GA33-0013-3

3725 Communication Controller

Principles of Operation

System/370, 30xx, and 4300 Processors

IBM

3725
Communication Controller

Principles of Operation

System/370, 30xx, and 4300 Processors

Publication Number
GA33-0013-3

File Number
S/370/30xx/4300-09

Fourth Edition (January 1984)

This is a major revision of GA33-0013-2, which is now obsolete. Changes to the text or to an illustration are indicated by a vertical line to the left of the change.

Reference in this publication to IBM products, programs, or services does not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

The information contained in this manual is subject to change from time to time. Any such changes will be reported in subsequent revisions or Technical Newsletters.

Publications are not stocked at the addresses given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to either of the following:

- International Business Machines Corporation, Department 812LG, 1133 Westchester Avenue, White Plains, New York 10604, USA
- IBM France, Centre d'Etudes et Recherches, Department 0798, 06610 La Gaude, France

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

PREFACE

This publication is intended to help users to write a control program or modify an IBM supplied Network Control Program or Emulation Program. The reader should have an understanding of basic data communication and a thorough knowledge of System/370 channel operations. The Introduction to the IBM 3725 Model 1 Communication Controller (GA33-0010), or the Introduction to the IBM 3725 Model 2 Communication Controller (GA33-0021) is a prerequisite publication. A related publication is the IBM 3725 Communication Controller Operating Guide (GA33-0014).

This manual is divided into five chapters and seven appendixes.

Chapter 1. Introduction

Provides a general description of the 3725 and its functional units.

Chapter 2. Structure of the 3725

Describes the system structure, the storage scheme, the registers, the interrupt scheme, and the interrupt levels assigned to each adapter.

Chapter 3. Instruction Set

Describes each of the 3725 instructions with their mnemonics, format, and condition codes.

Chapter 4. Central Control Unit (CCU)

Describes the operation and programming requirements of the Central Control Unit.

Chapter 5. Channel Adapter

Describes the operation and programming requirements of the Channel Adapter and how it communicates with the CCU.

Appendix A. CCU External Registers

Provides summary information on the CCU registers in convenient tabular form.

Appendix B. CA Input/Output Instruction Summary Charts

Provides summary information on the channel adapter input/output instructions in convenient tabular form.

Appendix C. Communication Scanner Commands

Provides summary information on the communication scanner commands in convenient tabular form.

Appendix D. MOSS Commands

Provides summary information on the MOSS commands in convenient tabular form.

Appendix E. Redrive Logic

Provides information on the use of the redrive logic.

Appendix F. Initial Program Load

Provides information on initial program loading (IPL).

Appendix G. Branch Trace

Provides information on branch tracing operations.

CONTENTS

CHAPTER 1. INTRODUCTION

CHARACTERISTICS OF THE 3725

The IBM 3725 Communication Controller belongs to the same family as the IBM 3704 and 3705 Communications Controllers, but differs from them in having a faster instruction execution time, larger storage capacity, improved flexibility, enhanced functions, and higher throughput.

These improvements are due to a new architecture based on the principle of 'distributed intelligence', and made possible by the technological advances of large scale integration (LSI). Instead of one main processor executing all machine functions, many of the more dedicated functions are executed by microprocessors distributed throughout the machine. In this way, each of the communication scanners in the 3725 has its own microprocessor, and the Maintenance and Operator Subsystem (MOSS) has another. The user does not have to program the communication scanners and the MOSS. A diskette unit which forms part of the MOSS contains the microcode for these two devices; it is loaded automatically at IPL time.

The MOSS uses a display console which replaces all the control panel functions of older machines, but has the advantage of being able to display simultaneously much larger amounts of information. This facility makes it much simpler for the operator to control the 3725 and its attached network.

There are also many advantages to the programmer, as for example:

- Display large storage areas and all register contents at the same time.
- Change storage and registers.
- Trace program flow.
- Stop on selected instruction or data addresses.

3725 SIMPLIFIED BLOCK DIAGRAM

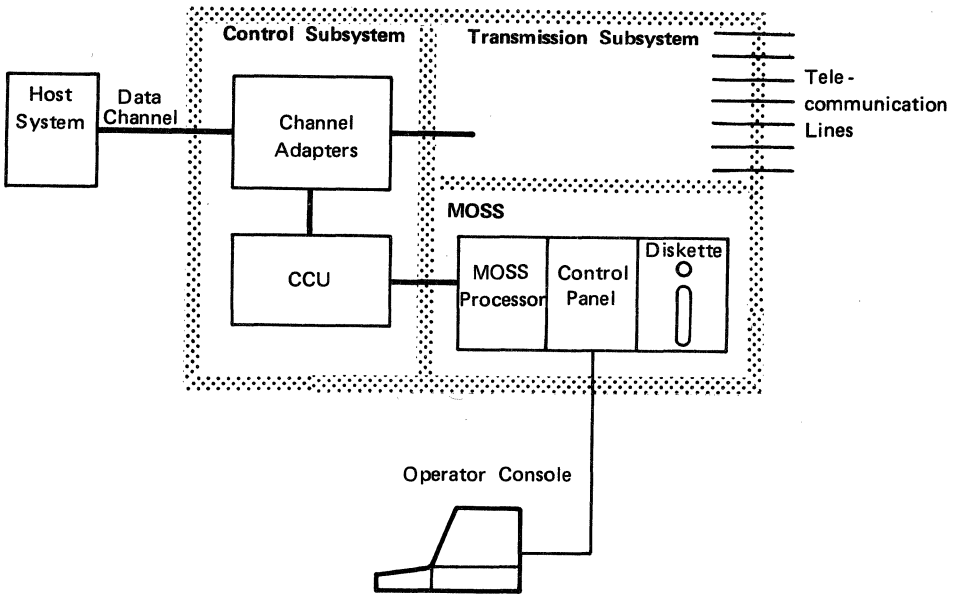


Figure 1-1. 3725 Block Diagram

Figure 1-1 shows a simplified block diagram of the 3725. The 3725 comprises:

- A control subsystem, consisting of a central control unit (CCU) with from 512 kilobytes to 2 megabytes of associated storage, and, for channel-attached controllers, one or more channel adapters.
- A transmission subsystem, consisting of one or more communication scanners and the associated line coupling hardware.
- A maintenance and operator subsystem (MOSS), with its diskette drive, control panel, and associated IBM 3727 Operator Console.

SYSTEM PROGRAMS

For scheduling and controlling the resources of the 3725, IBM provides a licensed program product, ACF/NCP (Advanced Communications Function/Network Control Program) Version 2.

IBM also provides a set of system support programs, called ACF/SSP (Advanced Communications Function/System Support Programs). These are host processor programs used primarily to generate or assemble a user's control program, and to provide IPL and dump facilities for the controller.

Also available is an IBM licensed program called the Emulation Program (EP). The emulation program can only run in a controller attached directly to a channel of the host processor. The program emulates most of the functions of the IBM 2701 Data Adapter Unit, the IBM 2702 Transmission Control, and the IBM 2703 Transmission Control.

CONTROL SUBSYSTEM

CENTRAL CONTROL UNIT (CCU)

The Central Control Unit (CCU) contains the circuits and data flow paths to execute the instruction set, and to control the storage, the channel adapters, and the communication scanners. The CCU is interrupt driven, and can operate at five different interrupt levels under the control of the resident control program.

STORAGE

The 3725 includes from 512 kilobytes to 2 megabytes of main storage by increments of 256 kilobytes. This storage is used to contain the control program. It is also used as a temporary storage area for data as it is being assembled or disassembled.

An advanced storage error checking system is used which detects and corrects all single bit errors, and detects all double bit errors.

Storage Protection

The storage protection mechanism divides storage into blocks of 2048 bytes, each block having a separate storage key. Storage writing and instruction execution is allowed only if the user key matches the storage key (a user being defined as a program level or a cycle stealing device). Storage protection is valid at all program levels and for all adapters.

In addition, a read only key is used on the 3725 to protect certain areas of storage. Again, the storage is divided into blocks of 2048 bytes, each block having a single bit key indicating whether the storage area is read/write, or read only.

CHANNEL ADAPTER (CA)

Unlike previous communication controllers, which used several different types of channel adapter, the 3725 has only one type of channel adapter. It adapts itself to the different types of channel (byte multiplex, block multiplex, selector).

The 3725 may be equipped with a maximum of 6 channel adapters, or up to 4 channel adapters equipped with the two-processor switch, or any mix of the two, provided that the total number of channel connections used does not exceed eight.

Model 2 only

The 3725 Model 2 is limited to a maximum of 2 channel adapters.

Two-Processor Switch (TPS)

The channel adapter may be fitted with a two-processor switch which allows it to be attached to two channels at once. There are two possibilities:

Tightly Coupled Processors

For tightly coupled multiprocessors, the channel adapter may communicate with both interfaces alternately (but not simultaneously) using the native sub-channel. The 3725 appears as the same I/O unit to the hosts, and can be accessed by both interfaces in exactly the same manner. This facility allows the access methods for the 3725 to run on either processor, the path to the 3725 being transparent to the access method.

Independent Host Processors

It is also possible to connect two independent host processors to a 3725 via the two-processor switch. In this case, the two processors may not access the 3725 alternately, but only after manual switching between the channels via the 3725 control panel.

Model 2 only

The 3725 Model 2 cannot be equipped with two-processor switches.

TRANSMISSION SUBSYSTEM (TSS)

The transmission subsystem consists of:

- The line attachment bases (up to 8), each containing 1 or 2 communication scanners (see notes below for 3725 Model 2 limitations).

- The communication scanners.
- The line interface couplers (up to 8 per scanner, depending on the configuration of the subsystem). See the notes below for 3725 Model 2 limitations.
- Interface clocking

The transmission subsystem does not include modems, as these are completely external to the 3725.

LINE ATTACHMENT BASE (LAB)

The 3725 may contain up to 8 line attachment bases, each of which contains 1 or 2 communication scanners. Each LAB supports a maximum of 8 line interface couplers (LICs), and each LIC supports up to 4 lines, giving a total of 32 lines per LAB. There are two types of line attachment base:

1. Line Attachment Base Type A.

   The line attachment base type A (LAB A) contains a single communication scanner, which therefore supports up to 32 lines. LAB A is normally used to support a relatively large number of low- and medium-speed lines.

2. Line Attachment Base Type B.

   The line attachment base type B (LAB B) contains two communication scanners, each of which therefore supports up to 16 lines. LAB B is normally used to support a relatively small number of medium- and high-speed lines.

Whatever type of line attachment base is used, the maximum number of lines that can be attached to the 3725 is 256, that is, 8 LABs supporting a maximum of 32 lines each: 8×32 = 256.

Notes:

1. The first two installed LABs are always LAB Type A, so that the number of installed scanners cannot exceed 14.
2. The figure of 256 lines given above is a maximum. The actual number of lines that may be connected to each line attachment base depends on the type and the speed of the lines that it supports.

   Notes for Model 2 only

3. The 3725 Model 2 can be equipped with one line attachment base only (type A) containing a single communication scanner.

4. A maximum of six line interface couplers may be attached to the unique communication scanner; the maximum number of attached lines is therefore 24.

COMMUNICATION SCANNER

Only one type of communication scanner is used in the 3725. Each scanner has its own microprocessor and associated microcode, providing increased throughput and improved flexibility. As noted above, each scanner may support up to 32 lines if it is installed in a LAB type A, and up to 16 lines if it is installed in a LAB type B. The scanner provides the connection between the line interface coupler hardware and the central control unit. The main functions of the communication scanner are:

- Execute data link control (DLC) functions
- Provide serializer/deserializer facilities
- Provide buffering facilities
- Provide the correct interfaces for the different line types and line speeds (except for the hardware interface, which is done by the line interface coupler).
- Recognize certain control characters (depending on the type of line).
- Provide control character insertion/deletion (depending on the type of line).
- Provide code translation for BSC lines. The scanner translates from ASCII to EBCDIC when receiving from the line, and from EBCDIC to ASCII when transmitting to the line.

LINE INTERFACE COUPLERS (LIC)

The line interface couplers (LICs) provide the functions necessary to interface the communication scanner with the various external facilities such as modems and other data circuit terminating equipment (DCE), and with direct attached terminals. There are only five types of line interface coupler. They provide the same functions as the line sets of the 370X Communications Controllers. A LIC may connect from 1 to 4 lines, depending on the type of coupler. Different types of LIC may be mixed on the same LAB.

The characteristics of the five types of LIC are summarized below:

LIC	Lines	Interface	Protocol	Speed
1	4	V24/V25/RS-232C/RS-366	SDLC (DX) BSC Start/Stop	19.2 Kbps max 19.2 Kbps max 9.6 Kbps max
2	1	Wide band data station US 5703/8803/5701/8801	SDLC (DX) BSC	230.4 Kbps max 64.0 Kbps max
3	1	V35	SDLC (DX) BSC	256.0 Kbps max 64.0 Kbps max
4A	4	X.21 (medium speed)	SDLC (DX)	9.6 Kbps max
4B	1	X.21 (high speed)	SDLC (DX)	256.0 Kbps max

DX = duplex

INTERFACE CLOCKING

Business machine clocking is available by groups of 16 lines, with a maximum of two groups (32 lines) to each communication scanner. The lines may be full- or half-duplex, at 50, 110, 134.5, 200, 300, 600, and 1200 bps.

Business machine clocking also supports direct attachments at 2400, 4800, 9600, 19200, and 56000 bps.

Direct Attachment

Low-Speed Direct Attachment

Low-speed direct attachment is available for the 3725 using the LIC 1 with business machine clock rates of 2400, 4800, 9600 or 19200 bps.

High-Speed Direct Attachment

High-speed direct attachment is available for the 3725 using the LIC 3 with business machine clock rates of 19.2 or 56 kbps. The high-speed direct attachment allows communication without modems over a half-duplex line connecting two 3725s (or a 3725 and a 3705).

X.21 Direct Attachment

X.21 direct attachment is available for the 3725 via:

1. LIC 4A with business machine clock rates of 2400, 4800, or 9600 bps.

2. LIC 4B with business machine clock rates up to 56 kbps.

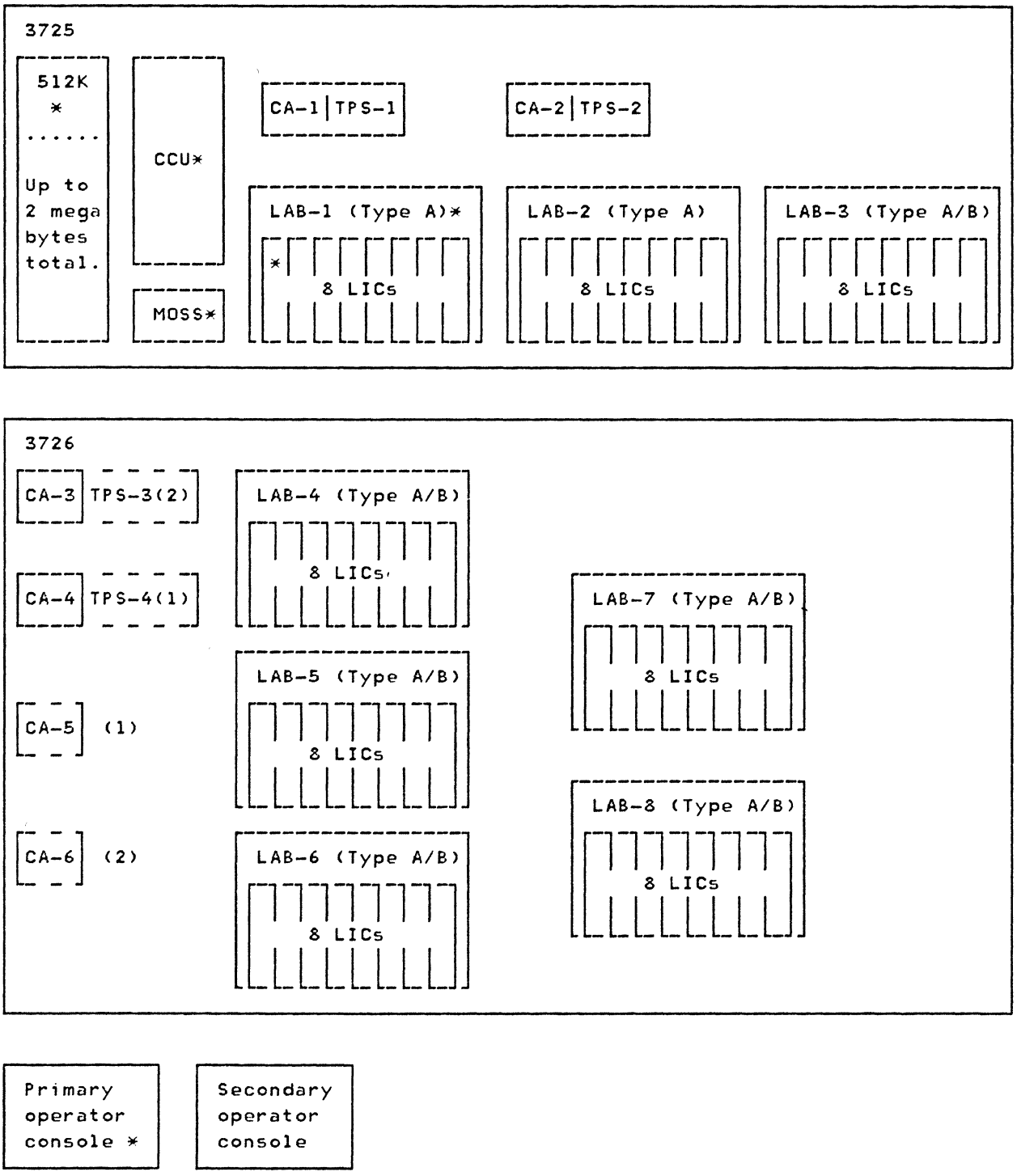
MAINTENANCE AND OPERATOR SUBSYSTEM (MOSS)

The maintenance and operator subsystem (MOSS) is used for loading and supervising the 3725, for running the problem determination procedures, and for program and hardware maintenance. It continually monitors the operation of the 3725, compiling and storing error data, executing recovery routines, and issuing alarm messages. It includes the following features:

- An independent processor called the MOSS processor
- The MOSS microcode.
- An attachment for the IBM 3727 Operator Console. If required, the attachment may be switched to an optional remote 3727 Operator Console. This remote terminal may be located at up to 150 metres (500 feet) from the communication controller.
- A magnetic diskette unit
- A control panel

PHYSICAL CHARACTERISTICS

The figure below shows the minimum and maximum configurations of the 3725:



Notes:

1. CA-5 and TPS-4 are mutually exclusive.

2. CA-6 and TPS-3 are mutually exclusive.

3. * Denotes minimum configuration. If the 3725 is channel-attached, at least one channel adapter is required.

4. 3725 Model 2 configuration is the same as the minimum configuration as indicated by the previous note, with the following exceptions:

   a. LAB-1 contains a maximum of 6 LICs, so that the maximum number of attached lines is 24.

   b. The maximum number of attached channel adapters is 2.

   c. Channel adapters cannot be equipped with the two-processor switch.

GENERAL PROGRAMMING CONCEPTS

The control program, resident in 3725 storage, controls the transfer of data as it passes through the controller between the stations in the network and the host processor.

Communication between the channel adapter, the communication scanner, and the control program is by means of interrupts, I/O instructions, and cycle stealing.

Data entered at a terminal is received by the line interface coupler (LIC). The communication scanner recognizes that service is required and receives the data from the terminal via the LIC, afterwards transmitting it to the CCU. The control program places the data in storage, where it is available for retransmission.

When data is to be transmitted from the host processor to a terminal, the process is reversed. The host processor channel sends the data to the 3725 channel adapter, which places the data in storage under program control. An interrupt is then signaled to the control program, which sends the data via the communication scanner and the LIC to the terminal.

3725 INPUT/OUTPUT INSTRUCTIONS

The control program in the 3725 uses input/output instructions as the primary link between the program and the 3725 hardware. The input/output instructions are of four types:

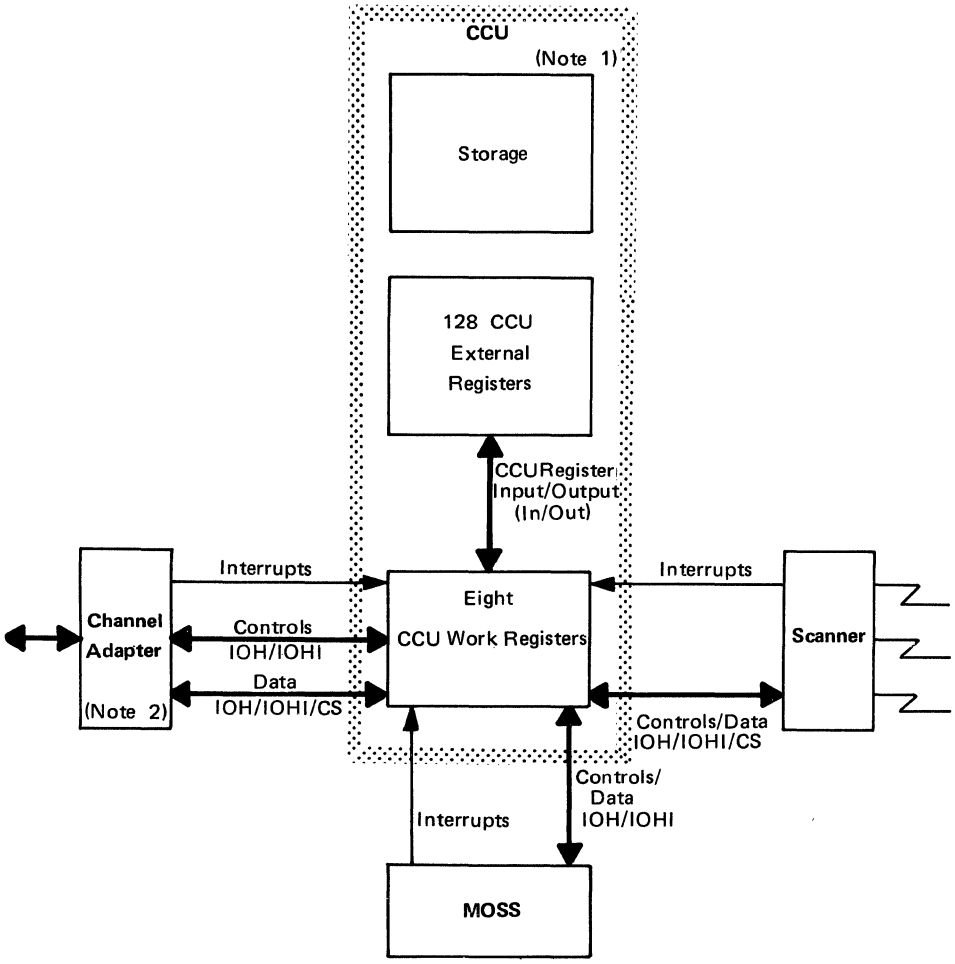
- CCU register input

- CCU register output
- Adapter input/output (IOH)
- Adapter input/output immediate (IOHI)

The first two of these instructions are for use within the CCU and for communication with the MOSS. The two others are for use with the channel adapter and the communication scanners. In the case of IOH and IOHI instructions, a bit in the instruction determines whether a particular input/output operation is input or output.

CHAPTER 2. STRUCTURE OF THE 3725

This chapter describes the structure, the storage addressing scheme, the registers, the interrupt system, and the program levels used in the 3725. The user needs a thorough understanding of these facilities in order to program the controller efficiently.



Notes:
1. The CCU is treated in detail in Chapter 4.
2. The channel adapter is treated in detail in Chapter 5.

Figure 2-1. System Structure

Figure 2-1 shows the structure of the 3725. Only one channel adapter and one communication scanner are shown on this figure.

STORAGE

Byte locations in storage are numbered consecutively starting with 0; the address of a byte is the same as its numbered position. A group of bytes in storage is addressed by the high order byte of the group, the number of bytes in the group being either implied, or explicitly defined by the operation.

STORAGE ADDRESSING SCHEME

The 3725 storage addressing scheme uses a 22 bit address, of which the high order bit must be zero. The resulting 21-bit binary address (contained in three bytes) can thus address up to 2 megabytes.

Note to Model 2

The maximum memory size in the 3725 Model 2 is 512 K.

Note: in the remainder of this manual, addresses and registers capable of holding 22 bits are considered to be 3 bytes long.

The three bytes are called (from high order to low order) byte X, byte 0, and byte 1 as shown below:

Byte X	Byte 0	Byte 1
2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7

high order byte — low order byte

Byte X is also called the 'extension byte'; bit 2 of this byte must be zero.

All the general registers, and all registers involved in storage addressing are structured in this way. Some of them (the High Resolution Timer/Utilization Counter, for example) use all 22 bits. All three bytes of the address form an integral part of the register regardless of the address being operated on. There are two exceptions only to this rule:

- Byte X is ignored for output (write) instructions that do not address storage. Its bits may be set to either 0 or 1.
- Byte X is set to all zeros for input (read) instructions that do not address storage.

Notes:

1. If an address is used which is less than the storage wrap point (4194303 bytes), but greater than the number of installed storage positions, an Address Exception Check occurs, and causes a level 1 interrupt or a CCU

hard check. An Address Exception Check is also raised if a storage position is addressed for which the corresponding address exception key is 0.

2. Storage addressing wraps at 4194303. If an address as calculated exceeds this figure, then the true storage position addressed is the calculated address, minus 4194303. An Address Exception Check is raised if storage protection is enabled.

STORAGE BOUNDARIES

Instructions and half/fullword operands must be located on integral halfword boundaries in storage; the address of a half/fullword must be a multiple of 2.

Storage addresses are expressed in binary form. Thus an integral address for a half/fullword must have the last binary digit equal to 0.

STORAGE PROTECTION

The storage protection circuits contain three separate mechanisms:

Storage Protection by User Protect Key

The word user in this context means a program level or a cycle stealing device.

Storage is divided into blocks of 2048 bytes, each block of storage being associated with a three-bit storage key, located in a special store. Each time that an attempt is made to write a storage position, or execute an instruction, the storage key is read out of the corresponding position of the special store. The storage keys are set up by program.

In the same way, each user is assigned a three-bit protect key, located in a hardware register. This three-bit key is active whenever the corresponding program level or cycle-stealing device is active. The user protect keys are set up by program.

Whenever a storage location is addressed, the storage key is read out and compared with the user's protect key. Writing in the storage location or instruction fetching is allowed only if the two keys match. There are slight differences of operation depending on whether the storage position being addressed contains an instruction (instruction fetch phase), or data (execute phase).

If the storage location being addressed contains an instruction, the keys must match absolutely. If the two keys do not match, a storage protect exception level 1 interrupt is set.

If the storage position being addressed contains data, the keys must satisfy one of the following conditions:

- The two keys are equal.
- The storage key is X'7'. This bit combination signifies that the storage location is unprotected (for data only).
- The user protect key is X'0'. This bit combination allows the user to operate anywhere in storage (for data only).

If none of these conditions are filled, a storage exception level 1 interrupt is set.

Notes:

1. When an EXIT instruction is executed in program levels 1 through 4, the user key is set to zero for that level.
2. When an EXIT instruction is executed in program level 5, the user key remains unchanged.
3. If an Output X'73' instruction is executed to alter the user key for the current program level, a branch instruction must follow immediately. This branch instruction must cause a branch to a storage area having its storage key equal to the new user key.
4. If an Output X'73' instruction is executed which sets the user key for a block containing an instruction that would otherwise be among the next 4 to be executed, a branch instruction must be executed immediately. This branch instruction must cause a branch to a storage area having its storage key equal to the new user key.
5. Changing or setting the user key, storage key, read-only bit, or address exception bit via an Output X'73' instruction can only be done from a zone having the user key set to zero. If the user key is not zero, the instruction is executed, but the key is not altered.

Read-Only Protection

Each 2K block of storage also has a read-only bit. If this bit is set to 1, the block of storage may not be written, but only read. Any attempt to write in a read-only protected area causes a Level 1 interrupt.

Addressing Exception Protection

Each 4K block of storage has a bit which indicates whether the block of storage is physically present or not. Any attempt to read or write in a non-existent storage location causes a Level 1 address exception interrupt.

CENTRAL CONTROL UNIT (CCU)

CCU REGISTERS

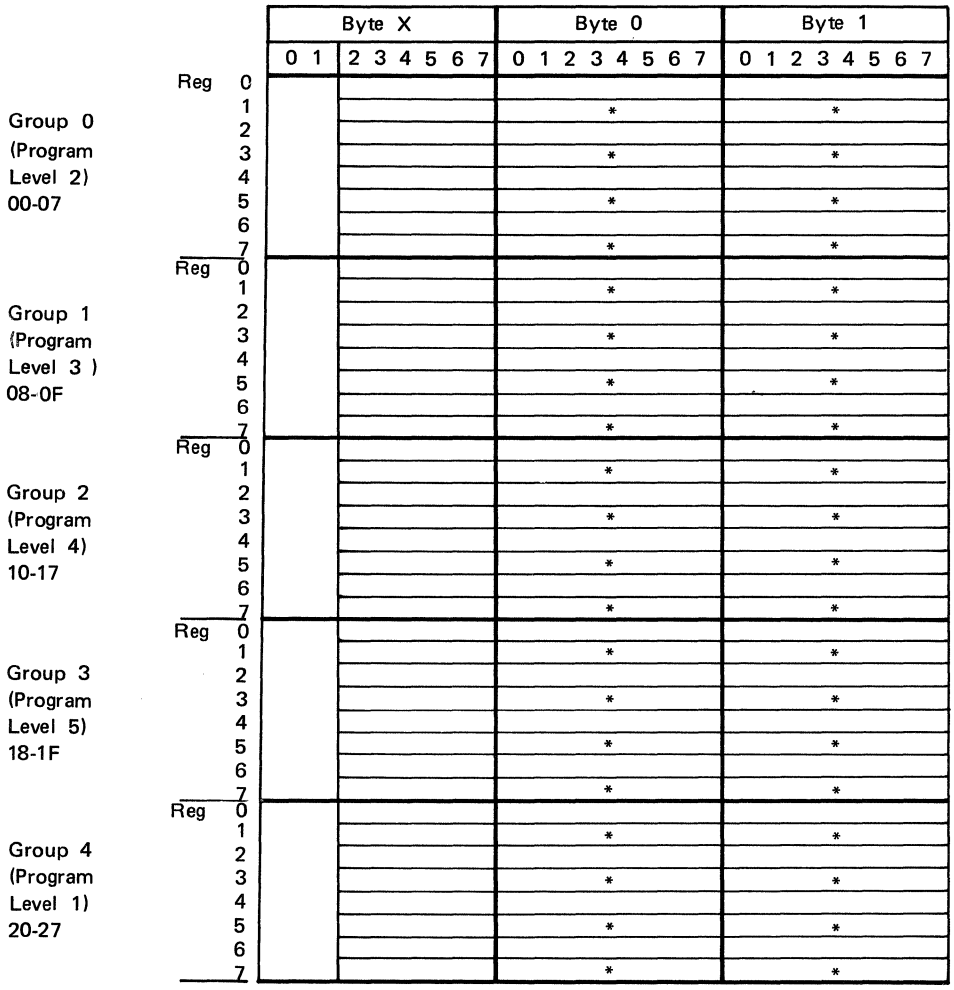
All the CCU registers are three bytes long, but the third (high order) byte is used only when the register in question contains an address; the third byte (also called byte X or the 'extension' byte) is then used for storing the high order bits of the address. The CCU has two types of register:

- The 8 CCU work registers, which are accessible by the program directly
- The 128 CCU external registers, which are accessible to the program indirectly by means of input/output instructions. Most of these registers are located in high-speed local storage, and can be both written and read by the CCU program. Certain registers, however, are implemented in hardware. Some of these registers are double registers; that is one register can only be written by the program, the other can only be read. This is the case of register X'72' for example, where Input X'72' means 'load a general purpose register with the contents of the operator function select control'. Output X'72', on the other hand, means 'set the program display register with the contents of a general purpose register'.

The function of each CCU register is described in detail in Chapter 4.

CCU General Registers

The first 40 external registers (X'00' through X'27') are called general registers. They are located in local storage and are addressed as external I/O registers using the 'Input' and 'Output' instructions. The general registers have specific functions as shown in the figure below:



* Indicates selectable bytes of general registers

Figure 2-2 General Register Groups

The 40 general registers are divided into five groups, numbered 0 through 4, and each containing 8 registers. Within each group, the registers are numbered 0 through 7. Each group is assigned to a specific program level (see below under the heading Program Levels). When an interrupt level is exited (using the EXIT instruction) and the interrupted program level re-entered, the contents of the corresponding group of 8 local storage registers is transferred by hardware into the 8 work registers. As instructions are executed, the 8 local storage registers are updated with the contents of the work registers. This allows the control program working at one level to be interrupted

by a higher priority level without the need for saving registers. The figure below shows the relationship between the CCU work registers and the general registers.

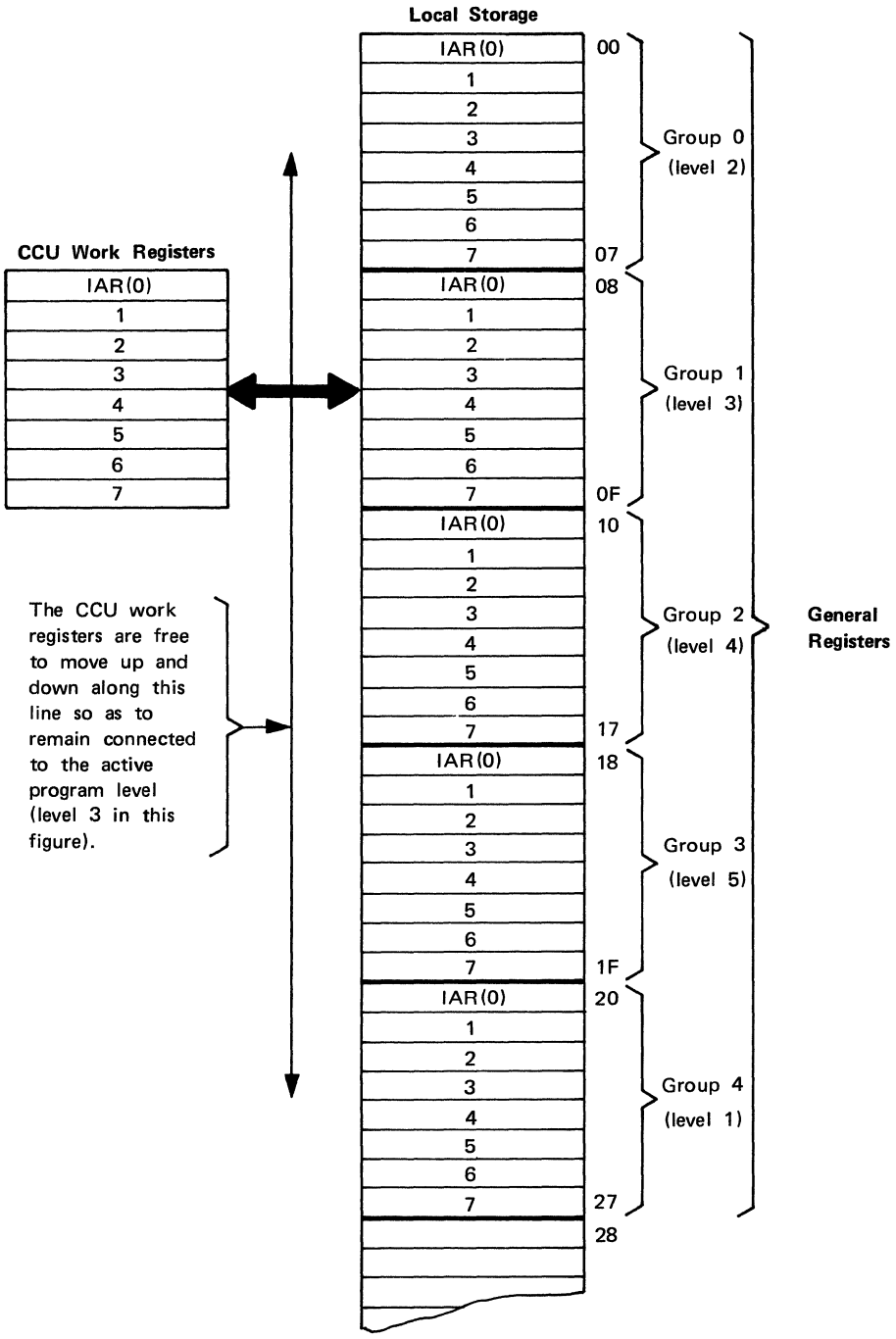


Figure 2-3. Work Registers and Local Storage

The general registers of the other (non-active) groups are considered as external registers by the active program level, which can access them in the normal way by means of CCU Register Input/Output instructions.

The first general register of each group (register 0) is used as the instruction address register (IAR) of the corresponding program level.

CONDITION LATCHES

Each program level has associated with it a pair of latches called the C and Z latches. They reflect the results of many of the instructions, and provide a means of branching on these results. Note, however, that some instructions do not change the C and Z latches; they stay set with their previous values.

- C (carry) usually indicates a carry out of the high order position of the register being operated on, but it can also mean 'less than' or 'not equal to' depending on the instruction.

- Z (zero) usually indicates that the result of the operation is zero, but it can also mean 'the two operands are equal'.

PROGRAM LEVELS

The 3725 hardware has five operational program levels. Each program level operates in a similar way to a subroutine, and is responsible for particular phases of 3725 operation.

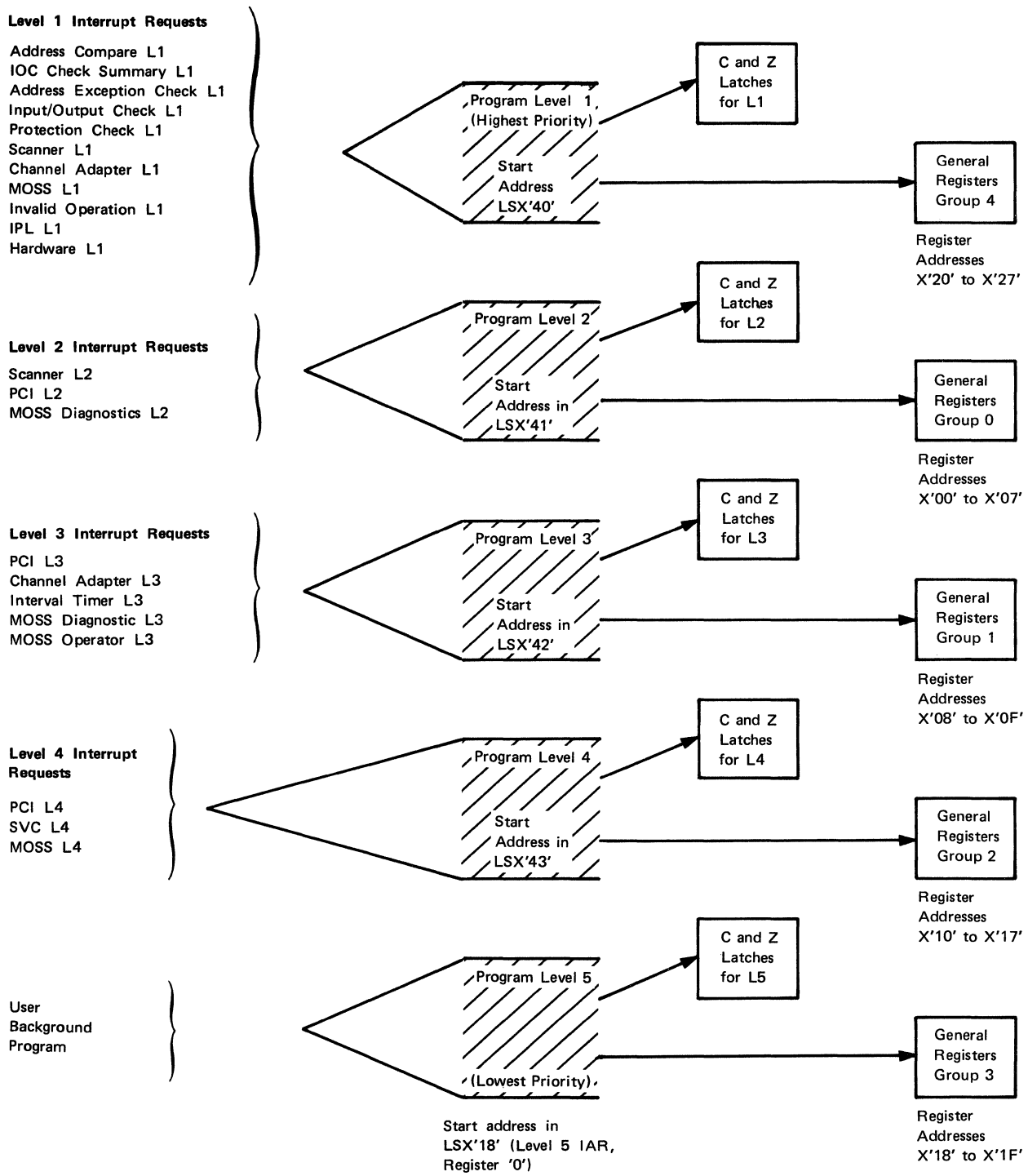


Figure 2-4. Program Levels

The organization of the different program levels is shown in figure 2-4. Each of the five program levels has a different priority; program level 1 has the highest priority, and program level 5 the lowest priority. Program levels 1 through 4 (referred to as the interrupt program levels) provide the connection between the hardware units and program level 5 (referred to as the background program level).

The functions assigned to the different program levels are as follows:

Background Program Level 5

This program level has the lowest priority level, and is only active when none of the other four levels requires service.

Program level 5 cannot interrupt any other program level. However, if program level 5 issues the 'exit' instruction, this generates a supervisor call interrupt at program level 4, and so allows level 5 to communicate with level 4.

Interrupt Program Level 4

The functions assigned to this level are:

- Program-controlled interrupt requests at level 4.
- Supervisor call (SVC) request (generated when the exit instruction is executed at level 5).
- MOSS interrupt level 4.

Interrupt Program Level 3

Level 3 is used for most of the interactions between the host processor and the channel adapter.

The functions assigned to this level are:

- Program controlled interrupts at level 3.
- Channel adapter interrupts
- Interval timer interrupts at 100 ms intervals.
- MOSS diagnostic interrupts at level 3.
- MOSS operator interrupt (corresponding to the interrupt button function).

Level 3 interrupts are less critically time-dependent than those assigned to level 2.

Interrupt Program Level 2

Level 2 is used for most of the interactions between the host processor and the communication scanner. The functions assigned to this level are:

- Communication scanner interrupts.
- Program controlled interrupts at level 2.
- MOSS diagnostic interrupts at level 2.

Interrupt Program Level 1

This is the highest priority program level. It is entirely hardware driven, and is used to service 'trouble' and other unusual conditions. Conditions that can cause a level 1 interrupt are:

- Address Compare interrupts
- IOC Check Summary
- Address Exception Check
- Input/Output Check
- Protection Check
- Communication Scanner Checks
- Channel Adapter Checks
- MOSS level 1 interrupts
- Invalid Operations Check
- IPL Check
- Hardware level 1 interrupt (if in the 'bypass CCU check' mode).

Communication scanner and channel adapter checks may be masked, but only if the CCU is in test mode.

The 3725 is an interrupt driven machine, operating in response to requests from the control program and from the hardware. Since these requests have varying degrees of urgency, a priority system is used. Each program, CCU, channel adapter, and communication scanner request is assigned a particular priority level. Any request for the use of the controller coming from either the control program or the hardware is called an interrupt request.

Each interrupt request is assigned to a program level. As we have seen above, the program levels are numbered from 1 to 5 and determine the priority structure. The priority level decreases as the program level number increases; level 1 has the highest priority, and level 5 the lowest.

The machine contains a mechanism that determines when, and in what order, interrupts may occur. If an interrupt request is allowed, the change from the current program level to the interrupting program level takes place immediately after completion of the current instruction. If several interrupt requests having different priorities are present at the same time, the one with the highest priority obtains the use of the controller. When a particular level is using the controller, it may be interrupted in its turn by a new interrupt request at a higher level.

When an interrupt occurs (after completion of the current instruction) instruction execution at the lower priority program level is suspended until execution at the higher level is complete. If a new interrupt request at the same level (or at a lower level) occurs, it is stacked until servicing of the current interrupt is terminated.

The 3725 will not allow a particular interrupt if any of the following conditions exist:

- A higher priority interrupt request is present
- The new interruption is at the same level as the one currently being processed
- The interrupt request or the program level to be interrupted is masked

At the moment that an interrupt is honored, a latch called the interrupt entered latch is set on. The 'interrupt entered' latch is a hardware latch that tells the 3725 that the associated program level has been entered. As long as this latch is on, no other interrupt requests to that level are honored. This prohibits interrupts at the same or at a lower level that could destroy essential information. The 'interrupt entered' latch is not reset when its program level is interrupted by a higher priority level. It can only be turned off by an 'exit' instruction at its own program level, or by a reset condition in the controller.

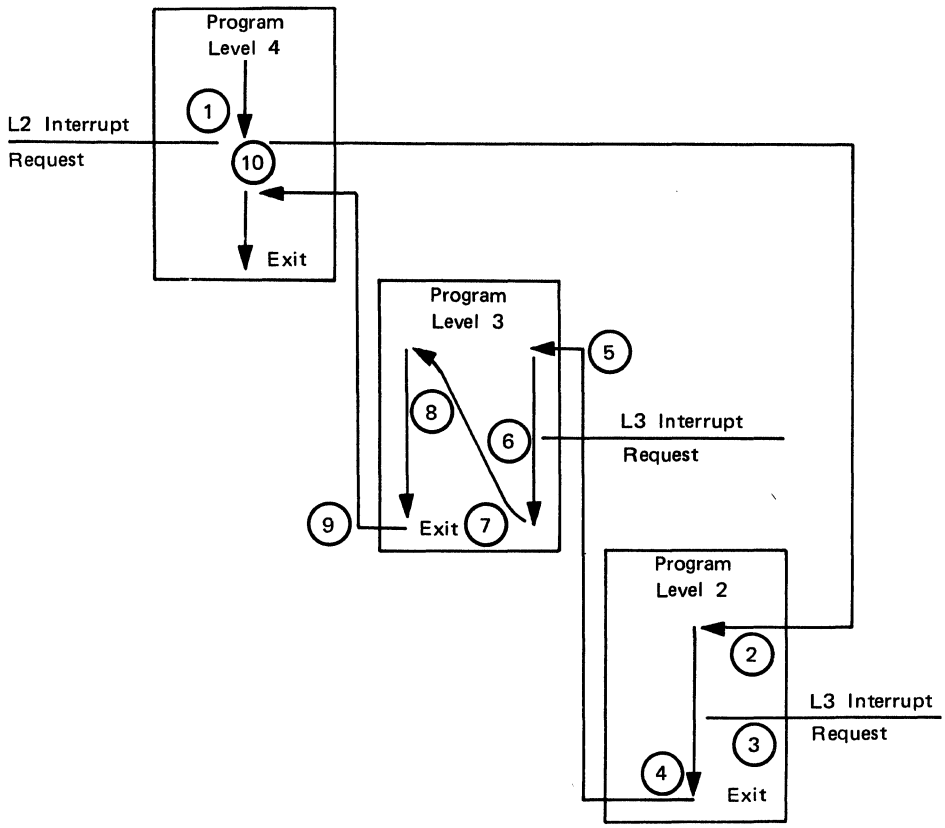


Figure 2-5. Interrupt Priority Example

Figure 2-5 shows an example of a sequence of interrupts. Assume that the program is being executed at level 4.

1. A level 2 interrupt request occurs.

2. The controller hardware sets the 'interrupt entered' latch of level 2, and forces a branch to the start address of program level 2, which starts running.

3. During the execution of the level 2 program, a level 3 interrupt request occurs. The interrupt is not honored, because the level 2 program being executed is of higher priority. The level 3 interrupt is not lost, however, but is stored temporarily.

4. The level 2 program signals that it has ended by means of an 'exit' instruction, which resets the level 2 'interrupt entered' latch.

5. The controller now allows the next highest interrupt level to be processed. The level 3 interrupt, previously noted, sets the level 3 'interrupt entered' latch and causes entry into the level 3 program at its starting address.

6. Another interrupt occurs at level 3 during the execution of this same level. This interrupt cannot be honored, as the same level is currently being executed. It is therefore stored temporarily as before.

7. The level 3 program signals its end by means of the 'exit' instruction, which resets the level 3 'interrupt entered' latch.

8. The pending level 3 interrupt sets again the level 3 'interrupt entered' latch, causing re-entry into the level 3 program at its starting address.

9. The level 3 program signals its end by means of the 'exit' instruction, which resets the level 3 'interrupt entered' latch.

10. No further interrupts are pending. However, the level 4 'interrupt entered' latch is still on, as the level 4 program has not run to completion (it has not yet signaled its end by means of the 'exit' instruction). The level 4 program therefore continues until it has finished its current task, and then executes the 'exit' instruction. The level 4 'interrupt entered' latch is turned off, and as no further interrupts are pending, the control program returns to background level 5.

There are times when it is not desirable to interrupt a program with a higher priority request. In these cases, a mask can be set to prevent interrupts to a particular program level.

When an interrupt occurs, instruction execution at that interrupt level begins with the instruction located at the starting address for that level. The starting addresses of the interrupt levels are contained in registers X'40' through X'43'. These assignments are shown below:

Level	Register
1	X'40'
2	X'41'
3	X'42'
4	X'43'

The instruction strings beginning at the addresses contained in these locations direct the control program to the correct routine to handle a particular interrupt.

Notes:

1. The interrupt starting addresses must be loaded into the registers at IPL time using Output instructions X'40' through X'43'.

2. Only the beginning of the interrupt routine is located at the start address; the remainder of the routine can be located anywhere in storage, and is reached by means of a branch instruction.

3. Level 5 is not entered by an interrupt; instruction execution begins at the address in the level 5 Instruction Address Register (register 0, Input X'18').

Some routines may be used by more than one program level. However, the execution of the routine always occurs at the priority level of the currently active program level.

When a program level has completed its interrupt servicing, it must execute an 'exit' instruction (always the last instruction of an interrupt routine). The exit instruction causes the 'interrupt entered' latch for that level to be reset, and allows control to be passed to the next higher priority program level requiring service.

Notes:

1. A program controlled interrupt (PCI) is available at levels 2, 3, and 4. Level 5 cannot generate a PCI directly, but only via the 'Exit' instruction as described in the next note.

2. When the 'Exit' instruction is executed at program level 5, the operation is modified, and a supervisor call interrupt to level 4 (SVC L4) is set. This is the only way in which program level 5 can generate an interrupt request.

CHANNEL ADAPTER (CA)

The channel adapter communicates with the CCU in three different ways:

1. By means of IOH and IOHI instructions. The IOH and IOHI instructions are used to move control information between the CCU and the channel adapter registers.

   These instructions may also be used by the control program to read and write data in the Programmed Input/Output (PIO) mode. As this mode of operation is slow compared to the Adapter Input/Output (AIO) method, it should normally be used only if throughput considerations allow.

2. By means of Cycle Stealing. Cycle stealing is used for the high speed transfer of data between the CCU and the channel adapter. Once the operation has been initialized by means of IOH or IOHI instructions, the operation continues without further intervention by the program until all the data has been transferred.

3. By means of interrupts. This is the method used by the channel adapter to obtain the attention of the control program. Channel adapter interrupts are at two different levels:

   a. At level 3 for normal interrupts.

   b. At level 1 for error interrupts.

The channel adapter and its connection with the CCU are described in detail in Chapter 5.

COMMUNICATION SCANNER

The communication scanner interacts with the CCU in three different ways:

1. By means of IOH and IOHI instructions. The IOH and IOHI instructions are used to move control information between the CCU and the communication scanner registers.

2. By means of Cycle Stealing. Cycle stealing is used for the high speed transfer of control information and data between the CCU and the communication scanner. Once the operation has been initialized by means of IOH or IOHI instructions, the operation continues without further intervention by the program until all the control information and data has been transferred.

3. By means of interrupts. This is the method used by the communication scanner to obtain the attention of the control program. Communication scanner interrupts are at two different levels:

   a. At level 2 for normal interrupts.

   b. At level 1 for error interrupts.

MAINTENANCE AND OPERATOR SUBSYSTEM (MOSS)

The MOSS interacts with the CCU and main storage in several different ways:

1. The MOSS can read and write anywhere in main storage. This facility must of course be used with great care.

2. As part of the general facility described in (1) above, the top 2K of main storage is reserved for communications between the MOSS and the CCU. Two 32-byte areas of this high storage are called the 'In Mailbox' and the 'Out Mailbox'. Most of the normal communications between the CCU and the MOSS take place via these mailboxes. When a mailbox has been filled, the CCU (or the MOSS) raises an interrupt to inform the MOSS (or the CCU) that the mailbox is available. See (3) below for details of the interrupt.

3. The following interrupts are used:

   a. Level 4 for service interrupts. This is the mechanism used by the MOSS to inform the CCU that it has filled a mailbox. A similar mechanism is used by the CCU to inform the MOSS that it has filled a mailbox.

b. Level 3 which is used as a console interrupt key.

c. Levels 2 and 3 for diagnostic interrupts.

d. Level 1 for error interrupts.

4. The CCU Register Input and CCU Register Output instructions are used to move control information between the CCU and the MOSS registers.

CHAPTER 3. INSTRUCTION SET

The 3725 Communication Controller makes use of a set of 53 instructions that can be used to tailor the control program to meet the specific needs of the data communication system. The instruction set provides the greatest possible program flexibility in the smallest amount of storage.

This chapter gives the general instruction formats, followed by a detailed description of each instruction.

Important Note: readers familiar with the instruction set of the IBM 3704/5 Communications Controllers should note that there are minor differences between the 3704/5 and the 3725 in the following instructions:

> Load, Load Halfword, Insert Character, Insert Character and Count, Load Address, Store, Store Halfword, Store Character, Store Character and Count, Branch and Link, Exit, CCU Register Input, CCU Register Output.

The following instructions are new:

> Adapter Input/Output, Adapter Input/Output Immediate.

Programming Note

Modifying an instruction during program execution is not advisable. If instruction modification is done, at least four CCU cycles must be allowed before executing the modified instruction. Failure to observe this rule may cause unpredictable results.

INSTRUCTION FORMAT

All instructions have a length of one half word, with the exception of the 'Load Address', 'Branch and Link', and 'Adapter Input/Output Immediate' instructions which have a length of two halfwords. There are eight basic instruction formats:

- Register to Immediate Operand Instructions (RI)
- Register to Register Instructions (RR)
- Register to Storage Instructions (RS)
- Register to Storage with Addition Instructions (RSA)
- Branch Operations (RT)
- Register to Immediate Address Instructions (RA)

- Exit Instruction (EXIT)
- Input/Output Instructions (RE)

INSTRUCTION SET SUMMARY

The table below shows the basic mnemonic names and assembler operand field designations for each instruction. For the explanation of the terms occurring in the 'Operand Field Format' column, refer to the notes in the next section for each different type of instruction.

Instruction	Format Code	Mnemonic	Operand Field Format
Adapter Input/Output	RR	IOH	R1,R2
Adapter Input/Output Immediate	RI	IOHI	R,I
Add Character Register	RR	ACR	R1(N1),R2(N2)
Add Halfword Register	RR	AHR	R1,R2
Add Register	RR	AR	R1,R2
Add Register Immediate	RI	ARI	R(N),I
AND Character Register	RR	NCR	R1(N1),R2(N2)
AND Halfword Register	RR	NHR	R1,R2
AND Register	RR	NR	R1,R2
AND Register Immediate	RI	NRI	R(N),I
Branch	RT	B	T
Branch and Link	RA	BAL	R,A
Branch and Link Register	RR	BALR	R1,R2
Branch on Bit	RT	BB	R(N,M),T
Branch on Count	RT	BCT	R(N),T
Branch on C Latch	RT	BCL	T
Branch on Z Latch	RT	BZL	T
CCU Register Input	RE	IN	R,E
CCU Register Output	RE	OUT	R,E
Compare Character Register	RR	CCR	R1(N1),R2(N2)
Compare Halfword Register	RR	CHR	R1,R2
Compare Register	RR	CR	R1,R2
Compare Register Immediate	RI	CRI	R(N),I
Exclusive OR Character Register	RR	XCR	R1(N1),R2(N2)
Exclusive OR Halfword Register	RR	XHR	R1,R2
Exclusive OR Register	RR	XR	R1,R2
Exclusive OR Register Immediate	RI	XRI	R(N),I
Exit	EXIT	EXIT	-
Insert Character	RS	IC	R(N),D(B)
Insert Character and Count	RSA	ICT	R(N),B
Load	RS	L	R,D(B)
Load Address	RA	LA	R,A
Load Character Register	RR	LCR	R1(N1),R2(N2)
Load Character with Offset Reg	RR	LCOR	R1(N1),R2(N2)
Load Halfword	RS	LH	R,D(B)
Load Halfword Register	RR	LHR	R1,R2
Load Halfword with Offset Reg	RR	LHOR	R1,R2
Load Register	RR	LR	R1,R2
Load Register Immediate	RI	LRI	R(N),I
Load with Offset Register	RR	LOR	R1,R2
OR Character Register	RR	OCR	R1(N1),R2(N2)
OR Halfword Register	RR	OHR	R1,R2
OR Register	RR	OR	R1,R2
OR Register Immediate	RI	ORI	R(N),I
Store	RS	ST	R,D(B)
Store Character	RS	STC	R(N),D(B)
Store Character and Count	RSA	STCT	R(N),B
Store Halfword	RS	STH	R,D(B)
Subtract Character Register	RR	SCR	R1(N1),R2(N2)
Subtract Halfword Register	RR	SHR	R1,R2
Subtract Register	RR	SR	R1,R2
Subtract Register Immediate	RI	SRI	R(N),I
Test Register Under Mask	RI	TRM	R(N),I

INSTRUCTION SET BY TYPE OF INSTRUCTION

The table below shows the operation code bit structure and the operand fields for each instruction. In this section, the instructions are grouped by type of instruction.

				Format																
Name	Instruction	CZ	N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
LRI	Load Reg Immediate	Y	1	1	0	0	0	0	R		N	Immediate Data								(1,2)
ARI	Add Reg Immediate	Y	1	1	0	0	1	0	R		N	Immediate Data								(1,2)
SRI	Subt Reg Immediate	Y	1	1	0	1	0	0	R		N	Immediate Data								(1,2)
CRI	Comp Reg Immediate	Y	1	1	0	1	1	0	R		N	Immediate Data								(1,2)
XRI	XOR Reg Immediate	Y	1	1	1	0	0	0	R		N	Immediate Data								(1,2)
ORI	OR Reg Immediate	Y	1	1	1	0	1	0	R		N	Immediate Data								(1,2)
NRI	AND Reg Immediate	Y	1	1	1	1	0	0	R		N	Immediate Data								(1,2)
TRM	Test R under Mask	Y	1	1	1	1	1	0	R		N	Mask Bits								(1,2)
BALR	Branch & Link Reg	N	4	0	R2			0	R1			0	1	0	0	0	0	0	0	(3)
LHR	Load Halfword Reg	Y	1	0	R2			0	R1			1	0	0	0	0	0	0	0	(3)
LR	Load Register	Y	1	0	R2			0	R1			1	0	0	0	1	0	0	0	(3)
AHR	Add Halfword Reg	Y	1	0	R2			0	R1			1	0	0	1	0	0	0	0	(3)
AR	Add Register	Y	1	0	R2			0	R1			1	0	0	1	1	0	0	0	(3)
SHR	Subt Halfword Reg	Y	1	0	R2			0	R1			1	0	1	0	0	0	0	0	(3)
SR	Subtract Register	Y	1	0	R2			0	R1			1	0	1	0	1	0	0	0	(3)
CHR	Comp Halfword Reg	Y	1	0	R2			0	R1			1	0	1	1	0	0	0	0	(3)
CR	Compare Register	Y	1	0	R2			0	R1			1	0	1	1	1	0	0	0	(3)
XHR	XOR Halfword Reg	Y	1	0	R2			0	R1			1	1	0	0	0	0	0	0	(3)
XR	XOR Register	Y	1	0	R2			0	R1			1	1	0	0	1	0	0	0	(3)
OHR	OR Halfword Reg	Y	1	0	R2			0	R1			1	1	0	1	0	0	0	0	(3)
OR	OR Register	Y	1	0	R2			0	R1			1	1	0	1	1	0	0	0	(3)
NHR	AND Halfword Reg	Y	1	0	R2			0	R1			1	1	1	0	0	0	0	0	(3)
NR	AND Register	Y	1	0	R2			0	R1			1	1	1	0	1	0	0	0	(3)
LHOR	Load HW w. Offset	Y	1	0	R2			0	R1			1	1	1	1	0	0	0	0	(3)
LOR	Load w. Offset	Y	1	0	R2			0	R1			1	1	1	1	1	0	0	0	(3)
LCR	Load Character Reg	Y	1	0	R2		N	0	R1		N	0	0	0	0	1	0	0	0	(4,5)
ACR	Add Character Reg	Y	1	0	R2		N	0	R1		N	0	0	0	1	1	0	0	0	(4,5)
SCR	Subt Character Reg	Y	1	0	R2		N	0	R1		N	0	0	1	0	1	0	0	0	(4,5)
CCR	Comp Character Reg	Y	1	0	R2		N	0	R1		N	0	0	1	1	1	0	0	0	(4,5)
XCR	XOR Character Reg	Y	1	0	R2		N	0	R1		N	0	1	0	0	1	0	0	0	(4,5)
OCR	OR Character Reg	Y	1	0	R2		N	0	R1		N	0	1	0	1	1	0	0	0	(4,5)
NCR	Add Character Reg	Y	1	0	R2		N	0	R1		N	0	1	1	0	1	0	0	0	(4,5)
LCOR	Load Char w Offset	Y	1	0	R2		N	0	R1		N	0	1	1	1	1	0	0	0	(4,5)
L	Load	Y	4	0	Base Reg			0	R			0	Displacement					1	0	(6,7)
ST	Store	N	2	0	Base Reg			0	R			1	Displacement					1	0	(6,7)
LH	Load Halfword	Y	3	0	Base Reg			0	R			0	Displacement						1	(6,7)
STH	Store Halfword	N	1	0	Base Reg			0	R			1	Displacement						1	(6,7)
IC	Insert Character	Y	3	0	Base Reg			1	R		N	0	Displacement							(1,2,7)
STC	Store Character	N	1	0	Base Reg			1	R		N	1	Displacement							(1,2,7)
ICT	Insert Char & Ct	N	4	0	Base Reg			0	R		N	0	0	0	1	0	0	0	0	(1,2,8)
STCT	Store Char & Ct	N	3	0	Base Reg			0	R		N	0	0	1	1	0	0	0	0	(1,2,8)
B	Branch	N	4	1	0	1	0	1	Displacement											(9)
BZL	Branch on Z latch	N	2	1	0	0	0	1	Displacement											(9)
BCL	Branch on C latch	N	2	1	0	0	1	1	Displacement											(9)
BCT	Branch on Count	N	3	1	0	1	1	1	R		N	1	Displacement							(1,2,9)
BB	Branch on Bit	N	2	1	1	M	M	1	R		N	M	Displacement							(1,2,9,10)
BAL	Branch and Link	N	4	1	0	1	1	1	R			0	1	Addr Byte Ext						Addr Bytes 0 and 1 (6,11)
LA	Load Address	N	1	1	0	1	1	1	R			0	0	Addr Byte Ext						Addr Bytes 0 and 1 (6,11)
EXIT	Exit	N	9	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	
IN	CCU Reg Input	N	1	0	E			0	R			E				1	1	0	0	(6,12,13)
OUT	CCU Reg Output	N	1	0	E			0	R			E				0	1	0	0	(6,12,13)
IOH	Adapter I/O	Y	7	0	R2			0	R1			0	1	0	1	0	0	0	0	(3,12,14)
IOHI	Adapter I/O Immed	Y	6	0	0	0	0	0	R			0	1	1	1	0	0	0	0	Depends on adapter (3,12,15)

The column C,Z indicates whether the C and Z registers are changed by the execution of the instruction (Y = yes; N = no).

The column N indicates the minimum number of 200ns CCU cycles required to execute the instruction; in practice there may be more. In the case of the IOH and IOH instructions, the number of execution cycles is increased by the adapter response time.

1. The R field addresses the general registers. As the R field is only two bits long, these bits form the two high order bits of the register address. The low order bit of the address is created by hardware, and is always 1. This means that only odd numbered general registers (1, 3, 5, 7) can be addressed.

2. The bit marked N is used to select one (or sometimes both) of the bytes of the general register selected by the associated R field.

3. The R1 and R2 fields address the general registers. As the R1 and R2 fields are three bits long, these bits can take any value from 0 to 7, and all three bytes of the register are used in the operation.

4. The R1 and R2 fields address the general registers. As the R1 and R2 fields are only two bits long, these bits form the two high order bits of the register address. The low order bit of the address is created by hardware, and is always 1. This means that only odd numbered general registers (1, 3, 5, 7) can be addressed.

5. The bit marked N is used to select one (or sometimes both) of the bytes of the general register selected by the associated R1/R2 field.

6. The R field addresses the general registers. As the R field is three bits long, these bits can take any value from 0 to 7, and all three bytes of the register are used in the operation.

7. The effective storage address is formed by adding the displacement to the contents of the base register selected by bits 1-3.

8. The contents of the base register specified are incremented by 1 after storage access.

9. The displacement field is added to the address of the next sequential instruction (contained in general register 0) to form the branch address.

10. The three bits of the M(ask) field specify the bit to be tested.

11. The 20 bits contained in the extension byte and in bytes 0 and 1 form an address. In the case of the branch and link instruction, these 20 bits form the branch address. In the case of the load address instruction, the 20 bits are treated as immediate data and loaded into the register specified by R.

12. The IN and OUT instructions can only address the CCU external registers. The IOH and IOHI instructions can only address the adapter external registers. See below for lists of these registers.

13. The E field consists of 7 bits and addresses one of the 128 external CCU registers.

14. The contents of R2 includes the address of the adapter external register.

15. The second half word contains the address of the adapter external register.

INSTRUCTION SET DETAILED BIT STRUCTURE

In this section, the instructions are grouped logically, that is, it starts with all the LOAD instructions, and then continues with all the STORE instructions, etc.

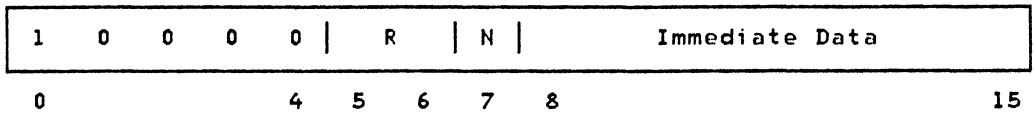
Each table is followed by a detailed description of the individual instructions of that type.

LOAD INSTRUCTIONS

			Format															
Name	Instruction	Type	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LRI	Load Reg Immediate	RI	1	0	0	0	0	R		N	0	Immediate Data						
LR	Load Register	RR	0	R2			0	R1			1	0	0	0	1	0	0	0
LOR	Load Reg w. Offset	RR	0	R2			0	R1			1	1	1	1	1	0	0	0
LHR	Load Halfword Reg	RR	0	R2			0	R1			1	0	0	0	0	0	0	0
LHOR	Load H/W w. Offset	RR	0	R2			0	R1			1	1	1	1	0	0	0	0
LCR	Load Character Reg	RR	0	R2		N	0	R1		N	0	0	0	0	1	0	0	0
LCOR	Load Char w Offset	RR	0	R2		N	0	R1		N	0	1	1	1	1	0	0	0
L	Load	RS	0	Base Reg			0	R			0	Displacement					1	0
LH	Load Halfword	RS	0	Base Reg			0	R			0	Displacement						1
IC	Insert Character	RS	0	Base Reg			1	R		N	0	Displacement						
ICT	Insert Char & Ct	RSA	0	Base Reg			0	R		N	0	0	0	1	0	0	0	0
LA	Load Address	RA	1	0	1	1	1	R			0	0	0	0	Ad Byte Ext			
			Addr Byte 0								Addr Byte 1							

Load Register Immediate

LRI R(N),I RI



The second operand (Immediate Data field) is loaded into the first operand (byte 0 if N = 0, or byte 1 if N = 1, of the register specified by R). The non-selected bytes of the register remain unchanged. The two bits of R form the two high-order bits of the register address; the low order bit is forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

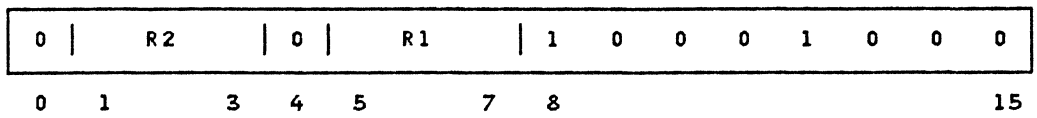
Resulting Condition Latches:

C: the result in the selected byte of R ≠ 0.

Z: the result in the selected byte of R = 0

Load Register

LR R1,R2 RR



The second operand (bytes 0 and 1, and byte X of R2) is loaded into the first operand (bytes 0 and 1, and byte X of the register specified by R1). The second operand remains unchanged, and the condition latches are set according to the result in the first operand.

Resulting Condition Latches:

C: the result in R1 ≠ 0.

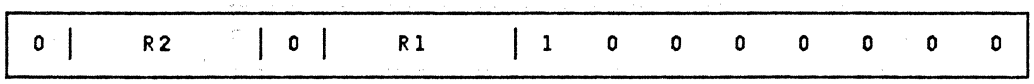
Z: the result in R1 = 0

Programming Note

If register 0 is specified as R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

Load Halfword Register

LHR R1,R2 RR



The second operand (bytes 0 and 1, but not byte X of R2) is loaded into the first operand (bytes 0 and 1, but not byte X) of the register specified by R1). The second operand remains unchanged, and the condition latches are set according to the result in the first operand.

Resulting Condition Latches:

C: the result in bytes 0 and 1 of R ≠ 0.

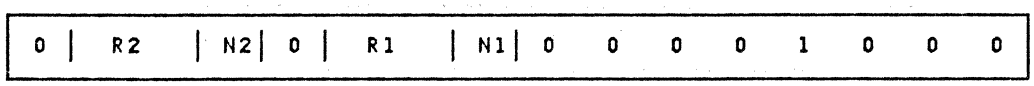
Z: the result in bytes 0 and 1 of R = 0

Programming Note

If register 0 is specified as R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

Load Character Register

LCR R1(N1),R2(N2) RR



The second operand (R2, byte 0 if N2 = 0, or byte 1 if N2 = 1) is loaded into the first operand (R1, byte 0 if N1 = 0, or byte 1 if N1 = 1). The non-selected bytes of R1 remain unchanged. The two bits of R2 and R1 form the two high-order bits of the register addresses; the low order bits are forced to 1 by hardware. For this reason, R2 and R1 are always odd-numbered registers (1, 3, 5, 7).

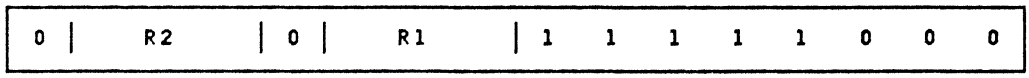
Resulting Condition Latches:

C: the selected byte of R1 contains an even number of 1 bits, or is zero.

Z: the selected byte of R1 = 0

Load Register with Offset

LOR R1,R2 RR



The second operand (bytes 0 and 1, and byte X of R2) is shifted right one bit position, and the result loaded into the first operand (bytes 0 and 1, and byte X, of the register specified by R1). A 0 bit is inserted into the high-order bit position of R1, byte X.

Resulting Condition Latches:

C: a 1 bit was shifted out of byte 1, bit 7 of R2.

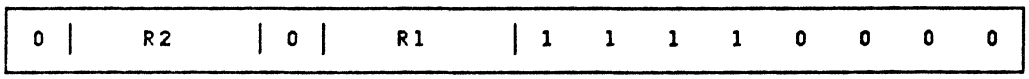
Z: the result in R1 = 0.

Programming Note

If register 0 is specified as R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

Load Halfword Register with Offset

LHOR R1,R2 RR



The second operand (bytes 0 and 1, but not byte X of R2) is shifted right one bit position, and the result loaded into the first operand (bytes 0 and 1, but not byte X, of the register specified by R1). A 0 bit is inserted into the high-order bit position of R1, byte 0.

Resulting Condition Latches:

C: a 1 bit was shifted out of byte 1, bit 7 of R2.

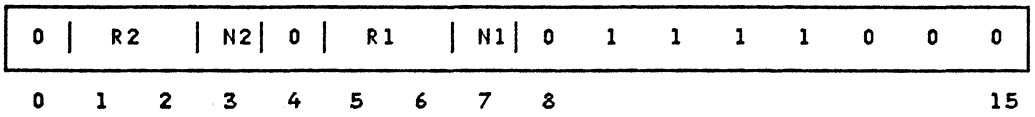
Z: the result in bytes 0 and 1 of R1 = 0.

Programming Note

If register 0 is specified as R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

Load Character Register with Offset

LCOR R1(N1),R2(N2) RR



The second operand (R2, byte 0 if N2 = 0, or byte 1 if N2 = 1) is shifted right one bit position, and the result is loaded into the first operand (R1, byte 0 if N1 = 0, or byte 1 if N1 = 1). A 0 bit is inserted into the high-order bit position of the selected byte of R1. The non-selected bytes of R1 remain unchanged. The two bits of R1 and R2 form the two high-order bits of the register addresses; the low order bits are forced to 1 by hardware. For this reason, R1 and R2 are always odd-numbered registers (1, 3, 5, 7).

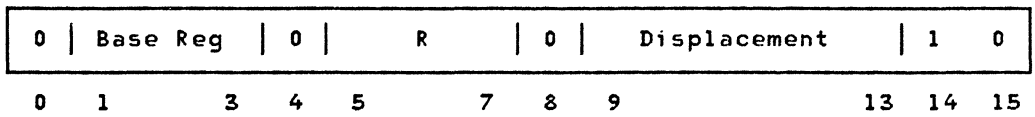
Resulting Condition Latches:

C: a 1 bit was shifted out of bit 7 of the selected byte of R2.

Z: the result in the selected byte of R1 = 0.

Load

L R,D(B) RS



The load instruction loads the data (the second operand) from a four-byte field in storage into R, the first operand register. The four byte field containing the second operand must be on a halfword boundary. As the general registers are only 3 bytes long (bytes X, 0, and 1), only the three low-order bytes (22 bits) of the storage location are used.

The storage address is formed by adding the displacement value to the contents of the base register specified by B. The displacement field allows for a displacement of between 0 and 124 bytes in multiples of 4 (31 fullwords).

Resulting Condition Latches:

C: the result in R ≠ 0

Z: the result in R = 0.

Programming Notes

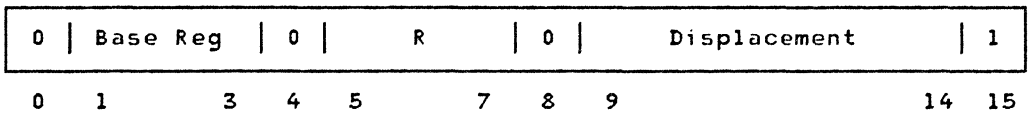
1. The low order-bit of the address is ignored since storage is addressed on halfword boundaries with this instruction.

2. If register 0 (instruction address register) is specified in the R field, the instruction causes an unconditional branch to the address loaded into register 0; the condition latches are unchanged.

3. If the base register specified is R0, the contents of R0 are not used as the address. Instead, a program settable address located in CCU external register X'46' is used as a base address instead of the contents of register 0. This permits direct addressing of the 32 fullwords starting at the address contained in register X'46' without having to load a base register. After program loading, the contents of register X'46' are unpredictable and must be initialized before use.

Load Halfword

LH R,D(B) RS



The load halfword instruction loads the data (the second operand) from a halfword field in storage into bytes 0 and 1 of R, the first operand register. Byte X of R is set to 0.

The storage address is formed by adding the displacement value to the contents of the base register specified by B. The displacement field allows for a displacement of between 0 and 126 bytes in multiples of 2 (63 halfwords).

Resulting Condition Latches:

C: the result in bytes 0 and 1 of R ≠ 0

Z: the result in bytes 0 and 1 of R = 0.

Programming Notes

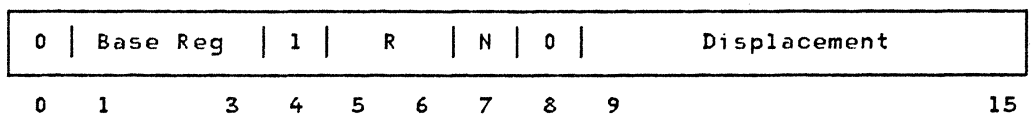
1. The low order-bit of the address is ignored since storage is addressed on halfword boundaries with this instruction.

2. If register 0 (instruction address register) is specified in the R field, the instruction causes an unconditional branch to the address loaded into register 0; the condition latches are unchanged.

3. If the base register specified is R0, the contents of R0 are not used as the address. Instead, a program settable address located in CCU external register X'45' is used as a base address instead of the contents of register 0. This permits direct addressing of the 64 halfwords starting at the address contained in register X'45' without having to load a base register. After program loading, the contents of register X'45' are unpredictable and must be initialized before use.

Insert Character

IC R(N),D(B) RS



The insert character instruction loads the 8-bit character at the second operand address into byte 0 (N = 0) or byte 1 (N = 1) of the register specified by R. The remaining bits of the register are unchanged.

The storage address is formed by adding the displacement value to the contents of the base register specified by B. The displacement field allows for a displacement of between 0 and 127 bytes. The two bits of R form the two high-order bits of the register address; the low order bits are forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

Resulting Condition Latches:

C: the selected byte of R contains an even number of 1 bits, or is zero.

Z: the selected byte of R is equal to 0.

Programming Note

If the base register specified is R0, the contents of R0 are not used as the address. Instead, a program settable address located in CCU external register X'44' is used as a base address instead of the contents of register 0. This permits direct addressing of the 128 bytes starting at the address contained in register X'44' without having to load a base register. After program loading, the contents of register X'44' are unpredictable and must be initialized before use.

Insert Character and Count

ICT R(N),B RS

The eight bit character at the second operand address (contained in the base register designated by the B field) is loaded into byte 0 (N = 0) or byte 1 (N = 1) of the register specified by R. After the storage address has been obtained from the base register B, the contents of the base register is incremented by 1. After execution of the instruction, the base register normally contains an address 1 byte greater than before execution (see however programming note 2 below). The two bits of R form the two high-order bits of the register address; the low order bit is forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

Resulting Condition Latches: unchanged

Programming Notes

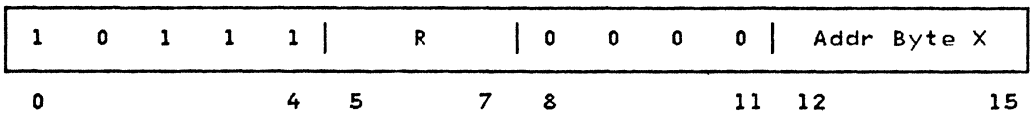
1. If register 0 is specified in the B field, it causes an invalid operation check.

2. If R and B specify the same (odd) register, its contents are incremented by 1 before the character is inserted into the selected byte of the register.

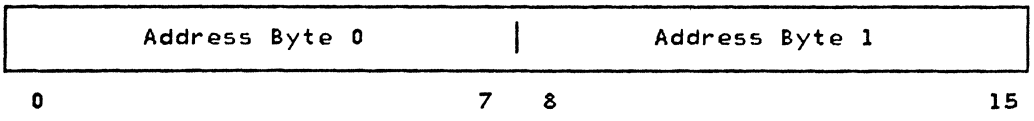
Load Address

LA R,A RA

First halfword



Second halfword



The load address instruction is a 4-byte instruction. The second operand (address field) is treated as an immediate operand, and is loaded into the first operand specified by R.

Resulting Condition Latches: unchanged

Programming Note

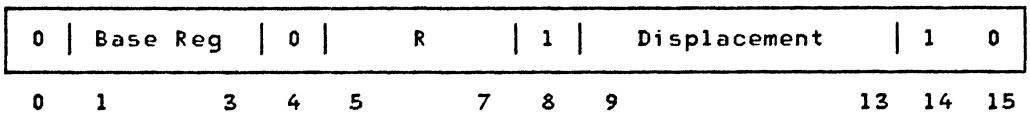
If register 0 is specified by R, a branch results to the address contained in the address field.

STORE INSTRUCTIONS

Name	Instruction	Type	Format
ST	Store	RS	0 | Base Reg | 0 | R | 1 | Displacement | 1 0
STH	Store Halfword	RS	0 | Base Reg | 0 | R | 1 | Displacement | 1
STC	Store Character	RS	0 | Base Reg | 1 | R | N | 1 | Displacement
STCT	Store Char & Ct	RSA	0 | Base Reg | 0 | R | N | 0 0 1 1 0 0 0 0

Store

ST R,D(B) RS



The store instruction stores the contents of the first operand (the register specified by R) into the second operand in storage. The address of the second operand must be on a halfword boundary. As the general registers are only 3 bytes long (bytes X, 0, and 1), only the three low-order bytes (22 bits) of the storage location are affected.

The storage address is formed by adding the displacement value to the contents of the base register specified by B. The displacement field allows for a displacement of between 0 and 124 bytes in multiples of 4 (31 fullwords).

Resulting Condition Latches: unchanged

Programming Notes

1. The low order-bit of the address is ignored since storage is addressed on halfword boundaries with this instruction.

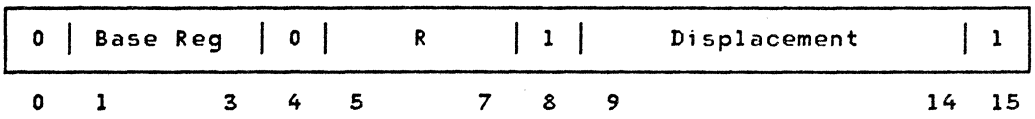
2. If the base register specified is R0, the contents of R0 are not used as the address. Instead, a program settable address located in CCU external register X'46' is used as a base address instead of the contents of register 0. This permits direct addressing of the 32 fullwords starting at the address contained in register X'46' without having to load a base

register. After program loading, the contents of register X'46' are unpredictable and must be initialized before use.

3. If the general register specified by R is R0, all zeros are stored at the storage location, instead of the contents of R0.

Store Halfword

STH R,D(B) RS



The store halfword instruction stores bytes 0 and 1 of the first operand (the register specified by R) into the second operand in storage.

The storage address is formed by adding the displacement value to the contents of the base register specified by B. The displacement field allows for a displacement of between 0 and 126 bytes in multiples of 2 (63 halfwords).

Resulting Condition Latches: unchanged

Programming Notes

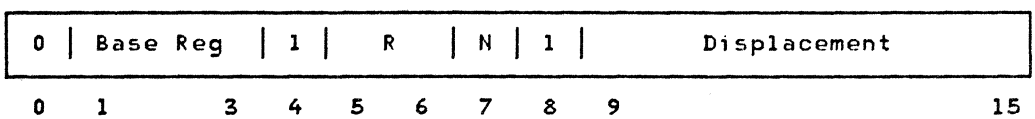
1. The low order-bit of the address is ignored since storage is addressed on halfword boundaries with this instruction.

2. If the general register specified by R is R0, all zeros are stored at the storage location, instead of the contents of R0.

3. If the base register specified is R0, the contents of R0 are not used as the base address. Instead, a program settable address located in CCU external register X'45' is used as a base address instead of the contents of register 0. This permits direct addressing of the 64 halfwords starting at the address contained in register X'45' without having to load a base register. After program loading, the contents of register X'45' are unpredictable and must be initialized before use.

Store Character

STC R(N),D(B) RS



The store character instruction stores byte 0 (N = 0) or byte 1 (N = 1) of the register specified by R into the second operand address in storage.

The storage address is formed by adding the displacement value to the contents of the base register specified by B. The displacement field allows for a displacement of between 0 and 127 bytes. The two bits of R form the two high-order bits of the register address; the low order bits are forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

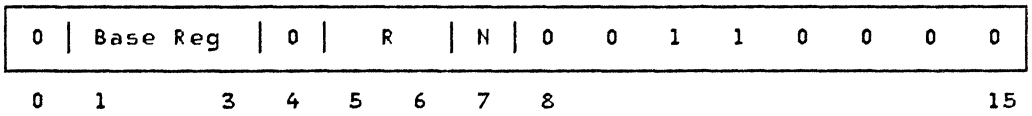
Resulting Condition Latches: unchanged

Programming Note

If the base register specified is R0, the contents of R0 are not used as the address. Instead, a program settable address located in CCU external register X'44' is used as a base address instead of the contents of register 0. This permits direct addressing of the 128 bytes starting at the address contained in register X'44' without having to load a base register. After program loading, the contents of register X'44' are unpredictable and must be initialized before use.

Store Character and Count

STCT R(N),B RS



The eight bit character in byte 0 (N = 0) or byte 1 (N = 1) of the register specified by R is loaded into the location specified by the second operand address (contained in the base register designated by the B field). After the storage address has been obtained from the base register B, the contents of the base register is incremented by 1. After execution of the instruction, the base register normally contains an address 1 byte greater than before execution (see however programming note 2 below). The two bits of R form the two high-order bits of the register address; the low order bit is forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

Resulting Condition Latches: unchanged

Programming Notes

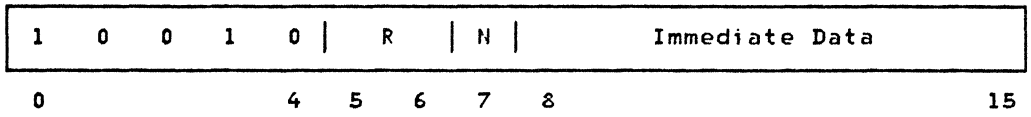
1. If register 0 is specified in the B field, it causes an invalid operation check.

2. If R and B specify the same (odd) register, its contents are incremented by 1 before the selected byte of that register is stored.

Name	Instruction	Type	Format
ARI	Add Reg Immediate	RI	1 0 0 1 0 | R | N | 0 | Immediate Data
AR	Add Register	RR	0 | R2 | 0 | R1 | 1 0 0 1 1 0 0 0
AHR	Add Halfword Reg	RR	0 | R2 | 0 | R1 | 1 0 0 1 0 0 0 0
ACR	Add Character Reg	RR	0 | R2 | N | 0 | R1 | N | 0 0 0 1 1 0 0 0

Add Register Immediate

ARI R(N),I RI



The second operand (Immediate Data field) is added to the first operand (byte 0 if N = 0, or bytes 0 and 1 if N = 1, of the register specified by R). The sum is then placed in the first operand location. If N = 0, byte 1 of the register remains unchanged. The two bits of R form the two high-order bits of the register address; the low order bit is forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

Resulting Condition Latches:

1. N = 0

   C: an overflow occurred from byte 0 of R.

   Z: the result in byte 0 of R = 0

2. N = 1

   C: an overflow occurred from bytes 0 and 1 of R.

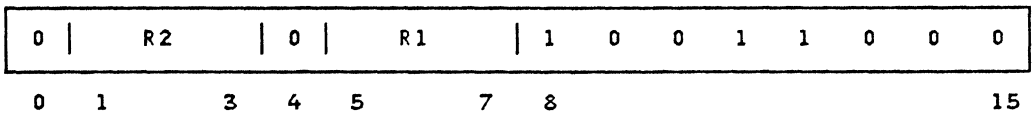
   Z: the result in bytes 0 and 1 of R = 0

Programming Note

The first operand includes byte X of the register specified by R. However, byte X does not affect the setting of the condition latches.

Add Register

AR R1,R2 RR



The second operand (bytes 0 and 1, and byte X of R2) is added to the first operand (bytes 0 and 1, and byte X of the register specified by R1. Addition of the register operands is performed logically without regard to sign. The second operand remains unchanged, and the condition latches are set according to the result in the first operand.

Resulting Condition Latches:

C: an overflow occurred from R1.

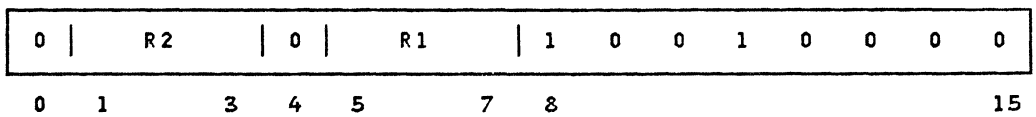
Z: the result in R1 = 0

Programming Note

If register 0 is specified by R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

Add Halfword Register

AHR R1,R2 RR



The second operand (bytes 0 and 1, but not byte X of R2) is added to the first operand (bytes 0 and 1, but not byte X, of the register specified by R1). Addition of the operands is performed logically without regard to sign. The second operand remains unchanged, and the condition latches are set according to the result in the first operand.

Resulting Condition Latches:

C: an overflow occurred from byte 0 of R1.

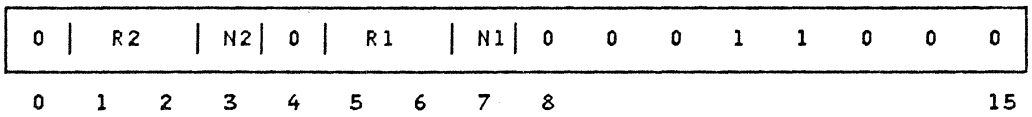
Z: the result in bytes 0 and 1 of R1 = 0

Programming Note

If register 0 is specified as R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

Add Character Register

ACR R1(N1),R2(N2) RR



The second operand (R2, byte 0 if N2 = 0, or byte 1 if N2 =1) is added to the first operand (R1, bytes X and 0 if N1 = 0, or bytes X, 0, and 1 if N1 = 1). The sum is then placed in the first operand location. If N1 = 0, byte 1 of the register remains unchanged. The two bits of R1 and R2 form the two high-order bits of the register addresses; the low order bits are forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

Resulting Condition Latches:

1. N = 0

   C: an overflow occurred from byte 0 of R1.

   Z: the result in byte 0 of R1 = 0

2. N = 1

   C: an overflow occurred from byte 0 of R1.

   Z: the result in bytes 0 and 1 of R1 = 0

Programming Note

The first operand includes byte X of the register specified by R1. However, byte X does not affect the setting of the condition latches.

SUBTRACT INSTRUCTIONS

Name	Instruction	Type	Format 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SRI	Subt Reg Immediate	RI	1	0	1	0	0	R		N	0	Immediate Data						
SR	Subtract Register	RR	0	R2			0	R1			1	0	1	0	1	0	0	0
SHR	Subt Halfword Reg	RR	0	R2			0	R1			1	0	1	0	0	0	0	0
SCR	Subt Character Reg	RR	0	R2		N	0	R1		N	0	0	1	0	1	0	0	0

Subtract Register Immediate

SRI R(N),I RI

1	0	1	0	0	R	N	Immediate Data
0				4	5 6	7	8 15

The second operand (Immediate Data field) is subtracted from the first operand (byte 0 if N = 0, or bytes 0 and 1 if N = 1, of the register specified by R). The result is then placed in the first operand location. The two bits of R form the two high-order bits of the register address; the low order bit is forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

Before the subtraction is performed, the second operand is expanded with high-order zeros to equal the size of the first operand. Subtraction is performed by adding the two's complement of the second operand to the first operand and setting the appropriate condition latch. If the difference is less than zero, the result of the subtraction is in the two's complement form.

Resulting Condition Latches:

1. N = 0

   C: the result in byte 0 of R < 0

   Z: the result in byte 0 of R = 0

2. N = 1

   C: the result in bytes 0 and 1 of R < 0

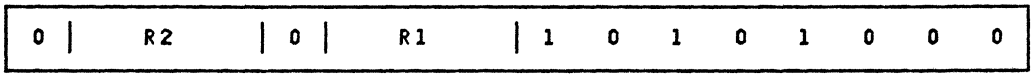
   Z: the result in bytes 0 and 1 of R = 0

Programming Note

The first operand includes byte X of the register specified by R. However, byte X does not affect the setting of the condition latches.

Subtract Register

SR R1,R2 RR



The second operand (bytes 0 and 1, and byte X of R2) is subtracted from the first operand (bytes 0 and 1, and byte X, of the register specified by R1). Subtraction is performed by adding the two's complement of the second operand to the first operand. If the difference is less than zero, the result is in the two's complement form. The second operand remains unchanged, and the condition latches are set according to the result in the first operand.

Resulting Condition Latches:

C: the result in R1 < 0.

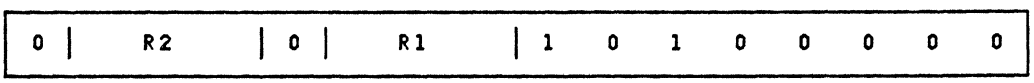
Z: the result in R1 = 0

Programming Note

If register 0 is specified as R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

Subtract Halfword Register

SHR R1,R2 RR



The second operand (bytes 0 and 1, but not byte X of R2) is subtracted from the first operand (bytes 0 and 1, but not byte X, of the register specified by R1). Subtraction is performed by adding the two's complement of the second operand to the first operand. If the difference is less than zero, the result is in the two's complement form. The second operand remains unchanged, and the condition latches are set according to the result in the first operand.

Resulting Condition Latches:

C: the result in bytes 0 and 1 of R1 < 0.

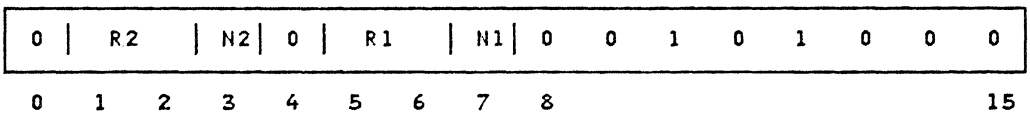
Z: the result in bytes 0 and 1 of R1 = 0

Programming Note

If register 0 is specified as R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

Subtract Character Register

SCR R1(N1),R2(N2) RR



The second operand (R2, byte 0 if N2 = 0, or byte 1 if N2 = 1) is subtracted from the first operand (R1, bytes X and 0 if N1 = 0, or bytes X, 0, and 1 if N1 = 1). The result is then placed in the first operand location. The two bits of R1 and R2 form the two high-order bits of the register addresses; the low order bit is forced to 1 by hardware. For this reason, R1 and R2 are always odd-numbered registers (1, 3, 5, 7).

Before the subtraction is performed, the second operand is expanded with high-order zeros to equal the size of the first operand. Subtraction is performed by adding the two's complement of the second operand to the first operand and setting the appropriate condition latch. If the difference is less than zero, the result of the subtraction is in the two's complement form.

Resulting Condition Latches:

1. N = 0

   C: the result in byte 0 of R1 < 0

   Z: the result in byte 0 of R1 = 0

2. N = 1

   C: the result in bytes 0 and 1 of R1 < 0

   Z: the result in bytes 0 and 1 of R1 = 0

Programming Note

The first operand includes byte X of the register specified by R1. However, byte X does not affect the setting of the condition latches.

COMPARE INSTRUCTIONS

Name	Instruction	Type	Format 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
CRI	Comp Reg Immediate	RI	1 0 1 1 0 | R | N | 0 | Immediate Data
CR	Compare Register	RR	0 | R2 | 0 | R1 | 1 0 1 1 1 0 0 0
CHR	Comp Halfword Reg	RR	0 | R2 | 0 | R1 | 1 0 1 1 0 0 0 0
CCR	Comp Character Reg	RR	0 | R2 | N | 0 | R1 | N | 0 0 1 1 1 0 0 0
TRM	Test R under Mask	RI	1 1 1 1 0 | R | N | Mask Bits

Compare Register Immediate

CRI R(N),I RI

1	0	1	1	0	R	N	Immediate Data
0				4	5 6	7	8 15

The second operand (Immediate Data field) is compared with the first operand (byte 0 if N = 0, or byte 1 if N = 1, of the register specified by R), and the appropriate condition latch is set. The instruction performs a logical compare without regard to an eventual sign bit; all bits of each operand participate in the comparison. The two bits of R form the two high-order bits of the register address; the low order bit is forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

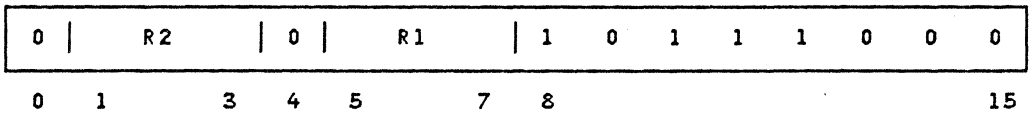
Resulting Condition Latches:

C: the value in the selected byte of R < I

Z: the value in the selected byte of R = I

Compare Register

CR R1,R2 RR



The second operand (bytes 0 and 1, and byte X of R2) is compared with the first operand (bytes 0 and 1, and byte X, of the register specified by R1). This instruction performs a logical compare without regard to the sign bit. The condition latches are set according to the result of the comparison.

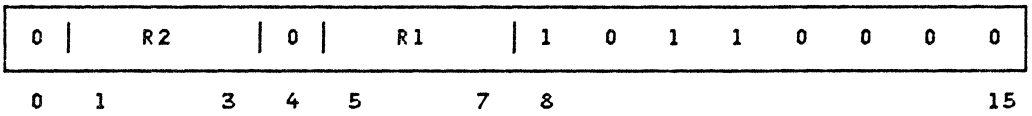
Resulting Condition Latches:

C: value in R1 < value in R2.

Z: value in R1 = value in R2.

Compare Halfword Register

CHR R1,R2 RR



The second operand (bytes 0 and 1, but not byte X of R2) is compared with the first operand (bytes 0 and 1, but not byte X, of the register specified by R1). This instruction performs a logical compare without regard to the sign bit. The condition latches are set according to the result of the comparison.

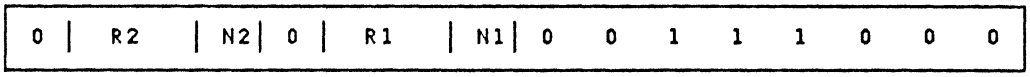
Resulting Condition Latches:

C: bytes 0 and 1 of R1 < bytes 0 and 1 of R2.

Z: bytes 0 and 1 of R1 = bytes 0 and 1 of R2.

Compare Character Register

CCR R1(N1),R2(N2) RR



The second operand (R2, byte 0 if N2 = 0, or byte 1 if N2 = 1) is compared with the first operand (R1, byte 0 if N1 = 0, or byte 1 if N1 = 1), and the appropriate condition latch is set. The instruction performs a logical compare without regard to an eventual sign bit; all bits of each operand participate in the comparison. The two bits of R1 and R2 form the two high-order bits of the register addresses; the low order bits are forced to 1 by hardware. For this reason, R1 and R2 are always odd-numbered registers (1, 3, 5, 7).

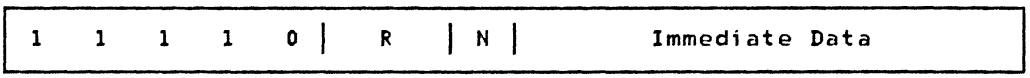
Resulting Condition Latches:

C: the value in the selected byte of R1 < selected byte of R2.

Z: the value in the selected byte of R1 = selected byte of R2.

Test Register Under Mask

TRM R(N),I RI



The second operand (Immediate Data field) is used as a mask to test the bits of the first operand (byte 0 if N = 0, or byte 1 if N = 1, of the register specified by R). A mask bit of 1 indicates that the corresponding register bit is to be tested; when the mask bit is 0, the register bit is ignored. Testing is done by ANDing the selected byte of the register with the immediate operand. The contents of the register remain unchanged. The two bits of R form the two high-order bits of the register address; the low order bit is forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

Resulting Condition Latches:

C: the result of testing the selected byte of R ≠ 0.

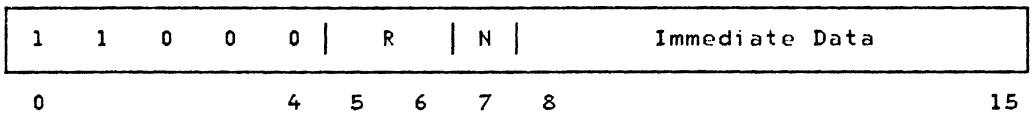
Z: the result of testing the selected byte of R = 0.

XOR INSTRUCTIONS

Name	Instruction	Type	Format 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
XRI	XOR Reg Immediate	RI	1	1	0	0	0	R		N	0	Immediate Data						
XR	XOR Register	RR	0	R2			0	R1			1	1	0	0	1	0	0	0
XHR	XOR Halfword Reg	RR	0	R2			0	R1			1	1	0	0	0	0	0	0
XCR	XOR Character Reg	RR	0	R2		N	0	R1		N	0	1	0	0	1	0	0	0

XOR Register Immediate

XRI R(N),I RI



The second operand (Immediate Data field) is exclusive ORed with the first operand (byte 0 if N = 0, or byte 1 if N = 1, of the register specified by R). The resulting byte is placed in the first operand location, and the appropriate condition latch is set. The remaining bytes of R are unchanged.

Operands are treated as unstructured logical quantities, and the connective exclusive OR is applied bit by bit. A bit position in the result is set to 1 if the corresponding bit positions in the two operands are unlike; otherwise, the resulting bit position is set to 0. The two bits of R form the two high-order bits of the register address; the low order bit is forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

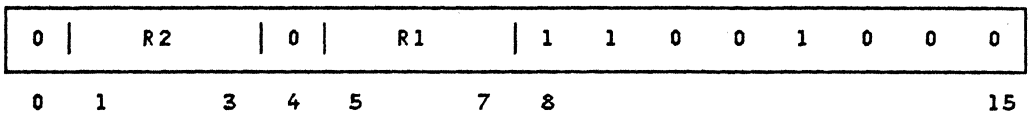
Resulting Condition Latches:

C: the result in the selected byte of R ≠ 0.

Z: the result in the selected byte of R = 0

XOR Register

XR R1,R2 RR



The second operand (bytes 0 and 1, and byte X of R2) is exclusive ORed with the first operand (bytes 0 and 1, and byte X, of the register specified by R1). Operands are treated as unstructured logical quantities, and the connective exclusive OR is applied bit by bit. A bit position in the result is set to 1 if the corresponding bit positions in the two operands are unlike; otherwise, the resulting bit is set to 0. The second operand is unchanged, and the condition latches are set according to the result of the operation.

Resulting Condition Latches:

C: the result in R1 ≠ 0.

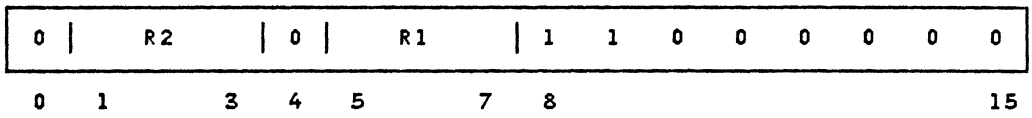
Z: the result in R1 = 0.

Programming Note

If register 0 is specified as R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

XOR Halfword Register

XHR R1,R2 RR



The second operand (bytes 0 and 1, but not byte X of R2) is exclusive ORed with the first operand (bytes 0 and 1, but not byte X, of the register specified by R1). Operands are treated as unstructured logical quantities, and the connective exclusive OR is applied bit by bit. A bit position in the result is set to 1 if the corresponding bit positions in the two operands are unlike; otherwise, the resulting bit is set to 0. The second operand is unchanged, and the condition latches are set according to the result of the operation.

Resulting Condition Latches:

C: the result in bytes 0 and 1 of R1 ≠ 0.

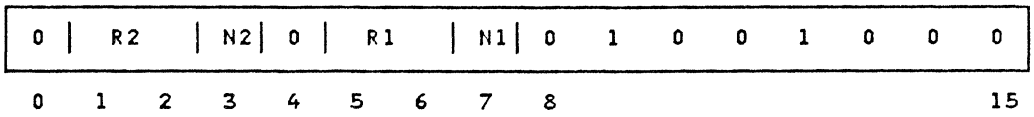
Z: the result in bytes 0 and 1 of R1 = 0.

Programming Note

If register 0 is specified as R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

XOR Character Register

XCR R1(N1),R2(N2) RR



The second operand (R2, byte 0 if N2 = 0, or byte 1 if N2 = 1) is exclusive ORed with the first operand (R1, byte 0 if N1 = 0, or byte 1 if N1 = 1). The result is placed in the first operand location, and the appropriate condition latch is set. The remaining bytes of R are unchanged.

Operands are treated as unstructured logical quantities, and the connective exclusive OR is applied bit by bit. A bit position in the result is set to 1 if the corresponding bit positions in the two operands are unlike; otherwise, the resulting bit position is set to 0. The two bits of R1 and R2 form the two high-order bits of the register addresses; the low order bits are forced to 1 by hardware. For this reason, R1 and R2 are always odd-numbered registers (1, 3, 5, 7).

Resulting Condition Latches:

C: the result in the selected byte of R ≠ 0.

Z: the result in the selected byte of R = 0

OR INSTRUCTIONS

Name	Instruction	Type	Format 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
ORI	OR Reg Immediate	RI	1 1 0 1 0 | R | N | 0 | Immediate Data
OR	OR Register	RR	0 | R2 | 0 | R1 | 1 1 0 1 1 0 0 0
OHR	OR Halfword Reg	RR	0 | R2 | 0 | R1 | 1 1 0 1 0 0 0 0
OCR	OR Character Reg	RR	0 | R2 | N | 0 | R1 | N | 0 1 0 1 1 0 0 0

OR Register Immediate

ORI R(N),I RI

1	1	0	1	0	R	N	Immediate Data
0				4	5 6	7	8 15

The second operand (Immediate Data field) is ORed with the first operand (byte 0 if N = 0, or byte 1 if N = 1, of the register specified by R). The result of the operation is placed in the first operand location; the remaining bytes of the register remain unchanged.

Operands are treated as unstructured logical quantities, and the connective inclusive OR is applied bit by bit. A bit position in the result is set to one if the corresponding bit in either one or both of the operands contains a 1; otherwise, the result bit remains at zero. All values of operands and result are valid. The two bits of R form the two high-order bits of the register address; the low order bit is forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

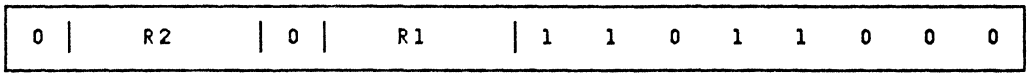
Resulting Condition Latches:

C: the result in the selected byte of R ≠ 0.

Z: the result in the selected byte of R = 0

OR Register

OR R1,R2 RR



The second operand (bytes 0 and 1, and byte X of R2) is ORed with the first operand (bytes 0 and 1, and byte X, of the register specified by R1). Operands are treated as unstructured logical quantities, and the connective inclusive OR is applied bit by bit. A bit position in the result is set to 1 if the corresponding bit position in either one or both of the operands contains a 1. Otherwise the result bit contains 0. Any value in the operands or in the result is valid. The second operand is unchanged, and the condition latches are set according to the result of the operation.

Resulting Condition Latches:

C: the result in R1 ≠ 0.

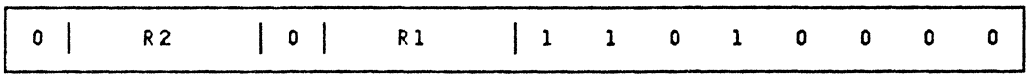
Z: the result in R1 = 0.

Programming Note

If register 0 is specified as R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

OR Halfword Register

OHR R1,R2 RR



The second operand (bytes 0 and 1, but not byte X of R2) is ORed with the first operand (bytes 0 and 1, but not byte X, of the register specified by R1). Operands are treated as unstructured logical quantities, and the connective inclusive OR is applied bit by bit. A bit position in the result is set to 1 if the corresponding bit position in either one or both of the operands contains a 1. Otherwise the result bit contains 0. Any value in the operands or in the result is valid. The second operand is unchanged, and the condition latches are set according to the result of the operation.

Resulting Condition Latches:

C: the result in bytes 0 and 1 of R1 ≠ 0.

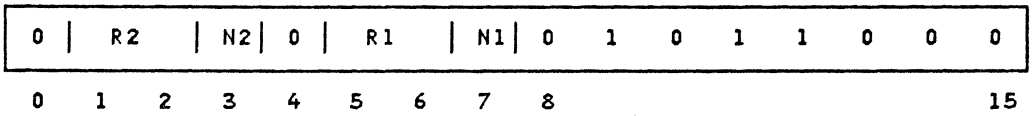
Z: the result in bytes 0 and 1 of R1 = 0.

Programming Note

If register 0 is specified as R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

OR Character Register

OCR R1(N1),R2(N2) RR



The second operand (R2, byte 0 if N2 = 0, or byte 1 if N2 = 1) is ORed with the first operand (R1, byte 0 if N1 = 0, or byte 1 if N1 = 1). The result of the operation is placed in the first operand location; the remaining bytes of the register remain unchanged.

Operands are treated as unstructured logical quantities, and the connective inclusive OR is applied bit by bit. A bit position in the result is set to one if the corresponding bit in either one or both of the operands contains a 1; otherwise, the result bit remains at zero. All values of operands and result are valid. The two bits of R1 and R2 form the two high-order bits of the register addresses; the low order bits are forced to 1 by hardware. For this reason, R1 and R2 are always odd-numbered registers (1, 3, 5, 7).

Resulting Condition Latches:

C: the result in the selected byte of R ≠ 0.

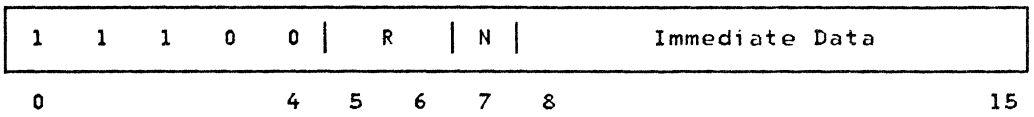
Z: the result in the selected byte of R = 0

AND INSTRUCTIONS

			Format													
Name	Instruction	Type	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
NRI	AND Reg Immediate	RI	1	1	1	0	0	R		N	0	Immediate Data						
NR	AND Register	RR	0	R2			0	R1			1	1	1	0	1	0	0	0
NHR	AND Halfword Reg	RR	0	R2			0	R1			1	1	1	0	0	0	0	0
NCR	AND Character Reg	RR	0	R2		N	0	R1		N	0	1	1	0	1	0	0	0

AND Register Immediate

NRI R(N),I RI



The second operand (Immediate Data field) is ANDed with the first operand (byte 0 if N = 0, or byte 1 if N = 1, of the register specified by R). The result of the operation is placed in the first operand location; the remaining bytes of the register remain unchanged.

Operands are treated as unstructured logical quantities, and the connective AND is applied bit by bit. A bit position in the result is set to one if the corresponding bit position in both operands contains a 1; otherwise the result bit is set to zero. All bits of each operand participate in the operation. All values of operands and result are valid. The two bits of R form the two high-order bits of the register address; the low order bit is forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

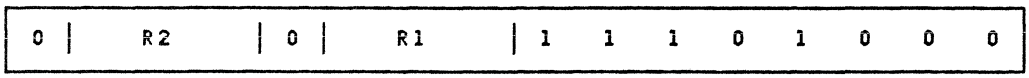
Resulting Condition Latches:

C: the result in the selected byte of R ≠ 0.

Z: the result in the selected byte of R = 0

AND Register

NR R1,R2 RR



The second operand (bytes 0 and 1, and byte X of R2) is ANDed with the first operand (bytes 0 and 1, and byte X, of the register specified by R1). Operands are treated as unstructured logical quantities, and the connective AND is applied bit by bit. A bit position in the result is set to 1 if the corresponding bit positions in both operands contains a 1; otherwise the result bit is set to 0. All bits of each operand participate in the operation. Any value in the operands or in the result is valid. The second operand is unchanged, and the condition latches are set according to the result of the operation.

Resulting Condition Latches:

C: the result in R1 ≠ 0.

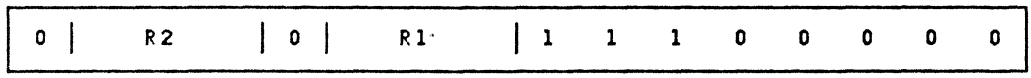
Z: the result in R1 = 0.

Programming Note

If register 0 is specified as R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

AND Halfword Register

NHR R1,R2 RR



The second operand (bytes 0 and 1, but not byte X of R2) is ANDed with the first operand (bytes 0 and 1, but not byte X, of the register specified by R1). Operands are treated as unstructured logical quantities, and the connective AND is applied bit by bit. A bit position in the result is set to 1 if the corresponding bit positions in both operands contains a 1; otherwise the result bit is set to 0. All bits of each operand participate in the operation. Any value in the operands or in the result is valid. The second

operand is unchanged, and the condition latches are set according to the result of the operation.

Resulting Condition Latches:

C: the result in bytes 0 and 1 of R1 ≠ 0.

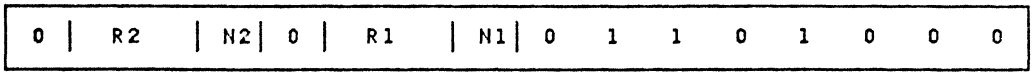
Z: the result in bytes 0 and 1 of R1 = 0.

Programming Note

If register 0 is specified as R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

AND Character Register

NCR R1(N1),R2(N2) RR



The second operand (R2, byte 0 if N2 = 0, or byte 1 if N2 = 1) is ANDed with the first operand (R1, byte 0 if N1 = 0, or byte 1 if N1 = 1). The result of the operation is placed in the first operand location; the remaining bytes of the register remain unchanged.

Operands are treated as unstructured logical quantities, and the connective AND is applied bit by bit. A bit position in the result is set to one if the corresponding bit position in both operands contains a 1; otherwise the result bit is set to zero. All bits of each operand participate in the operation. All values of operands and result are valid. The two bits of R1 and R2 form the two high-order bits of the register addresses; the low order bits are forced to 1 by hardware. For this reason, R1 and R2 are always odd-numbered registers (1, 3, 5, 7).

Resulting Condition Latches:

C: the result in the selected byte of R ≠ 0.

Z: the result in the selected byte of R = 0

Name	Instruction	Type	Format 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
B	Branch	RT	1	0	1	0	1	Displacement										
BCL	Branch on C latch	RT	1	0	0	1	1	Displacement										
BZL	Branch on Z latch	RT	1	0	0	0	1	Displacement										
BCT	Branch on Count	RT	1	0	1	1	1	R		N	1	Displacement						
BB	Branch on Bit	RT	1	1	M	M	1	R		N	M	Displacement						
BALR	Branch & Link Reg	RR	0	R2			0	R1			0	1	0	0	0	0	0	0
BAL	Branch and Link	RA	1	0	1	1	1	R			0	1	0	0	Ad Byte Ext			
			Addr Byte 0								Addr Byte 1							

Branch

B T RT

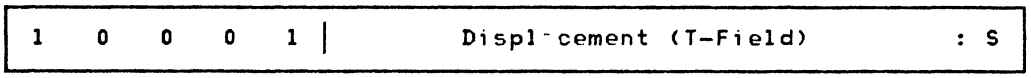
1	0	1	0	1	Displacement (T-field) : S
0			4	5	15

This instruction causes an unconditional branch to the branch address. The branch address is formed by adding the displacement value in the T field to the address of the next sequential instruction. Therefore, the branch address is always calculated relative to the next sequential instruction. The T-field allows a displacement of between +1023 and -1023 halfwords. The low order bit position (bit 15: marked S for sign) indicates whether the displacement is positive (0), or negative (1).

Resulting Condition Latches: unchanged

Branch on Z Latch

BZL T RT



This instruction tests the state of the Z condition latch associated with the active group of general registers. If the tested latch is not set (0), the next sequential instruction is executed. If the tested latch is set (1), the instruction located at the branch address is executed. The branch address is formed by adding the displacement value in the T field to the address of the next sequential instruction. Therefore, the branch address is always calculated relative to the next sequential instruction. The T-field allows a displacement of between +1023 and -1023 halfwords. The low order bit position (bit 15: marked S for sign) indicates whether the displacement is positive (0), or negative (1).

Resulting Condition Latches: unchanged

Branch on C Latch

BCL T RT

This instruction behaves in exactly the same way as the 'Branch on Z Latch' instruction, except that it is the C latch which is tested.

Branch on Count

BCT R(N),T RT

The count value contained in the register specified by R is decremented by one and then tested for zero. If the result is zero, the next sequential instruction is executed. If the result is not zero, the instruction located at the branch address is executed. The count is contained in byte 0 only (if N = 0) or in both bytes 0 and 1 (if N = 1) of the register. The branch address is formed by adding the displacement value in the T field to the address of the next sequential instruction. Therefore, the branch address is always calculated relative to the next sequential instruction. The T-field allows a displacement of between +63 and -63 halfwords. The low order bit position (bit 15: marked S for sign) indicates whether the displacement is positive (0), or negative (1).

The two bits of R form the two high-order bits of the register address; the low order bit is forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

Resulting Condition Latches: unchanged

Programming Notes

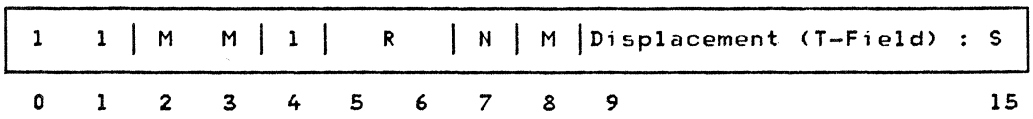
1. If N = 0, and before execution the count in byte 0 was zero, the effective count is 256.

2. If N = 1, and before execution the count in bytes 0 and 1 was zero, the effective count is 65536.

3. Byte X is ignored.

Branch on Bit

BB R(N,M),T RT



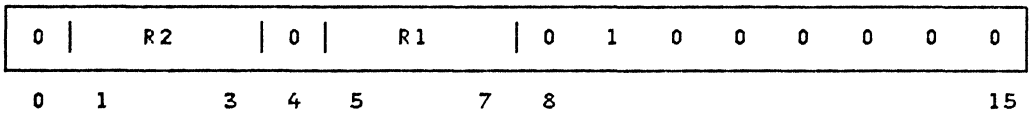
This instruction tests the state of a specified bit in a general register. The three bits of the M (mask) field specify which one of the 8 bits of byte 0 (N = 0) or byte 1 (N = 1) of the register is to be tested. If the bit tested is zero, the next sequential instruction is executed. If the bit tested is a one, the instruction located at the branch address is executed. The branch address is formed by adding the displacement value in the T field to the address of the next sequential instruction. Therefore, the branch address is always calculated relative to the next sequential instruction. The T-field allows a displacement of between +63 and -63 halfwords. The low order bit position (bit 15: marked S for sign) indicates whether the displacement is positive (0), or negative (1).

The two bits of R form the two high-order bits of the register address; the low order bit is forced to 1 by hardware. For this reason, R is always an odd-numbered register (1, 3, 5, 7).

Resulting Condition Latches: unchanged

Branch and Link Register

BALR R1,R2 RR



The address of the next sequential instruction (from the IAR, R0) is stored as link information in the register specified by R1. Subsequently, the instruction address in register 0 is replaced by the branch address (the contents of the register specified by R2), and the branch is executed. The branch address is obtained from R2 before the link information is stored in R1.

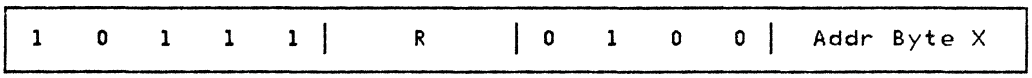
Programming Note

Since register 0 is the instruction address register (IAR), no linkage is provided if it is specified in the R1 field, and no branch occurs if it is specified in the R2 field.

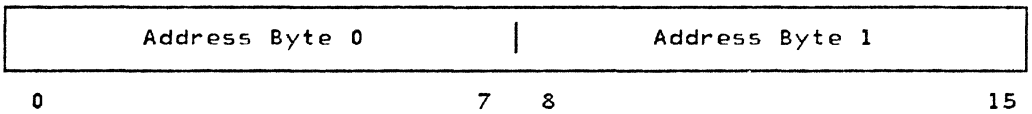
Branch and Link

BAL R,A RA

First halfword



Second halfword



The branch and link instruction is a 4-byte instruction which causes an unconditional branch. The address of the next sequential instruction is stored as link information in the register specified by R. The instruction address in register 0 is then replaced by the branch address taken from the 20-bit address field, and the branch is executed.

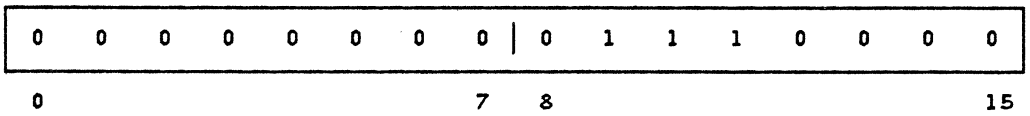
Resulting Condition Latches: unchanged

Programming Note

Since register 0 is the instruction address register, no linkage is provided if it is specified in the R field.

EXIT

EXIT - EXIT



The exit instruction is used to exit from the active program level. The interrupt priority logic then determines which group of general registers to select as the active group for the next program operation, and automatically loads the work registers with the contents of the active group. This instruction also resets the 'interrupt entered' latch for the program level that executes it.

If the exit instruction is executed at program level 5, the level 4 supervisor call interrupt request (SVC L4) is set. The next instruction is then normally the instruction at the starting address for program level 4. However, if other interrupt requests of higher priority are present, the next instruction executed is the instruction at the starting address of the highest priority program level requesting an interrupt.

Resulting Condition Latches: unchanged

Programming Notes

1. If program level 4 is masked, and no unmasked higher level interrupts are pending, program execution continues at level 5 after the SVC level 4 interrupt request is set.

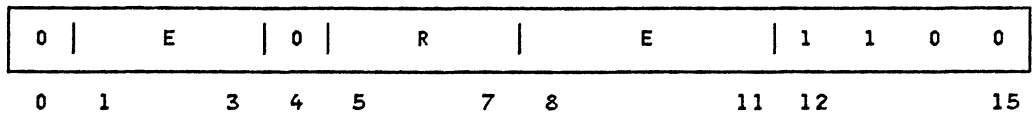
2. If this instruction follows an IOH or IOHI instruction (used to clear an adapter interrupt), at least 8 CCU cycles must separate the execution of the IOH/IOHI and the Exit instructions.

3. When executing at any interrupt level, if another interrupt is pending when the EXIT instruction is executed, the CCU momentarily returns to the original interrupted level without instruction execution, and then goes to the new interrupt level. This action is invisible, except if branch tracing is done; in this case, the passage through the original interrupted level is recorded in the branch trace list.

Name	Instruction	Type	Format
IN	CCU Reg Input	RE	0 | E | 0 | R | E | 1 1 0 0
OUT	CCU Reg Output	RE	0 | E | 0 | R | E | 0 1 0 0
IOH	Adapter I/O	RR	0 | R2 | 0 | R1 | 0 1 0 1 0 0 0 0
IOHI	Adapter I/O Immed	RI	0 0 0 0 0 | R | 0 1 1 1 0 0 0 0
			(Second half word depends on the adapter)

CCU Register Input

IN R,E RE



The input instruction loads the general register specified by R with the contents of one of the 128 CCU registers, as specified by the E field. Throughout this manual, CCU Register Input instructions are referred to in the form: Input X'nn' where 'nn' is the hexadecimal address of the CCU register. See the Appendixes to this manual for a list of the external registers and for their detailed bit structure.

Resulting Condition Latches: unchanged

Programming Notes

1. If register 0 of the active group of general registers is specified by R, this instruction causes a branch to the address formed in register 0.

2. This instruction is a privileged instruction, executable only at program levels 1, 2, 3, or 4. Any attempt to execute the instruction at program level 5 causes a level 1 interrupt request (Level 5 I/O Error) to be set.

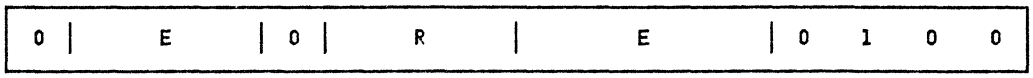
3. The following input external register addresses are invalid: X'28' through X'2F', X'49' through X'4F', X'60' through X'6F', and X'78'. If the control program tries to read one of these registers, an invalid

operation condition is detected, and a level 1 interrupt is set. The instruction is not executed.

4. If the control program uses an IOH or IOHI instruction to clear an adapter interrupt, at least 8 CCU cycles must be performed before executing an Input X'77' or an Input X'7E' instruction.

CCU Register Output

OUT R,E RE



The output instruction loads one of the 128 CCU registers specified by the E field with the contents of the general register specified by R. Throughout this manual, CCU Register Output instructions are referred to in the form: Output X'nn' where 'nn' is the hexadecimal address of the CCU register. All registers are three bytes long. See the Appendixes to this manual for a list of the CCU registers and for their detailed bit structure.

Resulting Condition Latches: unchanged

Programming Notes

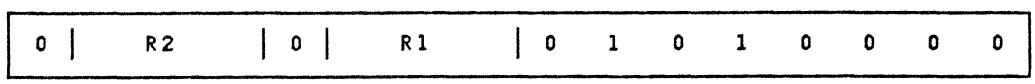
1. If register 0 of the active group of general registers is specified by E, this instruction causes a branch to the address formed in register 0.

2. This instruction is a privileged instruction, executable only at program levels 1, 2, 3, or 4. Any attempt to execute the instruction at program level 5 causes a level 1 interrupt request (Level 5 I/O Error) to be set.

3. The following output external register addresses are invalid: X'28' through X'2F', X'49' through X'4F', X'60' through X'6F', and X'75'. If the control program tries to write into one of these registers, an invalid operation condition is detected, and a level 1 interrupt is set. The instruction is not executed.

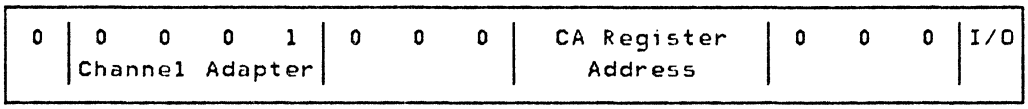
Adapter Input/Output

IOH R1,R2 RR



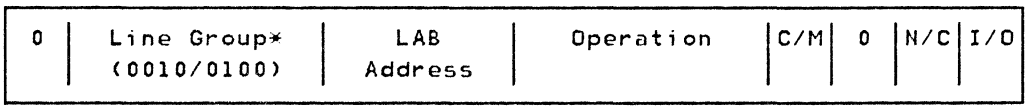
This instruction transfers the contents of the register specified by R1 to the channel adapter/communication scanner, or places information coming from the channel adapter/communication scanner into the register specified by R1. The adapter, the adapter command or register, and the direction of data movement are all specified by the contents of R2. This instruction cannot be used to address registers within the CCU. For the instruction to execute correctly, R2 must be loaded as follows:

Channel Adapter



I/O = input/output bit: 0 = output, 1 = input

Communication Scanners



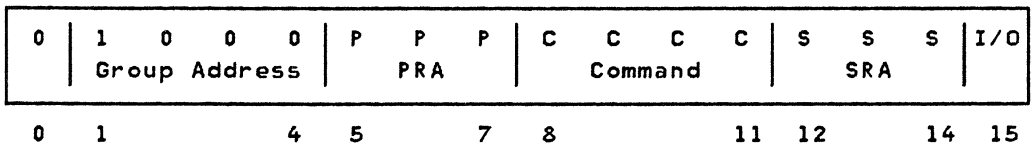
* 0010 selects the first group of 16 lines; 0100 selects the second group. In the case of LAB 2, this division corresponds to the two communication scanners.

C/M = CCU/MOSS bit: 0 = initiated by CCU, 1 = initiated by MOSS

N/C: 0 = normal mode, 1 = character mode

I/O = input/output bit: 0 = output, 1 = input

Redrive Logic



Bits 1 through 4 contain the group address (always 1 0 0 0)

Bits 5 through 7 contain the primary redrive address (PRA)

Bits 8 through 11 indicate the redrive command

Bits 12 through 14 contain the secondary redrive address (SRA)

I/O = input/output bit: 0 = output, 1 = input

Resulting Condition Latches:

C: the exception line was raised by the adapter

Z: the exception line was not raised by the adapter

Programming Notes

1. This instruction is a privileged instruction, executable only at program levels 1, 2, 3, or 4. Any attempt to execute it at program level 5 causes a level 1 input/output check interrupt request (level 5 I/O error) to be set.

2. A time out level 1 interrupt request occurs if no valid response is received from the adapter within a specified time.

3. Byte X of register R2 is not used.

4. Byte X of register R1 is set to all zeros if the operation is read.

5. If register 0 is specified in the R1 field, an Invalid OP Check L1 interrupt request occurs; the instruction is not executed.

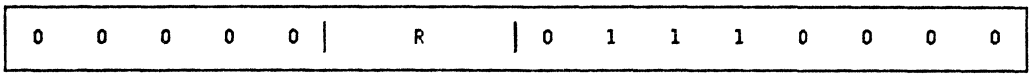
6. If the R2 field is 0, CCU external register X'48' (IOH Address Substitution) is used in place of register 0 to specify the external adapter register.

7. If this instruction is used to clear an interrupt and is followed by an Exit instruction, at least 8 CCU cycles must separate the execution of the IOH and the Exit instructions.

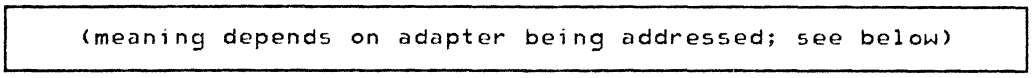
Adapter Input/Output Immediate

IOHI R,A RA

First halfword

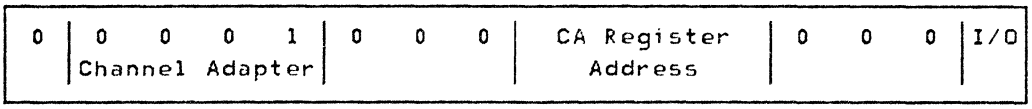


Second halfword



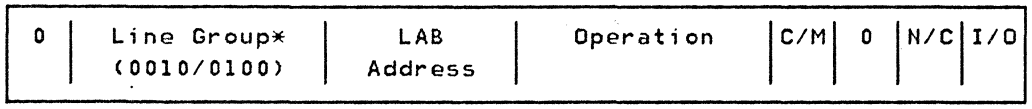
This instruction transfers the contents of the register specified by R to the channel adapter/communication scanner, or places information coming from the channel adapter/communication scanner into the register specified by R. The adapter, the adapter command or register, and the direction of data movement are all specified by the contents of the second halfword. This instruction cannot be used to address registers within the CCU. For the instruction to execute correctly, the second halfword must be composed as follows:

Channel Adapter



I/O = input/output bit: 0 = output, 1 = input

Communication Scanners



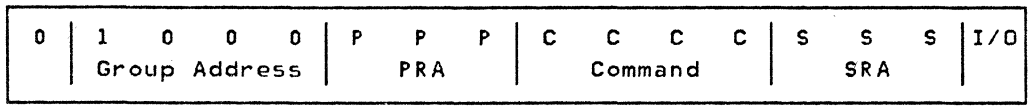
* 0010 selects the first group of 16 lines; 0100 selects the second group. In the case of LAB 2, this division corresponds to the two communication scanners.

C/M = CCU/MOSS bit: 0 = initiated by CCU, 1 = initiated by MOSS

N/C: 0 = normal mode, 1 = character mode

I/O = input/output bit: 0 = output, 1 = input

Redrive Logic



Bits 1 through 4 contain the group address (always 1 0 0 0)

Bits 5 through 7 contain the primary redrive address (PRA)

Bits 8 through 11 indicate the redrive command

Bits 12 through 14 contain the secondary redrive address (SRA)

I/O = input/output bit: 0 = output, 1 = input

Resulting Condition Latches:

C: the exception line was raised by the adapter

Z: the exception line was not raised by the adapter

Programming Notes

1. This instruction is a privileged instruction, executable only at program levels 1, 2, 3, or 4. Any attempt to execute it at program level 5 causes a level 1 input/output check interrupt request (level 5 I/O error) to be set.

2. A time out level 1 interrupt request occurs if no valid response is received from the adapter within a specified time.

3. Byte X of register R is not used, and is set to all zeros if the operation is read.

4. If register 0 is specified in the R field, an EXIT operation occurs.

5. If this instruction is used to clear an interrupt and is followed by an Exit instruction, at least 8 CCU cycles must separate the execution of the IOHI and the Exit instructions.

CHAPTER 4. CENTRAL CONTROL UNIT (CCU)

This chapter is intended to give the reader a basic understanding of the operation of the Central Control Unit (CCU) and the requirements necessary to control its operation.

The CCU is interrupt driven, that is, it operates exclusively in response to interrupts coming from the channel adapters, the communication scanners, the MOSS, and from other program levels via program controlled interrupts (PCIs). However, the interrupt system is not described here; for an overview, refer to the sections 'Program Levels' and 'Interrupts' in Chapter 2. For details of the channel adapter interrupts, refer to chapter 5.

The CCU contains the circuits and data flow paths needed for the following operations:

- Accept interrupts
- Execute the instruction set
- Address storage
- Perform arithmetic and logical processing of data
- Control the attached adapters

Operation of the CCU is under the control of the programs in storage.

The data flow in the CCU is implemented in hardware, controlled by the control program. The data flow for a particular operation is determined by the instruction, cycle steal, or control operation that is being executed.

CCU REGISTERS

The CCU contains a number of registers, most of which are accessible to the program. The bits of these registers are described in detail under the heading CCU Input/Output Instructions in this chapter.

OPERATION REGISTER

The operation register holds the first 16 bits of the instruction currently being executed. It is not available to the program, but may be displayed on the screen.

STORAGE ADDRESS REGISTER (SAR)

The storage address register holds the storage address that is being used, or that was last used. It is not available to the program, but may be displayed on the screen.

CCU WORK REGISTERS

The CCU work registers are a group of 8 hardware registers for the immediate use of the CCU. They may all be used by the program without restriction, with the exception of register 0 which is used as the Instruction Address Register.

Instruction Address Register (IAR)

The IAR is an implied base register, and always contains the address of the next instruction to be executed. It is always incremented to point to the next sequential instruction before the current instruction is executed. In most cases, the next halfword in storage contains the next instruction to be executed. Sometimes, however, the contents of the IAR are changed as the result of the instruction being executed. Execution of a branch instruction, for example, can cause the IAR to be loaded with a storage address other than the next sequential instruction. Refer to the descriptions of the individual instructions in Chapter 3 for the results of using register 0, and the precautions to be taken.

CCU EXTERNAL REGISTERS

The CCU register addressing scheme can address up to 128 registers; however, not all of these register addresses are used.

Note that some instructions can address two different registers, one of which is only used for input, the other being used only for output. For example, the instruction Input X'71' reads the contents of local storage position X'71', but Output X'71' sets hardware display register 1.

In addition, some instructions do not address a register at all. This is the case of Output X'7B', for example, which forces a program controlled interrupt at level 2.

The table below shows the name and address of the CCU extèrnal registers:

Code	Meaning	Ext reg addr	Type
00-27	General Registers	LS X'00'-X'27'	I/O
30-35	Channel Adapter CS Address Pointers	LS X'30'-X'35'	I/O
36-3E	Reserved Pointer Registers	LS X'36'-X'3E'	I/O
3F	Communication Scanner CS Address Pointer	LS X'3F'	I/O
40-43	Interrupt Start Addresses	LS X'40'-X'43'	I/O
44	Byte Operations Base Register	LS X'44'	I/O
45	Halfword Operations Base Register	LS X'45'	I/O
46	Fullword Operations Base Register	LS X'46'	I/O
48	IOH Address Substitution Register	LS X'48'	I/O
50-5F	Programmable Registers	LS X'50'-X'5F'	I/O
70	Storage Size Installed	HW register	In
71	Operator Address/Data Entry Register	LS X'71'	In
71	Display Register 1	HW register	Out
72	Operator Display/Function Select Control	LS X'72'	In
72	Display Register 2	HW register	Out
73	Insert Storage Protect/Address Exception Key	HW register	In
73	Set Storage Protect/Address Exception Key	HW register	Out
74	Lagging Address Register	HW register	In
75	CCW for AIO Operations	HW register	In
76	CCU Level 1 Interrupt Requests on I/O ops.	HW register	In
76	Miscellaneous Control 1	HW register	Out
77	Adapter Levels 2 and 3 Interrupt Requests	HW register	In
77	Miscellaneous Control 2	HW register	Out
79	Utility	HW register	In
79	Utility	HW register	Out
7A	High Res. Timer/Utilization Counter	HW register	In
7A	High Res. Timer/Utilization Counter Control	HW register	Out
7B	Branch Trace Address Pointer	LS X'7B'	In
7C	Branch Trace Buffer Count	LS X'7C'	In
7D	CCU Hardware Check Register	HW register	In
7E	CCU Level 1 Interrupt Requests	HW register	In
7E	Set Program Interrupt Mask Bits	HW register	Out
7F	CCU L2, 3, or 4 Interrupt Requests	HW register	In
7F	Reset Program Interrupt Mask Bits	HW register	Out

Note: output instructions corresponding to codes X'70', X'78', X'7B', X'7C', and X'7D' exist, but do not correspond to any register; instead, they perform a function.

General Registers

The first 40 external registers (00 through 27) are called the General Registers. They are divided into five groups, numbered 0 through 4, and each containing 8 registers numbered 0 through 7. Each group is assigned to a specific program level.

The first general register of each group (register 0) is used as the instruction address register (IAR) of the corresponding program level. As the program is executed, the 8 local storage registers (with the exception of the IAR) are updated with the contents of the work registers when the register is used in the execution of an instruction. Refer to figure 2-3 for the relationship between the work registers and the general registers.

Programming Notes:

1. When a given program interrupt level is entered, the work registers must be initialized as required, with the exception of register 0. Register 0 contains the IAR, and is, of course, always correct at the entry to the interrupt.

2. In general, when a given level is EXITed, the work registers are restored automatically to the values they contained before the interrupt.

3. The other groups of registers (corresponding to the non-active program levels) cannot be accessed via the usual instructions. They must be accessed via the CCU Register In/Out instructions using the following addresses:

General register group (in local storage)	External (and LS) hexadecimal addresses
0 (program level 2)	00 through 07
1 (program level 3)	08 through 0F
2 (program level 4)	10 through 17
3 (program level 5)	18 through 1F
4 (program level 1)	20 through 27

LOCAL STORAGE MAP

LS Addr	Register Functions	Accessed by:
00 - 07	General Register Group 0 (Interrupt Level 2)	In/Out 00-07
08 - 0F	General Register Group 1 (Interrupt Level 3)	In/Out 08-0F
10 - 17	General Register Group 2 (Interrupt Level 4)	In/Out 10-17
18 - 1F	General Register Group 3 (Interrupt Level 5)	In/Out 18-1F
20 - 27	General Register Group 4 (Interrupt Level 1)	In/Out 20-27
28 - 2F	Invalid	
30	Cycle Steal Address Pointer Register - CA 1	In/Out 30
31	" 2	In/Out 31
32	" 3	In/Out 32
33	" 4	In/Out 33
34	" 5	In/Out 34
35	" 6	In/Out 35
36 - 3E	Pointer Registers 6 through E	In/Out 36-3E
3F	Communication Scanner CS Address Pointer Reg.	In/Out 3F
40	Interrupt start address - Level 1	In/Out 40
41	" " 2	In/Out 41
42	" " 3	In/Out 42
43	" " 4	In/Out 43
44	Byte Operations Base Register	In/Out 44
45	Halfword Operations Base Register	In/Out 45
46	Fullword Operations Base Register	In/Out 46
47	(reserved)	
48	IOH TA substitution register	In/Out 48
49 - 4F	Invalid	
50 - 5F	Programmable registers	In/Out 50-5F
60 - 6F	Invalid	
70	Not used - address corresponds to a H/W register	
71	Operator address/data entry	In 71
72	Operator function select control	In 72
73	Not used - address corresponds to a H/W register	
74	(reserved)	
75 - 7A	Not used - addresses correspond to H/W registers	
7B	Branch trace address pointer	In 7B
7C	Branch trace buffer count	In 7C
7D - 7F	Not used - addresses correspond to H/W registers	

CCU INPUT/OUTPUT INSTRUCTIONS

Most CCU input/output instructions address specific registers, although some only execute a function without using the contents of a register. In the text below, to avoid dispersion, the contents of the register is described in detail after the instruction which addresses it.

Type	Code	Meaning	Ext reg addr
I/O	00-27	General Registers	LS X'00'-X'27'
I/O	30-35	Channel Adapter CS Address Pointers	LS X'30'-X'35'
I/O	36-3E	Reserved Pointer Registers	LS X'36'-X'3E'
I/O	3F	Communication Scanner CS Address Pointer	LS X'3F'
I/O	40-43	Interrupt Start Addresses	LS X'40'-X'43'
I/O	44	Byte Operations Base Register	LS X'44'
I/O	45	Halfword Operations Base Register	LS X'45'
I/O	46	Fullword Operations Base Register	LS X'46'
I/O	48	IOH Address Substitution Register	LS X'48'
I/O	50-5F	Programmable Registers	LS X'50'-X'5F'
In	70	Storage Size Installed	HW In X'70'
Out	70	Hardstop	Function (1)
In	71	Operator Address/Data Entry Register	LS X'71'
Out	71	Display Register 1	HW Out X'71'
In	72	Operator Display/Function Select Control	LS X'72'
Out	72	Display Register 2	HW Out X'72'
In	73	Insert Storage Protect/Address Exception Key	HW In X'73'
Out	73	Set Storage Protect/Address Exception Key	HW Out X'73'
In	74	Lagging Address Register	HW In X'74'
In	75	CCW for AIO Operations	HW In X'75'
In	76	CCU Level 1 Interrupt Requests on I/O ops.	HW In X'76'
Out	76	Miscellaneous Control 1	HW Out X'76'
In	77	Adapter Levels 2 and 3 Interrupt Requests	HW In X'77'
Out	77	Miscellaneous Control 2	HW Out X'77'
Out	78	Force ALU Checks	Function (1)
In	79	Utility	HW In X'79'
Out	79	Utility	HW Out X'79'
In	7A	High Res. Timer/Utilization Counter	HW In X'7A'
Out	7A	High Res. Timer/Utilization Counter Control	HW Out X'7A'
In	7B	Branch Trace Address Pointer	LS X'7B'
Out	7B	Set PCI Level 2	Function (1)
In	7C	Branch Trace Buffer Count	LS X'7C'
Out	7C	Set PCI Level 3	Function (1)
In	7D	CCU Hardware Check Register	HW In X'7D'
Out	7D	Set PCI Level 4	Function (1)
In	7E	CCU Level 1 Interrupt Requests	HW In X'7E'
Out	7E	Set Program Interrupt Mask Bits	HW Out X'7E'
In	7F	CCU L2, 3, or 4 Interrupt Requests	HW In X'7F'
Out	7F	Reset Program Interrupt Mask Bits	HW Out X'7F'

Notes:

1. These instructions perform a function.

2. Input/Output instructions not listed in this table are invalid.

INPUT/OUTPUT X'00' THROUGH X'27' (GENERAL REGISTERS)

A given program level can only access its own group of general registers directly. It can get access to the general registers associated with the other program levels by means of the Input/Output X'00' through X'27' instructions. The bit assignments of these registers are, in general, not fixed, but vary with the use of the register. There is however one exception: the first register of each group always contains the address of the next sequential instruction in that interrupt level.

INPUT/OUTPUT X'28' THROUGH X'2F' (RESERVED)

The 8 registers addressed by these instructions are reserved. If the control program tries to read or write one of these registers, an invalid operation condition is detected, and a level 1 interrupt request is set.

INPUT/OUTPUT X'30' THROUGH X'35' (CHANNEL ADAPTER CS ADDRESS POINTERS)

These registers are used as channel adapter cycle steal address pointer registers (CSARs). They are assigned as follows:

Register	Channel adapter
X'30'	1
X'31'	2
X'32'	3
X'33'	4
X'34'	5
X'35'	6

Programming Note:

The contents of the cycle steal address pointer register are unpredictable at the end of any data transfer, and should not be used to determine the exact number of bytes that were transferred. The exact number of bytes transferred may be determined via an Input X'C' instruction.

INPUT/OUTPUT X'36' THROUGH X'3E' (RESERVED POINTER REGISTERS)

The registers addressed by these instructions are reserved, and should not be used by the program; however, they are not invalid.

INPUT/OUTPUT X'3F' (COMMUNICATION SCANNER CS ADDRESS POINTER)

This register is used as the communication scanner cycle steal address pointer register (CSAR).

Programming Note: this register is normally set automatically by a communication scanner before it starts a cycle steal operation. It should not normally be set by the CCU.

INPUT/OUTPUT X'40' THROUGH X'43' (INTERRUPT START ADDRESSES)

In previous communication controllers, the interrupt start addresses were fixed by hardware. In the 3725, the start addresses of the four interrupt levels are contained in the registers numbered X'40' through X'43, and may be set by the control program via the Input/Output X'40' through X'43' instructions. The Output instruction is used to load a register, and the Input instruction to examine it. The bit structure is of the normal 3-byte address type (bytes X, 0, and 1).

INPUT/OUTPUT X'44' (BYTE OPERATIONS BASE REGISTER)

The register addressed by these instructions is used by the 'Insert Character' and 'Store Character' instructions when the base register field (B-field) is defined as '0'. Under these conditions, the contents of external register X'44' is used as the base register instead of register 0.

INPUT/OUTPUT X'45' (HALFWORD OPERATIONS BASE REGISTER)

The register addressed by these instructions is used by the 'Load Halfword' and 'Store Halfword' instructions when the base register field (B-field) is defined as '0'. Under these conditions, the contents of external register X'45' is used as the base register instead of register 0.

INPUT/OUTPUT X'46' (FULLWORD OPERATIONS BASE REGISTER)

The register addressed by these instructions is used by the 'Load' and 'Store' instructions when the base register field (B-field) is defined as '0'. Under these conditions, the contents of external register X'46' is used as the base register instead of register 0.

INPUT/OUTPUT X'48' (IOH ADDRESS SUBSTITUTION REGISTER)

The register addressed by these instructions is used by the 'IOH' instruction when the R2 field is defined as '0'. Under these conditions, the contents of external register X'48' is used to define the address of the external adapter instead of the contents of register 0.

INPUT/OUTPUT X'49' THROUGH X'4F' (RESERVED)

The registers addressed by these instructions are reserved. If the control program tries to read or write one of these registers, an invalid operation condition is detected, and a level 1 interrupt request is set.

INPUT/OUTPUT X'50' THROUGH X'5F' (PROGRAMMABLE REGISTERS)

The registers addressed by these instructions have no specific functions and are available for use by the program.

INPUT/OUTPUT X'60' THROUGH X'6F' (RESERVED)

The registers addressed by these instructions are reserved. If the control program tries to read or write one of these registers, an invalid operation condition is detected, and a level 1 interrupt request is set.

INPUT X'70' (STORAGE SIZE INSTALLED)

This instruction causes the register specified by R to be loaded with a bit combination that indicates the amount of storage installed in the controller.

The register bits are set as shown in the table below:

Byte	Bit	Meaning
0	0	0
	1	1
	2	0
	3	2048 K
	4	1024 K
	5	512 K
	6	256 K
	7	Storage not a multiple of 256K
1	0-7	0

The effect of bits 4 through 6 is additive, so that a machine with 768K would have byte 1, bit 4 off and bits 5 and 6 on.

Byte 0, Bit 7 - Storage not a Multiple of 256K: this bit, when on, indicates that the storage configuration is non-standard, that is, it is not a multiple of 256k bytes.

OUTPUT X'70' (HARDSTOP)

This instruction causes the controller to enter the 'hardstop' state. In this state, program execution, program interrupts, and adapter cycle stealing are blocked. Since this instruction performs a function, the bit settings of the general register are ignored.

Note: there is no way to leave the hardstop state except via the MOSS or IPL.

INPUT X'71' (OPERATOR ADDRESS/DATA ENTRY REGISTER)

This instruction causes the register specified by R to be loaded with the contents (three bytes: X, 0, 1) of the operator address/data entry register. This data is used in control panel functions. The operator address/data entry register is itself set by the operator via the MOSS. The bits have the following meaning:

Byte	Bit	Meaning
X	2-7	Operator address/data register byte X, bits 2-7
0	0-7	Operator address/data register byte 0, bits 0-7
1	0-7	Operator address/data register byte 1, bits 0-7

OUTPUT X'71' (DISPLAY REGISTER 1)

This instruction transfers the contents of the register specified by R (three bytes: X, 0, 1) to display register 1; at the same time, the MOSS is informed that the register must be displayed. The bits of this register have the following meaning:

Byte	Bit	Meaning
X	2-7	Display register 1 byte X, bits 2-7
0	0-7	Display register 1 byte 0, bits 0-7
1	0-7	Display register 1 byte 1, bits 0-7

Notes:

1. This instruction should not be executed more than once every 500 milliseconds. This is to allow the MOSS to control the 3725 correctly.

2. When a message is displayed on the MOSS display via the X'71' (or X'72') register, the keyboard is locked for the duration of the display function. For this reason, Output X'71' (or X'72') should be used with care (for example, only after one of the registers is updated) to avoid lowering the availability of the keyboard.

INPUT X'72' (OPERATOR DISPLAY/FUNCTION SELECT CONTROL)

This instruction causes the register specified by R to be loaded with the contents (two bytes: 0, 1) of the operator display/function select register. The operator function select register is itself set by the operator via the MOSS. The bits of this register have the following meaning:

Byte	Bit	Meaning
0	0	Function select 8
	1	Function select 9
	2	Function select 10
	3	Function select 11 (storage address)
	4	Function select 12 (register address)
	5	Function select 13
	6	Function select 14
	7	Function select 15
1	0	Function select 16
	1	Function select 1
	2	Function select 2
	3	Function select 3
	4	Function select 4
	5	Function select 5
	6	Function select 6
	7	Function select 7

Byte 0, bit 0: this bit indicates that function 8 has been selected by the MOSS operator.

Byte 0, bit 1: this bit indicates that function 9 has been selected by the MOSS operator.

Byte 0, bit 2: this bit indicates that function 10 has been selected by the MOSS operator.

Byte 0, bit 3: this bit indicates that function 11 (storage address) has been selected by the MOSS operator.

Byte 0, bit 4: this bit indicates that function 12 (register address) has been selected by the MOSS operator.

Byte 0, bit 5: this bit indicates that function 13 has been selected by the MOSS operator.

Byte 0, bit 6: this bit indicates that function 14 has been selected by the MOSS operator.

Byte 0, bit 7: this bit indicates that function 15 has been selected by the MOSS operator.

Byte 1, bit 0: this bit indicates that function 16 has been selected by the MOSS operator.

Byte 1, bit 1: this bit indicates that function 1 has been selected by the MOSS operator.

Byte 1, bit 2: this bit indicates that function 2 has been selected by the MOSS operator.

Byte 1, bit 3: this bit indicates that function 3 has been selected by the MOSS operator.

Byte 1, bit 4: this bit indicates that function 4 has been selected by the MOSS operator.

Byte 1, bit 5: this bit indicates that function 5 has been selected by the MOSS operator.

Byte 1, bit 6: this bit indicates that function 6 has been selected by the MOSS operator.

Byte 1, bit 7: this bit indicates that function 7 has been selected by the MOSS operator.

OUTPUT X'72' (DISPLAY REGISTER 2)

This instruction transfers the contents of the register specified by R (three bytes: X, 0, 1) to display register 2; at the same time, the MOSS is informed that the register must be displayed. The bits of this register have the following meaning:

Byte	Bit	Meaning
X	2-7	Display register 2 byte X, bits 2-7
0	0-7	Display register 2 byte 0, bits 0-7
1	0-7	Display register 2 byte 1, bits 0-7

Notes:

1. This instruction should not be executed more than once every 500 milliseconds. This is to allow the MOSS to control the 3725 correctly.

2. When a message is displayed on the MOSS display via the X'72' (or X'71') register, the keyboard is locked for the duration of the display function. For this reason, Output X'72' (or X'71') should be used with care (for example, only after one of the registers is updated) to avoid lowering the availability of the keyboard.

INPUT X'73' (INSERT STORAGE PROTECT/ADDRESS EXCEPTION KEY)

This instruction is associated with storage protection. It causes the key that was addressed by the last Output X'73' (Set Key) instruction to be loaded into byte 1, bits 5-7 of the register specified by R as shown in the table below:

Byte	Bit	Meaning
0	0-7	(not used)
1	0	(not used)
	1	(not used)
	2	(not used)
	3	(not used)
	4	(not used)
	5	Key Bit 0
	6	Key Bit 1
	7	Key Bit 2

Programming Notes:

1. If the last Output X'73' instruction addressed a storage key, an exception key, or a read-only key, the next Input X'73' instruction will read back this key, even though the key was not set (byte 1, bit 4 off).

2. If the last Output X'73' instruction addressed a user protect key, the next Input X'73' instruction reads back the key that was set by the last Output X'73' instruction that changed the user protect key. All other Output X'73' instructions have no effect.

OUTPUT X'73' (SET STORAGE PROTECT/ADDRESS EXCEPTION KEY)

This instruction is associated with storage protection. The bits of this register have the following meaning:

Byte	Bit	Meaning
X	2	Storage key address bit 0
	3	Storage key address bit 1
	4	Storage key address bit 2
	5	Storage key address bit 3
	6	Storage key address bit 4
	7	Storage key address bit 5
0	0	Storage key address bit 6
	1	Storage key address bit 7
	2	Storage key address bit 8
	3	Storage key address bit 9/user key address bit 0
	4	Storage key address bit 10/user key address bit 1
	5	User key address bit 2
	6	User key address bit 3
	7	User key address bit 4
1	0	(not used)
	1	Enable storage protect/address exception
	2	Key type bit 0
	3	Key type bit 1
	4	Modify key value
	5	Key bit 0
	6	Key bit 1
	7	Key bit 2

Byte X, Bits 2 through 7, and Byte 0, Bits 0 through 7 - Address: these bits contain the address of the key to be set. The storage, address exception, and read-only keys require a 14-bit address, and the user key a 5-bit address. The use of these addresses is described below under the heading 'CCU Special Topics', section 'Storage Protect'.

Byte 1, Bit 1 - Enable Storage Protect/Address Exception: this bit, when on, enables the storage protect/address exception mechanism. Once enabled, there is no way to disable storage protect/address exception except via a power off, or a power on reset.

Byte 1, Bits 2 and 3 - Key Type: these bits select the type of key as shown in the following table:

Byte 1 bit		
2	3	Meaning
0	0	User protect key
0	1	Storage key
1	0	Exception key
1	1	Read-only key

Byte 1, Bit 4 - Modify Key Value: this bit, when on, indicates that the addressed key is to be set with the key bits contained in byte 1, bits 5-7. If the bit is off, the addressed key is not set.

Byte 1, Bits 5 through 7 - Key Bits: these bits indicate the key that is to be set in the addressed location.

INPUT X'74' (LAGGING ADDRESS REGISTER)

This instruction causes the register specified by R to be loaded with the contents (three bytes: X, 0, 1) of the lagging address register (LAR).

The LAR is essentially a 'came from' register. When displayed by the operator or by the program, it contains the address of the last instruction that was being executed prior to the current instruction (if any). The LAR is loaded from the instruction address register (IAR) at the beginning of each instruction.

The program may load the contents of the LAR into a general register by executing the instruction. From there, the program may use the LAR directly, or it may display it on the control panel by using the general register as input to the display register.

Programming Note

The address contained in the LAR is that of the last instruction executed immediately before the Input X'74'. In order to preserve the contents of the LAR after a level 1 interrupt, the first instruction executed after entering level 1 should be an Input X'74' instruction.

Contents of the LAR after an Unusual Condition

In normal operation, the LAR behaves as described above. However, certain check conditions and control panel operations may give a different result in the LAR:

1. Invalid Op Code Check: the LAR contains the address of the last instruction executed before the one that caused the check.

2. Input/Output Check at Level 5: the LAR contains the address of the last instruction executed before the one that caused the check.

3. Storage Protect Check on Instruction Fetch: the LAR contains the address of the last instruction executed before the one that caused the check.

4. Address Exception Check on Instruction Fetch: the LAR contains the address of the last instruction executed before the one that caused the check.

5. Storage Protect Check on Store or Load Instruction: the LAR contains the address of the instruction that caused the check, or this address incremented by 2.

6. Address Exception Check on Store or Load Instruction: the LAR contains the address of the instruction that caused the check, or this address incremented by 2.

7. Adapter Interconnection Check on IOH or IOHI Instruction: the LAR contains the address of the IOH or IOHI instruction that caused the check.

8. Adapter Check: the contents of the LAR is not predictable.

9. Adapter Interconnection Check during Adapter Cycle Steal: the contents of the LAR is not predictable.

10. I-Fetch Address Compare Stop/Interrupt: all instructions except IOH/IOHI are executed before the address compare stop/interrupt occurs. The LAR therefore contains the address of the instruction that was executed before the one that caused the address compare.

    Note: IOH/IOHI instructions may or may not be executed before the address compare stop/interrupt. Because of the impossibility of knowing whether or not the IOH/IOHI instruction has been executed, it is recommended not to set the compare address to either of these two instructions.

11. Instruction Access to Storage Address Compare Stop/Interrupt: the LAR contains the address of the instruction that loaded from or stored into the indicated storage location.

12. Instruction Step Mode: the LAR contains the address of the last instruction executed.

13. Program Stop Mode: the LAR contains the address of the last instruction executed.

INPUT X'75' (CCW FOR AIO OPERATIONS)

This instruction causes the register specified by R to be loaded with certain bits of the CCW when the current AIO operation causes an error condition which stops the AIO on the I/O bus. This allows the control program and the MOSS to determine which adapter caused the error condition. This register is not writable by the control program. If the control program tries to write this register, an invalid operation condition is detected, and a level 1 interrupt request is set. The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	CCW Bit 5 (0 = CA AIO, 1 = scanner AIO)
	1	CCW Bit 11 (pointer no./scanner address bit 0)
	2	CCW Bit 12 (pointer no./scanner address bit 1)
	3	CCW Bit 13 (pointer no./scanner address bit 2)
	4	CCW Bit 14 (pointer no./scanner address bit 3)
	5	(not used)
	6	(not used)
	7	(not used)
1	0-7	(not used)

If byte 0, bit 0 = 0, the CCW has come from the channel adapter, and byte 0, bits 1 through 4 contain a pointer number indicating one of the 6 channel adapter cycle steal address pointer registers:

Register	Channel adapter
X'30'	1
X'31'	2
X'32'	3
X'33'	4
X'34'	5
X'35'	6

If byte 0, bit 0 = 1, the CCW has come from the communication scanner, and byte 0, bits 1 through 4 contain the address of the scanner. In this case, register X'3F' (Cycle Steal Address Register) is used as the communication scanner cycle steal (common) address pointer register.

INPUT X'76' (CCU LEVEL 1 INTERRUPT REQUESTS ON I/O OPERATIONS)

The register addressed by this instruction contains information identifying the source of a level 1 interrupt request. The bits of this register have the following meaning:

Byte	Bit	Meaning
0	0	Addressing exception during I/O operations
	1	Storage protection check during I/O operations
	2	Invalid CCW during I/O operations
	3	(not used)
	4	Time out condition
	5	Bus in parity check
	6	Adapter initiated operation
	7	MOSS initiated operation
1	0-7	(not used)

Note: if either or both of bits 4 and 5 of byte 0 is on, bits 0 through 3 of byte 0 have a special meaning for maintenance purposes, and should normally be ignored. For the special meaning of these bits, refer to the maintenance documentation.

OUTPUT X'76' (MISCELLANEOUS CONTROL 1)

The register addressed by this instruction contains miscellaneous control information, used mainly to set/reset interrupt requests. The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	Reset errors detected during I/O operations
	1	(not used)
	2	(not used)
	3	Control program to MOSS request
	4	Control program to MOSS response
	5	(not used)
	6	(not used)
	7	(not used)
1	0-7	(not used)

Byte 0, Bit 0 - Reset Errors Detected during I/O this bit, when on, resets all CCU errors that were detected during I/O operations.

Byte 0, Bit 3 - Control Program to MOSS Request: this bit, when on, raises a MOSS interrupt to inform the MOSS that the control program requires the execution of a MOSS function as defined in the CCU to MOSS Request Control Block (Mailbox) in CCU storage.

Byte 0, Bit 4 - Control Program to MOSS Response: this bit, when on, raises a MOSS interrupt to inform the MOSS that the control program has executed a request from the MOSS, and that the result is available in the CCU to MOSS Response Control Block (Mailbox) in CCU storage.

INPUT X'77' (ADAPTER LEVELS 2 AND 3 INTERRUPT REQUESTS)

The register addressed by this instruction contains information about the source of interrupts at levels 2 and 3. The bits of this register have the following meaning:

Byte	Bit	Meaning
0	0	(not used)
	1	Scanner level 2 interrupt
	2	(not used)
	3	(not used)
	4	(not used)
	5	(not used)
	6	(not used)
	7	(not used)
1	0	Level 3 channel adapter interrupt
	1	(not used)
	2	(not used)
	3	(not used)
	4	(not used)
	5	(not used)
	6	(not used)
	7	(not used)

Byte 0, Bit 1 - Scanner Level 2 Interrupt: this bit, when on, indicates that one of the communication scanners has raised a level 2 interrupt.

Byte 1, Bit 0 - Level 3 Channel Adapter Interrupt: this bit, when on, indicates that one of the channel adapters has raised a level 3 interrupt.

Programming Note: these interrupts are cleared at the adapter by means of the appropriate IOH/IOHI instruction. After clearing the interrupt in this way, a minimum of 8 CCU cycles must be allowed before executing an EXIT or an Input X'77' or X'7E' instruction.

OUTPUT X'77' (MISCELLANEOUS CONTROL 2)

The register addressed by this instruction contains miscellaneous control information, used mainly to set/reset interrupt requests. The bits of this register have the following meaning:

Byte	Bit	Meaning
0	0	Reset IPL level 1 interrupt
	1	Reset CCU hardware checks
	2	Reset MOSS panel interrupt request level 3
	3	Reset MOSS diagnostic interrupt request level 3
	4	Reset MOSS service interrupt request level 4
	5	Reset MOSS service interrupt response level 4
	6	(not used)
	7	Reset program controlled interrupt level 2
1	0	Reset MOSS inoperative level 1 interrupt
	1	Reset interval timer level 3 interrupt
	2	Reset program controlled interrupt level 3
	3	Reset MOSS diagnostic interrupt request level 2
	4	Reset address compare level 1 interrupt
	5	Reset program errors
	6	Reset program controlled interrupt level 4
	7	Reset supervisor call level 4 interrupt

Byte 0, Bit 0 - Reset IPL Level 1 Interrupt: this bit, when on, resets an IPL level 1 interrupt.

Byte 0, Bit 1 - Reset CCU Hardware Checks: this bit, when on, causes all CCU hardware checks to be reset.

Byte 0, Bit 2 - Reset MOSS Panel Interrupt Request Level 3: this bit, when on, resets the MOSS panel interrupt request level 3.

Byte 0, Bit 3 - Reset MOSS Diagnostic Interrupt Request Level 3: this bit, when on, resets the MOSS diagnostic interrupt request level 3.

Byte 0, Bit 4 - Reset MOSS Service Interrupt Request Level 4: this bit, when on, resets the MOSS service interrupt request level 4.

Byte 0, Bit 5 - Reset MOSS Service Interrupt Response Level 4: this bit, when on, resets the MOSS service interrupt response level 4.

Byte 0, Bit 7 - Reset Program Controlled Interrupt Level 2: this bit, when on, resets the program controlled interrupt at level 2.

Byte 1, Bit 0 - Reset MOSS Inoperative Level 1 Interrupt: this bit, when on, resets the MOSS inoperative interrupt request at level 1.

Byte 1, Bit 1 - Reset Interval Timer Level 3 Interrupt: this bit, when on, resets the interval timer level 3 interrupt.

Byte 1, Bit 2 - Reset Program Controlled Interrupt Level 3: this bit, when on, resets the program controlled interrupt at level 3.

Byte 1, Bit 3 - Reset MOSS Diagnostic Interrupt Request Level 2: this bit, when on, resets the MOSS diagnostic interrupt request level 2.

Byte 1, Bit 4 - Reset Address Compare Level 1 Interrupt: this bit, when on, resets the address compare interrupt request level 1.

Byte 1, Bit 5 - Reset Program Errors: this bit, when on, resets all program errors.

Byte 1, Bit 6 - Reset Program Controlled Interrupt Level 4: this bit, when on, resets the program controlled interrupt at level 4.

Byte 1, Bit 7 - Reset Service Level 4 Interrupt: this bit, when on, resets the supervisor call interrupt at level 4.

OUTPUT X'78' (FORCE ALU CHECKS)

This instruction provides the means of testing the ALU compare circuit under diagnostic control. It causes one of the two redundant ALUs to be degated, thus forcing error parity on the data at the output of the ALU. Since this instruction performs a function, the bit settings of the register are not used.

INPUT X'79' (UTILITY)

This instruction causes the register specified by R to be loaded with information indicating:

- The state of the program level 5 C and Z condition latches.
- The last program level that was active before a level 1 interrupt.

The bits of this register have the following meaning:

Byte	Bit	Meaning
0	0	(not used)
	1	(not used)
	2	(not used)
	3	(not used)
	4	(not used)
	5	(not used)
	6	Program level 5 C latch
	7	Program level 5 Z latch
1	0	Program level 2 interrupted by level 1 (note)
	1	Program level 3 interrupted by level 1 (note)
	2	Program level 4 interrupted by level 1 (note)
	3	Program level 5 interrupted by level 1 (note)
	4	(not used)
	5	(not used)
	6	(not used)
	7	(not used)

Note: one only of these bits is set if the instruction is executed in program level 1. If the instruction is executed in any other level, or in level 1 if it is reentered immediately after a level 1 exit, these four bits are set to 0.

Byte 0, bit 6 - Program Level 5 C Latch: this bit indicates that the 'C' condition latch for program level 5 is on.

Byte 0, bit 7 - Program Level 5 Z Latch: this bit indicates that the 'Z' condition latch for program level 5 is on.

Byte 1, bit 0 - Program Level 2 Interrupted by Program Level 1: this bit indicates that program level 2 was interrupted by program level 1 (see note below).

Byte 1, bit 1 - Program Level 3 Interrupted by Program Level 1: this bit indicates that program level 3 was interrupted by program level 1 (see note below).

Byte 1, bit 2 - Program Level 4 Interrupted by Program Level 1: this bit indicates that program level 4 was interrupted by program level 1 (see note below).

Byte 1, bit 3 - Program Level 5 Interrupted by Program Level 1: this bit indicates that program level 5 was interrupted by program level 1 (see note below).

Note to byte 1, bits 0-3: when an Input X'79' is executed in program level 1, one of these bits is set to 1 to indicate the program level that was running when control was passed to level 1. The other bits are set to 0.

When an Input X'79' is executed in levels other than level 1, all 4 bits are set to 0.

Programming Note: if this instruction follows an Output X'79' instruction, at least one CCU cycle must separate the Output and Input instructions.

OUTPUT X'79' (UTILITY)

This instruction is used to set and reset various hardware latches. The bits of register X'79' have the following meaning:

Byte	Bit	Meaning
0	0	(not used)
	1	(not used)
	2	Set programmed IPL request
	3	(not used)
	4	Remote power off
	5	Inhibit program level 5 C and Z latches replacement
	6	Set program level 5 C latch
	7	Set program level 5 Z latch
1	0	(not used)
	1	(not used)
	2	Set AIO stop mode
	3	Reset AIO stop mode
	4	Set bypass CCU check stop mode
	5	Reset bypass CCU check stop mode
	6	Scope sync pulse 1
	7	Scope sync pulse 2

Byte 0, Bit 2 - Set Programmed IPL Request: this bit, when on, causes an interrupt to the MOSS to indicate that IPL is required (because the program is about to ABEND).

Programming Note:

This bit only causes an interrupt to the MOSS; the control program is not stopped. If the program must stop after the IPL request, an Output X'70' (Hardstop) instruction must also be executed by the control program.

Byte 0, Bit 4 - Remote Power Off: this bit, when on, raises a line to the power subsystem causing it to power down.

Byte 0, Bit 5 - Inhibit Program Level 5 C and Z Latches Replacement: this bit, when on, prevents byte 0, bits 6 and 7 from changing the state of the 5C and 5Z latches.

Byte 0, Bit 6 - Set Program Level 5 C Latch: this bit, when on, sets the program level 5 C latch to 1.

Byte 0, Bit 7 - Set Program Level 5 Z Latch: this bit, when on, sets the program level 5 Z latch to 1.

Byte 1, Bit 2 - Set AIO Stop Mode: this bit, when on, sets the AIO stop mode, causing all AIO transfers to stop.

Byte 1, Bit 3 - Reset AIO Stop Mode: this bit, when on, resets the AIO stop mode.

Byte 1, bit 4 - Set Bypass CCU Check Stop Mode: when this bit is set to 1, it prevents CCU hardware checks from setting a CCU hardstop and a MOSS interrupt.

Byte 1, bit 5 - Reset Bypass CCU Check Stop Mode: when this bit is set to 1, it resets the CCU check bypass mode to allow CCU hardstops and MOSS interrupts.

Byte 1, bit 6 - Scope Sync Pulse 1: this bit, when on, generates a scope synchronization pulse at scope sync point no. 1 when the instruction is executed.

Byte 1, bit 7 - Scope Sync Pulse 2: this bit, when on, generates a scope synchronization pulse at scope sync point no. 2 when the instruction is executed.

Programming Note: if this instruction is to be followed by an Input X'79' instruction, at least one CCU cycle must separate the Output and Input instructions.

INPUT X'7A' (HIGH RESOLUTION TIMER/UTILIZATION COUNTER)

This instruction causes the register specified by R to be loaded with the contents (three bytes: X, 0, 1) of the high resolution timer/utilization counter. The correspondence between the bits is shown in the table below: The bits of this register have the following meaning:

Byte	Bit	Meaning
X	2	Timer Bit 0
	3	Timer Bit 1
	4	Timer Bit 2
	5	Timer Bit 3
	6	Timer Bit 4
	7	Timer Bit 5
0	0	Timer Bit 6
	1	Timer Bit 7
	2	Timer Bit 8
	3	Timer Bit 9
	4	Timer Bit 10
	5	Timer Bit 11
	6	Timer Bit 12
	7	Timer Bit 13
1	0	Timer Bit 14
	1	Timer Bit 15
	2	Timer Bit 16
	3	Timer Bit 17
	4	Timer Bit 18
	5	Timer Bit 19
	6	Timer Bit 20
	7	Timer Bit 21

OUTPUT X'7A' (HIGH RESOLUTION TIMER/UTILIZATION COUNTER CONTROL)

The bits of the register addressed by this instruction have the following meaning:

Byte	Bit	Meaning
0	0	Timer/counter (1 = reset timer/enable count)
	1	High/low resolution (1 = low resolution)
	2	Timer/utilization counter (0 = timer)
	3	(not used)
	4	(not used)
	5	(not used)
	6	(not used)
	7	(not used)
1	0-7	(not used)

Byte 0, Bit 0 - Enable/Disable Timer/Counter: this bit, when on, indicates that high/low resolution timer/counter is enabled when the bit is off, the timer/counter is disabled.

Byte 0, Bit 1 - High/Low Resolution: this bit, when off, indicates that the timer/counter is set in the high resolution mode; when on, it indicates that the timer/counter is in the low resolution mode.

Byte 0, Bit 2 - Timer/Utilization Counter: this bit, when off, indicates that the high/low resolution timer is selected; when the bit is on, the utilization counter is selected.

INPUT X'7B' (BRANCH TRACE ADDRESS POINTER)

This instruction transfers the contents of the branch trace address pointer to the register specified by the R field. The bits of this register have the following meaning:

Byte	Bit	Meaning
X	4-7	Branch trace address pointer byte X, bits 4-7
0	0-7	Branch trace address pointer byte 0, bits 0-7
1	0-7	Branch trace address pointer byte 1, bits 0-7

OUTPUT X'7B' (SET PCI LEVEL 2)

This instruction sets a program controlled interrupt (PCI) at level 2. This allows a program level to transfer a processing requirement to a program level of different priority. A program controlled interrupt request is immediately effective. Since this instruction performs a function, the bit settings of the register are not used.

INPUT X'7C' (BRANCH TRACE BUFFER COUNT)

This instruction transfers the contents of the branch trace buffer count register to the register specified by the R field. The bits of this register have the following meaning:

Byte	Bit	Meaning
0	0	Branch trace buffer count bit 0
	1	Branch trace buffer count bit 1
	2	Branch trace buffer count bit 2
	3	Branch trace buffer count bit 3
	4	Branch trace buffer count bit 4
	5	Branch trace buffer count bit 5
	6	Branch trace buffer count bit 6
	7	Branch trace buffer count bit 7
1	0	Branch trace buffer count bit 8
	1	Branch trace buffer count bit 9
	2	Branch trace buffer count bit 10
	3	Branch trace buffer count bit 11
	4	Branch trace buffer count bit 12
	5	(not used)
	6	(not used)
	7	(not used)

Note: byte 1, bits 5 through 7 are ignored as the actual length of the buffer is always a multiple of 8 bytes.

OUTPUT X'7C' (SET PCI LEVEL 3)

This instruction sets a program controlled interrupt (PCI) at level 3. This allows a program level to transfer a processing requirement to a program level of different priority. A program controlled interrupt request is immediately effective. Since this instruction performs a function, the bit settings of the register are not used.

INPUT X'7D' (CCU HARDWARE CHECK REGISTER)

This instruction causes the register specified by R to be loaded with the contents of the CCU hardware check register (two bytes). For the meaning of the bits of this register, refer to the maintenance documentation.

OUTPUT X'7D' (SET PCI LEVEL 4)

This instruction sets a program controlled interrupt (PCI) at level 4. This allows a program level to transfer a processing requirement to a program level of different priority. A program controlled interrupt request is immediately effective. Since this instruction performs a function, the bit settings of the register are not used.

INPUT X'7E' (CCU LEVEL 1 INTERRUPT REQUESTS)

This instruction sets the bits in the register specified by R to indicate which type of interrupt request level 1 is set. The register also includes the CCU hardware error and adapter error summary bits. The bits of this register have the following meaning:

Byte	Bit	Meaning
0	0	MOSS inoperative
	1	CCU hardware error summary
	2	(not used)
	3	Level 5 I/O error
	4	Invalid operation
	5	Adapter level 1 interrupt request
	6	(not used)
	7	CCU level 1 interrupts during I/O summary
1	0	Address compare level 1 interrupt
	1	Addressing exception on instruction fetch
	2	Storage protect exception on instruction fetch
	3	Addressing exception on program execution
	4	Storage protect exception on program execution
	5	(not used)
	6	IPL level 1 interrupt
	7	(not used)

Byte 0, Bit 0 - MOSS Inoperative: this bit, when on, indicates that the MOSS is inoperative.

Byte 0, Bit 1 - CCU Hardware Error Summary: this bit, when on, indicates that a bit has been set in register X'7D' (CCU hardware check register). The bit by itself does not cause a level 1 interrupt. Register X'7D' may be examined via the Input X'7D' instruction.

Note: this bit may also be set if a program error (such as an Invalid Operation) occurs whilst in level I.

Byte 0, Bit 3 - Level 5 I/O Error: this bit, when on, indicates that the program attempted to execute an I/O instruction while running in level 5.

Byte 0, Bit 4 - Invalid Operation: this bit, when on, indicates that the program has attempted to execute an invalid operation code, or an I/O instruction to an invalid external register address.

Byte 0, Bit 5 - Adapter Level 1 Interrupt Request: this bit, when on, indicates that one of the channel adapters or communication scanners has raised a level 1 interrupt.

Byte 0, Bit 7 - CCU Level 1 Interrupts during I/O Summary: this bit, when on, indicates that one or more bits have been set in register X'76' (CCU level 1 interrupt on I/O operations) to indicate the cause of the level 1 interrupt. Register X'76' may be examined via the Input X'76' instruction.

Byte 1, Bit 0 - Address Compare Level 1 Interrupt: this bit, when on, indicates that an address compare has occurred.

Byte 1, Bit 1 - Addressing Exception on Instruction Fetch: this bit, when on, indicates that an addressing exception occurred during instruction fetch.

Byte 1, Bit 2 - Storage Protect Exception on Instruction Fetch: this bit, when on, indicates that a storage protection violation occurred during instruction fetch.

Byte 1, Bit 3 - Addressing Exception on Program Execution: this bit, when on, indicates that an addressing exception occurred during instruction execution.

Byte 1, Bit 4 - Storage Protect Exception on Program Execution: this bit, when on, indicates that a storage protection violation during instruction execution.

Byte 1, Bit 6 - IPL Level 1 Interrupt: this bit, when on, indicates that the MOSS has raised a level 1 interrupt request to force an IPL from the MOSS.

OUTPUT X'7E' (SET PROGRAM INTERRUPT MASK BITS)

This instruction is used to set the program level interrupt mask. When a mask bit is on, interrupt requests for the corresponding program level are ignored. When the mask bit for program level 5 is on, program execution at that level is suspended. The mask bits have the following meaning:

Byte	Bit	Meaning
0	0-7	(not used)
1	0	(not used)
	1	Mask adapter program level 1 requests
	2	Mask program level 2 requests
	3	Mask program level 3 requests
	4	Mask program level 4 requests
	5	Mask program level 5 execution
	6	(not used)
	7	(not used)

INPUT X'7F' (CCU L2, 3, OR 4 INTERRUPT REQUESTS)

This instruction sets the bits of the register specified by R to indicate which level (2, 3, or 4) and type of interrupt is set. The bits of this register have the following meaning:

Byte	Bit	Meaning
0	0	Program controlled interrupt (PCI) level 2
	1	MOSS diagnostic interrupt request level 2
	2	MOSS diagnostic interrupt request level 3
	3	MOSS service interrupt request level 4
	4	MOSS service interrupt response level 4
	5	(not used)
	6	CE/operator interrupt request level 3
	7	Program controlled interrupt (PCI) level 4
1	0	(not used)
	1	(not used)
	2	(not used)
	3	(not used)
	4	(not used)
	5	Interval timer interrupt request level 3
	6	Program controlled interrupt (PCI) level 3
	7	Supervisor Call level 4

Byte 0, Bit 0 - Program Controlled Interrupt (PCI) Level 2: this bit, when on, indicates that a program controlled interrupt has occurred at level 2.

Byte 0, Bit 1 - MOSS Diagnostic Interrupt Request Level 2: this bit, when on, indicates that the MOSS has set a level 2 interrupt to the CCU for diagnostic purposes. The interrupt request may be reset by executing on Output X'77' instruction with the 'Reset MOSS Diagnostic Interrupt Request Level 2' (byte 1, bit 3).

Byte 0, Bit 2 - MOSS Diagnostic Interrupt Request Level 3: this bit, when on, indicates that the MOSS has set a level 3 interrupt to the CCU for diagnostic purposes. The interrupt request may be reset by executing on Output X'77' instruction with the 'Reset MOSS Diagnostic Interrupt Request Level 3' (byte 0, bit 3).

Byte 0, Bit 3 - MOSS Service Interrupt Request Level 4: this bit, when on, indicates that the MOSS has requested a level 4 interrupt to request the control program to execute a function defined in the CCU Request Control Block (Mailbox) in CCU storage. The interrupt request may be reset by executing on Output X'77' instruction with the 'Reset MOSS Service Interrupt Request Level 4' (byte 0, bit 4).

Byte 0, Bit 4 - MOSS Service Interrupt Response Level 4: this bit, when on, indicates that the MOSS has requested a level 4 interrupt to inform the control program that a MOSS function requested by the CCU has been executed,

and that the response is available in the CCU Response Control Block (Mailbox) in CCU storage. The interrupt request may be reset by executing on Output X'77' instruction with the 'Reset MOSS Service Interrupt Response Level 4' (byte 0, bit 5).

Byte 0, Bit 6 - CE/Operator Interrupt Request Level 3: this bit, when on, indicates that the CE or operator has requested a CCU level 3 interrupt via the MOSS. It replaces the panel interrupt button, and is used to inform the program that it must read the data entry and function select switches and/or set the display indicators. The interrupt request may be reset by executing on Output X'77' instruction with byte 0 bit 2 (Reset MOSS Panel Interrupt Request Level 3) set to 1.

Byte 0, Bit 7 - Program Controlled Interrupt (PCI) Level 4: this bit, when on, indicates that a program controlled interrupt has occurred at level 4.

Byte 1, Bit 5 - Interval Timer Interrupt Request Level 3: this bit, when on, indicates that an interval timer interrupt has occurred at level 3.

Byte 1, Bit 6 - Program Controlled Interrupt (PCI) Level 3: this bit, when on, indicates that a program controlled interrupt has occurred at level 3.

Byte 1, Bit 7 - Supervisor Call Level 4: this bit, when on, indicates that a supervisor call request at level 5 has occurred at level 4.

OUTPUT X'7F' (RESET PROGRAM INTERRUPT MASK BITS)

This instruction is used to reset the program level interrupt mask. The mask bits have the following meaning:

Byte	Bit	Meaning
0	0-7	(not used)
1	0	(not used)
	1	Unmask adapter program level 1 requests
	2	Unmask program level 2 requests
	3	Unmask program level 3 requests
	4	Unmask program level 4 requests
	5	Unmask program level 5 execution
	6	(not used)
	7	(not used)

Programming Note:

If an interrupt for a particular level is pending when the mask bit is reset, the interrupt for that level takes place before the next instruction is executed.

CCU ERROR HANDLING

CCU HARDWARE ERRORS

These errors all cause a CCU hardstop and send a high priority interrupt request to the MOSS (the MOSS keeps running, unless it is stopped by the same failure). At the same time, a level 1 interrupt occurs in the CCU. Unless the CCU is running in the bypass check stop mode, this interrupt is queued but not executed.

The MOSS then executes a series of tests to collect error information, which is available to the MOSS operator.

Note: this information is not available to the control program, unless the error was forced by diagnostic routines running in the CCU.

The MOSS may now cause a re-IPL to take place.

CCU PROGRAM ERRORS

Program errors all cause a level 1 interrupt to the CCU, unless the CCU is already running in level 1, in which case a hardstop occurs. Error information can be obtained by the program by executing an Input X'7E' (CCU Level 1 Interrupt Requests) instruction.

Programming Note:

Storage protection/addressing exception errors may be unrecoverable if they occur during program execution (load/store instruction) as in some cases, the instruction immediately following will have been executed before the interrupt to level 1.

STORAGE PROTECTION

Setting Up the User Protect Key

To set the user protect key, an Output X'73' instruction must be executed with byte 1, bits 2 and 3 set to 00. Byte 0, bits 3 through 7 contain the 5-bit user key address, and byte 1, bits 5 through 7 contain the key to be set. Byte 1, bit 4 must be on to set the key. This is shown in the figure below:

Byte	Bit	Meaning	
Ext		(not used)	
0	0	(not used)	
	1	(not used)	
	2	(not used)	
	3	User key addr	)
	4	User key addr	)
	5	User key addr	)-User protect key address
	6	User key addr	)
	7	User key addr	)
1	0	(not used)	
	1	Enable	Bit must be on to enable SP
	2	Key Type Bit 0	0)(These two bits indicate that
	3	Key Type Bit 1	0)(the user key is to be set
	4	Set Bit	1 Bit must be on to set the key
	5	Key Bit 0	)
	6	Key Bit 1	)-Key value
	7	Key Bit 2	)

22 different user key addresses are possible. They are assigned as follows:

User Key Address	User
X'00'	Channel Adapter 1
X'01'	Channel Adapter 2
X'02'	Channel Adapter 3
X'03'	Channel Adapter 4
X'04'	Channel Adapter 5
X'05'	Channel Adapter 6
X'06'	(reserved)
X'07'	(reserved)
X'08'	(reserved)
X'09'	(reserved)
X'0A'	(reserved)
X'0B'	(reserved)
X'0C'	(reserved)
X'0D'	(reserved)
X'0E'	(reserved)
X'0F'	All Communication Scanners
X'10'	(reserved)
X'11'	Program Level 1
X'12'	Program Level 2
X'13'	Program Level 3
X'14'	Program Level 4
X'15'	Program Level 5

Setting Up the Storage Key

To set the storage key, an Output X'73' instruction must be executed with byte 1, bits 2 and 3 set to 01. Byte Ext, bits 2 through 7 and byte 0, bits 0 through 4 contain the 11-bit address of the 2048-byte block, and byte 1 bits 5 through 7 contain the key to be set. Byte 1, bit 4 must be on to set the key. This is shown in the following figure:

Byte	Bit	Meaning	
Ext	2	SKA Bit 0	)
	3	SKA Bit 1	)
	4	SKA Bit 2	)-Key address 0-5
	5	SKA Bit 3	)
	6	SKA Bit 4	)
	7	SKA Bit 5	)
0	0	SKA bit 6	)
	1	SKA Bit 7	)
	2	SKA Bit 8	)-Key address 6-10
	3	SKA Bit 9	)
	4	SKA Bit 10	)
	5	0	
	6	0	
	7	0	
1	0	(not used)	
	1	Enable	Bit must be on to enable SP
	2	Key Type Bit 0	0)(These two bits indicate that
	3	Key Type Bit 1	1)(the storage key is to be set
	4	Set Bit	1 Bit must be on to set the key
	5	Key Bit 0	)
	6	Key Bit 1	)-Key value
	7	Key Bit 2	)

Setting Up the Address Exception Key

To set the address exception key, an Output X'73' instruction must be executed with byte 1, bits 2 and 3 set to 10. Byte Ext, bits 2 through 7 and byte 0, bits 0 through 4 contain the 11-bit address, and byte 1 bit 7 must be set to 0 for each block of storage (4096 bytes) that is installed, and to 1 for each block that is not installed. Byte 1, bit 4 must be on to set the key. This is shown in the following figure:

Byte	Bit	Meaning	
Ext	2	SKA Bit 0	)
	3	SKA Bit 1	)
	4	SKA Bit 2	)-Key address 0-5
	5	SKA Bit 3	)
	6	SKA Bit 4	)
	7	SKA Bit 5	)
0	0	SKA bit 6	)
	1	SKA Bit 7	)
	2	SKA Bit 8	)-Key address 6-10
	3	SKA Bit 9	)
	4	SKA Bit 10	)
	5	0	
	6	0	
	7	0	
1	0	(not used)	
	1	Enable	Bit must be on to enable SP
	2	Key Type Bit 0	1)(These two bits indicate that
	3	Key Type Bit 1	0)(the exception key is to be set
	4	Set Bit	1 Bit must be on to set the key
	5	0	
	6	0	
	7	Read-only bit	Bit set for non-installed blocks

Setting Up the Read-Only Key

To set the read-only key, an Output X'73' instruction must be executed with byte 1, bits 2 and 3 set to 11. Byte Ext, bits 2 through 7 and byte 0, bits 0 through 4 contain the 11-bit address of the 2048-byte block, and byte 1 bit 7 must be set to 1 for each block of storage that is designated as read-only. Byte 1, bit 4 must be on to set the key. This is shown in the following figure:

Byte	Bit	Meaning	
Ext	2	SKA Bit 0	)
	3	SKA Bit 1	)
	4	SKA Bit 2	)-Key address 0-5
	5	SKA Bit 3	)
	6	SKA Bit 4	)
	7	SKA Bit 5	)
0	0	SKA bit 6	)
	1	SKA Bit 7	)
	2	SKA Bit 8	)-Key address 6-10
	3	SKA Bit 9	)
	4	SKA Bit 10	)
	5	0	
	6	0	
	7	0	
1	0	(not used)	
	1	Enable	Bit must be on to enable SP
	2	Key Type Bit 0	1)(These two bits indicate that
	3	Key Type Bit 1	1)(the read-only key is to be set
	4	Set Bit	1 Bit must be on to set the key
	5	0	
	6	0	
	7	Read-only bit	Bit set for read-only blocks

TIME MEASUREMENT IN THE 3725

The 3725 includes a general purpose timer and a utilization counter. As they use the same hardware mechanism, these two features are mutually exclusive, and cannot be used simultaneously. Neither feature causes an interrupt.

The 3725 also includes an interrupting timer, causing an interrupt every 100 milliseconds.

High/Low Resolution Timer

The high/low resolution timer is a 22-bit counter that is incremented by the system clock. It may be programmed to operate in two different modes:

- High resolution: the timer provides intervals from 0 to 0.838 seconds by increments of 200 nanoseconds.

- Low resolution: the timer provides intervals from 0 to 57.2 minutes by increments of 819 microseconds.

The timer is controlled by byte 0, bits 0 through 2 of the Output X'7A' as follows:

Bit	Meaning
0	Enable/disable timer (1 = enable)
1	High/low resolution (1 = low resolution)
2	Must be 0 to select the timer

The timer may be read via the Input X'7A' instruction.

Programming Notes

1. The timer increments continuously. To make a measurement, the program must issue an Output X'7A' instruction to reset the timer and to select the correct mode. When the event being measured occurs, the program must issue an Input X'7A' instruction to obtain the value of the counter, and multiply it by the correct factor to obtain the time in seconds.

2. The timer cannot be used in single step mode.

3. Timer overflow is never signalled. It is therefore up to the user to select the high/low resolution mode and to ensure that the measurement interval fits into the maximum count of the counting mechanism:

   a. If the interval is less than 0.838 seconds, either the high or the low resolution mode may be used.

   b. If the interval is between 0.838 seconds and 57.2 minutes, the low resolution mode must be used.

   c. If the interval is greater than 57.2 minutes, the high/low resolution counter should not be used. The 100-millisecond interrupting timer should be used instead.

Utilization Counter

The utilization counter is a 22-bit counter that is incremented by the system clock; it counts CCU busy time (including cycle steal). It operates in either high- or low-resolution mode:

- High resolution: the counter provides intervals from 0 to 0.838 seconds by increments of 200 nanoseconds.

- Low resolution: the counter provides intervals from 0 to 57.2 minutes by increments of 819 microseconds. It may be selected by executing the Output X'7A' instruction with byte 0, bit 2 set to 1; the counter may then be read via the Input X'7A' instruction.

Programming Notes

1. The counter increments continuously. To measure CCU busy time, the program must issue an Output X'7A' instruction to reset the counter and to select the correct mode. At the end of the measuring period, the program must issue an Input X'7A' instruction to obtain the value of the counter, and, if necessary, multiply it by the correct factor to obtain the time in seconds.

2. The counter cannot be used in single step mode.

3. Counter overflow is never signalled. It is therefore up to the user to select the high/low resolution mode and to ensure that the measurement interval fits into the maximum count of the counting mechanism:

   a. If the interval is less than 0.838 seconds, either the high or the low resolution mode may be used.

   b. If the interval is between 0.838 seconds and 57.2 minutes, the low resolution mode must be used.

   c. If the interval is greater than 57.2 minutes, the high/low resolution counter should not be used. The 100-millisecond interrupting timer should be used instead.

100-Millisecond Interval Timer (Interrupting)

The interval timer provides an interrupt request at program level 3 every 100 milliseconds. It may be used to maintain a real-time clock in storage, perform long and short I/O timeouts, and perform supervisory functions on a periodic basis. The interrupt may be reset by executing an Output X'77' instruction with the 'Reset Interval Timer Level 3 Interrupt' bit (byte 1, bit 1) set to 1.

The 100-millisecond timer interrupt is disabled for program stop or single instruction step mode.

CCU DIAGNOSTIC FACILITIES

The CCU has a certain number of test facilities to allow the control program to test the controller hardware.

Bypass CCU Check Stop/MOSS Interrupt

The CCU check stop/MOSS interrupt caused by a CCU hardware check may be masked by executing an Output X'79' (Utility) instruction with byte 1, bit 4 (Set Bypass CCU Check Stop Mode) set to 1. It may be un-masked by executing an Output X'79' instruction with byte 1, bit 5 (Reset Bypass CCU Check Stop Mode) set to 1.

Note: enabling/disabling the CCU check stop/MOSS interrupt may also be controlled from the MOSS.

Inhibit Channel Adapter/Communication Scanner Level 1 Interrupt

Level 1 interrupts caused by a channel adapter or communication scanner can be masked by executing an Output X'7E' (Set Program Interrupt Mask Bits) instruction with byte 1, bit 1 (Mask Adapter Program Level 1 Requests) set to 1.

The interrupts may be unmasked by executing an Output X'7F' (Reset Program Interrupt Mask Bits) instruction with byte 1, bit 1 set to 1.

Force CCU Checks

The control program can force wrong parity on the ALU output by executing an Output X'78' instruction. This data with bad parity may be used to perform further checking by moving it about the CCU.

CHAPTER 5. CHANNEL ADAPTER

The CCU is an interrupt-driven processor, and almost all processing for the channel adapter is done in response to channel adapter interrupts. This chapter is therefore organized in the following way:

- Section 1 contains basic information concerning the channel adapter.
- Section 2 describes the channel adapter interrupt system. This provides the means by which the channel adapter indicates to the CCU that it requires service.
- Section 3 describes the channel adapter I/O system. This provides the means by which the CCU responds to the channel adapter interrupts.
- Section 4 describes programming considerations for the handling of interrupts, commands, and initial status responses.
- Section 5 describes the two-processor switch.
- Section 6 describes a number of special topics.

SECTION 1. CHANNEL ADAPTER BASIC INFORMATION

The 3725 channel adapter allows the 3725 to be attached to the selector, block multiplexer, or byte multiplexer channels of a S/370 or similar host. Up to six channel adapters may be attached to the 3725 (up to 2 in the 3725 Model 2). The channel adapter is an optional feature.

MODES OF OPERATION

The channel adapter may run in either Native Subchannel mode (NSC), or in Emulation Subchannel mode (ESC). With proper programming support, the channel adapter allows the controller to operate in either NSC mode, or in ESC mode, or in both modes simultaneously.

The NSC mode is supported for all types of host channel (byte multiplex, selector, block multiplex), and allows the servicing of any number of lines up to 256 using only one unique host subchannel address. Line address decoding is handled entirely by the control program.

The ESC mode is supported for byte multiplex channels only, and allows the controller to emulate the 2701, 2702, and 2703 control units using existing host programs and subchannel addresses. A separate subchannel address is required for each line.

Notes:

1. Initial program load (IPL) must always be done in NSC mode.

2. Many of the channel adapter operations are identical in both NSC and ESC modes. Throughout this chapter, the exceptions and/or differences in operation due to NSC or ESC mode are noted by "NSC" or "ESC" at the start of the paragraph that describes the particular operation. All text that is not specially marked "NSC" or "ESC" may be assumed to apply equally to both modes of operation.

BASIC OPERATION AND DATA FLOW

The channel adapter receives an address and a command from the host processor and determines whether the host wants to communicate in NSC or ESC mode; the correct mode of operation is then set. The channel adapter then requests a level 3 interrupt to make this information available to the 3725 control program via Input instructions.

In PIO operation, the data coming from the host channel interface is placed in the data buffers, from where the control program must retreive it by executing Input instructions. In AIO mode, the data from the host is placed directly

into main storage. Channel End and Device End status are generated by the control program when the complete message or block of data has been received.

When the data is going to the host channel, the control program sends an Attention status to the channel. The host processor then initiates an initial selection sequence with a read command. The control program in PIO or AIO mode must then load the buffers with the data to be transferred. The data from the buffers can now be transferred across the channel interface. Channel End and Device End status are generated by the control program when the complete message or block of data has been sent.

DATA TRANSFER METHODS

Two methods may be used to transfer data between the channel adapter and the host channel:

- Program-initiated operation (PIO)
- Adapter-initiated operation (AIO)

Program-Initiated Operation (PIO)

With PIO, data buffering at the channel adapter interface is provided for up to four bytes of data, the buffers being serviced by programming. Program intervention is required for every four bytes. PIO is relatively slow, and should not be used on channel adapters connected to selector or block multiplexer channels. It may be used for channels attached to byte multiplexer channels in cases where performance is not critical.

Adapter-Initiated Operation (AIO)

With AIO, data buffering at the channel adapter interface is provided for 16 bytes of data, the buffers being serviced by cycle stealing. Up to 255 bytes of data may be transferred by this method before program intervention becomes necessary. AIO should always be used on selector or block multiplex channels.

CONTROLLING THE CHANNEL ADAPTER

The channel adapter is controlled by instructions issued by the control program. These instructions are of two types only: 'Adapter Input/Output' (IOH) and 'Adapter Input/Output Immediate' (IOHI). Using these instructions,

the channel adapter registers may be examined or set, buffers may be loaded or read, and cycle stealing may be initiated.

Note: throughout this chapter, the channel IOH and IOHI instructions are referred to as Input X'n' or Output X'n' for simplicity.

Access to a channel adapter may be obtained at program levels 3 and 1. Program level 3 is used for all routine servicing of the channel adapter. This level is entered via a level 3 interrupt, initiated by either an event occurring on the channel interface, or by a program controlled interrupt (PCI) from another program level. Program level 1 is used for servicing channel adapter error conditions. See under the heading 'Input X'D' (Channel Adapter Interrupt Check Register) for more details.

Programming Note

All IOH/IOHI instructions are privileged, that is, any attempt to execute them in background program level 5 causes an input/output check, and a level 1 interrupt.

CHANNEL ADAPTER STATES

The channel adapter may be in one of five states:

- Ready state
- Initial selection state
- Data transfer state
- Status transfer state
- Disabled state

Ready State

In the ready state, the channel adapter may accept instructions, but is not in one of the three active states (initial selection, data transfer, status transfer).

Initial Selection State

The channel adapter enters the initial selection state when an initial selection is started by the host processor. The channel adapter continually monitors its channel interface for one of its assigned addresses. When one of

these addresses is detected, the channel adapter enters the initial selection state, and proceeds with the initial selection. If a standard command is sent on initial selection, and is received without error (correct parity), an initial status of all zeros is returned to the channel, unless the command is I/O No-Op or Test I/O.

Note: non-standard commands sent to a block multiplex or selector channel receive an initial status of channel end.

During initial selection, the I/O device address and the channel command are stored in the initial selection address and command register. The initial selection hardware then causes a level 3 interrupt, and control is passed to the level 3 interrupt program.

Data Transfer State

The channel adapter enters the data transfer state when the control program initiates a data transfer sequence. Data is transferred across the interface from the host channel to the channel adapter, or from the channel adapter to the host, by hardware. When the data transfer is ended, the channel adapter hardware calls the control program with a level 3 interrupt request.

Status Transfer State

The channel adapter enters the status transfer state when the control program initiates a status transfer sequence. During this sequence, the status byte is transferred to the host. When the status transfer is ended, the channel adapter hardware calls the control program with a level 3 interrupt request.

Disabled State

The channel adapter is in the disabled state when it is not enabled by the control program or by the panel switches. In the disable state, the selection signals are propagated to the next adapter on the channel.

CHANNEL ADAPTER DEVICE ADDRESSES

Channel adapter device addresses are required on two separate occasions:

- At initial selection, the channel adapter must be able to recognize the device address presented to it

- On a byte multiplex channel, the channel adapter must present a valid device address to the channel before it can transfer data or status information.

Channel Adapter Device Addresses for Initial Selection

The address byte presented by the channel during initial selection must have correct parity, or the channel adapter will not decode the device address. If the parity is correct, the channel adapter will recognize a device address or addresses determined by plug options wired by the customer engineer from information supplied by the user.

NSC: the NSC device address can be assigned any value in the range 0 through 255. If the two-processor switch is installed on a channel, the two NSC interfaces (A and B) are assigned separately, and may be either the same or different. As the NSC uses only one subchannel address, the line address must be transferred from the host in the form of data. The location and the format of the terminal addresses must be coordinated between the host access method and the control program.

ESC: the ESC device addresses must form a group of contiguous addresses. The lowest address in the group may be set to 0, or to any multiple of 16 from 16 through 240. The highest address in the group may be set to one of the values 4n-1, where n = 1 through 64, that is, from 3 through 255 by steps of 4. If the two-processor switch is installed, the addresses are the same for interfaces A and B.

Programming Notes:

1. If emulation is not required, the machine can be wired at installation time so that the lowest ESC address is higher than the highest address. In this way, no ESC addresses are recognized.

2. The address assigned for the NSC may be one of the addresses in the range assigned to the ESC. The NSC address has priority, and the address is lost to the ESC.

3. After power on, the channel adapter does not immediately recognize the ESC addresses, even if the interface is enabled. To make the ESC addresses operational, the program must issue an Output X'7' instruction to set the 'Set ESC Operational' bit (byte 1, bit 5) to 1. If the host sends Start I/O to an ESC address before this bit is set, the resulting condition code for the host instruction is set to 3 (not operational).

4. An initial selection causes a channel adapter initial selection level 3 interrupt. The program may determine the I/O device address by issuing an Input X'1' instruction. Once the program has set the 'Set ESC Operational' bit as described above, all the assigned ESC addresses become operational, and the interrupt request may be caused by an initial selection sequence for any of these addresses, or for the NSC address. The

program must therefore be prepared to handle initial selection sequences for all assigned operational addresses, both NSC and ESC.

Channel Adapter Device Addresses for Data/Status Transfer

When the control unit initiates a data/status transfer, it must provide the correct device address associated with the transfer.

NSC: for control unit initiated data/status transfers on the NSC, the plugged hardware address is used.

ESC: as the device address for control unit initiated data/status transfers on the ESC is variable, it must be provided by the control program. This is done by executing an Output X'3' instruction, with the data/status transfer address in byte 0. The channel adapter hardware checks only that the issued address is in the range of plugged addresses; if the address is outside this range, the address compare error bit (byte 0, bit 5) is set in register X'D', and a level 1 interrupt occurs.

Programming Notes:

1. If the address presented is incorrect, but within the plugged range, the channel adapter has no means of detecting the error; no error is signaled, but improper channel operation will occur.

2. The I/O address that was presented to the channel by the last Output X'3' instruction may be determined by executing an Input X'3' instruction. The Input X'3' instruction should only be issued in interrupt level 3.

SECTION 2. CHANNEL ADAPTER INTERRUPT REQUESTS

The channel adapter can raise interrupt requests at level 1 and at level 3.

- Level 1 interrupt requests are caused by check or error conditions.
- Level 3 interrupt requests are caused by two different conditions:
  - Initial selection interrupt requests are raised when the channel adapter receives an address and a command across the channel interface.
  - Data/status interrupt requests are raised when the channel adapter requires data or status service.

LEVEL 1 INTERRUPT REQUESTS

When an error condition is detected in the channel adapter, a level 1 interrupt occurs, and a bit is set in the Channel Adapter Level 1 Interrupt Check Bit register to indicate the type of error.

LEVEL 3 INTERRUPT REQUESTS

There are two types of interrupt request at level 3:

- Channel Adapter Initial Selection Level 3 interrupt request
- Data/Status Transfer Level 3 interrupt request

Channel Adapter Initial Selection Level 3 Interrupt Request

This type of interrupt request may be due to:

- An initial selection sequence
- A system reset sequence
- An NSC status cleared indication
- An ESC TIO status cleared indication

When an Initial Selection interrupt request occurs, the condition causing the interrupt may be determined by executing an Input X'0' (Initial Selection Control register) instruction.

Channel Adapter Data/Status Level 3 Interrupt Request

This type of interrupt request may be set by:

- The end of an inbound data transfer sequence
- The end of an outbound data transfer sequence
- The end of a status transfer sequence
- Any level 1 interrupt occurring during any one of the above three data/status transfers.
- A Suppress Out Monitor condition
- A program requested interrupt

When a Data/Status interrupt request occurs, the condition causing the interrupt may be determined by executing an Input X'2' (Data/Status Control register) instruction.

SECTION 3. CHANNEL ADAPTER INPUT/OUTPUT

The channel adapter contains a number of registers, most of which are accessible to the program via the IOH/IOHI instructions. These registers are described in detail below under the heading 'Channel Adapter IOH/IOHI Instructions - Detailed Bit Structure'. One very important register used for channel adapter operations in AIO mode is physically located in the CCU, and is accessed by CCU Input and Output instructions (not IOH/IOHI):

Input/Output X'30' through X'35' - Fixed Pointer Registers

The Output instruction loads the CCU pointer address with the cycle steal data address for the channel adapter; the Input instruction may be used to read it back. The correspondence between register and channel adapter is as follows:

Register	Channel adapter
X'30'	1
X'31'	2
X'32'	3
X'33'	4
X'34'	5
X'35'	6

CHANNEL ADAPTER IOH/IOHI INSTRUCTIONS

The channel adapter IOH/IOHI instructions are used to transfer the contents of one of the general registers to one of the channel adapter registers (register X'n') or vice versa.

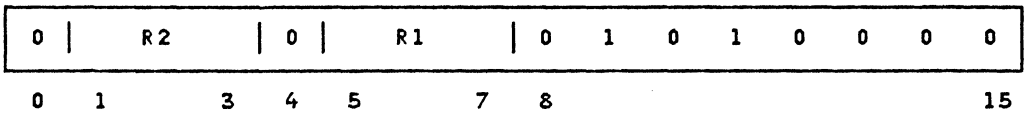
There are two types of channel adapter input/output instruction:

- Adapter Input/Output (IOH)
- Adapter Input/Output Immediate (IOHI)

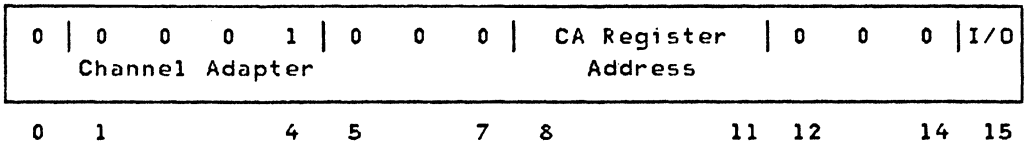
They are used by the channel adapter as follows:

ADAPTER INPUT/OUTPUT (IOH)

This instruction transfers the contents of the register specified by R1 to the channel adapter, or places information coming from the channel adapter into the register specified by R1. The adapter, the adapter command or register, and the direction of data movement are all specified by the contents of R2.



R2 must be loaded as follows:



Bits 1 through 4 (= 0001) indicate the channel adapters

Bits 5 through 7 must be zero.

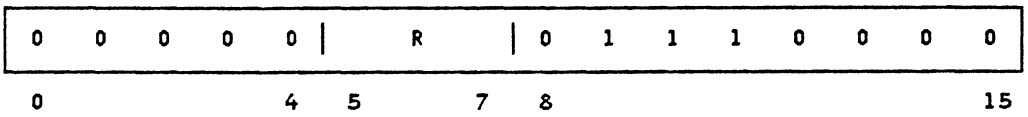
Bits 8 through 11 indicate the CA register address (X'x') to be used.

I/O = input/output bit: 0 = output, 1 = input

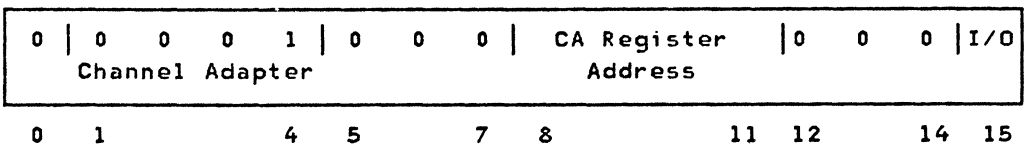
ADAPTER INPUT/OUTPUT IMMEDIATE (IOHI)

This instruction transfers the contents of the register specified by R to the channel adapter, or places information coming from the channel adapter into the register specified by R. The adapter, the adapter register, and the direction of data movement are all specified by the contents of the second halfword.

First halfword



Second halfword



Bits 1 through 4 (= 0001) indicate the channel adapters

Bits 5 through 7 must be zero.

Bits 8 through 11 indicate the CA register address (X'x') to be used.

I/O = input/output bit: 0 = output, 1 = input

CHANNEL ADAPTER ADDRESSING

As may be seen in the previous section, the IOH and IOHI instructions do not contain an explicitly defined address. The channel adapter is addressed indirectly via the contents of a special 3-bit register, controlled by byte 0 of the Output X'7' instruction. The table below shows the bits of the instruction used to control this 3-bit register:

Byte	Bit	Meaning
0	0	Enable auto-selection (all channel adapters)
	1	Disable auto-selection (all channel adapters)
	2	Select CA addressed by bits 4-6
	3	Execute output on CA addressed by bits 4-6
	4	Channel address bit 0)
	5	Channel address bit 1) CA address 1-6
	6	Channel address bit 2)
	7	Channel adapter reset

Bits 4 through 6 define the channel adapter as follows:

Bit 4 5 6	Channel Adapter
0 0 0	1
0 0 1	2
0 1 0	3
0 1 1	4
1 0 0	5
1 0 1	6

There are two modes of operation:

- Explicit selection by the control program.
- Selection by the auto-selection mechanism.

CHANNEL ADAPTER SELECTION BY THE CONTROL PROGRAM

This mode of channel adapter selection is controlled by byte 0, bits 2 through 6 of the Output X'7' instruction. Bits 4 through 6 contain the address of the channel adapter to be selected, and bits 2 and 3 control the operation:

- If bit 2 (Select Channel Adapter Addressed by Bits 4 through 6) is set to 1, the addressed channel adapter is selected for all subsequent channel adapter instructions until it is turned off.

  Note: the channel adapter (if any) that was selected by the auto-selection mechanism is changed.

- If bit 3 (Execute Output on Channel Adapter Addressed by Bits 4 through 6) is set to 1, the addressed channel adapter is temporarily selected for this instruction only. This mode is used to set up a program requested interrupt on the addressed channel via byte 1, bit 1 (Set Program Requested Interrupt) of this same Output X'2' instruction.

  Note: the channel adapter (if any) that was selected by the auto-selection mechanism, or by byte 0, bit 3 is simply overridden, not changed.

CHANNEL ADAPTER SELECTION BY THE AUTO-SELECTION MECHANISM

This mode of channel adapter selection is controlled by byte 0, bits 0 and 1 of the Output X'7' instruction.

Bit 0 enables auto-selection and bit 1 disables it; both bits must not be on at the same. Auto-selection works as follows:

1. The control program enables auto-selection on all channel adapters by performing an Output X'7' instruction with byte 0, bit 0 set to 1.

2. When a level 3 interrupt is pending, the control program executes an Input X'F' (Channel Adapter Level 3 Interrupt Requests) instruction. If the 'Auto-Selection Complete' latch is not already set, the channel adapter with the highest priority interrupt request pending is selected; its address is available in byte 0, bits 4 through 6 of register X'F'. At the same time, the 'Auto-Selection Complete' latch is set to prevent further auto-selection until the current interrupt has been serviced and reset.

3. The control program executes the interrupt handling routine. All instructions to the channel adapter are now routed automatically to the selected channel adapter.

4. At the end of the interrupt handling routine, the control program must reset the 'Auto-Selection Complete' latch by executing one of the following instructions:

   - Output X'0' (bit configuration is ignored).

   - Output X'2' with either byte 0, bit 5 (Reset Initial Selection Interrupt), or byte 0, bit 6 (Reset Data/Status Interrupt) set to 1 as appropriate.

- Output X'7' with byte 1, bit 3 (Reset System Reset/NSC Address Active), if the interrupt was due to a System Reset.

- Output X'B' (bit configuration is ignored).

The order of priorities in the auto-selection mechanism is as follows:

1. Priority outbound level 3 interrupt.

2. Outbound level 3 interrupt.

3. Initial selection level 3 interrupt.

4. Inbound level 3 interrupt.

5. All other level 3 interrupts.

CHANNEL ADAPTER IOH/IOHI INSTRUCTIONS - DETAILED BIT STRUCTURE

INPUT X'0' (INITIAL SELECTION CONTROL REGISTER)

The register addressed by this instruction is set by the channel adapter hardware and contains information that identifies the event that set the channel adapter Initial Selection Level 3 interrupt. The instruction should be issued only when servicing a channel adapter initial selection level 3 interrupt request. This type of interrupt request may be set by:

1. the completion of an initial selection sequence.

2. the detection of various reset sequences.

During a normal initial selection sequence, the initial selection interrupt bit (byte 0, bit 0) is set. The remaining bits give supplementary information, or indicate certain reset and error conditions. The bits of this register have the following meaning:

Byte	Bit	Meaning
0	0	Initial selection interrupt
	1	Interface disconnect
	2	Selective reset
	3	Channel bus out check
	4	Emulation subchannel operation (ESC = 1)
	5	Initial status byte stacked
	6	Status byte cleared
	7	System reset
1	0-7	(not used)

Byte 0, Bit 0 - Initial Selection Interrupt: this bit, when on, indicates that a normal initial selection sequence has occurred. The initial status byte sent to the host processor is X'00', unless it is a command for the NSC and the channel command is non-standard; the initial status byte is then X'08', with the Channel End bit on. The I/O device address and the I/O command byte may be determined by executing an input X'1' instruction. If this bit is on, indicating a normal initial selection sequence, all other bits of this register should be off, with the possible exception of the ESC selection bit (byte 0, bit 4).

If byte 0, bit 0 is off, it indicates that the interrupt request was due to the detection of another condition as defined by the remaining bits of the register.

Programming Note:

Certain normal initial selection sequences do not cause the channel adapter initial selection level 3 interrupt to be set, and furthermore, do not set byte 0, bit 0. These sequences are as follows:

1. An initial selection sequence in which the channel adapter responds automatically with a 'Control Unit Busy' status indication (X'70' = 'Status Modifier', 'Control Unit End', and 'Busy') due to the channel adapter initial selection level 3 interrupt request having been already previously set.

2. An initial selection sequence for a channel I/O command byte X'03' ('I/O No-Op'). The initial status byte of 'Channel End' and 'Device End' is generated automatically by the channel adapter hardware.

3. A 'Test I/O' command has been sent to the NSC address when it was free of commands.

Byte 0, Bit 1 - Interface Disconnect: this bit, if on, indicates that the channel adapter level 3 interrupt request was caused by the detection of an interface disconnect sequence (Halt I/O) during initial selection. The addressed subchannel can be determined via an Input X'1' instruction.

Byte 0, Bit 2 - Selective Reset: this bit, if on, indicates that the channel adapter level 3 interrupt request was caused by the detection of a selective reset sequence during initial selection. The addressed subchannel can be determined via an Input X'1' instruction.

Note:

The selective reset does not cause a reset of the channel adapter. If the unique NSC address (byte 0, bit 4 off), has received the selective reset, the program should execute an Output X'7' instruction with the 'reset system reset/NSC address active' bit (byte 1, bit 3) equal to 1. This resets the channel adapter hardware associated with the unique NSC address.

Byte 0, Bit 3 - Channel Bus Out Check: this bit, if on, indicates that the channel adapter level 3 interrupt request was caused by the detection of bad (even) parity on the I/O channel interface bus out when the channel I/O command byte was presented during initial selection. The channel adapter responds automatically with 'Unit Check' status. The addressed subchannel can be determined via an Input X'1' instruction.

Byte 0, Bit 4 - Emulation Subchannel Operation: this bit, if on, indicates that the channel adapter initial selection address and command register (X'1') contains an ESC address.

If the bit is off, the channel adapter initial selection address and command register (X'1') contains the unique NSC address.

Note: this bit is set to zero if the initial selection level 3 interrupt was due to a system reset.

Byte 0, Bit 5 - Initial Status Byte Stacked: this bit, if on, indicates that the channel adapter level 3 interrupt request was caused by the completion of an initial selection sequence in which the initial status byte presented to the channel has been stacked. The addressed subchannel and I/O command can be determined via an Input X'1' instruction.

The initial status byte stacked indication can occur in the following situations:

1. NSC (Byte 0, bit 4 = 0). The NSC status byte, prepared by the program, was presented to the host during an initial selection sequence, but was stacked (not accepted) by the host. As a result, a channel adapter initial selection level 3 interrupt request is set.

2. ESC (Byte 0, bit 4 = 1). The ESC status byte, prepared by the control program for a given subchannel, was presented to the host in response to an ESC 'Test I/O' command issued to the same subchannel. However, the status was stacked (not accepted) by the host.

Byte 0, Bit 6 - Status Byte Cleared: the interpretation of this bit depends on the state of Byte 0, bit 4:

1. NSC (Byte 0, bit 4 = 0). The NSC status byte, either prepared by the program in the X'6' register, or an early Channel End status that has been previously stacked, has been transferred to the host during an initial selection sequence. Thus, the NSC status byte has been cleared, and this has resulted in the setting of the channel adapter initial selection level 3 interrupt request. If the command during this initial selection sequence was 'Test I/O' (X'00'), the status is presented normally. If, however, the command was anything other than 'Test I/O', the 'Busy' bit is presented in addition to the other status bits. The control program must perform the channel adapter state and the status sent by performing the appropriate IOH. If the 'NSC address active' bit (Input X'7', byte 0, bit 5) is found to be on it means that the Device End status must be sent; if not, the program only resets the level 3 interrupt request.

   Note: byte 0, bit 0 is not set.

2. ESC (Byte 0, bit 4 = 1). The channel adapter initial selection level 3 interrupt request results from the completion of an initial selection sequence in which a 'Test I/O' command to an ESC subchannel was serviced, and TIO status had previously been set up for that subchannel. The subchannel that was serviced, and the status that was presented, can be determined by executing an Input X'B' instruction.

   Note: byte 0, bit 0 is not set.

Byte 0, Bit 7 - System Reset: This bit, when on, indicates that a system reset sequence has occurred on the channel, causing the channel adapter to reset also. This means that if this bit is on, all other bits obtained by the execution of the Input X'0' instruction are automatically 0.

Note: since a system reset may occur at any time, all indications of previous channel sequences that have not yet been serviced are lost to the CCU program.

OUTPUT X'0' (RESET INITIAL SELECTION)

This instruction resets all the initial selection hardware latches and also the channel adapter level 3 interrupt request resulting from an initial selection sequence. As this instruction performs a function, the bit settings of the register are not used.

Note: this instruction does not reset a 'system reset' condition, nor the resulting channel adapter level 3 interrupt request.

INPUT X'1' (INITIAL SELECTION ADDRESS AND COMMAND REGISTER)

The register addressed by this instruction is set by the channel adapter hardware with the address and the command received from the channel. The instruction should be issued only if a channel adapter initial selection interrupt has been set at level 3, and an Input X'0' has shown that the interrupt is due to an initial selection sequence. The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0-7	Address byte bits 0-7 (initial selection addr)
1	0-7	I/O cmd byte bits 0-7 (initial selection cmd)

OUTPUT X'1' (INITIAL SELECTION ADDRESS AND COMMAND REGISTER)

This instruction allows the program to set register X'1' with an initial selection address and command for diagnostic purposes only.

INPUT X'2' (DATA/STATUS CONTROL REGISTER)

The register addressed by this instruction is used to identify the event(s) which caused a channel adapter Data/Status level 3 interrupt. The instruction must normally be issued only for servicing a channel adapter level 3 data/status interrupt.

Note: if a system reset sequence occurs before the Input X'2' is executed, the bits which define the cause of the interrupt request, and the interrupt request itself are reset.

The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	Outbound data transfer sequence
	1	Inbound data transfer sequence
	2	Status transfer sequence
	3	Emulation subchannel operation (ESC = 1)
	4	Channel end presented
	5	Channel stop/interface disconnect
	6	Suppress out monitor interrupt
	7	Program requested interrupt
1	0	Channel bus out check
	1	Selective reset
	2	Suppress out
	3	Ending status stacked
	4	Priority outbound service
	5	Residual byte count bit 5
	6	Residual byte count bit 6
	7	Residual byte count bit 7

Byte 0, Bit 0 - Outbound Data Transfer Sequence: this bit, when on, indicates that the channel adapter data/status level 3 interrupt was caused by the ending of an outbound (3725 to host) data transfer sequence. If this bit is on, byte 0, bit 1 should be off. On a block multiplex channel, byte 0, bit 2 may also be on if the block is the last data block.

In PIO mode, byte 1, bits 5 through 7 indicate the residual byte count.

ESC: if the transfer was over an emulator subchannel (byte 0, bit 3 = 1), an Input X'3' instruction should be executed to determine the address of the subchannel.

Byte 0, Bit 1 - Inbound Data Transfer Sequence: this bit, when on, indicates that the channel adapter data/status level 3 interrupt was caused by the ending of an inbound (host to 3725) data transfer sequence. If this bit is on, byte 0, bits 0 and 2 should both be off.

In PIO mode, byte 1, bits 5 through 7 indicate the residual byte count; the data bytes transferred from the host processor may be obtained by executing the Input X'4' and X'5' instructions.

ESC: if the transfer was over an emulator subchannel (byte 0, bit 3 = 1), an Input X'3' instruction should be executed to determine the address of the subchannel.

Byte 0, Bit 2 - Status Transfer Sequence: this bit, when on, indicates that the channel adapter data/status level 3 interrupt was caused by the ending of a status transfer sequence. If this bit is on, byte 0, bit 1 should be off. Byte 0, bit 0 may be on or off.

ESC: if the transfer was over an emulator subchannel (byte 0, bit 3 = 1), an Input X'3' instruction should be executed to determine the address of the subchannel over which the transfer occurred and to obtain the status that was presented.

Byte 0, Bit 3 - Emulation Subchannel Operation:

NSC (Byte 0, Bit 3 Off)

The transfer sequence defined by byte 0, bits 0 through 2, was performed on the native subchannel using the assigned hardware address. In the case of a status transfer, the status was taken from the NSC status register (X'6').

ESC (Byte 0, Bit 3 On)

The transfer sequence defined by byte 0, bits 0 through 2, was performed on the ESC subchannel using the ESC address (and the status, in the case of a status transfer) taken from the ESC address and status register (X'3').

Byte 0, Bit 4 - Channel End Presented: this bit only applies to a native subchannel working with a block multiplex or selector channel; for all other combinations, it is not used.

Note: when this bit is on, byte 0, bit 1, or byte 0, bits 0 and 2 are also on.

When the bit is on, it indicates that the hardware has presented, or has tried to present, a 'Channel End' status to the host during a data transfer; a level 3 interrupt request is set up at the same time.

The hardware 'Channel End' is presented when the CCU program sets up an outbound (3725 to host) data transfer sequence and a status transfer sequence at the same time (Output X'2', byte 0, bits 0 and 2 both on), and the data transfer to the host occurs normally.

The hardware 'Channel End' is also presented during any data transfer sequence if the host issues a 'Channel Stop' command.

Byte 0, Bit 5 - Channel Stop/Interface Disconnect: this bit, when on, indicates that during an inbound or outbound data transfer sequence, a channel stop or interface disconnect (Halt I/O) sequence has occurred (the CCU program

cannot distinguish between the two). The transfer sequence is ended; the residual byte count may be greater than zero, and indicates the number of bytes that were not transferred (the residual byte count is contained in register X'2', byte 1, bits 5 through 7 for PIO operations, and in register X'C', byte 1, bits 0 through 7 for AIO operations).

The bit, when on, may also indicate that an interface disconnect (Halt I/O) occurred during a status transfer sequence.

Byte 0, Bit 6 - Suppress Out Monitor Interrupt: this bit, when on, indicates that the channel adapter data/status level 3 interrupt was set because the 'suppress out' tag line was found to be inactive after the control program had told the channel adapter to monitor for this condition. If one of the transfer bits (byte 0, bits 0 through 2) is on, this bit should be 0 for correct operation. The program signals to the channel adapter to monitor for the inactive condition of 'suppress out' by executing an Output X'7' instruction with the 'Set Suppress Out Monitor Interrupt' bit (Byte 1, bit 0) set to 1.

Byte 0, Bit 7 - Program Requested Interrupt: this bit, when on, indicates that the channel adapter data/status level 3 interrupt was set because the program requested an interrupt at level 3 by executing an Output X'7' instruction with byte 1, bit 1 set 1. An Output X'2' instruction should be executed to reset this bit and the resulting channel adapter data/status level 3 interrupt request.

Byte 1, Bit 0 - Channel Bus Out Check: this bit, when on, indicates that during an inbound (host to 3725) data transfer sequence, a bad (even) parity condition was detected on bus out during the transfer of a data byte and that the transfer sequence was terminated. The number of bytes transferred prior to the bus out check may be found by examining the residual byte count (the residual byte count is contained in register X'2', byte 1, bits 5 through 7 for PIO operations, and in register X'C', byte 1, bits 0 through 7 for AIO operations).

Byte 1, Bit 1 - Selective Reset: this bit, when on, indicates that a selective reset sequence occurred during the transfer.

Byte 1, Bit 2 - Suppress Out: this bit, when on, indicates that the 'Suppress Out' tag line on the channel was active at the time that the Input X'2' instruction was executed.

Byte 1, Bit 3 - Ending Status Stacked: this bit, when on, indicates that during a final status transfer sequence, the status byte was not accepted by the channel (stacked).

ESC

The status byte that was presented may be examined by executing an Input X'3' instruction.

Byte 1, Bit 4 - Priority Outbound Service: this bit, when on, indicates to the auto-selection logic that an ESC address that has completed an outbound (3725 to host) data transfer has the highest priority.

Byte 1, Bits 5 through 7 - Residual Byte Count: these bits only apply to PIO operations. They contain the residual byte count (number of bytes that were not transferred) for inbound or outbound data transfer sequences.

Note: For AIO operations, the residual byte count is obtained by executing an Input X'C' instruction.

OUTPUT X'2' (DATA/STATUS CONTROL REGISTER)

The register addressed by this instruction is used to control the operation of the channel adapter. This instruction also resets the Program Requested Interrupt and Suppress Out Monitor bits. It should be issued only if a level 3 channel adapter initial selection interrupt has been set. The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	Set/reset outbound data transfer sequence (Note)
	1	Set/reset inbound data transfer sequence (Note)
	2	Set/reset status transfer sequence (Note)
	3	Set/reset ESC operation (Note)
	4	Set/reset PIO mode (Note)
	5	Reset initial selection interrupt
	6	Reset data/status interrupt
	7	(not used)
1	0	Set monitor for circle B
	1	(not used)
	2	Set monitor for 2848 ETX
	3	Set suppressible status
	4	Priority outbound service
	5	Request byte count bit 5
	6	Request byte count bit 6
	7	Request byte count bit 7

Note: Set = 1; reset = 0.

Byte 0, Bit 0 - Set/Reset Outbound Data Transfer Sequence: this bit, when on, sets the Outbound Data Transfer bit and causes the channel adapter to initiate an outbound (3725 to host) data transfer sequence.

If the bit is off, it resets the Channel Adapter Outbound Data Transfer Sequence bit (if it is on).

NSC: if the transfer is over the native subchannel (byte 0, bit 3 = 0), the transfer is initiated using the assigned NSC address.

ESC: if the transfer is over an emulator subchannel (byte 0, bit 3 = 1), the transfer is initiated using the address contained in register X'3'.

Byte 0, Bit 1 - Set/Reset Inbound Data Transfer Sequence: this bit, when on, sets the Inbound Data Transfer bit and causes the channel adapter to initiate an inbound (host to 3725) data transfer sequence.

If the bit is off, it resets the Channel Adapter Inbound Data Transfer Sequence bit (if it is on).

NSC: if the transfer is over the native subchannel (byte 0, bit 3 = 0), the transfer is initiated using the assigned NSC address.

ESC: if the transfer is over an emulator subchannel (byte 0, bit 3 = 1), the transfer is initiated using the address contained in register X'3'.

Byte 0, Bit 2 - Set/Reset Status Transfer Sequence: this bit, when on, sets the Status Transfer bit and causes the channel adapter to initiate a status transfer sequence.

If the bit is off, it resets the Channel Adapter Status Transfer Sequence bit (if it is on).

NSC: if the transfer is over the native subchannel (byte 0, bit 3 = 0), the transfer is initiated using the assigned NSC address and the NSC status byte (from register X'6').

The NSC status remains available until it is accepted by the host channel.

Note (NSC only): Byte 0, bits 0 and 2 may be on together; the channel adapter hardware then presents a Channel End status to the host.

ESC: if the transfer is over an emulator subchannel (byte 0, bit 3 = 1), the transfer is initiated using the ESC address and status contained in register X'3'.

When the status is presented to the channel, the 'ESC TIO Status Available' latch is reset. If the status is stacked, the ESC address and the status that was stacked are moved by hardware to the ESC TIO Address and Status byte register (X'B'); at the same time, the ESC TIO Status Available latch is set.

Byte 0, Bit 3 - Set/Reset ESC Operation:

NSC (Byte 0, Bit 3 Off)

The transfer sequence defined by byte 0, bits 0 through 2, is initiated on the native subchannel using the assigned hardware address. In the case of a status transfer, the status is taken from the NSC status register (X'6').

ESC (Byte 0, Bit 3 On)

The transfer sequence defined by byte 0, bits 0 through 2, is initiated on the emulation subchannel using the ESC address (and the status, in the case of a status transfer) taken from the ESC address and status register (X'3').

Byte 0, Bit 4 - Set/Reset PIO Mode:

AIO Mode (Byte 0, Bit 4 Off)

The data transfer sequence defined by byte 0, bit 0 or 1, is executed in AIO mode using the cycle steal mechanism. Byte 1, bits 5 through 7 are not used. The request byte count and the special control functions are taken from register X'C'.

PIO Mode (Byte 0, Bit 4 On)

The data transfer sequence defined by byte 0, bit 0 or 1, is executed in PIO mode with program intervention required every 4 bytes. The request byte count is taken from byte 1, bits 5 through 7, while byte 1, bits 0 and 2 enable certain special control functions. The transfer byte count and special control bits contained in register X'C' are not used for PIO operations.

Byte 0, Bit 5 - Reset Initial Selection Interrupt: This bit, when on, causes the channel adapter to reset all the initial selection hardware latches and also the channel adapter level 3 interrupt request resulting from an initial selection sequence. This bit does not reset a 'system reset' condition, nor the resulting channel adapter level 3 interrupt request.

Byte 0, Bit 6 - Reset Data/Status Interrupt: This bit, when on, causes the channel adapter to reset the following bits in the Data/Status Control register (Input X'2'):

- Channel stop/interface disconnect (byte 0, bit 5)
- Channel bus out check (byte 1, bit 0)
- Selective reset (byte 1, bit 1)
- Stacked ending status (byte 1, bit 3)

The channel adapter data/status level 3 interrupt is also reset. In addition, if one of the transfer bits (byte 0, bits 0 through 2) is on, the channel adapter hardware also raises the 'Request In' channel interface tag line in order to initiate the transfer sequence.

Byte 1, Bit 0 - Set Monitor for Circle B: this bit turns on the monitoring for the 'Circle B' character on inbound data transfer only. If the bit is off, the monitoring is reset.

Byte 1, Bit 2 - Set Monitor for 2848 ETX: this bit turns on the monitoring for the 2848 'ETX' character on inbound data transfer only. If the bit is off, the monitoring is reset.

Byte 1, Bit 3 - Set Suppressible Status: this bit is set by the control program after a status has been stacked on the channel to inhibit a status transfer to the host as long as 'Suppress Out' is active. It is also set:

- When the control program presents an ending status to a selective reset.
- When an asynchronous 'Attention' status occurs.

Byte 1, Bit 4 - Priority Outbound Service: this bit applies to outbound (3725 to host) operations on the ESC only. When on, the bit indicates an ESC priority outbound data transfer sequence to the auto-selection logic. Priority outbound service and NSC interrupts have the highest auto-select priority.

Byte 1, Bits 5 through 7 - Request Byte Count: These bits apply to PIO operations only; for AIO operations, they are ignored. They are used to to indicate the number of bytes to be transferred to or from the host. A maximum of 4 bytes can be transferred at one time.

INPUT/OUTPUT X'3' (ESC ADDRESS AND STATUS BYTE REGISTER)

The register addressed by this instruction contains the following information:

- When transferring data to or from the channel, byte 0 contains the address to be used by the ESC. Byte 1 is not used.
- When transferring status information to the channel, byte 0 contains the address to be used by the ESC; byte 1 contains the status.

These two instructions should be issued only if a level 3 channel adapter data/status interrupt has been set, prior to informing the channel adapter that an ESC data/status transfer is required.

Notes:

1. The ESC I/O device address provided by the last Output X'3' instruction executed is presented to the channel in all subsequent ESC transfer sequences. This instruction must therefore be executed each time a transfer sequence is required for a different I/O address.
2. For a status transfer sequence, the ESC final status must be provided in byte 1.
3. For reasons of compatibility, the program should ensure that the ESC status bits that are provided are consistent with those that are set under similar circumstances by the 2701, 2702, or 2703.
4. The Input X'3' instruction may be used to determine the ESC address and status that were provided to the channel when Output X'3' was last executed. It may therefore be used for checking purposes, or to obtain the information if it is not kept elsewhere.

The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0-7	ESC address byte bits 0-7 (data/status transfer)
1	0	ESC status byte bit 0 (attention)
	1	ESC status byte bit 1 (status modifier)
	2	ESC status byte bit 2 (control unit end)
	3	ESC status byte bit 3 (Busy)
	4	ESC status byte bit 4 (channel end)
	5	ESC status byte bit 5 (device end)
	6	ESC status byte bit 6 (unit check)
	7	ESC status byte bit 7 (unit exception)

INPUT/OUTPUT X'4' AND X'5' (DATA BUFFER REGISTERS)

The registers addressed by these instructions are used to hold data during data transfers in either direction between the channel adapter and the host channel. The way in which these registers are used depends on whether the buffers are being loaded in program initiated operation mode, or in adapter initiated operation mode. The bits of the registers have the following meaning:

Register X'4' (Data Buffer Bytes 1 and 2 or 5 and 6)

Byte	Bit	Meaning
0	0-7	Data buffer byte 1 or 5 bits 0-7
1	0-7	Data buffer byte 2 or 6 bits 0-7

Register X'5' (Data Buffer Bytes 3 and 4 or 7 and 8)

Byte	Bit	Meaning
0	0-7	Data buffer byte 3 or 7 bits 0-7
1	0-7	Data buffer byte 4 or 8 bits 0-7

Note: bytes 5 through 8 are used for diagnostic purposes only.

Program-Initiated Operation (PIO)

To transfer data in PIO mode, byte 0, bit 4 must be set to 1. Four single-byte buffers are used for data transfer, two bytes (1 and 2) being contained in register X'4', and two bytes (3 and 4) in register X'5'.

During an inbound operation (host to 3725: byte 0, bit 1 of register X'2' is on) the four buffers are loaded by the channel adapter hardware. When the four bytes have been loaded, a data/status level 3 interrupt occurs. The level 3 control program must then empty the buffers by software using the Input X'4' and X'5' instructions. Program intervention is thus required, in general, every 4 bytes.

Programming Notes:

1. After power on, the buffers should be loaded via the Output X'4' and X'5' commands to prevent parity errors when reading.

2. Before executing the Input X'4' and X'5' instructions, the program should first examine the residual byte count in register X'2':

   - Buffer byte 1 contains valid data if the residual count is less than the requested transfer count (issue Input X'4' instruction).

   - Buffer byte 2 contains valid data if the residual count is at least 2 less than the requested transfer count (issue Input X'4' instruction).

   - Buffer byte 3 contains valid data if the residual count is at least 3 less than the requested transfer count (issue Input X'4' and X'5' instructions).

   - Buffer byte 4 contains valid data only if the residual count is 0 and a 4-byte transfer was requested (issue Input X'4' and X'5' instructions).

During an outbound (3725 to host) operation, the four buffers are loaded by the program using the Output X'4' and X'5' instructions. The program must then start the hardware data transfer by setting on byte 0, bit 0 in the Data/Status Control Register (X'2'). When the four data bytes have been transferred across the channel, the hardware causes a data/status level 3 interrupt to ask for four more bytes to be loaded into the buffers.

Programming Note:

To ensure data integrity, the request byte count contained in byte 1, bits 5 through 7 of register X'2' must be consistent with the number of data bytes loaded into the data buffers.

Adapter-Initiated Operation (AIO)

In AIO mode, two 8-byte buffers are used alternately for data transfer. They are called the A and B Data Buffers. The operation of loading and unloading the buffers is done entirely by hardware; program intervention is never required during normal operation. The two buffers are switched between the CCU and the channel adapter:

- During a host write operation, one Data Buffer is loaded by the channel adapter hardware while the other is emptied by the CCU cycle steal mechanism.

- During a host read operation, one Data Buffer is loaded by the CCU cycle steal mechanism while the other is emptied by the channel adapter hardware.

There are, however, occasions when it is necessary for the program to load or read the contents of the A/B Data Buffers, for example, during diagnostic routines, or for recovery purposes. The program can access them via the

Input/Output X'4' and X'5' instructions. The A or B Data Buffer must first be selected via the A or B Data Buffer Diagnostic bit in the NSC Status/Control register (byte 0, bit 6). If the bit is off, the A Data Buffer is selected; if it is on, the B Data Buffer is selected. The Input/Output X'4/5' instructions can then access bytes 1 through 4 directly, and bytes 5 through 8 by first setting on the Diagnostic Storage Mode bit in the NSC Status/Control register (Output X'6', byte 0, bit 3). The following diagram should make this clear:

	Diagnostic Storage Mode Bit	
Instruction	Off	On
IOH/IOHI X'4'	Bytes 1 and 2	Bytes 5 and 6
IOH/IOHI X'5'	Bytes 3 and 4	Bytes 7 and 8

Programming Note

After power on, the buffers should be loaded via the Output X'4' and X'5' commands to prevent parity errors when reading.

INPUT X'6' (NSC STATUS/CONTROL REGISTER)

The register addressed by this instruction contains the current NSC status byte. The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	Channel adapter switched to interface B
	1	Channel adapter switched to interface A
	2	(not used)
	3	(not used)
	4	(not used)
	5	(not used)
	6	(not used)
	7	(not used)
1	0	NSC status byte bit 0 (attention)
	1	NSC status byte bit 1 (status modifier)
	2	NSC status byte bit 2 (control unit end)
	3	NSC status byte bit 3 (Busy)
	4	NSC status byte bit 4 (channel end)
	5	NSC status byte bit 5 (device end)
	6	NSC status byte bit 6 (unit check)
	7	NSC status byte bit 7 (unit exception)

Byte 0, Bit 0 - Channel Adapter Switched to Interface B: this bit, when on, indicates that the channel adapter is switched to interface B; byte 0, bit 1 cannot be on at the same time.

Note:

The execution of an Output X'6' instruction with byte 0, bit 1 set to 1 will also make this bit active.

Byte 0, Bit 1 - Channel Adapter Switched to Interface A: this bit, when on, indicates that the channel adapter is switched to interface A; byte 0, bit 0 cannot be on at the same time.

Note:

The execution of an Output X'6' instruction with byte 0, bit 0 set to 1 will also make this bit active.

Byte 1, Bits 0 through 7 - NSC Status Byte: these are the bits that were set into the NSC status register when an Output X'6' instruction was executed. The bits have the usual meanings of the device status byte.

OUTPUT X'6' (NSC STATUS/CONTROL REGISTER)

The register addressed by this instruction is used to set the current status of the NSC. This status is gated over the channel interface during NSC status transfer sequences. It is also used to set certain conditions in the adapter.

The instruction should be executed before signalling to the channel adapter that an NSC final status transfer sequence is required, if the status byte has not been previously given to the channel adapter. If the status byte has been given previously to the channel adapter, but has been stacked by the channel, it need not be given again.

The instruction should only be executed when an initial selection, data/status, or program controlled interrupt is set. When the NSC final status transfer sequence occurs, the status byte provided by this output is presented to the channel.

This instruction should also be used when presenting an asynchronous status, when presenting the final status byte ending a channel I/O command on the NSC, or if an early channel end is stacked.

The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	Set force A busy
	1	Set force B busy
	2	Force error
	3	Diagnostic storage mode (set = 1; reset = 0)
	4	(not used)
	5	Check the checkers
	6	A/B data buffer diagnostic mode
	7	Reset to neutral state
1	0	Set NSC status byte bit 0 (attention)
	1	Set NSC status byte bit 1 (status modifier)
	2	Set NSC status byte bit 2 (control unit end)
	3	Set NSC status byte bit 3 (busy)
	4	Set NSC status byte bit 4 (channel end)
	5	Set NSC status byte bit 5 (device end)
	6	Set NSC status byte bit 6 (unit check)
	7	Set NSC status byte bit 7 (unit exception)

Byte 0, Bit 0 - Set Force A Busy: this bit, when on, forces interface A of a two-processor switch (TPS) into the busy state, and sets interface B into the long term allegiance state (see under the heading 'Two-Processor Switch Feature' below for a definition of these terms).

Byte 0, Bit 1 - Set Force B Busy: this bit, when on, forces interface B of a two-processor switch (TPS) into the busy state, and sets interface A into the long term allegiance state.

Programming Note:

If for any reason both bits 0 and 1 are active at the same time, interface B is forced to the busy state.

Byte 0, Bit 2 - Force Error: this bit, when on, is used to force errors on the channel interface and driver receiver cards for checking purposes. It should only be set when the channel adapter is in the disabled state.

Byte 0, Bit 3 - Diagnostic Storage Mode: this bit, when on, sets the diagnostic storage mode latch. This causes the Input/Output X'4' and X'5' instructions to access bytes 5 through 8 of one of the two 8-byte data buffers instead of bytes 1 through 4. When the bit is off, it resets the diagnostic storage mode latch, and the instructions mentioned above go back to accessing bytes 1 through 4.

Byte 0, Bit 5 - Check the Checkers: this bit, when on, is used in conjunction with byte 0, bit 2 to allow the checkout of various checking circuits in the channel adapter.

Note: bits 2 and 5 need not necessarily be on at the same time.

Byte 0, Bit 6 - A/B Data Buffer Diagnostic Mode: this bit is used to select one of the two 8-byte data buffers, A and B. When the bit is off, the Input/Output X'4' and X'5' instructions access data buffer A . When the bit is on, the Input/Output X'4' and X'5' instructions access data buffer B

Byte 0, Bit 7 - Reset to Neutral State: this bit, if on, returns the channel adapter to the neutral state from the long-term allegiance state (see under the heading 'Two-Processor Switch Feature' below for a definition of these terms).

Byte 1, Bits 0 through 7 - Set NSC Status Byte: when the Output X'6' instruction is executed, these bits are transferred to the NSC status byte. They have the usual meanings of the device status byte.

INPUT X'7' (CHANNEL ADAPTER CONDITION REGISTER)

The register addressed by this instruction contains information mainly concerning the enabled/disabled status of the channel adapter interfaces.

Note: the channel adapter condition register contains information concerning all the channel adapters.

The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	Channel adapter 5 enabled
	1	(not used)
	2	Channel adapter 6 enabled
	3	(not used)
	4	(not used)
	5	NSC address active
	6	PIO mode
	7	(not used)
1	0	Channel adapter 1 interface A enabled
	1	Channel adapter 1 interface B enabled
	2	Channel adapter 2 interface A enabled
	3	Channel adapter 2 interface B enabled
	4	Channel adapter 3 interface A enabled
	5	Channel adapter 3 interface B enabled
	6	Channel adapter 4 interface A enabled
	7	Channel adapter 4 interface B enabled

Byte 0, Bit 0 - Channel Adapter 5 Enabled: this bit is set to 1 by the channel hardware when channel adapter 5 is enabled. It is reset when channel adapter 5 is disabled.

Byte 0, Bit 2 - Channel Adapter 6 Enabled: this bit is set to 1 by the channel hardware when channel adapter 6 is enabled. It is reset when channel adapter 6 is disabled.

Byte 0, Bit 5 - NSC Address Active: this bit is set by hardware under the following conditions:

- When the NSC is initially selected by accepting a command.
- When the program executes an Output X'2' instruction with byte 0, bit 2 ('Status Transfer Sequence') on.

The bit remains active until the 'Device End' ending status is accepted for this command.

The bit is reset when an NSC 'Device End' status is accepted by the host channel on an NSC final status transfer.

Byte 0, Bit 6 - PIO Mode: this bit, when on, indicates that the last Output X'2' instruction set the PIO data transfer mode; all subsequent data transfer sequences use the PIO mode until it is reset. When the bit is off, it indicates that the last Output X'2' instruction set the AIO data transfer mode.

Programming Note:

After a reset, the bit is set to 0, forcing AIO mode.

Byte 1, Bit 0 - Channel Adapter 1, Interface A Enabled: this bit is set to 1 by the channel hardware when interface A of channel adapter 1 is enabled and reset when it is disabled. If a TPS is not installed on this channel, it is set when channel adapter 1 is enabled and reset when it is disabled.

Byte 1, Bit 1 - Channel Adapter 1, Interface B Enabled: this bit is set to 1 by the channel hardware when interface B of channel adapter 1 (equipped with a TPS) is enabled.

Byte 1, Bit 2 - Channel Adapter 2, Interface A Enabled: this bit is set to 1 by the channel hardware when interface A of channel adapter 2 is enabled and reset when it is disabled. If a TPS is not installed on this channel, it is set when channel adapter 2 is enabled and reset when it is disabled.

Byte 1, Bit 3 - Channel Adapter 2, Interface B Enabled: this bit is set to 1 by the channel hardware when interface B of channel adapter 2 (equipped with a TPS) is enabled.

Byte 1, Bit 4 - Channel Adapter 3, Interface A Enabled: this bit is set to 1 by the channel hardware when interface A of channel adapter 3 is enabled and reset when it is disabled. If a TPS is not installed on this channel, it is set when channel adapter 3 is enabled and reset when it is disabled.

Byte 1, Bit 5 - Channel Adapter 3, Interface B Enabled: this bit is set to 1 by the channel hardware when interface B of channel adapter 3 (equipped with a TPS) is enabled.

Byte 1, Bit 6 - Channel Adapter 4, Interface A Enabled: this bit is set to 1 by the channel hardware when interface A of channel adapter 4 is enabled and reset when it is disabled. If a TPS is not installed on this channel, it is set when channel adapter 4 is enabled and reset when it is disabled.

Byte 1, Bit 7 - Channel Adapter 4, Interface B Enabled: this bit is set to 1 by the channel hardware when interface B of channel adapter 4 (equipped with a TPS) is enabled.

OUTPUT X'7' (CHANNEL ADAPTER CONTROL REGISTER)

This instruction is recognized by all channel adapters. The main purpose of this register is to select one of the six channel adapters:

- For the duration of the instruction (temporary selection).
- Until the channel adapter selection is changed, either by the auto-selection mechanism, or by another Output X'7' instruction.

This instruction is also used to control channel adapter operations by setting and resetting control latches. The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	Enable auto-selection) cannot be on at
	1	Disable auto-selection) the same time.
	2	Select channel adapter addressed by bits 4-6
	3	Execute output on CA addressed by bits 4-6
	4	Channel adapter selection bit 0)
	5	Channel adapter selection bit 1) CA address 1-6
	6	Channel adapter selection bit 2)
	7	Channel adapter reset
1	0	Set suppress out monitor
	1	Set program requested interrupt
	2	Reset channel adapter interrupt level 1 checks
	3	Reset system reset/NSC address active
	4	Set allow channel interface enable (A and B)
	5	Set ESC operational
	6	Set ESC command free
	7	Set allow channel interface disable (A and B)

Byte 0, Bit 0 - Enable Auto-Selection: this bit, when on, enables the auto-selection mechanism for all channel adapters.

Byte 0, Bit 1 - Disable Auto-Selection: this bit, when on, disables the auto-selection mechanism for all channel adapters.

Programming Note to Bits 0 and 1: byte 0, bits 0 and 1 must not both be on together.

Byte 0, Bit 2 - Select Channel Adapter Addressed by Bits 4 through 6: this bit, when on, causes the channel adapter whose address is contained in byte 0, bits 4 through 6 to be selected for all subsequent channel operations.

Byte 0, Bit 3 - Execute Output on Channel Adapter Addressed by Bits 4 through 6: this bit, when on, allows a particular channel adapter to be temporarily selected in order to set or reset one or more of the conditions specified by

byte 0, bit 7, and byte 1, bits 0 through 7. Byte 0, bits 4 through 6 indicate in which channel adapter the condition is to be set or reset.

Note: the channel adapter which was selected by the auto-selection mechanism, or by the execution of this instruction with byte 0, bit 2 on, is not changed.

Programming Notes to Bits 2 and 3:

1. If one of the installed channel adapters has already been selected, either by the auto-selection mechanism or by a previous Output X'7' instruction, bits 2 and 3 may both be off.

2. If none of the installed channel adapters has been previously selected, either bit 2 or bit 3 must be on, and a valid channel adapter address must be contained in bits 4 through 6.

Byte 0, Bits 4 through 6 - Channel Adapter Selection Bits: these bits form the address of the channel adapter selected by byte 0, bits 2 or 3, either temporarily, or for subsequent instructions. The three bits are decoded as follows:

Bit 4 5 6	Channel Adapter
0 0 0	1
0 0 1	2
0 1 0	3
0 1 1	4
1 0 0	5
1 0 1	6

Note: if the Output X'7' instruction addresses a channel adapter that is not installed, the CCU hardware times out and sets the 'PIO Halt Remember' bit in register X'D'. The lagging address register (LAR) points to the failing instruction. All channel adapters raise a level 1 interrupt.

Byte 0, Bit 7 - Channel Adapter Reset: this bit simulates a 'power-on reset' in the channel adapters. It should be executed only when the channel interface is not enabled, or as a last resort when the channel adapter is hung up on the interface.

Programming Note:

The program should first execute an Output X'7' instruction with byte 1, bit 7 'Set Allow Interface Disable' = 1 in an attempt to disable both interfaces A and B (if installed) before executing this instruction.

Byte 1, Bit 0 - Set Suppress Out Monitor: this bit, when on, causes the channel adapter to monitor for the inactive state of the 'Suppress Out' tag line. If this inactive state is detected, the channel adapter sets a data/status level 3 interrupt request, and also byte 0, bit 6 of register X'2' ('Suppress Out Monitor Interrupt'), which may be read using an Input X'2' instruction.

Programming Note:

This bit may be used by the program after a stacked status condition to cause the channel adapter to signal when the suppress status indication has been removed.

Byte 1, Bit 1 - Set Program Requested Interrupt: this bit, when on, causes a channel adapter data/status interrupt request and byte 0, bit 7 of register X'2' ('Program Requested Interrupt') to be set immediately, unless one of the following conditions occurs:

1. A data/status transfer sequence has been initiated.

2. The host is initiating an initial selection sequence on the channel.

3. Chaining is indicated.

4. In two-processor switch operation, if a tagged 'Device End' status is being presented to a previous 'Busy' status.

In this case, the level 3 data/status interrupt is not set until these sequences are complete.

Byte 1, Bit 2 - Reset Channel Adapter Interrupt Level 1 Checks: this bit, when on, causes the channel adapter to reset the channel adapter level 1 check latches, which in turn resets the channel adapter level 1 interrupt request.

Byte 1, Bit 3 - Reset System Reset/NSC Address Active: this bit, when on, causes the channel adapter to reset 'System Reset' (Input X'0', byte 0, bit 7) and 'NSC Active' (Input X'7', byte 1, bit 5). If the channel adapter initial level 3 interrupt request is found to be due to the detection of a system reset sequence, this bit must be used to reset 'system reset' and the associated level 3 interrupt.

Byte 1, Bit 4 - Set Allow Channel Interface Enable (A and B): this bit, when on, causes the channel adapter to set the 'Allow Channel Interface Enable' latch for both interfaces A and B. When the 'Enable Interface A' and/or 'Enable Interface B' signal(s) is sent to the channel adapter from the MOSS, the appropriate interfaces are enabled if 'Select Out' from the channel is not active. This bit must not be on simultaneously with byte 1, bit 7.

Programming Note:

After a power on reset, the channel interface cannot be enabled until the Output X'7' instruction is executed with this bit on.

Byte 1, Bit 5 - Set ESC Operational: this bit, when on, causes the channel adapter to make the emulator subchannel (ESC) addresses operational.

Byte 1, Bit 6 - Set ESC Command Free: this bit, when on, causes the channel adapter to reset the 'ESC Command Active' latch (this latch is set when the channel adapter hardware detects an initial selection sequence to an ESC address). When the latch is off, it indicates that the ESC part of the channel adapter is free of commands. Since the channel adapter cannot disable the

interface until it is free of commands, the program must set this bit whenever it detects that it is free of ESC commands; if the hardware has previously set this latch due to an ESC initial selection, the channel adapter will not disable the interface.

Byte 1, Bit 7 - Set Allow Channel Interface Disable (A and B): this bit, when on, causes the channel adapter to set the 'Allow Channel Interface Disable' latch for both interfaces A and B. This latch overrides channel adapter enable/disable when the interfaces are free of commands, no chaining is specified, not in an initial selection sequence, there is no pending device end status, the 'Operational In' driver is not active, and 'Select Out' is not active. This bit must not be on simultaneously with byte 1, bit 4.

Programming Note:

If a status with at least the Unit Check bit set (Output X'6', byte 1, bit 6) is exchanged before the enable/disable latch has been overriden, a channel adapter reset (Output X'7', byte 0, bit 7) must be sent to the channel adapter. This reset must be sent within 500 miliseconds after the channel adapter has been found to be in the disabled state (as indicated by the corresponding bit(s) in the Input X'7' instruction).

INPUT/OUTPUT X'B' (ESC TEST I/O ADDRESS AND STATUS REGISTER)

The register addressed by these instructions contains the Test I/O address in byte 0 and the Test I/O status in byte 1. The register is loaded either by the control program via the Output X'B' instruction, or by the stacking of a final status on the ESC.

The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0-7	ESC TIO address byte bits 0-7
1	0	ESC TIO status byte bit 0 (attention)
	1	ESC TIO status byte bit 1 (status modifier)
	2	ESC TIO status byte bit 2 (control unit end)
	3	ESC TIO status byte bit 3 (busy)
	4	ESC TIO status byte bit 4 (channel end)
	5	ESC TIO status byte bit 5 (device end)
	6	ESC TIO status byte bit 6 (unit check)
	7	ESC TIO status byte bit 7 (unit exception)

INPUT X'C' (CYCLE STEAL MODE CONTROL REGISTER)

The register addressed by these instructions is used in AIO mode; it contains various cycle steal controls in byte 0, and a residual byte count in byte 1. The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	SYN monitor latch
	1	DLE remember control latch
	2	USASCII monitor control latch
	3	EBCDIC monitor control latch
	4	(not used)
	5	(not used)
	6	(not used)
	7	(not used)
1	0-7	Residual byte count bits 0-7

Byte 0, Bit 0 - SYN Monitor Latch: this bit is used only on BSC inbound (host to 3725) operations. It indicates that a stream of four SYN characters was detected in the incoming data stream during the current operation. See under the heading 'BSC Control Character Recognition' for the use of this bit.

Byte 0, Bit 1 - DLE Remember Control Latch: this bit is used only on BSC inbound (host to 3725) operations and indicates the state of the 'DLE Remember Control' latch. This latch is set by the channel adapter hardware each time that a DLE character is detected in the incoming data stream; it is reset by the following character (if it is not another DLE). If the DLE is the last character in the transfer sequence, the latch will stay on, and the input instruction will find byte 0, bit 1 on. See under the heading 'BSC Control Character Recognition' for the use of this bit.

Programming Note:

If the Input X'C' instruction finds this bit on, it must restore the bit to activity when a new inbound transfer sequence is initiated for the same subchannel address (this is to continue the test for the start of transparent mode).

Byte 0, Bit 2 - USASCII Monitor Control Latch: this bit is used only on BSC inbound (host to 3725) operations and indicates that monitoring was carried out by the channel adapter hardware on the last transfer sequence for certain USASCII control characters. See under the heading 'BSC Control Character Recognition' for the use of this bit.

Byte 0, Bit 3 - EBCDIC Monitor Control Latch: this bit is used only on BSC inbound (host to 3725) operations and indicates that monitoring was carried out by the channel adapter hardware on the last transfer sequence for certain EBCDIC control characters. See under the heading 'BSC Control Character Recognition' for the use of this bit.

Byte 1, Bits 0 through 7 - Residual Byte Count: this byte contains the residual byte count for the transfer sequence that has just ended. On outbound (3725 to host) operations, the residual count indicates the number of bytes that were not transferred to the host. On inbound (host to 3725) operations, the residual count indicates the difference between the number of bytes requested, and the number of bytes actually transferred from the host.

OUTPUT X'C' (CYCLE STEAL MODE CONTROL REGISTER)

The register addressed by these instructions is used in AIO mode; it contains various cycle steal controls in byte 0, and a byte count in byte 1. The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	SYN monitor control latch (Note)
	1	DLE remember control latch (Note)
	2	USASCII monitor control latch (Note)
	3	EBCDIC monitor control latch (Note)
	4	(not used)
	5	(not used)
	6	(not used)
	7	(not used)
1	0-7	Request byte count bits 0-7

Note: 1 = set; 0 = reset

Byte 0, Bit 0 - SYN Monitor Control Latch: this bit is used only on BSC inbound (host to 3725) operations. When on, it sets the 'SYN Monitor Control' latch. It causes the hardware to monitor the data coming from the host for SYN characters. If four consecutive SYN characters are detected in the incoming data stream, the data transfer is terminated, and a channel adapter data/status level 3 interrupt is requested. The 'SYN Monitor Control Latch' is reset when any non-SYN character is detected. When the bit is off, the 'SYN Monitor Control' latch is reset, and SYN monitoring is stopped.

Byte 0, Bit 1 - DLE Remember Control Latch: this bit is used only on BSC inbound (host to 3725) operations and is used to restore the state of the 'DLE Remember Control' latch at the start of a new transfer sequence (this is to continue the test for the start of transparent mode). See under the heading 'BSC Control Character Recognition' for the use of this bit.

Byte 0, Bit 2 - USASCII Monitor Control Latch: this bit is used only on BSC inbound (host to 3725) operations. When on, it indicates that monitoring for certain USASCII control characters is to be carried out by the channel adapter hardware. See under the heading 'BSC Control Character Recognition' for the use of this bit.

Byte 0, Bit 3 - EBCDIC Monitor Control Latch: this bit is used only on BSC inbound (host to 3725) operations. When on, it indicates that monitoring for certain EBCDIC control characters is to be carried out by the channel adapter hardware. See under the heading 'BSC Control Character Recognition' for the use of this bit.

Byte 1, Bits 0 through 7 - Request Byte Count: this byte contains the count of the number of bytes that are to be transferred to/from the host.

Programming Note:

The Request Byte Count loaded into byte 1 of register X'C' must never be greater than the storage size of the buffer reserved for the AIO data transfer. The total number of bytes transferred by the channel adapter will never be greater than the initial value loaded by the Output X'C' instruction.

The contents of the cycle steal address pointer register are unpredictable at the end of any data transfer, and should not be used to determine the exact number of bytes that were transferred. The exact number of bytes transferred may be determined via an Input X'C' instruction. The cycle steal pointer must always be set via an Output X'30' through X'35' instruction before starting an AIO transfer using an Output X'2' instruction.

INPUT X'D' (CHANNEL ADAPTER LEVEL 1 INTERRUPT CHECK REGISTER)

The register addressed by this instruction is set by hardware with the various checks that can cause a level 1 interrupt. Other bits do not themselves cause a level 1 interrupt, but may help to localize the cause of the interrupt. The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	PIO bus parity error
	1	Internal bus parity error
	2	CCU interconnection card check
	3	(not used)
	4	Channel interface card check
	5	Address compare error
	6	Initiate service latch ungated
	7	(not used)
1	0	Output exception check
	1	PIO halt remember
	2	Cycle steal halt remember
	3	Bus in check interface A
	4	Ground Fault Error
	5	Bus in check interface B
	6	Driver/receiver card check interface A
	7	Driver/receiver card check interface B

Byte 0, Bit 0 - PIO Bus Parity Error: this bit, when on, indicates that a bad parity has been detected on the PIO bus between the CCU and the channel adapter. If the error was detected on data transferred from the CCU to the channel adapter, this bit is set. This bit does not cause a level 1 interrupt request.

Byte 0, Bit 1 - Internal Bus Parity Error: this bit, when on, indicates that a bad parity has been detected between the channel interface card and the CCU interconnection card.

Byte 0, Bit 2 - CCU Interconnection Card Check: this bit, when on, indicates that a hardware failure has been detected on the CCU interconnection card. Four different hardware failures may cause this check:

1. PIO bus parity error on inbound (host to 3725) operations.

2. Internal bus parity error on outbound (3725 to host) operations.

3. I/O command decoder failure.

   Note: if an invalid instruction (X'8', X'9', or X'A') is executed, the I/O command decoder will fail and cause this check; in this case, it is a program check.

4. Byte counter failure.

Byte 0, Bit 4 - Channel Interface Card Check: this bit, when on, indicates that a hardware failure has been detected on the channel interface card.

Byte 0, Bit 5 - Address Compare Error: this bit, when on, indicates that the program has addressed an emulator subchannel that is outside of the plugged address range.

Byte 0, Bit 6 - Initiate Service Latch Ungated: this bit, when on, is set on at the start of a data or status transfer initiated by the control program. It indicates that the channel adapter is starting a control unit initiated sequence by raising 'Request In' to the host, or that the channel adapter is actually transferring data or a status to the host. This bit does not cause a level 1 interrupt request.

Byte 1, Bit 0 - Output Exception Check: this bit, when on, indicates that the channel adapter hardware has detected an invalid Output instruction. Output instructions, with the single exception of Output X'7', are not allowed during data/status transfer. The Output X'B' instruction is only allowed during an initial select level 3 interrupt. This bit does not cause a level 1 interrupt request.

Byte 1, Bit 1 - PIO Halt Remember: this bit, when on, indicates that the CCU has detected an error during an Input/Output operation, and has activated the 'Halt' signal. This bit does not cause a level 1 interrupt request.

Byte 1, Bit 2 - Cycle Steal Halt Remember: this bit, when on, indicates that the CCU has detected an error during cycle stealing, and has activated the 'Halt' signal. This bit does not cause a level 1 interrupt request.

Byte 1, Bit 3 - Bus In Check Interface A: this bit, when on, indicates that a hardware failure has occurred on the channel adapter internal bus path during a data or address transfer to the host. This condition was detected on interface A.

Byte 1, Bit 4 - Ground Fault Error: this bit, when on, indicates that a channel bus in or tag in (except Select In) signal is shorted to ground.

Byte 1, Bit 5 - Bus In Check Interface B: this bit, when on, indicates that a hardware failure has occurred on the channel adapter internal bus path during a data or address transfer to the host. This condition was detected on interface B.

Byte 1, Bit 6 - Driver/Receiver Card Check Interface A: this bit, when on, indicates:

1. A hardware failure has been detected on the interface A driver/receiver card caused by a parity error on the channel interface card during an inbound (host to 3725) data transfer. This parity error is detected when the bus out check is inactive, but the channel interface card detects bad parity on data transferred from the bus out register.

2. A hardware failure was detected in the Bus In or Tag In (except Select In) interface drivers.

Byte 1, Bit 7 - Driver/Receiver Card Check Interface B: this bit, when on, indicates:

1. A hardware failure has been detected on the interface B driver/receiver card caused by a parity error on the channel interface card during an inbound (host to 3725) data transfer. This parity error is detected when the bus out check is inactive, but the channel interface card detects bad parity on data transferred from the bus out register.

2. A hardware failure was detected in the Bus In or Tag In (except Select In) interface drivers.

Programming Note to Byte 0, Bit 0 and Byte 1, Bits 1 and 2: these bits do not cause a level 1 interrupt request to the CCU. The channel adapter hardware reports the error, and the control program takes appropriate action, thus avoiding double reporting of errors.

INPUT X'E' (CHANNEL ADAPTER LEVEL 1 INTERRUPT REQUESTS)

The register addressed by this instruction indicates which channel adapter(s) has a level 1 interrupt pending. It may be read via the Input X'E' instruction when servicing a level 1 interrupt.

Note: the channel adapter level 1 interrupt requests register contains information concerning all the channel adapters.

The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	Channel adapter 5 level 1 interrupt request
	1	(not used)
	2	Channel adapter 6 level 1 interrupt request
	3	Channel adapter (any) level 1 interrupt request
	4	Channel adapter address bit 0)
	5	Channel adapter address bit 1) CA address 1-6
	6	Channel adapter address bit 2)
	7	(not used)
1	0	Channel adapter 1 level 1 interrupt request
	1	(not used)
	2	Channel adapter 2 level 1 interrupt request
	3	(not used)
	4	Channel adapter 3 level 1 interrupt request
	5	(not used)
	6	Channel adapter 4 level 1 interrupt request
	7	(not used)

Byte 0, Bit 0 - Channel Adapter 5 Level 1 Interrupt Request: this bit, when on, indicates that channel adapter 5 has a level 1 interrupt request.

Byte 0, Bit 2 - Channel Adapter 6 Level 1 Interrupt Request: this bit, when on, indicates that channel adapter 6 has a level 1 interrupt request.

Byte 0, Bit 3 - Channel Adapter (Any) Level 1 Interrupt Request: this bit, when on, indicates that one or more of the channel adapters has a level 1 interrupt request.

Byte 0, Bits 4 through 6 - Channel Adapter Address: these bits identify the currently selected channel adapter, as follows:

Bit 4 5 6	Channel Adapter
0 0 0	1
0 0 1	2
0 1 0	3
0 1 1	4
1 0 0	5
1 0 1	6

Byte 1, Bit 0 - Channel Adapter 1 Level 1 Interrupt Request: this bit, when on, indicates that channel adapter 1 has a level 1 interrupt request.

Byte 1, Bit 2 - Channel Adapter 2 Level 1 Interrupt Request: this bit, when on, indicates that channel adapter 2 has a level 1 interrupt request.

Byte 1, Bit 4 - Channel Adapter 3 Level 1 Interrupt Request: this bit, when on, indicates that channel adapter 3 has a level 1 interrupt request.

Byte 1, Bit 6 - Channel Adapter 4 Level 1 Interrupt Request: this bit, when on, indicates that channel adapter 4 has a level 1 interrupt request.

INPUT X'F' (CHANNEL ADAPTER LEVEL 3 INTERRUPT REQUESTS)

The register addressed by this instruction indicates which channel adapter is currently selected, and the status of its level 3 interrupts. It may be read via the instruction when servicing a level 3 interrupt.

If the 'Auto-Selection Complete' latch is not set, execution of this instruction initiates the auto-select mechanism (the 'Auto-Selection Complete' latch is set when the interrupt request information is presented in response to the Input X'F' instruction). The latch is reset by:

- An Output X'0' instruction.
- An Output X'2' instruction with either byte 0, bit 5 or 6 set to 1.
- An output X'7' instruction with byte 1, bit 3 set to 1 (for a system reset only).
- An Output X'B' instruction.

For a full description of the auto-selection mechanism, refer to the section 'Channel Adapter Selection by the Auto-Selection Mechanism' at the beginning of this chapter.

The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	(not used)
	1	Two-processor switch installed
	2	Selected CA initial selection L3 interrupt request
	3	Selected CA data/status L3 interrupt request
	4	Channel adapter address bit 0)
	5	Channel adapter address bit 1) CA address 1-6
	6	Channel adapter address bit 2)
	7	(not used)
1	0-7	(not used)

Byte 0, Bit 1 - Two-Processor Switch Installed: this bit, when on, indicates that the currently selected channel adapter is equipped for two-processor switch operation.

Byte 0, Bit 2 - Selected CA Initial Selection L3 Interrupt Request: this bit, when on, indicates that the currently selected channel adapter has an initial selection level 3 interrupt request pending.

Byte 0, Bit 3 - Selected CA Data/Status L3 Interrupt Request: this bit, when on, indicates that the currently selected channel adapter has a data/status level 3 interrupt request pending.

Byte 0, Bits 4 through 6 - Channel Adapter Address: these bits identify the currently selected channel adapter, as follows:

Bit 4 5 6	Channel Adapter
0 0 0	1
0 0 1	2
0 1 0	3
0 1 1	4
1 0 0	5
1 0 1	6

SECTION 4. CHANNEL ADAPTER PROGRAMMING CONSIDERATIONS

CHANNEL ADAPTER INTERRUPT REQUEST HANDLING

LEVEL 1 INTERRUPT REQUESTS

When an error condition is detected in the channel adapter, a level 1 interrupt occurs, and a bit is set in the Channel Adapter Level 1 Interrupt Check Bit register (X'D') to indicate the type of error, as follows:

Byte	Bit	Meaning
0	0	PIO Bus Parity Error
	1	Internal Bus Parity Error
	2	CCU Interconnection Card Check
	3	(not used)
	4	Channel Interface Card Check
	5	Address Compare Error
	6	(not used)
	7	(not used)
1	0	Output Exception Check
	1	PIO Halt Memory Latch
	2	AIO Halt Memory Latch
	3	Bus In Check Interface A
	4	Bus In Check Interface B
	5	(not used)
	6	Driver/Receiver Card Check I/F A
	7	Driver/Receiver Card Check I/F B

These bits are available to the program via the Input X'D'instruction.

LEVEL 3 INTERRUPT REQUESTS

There are two types of interrupt request at level 3:

- Channel Adapter Initial Selection Level 3 interrupt request
- Data/Status Transfer Level 3 interrupt request

The type of interrupt request, and the channel number, may be identified by issuing an Input X'F' (Channel Adapter Level 3 Interrupt Requests) instruction:

- Byte 0, bit 2 indicates a Channel Adapter Initial Selection Interrupt Request.

- Byte 0, bit 3 indicates a Channel Adapter Data/Status Interrupt Request.

Both types of interrupt requests may be active at the same time; this condition is indicated by both bits being on in the register.

Channel Adapter Initial Selection Level 3 Interrupt Request

When an Initial Selection interrupt request occurs, the cause of the interrupt may be determined by executing an Input X'0' (Initial Selection Control register) instruction:

- Byte 0, bit 0 indicates that the interrupt was caused by a normal initial selection interrupt request.

- Byte 0, bit 1 indicates that the interrupt request was caused by an interface disconnect sequence (Halt I/O).

- Byte 0, bit 2 indicates that the interrupt request was caused by a selective reset.

- Byte 0, bit 3 indicates that the interrupt request was caused by the detection of bad (even) parity on the I/O channel interface bus out when the channel I/O command byte was presented during initial selection.

- Byte 0, bit 7 indicates that the interrupt request was caused by a system reset.

Note: if the interrupt was caused by a system reset, all the bits of register X'0' will be off, except the System Reset bit (byte 0, bit 7). This is because the system reset sequence resets all the latches in the channel, with the exception of the system reset bit. This means that if an initial selection sequence occurs just before a system reset sequence, all the indications will be lost to the program, unless the initial selection interrupt was completely serviced before the system reset occurred.

Once the Channel Adapter Initial Selection Level 3 Interrupt Request is set, the channel adapter hardware replies with a short control unit busy (CU End, Status Modifier, and Busy) to all attempts at initial selection until the control program tells the channel adapter to reset the condition that caused the initial selection interrupt. During this period, no channel commands can be accepted. The control program should therefore tell the channel adapter to reset the interrupting condition as soon as possible.

Note: During this period, subsequent data/status transfer sequences are also inhibited. This means that it is possible to have both an initial selection request and a data/status request simultaneously only if the data/status request occurs first, and if the subsequent initial selection request is not due to a System Reset condition.

A Channel Adapter Initial Selection Level 3 Interrupt Request may be reset either by executing an Output X'0' instruction (the bit configuration is ignored), or by executing an Output X'2' instruction with byte 0, bit 5 (reset initial selection) set to 1.

Notes:

1. If the interrupt was due to a System Reset, it must be reset by executing an Output X'7' instruction with byte 1, bit 3 (reset system reset/NSC address active) set to 1.

2. When executing Output X'2' or Output X'7' instructions to reset interrupt requests, the control program must be careful to set/reset all bits correctly to achieve the desired result.

Channel Adapter Data/Status Level 3 Interrupt Request

When a Data/Status interrupt request occurs, the cause of the interrupt may be determined by executing an Input X'2' (Data/Status Control register) instruction:

- Byte 0, bit 0 indicates that the interrupt was caused by the ending of an outbound (3725 to host) data transfer sequence.

- Byte 0, bit 1 indicates that the interrupt was caused by the ending of an inbound (host to 3725) data transfer sequence.

- Byte 0, bit 2 indicates that the interrupt was caused by the ending of a status transfer sequence.

- Byte 0, bit 7 indicates that the interrupt was a program requested interrupt.

Note: if a system reset occurs immediately afterwards, all the bits of register X'2' will be off, but the System Reset bit (byte 0, bit 7 of register X'0' will be on. This is because the system reset sequence resets all the latches in the channel, with the exception of the system reset bit. This means that if a data/status sequence occurs just before a system reset sequence, all the indications will be lost to the program, unless an Input X'2' instruction was executed before the system reset occurred.

A Channel Adapter Data/Status Level 3 Interrupt Request may be reset by executing an Output X'2' instruction with byte 0, bit 6 (reset data/status interrupt) set to 1. Unless the control program wants to immediately initiate another transfer sequence, byte 0, bit 0 (reset outbound data transfer), bit 1 (reset inbound data transfer) and bit 2 (reset status transfer) should be set to 0. The execution of an Output X'2' instruction also resets the 'program requested interrupt' (byte 0, bit 7 of Input X'2'), and the 'suppress out monitor interrupt' (byte 0, bit 6 of Input X'2').

Note: when executing Output X'2' or Output X'7' instructions to reset interrupt requests, the control program must be careful to set/reset all bits correctly to achieve the desired result.

Simultaneous Initial Selection and Data/Status Interrupts

When servicing channel adapter level 3 interrupts, the control program should first check whether both Initial Selection and Data/Status interrupts are set, or only one of them.

If only a Channel Adapter Initial Selection Level 3 interrupt request is set, this may be serviced without worrying about the possibility of a Channel Adapter Data/Status Level 3 request occurring afterwards, since any pending data/status transfer sequences are inhibited until the initial selection request is reset. However, if Output X'2' is used to reset the interrupt, it must be remembered that this instruction also controls data/status transfers, and the control program must be prepared to reinitiate a pending transfer sequence if it still requires that transfer to be executed.

If both interrupt requests are set (initial selection request received after data/status request), the program must service both requests before executing the Output X'2' instruction. Remember too that a system reset sequence can occur before the Input X'2' is executed.

CHANNEL COMMANDS

Channel commands are commands issued by the channel to the 3725. All I/O command byte combinations are valid to the channel adapter hardware provided that good parity is found on the channel interface bus out.

The following channel commands are standard:

		Applicable to	
Hex	Meaning	NSC	ESC
00	Test I/O (TIO)	*	*
01	Write	*	*
02	Read	*	*
03	I/O No-op	*	*
04	Sense	*	*
05	Write IPL	*	
09	Write break	*	
E4	Sense ID	*	

Test I/O (TIO) - (X'00')

NSC

When this command is issued to the NSC address, the channel adapter replies with the current status of the NSC:

- If the NSC is free of commands, the channel adapter replies with an X'00' with a hardware generated status byte during the initial status presentation to the Test I/O command.
- If the NSC is active, and the status is not available in the NSC status register, the channel adapter replies with a hardware generated busy (X'10') status during the initial status presentation to the test I/O command.
- If the NSC has a pending status available in the NSC (software) status register, this status is sent to the channel in response to the Test I/O command. An initial selection level 3 interrupt is raised, with byte 0,

bit 6 (Status Byte Cleared) set in register X'0'. There is no busy bit in this status.

Note: the TIO command must be recognized by the control program if there is a status stacked.

ESC

- When this command is issued to an ESC address, the line address presented to the channel adapter on the channel bus out is compared with the line address contained in register X'B'. If the two addresses compare equal, the status contained in Byte 1 of register X'B' is presented to the channel.

- If the two addresses do not compare (this will usually be the case), the Test I/O status is not immediately available for that address; the channel adapter hardware presents a short control unit busy status X'70' (Status Modifier, Control Unit End, and Busy), and raises a channel adapter initial selection level 3 interrupt request.

  The control program must obtain the ESC address and command via an Input X'1' instruction, and then execute an Output X'B' instruction to load register X'B' with the ESC address of the line to be serviced in Byte 0, and the status of the line in Byte 1.

  When the next initial selection sequence occurs, the channel adapter hardware compares the address on bus out with the contents of Byte 0 of register X'B'. If it is for the same address, an address compare equal occurs, and the status, contained in Byte 1 of register X'B', is sent to the channel. At the same time, register X'0' Byte 0, Bit 4 (ESC Operation), and Byte 0, Bit 6 (Status Byte Cleared) are set on by hardware, and a channel adapter initial selection level 3 interrupt occurs. The control program may obtain this information by executing an Input X'0' instruction.

Notes:

1. If the addresses do not compare equal during the initial selection sequence, or if the command is not Test I/O, the hardware handles the selection sequence as a standard initial selection.

2. From the time that the Test I/O command is first issued until the control program reacts by executing the Output X'B' instruction, the channel adapter hardware responds to all initial selection sequences from the host with the short control unit busy status X'70' (Status Modifier, Control Unit End, and Busy).

Write - (X'01')

This command is used to transfer data or control information from the host to the 3725. When the command is issued, the channel adapter hardware accepts the command and returns an initial selection status of X'00'. The control program must decode the command and initiate the appropriate action.

Read - (X'02')

This command is used to transfer data from the 3725 to the host. When the command is issued, the channel adapter hardware accepts the command and returns an initial selection status of X'00'. The control program must decode the command and initiate the appropriate action.

I/O No-Op - (X'03')

NSC

This command is a 'dummy' command. When the command is issued, the channel adapter hardware returns an immediate initial selection status of channel end and device end (X'0C') if the channel adapter is free.

If the channel adapter has an Initial Selection or a Program Requested Interrupt pending, a Control Unit Busy status (X'70') with bits 1 (Status Modifier), 2 (Control Unit End), and 3 (Busy) is returned instead.

If the original channel end/device end status is not stacked, no initial selection level 3 interrupt occurs.

If a pending status is available (previous NSC status byte stacked) the channel adapter presents this stacked status to the No-Op command, along with the busy bit. An initial selection level 3 interrupt occurs with byte 0, bit 6 (Status Byte Cleared) in register X'0'.

Note: the No-Op command must be recognized by the control program if there is a status stacked.

ESC

This command is a 'dummy' command. When the command is issued to a valid ESC address, the channel adapter hardware returns an immediate initial selection status of channel end and device end (X'0C') if the channel adapter is free.

If the channel adapter has an Initial Selection or a Program Requested Interrupt pending, a Control Unit Busy status (X'70') with bits 1 (Status Modifier), 2 (Control Unit End), and 3 (Busy) is returned instead.

Sense - (X'04')

This command is used to transfer a single byte of sense information from the 3725 to the host. When the command is issued, the channel adapter hardware accepts the command and returns an initial selection status of X'00'. The normal ending status is channel end and device end (X'0C'), unless a Halt I/O command is detected when the channel adapter is not initialized. In this case, the response is channel end, device end, and unit check (X'0E'). The control program must decode the command create the correct sense byte, and send it to the host. The transfer takes place in the same way as a data transfer with a single byte of data.

Write IPL - (X'05')

This command prepares the 3725 to receive the control program from the host. No data is actually transferred by the command.

1. The initial status is always channel end alone, thus causing the channel to disconnect immediately. This is necessary because the 3725 may or may not be ready; if the 3725 is not ready, it may take several minutes before the loader program is loaded and ready to receive the control program from the host.

2. A level 3 interrupt to the CCU is raised with the channel end; at the same time, the MOSS is informed of the 'Write IPL' command by hardware, and initializes the 3725 with the loader program, if necessary.

3. When the loader program in the 3725 is ready, it sends device end to the channel. If the 3725 is ready immediately, the device end follows the channel end almost at once; however, they never occur together in the same status presentation.

Write Break - (X'09')

This command is exactly the same as the normal write command, with one exception: the command code to be found in byte 1 of register X'1' is X'09' instead of X'01'. This allows the host to inform the control program of the point it has reached in the host CCW chain. The control program should react in the same way as to a normal write command.

Sense ID - (X'E4')

This command is used to determine the unit type. The control program must set up an outbound transfer sequence (3725 to host) to transfer the Unit Identification and Level.

Non-Standard Commands

As previously stated, the channel adapter recognizes all I/O command byte combinations as valid, provided that correct parity is detected on the channel interface bus out. It is the responsibility of the control program to test for validity at the control program level. If an invalid command is received at this level, the control program must end the command by setting up a final status transfer with at least Channel End, Device End, and Unit Check (X'0E').

NSC

When a non-standard command is received by the channel adapter, it replies with an initial status of channel end (X'08') and raises an initial selection level 3 interrupt to the CCU.

If the CE status is stacked, a level 3 interrupt is presented to the CCU to indicate the stacked status. The channel end status is available until an Output X'6' is performed, or until a Halt I/O or a Selective Reset is sent from the host.

ESC

When a non-standard command is received by the channel adapter, it replies with an all-zero initial status and raises an initial selection level 3 interrupt to the CCU. If the ESC control program determines that a particular command byte is invalid, it must terminate the command with an ending status of at least X'0E' (Channel End, Device End, and Unit Check) to that ESC address.

CHANNEL INITIAL STATUS

NSC Initial Status

At initial selection, the status returned to the channel may be one of the following:

1. X'00' - All Zero Status

   This status is returned to the channel when:

   a. the hardware has accepted a standard command.

   b. the channel command is Test I/O and the channel adapter is free of commands.

2. X'02' - Unit Check

   This status is returned to the channel when the NSC hardware detects an even parity on the Channel Bus Out for the command byte.

3. X'08' - Channel End

   This status is returned to the channel by the NSC hardware as an immediate Initial Status when the command is a control type command.

4. X'0C' - Channel End and Device End

   This status is returned to the channel by the NSC hardware as an immediate Initial Status to a No-Op command.

5. X'10' - Busy

   This status is returned to the channel by the NSC hardware when the NSC is already active with another command and has not yet presented a final status for that command.

6. X'70' - Status Modifier, Control Unit End, and Busy

   This status is returned to the channel when:

   a. the channel adapter has an Initial Selection Level 3 interrupt request pending. This is because the channel adapter has accepted a previous command and has not yet reset the interrupt request.

   b. the channel adapter has detected a System Reset and caused an Initial Selection Level 3 interrupt, but the control program has not yet reset the interrupt.

   c. the channel adapter has detected a Selective Reset during a service transfer sequence and caused a Data/Status Transfer Level 3 interrupt, but the control program has not yet reset the interrupt.

   d. a program requested interrupt is pending, but the control program has not yet reset the interrupt.

7. Any Pending Status without the Busy Bit

   This initial status is returned by the NSC hardware when a Test I/O command is issued to the NSC and a hardware generated status is pending.

8. Any Pending Status with the Busy Bit

   This initial status is returned by the NSC hardware when any command other than Test I/O is issued to the NSC and a hardware generated status is pending.

ESC Initial Status

At initial selection, the status returned to the channel may be one of the following:

1. X'00' - All Zero Status

   This status is returned to the channel when the channel adapter accepts an initial selection command byte other than I/O No-Op (X'03') or Test I/O (X'00'), and none of the Control Unit Busy (X'70') conditions (see below) are active.

2. X'02' - Unit Check

   This status is returned to the channel when the ESC hardware detects an even parity on the Channel Bus In for the command byte.

3. X'0C' - Channel End and Device End

   This status is returned to the channel by the ESC hardware as an immediate Initial Status to a No-Op command.

4. X'70' - Status Modifier, Control Unit End, and Busy

   This status is returned to the channel for a Test I/O command when:

   a. an Initial Selection level 3 interrupt request is pending.

   b. a Test I/O command has been issued to an ESC address, but ESC TIO status is not set for that address.

   c. a command is still in progress on another ESC address (ending status not yet accepted).

5. ESC Test I/O Pending Status

   This initial status is returned by the ESC hardware when a TIO command is issued to an ESC address and the ESC TIO Address/Status register (X'B') has been loaded with that subchannels address and status.

Stacked Initial Status

Some initial status responses to channel commands may be stacked by the host. When this happens, the channel adapter hardware causes a channel adapter Initial Selection Level 3 interrupt request.

The initial statuses listed below may be presented by the channel adapter hardware and could be stacked by the channel:

All Zero Initial Status (X'00')

This status is never stacked by the channel unless the command is Test I/O to the NSC. See below under the heading 'Any Initial Status on Test I/O (NSC)'.

Channel End/Device End Initial Status to I/O No-Op (X'0C')

The device address and command may be obtained by executing an Input X'1' instruction. When Input X'0' is executed, byte 0, bit 5 (Stacked Initial Status) will be active.

Unit Check Initial Status (X'02')

This is caused by a bad parity on Bus Out during command byte transfer. The device address may be obtained by executing an Input X'1' instruction. When Input X'0' is executed, byte 0, bit 3 (Channel Bus Out Check), and bit 5 (Stacked Initial Status) will be active.

Any Initial Status on Test I/O (NSC)

This is the only case in which an all zero status may have been stacked. The NSC address and command may be obtained by executing an Input X'1' instruction. When Input X'0' is executed, byte 0, bit 5 (Stacked Initial Status) will be active. When a Test I/O initial status is stacked for the NSC address, the control program should not execute an Output X'6' (NSC Status/Control Register) to put the stacked status in the NSC Status Register. This is because the NSC hardware saves the stacked (pending) status from a Test I/O command into the NSC Status register. The NSC status register contains X'00' if the channel adapter is free of commands or contains the pending or stacked status otherwise. The channel adapter hardware does not reset this register until the host channel has accepted it. The control program can present this status by executing an Output X'2' instruction, unless of course a TIO command retrieves it first.

Any Initial Status on Test I/O (ESC)

The ESC address and command may be obtained by executing an Input X'1' instruction. When Input X'0' is executed, byte 0, bit 5 (Stacked Initial Status) will be active. From this point onwards, the control program should treat this stacked Test I/O status as if it were a status that was stacked during an ESC final status transfer.

SECTION 5. TWO-PROCESSOR SWITCH FEATURE

Note for 3725 Model 2 only

The 3725 Model 2 cannot be equipped with a two-processor switch.

The two-processor switch feature (TPS) is an optional feature that allows the 3725 to be attached to a pair of symmetrical tightly-coupled multiprocessors. Additionally, this feature allows the 3725 to be attached to two different channel interfaces on a single host as an I/O device with alternate path capability. Each channel interface has its own NSC address. ESC operation is only allowed when the channel adapter is operating in a partitioned mode; i.e. only one of the two channel interfaces is enabled at any one time.

In addition to the usual features of a normal channel adapter, a channel adapter fitted with the two-processor switch feature adds the following capabilities:

- Each channel interface may be enabled independently.
- Both channel interfaces (A and B) may be enabled (on line) simultaneously. However, simultaneous operation over the two interfaces is not permitted.
- When both interfaces are enabled at the same time, contention problems are resolved automatically by the hardware.
- The interfaces may be enabled or disabled via the control program. If an interface is disabled, it bypasses 'Select Out'.
- The channel adapter hardware automatically provides 'allegiance' to a single channel interface for the entire duration of a channel I/O operation, that is to say, from initial selection and reception of the first command right up to the reception of a 'Device End' ending status for the last command that does not indicate command chaining.
- During the period when the allegiance of the channel adapter hardware is directed to one channel interface, any initial selection attempt by the other channel results in an 'Abbreviated Device Busy' (X'10') status to that channel.

  When the other channel has ended its I/O operation, the channel adapter hardware automatically presents a 'Device End' status to the other channel (the one that received the 'Abbreviated Device Busy' status).
- Either interface may be enabled or disabled from the control panel. If only one interface is enabled in this way at any one time, it is called the manual partitioning mode. Operation with both NSC and ESC addresses is permitted in this mode.

STATES OF A CHANNEL ADAPTER PLUS TPS

When both channel interfaces are enabled, the channel adapter is in one of two states:

Neutral State

In this state, the channel adapter is not switched to either interface, but is available to both.

Switched State

In this state, the channel adapter has allegiance to one of the interfaces; that is, an initial selection sequence from the host has switched the channel adapter to that interface. Channel commands can now only be accepted on this interface from the channel that sent the command.

Note, however, that the channel adapter continues to monitor the activity on the other interface, and responds to initial selection requests on this interface either by temporarily suspending the completion of the initial selection sequence, or by responding with the short 'Device Busy' status (X'10'). During this status sequence, the channel adapter remains connected to the other interface.

TYPES OF ALLEGIANCE

Allegiance to an interface may be either short- or long-term, and may change from short- to long-term during the operation. Short- and long-term allegiance is discussed in detail below under the heading 'Duration of Channel Interface Allegiance'.

Instantaneous Allegiance

Instantaneous allegiance is a short-term allegiance. The channel adapter enters this state when it traps 'Select Out' because of a poll to 'Request In' (from the neutral state), or because of an initial selection from the host. Instantaneous allegiance causes the channel adapter hardware to switch temporarily to that interface during the initial selection sequence (until Status In is raised). Instantaneous allegiance ends with the presentation of initial status by the channel adapter.

Implicit Allegiance

Implicit allegiance is a long-term allegiance, and covers the entire execution of an I/O operation. It starts when the channel adapter replies with an 'all zeros' status to an initial selection sequence with a channel command that requires information transfer to complete it:

- Data transfer command
- Ending status to complete a command
- A No-Op command with the command chaining bit on.

It ends when the channel accepts a 'Device End' status for the last command without command chaining indicated.

Any attempt by the channel attached to the opposite interface to select the channel adapter during this time is rejected with a 'Device Busy' status (X'10') during a short busy sequence on the channel interface.

Note: if the I/O operation was ended by an 'interface disconnect', the implicit allegiance lasts until either:

- The control program returns the channel adapter to the neutral state by executing an Output X'6' (NSC Status/Control Register) with byte 0, bit 7 (Reset to Neutral State) set to 1.
- The control program presents 'Device End' status to the channel.

Contingent Allegiance

Contingent allegiance is a type of long-term allegiance. It occurs only for a channel adapter that has ended a command sequence with a 'Unit Check' indication in the status byte. The intention is to ensure that exactly the same path may be used by the host to recover sense data from the 3725 after the host has received a 'Unit Check' status. While in the contingent allegiance state, any attempt at initial selection by the other interface receives a device busy status (X'10') in reply. Contingent allegiance ends when the control program decodes a channel command (other than Test I/O or No-Op).

DURATION OF CHANNEL INTERFACE ALLEGIANCE

When the channel adapter owes allegiance to one of the interfaces, this allegiance may be either short- or long-term. Both implicit and contingent allegiance are long-term allegiances. A short-term (or instantaneous) allegiance exists only for the duration of initial selection.

If, during the period of short-term allegiance on one interface, an initial selection attempt occurs on the other, the channel adapter logic detects the rise of the 'Select Out' tag, but then temporarily blocks the completion of the selection sequence. There are now two possibilities:

1. The first interface passes to the neutral state (this is the case, for instance, when an asynchronous status has been presented, and either accepted or stacked, or if a No-Op without chaining has been issued by the host. The second interface itself now enters the short-term allegiance state and continues its initial selection routine. The 'Device Busy' status is not issued.

2. The first interface passes from short- to long-term allegiance. This occurs if a command was accepted by the channel during the initial selection. The opposite interface now replies to its initial selection with the short 'Device Busy' (X'10') status.

STATUS PRESENTATION

In the discussion which follows, an untagged status is a status that is offered to both channel interfaces at the same time. A tagged status is a status that must be offered to a particular interface only.

Untagged Asynchronous Status Presentation

In a Tightly-Coupled Symmetric environment, if the control program wants to present an asynchronous status, it sets this status in the channel adapter logic by means of an Output X'6' (NSC Status/Control Register) instruction, and then starts the transfer by means of an Output X'2' (Data/Status Control Register) instruction, in the usual way.

If both interfaces are enabled, the hardware treats this status as an untagged status, and offers it to both channels by raising the 'Request In' tag to both interfaces. The first channel to poll (by raising the 'Select Out' tag in response to 'Select In') is connected to the channel adapter and receives the status. For the duration of this sequence (considered by the adapter as a short-term allegiance), any channel polls on the opposite interface are ignored by bypassing 'Select Out'. A channel initiated initial selection attempt on the other interface is temporarily suspended by trapping 'Select Out' as described above.

If the status just presented is accepted or stacked by the channel, the channel adapter sets a Data/Status Level 3 Interrupt and returns to the neutral state; any Control Unit initiated polls while the channel adapter is in this Data/Status Level 3 Interrupt state are bypassed on both interfaces. For a stacked status, when the control program re-initiates the status transfer via another Output X'2' instruction, the status is offered to both interfaces.

A Channel Initiated initial selection sequence on either interface causes the channel adapter to switch to the short-term allegiance state for that interface, present the status (accompanied by the 'Busy' bit if the command is anything other than Test I/O), cause an Initial Selection Level 3 Interrupt request, and return to the neutral state.

During this interrupt, any other Channel Initiated initial selection sequences to either interface cause the channel adapter to present Control Unit Busy (X'70') as initial status and return to the neutral state.

Tagged Status Presentation

When the channel adapter has presented a 'Busy' status (X'10') for a Channel Initiated initial selection routine on one interface while the other is in the long-term allegiance state, the channel adapter presents a tagged (over the same interface) asynchronous Device End status when the channel adapter returns to the neutral state (see however the note below) and none of the following conditions is present:

- An initial selection level 3 interrupt request is pending due to a normal initial selection, or to a system reset or selective reset during initial selection.
- A data/status level 3 interrupt request is pending due to a selective reset during data/status transfer.
- A program requested interrupt is pending.

The Device End status can only be presented when these pending interrupts interrupts have been reset.

Note: if a disable has been requested (an Output X'7', with byte 1, bit 7 = Set Allow Channel Interface Disable was executed, or the channel switch on the control panel was switched from the ON to the OFF position while the interface was in the long term allegiance state) before the channel adapter returns to the neutral state, then the tagged device end status is not sent.

During this time, if the opposite channel polls the adapter in response to a 'Request In', the resulting 'Select Out' tag is bypassed.

Similarly, if a Channel Initiated initial selection sequence occurs on the opposite channel and the other interface is in instantaneous allegiance, the sequence is temporarily suspended until the Device End status has been presented or stacked. The sequence is then completed at the end of the Device End presentation.

Note: if an asynchronous status is set up by the control program while a tagged Device End status is pending for either interface, this asynchronous status is presented along with the Device End for that interface.

EFFECT OF SYSTEM RESET

System Reset over Interface with Allegiance

When the channel adapter recognizes the system reset, it completely resets the adapter, ends the channel adapter allegiance, and sets an Initial Selection Level 3 interrupt request. However, if a Device End status caused by a previous Device Busy status over the opposite interface is pending, it is not reset.

During this time, if the opposite channel polls the adapter in response to a 'Request In' from some other control unit, the resulting 'Select Out' tag is bypassed.

Similarly, any Channel Initiated initial selection sequences to either interface cause the channel adapter to switch to that interface, present Control Unit Busy (X'70') as initial status, and return to the neutral state.

System Reset over Interface without Allegiance

When a system reset occurs for the interface that does not have allegiance, it resets the pending tagged Device End status (if any) for that interface. The remaining adapter hardware is not reset, and there is no Initial Selection Level 3 Interrupt request.

System Reset when Adapter is in Neutral State

When a system reset occurs for a channel adapter in the neutral state, it resets only a pending tagged Device End status (if any) for the interface over which the system reset was received. The remaining adapter hardware is not reset, and there is no Initial Selection Level 3 Interrupt request.

Note: if a system reset is presented to both interfaces simultaneously, it completely resets the adapter and sets an Initial Selection Level 3 interrupt request.

EFFECT OF SELECTIVE RESET

Selective Reset over Interface with Allegiance

When the channel adapter recognizes the selective reset, it returns to the neutral state and sets an Initial Selection Level 3 interrupt request. If the selective reset is received on an interface having a tagged Device End pending due to a previous presentation of Busy, this tagged Device End is reset.

During the Initial Selection Interrupt, if the opposite channel polls the adapter in response to a 'Request In' from some other control unit, the resulting 'Select Out' tag is bypassed.

Similarly, if a Channel Initiated initial selection sequence occurs to either interface, the channel adapter switches to that interface, enters the short-term allegiance state, presents Control Unit Busy (X'70') as initial status, and returns to the neutral state.

Selective Reset over Interface without Allegiance

A selective reset cannot occur for the interface that does not have allegiance; therefore, the channel adapter hardware is not reset, and there is no Initial Selection Level 3 Interrupt request.

SECTION 6. CHANNEL ADAPTER - SPECIAL TOPICS

BSC CONTROL CHARACTER RECOGNITION

The 3725 contains hardware circuits which search for the following characters or sequence of characters in the incoming data stream from the host:

1. End of Transmission Block (ETB)
2. End of Text (ETX)
3. Data Link Escape (DLE) followed by Start of Text (STX).

These hardware circuits may be activated by setting bit 2 (ASCII) or bit 3 (EBCDIC) in byte 0 of the Cycle Steal Mode Control register (X'C').

A further hardware circuit searches for SYN characters. It is activated by setting byte 0, bit 0 of the Cycle Steal Mode Control register (X'C').

The use of these circuits depends on whether the transmission is taking place in normal or in transparent mode.

Normal Text Mode Operation

The control program must select the monitoring required by setting the appropriate bit in the Cycle Steal Mode Control register via an Output X'C' instruction:

- Set byte 0, bit 2 on if ASCII monitoring is required
- Set byte 0, bit 3 on if EBCDIC monitoring is required
- Set both bits off if monitoring is not required

When monitoring for ASCII or EBCDIC control characters, if an ETB or ETX character is detected in the stream coming from the host channel, the transfer sequence is terminated after the ETB/ETX character has been transferred to the channel adapter. Channel stop is set in register Input X'2' by hardware.

Transparent Text Mode Operation

Suppose that the channel adapter is working in normal text mode, and has been set to monitor for ETB/ETX in either ASCII or EBCDIC. If ETB/ETX is detected, the transfer sequence is ended as described above.

At the same time, however, the channel adapter searches for the two-character sequence DLE/STX, which indicates the start of transparent text. When the sequence DLE/STX is detected, the ASCII/EBCDIC monitor control latch is reset, monitoring is stopped, and the transparent text mode is entered. The control program is informed of this event by the absence of the monitor bits in the Cycle Steal Mode Control register.

Note: it may happen that DLE is the last character of one host write operation and STX is the first character of the next. To ensure correct operation when this occurs, a special bit, the DLE Remember Latch (bit 1), is set in the Cycle Steal Mode Control register. The latch is set when a DLE character is detected, and reset when a non-DLE character is transferred. If the DLE character is the last character in a host write operation, when the program issues an Input X'C' instruction in response to the level 3 interrupt request (Channel End), it will find that the DLE Remember latch is on. The next time that a host write operation occurs for the same address, the control program must restore the DLE Remember bit via an Output X'C' instruction. If the first character of the host write operation is STX, the transparent text mode is entered.

Monitoring for SYN Characters

The 3725 also contains hardware that searches the incoming data stream from the host for SYN characters. This hardware is activated via one of the ASCII/EBCDIC Control Latches as described above; in addition, a special bit (bit 0) in the Cycle Steal Mode Control register called the SYN Monitor Control Latch must be set. If register X'C', byte 0, bit 2 is on, the circuits search for ASCII SYN characters; if byte 0, bit 3 is on, the hardware searches for EBCDIC SYN.

Each time that the hardware detects a SYN character, it sets Byte 0, bit 0 (SYN Monitor Latch) of Input register X'C'. The next non-SYN character resets it.

If four consecutive SYN characters are detected, the channel adapter disconnects from the channel, and raises a Data/Status level 3 interrupt. Byte 0, bit 0 of register X'C' is left in the on state, and can be read via an Input X'C' instruction.

270X EMULATION CONSIDERATIONS

When emulating a 270X on the 3725, attention must be paid to the following points:

2702/2703 Two-Channel Switch Support

The 3725 does not support the 2702/2703 automatic two-channel switch hardware feature. If the channel issues a 'Reserve' or a 'Release' command, the 3725 must reject it by returning an ending status of 'Channel End', 'Device End', and 'Unit Check', and by setting the 'Command Reject' bit in the sense byte.

Busy Response to Start I/O and Test I/O

Under normal operating conditions, both the 2702 and 2703, being multiple subchannel adapters, can present a control unit busy condition to a Start I/O or a Test I/O instruction. The reason for this busy condition is that the control unit must store a received command before another operation can be initiated. This busy condition lasts for a maximum of 1 millisecond for the 2702, and 90 microseconds for the 2703.

The 3725 being also a multiple subchannel adapter (in 270X emulation mode) can also present a control unit busy condition since it cannot accept a new command until the previous command has been serviced by the control program. In general, if a Start I/O or a Test I/O terminates with a condition code of 1 and a Control Unit Busy status, the instruction must be reissued until the busy condition ends.

APPENDIX A. CCU EXTERNAL REGISTERS

INPUT/OUTPUT X'00' THROUGH X'27' (GENERAL REGISTERS)

The bit assignments of these registers are, in general, not fixed, but vary with the use of the register. There is however one exception: the first register of each group always contains the address of the next sequential instruction in that interrupt level.

INPUT/OUTPUT X'28' THROUGH X'2F' (RESERVED)

The 8 registers addressed by these instructions are reserved.

INPUT/OUTPUT X'30' THROUGH X'35' (CYCLE STEAL ADDRESS REGISTERS)

Register	Channel adapter
X'30'	1
X'31'	2
X'32'	3
X'33'	4
X'34'	5
X'35'	6

INPUT/OUTPUT X'36' THROUGH X'3E' (POINTER REGISTERS)

INPUT/OUTPUT X'3F' (COMMUNICATION SCANNER CS ADDRESS)

INPUT/OUTPUT X'40' THROUGH X'43' (INTERRUPT START ADDRESS)

INPUT/OUTPUT X'44' (BYTE OPERATIONS BASE REGISTER)

INPUT/OUTPUT X'45' (HALFWORD OPERATIONS BASE REGISTER)

INPUT/OUTPUT X'46' (FULLWORD OPERATIONS BASE REGISTER)

INPUT/OUTPUT X'48' (IOH ADDRESS SUBSTITUTION REGISTER)

INPUT/OUTPUT X'49' THROUGH X'4F' (RESERVED)

INPUT/OUTPUT X'50' THROUGH X'5F' (PROGRAMMABLE REGISTERS)

INPUT/OUTPUT X'60' THROUGH X'67' (RESERVED)

INPUT X'68' (ZERO REGISTER)

INPUT/OUTPUT X'69' THROUGH X'6F' (RESERVED)

INPUT X'70' (STORAGE SIZE INSTALLED)

Byte	Bit	Meaning
0	0	0
	1	1
	2	0
	3	2048 K
	4	1024 K
	5	512 K
	6	256 K
	7	Storage not a multiple of 256K
1	0-7	0

OUTPUT X'70' (HARDSTOP)

INPUT X'71' (OPERATOR ADDRESS/DATA ENTRY REGISTER)

Byte	Bit	Meaning
X	2-7	Operator address/data register byte X, bits 2-7
0	0-7	Operator address/data register byte 0, bits 0-7
1	0-7	Operator address/data register byte 1, bits 0-7

OUTPUT X'71' (DISPLAY REGISTER 1)

Byte	Bit	Meaning
X	2-7	Display register 1 byte X, bits 2-7
0	0-7	Display register 1 byte 0, bits 0-7
1	0-7	Display register 1 byte 1, bits 0-7

INPUT X'72' (OPERATOR DISPLAY/FUNCTION SELECT CONTROL)

Byte	Bit	Meaning
0	0	Function select 8
	1	Function select 9
	2	Function select 10
	3	Function select 11 (storage address)
	4	Function select 12 (register address)
	5	Function select 13
	6	Function select 14
	7	Function select 15
1	0	Function select 16
	1	Function select 1
	2	Function select 2
	3	Function select 3
	4	Function select 4
	5	Function select 5
	6	Function select 6
	7	Function select 7

OUTPUT X'72' (DISPLAY REGISTER 2)

Byte	Bit	Meaning
X	2-7	Display register 2 byte X, bits 2-7
0	0-7	Display register 2 byte 0, bits 0-7
1	0-7	Display register 2 byte 1, bits 0-7

INPUT X'73' (INSERT STORAGE PROTECT/ADDRESS EXCEPTION KEY)

Byte	Bit	Meaning
0	0-7	(not used)
1	0	(not used)
	1	(not used)
	2	(not used)
	3	(not used)
	4	(not used)
	5	Key Bit 0
	6	Key Bit 1
	7	Key Bit 2

OUTPUT X'73' (SET STORAGE PROTECT/ADDRESS EXCEPTION KEY)

Byte	Bit	Meaning
X	2	Storage key address bit 0
	3	Storage key address bit 1
	4	Storage key address bit 2
	5	Storage key address bit 3
	6	Storage key address bit 4
	7	Storage key address bit 5
0	0	Storage key address bit 6
	1	Storage key address bit 7
	2	Storage key address bit 8
	3	Storage key address bit 9/user key address bit 0
	4	Storage key address bit 10/user key address bit 1
	5	User key address bit 2
	6	User key address bit 3
	7	User key address bit 4
1	0	(not used)
	1	Enable storage protect/address exception
	2	Key type bit 0
	3	Key type bit 1
	4	Modify key value
	5	Key bit 0
	6	Key bit 1
	7	Key bit 2

INPUT X'74' (LAGGING ADDRESS REGISTER)

INPUT X'75' (CCW FOR AIO OPERATIONS)

Byte	Bit	Meaning
0	0	CCW Bit 5 (0 = CA AIO, 1 = scanner AIO)
	1	CCW Bit 11 (pointer no./scanner address bit 0)
	2	CCW Bit 12 (pointer no./scanner address bit 1)
	3	CCW Bit 13 (pointer no./scanner address bit 2)
	4	CCW Bit 14 (pointer no./scanner address bit 3)
	5	(not used)
	6	(not used)
	7	(not used)
1	0-7	(not used)

INPUT X'76' (ADAPTER LEVEL 1 INTERRUPT REQUESTS)

Byte	Bit	Meaning
0	0	Addressing exception during I/O operations
	1	Storage protection check during I/O operations
	2	Invalid CCW during I/O operations
	3	(not used)
	4	Time out condition
	5	Bus in parity check
	6	Adapter initiated operation
	7	MOSS initiated operation
1	0-7	(not used)

OUTPUT X'76' (MISCELLANEOUS CONTROL 1)

Byte	Bit	Meaning
0	0	Reset errors detected during I/O operations
	1	(not used)
	2	(not used)
	3	Control program to MOSS request
	4	Control program to MOSS response
	5	(not used)
	6	(not used)
	7	(not used)
1	0-7	(not used)

INPUT X'77' (ADAPTER LEVELS 2 AND 3 INTERRUPT REQUESTS)

Byte	Bit	Meaning
0	0	(not used)
	1	Scanner level 2 interrupt
	2	(not used)
	3	(not used)
	4	(not used)
	5	(not used)
	6	(not used)
	7	(not used)
1	0	Level 3 channel adapter interrupt
	1	(not used)
	2	(not used)
	3	(not used)
	4	(not used)
	5	(not used)
	6	(not used)
	7	(not used)

OUTPUT X'77' (MISCELLANEOUS CONTROL 2)

Byte	Bit	Meaning
0	0	Reset IPL level 1 interrupt
	1	Reset CCU hardware checks
	2	Reset MOSS panel interrupt request level 3
	3	Reset MOSS diagnostic interrupt request level 3
	4	Reset MOSS service interrupt request level 4
	5	Reset MOSS service interrupt response level 4
	6	(not used)
	7	Reset program controlled interrupt level 2
1	0	Reset MOSS inoperative level 1 interrupt
	1	Reset interval timer level 3 interrupt
	2	Reset program controlled interrupt level 3
	3	Reset MOSS diagnostic interrupt request level 2
	4	Reset address compare level 1 interrupt
	5	Reset software checks
	6	Reset program controlled interrupt level 4
	7	Reset supervisor call level 4 interrupt

OUTPUT X'78' (FORCE ALU CHECKS)

INPUT X'79' (UTILITY)

Byte	Bit	Meaning
0	0	(not used)
	1	Probe condition satisfied
	2	Probe address compare received
	3	(not used)
	4	(not used)
	5	(not used)
	6	Program level 5 C latch
	7	Program level 5 Z latch
1	0	Program level 2 interrupted by level 1
	1	Program level 3 interrupted by level 1
	2	Program level 4 interrupted by level 1
	3	Program level 5 interrupted by level 1
	4	(not used)
	5	(not used)
	6	(not used)
	7	(not used)

OUTPUT X'79' (UTILITY)

Byte	Bit	Meaning
0	0	(not used)
	1	(not used)
	2	Set programmed IPL request
	3	(not used)
	4	Remote power off
	5	Inhibit program level 5C, 5Z replace
	6	Set program level 5 C latch
	7	Set program level 5 Z latch
1	0	(not used)
	1	(not used)
	2	Set AIO stop mode
	3	Reset AIO stop mode
	4	Set bypass CCU check stop mode
	5	Reset bypass CCU check stop mode
	6	Scope sync pulse 1
	7	Scope sync pulse 2

INPUT X'7A' (HIGH RESOLUTION TIMER/UTILIZATION COUNTER)

Byte	Bit	Meaning
X	2	Timer Bit 0
	3	Timer Bit 1
	4	Timer Bit 2
	5	Timer Bit 3
	6	Timer Bit 4
	7	Timer Bit 5
0	0	Timer Bit 6
	1	Timer Bit 7
	2	Timer Bit 8
	3	Timer Bit 9
	4	Timer Bit 10
	5	Timer Bit 11
	6	Timer Bit 12
	7	Timer Bit 13
1	0	Timer Bit 14
	1	Timer Bit 15
	2	Timer Bit 16
	3	Timer Bit 17
	4	Timer Bit 18
	5	Timer Bit 19
	6	Timer Bit 20
	7	Timer Bit 21

OUTPUT X'7A' (HIGH RESOLUTION TIMER/UTILIZATION COUNTER CONTROL)

Byte	Bit	Meaning
0	0	Timer/counter (1 = reset timer/enable count)
	1	High/low resolution (1 = low resolution)
	2	Timer/utilization counter (0 = timer)
	3	(not used)
	4	(not used)
	5	(not used)
	6	(not used)
	7	(not used)
1	0-7	(not used)

INPUT X'7B' (BRANCH TRACE ADDRESS POINTER)

Byte	Bit	Meaning
X	2-6	Branch trace address pointer byte X, bits 2-7
0	0-7	Branch trace address pointer byte 0, bits 0-7
1	0-7	Branch trace address pointer byte 1, bits 0-7

OUTPUT X'7B' (SET PCI LEVEL 2)

INPUT X'7C' (BRANCH TRACE BUFFER COUNT)

Byte	Bit	Meaning
0	0	Branch trace buffer count bit 0
	1	Branch trace buffer count bit 1
	2	Branch trace buffer count bit 2
	3	Branch trace buffer count bit 3
	4	Branch trace buffer count bit 4
	5	Branch trace buffer count bit 5
	6	Branch trace buffer count bit 6
	7	Branch trace buffer count bit 7
1	0	Branch trace buffer count bit 8
	1	Branch trace buffer count bit 9
	2	Branch trace buffer count bit 10
	3	Branch trace buffer count bit 11
	4	Branch trace buffer count bit 12
	5	(not used)
	6	(not used)
	7	(not used)

OUTPUT X'7C' (SET PCI LEVEL 3)

INPUT X'7D' (CCU HARDWARE CHECK REGISTER)

OUTPUT X'7D' (SET PCI LEVEL 4)

INPUT X'7E' (CCU LEVEL 1 INTERRUPT REQUESTS)

Byte	Bit	Meaning
0	0	MOSS inoperative
	1	CCU hardware error summary
	2	(not used)
	3	Level 5 I/O error
	4	Invalid operation
	5	Adapter level 1 interrupt request
	6	(not used)
	7	CCU level 1 interrupts during I/O summary
1	0	Address compare level 1 interrupt
	1	Addressing exception on instruction fetch
	2	Storage protect exception on instruction fetch
	3	Addressing exception on program execution
	4	Storage protect exception on program execution
	5	(not used)
	6	IPL level 1 interrupt
	7	(not used)

OUTPUT X'7E' (SET PROGRAM INTERRUPT MASK BITS)

Byte	Bit	Meaning
0	0-7	(not used)
1	0	(not used)
	1	Mask adapter program level 1 requests
	2	Mask program level 2 requests
	3	Mask program level 3 requests
	4	Mask program level 4 requests
	5	Mask program level 5 execution
	6	(not used)
	7	(not used)

INPUT X'7F' (CCU L2, 3, OR 4 INTERRUPT REQUESTS)

Byte	Bit	Meaning
0	0	Program controlled interrupt (PCI) level 2
	1	MOSS diagnostic interrupt request level 2
	2	MOSS diagnostic interrupt request level 3
	3	MOSS service interrupt request level 4
	4	MOSS service interrupt response level 4
	5	(not used)
	6	CE/operator interrupt request level 3
	7	Program controlled interrupt (PCI) level 4
1	0	(not used)
	1	(not used)
	2	(not used)
	3	(not used)
	4	(not used)
	5	Interval timer interrupt request level 3
	6	Program controlled interrupt (PCI) level 3
	7	Supervisor call level 4

OUTPUT X'7F' (RESET PROGRAM INTERRUPT MASK BITS)

Byte	Bit	Meaning
0	0-7	(not used)
1	0	(not used)
	1	Unmask adapter program level 1 requests
	2	Unmask program level 2 requests
	3	Unmask program level 3 requests
	4	Unmask program level 4 requests
	5	Unmask program level 5 execution
	6	(not used)
	7	(not used)

APPENDIX B. CA INPUT/OUTPUT INSTRUCTION SUMMARY CHARTS

HARDWARE STATUS BYTE REGISTER

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

INPUT X'0' (INITIAL SELECTION CONTROL REGISTER)

Byte	Bit	Meaning
0	0	Initial selection interrupt
	1	Interface disconnect
	2	Selective reset
	3	Channel bus out check
	4	Emulation subchannel operation
	5	Stacked initial status
	6	Status byte cleared
	7	System reset
1	0-7	(not used)

OUTPUT X'0' (RESET INITIAL SELECTION)

INPUT X'1' (INITIAL SELECTION ADDRESS AND COMMAND REGISTER)

Byte	Bit	Meaning
0	0-7	Address byte bit 0-7 (initial selection address)
1	0-7	I/O cmnd byte bit 0-7 (initial selection cmnd)

OUTPUT X'1' (INITIAL SELECTION ADDRESS AND COMMAND REGISTER)

Byte	Bit	Meaning
0	0-7	Address byte bit 0-7 (initial selection address)
1	0-7	I/O cmnd byte bit 0-7 (initial selection cmnd)

INPUT X'2' (DATA/STATUS CONTROL REGISTER)

Byte	Bit	Meaning
0	0	Outbound data transfer sequence
	1	Inbound data transfer sequence
	2	Status transfer sequence
	3	NCP subchannel if 0; EP subchannel if 1
	4	Channel end presented
	5	Channel stop/interface disconnect
	6	Suppress out monitor interrupt
	7	Program requested interrupt
1	0	Channel bus out check
	1	Selective reset
	2	Suppress out
	3	Stacked ending status
	4	Priority outbound service
	5	Residual byte count bit 5
	6	Residual byte count bit 6
	7	Residual byte count bit 7

OUTPUT X'2' (DATA/STATUS CONTROL REGISTER)

Byte	Bit	Meaning
0	0	Set/reset outbound data transfer sequence (Note)
	1	Set/reset inbound data transfer sequence (Note)
	2	Set/reset status transfer sequence (Note)
	3	Set/reset ESC operation (Note)
	4	Set/reset PIO mode (Note)
	5	Reset initial selection interrupt
	6	Reset data/status interrupt
	7	(not used)
1	0	Set monitor for circle B
	1	(not used)
	2	Set monitor for 2848 ETX
	3	Set suppressible status
	4	Priority outbound service
	5	Request byte count bit 5
	6	Request byte count bit 6
	7	Request byte count bit 7

Note: Set = 1; reset = 0.

INPUT/OUTPUT X'3' (ESC ADDRESS AND STATUS BYTE REGISTER)

Byte	Bit	Meaning
0	0-7	Address byte bits 0-7 (data/status transfer)
1	0	ESC status byte bit 0 (attention)
	1	ESC status byte bit 1 (status modifier)
	2	ESC status byte bit 2 (control unit end)
	3	ESC status byte bit 3 (Busy)
	4	ESC status byte bit 4 (channel end)
	5	ESC status byte bit 5 (device end)
	6	ESC status byte bit 6 (unit check)
	7	ESC status byte bit 7 (unit exception)

INPUT/OUTPUT X'4' AND X'5' (DATA BUFFER REGISTERS)

Register X'4' (Data Buffer Bytes 1 and 2 or 5 and 6)

Byte	Bit	Meaning
0	0-7	Data buffer byte 1 or 5 bits 0-7
1	0-7	Data buffer byte 2 or 6 bits 0-7

Register X'5' (Data Buffer Bytes 3 and 4 or 7 and 8)

Byte	Bit	Meaning
0	0-7	Data buffer byte 3 or 7 bits 0-7
1	0-7	Data buffer byte 4 or 8 bits 0-7

INPUT X'6' (NSC STATUS/CONTROL REGISTER)

Byte	Bit	Meaning
0	0	Channel adapter switched to interface B
	1	Channel adapter switched to interface A
	2	(not used)
	3	(not used)
	4	(not used)
	5	(not used)
	6	(not used)
	7	(not used)
1	0	NSC status byte bit 0 (attention)
	1	NSC status byte bit 1 (status modifier)
	2	NSC status byte bit 2 (control unit end)
	3	NSC status byte bit 3 (Busy)
	4	NSC status byte bit 4 (channel end)
	5	NSC status byte bit 5 (device end)
	6	NSC status byte bit 6 (unit check)
	7	NSC status byte bit 7 (unit exception)

OUTPUT X'6' (NSC STATUS/CONTROL REGISTER)

Byte	Bit	Meaning
0	0	Set force A busy
	1	Set force B busy
	2	Force error
	3	Diagnostic storage mode (set = 1; reset = 0)
	4	(not used)
	5	Check the checkers
	6	A/B data buffer diagnostic mode
	7	Reset to neutral state
1	0	Set NSC status byte bit 0 (attention)
	1	Set NSC status byte bit 1 (status modifier)
	2	Set NSC status byte bit 2 (control unit end)
	3	Set NSC status byte bit 3 (busy)
	4	Set NSC status byte bit 4 (channel end)
	5	Set NSC status byte bit 5 (device end)
	6	Set NSC status byte bit 6 (unit check)
	7	Set NSC status byte bit 7 (unit exception)

INPUT X'7' (CHANNEL ADAPTER CONDITION REGISTER)

Byte	Bit	Meaning
0	0	CA5 enabled
	1	(not used)
	2	CA6 enabled
	3	(not used)
	4	(not used)
	5	NSC address active
	6	PIO mode
	7	(not used)
1	0	CA1 interface A enabled
	1	CA1 interface B enabled
	2	CA2 interface A enabled
	3	CA2 interface B enabled
	4	CA3 interface A enabled
	5	CA3 interface B enabled
	6	CA4 interface A enabled
	7	CA4 interface B enabled

OUTPUT X'7' (CHANNEL ADAPTER CONTROL REGISTER)

Byte	Bit	Meaning
0	0	Enable auto-select) cannot be on at
	1	Disable auto-select) the same time
	2	Select CA addressed by bits 4-6
	3	Execute output on CA addressed by bits 4-6
	4	Channel address bit 0)
	5	Channel address bit 1) CA address 1-6
	6	Channel address bit 2)
	7	Channel adapter reset
1	0	Set suppress out monitor
	1	Set program requested interrupt
	2	Reset channel adapter interrupt level 1 checks
	3	Reset system reset/NSC address active
	4	Set allow channel interface enable (A and B)
	5	Set ESC operational
	6	Set ESC command free
	7	Set allow channel interface disable (A and B)

Note: bits 4 and 7 cannot be active at the same time.

INPUT/OUTPUT X'B' (ESC TEST I/O ADDRESS AND STATUS REGISTER)

Byte	Bit	Meaning
0	0-7	ESC TIO address byte bits 0-7
1	0	ESC TIO status byte bit 0 (attention)
	1	ESC TIO status byte bit 1 (status modifier)
	2	ESC TIO status byte bit 2 (control unit end)
	3	ESC TIO status byte bit 3 (Busy)
	4	ESC TIO status byte bit 4 (channel end)
	5	ESC TIO status byte bit 5 (device end)
	6	ESC TIO status byte bit 6 (unit check)
	7	ESC TIO status byte bit 7 (unit exception)

INPUT X'C' (CYCLE STEAL MODE CONTROL REGISTER)

Byte	Bit	Meaning
0	0	SYN monitor latch
	1	DLE temporary latch
	2	USASCII monitor control latch
	3	EBCDIC monitor control latch
	4	(not used)
	5	(not used)
	6	(not used)
	7	(not used)
1	0-7	Residual byte count bits 0-7

OUTPUT X'C' (CYCLE STEAL MODE CONTROL REGISTER)

Byte	Bit	Meaning
0	0	SYN monitor control latch (Note)
	1	DLE remember control latch (Note)
	2	USASCII monitor control latch (Note)
	3	EBCDIC monitor control latch (Note)
	4	(not used)
	5	(not used)
	6	(not used)
	7	(not used)
1	0-7	Residual byte count bits 0-7

Note: 1 = set; 0 = reset

INPUT X'D' (CHANNEL ADAPTER LEVEL 1 INTERRUPT CHECK REGISTER)

Byte	Bit	Meaning
0	0	PIO bus parity error
	1	Internal bus parity error
	2	CCU interconnect card check
	3	(not used)
	4	Channel interface card check
	5	Address compare error
	6	Initiate service latch ungated
	7	(not used)
1	0	Output exception check
	1	PIO halt remember latch
	2	Cycle steal halt remember latch
	3	Bus in check interface A
	4	Ground fault error
	5	Bus in check interface B
	6	Driver/receiver card check I/F A
	7	Driver/receiver card check I/F B

INPUT X'E' (CHANNEL ADAPTER LEVEL 1 INTERRUPT REQUESTS)

Byte	Bit	Meaning
0	0	Channel adapter 5 level 1 interrupt request
	1	(not used)
	2	Channel adapter 6 level 1 interrupt request
	3	Channel adapter (any) level 1 interrupt request
	4	Channel address bit 0)
	5	Channel address bit 1) CA address 1-6
	6	Channel address bit 2)
	7	(not used)
1	0	Channel adapter 1 level 1 interrupt request
	1	(not used)
	2	Channel adapter 2 level 1 interrupt request
	3	(not used)
	4	Channel adapter 3 level 1 interrupt request
	5	(not used)
	6	Channel adapter 4 level 1 interrupt request
	7	(not used)

INPUT X'F' (CHANNEL ADAPTER LEVEL 3 INTERRUPT REQUESTS)

Byte	Bit	Meaning
0	0	(not used)
	1	Two processor switch installed
	2	Selected CA initial selection L3 request
	3	Selected CA data/status L3 request pending
	4	Channel address bit 0)
	5	Channel address bit 1) CA address 1-6
	6	Channel address bit 2)
	7	(not used)
1	0-7	(not used)

APPENDIX C. COMMUNICATION SCANNER COMMANDS

Grouped by Function

Common Commands.

Command	Hex
Set Mode	X'01'
Enable	X'02'
Disable	X'03'
Monitor Incoming Call	X'04'
Dial	X'05'
Change	X'06'
Wrap	X'07'
Raise DTR	X'08'
Flush Data	X'09'
Reset-D	X'0B'
Reset-N	X'0C'
Halt	X'F0'
Halt Immediate	X'F1'

NCP Commands:

Command	Hex
SDLC Transmit Control	X'10'
SDLC Transmit Data	X'11'
SDLC Transmit Continue	X'1D'
SDLC Receive Monitor	X'12'
SDLC Receive	X'13'
SDLC Receive Continue	X'14'
X.21 Call Request	X'15'
X.21 Monitor Incoming Call	X'16'
X.21 Clear Request	X'17'
NCP BSC Control	X'18'
NCP BSC Transmit	X'19'
NCP BSC Transmit Continue	X'1A'
NCP BSC Receive	X'1B'
NCP BSC Receive Continue	X'1C'

EP Commands:

Command	Hex
EP BSC Transmit Initial	X'20'
EP BSC Transmit SYN	X'21'
EP BSC Transmit Data	X'22'
EP BSC Poll	X'23'
EP BSC Receive	X'24'
EP BSC Receive Continue	X'25'
EP BSC Prepare	X'26'
EP BSC Monitor for Phase	X'27'
EP BSC Address Prepare	X'28'
EP BSC Search	X'29'

Character Mode Command:

Command	Hex
Write ICW	X'40'
Start/Stop Transfer	X'41'
Read ICW	X'F3'

Miscellaneous Commands:

Command	Hex
IBM 326X Test	X'2B'
Trace	X'2C'
Stop Trace	X'2D'
Line wrap	X'2E'

COMMANDS IN NUMERICAL ORDER

Hex	Command
X'01'	Set Mode
X'02'	Enable
X'03'	Disable
X'04'	Monitor Incoming Call
X'05'	Dial
X'06'	Change
X'07'	Wrap
X'08'	Raise DTR
X'09'	Flush Data
X'0B'	Reset-D
X'0C'	Reset-N
X'10'	SDLC Transmit Control
X'11'	SDLC Transmit Data
X'12'	SDLC Receive Monitor
X'13'	SDLC Receive
X'14'	SDLC Receive Continue
X'15'	X.21 Call Request
X'16'	X.21 Monitor Incoming Call
X'17'	X.21 Clear Request
X'18'	NCP BSC Control
X'19'	NCP BSC Transmit
X'1A'	NCP BSC Transmit Continue
X'1B'	NCP BSC Receive
X'1C'	NCP BSC Receive Continue
X'1D'	SDLC Transmit Continue
X'20'	EP BSC Transmit Initial
X'21'	EP BSC Transmit SYN
X'22'	EP BSC Transmit Data
X'23'	EP BSC Poll
X'24'	EP BSC Receive
X'25'	EP BSC Receive Continue
X'26'	EP BSC Prepare
X'27'	EP BSC Monitor for Phase
X'28'	EP BSC Address Prepare
X'29'	EP BSC Search
X'2B'	IBM 326X Test
X'2C'	Trace
X'2D'	Stop Trace
X'2E'	Line Wrap
X'40'	Write ICW
X'41'	Start/Stop Transfer
X'F0'	Halt
X'F1'	Halt Immediate
X'F3'	Read ICW

APPENDIX D. MOSS COMMANDS

MAILBOX OUT COMMANDS

Mailbox In commands always have the high order command bit off. The hexadecimal values of these commands are therefore always in the range X'0x' through X'7x'.

Command	Hex
Transfer Path Information Unit Out Command (SNA Only)	X'06'
Box Error Records Command	X'07'
Buffers Now Available Command	X'08'
Wrap Test Results Command	X'09'
Time/Date Valid Command	X'0C'
Control Program Parameters Command	X'23'
Request Hardware Configuration Data File Command	X'24'
Control Program Initialization Complete Command	X'25'
Control Program Loaded Command	X'41'
Roll In Saved Storage For Dump Command	X'42'

MAILBOX IN COMMANDS (MOSS TO CCU)

Mailbox In commands always have the high order command bit on. The hexadecimal values of these commands are therefore always in the range X'8x' through X'Fx'.

Command	Hex
Transfer Path Information Unit In Command (SNA Only)	X'86'
Wrap Test Request Command	X'89'
Connect Scanner Command	X'8D'
Request Buffer Command	X'8E'
Free Buffer Command	X'8F'
MOSS Offline Command	X'90'
MOSS Online Command	X'91'
Control Program Parameters Saved Command	X'A3'
Configuration Data File Information Available Command	X'A4'
Scanner IML Complete to Load/Dump Command	X'C1'
Roll In Complete to Load/Dump Command	X'C2'

APPENDIX E. REDRIVE LOGIC

The redrive logic is logically situated between the internal logic and the channel adapters or communication scanners. The individual redrives are addressed by the Primary Redrive Address (PRA) and Secondary Redrive Address (SRA) fields contained in register R2 of an IOH instruction, or in the second halfword of an IOHI instruction. The chart below and figure E1 define the adapter associated with each redrive and the PRA/SRA bit structure required to address the redrives.

All configurations except Model 2

PRA (Byte 0 bits)			SRA (Byte 1 bits)			
5	6	7	4	5	6	Controlled Adapters
0	0	0	0	0	0	Channel Adapter 1 and Line Attachment Base 1
0	0	0	0	0	1	Channel Adapter 2 and Line Attachment Base 2
0	0	0	0	1	0	Line Attachment Base 3
0	0	0	0	1	1	Channel Adapters 3, 4, 5, and 6
0	0	1	0	0	0	Frame Redrive for LABs 4 through 8
0	0	1	0	1	1	Line Attachment Base 4
0	0	1	1	0	0	Line Attachment Base 5
0	0	1	1	0	1	Line Attachment Base 6
0	0	1	1	1	0	Line Attachment Base 7
0	0	1	1	1	1	Line Attachment Base 8

where:

PRA = primary redrive address

SRA = secondary redrive address

Model 2 only

PRA (Byte 0 bits)			SRA (Byte 1 bits)			
5	6	7	4	5	6	Controlled Adapters
0	0	0	0	0	0	Channel Adapters 1/2 and Line Attachment Base

where:

PRA = primary redrive address

SRA = secondary redrive address

The following diagram should make this clear:

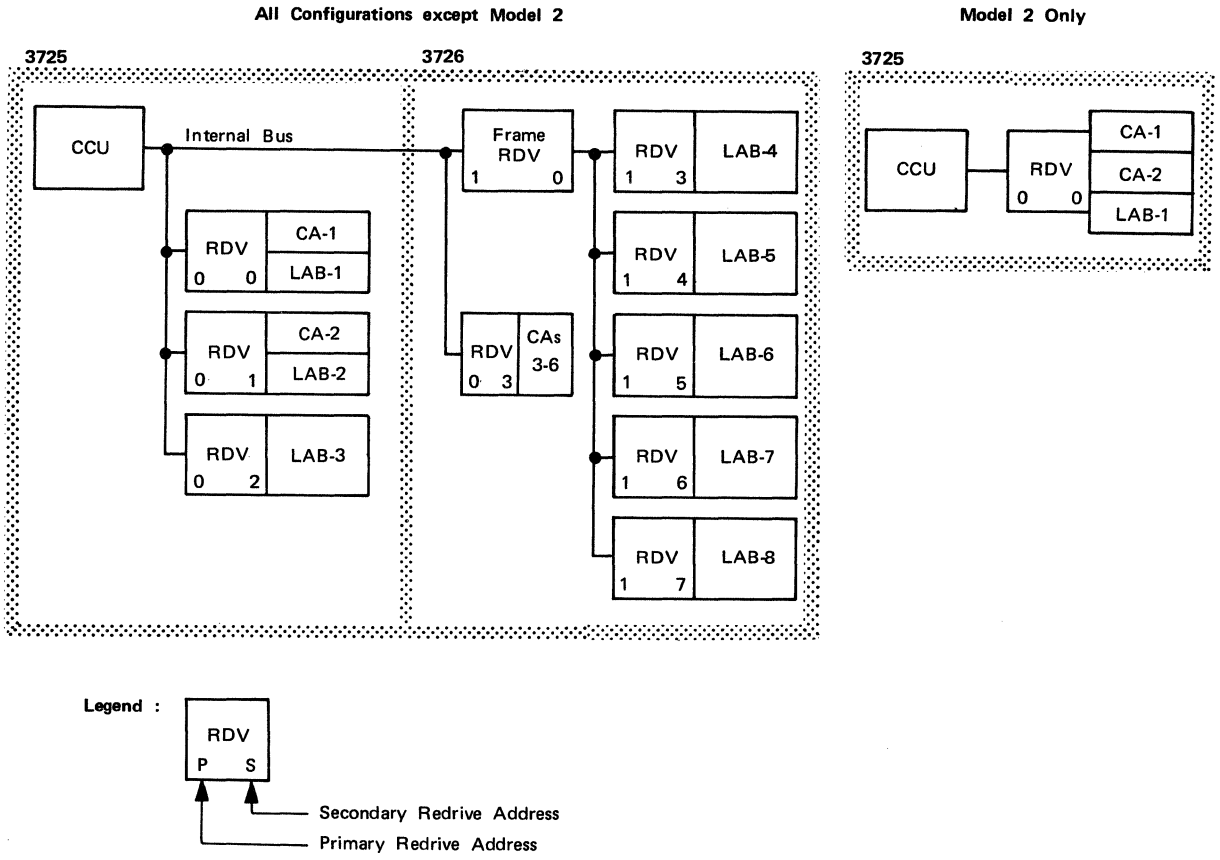


Figure E-1. Redrive Logic

REDRIVE IOH/IOHI INSTRUCTIONS

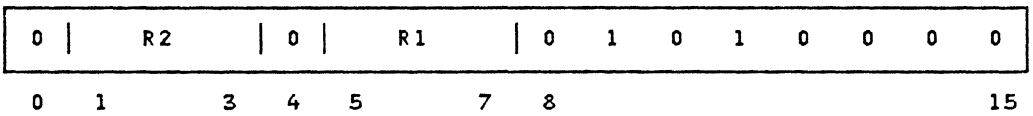
The redrive IOH/IOHI instructions are used to transfer the contents of one of the general registers to a selected redrive register or vice versa. There are two types of redrive input/output instruction:

- Adapter Input/Output (IOH)
- Adapter Input/Output Immediate (IOHI)

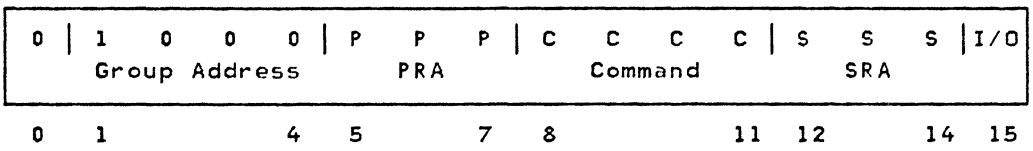
They are used as follows:

ADAPTER INPUT/OUTPUT (IOH)

This instruction transfers the contents of the register specified by R1 to the redrive logic, or places information coming from the redrive logic into the register specified by R1. The redrive address, the command, and the direction of data movement are all specified by the contents of R2.



R2 must be loaded as follows:



Bits 1 through 4 contain the group address (always 1 0 0 0)

Bits 5 through 7 contain the primary redrive address (PRA)

Bits 8 through 11 indicate the redrive command

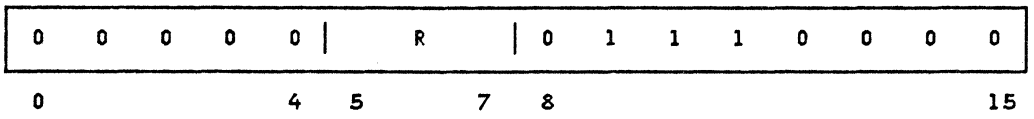
Bits 12 through 14 contain the secondary redrive address (SRA)

I/O = input/output bit: 0 = output, 1 = input

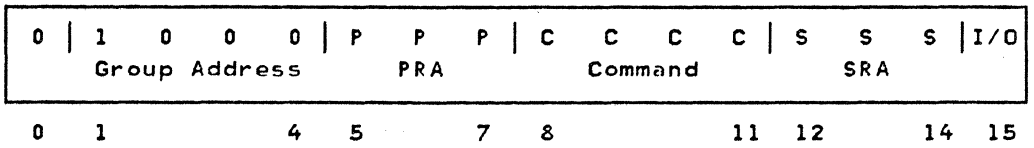
ADAPTER INPUT/OUTPUT IMMEDIATE

This instruction transfers the contents of the register specified by R to the redrive logic, or places information coming from the redrive logic into the register specified by R. The redrive address, the command, and the direction of data movement are all specified by the contents of the second halfword.

First halfword



Second halfword



Bits 1 through 4 contain the group address (always 1 0 0 0)

Bits 4 through 7 contain the primary redrive address (PRA)

Bits 8 through 11 indicate the redrive command

Bits 12 through 14 contain the secondary redrive address (SRA)

I/O = input/output bit: 0 = output, 1 = input

REDRIVE COMMANDS - DETAILED BIT STRUCTURE

The high order bit of the redrive command indicates whether the command is broadcast (high order bit = 0) or addressed to a particular redrive address.

COMMAND INPUT X'0' (POLL)

The high order bit of the command is always off, indicating a broadcast command. The register addressed by the instruction is loaded from the redrive logic and contains the redrive address and the event that set the redrive level 1 interrupt. If more than one redrive has a level 1 interrupt pending, multiple Input X'0's must be executed, since the redrive logically closer to the CCU on the internal bus presents its poll data first. The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	1
	1	Enable/disable (enable = 0, disable = 1)
	2	PRA Bit 4
	3	PRA Bit 2
	4	PRA Bit 1
	5	SRA Bit 4
	6	SRA Bit 2
	7	SRA Bit 1
1	0	In Bus Parity Error
	1	Out Bus Parity Error
	2	Out Bus Tag Check
	3	In Bus Tag Check
	4	Halt Remember
	5	Secondary Select Out
	6	Secondary Cycle Steal Grant
	7	Command Reject

The bits of byte 0 have the following meaning:

Byte 0, Bit 1 - Enable/disable: this bit, if off, indicates that the redrive address in byte 0, bits 2 through 7 is enabled; if on, it indicates that the redrive address is disabled.

Byte 0, Bits 2 through 4 - Primary Redrive Address (PRA).

Byte 0, Bits 5 through 7 - Secondary Redrive Address (SRA).

Redrive Error Register: Byte 1 is collectively called the Redrive Error Register. Its bits have the following meaning:

Byte 1, Bit 0 - In Bus Parity Error: this bit, if on, indicates that a parity error has occurred on the In Bus between the adapter and the redrive logic.

Byte 1, Bit 1 - Out Bus Parity Error: this bit, if on, indicates that a parity error has occurred on the Out Bus between the adapter and the redrive logic.

Byte 1, Bit 2 - Out Bus Tag Check: this bit, if on, indicates that a tag check (incorrect sequence of tags) has occurred on the Out Bus between the adapter and the redrive logic.

Byte 1, Bit 3 - In Bus Tag Check: this bit, if on, indicates that a tag check (incorrect sequence of tags) has occurred on the In Bus between the adapter and the redrive logic.

Byte 1, Bit 4 - Halt Remember: this bit, if on, indicates that the redrive logic has been selected, but a Halt signal was received before the operation ended.

Byte 1, Bit 5 - Secondary Select Out: this bit, if off, indicates that the Select Out tag was propagated to the next redrive address; if on, it indicates that the tag was not propagated.

Note: the bit is set when the select out tag is sent to the adapter controlled by the redrive logic, and is reset automatically if the tag is propagated to the next redrive logic.

Byte 1, Bit 6 - Secondary Cycle Steal Grant: this bit is set when the secondary cycle steal grant signal is sent to the adapter controlled by the redrive logic, and is reset automatically by:

- The next cycle steal operation if the redrive logic propagates the cycle steal grant signal to the next redrive address.
- The next PIO operation on the bus.

Byte 1, Bit 7 - Command Reject: this bit, if on, indicates that an invalid command was sent to the the addressed redrive logic.

COMMAND OUTPUT X'0' OR X'8' (WRITE ERROR REGISTER)

This command is used to set the redrive logic error register for diagnostic purposes. The high order bit of the command may be off, indicating a broadcasted command, or on if the command is addressed to a specified redrive logic. The error register(s) of the redrive logic(s) addressed by the instruction is loaded from the specified register. Byte 1 is not used. If the command is broadcast, all redrive logic error registers are set. The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	In Bus Parity Error
	1	Out Bus Parity Error
	2	Out Bus Tag Check
	3	In Bus Tag Check
	4	Halt Remember
	5	Secondary Select Out
	6	Secondary Cycle Steal Grant
	7	Command Reject
1	0/7	(not used)

Redrive Error Register: Byte 0 is collectively called the Redrive Error Register. For a detailed description of its bits, see under the 'Poll' command above.

COMMAND INPUT X'1' OR X'9' (READ ERROR REGISTER)

The high order bit of the command may be off, indicating a broadcast command, or on if the command is addressed to a specified redrive logic. The register addressed by the instruction is loaded from the redrive logic and contains the redrive address and the event that set the redrive level 1 interrupt. If the command is broadcast, and more than one redrive has a level 1 interrupt pending, the redrive logically closer to the CCU on the internal bus presents its error register data first. Additional Input X'1' or X'9' instructions must be executed if more than one redrive has a level 1 interrupt pending. The bits of the register have the following meaning:

Byte	Bit	Meaning
0	0	1
	1	Enable/disable (enable = 0, disable = 1)
	2	PRA Bit 4
	3	PRA Bit 2
	4	PRA Bit 1
	5	SRA Bit 4
	6	SRA Bit 2
	7	SRA Bit 1
1	0	In Bus Parity Error
	1	Out Bus Parity Error
	2	Out Bus Tag Check
	3	In Bus Tag Check
	4	Halt Remember
	5	Secondary Select Out
	6	Secondary Cycle Steal Grant
	7	Command Reject

Redrive Error Register: For a detailed description of the bits of this command, see under 'Command Input X'0' (Poll) on page E-6.

COMMAND OUTPUT X'1' OR X'9' (DISABLE DRIVERS)

This command is used to inhibit the inputs of the dependant adapter(s) and dependant redrive logics (if any) from the bus. It is, however, still possible to send information to the dependant logics, and to send and receive information from the redrive logic itself. The high order bit of the command may be off, indicating a broadcast command, or on if the command is addressed to a specified redrive logic. As this is a command that performs a function, the contents of the register addressed by the command are ignored.

COMMAND OUTPUT X'2' OR X'A' (ENABLE DRIVERS)

This command is used to enable the addressed redrive logic and the adapter(s) and redrive logics dependant on it. The high order bit of the command may be off, indicating a broadcast command, or on if the command is addressed to a specified redrive logic. As this is a command that performs a function, the contents of the register addressed by the command are ignored.

COMMAND OUTPUT X'5' OR X'C' (RESET)

This command is used to reset all latches of the addressed redrive logic(s), except the enable/disable latch. The high order bit of the command may be off, indicating a broadcast command, or on if the command is addressed to a specified redrive logic. As this is a command that performs a function, the contents of the register addressed by the command are ignored.

APPENDIX F. INITIAL PROGRAM LOAD (IPL)

The initial program load (IPL) mechanism controls the loading of the control program into the 3725. Loading takes place via one of the ports defined in the IPL port table, using either a channel adapter (channel-attached 3725) or a transmission line and one of the communication scanners (link-attached 3725).

Initial program load is performed under the following conditions:

- When the system is first powered up.
- When power is lost and an auto-start occurs.
- When processing cannot continue because of a controller error condition.
- When the channel adapter decodes a 'Write IPL' command from the host.

SEQUENCE OF IPL

IPL takes place in several phases:

- Phase 0: load the MOSS.
- Phase 1: initialize and test the CCU.
- Phase 2: load the Controller Loader Dump Program (CLDP).
- Phase 3: load the communication scanners.
- Phase 4: load and/or initialize the control program.

Note: for a channel-attached 3725, phases 3 and 4 take place simultaneously; for a link-attached 3725, phase 4 cannot take place until phase 3 (load the communication scanners) has been completed.

PHASE 0: LOAD THE MOSS

This phase loads the MOSS from the MOSS diskette, executes a series of internal tests, and initializes the MOSS. The hexadecimal display indicates 3 digits from X'F00' through X'FE0 while the MOSS is being loaded and initialized; at the end of this phase, the indicators show X'FEF'.

Apart from the changing digits on the hexadecimal display, this phase is invisible to the user, except in the case of a power on sequence. In this case, the MOSS General Menu is displayed and the MOSS status indicates 'MOSS ALONE'.

Note: phase 0 only takes place after a 'cold start', that is, under the following conditions:

- At power on time.
- After and auto restart.
- If the START switch is pressed at the control panel.

PHASE 1: INITIALIZE AND TEST THE CCU

During this phase, the MOSS initializes the CCU, as follows:

- The CCU latches are initialized.
- The local storage registers are set to all zeros.
- The main storage is set to all zeros with good parity (power on IPL only).
- The storage protect/address exception mechanism is disabled (power on IPL only).
- The channel adapter registers are initialized with good parity (power on IPL only).
- The CCU and internal bus are tested.
- The communication scanners are tested.

The hexadecimal display indicates the 3 digits X'FF1' during phase 1; at the end of this phase, the indicators show X'FF2'.

Apart from the changing digits on the hexadecimal display, and the changing CCU status on the MOSS Machine Status Area, this phase is also invisible to the user.

PHASE 2: LOAD THE CONTROLLER LOADER DUMP PROGRAM (CLDP).

During this phase, the following events take place:

- The CLDP loader program and the IPL Ports table are loaded into the CCU from the MOSS diskette.
- Control is passed to the CLDP.

While the MOSS is entering IPL Phase 2, the CLDP does the following:

- Enables the channel adapters (channel attached 3725s only).

  Note: until the channel adapters are enabled, they do not reply to initial selection, but simply propagate Select Out to the next device.

- Signals to the hosts that IPL is required:

  - Channel-attached 3725: an asynchronous Device End/Unit Check (DE/UC) status is sent to all hosts to signal that control program loading may begin.

  - Link-attached 3725: the CLDP program must wait for the communication scanners to be loaded and initialized before it can communicate with the host(s).

- Monitors the IPL ports.

At the end of this phase, the indicators show X'FF3', and the message

ENABLED PORTS CA xxxxxx L NNNNNNNN

is displayed on the operator console.

where:

xxxxxx is a pattern of mixed 'Y' and 'N' characters to indicate the enabled (Y) or disabled (N) state of channel adapters 1 through 6.

Note: the 'L NNNNNNNN' pattern indicates that all the Link IPL ports are disabled at this point in time.

PHASE 3: LOAD THE COMMUNICATION SCANNERS.

During this phase, the following events take place:

- All scanners are loaded and initialized in parallel.

- The MOSS monitors the scanner loading process, and sends to the CLDP a list of the scanners that have been successfully initialized.

  Note: control program loading starts here for link-attached 3725s.

At the end of this phase, the indicators show X'FF4', and the message

ENABLED PORTS CA xxxxxx L yyyyyyyy

is displayed on the operator console.

where:

xxxxxx is a pattern of mixed 'Y' and 'N' characters to indicate the enabled (Y) or disabled (N) state of channel adapters 1 through 6.

yyyyyyyy is a pattern of mixed 'Y' and 'N' characters to indicate the enabled (Y) or disabled (N) state of Link IPL ports 1 through 8.

PHASE 4: LOAD AND/OR INITIALIZE THE CONTROL PROGRAM.

For a channel-attached 3725, control program loading may take place simultaneously with phase 3; in this case, phase 4 is limited to the initialization of the control program already loaded. For a link-attached 3725, control program loading cannot take place until phase 3 (load the communication scanners) has been completed.

Channel-Attached 3725 Loading

During this step, the following events take place:

1. The host sends a Write IPL (X'05') command to the channel adapter to inform the CCU that the host is ready to send the control program modules. No data is actually sent via this command. The channel adapter replies with an initial status of Channel End alone; this allows the channel to disconnect immediately. This is necessary, because the 3725 may not be ready at this time (for example, after a restart, when the 3725 must be reinitialized).

2. At the same time that the channel adapter sends channel end to the host in reply to Write IPL, it raises a level 3 interrupt to the CCU, and informs the MOSS by hardware. If necessary, the MOSS initializes the CCU, and loads it with the CLDP.

3. When the CLDP is ready, it sends a Device End status to the host (this Device End may be immediate in the case of normal power on, or delayed in the case of a restart, for example). At this point, the hexadecimal indicators display X'FF5', and the message:

   CA IPL DETECTED ON CA x

   is displayed on the operator console, followed a little later by a second message:

   LOAD IN PROGRESS ON CA x

   where:

   x is the number of the channel adapter 1 through 6.

Note: Channel End and Device End never occur in the same status in reply to a Write IPL command.

4. At this point, the host transfers the control program load module from the host to the CCU, using the normal Write (X'01') command for each block of text. The CLDP reads each block of text, and answers each with a Channel End/Device End status.

5. On the last block of text, the host sends a Write Break (X'09') command to the CLDP, followed by a final Write (X'01') command containing the control program entry point; this causes the CLDP to signal to the MOSS that the control program is loaded, and to transfer control to the control program. At this point, the hexadecimal indicators display X'FF7', and the message

   CONTROL PROGRAM LOADED

   is displayed on the operator console.

6. Under the control of the control program, CCU software initialization takes place. In particular, the CCU receives the CDF parameters from the MOSS, and the MOSS receives the Control Program Initialization Table (CPIT) from the CCU.

7. When CCU initialization is complete, the hexadecimal indicators display X'000' and the message

   IPL COMPLETE

   is displayed on the operator console; the 3725 is ready.

   Note: if an error is detected during the IPL, the hexadecimal indicators display X'FFE', and the message

   IPL COMPLETE + ERRORS

   is displayed on the operator console.

Link-Attached 3725 Loading

At the end of phase 3, the scanners are ready to transmit and receive data. The FF4 indication appears on the hexadecimal display, and the message:

ENABLED PORTS CA xxxxxx L yyyyyyyy

is displayed, where yyyyyyyy indicates those lines that are designated as IPL ports.

The transfer of the control program now takes place via one of the designated IPL ports. A series of indications on the hexadecimal display and on the operator console allows the operator to follow the operation.

1. When one of the hosts is ready to send the load modules to the 3725, the hexadecimal indicators display X'FF6', and the message:

   LINK IPL DETECTED ON L xxx

   is displayed on the operator console, followed a little later by a second message:

   LOAD IN PROGRESS ON L xxx

   where:

   xxx is the address of the link from 0 through 255.

2. The host now transfers the control program load module from the host to the 3725, via a channel-attached 3725 (or 3705) using the SDLC protocol under the control of the CLDP.

3. When the last block of data has been loaded into the 3725, the hexadecimal indicators display X'FF7', and the message

   CONTROL PROGRAM LOADED

   is displayed on the operator console.

4. Under the control of the control program, CCU software initialization takes place. In particular, the CCU receives the CDF parameters from the MOSS, and the MOSS receives the Control Program Initialization Table (CPIT) from the CCU.

5. When CCU initialization is complete, the hexadecimal indicators display X'000' and the message

   IPL COMPLETE

   is displayed on the operator console; the 3725 is ready.

   Note: if an error is detected during the IPL, the hexadecimal indicators display X'FFE', and the message

   IPL COMPLETE + ERRORS

   is displayed on the operator console.

APPENDIX G. BRANCH TRACE

BRANCH TRACE INTRODUCTION

Branch trace is intended as a general debugging tool for the control program. It records, in the branch trace table, the addresses where branches are taken during CCU instruction execution. Interrupts and returns from interrupts via the EXIT instruction are considered as branches for branch trace operations.

Notes:

1. The range of addresses to be traced may be selected by the user via the MOSS.

2. The interrupt levels to be traced may also be selected by the user.

BRANCH TRACE TABLE

The branch trace table consists of a series of eight-byte entries, one for each branch. The following information is recorded in the table:

- The 'come from' program level.
- The 'come from' instruction address.
- The 'go to' program level.
- The 'go to' instruction address.

The information actually recorded varies slightly depending on the type of branch. The following table should make this clear:

	Trace table entry byte							
	0	1	2	3	4	5	6	7
Type of branch	Come from level	Come from address			Go to level	Go to address		
True branch	CPL	Instruction address			CPL	Branch address		
IAR modification	CPL	Instruction address			CPL	New IAR (Reg 0)		
Program interrupt	Old level	Address of last inst executed in old lev.			New level	Address of 1st inst. executed in new lev.		
EXIT instruction	Level EXIT'd	Address of EXIT instruction			New level	Address of 1st inst. executed in new lev.		

Where:

CPL = current program level

The 'come from' and 'go to' levels are encoded as follows:

Level	Hex value
1	X'01'
2	X'02'
3	X'03'
4	X'04'
5	X'05'

SETTING UP THE BRANCH TRACE

The branch trace is set up from the operator console via the MOSS. The user must pass to the MOSS the address of a suitable buffer area, and an initial buffer count (in multiples of eight bytes), for the branch trace table (alternatively, these two parameters may be set by the control program, if one is resident). At this time, other parameters may also be specified, such as:

- Range of addresses to be traced.
- Interrupt levels to be traced.

- Whether or not branch trace wrapping is required. If wrapping is allowed, when the branch trace table is full, the following entries overwrite the earlier entries in the buffer, which are therefore lost.

- Whether or not stop on address is simultaneously required.

Note: the combination of branch trace wrap and stop on address may be used to record the last 'n' branch traces before the stop on address occurred, where n is equal to or less than the number of entries reserved for the branch trace table.

After this initial setting up, the operation is totally transparent to the user.

Programming Notes:

1. When branch tracing is in operation, some degradation of instruction execution time occurs.

2. The address of the branch trace table, as received from the MOSS is available to the control program via the Input X'7B' instruction.

3. The branch trace buffer count, as received from the MOSS is available to the control program via the Input X'7C' instruction.

4. To avoid filling the branch trace table with unwanted timer interrupt traces, the code traced should not include any level 3 code associated with the servicing of timer interrupts.

5. Local storage register X'18' (the IAR of level 3) should be set to a storage address outside of the range of the storage block being traced. Register X'18' must only be set to this value while the CCU is in the 'Wait' state.

6. If no control program is resident in CCU storage, the address and the maximum number of entries must be fixed by the user when he calls the function from the MOSS. If a control program is resident, the address and the count may be set up by the program, and passed to the MOSS. However, the user may still modify these two parameters via the MOSS.

Index

A

B

E

F

R

S

X

Z

Numeric

3725 Communication Controller
Principles of Operation
Order No. GA33-0013-3

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

. . . . . .
. . . . . .
. . . . . .
. . . . . .

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

What is your occupation? . . . . . .

Number of latest Newsletter associated with this publication: . . . . . .

Thank you for your cooperation. No postage stamp necessary if mailed in the USA. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Comment Form

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 812LG
1133 Westchester Avenue
White Plains, New York 10604

Fold and tape | Please Do Not Staple | Fold and tape

IBM

Cut or Fold Along Line

3725 Communication Controller, Principles of Opera (File No. S/370/30xx/4300-09) Printed in USA GA33-001

Publication Number
GA33-0013-3

File Number
S/370/30xx/4300-09

Printed in
USA

IBM