B-register contents. The result of the decimal add is tested and corrected by the Decimal Corrector circuitry on the output of the adder.

If the complement control is active and the operation is either binary or decimal, the contents of the B-register are inverted and gated to the adder.

Adder

The adder is made up of AND and OR circuits that receive input data from the A-register gating circuits and the B-register true complement circuits. This data is gated

through the adder by the ALU control lines generated for a particular operation.

Outputs of the adder are to the decimal corrector, Z-bus, S-register, dynamic condition register, modifier, and Z-bus check network.

The adder circuits are duplicated for both the plus and minus levels (2-wire adder).

ALU Controls

The manner in which the data is to be handled through the adder is specified by the control lines generated in the ALU control circuits. Refer to MDM 4-36.

The arithmetic control words (word type 1 and word type 3) specify the effective operation to be performed by the adder. The decode of the arithmetic functions determines the actual ALU controls generated.

The output control lines from the ALU control circuits are:

AND
XCR
LOGICAL

The output of the True/Complement controls are:

TRUE
COMPLEMENT

The combinations of these lines that may be developed are:

Arithmetic Function	ALU Controls
* (AND)	AND, LOGICAL, TRUE
*- (Compl'nt AND)	AND, LOGICAL, COMPLEMENT
¤ (Exclusive OR)	XOR, LOGICAL, (NOT) AND, TRUE
$ (OR)	LOGICAL, (NOT) XOR, (NOT) AND, TRUE
+ (Plus)	XOR, (NOT) LOGICAL, (NOT) AND, TRUE
- (Minus)	XOR, (NOT) LOGICAL, (NOT) AND, COMPLEMENT
+- (Binary add)	XOR (other lines same as + or - above)
@ (Decimal add)	XOR (other lines same as + or - above)

The XCR control is brought up for any other control word that specifies an ALU function.

Various combinations of the active ALU control lines and the not-active control lines provide the necessary gating in the adder for any operation.

When the combination of XOR and (not) LOGICAL control lines are active, carries from one adder position to another are allowed to occur. All other combinations of ALU controls inhibit carry handling.

Carry Insert

The carry-insert circuits generate a +1 input to the low-order (bit-7) position of the adder. The carry insert is generated for the conditions shown in Figure 2-13.

The Move/Arithmetic control word may call for the generation of the carry insert with the following format examples.

```
*AC=A+B+C
 AC=AL+B+C
*AC=A-B+C
 AC=0-B+C
 A=A+-B+C
 AC=A@B+C
```

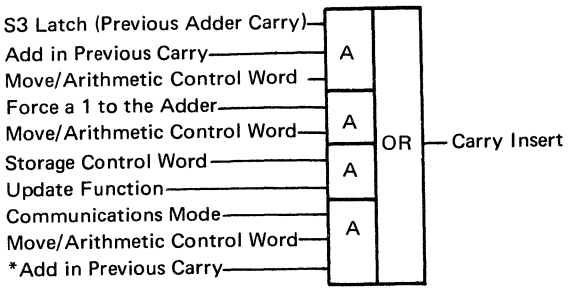
These formulas interrogate the S3 latch that indicates the status of the previous adder carry (carry from high-order adder position).

*These 2 formulas when performed in Communications Mode effectively force a 1 to be added to bit 7 of the adder; the S3 latch is not sampled.

```
AC=A+B+1
A=A-B+1
```

These formulas force a 1 to the low-order (bit-7) position of the adder.

The storage control word with an update function indicated causes the carry insert to be generated. The carry insert is used in conjunction with a generated B=1 control line to effectively cause an update of 2. For updates of 1, the carry insert is generated alone.



*The S3 latch is not sampled in this operation. The carry is effectively forced.

Figure 2-13. Carry-Insert Generation

Decimal Corrector

The decimal corrector circuits are located on the output of the adder and are active for decimal arithmetic operations.

The decimal corrector is divided into two individual elements: decimal corrector high and decimal corrector low. The decimal corrector high samples the 0-bit carry and the status of bits 0, 1, and 2. If a 0-bit carry occurs during the decimal arithmetic operations, bits 0, 1, and 2 are gated directly to the Z-bus. If a carry does not occur, a value of 1 is added to the 0- and 2-bit positions and the result is gated to the Z-bus. Carries are allowed from positions 1 and 2; the 0-bit carry is ignored on the correction operation.

The decimal corrector low works in the same manner as the high circuits. The 4-bit carry and the status of bits 4, 5, and 6 are sampled. If a 4-bit carry is detected, bits 4, 5, and 6 are gated directly to the Z-bus. If no 4-bit carry is detected, a value of 1 is added to bits 4 and 6 and the result is gated to the Z-bus. Carries are allowed from bits 5 and 6; the 4-bit carry is ignored on the correction operation.

ALU Check

The plus and minus lines for each bit of the adder, and the 0-bit adder carry are sampled by the ALU check circuits. If the plus and minus lines of any one of these bits are both plus or minus at the same time, an ALU check is generated, and the ALU check latch is set.

The ALU check is inhibited during a Branch on Mask or Branch on Condition control word, and the first cycle of a storage control word.

The ALU check line causes the hard-stop latch to be set if the check-control switch is in the process position and MMSK latch 8 or 9 is on, or if the check-control switch is in the stop position.

The ALU check latch turns on the ALU error indicator and activates the bit-5 position of the machine-check register (MC).

ALU Modifier

- Receives data from B-register.
- Modifies high-order byte of address or data halfword.
- Output to LS-data assembler, dynamic condition register, and S-register.

The modifier is located on the output of the B-register (Figure 2-14), and receives data every cycle from the B-register. The data is either modified by plus or minus 1, or left undisturbed.

Although the modifier circuits are always active, the output of the modifier is gated to the LS-data assembler only for a storage control word (word type 2) address modification or double-byte modify operation. The modifier output is gated also to the LS-data assembler on a display local-storage operation. The modifier output is the byte of the LS-register addressed for display.

During a storage control word operation involving address modification, the modifier receives the high-order byte of the address register being used. The low byte of the address register is gated to the ALU, and is modified. If the modification was an increment, and an adder carry was generated, the modifier contents are incremented by +1 and gated to the LS-data assembler.

If the modification was a decrement and no adder carry was generated, the contents of the modifier are decremented by 1 and gated to the LS-data assembler.

If an increment operation was specified and an adder carry was not generated, or if a decrement operation was specified and an adder carry was generated, the contents of the modifier are gated undisturbed to the LS-data assembler.

The modifier output is also gated to the input circuitry of the dynamic condition (DYN) register. If the storage control word is in operation and the modifier output is zero, the bit-6 latch of the DYN register is set.

The storage control word double-byte modify operation is handled by the modifier in the same manner as just described. The high-order byte, gated to the modifier for a double-byte modify, is not necessarily from an address register source but can be from any LS-register addressed by the storage control word.

If the storage control word double-byte modify operation is in process and the output of the modifier and the Z-bus are not zero, the S-register 1 latch is set.

During the second cycle of a file-share operation, the signal (count gone to zero) is generated if the modifier and Z-bus outputs are both zero.

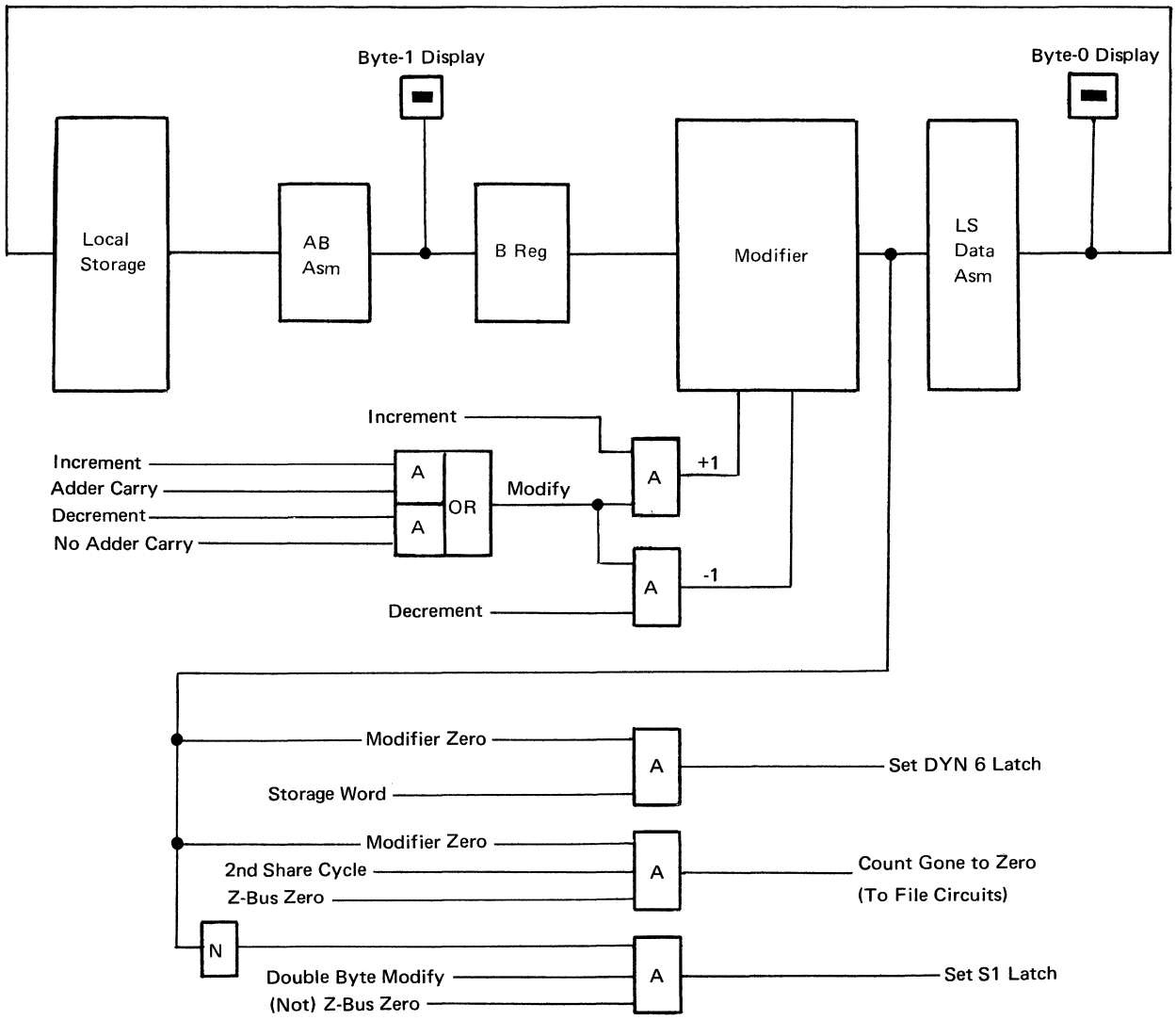


Figure 2-14. Modifier Controls

LOCAL STORAGE (FIGURE 2-15)

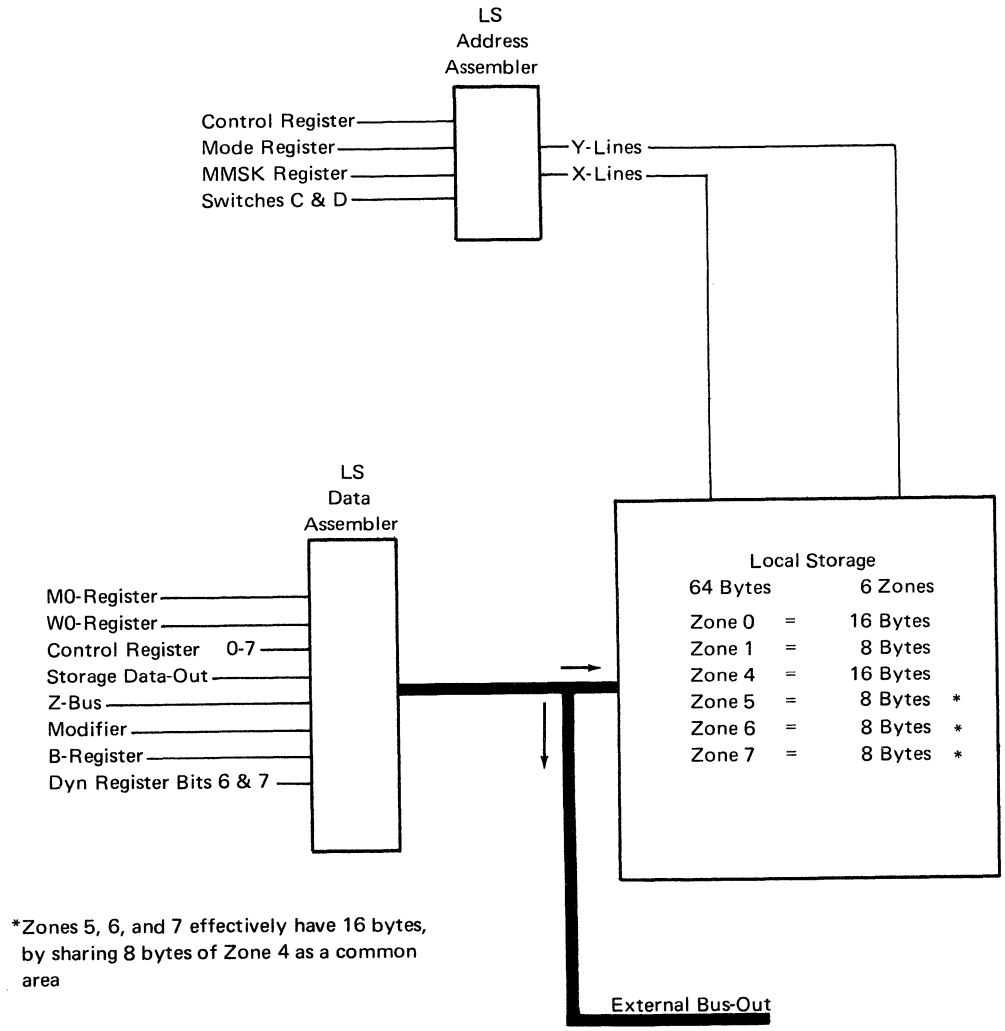
- Comprised of a 64-byte high-speed monolithic stack.
- Addressed through the local-storage address assembler.
- Receives data from the local-storage data assembler.
- Used mainly as intermediate storage for data handled by the system.

The 64-byte local storage is employed as an intermediate storage device for data and addresses used during the execution of some machine-language instruction or I/O operation. Addressing is performed by 8 X-lines and 8 Y-lines plus a read or write control pulse. Refer to MDM 5-2 through 5-13 for local storage addressing and timing examples.

Local storage is divided into six zones that are assigned for use by certain Model 25 modes of operation. The six zones are:

Zone 0, CPU mode, 16 bytes
Zone 1, 2311 mode, 8 bytes
Zone 4, Backup area, 16 bytes
Zone 5, Communications, 8 bytes
Zone 6, 2540 mode, 8 bytes
Zone 7, Channel mode, 8 bytes

The zones of local storage are addressed, dynamically through the combined decode of the AS- or BS-field of a control word, and through the mode or MMSK register status. Any byte in local storage can also be addressed manually, by the settings of console switches C and D, for alteration or display.



*Zones 5, 6, and 7 effectively have 16 bytes, by sharing 8 bytes of Zone 4 as a common area

Figure 2-15. Local Storage (Part 1 of 2)

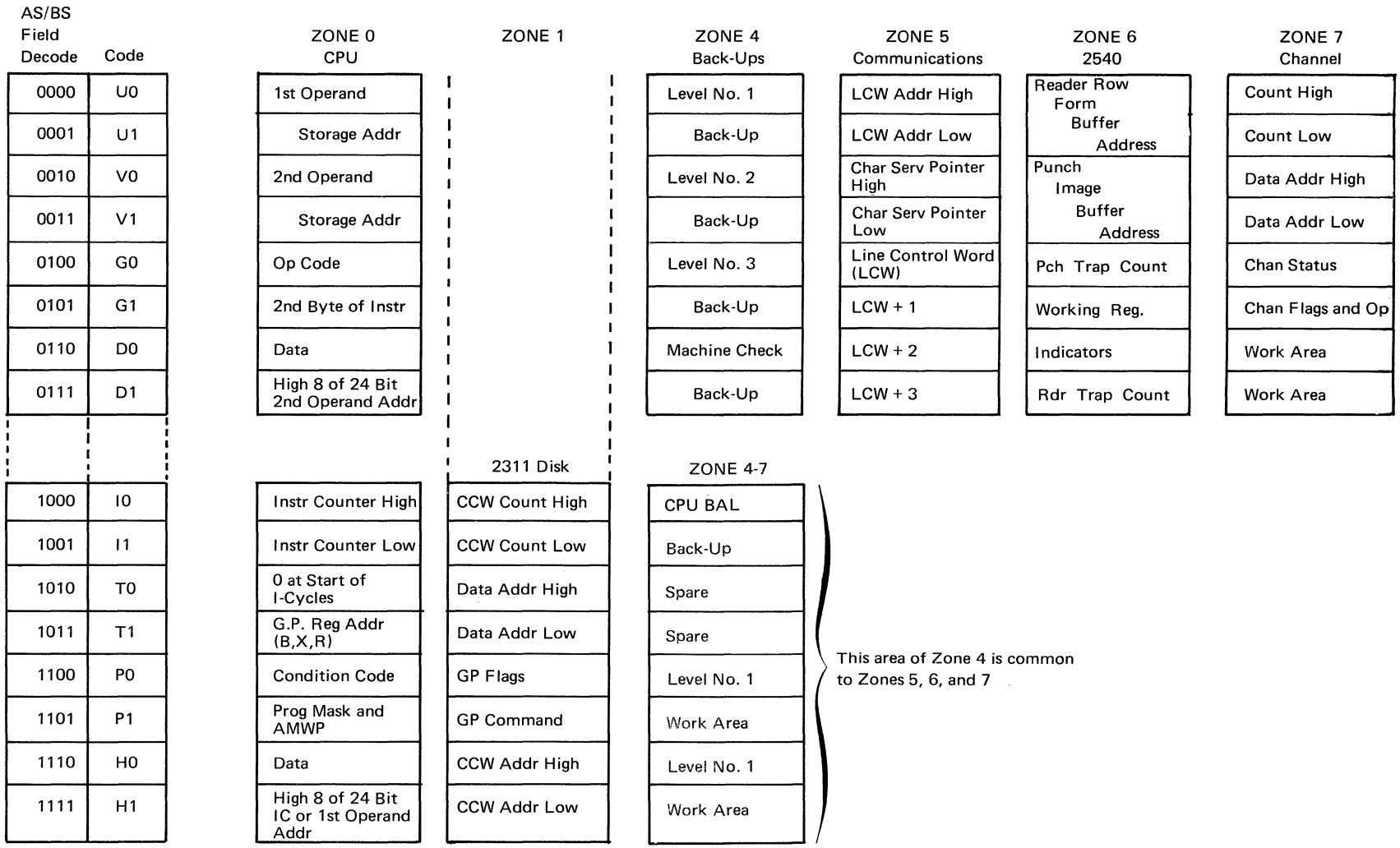


Figure 2-15. Local Storage (Part 2 of 2)

Zone 0

Zone 0 is addressed when the Model 25 is operating in CPU mode. There are 16 bytes within the zone-0 area addressed by X-lines 0-7 and Y-lines 0 and 1.

Each byte within zone 0 is assigned a particular function; however, the use of these bytes is not restricted to these functions entirely. The zone-0 area is used as a working area by some of the microroutines when the assigned function has been performed or has been saved in some other storage area.

AS/BS Decode (hex)	Sym	X	Y	Assigned Function
0	U0	0	0	First operand storage address
1	U1	1	0	First operand storage address
2	V0	2	0	Second operand storage address
3	V1	3	0	Second operand storage address
4	G0	4	0	Op-code byte of machine language instruction
5	G1	5	0	Second byte of machine language instruction
6	D0	6	0	Data
7	D1	7	0	High byte of 24-bit second operand address
8	I0	0	1	High byte of instruction counter
9	I1	1	1	Low byte of instruction counter
A	T0	2	1	0's at start of I-cycles
B	T1	3	1	General purpose register address
C	P0	4	1	Condition code (bits 0-3)
D	P1	5	1	Program mask (bits 0-3) AMWP (bits 4-7)
E	H0	6	1	Data
F	H1	7	1	High byte of 24-bit instruction counter or high byte of first operand address.

Zone 0 is also used by the 1052 microroutines and is assigned the following tentative function.

AS/BS Decode (hex)	Sym	X	Y	Assigned Functions
0	U0	0	0	Count Field
1	U1	1	0	
2	V0	2	0	Data Address
3	V1	3	0	
4	G0	4	0	Channel Status
5	G1	5	0	Flags/Op Byte
6	D0	6	0	UCW Address
7	D1	7	0	UCW Address
8	I0	0	1	
9	I1	1	1	
A	T0	2	1	
B	T1	3	1	
C	P0	4	1	
D	P1	5	1	Sense Byte
E	H0	6	1	Unit Status
F	H1	7	1	

Zone 1

Zone 1 is addressed when Model 25 is operating in 2311 mode. There are 8 bytes within the zone-1 area addressed by X-line 0-7 and Y-line 2.

The eight bytes of zone 1 are assigned particular functions, but are used as working areas when the assigned function has been performed or the contents of zone 1 have been saved.

AS/BS Decode (hex)	Sym	X	Y	Assigned Function
8	I0	0	2	High byte, CCW count
9	I1	1	2	Low byte, CCW count
A	T0	2	2	High byte data address
B	T1	3	2	Low byte data address
C	P0	4	2	File attachment flags
D	P1	5	2	File attachment command
E	H0	6	2	Next CCW address high byte
F	H1	7	2	Next CCW address low byte

Zone 4

Zone 4 is the backup area for all modes of operation except 2311 mode. There are 16 bytes within the zone-4 area that are addressed through combinations of X-lines 0-7 and Y-lines 3 and 4.

When CPU mode is being used, the backup area can be accessed in 2 ways: the MODE register set to point to the LS backup area (bits 5, 6, and 7 set to 1, 0, 0) or a BAL or RTN control word being executed. The BAL and RTN control words force the I-register of the backup area to be addressed.

In 2540, communications, or channel mode, the low-order eight bytes of the backup area (I0-H1) are addressed automatically. That is, if a control statement is written naming an LS byte I0-H1 and the microroutine is in 2540, communications, or channel mode, the backup area is addressed. All 16 bytes of the backup area can be addressed manually by console switch C and D. Backup area assignments are as follows.

AS/BS Decode (hex)	Sym	X	Y	Assigned Function
0	U0	0	4	High-order level-1 backup address
1	U1	1	4	Low-order level-1 backup address
2	V0	2	4	High-order level-2 backup address
3	V1	3	4	Low-order level-2 backup address
4	G0	4	4	High-order level-3 backup address
5	G1	5	4	Low-order level-3 backup address
6	D0	6	4	High-order machine check backup address
7	D1	7	4	Low-order machine check backup address
8	I0	0	3	High-order CPU Branch and Link backup address
9	I1	1	3	Low-order CPU Branch and Link backup address
A	T0	2	3	Spare
B	T1	3	3	Spare
C	P0	4	3	Level-1 work area
D	P1	5	3	Level-1 work area
E	H0	6	3	Level-1 work area
F	H1	7	3	Level-1 work area

The level-1, -2, and -3 backup registers are used to retain the address of the next sequential control word of routine that was interrupted by one of the level-1, -2, or -3 traps. When the trap routine is completed, this backup address is restored to the M-register, and the interrupted routine is continued.

Zone 5

Zone 5 is addressed in communications mode (MMSK bits 5 or 6 on). There are eight bytes in the zone-5 area addressed by X-lines 0-7 and Y-line 5. The functional assignments for these eight bytes are as follows.

AS/BS Decode (hex)	Sym	X	Y	Assigned Function
0	U0	0	5	High byte of LCW address
1	U1	1	5	Low byte of LCW address
2	V0	2	5	High char service pointer
3	V1	3	5	Low char service pointer
4	G0	4	5	Line Control Word (LCW)
5	G1	5	5	LCW + 1
6	D0	6	5	LCW + 2
7	D1	7	5	LCW + 3

Zone 6

Zone 6 is addressed in 2540 mode (MMSK bits 3 or 4 on). There are eight bytes in the zone-6 area addressed by X-lines 0-7 and Y-line 6. Zone 6 also can be addressed by setting the MODE register bits 5, 6, 7, to 1, 1, 0, respectively.

AS/BS Decode (hex)	Sym	X	Y	Assigned Function
0	U0	0	6	2540 Reader row address
1	U1	1	6	2540 Reader row address
2	V0	2	6	2540 Punch Image
3	V1	3	6	2540 Buffer Address
4	G0	4	6	2540 Punch trap count
5	G1	5	6	2540 Not used
6	D0	6	6	2540 Indicator bits
7	D1	7	6	2540 Reader trap count

Zone 7

Zone 7 is addressed in channel mode (MMSK bits 0 or 2 on). There are eight bytes in the zone-7 area addressed by X-lines 0-7 and Y-line 7. The functional assignments for these bytes are as follows.

AS/BS Decode (hex)	Sym	X	Y	Assigned Function
0	U0	0	7	Count
1	U1	1	7	Count
2	V0	2	7	High-byte data address
3	V1	3	7	Low-byte data address
4	G0	4	7	Channel status
5	G1	5	7	Op Flag
6	D0	6	7	Channel identification
7	D1	7	7	Burst channel unit address or byte channel UCW address

LOCAL STORAGE ADDRESS ASSEMBLER

- The local storage address assembler creates the X- and Y-lines that address a specific byte in the local storage stack.
- The X-lines (X0-X7) are selected by combinations of AS- and BS-decodes and MODE or MMSK status.
- The Y-lines (Y0-Y7) are selected by combinations of MODE or MMSK status and control register decodes.
- Console switches C and D are used to address the local storage area manually.

The local storage address assembler comprises a group of circuits that condition the proper X- and Y-lines to address a specific byte in the local storage stack. An example of addressing a particular byte in the stack with a given set of conditions is given here. For a detailed logical picture of overall local storage addressing, refer to MDM 4-30. For examples of specific addressing, refer to MDM 5-2 through 5-13.

The example of local-storage addressing (Figure 2-16) assumes the following.

1. The MODE Register is set to access LS zone-4 bits 5, 6, and 7 set to 1, 0, 0, respectively.

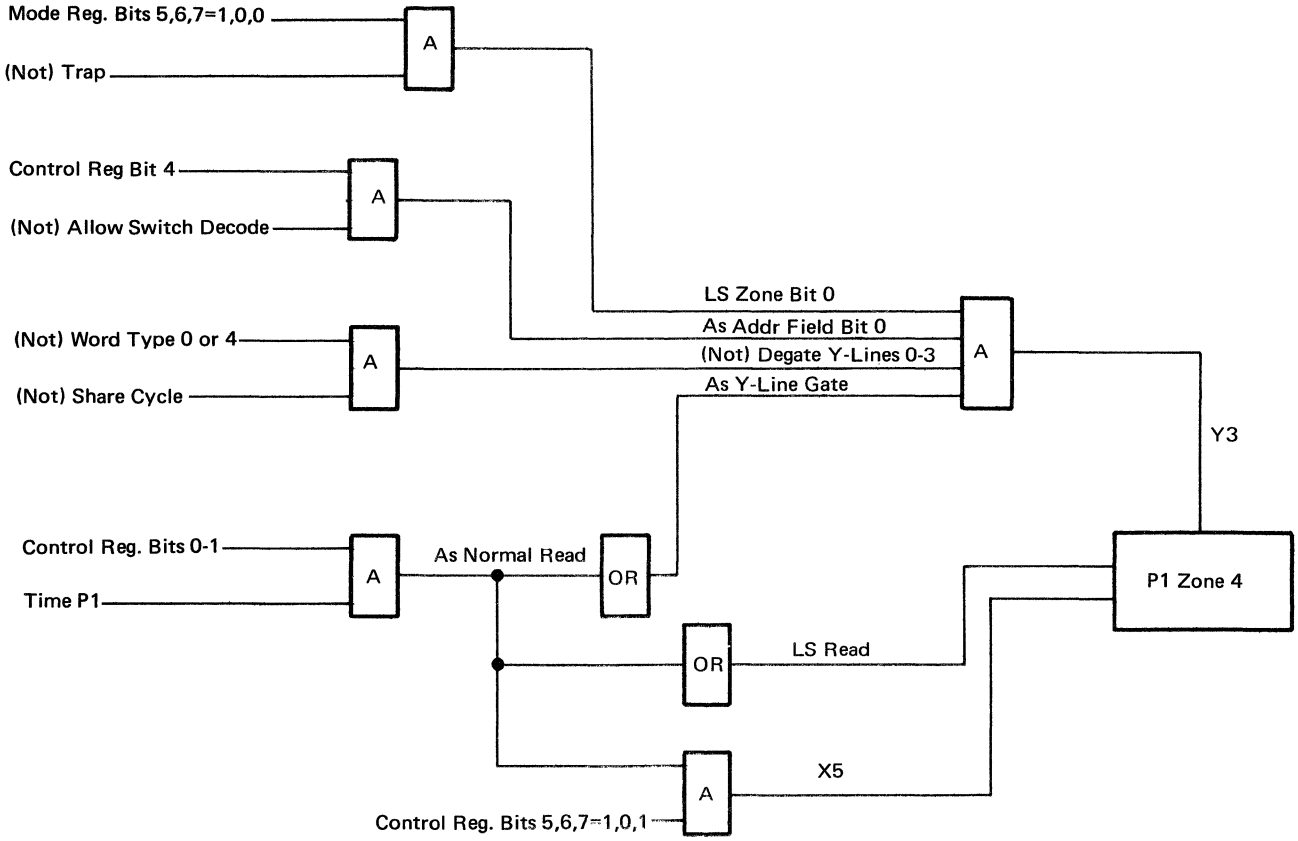


Figure 2-16. LS-Addressing Example

2. A Branch on Condition word is in the control register with an AS-field decode of (1101).
3. The microprogram in process is not a trap routine (MMSK latches all off).

LOCAL STORAGE DATA ASSEMBLER

- The local storage data assembler receives data for gating from the:
  Storage data bus-out
  Z-bus
  Modifier
  B-register
  Dynamic condition register (bits 6 and 7)

- The local storage data assembler receives data for display from the:
  M0-register
  W0-register
  Control register (bits 0-7)
  Storage data bus-out byte 0

The local storage data assembler receives data from the storage data bus-out (SDBO) when a core storage area is read out by a storage control word (word type 2). The data is gated to the LS-data assembler as a halfword of information (bytes 0 and 1) and byte selected at the assembler. Byte 0 of the SDBO is also gated to the LS-data assembler when core storage data is displayed. Byte 1 of the SDBO is gated to the B/A assembler for this same display request.

The Z-bus is gated to the local storage data assembler when:
1. Any control word except Branch on Mask or Branch on Condition is executed.
2. Display ALU is executed.

The Z-bus is gated as specified in Item 1 but is not necessarily used by the operation in process. Further gating of bus information to either local storage or external bus-out depends on the control word in the control register.

The modifier is gated to the local storage data assembler when a storage control word is executed. The modifier output is the high-order byte of the address register and can be updated during storage word execution. The modifier is

also gated to the LS-data assembler when 'display local storage' is executed. The modifier contains the LS-byte addressed by the console switches.

The B-register is gated to the local storage data assembler when 'display B/A' is executed.

The B-register is gated to the local storage data assembler when a manual store-LS operation is executed. The B-register contains the data from console switches A and B.

Dynamic condition register bits 6 and 7 are gated to the LS-data assembler when storing the backup address during a link function of a Set/RST word or a Branch Unconditional word. Bits 6 and 7 are gated to the high-order 2-bits of the backup area.

The M0-register, W0-register, byte 0 of the control register, and byte 0 of the storage data bus-out are gated through the LS-data assembler when a display of those particular bytes is called for.

CPU CHECKS

- A CPU check occurs when a parity error is detected and allowed at the A-register, B-register, storage data bus-out, storage address lines, or when an ALU check of the Z-bus is detected and allowed.

- An additional CPU check occurs on a storage protection parity check when the storage protection feature is installed.

The 'allow parity check' latch (MDM 4-10 Part 1) must be on before any CPU check can be allowed. The 'allow parity check' latch is turned on at T8 time of the first clock cycle after:

1. The start key is pressed.
2. The system-reset key is pressed.
3. The load key is pressed.
4. The CSL key is pressed.
5. Any scan operation and the 'clock start interlock' latch is off.

A CPU check occurs when the 'allow parity check' latch is on and a parity error is detected at the A-register, B-register, storage data bus-out, storage address lines, storage protection data, or when an ALU check is detected. A machine check trap can be taken and/or the CPU clock can be stopped by setting the hard-stop latch (Figure 2-17).

A-Register Check: The 'A-register check' latch is set when the A-register has an even number of bits set and the A-register check gate gate is active (MDM 4-10 Part 3).

B-Register Check: The 'B-register check' latch is set when the B-register has an even number of bits set and the B-register check gate is active (MDM 4-10 Part 3).

ALU Check: The 'ALU check' latch is set when an ALU check is detected. An ALU check occurs when a Z-bus plus-bit and minus-bit lines are both plus or minus at the same time. Recognition of an ALU check is blocked for the first cycle of a storage word and for word types 5, 6, and 7.

Storage Data Check: The 'storage data check' latch is set during the second cycle of a storage word when a byte of data read from storage is out of parity (even number of bits) and the parity error is allowed. Storage data check can occur only during a scan or during the data cycle of a word type 2. During word type 2, both bytes are normally checked, but only the byte affected is checked if byte select is on. Byte select depends on the setting of control register bit 7 (MDM 4-10 Part 1).

For a storage word, the storage data register contains the control word during the first cycle. During the second cycle, the storage data register contains the data written into storage (regenerated or replacement). The byte(s) gated for parity error recognition depends on control register bit 7. If control register bit 7 is:

1. equal to 0 (read halfword storage), both regenerated bytes are gated.
2. equal to 1 (read byte storage), only the selected byte is gated.
3. equal to 0 (store halfword storage), both replacement bytes are gated.
4. equal to 1 (store byte storage) only the selected (replacement) byte is gated.

For a scan operation, both bytes are checked.

Control Word Check: If either byte of the storage data register is out of parity and the control cycle latch is on, the control word check latch is set.

Storage Address Check: SAR lines 0 through 14, M1-register parity bit and 7-bit, and M0-register parity bit corrected are combined and checked for even parity. If even parity is detected, the 'storage address check' latch is set.

Storage Protection Check: Storage protection parity check is a check of STP1 bits 0 through 3 and parity for even parity. When the 'allow STP PC' line is active and even parity is detected, the

'storage protection check' latch is set (MDM 4-90).

Hard-Stop Latch

The hard-stop latch has three possible set conditions (Figure 2-17):

1. Any CPU check latch being on with the check control switch at the process position and MMSK register bit 8 or 9 on.
2. Any CPU check latch being on with the check control switch at the stop position.
3. A channel parity error is recognized and the check control switch at the stop position (DR bit 0=0).

When the hard stop latch is set, the clock is stopped at the end of that 900-nanosecond cycle.

The hard-stop latch is reset when one of these keys is pressed.

1. CSL
2. SYSTEM RESET
3. LOAD
4. POWER ON
5. CHECK RESET

Machine-Check Latch

The machine-check latch has two possible set conditions as shown in Figure 2-17. The operation set up by the machine latch being set is shown on MDM 4-10 Part 2.

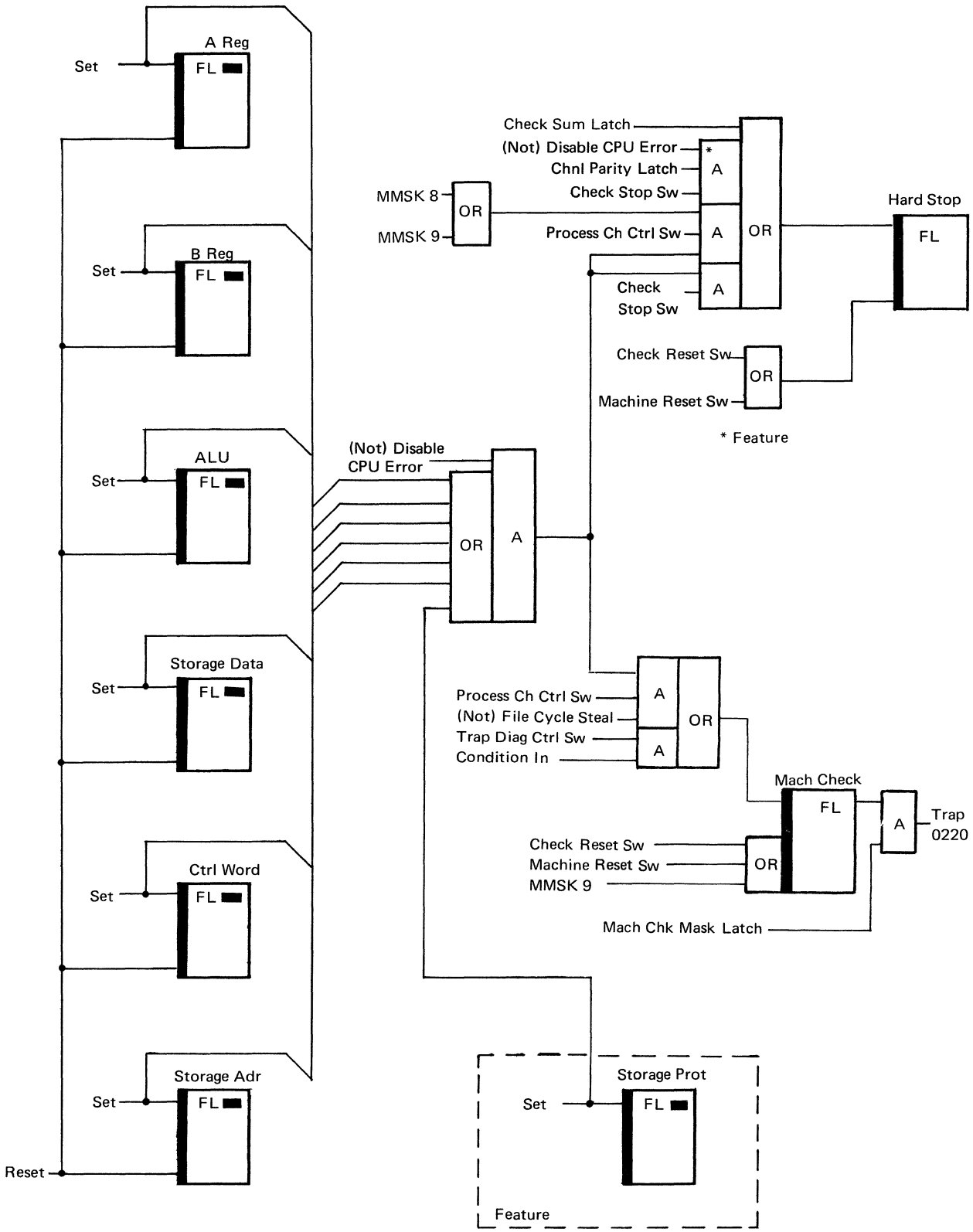


Figure 2-17. CPU Check Latches

MICROPROGRAM CONCEPTS

- The purpose of this section is to:
  1. Define the unique terms and symbols used in microprogram descriptions.
  2. Describe functions of microprograms that reside in the control storage area of the System/360 Model 25.
  3. Describe the tie-in between microprograms and machine language programs.
  4. Describe the control words that make up microprograms.

DEFINITIONS

- Terms and symbols used in the descriptions in this chapter are defined here.

Microprogram: A series of controls words in control storage performing a specific function in the overall operation of the Model 25.

Control Word: A bit-significant halfword that has specific fields defined for controlling data flow of the system. Seven control words are defined for the Model 25:

Set/Reset
Arithmetic Constant
Storage
Move/Arithmetic
Branch Unconditional
Branch on Mask
Branch on Condition

These words are described in detail in this chapter.

Core-Load: A core load is all of the control words and data loaded into the control-storage area at any one time by the control storage load program (CSL).

The system configuration and features list determine the contents of the core load to be used.

Byte-Source: A byte source is any eight-bit facility, either local storage or external, that can be addressed by any of the defined control words.

Certain of the external facilities are less than eight bits, but in operation are handled as byte sources.

High Digit: A high digit is the hex value represented in the high 4-bits of a byte source (bits 0-3).

Low Digit: A low digit is the hex value represented in the low 4-bits of a byte source (bits 4-7).

Microlisting: The source document for the control storage area of main storage. The microlisting contains symbolic information for each microprogram plus the assigned addresses and bit structure of the control words. The microlistings contain the cross reference data needed to trace both backward and forward through the microprograms.

The microlisting is described in detail under Microlistings in this chapter.

Trap Address: A trap address is an assigned fixed address that is the start of a microroutine handling priority operations. These operations are generally I/O or channel-data handling requests.

The trap address is forced by the priority circuits onto the main-storage address lines. The M-register address is not gated out at this time.

Branch Set: A branch set is a table of 4, 8, or 16 control words having the same mnemonic word label but set apart by a leg identifier (hex character). Each control word of a branch set is referred to as a leg of that set. For example, a 4-legged branch set might be labeled:

Branch Set Label	Leg Identifier
CNTRL	0
CNTRL	1
CNTRL	2
CNTRL	3

Branch sets appear sequentially in the listings and are assigned sequential addresses by the micorprogram assembler. The words of a branch set are generally accessed by a Branch on Mask control word.

Link Address: An address stored into a backup register in local storage when leaving a routine because of a trap or a branch-and-link function. This address is generally the address of the next control word that would have been executed had the trap or branch-and-link function not occurred.

The link address is restored to the M-register by a return function of the Set/Reset word.

Figure 3-1. Microroutine Sample

BSTP DESCRIPTIVE TEXT

ENTRY TO THE --BSTP-- ROUTINE IS MADE FROM THE --BSWI-- ROUTINE WHEN AN INSTRUCTION STEP OPERATION OR A SET IC OPERATION IS PERFORMED. THE ENTRY IS MADE TO TEST THE 1052 AND DETERMINE IF THE INSTRUCTION COUNTER CAN BE TYPED OUT.

IF THE 1052 IS NOT BUSY, THE --ALDP-- ROUTINE IS BRANCHED TO, AND THE INSTRUCTION COUNTER IS TYPED OUT.

IF THE 1052 IS BUSY, THE INSTRUCTION COUNTER IS NOT TYPED OUT. THE --BSWI-- ROUTINE IS BRANCHED TO. THE CPU KEY, P-REG, AND I-REG ARE RESTORED AND THE --CICY-- ROUTINE IS BRANCHED TO.

ADDR	WORD	NEXTADDR	NAME	NEXTLABEL	STATEMENT	COMMENTS	STATEMENT NO.
			T		INST TYPEOUT		BSTP 001
0888	2404		TYPADD		SET MODE K=20		BSTP 002
088A	CF91	BSTP 006		BUSY	BR IF TU0=1	CHECK AND BRANCH IF READ LATCH	BSTP 003
088C	DF91	BSTP 006		BUSY	BR IF TU1=1	CHECK AND BRANCH IF WRT LATCH	BSTP 004
088E	CE92	BSTP 007		OK	BR IF TT0=0	CHECK AND BRANCH IF NO ATTEN	BSTP 005
0890	AB20	BSWI 061	BUSY	BSWI RESTRH	BR	NO TYPE OUT WILL OCCUR	BSTP 006
0892	8ACE	CCOM181	OK	CCOM STORH1	BAL	SAVE H1	BSTP 007
0894	4E86				H=I	PUT IC IN H REG	BSTP 008
0896	26C3				D0=0$K0C	BUILD	BSTP 009
0898	2707				D1=0	FLAGS	BSTP 010
089A	2213				V0=0$K01	SET UP COUNT	BSTP 011
089C	96C4	ALDP 157		ALDP ENTRY	BR	GO DISPLAY	BSTP 012

****
* CROSS REFERENCE FOR CSECT BSTP*
****

BSTP 002	BSWI 106	
BSTP 006	BSTP 003	BSTP 004
BSTP 007	BSTP 005	

MICROPROGRAM FUNCTIONS

- Read, decode, and operate on machine language instructions and data located in the program storage area of main storage.
- Handle I/O operations for the Integrated attachments and the channel.
- Handle console operations.
- Handle machine errors.
- Perform diagnostic operations.

The execution of these functions requires many steps in one or more microroutines. The steps in the microroutine are control words that are read out and executed in a particular sequence to perform any of the functions controlled through microprogramming.

MICROLISTINGS

- Each complete control storage load is represented by a microlisting.
- Microlistings contain the actual contents of the control-storage area and the symbolic information that generates the control storage contents.
- Microlistings also contain descriptive text and/or flowcharts of each microroutine.

Each microroutine is labeled with a name comprised of four alphameric characters, the first of which is alpha (Figure 3-1). All microroutines on a microlist are arranged in alphabetic order by routine name. The name of a routine appears on every page of the listing that the routine occupies, and each new routine starts a new page in the listing.

Each line of a microroutine having symbolic information or comments has a sequence number assigned to it. The sequence numbers are not related to the sequence of steps taken by the microroutine, but indicate the order in which the routine was assembled. There may be missing sequence numbers in a routine. This indicates that one or more statements in that routine were selected out because of the particular machine and feature configuration reflected by the listing. The sequence numbers are used in the cross-referencing information for ease of tracing. For each control word on the listing, there is a hex representation of that control word and hex address at which that control word resides in control storage.

Some control words are assigned word labels that appear on the listings. These labels are referenced by other control words that can branch to the labeled word. Refer to MDM 5-14 for microlisting example. Whenever a word is branched-to, its word label appears in the next address-label column of the branch word. If the word branched-to is in another routine, the routine name is given followed by the word label in that routine.

The next address column of the listings gives the routine name and sequence number that is branched-to by the statement in that particular line. If the next address is sequential, the next address column is generally left blank.

Following each microroutine is a cross-reference table. This table lists sequence numbers on the particular routines that are branched-to from some control word other than the previous one. If more than one control word can branch to a particular word in a routine, it is listed in the line with the particular word.

If more words branch to a particular word than can fit on one line, they are listed on sequential lines.

Microprogram, Machine Language Tie-In

A typical operation involving a machine-language instruction and its execution by microroutines is as follows.

1. The machine language instruction resides in the program storage area of main storage.
2. The address of this machine instruction is contained in the instruction counter (I-register of LS zone 0).
3. The machine instruction is read from is decoded by the CICY routine; the proper microroutine that starts the execution of this instruction is selected and branched-to.
5. The microroutine branched-to performs the designated steps called for by the type of instruction being executed. More than one microroutine can be involved in this operation.
6. When all the objectives of the machine-language instruction have been fulfilled, control is turned back to the CICY routine and another machine-language instruction is read from program storage.

Figure 3-2. Bit Significance Chart for Set/Rst Word (Word Type 0)

Word Type			K-High	Set/Rst Source Field				BS/K-Low				K-High			Word Type
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0=RST 1=Set	K-High 0 Bit	0000 = S-Reg 0010 = MMSK 0100 = MODE 0110 = BC 1000 = DR Decodes given are common to all modes. See Figure 3-4 for additional external facilities.				When MMSK addressed for link or return 0000 = U-Reg 0010 = V-Reg 0100 = G-Reg 0110 = D-Reg 1000 = I-Reg 1010 = T-Reg 1100 = P-Reg 1110 = H-Reg When used as K Low Field Bit 8 = KL Bit 4 Bit 9 = KL Bit 5 Bit 10 = KL Bit 6 Bit 11 = KL Bit 7				Bit 12 = KH Bit 1 Bit 13 = KH Bit 2 Bit 14 = KH Bit 3			0

When the MMSK facility is addressed by the Set/Rst Source field, control word bit 11=0 means to perform a link or return function, control word bit 11=1 means no link or return function
The registers and decodes are valid when MMSK is addressed for a link or return function.

CONTROL STORAGE LOAD (CSL)

- The CSL procedure is used to load the control-storage area of core storage.
- This procedure is used whenever control-storage initialization of the system is required: in switching modes, after major system maintenance, for loading microdiagnostics.

Computation and I/O operations are controlled by microprograms residing in the control-storage section of core storage. For these microprograms to be loaded into control storage initially or to be reloaded, the CSL procedure must be followed.

For the CSL procedure to operate correctly, the control program load routine BCPL must be in the control storage area. If the BCPL routine is not resident or has been altered, the hand-load section of BCPL must be entered manually before the CSL procedure is initiated.

The hand-load procedure and contents are described as lead-in text on the listings preceding the BCPL routine.

CSI Using Integrated 2540

1. Set console switches A and B to position E.
2. Press the CSL key.
3. Place the CSL card deck in the 2540 reader and ready the reader.

The CSL indicator remains on until the CSL is completed. Refer to MDM 5-20 for CSL diagram.

CSI Using a Reader on the Standard Interface

1. Place the CSL card deck in the reader and make the reader ready.
2. Set console switches A and B to position C.
3. Set console switches C and D to represent the unit address assigned to the reader.
4. Press the CSL key.

The CSL indicator remains on until CSL is completed.

CSI Using Integrated 2560

1. Set console switches A and B to position D.
2. Place the CSL card deck in the 2560 reader and ready the reader.
3. Press the CSL key.

The CSL indicator remains on until CSL is completed.

SET/RESET WORD

- Defined as word type 0 (control word bits 0, 1, and 15 = 0).
- Sets or resets bits of a designated external facility.
- Stores link address into local storage when used as link word.
- Retrieves link address from local storage when used as return word.

Bit Significance Fields (Figure 3-2)

Bits 0, 1, and 15 are the word-type field and are 0's for the Set/Reset word.

Bit 2 is the reset/set field. If bit 2=0, the bits of the facility addressed corresponding to the bits of the K-field are reset. If bit 2=1, the bits of the facility addressed corresponding to the bits of the K-field are set.

Bits 4, 5, 6, and 7 are the set/reset source field, and designate, by assigned bit values, the facility to be addressed and set or reset by the K-field.

Bits 3, 12, 13, and 14 make up the K-field high 4 bits; bit 3 contains the value of K-high 0 bit; bits 12, 13, and 14 contain the values of K-high bits 1, 2, and 3, respectively.

Bits 8, 9, 10, and 11 normally make up the K-field low 4 bits, or make up the local storage register address when the special link or return function is called for. As the low K-field, bits 8, 9, 10, and 11 contain the value of K-low bits 4, 5, 6, and 7. When used as a local storage register address, bits 8, 9, and 10 address the local-storage register; bit 11 is 0, signifying that a link or return function is to be performed.

Description

The major function of the Set/Reset word is to set or reset specified bits of a designated external facility. The bits to be set or reset are designated by corresponding bits of the K-field of the Set/Reset word. For example:

```
                K-Field = 3A (Hex)
Source bits affected = 0123 4567
                       ↑↑ ↑ ↑
     K-Bit Structure = 0011 1010
```

Most external facilities that can be addressed by the Set/Reset word are affected on a bit basis. Only the bits of the external for which there is a

K Field Use for Externals Affected on a Bit Basis			
	Bit Numbers	0123 4567	
	K Field=	0101 0011	53 (Hex)
SET Function			
	External Status before=	0011 1000	38 (Hex)
	External Status after=	0111 1011	7B (Hex)
	Bit Numbers	0123 4567	
	K Field=	1001 0111	97 (Hex)
SET Function			
	External Status before=	1011 1100	BC (Hex)
	External Status after=	0010 1000	28 (Hex)
K Field Use for Externals Affected on a Byte Basis			
	Bit Numbers	0123 4567	
	K Field=	1010 0111	A7 (Hex)
SET Function			
	External Status before=	0111 1111	7F (Hex)
	External Status after=	1010 0111	A7 (Hex)
	Bit Numbers	0123 4567	
	K Field=	0001 0001	11 (Hex)
SET Function			
	External Status before=	1101 0100	D4 (Hex)
	External Status after=	0001 0001	11 (Hex)

Figure 3-3. K-Field Function Set/Reset Word

corresponding bit in the K-field are set or reset. Refer to Figure 3-3.

Certain external facilities, however, are set directly with the hex value of the K-field.

These externals are always addressed by the set function of the Set/Reset word and are affected on a byte basis.

The set and reset of the external facility, MMSK, requires a slightly different interpretation of the K-field of the Set/Reset word. These exceptions are covered in the following examples.

Examples

These examples show the Set/Reset word as it appears in the statement area of the microlistings, the bit structure assigned by the MAS program, and the description of data handling.

Statement **SET S K=84**

	WT	R/S	KH	S/R Source				KL/BS				KH			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0

The bits specified by the K-field (84) set the corresponding bits of the S-register. The K-field equals 1000 0100 (binary). Therefore, bits 0 and 5 are addressed by the K-field and are set by the set function. The other bits of the S-register are not affected.

Note: Bits 1 and 2 of the S-register cannot be set by the set function of the Set/Reset word but may be reset by the reset function of the Set/Reset word. Refer to MDM 5-1.

Statement **RST S4**

	WT		R/S	KH	S/R Source				KL/BS				KH			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

This statement results in a K-field of 08 in the control word to enable the proper addressing of bit 4 of the S-register. Bit 4 of the S-register is reset by the reset function of this word. The other S-register bits are not affected.

The preceding statement could have been written as RST S K=08 by the microprogrammer, and would have resulted in the same function.

Statement **LINK U MMSK4=1**

	WT		R/S	KH	S/R Source				KL/BS				KH			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0

This statement causes the special link function to be performed. The link statement is generally the first word of a trap routine. The following hardware operations are accomplished by the example statement.

1. The address contained in the M0- and M1-registers is stored in the U-register of local storage zone 4. This address is the address of the control word that would have been executed if the trap had not occurred.
2. The status of bit 3 of the dynamic condition register is stored in bit 7 of the U1-register; the status of bits 6 and 7 of the dynamic condition register is stored into bits 0 and 1 of the U0-register.
3. Latch 4 of the MMSK register is set on, forcing local-storage zone 6 and mode 2. MMSK latch 4 brings up the 2540 punch priority-control lines that remain in control of local storage and external facility addressing until the trap routine is completed and the latch is reset. (Refer to MDM 5-2.)

Statement **RTN V MMSK1=0**

	WT		R/S	KH	S/R Source				KL/BS				KH			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0

This statement causes the special return function of the Set/Reset word to be performed. The return function is generally the last control word of a trap routine and causes a return to the interrupted microroutine.

The following hardware steps are executed by the return function.

1. The address stored in the V-register of local-storage zone 4 is read out and placed in the M0- and M1-registers.
2. The status of bits 3, 6, and 7 of the dynamic condition register, prior to the trap, is retrieved from the V-register of local-storage zone 4 and restored into the dynamic condition register.
3. Latch 1 of the MMSK register is reset, returning the CPU to the mode of operation in control before the trap occurred.

Statement **RTN**

	WT		R/S	KH	S/R Source				KL/BS				KH			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	0	0	1	0	0	1	0	1	0	0	0	1	1	1	0

This statement causes the special return function of the Set/Reset word to be performed. The RTN word is generally used to return to a microprogram that had been branched and linked from, using the address from the CPU backup register. The hardware steps performed by this statement are:

1. The address from the I-register LS zone 4 is read out and placed in the M0-, M1-registers.
2. Dynamic condition register bit 3 is restored from bit 7 of the local storage I1, and bits 6 and 7 are restored from bits 0 and 1 of local storage I0.

Statement **SET MMSK K=31**

	WT		R/S	KH	S/R Source				KL/BS				KH			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	0	1	0	0	0	1	0	0	0	0	1	0	1	1	0

Whenever the MMSK register is addressed by a SET or RST function, the K-field is interpreted in the following manner.

The K-high digit represents the MMSK latch to be operated on.

Bit 11 of the control word is always 1 when the external facility MMSK is addressed and a link or return function is not called for.

The external facilities shown in Figures 3-4 and 3-5 are the assigned externals that may be addressed by the Set/Reset word. Any external facilities not appearing in these figures cannot be addressed by the Set/Reset word.

Set/Reset Source Field	CPU Mode	1403 Mode	1052 Mode	2311 Mode	2540 Mode	Communi-cations Mode	Channel Mode
0000	S	S	S	S	S	S	S
0001							
0010	MMSK	MMSK	MMSK	MMSK	MMSK	MMSK	MMSK
0011							
0100	*MODE	*MODE	*MODE	*MODE	*MODE	*MODE	*MODE
0101							
0110	BC	BC	BC	BC	BC	BC	BC
0111							
1000	*DR	*DR	*DR	*DR	*DR	*DR	*DR
1001					RPD		
1010							
1011				*DIAB	R		*GA
1100				*DIAC			
1101		PRA		FIA	RP	CSETF	GB
1110				FIB			
1111	CPF	PRB	TA	FIC	P	CCTRL	GC

*These externals are addressed by the set function only, and the K-value is set directly into the facility addressed.

Figure 3-4. Set/Reset Word Source Mnemonics

C-Reg Bits 4-7 = (0000) S = K (All Modes)

K-VALUES	SET FUNCTION SET/RESET BIT = 1	RESET FUNCTION SET/RESET BIT = 0
KH Bit 0	Set S0 Latch	Reset S0 Latch
KH Bit 1	Spare *	Reset S1 Latch
KH Bit 2	Spare *	Reset S2 Latch
KH Bit 3	Set S3 Latch	Reset S3 Latch
KL Bit 0	Set S4 Latch	Reset S4 Latch
KL Bit 1	Set S5 Latch	Reset S5 Latch
KL Bit 2	Set S6 Latch	Reset S6 Latch
KL Bit 3	Set Channel-0 Interrupt Latch	Reset Channel-0 Interrupt Latch

* Bit 1 and 2 of the S-Register cannot be set by the Set/Reset word.

C-Reg Bits 4-7 = (0010) MMSK = K (All Modes)

KH Value 0	Set MMSK Bit 0	Reset MMSK Bit 0
KH Value 1	Set MMSK Bit 1	Reset MMSK Bit 1
KH Value 2	Set MMSK Bit2	Reset MMSK Bit 2
KH Value 3	Set MMSK Bit 3	Reset MMSK Bit 3
KH Value 4	Set MMSK Bit 4	Reset MMSK Bit 4
KH Value 5	Set MMSK Bit 5	Reset MMSK Bit 5
KH Value 6	Set MMSK Bit 6	Reset MMSK Bit 6
KH Value 7	Set MMSK Bit 7	Reset MMSK Bit 7
KH Value 8	Set MMSK Bit 8	Reset MMSK Bit 8
KH Value 9	Set MMSK Bit 9	Reset MMSK Bit 9
KH Value A	Spare	Spare
KH Value B	Spare	Spare
KH Value C	Spare	Spare
KH Value D	Spare	Spare
KH Value E	Spare	Spare
KH Value F*	Spare	Spare

* There is no F-latch available. The F-decode is forced for the RTN and LINK functions where the MMSK register is not part of the statement. The F-decode is also forced for communications channel BAL function.

C-Reg Bits 4-7 = (0100) MODE = K* (All Modes)

KH Bit 0	1400 Emulator Mode	
KH Bit 1	Not Assigned	
KH Bit 2	External Gating Control	
KH Bit 3	External Gating Control	* The K-value is set directly into the MODE register
KL Bit 0	External Gating Control	
KL Bit 1	Local Storage Zones	
KL Bit 2	Local Storage Zones	
KL Bit 3	Local Storage Zones	

Figure 3-5. Set/Reset Word External Bit Assignment (Part 1 of 7)

K VALUES	SET FUNCTION SET/RESET BIT = 1	RESET FUNCTION SET/RESET BIT = 0
	C-Reg Bits 4-7 = (0110) BC = K (All Modes)	
KH Bit 0	Set Instruction Step Latch (If in Instruction Step Mode)	Set Soft Stop Latch
KH Bit 1	Set Timer Interrupt	Reset CSL Latch
KH Bit 2		Reset System Reset Latch
KH Bit 3		Reset IC Latch
KL Bit 0		Reset File SP Violation Latch, and CPU Check Latches
KL Bit 1		Reset Console Interrupt
KL Bit 2		Reset IPL Latch
KL Bit 3	Set Logout Latch	Reset Logout Latch
	C-Reg Bits 4-7 = (1000) DR = K** (All Modes)	
KH Bit 0	Disable CPU Error Stop	
KH Bit 1	Force ALU Error	
KH Bit 2	Force Storage Parity	
KH Bit 3	Block all Normal I/O Trap Requests, allow Diagnostic Trap Requests	** The K-value is set drectly into the DR-register
KL Bit 0	Spare	
KL Bit 1	Force Ext to AB	
KL Bit 2	Set CSL Check Indicator	
KL Bit 3	Turns Diagnostic Branch Latch On when set, and a Branch or Return Word Executed	
	C-Reg Bits 4-7 = (1111) CPF = K (CPU Mode)	
KH Bit 0	Set Direct Control Read Enable Latch	Reset Direct Control Read Enable Latch
KH Bit 1	Spare	Spare
KH Bit 2	Spare	Spare
KH Bit 3	Spare	Spare
KL Bit 0	Spare	Spare
KL Bit 1	Spare	Spare
KL Bit 2	Spare	Spare
KL Bit 3	Spare	Spare
	C-Reg Bits 4-7 = (1101) PRA = K (1403 Mode)	
KH Bit 0	Set Print Gate Latch	Reset Hammer Check, PLB Parity, Channel 9, 12 Latches
KH Bit 1	Set Print Request Latch	Reset Print Request, Request Queued, and Ready Request Hold Latches
KH Bit 2	Set Read Control Latch	Spare
KH Bit 3	Set Print Busy Latch	Reset Print Busy Latch
KL Bit 0	Set Request Queued Latch	Spare
KL Bit 1	Spare	Spare
KL Bit 2	Set Block 2 Homes	Force 1403 Machine Reset
KL Bit 3	Reset Printer in Sync Latch Set Diagnostic Write Latch	Spare

Figure 3-5. Set/Reset Word External Bit Assignment (Part 2 of 7)

C-Reg Bits 4-7 = (1111) PRB = K (1403 Mode)

K-VALUES	SET FUNCTION SET/RESET BIT = 1	RESET FUNCTION SET/RESET BIT 0
KH Bit 0	Set Single Cycle Mode Latch	Spare
KH Bit 1	Reset Single Cycle Mode Latch	Set Single Cycle Clock Run Latch
KH Bit 2	Spare	Spare
KH Bit 3	Set S7 Diagnostic Gate Latch (S7 Request Latch)	Spare
KL Bit 0	Reset S7 Diagnostic Gate Latch	Set Diagnostic PSS Pulse Latch Set PSS Gate
KL Bit 1	Set Diagnostic Decode-2 Latch	Reset Diagnostic PSS Pulse Latch Reset PSS Gate
KL Bit 2	Set Diagnostic Decode-3 Latch	Spare
KL Bit 3	Set Diagnostic Decode-4 Latch	Spare

C-Reg Bits 4-7 = (1111) TA = K (1052 Mode)

KH Bit 0	Set Read Latch	Reset Read Latch
KH Bit 1	Set Write Latch	Reset Write Latch
KH Bit 2	Set Microforce Latch	Reset Microforce Latch
KH Bit 3	Set Alter/Display Active	Reset Alter/Display Active
KL Bit 0	Spare	Spare
KL Bit 1	TA Diag SS Set	TA Attach Reset
KL Bit 2	Initialize Printer	Share Reset
KL Bit 3	Force Share Request	Attention Reset

C-Reg Bits 4-7 = (1101) FIA = K (2311 Mode)

KH Bit 0	Set GO Latch	Reset Trap Latch
KH Bit 1	Set Interrupt Latch	Reset Interrupt Latch
KH Bit 2	Set CCW Count Zero	Reset PCI Latch
KH Bit 3	Set CUB Latch	Reset CUB Latch
KL Bit 0	Spare	Spare
KL Bit 1	Spare	Spare
KL Bit 2	Spare	Spare
KL Bit 3	Spare	Spare

C-Reg Bits 4-7 = (1110) FIB = K (2311 Mode)

KH Bit 0	Set Not Oriented	Chain End Reset
KH Bit 1	Set Flag Reg to Zero	Initial Reset
KH Bit 2	Spare	Cold Start Reset
KH Bit 3	Spare	Spare
KL Bit 0	Spare	Spare
KL Bit 1	Spare	Spare
KL Bit 2	Spare	Spare
KL Bit 3	Spare	Spare

Figure 3-5. Set/Reset Word External Bit Assignment (Part 3 of 7)

C-Reg Bits 4-7 = (0011) DIAC = K* (2311 Mode)

K-VALUES	SET FUNCTION SET/RESET BIT = 1	RESET FUNCTION SET/RESET BIT = 0
KH Bit 0	+ Clock Bit (Read)	
KH Bit 1	Write Phase A	
KH Bit 2	Selected Index	*The K-value is set directly into the DIAC facility.
KH Bit 3	Diagnostic Mode	
KL Bit 0	Read Data Bit	
KL Bit 1	Set Mode Latch (1401)	
KL Bit 2	Erase Gate Reset	
KL Bit 3		

C-Reg Bits 4-7 = (0001) DIAB = K* (2311 Mode)

KH Bit 0	Gate Diag Addr 0	
KH Bit 1	Gate Diag Addr 1	
KH Bit 2	Gate Diag Addr 2	
KH Bit 3	Gate Diag Addr 3	
KL Bit 0	Gate Diag Addr 4	*The K-value is set directly into the DIAB facility
KL Bit 1	Gate Diag Addr 5	
KL Bit 2	Gate Diag Addr 6	
KL Bit 3	Diagnostic Compare Gate	

C-Reg Bits 4-7 = (1111) FIC = K (2311 Mode)

KH Bit 0	Set Move Latch (1401)	Reset Move Latch (1401)
KH Bit 1	Spare	Spare
KH Bit 2	Spare	Spare
KH Bit 3	Spare	Spare
KL Bit 0	Spare	Spare
KL Bit 1	Spare	Spare
KL Bit 2	Spare	Spare
KL Bit 3	Spare	Spare

C-Reg Bits 4-7 = (1001) RPD = K (2540 Mode)

KH Bit 0	Set R/P Diagnostic Latch	Reset R/P Diagnostic Latch
KH Bit 1	Set R/P Diagnostic Impulse Latch	Reset R/P Diagnostic Impulse Latch
KH Bit 2	Set R/P Diagnostic Single Cycle Latch	Reset R/P Diagnostic Single Cycle Latch
KH Bit 3	Spare	Reset R/P Reset Shift Register
KL Bit 0	Spare	Reset R/P Diagnostic Machine Reset
KL Bit 1	Spare	Spare
KL Bit 2	Spare	Spare
KL Bit 3	Spare	Spare

Figure 3-5. Set/Reset Word External Bit Assignment (Part 4 of 7)

K-VALUES	SET FUNCTION SET/RESET BIT = 1	RESET FUNCTION SET/RESET BIT = 0
	C-Reg Bits 4-7 = (1011) R = K (2540 Mode)	
KH Bit 0	Set Reader Equipment Check Gate Latch	Reset Reader Validity and Reader Equipment Latches (Reader Sense Byte)
KH Bit 1	Set Read Feed Latch	Spare
KH Bit 2	Set Stacker Select R2 Command Latch	Spare
KH Bit 3	Set Stacker Select R3 Command Latch	Spare
KL Bit 0	Set Reader Validity Latch	Spare
KL Bit 1	Set Reader Equipment Sense Latch (Provided Reader Equipment Check Gate Latch On)	Spare
KL Bit 2	Set Reader Status Latch	Reset Reader Status (Reader DE, Reader Status and Reader Queued Latches)
KL Bit 3	Set Reader Queued Latch	Spare
	C-Reg Bits 4-7 = (1101) RP = K (2540 Mode)	
KH Bit 0	Unit Exception Latch	Spare
KH Bit 1	1400 Latch Causing 1400 Delayed Feed	Spare
KH Bit 2	Reader Command Interlock Latch	Reset Reader Command Interlock Latch
KH Bit 3	Reader Overrun Interlock 1 and 2 Latches Reader-Overrun Latch (Set Reader Mach Check)	Spare
KL Bit 0	Punch Command Interlock Latch	Reset Punch Command Interlock Latch
KL Bit 1	Punch Overrun Interlock 1 and 2 Latches Punch Overrun Latch (Set Punch Mach Check)	Spare
KL Bit 2	Spare	Spare
KL Bit 3	Spare	Spare
	C-Reg Bits 4-7 = (1111) P = K (2540 Mode)	
KH Bit 0	Punch Equipment Check Latch	Reset PFR Validity and Punch Equipment Check Latch (Punch Sense Byte)
KH Bit 1	Punch Feed Latch	Spare
KH Bit 2	Stacker Select P2 Command Latch	Spare
KH Bit 3	Stacker Select P3 Command Latch	Spare
KL Bit 0	PFR Validity Latch	Spare
KL Bit 1	PFR Restart Gate Latch	Spare
KL Bit 2	Punch Status Latch	Reset (Punch Device End Latch, Punch Status Latch and Punch Queued Latch.)*
KL Bit 3	Punch Queued Latch	Spare

* K=low Bit 2 and Not Punch Intervention Latch sets Block Punch Initial Ready Latch

	C-Reg Bits 4-7 = (1101) CSETF = K (Start/Stop) (Communications Mode)	
KH Bit 0	Restart Scanner	ICC Reset
KH Bit 1	Spare	Spare
KH Bit 2	Spare	Spare
KH Bit 3	Set Start/Stop Character Trap	Reset Start/Stop Character Trap
KL Bit 0	Set Start/Stop Chain Trap	Reset Start/Stop Chain Trap
KL Bit 1	Set Timeout Enable and Remember	Reset Timeout Enable and Remember
KL Bit 2	Set Timeout Trap (Diagnose)	Reset Timeout Trap and Remember
KL Bit 3	Set ICC I/O Interrupt Indicator	Reset ICC I/O Interrupt Indicator

Figure 3-5. Set/Reset Word External Bit Assignment (Part 5 of 7)

K-VALUES	SET FUNCTION SET/RESET BIT = 1	RESET FUNCTION SET/RESET BIT = 0
	C-Reg Bits 4-7 = (1101) CSETF = K-Synchronous (Communications Mode)	
KH Bit 0	Restart Scanner	ICC Reset
KH Bit 1	Set Sync Char Trap Latch and Chain Trap Latch	Reset Sync Char Trap
KH Bit 2	Set Sync Chain Trap and Reset Sync Char Trap	Reset Sync Char Trap, Chain Trap, and Prepare CMD Latch
KH Bit 3	Set Start/Stop Character Trap	Reset Start/Stop Character Trap
KL Bit 0	Set Start/Stop Chain Trap	Reset Start/Stop Chain Trap
KL Bit 1	Set Timeout Enable	Reset Timeout Trap, Enable, and Remember
KL Bit 2	Set Timeout Trap (Diagnose)	Reset Timeout Request and Remember
KL Bit 3	Set ICC I/O Int Indicator	Reset ICC I/O Int Indicator
	C-Reg Bits 4-7 = (1111) CCTRL = K Start/Stop (Communications Mode)	
KH Bit 0	Spare	Selective Reset
KH Bit 1	Set LA Transmit Latch	Reset LA Transmit Latch
KH Bit 2	Set LA Enable Latch	Reset LA Enable Latch
KH Bit 3	Set LA Data Set Ready Trap Latch (Diagnose)	Reset LA Data Set Ready Trap Latch (Diagnose)
KL Bit 0	Start Strobe	Stop Strobe
KL Bit 1	Set Diag Control Ltach (RECV) Set Fractional Stop Latch (XMIT)	Reset Fractional Stop Bit Overflow, Diag. Control
KL Bit 2	Set Receive End Latch	Reset Receive End Latch and Set Line Quiet Latch
KL Bit 3	Spare	Spare
	C-Reg Bits 4-7 = (1111) CCTRL = K-Synchronous (Communications Mode)	
KH Bit 0	Set Test Mode Latch	Selective Reset
KH Bit 1	Set LA Transmit Latch	Reset LA Transmit Latch
KH Bit 2	Set LA Enable	Reset Enable and Test Mode
KH Bit 3	Set Prepare Command Latch	Data Set Ready Request
KL Bit 0	Set Character Phase (Diagnose)	Reset Character Phase (Diagnose)
KL Bit 1	Set 1-Second Timeout	Reset 1-Second Timeout
KL Bit 2	Set 3-Second Timeout	Reset 3-Second Timeout
KL Bit 3	Set Interface B	Reset Interface B
	C-Reg Bits 4-7 = (1011) *GA = K (Channel Mode)	
KH Bit 0	Selective Reset (Note 1)	Note 1: This function yields a Selective Reset Sequence, ending with Suppress-Out and Operational-Out both up
KH Bit 1	Service Out	
KH Bit 2	Address Out	
KH Bit 3	Command Out	
KL Bit 0	Initial Select	
KL Bit 1	Select Out	
KL Bit 2	Channel Reset (Diagnostic Only)	
KL Bit 3	Spare	

*Directly replaced by K-value; Set/Reset Bit will be 1

Figure 3-5. Set/Reset Word External Bit Assignment (Part 6 of 7)

C-Reg Bits 4-7 = (1101) GB = K (Chnanel Mode)

K-VALUES	SET FUNCTION SET/RESET BIT = 1	RESET FUNCTION SET/RESET BIT = 0
KH Bit 0	Set Data Chain Request Latch	Reset Data Chain Request Latch
KH Bit 1	Set Channel Identification Latch: Selector	Set Channel Identification Latch: Multiplexer
KH Bit 2	Set Burst Latch	Reset Burst Latch and Buffered Device Latch
KH Bit 3	Set Buffered Device Latch	Reset Channel Parity Check Latch
KL Bit 0	Set Channel Diagnostic Latch	Reset Channel Diagnostic Latch
KL Bit 1	Set Channel 1-Interrupt Latch	Reset Channel 1-Interrupt Latch
KL Bit 2	Set Chain Detect Latch	Spare
KL Bit 3	Set Suppress-Out Latch	Reset Suppress-Out Latch

Figure 3-5. Set/Reset Word External Bit Assignment (Part 7 of 7)

ARITHMETIC CONSTANT WORD

- Defined as word type 1 (control word bits 0, 1, and 15 = 001).
- Performs an arithmetic operation using a byte from local storage and a K-value from the control word.

Bit Significance Fields (Figure 3-6)

Bits 0, 1, and 15 are the word type field and are set to 0, 0, 1, respectively.

Bits 2, 3, 12, 13, and 14 make up the functional decode. This field is set according to the statement the microprogrammer writes. The functional decode determines the ALU operation, the A- and B-register gating, and the destination of the ALU result.

Bits 4, 5, 6, and 7 make up the A-source field and indicate which byte of a designated local-storage register is gated to the A-register of ALU.

Bits 8, 9, 10, and 11 make up the K-field and contain the hex digit designated in the statement written by the microprogrammer. The K-field is gated to both the high and low 4 bits of the B-register. The manner in which the K-field is written in the statement determines how the K-value is gated from the B-register.

Figure 3-6. Bit Significance Chart for Arithmetic Constant Word (Word Type 1)

Word Type		Function Decode		A-Source				K-Field				Function Decode			Word Type
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	00		0000 = U0				0000				000 Z=A + KL			1
		00		0001 = U1				0001				001 Z=A,A−, KL			
		00		0010 = V0				0010				010 Z=A,A−, KH			
		00		0011 = V1				0011				011 Z=A,A−, KK			
		00		0100 = G0				0100				100 Z=A + KH			
		00		0101 = G1				0101				101 Z=A, OE, KL			
		00		0110 = D0				0110				110 Z=A,OE,KH			
		00		0111 = D1				0111				111 Z=A,OE,KK			
		01		1000 = I0				1000				000			
		01		1001 = I1				1001				001 A=A,A−,KL			
		01		1010 = T0				1010				010 A=A,A−, KH			
		01		1011 = T1				1011				011 A=A,A−,KK			
		01		1100 = P0				1100				100			
		01		1101 = P1				1101				101 A=A,OE,KL			
		01		1110 = H0				1110				110 A=A,OE,KH			
		01		1111 = H1				1111				111 A=A, OE, KK			
		10		The Function decode is a combination of control word bits 2, 3, 12,13, and 14.								000			
		10										001 A=0,OR,KL			
		10										010 A=0,OR,KH			
		10										011 A=0,OR,KK			
		10										100			
		10										101 A=A + KL			
		10										110 A=A + KH			
		10										111 A=A + KK			
		11		The A-symbol used in columns 12, 13, and 14 can be any of the local storage byte sources listed under the A-source field.								000			
		11										001 A=A,OR,KL			
		11										010 A=A,OR,KH			
		11										011 A=A,OR,KK			
		11										100 A=0−KH			
		11										101 A=A−KL			
		11										110 A=A−KH			
		11										111 A=A−KK			

Examples

These examples show some arithmetic constant words as they appear in the statement area of the microlistings, the bit structure assigned by the MAS program, and the description of the data handling.

Statement **V1=V1+KAA**

	WT		FD		AS				K				FD			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	0	1	0	0	0	1	1	1	0	1	0	1	1	1	1

The contents of local-storage byte V1 is gated to the A-register of ALU. The K-value of A is gated to both the high and low 4 bits of the B-register of ALU. The values in A and B are added together, and the result is gated back to local-storage byte V1. Refer to MDM 5-4.

Statement **Z=I1*-K0E**

	WT		FD		AS				K				FD			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	0	0	0	1	0	0	1	1	1	1	0	0	0	1	1

The contents of local-storage byte I1 is gated to the A-register input of ALU. A K-value of E is gated to both the high and low 4 bits of the B-register. The functional decode gates the low four bits of the B-register to the ALU and complement-ANDs them to the contents of the A-register. The result is placed on the Z-bus to set the dynamic condition register with the ALU result. Refer to MDM 5-5.

Statement **G1=0$K77**

	WT		FD		AS				K				FD			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	0	1	0	0	1	0	1	0	1	1	1	0	1	1	1

The contents of G1 is gated to the A-register of ALU. The K-value of 7 is gated to both the high and low 4 bits of the B-register. The functional decode:

1. blocks the gate of the A-register to ALU
2. gates the B-register to ALU
3. ORs the B-register contents with a 0-value from the A-register output.

The result of the OR (hex 77) is gated to the G1 byte of local storage.

Statement **V1=V1-K33**

	WT		FD		AS				K				FD			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	0	1	1	0	0	1	1	0	0	1	1	1	1	1	1

The contents of V1 are gated to the A-register of ALU. The K-value 3 is gated to both the high and low 4 bits of the B-register. The functional decode:

1. gates the A-register straight to the adder
2. causes the complement of the B-register contents to be gated to the adder
3. causes an effective add of the A-register and complemented B-register contents
4. gates the result back to the V1-register.

STORAGE WORD

- Defined as word type 2 (bits 0, 1, and 15 = 0, 1, 0 respectively).
- Only word able to access program storage.
- The operations of the storage word are to:
  1. read data (byte or halfword) from program, control, or auxiliary storage
  2. store data (byte or halfword) into program, control, or auxiliary storage
  3. do halfword updates of storage addresses or data counts
  4. move halfwords from one pair of local registers to another.

Word Type	Storage Control	Data Register Address	Address Register Address or K-Field	Modifier Control	Word Type
0 1	2 3	4 5 6 7	8 9 10 11	12 13 14	15
0	00=Read Control	This field may address Local Storage for a byte or halfword access.	For Indirect Addressing	000=No update	0
1	01=Read Aux/Prog	Byte selection of Local Storage is limited to the odd address, where control word bit 7=1.	The Address Register selected by this field is a halfword Local-Storage Register	001=Direct Address	
	10=Store Control	Halfword selection is limited to even addresses, where bit 7=0	Bit 11 does not enter into the addressing of the address register.	010=No access, + update	
	11=Store Aux/Prog	External facility addressing is always done in byte mode, where control word bit 7=1.	Bit 11 designates the following:	011=No access, – update	
			Bit 11=0 Bits 2,3=01 — Read Aux	100=LS Data Reg,Access, + update	
			Bit 11=1 Bits 2,3=01 — Read Prog	101=LS Data Reg,Access, – update	
			Bit 11=0 Bits 2,3=11 — Stor Aux	110=Ext Data Reg,Access, + update	
			Bit 11=1 Bits 2,3=11 — Store Prog	111=Ext Data Reg,Access, – update	

When no access to storage is designated, modifier control decode of 010 or 011, Bits 2 and 3 determine the value to update by:

Bits 2 and 3		Value
00	=	0
01	=	±1
10	=	±2
11	=	±3

Local Storage

*	0000	U0
	0001	U1
*	0010	V0
	0011	V1
*	0100	G0
	0101	G1
*	0110	D0
	0111	D1
*	1000	I0
	1001	I1
*	1010	T0
	1011	T1
*	1100	P0
	1101	P1
*	1110	H0
	1111	H1

* Even registers can be used as address registers (ctrl bits, 8, 9, 10, 11)

For Direct Address

Bits 8, 9, 10, and 11 are forced to values that will result in the K-address for the M1-Register. The M1-Register is forced to the following values for the K-field values indicated:

Bits 8, 9, 10, 11	M1-Register in Hex
0000	88
0001	8A
0010	8C
0011	8E
0100	98
0101	9A
0110	9C
0111	9E
1000	A8
1001	AA
1010	AC
1011	AE
1100	B8
1101	BA
1110	BC
1111	BE

M0 = X0 for direct access of Auxiliary Stoarage

M0 = X3 for direct access of Control Storage

Figure 3-7. Bit Significance Chart for Storage Word (Word Type 2)

Bit Significance Fields (Figure 3-7)

Bits 0, 1, and 15 are the word-type field and are set to 0, 1, 0 respectively to designate word type 2.

Bits 2 and 3 designate the storage operation to be performed.

During indirect addressing of storage, bits 2 and 3 work in conjunction with bit 11 to determine the operation. Indirect addressing means accessing storage by using the contents of a specified LS-register to load the M-register.

Bits 2 & 3	Bit 11	Function
00	0	Read Control Storage
01	1	Read Program Storage
01	0	Read Auxiliary Storage
10	0	Store Control Storage
11	1	Store Program Storage
11	0	Store Aux Storage

When storage is not accessed, bits 2 and 3 determine the value of the address update to be performed.

Bits 2 and 3	Update
00	0
01	+ or -1
10	+ or -2
11	+ or -2

Bits 12, 13, and 14 determine the update function (+ or -).

Bits 4, 5, and 6 make up the data-register address field. During a read operation, the local-storage register or external facility that receives the data is designated in this field.

During a store operation, this field indicates the local-storage register or external facility that contains the data to be stored.

During a move or update-only function, this field designates the local-storage register that receives halfword of data.

Bit 7 is the byte-select field. This bit determines whether a byte or halfword data operation is to be performed.

When bit 7=1 during a read operation, the single byte specified by the M-register is placed into the odd data-register location addressed by bits 4-7.

When bit 7=1 during a store operation, the single byte from the odd data-register location addressed by bits 4-7 is stored at the byte address specified by the M-register.

When bit 7=0 during a read operation, the halfword of data addressed by the M-register is placed into the halfword data register specified by bits 4, 5, and 6.

When bit 7=0 during a store operation, the halfword of data from the data register specified by bits 4, 5, and 6 is stored at the halfword location addressed by the M-register.

Note: Bit 7 is always 1 when external facilities are addressed.

Bits 8, 9, and 10 make up the address-source field. During indirect addressing, this field designates the register in local storage that contains the storage address to be used in the operation.

When a no-access operation is indicated, this field designates the local-storage register that contains the data or address to be updated and/or moved.

When a direct address operation is indicated, this field, along with bit 11, is used to develop the effective low-order byte address for the M1-register in the following manner.

M1 Register	Bits
0	forced to a 1
1	forced to a 0
2	Control register bit 8
3	Control register bit 9
4	forced to a 1
5	Control register bit 10
6	Control register bit 11
7	forced to a 0

M0=00 for direct access of auxiliary storage; M0=03 for direct access of control storage.

Bits 12, 13, and 14 make up the modifier control field. The bit combinations and the functions designated are as follows.

Bits 12-14: 000
Function: (0)
Designation: An access of storage is made, but no address update is designated.

Bits 12-14: 001
Function: (M=K)
Designation: Direct addressing of storage is performed. The address for the M1-register is developed from bits 8, 9, 10, and 11 and forced status.

Bits 12-14: 010
Function: (No ACC +)
Designation: No storage access is done. A plus address update is performed using the value specified in bits 2 and 3.

Bits 12-14: 011
Function: (No ACC -)
Designation: No storage access is done. A minus address update is performed, using the value specified in bits 2 and 3. This decode is also used when a 'no storage access, no update' function is designated.

Bits 12-14: 100
Function: (STOR LS+)
Designation: An access of storage is made. The data is read from, or stored into, some local-storage register. A plus update is performed using a value implied by the operation. For instance, halfword operations imply an update by a value of 2; byte operations imply an update by a value of 1.

Bits 12-14: 101
Function: (STOR LS-)
Designation: An access of storage is made. The date is read from, or stored into, some local storage register. A minus update is performed using an implied value. For instance, halfword operations imply an update by a value of 2; byte operations imply an update by a value of 1.

Bits 12-14: 110
Function: (STOR EXT+)
Designation: An access of storage is made. The data is read from, or stored into, some external facility. A plus update is performed using implied values as explained for decode 100.

Bits 12-14: 111
Function: (STOR EXT-)
Designation: An access of storage is made. The data is read from, or stored into, some external facility. A minus update is performed using implied values as explained for decode 101.

Description

The primary function of the storage word is to move data from/to program, control, or auxiliary storage, to/from an external facility or a local-storage register. The read or store operations that access local storage may be either byte or halfword operation. The read or store operations that access external facilities can be only byte operations.

Program storage is addressed only in an indirect manner, using the contents of the halfword local-storage register specified as the address register. This indirect address can be updated by values of 1 or 2, depending on the operation being performed.

Control and auxiliary storage can be addressed either indirectly (the same as program storage) or directly. The direct addressing operation is called for by the statement format and is described fully in the storage word examples.

A function (called double-byte modify) is also available to update and move halfword data from one local-storage register to another, or to update the halfword data in a local-storage register only.

Examples

These examples show some of the storage words as they appear in the statement area of the microlistings, the bit structure assigned by the MAS program, and the description of the data handling.

Statement **RDH U V**

	WT		SC		DATA			BS	ADDR			MS	MC			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	1	0	1	0	0	0	0	0	0	1	1	0	0	0	0

The operation indicated by the statement is a read halfword (RDH) from some program storage location addressed by the contents of the address register (V). The halfword read from program storage is placed in the data register (U) of local storage.

No address update is called for by the statement.

Statement **RDH G V+2**

	WT		SC		DATA			BS	ADDR			MS	MC			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	1	0	1	0	1	0	0	0	0	1	1	1	0	0	0

The operation indicated by the statement is a read halfword (RDH) from some program storage location addressed by the contents of the address register (V). The halfword read from program storage is placed in the data register (G) of local storage.

An update of the address register contents is performed (V+2) after the read operation is complete.

Statement **STB D1 AS,G**

	WT		SC		DATA			BS	ADDR			MS	MC			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0

The operation indicated by the statement is a store byte (STB) into auxiliary storage (AS) addressed by the contents of address register (G). The data byte to be stored is taken from the data register byte 1 (D1) of local storage.

No address update is called for by this statement.

Statement **RDB U1 CS,H-1**

	WT		SC		DATA			BS	ADDR			MS	MC			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	1	0	0	0	0	0	1	1	1	1	0	1	0	1	0

The operation indicated by the statement is a read byte (RDB) from some control storage (CS) location addressed by the contents of the address register (H). The byte read out of control storage is placed into byte 1 of the data register (U1) in local storage.

The contents of the address register (H) is decremented by 1 as indicated in the statement (H-1). The address is updated after the read operation is complete.

Statement **D=D-2**

	WT		SC		DATA			BS	ADDR			MS	MC			WT
	0	1	2	3	4	5	6	7	8	9	10	12	12	13	14	15
Bit Structure	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0

The operation indicated by the statement is a no-access, update and move (double-byte modify). The contents of the D-register of local storage are decremented by 2 and the result is moved back to the D-register. Refer to MDM 5-7 for an example of a double-byte modify.

Statement **V=U**

	WT		SC		DATA			BS	ADDR			MS	MC			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0

The operation indicated by the statement is a no-access move. The contents of the U-register of LS are moved and placed into the V-register of LS.

The MC-field bits assigned by the MAS program indicate a no-access minus-update function; however, bits 2 and 3 indicate an update value of 0, effectively producing a no-update function.

Statement **RDH U DA,BA**

	WT		SC		DATA			BS	ADDR			MS	MC			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	1	0	1	0	0	0	0	1	1	0	1	0	0	0	0

The operation indicated by the statement is a read halfword (RDH) from auxiliary storage, directly addressed (DA) using the K-value BA for the M1-register. The M0-register is set to 0, and hardware gating is brought up to address the CPU auxiliary storage area. Refer to MDM 4-20 for M-register gating.

Statement **RDB GB/OUT CS,H-1**

	WT		SC		DATA			BS	ADDR			MS	MC			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	1	0	0	1	1	1	1	1	1	1	0	1	1	1	0

The operation indicated by the statement is a read byte (RDB) from a control storage location (CS) indirectly addressed by the contents of the address register (H). The data byte read from control storage is gated to the external facility GB/OUT. The contents of the address register (H) are decremented by 1 after the read operation has been performed.

Figure 3-8. Bit Significance Chart for Move/Arithmetic Word (Word Type 3)

Word Type		Functional Decode		A-Source Field				B-Source Field				Functional Decode			Word Type
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	00		0000 = U0 0001 = U1 0010 = V0 0011 = V1 0100 = G0 0101 = G1 0110 = D0 0111 = D1				0000 = U0 0001 = U1 0010 = V0 0011 = V1 0100 = G0 0101 = G1 0110 = D0 0111 = D1				000 001 B=AXH+BL 010 B=AXL,OR,BH 011 100 B=A + B 101 B-AH + BL 110 B=AL + BH 111 EXT=B			1
		01		1000 = I0 1001 = I1 1010 = T0 1011 = T1 1100 = P0 1101 = P1 1110 = H0 1111 = H1				1000 = 10 1001 = 11 1010 = T0 1011 = T1 1100 = P0 1101 = P1 1110 = H0 1111 = H1				000 B=AX 001 B=AXH 010 B=AXL 011 B=0 (STOP) 100 B=A 101 B=AH 110 B=AL 111 B=EXT			
		10		If Functional Decode (Bits 2, 3, 12, 13, and 14) Equal 00111, the AS-field addresses an external byte facility				If Functional Decode (Bits 2, 3, 12, 13, and 14) equal 01111, the AS-field addresses an external byte facility				000 A=A,OE,B 001 A=A + B 010 A=A,OR,B 011 A=A,A,B 100 AC=A + B + 1 101 AC=A + B 110 AC=A + B + C 110 AC=AL + B + C			
		11		The Function decode is a combination of control word bits 2,3,12,13 and 14. The-A symbol used in columns 12, 13, and 14 can be any of the local-storage byte sources listed under the A-source field. The B-symbol used in columns 12, 13, and 14 can be any of the local-storage byte sources listed under the B-source field The C-symbol used in columns 12, 13 and 14 denotes adder carry or S3 latch.								000 A=A -B + 1 001 A=A-B 010 011 100 AC=A-B + C 101 AC=0-B + C 110 A=A +-B + C 111 AC=A @ B + C			

Note: There are storage bit configurations forced by circuitry into the control register for 2311 file-share operations. The functions and descriptions of the forced storage words are covered in the IBM 2311 Disk Attachment FETOM, Form Y24-3534.

MOVE/ARITHMETIC WORD

- Defined as word type 3 (control word bits 0, 1, and 15 = 011).
- Performs arithmetic operations on local storage data.
- Moves data from local storage to external facilities.
- Moves data from external facilities to local storage.

Bit Significance Fields (Figure 3-8)

Bits 0, 1, and 15 are the word type field and are set to 0, 1, 1 respectively to indicate word type 3.

Bits 2, 3, 12, 13, and 14 make up the functional decode field. The bit structure assigned to this field relates to the hardware, the gating for the ALU inputs and outputs, and the operation that the ALU performs.

Bits 4, 5, 6, and 7 make up the A-source field decode. This field represents the local-storage byte that is gated to the A-register of ALU for arithmetic operations and for moves from local storage to local storage. For moves involving an external facility, this field represents the external facility address, whether the external is a source or a destination.

Note: For moves involving an external facility, the B-register alone is used to route data through the ALU to the LS-data assembler.

Bits 8, 9, 10, and 11 make up the B-source field. For arithmetic operations, this field represents the local-storage byte gated to the B-register of ALU. For move operations, it represents either the local-storage source or destination byte.

The functions of the Move/Arithmetic word are as follows.

1. Perform arithmetic operations on data from local-storage byte sources. Arithmetic operations that can be performed are:
   - ¤ Exclusive OR
   - + Add
   - - Subtract
   - $ CR
   - * AND
   - *- Complement AND
   - @ Decimal add (S0=0 means add, S0=1 means subtract)
   - +- Binary add (S0=0 means add, S0=1 means subtract).
2. Move data from one local-storage byte address to another. There are times when a microprogrammer writes a statement that effectively moves data from a local-storage byte location to itself; i.e., V1=V1, G0=G0. This places the contents of the byte source on the Z-bus of ALU. Here it is sampled, and the status of the Z-bus is set into the dynamic condition register.
3. Move a data byte from local storage to an external facility. The local-storage data byte is gated to the AB-assembler, the B-register, through the ALU to the Z-bus, to the local storage data assembler, to the external bus-out, and finally to the external facility.
4. Move the data from an external facility to a local-storage byte location. The external data is gated on the external bus-in lines to the AB-assembler, to the B-register, through the adder onto the Z-bus, to the local-storage data assembler, then to the local-storage byte location specified.

Examples

These examples show some of the Move/Arithmetic words as they appear in the statement area of the microlistings. The bit structure assigned by the MAS program for the statement, and the results of the word being executed, are also given.

Statement **U1C=U1+H0**

	WT		FD		AS				BS				FD			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	1

The contents of local-storage byte U1 are gated to the A-register. The contents of local-storage byte H0 are gated to the B-register. A binary add is performed on the two bytes, and the result is gated back to the U1 byte of LS.

A carry from the 0-bit of the adder sets latch 3 of the S-register. If no carry occurs, latch 3 of the S-register is reset.

This function is designated by a C written in the statement following the destination byte mnemonic. Refer to MDM 5-9.

The dynamic condition register also is set to indicate the state of the ALU result and the adder carry.

Statement **GB/OUT=T1**

	WT		FD		AS				BS				FD			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	1	1	1	0	1	0	0	0	1	0	0	1	1	1	1

The contents of LS byte T1 are gated through the AB-assembler, B-register, ALU, on the Z-bus to the LS-data assembler, out on the external bus-out, to the external facility named GB/OUT.

For the external facility GB/OUT to be addressed, the MODE register must be set to indicate channel mode, or the MMSK register must be set to indicate channel high or low priority.

Statement **G0C=G0@G0+C**

	WT		FD		AS				BS				FD			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	1	1	1	0	1	0	0	0	1	0	0	1	1	1	1

The contents of LS byte G0 are gated to both the A-register and B-register. A decimal add or subtract, depending on the setting of S0, is performed on the A- and B-register contents. The status of the S3 latch is added in with the values from the A- and B-registers, and the S3 latch is set by the resultant adder carry. If the S3 latch is on and no carry occurs, the S3 latch is reset.

The dynamic condition register is set with the results of the arithmetic operation.

Statement **STOP**

	WT		FD		AS				BS				FD			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	1

This special function word is used to stop the CPU clock at the end of the control-word cycle. The manipulation of the data flow caused by this word is not significant; it is just allowed to happen.

Byte 0 of the U-register of LS is gated to both the A- and B-registers. The A-register is blocked and does not enter the ALU circuits. The B-register is gated straight through the ALU circuits to the Z-bus, to the LS-data assembler, and then is written back into byte 0 of the U-register.

At the end of this cycle, the CPU clock is stopped.

BRANCH UNCONDITIONAL WORD

- Defined as word type 4 (control word bits 0, 1, and 15 = 1, 0, 0 respectively).
- Branches unconditionally to any designated address in control storage.
- Stores the next sequential address in the CPU backup register, I in LS zone 4.

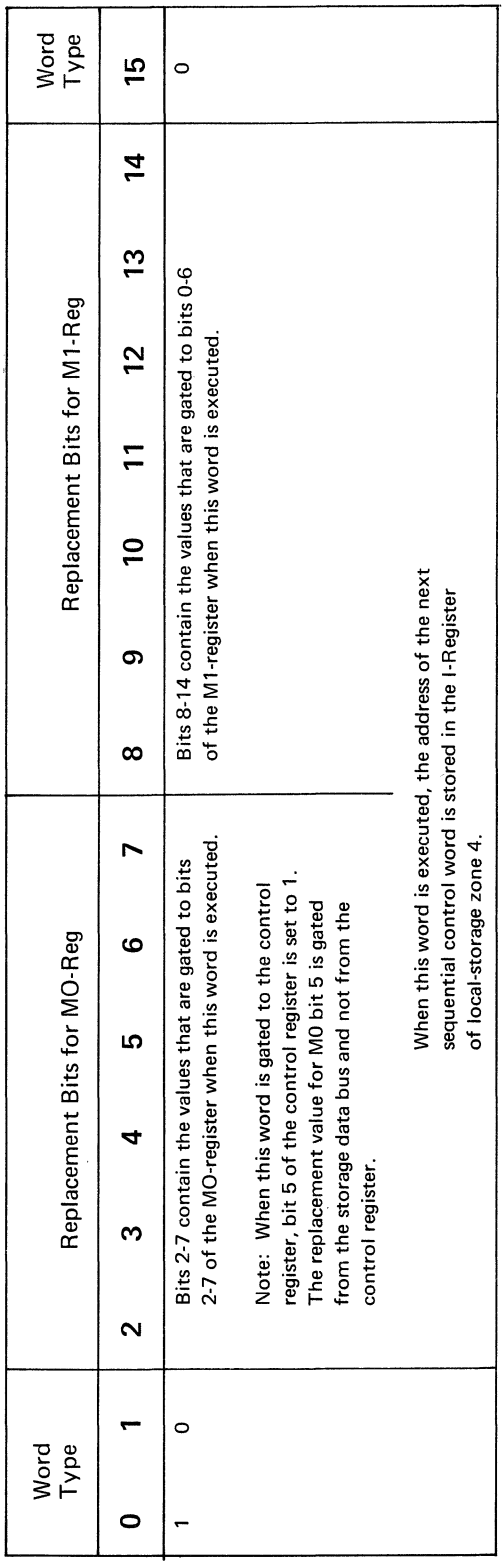


Figure 3-9. Bit Significance Chart for Branch Unconditional Word (Word Type 4)

Bit Significance Fields (Figure 3-9)

Bits 0, 1 and 15 are the word-type field and are set to 1, 0, 0 respectively to designate the Branch Unconditional word.

Bits 2 through 7 are the replacement bits that are gated to bits 2 through 7 of the M0-register.

Bits 8 through 14 are the replacement bits that are gated to bits 0 through 6 of the M1-register.

The replacement bits are assigned by the MAS program. The address assigned to the branched-to next-address label is found by the MAS program, and the corresponding bits of that address are placed in the Branch Unconditional word as replacement bits.

Description

The Branch Unconditional word appears in two different forms in the statement area of the microlistings: BR or BAL. In either case, the same function is performed and the same word type is designated. The microprogrammer generally writes BR when a return to the next sequential instruction is not anticipated. The BAL statement implies that a return to the next sequential instruction will occur when some subroutine starting at the branched-to address is finished.

Examples

Next Address Label: **MCHK** Statement: **BR**

	WT		RP for M0						RP for M1							WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	1	0	1	1	0	0	1	0	1	0	1	1	0	0	1	0

In this example, assume that the address assigned to the next-address label MCHK is 32B2 (hex). The replacement bits assigned for this example word correspond to the bits needed to address this next-address label.

When the Branch Unconditional word is gated to the control register, bit 5 of the control register is set to 1. Setting bit 5 of the control register is necessary for the other branch control words but is not used or suppressed for word type 4. The replacement bit assigned for bit 5 is gated from the SDB when the branch is executed. The following steps are taken to execute the unconditional branch word.

1. The address of the BR word is updated by 2 and stored in the I-register of LS zone 4.
2. The status of dynamic condition register bits 6 and 7 is stored in bits 0 and 1 of byte 0 of the I-register (restored with a return word).
3. The status of dynamic condition register bit 3 is stored in bit 7 of byte 1 of the I-register (restored with a return word).
4. The replacement bits from the control register are gated to the proper bits of the M0- and M1-registers. The replacement value for bit 5 of the M0-register is gated from the SDB bit-5 position.
5. Bits 0 and 1 of the M0-register are forced to an assigned value depending on the size of program storage.
   Bits 0, 1 = 11 for 48K program storage
   Bits 0, 1 = 10 for 32K program storage
   Bits 0, 1 = 10 for 24K program storage
   Bits 0, 1 = 01 for 16K program storage.
6. Bit 7 of the M1-register is forced to 0 to keep the address on a halfword boundary.

The functions described for the example word, BR, also apply if the statement were written BAI. Refer to MDM 5-11 for a detailed operation diagram of the BR word.

BRANCH ON MASK WORD

- Defined as word type 5 (control word bits 0,1, and 15 equal to 1,0,1 respectively).
- Performs 4-,8-, or 16-way branching.
- Performs special 2-way branch.

Bit Significant Fields (Figure 3-10)

Bits 0,1, and 15 are the word type field and are set to 1,0,1 respectively to indicate a Branch on Mask word.

Bits 2 and 3 designate the type of branch to be performed. These bits are the controls for gating the status of addressed bits to M1-register bit positions 3, 4, 5, and 6.

This function is covered more fully in this section under Examples.

Bit 5 determines the A-register gating. Bit 5 is assigned by the MAS program, depending on the bits called for in the statements.

If the bits called out in the statement are in the high four-bit group (0-3) of the designated byte source, the A-cross-low gate (AXL) is specified (bit 5 set to 1).

If the bits called out in the statement are in the low four-bit group (4-7) of the designated byte source, the A-low gate (AL) is specified (bit 5 set to 0).

If the 'byte source not zero' function is specified, the A-low gate and the A-high gate are specified.

Note: The status of bit 5 is taken from the SDB when this word is being operated on.

Bits 4, 6, and 7 determine the source of the bits to enter into the branch addressing. When the Branch on Mask word is placed into the control register, bit 5 of the control register is set to 1. Bit 5 then becomes a fixed portion of the source address along with bits 4, 6, and 7. This restricts the addressable facilities to those that have bit 5=1 in their source address decode. Refer to Figure 3-11.

Bit 8 determines the area in which the byte source is located. Bit 8=0 means the byte source is in local storage. Bit 8=1 means the byte source is an external facility.

Bits 9, 10, and 11 contain the replacement values to be gated to the M1-register when the Branch on Mask word is executed. Bit 9 contains the replacement value for bit 1 of the M1-register. Bit 10 contains the replacement value for bit 2 of the M1-register. Bit 11 contains the replacement value for bit 0 of the M1-register.

Bits 12, 13, and 14 contain the replacement values to be gated to the M0-register when the Branch on Mask word is executed.

Bits 12, 13, and 14 are gated to bits 5, 6, and 7 respectively of the M0-register.

Description

The function of the Branch on Mask word is to branch to a particular member of a specified branch set. The member of the branch set is determined by the status of specified bits of the byte source addressed.

The capability to branch to any member of a 4-, 8-, or 16-way branch set is available in this word. These branch capabilities are covered in detail under Examples.

Special 2-way Branch: An entire register is tested (all 8 bits) for a nonzero status. If nonzero, a branch occurs by forcing M1-register bits 3, 4, 5, and 6 to zero.

Word Type		Branch Control		A-Source Field					Replacement Bits for M1-Register			Replacement Bits for M0-Register			Word Type
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	00 = 4 way 01 = 8 way 10 = 2 way (A-Source Not 0) 11 = 16 Way			0 = STR 1 = CROSS			0 = LS 1 = EXT	Bit 9 contains replacement value for bit 1 of the M1-register Bit 10 contains replacement value for bit 2 of the M1-register Bit 11 contains replacement value for bit 0 of the M1-register			Bits 12, 13, and 14 contain the replacement values for bits 5, 6, and 7 of the M0-register respectively.			1

Local Storage

0000, 0100 = G0

0001, 0101 = G1

0010, 0110 = D0

0011, 0111 = D1

1000, 1100 = P0

1001, 1101 = P1

1010, 1110 = H0

1011, 1111 = H1

External facilities that may be an A-source are shown in Figure 3-11.

Note: Control word bit 5 is used to control A-register gating. The status of bit 5 is taken from the storage data bus-in. When the branch on mask word is placed into the control register, bit 5 is set to a 1 and becomes part of the A-source address. A-sources are, therefore, restricted to those that have at least a bit 5=1 in their source address decode.

Figure 3-10. Bit Significance Chart for Branch on Mask Word (Word Type 5)

Control Reg Bits		Mode Reg Bits		
*4567	8	Ext Ctrl 234	LS Zone 567	LOCATION ADDRESSED
0100	0	XXX	XXX	Byte 0 of G-Register,
0101	0	XXX	XXX	Byte 1 of G-Register,
0110	0	XXX	XXX	Byte 0 of D-Register,
0111	0	XXX	XXX	Byte 1 of D-Register,
				LS Zones 000, 100, 101, 110, 111
1100	0	XXX	XXX	Byte 0 of P-Register,
1101	0	XXX	XXX	Byte 1 of P-Register,
1110	0	XXX	XXX	Byte 0 of H-Register,
1111	0	XXX	XXX	Byte 1 of H-Register,
				LS Zones 000, 001, 100, 101, 110, 111
0100	1	000	XXX	Dynamic Condition Register (DYN)
0101	1	000	XXX	Status Register (S)
0110	1	000	XXX	Microprogram Mask Register (bits 0-7) MMSK
0111	1	000	XXX	Branch Conditions (BA)
1100	1	000	XXX	Diagnostic Register (DR)
1101	1	000	XXX	
1110	1	000	XXX	Error Register (MC)
1111	1	000	XXX	Soft-Stop Branch Conditions (BB)
0100	1	001	XXX	Dynamic Condition Register (DYN)
0101	1	001	XXX	Disk Attachment Status (DASI)
0110	1	001	XXX	Microprogram Mask Register (bits 0-7) (MMSK)
0111	1	001	XXX	Branch Conditions (BA)
1100	1	001	XXX	File Gated Attention (FGA)
1101	1	001	XXX	File Flags In (FFI)
1110	1	001	XXX	Disk Status (DS)
1111	1	001	XXX	File Op Register (FOP)
0100	1	110/010	XXX	Dynamic Condition Register (DYN)
0101	1	110/010	XXX	Status Register (S)
0110	1	110/010	XXX	Microprogram Mask Register (MMSK)
0111	1	110/010	XXX	Branch Conditions (BA)
1100	1	110/010		
1101	1	110/010	XXX	Reader Branch Conditions (RS)
1110	1	110/010	XXX	Reader-Punch Branch Conditions (RPS)
1111	1	110/010	XXX	Punch Branch Conditions (PS)
0100	1	011	XXX	Dynamic Condition Register (DYN)
0101	1	011	XXX	Status Register (S)
0110	1	011	XXX	Microprogram Mask Register bits 0-7 (MMSK)
0111	1	011	XXX	Branch Conditions (BA)
1100	1	011	XXX	
1101	1	011	XXX	
1110	1	011	XXX	Sense/Status Conditions (PRS)
1111	1	011	XXX	1403 Diagnostic Conditions (PRD)
0100	1	100	XXX	Dynamic Condition Register (DYN)
0101	1	100	XXX	Status Register (S)
0110	1	100	XXX	Microprogram Mask Register bits 0-7 (MMSK)
0111	1	100	XXX	Branch Conditions (BA)
1100	1	100	XXX	
1101	1	100	XXX	
1110	1	100	XXX	1052 Branch Conditions (TT)
1111	1	100	XXX	1052 Branch Conditions (TU)

CPU Mode

2311 Disk Mode

2540 Reader or Punch Mode

1403 Mode

1052 Mode

*Bit 5 may be a 0 in the control word in storage, but is forced on in the control register.

Figure 3-11. Addressing Capabilities for Branch on Mask Word (Part 1 of 2)

	Control Reg Bits		Mode Reg Bits		
	*4567	8	Ext Ctrl 234	LS Zone 567	LOCATION ADDRESSED
Communications Mode	0100	1	101	XXX	Dynamic Condition Register (DYN)
	0101	1	101	XXX	Status Register (S)
	0110	1	101	XXX	Microprogram Mask Register (bits 0-7) (MMSK)
	0111	1	101	XXX	Branch Conditions (BA)
	1100	1	101	XXX	Line Adapter Conditions (LACON)
	1101	1	101	XXX	Communication Channel Branch Condition (LASTAT)
	1110	1	101	XXX	Dial In (DILIN)
	1111	1	101	XXX	Communications Channel Branch Condition (GSTAT)
Channel Mode	0100	1	111	XXX	Dynamic Condition Register (DYN)
	0101	1	111	XXX	Status Register (S)
	0110	1	111	XXX	Microprogram Mask Register bits 0-7 (MMSK)
	0111	1	111	XXX	Branch Conditions (BA)
	1100	1	111	XXX	Channel Branch Conditions (GS)
	1101	1	111	XXX	Channel Branch Conditions (GT)
	1110	1	111	XXX	Channel Diagnostic Conditions (GD)
	1111	1	111	XXX	Channel Bus In (GB/IN)

*Bit 5 may be a 0 in the control word in storage, but is forced on in the control register.

Figure 3-11. Addressing Capabilities for Branch on Mask Word (Part 2 of 2)

Examples

Next Address Label: **CHECK N** Statement: **N = D1 BITS123**

	WT		BC		AS				LS Ext	RP for M1			RP for M0			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	1	0	0	1	0	1	1	1	0	x	x	x	x	x	x	1

Byte D1 from local storage is addressed, and its status is gated through the AB-assembler to the A-register.

SDB bit 5 brings up the AXL gate for the A-register, and bits 0-3 are crossed and gated out to the branch circuits as the low 4-bit output of the A-register.

Bits 2 and 3 of the control word allow the status of bits 5, 6, and 7 of the A-register output to be gated to bits 4, 5, and 6 of the M1-register. (A-register bits 5, 6, and 7 contain the actual status of bits 1, 2, and 3 of the D1 local-storage byte source.) Bit 3 of the M1-register is set to 0. Bits 0, 1, and 2 of the M1-register are set to the value of control word bits 11, 9, and 10 respectively. Bits 5, 6, and 7 of the M0-register are set to the value of control word bits 12, 13, and 14 respectively.

Bits 0-4 of the M0-register are set to the value contained in the corresponding bits of the W0-register.

The branch set that is branched-to is labeled CHECK (some hex digit). There are eight members of this branch set labeled sequentially: CHECK 0, CHECK 1, CHECK 2,....., CHECK 7. The hex digit of the label of the branch member branched-to in the example corresponds to the value contained in bits 1, 2, and 3 of the D1 byte of local storage. Refer to MDM 5-12.

Next Address Label: **TABLE N** Statement: **BR IF H1 = NZ**

	WT		BC		AS				LS Ext	RP for M1			RP for M0			WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	1	0	1	0	1	0	1	1	0	1	1	0	0	1	1	1

Assume the next address label is located at hex address 3B60.

The local-storage byte source H1 is addressed, and its contents gated through the AB-assembler to the A-register. The A-register contents are gated to the zero test circuits. If the test indicates that H1 is nonzero, bits 3-7 of the M1-register are forced to 0. Bits 0, 1, and 2 of the M1-register are set to the value of control word bits 11, 9, and 10 respectively. Bits 5, 6, and 7 of the M0-register are set to the value of control word bits 12, 13, and 14 respectively. Bits 0-4 of the M0-register are set to the value of the corresponding bits of the W0-register.

If the test indicates H1 is equal to zero, the next sequential control word is read out and operated on.

BRANCH ON CONDITION WORD

- Defined as word type 6 or 7 (control word bits 0, 1, and 15 equal to 1,1,0 or 1,1,1 respectively).
- Tests one of the eight bits of a designated byte source for a 0 or 1 condition.
- Branches if the condition of the bit tested compares to the condition indicated by the Branch on Condition statement.

Bit Significance Fields (Figure 3-12)

Bits 0, 1, and 15 are the word type field. Bits 0 and 1 are set to 1 to designate a Branch on Condition word. Bit 15 is set to indicate the condition to be tested for. Bit 15=0, test for 0. Bit 15=1, test for 1.

Bits 2, 3, and 5 determine the bit of the designated byte source to be tested. The test circuits receive bits to be tested from the low 4 bits of the A-register. Bit 5 of the control word gates the proper 4-bit group in the A-register to the branch circuits, either by crossing the high 4-bit group to the low 4-bit group, or by gating the low 4-bit group straight.

Bit Tested	Bits 2, 3	SDB Bit 5	A-Reg Gate
0	00	1	AX Low
1	01	1	AX Low
2	10	1	AX Low
3	11	1	AX Low
4	00	0	A Low
5	01	0	A Low
6	10	0	A Low
7	11	0	A Low

AX Low = cross the high 4 bits (0-3) of the A-register to the low 4 bits (4-7), and gate just these low 4 bits out.

A Low = gate only the low 4 bits (4-7) of the A-register out.

Note: When the Branch on Condition word is read into the control register, bit 5 is forced to a 1. This restricts the byte sources that may be tested to those that have at least a bit 5=1 in the AS-decode. Refer to Figure 3-13. The straight/cross gate for the A-register is controlled from the SDB bit 5 for this word.

Bits 4, 5, 6, and 7 make up the address of the byte source that is read out and tested. Bit 5 is forced on when this control word type is set into the C-register, regardless of the bit-5 status in control storage.

The address contained in this field, along with the external gating controls and the LS-zone, determine exactly the byte source to be read.

Bit 8 determines if the decode of bits 4, 5, 6, and 7 indicate a local storage or an external facility address. Bit 8=0 indicates local storage. Bit 8=1 indicates external facilities.

Bits 9 through 14 contain the replacement values for bits 1 through 6 of the M1-register, if the branch condition is met and the branch is to be taken.

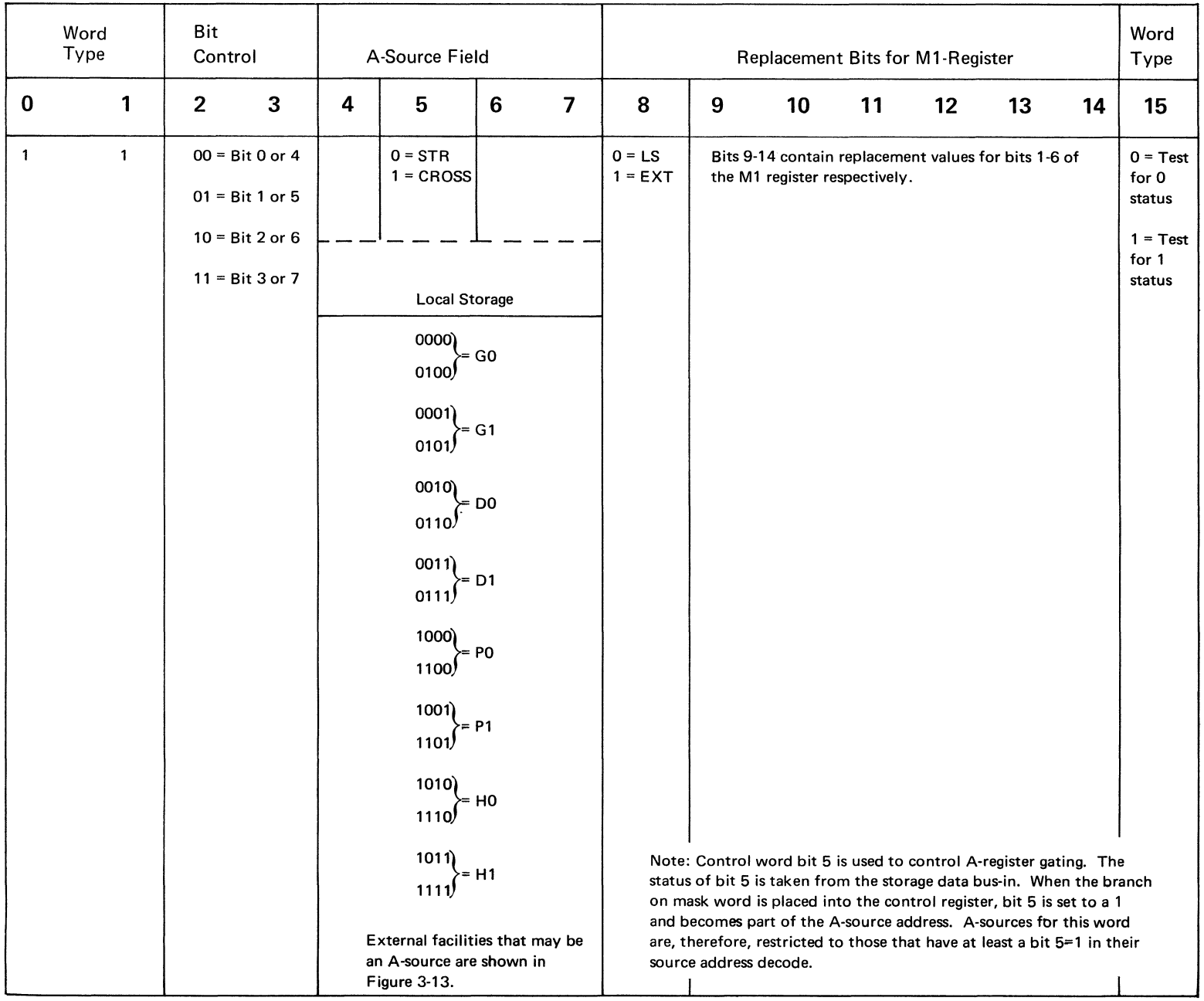


Figure 3-12. Bit Significance Chart for Branch on Condition Word (Word Type 6 or 7)

Control Reg Bits		Mode Reg Bits		
*4567	8	Ext Ctrl 234	LS Zone 567	LOCATION ADDRESSED
0100	0	XXX	XXX	Byte 0 of G-Register,
0101	0	XXX	XXX	Byte 1 of G-Register.
0110	0	XXX	XXX	Byte 0 of D-Register,
0111	0	XXX	XXX	Byte 1 of D-Register,
1100	0	XXX	XXX	Byte 0 of P-Register,
1101	0	XXX	XXX	Byte 1 of P-Register,
1110	0	XXX	XXX	Byte 0 of H-Register,
1111	0	XXX	XXX	Byte 1 of H-Register,
0100	1	000	XXX	Dynamic Condition Register (DYN)
0101	1	000	XXX	Status Register (S)
0110	1	000	XXX	Microprogram Mask Register (bits 0-7) MMSK
0111	1	000	XXX	Branch Conditions (BA)
1100	1	000	XXX	Diagnostic Register (DR)
1101	1	000	XXX	
1110	1	000	XXX	Error Register (MC)
1111	1	000	XXX	Soft-Stop Branch Conditions (BB)
0100	1	001	XXX	Dynamic Condition Register (DYN)
0101	1	001	XXX	Disk Attachment Status (DASI)
0110	1	001	XXX	Microprogram Mask Register (bits 0-7) (MMSK)
0111	1	001	XXX	Branch Conditions (BA)
1100	1	001	XXX	File Gated Attention (FGA)
1101	1	001	XXX	File Flags In (FFI)
1110	1	001	XXX	Disk Status (DS)
1111	1	001	XXX	File Op Register (FOP)
0100	1	110/010	XXX	Dynamic Condition Register (DYN)
0101	1	110/010	XXX	Status Register (S)
0110	1	110/010	XXX	Microprogram Mask Register. (MMSK)
0111	1	110/010	XXX	Branch Conditions (BA)
1100	1	110/010	XXX	
1101	1	110/010	XXX	Reader Branch Conditions (RS)
1110	1	110/010	XXX	Reader-Punch Branch Conditions (RPS)
1111	1	110/010	XXX	Punch Branch Conditions (PS)
0100	1	011	XXX	Dynamic Condition Register (DYN)
0101	1	011	XXX	Status Register (S)
0110	1	011	XXX	Microprogram Mask Register bits 0-7 (MMSK)
0111	1	011	XXX	Branch Conditions (BA)
1100	1	011	XXX	
1101	1	011	XXX	
1110	1	011	XXX	Sense/Status Conditions (PRS)
1111	1	011	XXX	1403 Diagnostic Conditions (PRD)
0100	1	100	XXX	Dynamic Condition Register (DYN)
0101	1	100	XXX	Status Register (S)
0110	1	100	XXX	Microprogram Mask Register bits 0-7 (MMSK)
0111	1	100	XXX	Branch Conditions (BA)
1100	1	100	XXX	
1101	1	100	XXX	
1110	1	100	XXX	1052 Branch Conditions (TT)
1111	1	100	XXX	1052 Branch Conditions (TU)

LS Zones 000, 100, 101, 110, 111

LS Zones 000, 001, 100, 101, 110, 111

CPU Mode

2311 Disk Mode

2540 Reader or Punch Mode

1403 Mode

1052 Mode

*Bit 5 may be a 0 in the Control Word in storage, but is forced on in the Control Register.

Figure 3-13. Addressing Capabilities for Branch on Condition Word (Part 1 of 2)

	Control Reg Bits		Mode Reg Bits		
	*4567	8	Ext Ctrl 234	LS Zone 567	LOCATION ADDRESSED
Communications Mode	0100	1	101	XXX	Dynamic Condition Register (DYN)
	0101	1	101	XXX	Status Register (S)
	0110	1	101	XXX	Microprogram Mask Register (bits 0-7) (MMSK)
	0111	1	101	XXX	Branch Conditions (BA)
	1100	1	101	XXX	Line Adapter Conditions (LACON)
	1101	1	101	XXX	Communication Channel Branch Condition (LASTAT)
	1110	1	101	XXX	Dial In (DILIN)
	1111	1	101	XXX	Communications Channel Branch Condition (GSTAT)
Channel Mode	0100	1	111	XXX	Dynamic Condition Register (DYN)
	0101	1	111	XXX	Status Register (S)
	0110	1	111	XXX	Microprogram Mask Register bits 0-7 (MMSK)
	0111	1	111	XXX	Branch Conditions (BA)
	1100	1	111	XXX	Channel Branch Conditions (GS)
	1101	1	111	XXX	Channel Branch Conditions (GT)
	1110	1	111	XXX	Channel Diagnostic Conditions (GD)
	1111	1	111	XXX	Channel Bus In (GB/IN)

*See Note Part 1.

Figure 3-13. Addressing Capabilities for Branch on Condition Word (Part 2 of 2)

Description

The function of the Branch on Condition word is to test the status of a bit of a designated byte source. If the status of the bit corresponds to the condition called for by the Branch on Condition word, a branch is taken. The branch is performed by replacing bits 1-6 of the M1-register with the value contained in bits 9-14 of the Branch on Condition word. Bit 7 of M1 is forced to a 0. Bit 0 of M1 is set to the value of bit 0 of the W1-register. The M0-register is set to the value of the W0-register.

If the status of the tested bit does not correspond to the condition called for, the next sequential control word is read out and operated on.

Examples

These are examples of the Branch on Condition word as they appear in the statement area of the microlistings, the bit structure assigned by the MAS program, and the description of data handling.

Next Address Label: **INTPR** Statement: **BR IF G1 BIT3=0**

	WT		BC		AS				LS Ext	RP for M1						WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	1	1	1	1	0	1	0	1	0	0	1	0	1	0	0	0

Assume the next address label is located at hex address 4CA8.

The MAS program assigns the replacement bits for the control word in the following manner.

1. The label INTPR is looked up in an assigned address table.
2. The status that would be placed into bits 1-6 of the M1-register, to address label INTPR, is assigned to bits 9-14 of the control word.

Byte G1 from local storage is read out and gated through the AB-assembler to the A-register. SDB bit 5 is used to form the AXL gate for the A-register. Bits 0, 1, 2, and 3 of the G1-register are gated to the low four bits of the A-register output. These four bits are gated to branch

circuits where control register bits 2 and 3 determine which bit of this four-bit group is tested. Bit 3 is selected and tested for a 0 status. If bit 3=0, the branch is taken; if bit 3=1, the next sequential control word is read out and operated on.

Next Address Label: **ADDLE** Statement: **BR IF TT BIT6=0**

	WT		BC		AS				LS Ext	RP for M1						WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	1	1	1	0	1	0	1	0	1	1	1	0	1	1	0	0

Assume the next address label is located at hex address 156C.

The external facility, TT, is addressed and its status is gated on the external bus-in to the AB assembler. The AB-assembler gates the TT status to the A-register. The SDB bit-5 status (0) brings up the A-low gate for the A-register. Bits 4-7 of the A-register are gated to branch test circuits, and bits 2 and 3 of the control register determine which bit of this four-bit group is tested. Bit 6 is selected and tested for a 0 status. If bit 6=0, the branch is taken; if bit 6=1, the next sequential control word is read out and operated on. Refer to MDM 5-13.

Next Address Label: **CTRL** Statement: **BR IF LZ=0**

	WT		BC		AS				LS Ext	RP for M1						WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	1	1	1	1	0	0	0	0	1	1	1	1	1	0	1	1

Assume the next address label is located at hex address 1AFA.

Bit 7 of the dynamic condition register (DYN) is tested for a status of 1 to determine if the low Z-bus is equal to zero.

The external facility DYN is addressed, and its status is gated on the external bus-in to the AB-assembler. The DYN status is then gated to the A-register. SDB bit-5 status (0) brings up the A-low gate for the A-register. Bits 4-7 of the A-register are gated to branch circuits, and bits 2 and 3 of the control register determine that bit 7 is to be tested. If bit 7 of DYN=1, branch to hex address 1AFA. If bit 7 of DYN=0, step to next sequential control word.

Next Address Label: **ADDLP** Statement: **BR IF H00=1**

	WT		BC		AS				LS Ext	RP for M1						WT
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit Structure	1	1	0	0	1	1	1	0	0	0	1	1	1	1	1	1

Assume the next address label is located at hex address 7DBE.

Byte H0 of local storage is addressed and its status is gated through the AB-assembler to the A-register. SDB bit-5 status (1) brings up the A-cross-low gate for the A-register. Bits 0-3 of the A-register are gated out in the low four-bit positions of the A-register output to the branch circuits.

Bits 2 and 3 of the control register indicate that bit 0 is to be tested. Bit 15 of the control register specifies that a condition of 1 is to be tested for. If bit 0 of H0=1, branch to the hex address 7DBE. If bit 0 of H0=0, step to the next sequential control word.

SYSTEM/360 MODEL 25 1401/1460 AND 1440 COMPATIBILITY FEATURES

- 1401/1460 and 1440 Compatibility features enable a rapid and simplified transfer from 1401, 1460 and 1440 applications to the IBM System/360 Model 25.

- The 1401/1460 Compatibility feature provides compatibility of 1401 and 1460 programs; the 1440 Compatibility feature provides compatibility of 1440 programs.

- The 1401/1460 feature and the 1440 feature are mutually exclusive in operation. Separate Control Storage Load decks are used to initialize for either 1401/1460 or 1440 operation according to the user's applications.

The 1401/1460 and 1440 Compatibility features provide compatibility with an existing 1401, 1460, or 1440 system configuration. Compatibility features permit 1401, 1460, and 1440 object programs, using comparable I/O devices included for the feature, to be executed on a System/360 Model 25 without modification.

The 1401/1460 Compatibility feature provides compatibility of 1402 and 1403 operations (except multiple printer operations) for 1401 and 1460 programs. The 2540 card read-punch and 1403 printer units are attached to integrated control units. A channel-attached 1403 or 1403-N1 printer also can be used.

The 1440 Compatibility feature provides compatibility for 1442, 1443, or 1403 operations for 1440 programs. The 1442-N1 card read-punch or 1442-N2 card punch and the 1443 or 1403 are attached to the multiplexer or selector channel by standard interface. The integrated 1403 also can be used in place of a 1403 or 1443 on channel. The integrated 2540 read punch can be used (with some limitations) in place of a 1442 or 1444.

The 1400-series object program is stored in upper main storage of the 2025. Address conversion and character conversion occurs as needed. The original 1400 program sequence and character configuration are not disturbed. Microprogram routines perform the same functions in 1400 Compatibility mode as circuitry performed in 1401, 1460, or 1440 systems.

A System/360 Model 25 with comparable I/O devices can assume the functions of any 1401, 1460, or 1440 system having the following IBM I/O units.

1402 Card Read Punch
1442 Card Read Punch
1444 Card Punch
1403 Printer
1443 Printer
1407 Console Inquiry Station
1447 Console Inquiry, Models 2 or 3
1311 Disk Storage Drive
729 or 7330 Magnetic Tape Units
7335 Magnetic Tape Unit

The functions of the input/output units that cannot be attached to the System/360 Model 25 are provided by similar units attached to the system; i.e., the Console Printer-Keyboard is used for 1407 operations.

A general knowledge and familiarity with 1400-series system operation is prerequisite to understanding 1400-mode operation of the System/360 Model 25. Refer to the 1400 system manuals as necessary to review details of 1400 operations.

For brevity and ease of discussion, terms such as "1400-mode", "1400 compatibility", "1400 type", "1400-series", etc. appear in this manual. These terms pertain to the 1401/1460, and 1440 Compatibility features and should not be construed as referring to any other 1400-series system or unit.

IMPLEMENTATION

- Separate and distinct control storage load routines initialize auxiliary and control storage areas to control data flow in a manner that emulates the 1401/1460, or 1440 system. Circuit changes are minor.

- I0I1, U0U1, V0V1, and G1 and D1 in local storage zone 0 perform the functions of I-Star, A-Star, B-Star, and Op-reg and A-reg respectively.

- Auxiliary storage is loaded with conversion tables, constants, and other control factors required to absorb the differences in code structure and storage addressing between the 1401/1460 or 1440 system and the System/360.

The 1401/1460 or 1440 Compatibility feature is implemented on the System/360

Model 25 largely by unique control program routines that utilize local and auxiliary storage extensively. Auxiliary storage normally provides residence for general purpose, floating point, and condition registers, multiplexer channel unit control words (UCWs), etc. For 1401/1460 or 1440 compatibility, however, auxiliary storage is loaded with conversion tables, constants, and other control factors required by the control program during compatibility mode operations.

In addition to constants and conversion tables, considerable information such as storage size, tape densities, unit addresses, system features, and other characteristics unique to a particular 1401/1460 or 1440 System/360 emulation are entered into auxiliary storage. Auxiliary storage and local storage also provide data registers and status indicators that are used by the microprogramming routines. For example, the I0I1, U0U1, V0V1, G1 and D1 registers in local storage zone 0 perform the functions of I-Star, A-Star, B-Star, Op-reg and A-reg respectively.

The 1401/1460 and 1440 Compatibility features require only minor modification to basic circuit design. The additional special circuitry consists of approximately 9 cards in the CPU. The additional circuitry is necessary to emulate 1402 reader characteristics, file unit characteristics, and to facilitate storage-wrap detection.

All CPU circuitry is utilized for 1401/1460 and 1440 compatibility mode operation except:

- Storage protection feature
- Interval timer feature
- Channel-to-channel adapter feature.

COMPATIBILITY INITIALIZATION

- Compatibility initialization is by the Control Storage Load routine with the 1400-mode CSL deck.
- Control Storage Loading loads the control storage area with the microprogram routines necessary to emulate the 1401/1460 or 1440 operations.
- Control Storage Loading also loads auxiliary storage with the necessary conversion factors, tables, constants, etc.
- Control Storage Loading for 1400-mode follows the same general pattern as for other modes of operation.

A Control Storage Loading procedure (CSL), using a special card deck, loads either the 1401/1460 or 1440 compatibility mode control program. CSL of the 1401/1460 or 1440 compatibility mode control program performs the same general functions as described in the section for CSL in System/360 mode. Many of the same basic diagnostic routines are used.

The control program interprets instructions, performs arithmetic and data-handling functions, etc. in 1400 mode. In addition, the initialization routine loads auxiliary storage with fixed information required by the microprogram routines to absorb the difference in code structure and storage addressing between the emulated 1400 system and the System/360. Certain variable information (tape densities, unit addresses, etc.) also must be entered before the Model 25 can execute 1400 instructions.

OP-CODE CONVERSION AND RECOGNITION

- 1400 system op-codes are converted to bit significant characters during I-cycles to facilitate recognition by the microprogram.
- An op-code conversion table is stored in auxiliary storage.
- 1400-system op-code character bits 0 and 1 are forced on before using the character to address the conversion table.
- The converted op-code is stored in the G1-register in local storage during I-cycles.

The EBCDIC bit configurations of 1400-system op-codes do not readily indicate to 2025 microprogramming the type of op-code being handled. A conversion table, stored in auxiliary storage module 2, groups similar op-codes together and converts the bits to a configuration that is bit-sensitive for easy identification by the microprogram.

When the microprogram reads a 1400-system op-code from storage in EBCDIC form, it forces on the 0 and 1 bits of the op-code. Figure 4-1 shows that if the 0 and 1 bits of all 1400-system characters are forced on, there are still enough unique bit combinations for all characters except blank, -, and &. These three are not valid 1400-system op-codes, and they are detected as such by a microprogram step.

4567 ↓	WITH WORDMARK 0123 →				NO WORDMARK				WITH WORDMARK				NO WORDMARK			
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	Blank	&	-		Blank	&	--		?	I	‡	0	?	I	‡	0
0001			/				/		A	J		1	A	J		1
0010									B	K	S	2	B	K	S	2
0011									C	L	T	3	C	L	T	3
0100									D	M	U	4	D	M	U	4
0101									E	N	V	5	E	N	V	5
0110									F	O	W	6	F	O	W	6
0111									G	P	X	7	G	P	X	7
1000									H	Q	Y	8	H	Q	Y	8
1001									I	R	Z	9	I	R	Z	9
1010				ƀ				ƀ								
1011	.	$	'	#	.	$	'	#								
1100	¤	*	%	@	¤	*	%	@								
1101	[	]	ᵛ	:	[	]	ᵛ	:								
1110	<	;	\	>	<	;	\	>								
1111	‡	△	⧻	√‾	‡	△	⧻	√‾								

Figure 4-1. 1400 Defined Characters

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C	(?) 1C	(A) 18	(B) 0B	(C) 1F	(D) 12	(E) 16	(F) 2A	34	(H) B1	34	34	(·) 02	(¤) 15	34	34	34
D	(!) 1D	34	(K) 29	(L) 90	(M) 80	(N) 06	34	(P) 1E	(Q) F1	34	34	34	34	34	34	34
E	34	(/) 05	(S) 19	34	(U) 20	(V) 0E	(W) 0F	34	(Y) 13	(Z) 17	34	(,) 04	(%) 1B	34	34	34
F	34	(1) 21	(2) 22	(3) 23	(4) 24	(5) 25	(6) 26	(7) 27	(8) 06	(9) 06	34	(#) 14	(@) 1A	34	34	34

Notes:

1. The 1400-series operation code is shown in parentheses.
2. The code 34 is used to indicate an invalid 1400-series operation code.
3. The code 06 is used to indicate a No-Op (no operation).
4. Notice that several 1400-series special features, such as Branch if Bit Equal (W op-code), Divide (% op-code), etc. are standard with the compatibility feature. These 1400-series operations can be made invalid by inserting the invalid code (34) in the corresponding table location, if desired.

Figure 4-2. 1400 Operation-Code Decode Table

The modified op-code, formed when the 0 and 1 bits are forced on, is used to address auxiliary storage and bring out the converted op-code stored by the control storage load routine. The converted op-code bears no logical resemblance to the original. The new op-code character has a bit configuration that is more readily tested to determine the type of operation desired (I/O, CPU, miscellaneous).

The op-code conversion table is used in the following manner. Assume the op-code read out of the 1400-system program is the edit operation. The hexadecimal bit configuration of E with a WM in EBCDIC is 85 (1000 0101). A microprogram step forces on bits 0 and 1. This changes the configuration to C5 (1100 0101).

C5 is used to address the op-code table in auxiliary storage (Figure 4-2). From auxiliary storage Module-2 position C5, 16 (0001 0110) is read out and placed in the G1-register of local storage zone 0. The 16 is bit-sensitive to the microprogram as the edit op-code.

Any invalid EBCDIC op-code configuration that addresses the op-code table brings out a byte containing 34. This is detected as an error by a microprogram step.

CHARACTER CONFIGURATION

- The BCD characters of the 1400 system being emulated are represented in the 2025 in System/360 EBCDIC.
- Absence of a 1-bit in EBCDIC representation indicates that a wordmark is associated with the character.
- Character conversion from EBCDIC to BCD and vice-versa is through a table lookup.

The BCD characters of the 1400 system being emulated in 1400 compatibility mode are represented in the 2025 in EBCDIC. This utilizes all eight data bits plus a parity bit. Note in Figure 4-1 that all valid BCD characters represented in EBCDIC contain a bit in byte position 1. Wordmark notation is achieved by manipulating bit position 1 while still maintaining EBCDIC compatibility. The absence of a 1-bit in the EBCDIC respresentation indicates that a wordmark is associated with the character. Thus the character A without a wordmark is 1000 0001.

Occasionally it is necessary to translate characters from EBCDIC to BCD and back again to process certain 1400-system instructions: bit test, move zone, or move numeric. Conversion is accomplished by a table lookup procedure that uses tables stored in auxiliary storage. These tables are read into storage as part of the initializing routine.

In utilizing conversion tables, if a wordmark is present with the character, the microprogram eliminates it (by turning on bit 1) before the table lookup is executed.

It might be helpful to examine the relationship between A and B zone bits in BCD representation and EBCDIC bit structure. Bits 2 and 3 in EBCDIC represent the B and A bits in BCD configuration as follows:

EBCDIC	00	01	10	11	┐
					├-Bits 2, 3
BCD	11	10	01	00	┘
Zones	BA	B	A	no zones.	

To illustrate the use of the table in auxiliary storage (Figure 4-3), convert a character from EBCDIC to BCD. The character C in EBCDIC is C3 (1100 0011). By going to auxiliary storage module-0 location C3, you find the configuration 33 or 0011 0011, which is CBA21, the BCD configuration for a C.

The table is also addressed by EBCDIC characters that are read in from cards to determine BCD validity. If an invalid EBCDIC character addresses the EBCDIC to BCD conversion table, a hexadecimal character is read out. If the 1 bit is on, it is detected as an error by the microprogram.

1401/1460 AND 1440 PROGRAM ADDRESSING

- The 1401/1460 or 1440 compatibility mode storage area is located contiguously in the upper part of program storage area in 2025 main storage.
- A halfword in auxiliary storage contains a storage bias constant (or offset) that is unique for each combination of 1400/25 size.
- A conversion table in auxiliary storage converts BCD addresses to binary addresses with consideration of the bias constant.
- An area below the bias is used for File Tables.

For 1401/1460 or 1440 compatibility mode operation, object programs are loaded into upper program storage area of main storage in the IBM 2025 Processing Unit. If a 1401 program written for 4K of storage is to be run on a 2025 with 16,384 positions of storage, the program is stored in IBM 2025 program storage area locations 12,384 to 16,383. This allows the 2025 to detect 1401 high storage-wrap errors through existing circuits.

The 2025 uses a conversion table in auxiliary storage to convert 1401 BCD addresses to System/360 binary addresses. The actual values in the conversion table include the bias constant value that is unique to the 24 possible Model 25 program

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	00	40	40	40	40	40	40	40	40	40	40	3B	3C	3D	3E	3F
5	30	40	40	40	40	40	40	40	40	40	40	2B	2C	2D	2E	2F
6	20	11	40	40	40	40	40	40	40	40	40	1B	1C	1D	1E	1F
7	40	40	40	40	40	40	40	40	40	40	10	0B	0C	0D	0E	0F
C	3A	31	32	33	34	35	36	37	38	39	40	40	40	48	45	5C
D	2A	21	22	23	24	25	26	27	28	29	40	40	40	40	46	5D
E	1A	40	12	13	14	15	16	17	18	19	40	40	40	49	4F	56
F	0A	01	02	03	04	05	06	07	08	09	40	40	40	40	44	5F

Figure 4-3. EBCDIC-to-BCD Translate Table

storage size and/or 1400 program size combinations. Refer to Figure 4-4. These values are set at CSL time.

For example, in running a 1401 program written for 4K of storage on a 2025 with 16,384 positions of storage, the storage bias constant would be:

16,384 (2025 storage size)
- 4,000 (1401 storage size)
12,384 or 3060 in hexadecimal

In addition to the program storage area utilized by the 1400-series object program, a 214-byte area located below the address specified by the storage bias constant is used for file tables. This address varies with storage sizes and is developed during the system reset routine. See Figure 4-4 for the appropriate start address of this table.

ADDRESS CONVERSION

- 1400 addresses are stored in EBCDIC form.
- Address conversion is by microprogramming steps that utilize tables in auxiliary storage.
- A 2-byte binary address is developed from 3 characters in EBCDIC code.
- Units and hundreds zone bits determine thousands (tens zone bits provide indexing).

For an example of address conversion, assume that a 2025 with 16,384 positions of

25 / 1400	16,384		24,576		32,768		49,152	
	Bias Constant	File Table	Bias Constant	File Table	Bias Constant	File Table	Bias Constant	File Table
16K	0180	0000	2180	2000	4180	4000	8180	8000
12K	1120	1000	3120	3000	5120	5000	9120	9000
8K	20C0	1F00	40C0	3F00	60C0	5F00	A0C0	9F00
4K	3060	2F00	5060	4F00	7060	6F00	B060	AF00
2K	3830	3700	5830	5700	7830	7700	B830	B700
1.4K	3A88	3900	5A88	5900	7A88	7900	BA88	B900

Bias Constant: Located in byte 88-89, (K0) in Module 0.

File Table Address: Developed from bias constant during file routine.

Figure 4-4. Storage Bias Constant and File Table Addresses

program storage is emulating a 1401 progra written for a 12K 1401. Figure 4-4 indicates that the bias constant is 1120.

The objective in 1401/1460 or 1440 compatibility address conversion is to convert the 1400 3-character address that is in storage to a two-byte binary address that includes the storage bias offset factor. Microprogramming and conversion tables facilitate the conversion. A-Star address development will be discussed during I-Phase for the instruction BA5SE.

To convert the 1400 address to a binary address, the microprogram uses digits (and zones) in the 1400 address to read out tables in auxiliary storage module 2. Thus, to convert the hundreds position (example: A=1100 0001) of a 1400 address to a binary value, microprogram steps manipulate the character to facilitate addressing the auxiliary storage table. The microprogram first crosses the byte (1100 0001 becomes 0001 1100), then adds the low bits (4-7) to the byte:

```
0001 1100
  +  1100
---------
0010 1000
```

The result (28) addresses auxiliary storage hundreds and thousands translate table (Figure 4-5), row 2 column 8. This example just given is for a BCD character with AB-zones, (3,000). For a BCD character with B-zone only (2,000), column-A is addressed; for a character with A-zone only (1,000), column-C is addressed; and for a no-zone (numeric) character, column-E is addressed. The bit configuration of the / (slash) does not fit into the routine utilized by other characters (at hundreds position) for addressing auxiliary storage. The translation of the / is handled separately and addresses the halfword at location 2016.

In converting the three-character 1400 address to a two-byte binary address, the microprogram accumulates the binary values of the three characters in the 1400 address plus the factor for memory bias. Consider the sample address A5S. This represents a decimal address of 7,152 (Hundreds Zones = BA = 3000, Units Zones = A = 4000).

First examine this address decimally, and convert it using decimal values to illustrate how the result is obtained. The desired address consists of the 1400 address 7,152 plus the storage bias constant of 4,384 for a total of 11,536.

AUXILIARY STORAGE MODULE 2

	6 7	8 9	A B	C D	E F
1X	B+044C	B+0BB8	B+07D0	B+03E8	B+0000
2X		B+0C1C	B+0834	B+044C	B+0064
3X		B+0C80	B+0898	B+04B0	B+00C8
4X		B+0CE4	B+08FC	B+0514	B+012C
5X		B+0D48	B+0960	B+0578	B+0190
6X		B+0DAC	B+09C4	B+05DC	B+01F4
7X		B+0E10	B+0A28	B+0640	B+0258
8X		B+0E74	B+0A8C	B+06A4	B+02BC
9X		B+0ED8	B+0AF0	B+0708	B+0320
AX		B+0F3C	B+0B54	B+076C	B+0384

Note: B=Halfword bias constant in K0 (Module 0; 88-89)
See previous figure for value.

Hundreds/Thousands/Bias Conversion

AUXILIARY STORAGE MODULE 2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0X	00	0A	14	1E	28	32	3C	46	50	5A	01	01	01	01	01	01

Tens Conversion

Figure 4-5. Decimal-to-Hexadecimal Conversion

Follow this example through conversion showing logically what happens without regard to exact sequence of microprogramming steps. Use the term "accumulator" to refer to the place of address development.

Example: A5S in 1400-program size 12K in 16K 2025.

1. Read out hundreds character A = BA1 = 1100 0001. Zone bits (2, 3) = 00 = BA = 3000. By microprogram manipulation, auxiliary storage module 2, bytes 28 and 29 are addressed.
2. Value in auxiliary storage hundreds/thousands/bias translation table equals 3100 + Bias or hex 0C1C + 1120 = 1D3C. This value is entered in the accumulator.
3. Read tens character 5 = C41 = 1111 0101. There are no zones. If tens byte were zoned, indexing would be required. The tens numeric portion is used to address byte 05 of auxiliary storage module 2. The value (32) in this byte is entered in the accumulator.
4. Read units character S = CA2 = 1110 0010. Units numeric (2) is entered directly to accumulator. Units zone bits (2, 3 = 10) are translated by the microprogram. 10 = 4000 = 0FA0.
5. The accumulator now contains 1D3C (0C1C + 1120) +32 +2 + 0FA0 = 2D10 = 11,536.

Factor	Action	Accumulator	Auxiliary Storage
Hundreds/Thousands/Bias	3100+1120=1D3C	1D3C	Module 2-28
Tens	1D3C+32=1D6E	1D6E	Module 2-05
Units/Thousands	1D6E+2+0FA0=2D10	2D10	(Microprogram Steps)

Figure 4-6. Address Conversion Summary

Figure 4-6 summarizes the example.

ADDRESS ERROR DETECTION

- Microprogram routines detect invalid characters during the conversion of 1400-system addresses to 2025 addresses.
- High storage wrap is detected by the same circuitry as in System/360 mode operation.
- The value 8F is placed in 1400-address 000 minus one to facilitate low storage wrap error detection.

For 1401/1460 or 1440 compatibility operation, the upper position of 1400-system compatibility storage is coincident with the upper position of 2025 storage. This enables high storage wrap-error detection by the same circuitry as in System/360 mode operation.

For an error to be detected when the equivalent 1400-address 000 is modified by minus one (low storage wrap error), an invalid character (8F) is placed in 2025 storage, one core location below the address assigned to 1400 compatibility storage location 000. If this location is addressed, the invalid character causes an invalid address stop.

The invalid character, 8F, is placed in storage as required by the control storage load routine.

During the conversion of 1400-addresses to 2025 addresses, an error-detection procedure detects invalid characters. At the beginning of I-cycles, status bits are turned on indication invalid A-and B-addresses. As the addresses are converted and found to be valid, the corresponding status bit is turned off. Thus, an op-only, I4, or I5 instruction can have an invalid address because of the absence of one or both addresses. This is not an error condition.

AUXILIARY STORAGE

The auxiliary storage section of the 2025 main storage normally provides residence for general-purpose registers, floating point registers, condition registers, unit-control words for channel, temporary work areas, translate tables, etc.

For operation to be performed in 1401/1460 or 1440 compatibility mode, auxiliary storage must be loaded with certain fixed information required by the microprogram routines to absorb the difference in code structure and storage addressing between the 1401/1460 or 1440 systems and System/360. Variable information such as tape densities, unit addresses, storage size, etc. also must be entered before the 2025 can execute 1401/1460 or 1440 instructions. Other areas of auxiliary storage are initialized (either to 0 or to some required value.) These areas provide status indication and control information for the microprogram, backup registers, etc.

Because auxiliary storage performs such a vital role in 1401/1460 and 1440 compatibility mode operation, each of the 2048 bytes is quite significant. Customer modifications, such as alteration of op-code tables, nonstandard special character print arrangements, etc., demand an even greater involvement with details in auxiliary storage.

Placement of the 1400 compatibility requirements (8 modules of 256 bytes each) is in auxiliary storage modules 0 through 7 for 16K, 32K and 48K units. In the 24K CPU, placement is in modules 0 through 5 and 8 and 9 because modules 6 and 7 are not available.

An examination of the 2048 bytes of auxiliary storage (Figure 4-7), row by row and byte by byte or bit by bit, may explain the function more clearly.

The cross-reference listing portion of the microlisting for the CSL deck includes a printout of the contents of each byte of auxiliary storage prior to initialization.

The system initializer card tailors the system to the user's compatibility application with regard to program control, bias constant, I/O addresses, etc.

For example, before the last card of the CSL deck is processed, the Hundreds/Thousands/Bias table in auxiliary storage module 2 does not contain a bias factor. The auxiliary storage contents without the bias factor are shown in the cross-reference listing. When the last card is processed, the bias factor is added into all the appropriate bytes.

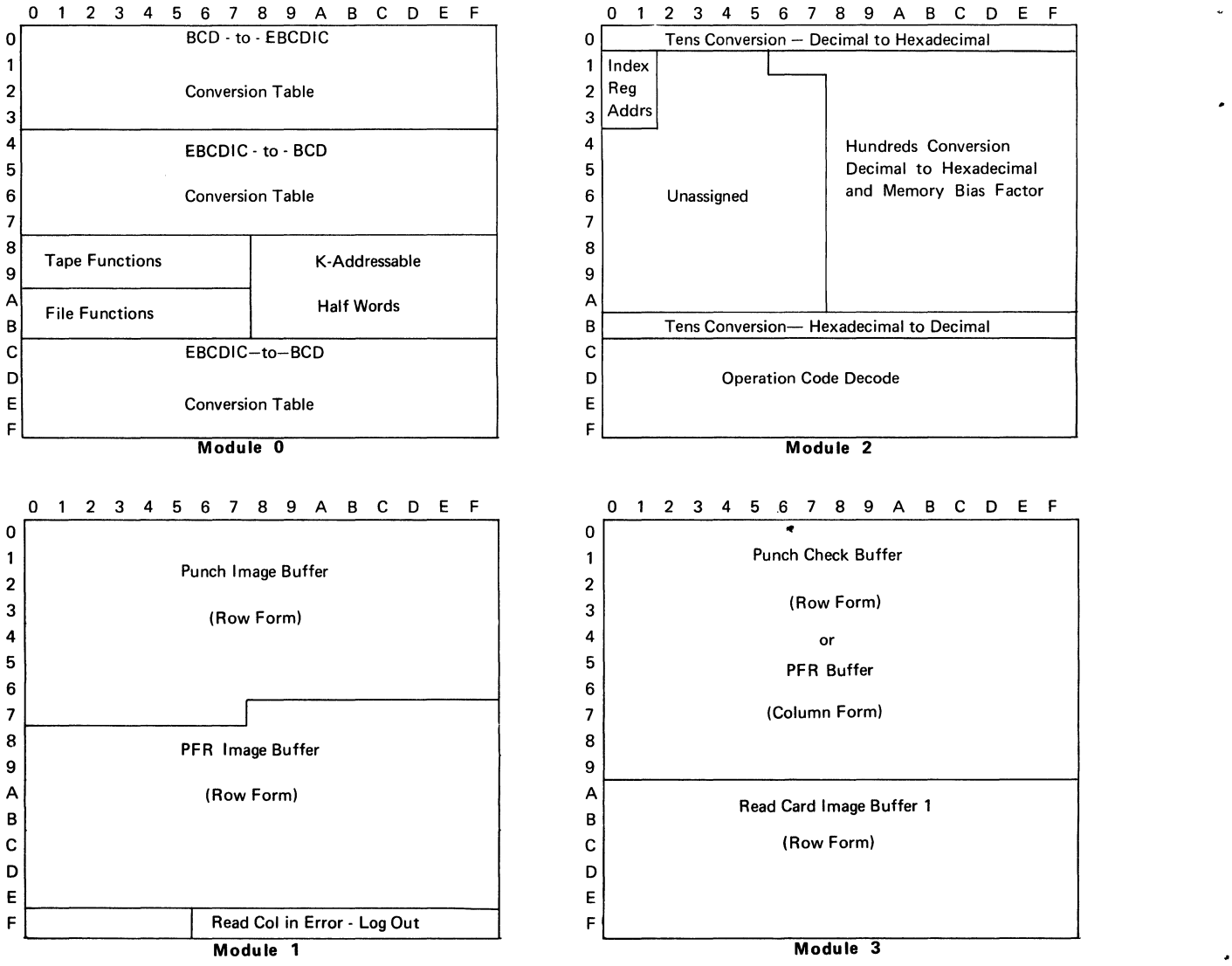


Figure 4-7. 1401/1460 and 1440 Compatibility Features Aux. Storage Map (Part 1 of 2)

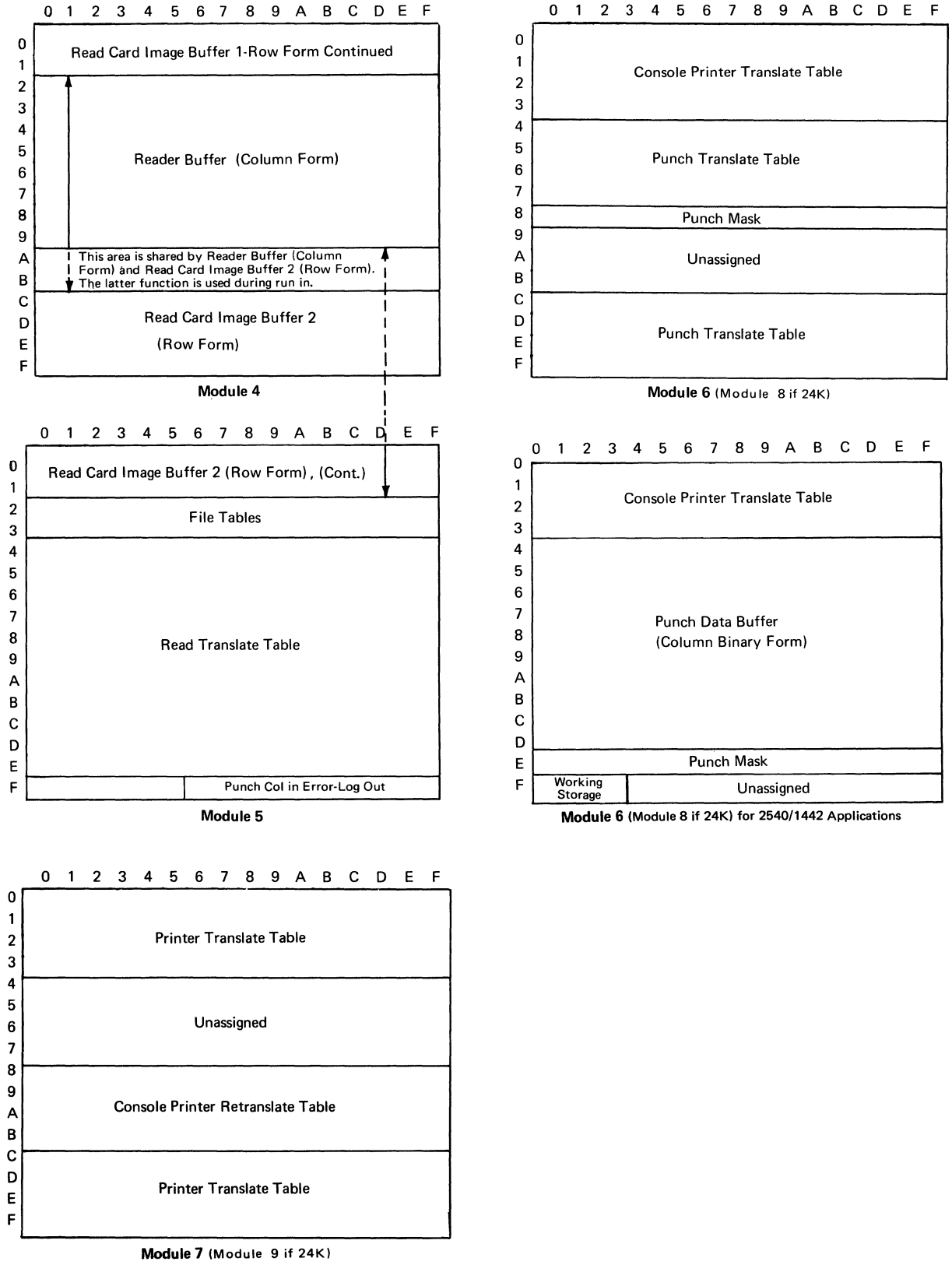


Figure 4-7. 1401/1460 and 1440 Compatibility Features Aux. Storage Map (Part 2 of 2)

AUXILIARY STORAGE MODULE 0

Rows 0-3; BCD to EBCDIC Conversion Table: These four rows provide the constants for translating the 64 possible BCD configurations to appropriate EBCDIC bytes.

Rows 4-7 and C-F; EBCDIC to BCD Conversion Table: These eight rows provide a 128-character table that contains a BCD code corresponding to each EBCDIC character. EBCDIC to BCD translation is required during execution of instructions such as Bit Tests, Move Zone, and Move Numeric.

Rows 8 and 9, Bytes 0 through 7; Tape Assignments: The use of the tape control and initial load bytes in auxiliary storage module 0 is as follows.

Byte	Bit	Use
	0	Last command Write or WTM
80	1-3	Last 1400 tape unit addressed (initialized to 000)
	4-7	Tape Control Unit Address
81	0-1	Tape density for 1400-mode tape drive 1: 00=200 bpi on 7-track tape drive 01=556 bpi on 7-track tape drive 10=800 bpi on 7-track tape drive 11=800 bpi on 9-track tape drive
	2	If on, a backspace was the last operation performed on 1400-mode tape drive 1 (initialized to 0).
	3	If on, and end-of-file condition is outstanding on 1400-mode tape drive 1 (initialized to 0).
	4-7	System/360 unit address assigned to be 1400-mode tape drive 1.
82-86		Same as byte 81, for 1400-mode tape drives 2 through 6.
87		Last status byte received from tape-control unit (initialized to 40).
88-8F		Refer to K-addressable halfwords.
90		Tape sense byte 0
91		Tape sense byte 1
92		Tape sense byte 2 (Tape Track in Error)
93		Tape sense byte 3
94-95		Tape C-Star backup location
96	0	Temporary command information
	1	1400 tape drive 1 EOF block bit
	2	1400 tape drive 2 EOF block bit
	3	1400 tape drive 3 EOF block bit
	4	1400 tape drive 4 EOF block bit
	5	1400 tape drive 5 EOF block bit
	6	1400 tape drive 6 EOF block bit
	7	Temporary command information
97		MCS control; Print Character Counter Length (PCCL)
		Bit Significance
	0	Unassigned
	1	240 character
	2	120 character
	3	80 character
	4	60 character
	5	48 character
	6	40 character
	7	16 character

Rows A and B, bytes 0 through 7 are used for Disk File functions:

Byte	Bit	Compatibility Mode Use
A0-A1		1400 Drive 0 Assignment
A0	0-3	Must be set to 0000
	4-7	Modular addresses 0 to 19,999=0000 Modular addresses 20K to 39,999=0010 Modular addresses 40K to 59,999=0100 Modular addresses 60K to 79,999=0110 Modular addresses 80K to 99,999=1000
A1	0-3	System/360 File Select Addresses: File 0 = 1000 File 1 = 0100 File 2 = 0010 File 3 = 0001
	4-5	Not used
	6	Compare Disable. 6=0; compare disable is inactive. A successful address compare on a 1400-series indelible address (I/A) must occur before an I/A read or write can be executed. 6=1; compare disable is active. Read or write with I/A is executed without first doing an address compare on the 1400-series indelible address. This bit should be set to 1 only when initializing a disk pack in 1400-mode.
	7	Module overflow detection. 7=0; module overflow detection is active. The 1400-program module value within each disk-control field is compared against a module value preset in auxiliary storage. If the module values disagree, a coded stop occurs

with 60 displayed. In addition, Unequal Address Compare is set. Correct module values must be set in auxiliary storage 0 locations A0, A2, etc.
7 = 1; module overflow detection is inactive. (Most user applications set bit 7 to 1.)

A2-A3, A4-A5, A6-A7 and B6-B7 represent 1400-drives 2, 4, 6, and 8 respectively. Assignments follow the same structure format as A0-A1. Compare Disable and Module Overflow Detection must be set only in the A1 byte.

B0-B5 Disk Error information for diagnostic use.

Rows 8 through B, Bytes 8 through F, K-addressable Halfwords: The 16 K-addressable halfwords of auxiliary storage module 0 are used for 1401/1460 and 1440 compatibility mode.

K-Adr	Byte	Bit	Use
K0	88-89		Bias constant. Refer to Figure 4-4 for correct value.
K1	8A-8B		Working storage
K2	8C-8D		B-Star backup for local storage V0, V1
K3	8E-8F		A-Star backup for local storage U0, U1
K4	98-99		Working storage
K5	9A-9B		Working storage
K6	9C (Integrated Attachment Printer)		
		0	132 print position 1403
		1	Channel 9
		2	Channel 12
		3	Invalid channel
		4	Secondary bit
		5	Device-end
		6	Print wordmark operation
		7	Printer error
K6	9C (Channel Attached Printer)		
		0	132-position 1403; or greater than 120 positions if 1443.
		1	Channel 9
		2	Channel 12
		3-7	Device address
	9D		Forms command
		0	1403 Printer (Integrated Attachment only)
		1	On Channel (Integrated Attachment only)
		2	
		3	
		4	Go to Set Up after Rmt/Rst (Integrated Attachment only)
		5	From General Stop Loop (Integrated Attachment only)
		6	Last CMD Skip or Spc Sup.
		7	Forms After Cmd
K7	9E		Reader-1 address
	9F		Reader-2 address
K8	A8	0	Sense switch A (last card)
		1	Sense switch B
		2	Sense switch C
		3	Sense switch D
		4	Sense switch E
		5	Sense switch F
		6	Sense switch G
		7	Sense switch A for second serial reader punch
	A9	0	High-compare result (U)
		1	Unequal-compare result (/)
		2	Low-compare result (T)
		3	Equal-compare result (S)
		4	Inquiry clear indicator (*)
		5	Overflow indicator (Z)
		6	Inquiry request indicator (Q)
		7	Not used
K9	AA	0	Not Move or Load instruction
		1	H-Typehead installed
		2	Secondary bit
		3	Load mode
		4	Stop message being performed (0-4 apply to PR-KB)
		5	Reader error
		6	Punch error
		7	Printer error
	AB		Unassigned

KA	AC	0	48K
		1	24K, 32K
		2	16K, 32K, or 48K
		3-7	Initialized to 1 0-7: High-order byte of highest storage address--initialized to 3F, 5F, 7F, or BF
	AD		1442/1443 status
		0	Skip GMWM check
		1	Channel-end only
		2	Last card on
		3	Column binary mode
		4	Print operation
		5	Output command
		6-7	00=R/W, 01=Test I/O, 10=Sense, 11=Control
KB	AE	0-7	Previous file operation
	AF	0	Read-back check interlock
		1	Recalibrate sequence bit
		2	Unequal-address compare (X)
		3	Access busy (\)
		4	Wrong-length record (W)
		5	Any disk error condition (Y)
		6	Disk error (V)
		7	Not ready (N)
KC	B8	0	I/O check-stop switch
		1	Not used
		2	Expanded print-edit feature
		3	Period-comma inversion (World Trade only)
		4	Column binary feature
		5	No punch buffer
		6	Model-G emulation
		7	51-column cards Usage of bits 5, 6 and 7 when using 2540 as 1442: 5 Simulate Read and Punch in same card 6 Select read error card to pocket R2 7 Use 2540 as 1442
	B9	0	Alternate read mode
		1	Not used
		2	Tape erase (initialized off)
		3	Alternate 9-track tape
		4	Temporary status
		5	Channel printer
		6-7	Sterling feature (World Trade only).
KD	BA	0	System/360 tape drive 0 is a 9-track unit.
		1	System/360 tape drive 1 is a 9-track unit.
		2	System/360 tape drive 2 is a 9-track unit.
		3	System/360 tape drive 3 is a 9-track unit.
		4	System/360 tape drive 4 is a 9-track unit.
		5	System/360 tape drive 5 is a 9-track unit.
		6	System/360 tape drive 6 is a 9-track unit.
		7	System/360 tape drive 7 is a 9-track unit.
KD	BB	0	0=0 tape drive 0=800 bpi density 0=1--1600 bpi
		1	1=0 tape drive 1=800 bpi density 1=1--1600 bpi
		2	2=0 tape drive 2=800 bpi density 2=1--1600 bpi
		3	3=0 tape drive 3=800 bpi density 3=1--1600 bpi
		4	4=0 tape drive 4=800 bpi density 4=1--1600 bpi
		5	5=0 tape drive 5=800 bpi density 5=1--1600 bpi
		6	6=0 tape drive 6=800 bpi density 6=1--1600 bpi
		7	7=0 tape drive 7=800 bpi density 7=1--1600 bpi
KE-KF	BC-BF		Working Storage

AUXILIARY STORAGE MODULE 1

Rows 0-7(7): Punch Image Buffer, (row form)
Rows 7(8)-E: PFR Image Buffer, (row form)
Row F: (F6-FF) Logout area for card read errors

AUXILIARY STORAGE MODULE 2

Row 0; Tens Conversion Constants: This row provides tens conversion for translating BCD to EBCDIC. 0A through 0F contain invalid digit values.

Rows 1, 2, and 3; Bytes 0 and 1; Index Register Addresses: These three halfwords give the addresses for the index registers for the tens character zones A, B, and AB respectively.

Rows 1-A, Bytes 8-F; Hundreds, Thousands and Bias Conversion Table: The values in this table are determined by the sizes of the 1401/1460 or 1440 being emulated and the size of the 2025 program storage area. Entry into the table is according to hundreds and hundreds zones (thousands). Thus, the result from the table solves hundreds, thousands, and bias translation

simultaneously. The halfword at 16 is an extension of the table and is used to translate the / character.

Row B; Hexadecimal to Decimal Conversion Table: This row gives the decimal equivalent of the 2nd-order hex digit value.

Rows C through F; Operation Code Decode Table: These four rows provide the translation of 1401/1460 and 1440 operation codes to a bit significant form that is usable by the 2025. In the chart (Figure 4-2), the 1400-series op-code is shown in parentheses. Code 34 is used to indicate an invalid 1400-series op-code. Code 06 is used to indicate a No-Op.

AUXILIARY STORAGE MODULE 3

Rows 0-9: Punch Check Buffer (row form) or PFR Buffer (column form).
Rows A-F: Read Image Buffer 1 (row form). Partial; remainder of buffer area is in module 4, rows 0 and 1.

AUXILIARY STORAGE MODULE 4

Rows 0-1: Read Image Buffer 1 (row form) continued from module 3.
Rows 2-B: Reader Buffer, (column form).
Rows A-F: Read Image Buffer 2 (row form), Partial; remainder of buffer is in module 5, rows 0 and 1. Rows A and B are shared with the column form buffer.

AUXILIARY STORAGE MODULE 5

Rows 0 and I: Read Image Buffer 2 (row form) Continued from module 4.
Rows 2 and 3: Disk File Tables, conversion tables for cylinder and head decode, file op-code conversion, etc.
Rows 4-E: read translate table.
Row F: (F6-FF) Logout area for card punch errors.

AUXILIARY STORAGE MODULE 6 (MODULE 8 IN 24K SYSTEMS)

Rows 0-3: Console Printer Translate Table
Rows 4-7: Punch Translate Table
Row 8: Punch Mask
Rows 9-B: Unassigned
Rows C-F: Punch Translate Table

Differences for 2540/1442 applications:
Rows 4-D; Punch Data Buffer (column binary form)
Row E; Punch Mask
Row F; Working storage and unassigned

AUXILIARY STORAGE MODULE 7 (MODULE 9 IN 24K SYSTEMS)

Rows 0-3: Printer Translate Table (Channel)
Row 4: Console Printer working storage area
Rows 5-7: Unassigned
Rows 8-B: Console Printer Re-Translate Table
Rows C-F: Printer Translate Table (Integrated)

LOCAL STORAGE

The 64-byte local storage is divided into the following six zones that are used in the same general manner in 1400 mode as in System/360 mode. The six zones are:
Zone 0, CPU Mode; 16 bytes
Zone 1, 2311 Mode; 8 bytes
Zone 4, Backup area; 16 bytes
Zone 5, Undefined; 8 bytes
Zone 6, 2540 Mode; 8 bytes
Zone 7, Channel Mode; 8 bytes

Local storage is used when operating in 1401/1460 or 1440 compatibility mode for intermediate storage for factors unique to this mode of operation. Example: I-Star, A-Star, B-Star, Op-register and A-register. In addition, local storage is used for problem-program factors and microprogram factors. Also, data stored in local storage is used by control words to perform some problem-program functions.

ZONE 0

Zone 0 is addressed when the Model 25 is operating in CPU mode. There are 16 bytes within the zone-0 area addressed by X-lines 0-7 and Y-lines 0 and 1. Not all bytes have assigned functions.

AS/BS Decode	Sym	X	Y	Assigned Function
0	U0	0	0	A-Address register
1	U1	1	0	A-Address register
2	V0	2	0	B-Address register
3	V1	3	0	B-Address register
4	G0	4	0	Working register
5	G1	5	0	Op-register
6	D0	6	0	Status Indicators
7	D1	7	0	A-Register
8	I0	0	1	I-Address register
9	I1	1	1	I-Address register
A	T0	2	1	Working register
B	T1	3	1	Working register
C	P0	4	1	Working register
D	P1	5	1	Working register
E	H0	6	1	Working register
F	H1	7	1	Working register

ZONE 1

Zone 1 is addressed when the Model 25 is operated in 2311 mode. There are 8 bytes within the zone-1 area addressed by X-lines 0-7 and Y-line 2, as follows.

AS/BS Decode	Sym	X	Y	Assigned Function
8	I0	0	2	Count register
9	I1	1	2	Count register
A	T0	2	2	Data Address register
B	T1	3	2	Data Address register
C	P0	4	2	
D	P1	5	2	
E	H0	6	2	
F	H1	7	2	

ZONE 4

Zone 4 is used as the backup area for all modes of operation except 2311 mode. There are 16 bytes within the zone-4 area addressed by X-lines 0-7 and Y-lines 3 and 4.

For CPU mode operation, the backup area can be addressed by setting the mode register bits 5, 6, and 7 to 1,0,0. For operation in 2540 or channel mode (MMSK bits 0, 2, 3, or 4 on), the low-order 8 bytes of zone 4 can be addressed. None of the bytes of the backup area can be addressed when operating in 2311 mode (MMSK bit 1 on).

All 16 bytes of the backup area can be addressed manually by console switches C and D. Assignments are as follows.

AS/BS Decode	Sym	X	Y	Assigned Function
0	U0	0	4	High-order Level-1 backup address
1	U1	1	4	Low-order Level-1 backup address
2	V0	2	4	High-order Level-2 backup address
3	V1	3	4	Low-order Level-2 backup address
4	G0	4	4	High-order Level-3 backup address
5	G1	5	4	Low-order Level-3 backup address
6	D0	6	4	High-order Machine-Check backup address
7	D1	7	4	Low-order Machine-Check backup address
8	I0	0	3	High-order CPU Branch and Link backup address
9	I1	1	3	Low-order CPU Branch and Link backup address
A	T0	2	3	Spare
B	T1	3	3	Spare
C	P0	4	3	Workarea
D	P1	5	3	Workarea
E	H0	6	3	Workarea
F	H1	7	3	Workarea

ZONE 6

Zone 6 is addressed when operating in 2540 mode (MMSK bits 3 or 4 on). There are eight bytes in the zone-6 area addressed by X-lines 0-7 and Y-line 6. Zone 6 may be addressed also when in CPU mode by setting the MODE register bits 5, 6, 7 to 1, 1, 0 respectively.

AS/BS Decode	Sym	X	Y	Assigned Function
0	U0	0	6	Reader Image Buffer Address
1	U1	1	6	Reader Image Buffer Address
2	V0	2	6	Punch Image Buffer Address
3	V1	3	6	Punch Image Buffer Address
4	G0	4	6	Reader/Punch Status
5	G1	5	6	Stacker-Select Information
6	D0	6	6	Punch Count
7	D1	7	6	Read Count

ZONE 7

Zone 7 is addressed when operating in channel mode (MMSK bits 0 or 2 on). There are 8 bytes in the zone-7 area addressed by X-lines 0-7 and Y-line 7. Functional assignments for these bytes are:

AS/BS Decode	Sym	X	Y	Assigned Function
0	U0	0	7	
1	U1	1	7	
2	V0	2	7	Data Address Register
3	V1	3	7	Data Address Register
4	G0	4	7	
5	G1	5	7	Data Register
6	D0	6	7	Status Register
7	D1	7	7	Word Separator

OPERATION

Operation of the System/360 Model 25 in 1400 compatibility mode utilizes the same microprogramming principles with the same capabilities and limitations as in System/360 mode. This section presents overall objectives of 1400 emulation without giving point-to-point detailed descriptions of control-word routines. Refer to Chapter 3 for a discussion of micorprogram control-word concepts, microlisting description, machine language tie-in, etc.

CPU OPERATIONS

- CPU Operations in 1400 mode are emulated by unique microroutines.
- The System/360 Model 25 in 1400 compatibility mode handles some conditions differently than the 1400 system.
- A minor circuit addition accommodates 1400-storage wrap detection.

CPU operational differences are largely concentrated in the execution of I-cycles. The variable length of 1400-series instructions necessitates a more complex microprogram routine for interpreting instructions and setting up conditions for execution. I-cycles are discussed more fully in the Section I-Cycles.

Execution of CPU operations such as Compare, Move, Add, etc., in 1400 mode are handled also by unique microprogram routines. These routines are patterned closely after System/360 mode routines. They are further clarified by comments that are part of the MAS program listing. This manual stresses I/O operations and the operational flow of microroutines. Major objectives are discussed. Refer to microroutine listings for step-by-step details of the operations.

CPU Differences: Functional differences are described in the SRL Manual IBM System/360 Model 25 1401/1460 and 1440 Compatibility Features, Form A24-3512.

Storage Wrap Detection Circuitry: As described in the section Address Error Detection, low-storage wrap detection in 1400 compatibility mode is accomplished by recognition of an 8F through FF character in the program storage area below the equivalent of 1400-storage location 000. Refer to MDM 4-14. Whenever the 8F-FF character is detected during a program memory-word cycle in 1400 mode, the line '8F Detected' is energized. This causes a storage wrap violation stop.

I-CYCLES

- 1400-series operation codes are converted to System/360 type (bit significant) operation codes.
- 1400-series addresses are converted to System/360 hexadecimal equivalent; address error detection is performed.
- Indexing is performed if designated.

The objectives of I-cycles in 1401/1460 or 1440 Compatibility mode are essentially the same as for I-cycles in the respective 1401/1460 or 1440 system. There are important additional (support) objectives that are necessary to accomplish the I-phase (conversion, etc.) in compatibility mode.

I-phase is of variable length.

Normally, a total of 9 I-cycles is possible; exact number of I-cycles taken is dependent upon the length of the instruction (Figure 4-8). One I-cycle is required for each character in the instruction word. One additional I-cycle is required to recognize the end of the instruction (the next instruction wordmark). In the microprogram listing comments, the instruction length designation refers to the cycle in which the WM is recognized. Example: I2 Cp is an instruction and a d-modifier, I4 is an instruction with an A-address, etc. Under some conditions, I-cycles occur until storage wrap or a WM is encountered.

I-OP	I-1 *	I-2 *	I-3	I-4 *	I-5 *	I-6	I-7 *	I-8 *
{wm OP	{wm OP of next instruction							
{wm OP	d	{wm OP of next instruction						
{wm OP	A	A	A	{wm OP of next instruction				
{wm OP	A	A	A	d	{wm OP of next instruction			
{wm OP	A	A	A	B	B	B	{wm OP of next instruction	
{wm OP	A	A	A	B	B	B	d	{wm OP of next instruction

* I - Phase ends on this cycle

Figure 4-8. 1400-Series Instruction Lengths

The conversion of 1400 decimal addresses to System/360 hexadecimal equivalent addresses is performed by table lookup, as described under Address Conversion. In a similar manner, operation codes are converted to obtain codes that are bit significant for easy identification by the microprogram. Refer to Op-Code Conversion and Recognition.

After the 1400-operation code is converted and placed in the G1-register of local storage, a variety of paths are available depending on type of instruction, length of instruction, etc.

I-Cycles Objectives

I-cycle objectives vary according to instruction length and type. There are 6 valid instruction lengths in 1400 compatibility mode: I1 (Operation-alone), I2, I4, I5, I7, and I8. Instruction length and op-code validity are indicated by the presence of wordmarks (bit 1 off).

I8 Objectives

1. Use normal I-cycle entry point (HISTRT). Read operation code; check for wordmark. Use table in auxiliary storage to convert 1400 series op-code to bit significant op-code and place it in local storage zone-0, G1-register.
2. Read I1 character (with no WM). Check for Q-op; initialize status register. Convert hundreds zones and bias from auxiliary storage table.
3. Read I2 (tens) character (without WM). Check for special character, invalid digit, etc. If there are tens zones, branch to the indexing routine (INDX). Perform the indexing operation and return at the SETTO label with complete indexed A-address.
4. If there are no tens zones, read I3 (units) character, and complete translation of A-address including thousands zones.
5. Read I4 character (without WM). Transfer address to A-Star, and set status. Examine and classify I4 character, (Branch, I/O Op, etc.). Translate I4 (address) from auxiliary storage.
6. Read I5 character (without WM). This step is a repetition of step 3, with I5 substituted for I2.
7. Repeat step 4 (if applicable) with I6 substituted for I3.
8. Read I7 character (without WM). Transfer address to B-Star. Set modifier character in D1-register.
9. Read I8 character (with WM) and set status. Decode the operation and branch to the appropriate routine for execution.

I7 Objectives

One through seven are the same as for I8 objectives.

8. Read I7 character (with WM). Transfer address to B-Star. Branch to OPI478. Decode operation and branch to appropriate routine for execution.

I5 Objectives

One through five refer to I8 objectives.

6. Read I5 character (with WM). Branch to I25OP. Set modifier (I4 character) into D1. Set status. Decode operation and branch to appropriate routine.

I4 Objectives

One through four refer to I8 objectives.
5. Read I4 (with WM), transfer address to A-Star and set status. Decode the operation and branch to appropriate routine for execution.

I2 Objectives

One and two refer to I8 objectives.
3. Read I2 character (with WM). Branch to I25OP. Set the modifier character in D1. Set status. Decode the operation and branch to the appropriate routine for execution.

I1 Objectives

1. Refer to I8 objectives.
2. Read the I1 character. Wordmark indicates Operation Alone. Set status and decode the operation. Branch to the appropriate routine for execution of the operation.

INDEXING

Indexing occurs during I-cycles but is accomplished by branching from the main I-cycle routine, executing the INDX microroutine, then returning to I-cycles. This can occur during A- and/or B-address generation.

The objectives of indexing in 1400 compatibility mode are essentially the same as for indexing in the 1400 system. As for I-cycles, there are some support objectives necessary to accomplish conversion.

Summary:
1. Branch on tens zone (I2 or I5 time) to indexing routine. P1 has the hundreds character necessary for total address. Use TLU, convert hundreds character to BCD form; set in D1.
2. Convert tens to BCD, set in H1. Save tens numeric in G0H. Read index register address into T-register.
3. Read units into P1. Read units BCD from auxiliary storage to P1. Set tens and units of basic address into G0. (G0H=tens, G0L=units.) Set units zones into H1HI.
4. Read units value from the index register into P1. Translate to BCD from auxiliary storage. Add units zones of the two fields, (address and index), set into H1.
5. Read tens value from index register into P1; convert to BCD. Read index register hundreds value into T1.
6. Move total units zones into T0. Combine index tens and units and add to regular A- or B-address tens and units.
7. Add hundreds from index and A- or B-address. Translate tens from decimal to hex, add tens and units.
8. Add tens and units to hundred/thousand/bias value for complete indexed address. Branch back to I-cycles for remainder of instruction read.

I/O OPERATIONS

- I/O operations in 1400 compatibility mode are executed in the equivalent of burst mode.
- Operation-code bit structure changes during the execution of an I/O operation.
- Reader-Punch and File operations utilize additional special circuitry.

Sequence of Operation for Combination Operations Is Print, Read, and Punch	
1400 Operation	
Read (1)	Starts as 21, and during read ending routine (LRDR) is changed to 20. In case of error (and the I/O Stop switch is on,) the code remains 21 and display stop occurs.
Print (2)	Starts as 22 and is reset to 00 at the end of the normal print routine (MPRT) before return to LOPD and ICYC.
Print, Read (3)	Starts as 23 and is changed to 21 at end of the print routine. Routine branches back to ICYC VALADR 0, is decoded as a 1 Op and executed.
Punch (4) Punch Col Bin (4C)	Starts as 24 and is changed to 20 during punch ending routine (LPCH). In case of error, code remains 24 and a display stop occurs.
Read, Punch (5)	Starts as 25 and is changed to 24 at read ending (LRDR). In case of error (with I/O Stop switch on) code remains 25 and display stop occurs. No error routine branches back to LOPD, performs punch op and is changed to 20.
Print, Punch (6)	Starts as 26 and is changed to 24 at end of Print routine. Routine branches back to ICYC VALADR 0 and is decoded as a punch op. Operation is then the same as a 4 op and is changed to 20 at the ending.
Print, Read, Punch (7)	Starts as 27, changes to 25 at end of print operation. 25 causes a read op, at end of read op code changes to 24 and causes a punch op. At end of punch op, code changes to 20. All error limitations for the various operations apply.
Punch Feed Read (4R)	Starts as 24. The d-modifier causes the op code to change to A4, then a normal punch routine is started. At the end of the punch routine the code is changed to 28 and a punch feed read operation is performed. At end op code changes to 20.
Read Column Binary (1C)	Starts as 21. Changes to 20 during read ending routine (RDREND). Error handling same as for Read op. (Validity not checked.)

Figure 4-9. Op-Code Information

The 1400-system I/O operations are executed in the equivalent of burst mode. I/O instructions are decoded in the I-cycle routine causing a branch to the appropriate I/O instruction routine for execution.

As some I/O operations are executed, the bit significant op-code changes indicating the status of the operation. For example, in executing the combination 'print-read-punch' instruction (for 1401/1460), the original converted op-code (27) changes to 25 (Read-Punch) upon completion of the print operation; to 24 (Punch) upon completion of the read operation; and to 20 (indicating operation completed) upon completion of the punch operation. Refer to Figure 4-9.

Various I/O operations are discussed under the appropriate headings. Attention is given to circuitry that is added to implement the differences in 1400-mode operations compared to System/360 mode operations. In most cases, operational differences are absorbed by microprogram routines and actual circuitry differences are slight.

1402 READ OPERATIONS

The functions of the 1402 attached to a 1401 or 1460 system are performed by an IBM 2540 Card Read Punch. When a 1402 is operating with a 1401 or 1460 system, there is a 10 millisecond period after a 'read a card' instruction during which a stacker-select instruction (to be effective) must be given.

When a 2540 is operating on a System/360 Model 25 in System/360 mode, a read command causes data to be transmitted to the CPU, but no card movement occurs.

To cause the 2540 (on a 2025 in 1400 compatibility mode) to duplicate the card feed action of a 1402 on a 1400 system, a 1400 delayed-read feed circuit has been added. Basically this circuit does the following: 5.5 milliseconds after the read command data transfer ends, a feed cycle occurs and a card is read and selected to the normal pocket. If, during the 5.5 ms timeout period, a 1401/1460 stacker-select command is sensed, a stacker-select command is issued to the 2540. This command causes the card to select to the stacker. Refer to MDM 4-104.

The microroutine for stacker select detects whether or not the 5.5 ms timeout is over by checking the RPS-2 bit. If the status of this bit indicates the 5.5 ms time has expired, the routine is ended, and no stacker select occurs.

Microroutine Objectives

Read a Card or Read a Card and Branch, 1401/1460 1 or 1 III: The read operation code is converted to 21 by a TLU routine during I-cycles. This code is bit significant and related to other I/O operations. This relationship becomes significant in the execution of combination I/O operations.

Operational Summary:
1. Decode 1 in I-cycle routine (ICYC) and branch to LOPD.
2. Set read in area, Column Form Reader Buffer, and count for either 80 or 51 column cards. Branch to LRXF ENTRY.
3. Transfer column form buffer (information from previous card) in auxiliary storage 4 into program storage (1400 001-080 equivalent) using TLU and read translate table in auxiliary storage 5. Check for validity and store blank (40) if character is invalid. Repeat data loop until character count is zero.
4. LXFR WAIT. Set Read operation indicator. Transfer row image buffer from auxiliary storage 3 and 4 into column image buffer in auxiliary storage 4. Set read count and buffer address; branch to LRDR RDREND.
5. LRDR RDREND. Reset previous errors, if any. Allow checks; set 1400 delayed feed. Read and clear EOF indicator. Check for error conditicns. Change op-code to 20. Branch back to LOPD NOTI25.
6. IOPD NCTI25. Test for combination operations. Branch if I4 or I5. Do IUBR for next instructicn address (set A-Star to I-Star), then return to ICY HISTRT. If not an I4 or I5 Op, go directly to ICY HISTRT for next sequential instruction.

Read Column Binary, or Read Column Binary and Branch; 1C or 1IIIC: The C-modifier distinguishes a read column binary operation from a read operation. The operation code is treated the same. The modifier sets column binary mode.

Operational Summary:
1. Decode 1 in I-cycles and branch to IOPD.
2. LOPD RDRPCH. Operation can be either I2 or I5; test for C-modifier and set column binary indicator.
3. Set read in area, column form buffer and count. Br to LRXF.
4. LRXF ENTRY. Transfer column form buffer into program storage area. Check validity. If valid, do TLU and store. If invalid, do not set error. Change character to 40 and store.
5. Upon completion of transfer to program storage area, set up for column binary mode. Read 2 characters at a time from column form buffer; convert to EBCDIC and store 401-480 and 501-580 in program storage area. When complete, branch to LXFR.
6. LXFR WAIT. Transfer row image buffer to column image buffer. Branch to LRDR.
7. LRDR RDREND. Reset previous errors, allow checks, set 1400 delayed feed, read and clear EOF indicator, check for error conditions. Change operation code and branch to LOPD.
8. LOPD NOTI25. Branch if I4 or I5. Do IUBR for next instruction address then return to ICY HISTRT. If not I4 or I5, go directly to ICY HISTRT for next sequential instruction.

1402 PUNCH OPERATIONS

1402 punch operations in 1400-compatibility mode are performed by the 2540. Differences that exist between the 1402 operations in the 1401/1460 systems and 1402 operations in the System/360 Model 25 using the 2540 are emulated by microprogram routines. There are no circuits added for 1402 punch compatibility.

A stacker-select command given in System/360 mode causes the card entering the punch station to be selected. A stacker-select command given for a 1402 punch in a 1401/1460 system causes the card at the punch-check station to be selected. In the 1401/1460 compatibility feature, microprogramming causes the stacker-select operation to emulate the 1401/1460 system.

Microroutine Objectives

Punch a Card, or Punch a Card and Branch; 4 or 4 III: The punch op-code is converted to bit significant value 24 by a TLU routine during I-cycles.

Operational Summary:
1. Decode 4 in ICYC and branch to LOPD.
2. LOPD RDRPCH. Test for column binary, combined or PFR. Branch to LPSU.
3. LPSU PCHSTI 0. Loop if busy, branch to intervention required if not ready, off line, etc. Set buffer addresses (punch image, punch check). Move data from punch image buffer (auxiliary storage 1) to punch check buffer (auxiliary storage 3).
4. Set up registers for addressing program storage area (1400 equivalent) 100-180.

Store punch complete indicator in location 100. Branch to LPXF.

5. LPXF BEGIN. Initialize for addressing translate table. Read data from program storage area (1400 equivalent 101-180), translate to row form, put in punch image buffer. When 80 columns are complete, branch to LPCH.
6. LPCH ENDING. Reset previous error conditions. Set punch image buffer address (1000). Set stackers and feed. Change op-code to 20. Go to LOPD.
7. LOPD NOTI25. Test for combination Ops. If I4 Op (4 III), do IUBR UNCDBR for next instruction address; if not I4, go to ICY HISTRT for next sequential instruction.

Punch Column Binary, or Punch Column Binary and Branch; 4 C or 4 III C: The punch column binary mode of operation is set by the C-modifier character.

Operational Summary:

1. Decode 4 op-code and branch to LOPD.
2. LOPD RDRPCH. I2 or I5 Op -- Read C-modifier, set column binary indicator. Branch to LPSU.
3. IPSU PCHSTT 0. Loop if busy, branch to intervention required if not ready, off line, error, etc. Set buffer addresses (punch image and punch check) and move data from punch image to punch check. Test for column binary mode, set up to address column binary area of 1400 program storage. Go to LPCB.
4. LPCB START. Update program storage address by 401. Initialize punch mask and read odd character from storage locations 401-480. Use TLU for BCD equivalent and build row image characters for upper half of card. When upper half is complete, read characters from 501-580 for lower half. When second half is complete, go to LPCH.
5. IPCH ENDING. Ending routine is the same as for punch. Refer to steps 6 and 7 of punch operation with substitution of I5 for I4 for branch instruction.

Punch Feed Read or Punch Feed Read and Branch; 4 R or 4 III R: This instruction cannot be combined. The op-code is converted to the regular punch op-code (24). The R-modifier sets the PFR indicator and changes the op-code to A4. The punch routine occurs first followed by the read operation.

Operational Summary:

1. Decode 4 Op in ICYC, branch to LOPD.
2. LOPD RDRPCH. Interpret I2 or I5. Test for PFR, read R-modifier and set PFR indicator (change op-code to A4). Branch to LPSU.
3. LPSU PCHSTT 0. Loop if busy, branch to intervention required stop, if applicable. Branch to PFRWRI, set PFR write mode. Set punch image and punch check buffer addresses. Transfer data and set conditions to address 101-180. Store punch completion indicator. Go to LPXF.
4. LPXF BEGIN. Initialize to address translate table. Read data from storage, translate to row form to build character in punch row image buffer. Branch to LPCH.
5. LPCH ENDING. Set punch image buffer address, set stackers and set feed. Change op-code to 28. Branch to LOPD.
6. LOPD NOTI25. Branch on PFR Op, (Bit 4 in op-code) to LPSU.
7. LPSU PCHSTT 0. Check for unusual conditions. Branch for PFREAD. Initialize for addressing column form buffer, and PFR image buffer. Branch to LXFR.
8. LXFR BYTECT. Transfer PFR image buffer (row form, auxiliary storage 1) into PFR buffer (column form, auxiliary storage 3). Branch to LRXF to transfer data to program storage.
9. LRXF ENTRY. Transfer PFR buffer, column form into program storage area with translation. Check validity. Branch to LRDR.
10. LRDR RDREND. Reset previous errors, allow checks. Branch for PFR ending. Read error control byte. Change op-code to 20. Branch to LOPD and to ICYC (via IUBR if applicable).

1403 PRINTER OPERATIONS

The System/360 Model 25 with the 1401/1460 Compatibility feature can process programs that utilize the 1403 printer. The printer can be either an integrated attachment or on channel.

The printer is buffered (standard feature) on the System/360 Model 25; however, there are no programming differences except for channel-9 and channel-12 interrogation. This is explained in the SRL Manual, Form A24-3512.

For print instructions, the microprogram analyzes the operation, sets up constants, PIBAR value, PLB count, data address, etc., and performs various details in preparation for print-instruction execution. The microprogram then enters a data loop that reads one character at a time from the program storage area (1400 equivalent 201-332), translates it by TIU from the printer translate table in auxiliary storage and loads it into the Print Line Buffer (PLB). The PLBAR (Print Line Buffer Address Register) is also loaded. This loop is repeated until the buffer is loaded, then print gate is set and the

printer performs the print-and-space operation. The operation code is reset and the program branches back to ICYC (via LOPD and via IUBR if applicable).

Microroutine Objectives

Print operations include (for the 1403 printer): 2 - print, 2 III - print and branch, 2 ¤ - print wordmarks, 2 III ¤ - print wordmarks and branch, 2S - print and space suppress, 2 III S - print space suppress and branch. For familiarization with the microprogram, consider a 2 instruction on an integrated attachment 1403 printer.

Operational Summary:

1. The 1401/1460 operation code 2 is converted to 22 during I-cycles and the microprogram branches to the routine MPRT.
2. MPRT PRTCMD. Check for validity, consistency of op-code with I/O installation. Check for forms after, print WM, etc. Assemble status bits.
3. Set data address equivalent to 1400 address 201. Set data count to 132. Branch to SIOROU.
4. Check for not ready, error, etc. Set up for control command. Branch to WRTCMD. Load carriage data. Set constants PLB count, PLB table address constants, PLBAR constant and count. Branch to PRBULO for data loop.
5. (PRBULO). Read data character from program storage. Adjust for TLU. Load PLBAR and set read call. Load translated character into PLB. Decrement data count; if not zero, decrement PLB count and continue buffer load. Update PLBAR count to address the buffer in subscan sequence. Repeat PRBULO to read another data character.
6. Step 5 is repeated for each data character. When buffer is loaded (data count and PLB both 0), the loop is exited to STRPRT.
7. Reload D-register, set print gate latch to start print operation. Set busy latch. Restore various conditions, store status, etc. Subtract 2 from op-code. Reset to 00 if not combination op; otherwise, execute next I/O Op. If not combination, branch to LOPD.
8. LOPD TESTBR. If I4 or I5 Op; do IUBR; otherwise, branch to ICYC HISTRT.

1443 PRINTER OPERATIONS

When using the 1440 Compatibility feature, operations of the 1443 printer can be emulated by a channel-attached 1443 Model N1 printer. The buffering available with the 1443-N1 is effective during the operation in compatibility mode without program modification.

There are some graphic differences that exist between the 1400-series 1443 typebars and the 1443-N1 typebars. These differences are resolved by initialization of auxiliary storage during CSL.

For print instructions, the microprogram analyzes the operation, sends device address, issues write command, sends service out and performs the necessary details in preparation for print-instruction execution. The microprogram then enters a data loop that sends 1 character at a time to the print buffer. Table lookup is performed for character conversion.

Detection of the last character to be printed terminates the data loop and performs the routine for disconnecting electrically from the channel. The op-code is changed to indicate completion of the print operation and the next instruction is read out.

Microroutine Objectives

The print operations for the 1443 printer are M %Y1 BBB W - print and single space and M %Y1 BBB S - print and suppress space. Consider the print and single space operation.

Operational Summary:

1. The 1440 Move I/O instruction (M) is converted to 80 during I-cycles. The microprogram branches to IOCM to decode the unit address.
2. IOCM LOAD. This routine decodes the unit address, Y, causing a branch to routine MKKK.
3. MKKK STRT43. Decode the modifier (S or W) and set the command (assume write and space). Verify channel device, read-printer address byte, set count. Branch to MAAA.
4. MAAA RESLCT. (Initial selection and sense shared with the 1442.) Block traps, set mode, set Service-Out. Wait for no Operational-In and set Address-Out/Select-Out. Wait for Operational-In and reset Address-Out. Wait for Address-In. Compare addresses and if OK, branch to MJJJ.
5. MJJJ PRNTR. (Continuation of initial selection.) Decode status bits 6, 7. Move command, analyze space-and-skip status. Branch to MKKK.
6. MKKK SETCMD. Examine unit check bit, set Command-Out, wait for Status-In. (Branch if busy), clear old status. (Branch if Unit Check or Channel End), set Service-Out, Allow Traps, and go to data loop in MLLL.
7. MLLL SVCIN. Wait for Service-In. Read

data character from B-field. (Branch if GMWM), set up TLU address, read translated character and put on bus-out. Set Service-Out.

8. Branch back to SVCIN and repeat step 7, (data loop) until either GMWM character indicates end of data field, or Status-In indicates that the print buffer is full.
9. Detection of GMWM sets Command-Out. The response, Status-In, is checked, then the routine branches to MBBB for normal 1443 end.
10. MBBB ENDOK. Set Service-Out, allow traps and CPU mode. Branch to ICYC HISTRT.

1442 CARD READ-PUNCH OPERATIONS

The functions of the 1442 Read Punch attached to a 1440 system are performed by a channel-attached IBM 1442-N1 or 1442-N2. A GMWM is required to stop the read or punch operations. Reader input is validity-checked.

For read instructions, the microprogram routine analyzes the operation, initializes registers, blocks traps, sets mode and performs initial selection routine. The operation type is decoded causing a branch to the data loop. The data loop reads 1 character at a time into the program storage area designated by the BBB address. This continues until either a GMWM is encountered in storage, or device-end from the reader indicates column 81, or both. When channel-end (GMWM) is encountered first, the routine waits for status-In with Device-End. The data loop is terminated, the ending sequence occurs and the microprogram returns to I-cycles for the next instruction.

Microroutine Objectives; Read Operations

The read (or input) operations for the 1442 Card Read-Punch are: read a card; M %G1/2 BBB R and read card image; M %G9/0 BBBR. For discussion, consider the read card instruction on a channel-attached 1442-N1.

Operational Summary:

1. The 1440 Move I/O instruction (M) is converted to 80 during the ICYC routine. The microprogram branches to IOCM to decode unit address.
2. IOCM LOAD. This routine decodes the unit address, G, and the microprogram branches to the modifier decode and initial selection routine, MAAA.
3. MAAA START. Save I- and A-Stars and status byte. Reset previous errors. Decode R-modifier character. Set read command. Read address and status of sense switches. Decode unit number 1 or 2 (9 or 0 for card image). Assume SRP1.
4. Store address and sense switches. Block traps and set mode. Set Service-Out and wait for No Op-in. Set Bus-Out, set Address-Out/Select-Out, wait for Op-In. Reset Address-Out, wait for Address-In and compare addresses. If Address OK, go to MBBB, go to MBBB, Read Command and status loop.
5. MBBB RDWR. Set Bus-Out, Command-Out and wait for Status-In. (initial status), check for unit exception, unit check (channel-end or device-end) busy, etc. If OK, branch to data loop, MCCC.
6. MCCC DATA. Wait for Service-In. Use Input routine. Get data from Bus-In. Read B-field and check for WM and GMWM. Check character for validity (quadrant 1 or 3). Use TLU and check 1-bit status of character from table. Set error if invalid. Store character from Bus-In. Set Select-Out, wait for next character. (MLP character is handled by manipulating character from table and storing it.) Repeat loop, bring in data and store it until GMWM is in storage or device-end (column 81). Normally, for 80 column read, Status-In will occur with Channel-End and Device-End both up. Status-In causes branch to MBBB, status loop.
7. MBBB STATLP. Check status, set Service-Out, Branch on Channel-End, Device-End. Check for errors. Go to End OK. Set Service-Out, allow traps, set CPU mode and return to ICYC.

Microroutine Objectives, Punch Operations

For Punch instructions, the same decode and selection routines are used as for the read instruction. The punch instruction uses the output portion of the data loop. The B-field character is read, tested for GMWM and put on the channel bus-out. This loop is repeated until a GMWM is encountered in storage, or device end occurs.

The punch (output) instructions for the 1442 card read-punch are punch and stop; M %Gn BBB P, punch and feed; M %Gn BBB G, and punch and skip; M %Gn BBB C. These same operations can be performed in punch image (column binary) mode. For general familiarization, consider the M %G1 BBB P operation.

Operational Summary:

1. The microprogram branches from ICYC to IOCM.
2. IOCM LOAD. Decode unit address G, branch to modifier decode and initial selection routine, MAAA.
3. MAAA START. Save registers, reset previous errors. Decode P-modifier, set punch and no go command. Read address and sense switches. Decode unit number. Store address and switches.
4. Block traps and set mode. Perform selection (refer to read operation), branch to MBBB.
5. MBBB RDWR. Complete selection, check conditions, branch to data loop, MCCC.
6. MCCC DATA. Wait for Service-In. Go to output routine. Read B-field, test for GMWM, set Bus-Out and Service-Out. Repeat loop until GMWM or Status-In/Device-End. On GMWM, send Command-Out, wait for Status-In. Normally, Channel-End and Device-End will be together. Branch to MBBB.
7. MBBB STATLP. Check status, set Service-Out, branch on CE and DE. Check for errors, go to End OK. Set Service-Out, allow traps, set CPU mode, return to ICYC.

DISK COMPATIBILITY OPERATION

- 1311 Disk Packs must be reloaded and reformatted (by utility programs) to enable operation with the 2025.
- The 1311 file must be loaded into cylinders 1 to 100 on the 2311 to provide compatibility. Microprograms increment and decrement 1400 cylinder values as needed to compensate.
- RBC can be overridden for diagnostics by altering one microprogram word.

The Disk Storage Operations compatibility feature permits processing IBM 1311 magnetic disk-file records in either track or sector mode in 1400 compatibility mode using the integrated attachment IBM 2311 Disk Storage Drive.

The 2311 uses the same disk pack (1316) as the 1311. However, because of increased recording density and different format, existing 1311 files must be reloaded using the 2311 on the 2025. The 1311 file must be loaded into cylinders 1 to 100 on the 2311.

System/360 Model 25 file-unit addresses must be assigned to correspond to 1311 module numbers. This is done during initial control storage load by the system initializer card. Ten bytes are reserved in auxiliary storage for this purpose.

Because formatting is a major consideration and must be done before compatibility operations can be performed, first examine formats before discussing actual operations.

Formats

Home Address: The home address is the binary equivalent of the cylinder and head physical location. Refer to Figure 4-10. Home addresses are prewritten by a utility program.

Home addresses are not used directly in the execution of compatibility mode file operations. The home address serves as a reference point. The symbols F, C, C, H and H stand for Flag, Cylinder, Cylinder, Head and Head. The values of the first C and the first H (from left to right) are fixed at 0.

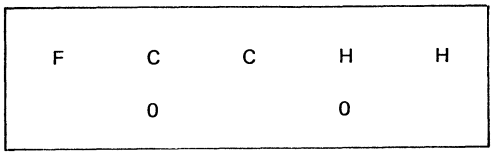


Figure 4-10. Home Address Format (Showing Examples)

Record 0 (R0): Record Zero (R0) for each track is prewritten by a utility program. Record zero is normally the same as the physical cylinder and head address. An exception exists when an alternate track is assigned to replace a damaged or defective track. Refer to Alternate Track Operation.

Record zero format, count, and data fields are shown in Figure 4-11. The data field is normally 8 characters. It is used for handling alternate track situations in 360 mode operations. The R0 data field is not used in 1400 compatibility mode.

R0 Count								R0 Data							
C	C	H	H	R	K_L	D_L	D_L	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8
0		0		0	0	0	8	X	X	X	X	X	X	X	X

Figure 4-11. Record Zero Format (Showing Fixed Values)

Count, Key and Data Fields (CKD): The count field provides an indelible address (I/A) for each record. Refer to Figure 4-12. The count field for sector mode operations specifies a data length of 100. The count field for record mode operations specifies a data length of 2980.

When writing a data field in load mode, the data transfer is truncated after the 90th data character is transferred to the file control unit. Short records are filled with valid blanks (40 hex) through the 90th character. The file control unit fills the 91st through the 100th character of the record with all zero-bit bytes.

When writing a data field in move mode and a short record is encountered, the 1400 compatibility microprogram sends valid blanks (40 hex) to the file control unit until a transfer of 100 or 2980 characters is completed. This is done for short records to provide correct mode detection when reading. Refer to Error Checking and Branch Conditions.

Sector Operations															
	Count								Data						
	C	C	H	H	R	K_L	D_L	D_L	D_1	D_2	D_3	⟨⟨	D_{98}	D_{99}	D_{100}
Numeric Value	0		0			0	0	100							
Hexadecimal Value	00		00			00	00	64							
Full Track Record Operations															
	Count								Data						
	C	C	H	H	R	K_L	D_L	D_L	D_1	D_2	D_3	⟨⟨	D_{2978}	D_{2979}	D_{2980}
Numeric Value	0		0			0	2980								
Hexadecimal Value	00		00			00	0BA4								

Figure 4-12. Count, Key, and Data Format (Showing (Fixed Values)

Record Number: Record numbers in the count field are numbered 1 through 20. 1400-series record numbers are incremented by 1 before being written on a disk and before executing compare address. Conversely, after a record number is read from a disk, it is decremented by 1 before it is stored in program storage area. This handling of record numbers permits a record zero, while maintaining a unique number for each data record.

Head Number: The head numbers that are written on the disk (0 through 9) are the same as the head value specified by the disk control field.

Cylinder Number: The cylinder numbers that are written on the disk are one greater than the cylinder value specified by the disk control field. These values are 1 through 100.

Module or Unit Number: In 360 format, a module number is not recorded on the disk pack as was done in 1400 disk operations. Thus, there is no protection against accidentally using the wrong pack. Module overflow detection, however, is provided.

Data Field: Data is written on the disk in EBCDIC code. Each record is written in the data field identified by its unique count field.

File Circuitry

SLT circuit cards are added for file compatibility. The additional circuits perform the functions indicated. They are identified by the logic page they appear on and by their logical function.

Logic FA 597 - Move/Load Latch: This latch is used for wordmark handling and mode error detection. It is set and reset by the microprogram set/reset word SET/RST DIAC K=04.

Logic FA 222 - Wordmark Latch: This latch is set or reset during the prefetch data cycles (KBBC) according to the bus-out 1-bit status (wordmark) of the data byte.

Logic FA 133 - Group Mark Word Mark latch (Block Share Request): This latch is set during the prefetch data cycles if a GMWM (0F) is read out of storage. This latch, when on, blocks share cycles.

Logic FA 611 - Mode Error Latch: This latch is set during read data operations if the mode of read operation being done is

not the same as the mode in which the data was written.

Logic FA 591 - Write Pad: This circuitry controls the forcing of data (blanks) on write ops in move mode when a short record is encountered.

Logic FA 595 - Scan and RBC: This circuitry controls the various gates etc., needed to control the compare circuits for scan and read back check. This page also contains a portion of the force data controls (write pad).

OPERATIONAL OBJECTIVE

Disk operations, as in 1400 series, fall into three major groups: Seek, Read or Write. The 1400 operation code is as follows:

M/L %Fn BBB R/W

Values of n and meanings are:
0 = Seek
1 = Sector mode
2 = Track Record mode
3 = RBC
4 = Not used
5 = Sector Control Overlay
6 = Sector mode with I/A
7 = Scan (Lo or Eq)
8 = Scan (Eq)
9 = Scan (Hi or Eq)
@ = Track Record mode with I/A

Seek Command

Seek commands are either return to home (RTH) or direct seek. Direct seek, an option with 1400 series, is standard with the disk storage operations feature of the 1400 compatibility feature. Refer to MDM 5-735.

RTH Seek: To execute an RTH seek, the disk control field (DCF) is decoded into a binary cylinder value. A table of cylinder values located in auxiliary storage is used by the microprogram to convert the 1400 DCF into a binary value. The microprogram decodes the present cylinder value of the 2311 by using external registers, then issues a seek command, either forward or reverse according to the computed difference of the cylinder values. For all 1400 seek commands (M/L %F0 BBB R/W) head number one is selected.

Direct Seek: To execute a direct seek, the disk control field is decoded to determine the number of cylinders and the direction of the seek movement. The operation is then handled the same as RTH seek.

Note: Prior to issuing any read or write data commands the microprogram selects the correct head as decoded from the disk control field.

RTH Seek (Recalibrate): A re-orientation/recalibrate seek command chain is initiated by the following sequence of events:
1. A read or write operation resulting in X and not WLR branch indicators
2. Seek command (Direct or Return to Home)
3. A second read or write resulting in X and not WLR

This sequence discriminates between cylinder overflows and access disorientation, and provides seek recovery without impairing throughput time.

Cylinder Displacement: During execution of a seek op-code, the cylinder value is incremented by one. Thus, a seek to cylinder 0 actually places the access at cylinder 1. This reserves cylinder 0 for the Initial Program Load (IPL) function. 1400-series programs occupy cylinders 1 through 100.

CAUTION
1311 Diagnostics (4F series) should not be run on a CE disk in compatibility mode. Adjustment data will be erased.

Direct Seek Special Case: Some existing customer programs cause a direct seek in the reverse direction (outward) with a number-of-cylinders value that results in a cylinder value of less than zero. The 1311 drive will seek to minus one cylinder, turn around and finish out the seek in the forward direction. This behavior has been simulated arithmetically for compatibility operation to provide program compatibility.

Read/Write Operations With Indelible Address

A head seek must precede all read/write commands to ensure that the head specified by the disk control field is the head selected. If a search equal ID command is issued to emulate an address compare function, two equal address compares are necessary. First a compare equal on a 1400 mode formatted indelible address must be made. (For disk packs not formatted in 1400 mode, this first compare is bypassed by setting the auxiliary storage position 00A1 bit 6 to 1.) Upon satisfying the first compare equal, the cylinder value in the compare argument is set to the seek cylinder value. The record number is set to 0 and an address compare equal on record zero must be satisfied. Refer to MDM 5-737.

This provides correct time orientation so that subsequent indelible address and data fields will be written in correct locations.

Substitution of the seek cylinder value is necessary to provide for cases where the disk control field contains an abnormal cylinder value. Cylinder-overflow recognition is provided when compare disable is active.

Cylinder Values: The cylinder value written on the disk pack is one higher than the value specified by the disk control field. Because all seeks are microprogram incremented by one, the R0 cylinder value, written in 360 mode, will be the same as the indelible address cylinder value written in 1400 mode (in most cases).

When indelible addresses are read from a disk, cylinder value is decremented by one before being placed in program storage.

All indelible address search arguments taken from the disk control field are microprogram incremented by one.

Head Values: The Head value during the address compare operation will be the same as the value specified by the DCF.

When formatting a disk in 1400 mode, the head specified by the disk control field must be the same as the head that executes the command. An exception is made for alternate track operation. Module Overflow Detection: This function is necessary when a 1400 program uses a change in module value in the disk control field to set the no address compare (No-X) branch indicator and branch to a seek routine.

Module overflow detection (Aux 0 - Byte A1 - Bit 7=0) is accomplished by comparing the disk control field module value to a corresponding module value that is preset in auxiliary storage. Module values are:

1400 Drive No.	Aux Storage	Normal Value
0	00A0	00
2	00A2	02
4	00A4	04
6	00A6	06
8	00B6	08

The module values must be set in auxiliary storage during the CSL routine. These values are entered via the system initializer card. When a module mismatch is detected, a coded stop occurs and 60 is displayed by the printer keyboard.

Module overflow or mismatch detection can be made inactive by setting byte A1 bit 7 to 1 in auxiliary storage 0.

Missing Address Mark Detection: The compatibility microprogram checks that the last record number read into memory on address ops is 16 or greater. If the last record number is less than 16, it is assumed possible that address marks might have been missed, and the no address compare (X) branch indicator is set.

Zero the DCF 6th Digit: When reading with indelible addresses into program storage area, the 6th digit of the disk control field is set to zero. This prevents residual characters in the program storage area from being transferred to tape or other permanent records.

Read Back Check (RBC): Read back check interlock is provided to ensure that a RBC instruction follows each write instruction. If an instruction other than RBC follows a write operation, the instruction is not executed and the system returns to I-cycles and stops.

It is possible to override RBC interlock for testing purposes by altering a branch condition in the microprogram. Refer to microroutine KAAA. The statement Branch to KEND STOP10 must be changed to No-Op (0000). Remember to restore the original statement upon completion of testing.

Wordmarks and Zone Bits in Control Fields

Disk Control Fields: The disk control field wordmarks, zone bits, and numeric values are unaltered during the execution of read/write operations with indelible addresses because no arithmetics are performed on the DCF.

Sector Count Field: The sector count is decremented each time a record is transferred to or from a disk. The sector count field is modified to conform with 1400 series arithmetic operations as follows:

Load Mode: Zone bits and wordmarks are removed from the sector count field.

Move Mode: Zone bits are removed, wordmarks are retained in the sector count field.

At the end of a read or write operation--(free of error conditions), the final A-Star value is BBB+9.

Read/Write Operations, Sector Mode

This operation is a normal customer application.

As previously stated, a head seek command precedes all read/write commands. A search equal identification (ID) command is issued to emulate an address compare. The record number in the search argument

will be the binary equivalent of the 1400-series record number, plus 1. Data record numbers on a disk are 1 through 20.

The head number in a search argument will be the binary equivalent of the 1400-series head number in the DCF. The cylinder value in a search argument will be the binary equivalent of the 1400-series cylinder number in DCF plus 1. Refer to MDM 5-739.

Scan Operations: The scan feature is standard with the disk storage operation compatibility feature.

Overflow Conditions: This section discusses the overflow conditions affecting

Head - When execution of a read/write operation requires head switching, the multitrack bit is set on. This causes the next head to search the first record.

Cylinder - When the execution of a read/write operation goes beyond the last record on surface 9, head value is set to 0 and cylinder count is incremented by 1. This forces a no-address-compare branch condition to be set following the next search. At this time WLR branch condition is also set to simulate 1400 operation.

Module - When the execution of a read/write operation exceeds the capacity of a drive unit or module, the disk control field is updated to the first address on the next module. If the 5th drive unit gets a module overflow (exceeds 099999), the resulting disk control field following incrementation is 000000.

Wordmarks and Zone Bit Handling--Sector Mode: The DCF wordmarks and zone bits are altered whenever the original sector count is two or greater. In load mode, the zone bits and wordmarks are removed from the disk control field in program storage. This does not include the alternate module select position that is not processed arithmetically during incrementation.

In move mode, except for alternate module select position, the DCF zone bits are removed and the wordmarks are saved in program storage.

The sector count field wordmarks and zone bits and final B-Star values are treated the same as for indelible address values.

The final A-Star value upon error free completion of a read/write operation is BBB+6.

Error Checking and Branch Conditions: Error conditions posted by the file control unit are interpreted and translated into 1400-series branch conditions as follows, (branch byte 00AE in auxiliary storage 0).

Bit	Condition	Symbol
0	RBC Interlock on	
1		
2	No Address Compare	X
3	Busy	\
4	Wrong-Length Record	W
5	Any Disk Condition	Y
6	Parity	V
7	Not Ready	N

An unusual condition in the file control unit as a response to a file select (initial status) is interpreted directly as a not-ready condition and the not-ready branch condition is set.

When an unusual condition is posted by the file control unit as a part of ending status, the microprogram examines the file external registers to determine the type of error that occurred and then post the corresponding 1400-error branch conditions in position 00AF in Auxiliary Storage 0. The following registers and bits are checked.

External Register: Disk Attachment Status-In (DASI) 0101
Bit: 1
Significance: Select Gated Attention
1400 Error: N

External Register: Terminating Conditions (TC) 1010
Bit: 0
Significance: Data Check in count
1400 Error: X+V
Bit: 1
Significance: Track Overrun
1400 Error: N
Bit: 2
Significance: No Record Found
1400 Error: X
Bit: 4
Significance: Data Check
1400 Error: V
Bit: 5
Significance: Overrun
1400 Error: V
Bit: 6
Significance: Track Condition check
1400 Error: Alternate Track Seek

External Register: File Gated Attention (FGA) 1100
Bit: 4
Significance: Wrong-Length Record
1400 Error: V (mode error)
Bit: 5
Significance: Missing address mark
1400 Error: X

External Register: Disk Status (DS) 1110
Bit: 0
Significance: Ready
1400 Error: N
Bit: 1
Significance: On line
1400 Error: N
Bit: 2
Significance: Unsafe
Bit: 5
Significance: End of Cylinder
1400 Error: X+W
Bit: 7
Significance: Seek Incomplete
1400 Error: N

If Busy is caused by a microprogram decision (example: busy because of a seek to an alternate track), the microprogram will loop in initial selection sequence until the busy is cleared, then issue the command.

If Busy occurs under conditions that the macro programmer would not anticipate (example: a RBC instruction is specified by the macro programmer and the programmer expects the file to be not busy), the microprogram must take the initiative looping until the busy condition is cleared, then issue the command.

In situations where a macro programmer can anticipate a busy condition, the busy branch condition is set when a busy condition is detected.

For Wrong-Length Record, a counter is set for each data transfer and decremented during the data transfer. When the group mark with a wordmark in program storage and a count of 0 do not coincide, the Wrong Length Record (WLR) branch condition is set.

On scan operations, WLR is set only if the groupmark with wordmark fails to precede the end of data field by two bytes or more.

Mode Checking: Checking is done by hardware circuitry to ensure that records are read from a file in the same mode as they were written on the file.

Read Load Mode Check--Each data character read from the disk is examined for a wordmark. If the record was written in load mode, the 91st character should always contain a wordmark. If this requirement is not met, the validity condition is set to simulate 1400-series setting. Reading full track record in load mode, the 2683rd character should always contain a wordmark.

Read Move Mode Check--Each data character is examined for a wordmark. The 100th character (and preceding 99 characters) should not have a wordmark. If this requirement is not met, the validity condition is set. Reading full track record in move mode, each character is examined also.

Alternate Track Operation

Alternate tracks can be assigned for any imperfect or damaged tracks on a disk pack. Provision is made to seek an alternate track when necessary, process, then return to the original track either to continue processing with the next sequential head or to end the operation.

An alternate track situation is recognized following an address compare operation (Search ID).

Alternate Track Formatting: Alternate track formatting (by a formatting program in 360 mode) is prerequisite to alternate track operations in 1400 compatibility mode.

Alternate tracks will normally be assigned to cylinders 200, 201, and 202, but this is not necessary on disks used only for 1400 file programs.

An alternate track can have a head assignment different from the head value of the original track. It is possible to write other than normal indelible address cylinder values on an alternate track, but not an abnormal head value.

A normal address is an address that could be expected to appear on the original track.

MAGNETIC TAPE OPERATIONS

When using the 1401/1460 or 1440 Compatibility feature, operations of the 729/7330/7335 Magnetic Tape units can be emulated by Selector Channel attached 2400 Model 1 or 2415 Tape units.

There are some important differences between tape operations in 1440 compatibility mode on the 2025 and in the 1401 or 1460. The 1401 or 1460 stores an end-of-file in tape unit as a tape indicate. Tape indicate is reset by unloading the tape unit or by branching on the end-of-file condition. In 1400 compatibility mode on the 2025, the end-of-file condition is stored as a bit in the tape unit control byte in auxiliary storage. This bit is reset by a rewind-unload instruction or a branch on end-of-file. The tape indicate in the tape unit is set only by the end-of-file reflective strip on the tape during a write

instruction and is reset by any backward command.

Because the end-of-file bit in auxiliary storage is not reset by manually unloading the tape unit, the operator must ensure that the bit is reset when reloading the tape unit to eliminate false end-of-file conditions.

A tape error during initial program load causes a microprogram stop with a coded error message. The tape control information is in auxiliary storage module 0, bytes 80 through 87. Refer to Auxiliary Storage, for the significance of control bytes.

For tape instructions the operation is quite similar to other operations on channel. The microprogram routine analyzes the operation, initializes registers, blocks traps, sets mode, and performs the initial selection routine. The operation type is decoded causing a branch to the tape data loop. The data loop reads or writes one character at a time in burst mode. The operation is usually terminated by a GMWM. This resets burst mode and causes the end routine to be executed. The microprogram returns to I-cycles. A read operation could also be terminated by reading an interrecord gap.

Microroutine Objectives

Tape operations are handled by variations of basic 'I/O move and load' instructions. For familiarization with the microprograms that execute tape instructions, consider the instruction Write Tape with Wordmarks; L %U1 BBB W.

Operational Summary:

1. The 1401/1460 load I/O instruction (L) is converted to 90 during I-cycles. The microprogram branches to IOCM to decode the unit address.
2. IOCM LOAD. This routine decodes the unit address U causing a branch to the JTPE routine.
3. JTPE UADMCK. This routine tests the tape unit address and d-modifier. Check validity of address. Store registers, interpret Load Op and W-modifier. Branch to JODE.
4. JODE RDCRWR. This routine performs tape operation decode. Set the write command and get the control byte from auxiliary storage. Test for density and redundancy, TIE, etc. Go to channel routine JCHL.
5. JCHL CHNL. Tape channel routine. This routine performs initial channel selection; set Mode and Zone, block traps, set Address Out and Select-Out. Get Op-In and reset Address-Out. Allow R/P traps. Wait for Address-In and compare addresses. Addresses OK, set Command Out. Wait for Status-In. Test Bus-In for busy bit. Not busy, check if sense op.
6. Test status, check Channel-End and issue Service-Out. Wait for fall of Status-In. Test for Sense Op, set Channel Mode and Zone. Get data address. Allow level 1 traps and set burst mode latch. Wait for trap to JDTA.
7. JDTA 0170. Tape read and write data loop. This routine gets data from storage one byte at a time and puts it on bus-out to tape. Bring up Service-Out and Select-Out. Allow Service-In trap; return to JCHL. This two routine loop is repeated for each byte of data until a GMWM is encountered.
8. A GMWM detected in the JDTA routine causes reset of Select-Out and the burst mode latch. Burst mode off detected in JCHL causes a branch to the DTAEND entry of JDTA.
9. Data end performs write end routine, resets mask, restores regs, etc. as required to end the operation. The routine branches to I-cycles for the next instruction.

Read tape operations follow the same general pattern. The major differences are in the ending routine. When a GMWM is detected in storage before an inter record gap on tape, the 0170 data loop continues until status-in but no data is stored.

When an interrecord gap is detected (Status-in/Device-end) the routine traps to address 0180, routine JEND. This routine checks status, next data character for GMWM etc., then returns to the JCHL routine. The JCHL routine resets Select-Out, restores regs, etc. then branches back to the data loop routine. The CPU zone is set, MMSK conditions are reset, registers are restored and the routine returns to ICYC for the next instruction.

STORAGE PROTECTION

- The storage protection feature allows alteration within 2,048-byte blocks of main storage only when:
  1. The four-bit storage key for the block being addressed matches:
     a. the protection key of the current PSW (for CPU operations), or
     b. the protection key assigned to the associated I/O operation (for channel operations).
  2. The PSW or I/O protection key is zero.

- The feature provides protection only against erroneous alteration of specified problem program areas of core storage and not fetch protection (against undesired accesses to core storage).

- The 'set storage key' instruction is used to assign a key to the block of storage specified in the instruction.

- The 'insert storage key' instruction is used to inspect the storage key.

- A protection key mismatch during CPU operations causes the instruction to be suppressed or terminated.

- A violation due to an I/O operation causes the I/O function to be terminated and an indication to be stored in the CSW at the end of the operation.

Storage protection is a feature available for System/360 Model 25. The storage protection feature protects a block of main storage assigned to one program from being changed.

Two machine language instructions are provided the problem-programmer for use with the storage protection feature:

Name: Insert Storage Key
Mnemonic: ISK
Type: RR
Operand: R1, R2
Code: 09

Name: Set Storage Key
Mnemonic: SSK
Type: RR
Operand: R1, R2
Code: 08

Four microword instructions are provided the engineer for use with the storage protection feature:

SSK STP1 *HWLS±1 Set Storage Key
SSK STP1 AS,*HWLS±1 Set Storage Key
ISK STP0 *HWLS±2 Insert Storage Key
ISK STP0 AS,*HWLS±2 Insert Storage Key
(*HWLS=halfword local storage register)

There are two keys used with the storage protection feature: the storage key and the protection key. The storage key is the key assigned to each 2,048-byte block of storage and is stored in a special monolithic local storage unit. This special monolithic SLT card is the same card as used for the CPU local storage. The protection key is the key in the PSW (program status word) or CAW (channel address word) and is compared to the storage key to determine if the area is protected. Any time main storage is read out, the protection key in the PSW (bits 8-11) is compared to the storage key assigned that block of main storage. When the read is during an I/O operation, the storage key is compared to the protection key of the CAW, bits 0-3. The keys match (are equal) when both the storage key and protection key are the same, or if the protection key in the PSW or CAW has a 4-bit code of 0. Only if the information in the block is to be changed (any store operation) is the result of the compare used. If a mismatch (unequal compare) occurs at this time, the information read out of storage is regenerated at write time, and an interrupt occurs so the supervisor program can be alerted to the mismatch condition.

The storage key is not part of addressable main storage. A monolithic local-storage unit (STP1) with a capacity for 128 hexadecimal digits with parity, and controls, are added to the 2025 when storage protection is installed. STP1 is used to store the storage keys and is loaded from the CPU local storage, via 'external bus out' (Figure 4-13), nine bits at a time (including parity). Parity is odd.

The CSTS routine is an example of how STP1 is loaded with a storage key from a local storage register using two microword statements. The key is first transferred from a local storage register into STP0 (STP0=D1). STP0 is a mnemonic that specifies the Q-register in CPU mode and is used as a buffer between CPU local storage and STP1. The key is then stored into STP1 at an address designated by the contents of a local storage register specified in the statement: SSK STP1 U+1 (set storage key microstatement).

The STP1 matrix is eight bytes wide (X-axis), by eight bytes long (Y-axis), by nine bits deep (Z-axis). The storage keys are represented in STP1 bits 0-3 for storage-block protection, bits 0-3 for channel operations, and bits 4-7 for communications operations. STP1 addressing structure and storage allocations are discussed in section Storage Key.

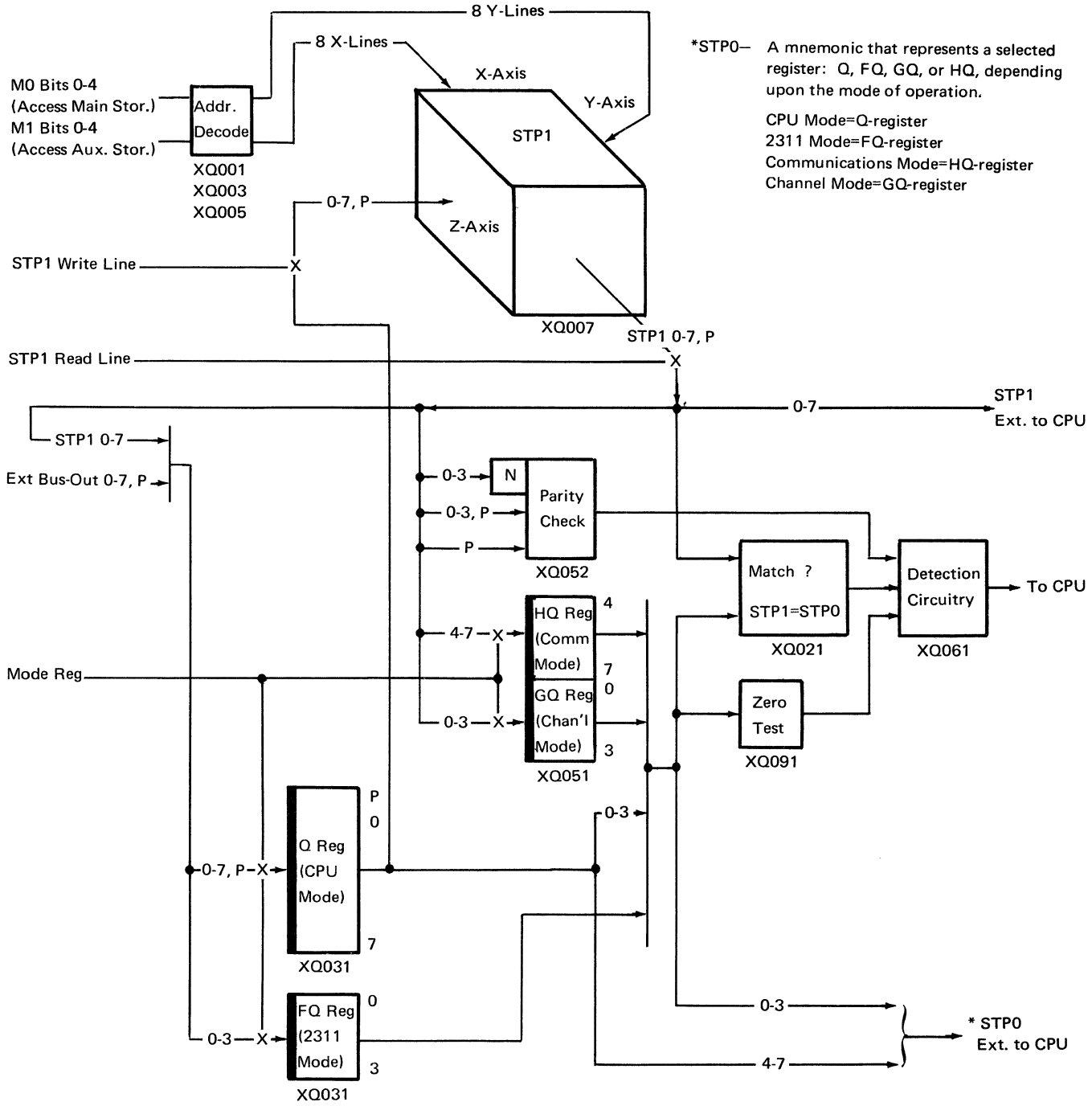


Figure 4-13. Storage Protection Data Flow

Twenty-four positions of STP1 are reserved for storing the storage key associated with each 2,048-byte block of main storage (0-48K). Thirty-two positions are reserved for storing the protection keys associated with channel operations keys (each possible UCW). This key is obtained from the CAW during the I/O start routine. Thirty-two positions are reserved for storing the protection key associated with communications keys.

The storage keys associated with each 2,048-byte block of main storage, the channel operations keys, and the communications keys are assigned by issuing the 'set storage key' instruction. The 'insert storage key' instruction is used to inspect the storage key.

STORAGE KEY

- A storage key is a 4-bit number assigned to a 2,048-block of storage.
- There are 16 different keys, 0-F.

To implement the storage protection feature, main storage is divided into blocks of 2,048 bytes. A processor with a main storage of 8,192 bytes has 4 blocks, and a processor with 49,152 bytes has 24 blocks. Each storage block of 2,048 bytes has a key associated with it. This key is four bits long and may contain any number from 0 through F. These numbers are referred to as storage keys. They can be assigned in any order and any of the possible 16 keys can be used regardless of storage size (Figure 4-14). Blocks of storage with the same key do not need to be consecutive blocks.

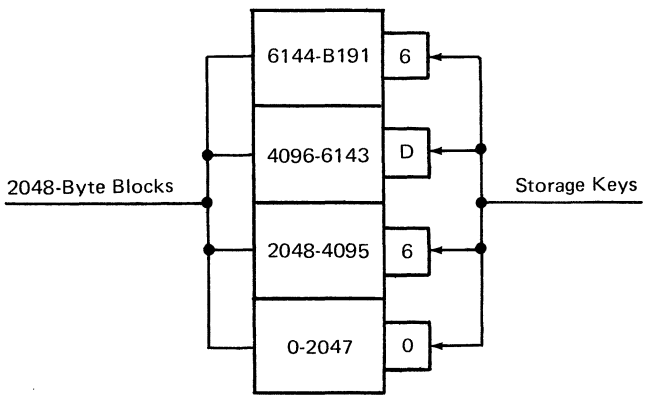


Figure 4-14. Storage Keys

The storage keys are stored in STP1 by the 'set storage key' instruction (see Set Storage Key). There are 24 positions reserved in STP1 (Figure 4-15) for program storage keys. Bits 4-7 of the STP1 locations addressed by X-lines 0-3, Y-lines 6-7 are not used, and are set to zeros. Unused channel and communication key locations are also set to zeros.

Figure 4-16 gives the program storage key allocations for STP1, and Figure 4-17 gives the channel and communications key allocations for STP1. (For systems with less than 48K bytes of program storage, core storage locations above the installed program storage are not protected.) Main storage block 0 to 2047 (0000-07FF) key is at location X0-Y0, block 2048 to 4095 (0800-0FFF) is addressed by X1-Y0 lines, etc.

For channel operations, bits 0-3 (Z-axis shown in Figure 4-15) are used for a maximum of 32 keys (defined within parentheses in Figure 4-17). For communications operations, bits 4-7 (Z-axis shown in Figure 4-15) are used for a maximum of 32 keys (defined within parentheses in Figure 4-17).

Up to eight 2K blocks of main storage (for 16K systems) can be protected by setting the proper key in the appropriate STP1 location. For 32K systems, as many as sixteen 2K blocks can be protected (24 blocks for 48K systems). Similarily, protection keys for up to 64 overlapped I/O operations can be set: 32 are assigned to communications subchannels and 32 are assigned to a System/360 multiplex channel.

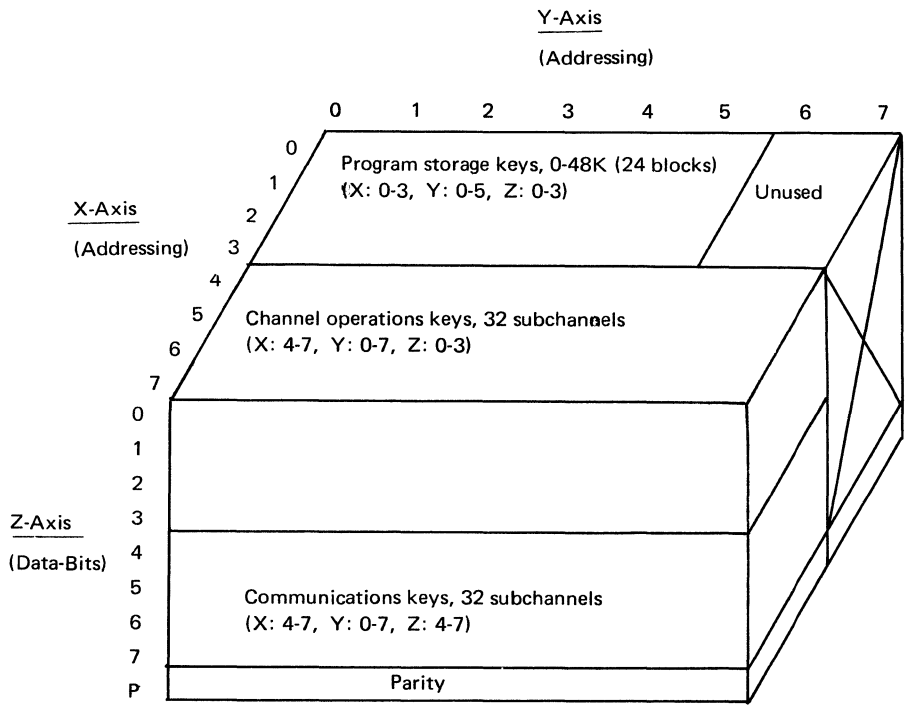


Figure 4-15. STP1 Allocations

Y-AXIS

X-AXIS	0	1	2	3	4	5	6*	7*
0	0000-07FF 0-2K	2000-27FF 8-10K	4000-47FF 16-18K	6000-67FF 24-26K	8000-87FF 32-34K	A000-A7FF 40-42K	C000-C7FF 48-50K	E000-E7FF 56-58K
1	0800-0FFF 2-4K	2800-2FFF 10-12K	4800-4FFF 18-20K	6800-6FFF 26-28K	8800-8FFF 34-36K	A800-AFFF 42-44K	C800-CFFF 50-52K	E800-EFFF 58-60K
2	1000-17FF 4-6K	3000-37FF 12-14K	5000-57FF 20-22K	7000-77FF 28-30K	9000-97FF 36-38K	B000-B7FF 44-46K	D000-D7FF 52-54K	F000-E7FF 60-62K
3	1800-1FFF 6-8K	3800-3FFF 14-16K	5800-5FFF 22-24K	7800-7FFF 30-32K	9800-9FFF 38-40K	B800-BFFF 46-48K	D800-DFFF 54-56K	F800-FFFF 62-64K

*These columns are shown for reference only. The storage locations represented are never used for program storage, and thus are not subject to the storage protection feature.

Figure 4-16. STP1 Allocations for Program Storage

X-Axis \ Y-Axis	0	1	2	3	4	5	6	7
	00-7 (1)	20-27 (5)	40-47 (9)	60-67 (13)	80-87 (17)	A0-A7 (21)	C0-C7 (25)	E0-E7 (29)
	08-0F (2)	28-2F (6)	48-4F (10)	68-6F (14)	88-8F (18)	A8-AF (22)	C8-CF (26)	E8-EF (30)
	10-17 (3)	30-37 (7)	50-57 (11)	70-77 (15)	90-97 (19)	B0-B7 (23)	D0-D7 (27)	F0-F7 (31)
	18-1F (4)	38-3F (8)	58-5F (12)	78-7F (16)	98-9F (20)	B8-BF (24)	D8-DF (28)	F8-FF (32)

Figure 4-17. STP1 Allocations for Channel and Communications Operations

PROTECTION KEY

- The protection key is a 4-bit number found in a PSW (for CPU operations) or a CAW (for channel operations).
- The protection key in the PSW or CAW is compared to the storage key assigned to a location in STP1.
- The result of the compare is used only for the storage modification cycle.

The protection key is in bit positions 8-11 of the PSW and bit positions 0-3 of the CAW. The PSW or CAW protection key is compared to the storage key in STP1 each time main storage is accessed.

When a main storage access is made for storing new data, STP1 is also accessed (addressed) using the contents of M-register (see Figure 4-13). The high-order digit of the information read out of STP1 contains the key associated with the block of main storage being accessed. This digit is matched with the contents of either the Q, FQ, GQ, or HQ-register, depending upon the mode (see Mode Register, Section 2) of operation. The result of the match is used only when the information in storage is to be modified. A read-write (regenerate) cycle, even if the keys are mismatched, is performed without an interrupt because storage is not modified.

When an auxiliary storage access is made to read out control information that governs a data transfer on either the I/O channel or the communications adapter, the GQ- or HQ-registers are loaded with a particular low-order digit from STP1 in parallel with the auxiliary storage read out. Because an auxiliary storage address, unique to a particular I/O channel or communications subchannel is used to read out the control information, this address is also used to specify a unique STP1 location. Each time auxiliary storage is accessed for this purpose, a simultaneous access of STP1 is made using the M-register contents. The low-order digit may be set into either the GQ-register or the HQ-register several times before the actual main storage data transfer occurs. However, when the data transfer does occur, the correct protection key will be resident in either the GQ- or the HQ-register. As a result of this procedure, adding the storage protection feature to the 2025 causes no functional change to the standard I/O-to-main-storage data transfer.

In effect, the protection key stored in STP1 extends the UCW by four bits. This allows a record of the protection key for each I/O unit on the multiplexer channel to be maintained.

If the storage protection feature is not installed, the protection key must be zero.

PROTECTION EXCEPTION

- When storage protection is violated, the protection exception is indicated in the PSW or CSW.
- Detection circuitry signals the system that a protection key mismatch has been detected, or that a parity error was found in the storage block key. In either case, the storage-protect check bit is set in the machine-check register (MC1).

If a mismatch of the two keys occurs during a store main-storage access (and the protection key is non-zero), the original contents of main storage are regenerated.

In addition, a microprogram trap is initiated to inform the problem program of the protection violation, providing the error did not occur during a 2311 cycle steal. The protection check bit is stored in the CSW. If the error did occur during a 2311 cycle steal, a 'file protection check latch' is set. This terminates the current 2311 operation, and eventually stores a 'protection check' in the CSW for the current file operations.

SETTING UP STORAGE PROTECTION

- The supervisor program assigns the storage keys to each block of storage.
- After the problem programs are loaded and the protection keys set, the supervisor program transfers control to a problem program.

The 'set storage key' (SSK) is a privileged (machine language) instruction. It may be issued only when bit 15 (problem state bit) of the PSW is zero. In a typical supervisor controlled operation, the supervisor causes a problem program to read into main storage. The supervisor sets the storage keys for the area of storage used by the problem program. The PSW used by the problem program is assembled by the supervisor program. The supervisor program, as well as the assembled PSW (Figure 4-18), reside in the program area of main storage in the 2025. This assembled PSW has a protection key that matches the storage key associated with the problem program.

Once the function of loading a problem program into main storage and assigning the keys for storage protection is done, the supervisor passes control to the problem program. This is done with the 'load PSW' instruction, which specifies the assembled PSW (Figure 4-19). The 'load PSW' instruction causes the protection key to be stored in a local storage register, and transferred (moved) to the Q-register (BPSW microroutine). The move from local storage to the Q-register is accomplished by the move/arithmetic word type 3, which specifies a local storage to external data transfer; i.e., STP0=X (where X represents the local storage register), and an AS-field combination selecting the Q-register.

The protection key in the PSW used by the supervisor program is generally zero. This allows the supervisor program to modify data anywhere in main storage. In Figure 4-19, the main storage area occupied by the supervisor program has a storage key of F. This means that unless a problem program has a key in its PSW of 0 or F, it cannot modify information in the area used by the supervisor program. This is unlikely because the supervisor program assigns the storage and protection keys.

The same storage key number can be set for more than one block of 2,048 bytes. However, each program in main storage should have a different storage key assigned, to protect one program from another. For instance, the supervisor program may take one block of 2,048 bytes, assigned a storage key of F. This storage key most likely was assigned by the supervisor program just after it was read into the system. The problem program is then read into the processor (as a result of a section of the supervisor program). The problem program in Figure 4-19 takes up three blocks of 2,048 bytes each. Each of the three blocks is assigned the same storage key (1, for example) by the supervisor program. The PSW for the problem program is given a protection key that matches its storage key. This allows the problem program to alter itself if necessary, but prevents it from altering another program.

It is possible to have two or more problem programs in main storage at once. Of course, just as in the supervisor controlled concept, only one program is being executed at any one time. Figure 4-20, shows that each problem program has a different storage key. The protection key for each program is also different; each matches the respective storage key.

Notice in Figure 4-20, the protection key of the supervisor program does not match its storage key. Because the protection key is zero, it does not have to match. A protection key of zero can unlock any area of main storage and alter its contents if necessary.

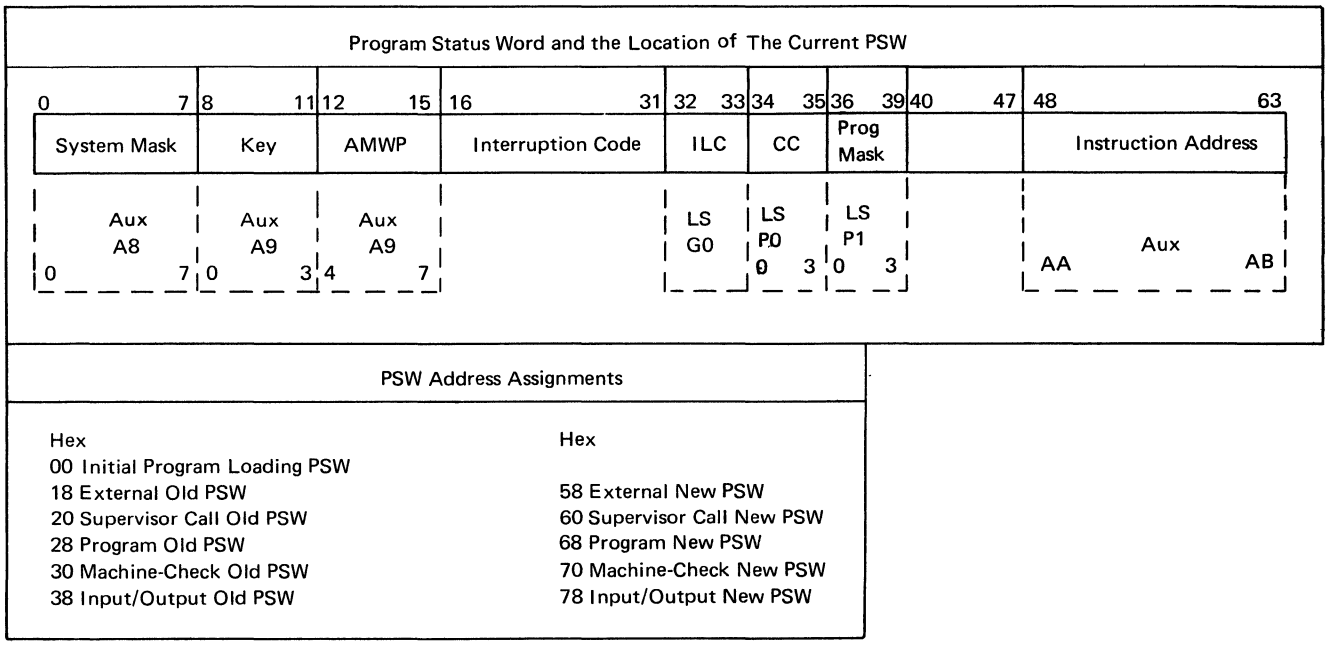


Figure 4-18. PSW

Assume: 1. That the problem program takes 5,000 bytes and begins at location 2048.
2. That the supervisor is in locations 000--2047 and has a storage key of F and a protection key of 0.

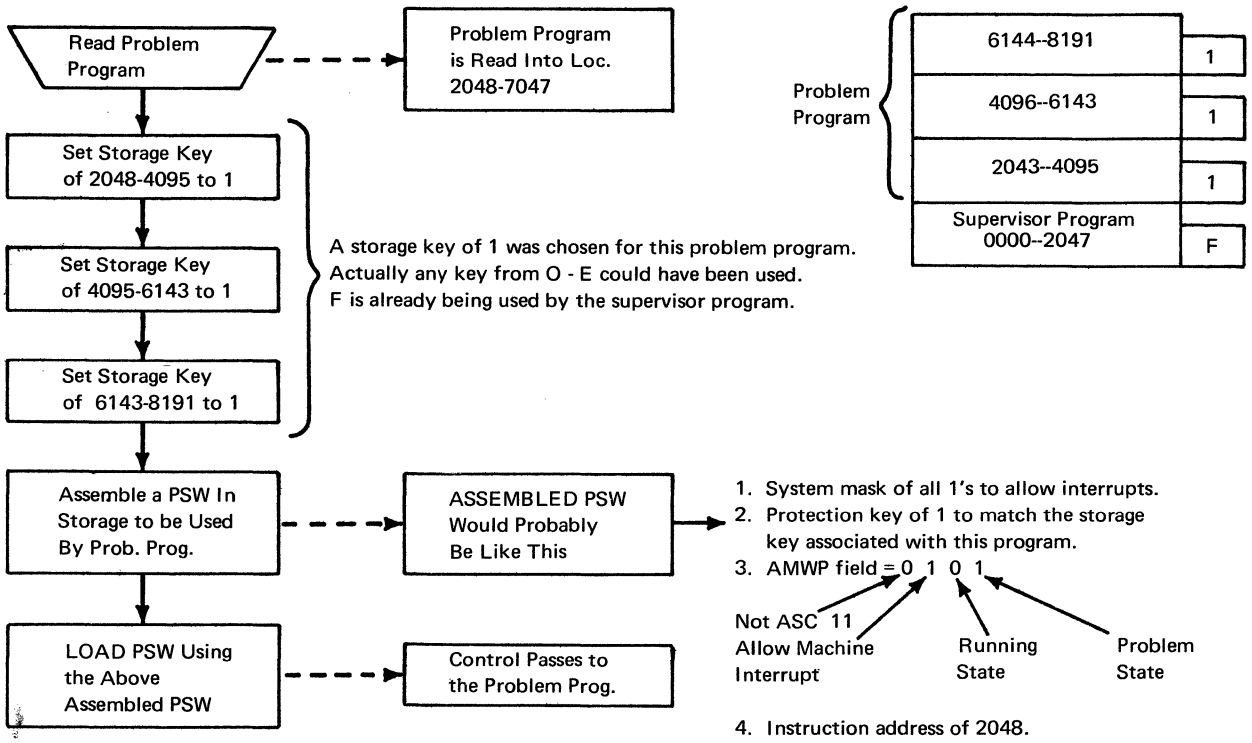


Figure 4-19. Using Storage Protection

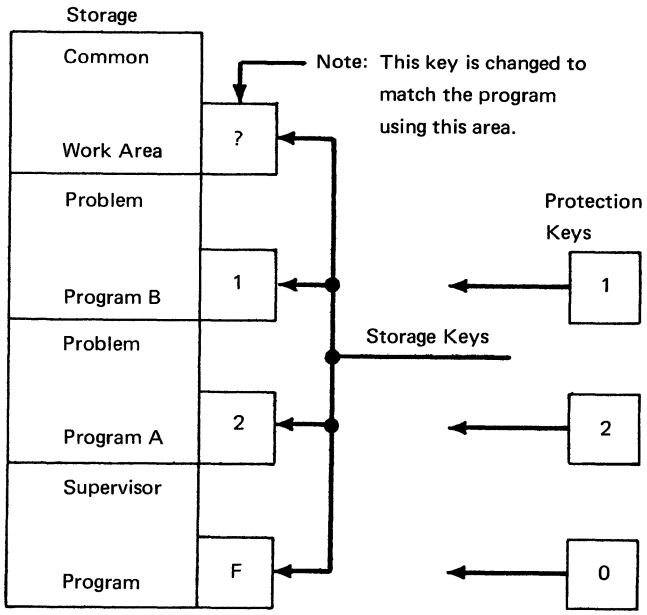


Figure 4-20. Storage and Protection Keys

Set Storage Key (Machine Language Instruction)

- SET STORAGE KEY (SSK) is a privileged instruction, used by the problem programmer.
- The instruction is of the RR format.
- After the instruction is decoded in I-cycles, the CSTS routine loads the storage key into the storage protection stack (STP1).

The key of the storage block addressed by the register designated by R2 is set according to the key in the register designated by R1.

The storage block of 2,048 bytes is addressed by bits 8-20 of the register designated by the R2 field. Bits 0-7 and 21-27 are ignored. Bits 28-31 must be zero. Otherwise, a specification exception causes a program interrupt.

The 4-bit storage key is obtained from bits 24-27 of the register designated by the R1 field. Bits 0-23 and 28-31 are ignored.

Insert Storage Key (Machine Language Instruction)

- INSERT STORAGE KEY (ISK) is a privileged instruction used by the problem programmer.
- The instruction is of the RR format.
- After the instruction is decoded in I-cycles, the CSTS routine reads out of STP1 the key assigned to a given block.

The key of the storage block addressed by the register designated by R2 is inserted in the register designated by R1.

A storage block is addressed by bits 8-20 of the register designated by the R2 field. Bits 0-7 and 21-27 are ignored. Bits 28-31 must be zero. Otherwise, a specification exception causes a program interruption. The 4-bit storage key is inserted in bits 24-27 of the register specified by the R1 field. Bits 0-23 of this register remain unchanged, and bits 28-31 are set to zero.

DISPLAYING A STORAGE PROTECT KEY

To display a storage key, the M-register must first be set up to the 2,048 bytes associated with the CPU storage key or the MPX UCW address associated with a particular I/O device. Therefore, a manual display of auxiliary storage must be done to see the key used by a start I/O, or of main storage to see a CPU key. After the manual display is done, turn the mode switch to ALU/EXT, dial switches CD to 03 (CPU mode external decode 3), and press the display switch to gate the storage key into byte-1.

The selector channel key can be displayed by using auxiliary storage address XX88 and following the display procedure described above for a storage key. This is possible since MPX UCW's cannot be used when channel burst mode is defined.

There is one exception to this display procedure. The M-register does not have to be set up to display the effective file storage key. This is possible because the file key is stored in hardware (not STP1). To display the effective file storage key, dial 12 in switch CD and dial ALU/EXT.

FUNCTIONAL UNITS (REFER TO FIGURE 4-13)

- STP1: A monolithic local-storage unit with a capacity for 128 hexadecimal digits with parity.

- STP0: An external source/destination that is either Q (CPU mode), FQ (2311 mode), GQ (channel mode), or HQ (communications mode), depending upon the contents of the mode register.

- Q: An 8-bit register containing a priority-level 0-key which is loaded from the main data flow via the external bus-out. The Q-register is also used as a buffer between STP1 and local storage.

- FQ: A 4-bit register containing a cycle-steal key used for the 2311 attachment in the Model 25.

- GQ: A 4-bit register containing the I/O channel key at the priority level assigned to the channel.

- HQ: A 4-bit register containing the communications channel key at the priority assigned to this channel.

- Storage Protect Match: Match circuitry that compares the output of STP1 with STP0 (a selected register: Q, FQ, GQ, or HQ). The selection depends upon CPU status and the mode register; that is, the I/O operation in progress.

- Zero Test: Zero test circuitry for registers Q, FQ, GQ, or HQ to determine if the protection key is zero during a store main-storage access.

Detection circuitry is also provided to signal the system that a protection key mismatch has been detected, or that a parity error was found in the storage-block key (see Figure 4-13). In either case, the storage protect check bit is set in the machine check register (MC1).

STP1 MAP

The monolithic local-storage unit (see Figure 4-15) used to store the storage keys has a capacity for 128 hexadecimal digits with a parity bit associated with each byte. The STP1 matrix is eight bytes wide (X-axis), by eight bytes long (Y-axis), by nine bits deep (Z-axis). The program storage key allocations are shown in Figure 4-16, and the channel and communications keys are shown in Figure 4-17.

THEORY OF OPERATION (MDM 4-90)

STP1 ADDRESSING STRUCTURE (FIGURE 4-21)

The particular STP1 location to be loaded is selected by bits from the BSM addressing register (M-register). These bits come from two different sets of address register bits, depending on whether the main-storage block key or a subchannel key is desired.

During any storage word that accesses main storage, STP1 is addressed by M0 bits 0, 1, 2, 3, and 4. (The highest order bit is not presently used because the addressable core storage is 48K.) During any storage word that accesses auxiliary storage, STP1 is addressed by M1 bits 0, 1, 2, 3, and 4, and with the communications or channel mode. No readout occurs with an auxiliary storage access if the mode register does not specify channel or communications mode.

LOADING STP1

STP1 is loaded from local storage nine bits at a time (including parity) through 'external bus out'. These storage keys are represented in STP1 bits 0-3 for program storage protection, bits 0-3 for channel operations, and bits 4-7 for communications operations.

A protection key for a particular 2K block of program storage is transferred from local storage to STP1 by the 'set storage key' instruction.

The storage key assigned to each 2,048-byte block of main storage is usually set into STP1 during the first part of the supervisor program.

The desired storage key is in bits 24-27 of the general register specified by the R1 field of the instruction (Figure 4-22). The block of storage (to which the key is assigned) is determined by the address in general register specified by the R2 field.

The microprogram sequence (Figure 4-23) performs the following steps using the Q-register as a buffer (CSTS Routine).

1. The 4-bit key is transferred to the Q-register using the mnemonic STP0 (Q-register in CPU mode).
2. The key is then taken from Q to STP1 at an address determined by the bits from the M-register (see Figure 4-21).

3a and 3b. Q is restored with the CPU key from auxiliary storage (CCOM routine).

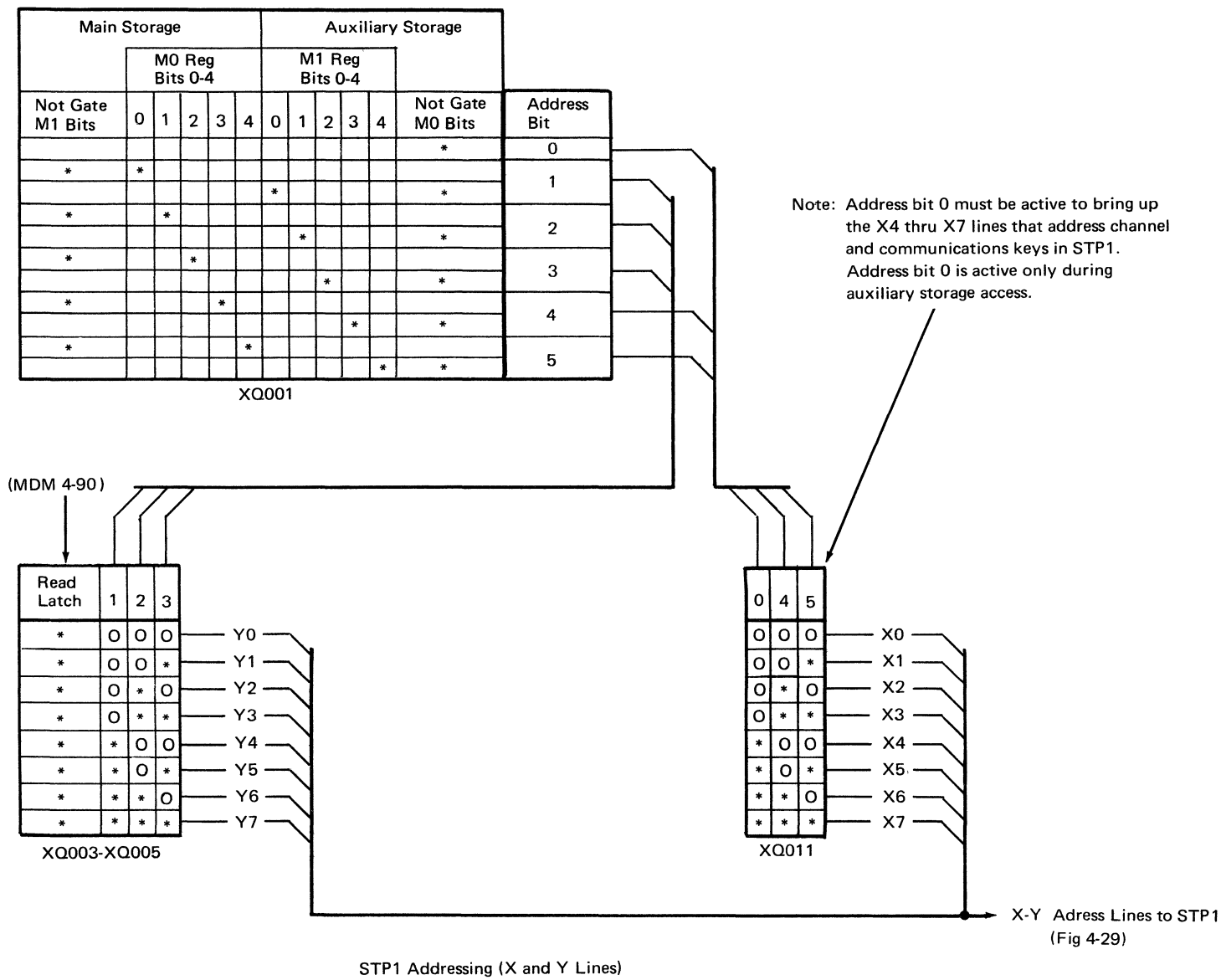


Figure 4-21. STP1 Addressing Structure

STP1 Addressing (X and Y Lines)

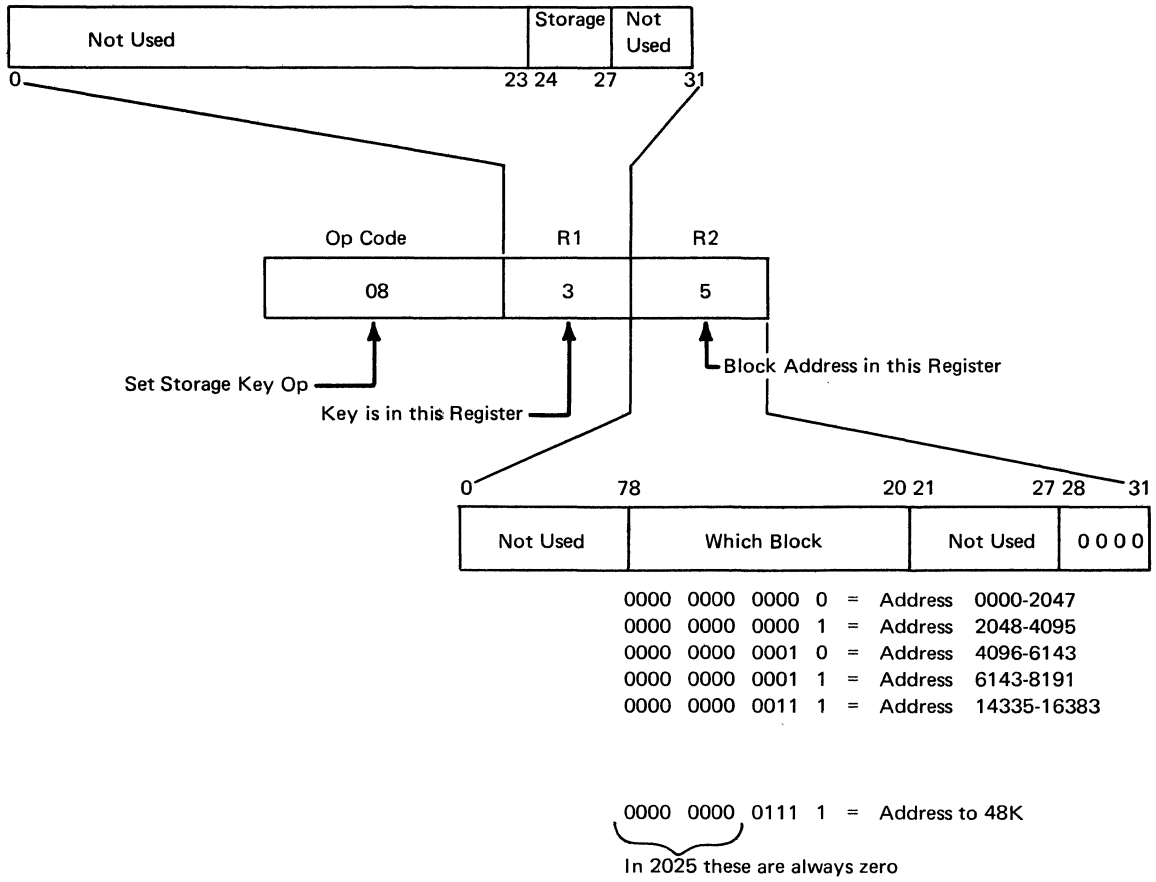


Figure 4-22. Set Storage Key Instruction Word Format

Assume:
1. CPU Mode
2. Key is in high-order digit of local storage byte D1.
3. Block address is in U.

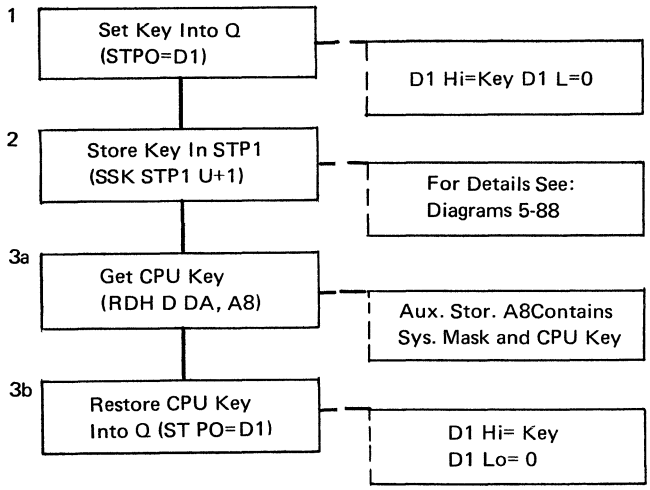


Figure 4-23. Set Key (for 2K Block of Main Storage) CSTS Routine

Set Storage Key Microword Cycle (Figure 4-24)

The SSK microword is a storage-word type-2 (external data register access and update type). STP1 is always specified as the data register, a halfword local storage register is specified as the address register, and the address is modified by plus or minus one.

The SSK microword statement can be one of two types:
1. SSK STP1 *HWLS±1 (*HWLS = halfword local storage register)
2. SSK STP1 AS,*HWLS±1 (*HWLS = halfword local storage register)

MDM 5-88 is an operational diagram showing an example of the SSK STP1 HWLS±1 microword statement. The indirect address is read from the halfword local storage register (HWLS) specified in the statement and placed in M-register. The data in the M-register is then used to address STP1; the storage key previously placed in the Q-register is gated into STP1 at the address specified in the M-register.

The SSK STP1 AS,HWLS±1 statement is identical in function except that the halfword local storage register addresses auxiliary storage instead of main storage.

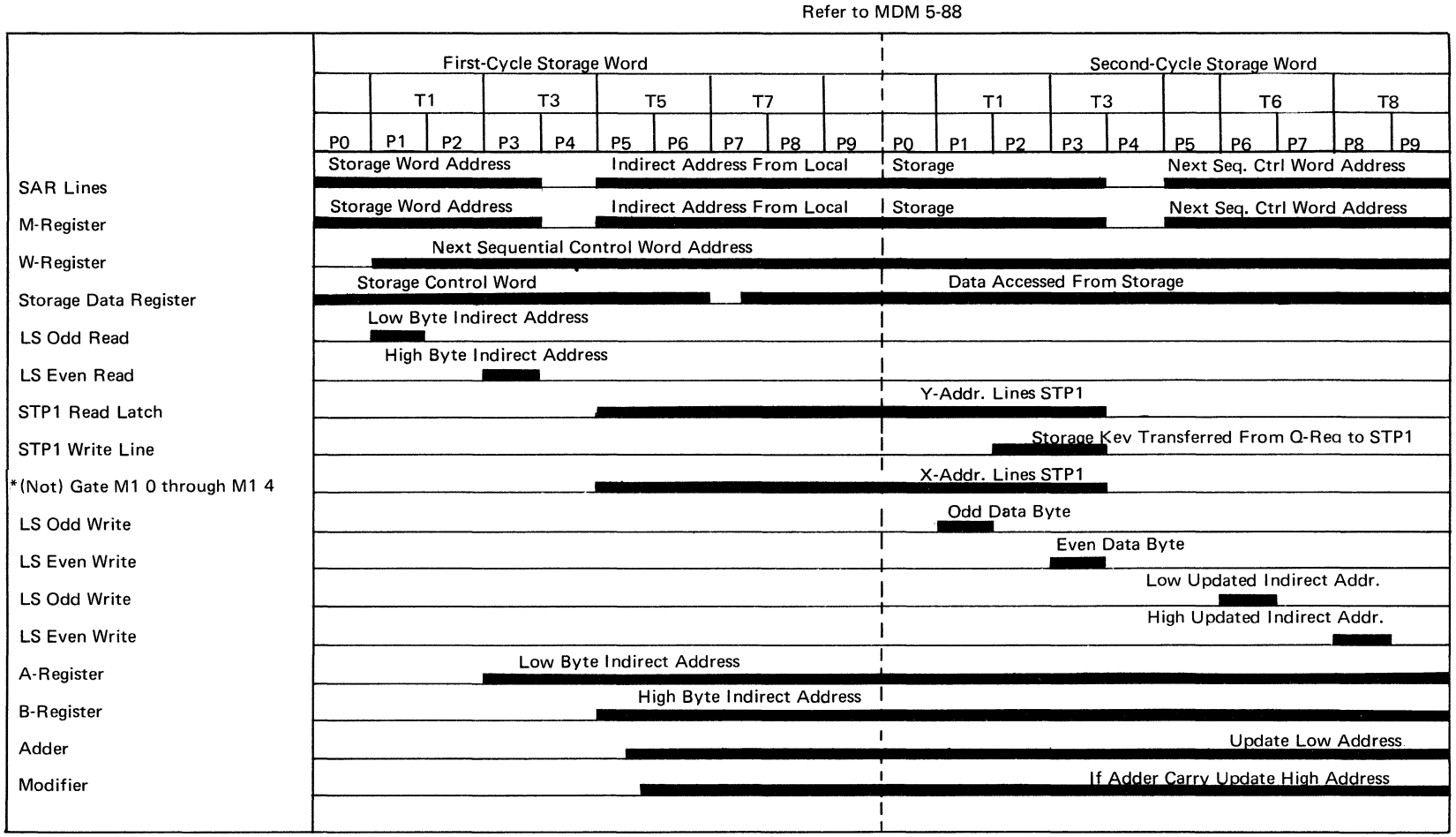
The 'STP1 read latch' (MDM 4-90, P1 5B) is turned on at T5 time during the 'first cycle storage word'. 'Read latch' (line) gates the decoded 'Y-address lines (to) STP1'. 'STP1 write line' is developed by 'AS-field decode 3', (not) '2311 mode or file cycle steal', and 'gate CPU to external pulse'. 'Gate CPU to external pulse' is brought up at T1 time during the second cycle storage word. The 'X-address lines (to) STP1' are gated by the (not) 'gate M1-0 through M1-4' line. The 'gate M1-0 through M1-4' line is active only during a 'use auxiliary store' storage word cycle. When the 'gate M1-0 through M1-4' line is active (auxiliary store access), M1 bits are gated to address STP1; when the line is inactive (main store access), M0 bits are gated to address STP1.

Main storage is also addressed by the M-register and data is accessed from storage, but the 'local storage data-assembler' is not gated, thereby preventing main storage data from entering local storage or being gated on 'external bus out'.

Figure 4-25 shows an expanded routine used to store a channel and/or a communications key into STP1. When only one key at an STP1 location is to be loaded or changed, the byte in STP1 is read out, the required digit is mixed with the digit to be saved, and the result is stored back into STP1. These steps are repeated (using predefined bit patterns in the local storage register) until the necessary keys are transferred to STP1.

Bits 4-7 must be zeros for bytes representing storage blocks (see Figure 4-15) because only bits 0-3 represent program storage keys in STP1. A channel key and a communications key can be stored by a single transfer from local storage because of their vertical alignment in STP1.

Figure 4-24. SSK STP1 U+1



*This line becomes active only for auxiliary storage access (MDM 4-90 P1 6A)

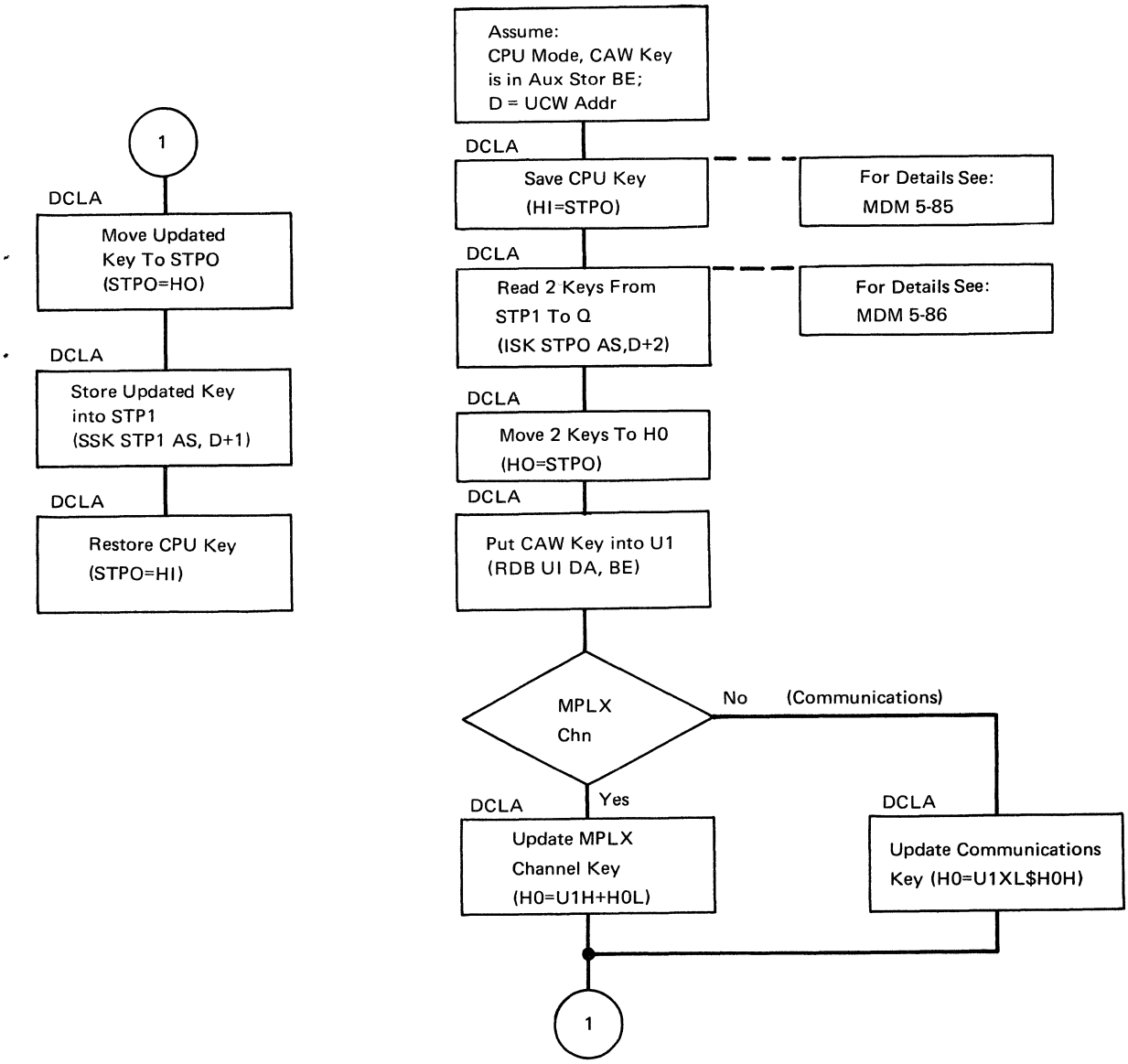


Figure 4-25. Storing Multiplex or Comm. Chan. Key from CAW into STP1 (DCLA Routine)

LOADING Q- AND FQ-REGISTERS

These registers are loaded by a move/arithmetic word, which specifies a 'local storage to external' transfer, and an AS-field combination selecting either Q or FQ (Figure 4-26). The bits are read out of local storage and sent (via the 'external bus out') to the selected register as shown in Figures 4-27 and 4-28.

The Q- and FQ-registers can be read into the main data flow by a move/arithmetic word that specifies an 'external to local storage' transfer and an AS-field combination selecting either Q or FQ. The same combination is actually used for both Q and FQ, but it is the mode register that selects which of these registers is transferred to or from the main data flow (defined in Main Storage Access).

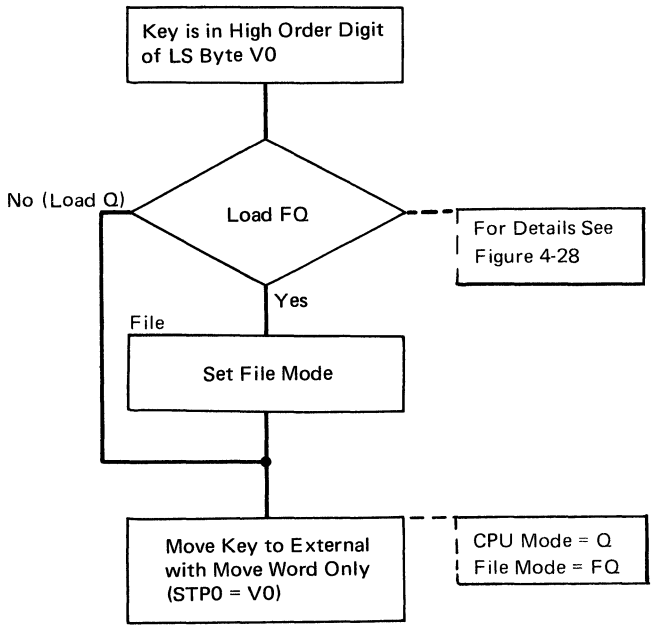


Figure 4-26. Q- or FQ-Register Loading of CAW key

LOADING GQ- AND HQ-REGISTERS (FIGURE 4-29)

These registers are not available to the main data flow and are loaded only from STP1 during all auxiliary storage access cycles. Either GQ or HQ is selected, depending upon the mode (channel or communications).

MAIN STORAGE ACCESS (FIGURE 4-29)

When a main storage access is made to store new data, STP1 is also accessed (addressed) using the contents of the M-register.

Note, in Diagram 4-90 Part 1, that the 'STP1 read line' and the 'read latch' lines are turned on when the 'STP1 read latch' is on. The 'STP1 read line' is delayed 30 nsec. to ensure the gating-time for the X- and Y-lines addressing STP1. The high-order digit of the information read out of STP1 contains the key associated with the block of main storage being accessed. This digit is matched with the contents of either Q-, FQ-, GQ-, or HQ-register, depending upon the mode of operation (Figure 4-30):

Q--if a CPU operation is in progress, or a 1052, 2540, or 1403 main storage data transfer is in progress

FQ--if a 2311 cycle steal is in progress
GQ--if the I/O channel is transferring data
HQ--if the communications channel is transferring data.

When an auxiliary storage address unique to a particular I/O channel or communications subchannel is used to read out the control information, this address is also used to specify a unique STP1 location. Each time auxiliary storage is accessed for this purpose, a simultaneous access of STP1 is made using the M-register contents. The GQ- or HQ-registers are loaded with a particular low-order digit from STP1 in parallel with the reading out from auxiliary storage of control information that governs a data transfer on either the I/O channel or the communications adapter. The low-order digit may be set into either GQ or HQ several times before the actual main storage data transfer occurs. However, when the data transfer does occur, the correct protection key is resident in either GQ or HQ.

ZERO TEST (PROTECTION KEY 0)

A zero test is made for registers Q, FQ, GQ, and HQ (Figure 4-30). The output of these registers (STP0 bits 0-3) is gated to an OR block. A protection key of 0 is indicated when this OR block is inactive (not conditioned).

PARITY CHECK (FIGURE 4-30)

A parity check is made of STP1 bits 0-3 and P. Parity is odd. A parity error found in the storage-block key sets the storage-protect bit in the machine-check register (MC1) in storage. The console storage protect light is only set by a parity check (not by a trap or mismatch).

MATCH CIRCUITRY (STP0=STP1)

Match circuitry (Figure 4-30) compares the output of STP1 with STP0 (a selected register Q, FQ, GQ, or HQ.) The selection of the particular register depends upon the CPU status and the mode register; that is, the I/O operation in progress.

The bits from the Q, FQ, GQ, and HQ are compared in an exclusive OR network, the output of which is ANDed together. An active output from any exclusive OR block deconditions the 'HDWE and QLS keys match' line signaling a mismatch of the storage key and protection key.

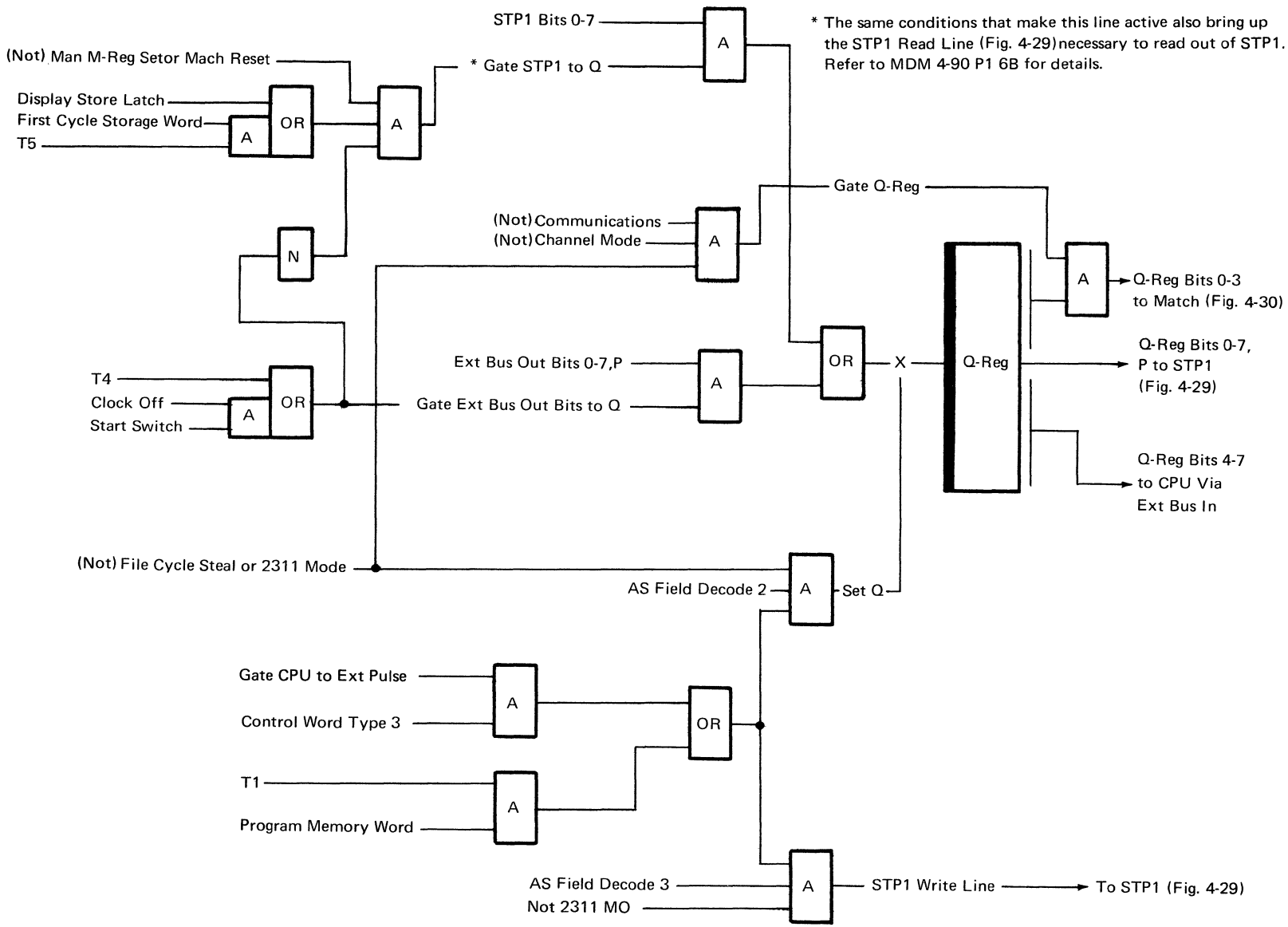
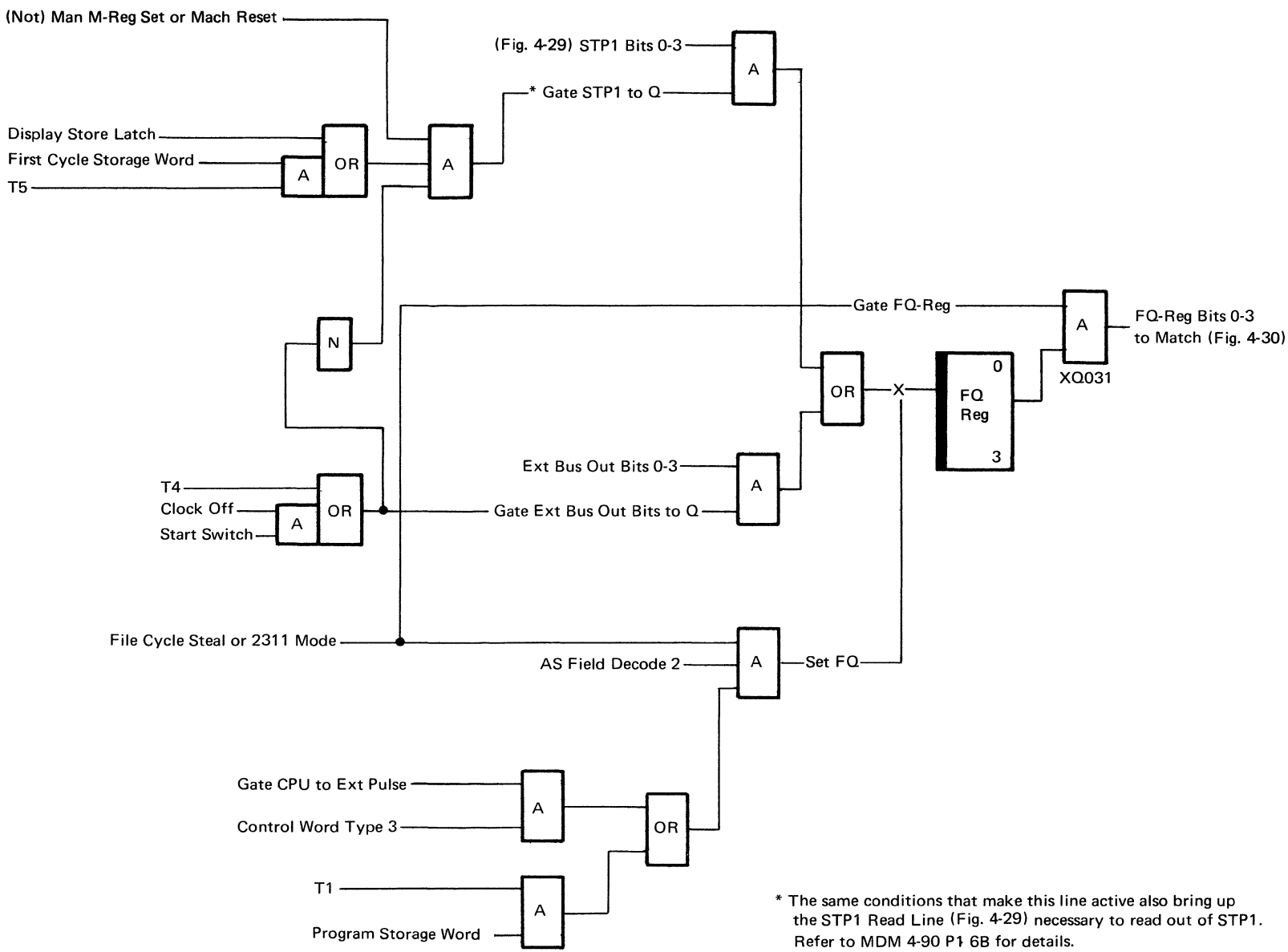


Figure 4-27. Q-Register Addressing

Figure 4-28. FQ-Register Addressing



* The same conditions that make this line active also bring up the STP1 Read Line (Fig. 4-29) necessary to read out of STP1. Refer to MDM 4-90 P1 6B for details.

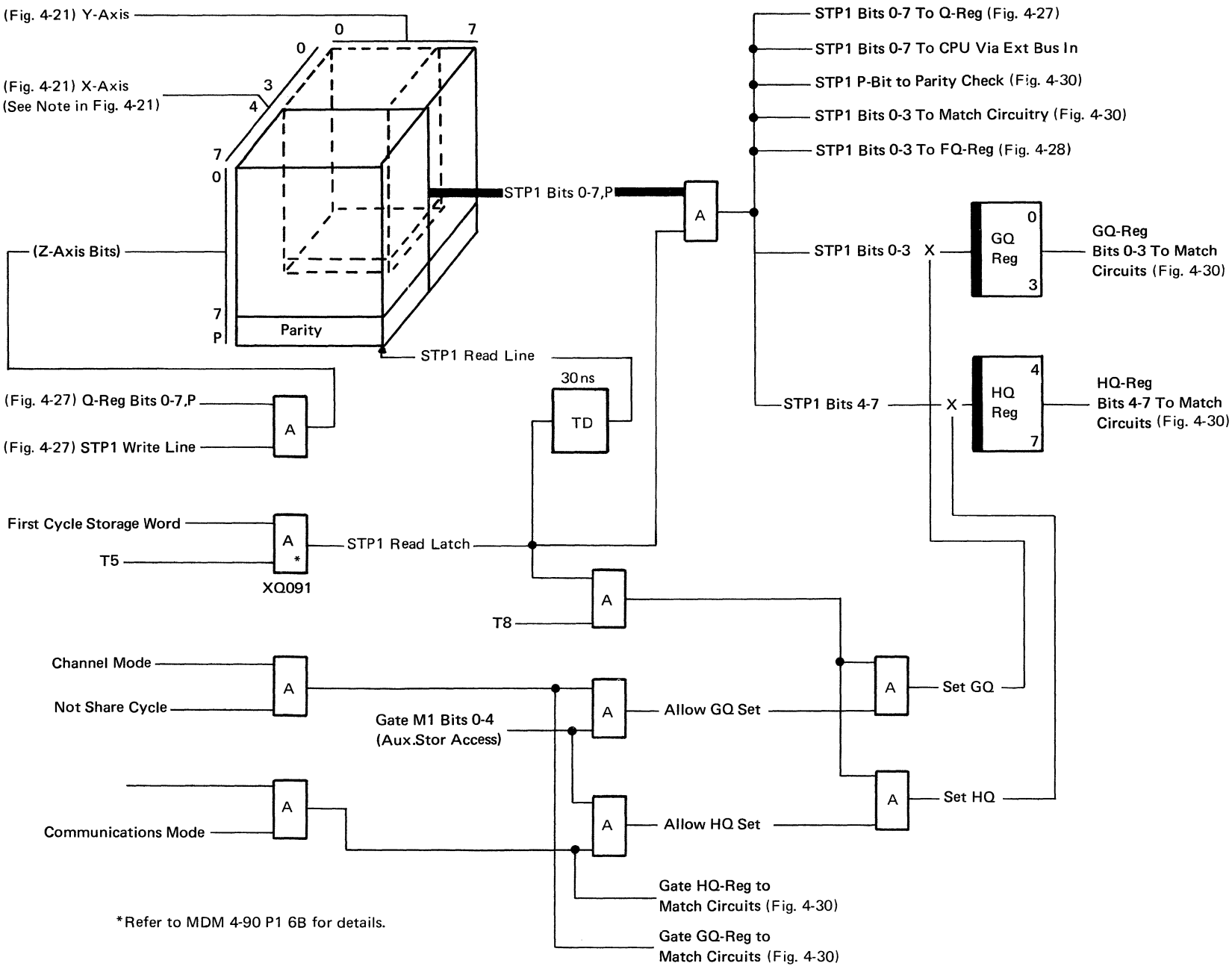


Figure 4-29. STP1, GQ-Register, and HQ-Register Addressing

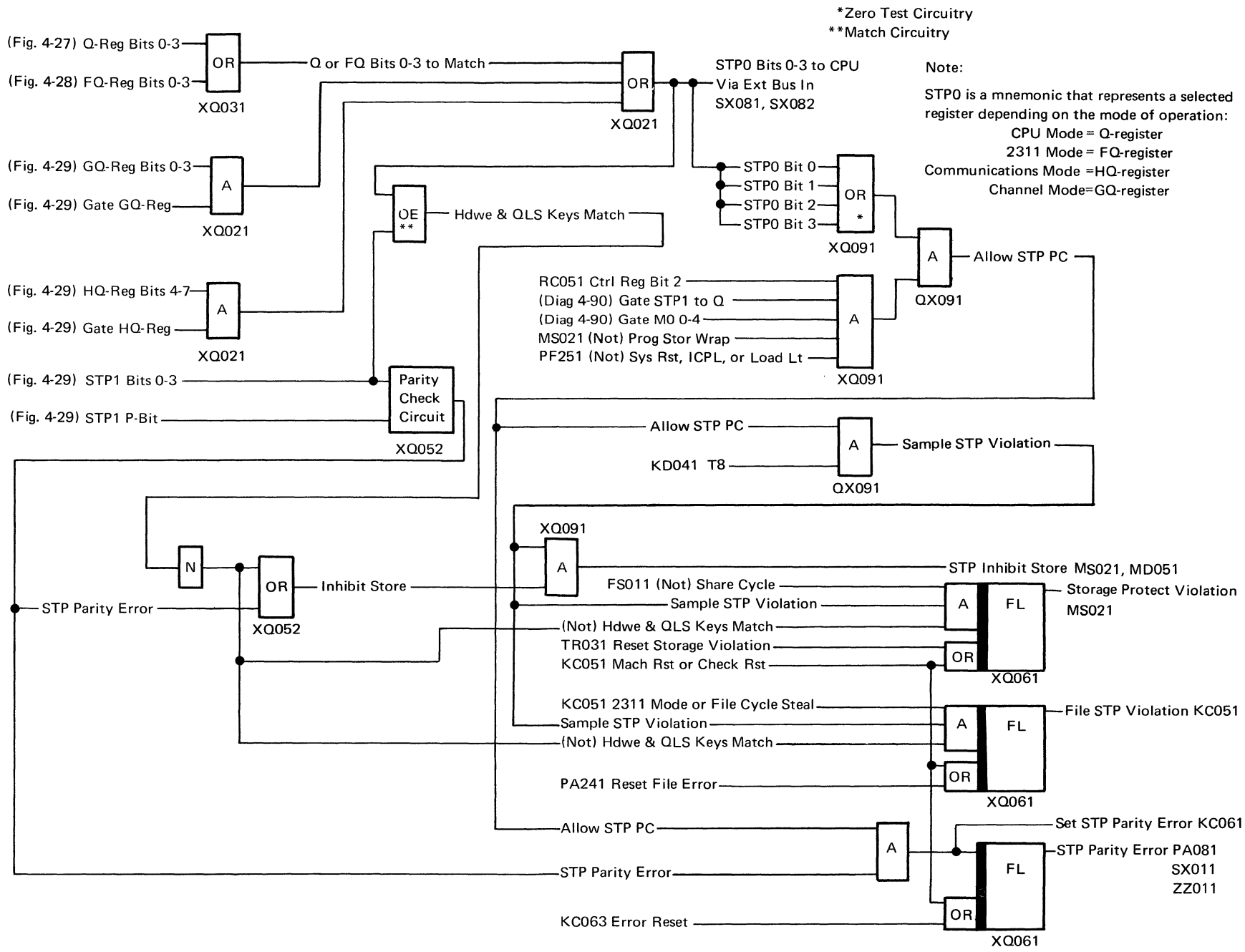


Figure 4-30. STP Controls, Match, and Error Detection

INSERT STORAGE KEY (CSTS ROUTINE)

If it is necessary to find out what number has been assigned to a block of storage, the privileged instruction, 'insert storage key', is used (Figure 4-31). This instruction takes the address in R2 and sets the address in U. The indirect address from U is placed in the M-register and is used to read out the storage key from STP1 into the Q-register. Main storage is also addressed by the M-register and data is read out, but the local storage data assembler is not gated, thus preventing local storage access.

The Q-register is then stored in a local storage register and set into the third byte of R1. A record has now been made of the storage key assigned to that block of storage.

The CPU key is fetched from auxiliary storage location A8 and stored in a local storage register. STP0 (Q-register in CPU mode) is addressed and the CPU key is moved from the local storage register to Q.

Figure 4-32 is an example of the routine necessary to fetch a block key for a particular 2K block of main storage. Figure 4-33 shows a corresponding routine (CSW store) to read a multiplex or communications channel key from STP1. Both operations are done in CPU mode.

Insert Storage Key (Microword Statement)

The ISK microword is a storage-word type-2 (external data register access and update type). STP0 is always specified as the data register, a halfword local storage register is specified as the address register, and the address is modified by plus or minus two.

The ISK microword statement can be one of two types:
1. ISK STP0 *HWLS±2
2. ISK STP0 AS, *HWLS±2
* HWLS=halfword local storage register
MDM 5-87 is an operational diagram showing an example of the ISK STP0 HWLS±2 microword statement. The indirect address is read from the halfword local storage register (HWLS) specified in the statement and placed in the M-register. The data in the M-register is then used to address STP1; and the contents of the STP1 location addressed are gated to the Q-register in CPU mode.

MDM 5-87 is an operational diagram showing an example of the ISK STPO AS,HWLS±2 microword statement. In this case, the halfword local storage register

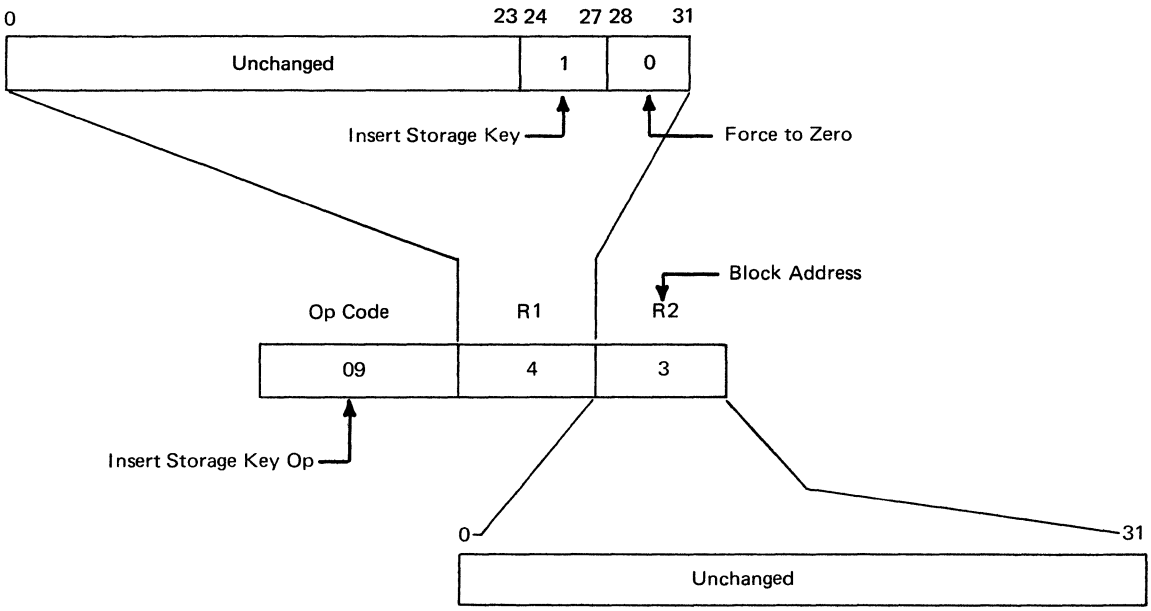


Figure 4-31. Insert Storage Key Instruction Word Format

addresses auxiliary storage instead of main storage. Depending upon the mode register, the storage key addressed by the contents of the M-register would be gated to the Q-register in CPU mode, the FQ-register in 2311 mode, the HQ-register in communications mode, or the GQ-register in channel mode.

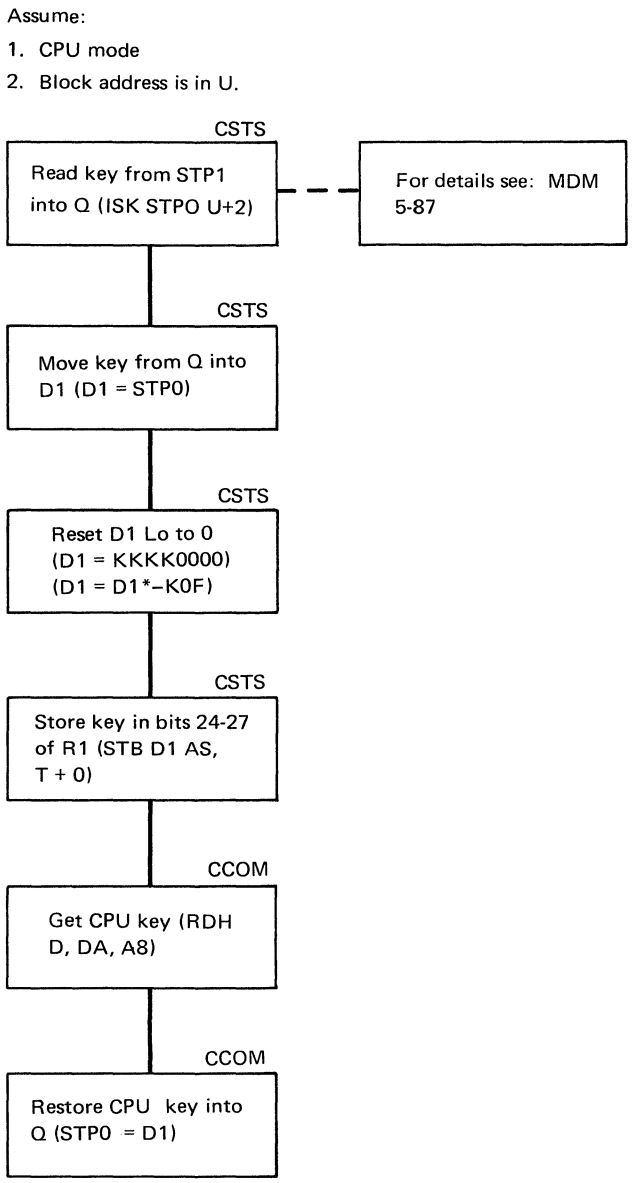


Figure 4-32. Insert Storage Key (Main Storage) CSTS Routine

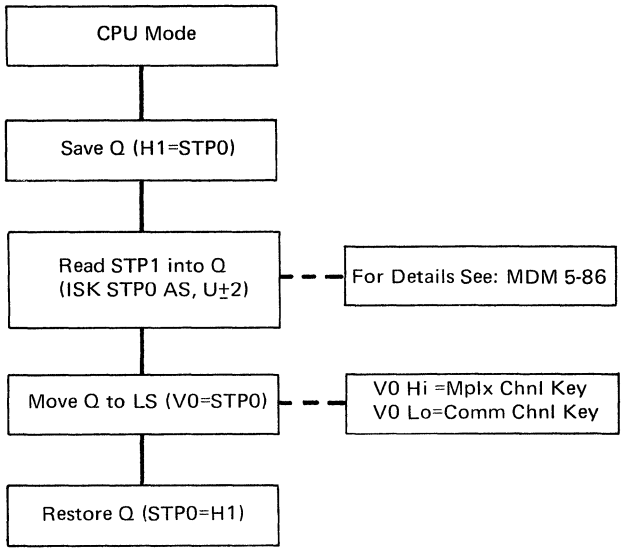


Figure 4-33. Insert Storage Key (CSW Store)

INTERVAL TIMER

- The interval timer feature consists of three bytes in main storage locations 50, 51, and 52.
- The value in the timer is decreased for intervals of time.
- An external interrupt is signaled when the timer goes from a positive to a negative value.

Consider an application of the interval timer feature. Assume that a customer must run two jobs during the day. Job #1 takes seven hours. The information to run Job #2 is not available until 2 pm. By using the interval timer feature, the customer, in effect, can instruct the System/360 to stop working on Job #1 and start on Job #2 at 2 P.M. If the customer knows that Job #2 is usually completed in 15 minutes, he could set the timer for 17 minutes. This would allow a 2 minute safety margin. At the end of 17 minutes the work on Job #2 is halted (whether completed or not) and processing of Job #1 is resumed.

To use this feature, you set a certain value in main storage locations 50, 51, and 52 (by machine language instructions). This starts a counter that keeps track of time. The value that is set in main storage represents total elapsed time. When the counter value is subtracted from the timer value often enough, the timer value goes from a positive to a negative value. At this time, an external interrupt is taken to whatever has been previously

set up by the customer. In this example, it would be the start of a routine to handle Job #2.

Examine the computation of timer values. The high-order bit of location 50 is reserved for sign control. This leaves 23 bit positions free for data. A value of over 16,700,000 can be set with 23 bit positions.

A microprogram routine subtracts 300 from the timer during each second of elapsed time. Thus, the full cycle time of the timer is about 15.5 hours. Because 300 is subtracted from the timer for an elapsed time of one second, the timer must be set to the value 1,080,000 for each hour (300/sec x 60 sec x 60 min) of elapsed time that is desired.

60-CYCLE OPERATION

- The value that is subtracted from the timer is determined by the setting of the 4-position binary connected TIM counter.
- The (TIM) counter is driven at a 60-cycle rate.
- The (TIM) counter is FULL every .25 seconds.
- Any latch on in the (TIM) counter causes a timer update at the start of I-cycles.

The (TIM) counter keeps track of actual time. A 60 cycle pulse from the power supply provides the drive. Though this 4-position counter is FULL with only 15 impulses (.25 sec), any position of this counter that is set prior to I-cycles causes a timer update. Refer to MDM 4-91.

The update loop (in the BSWI routine) takes the value in the (TIM) counter, multiplies it by 5, and subtracts the product from the timer value. When the timer value located at program storage address 50 (hex) is decremented from a positive value to a negative value, an external interrupt is initiated (Figure 4-34).

The external interrupt, if allowed and masked on (System Mask bit 7), causes the interruption code to be stored in the external old PSW. The external new PSW is read out and placed in command. If no sign change occurs, the update routine returns to the I-cycles routine (CICY) for the next instruction.

For every 1/4 of a second, a value of 75 (decimal) must be subtracted from the timer value in program storage. This fulfills the requirement of subtracting a decimal value of 300 from the timer every second of elapsed time.

If the update routine (BSWI) is entered and the external facility (TIM) contains all ones, the update sequence is bypassed, and a return to the I-cycles routine is made.

Note: The external facility (TIM) is the inverse of the hardware counter.

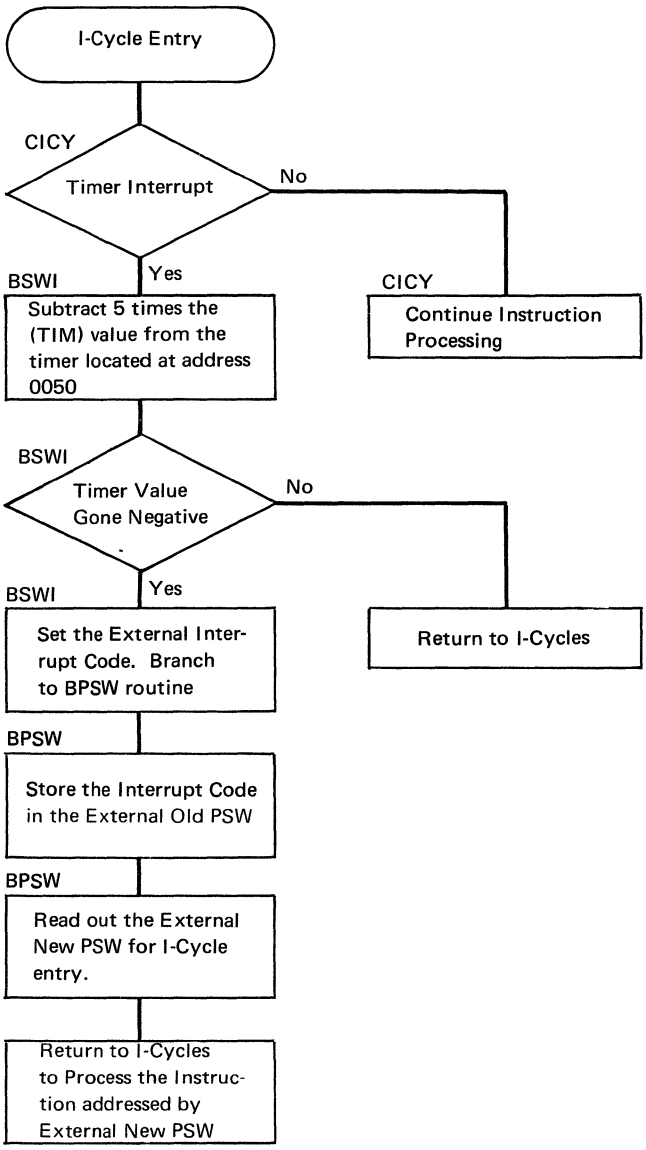


Figure 4-34. Microprogram Handling of Timer Update

50-CYCLE OPERATION

- The (TIM) counter is driven at a 50-cycle rate.
- The (TIM) counter is full every .3 seconds.
- Counter value multiplied by 6 is subtracted from the timer during timer update.

If the 2025 is operating on 50-cycle alternating current, a slight change in the timer update controls is necessary. Instead of multiplying the counter value by 5 to get the correct number to subtract from the timer, the 50-cycle machine timer update routine multiplies the counter value by 6. This is necessary because on a 50-cycle machine, the counter is full every .3 seconds instead of every .25 seconds.

By using a multiplier of 6 for 50 cycle machines, and a multiplier of 5 for 60 cycle machines, the same value (300) is subtracted from the timer on all machines. This makes timer programming compatible for all machines.

DIRECT CONTROL FEATURE

- Provides a means of communicating between 2 CPUs, or between a CPU and external devices.
- Communication is primarily control information.
- The write direct and read direct instructions are provided with this feature.

A CPU communicates with external devices by using the external interruption mechanism and the read-direct and write-direct instructions.

The direct control interface, Figure 4-35, permits communication between CPUs or between the CPU and an external device. The interface consists of:

Direct Control Bus-Out (8 lines)
Timing Signal Bus-Out (8 lines)
Direct Control Bus-In (8 lines)
External Signal Bus-In (8 lines)
Write-Out (1 line)
Read-Out (1 line)
Hold-In (1 line)
Read-In (1 line)

DIRECT CONTROL BUS-OUT

The direct control bus-out is a set of eight lines from the CPU to the external device. These eight lines are the output of the external register, JO, which is loaded through microprogramming as a result of the write direct instruction.

Data on the direct control bus-out is placed only during the execution of the write-direct instruction. The data on the lines represents the byte at the location designated by the operand address of the last write-direct instruction. The data placed on the direct control bus-out remains valid until intentionally changed, as for example, at the execution of the next write direct. The write-out pulse overlaps a change on the direct control bus-out by 100 nanoseconds, i.e., data already on the direct-out lines is valid for at least 100 nanoseconds after the rise of the write-out pulse to its up-level, and new data is valid at least 100 nanoseconds before the fall of the write-out pulse below its up-level.

TIMING SIGNAL BUS-OUT

The timing signal bus-out is the output of the external register, JA set by either write-direct or read-direct instructions.

Timing signal bus-out is a set of eight lines from the CPU to the external equipment. The external equipment could be another CPU, in which case the timing signal bus-out is connected to external-signal bus-in of the other CPU. The Sig-Out-0 and Sig-Out-1 lines are terminated and serve no purpose.

During a read direct or a write direct the eight bits contained in the instruction, positions 8-15, are sent out as eight timing pulses on these bus lines. The timing pulses have a duration of 500nsec to 1000nsec. The leading edge of the timing pulses coincides with the leading edge of the write-out or the read-out signal.

When the timing signal bus-out is connected to external signal bus-in of another CPU, the timing pulses on position 2 to position 7 cause an external signal interruption at the receiving CPU.

DIRECT CONTROL BUS-IN

The direct control bus-in is a set of eight lines from the external equipment to the CPU. The external equipment could be another CPU; in this case the direct control bus-in connects to the direct control bus-out of the other CPU.

The direct control bus-in to the 2025 is set into the JI external facility when allowed by the Hold-In control.

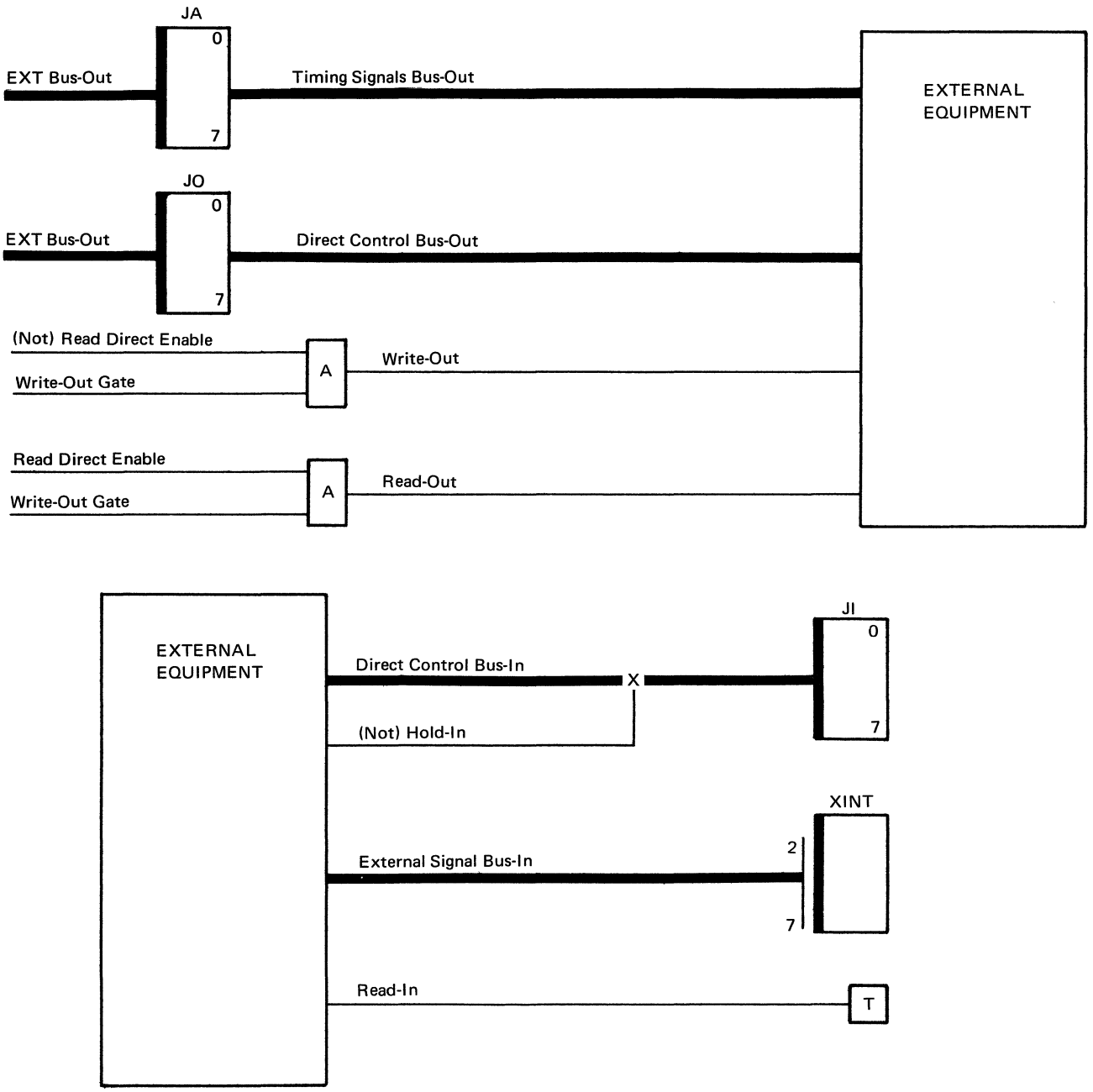


Figure 4-35. Direct Control Interface for 2025

The data appearing on direct control bus-in is read by the CPU only during execution of read direct. The data is stored in the location designated by the operand address of read-direct instruction.

EXTERNAL SIGNAL BUS-IN

External signal bus-in is a set of eight lines from the external equipment to the CPU. The external equipment could be another CPU, in which case the external signal bus-in connects to the timing signal bus-out of the other CPU. The Sig-Out-0 and Sig-Out-1 provide termination and serve no other function.

The external-signal pulses have a minimum active duration of 500nsec and maximum active duration of 1000nsec. The

external signal has a minimum inactive duration of at least 500nsec.

The external-signal pulse may occur at any time and has no relation to the timing of other signals on the direct control interface.

The purpose of the external signals bus is to provide a path to the external interruption mechanism of the CPU.

The XINT external facility receives the input of the external signals bus-in.

External interruption can occur only when system mask bit 7 is a one, and after the current instruction is completed. The interruption causes the current PSW to be stored at location 24 and the external new PSW to be fetched from location 88.

A total of six signal lines make up the external signal bus-in (Sig-In-2 through Sig-In-7). As a result of an external interruption, external signals are placed in bit locations 26 to 31 of the external old PSW.

An external signal request can occur at any time. The requests are pending until honored by the CPU. All pending requests are presented simultaneously when an external interruption occurs. Each request is presented only once. When several requests from one source are made before the interruption is taken, only one interruption occurs.

Sig-In-0 and Sig-In-1 provide no function except as a termination of the Sig-Out-0 and Sig-Out-1 lines from another CPU.

WRITEOUT

Writeout is a line from the CPU to external equipment. The external equipment could be another CPU, in which case the writeout line is connected to the hold-in line of the other CPU.

The function of the writeout line is to signal the external equipment when the CPU is placing data on the direct-out lines, and to indicate the data is presently invalid. The down-level of writeout indicates the data on the direct-out lines is valid.

READOUT

Readout is a line that connects the CPU to the external equipment. The external equipment could be another CPU, in which case the readout line is terminated without serving any function.

The purpose of the readout line is to signal the external equipment that a read direct is being executed, and that the external equipment must provide valid data on the direct control bus-in, as indicated by the down-level of the hold-in signal.

The leading edge of the readout signal must coincide with the leading edge of the pulses on the timing signal bus-out.

HOLD-IN

Hold-in is a line from the external equipment to the CPU. The external equipment could be another CPU, in which case the hold-in line is connected to the writeout line of the other CPU. The purpose of the hold-in signal is to prevent the CPU from reading the data from the direct control bus-in until such data is valid.

The hold signal is active for at least 100nsec on either side of any signal change on direct control bus-in.

After the readout pulse is generated, during the execution of read direct, the CPU tests for (not) hold-in condition to read the direct control bus-in.

Because the CPU hangs up waiting for a (not) hold-in condition, the external equipment maintains the hold-in inactive for at least 500nsec after the termination of every readout pulse.

The hold-in signal can occur at any time; it does not have to be synchronized with the readout pulse.

READ-IN

Read-in provides no function except as a termination for the readout line.

APPLICATION

- The CPU can initiate a read or write command over the direct control interface.
- One byte of information is transferred for each read or write command.

Two situations in the following text are used to illustrate the sequence of operation for the direct control feature. For illustration purposes, two CPUs are assumed to be attached by The second CPU also could be a control unit for any

machine (IBM or non-IBM) the customer desires to attach to the direct control interface.

Situation A

CPU #1 desires to send a byte of information to CPU #2.

1. CPU #1 executes a write-direct (84) instruction. Byte 2 of the instruction (IZ field) is placed on timing signal bus-out 0-7. The character in the storage location is read out and placed on direct control bus-out 0-7 (JO-register).
2. CPU #2 receives the timing signals on external signal bus-in 0-7. Receiving the timing signals causes CPU #2 to take an interrupt. CPU #2 tests the interruption code in the old PSW and determines the cause of the interrupt. CPU #2 executes a read direct (85) instruction, and reads into storage the character present on direct control bus-in. The storage address is determined by the B1+D1 address in the read direct (85) instruction. CPU #2 also transmits the I2-field of the read direct instruction back to CPU #1 on timing signal bus-out 0-7.
3. CPU #1 receives the signal on external signal bus-in, and an interrupt occurs. CPU #1 tests the interrupt and determines that CPU #2 received the character CPU #1 transmitted. CPU #1 is now able to execute another write-direct instruction if necessary.

Situation B

CPU #1 is interrupted by direct control because CPU #2 is ready to send CPU #1 a byte of information.

1. CPU #1 receives a signal on external signal bus-in 0-7. This signal causes CPU #1 to take an interrupt. Interrogation of the old PSW determines that a character is being sent to CPU #1 on the direct control interface.
2. CPU #1 executes a read direct (85) instruction and stores the character present on direct control bus-in into storage. The location in storage for that character is determined by the B1- and D1-field of the instruction. CPU #1 also transmits the I2-fields of the instruction out on timing signal bus-out to CPU #2. These timing signals notify CPU #2 that CPU #1 received the character on direct control bus-in.

Multiple devices can be attached to the direct control interface. When multiple devices are attached to the direct control interface, the timing signal bus-out bit configurations normally are used to select the I/O device desired.

EXTERNAL INTERRUPT

The External Interrupt feature provides a means for external devices to signal the 2025 that some action must be taken by the system to handle the cause of the interrupt request.

External signals are set into the external facility XINT. (Refer to MDM 4-91.) When the processing of the current machine language instruction is completed and a return to the microroutine CICY is made, the interrupt is recognized and handled.

The handling of the interrupt is microprogram controlled and accomplishes the following.

1. Bits 2-7 of the XINT external facility are placed in the interruption code portion (bits 26-31) of the external old PSW located at hex location 18.
2. The external new PSW, hex location 58, is read out, and the instruction addressed by the instruction address of the external new PSW is executed.

- All ac and dc power requirements for the CPU, including features and integrated attachment circuitry, are supplied through the 2025 power system.
- Standard SLT Mid Pac supplies are used for most dc voltages.

Input power for the Model 25 for domestic installations is 208 or 230 volts ± 10%, 60 ± .5 Hz, 60 amperes, via a three phase, four-wire (fourth wire is equipment ground) shielded cable. Input power for the Model 25 for World Trade installations is 200, 220, 235/380, or 408 volts ± 10%, 50 ± .5 Hz 60/25 amperes via a three-phase, five-wire (fourth wire is neutral for a WYE system and not used on Delta, and the fifth wire is equipment ground) shielded cable. An auto-transformer is provided for 200, 235, or 408 volts 50 Hz inputs.

The 2025 uses the middle power package (Mid-Pac) system to develop the regulated dc voltages needed by the CPU logic, memory and integrated attachments. In the Mid-Pac system, ferroresonant transformers feed full wave rectifiers and filter capacitors to convert the system ac input voltage to filtered, unregulated dc voltages. Mid-Pac regulators, operating in series with the unregulated dc voltages, provide load regulation by means of differential amplifiers and series power transistors (Figure 5-1). Some dc outputs of the ferroresonant transformer/rectifier/filter assemblies are used in the filtered but unregulated state. For these special voltage requirements, Mid-Pac regulators are omitted. For a more complete description of the Mid-Pac power system, see SLT Power Supplies, Form 223-2799.

There are five non-Mid-Pac power supplies in the 2025 Processing Unit. The 24-volt dc control voltage required by the power control circuits consists of a step-down transformer (T3) feeding a full-wave rectifier and filter. On 60 Hz machines, an isolation step-down transformer (T4) produces 110V ac for convenience outlets located in the processing unit and in the integrated attached I/O devices. The time meters are powered and controlled by means of a separate ac power supply. Power supply 13 is a ferro-regulated ac supply that provides 7.25V ac for powering console signal lamps in the system. Power supply 14 is a special supply for producing the +60 volts required by hammer driver circuits for the integrated 1403.

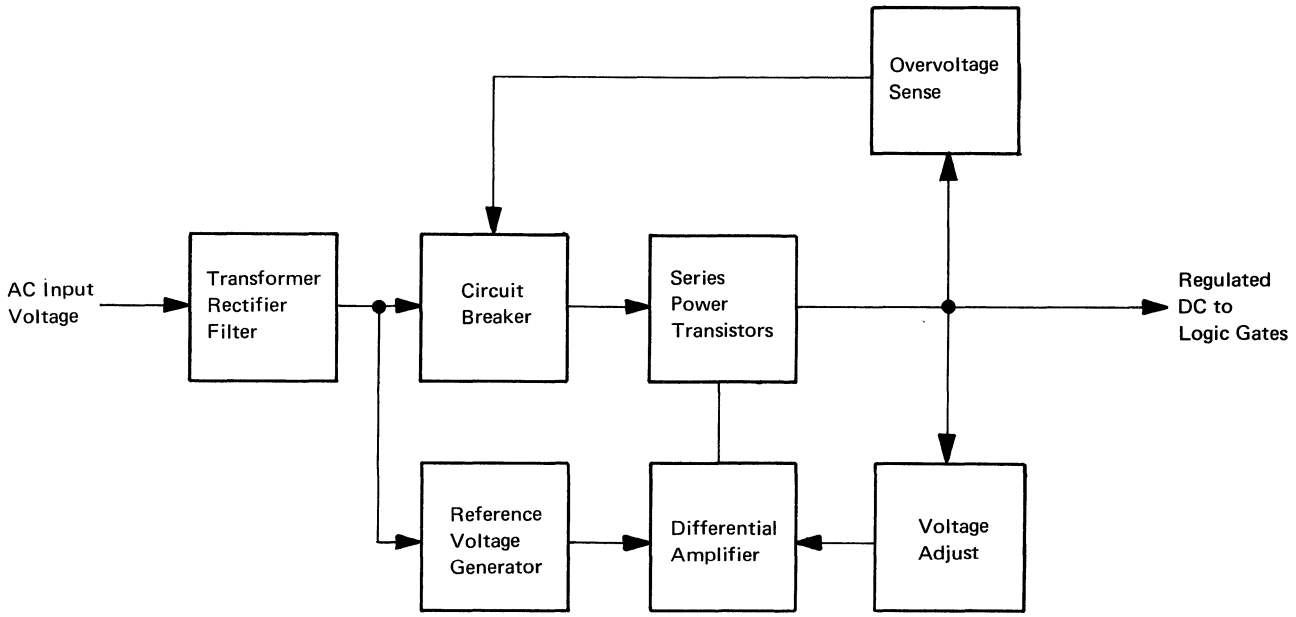


Figure 5-1. Mid-Pac Power System

POWER-ON SEQUENCE

- Before a power-on sequence can be initiated, the EPO pull switch must be in its nonoperated state.
- Overvoltage, overcurrent, overtemperature must be normal, power sequence to all I/O channel-related devices must be complete, and the voltage sense circuit must sense the correct voltages before power-on sequence is complete.

For the following discussion, refer to the power-on sequence chart on logic page YA031. With mainline power applied to the processing unit, but before the power-on key is pressed, the convenience outlets are powered and power is applied to the 24-volt power control voltage power supply (see Logic page YA092A). If the EPO circuits are normal, relay K1 is up to close the 24-volt supply to the power circuits. With power control voltage up, thermal sense relay, K20, picks provided all thermal sense contacts are in their normal (closed) state. Provided all power supply circuit breakers are normal, pressing the power-on key picks power control relay K21 (YA161). This picks relay K14 to start the power-on sequence (YA151F). Relay K14 picks relay K2 (YA151B). When relay K2 closes, ac power is supplied to the 1052, the 2311, the 2540, dc power supplies, meter power pac, and all fans. Within about 80 milliseconds, the following dc voltages are up at their respective power supply terminals: -3, +3, -6, +6, -12, +12, -20, +20, -36, +48, +60. With the key logic voltages up, voltage sense relay K18 picks and this picks relay K22, the I/O interface control relay to provide a power cycle reset to the I/O channel. Relay K18 also picks relay K12 which completes the circuits to provide the controlled -20 to the 2540, and the controlled -30 to the memory units. Relay K12 also allows ac power to turn on in the 2560 (Relay K3 picks).

Special voltage sense relay, K19, picks if all the following voltages are sensed at their correct levels: -6, -20, -30, -36, and +48. Relay K19 must pick before the power on sequence can be completed. Relays K11 and K4 are then picked through the shoe connector, CCPR (YA 131). Relay K11 turns on the special controlled +60 volts to the printer. Relay K4 completes the circuit for one phase of the ac power to the 1403 main motors (YA131). Relay K4 picks relay K6 to complete the other two phases of ac power through an SCR switch (YA131). The SCR switches provide arcless switching for motor control.

Provided power is on in the 1403, relay K5 picks. This completes the circuits for another controlled +60 supply and a controlled +6 to the printer. If printer T-casting is closed, relay K10 picks to control ac power to the chain and ribbon motors. Relay K10 picks relay K8, and K8 picks K7. Relays K7 and K8 control the SCR switches that provide arcless switching for phase 2 and phase 3 of ac power supplied to chain and ribbon motors.

Special voltage sense relay, K19, starts the power-on sequence for I/O devices connected to the System/360 standard channel (YA171). The I/O EPO relay, K25, is on as long as the EPO circuits have not been activated. When K19 picks, the circuit is completed to power-sequence control circuits in the first I/O device (connected to power connector J1). When the power-on sequence in that device is complete, a relay in the device completes the circuit between pins 3 and 4 of the power connector. This allows relay K26 to pick to start the power-on sequence of the next device. This process continues until all channel I/O device control units have cycles power on. When the last device powers on, relay K17 picks. If any I/O devices have their local/remote switches in the local position, those devices are bypassed in the power-on sequence (YA171B). K17 causes the power-on key white backlight to turn on (YA151E) indicating that the power-on sequence is complete. When relay K17 picks, the power-on reset relay, K2 drops to complete the power-on reset (YA161J).

POWER-OFF SEQUENCE

- System power is turned off in varying ways, depending upon the power-off reason.
- Power can be turned off from the console (power-off key or EPO pull-switch), or by one of several system protective devices (over-temperature, overvoltage, overcurrent, low-voltage sense).

POWER-OFF KEY

- Pressing the power-off key causes a power-off sequence in the processing unit and in any processing unit controlled devices.
- After the power-off-sequence, the 24V ac power control voltage and convenience outlet voltages are still available.

Pressing the power-off key breaks the circuit to relay K21 to start the power-off sequence. When K21 drops, K22 and K12 also drop. relay K22 causes a power cycle reset (Ya161). This prevents spurious I/O

interface signals from occurring during the power-off sequence. If there are no System/360 I/O channel-connected devices, the drop of K21 also drops K17 to provide a power-on reset during the remainder of the power-off sequence. This prevents any unwanted logic action during the power-off sequence. If there are System/360 I/O channel-connected devices, the power-on reset occurs after the I/O devices have completed their power off-sequences. Relay K17 also causes the white portion of the console power indicator to turn off and the red portion to light.

When relay K21 drops, relay K12 is dropped. Relay K12 causes power to turn off at any System/360 channel I/O devices set up for remote power control (YA171), and initiaies the power-off sequence for motors and control voltages for the integrated 1403 attachment. Relay K12 also initiates power off for the integrated 2560 attachment voltages by dropping relay K3. Power-off sequence for the integrated 2540 attachment is also begun when relay K12 points open the -20V dc supply to the 2540. Relay K12 also opens the input circuit to the -30V dc memory power supply as well as the output of this supply (Ya201). Removing the memory-driver supply voltage completely before removing any logic voltages ensures that memory data will not be affected by the power-off sequence.

When relay K12 opens the -30 volt supply circuit, relay K19, the special voltage sense relay drops. When the K19 points open, a special controlled ground signal to the 2311 opens. This notifies the 2311s to retract the head assemblies in preparation for system power off. (The heads must be retracted before the 2311 drive motors are stopped because the heads depend on disk surface air movement for proper head-to-disk distance.) When all heads have been retracted, relay K14 drops (YA151), and the system console red power light turns off. Relay K14 drops relay K2, the logic and ac power control relay. With input power open to all logic supplies, relay K18 (logic voltage sense) drops and the power on reset drops (power on reset is supplied from the -3V supply). The 24-volt control voltage remains on, as well as the convenience outlet voltages for all units.

EMERGENCY POWER OFF (EPO)

- Pulling the EPO switch removes all system ac and dc power except for the 24V ac power control voltage.
- When pulled, the EPO switch latches in the out position and must be restored by a CE.
- When two processing units are tied together, power for both systems can be removed by operating either console EPO pull switch.

Operating the EPO pull switch on the console opens the circuit to relay K1 (YA151). When relay K1 drops, all convenience outlet power drops, and the +24 volt control voltage tc all control relays is opened (YA151). This causes all control relays to drop, including relays K2 and K3. Relay K2 opens the ac supply to all logic and control voltages as well as the ac supply to all cooling fan motors, integrated attached device motors, etc. Relay K3 opens the ac supply to the 2560 motors.

The EPO pull switch is a two-pole switch. The second pole is connected to two terminals of J24 (YA151). A second system, when connected to J24, is controlled by this second switch pole (Figure 5-2). Likewise, this system is controlled by a second pole on the EPO switch of a second system. This second switch pole is connected in series with the first system EPO switch via jack J24 (YA151A, YA151J).

When the EPO pull switch has been operated, a mechanical latch holds the switch in the transferred position. This mechanical latch must be restored by a CE who must gain access to the rear of the console panel. Once the EPO pull switch has been restored, power can be turned on via the system power-on key with no further restore procedures required.

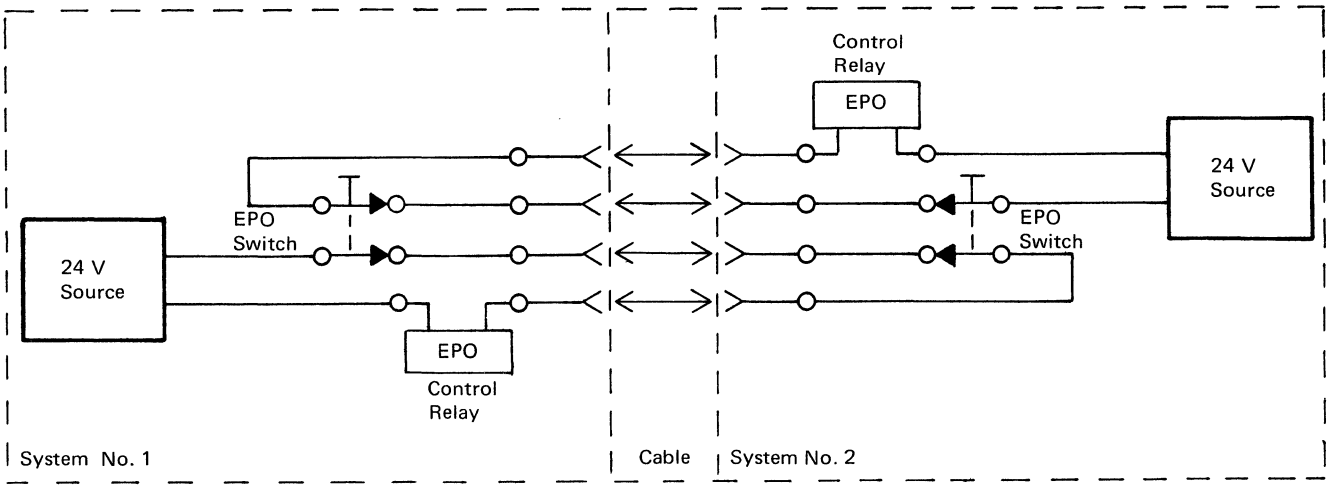


Figure 5-2. Two-System EPO Control

OVERCURRENT, OVERVOLTAGE DETECTION

- The Model 25 power system contains overcurrent detection for all dc supplies and overvoltage protection for certain critical dc supplies.

- Overcurrent or overvoltage detection causes a power-off sequence ending with the console power check light on.

Each dc power supply has an overcurrent-sensing circuit breaker. An overcurrent condition on any supply causes the associated circuit breaker to trip mechanically, and the CB-trip light to turn on (light DS1 on the power control chassis, logic page YA021). A normal power-off sequence occurs when the CB-trip sensing circuitry opens the pick circuit for power control relay K21 (YA161). In addition to lighting the DS1 light, the CB-trip sensing circuit picks the power check interlock relay, K16. This lights the console power check light and interlocks the power-on sequence.

To restore power on, the condition causing the CB-trip must be corrected and the tripped CB must be reset. Pressing the console power-off key resets the power interlock by dropping the power check interlock relay, K16 (YA161). Normal power-on sequence can then be accomplished by pressing the power-on key.

Certain key power supplies have direct overvoltage detection. These are PS-1 (-6), PS-3 (-12), PS-8 (+12), PS-10 (+6), and PS-11 (+6). Detection of an overvoltage condition in these supplies causes the CB for the respective supply to trip, and the power-off sequence to occur as previously described.

VOLTAGE SENSING SYSTEM

- A voltage sensing system determines that all dc voltages are on and are supplying at least a certain minimum output.

- Failure in any dc supply causes power to be sequenced off and the power check light on the system console to be turned on.

A special voltage sensing system monitors the output of all logic power supplies to determine that these supplies are on and are supplying at least a certain minimum output. (This system consists of four SMS cards mounted on a small gate on the left side of the processing unit and one relay (K23) mounted on the power control relay chassis. See logic pages YA021 and YA181.)

Loss of any sensed voltage below the required minimum causes relay K19 to drop (YA161). Relay K19-4 points (YA151G) close to pick the power check interlock relay, K16. Relay K16 drops the power-on pick relay, K14. Relay K14 drops main power relay K2 and all logic and fan power is removed (YA151C). The console power check light is on, but neither red nor white power-on lights are on.

To restore power, the power-off key must be pressed to reset the power check light and the condition causing the voltage sense failure detection must be corrected. Then, pressing power on causes a normal power-on sequence.

THERMAL SENSING

- Thermal sensing switches are provided on logic gates and core storage gates, and in the power supply area.
- Sensing an overtemperature condition causes a power check with a normal power-off sequence.

Thermal sensing switches provide overtemperature protection in various areas throughout the processing unit (see logic page YA181). An overtemperature condition causes an associated overtemperature switch to transfer open. This drops the thermal sense control relay, K20 (YA161). Relay K20 turns on the thermal sense light on the power tower (light DS2), and picks the power check interlock relay K16. Relay K16 causes a power-off sequence and lights the console power check light.

To restore power after a thermal trip, the system power-off key must be pressed to reset the power check. Then the condition causing the thermal trip must be corrected, and the thermal reset switch (S1) on the power tower must be pressed (see logic page YA021). This drops the power control relay K21 so the thermal sense relay K20 can be repicked. Note that a failure of the temperature controlling circuit in either memory will cause the respective heater to stay on, thus keeping the thermal sense control relay down. In this case, power cannot be restored until the condition causing the memory heater to stay on is corrected. Once the thermal condition is corrected and the thermal sense circuit is reset, pressing system power on starts a normal power-on sequence. For additional information about the memory heaters, refer to Power Supply and Temperature Compensation in the Core Storage section of Chapter 2.

POWER DISTRIBUTION

- Power distribution for the processing unit and all integrated devices is through the 2025 processing unit.
- Power for channel-connected devices may be controlled by the processing unit, but enters at the devices themselves.

Three-phase power enters the 2025 and passes through a line filter. All three phases of power are controlled by the master circuit breaker, CB-1 (Figure 5-3). From the output of CB-1, power is split into single and three-phase requirements for all power supplies, motors, blowers, and convenience outlets for the processing unit and all integrated devices.

Devices attached via the System/360 standard channel do not receive their power from the 2025. However, these devices may be power-controlled by the processing unit via power control cables connected between the devices and the processing unit (see logic page YA171). The EPO interconnecting cables for these devices are plugged into jacks provided in the power-on control circuits (YA171). A special jumper is connected to pick relay K17 from the next I/O relay for the last device attached to the System/360 standard channel.

MARGINAL CHECKING

- Designed marginal check facility exists for the natively attached 2540.
- The +12M voltage may be varied ±3 volts for isolating and diagnosing failures.

A designed marginal check facility exists only for the natively attached 2540. A jack receptacle on the power control chassis allows plugging a portable marginal check power supply so the +12M voltage can be varied ± 3 volts for certain circuits in the 2540. (Marginal checking for a 2540 attached to the channel is provided via the 2821 control unit.)

All other dc voltages may be varied within their stated tolerances by means of the individual power-supply amplifier card adjusting potentiometers. However, varying these voltages has only limited diagnostic value.

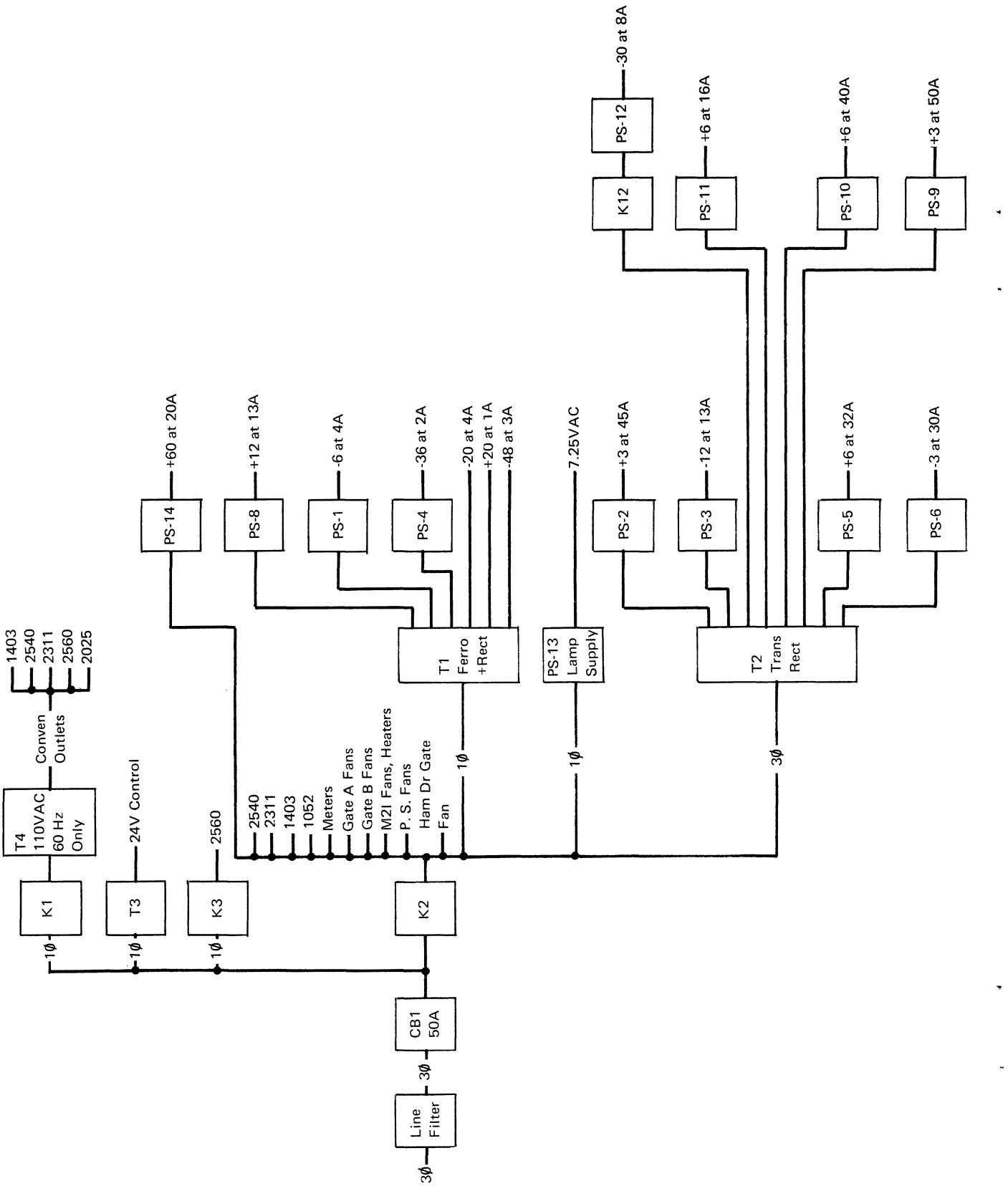


Figure 5-3. System/360 Model 25 Power Distribution

SYSTEM CONTROL PANEL

- The system control panel (Figure 6-1) is made up of four sections.
  1. Displays
  2. Operator Control Section
  3. Operator Intervention Section
  4. Customer Engineering Controls

The Model 25 system is manually controlled by fifteen pushbutton switches and seven rotary switches. Two of the rotary switches control the mode of operation and check control. Four hex-coded switches are used for addressing and manual entry of data. One rotary switch is used for diagnostic control.

INDICATORS OR DISPLAYS

Approximately 50 indicators are available for displaying data and status information. Error and status displays use about half of these. The remainder are used to display the storage address register and other selected registers or locations.

BYTE-0 AND BYTE-1 INDICATORS

The byte-0 indicators are on the output of the local storage data assembler. The byte-0 indicators are used, during display operations, to display:

M0-register P,0-7
W0-registers P,0-7
Storage data-out P,0-7
Modifier P,0-7
B-register P,0-7
Z-bus P,0-7
Control register 0-7.

The byte-1 indicators are on the output of the AB-assembler. The byte-1 indicators are used, during display operation, to display:

M1-register P,0-6
W1-register P,0-6
Control register 8-15
A-register P,0-7
External bus-in P,0-7
Local-storage P,0-7
Storage data-out P,8-15.

CPU STATUS INDICATORS

Data Cycle, File: This light is on during a file share cycle.

1400 Mode (Bit 0): Indicates operation in 1400 emulator mode.

2020 Mode (Bit 1): Indicates operation in System/360 Model 20 compatibility mode.

EXT Mode (Indicators 2, 3, and 4): With 2020 mode indicator (Bit 1) either off or on, the mode for external gating control shown by indicators 2, 3, and 4 is:

234	Mode Indicated
000	CPU
001	2311
010	2540 Punch
011	1403
100	1052
101	Communications
*110	2540 Reader
111	Channel

*The 2020 mode indicator combination off means 2540 reader mode. The 2020 mode indicator combination on means 2560 mode.

Note: The EXT mode indicators 2, 3, and 4 are not connected directly to the mode register. See MDM 4-14.

LS Zone (Indicators 5, 6, and 7): These indicators show the local-storage zone that is being used for a particular operation.

567	Local Storage Zone
000	Zone 0 - CPU
001	Zone 1 - 2311
010	Zone 2 - (Not used)
011	Zone 3 - (Not used)
100	Zone 4 - Backup area
101	Zone 5 - Communications
110	Zone 6 - 2540
111	Zone 7 - Channel

Note: LS Zone reg indicators 5, 6, and 7 are not connected directly to the mode register. See MDM 4-32, part 3.

Address Match (ADR MATCH): If the MODE switch is in the AS ADR STOP position and the address in switches A, C, and D compares to the address in bits 0-3 of the M0-register and M1-register (bits 0-7), the clock is stopped and the ADR MATCH indicator is turned on. (M0 bits 0-4 are not used to access auxiliary storage.) The clock stop occurs before the word at the matching address is read out.

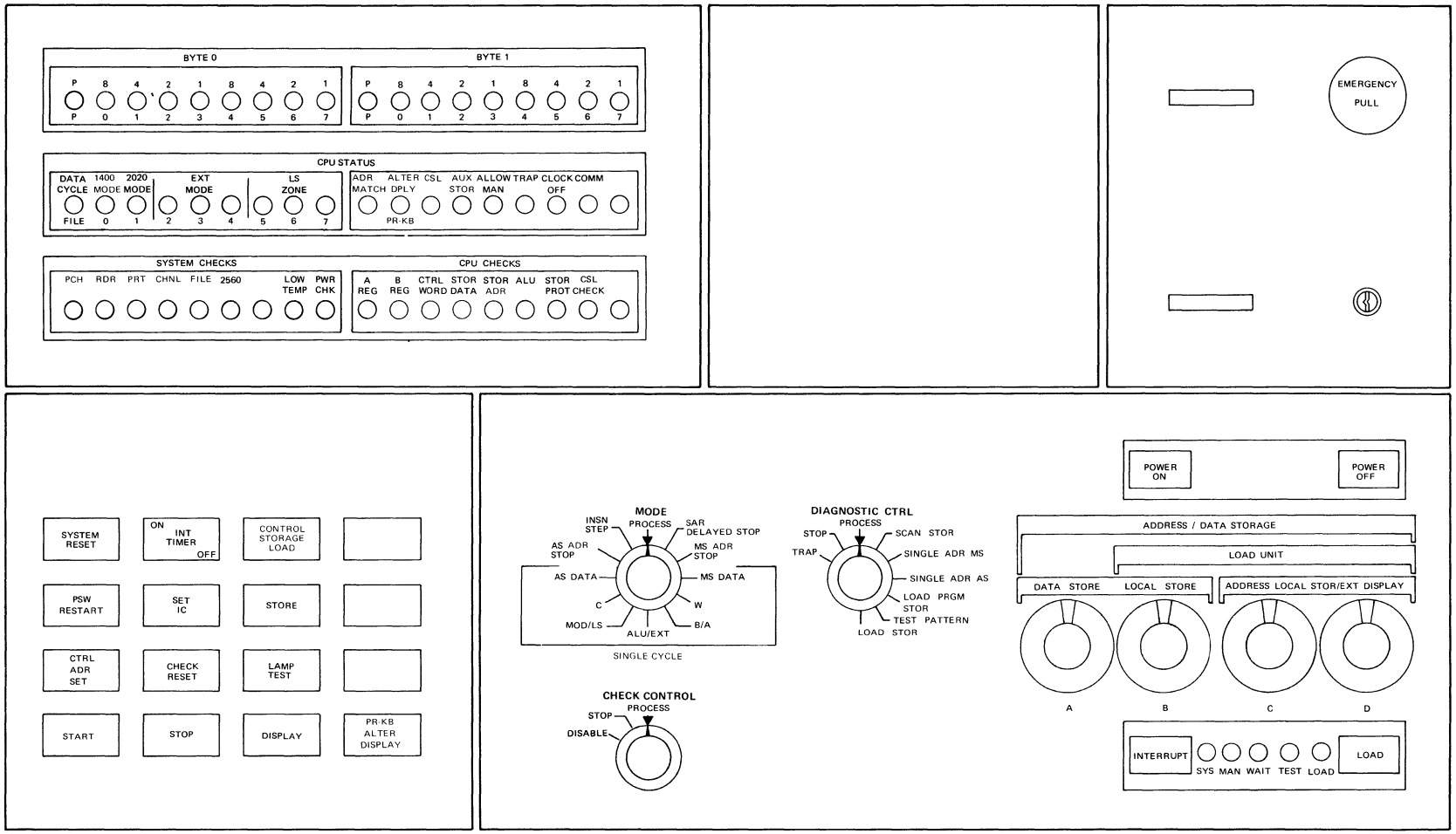


Figure 6-1. System Control Panel

If the MODE switch is in the MS ADR STOP position and the address in switches A,B,C, and D compares to the address in M0 and M1, the clock is stopped and the ADR MATCH indicator is turned on.

If the MODE switch is in the SAR DELAYED STOP position and the address in switches A,B,C, and D compares to the address in M0 and M1, the current macro instruction is completed, the 'address match soft stop' latch is set, the ADR MATCH indicator is turned on, and the soft-stop loop is entered.

Alter/Display Printer-Keyboard (ALTER DPLY PR-KB): This indicator is turned on when the 'alter/display active' latch is set. The indicator remains on until the alter/display operation is complete and the 'alter/display active' latch is reset. See also the description of this indicator in the section Keys and Indicators (PR-KB).

Control Storage Load (CSL): This indicator is turned on when the CSL latch is set. The CSL indicator remains on until the control-storage load routine is completed and the CSL latch is reset. A flowchart of the CSL operation using the 2540 reader is shown on MDM 5-20.

Auxiliary Storage (AUX STOR): This indicator turns on when:
1. In the first cycle of a storage control word that addresses auxiliary storage.
2. The MODE switch is in the AS DATA position.
3. The DIAGNOSTIC CTRL switch is in the SCAN STOR position and the AUX SCAN latch is set.

Allow Manual Operation (ALLOW MAN): This indicator turns on to notify the operator when a manual alter or display operation of main or auxiliary storage can be accepted. The light is on when:
1. The mode switch is in either the MS-data or AS-data position.
2. The system clock has been stopped, and completion of both cycles of a storage microprogram word.
3. The machine is not stopped after the first cycle of a storage word.

Trap: This indicator turns on for any trap request. a trap request bypasses normal core-storage addressing and forces a predetermined control-storage address. The trap addresses are defined in Chapter 2 under Traps and Priority.

Clock Off: The clock-off indicator turns on when the clock is stopped. That is, the clock-start latch has been turned off and the clock-off latch is set. The following conditions result in a clock-off indication.

1. Hard-stop latch on. This latch is turned on by a parity error when the check control switch is in the check-stop position.
2. Clock stop pulse. This signal is generated by:
   a. Any single-cycle or address-stop position of the mode switch.
   b. A stop control word.
   c. Several other conditions that include diagnostic functions. (See MDM 4-11.)
3. 'Machine reset sw' signal. This line is activated when one of the following keys is held in the operated position: System Reset, Load, Control Storage Load, or Start. The clock is restarted after the key is released.

Communications (COMM): This indicator turns on for any communications data cycle.

SYSTEM CHECKS

Punch (PCH): This indicator turns on for any punch condition in the integrated 2540 attachment requiring operator attention. These conditions include:
1. Punch interrogation (not ready)
2. Punch holecount check
3. Punch address register error
4. Overrun
5. Punch shift sync error
6. PFR validity error.

The PCH indicator and associated logic are shown on MDM 4-105.

Reader (RDR): This indicator turns on for any reader condition in the 2540 integrated attachment that requires operator attention. These conditions include:
1. Reader interrogation (not ready)
2. Reader holecount check
3. Reader address register error
4. Overrun
5. Reader shift sync error
6. Reader validity error.

The RDR indicator and associated logic are shown on MDM 4-104.

Printer (PRT): This indicator turns on for any condition in the 1403 integrated attachment that requires operator attention. These conditions include:
1. Hammer-check error
2. Print-line buffer parity check
3. Program sense reset and (not) print ready. Printer microprogram issues a reset to PRA bit 0 to turn off hammer check, PLB parity, and channel 9 and 12 latches when printer is not ready. The PRA 0 reset occurs at the beginning of a start I/O for all commands other than a sense or No-Op.
4. CE switch S3 is on.

Channel (CHNL): This indicator applies to either the byte channel or the burst channel depending on which option is loaded in control storage. The CHNL indicator turns on for an A-register parity error that is detected when the channel microprogram is sending information to the CPU on channel bus-in (GB/IN). This information can be:
1. Data being sent to core storage
2. Address or status to local storage.

File: This indicator turns on for the following CPU-detected file errors.
1. Machine check error
2. Storage protect error
3. Storage wrap error.

2560: This indicator turns on for any 2560 condition requiring operator attention.

Low Temperature (LOW TEMP): This light turns on when the temperature at the core storage array(s) is below 96±5 degrees. When power is turned on initially, this light comes on and remains on until the array is at proper operating temperature (about two minutes). The low-temperature condition, itself, does not prevent an attempt to operate the machine.

Power Check (PWR CHK): This light turns on and a normal power-off sequence occurs for any of the following conditions.

1. Overcurrent sense. Each dc power supply has overcurrent detection. A circuit breaker on the supply trips if an overcurrent condition occurs. The CB-trip light (indicator DS1 on the power control chassis) turns on to identify the cause of the power check.
2. Overvoltage sense. Certain key power supplies have overvoltage detection. These are PS-1 (-6), PS-3 (-12), PS-8 (+12), PS-10 (+6), and PS-11 (+6). Detection of an overvoltage condition in any of these supplies trips the circuit breaker for that supply and turns on the CB-trip light.
3. Voltage sense. Loss of output or low output from any dc supply is detected by a voltage sensing system. This system includes both relays and voltage sensing cards.
4. Thermal sense. Thermal sensing switching is provided on logic gates, core storage, and power supply areas. Whenever any of these thermal switches sense a temperature in excess of its specified limit, a contactor drops and the thermal-trip light (on the power control chassis) turns on.

After a power check, power-on sequence cannot be accomplished until (1) the cause of the power check is corrected (if still present), (2) the circuit breaker or thermal reset switch is operated, and (3) the power-off key is operated to reset the power-check light.

The power-check light also turns on when no power-down sequence is possible because any of fuses F30 through F35 have blown. These fuses are in the 1403 attachment. Because print checks and coil-protect checks can occur if these fuses are open, this indication is provided. The light is turned off when the fuse is replaced.

CPU CHECKS

Parity Indicators: These lights, when on, indicate that incorrect parity was detected at the point checked. The parity indicators are:

A-register (A-REG). Checked at A-Reg output.

B-register (B-REG). Checked at B-Reg output.

Control word (CTRL WORD). Checked at the storage data register (storage data out bus) for both input and output of core storage.

Storage data (STOR DATA). Checked at the storage data register (storage data out bus) for both input and output of core storage.

Storage address (STOR ADR). Checked at the output of the M0- and M1-registers.

ALU: Turns on when an error is detected in the ALU circuits during an ALU cycle. The ALU check is not a parity check but an actual bit check of the two-wire adder.

Storage Protect (STOR PROT): Turns on to indicate that the data read out of the storage-protect buffer has incorrect parity.

CSL Check: This indicator is turned on when a control-storage loading error is detected by the checksum microprogram routine. This routine is initiated after each control-storage load, or when the system reset or load keys are pressed, to ensure that control storage was properly loaded and that it has not been changed.

Proper problem-program operation is assured only when the system is re-initialized using the appropriate CSL deck. If the CSL check indication persists, check the CSL deck for missing cards, and see that the deck being used matches the engineering level of the system.

In CE mode only, the check-sum error is automatically corrected. This procedure turns off the error indication and generates a correcting factor for the checksum. This correcting factor is

included for subsequent checks (system reset or load) until another CSL is attempted.

The checksum microprogram routine turns on the CSL check light by setting diagnostic register bit 6 (DR6) with a set/reset word. For detailed information about the checksum routine, refer to the BCHK routine and text on the microlistings.

OPERATOR CONTROL SECTION

- This section provides:
  Control and indication of power
  Indication of system status
  Operator-to-machine communication
  Initial program loading.

This section of the system control panel contains only the controls required by the operator when the CPU is operating under full supervisor control. Under supervisor control, a minimum of direct manual intervention is required because the supervisor performs operations such as store and display.

SWITCHES AND KEYS

Emergency Pull Switch: Pulling this switch turns off all power beyond the power-entry terminal on every unit that is part of the system or that can be switched onto the system. Therefore, the switch controls the system proper and all off-line and shared control units and I/O devices. The contents of core storage can be affected when this switch is operated because power is not sequenced down. Refer to Emergency Power-Off (EPO) in Chapter 5.

The switch latches in the out position and can be restored by maintenance personnel only.

When the emergency pull switch is in the out position, the power-on key is ineffective.

Power-On Key: This key is pressed to initiate power-on sequence of the system. See Power-On Sequence in Chapter 5.

As part of the power-on sequence, a system reset is performed in such a manner that the system performs no instructions or I/C operations until explicitly directed. The contents of core storage are preserved.

The power-on key is backlighted to indicate when the power-on sequence is completed. The key is effective only when the emergency pull switch is in its normal position.

The power-on key stays pink to indicate that the system power-on or power-off sequence cannot be completed because of a malfunction in a power supply or on I/O device.

Power-Off Key: The power-off key is pressed to initiate the power-off sequence of the system. See Power-Off Sequence under Chapter 5.

The contents of core storage are preserved. The power-off key overrides the power-on key.

Interrupt Key: The interrupt key is pressed to request an external interruption. Circuit and microprogram objectives for the interrupt key are as follows.

When the interrupt key is pressed, then released, the console interrupt latch is set. This latch resets bit 3 of the BB external field unless the external interrupt is masked off (PSW bit 7). Also (not) S7 branch condition is activated. On the basis of these two bits (S7 and BB3), the microprogram branches to take the interrupt by loading the external new PSW.

The interruption is taken when not masked off, and when the CPU is not in the stopped state. Otherwise, the interruption-request remains pending. Bit 25 in the interruption-code portion of the current PSW is set to 1 to indicate that the interrupt key is the source of the external interruption. The key is effective only while power is on the system.

Load Key: The load key is pressed to start initial program loading (MDM 5-22). A system reset is performed before the load operation. The key is effective only while power is on the system and a valid control-storage load microprogram resides in core storage.

Load-Unit Switches: Three rotary switches (switches B, C, and D) provide the I-O address bits used for initial program loading. (See Figure 6-2).

All are 16-position rotary switches labeled with the hexadecimal characters 0-F. The channel address (0 or 1) is loaded from switch B and the unit address from switches C and D.

INDICATORS

System (SYS): This indicator is on whenever the customer or CE use-meter is recording time. See Metering Switch.

Manual (MAN): Whenever the CPU clock is stopped (clock off indicator) or the CPU is in a soft-stop loop, this indicator is on.

Wait: This light is on when the CPU is in the wait state (i.e., CPU clock is running but instruction execution is not taking place). The machine is placed in the wait state after loading a PSW having the wait bit on. I/O requests and interrupts are handled while in the wait state. The machine remains in the wait state until a new PSW with the wait bit off is loaded by an interrupt.

Test: This light is on when either the mode switch, the diagnostic control switch, or the check-control switch is not in the process position.

Load: Whenever a load of the machine-language instructions or problem program is in process, this indicator is on. It turns on after the load key has been pressed and released, and turns off when the initial PSW is successfully loaded. Refer to MDM 5-22.

OPERATOR INTERVENTION SECTION

- This section contains the controls required to intervene in the normal programmed operation.
- These controls are intermixed with customer engineering controls.

System Reset: Pressing this key resets the CPU clock and all registers and controls in the CPU and I/O units. Refer to MDM 5-15. On the release of this key the system reset latch is set causing a trap address of 0240 to be forced on the SAR lines. This control storage address is the start of the resident microdiagnostic that performs a complete basic CPU check before branching to the system reset microroutine. MDM 5-16 shows the microprogram routines involved in a system reset.

The system reset key must be pressed after turning system power on. This places the system in reset status before any other function is performed.

The system reset key should be pressed only when attached devices are not in mechanical motion (actual printing, disk seek, etc.). Otherwise, the results are unpredictable: a system reset could cause undetected errors. If there is doubt about the mechanical status of I/O devices when resetting the system with this key, operations should be terminated. The program should then be restarted at the last checkpoint.

Control Address Set (CTRL ADR SET): This switch is effective only when the CPU clock is stopped. (clock is stopped when the mode switch is in one of the single cycle positions.) When the switch is pressed, the four address switches (A, B, C, and D) are gated into the M0,M1-register. When the switch is released, the 'control address set' latch and the start latch are turned on. The start latch on causes a clock start. The 'control address set' latch prevents the normal set pulse to the M0,M1-register during the first cycle after the 'control address set' operation. The latch is reset with the first T7 pulse.

Check Reset: Pressing this button resets the check latches in the processing unit to the no-error state.

Stop: Pressing the stop key with the system in process mode causes a branch into the soft-stop routine at the end of the current machine-language instruction, and turns on the manual light.

Control Storage Load: Pressing this key causes a machine reset (MDM 5-15). Releasing the key turns on both the CSL latch and the start latch. The CSL latch on causes a trap to the control-storage load routine. The latch resets at completion of the routine. A system reset is performed in conjunction with the CSL operation. Refer to MDM 5-19, 5-20.

Store: Any core-storage or local-storage location can be altered from the console. The store switch is effective when the mode-control switch is in any of the following positions.

1. MS Data
2. AS Data
3. Mod/LS

The detailed store operation is described under the mode-switch functions.

Lamp Test: When this key is pressed, all console indicators should light. The lamp test key can be pressed at any time without affecting system operation.

Display: This button is operative only when the allow manual indicator is on. The display button displays data using the byte-0 and byte-1 display indicators. The data displayed depends on which single-cycle position of the mode switch is used.

When the mode switch is set to either the MS-data or AS-data position of the mode switch, the display button causes:

1. Address in switches A, B, C, and D to access storage.
2. Read out of the two data bytes.
3. Display of the data in the byte-0 and byte-1 indicators.

R0 displays in byte-0 and R1 in byte 1. The clock is not started for this operation.

For display operations using other positions of the mode switch, see Alter/Display Functions.

For detailed descriptions of the display operations see Console Procedures.

Note: Parity is not checked during a manual store or display operation of memory.

PR-KB Alter/Display: Pressing this key causes a printer-keyboard request. When the request is honored, the keyboard unlocks, the PR-KB alter/display indicator turns on, and the proceed light turns on. The printer-keyboard can then be used for alter/display. See also PR-KB Alter/Display under Keys and Indicators PRKB. For operation details, see PR-KB Alter/Display Facility and MDM 5-77 to 5-79.

Address/Data Storage Switches: These four switches (A,B, C, and D) are used to address a location in any storage area and can be manipulated without disrupting CPU operation (Figure 6-2). For specific functions, they specify the data to be stored in the location designated by the mode switch. Refer to Mode Switch and Console Procedures for additional information.

Start Key: Pressing the start key resets the start-interlock latch (MDM 4-1). This allows the start latch to turn on when the start key is released. The start latch on generates a clock-start signal to start the machine. The processor continues to operate in the mode selected by the setting of the mode switch.

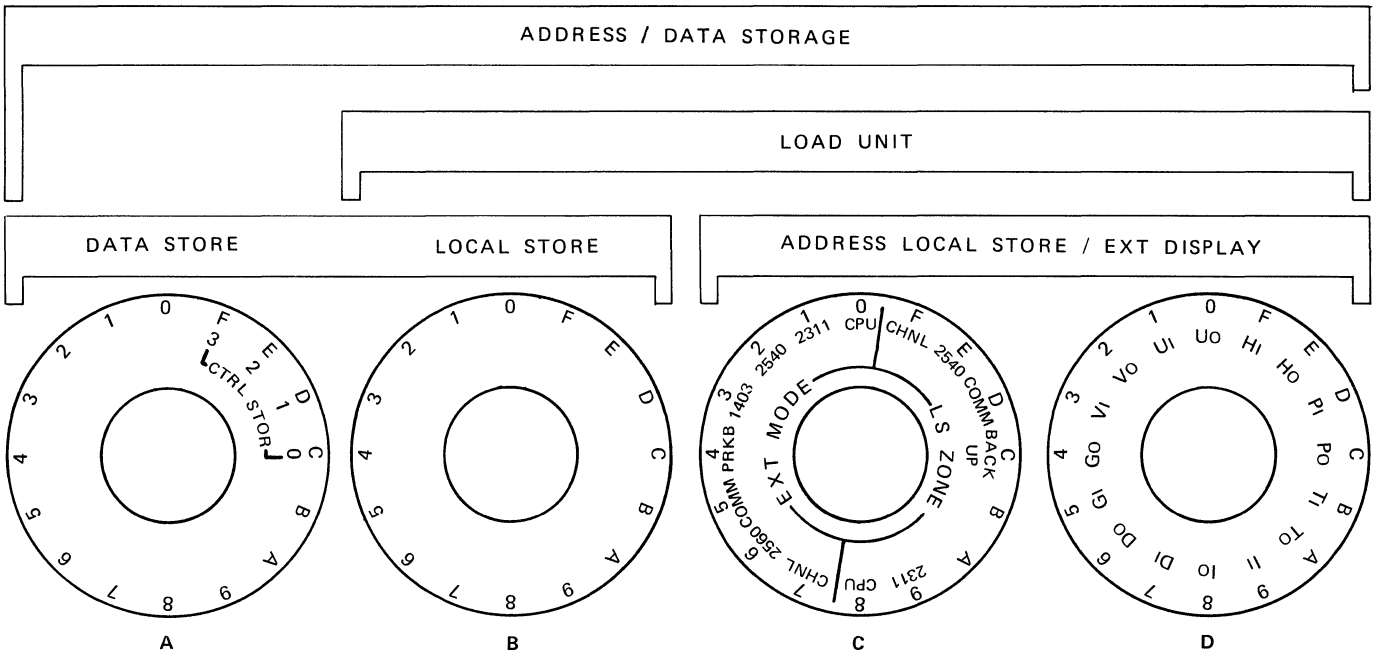
Interval Timer Switch (INT TIMER): This switch (MDM 4-91) is present only if the interval-timer feature is installed. The switch enables the interval timer. See Interval Timer in Chapter 4.

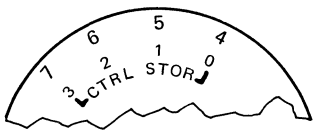
Set IC: This switch is effective when the CPU is in the soft-stop state. Pressing and releasing the switch turns on the Set IC latch. The Set-IC latch on causes a branch from the soft-stop routine and gates the contents of switches A, B, C, and D into the instruction counter. After the instruction counter has been set, the new instruction address will be printed out on the PR-KB if the PR-KB is available. At the end of the Set IC routine, the IC latch is reset and the machine goes to a soft-stop state. Refer to MDM diagram 5-14.

PSW Restart: This key is used in conjunction with the system reset key to reload the current PSW from the IPL PSW program storage locations 0000-0007). To accomplish this PSW restart, press the PSW restart key first, then press the system reset key. The program will begin execution without pressing the start key.

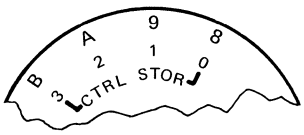
Pressing the PSW restart key sets the PSW restart latch. The system reset microprogram interrogates the PSW restart latch by testing bit 6 of the DR external facility.

When a system reset is performed without pressing the PSW restart key, the PSW is not reloaded and the current PSW remains in effect.





Switch A For 16K Systems



Switch A for 24K and 32K Systems

Note:
The full view of Switch A shows control storage positions for 48K systems, the partial views of Switch A are used for 16K, 24K, and 32K systems.

Figure 6-2. Control Panel Switches A, B, C, and D

CUSTOMER ENGINEERING CONTROLS

MODE SWITCH

- The mode of system operation is controlled by the setting of this switch.
- The positions of the switch either put the machine into an operating mode or a single-cycle mode.

Operating Mode Positions (MDM 4-4)

Process: This is the normal operating position for the system. If the clock stops with the switch in this position, byte 0 and byte 1 display the contents of the storage address register (M0,M1).

The mode switch can be switched between these first five positions without normal machine operation except as described for the switch settings.

Instruction Step (INSN STEP): In this position, the machine performs one machine-language instruction for each operation of the start button. The machine-language instruction can involve one or more microroutines. At the conclusion of the instruction, the machine branches to the soft-stop microroutine. In single-instruction operation, the console typewriter (if not busy) prints out the next instruction address after each instruction is executed. Unless a machine

check occurs, I/O overruns are not possible when the switch is in this position.

AS-Address Stop (AS ADR STOP): The machine is in the normal operating mode. When the address in switches A, C, and D compares to the address in M0 (0-3) and M1 (0-7), the machine stops at the completion of the control word in process and the address-match indicator turns on. (If the control word is a storage word, the machine stops at the end of the first cycle.) The storage address register is displayed in the byte-0 and byte-1 indicators.

SAR Delayed Stop: At this switch setting, the machine operates normally until the address in switches A,B,C,D compares to the address in the M0,M1-register. The machine completes the machine-language instruction in progress then branches to a soft-stop routine, similar to the single-instruction operation mentioned previously. The address-match indicator turns on.

Main Storage Address Stop (MS ADR Stop): The machine is in the normal operating mode. When the address of switches A,B,C,D compares to the address in M0,M1, the machine stops at the completion of the control word in process and the address-match indicator turns on. (If the control word is a storage word, the machine stops after the first cycle.) The storage address register is displayed in the byte-0 and byte-1 indicators. The match occurs for addresses in either program or control storage.

Note: An address match can occur in the checksum microroutine following operation of the system reset or IPL keys.

Single-Cycle Mode Positions

When the mode switch is in any of the remaining positions, pressing the start key causes the CPU to operate for only one cycle (overruns can occur for operations involving non-buffered devices). During this cycle, the CPU performs the function specified by the control word being executed. At the end of the cycle, the display in byte 0 and byte 1 is the storage address register, M0 and M1, except for the MS-data and AS-data positions. The data displayed in byyes 0 and 1 in the AS-data or the MS-data position is the output of core storage.

Note: When single cycling through the soft-stop loop in the BSWS routine, looping does not occur because the start key resets the soft-stop latch. To block this reset and allow only normal exits from the soft-stop loop, place the diagnostic control switch in the stop position.

Alter/Display Functions

The store pushbutton is effective only when the mode switch is in one of the following positions.

1. MS Data
2. AS Data
3. Mod/LS

The display pushbutton is effective only when the mode switch is in one of the following single-cycle positions.

1. W
2. B/A
3. C
4. ALU/EXT
5. Mod/LS
6. AS Data
7. MS Data

All of the preceding single-cycle positions except AS-data and MS-data display the M0,M1-registers in the byte-0 and byte-1 indicators unless the display switch is pressed. For displaying items 1 through 5 in the foregoing list, refer to Figure 6-3. For displaying MS- or AS-data, see Display in the Operator Intervention Section. For additional information, see Console Procedures.

Mode Switch Position	Information Displayed in Byte 0 and Byte 1 (1)	Operations Possible in Switch Position
W	W0, W1	Single Cycle or Display
C	C0, C1	Single Cycle or Display
B/A	B-Register, A-Register	Single Cycle or Display
MOD/LS	Modifier/Local Storage (2)	Single Cycle, Store (3) or Display
ALU/EXT	ALU/External Location (4)	Single Cycle or Display

(1) The indicated information is displayed only while the display key is held pressed. At all other times, the display is of the storage address register (M0, M1)

(2) Byte 1 displays the local-storage location addressed by switches C and D. Byte 0 displays the modifier.

(3) The alter/display local storage operation is explained under Console Procedures.

(4) The external location to be displayed is addressed by switches C and D and displayed in the byte-1 indicators.

The ALU output is displayed in the byte-0 indicators.

Figure 6-3. Mode Switch, Single-Cycle Positions

CHECK CONTROL SWITCH

This three-position rotary switch provides a stop-on-error function for the CPU. The switch positions and the action taken are as follows.

Position	Action
Process	Normal operating position
Disable	All machine checks are ignored, but the appropriate check latch is set.
Stop	A machine check, detected while the switch is in this position, causes a hard stop (clock stop) at the completion of the word in progress. If the check occurs during the first cycle of a storage word, the machine stops at the end of the first cycle.

DIAGNOSTIC CONTROL SWITCH (FIGURE 6-1)

Process: This is the normal position when the machine is operating, or when no scan/diagnostic functions are being performed. When the scan switch is taken out of the process position, the system is put into a test status.

Scan and Load Positions

The hardware scan and load operations are started by pressing SYSTEM RESET, then START. This starts the scan or load operation at address 0000. The scan or load continues until:

1. the system-reset key is pressed,
2. the mode switch is moved to AS ADR STOP or MS ADR STOP (stop on address match), or
3. the diagnostic control switch is moved to a different position.

If the scan or load operation is to be started at an address other than 0000, switches A, B, C, and D are set to the desired address. Pressing CONTROL ADDRESS SET places the address into the M0,M1-register. Pressing the start key initiates a clock start and the operation continues until terminated as indicated in the preceding.

Note: It may be confusing when scanning and observing auxiliary storage locations if the diagnostic switch is on SCAN, LOAD STORAGE, or TEST PATTERN. The second hexadecimal digit of the auxiliary storage address (0X00) is ignored when auxiliary storage is addressed. In scan mode, the storage address is advanced by +2 on each cycle. As a result, addresses 0000, 0100, ..., and 0F00 will access the same auxiliary storage location, 0000. If an AS ADDR STOP is done on any auxiliary storage address, the machine will stop sixteen times because each location is accessed sixteen times.

If the scan or load operation is to be started at an address other than 0000, switches A, B, C, and D are set to the desired address. Pressing CONTRCL ADDRESS

SET places the address into the M0, M1-register. Pressing the start key initiates a clock start, and the operation continues until terminated as in the preceding.

Scan Storage (SCAN STOR): Every position of core storage is read and regenerated in sequence. A parity error on storage data or a storage address causes a hard stop. The check-control switch must be in the stop position to cause a hard stop.

Note: Read call is blocked for the CSL of control storage except when using a single-cycle switch position.

This switch position is also useful in displaying sequential storage locations when used with the mode switch in either the MS DATA or AS DATA position. With the mode switch in one of these positions, byte-0 and byte 1 indicators display the addressed location each time the start key is pressed.

Single Address Positions (SINGLE ADR AS or MS): Initiated by pressing CONTROL ADDRESS SET and then the start key (must be in single-cycle mode). This sets the switches into M0 and M1. A reset to the 'control address set' latch is prevented in the single-address position. This, in turn, prevents a set to the M-register from the W-register. Thus, storage is accessed at the same address each time. The single-address function is a storage readout and regeneration operation. The appropriate single-address position must be used for the storage being accessed, AS for auxiliary storage and MS for program storage.

Load Program Storage (LOAD PRGM STOR): The data in switches A, B, C, and D is loaded into every position of program storage. When the control storage area of storage is reached, the locations are accessed in sequence, but the store lines are activated only in process mode. A storage scan should follow the 'load program storage' to determine if the data is stored properly.

Test Pattern: The purpose of this switch position is to exercise core storage with one of the following test-pattern combinations set from switches A, B, C, and D: FF00, 00FF, 01FE, or FE01. After a preliminary scan on the existing data, the test pattern is loaded from the switches. Switches C and D are stored in the first four bytes of storage, and switches A and B are stored in the next four bytes. This pattern progresses through the first 256 bytes of storage, after which the pattern is reversed (if hex address 0000 contains FFFF, address 0100 contains 0000).

Each storage address is accessed four times before a +2 address update occurs. During these accesses, the readout, complement and stor, readout, and recomplement and store sequence is performed as follows:

1st cycle	Readout	FEFE
2nd cycle	Store	0101
3rd cycle	Readout	0101
4th cycle	Store	FEFE

This process continues for each program, control, and auxiliary storage location except the resident CSL area of control storage, which is protected. To correct a storage data check, the load storage switch should be used.

Load Storage (LOAD STOR): Similar to 'load program storage' except that program, control, and auxiliary storage are loaded from the switches. The CSL area is not protected in single-cycle mode.

Stop: Stops the clock on a signal wired on the backpanel by the CE (MDM 4-11). If nothing is wired, the clock stops every cycle.

Trap: Used to force a trap to the CE Trap area of control storage. With the diagnostic control switch in the trap position and either (1) no wiring to the CE Panel, or (2) a positive pulse to the IN hub of the CE Panel, the machine-check latch is set and a trap is taken to 0280. If priority is not established in the trap routine, the trap to 0280 is repeated every other cycle. The CE trap routine can be used to log information on the printer-keyboard.

METERING SWITCH

- The metering switch enables one use meter and disables the other one.
- The metering switch is operated by the CE key.
- Two positions of the metering switch are:
  1. Normal -- Enable process meter, disable CE meter.
  2. CE -- Disable process meter, enable CE meter.

The 2025 console is provided with two direct-reading meter counters that record operating time: a customer meter and a customer engineer meter. The position of a key switch determines whether the customer meter or the CE meter is operating. The Customer Engineer holds the key for this switch, and whenever he is performing either scheduled or unscheduled maintenance in the CPU, he sets the switch to cause the

CE meter to operate. One of these meters (determined by key switch setting) operates whenever:

1. The CPU clock is running and the CPU is not in the wait state, or not in the soft stop loop.
2. The metering-in signal is up on an I/O channel.
3. Any file is selected.

The last two conditions will not cause the meter to run if the hard-stop latch is set. The meter, after being started, is forced to operate for a minimum of 400 milliseconds.

The system indicator is on whenever either meter is running.

CONSOLE PROCEDURES

Manual Display of Core Storage

The display of any storage location can be performed from the console. MDM 4-5 and the following describe the various display functions possible.

1. A display following a display. The address of the storage location to be displayed is set up in switches A, B, C, and D. Pressing the display switch gates the contents of the address switches A, B, C, and D directly through the M0,M1 assembler under manual control, and also provides a set pulse to the M-register. Pressing the display switch also resets the display storage interlock latch. Releasing the display switch turns on the display storage latch. This generates a manual read call to storage. The read-call to storage is turned off by the returning read-echo signal. Display gates are up during the complete operation and gate R0 (byte 0 of the addressed location) to the byte-0 indicators and R1 to the byte-1 indicators.

2. A display following a machine cycle. Under this condition, the address in the M-register must be transferred to the W-register before a storage location can be accessed. At the time the display key is pressed, signals are generated to set the W-register with the address in the M-register. As the display key is released, the display operation takes place as described under item 1 preceding. Note that the first machine cycle after a manual display is a no-op because the control register is reset during this first cycle.

Manual Store Operation (Core Storage)

This procedure is used to alter the contents of a particular storage location. To be effective, the store operation must be preceded by a display of that location. MDM 4-7 is a timing chart for the store operation. Displaying the storage location to be accessed leaves the manual inhibit latch on. Switches A, B, C, and D can be changed to the two bytes of data to be stored at the displayed location. If the address is in control storage, the CE Key switch must be set for CE mode during the store operation.

When the store switch is pressed, the data in switches A and B is gated through the external and AB-assemblers, and is set into the A- and B-register. Also, the alter storage interlock latch is turned off.

At this time, the manual store-0 line and a manual read-call are generated and sent to storage, and the B-register is stored. The read-echo signal returning from storage turns off the manual-inhibit latch, which turns off the read-call and store-0 lines.

The byte of data from the B-register is stored in R0 (byte 0 of the addressed location), and the data on the R1 lines (byte 1) is regenerated.

When the store switch is released, the data in switches C and D is gated through the external and AB-assemblers and is set into the A- and B-register. Then, the alter storage latch, manual-inhibit latch, and inhibit manual alter latch are turned on. A manual read-call and the manual store-1 line are generated with the alter storage latch. A read-echo from storage turns off the alter storage latch, which turns off the manual read-call and the manual store-1 line. The A-register is stored using R1 lines (byte 1) and regeneration is allowed on the R0 lines (byte 0).

The inhibit manual alter latch (left on at the end of the store operation) is used for interlocking so that a store operation cannot be followed by another store.

The manual-inhibit latch (left on after a store or a display) prevents storing the M-register into the W-register.

If the data in either byte location is to remain unchanged, the switches must still be set to the data previously displayed for that byte.

Alter/Display Local Storage

To perform either of these operations, the mode control switch must be set to the MOD/LS position.

Any position in local storage can be displayed and/or altered. The address to be accessed is selected by switches C and D. Switches A and B supply the byte of data for an 'alter local store' operation. The multiple labeling on the knob of switch C is to enable selection of main storage addresses, local storage zones, or the external-mode address for the various manual operations. Multiple labeling on the knob of switch D is for selection of bytes U0, U1, V0, V1, etc.

The switch-C decode (bits 1, 2, and 3) supplies three bits for zone selection that are the equivalent of mode register bits 5, 6, and 7. These are gated into the mode register bit circuits. The control-register bits are degated during a manual operation.

Switch D, bits 0, 1, 2, and 3 supply the bits that are decoded into the 0-F lines.

The mode register remains unchanged during an alter or display local storage, and normal machine operation can continue.

ALU/External Display

The data displayed in byte 0 and byte 1 is the M0 and M1 registers whenever the mode switch is turned to this position, or when single-cycling the machine in this mode.

During the operation of the display switch, the ALU output is gated through the LS-assembler and is displayed in the byte-0 indicators. Note that the controls specified by the control word are active. Byte 1 indicators display the contents of the address specified by switches C and D.

Switch-C bits 1, 2, and 3 correspond to the information normally set up in the mode-zone register in positions 2, 3, and 4. Switch D bits 0-3 are decoded to the 0-F addressing within any mode.

The store switch is ineffective in this mode position. Upon release of the display switch, byte-0 and byte-1 indicators will again display the M0- and M1-register.

The mode register remains unchanged during this display, and normal machine operation can continue.

CONSOLE PRINTER-KEYBOARD

COMPREHENSIVE INTRODUCTION

- Required feature, integrated I/O device on integrated interface.
- Provides facilities for: program-controlled operations, alter display, instruction step typeout, and logout.

The Console Printer-Keyboard (PR-KB) is a required feature of the Model 25. It is controlled through the integrated PR-KB attachment and associated PR-KB microprograms (Figure 6-4). Although the device is not on the standard interface of channel 0, this channel is used in the address portion of the I/O instruction.

Standard 360 I/O instructions are used; i.e., Start I/O, Halt I/O, and Test I/O. CCW commands are Read, Write, Write with ACR (automatic carrier return), No-Op, Sense and TIC. Status and sense information is stored for the program-controlled operations when applicable.

Instruction-step typeout is a manual operation associated with the instruction step position of the mode switch on the CPU console. The PR-KB types out the address of the next instruction after each instruction is executed.

The logout function of the PR-KB provides a means of typing out the diagnostic logout area of program storage (hexadecimal addresses 80-85).

For a basic introduction to the PR-KB, refer to IBM System/360 Model 25 Functional Characteristics, Form A24-3510. Use the data flow diagram (MDM 3-6) to support the following text.

INPUT DATA CODE

- Data is sent from the keyboard to the CPU using an 8-bit code consisting of 6 BCD bits plus 2 bits for upper or lower case.
- For CPU use, keyboard code characters are translated to EBCDIC by a PR-KB microprogram routine.

An 8-bit code is used to send keyboard data to the 2025. The keyboard generates the standard BCD code. The BCD code consists of 6 bits plus parity: B, A, 8, 4, 2, 1, and C. To make up the 8-bit code that is gated to the I/O CPU bus-in, the 6 BCD bits are used as the 6 low-order bits. The two high-order bits are 11 for uppercase or 00 for lowercase. The C-bit

is sent separately to the parity circuits in the CPU. Figure 6-5 shows both the uppercase and lowercase designations of the keyboard characters that are translated to EBCDIC by a PR-KB microprogram routine. The figure also shows the placement of the KB character bits on the CPU bus-in.

The keyboard characters that are translated to EBCDIC for use in CPU storage are:

26 Uppercase alpha: A-Z
26 Lowercase alpha: a-z
10 Numerics: 0-9
26 Special Characters (See chart)
1 Space or blank
1 Control code: New Line (carriage return and line feed)

The EOB (End of Block) and cancel keys do not have a special code. They have the same keyboard code as 5 and 0, respectively. The alternate coding key is tested to determine these two functions.

OUTPUT DATA CODE--EBCDIC

- Data is sent from the CPU to the PR-KB data register in EBCDIC.

The EBCDIC code of the CPU is used to send data to the PR-KB data register. However, the EBCDIC code is later translated to the tilt/rotate printing code by the PR-KB logic circuits.

During an input operation, keyboard characters are translated to EBCDIC before they are stored in memory and before they are returned to the PR-KB data register for printing. During an output operation, the EBCDIC characters in CPU storage are sent directly to the PR-KB data register. Figure 6-6 shows the EBCDIC output characters.

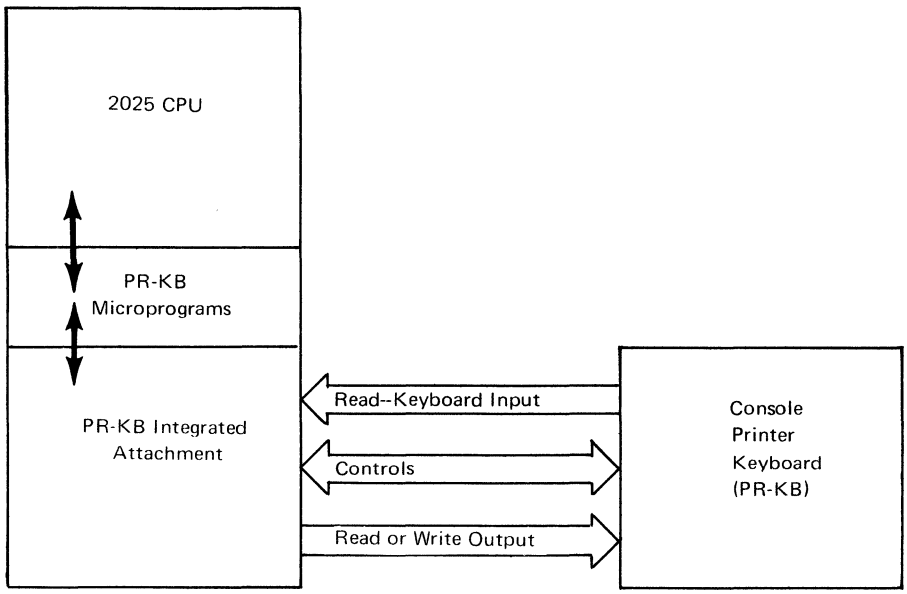


Figure 6-4. Console Printer-Keyboard Attachment

BCD Bits		A	B	BA		A	B	BA
	Space	@	-	&	Space	¢	—	+
1	1	/	j	a	=	?	J	A
2	2	s	k	b	⟨	S	K	B
2 1	3	t	l	c	;	T	L	C
4	4	u	m	d	:	U	M	D
4 1	EOB 5 Note 2	v	n	e	%	V	N	E
4 2	6	w	o	f	'	W	O	F
4 2 1	7	x	p	g	⟩	X	P	G
8	8	y	q	h	*	Y	Q	H
8 1	9	z	r	i	(	Z	R	I
8 2	Cancel 0 Note 2				)			
8 2 1	#	'	$	.	"	|	!	¬
8 4								
8 4 1			NEW LINE				NEW LINE	
8 4 2	UPPER CASE SHIFT			LOWER CASE SHIFT	UPPER CASE SHIFT			LOWER CASE SHIFT
8 4 2 1								

LOWER CASE UPPER CASE

Note 2: Alternate coding key must be pressed for EOB and Cancel

The keyboard code characters are sent to the CPU in 8-bit form as follows.

Bus-In Bits	KB Bits	
0	x	See Note 1
1	x	
2	B	
3	A	
4	8	
5	4	
6	2	
7	1	

Note 1:

x = 1, x = 1 — Uppercase

x = 0, x = 0 — Lowercase

Figure 6-5. Input Code from Keyboard

EBCDIC Bits 0123:

Bits 4567:	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	sp	&	–	–	sp	&	–	–	@	&	–	0	%	>	?	0
0001	sp	n	/	8	sp	n	/	8	a	j	/	1	A	J	_	1
0010	sp	o	b	,	sp	o	b	,	b	k	s	2	B	K	S	2
0011	sp	o	s	#	sp	o	s	#	c	l	t	3	C	L	T	3
0100	sp	2	@	@	sp	2	@	@	d	m	u	4	D	M	U	4
0101	sp	NL	/	8	sp	5	/	8	e	n	v	5	E	N	V	5
0110	sp	6	k	#	sp	6	k	#	f	o	w	6	F	O	W	6
0111	sp	6	s	#	sp	6	s	#	g	p	x	7	G	P	X	7
1000	h	q	y	y	H	Q	Y	Y	h	q	y	8	H	Q	Y	8
1001	i	r	z	9	i	r	z	9	i	r	z	9	I	R	Z	9
1010	.	$	.	,	¢	!	¢	:	@	$	–	0	%	>	?	0
1011	.	$	,	#	.	$	,	#	.	$	,	#	¢	!	:	#
1100	3	7	@	@	<	*	%	@	e	n	u	4	E	N	U	4
1101	1	5	/	8	(	)	_	'	e	n	v	5	E	N	V	5
1110	2	6	&	#	+	;	>	=	e	p	u	6	E	P	U	6
1111	0	4	–	9	|	¬	?	"	e	n	v	5	E	N	V	5

Note: Blocked areas show normal decoded Characters using 1052 keyboard Input

Figure 6-6. EBCDIC Output with Resulting PR-KB Graphics

TILT/ROTATE PRINTING CODE

- EBCDIC is translated to the Tilt/Rotate code for printing.

The Console Printer-Keyboard uses a printing mechanism that is similar to the IBM SELECTRIC® typewriter. Two tilt magnets (T1 and T2) and four rotate magnets (R1, R2 R2A, and R5) are used to position the print head. Various combinations of the magnets are energized to select characters for printing. A tilt/rotate translator in the PR-KB attachment converts the EBCDIC character to the tilt/rotate code used by the printer. Figure 6-7 shows the standard characters, EBCDIC code, and tilt/rotate code.

TILT/ROTATE CODE

Char	EBCDIC Hex	T1	T2	R5	R2A	R2	R1	Upper Case
¢	4A	0	0	0	0	0	0	1
.	4B	0	0	0	0	0	0	0
<	4C	1	1	1	1	1	0	1
(	4D	1	1	1	1	1	1	1
+	4E	1	1	0	1	1	0	1
|	4F	1	1	0	1	1	1	1
&	50	1	0	0	1	1	1	0
!	5A	1	0	0	0	0	0	1
$	5B	1	0	0	0	0	0	0
*	5C	1	1	1	1	0	0	1
)	5D	1	1	1	1	0	1	1
;	5E	1	1	0	1	0	0	1
¬	5F	1	1	0	1	0	1	1
-	60	0	1	0	1	1	1	0
/	61	0	1	1	1	1	1	0
,	6B	0	1	0	0	0	0	0
%	6C	0	0	0	1	1	1	1
_	6D	0	1	1	1	1	1	1
>	6E	1	0	0	1	1	1	1
?	6F	0	1	0	1	1	1	1
:	7A	0	1	0	0	0	0	1
#	7B	1	1	0	0	0	0	0
@	7C	0	0	0	1	1	1	0
▼	7D	1	1	1	0	0	1	1
=	7E	1	1	0	0	0	0	1
"	7F	1	1	0	0	0	1	1
a	81	0	0	1	1	1	1	0
b	82	0	0	0	1	1	0	0
c	83	0	0	1	1	1	0	0
d	84	0	0	0	1	0	1	0
e	85	0	0	1	1	0	1	0
f	86	0	0	0	1	0	0	0
g	87	0	0	1	1	0	0	0
h	88	0	0	1	0	0	1	0
i	89	0	0	0	0	0	1	0
j	91	1	0	1	1	1	1	1
k	92	1	0	0	1	1	0	0
l	93	1	0	1	1	1	0	0
m	94	1	0	0	1	0	1	0
n	95	1	0	1	1	0	1	0
o	96	1	0	0	1	0	0	0
p	97	1	0	1	1	0	0	0
q	98	1	0	1	0	0	1	0
r	99	1	0	0	0	0	1	0
s	A2	0	1	0	1	1	0	0
t	A3	0	1	1	1	1	0	0
u	A4	0	1	0	1	0	1	0
v	A5	0	1	1	1	0	1	0
w	A6	0	1	0	1	0	0	0
x	A7	0	1	1	1	0	0	0
y	A8	0	1	1	0	0	1	0
z	A9	0	1	0	0	0	1	0
0	F0	1	1	0	1	1	1	0
1	F1	1	1	1	1	1	1	0
2	F2	1	1	0	1	1	0	0
3	F3	1	1	1	1	1	0	0
4	F4	1	1	0	1	0	1	0
5	F5	1	1	1	1	0	1	0
6	F6	1	1	0	1	0	0	0
7	F7	1	1	1	1	0	0	0
8	F8	1	1	1	0	0	1	0
9	F9	1	1	0	0	0	1	0

Note: For alphabetic characters, only the lowercase is shown. The tilt/rotate code for an uppercase alphabetic character is the same as the corresponding lowercase character.

Figure 6-7. Tilt/Rotate Code

DATA FLOW

- MDM 3-6 shows the data flow for the console printer-keyboard.
- Controls-out are gated to the PR-KB attachment using a set/reset word with an external decode of F (TA-register).
- Branch conditions are gated to the CPU using a branch word with an external decode of F (TU-field) or E (TT-field).
- Input data from the keyboard is sent to the CPU using the TI-field (external decode A).
- Output data from the CPU is gated to the PR-KB TE data register using a move word with an external decode of F.

The external fields are described briefly in this section; however, further information appears in the Functional Units section.

Controls-Out (TA)

Microprogram controls from the CPU are gated to the TA-register in the PR-KB attachment on the external-out Interface. TA represents an external decode of F for a set/reset control word. Read and write operations are initiated and controlled through the TA external register.

Controls-In (TU and TT)

Input data and controls are sent to the CPU from the PR-KB attachment on the I/O CPU bus-in. The information on this bus is gated from several external fields. TU and TT are branch-condition fields. This means that the microprogram can use a branch word to test the condition of the latches and signal lines associated with each bit position of these fields.

Diagnostic Registers (TR and TD)

The TR and TD external fields are used for diagnostic purposes. During a read or write operation, the TR register can be used to gate the output of the tilt/rotate print translator (also function and case decode) to storage. Other conditions in the PR-KB can be gated to the CPU with the TD external field. A diagnostic routine can use these facilities to test for correct operation or to analyze malfunctions of the PR-KB and PR-KB integrated attachment.

Input Data (TI)

Input data from the keyboard is also gated to the I/O CPU bus-in. When the operator presses a character key during a read operation, the keyboard strobe bail contact initiates a PR-KB request. The microprogram recognizes this request and branches to the PR-KB request routine. The request routine uses a move/arithmetic control work with an external decode of A to gate the TI external field to local storage. If the operator presses a shift key, the uppercase store latch is set or reset to reflect the new shift (TI field bits 0 and 1). Also, the PR-KB takes a function cycle to set up the new shift. The input data from the keyboard does not go to the printer. Keyboard data is gated to the CPU by the microprogram. A translate routine translates the keyboard code to EBCDIC using a translate table in auxiliary storage. Then the microprogram gates the EBCDIC character back to the PR-KB attachment.

Output Data (TE)

Output data from the CPU is gated to the PR-KB attachment via the EXT bus-out. The EBCDIC data on this bus is stored in the TE data register when a move-arithmetic control word contains an external decode of F. During a read or write operation, the character in the data register (if not a function code) is translated, and the appropriate tilt/rotate magnets are energized to set up the print element. Then, the PR-KB cycle clutch is energized to print the character. A shift cycle occurs before the print cycle whenever a case change is detected in the data register. If the character is a function code, the appropriate function magnet is energized, i.e., space, or carriage return and/or line feed.

Share Request Control

The PR-KB request line is activated by the following.

1. Request key (attention).
2. Read or write share request.
3. Microforce (used at the end of an operation to handle I/O interrupt).
4. Ready key (used when the PR-KB goes from not-ready to ready).
5. Alter/display or logout request.

When the PR-KB request line is active, the microprogram branches to the PR-KB request routine to handle the particular condition.

FUNCTIONAL UNITS

PRINTER KEYBOARD

The mechanical principles of the PR-KB are described in FE Instruction--Maintenance Manual IBM 1052 Printer Keyboard, Form 225-3179; the mechanical principles of the

SELECTRIC I/O Printer are described in IBM SELECTRIC I/O Keyboardless Printer FETOM, Form 225-3353, and IBM SELECTRIC I/O Keyboardless Printer FEMM, Form 225-3207.

PR-KB AUXILIARY STORAGE AREAS

- Alter/Display Message Area--Auxiliary storage address 0x44 through 0x77.
- Sense Byte--0xF7.
- Unit Control Word (UCW) 0xF8.
- PR-KB Translate Table--5xB8 through 5xF5.

ALTER/DISPLAY MESSAGE AREA

The alter/display message area is located in the CPU module of auxiliary storage at address 0x44 through 0x77. This area is used by the Alter/Display (ALDP) microprogram routine. The following messages are stored in this area:

1. INVALID CHAR
2. INVALID ADDR

SENSE BYTE (PR-KB)

The PR-KB sense byte is located at auxiliary storage location 0xF7. Figure 6-8 shows the information stored in the sense byte. Whenever any bit is set in the sense byte, unit check is set in the UCW unit-status byte. The bit positions of the sense byte are defined as follows.

Bit 0 - Command Reject. This bit is set if a command not defined for the PR-KB is issued to the PR-KB.

Bit 1 - Intervention Required This bit is set only for a read or write command when:

1. The not-ready switch has been operated to place the PR-KB in a not-ready status, or
2. The forms switch indicates that the PR-KB requires forms loading.

Any command for the PR-KB other than a read or write is processed (even though either or both of the two preceding conditions exist) and intervention required is not set.

Bit 2 - not applicable to the PR-KB.

Bit 3 - Equipment Check. This bit is set when even parity is detected on a character code sent from the PR-KB to the CPU during a read command only. The even parity is detected in the CPU A-register and sets the KB check latch (TT bit 4).

Bits 4-7 - not applicable to the PR-KB.

PR-KB UNIT CONTROL WORD (UCW)

The PR-KB UCW is located at auxiliary storage location 0xF8. Figure 6-8 shows the information stored in the UCW. The byte format is the same as the UCW for other devices on channel 0 (integrated or standard interface). Some differences exist for the flags/op byte, channel-status byte, and the unit-status byte.

Flags/Op Byte

Bit 0--Chain Data (CD) Flag
Bit 1--Chain Command (CC) Flag
Bit 2--Suppress Length Indication (SLI) Flag
Bit 3--Skip Flag
Bit 4--Program Controlled Interruption (PCI) Flag.

The foregoing bits are set from the CCW flags.

Bit 5--Active Bit. This bit is set during initial selection for an I/O instruction. It remains on until channel-end status is cleared.

Bits 6-7 - Op 0 and Op 1. These bits are set from the CCW. They indicate the type of operation as follows.
01 - Write
10 - Read or Sense
11 - Write with ACR

Channel Status Byte

Bit 0 - Secondary Bit. This bit is on whenever status is queued at the I/O device. This secondary bit is set as follows.

1. At channel-end time for a read or write operation
2. When status must be cleared for not-ready to ready, attention (request key), intervention required, or an interface control check (false request).
3. The secondary bit is also set for the following if the PCI flag is set.
   a. Sense command
   b. No-op command
   c. Command reject
   d. Not ready when Start I/O is issued.

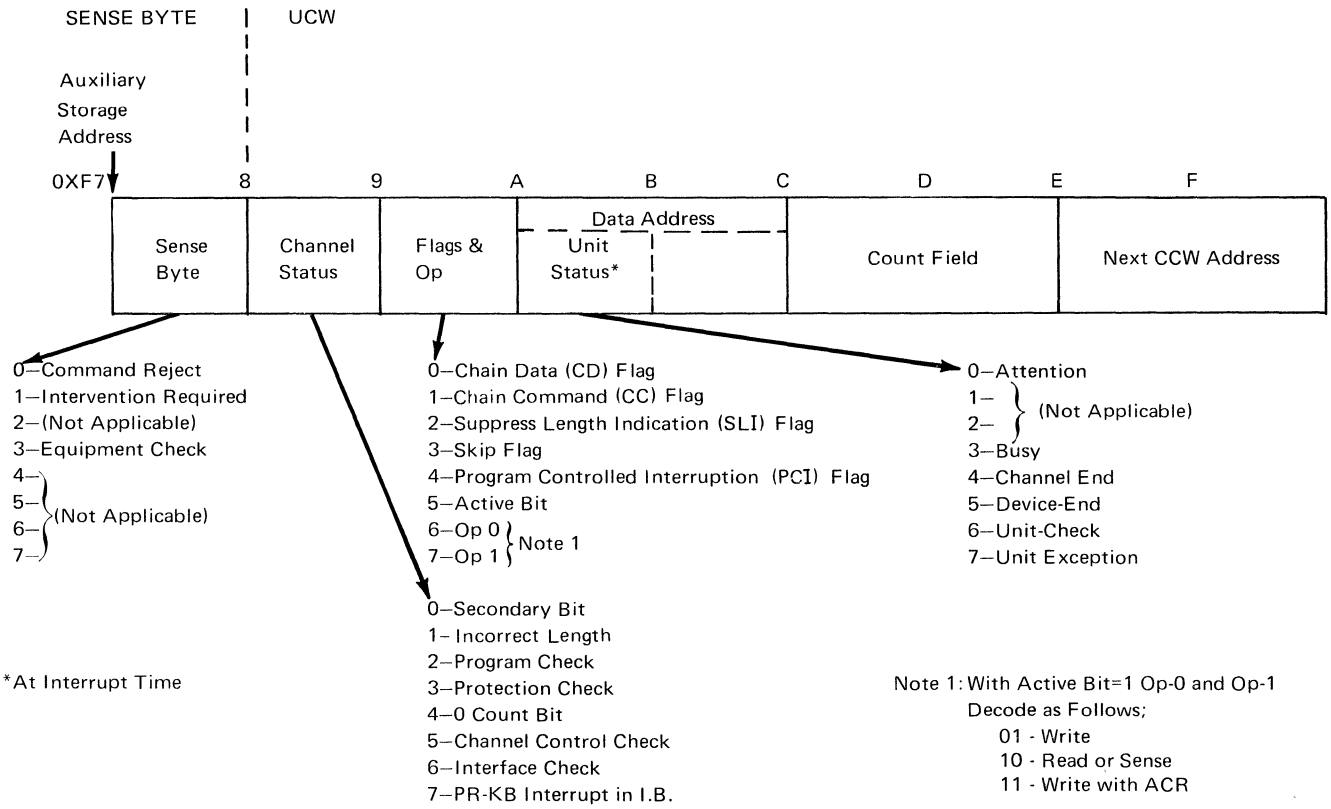


Figure 6-8. PR-KB Sense Byte and Unit Control Word

The secondary bit is not stored in the CSW. The corresponding bit in the CSW is the PCI indication.

Bit 1 - Incorrect Length (IL). This bit is set as follows.

1. IL is set for a read command during which the EOB (or cancel) key is operated when the count does not equal zero and the SLI flag is off. Any chain is terminated.
2. IL is set for a read command during which the count equals zero, and any key other than EOB or cancel is operated, and the SLI flag is off. Any chain is terminated.
3. IL is set for a read command during which an intervention required condition occurs (out of forms or not-ready switch is operated), and the count does not equal zero when either:
   a. The SLI flag is off, or
   b. The SLI flag is on and data chaining is also specified (CD flag on).
   Chaining is terminated.
4. IL is set for a write command if the SLI flag is not on. This occurs because one more data cycle is requested after the CCW data count has decremented to zero. If the CC flag is on, the operation (and chaining) is terminated. If the CD flag is on in the current write command, the incorrect-length indication does not occur for that write command. The last write command in the data chain, however, should have its SLI flag on to avoid the incorrect-length indication.
5. IL is set for a sense command if the data count specified is greater than 1 and the SLI flag is off.

Bits 2-3 - Program Check and Protection Check. Programming error detected during initial selection or a storage protect key violation during a read operation.

Bit 4 - 0-Count Bit. This bit is set when the UCW count field is decremented to zero (unless the CD flag is on). During the following request cycle, the 0-count bit causes the request routine to branch to the ending procedure. The 0-count bit is reset after the branch is accomplished. (The corresponding

bit in the CSW, channel data check, is not applicable to the PR-KB.)

Bit 5 - Channel Control Check. This bit is set as defined for channel 0.

Bit 6 - Interface Control Check. This bit is set if a false share cycle is requested. That is:
1. A not-ready-to-ready sequence has not been performed,
2. The request key has not been operated,
3. No program-controlled operation is in progress,
4. No status is outstanding for the PR-KB, and
5. No logout or alter/display operation is indicated in the attachment circuitry.

Bit 7 - PR-KB Interrupt in I.B. When the PR-KB has status ready to store in the CSW, an interrupt routine checks to see if the channel-0 interrupt buffer (I.B.) is available. If the I.B. is available, the PR-KB UCW address is placed in the I.B., and the 'PR-KB interrupt in I.B.' bit is set. This bit is reset when the status is stored in the CSW. (The corresponding bit in the CSW, chaining check, is not applicable to the PR-KB.)

Unit Status Byte

Bit 0 - Attention. This bit is set on when the request key is pressed, but only if no other PR-KB operation is in progress. If another operation is in progress, pressing the request key causes the attention-status bit to be set on after status for the other operation has been cleared at the UCW (i.e., accepted by the CPU program). If the other operation is for a logout or an alter/display operation (for which operation status is not presented), attention is not set on until the logout or alter/display is completed.

After attention status is set on:
1. If an I/O interruption for the PR-KB is processed, the CSW stored contains attention (CSW bit 32).
2. If a Start I/O is executed for the PR-KB before the I/O interruption can be processed, the CSW stored for the Start I/O contains attention (CSW bit 32) plus busy (CSW bit 35).
3. If a Test I/O is executed for the PR-KB before the I/O interruption is executed, the CSW stored for the Test I/O contains attention (CSW bit 32).
4. If a Halt I/O is executed for the PR-KB before the I/O interruption can be processed, the CSW is not stored and the condition code = 0 (interruption pending).

The preceding items 1, 2, and 3 clear the status at the PR-KB. Item 4 does not clear the status.

Bits 1-2 - not applicable to the PR-KB.

Bit 3 - Busy. This bit is set in the CSW stored as a result of execution of a Start I/O for the PR-KB only for the following conditions.
1. A program operation (other than a no-op command) has been completed to the point at which channel-end has been accepted by the CPU (an I/O interruption or Test I/O instruction has been processed to store the channel-end in a CSW) but device-end is now outstanding. Device-end (CSW bit 37) accompanies busy in the CSW for the Start I/O, and the status at the PR-KB is cleared.
2. Attention status (for a request-key operation) is outstanding for the PR-KB (i.e., the attention has not yet been cleared by an I/O interruption or Test I/O operation). Attention (CSW bit 32) accompanies the busy bit in the CSW stored for the Start I/O.
3. A device-end for a not-ready to ready sequence (the ready switch has been operated to place the PR-KB in a ready condition) is outstanding. Device-end (CSW bit 37) accompanies busy in the CSW stored for the Start I/O.
4. A program operation has been completed to the point at which channel-end has been accepted by the CPU (an I/O interruption or Test I/O instruction has been processed to store the channel-end in the CSW) but device-end is not yet available. The busy bit alone is presented in the CSW for the Start I/O, and the PR-KB status is not affected.

Busy is in a CSW stored as a result of execution of a Test I/O instruction only if the Test I/O for the PR-KB is executed after channel-end for a command has occurred and stored in the CSW, but before device-end for that same command has been set on.

Bit 4 - Channel End. This bit is set for any of the following conditions.
1. A zero data count has occurred for a write, write-with-ACR, read, or sense command. (For write, 'write with ACR', or read, channel-end is

set on during the share cycle after the one in which the zero data count is detected.)
2. At initial selection during execution of a no-op command when that command is accepted by the attachment.
3. The EOB key or the cancel key has been operated during a read instruction.
4. If a count greater than 1 is specified in a sense command, the operation is terminated after one byte is transferred.

If channel-end alone is available through channel-0 IB or has been stacked, it is cleared by an I/O interruption (or by a Test I/O) and stored in the CSW.

Channel-end and device-end are indicated in the CSW stored as a result of a Start I/O that initiates a no-op when command chaining is not specified.

Bit 5 - Device End. This bit is set for any of the following.
1. After a carrier return is mechanically started at the PR-KB by the control unit for a terminated read or write with ACR command.
2. On the share cycle following the one in which a zero data count condition occurs for a write (with no ACR) command.
3. When the attachment accepts a no-op command.
4. When the ready switch is operated to put the PR-KB in a ready condition.
5. During the share cycle in which a sense byte is sent to the CPU.

If a device-end has been generated or stacked, it is cleared during initial selection routines for a Start I/O if channel-end for the operation has already been stored in the CSW by an I/O interruption or Test I/O operation. Busy accompanies device-end in the CSW stored for the Start I/O.

Test I/O clears any outstanding device-end. A Halt I/O does not clear a device-end.

Bit 6 - Unit Check. This bit is set for any of the following reasons.
1. When a character with even parity is sent from the keyboard to the CPU during a read command operation, equipment check (sense bit 3) is also set on for this condition.
2. When the forms switch indicates that the PR-KB is out of paper or is in a not-ready condition (i.e., intervention light on the console is on), but then only:
   a. during a read or write (with or without ACR) command operation, or
   b. at initial selection for a read or write (with or without ACR) command, or
   c. during execution of a Test I/O instruction to the PR-KB.
   (Intervention required, sense bit 1, is also set on for this condition.)
3. If a command byte not defined for the PR-KB is sent to the attachment. (Command reject, sense bit 0, is also set for this condition.)

Bit 7 - Unit Exception. This bit is set on if the cancel key is operated, but only during a read command operation. The read operation is terminated (channel-end status is set on). If the count is not zero and the SLI flag is off for the read command, incorrect-length status (CSW bit 41) is also indicated during a subsequent I/O interruption or Test I/O operation.

PR-KB TRANSLATE TABLE (KEYBOARD CODE TO EBCDIC)

Figure 6-9 shows the PR-KB translate table. This table is located in auxiliary storage at locations 5XB8 through 5XF5. The table is used by the translate subroutine in the PR-KB microprogram routine to translate keyboard characters to EBCDIC.

TRANSLATE SUBROUTINE (KEYBOARD CODE TO EBCDIC)

- Translates characters from the keyboard to EBCDIC.
- Table lookup is used for special characters (except slash) and uppercase alphabetic characters.
- To translate lowercase alphabetic characters, B0 is exclusive ORed to the keyboard code for the character.
- For numeric characters, the keyboard code 8421 bits are preceded by hexadecimal F.

Figure 6-10 shows the PR-KB translate subroutine. This routine translates characters from the keyboard to EBCDIC. The keyboard characters are gated to the CPU bus-in using external field TI. Bits 0 and 1 of the TI field are 11 for upper case and 00 for lowercase. The translate subroutine uses the translate table in auxiliary storage to translate from keyboard code to EBCDIC. Only the EBCDIC

code for uppercase alphabetic characters and special characters is stored in the table. For lowercase characters, translation is accomplished by ORing B0 to the keyboard code for the character. Numeric characters do not use the table because the lower-order keyboard bits (8421) are the same as EBCDIC bits 4-7. To translate keyboard numeric to EBCDIC, set the high-order bits (0-3) to hexadecimal F. Special characters are handled individually. Refer to Figure 6-10 to follow the translation for any specific character.

Table Addr	Char	EBCDIC Hex	Table Addr	Char	EBCDIC Hex
50B8	Space	40	D8	—	6D
B9	=	7E	D9	J	D1
BA	<	4C	DA	K	D2
BB	;	5E	DB	L	D3
BC	:	7A	DC	M	D4
BD	%	6C	DD	N	D5
BE	'	7D	DE	O	D6
BF	>	6E	DF	P	D7
CO	*	5C	E0	Q	D8
C1	(	4D	E1	R	D9
C2	)	5D	E2	$	5B
C3	"	7F	E3	!	5A
C4	Space	00	E4	—	60
C5	Space	00	E5	New Line	15
C6	0	F0	E6	BKSP (Not used)	16
C7	#	7B	E7	Space	00
C8	¢	4A	E8	+	4E
C9	?	6F	E9	A	C1
CA	S	E2	EA	B	C2
CB	T	E3	EB	C	C3
CC	U	E4	EC	D	C4
CD	V	E5	ED	E	C5
CE	W	E6	EE	F	C6
CF	X	E7	EF	G	C7
D0	Y	E8	F0	H	C8
D1	Z	E9	F1	I	C9
D2	,	6B	F2	.	4B
D3	|	4F	F3	¬	5F
D4	@	7C	F4	&	50
D5	LF (not used)	25	F5	H. Tab (not used)	05
D6	Space	00			
D7	Space	00			

Figure 6-9. PR-KB Translate Table (Keyboard Code to EBCDIC)

TILT/ROTATE TRANSLATOR

- Translates EBCDIC to the PR-KB tilt/rotate print code.

- Output of the translator is fed to the PR-KB tilt/rotate magnets and to the TR external field (PR-KB diagnostic register).

The tilt/rotate translator (Figure 6-11) is located in the PR-KB attachment. This translator converts the EBCDIC character in the TE data register to the tilt/rotate code used by the console printer.

During an output operation, the microprogram sends the EBCDIC character to the TE data register. This is accomplished by a move/arithmetic word with an external AS-field decode of F. This control word together with PR-KB mode being set brings up the TE Gate signal, which allows the TE data register to be set.

The output of the TE data register is continually fed to the tilt/rotate translator. Therefore, when the cycle clutch is picked to take a print cycle, the output of the tilt/rotate translator is fed to the appropriate magnets.

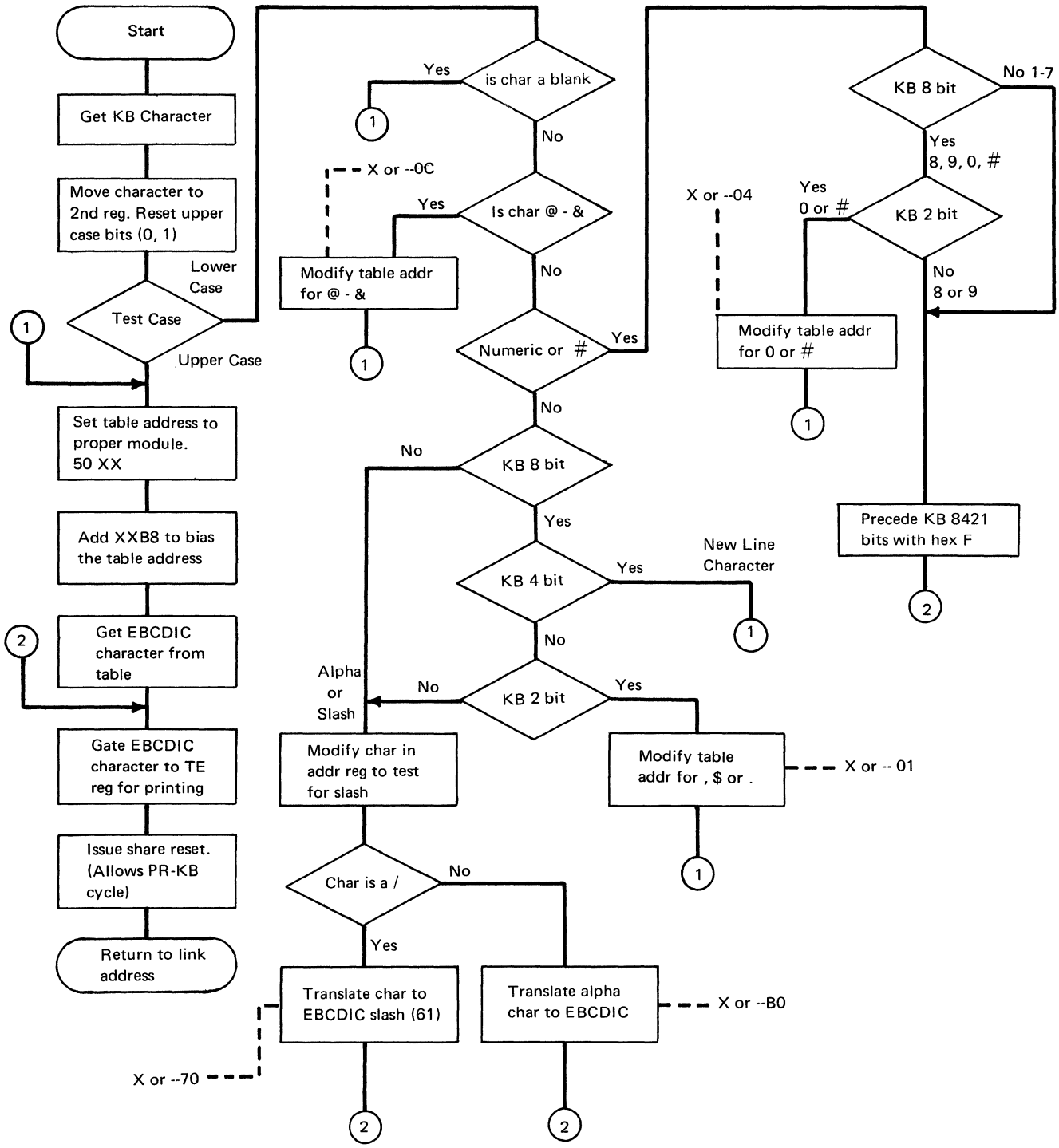


Figure 6-10. Translate Subroutine (Keyboard Code to EBCDIC)

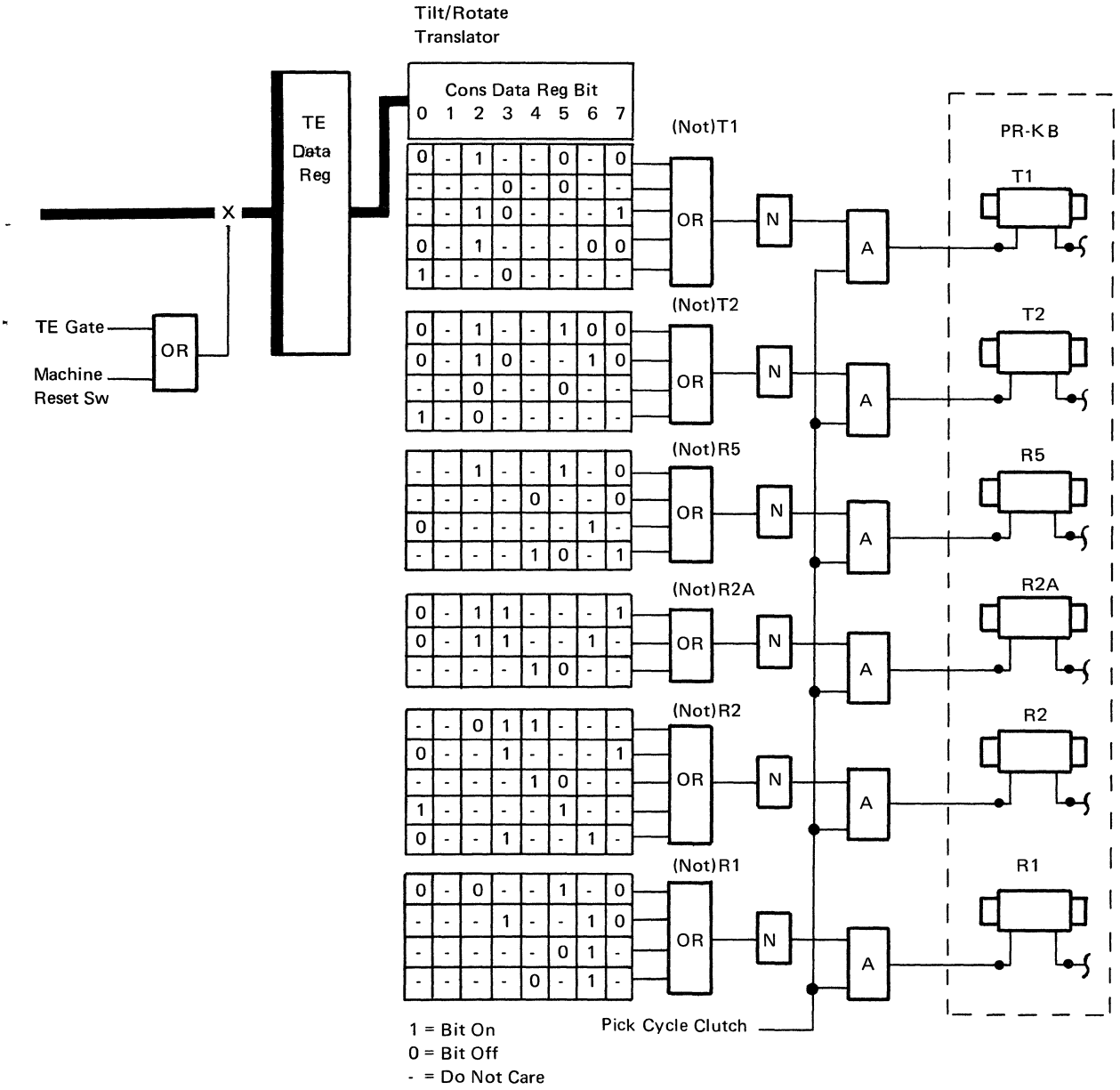


Figure 6-11. Tilt/Rotate Translator

EXTERNAL FIELDS--CPU TO PR-KB

- TA--Controls Out
- TE--Data Out

TA--Controls Out

- Microprogram initiates and controls PR-KB operations through the TA external field.

Figure 6-12 and MDM 4-71 show the TA external field. The TA external field consists of 8 bits. Each bit represents a latch (or single shot in the case of bit 5) that can be set and/or reset by a set/reset word with an external decode of F. For bits 5, 6, and 7, a different latch (or single shot) is set than is reset. The bit positions of the TA external field are defined as follows.

	Bit	
	0	Read Latch
	1	Write Latch
	2	Microforce Latch
	3	Alter-Display Active Latch
	4	(Not Used)
Set	5	TA Diagnostic Single-Shot Set
Reset		TA Attachment Reset
Set	6	Initialize Printer
Reset		Share Reset
Set	7	(Not Used)
Reset		Attention Reset

Figure 6-12. TA External Field

TA0 - Read Latch. This bit sets or resets the keyboard read latch to initiate or terminate a read operation.

TA1 - Write Latch. This bit sets or resets the console printer write latch to initiate or terminate a write operation.

TA2 - Microforce Latch. (Interrupt stacked.) This bit sets or resets the microforce latch. When status is ready for the CSW, the microprogram sets the microforce latch to force another request cycle when the channel-0 interrupt buffer is available.

TA3 - Alter Display Active Latch. This bit sets or resets the alter display active latch. This latch is set to indicate that an alter or display operation is in progress.

TA4 - Not used.

TA5 (set) - TA Diagnostic Single-Shot Set. With the set function of the set/reset word, this bit can be used in place of keyboard strobe to start the keyboard single shots. The read latch (TA0) must be set before TA5 is set.

TA5 (reset) - TA Attachment Reset. With the reset function of the set/reset word, this bit can be used to cause an attachment reset in the PR-KB without receiving a machine reset from the CPU.

TA6 (set) - Initialize Printer. With the set function of the set/reset word, this bit sets the initialize printer latch if the PR-KB is in uppercase. The purpose of the initialize printer latch is to set the PR-KB and the PR-KB attachment circuits for lowercase. If lowercase is already set, the set function of TA6 has no effect.

TA6 (reset) - Share Reset. With the reset function of the set/reset word, this bit resets the RD/WR share latch. Generally, the RD/WR share latch is reset at the same time as the read or write latch at the end of an operation, to stop the hardware-generated request cycles.

TA7 (set) - Not used.

TA7 (reset) - Attention Reset. With the reset function of the set/reset word, this bit resets the attention latch (set by request key). The microprogram issues 'attention reset' after the attention bit has been set in the unit-status register.

TE--Data Out

- Microprogram sends output data to the PR-KB attachment using external register TE (MDM 4-74).

The TE data register consists of 8 polarity hold latches to receive the 8-bit EBCDIC character on the external bus-out. The data register is set to the status of the external bus-out lines whenever 'machine reset' or 'TE gate' is activated. 'TE gate' is brought up when:

1. PR-KB Mode is set, and
2. a move/arithmetic word (type 3) with an AS-field decode of F (TE) is decoded in the control register.

This control word is issued by the microprogram when a data character is ready for printing. The output of the TE data register is fed to the tilt/rotate translator and to the function decode circuits.

EXTERNAL FIELDS--PR-KB TO CPU

- TI--Data In
- TU--Branch Conditions
- TT--Branch Conditions
- TR--Tilt/Rotate Diagnostic Register
- TD--PR-KB Diagnostic Register

TI--Data In

- The TI external field is used to send a keyboard character to the CPU.

Figure 6-13 shows the TI external field. This field consists of 8 bit-lines that are gated to the I/O CPU bus-in. The six low-order bits (BA8421) are fed directly from the keyboard bail contacts. The two high-order bits come from the uppercase store latch. These two bits are set to 11

for uppercase, 00 for lowercase. The uppercase store latch is set if the uppercase shift key (8421) is operated during a read operation. (The uppercase store latch is also set if an uppercase character is decoded during a write operation.) The TI external is gated to the CPU when:

1. PR-KB mode is set, and
2. the AS-decode in the control register is A (TI).

Bit	
0	Uppercase Store Latch*
1	Uppercase Store Latch*
2	Keyboard Bit B
3	Keyboard Bit A
4	Keyboard Bit 8
5	Keyboard Bit 4
6	Keyboard Bit 2
7	Keyboard Bit 1

* Bit 0=1, Bit 1=1 } Uppercase Bit 0=0, Bit 1=0 } Lowercase

Figure 6-13. TI External Field

TU--Branch Conditions

- The TU external field is used by the microprogram to test the status of eight latches in the PR-KB attachment.

Figure 6-14 and MDM 4-71 show the TU external field. This field consists of eight bit-lines that can be gated to the I/O CPU bus-in. Each bit-position represents the on-off status of a latch in the PR-KB attachment. The TU external field is gated to the CPU when:

1. PR-KB mode is set, and
2. the AS decode in the control register is F.

A branch on condition or branch on mask word can be used to test the various bit-positions of the TU field. The TU bit positions are defined as follows.

Bit	
0	Read Latch
1	Write Latch
2	Microforce Latch
3	Alter-Display Active Latch
4	Cycle Interlock Latch
5	Data Ready Latch
6	Initialize Printer Latch
7	Printer Busy Latch

Figure 6-14. TU External Field

TU0 - Read Latch. This bit indicates that a read operation is in progress. The read latch is set and reset by TA0.

TU1 - Write Latch. This bit indicates that a write operation is in progress. The write latch is set and reset by TA1.

TU2 - Microforce Latch. This bit indicates that an interrupt is stacked. The microprogram has set the microforce latch (with TA2) to force another request cycle when the interrupt buffer is available.

TU3 - Alter/Display Active Latch. This bit indicates that one of the following operations is in progress.

1. Alter
2. Display
3. Set IC
4. Instruction-Step Typeout
5. Logout

TU4 - Cycle Interlock Latch. This bit indicates that a PR-KB print or function cycle has been initiated by the PR-KB attachment. The cycle interlock latch resets after the mechanical cycle has started (printer busy).

TU5 - Data Ready Latch. This bit indicates that:

1. A character has been sent to the data register for a read or write operation, or
2. A change in shift has occurred and the PR-KB attachment is ready for the shift cycle, or
3. The initialize printer latch has been set by TA6. The data ready latch resets after the mechanical cycle has started (printer busy).

TU6 - Initialize Printer Latch. This bit indicates that the microprogram has set TA6 to put the PR-KB and PR-KB attachment in lowercase. A system

reset also turns on the initialize printer latch if the PR-KB is ready. The initialize printer latch is reset after the mechanical cycle has started (printer busy).

TU7 - Printer Busy Latch. This bit indicates that a mechanical cycle has started in the PR-KB (either print or function cycle). The printer busy latch remains on until the mechanical cycle is nearly complete.

TT Branch Conditions

- The TT external field is used by the microprogram to test the status of latches and signal lines in the PR-KB attachment.

Figure 6-15 shows the TT external field. This field consists of eight bit-lines that can be gated to the I/O CPU bus-in for testing by a branch word. The TT external field is gated to the CPU when:
1. PR-KB mode is set, and
2. the AS decode in the control word is E.

The TT bit positions are defined as follows.

Bit	
0	Attention Latch
1	Not Ready to Ready
2	Intervention Required
3	Alter-Display Latch
4	Keyboard Check Latch
5	Alternate Coding Key
6	PR-KB Request
7	Logout Latch

Figure 6-15. TT External Field

TT0 - Attention Latch. This bit indicates that the request key has been operated to set the attention latch. TA7 is used to reset the attention latch after the microprogram sets the attention bit in the unit-status register.

TT1 - Not-Ready to Ready. This bit indicates that the PR-KB has gone from a not-ready condition to a ready condition. The not-ready condition occurs if the not-ready key is operated or if the PR-KB runs out of forms. The PR-KB can then be made to go to the ready condition by inserting forms (if needed) and operating the ready key. The not-ready to ready condition causes a PR-KB request cycle. During this request cycle, the not-ready latch is reset by TA7.

TT2 - Intervention Required. This bit indicates that the PR-KB has run out of forms or the not-ready key has been operated.

TT3 - Alter/Display Latch. This bit indicates that the alter/display key (on the CPU console) has been operated to set the alter/display latch. This latch is reset when the microprogram sets the alter-display active latch (TA3).

TT4 - Keyboard Check Latch. This bit indicates that an A-register parity error occurred while a keyboard character (TI external field) was being gated to the CPU.

TT5 - Alternate Coding Key. This bit indicates that the alternate coding key is being operated.

TT6 - PR-KB Request. This bit indicates that the PR-KB is requesting a share cycle. A PR-KB request is generated by the following.
1. Attention-request key (TT0)
2. Read or write share request
3. Interrupt stacked (microforce latch, TA2)
4. Not-ready to ready (TT1)
5. Logout (TT7)
6. Alter/display (TT3)

TT7 - Logout Latch. This bit indicates that a logout operation has been initiated. The log-out latch is set by the microprogram using a set/reset word with an AS-field decode of 6 and K-low field of 3 (BC7).

TR--Tilt/Rotate Diagnostic Register

- The TR external field is used to send a byte of diagnostic information to the CPU. This byte reflects the print or function setup as translated (or decoded) by the PR-KB attachment.

The TR external field includes:

Bits 0-5 - The six TR bits from the tilt/rotate translator.

Bit 6 - Uppercase character indication. Translated from the TE data register output.

Bit 7 - Function cycle indication. Decoded in the TE data register or set from new-line latch.

The TR external field is gated to the CPU when:
1. PR-KB mode is set, and
2. the AS-decode in the control register is B.

TD -- PR-KB Diagnostic Register (Figure 6-16)

- The TD external field (MDM 4-75) is used by diagnostic microprograms to test the status of latches and signal lines in the PR-KB attachment.

Bit	
0	
1	
2	Read Write Share Request
3	New Line Latch
4	Key Switch CE Mode
5	Shift Cycle Latch
6	Lowercase Decode
7	Uppercase Decode

Figure 6-16. TD External Field

Bit 0 -- Not used.

Bit 1 -- Not used.

Bit 2 --Read-Write Share Request. This bit becomes active when the RD-WR share latch is turned on to request a read or write share cycle. TD bit 2 becomes inactive when the RD-WR share latch is turned off by share reset (TA6).

Bit 3 -- New Line Latch--This bit indicates that an end-of-line condition has occurred; i.e., right-hand margin switch operated. The new line latch is reset after the mechanical carriage return/line feed operation has started (printer busy).

Bit 4 - Key Switch CE Mode. This bit is on whenever the key switch on the CPU console is in the CE-mode position. When bit 4 is off, the alter/display routine protects control storage by making an alter operation invalid.

Bit 5 - Shift Cycle Latch. This bit indicates that a shift cycle has been initiated. A shift cycle is started if the shift key is operated (or released) during a read operation, or if a change in shift is decoded during a write operation. The shift cycle latch is turned off when the mechanical shift cycle is nearly over (printer not busy).

Bit 6 - Lowercase Decode. This bit indicates one of the following.
1. Read operation. A lowercase shift character is being received from the keyboard (keyboard bits B, A, 8, 4, 2). This occurs when the shift key is released.
2. Write operation. A lowercase character is being translated from the TE data register output.

Bit 7 - Uppercase Decode. This bit indicates one of the following.
1. Read operation. An uppercase shift character is being received from the keyboard (keyboard bits 8, 4, 2). This occurs when the shift key or lock key is operated.
2. Write operation. An uppercase character is being translated from the TE data register output.

CONSOLE PRINTER-KEYBOARD THEORY OF OPERATION

REQUEST KEY--ATTENTION STATUS

- Request key causes attention status to be stored in the CSW.

- If request is to be honored, the problem program (or supervisor) must test for attention status and then issue a Start I/O read command.

A read-request operation is initiated when the operator presses the request key on the console printer-keyboard. Pressing this key turns on the attention interlock latch. Releasing the key then sets the attention latch (bit 0 of the TT external field) and lights the attention indicator. Next, a PR-KB request (TT bit 6) is generated if the PR-KB is ready and not in run mode (busy with a previous read or write operation). If run mode is active, the PR-KB request is generated as soon as run mode becomes inactive. The PR-KB request activates not-S7 branch condition (not exceptional condition) and integrated I/O Request (BB Bit 1).

The control microprogram recognizes the status of S7 and branches to BSWI, the exceptional condition routine. Because BB bit 1 and TT bit 6 are set, the microprogram branches to the PR-KB request routine in DYPE. In the PR-KB request routine, the attention bit is turned on in the UCW unit-status byte. Also, the attention latch is reset (TA bit 7), and thereby the attention indicator is turned off. If the channel-0 interrupt buffer (I.B.) is available, the channel-0 interrupt latch is set (S7) to force an I/O interrupt. Also in preparation for the interrupt, the PR-KB UCW address is stored in the channel-0 interrupt buffer and the interrupt in I.B. bit is set in the channel-status byte of the UCW. Then, the microprogram returns to the BSWI routine. If the I.B. had not been available, the

microforce latch would have been set (TA bit 2) to force another PR-KB request to try again. When priority and masking permit, the microprogram branches to DCHN (channel-0 interrupt routine) to control the storing of the CSW and to initiate the interrupt (BPSW routine). Then if the attention status is to be honored, the problem program (or supervisor) must check for and recognize the attention bit, load the CAW with the address of the PR-KB read CCW, and issue a Start I/O instruction. The Start I/O instruction contains the PR-KB unit address (0XB7 in auxiliary storage).

I/O INSTRUCTIONS ROUTINE

- The DCLA routine fetches and checks CAW for Start I/O instructions.
- Microprogram branches to an appropriate routine on the basis of I/O instruction, active bit, and secondary bit.

I/O instructions are decoded in the I-cycles routine, CICY, causing a branch to the I/O instructions routine DCLA. In DCLA, the condition code is set to 0. (The condition code is changed later if the PR-KB is not available.) For Start I/O, the CAW is fetched and checked in the DCLA routine. For all PR-KB I/O instructions, the PR-KB is identified as an integrated unit on channel 0. The PR-KB UCW address (0xF8 in auxiliary storage) is derived from the I/O instruction unit address (0xB7 for the PR-KB). Local storage register G1 is used to hold the new flags/op byte. The active bit, G1 bit 5, is set on in the DCLA routine. The old active and secondary bits in the UCW ar tested, and the microprogram branches to another routine on the basis of these bits as follows.

START I/O WITH PRIOR STATUS TO CLEAR

- Prior status must be cleared when the active bit is off and the secondary bit is on at initial selection by Start I/O.
- Status is stored in the CSW with busy set in the unit-status byte.

If the active bit is off and the secondary bit is on in the PR-KB UCW when a Start I/O instruction is decoded in the DCLA routine, an I/O operation cannot be started. Instead, a status condition that has not been cleared must be stored in the CSW. To handle this condition, the microprogram branches to the DYPE routine.

If neither attention or device-end is set in the unit-status byte, the device is still busy with a previous operation. The CSW is stored with busy set (unit-status bit 3). Condition code 1 is set to indicate that a CSW was stored, then the microprogram returns to I-cycles.

If either attention or device-end is set, the status to be stored resulted from one of the following.

1. Attention status (request key)
2. Not-ready to ready
3. Read or Write with ACR device-end (channel-end status already cleared).

For any of these conditions, the 'interrupt in I.E.' bit and the channel-0 interrupt latch are reset (if on). This is done because an interrupt is no longer needed to store the CSW. The secondary bit is reset in the UCW to indicate no outstanding status. The CSW is stored with the busy bit set, then the microprogram sets condition code 1 and returns to I-cycles.

START I/O--PRINTER KEYBOARD NOT BUSY

- The DCLB routine fetches and checks the CCW for both initial selection and chained CCWs.
- Operation is decoded in the DYPE routine.

When a Start I/O instruction finds the PR-KB available, the microprogram branches to the DCLB routine. This routine fetches and checks the CCW. The CCW information and other pertinent information is stored for the next routine as follows.

Command code	*
Data Address	*
Flags/Op	*
Count	UCW
Next CCW Address	UCW
UCW Address	*

*See listing for local storage register assignment. Next, the microprogram branches to DYPE, the PR-KB routine. There the UCW is initialized. The secondary bit and the end-status indicator bit are reset in case they had been set. The end-status indicator bit is interrogated during PR-KB request cycles. When on, it indicates that final status is ready and the request was initiated to place the PR-KB UCW address in the interrupt buffer.

The DYPE routine is entered at the same point whether the CCW is being fetched for chaining or for Start I/O initial selection. If entry was for data chaining, the microprogram branches to the BSWI routine to test for the next I/O request. If the command is a new one, the microprogram proceeds to initialize for this command.

SENSE COMMAND

- The PR-KB sense byte (auxiliary storage 0xF7) is placed in the program storage location specified by the sense command data address.

- The sense command is executed even when the PR-KB is not ready.

When a sense command is decoded during Start I/C initial selection, the end-status bit is set to indicate final status. If the skip flag is not on, the sense byte is stored at the UCW data address. (If the skip flag is on, the sense byte is not stored.) The sense command for the PR-KB should have a count of one. If the count is not zero, after the count is decremented, or if the chain data flag is on, the incorrect length bit is set in the channel status byte. (Incorrect length is reset if the SLI flag is on.)

If the command chaining flag is off, the routine BALs to try for a normal interrupt. (See Common Interrupt Routine.) Upon return from the BAL the UCW is updated and the microprogram branches to BSWI before returning to I-cycles.

If the command chaining flag is on, the routine branches to the Common Interrupt Routine to try for a Program Controlled Interrupt (PCI). Then, the UCW is updated and the microprogram branches to the DCLB routine to fetch the next CCW.

INTERVENTION REQUIRED

- Intervention required (sense bit 1) is set when the PR-KB is not ready during Start I/O initial selection.

- The command is terminated and status is stored. Intervention required (sense bit 1) is set when TT bit 2 indicates that the PR-KB is not ready (not-ready key pressed or forms switch not operated). If the command is not sense, which is processed even when the PR-KB is not ready, unit check is set in the unit-status byte. Also, the current chaining flags are reset.

If the command was chained from a prior CCW, the operation is terminated by resetting the read latch, write latch, and RD-WR share latch. End status is set and the routine branches to the Common Interrupt Routine to try for a normal interrupt. Then the microprogram branches to BSWI before returning to I-cycles.

If the command was not chained from a prior CCW, the operation is terminated by turning off the active bit and storing status in the CSW immediately. Condition code 1 is set to indicate a CSW stored, then the microprogram returns to I-cycles.

NO-OPERATION COMMAND

- A no-op command sets channel-end and device-end, and stores a CSW unless the command is part of a chain.

When a no-op command is decoded, the current CC flag is tested to see if a command is chained to it (CC flag on). If the CC flag is on, the microprogram clears unit status, updates the UCW, then branches to the DCLB routine to fetch the next CCW. If no current CC flag, unit status is not reset and channel-end and device-end (set before testing the CC flag) remain on.

If the no-op command is not chained from a previous CCW, the active bit is reset and status is stored in the CSW. Condition code 1 is set and the microprogram returns to I-cycles. If the PCI flag is on, the secondary bit is set. This allows another attempt for a PCI.

If the no-op command is chained from a previous CCW, the operation is terminated by resetting the read latch, write latch, and RD-WR share latch. End status is set and the routine BALs to try for a normal interrupt. (See Common Interrupt Routine.) Upon return from the BAL, the UCW is updated. The microprogram returns to I-cycles via the BSWI routine.

READ COMMAND

- Figure 6-17 shows the major objectives of a PR-KB read operation, which is divided into two distinct parts:
  1. Initial Selection
  2. Request Cycles

Initial Selection is described here. Request Cycles are described under PR-KB Request, which describes the various types of request cycles including Read Share Request. Figure 6-17 ties the read operation together with a flowchart of the complete read operation. Figure 6-18 is a positive logic diagram of the read operation, and Figure 6-19 is a timing chart for the read operation.

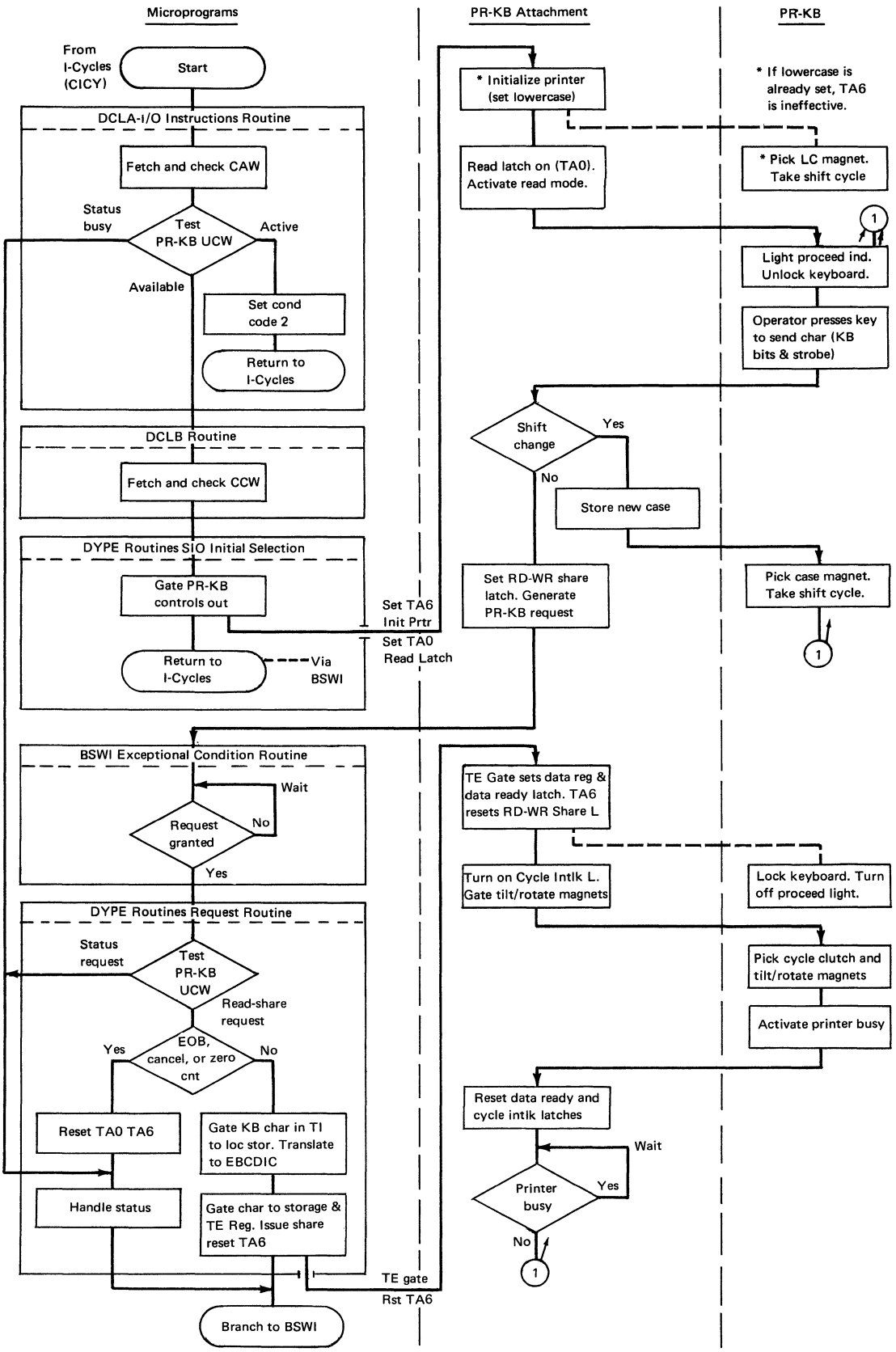


Figure 6-17. Read Operation Flowchart

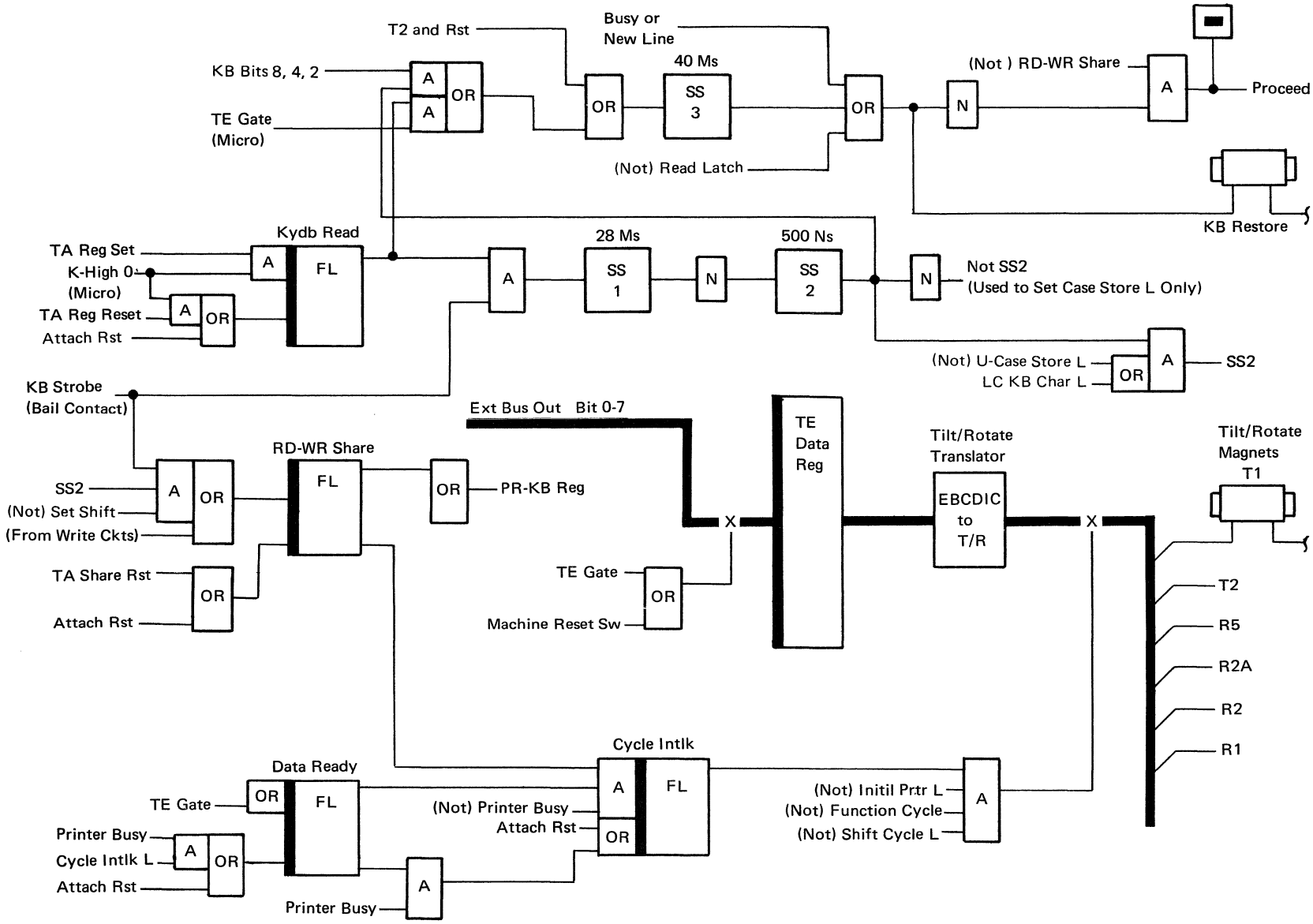


Figure 6-18. Read Controls

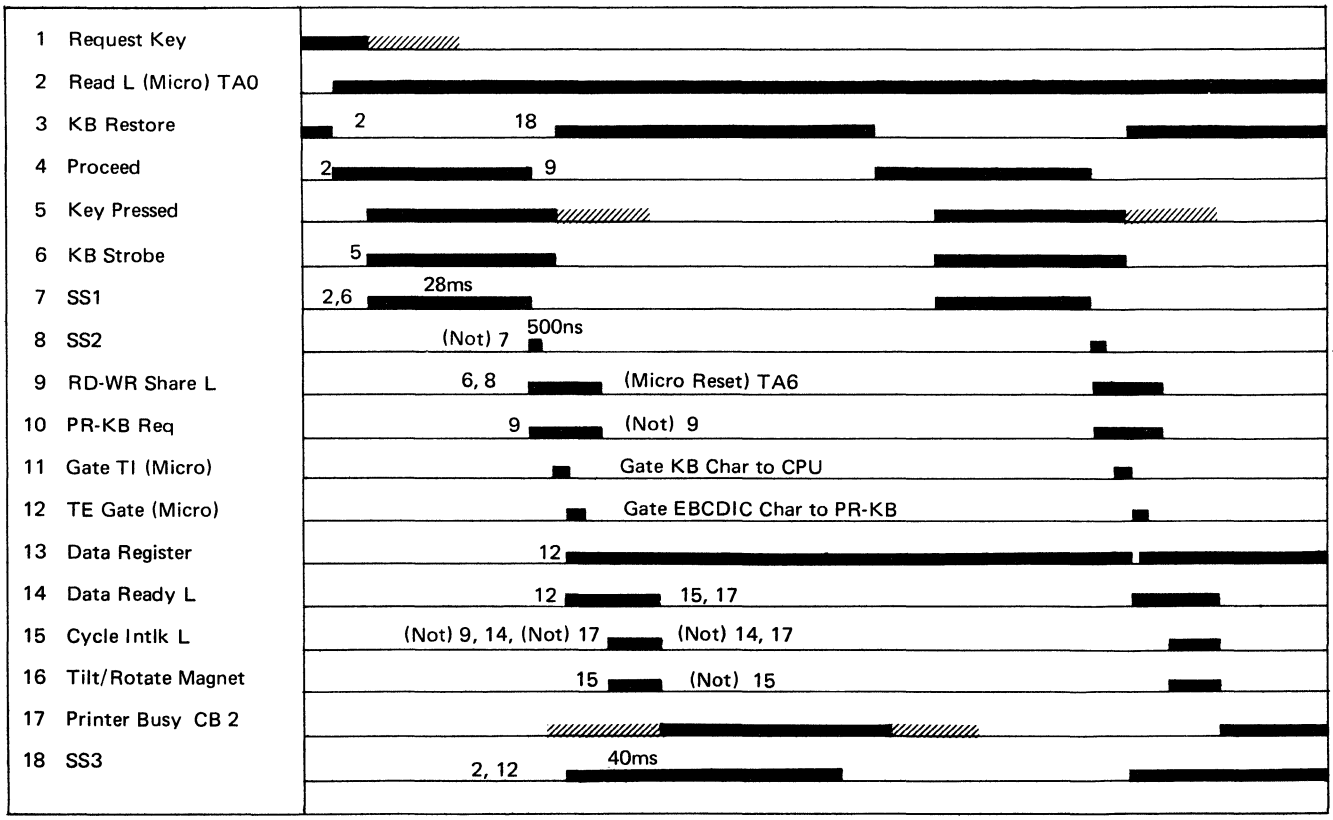


Figure 6-19. Read Timing Chart

Initial Selection--Read Command

- Microprogram sets TA bit 0, which turns on the read latch in the PR-KB attachment circuits.
- PR-KB attachment generates a PR-KB request when keyboard character is ready.
- Control microprogram recognizes request and branches to the PR-KB request routine.

The read command is decoded after the CCW has been fetched for Start I/O or command chaining. See Start I/O--Printer-Keyboard Not Busy for a description of the entry. The first step in the read operation initializes the printer using a set/reset word to set TA bit 6. If the PR-KB is already in lowercase, this step is ineffective. If the PR-KB is in uppercase, the 'initialize printer' latch is set. This latch affects the logic circuits as follows.

1. 'Data ready' latch turns on, then 'cycle interlock' latch.
2. Uppercase 'KB character' latch is reset.
3. 'Case store' latch is reset.
4. 'Shift cycle' latch is reset.
5. 'RD-WR share' is prevented for a write operation.
6. 'Cycle clutch' pick is prevented.
7. Lowercase magnet is picked.

Picking the lowercase magnet actuates the shift clutch in the PR-KB and puts the printer into lowercase. During this mechanical cycle, the shift contact (CB7) transfers. This brings up 'printer busy', which resets the 'data ready' latch, the 'initialize printer' latch, then the 'cycle interlock' latch.

Continuing with the microprogram, the read latch (TA bit 0) is turned on at the same time as the 'initialize printer' latch. After setting the read latch, the microprogram stores the updated UCW, then branches to the BSWI routine. In BSWI, CPU mode and zone are set. Also, the CPU storage protect key is restored. Then the microprogram returns to I-cycles, and continues with other routines until a 'request signal' is generated by the printer-keyboard. Recall that condition code 0 (PR-KB available) was previously set at the start of the DCLA routine.

In the PR-KB attachment, the read latch brings up read mode and run mode. These lines prevent the development of a request signal from: request key, logout, or alter/display. The request signal for the read operation is developed as follows.

1. The read latch drops the keyboard restore magnet and places it under control of single-shot 3 and the printer busy signal line.
2. The proceed indicator lights, and the operator presses a character key on the PR-KB.
3. The keyboard-bail contacts bring up 'KB strobe' and the KB bit lines for the selected character.
4. 'KB strobe' fires single-shot 1.
5. Single-shot 1 times out after 28ms and fires single-shot 2.
6. Single-shot 2 provides a 500ns pulse that ANDs with 'KB Strobe' and (not) set shift to set the 'RD-WR share' latch. This latch brings up the PR-KB request signal (TT bit 6).
7. The request signal activates '(not) S7 branch condition' and 'integrated I/O request' (BB bit 1).

The control microprogram recognizes the status of S7 and branches to BSWI, the exceptional condition routine. Because BB bit 1 and TT bit 6 are set, the microprogram branches to the PR-KB request routine. The UCW active bit=1 and the secondary bit=0 when the read operation initially enters the request routine. See Read Share Request for a description of request cycles and the continuation of the read operation. Briefly, the request routine:

1. Gates the keyboard character to local storage.
2. Translates the character to EBCDIC.
3. Sends the character back to the PR-KB for printing.

Keyboard Shift

- Upshift code from keyboard sets the uppercase KB-character latch and the U-case store latch.
- Downshift code from keyboard generates the lowercase KB-character and resets the U-case store latch.

Pressing either shift key during a read operation generates the uppercase code (842). When the shift key is released, the lowercase code (BA842) is generated. Pressing the lock key generates the uppercase code and locks the shift. The shift is unlocked by pressing either shift key. As before, the lowercase code is generated when the shift key is released.

Figure 6-20 is a timing chart for uppercase shift during a read operation. Shifting back to lowercase is similar, except that the lowercase KB character is generated and the U-case store latch is turned off.

A read share request is not generated for a shift cycle. This is blocked by the 'set shift' signal. After the new shift is set and the next character key is operated, the KB-character latch is reset and a normal read cycle is started.

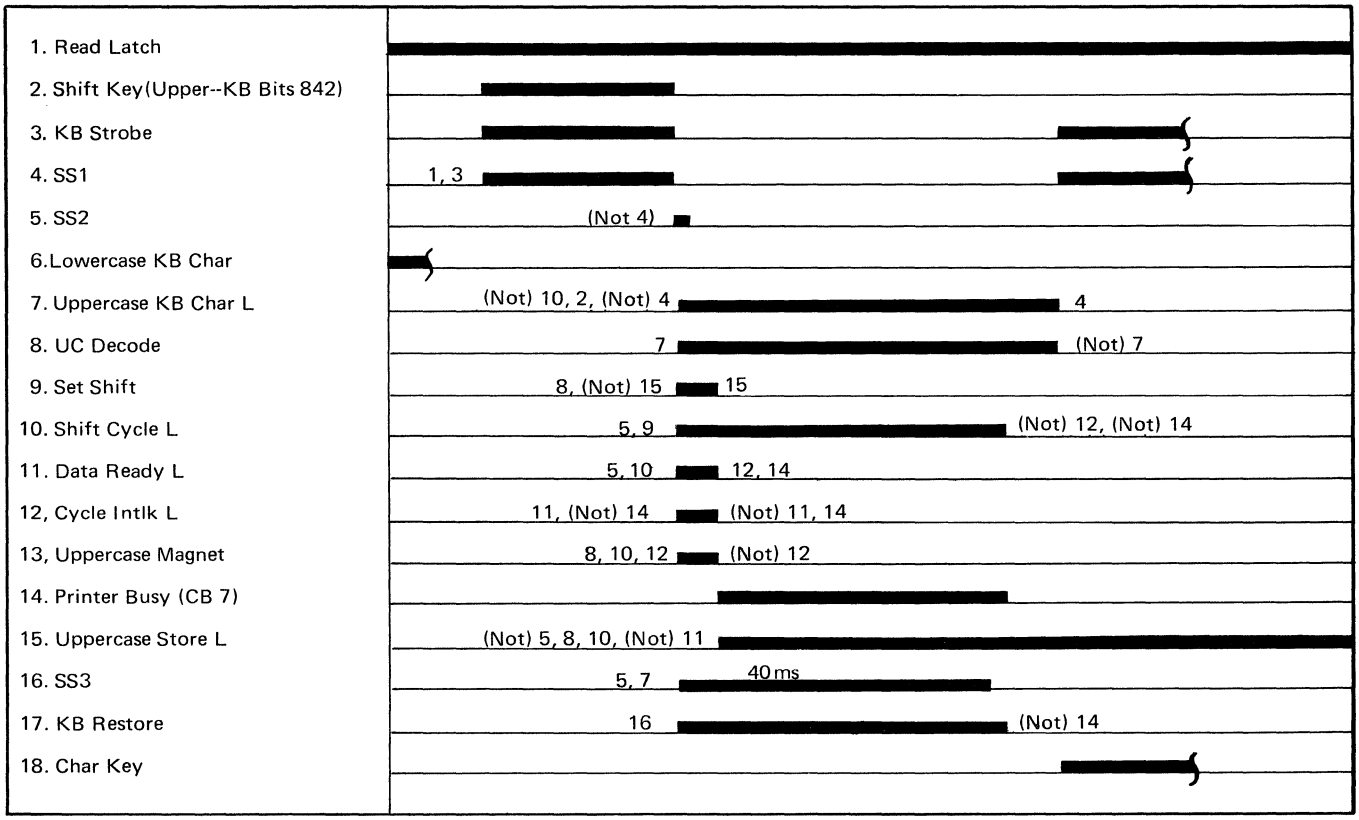


Figure 6-20. Uppercase Shift (Read Operation)

WRITE COMMAND

- Figure 6-21 shows the major objectives of a PR-KB write operation.

The PR-KB write operation is divided into two parts.

1. Initial Selection
2. Request Cycles

Initial Selection is described here. Request Cycles are described under PR-KB Request, which describes the various types of request cycles including Write Share Request. Figure 6-21 ties the write operation together with a flowchart of the complete write operation. Figure 6-22 is a positive logic diagram of the write operation, and Figure 6-23 is a timing chart for the write operation.

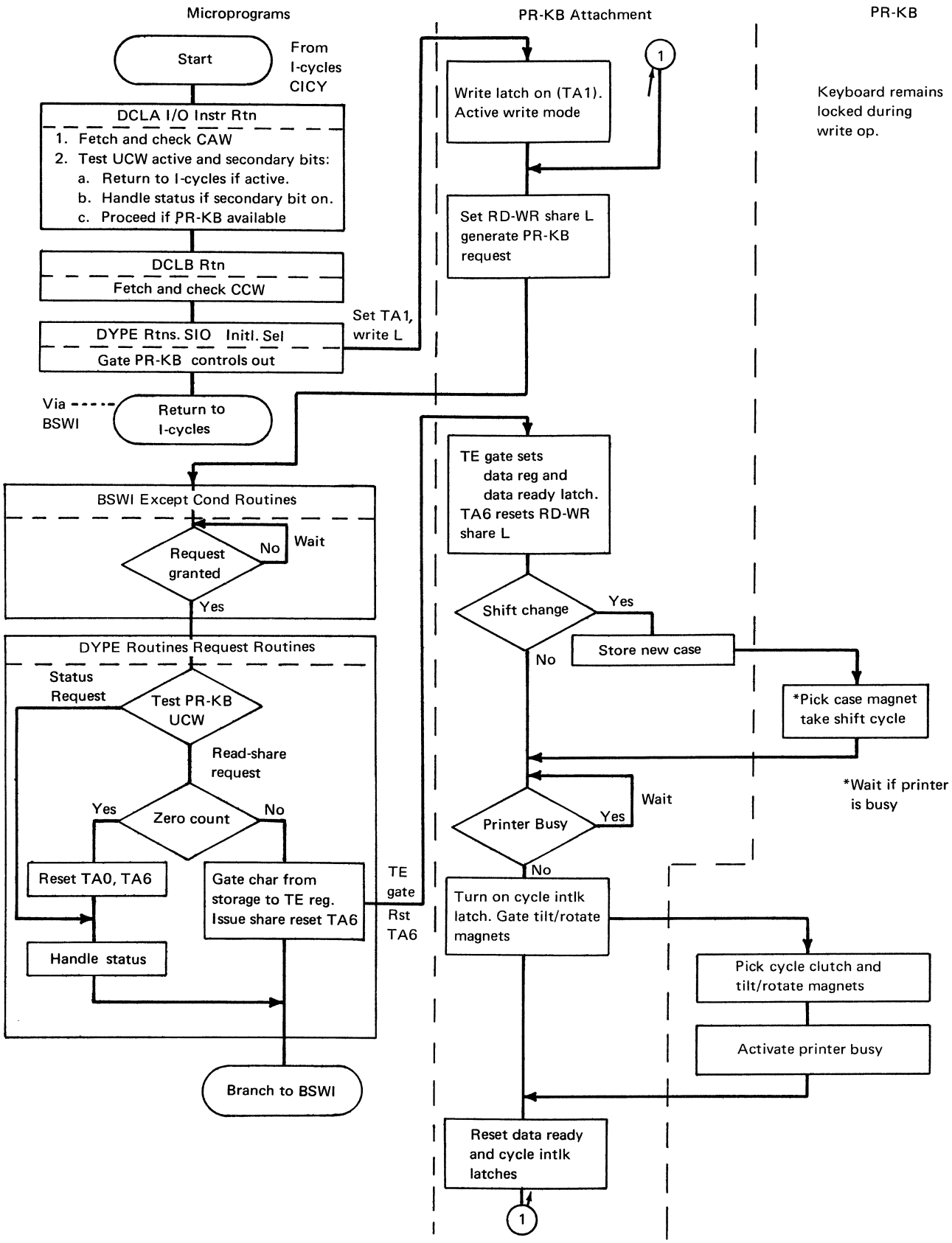


Figure 6-21. Write Operation Flowchart

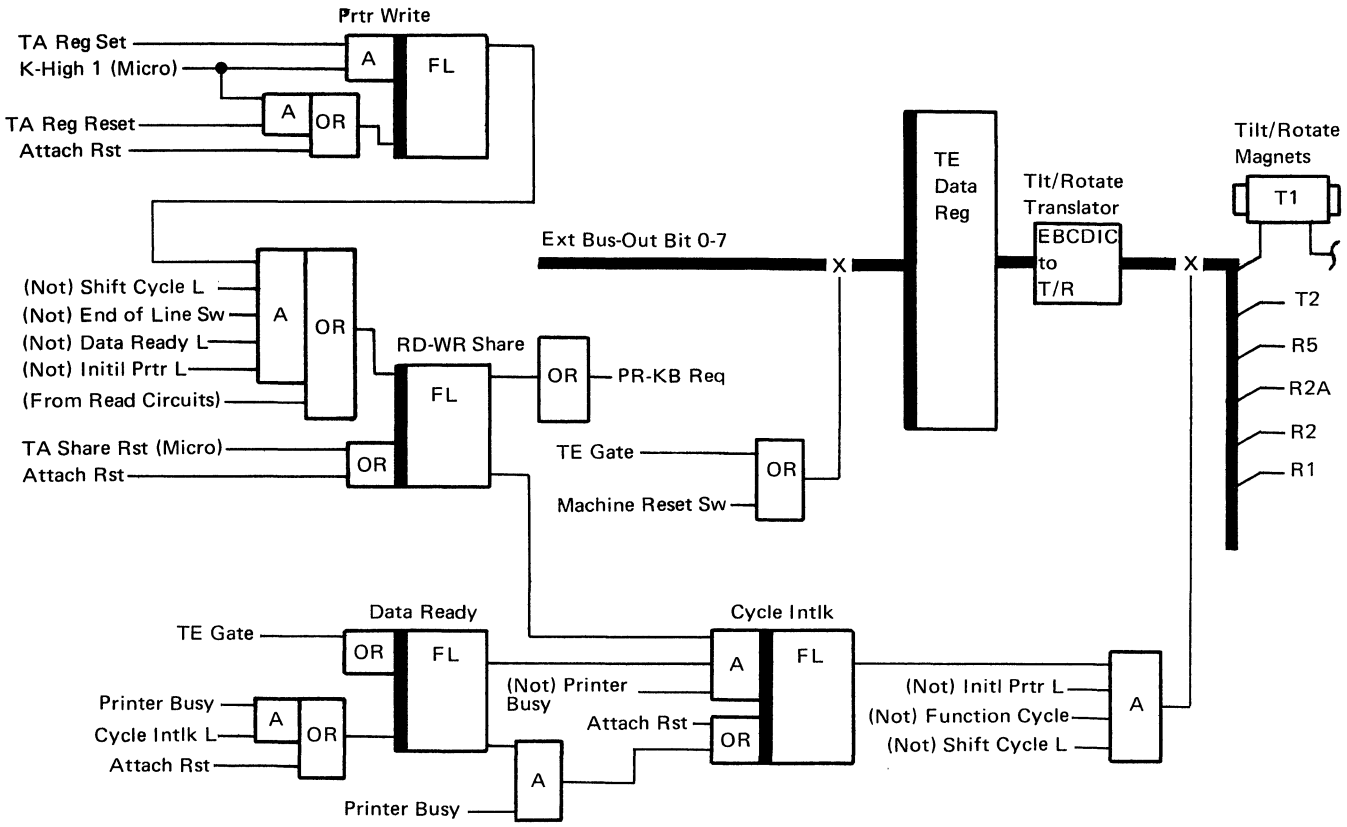


Figure 6-22. Write Controls

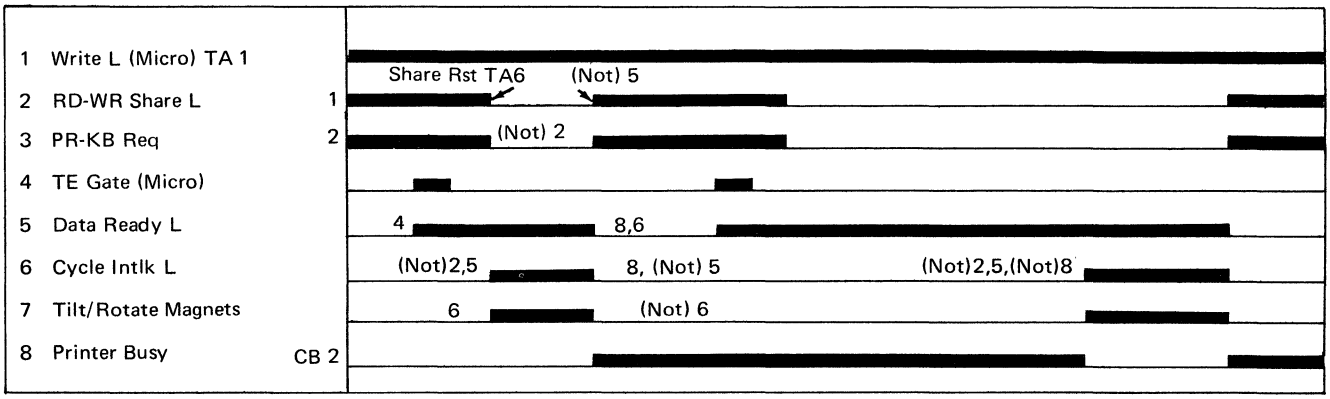


Figure 6-23. Write Timing Chart

Initial Selection--Write Command and Write with ACR Command

- Microprogram sets TA bit 1 to turn on the write latch in the logic circuits.
- Write latch and '(not) data ready' latch turn on 'RD-WR share' latch, generating a PR-KB request.
- Microprogram recognizes request and branches to PR-KB request routine.

The write commands are decoded after the CCW has been fetched for Start I/O or command chaining. See Start I/O--Printer Keyboard Not Busy for a description of the entry. The first step for the write with ACR operation sets bit 6 of the UCW

flags/op byte. Thus, the flags/op byte is 01 for write, and 11 for write with ACR. Next, the write latch is set (TA bit 1). The updated UCW is stored, then the microprogram branches to the BSWI routine. In BSWI, CPU mode and zone are set. Also, the CPU storage protect key is restored. Then the microprogram returns to I-cycles, continuing with other routines until the PR-KB generates a write request. Condition code 0 (console printer available) was previously set at the start of the DCLA routine.

In the console printer logic circuits, the write latch brings up write mode and run mode. These lines prevent the development of a request signal from: request key, logout, or alter/display. The request signal for the write operation comes from the 'RD-WR share' latch. The 'RD-WR share' latch is turned on by the write latch and '(not) data ready' latch. The request signal (TT bit 6) activates '(not) S7 branch condition' and 'integrated I/O request' (BB bit 1).

The control program recognizes the status of S7 and branches to BSWI, the exceptional condition routine. Because BB bit 1 and TT bit 6 are set, the microprogram branches to the PR-KB request routine. When the write operation enters the request routine initially, the UCW active bit=1 and the secondary bit=0. See Write Share Request for a description of the request cycles and the continuation of the write operation. The main function of the request routine is to gate the EBCDIC character to the PR-KB data register for printing.

PR-KB REQUEST ROUTINE

- The PR-KB request routine is used when a share request is required for the following.
  1. Attention request key
  2. Read share request
  3. Write share request
  4. Interrupt stacked share request (microforce)
  5. Ready share request (not-ready to ready)
  6. Logout
  7. Alter/display.

The PR-KB request routine is entered from BSWI, the exceptional condition routine. If the entry is for logout or alter/display, the microprogram branches to the BMCK routine for logout or the ALDP routine for alter/display. If not alter/display or logout, the request routine fetches the PR-KB UCW. Local storage registers are assigned for the UCW and sense byte (see microprogram listing for register assignment):

Sense Byte
Channel Status
Flags/Op Byte
Data Address
Count
Unit Status

See Figure 6-8 for layout of the UCW and Sense Byte.

After fetching the UCW, a test is made for ending status. When the end-status bit is set, the purpose of the request is to handle final status that has not been cleared. For this case, the request routine resets the write latch, read latch, and 'RD-WR share' latch. Also, the attention bit is set if the request is for attention status. Then the routine branches to try for the interrupt. See Common Interrupt Routine.

If ending status is not present, the routine tests the active bit (flags/op, bit 5). The active bit is on if an I/O instruction has been initiated, but channel-end status has not been stored in the CSW. Status is not normally stored for chained commands until the last command ends. After the active bit is tested, the routine checks the secondary bit. The secondary bit is on when any status is queued. It is reset when the status is stored in the CSW.

Active Bit = 0, Secondary Bit = 0

This condition indicates that the request was due to one of the following:

1. Not-ready to ready--set device-end in unit status.
2. Attention-request key--set attention bit.
3. Documentary alter/display switch--go to alter/display routine.
4. Interface error--set interface control check.

Except for alter/display, status must be stored in the CSW for each of the foregoing cases. Therefore, the secondary bit is set and the microprogram branches to the Common Interrupt Routine to try for the interrupt. Then the microprogram updates the UCW and branches to the BSWI routine (or ALDP routine if alter/display).

Active Bit = 0, Secondary Bit = 1

This combination can indicate either of two conditions:

1. Channel-end status for a read or write operation has been cleared by an interrupt or Test I/O. The current request sets device-end (if the attention bit is off) and resets the

read latch, write latch, and 'RD-WR share' latch. Then the microprogram branches to the Common Interrupt Routine to try for the interrupt. The microprogram updates the UCW and branches to the BSWI routine.

2. The request is another attempt to initiate an interrupt to clear status for one of the following:
   a. Not-ready to ready
   b. Attention-request key
   c. Interface control check (from a previous false request).

The end-status bit is also on for item 2 conditions. The microprogram tries for the interrupt as previously described.

Active Bit = 1, Secondary Bit = 1

This condition indicates:

1. Normal device-end. Channel-end (or channel-end and device-end) has occurred, but has not been cleared. (The end-status bit may be on also.) The current request sets device-end; also the read latch, write latch, and 'RD-WR share' latch are reset. The microprogram then branches to the Common Interrupt Routine to try for the interrupt. Then the microprogram updates the UCW and branches to the BSWI routine.
2. Command chaining time. The last cycle of the current command has ended. The UCW is updated with the current unit status, then the microprogram branches to the DCLB routine to fetch the chained command.

Active Bit = 1, Secondary Bit = 0

This condition indicates a read or write share request. The microprogram tests for intervention required (TT bit 2). in the sense byte. Then, the routine branches to do a carriage return/line feed and terminate the operation. See CR/LF Ending Routine.

If the microprogram does not branch on the preceding condition, the routine tests for a PCI flag. When this flag is on and the interrupt buffer is available, an interrupt is initiated. (See Common Interrupt Routine.) Whether or not the PCI flag was on, the routine next tests the op-bits (Flags/Op byte) and branches to do either a 'read share request' or a 'write share request'.

READ SHARE REQUEST

- Microprogram gates keyboard character to local storage using TI external field.
- Keyboard character is translated to EBCDIC by translate subroutine.
- EBCDIC character is gated to PR-KB data register by TE external field.
- In the logic circuits, the contents of the data register are translated to the tilt/rotate code and sent to the PR-KB tilt/rotate magnets.

After the request routine has determined that the request is for a read operation, the keyboard character is gated to local storage from the TI external field. (In hardware, the TI gate is brought up by 'Ext Decode A' from the control word. Next, TT bit 5 (alternate coding key) is tested. This bit is on if the operator has attempted an EOB or cancel. (See Read Operation Ending Procedure.)

If the alternate coding key is not active, the keyboard-check latch (TT bit 4) is tested. Equipment check is set in the sense byte when a keyboard check is detected. The 0-count bit is also tested in this section of the routine. This bit will be on if the UCW count was reduced to zero during the previous request. If the 0-count bit is on, the routine branches to the end of the read operation. (See Read Operation Ending Procedure.)

If the 0-count bit is off, the routine processes the read operation. The keyboard character is translated to EBCDIC by the translate subroutine. Then the character is gated back to the console-printer data register (external field TE). Also, a share reset (TA bit 6) is issued.

In the hardware, the TE gate turns on the 'data ready' latch and fires singleshot 3. The share reset (TA bit 6) turns off the 'RD-WR share' latch. 'Data Ready', '(not) printer busy', and '(not) RD-WR share' turn on the cycle interlock latch. This latch picks the cycle clutch in the console printer. 'Cycle interlock' also gates the translated output of the TE data register to the tilt/rotate magnets. The output of the tilt/rotate translator can be gated to the CPU by a control word that has an external decode of B (TR-register). The TR and TD-registers are used in the diagnostic routines.

Singleshot 3 energizes the KB-restore magnets for 40ms. The KB-restore magnets remain energized until singleshot 3 times out and busy drops (CB 2 N/C closes). Another key can be operated to initiate

another request any time after the keyboard is unlocked.

After issuing the share reset, the microprogram stores the character just received at the data address, then the data address is incremented. (If the skip flag is on in the UCW Flags/Op byte, the preceding step is performed.) The routine continues, decrementing the UCW count.

If the UCW count did not decrement to zero, the updated UCW (not unit status) is stored and the microprogram branches to the BSWI routine to test for the next I/O request.

If the UCW count is now zero, the chain data falg is tested. If the chain data flag is not set, the 0-count bit is set and the updated UCW (not unit status) is stored. Then the microprogram branches to the BSWI routine to test for the next I/O request.

If the chain data flag is on, the microprogram resets the secondary bit and updates the UCW including the unit-status byte. Then the microprogram branches to the DCLB chain routine to fetch the next CCW.

Read Operation Ending Procedure

The read operation can be terminated as follows.

1. EOB key with 0-count bit on. This is the normal ending. The routine branches to the CR/LF ending routine.
2. EOB key with 0-count bit off. The routine tests the SLI (suppress length indication) flag. Incorrect length is set in the channel status if the SLI flag is off. Then the routine branches to the CR/LF ending routine.
3. Cancel key. Ending is the same as for the EOB key except that the command chaining flag is reset and unit exception is set in unit status.
4. Intervention required. Bit 1 is set in the sense byte, then the routine branches to the CR/LF ending routine.

The CR/LF ending routine is described separately. This routine initiates the carriage return line feed and sets the secondary bit. Channel-end is set if not command chaining.

Anotner request cycle occurs after the CR/IF cycle. If the CC flag is not set, device-end is set during this request cycle. Also, the read latch, write latch, and 'RD-WR share' latch are reset. The end-status bit is set to indicate final status is ready. To attempt the interrupt, the request routine BALs to the Common Interrupt Routine. Upon return from the BAL, the microprogram branches to the BSWI routine to test for the next request.

If the CC flag is on during the last request cycle, the secondary bit is set and the updated UCW is stored. Then the microprogram branches to the DCLB routine to fetch the next CCW.

CR/LF Ending Routine

The CR/LF (Carriage Return Line Feed) routine is used during the ending procedure for a read or write with ACR operation.

The routine is also used when intervention required is detected at the start of a request cycle.

The first part of the subroutine performs the following steps:

1. Set up new-line character
2. Reset read, set write latches (TA bits 0 and 1)
3. Send new-line character to TE data register
4. Issue share reset (TA bit 6)
5. Set secondary bit in channel status register
6. Reset 0-count bit.

After the preceding steps are completed, the sense-byte register is tested. If any sense bit is on, unit check is set (unit-status bit 6) and the chaining flags are turned off.

Next the command chaining flag is tested. If the CC flag is off, channel-end is set and the microprogram BALs to try for a normal interrupt. (See Common Interrupt Routine.) After trying for the interrupt, the updated UCW is stored (including the unit status). If the CC flag is on, the UCW is stored, but no interrupt is attempted. The microprogram then branches to the BSWI routine to test for the next request.

Function Cycle

- Occurs when a space cr new-line character is decoded or the EOL switch is activated during either a read or write operation.

When the PR-KB is executing a read or write operation, a function cycle occurs if:

1. The character in the TE data register decodes as a space (not bits 0, 2, 3, 4),
2. The character in the TE data register decodes as a new line character (bits 3, 5, 7, and not bits 0, 1, 2, 4, 6), or
3. The new-line latch is turned on by the EOL (PR-KB RH margin) switch.

For a space function cycle, the space magnet is energized in the PR-KB. For a new-line function cycle, the CR-LF (carriage return line feed) magnet is energized. As soon as the PR-KB becomes not busy, the operation continues.

Write Share Request

- Microprogram gates the EBCDIC character to the PR-KB data register using TE external field.
- In the logic circuits, contents of the data register are translated to the tilt/rotate code and sent to the 1052 tilt/rotate magnets.

After the request routine has determined that the request is for a write operation, the 0-count bit (UCW channel status) is tested. This bit is on if the UCW count was reduced to zero during the previous request. If the 0-count bit is on, the routine branches to end the write operation. (See Write Operation Ending Procedure.)

If the 0-count bit is off, the routine proceeds with the write operation. The data character is read from program storage to local storage, then the UCW data address is incremented. The data character is sent to the console printer data register (external field TE). Next, the routine issues a share reset (TA bit 6) and decrements the UCW count field. The updated count is tested and the microprogram branches as follows.

Count = 0

1. If the UCW chain data flag is on, branch to the DCLB chain routine to fetch the chained CCW.
2. If no chain data flag, set 0-count bit in UCW channel status byte, then branch to BSWI routine to test for next I/O request.

Count Not Zero

Branch to BSWI routine to test for next I/O request.

When the microprogram addressed the TE external, the TE gate signal was brought up in the logic circuits. Besides gating the data register, the TE gate turns on the data-ready latch. The data-ready latch and '(not) RD/WR share' turn on the cycle interlock latch. (The RD-WR share latch was turned off when the microprogram issued share reset, TA bit 6.) The cycle interlock latch picks the cycle clutch in the console printer. The same line that picks the cycle clutch also gates the translated output of the TE data register to the tilt/rotate magnets.

After the mechanical cycle has started in the console printer, CB2 activates a line that turns on the printer-busy latch. 'Printer busy' resets the data-ready latch. Then the cycle-interlock latch is reset. Because the write latch remains on throughout the operation, the RD-WR share latch is turned back on when the data ready latch goes off. This generates the request for the next character. After the last character is sent to the console printer, one more request cycle occurs. This request cycle causes a branch to terminate the write operation as described earlier. The Write Operation Ending Procedure, which follows, describes the last request cycle(s).

Write Operation Ending Procedure

A write operation enters the ending procedure when the write request finds the 0-count bit set. The ending procedure resets the 0-count bit and tests the SLI flag. Incorrect length is set in channel status if the SLI flag is not set.

Write without ACR: If the operation is write (no ACR), the sense byte register is tested. If any sense bit is on, unit check is set (unit status bit 6) and the chaining flags are turned off. The CC flag is then tested and if it is off, channel-end, device-end, and the secondary bit are set. Also, the read latch, write latch, and 'RD-WR share' latch are reset. The end-status bit is set and the routine tries for the interrupt (See Common Interrupt Routine.) Then the microprogram branches to BSWI to test for the next request.

If the CC flag is on during the last request cycle, the secondary bit is set and the updated UCW is stored. Then the microprogram branches to the DCLB routine to fetch the next CCW.

Write with ACR: If the operation is write with ACR, the routine branches to the CR/LF ending routine. From this point, the operation is the same as the read operation. See Read Operation Ending Procedure.

Case Shift During Write Operation

- Automatic case shift occurs when the case of the character being translated differs from the case store latch.

Figure 6-24 is a timing chart for a shift to lowercase during a write operation. If the PR-KB is in uppercase ('U-case store' latch on) and a lowercase character is decoded, lowercase must be set in the attachment and the PR-KB must take a shift cycle. As soon as the shift cycle ends (printer not busy), the tilt/rotate

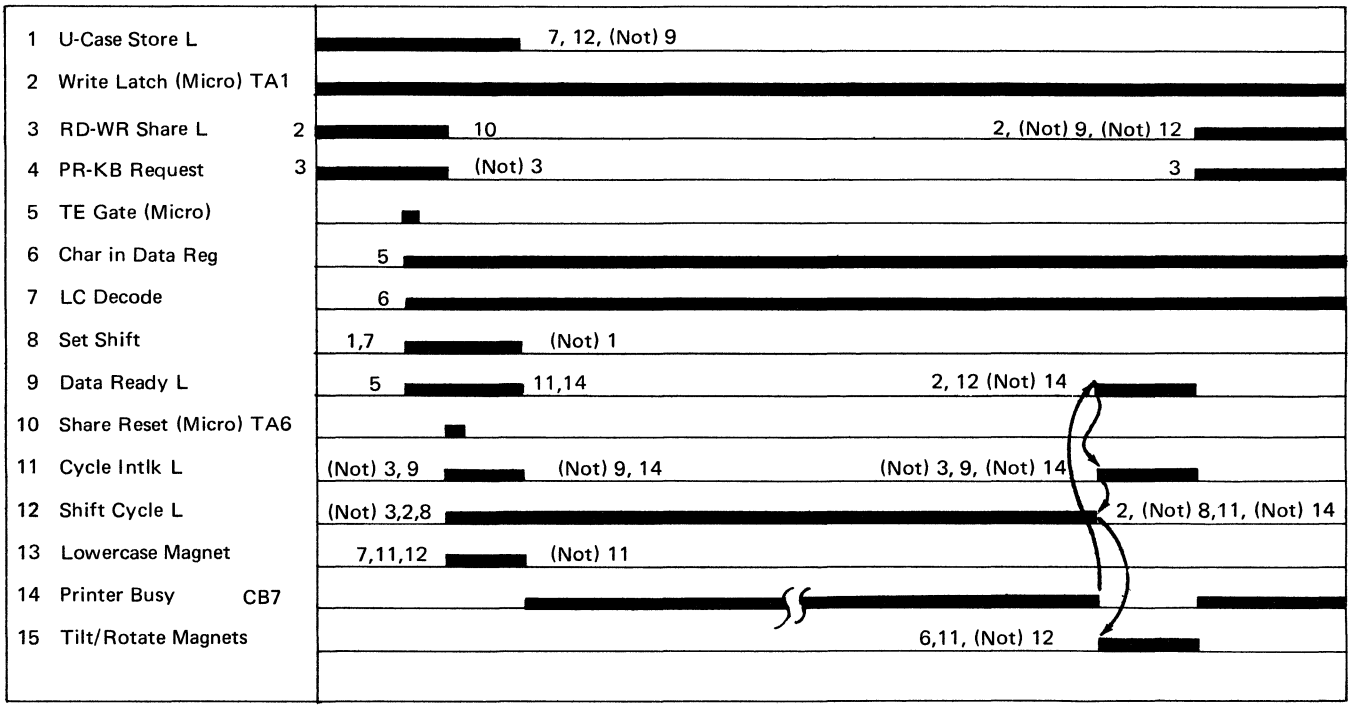


Figure 6-24. Shift to Lowercase during Write Operation

magnets for the character are energized and the print cycle is started. The timing chart shows the signals involved in the shift operation.

A shift from lowercase to uppercase is similar, except that the 'U-case store' latch is off at the start of the operation. This latch is turned on by an uppercase decode from the data register.

Common Interrupt Routine

- Used when channel-end status and/or device-end status is ready for the CSW.
- Initiates an interrupt if the I.B. is available.

This portion of the request routine initiates an interrupt if the interrupt buffer is available. If the interrupt buffer is not available, entry to this routine occurs again on the next request unless the status is cleared by Test I/O or Start I/O. Another request is microforced when the channel-0 interrupt latch (S7) goes off.

PCI Entry: This entry is used to try for a program-controlled interrupt. If the PCI flag is off, the microprogram returns to the link address. If the PCI flag is on, the 'PCI-alone' indicator bit is turned on for the I.B. (Interrupt Buffer). From this point, PCI is handled the same as the normal interrupt.

Normal Interrupt Entry: This entry is used to try for a normal interrupt; i.e., when channel-end and/or device-end status is ready. The 'interrupt in I.B.' bit (channel status bit 6) is tested first. If this bit is on, the PR-KB already has control of the interrupt buffer. In this case, the routine does not have to see if the I.B. is available. The PR-KB address is stored in the I.B. immediately. The 'PCI-alone' indicator is also stored in the I.B. (if it is on).

If the interrupt in I.B. bit is not on, priority hold (MMSK bit 7) is turned on. Priority hold prevents a level-1 trap by another device while the PR-KB is trying to put the UCW address in the channel-0 interrupt buffer. Next, the routine tests BA bit 0. If this bit is on, the interrupt buffer is being used by another device on channel 0. If this is the case, priority hold is reset. (The 'PCI-alone' indicator is also reset if it was turned on at the PCI entry.) If the CC flag is on, the microprogram returns to the link address. Otherwise (no CC flag), the end-status indicator is tested. If this bit is on, the microforce latch (TA bit 2) must be set

(to force another request) before returning to the link address.

If BA bit 0 is off when tested, the I.B. is available and the interrupt is initiated by performing the following steps.

Set channel 0 interrupt latch (S7)
Reset priority hold (MMSK bit 7)
Set I.B. bit in channel-status register
Store UCW address in interrupt buffer
Store 'PCI-alone' bit in I.B. (if it was turned on)
Issue share reset (TA bit 6)
Reset end-status indicator bit (if on).

After completing the above steps, the microprogram returns to the link address.

TEST I/O COMMAND (PR-KB)

- Test I/O checks the current status of the PR-KB and sets a condition code.

- A CSW is stored for:

1. channel-end,
2. device-end, or
3. PR-KB not operational.

A Test I/O command for the PR-KB is identified in the DCLA I/O Instructions routine. This command sets a condition code as follows.

Condition Code	
0	Available
1	CSW stored
2	Working

To determine the condition code and action to be taken, the microprogram first tests the active and secondary bits in the PR-KB UCW.

Active Bit = 1, Secondary Bit = 0

This condition indicates that the PR-KB is working. Condition code 2 is set, and the microprogram returns to I-cycles.

Active Bit = 1, Secondary Bit = 1

If the command chaining flag is on, the PR-KB is working, condition code 2 is set, and the microprogram returns to I-cycles.

If the command chaining flag is off, the microprogram branches to the PR-KB DYPE routine for further tests. In preparation for these tests the microprogram reads the sense byte, channel status, flags/op byte, and the unit status. Because Test I/O will clear any queued status, the microforce latch (TA2) is reset. If the program-check bit is off and the unit status equals zero, chaining is in progress. Condition code 2 is set, and the microprogram returns to I-cycles.

If the unit status is not zero or the program check bit is on, the active bit is reset because channel-end status is cleared. If the 'interrupt in I.B.' bit is set, it is reset and also the S7 channel-0 interrupt latch is reset. This is done because an interrupt will no longer be needed to clear status. When the device-end bit is on, device-end status is cleared. In this case channel status is set to zero in the updated UCW. When channel-end status alone will be cleared, channel status is stored in the UCW with the secondary bit on. The register used to store channel status in the UCW is not the same one that is used to store channel status in the CSW.

If the PCI flag is on, the PCI bit is set in the channel status to be stored in the CSW. The microprogram forms the remainder of the CSW, then stores the CSW at location 0040 of program storage. After setting CPU mode and condition code 1 (CSW stored), the microprogram returns to I-cycles.

Active Bit = 0, Secondary Bit = 1

The microprogram branches to the PR-KB DYPE routine to handle this condition. This routine reads the UCW and tests for the following conditions.

1. Device-end bit on. Device-end status to be stored in the CSW.
2. Attention bit on. Attention status to be stored.
3. Attention bit off and device-end bit off. Status will be stored with busy set in the unit status.

For conditions 1 and 2, the microprogram proceeds as follows: If the 'interrupt in I.B.' bit is set, it is reset and also the S7 channel-0 interrupt latch is reset because an interrupt is not needed to clear the status. The microforce latch (TA2) was previously reset for the same reason. Except for the status, all zeros are stored in the CSW. The microprogram then sets condition code 1 and returns to I-cycles.

For condition 3, busy is set in the unit status, then status and zeros are stored in the CSW as for the first two conditions.

Active Bit = 0, Secondary Bit = 0

The microprogram branches to the PR-KB DYPE routine to handle this condition.

If the PR-KB is operational (TT2 off), condition code 0 is set and the microprogram returns to I-cycles.

If the PR-KB is not operational, unit-check is set in the unit-status byte and the channel status byte is zeroed-out.

Channel status is reset. The CSW stored contains all zeros except for unit status. Condition code 1 is set. Then the microprogram returns to I-cycles.

HALT I/O COMMAND (PR-KB)

- Halt I/O sets a condition code based on the current status of the PR-KB and the action taken by the command.

A Halt I/O command for the PR-KB is identified in the DCLA I/O Instructions routine. This command sets the condition code as follows.

Condition Code	
0	Interrupt pending
1	CSW stored
2	--
3	PR-KB not operational

To determine the condition code and the action to be taken, the microprogram branches to the DYPE routine. The first halfword of the UCW is read, and the active and secondary bits are tested as follows.

Active Bit = 0, Secondary Bit = 0

1. Intervention Required (TT bit 2) on. For this condition, the microprogram sets condition code 3 (not operational), then returns to I-cycles.
2. Operational. For this condition, the microprogram sets the channel status to zero before storing status in the CSW. Then condition code 1 (CSW stored) is set and the microprogram returns to I-cycles.

Active Bit = 1, Secondary Bit = 0

For this condition, the operation is set up for termination as follows.
1. Read latch is reset (TA bit 0)
2. Write latch is set (TA bit 1)
3. 0-count bit is set
4. CC and CD flags are reset
5. Updated channel status and flags are stored in UCW.

Then unit status and zero channel status are stored in the CSW. The microprogram sets condition code 1 (CSW stored) and returns to I-cycles.

Secondary Bit = 1, CC Flag = 1

Channel-end is set in the channel-status byte, and the CC and CD flags are reset. Then the updated channel status and flags are stored in the UCW. The microprogram stores unit status and zero channel status in the CSW, sets condition code 1, and returns to I-cycles.

Secondary Bit = 1, CC Flag = 0, Active Bit = 0

The microprogram returns to I-cycles and sets condition code 0.

Secondary Bit = 1, CC Flag = 0, Active Bit = 1

1. Device-end set. The microprogram returns to I-cycles and sets condition code 0.
2. No device-end. The microprogram stores unit status and zero channel status in the CSW, sets condition code 1, and returns to I-cycles.

KEYS AND INDICATORS (PR-KB)

- Figure 6-25 shows the PR-KB keys and indicators.

Certain keys and indicators are not used when the 1052 Model 7 is attached to the System/360 Model 25. These unused items are all on the upper portion of the console and are indicated by an asterisk (*) in Figure 6-25.

INDICATORS (FIGURE 6-25)

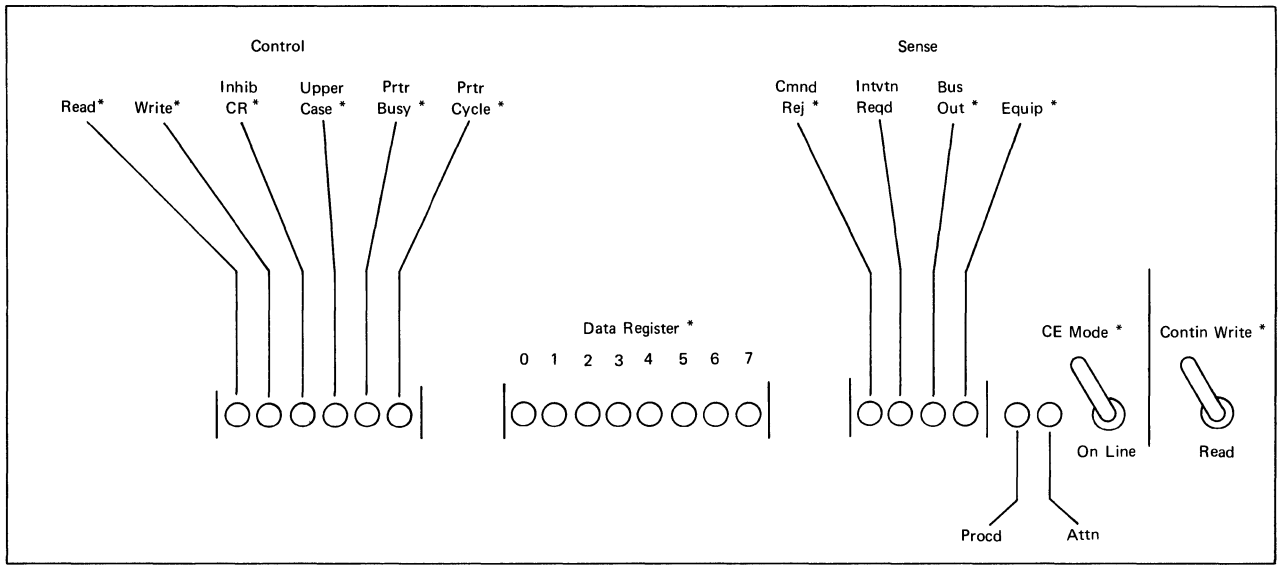
Intervention Required (INTVTN REQD)

This indicator is turned on when:
1. The forms switch indicates that the printer is out of forms, or
2. The printer not-ready key is operated. (Pressing this key turns on the not-ready latch, then resets the ready latch.)

The intervention required signal also goes to bit 2 of the TT external field. If a PR-KB operation (other than sense or no-op) is in process or is initiated, the PR-KB microprograms set bit 1 of the sense byte and unit-check bit 6 of the unit-status byte, then terminate the operation.

Proceed (PROCD)

The proceed indicator turns on each time the keyboard unlocks to allow keying during either a read or alter/display operation. The light turns off when the RD-WR share latch is turned on by single-shot 2 and KB strobe (operator has pressed a key). The keyboard unlocks and the indicator comes on again when single-shot 3 (40ms) times out and the printer becomes not-busy.



* Indicators or switches marked with an asterisk are not used.

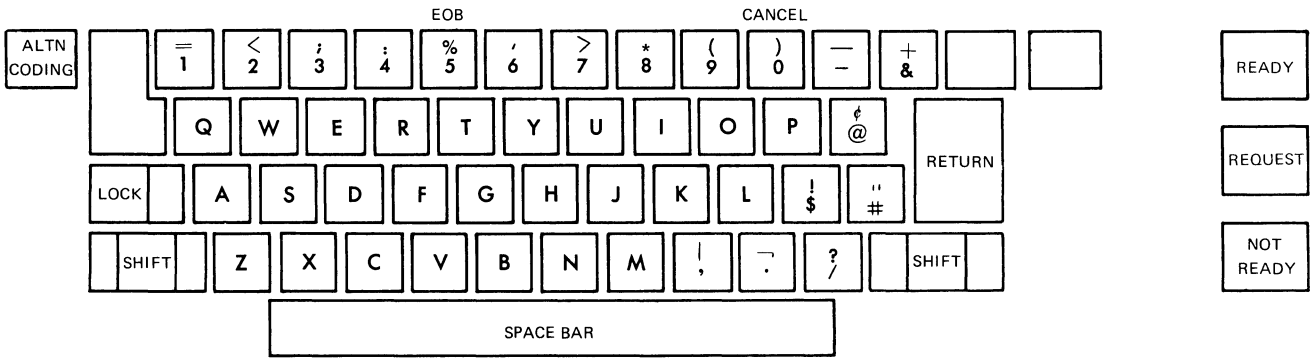


Figure 6-25. Printer-Keyboard

Attention (ATTN)

The attention indicator turns on when the request key is operated. Pressing this key turns off the attention interlock latch. Then, when the key is released, the attention latch and indicator turn on. The attention latch generates a PR-KB request to set attention status in the UCW unit-status byte. When this is accomplished, the microprogram issues Attention Rst (TA7), which sets the attention interlock latch and resets the attention latch and indicator. The attention status is subsequently presented to the problem program via the CSW.

Alter/Display Indicator on CPU Console (ALTER DPLY)

This indicator and the ALT/DISP ACTIVE latch (TA3) are turned on in the ALDP microprogram routine for the following operations.

1. PR-KB alter or display operation (initiated by operating the PR-KB alter/display key on the CPU console).
2. Set instruction counter (initiated by operating Set IC on the CPU console).
3. Instruction-step typeout (INSN STEP position of the mode switch on the CPU console).
4. Logout (initiated by machine check).

The ALT/DISP ACTIVE latch and alter/display indicator are turned off by the microprogram when the operation is completed.

KEYS (FIGURE 6-25)

Request

Pressing this key turns off the attention interlock latch. Then, when the key is released, the attention latch and indicator turn on. The attention latch generates a PR-KB request to set attention status. See also the ATTN (Attention) indicator description.

Ready

Operating this key turns on the ready latch (if forms are in place). This generates a not-ready to ready request. The PR-KB request microprogram routine sets device-end in the UCW unit-status byte and initiates an I/C interrupt.

Operating the ready key causes no action if the PR-KB is already in the ready state.

Not Ready

When this key is operated, the PR-KB goes not-ready. Pressing the key turns on the not-ready latch, then resets the ready latch. When the ready latch turns off, intervention required (TT2) is activated. See also the INTVTN REQD (Intervention Required) indicator description.

Alternate Coding (ALTN CODING)

This key must be held down in its operated position for the EOB or cancel keys to be operative. In the PR-KB DYPE and ALDP microprogram routines, a test is made for the alternate-coding key (TT5). If TT5 is active, tests are made for EOB (keyboard character 5) and for cancel (keyboard character 0). The alternate-coding relay is not picked.

End-of-Block (EOB)

The EOB key can be operated to either:
1. terminate a read command operation, or
2. conclude the current manual keying during an alter/display operation. (The alter/display operation, however, is not terminated unless the start key is operated after the EOB.)

The alternate coding key must be held in its operated position for the EOB key to be effective.

The EOB key sends a keyboard code to the CPU (character 5). In the PR-KB DYPE and ALDP microprogram routines, a test is made for the alternate-coding key (TT5) and the keyboard bits for the EOB character.

Cancel

This key performs the same functions as the EOB key. However, when the key is operated during a read command (not alter/display), unit exception is set in the UCW unit-status byte.

To test for Cancel, the microprogram first tests for the alternate-coding key (TT2) then for a keyboard character 0.

PR-KB Alter/Display (on CPU Console)

Operating this key causes a request for a PR-KB alter/display operation. Pressing the key turns off the ALT/DISP INTLK latch. When the key is released, the ALT/DISP latch (TT3) is turned on. This generates a PR-KB request as soon as any other operations are completed. When the PR-KB request is granted, the microprogram branches to the ALDP routine. The ALDP routine sets the ALT/DISP ACTIVE latch (TA3) and the read latch (TA0). Setting these two latches causes the alter/display indicator (on CPU console) and the proceed indicator to light.

PR-KB ALTER/DISPLAY FACILITY

This facility provides for the altering or displaying the contents of:
1. auxiliary storage,
2. control storage, (alter is possible only in CE mode), and
3. program storage.

Three advantages provided by this facility are:
1. Current I/O operations (data transfers and chaining) are not caused to overrun, as they might if alter/display operations were initiated from the system console.
2. A printed copy that specifies the operation (alter or display), the location(s) accessed, and the data used is provided by the console printer.
3. Ease of operation compared with the use of manual switches to alter and display data.

DESCRIPTION AND OPERATION

SETUP

To initiate an alter or display operation, the operator must first press the PR-KB alter/display button (on the CPU console) and then wait for the proceed light and the alter/display light to turn on. If the

keyboard-printer is busy executing another operation, these lights will not turn on until that operation is ended.

When these lights turn on, the operator types a two-character sequence, depending upon the desired operation. These characters are called the operation characters and can be:

First Character: a
  Specifies: alter operation
First Character: d
  Specifies: display operation

Second Character: a
  Specifies: auxiliary storage
Second Character: c
  Specifies: control storage (CE mode only for alter)
Second Character: p
  Specifies: program storage

Addressing

Next, the operator types an address for:

Area	Address Format (lowercase* hex)
auxiliary storage	xxxx
control storage	xxxx
program storage	xxxx

*Note: Uppercase can be used if the uppercase character does not differ from the lowercase character.

The four-digit hexadecimal address for control storage is typed as it appears on the microprogram listing, i.e., 0000 - through 3FFF only.

Execution of Alter and Display

After the operation character and address have been typed, the console printer advances to the beginning of a new line.

1. For a display operation: The keyboard is locked and printing begins, starting with the contents of the specified address. After 16 hexadecimal digits (4 halfwords) have been printed, the console printer stops, the keyboard unlocks, and the proceed lights turns on. At this point the operator has two choices:
   a. Successive blocks of 4 halfwords can be printed by pressing the spacebar (or any character) after each block is printed.
   b. This particular pass of the alter/display routine can be terminated by pressing either the EOB key or cancel key. This causes a return to the beginning of the routine, allowing the operator to initiate another alter or display operation.
2. For an alter operation: The keyboard is unlocked and the data to be stored can be typed immediately. However, it is good practice to first examine the typed input message so that its accuracy can be verified.

END OPERATION

Either the EOB key or the cancel key (in conjunction with the alternate coding key) can be used to end an alter or display operation, except when the keyboard is locked during a display printout. Pressing either key causes a return to the beginning of the alter/display routine so that the operator can perform another display or alter operation.

An alter or display operation is normally ended by pressing the EOB key after the required data has been typed. If the cancel key is used to end an alter operation, the last character is stored only if it completes a byte. (This is explained later in this section.)

An exit is not made from the alter/display routine until:
1. the EOB key or cancel key is operated to end the current alter or display operation, then
2. the CPU start key is pressed.

Therefore, the operator can perform as many alter and/or display operations as desired without exiting from the routine. Until the exit is accomplished, however, CPU instruction processing is stopped.

CPU instruction processing is resumed when, after an exit is made from the alter/display routine, the start key on the CPU console is pressed.

During an alter operation, data is altered on a byte basis. For example, assume that the byte data to be entered is 'af', but that the operator types 'bf' instead of 'af'. The 'bf' is stored. The operator should then press either the EOB or the cancel key to return to the beginning of the routine for a retry of the entire operation.

If the error is noticed before the 'f' is typed, the byte is not stored. If the cancel key is then operated (before the 'f' is entered), a return to the beginning of the routine is made, and the operation can be retried if desired. In either case, the entire manual operation must be repeated to store the 'af'. If the operator types one too many characters (such as 'aff' for the foregoing example), the operation would not have to be repeated.

MESSAGE FORMATS

The microprogram initiates a line feed and carrier return after sixteen halfwords have been printed on a line (for either alter or display). Data is column-justified and blanks are provided between halfwords for either display or alter data. Also, the routine provides offsetting of two spaces at the left margin if an odd starting address is specified for alter or display of main, auxiliary, or control storage (see following formats).

All input typing is in lowercase (uppercase can be used if the uppercase character does not differ from the lowercase character). All output printing of hexadecimal alpha characters is in uppercase. In the following formats, x equals a valid hexadecimal digit.

Alter Storage

The second operation character can be 'a' (auxiliary), 'p' (program). If the key switch is in the CE mode position, the operation character 'c' (control) can be used.

```
Input message:  aa xxxx
          Address-----↑
Input data (odd
  address):  xx xxxx xx (EOB)
Input data (even address):
  xxxx xxxx xxxx xxxx xxxx.....xxxx
  xxxx (EOB)
                  16th halfword----↑
```

Display Storage

The second operation character can be 'a' (auxiliary), 'c' (control), or 'p' (program).

```
Input message:  dp xxxx
         Address------↑

Output data
1.  odd address
    xx xxxx xxxx xxxx
2.  even address
    xxxx xxxx xxxx xxxx
```

ERRORS

The error message, INVALID CHAR, is printed if any one of the following operator errors is made.

1. An operation character is other than:
   First character: a or d
   Second character: a,c,p (c is invalid in customer mode for alter operation)
2. A character other than a valid hexadecimal digit is typed for an address or input alter data.
3. A key other than EOB or cancel is operated while the alternate coding key is pressed.

A keyboard check causes a return to the beginning of the alter/display routine.

The error message, INVALID ADDR, is printed if the program control, or auxiliary storage address is invalid for the system.

Note: Address checking is performed by the alter/display microprogram (not by storage-wrap checking circuits in the hardware).

The valid address boundaries for the various systems are as follows.

Size	Prog Stor	Cont Stor	Aux Stor
16K	0000-3FFF	0000-3FFF	0x00-7xFF
24K	0000-5FFF	0000-3FFF	0x00-5xFF and 8x00 BxFF
32K	0000-7FFF	0000-3FFF	0x00-BxFF
48K	0000-BFFF	0000-3FFF	0x00-FxFF

*The four-digit hexadecimal address for control storage is typed as it appears on the microprogram listing; i.e., 0000 through 3FFF.

The following are examples of the invalid address error message.

1. If the address typed in the input message is invalid, the error message is printed after the console printer advances to a new line.

```
Input message:  dp 4000
                   ↑Invalid address-16K
Data:           INVALID ADDR
                ↑Error Message
Input message:  aa 8000
                ↑Invalid Address (16K)
Data:           INVALID ADDR
                ↑Error Message
```

2. If the address goes invalid during the alter or display operation, the error message begins on the same line with the data.

```
Input message:  ac 3FFE
  (assume 16K system CE Mode)
Data:           xxxx INVALID ADDR
```

(Note: First two bytes are valid. Error message applies to addresses beyond this point.)

```
Input message:  dp 3FFF
Data:             xx INVALID ADDR
                ↑ 2 blanks for odd addr
```

PR-KB ALTER/DISPLAY MICROPROGRAM ROUTINE

MDM 5-77, 5-78, and 5-79 show the high-level flow of the ALDP routine for an alter or display operation. Additional information is included with the ALDP routine in the listings.

LOGOUT

MACHINE CHECK LOGOUT

- MDM Diagram 5-80 is a flowchart of the machine check trap operation. Machine check logout is shown in detail on this diagram.

The PR-KB logout is a microprogram function that prints (in hexadecimal notation) the information contained in the diagnostic logout area of program storage. This typeout is initiated for each machine check (when the machine check mask bit in the current PSW, bit 13, is set to 1). Problem programs are not affected by this logout.

Subsequent to the PR-KB logout, a System/360 machine check interruption is initiated. Existing restart procedures and problem programs that do not act directly upon the diagnostic logout area (such as BPS, DOS, OLTEP, etc.), are applicable to the Model 25 within the limits of core storage.

The program storage byte locations and contents of the diagnostic logout area are as follows.

Microprogram Mask (MMSK) register

Address (Hex)	Bit	Contents
80	0	Channel high trap
	1	2311 disk control trap
	2	Channel low trap
	3	2540 reader trap
	4	2540 punch trap
	5	Communications bit service
	6	Communications character service
	7	Level 1 priority hold
81		Branch condition (BA) register
	0	Channel-0 interrupt
	1	Mode bit 0
	2	Mode bit 1
	3	Mode bit 2
	4	IPL latch
	5	LS zone bit 0
	6	LS zone bit 1
	7	LS zone bit 2
82		Machine check (MC) register
	0	File control check
	1	Storage protection check
	2	Storage address check
	3	Control-word parity latch
	4	Storage-data parity latch
	5	ALU error latch
	6	A-register parity latch
	7	B-register parity latch
83	0-1	Logout Code (00 for machine check)
	2-7	Count of machine-check errors.
84-85		Address of control word at which the machine check occurred.

CHANNEL CONTROL CHECK LOGOUT

The channel microprogram initiates a logout on the 1052 only on channel control checks. A channel control check is a machine check while in a channel high or low priority trap.

The logout is as follows. When a channel control check trap occurs, this routine stores into the program storage starting at hexadecimal location 80 and prints out on the PR-KB (Console Printer Keyboard) through the ALDP routine the same information as previously listed for machine check logout.

INTERFACE CONTROL CHECK LOGOUT

The channel performs an asynchronous logout in main storage when an interface control check is detected by the microprogram. The following information is logged.

Hex Address	Contents
80	Trap priority register (MMSK)
81	Branch condition register (BA)
82	GS channel external conditions
83	Logout code and error count
84	GT channel external conditions
85	GD channel external conditions
86	Code byte
87	Unit address

The bit significance of the first two bytes (80 and 81) is the same as previously listed for machine check logout.

GS Channel External Conditions

Address (Hex)	Bit	Contents
82	0	Data chain request
	1	Buffered device latch
	2	Channel-1 burst latch
	3	Channel parity-error latch
	4	Initial selection latch
	5	Channel-1 interruption latch
	6	Spare
	7	Suppress control latch

Address (Hex)	Bit	Contents
83	0-1	Logout code (11 for interface control check)
	2-7	Count of errors

GT Channel External Conditions

Address (Hex)	Bit	Contents
84	0	Address-in
	1	(Not) Select-in
	2*	Service-in
	3*	Status-in
	4	Cperational-in
	5	(Not) Request-in
	6	Channel identification latch
	7	Channel diagncstic latch

*Service-In and Status-In conditions from the interface are detectable in GT only if neither Command-Out nor Service-Out has yet been brought up in response to Service-In or Status-In.

GD Channel External Conditions

Address (Hex)	Bit	Contents
85	0	Operational-out
	1	Service-out
	2	Address-out
	3	Command-out
	4	Spare
	5	Select-out
	6	Spare
	7	Suppress-out

Code Byte

Address (Hex)	Bit	Contents
86	0-1	(no significance)
	2	Detection via timeout
	3-6	(no significance)
	7	0 = status stored in CSW 1 = status stored in interrupt buffer

Address (Hex)	Bit	Contents
87	0-7	Unit address

Unit	Speed	Capacity
2025 Processing Unit	900 ns--Basic machine cycle 1.8 us--Core storage access 180 ns--Local storage access	Storage Capacity in Bytes Model D / DC / E / ED Program Storage: 16,384 / 24,576 / 32,768 / 49,152 Control Storage: 16,384 / 16,384 / 16,384 / 16,384 Auxiliary Storage: 2,084 / 2,560 / 3,072 / 4,096 Local Storage: 64 / 64 / 64 / 64
Printer--Keyboard (PR-KB) 1052 Model 7	15 characters/second (approx)	
Integrated 1403 Attachment 1403 Model 2 Printer	PRINT SPEED 600 lpm (nominal) 750 lpm (max) with MCS feature CARRIAGE SPEED 33 inches/second when line spacing 75 inches/second when skipping	132 print positions
1403 Model 7 Printer	PRINT SPEED 600 lpm (nominal) CARRIAGE SPEED 33 inches/second	120 print positions
Integrated 2311 Attachment 2311 Model 1 Disk Storage Drive	ACCESS TIME 25 ms--Adjacent cylinder 135 ms--cylinder 202 to cylinder 000 DATA TRANSFER 156,000 bytes/second (156 KB)	Maximum of 4 drives using 1316 Disk Pack 7.25 million bytes per disk pack
Integrated 2540 Attachment 2540 Card Read-Punch	READ SPEED 1000 cards/minute PUNCH SPEED 300 cards/minute	Read hopper including file—3100 cards Punch hopper--1350 cards
Multiplexer (Byte) Channel (Channel 0)	MAXIMUM DATA RATES * Degree of Overlap With Integrated Attachments / Without Data Chaining / With Data Chaining None: 27 KB / 13 KB Reader/Punch/Printer/PR-KB: 11 KB / 5.5 KB Disk-File only: 9.5 KB / 4.2 KB Reader/Punch/Printer/PR-KB/Disk-File: 4.5 KB / 2.2 KB	Addresses up to 152 I/O devices 8 control unit positions. 32 UCW positions (8 UCWs can address up to 16 devices each; 24 UCWs can address only 1 device each).
Selector (Burst) Channel (Channel 1)	MAXIMUM DATA RATES* Degree of Overlap With Integrated Attachments / Without Data Chaining / With Data Chaining None: 40 KB / 18.5 KB Reader/Punch/Printer/PR-KB: 40 KB / 10 KB	Addresses up to 256 I/O devices

*Refer to IBM System/360 Model 25 Functional Characteristics; Form A25-3510 for explanation of data rates, and for data rates when overlap is with the Integrated Communications Attachment. For information pertaining to power and cooling requirements, and overall dimensions and weights, see Systems Reference Library publication IBM System/360 Installation Manual---Physical Planning, Form C22-6820.

Figure A1.

Special circuits are covered in Chapter 2, Functional Units, and Chapter 5, Power Supplies.

Input Power: 200, 220, 235/380, 480 volts ±10%, 50 ±.5 hertz, 60 amperes via a three phase, five-wire shielded cable (fourth wire is neutral for a WYE system and not used on Delta system, and the fifth wire is equipment ground). The shield and ground wire must be bonded to the input line filter case at the cable entry point.

Interval Timer Feature: Provision is made to operate the timer at 50 hertz. See Interval Timer, 50-Cycle Operation in Chapter 4.

1402 Punch Operations 4-19
1402 Read Operations 4-18
1403 Printer Operations 4-20
1442 Card Read-Punch Operations 4-22
1443 Printer Operations 4-21
2311 Disk Data Transfer (Cycle Steal) 2-13
2560 (System Check Indicator) 6-4

AB-Register Assembler 1-6, 2-23
A-Register 2-23
A-Register Check 2-33
A-Register Gating 2-24
Adder 2-24
Address/Data Storage Switches 6-7
Address Conversion (1400) 4-5
Address Error Detection (1400) 4-7
Address Match Circuits 2-12
Address Match (Indicator) 6-1
Addressing, Storage 2-5
Allow Manual Operation (ALLOW MAN), Indicator 6-3
Alter/Display Local Storage 6-13
Alter/Display Message Area 6-19
ALTER DPLY, Alter/Display Indicator on CPU Console 6-46
Alter/Display Printer Keyboard (ALTER DPLY PR-KB) 6-3
Alternate Track Operation (1400) 4-28
ALU 2-24
ALU Check 2-26, 2-33
ALU Check (Indicator) 6-4
ALU Controls 2-25
ALU/External Display 6-13
ALU Modifier 2-26
Arithmetic Constant Word 3-15
Arithmetic and Logical Control Circuits 1-6, 2-22
Assembler, AB-Register 1-6, 2-23
Assembler, Local Storage Address 1-6, 2-31
Assembler, Local Storage Data 1-5, 2-32
Assembler, M-Register 1-4, 2-11
Assembler, Storage Address 1-5
Assembler, Storage Data 2-9
ATTN, Attention (PR-KB) 6-46
Auxiliary Storage 1-4
Auxiliary Storage (1400) 4-7
- Module 0 4-10
- Module 1 4-12
- Module 2 4-12
- Module 3 4-13
- Module 4 4-13
- Module 5 4-13
- Module 6 4-13
- Module 7 4-13

Auxiliary Storage Areas, PR-KB 6-19
Auxiliary Storage, XY-Decode Drive 2-9

B-Register 2-23
B-Register Check 2-33
B-Register Gating 2-24
BC-External Facility 2-20
Branch on Condition Word 3-30
Branch on Mask Word 3-26
Branch Set 3-1
Branch Unconditional Word 3-24
Byte-0 and Byte-1 Indicators 6-1
Byte Source 3-1

C-Register 1-5
Carry Insert 2-25
Case Shift During Write Operation (PR-KB) 6-42
Channel Control Check Logout (PR-KB) 6-50
Channels, Integrated 1-6
Character Configuration (1400) 4-4
Check Control Switch 6-10
Check Reset 6-6
Checks, CPU 2-33
Checks, System (Indicators) 6-3
CHNL (Channel System Check Indicator) 6-4
Clock Control 2-1
Clock, CPU 2-1
Clocks, Storage 2-5
Common Interrupt Routine (PR-KB) 6-43
Compatibility Features, 1401/1460 and 1440 4-1
Compatibility Initialization (1400) 4-2
Console and Maintenance Features 6-1
Console Printer-Keyboard 6-13
Console Procedures 6-12
Control Address Set 6-6
Control Panel, System 6-1
Control Register 2-21
Control Storage 1-3
Control Storage Load (CSL) 3-5
Control Storage Load (Key) 6-3, 6-6
Control Word 3-1
Control Word Check 2-33
Control Words 3-1
- WT 0 Set/Reset 3-5
- WT 1 Arithmetic Constant 3-15
- WT 2 Storage 3-17
- WT 3 Move/Arithmetic 3-23
- WT 4 Branch Unconditional 3-24
- WT 5 Branch on Mask 3-26
- WT 6 Branch on Condition 0 3-30
- WT 7 Branch on Condition 1 3-30

Core Array 2-3
Core Load 3-1
Core Storage 2-1
Core Storage, Manual Display 6-12
Core Storage, Manual Store Operation 6-12
CPU Checks 2-33
CPU Checks (Indicators) 6-4
- ALU 6-4
- CSL Check 6-4
- Parity Indicators 6-4
- Storage Protect (STOR PROT) 6-4

CPU Clock 2-1
CPU Differences (1400) 4-15
CPU Operations (1400) 4-15

CPU Status Indicators 6-1
1400 Mode (Bit 0) 6-1
2020 Mode (Bit 1) 6-1
Address Match (ADR MATCH) 6-1
Allow Manual Operation (ALLOW MAN) 6-3
Alter/Display, Printer-Keyboard (ALTER DPLY PR-KB) 6-3
Auxiliary Storage (AUX STOR) 6-3
Clock Off 6-3
Communications (COMM) 6-3
Control Storage Load (CSL) 6-3
Data Cycle, File 6-1
EXT Mode (Indicators 2, 3 and 4) 6-1
LS Zone (Indicators 5, 6, and 7) 6-1
Trap 6-3
CR/LF Ending Routine (PR-KB) 6-41
CSL, Using Integrated 2540 3-5
CSL, Using Integrated 2560 3-5
CSL, Using a Reader on the Standard Interface 3-5
Current Sources 2-7
Customer Engineering Controls 6-8
Cycle Controls 1-5
Cycle Steal, 2311 Disk Data Transfer 2-13

Data Flow 1-1
Data Flow (PR-KB) 6-18
Decimal Corrector 2-26
Diagnostic Control Switch 6-10
Direct Control Feature 4-52
Application 4-54
Direct Control Bus-In 4-52
Direct Control Bus-Out 4-52
External Signal Bus-In 4-53
Hold-In 4-54
Read-In 4-54
Readout 4-54
Timing Signal Bus-Out 4-52
Writeout 4-54
Disk Compatibility Operation (1400) 4-23
Disk Data Transfer, 2311 2-13
Disk Formats (1400) 4-23
Display 6-6
DYN Register (Dynamic Condition) 2-20

Emergency Power Off (EPO) 5-3
External Facilities 2-14
External Fields--CPU to PR-KB 6-25
External Fields--PR-KB to CPU 6-26
External Interrupt (1400) 4-55

File (System Check Indicator) 6-4
File Circuitry (1400) 4-24
FQ-Register, Loading 4-43
Function Cycle (PR-KB) 6-41

GQ-Register, Loading 4-44

Hard-Stop Latch 2-34
Halt I/O Command (PR-KB) 6-45
High-Digit 3-1
HQ-Register, Loading 4-44

I-Cycles (1400) 4-15
Objectives 4-16
I8 4-16
I7 4-16
I5 4-16
I4 4-17
I2 4-17
I1 4-17
Indexing (1400) 4-17
Indicators, Byte 0 and Byte 1 6-1
Indicators, CPU Status 6-1
Indicators or Displays 6-1
Indicators (Operator Control Section) 6-6
Load 6-6
Manual (MAN) 6-6
System (SYS) 6-6
Test 6-6
Wait 6-6
Initialization, Compatibility (1400) 4-2
Input Data Code (PR-KB) 6-13
Insert Storage Key (Machine Language Instruction) 4-37
Insert Storage Key (Microword Statement) 4-49
Integrated I/O and Channels 1-6
Interface Control Check Logout 6-50
Interval Timer 4-50
50-Cycle Operation 4-52
60-Cycle Operation 4-51
Interval Timer Switch (INT TIMER) 6-7
Intervention Required (PR-KB) 6-31
INTVTN REQD, Intervention Required (PR-KB) 6-45
I/O Instruction Routine (PR-KB) 6-30
I/O, Integrated 1-6
I/O Operations (1400) 4-17

Keys (PR-KB) 6-47
Alternate Coding (ALTN CODING) 6-47
Cancel 6-47
End-of-Block (EOB) 6-47
Not Ready 6-47
PR-KB Alter/Display (on CPU Console) 6-47
Ready 6-47
Request 6-47
Keys and Indicators (PR-KB) 6-45

Lamp Test 6-6
Link Address 3-1
Local Storage 2-28
Local Storage (1400) 4-13
Local Storage Address Assembler 1-6,2-31
Local Storage, Alter/Display 6-13
Local Storage Data Assembler 1-5, 2-32
Logout (PR-KB) 6-50
Channel Control Check 6-50
Interface Control Check 6-50
Machine Check 6-50
Low-Digit 3-1
LOW TEMP (Low Temperature Light) 6-4

Machine Characteristics A-1
Machine-Check Latch 2-34
Machine-Check Logout (PR-KB) 6-50
Magnetic Tape Operation (1400) 4-28
Main Storage 1-3
Marginal Checking 5-5
Match Circuitry (STP0=STP1) 4-44
Metering Switch 6-11
Microlisting 3-1

Microlistings 3-3
Microprogram 3-1
Microprogram Concepts 3-1
Microprogram Functions 3-3
MMSK Register 2-17
MODE Register 2-17
Mode Switch 6-8
Modifier, ALU 2-26
Move/Arithmetic Word 3-23
M-Register 1-4, 2-11

No-Operation Command (PR-KB) 6-31

Op-Code Conversion and Recognition (1400) 4-2
Operator Control Section 6-5
Output Data Code--EBCDIC (PR-KB) 6-14
Overcurrent Detection 5-4
Overvoltage Detection 5-4

Parity Check (Storage Protection) 4-44
PCH (Punch System Check Indicator) 6-3
Power Distribution 5-5
Power-Off Key 5-2
Power-Off Sequence 5-2
Power-On Sequence 5-2
Power Supplies and Control 5-1
Power Supply and Temperature Compensation 2-9
Priority 2-12
    Priority Level 0 2-14
    Priority Level 1 2-13
    Priority Level 2 2-13
    Priority Level 3 2-13
    Priority Level 4 2-13
Printer Keyboard 6-18
PR-KB Alter/Display 6-7, 6-47
PR-KB Alter/Display Microprogram Routine 6-50
PR-KB Alter/Display Facility, Description and Operation 6-47
PR-KB Auxiliary Storage Areas 6-19
PR-KB Request Routine 6-39
PR-KB Translate Table 6-22
PR-KB Unit Control Word (UCW) 6-19
PROCD, Proceed (PR-KB) 6-45
Program Addressing (1400) 4-4
Program Storage 1-3
Protection Exception 4-34
Protection Key 4-34
PRT (Printer System Check Indicator) 6-3
PSW Restart 6-7
PWR CHK (Power Check Light) 6-04

Q-Register, Loading 4-43

RDR (Reader System Check Indicator) 6-3
Read Command (PR-KB) 6-31
    Initial Selection 6-34
    Keyboard Shift 6-35
Read Operation Ending Procedure (PR-KB) 6-41
Read Share Request (PR-KB) 6-40
Read/Write Operation, Sector Mode, Disk (1400) 4-26
Read/Write Operations with Indelible Address, Disk (1400) 4-25
Request Key-Attention Status (PR-KB) 6-29

Seek Command, Disk (1400) 4-25
Sense Byte (PR-KB) 6-19
Sense Command (PR-KB) 6-31
Sense/Inhibit, Common 2-5
Set IC 6-7
Set/Reset Word 3-5
Set Storage Key (Machine Language Instruction) 4-37
Special Circuits B-1
S-Register 2-18
Start I/O (PR-KB) 6-30
Start Key 6-7
Stop 6-6
Storage Address Assembler 1-5
Storage Address Check 2-33
Storage Addressing 2-5
Storage Clocks 2-5
Storage Data Check 2-33
Storage Data Assembler 2-9
Storage Data Register 2-9
Storage Key 4-32
Storage Protect Key, Displaying 4-37
Storage Protection 4-30
Storage Protection Check 2-33
Storage Protection, Setting Up 4-35
Storage Word 3-17
Storage Wrap Detection Circuitry (1400) 4-15
Store 6-6
STP1 Addressing Structure 4-38
STP1, Loading 4-38
STP1, Map 4-38
Switches and Keys 6-5
    Emergency Pull Switch 6-5
    Interrupt Key 6-5
    Load Key 6-5
    Load-Unit Switches 6-5
    Power-Off Key 6-5
    Power-On Key 6-5
System Checks (Indicators) 6-3
    2560 6-4
    Channel (CHNL) 6-4
    File 6-4
    Low Temperature (LOW TEMP) 6-4
    Power Check (PWR CHK) 6-4
    Printer (PRT) 6-3
    Punch (PCH) 6-3
    Reader (RDR) 6-3
System Control 1-1
System Control Panel 6-1
System Reset (Key) 6-6

Test I/O Command (PR-KB) 6-44
Thermal Sensing 5-5
Tilt/Rotate Printing Code 6-16
Tilt/Rotate Translator 6-23
True/Complement Controls 2-24
Trap Address 3-1
Traps 2-12

UCW, PR-KB 6-19

Voltage Sensing System 5-4

Wordmarks and Zone Bits in Control Fields, Disk (1400) 4-26
W-Register 1-5, 2-11
Write Command (PR-KB) 6-36
    Initial Selection 6-38

Write Command (PR-KB) 6-36
Initial Selection 6-38
Write Operation Ending Procedure
(PR-KB) 6-24
Write Share Request (PR-KB) 6-42

X- and Y-Drive Gate and Selection 2-8
X-Decode Drive 2-8
X-Read Decode Drive 2-8
X-Write Decode Drive 2-8
XY-Decode Drive (Auxiliary Storage) 2-9

Y-Decode Drive 2-8
Y-Read Decode Drive 2-8
Y-Write Decode Drive 2-8

Zero Test (Protection Key 0) 4-44
Zone 0 (1400) 4-13
Zone 1 (1400) 4-14
Zone 4 (1400) 4-14
Zone 6 (1400) 4-14
Zone 7 (1400) 4-14
Zone 0, Local Storage 2-30
Zone 1, Local Storage 2-30
Zone 4, Local Storage 2-30
Zone 5, Local Storage 2-31
Zone 6, Local Storage 2-31
Zone 7, Local Storage 2-31

READER'S COMMENT FORM

IBM 2025 Processing Unit FETOM Y24-3527-0

● How did you use this publication?

As a reference source ☐

As a classroom text ☐

As ..... ☐

● Based on your own experience, rate this publication . . .

As a reference source:	.... Very Good	.... Good	.... Fair	.... Poor	.... Very Poor
As a text:	.... Very Good	.... Good	.... Fair	.... Poor	.... Very Poor

● What is your occupation? .....

● We would appreciate your other comments; please give specific page and line references where appropriate.

● Thank you for your cooperation. No postage necessary if mailed in U.S.A.

YOUR COMMENTS, PLEASE ....

Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM branch office serving your locality.

Cut Along Line

Fold Fold

FIRST CLASS
PERMIT NO. 170
ENDICOTT, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation
P. O. Box 6
Endicott, N. Y. 13760

Attention: Product Publications, Dept. 171

Fold Fold

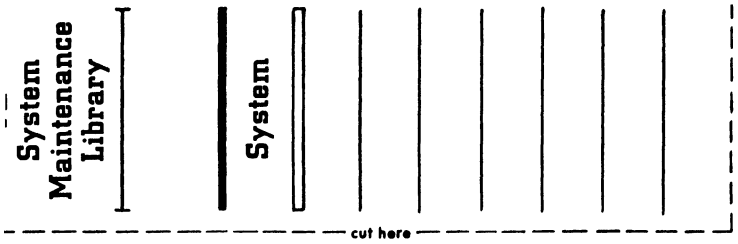
IBM

International Business Machines Corporation

Field Engineering Division

112 East Post Road White Plains, N. Y. 10601

Additional Comments:



Y24-3527-0

Printed in U.S.A. Y24-3527-0

IBM®
International Business Machines Corporation
Field Engineering Division
112 East Post Road, White Plains, N. Y. 10601