Systems

System/7
Functional Characteristics

This reference manual contains basic programming information for the IBM System/7 processor and I/O devices, and describes the operating characteristics of the system. The specific topics covered are:

- Processor functional characteristics
- Priority interruptions
- Operator console
- Instruction set
- I/O commands and device status
- Interval timers
- Operator station
- Asynchronous communications control
- 1130 attachment
- Analog input and output
- Digital input and output
- 2790 control

This manual is intended as a reference for designing a System/7 configuration and programming the applications of the system. Therefore, it is assumed that the reader has a background in basic data processing concepts and programming, especially those that are pertinent to sensor-based systems and applications.

Prerequisite to this publication is *IBM System/7 System Summary*, Order No. GA34-0002. Information on customer interfaces to the I/O modules (analog, digital, and 2790 control) appears in *IBM System/7 Installation Manual–Physical Planning*, Order No. GA34-0004.

IBM

First Edition (September 1970)

Changes are periodically made to the information herein. Before using this publication in connection with the operation of IBM systems, refer to the latest SRL Newsletter for the editions that are applicable and current.

The illustrations in this manual have a code number in the lower corner. This is a publishing control number and is not related to the subject matter. IBM System/7 units shown herein are at development level and may be subsequently modified in appearance.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.

A form for readers' comments is provided at the back of this publication. If the form has been removed, send your comments to IBM Corporation, General Systems Division, Product and Programming Publications, Department 707, Boca Raton, Florida 33432.

I/O MODULES

The System/7 is designed as a real-time system dedicated to applications requiring the use of sensor-based I/O devices. For this purpose, the I/O modules are provided. I/O module operations are controlled by the processor module. The functions of the processor module are described in a subsequent section. The I/O modules are:

- IBM 5014 Analog Input Module Models B01 and C01. Both can have as many as 128 analog input points, but they operate at different rates of speed (points per second).
- IBM 5012 Multifunction Module Model A01. Special features within this module are:
  - As many as 32 analog input points. All must be of the same speed. Performance of the analog input facility in this module is identical to that of the analog input module (models B01 and C01).
  - As many as two analog output points.
  - As many as 128 digital input points, of which the first two groups (of 16 points each) can have a process interrupt feature which generates an interruption from a change in status of a point.
  - As many as 64 digital output points.
  - One 2790 control feature for attaching a single loop of the IBM 2790 Data Communication System.

SYSTEM/7 CONFIGURATIONS

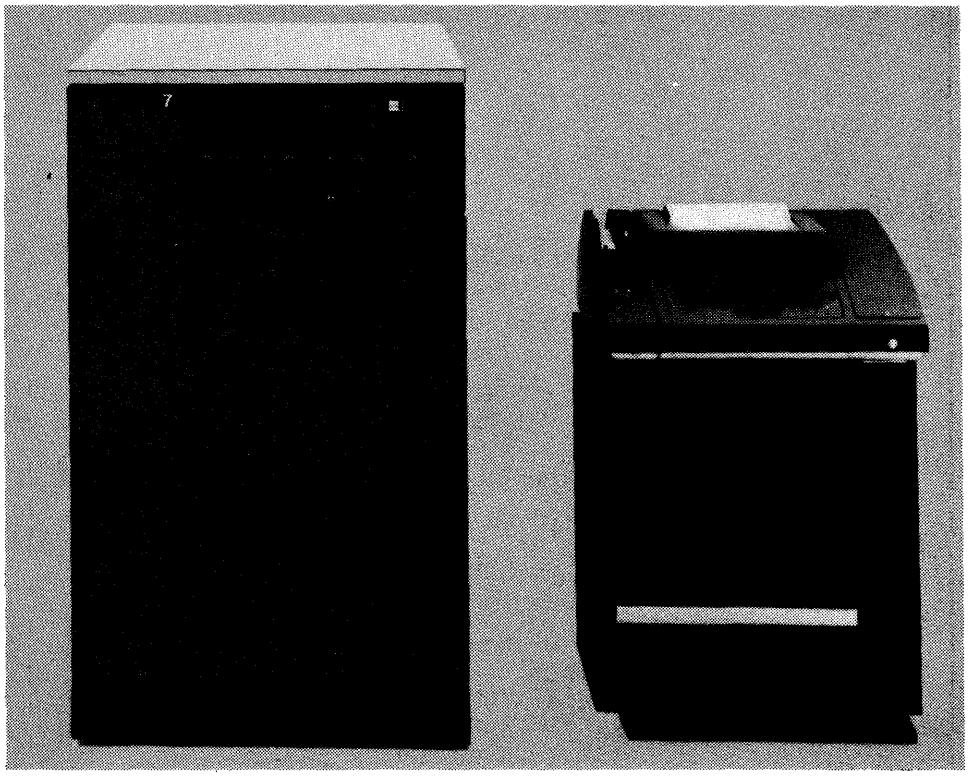
The System/7 is designed so that both its functional and physical structure are highly modular. This permits great flexibility in designing configurations for a broad spectrum of applications. Furthermore, the modularity allows easy expansion of a small initial installation to a more powerful configuration as future needs require.

System/7 can operate as a standalone computing system, or it can function as a satellite processor linked to a host processor which is on-site or at a remote distance.

However, the System/7 is structured independently of any host processor, and is designed to provide modular data acquisition and sensor-based I/O functions independent of the processor to which it may be linked. Therefore, the host processor can be modified or replaced without affecting the sensor-based I/O modules. This gives an installation more flexibility in its overall system planning and minimizes the disturbance to its machine interfaces.

The System/7 can be configured in any one of the following ways:

1. A standalone system with one I/O module of any type. (See Figure 1.)
2. A standalone system with as many as 11 I/O modules in any combination. Each system configuration includes programming support for one 2790 control feature.
3. One or more of the configurations discussed in items 1 and 2 can operate as satellite processors linked to the IBM System/360 (model 25 and up), the IBM System/370, or the IBM 1800 Data Acquisition and Control System. To accomplish this, each System/7 must have the asynchronous communications control feature.
4. Any one of the configurations discussed in items 1 and 2 can be attached directly to an IBM 1130 Computing System via the 1130 storage access channel. It is possible to initial program load (IPL) the System/7 from the 1130.



BR0712

Figure 1. IBM 5026 Model A02 Enclosure and IBM 5028 Operator Station

In all configurations, the IBM 5028 Operator Station is used for communication between the operator and the System/7. Input is presented via the keyboard or paper tape reader. Output is via the printer or paper tape punch. Preparation of programs and IPL of the system can be accomplished through the operator station.

GENERAL FUNCTIONAL DESCRIPTION

Every System/7 configuration has a 5010 Processor Module which controls the system, performing data processing operations and issuing I/O instructions to the sensor-based I/O modules. The main elements of the processor module are a processor, a monolithic storage, and a direct control channel.

The processor is a compact, 16-bit binary computer with a 400-ns storage cycle time. Sixty-four priority interruptions are organized on four priority levels, with 16 sublevels for each level. Interruption status-switching time is minimized by four sets of machine registers and program indicators, one set for each of the four interruption levels. Interruptions also occur for program check, machine check, and power failure conditions.

Performance is enhanced by the use of monolithic storage technology. Storage is organized into words of 16 bits (two bytes), plus a parity bit for each byte. The basic machine has 2k (2,048) words. Additional storage can be obtained in increments of 2k words to a maximum of 16k (16,384) words. Power to the system must be maintained to keep data in storage.

The direct control channel is the interface between the processor and the sensor-based I/O modules. It controls the System/7 internal interface to the attached I/O modules by interpreting I/O commands and transferring data to and from the I/O modules.

The direct control channel contains two standard interval timers. They function as separate program-controlled 16-bit binary counters that decrement at 50-μs intervals. The channel also contains the adapter for attaching the operator station; the optional asynchronous communications control for communicating with a System/360, a System/370, or an 1800 Data Acquisition and Control System; or the optional adapter for attachment to an 1130.

SYSTEM DATA FLOW

Data can enter or leave the System/7 by the operator station, the I/O modules, or a host system.

Data entered through the operator station can consist of programs to be executed, data to be used by the program, or direct commands from the operator to the system. Data leaving the system through the operator station can consist of messages to the operator requesting information or a specific action, or data to be placed on the printer and/or paper tape.

Data from the I/O modules, which requires processing or specific action by the System/7, represents the status of the attached devices. The data might be stored for future action by the System/7 itself or for future action by a host system. Data sent to the I/O modules causes them to perform control operations that affect the application.

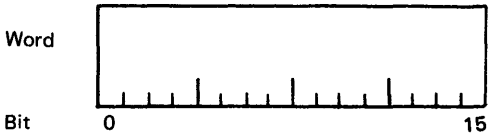
Generally, a host system transmits, to the System/7, control programs, data, and parameters within which certain processes are to be maintained. In turn, the System/7 sends, to the host system, collected data for processing, status information, or requests for new routines to handle situations which cannot be serviced by the routines that reside in the System/7.

DATA FORMATS

To be accessible to the program during processing, data must be in storage (or a register). Thus, input job data is read by an input device, placed in storage, and then processed. Results (output data) are sent to an output device.

Input data can be represented in a variety of ways depending upon the input medium used. A medium is the material on which data is recorded. For example, a paper tape that is punched with holes (which represent data) is a medium; the paper tape can be read by a paper tape reader.

In the System/7, data is read from or stored in storage on a word basis. A word is made up of 16 bits, numbered 0 to 15.



BR0829

The leftmost bit (0) is the high-order bit; the rightmost bit (15) is the low-order bit. Each bit can be set to either a binary 1 (on value) or a binary 0 (off value). For example, a word can contain:

Bit →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value →	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Or:

Bit →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value →	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

BR0830

Or, any combination is possible:

Bit →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value →	1	0	0	0	1	1	0	1	0	1	1	1	0	0	0	1
Or →	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0
Or →	1	1	1	1	0	0	0	0	1	1	1	1	0	0	1	1

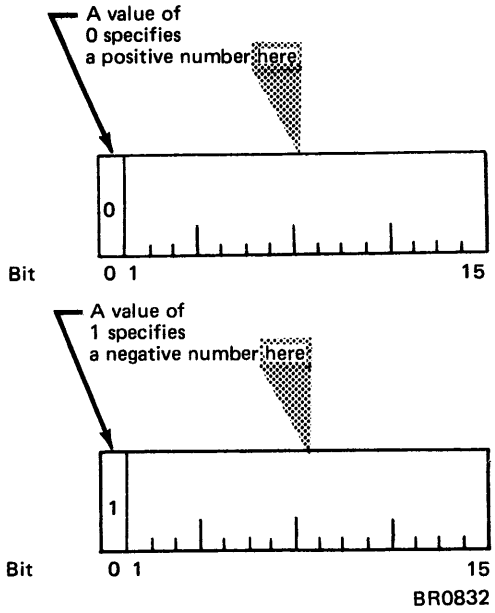
BR0831

Numeric, alphabetic, special character, or logical information can be represented by the bit values in a word, as determined by the combination of the bits in a word and the intention of the programmer who organizes the program and data in storage.

Numeric Data Formats for Arithmetic Operations

Numeric computations in the System/7 are performed in binary arithmetic by the arithmetic instructions. In other words, the System/7 operates on binary data to produce binary results.

The value of bit 0 (the high-order bit) of a word specifies whether the word represents a positive or negative number. If the value of bit 0 is 0, the number represented in the word is positive; if the value of bit 0 is 1, the number represented in the word is negative.



BR0832

The assumption has been made that the reader can, when given a numeric value represented by the symbols in the binary, hexadecimal, or decimal numbering system, convert those symbols to an equivalent numeric value in either of the other two systems. For example, the reader should be able to convert the binary symbols 11100010 to the hexadecimal representation E2, or to decimal representation 226. The reader should also be able to do simple addition or subtraction with numeric values represented by binary or hexadecimal symbols. (Refer to *Number Systems,* Order No. GC20-1618, for basic information on binary and hexadecimal numbering systems.)

The ranges of numeric values that can be represented in a 16-bit word are shown in binary and decimal system symbology as follows:

16-bit word					
Binary					Decimal
Bit 0 (sign)	Bits 1-15 (numeric value)				
0	000	0000	0000	0000	00000
+	to				to
0	111	1111	1111	1111	32,767
1	000	0000	0000	0000	-32,768
-	to				to
1	111	1111	1111	1111	-00001

BR0833

Negative numbers are represented in twos-complement binary form; the sign bit, containing a 1, signifies the negative number. For example, the number 1111 1111 1111 1111 is shown as having an equivalent decimal value of -00001. (The left 0's are shown since 16 binary bits can represent a five-digit decimal number.) This binary value is really the twos complement of negative 1. The absolute form can be obtained as follows:

1. Invert (change each position from 1 to 0):

   1111 1111 1111 1111

   becomes

   0000 0000 0000 0000

2. Add 1 to the inverted number:

```
0000 0000 0000 0000
                +1
-------------------
0000 0000 0000 0001
```

Negative results are always represented in twos-complement form, which specifically excludes a negative zero. Positive numbers are always represented in true binary notation with a 0 sign bit.

STORAGE ADDRESSES

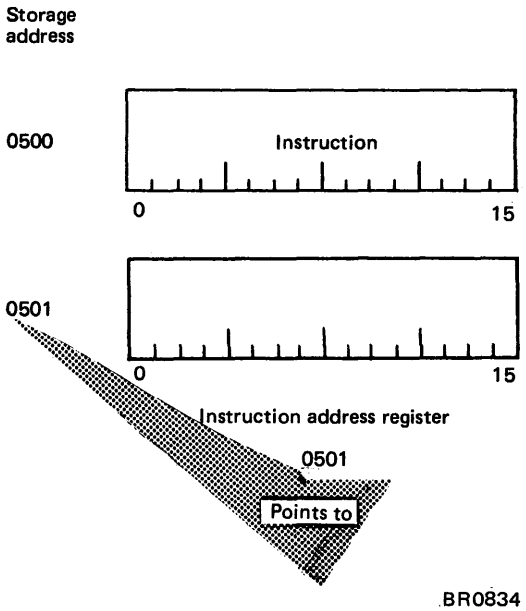
Storage capacity ranges from 2k (2,048) words to 16k (16,384) words. Storage between these limits can be obtained in 2k increments. The address for any storage location in System/7 can be contained in a 16-bit word:

Binary Address (16 bits)	*Hexadecimal Address*	*Decimal Address*
0000 0000 0000 0000	0000	00000
0000 1111 1111 1111	0FFF	4,095
0001 1111 1111 1111	1FFF	8,191
0011 1111 1111 1111	3FFF	16,383

REGISTERS

Instruction Address Register (IAR)

The instruction address register (shown on Figure 2) holds the address of the next instruction to be executed. For example, assume that the following instruction is being executed:



The 16-bit IAR contains the address of the storage word immediately following the instruction being executed. In most cases, this next word is the next instruction to be executed. Sometimes, however, the contents of the instruction address register are changed as a result of the instruction being executed. Execution of a branch instruction, for example, can cause accessing of the next instruction from a storage location other than the one immediately following the current instruction.

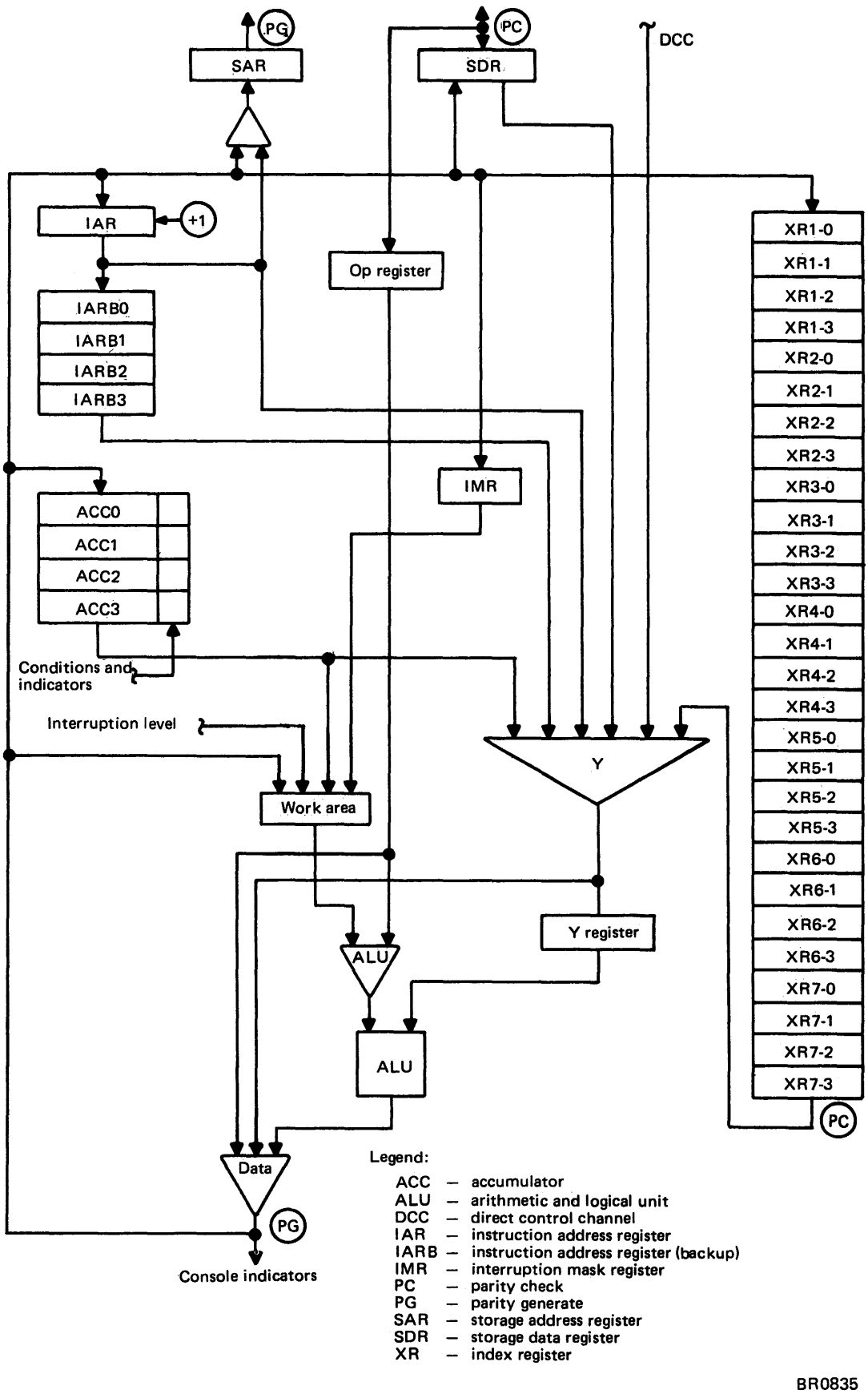


Figure 2. Processor Data Flow

The System/7 processor also has four backup instruction address registers (IARB0-IARB3), one for each priority interruption level. When a change of interruption levels occurs, the instruction address is automatically saved in the IARB for the level that is interrupted. When a return is made to that level, processing can be resumed at the location specified by the saved address.

The contents of any selected IAR or IARB can be displayed on the operator console, or console switches can be used to store a word into a selected IAR or IARB.

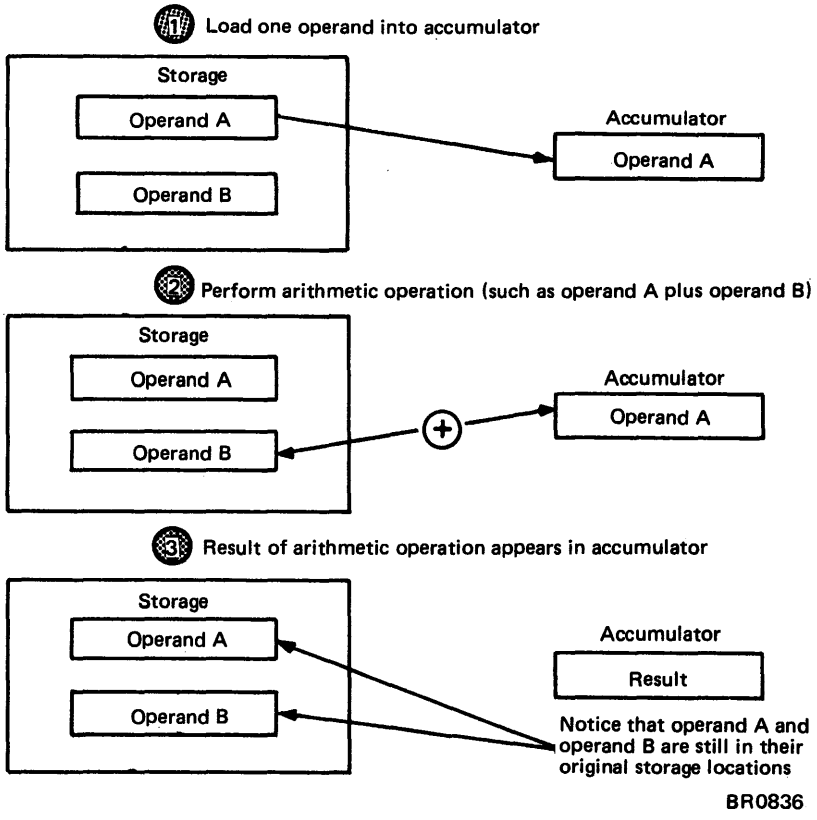
Index Registers (XR1-XR7)

The System/7 processor has seven index registers for each of the four priority interruption levels, for a total of 28. (See Figure 2.) Each index register has 18 bits (16 data bits plus 2 parity bits). This bit structure can represent either positive or negative numbers. (Refer to "Effective Address Generation" for a description of addressing concepts and how index registers can be used in generating effective addresses. Refer to the detailed instruction descriptions under "Instruction Set" for discussions of the ways in which the contents of the index registers can be manipulated in arithmetic and logical operations.)

The contents of any selected index register can be displayed on the operator console, or console switches can be used to store a word into a selected index register.

Accumulators (ACC0-ACC3)

The System/7 processor has four accumulators (ACC0-ACC3), one for each priority interruption level. (See Figure 2.) These 16-bit registers are used in arithmetic and logical operations. In general, instruction execution is performed using one specified operand (from storage or an index register) and one implied operand (from the value previously loaded into the accumulator). The result is in the accumulator at the end of the operation. This result is obtained by a machine function that depends upon the instruction that is executed.



The contents of any selected accumulator can be displayed on the operator console, or console switches can be used to store a word into a selected accumulator.

Interruption Mask Register (IMR)

This four-bit register (shown on Figure 2) has a bit corresponding to each of the four (0-3) priority interruption levels. The appropriate bit in the mask register is set on (given a 1 value) to enable the corresponding priority interruption level (i.e., interrupting sources assigned to that priority level can present interruptions to the system if their priority is higher than the current operating program). The contents of the register can be examined or modified by three instructions: AND to mask, OR to mask, and sense level and mask.

Miscellaneous Machine Registers

The four registers described in the following paragraphs function automatically as required by the operation in process. (See Figure 2.)

Storage Data Register (SDR)

Every word of data transferred to or from main storage by the processor passes through the SDR. The register holds 16 data bits and two parity bits.

Operation Register

This 16-bit register holds the first word of each instruction fetched from storage. The contents of the operation register can be displayed by the data lights on the operator console.

Storage Address Register (SAR)

The 16-bit storage address register contains the address of each location that is accessed in main storage by the processor. The SAR is displayed by the console address lights.

Y Operand Register

This 16-bit register holds the first operand of an instruction. The Y register can shift its contents to either the left or the right, for use in shift instructions.

PROGRAM INDICATORS

The System/7 has six indicators that show the status of a data operation result field. At the completion of most instructions, the result is automatically tested for certain conditions, and indicators are set accordingly. Some of the branching instructions can test the indicators for conditional branching situations.

Once an instruction sets the indicators either on or off, they remain in this state (and can be tested) until the execution of another instruction changes the indicators according to the result of the operation. (Refer to the detailed description of the individual instructions to determine the effect of each instruction on the setting of these six program indicators.)

Each of the four priority interruption levels has a set of program indicators.

Carry Indicator

The carry indicator is set on when an operation produces a result that exceeds the physical capacity of the 16-bit data flow. This happens when a 1 is shifted or carried out of bit 0 (the sign bit) by an add or shift left logical instruction. It can also occur when a subtract operation attempts a borrow beyond bit 0.

Overflow Indicator

The overflow indicator is set on if the result of an arithmetic operation exceeds the logical capacity of the 16-bit data flow. This happens when an add or subtract operation produces a result greater than $2^{15} - 1$ or less than -2^{15}. This range is from +32,767 to -32,768 in decimal notation and from 7FFF to 8000 in hexadecimal notation.

Result-Zero Indicator

An operation result field has a zero value when all of its bits are set to 0's. Such a condition sets this indicator on.

Result-Even Indicator

If the low-order bit (bit 15) of a result is 0, this indicator is set on to indicate that the operation has produced an even number.

Result-Positive Indicator

The high-order bit (bit 0) of a result is the sign bit. A 0 sign bit signifies a positive number, which causes this indicator to be set on. If the numeric value of the entire result field is zero, this indicator is set off because the result-positive definition does not include a zero value.

Result-Negative Indicator

The result-negative indicator is set on when the result of an operation is a negative number. This occurs when the sign bit (bit 0) is set to 1. A negative zero is not represented in the System/7.

PARITY

A System/7 word, whether used to hold data or an instruction, consists of 16 information bits. To ensure internal accuracy in data transmission and manipulation, System/7 has an internal checking system of odd parity. Every 16-bit word (bits 0-15) contains two 8-bit bytes (bits 0-7 and 8-15). Each byte must have an odd number of bits set on (having a value of 1) at all times. Obviously, this does not occur at all times because it is valid for data or instructions to be represented with an even number of bits set on. For example, 00000000 00001001 has no bits set on in the first byte and two bits set on in the second byte. Also, 11111111 11111111 has eight bits set on in each byte.

The ability for odd parity checking is designed into the system by adding two additional bits (called parity bits, P_0 and P_1) to each word. So, in reality, each word in the System/7 contains 18 bits. Each parity bit is associated only with its corresponding byte as shown by the following bit organization of a System/7 word:

P_0 0 1 2 3 4 5 6 7 P_1 8 9 10 11 12 13 14 15

Each parity bit is used to maintain odd parity in its associated byte by the following method. If the 8-bit byte already has odd parity, its parity bit is set to a 0 value. If the 8-bit byte has even parity, its parity bit is set to a 1 value to give the entity (byte plus parity bit) odd parity. The two previous examples would appear in main storage as:

P_0		P_1	
1	00000000	1	00001001
1	11111111	1	11111111

During program execution, any time that the value (contents) of a word is intentionally changed, the machine automatically modifies the parity bits to maintain odd parity.

Words entering the System/7 are analyzed to determine whether each byte has odd or even parity. Parity bits are then added to the word to maintain odd parity.

Every time a word is used in the system, each byte plus its parity bit is checked to ensure that an odd number of bits are set on. If an even number is detected (parity error) in the processor, a machine-check interruption occurs. A parity error detected within the direct control channel or I/O modules is recorded in the device status word.

All data used by a program flows through the storage data register. Data coming from a source (e.g., main storage) which already has parity bits is checked for odd parity. Data coming from a source (e.g., the accumulator) without parity bits is checked and parity bits are added to maintain odd parity.

Figure 2 shows the locations of the parity check and parity generate circuits in the processor.

POWER FAILURE AND THERMAL WARNING

Power failure detection, coupled with automatic restart, is a special feature that offers an early warning of possible power failures due to low voltage or loss of primary voltage. AC power falling below 90 percent of nominal voltage triggers the power failure detection circuit which signals the processor of the possibility of an imminent shutdown. The signal remains in effect for the duration of the low-voltage condition.

If the power failure detection–auto restart feature is installed, automatic restart can be enabled by an operator console switch selection.

The auto-restart switch is a two-position toggle console switch that can be set by the operator. In the off position, operator intervention is necessary to restart the system if power fails. When set in the on position, if a power failure occurs, the system is automatically reset and an attempt is made to restore power. If power is restored, automatic IPL takes place if the IPL tape is loaded and positioned correctly in the operator station.

The System/7 can operate in a temperature range from 40°F to 122°F (4.4°C to 50°C). Overtemperature or undertemperature conditions are shown by a console indicator which remains active until the condition is terminated.

Power failures caused by thermal conditions result in the same action as those caused by low voltage.

PRIORITY INTERRUPTION SYSTEM

System configurations can have a large number of devices and attachments capable of sending and receiving data. Information gathered from various sources can have different data rates. Handling requests for attention from I/O devices in serial fashion does not necessarily result in optimum response or throughput. Data sources that have rapidly changing information, or that require minimum processing per change, should have a higher processing priority than those sources that have slowly changing information, or that require longer processing time. A priority system is required to coordinate the real-time servicing of many varied I/O devices.

System/7 has such a system, whereby an executing program can be interrupted by another program with a higher priority. Status of the interrupted program is saved in the machine registers, the higher-priority program is performed, and then control of the system is restored to the interrupted program.

A program can be interrupted by some external condition such as a sensor-based I/O device's signaling its readiness to transmit data, or by an internal machine condition such as a parity error or an invalid machine operation-code.

The interruption system can stop the program in progress in such a way that it can be resumed quickly after the interruption has been analyzed and serviced. To accomplish this in a minimum of time, all registers and indicators that are pertinent to program operation are duplicated on each priority interruption level.

Priority Interruptions

Briefly, the priority interruption scheme works in the following manner. As long as no higher priority device requests attention, current program processing and communication with I/O devices continues to completion. The last instruction executed in the current program must be a level exit instruction. If no interruptions are pending, the machine goes into the wait state. In the wait state, no instruction processing is done, but the machine can recognize and accept any type of interruption for which it is enabled.

If interruptions are pending on the same or lower priority levels, upon completion of the current program and execution of the level exit instruction, the machine does not enter the wait state, but immediately transfers to service the highest pending priority interruption.

If a priority interruption occurs that is higher than the currently executing program, the current program's status is automatically saved in the machine registers assigned to the level that was interrupted, and processing transfers to the routine that services the higher-priority request. This program can, in turn, also be interrupted by a higher-priority interruption. The machine can accept interruptions between the execution of each instruction. The original program is not resumed until all higher-priority interruptions are serviced.

Interruption Levels

The System/7 priority interruption system has four levels of priority interruptions, designated as levels 0, 1, 2, and 3. System/7 design allows the user to assign each interrupting source a level and sublevel (there are 16 sublevels per level) dynamically via program control. This permits the user to easily restructure the priority organization as the application changes.

An interruption level designation of 0 is recognized as the highest priority by the System/7; level 3 has the lowest priority.

If processing is not being performed on any of the four interruption levels, the System/7 enters the wait state. A wait light and four level lights on the operator console indicate the operating status of the system.

Interruption Level Switching

System/7 is designed to facilitate program switching from one level to another with minimum overhead. This system overhead time would be significant if programming were required to save the status of an interrupted program each time a switch is made.

However, as shown on Figure 2, System/7 design provides each of the four interruption levels with its own set of registers (an accumulator, a set of six program indicators, an instruction address register, and a set of seven index registers). Program switching in System/7 is performed in less than 1 μs.

Interruption Level Stacking

Interruptions from devices are not serviced immediately if processing is taking place at the same, or at a higher, priority level. The system is capable of stacking or queueing pending interruptions.

Each of the four interruption levels has one buffer in the direct control channel. The first request for any particular level is placed in its corresponding buffer. For machine control to pass to that level, the request is moved out of the buffer and honored. If another request is pending at that level, it enters the buffer where it is available for servicing when the present program is completed and system control is returned to that level.

When the buffer is full (i.e., occupied by a request not yet honored), another device at the same level requesting service cannot be recognized. This interruption is stacked and must compete with any other requests made to access the buffer at that level, when the buffer becomes available again.

No matter how many interruptions are pending from devices on a particular priority level, the next one to be serviced when control returns is the request in the buffer.

Interruption Sublevel Branching

An I/O device can be assigned to the same interruption level as other I/O devices, i.e., given the same priority number. Although the I/O devices share a common priority, the user determines whether they are to share the same servicing routines. The user can choose to have them share routines or use their own routines.

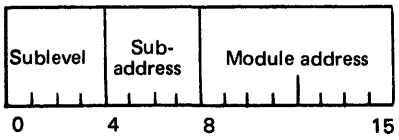
The user decides the priority of the various I/O devices by assigning them a level number (by a prepare I/O command). The level determines which branch table in storage is to be accessed for the starting address of the appropriate routine. One branch table must be established in main storage for each interruption level that is assigned (maximum of four).

Another parameter, the sublevel, is assigned by the user program to each device (by a prepare I/O command) so that the system can locate which address in the branch table is used to access the desired routine.

Upon detecting an interruption (from a level enabled for interruptions), the system examines the priority level to determine whether the interruption is to be permitted. If the same level or a higher level is processing, the request remains in the buffer. If not, the interruption request is honored. Using the level number, the system accesses the branch table reserved for that level. Using the sublevel as a displacement value, the system locates the desired address in the branch table and then branches to that address, which is the location of the first instruction in the routine that the user has established in main storage to service the interrupting source. This total interruption process is performed in less than 1 μs. (Refer to "Reserved Storage Locations.")

System/7 design allows for multiple interrupting sources (I/O devices) to be assigned to the same interruption level and sublevel. This permits a common interruption servicing routine to be used for multiple similar I/O devices. The device (out of two or more assigned to the same interruption level and sublevel) that actually caused the interruption is determined by the program's examining the accumulator contents immediately after the interruption.

The accumulator (associated with the level) contains the address of the I/O device causing the interruption, in the following format:



BR0837

During interruption presentation, an I/O device also presents a summary status bit, which is placed into the carry indicator associated with the interruption level. This summary status bit is 0 if a normal interruption condition occurred, or 1 if an exception or error interruption condition occurred.

Interruption Mask Register

To give the user additional control over the automatic interruption system, a 4-bit interruption mask register is provided. This register has a bit (0, 1, 2, or 3) corresponding to each of the four interruption levels.

With this register the user can further modify, in a preplanned manner, the sequence in which the priority interruptions are serviced. The appropriate bit in the mask register is set on if its corresponding interruption level is enabled (i.e., devices assigned to that priority level can interrupt the system). Setting the bit off disables the level so that no interruptions are honored on that level.

The contents of the mask register can be examined or modified by three instructions: AND to mask, OR to mask, and sense level and mask.

Interval Timer Interruptions

Two 16-bit interval timers are standard system features housed within the direct control channel. Each timer presents an interruption request to the system after counting a user-specified number of 50-μs time intervals. Though not strictly an I/O device, the timers present interruption requests to the system in the same manner as I/O devices and compete with them for system attention.

CLASS INTERRUPTIONS

Some internal machine conditions can also interrupt a processing program. Three types of class interruptions can occur in the system: program checks caused by invalid machine instructions, machine checks due to machine errors, and power/thermal failures.

Class interruptions cannot be disabled (i.e., masked off so that they do not interrupt the system). Conditions that cause class interruptions are serious enough that they must always be capable of interrupting system operations. Class interruptions automatically disable all four priority interruption levels and do not save the contents of machine registers. Therefore, the program that services the class interruptions must save register contents by storing, and must re-enable the mask register.

Class interruptions cause an interruption branch, allowing entry into error recovery routines to service the condition. The starting addresses of these routines are stored in fixed locations in reserved lower main storage. (Refer to "Reserved Storage Locations.")

Priority I/O interruptions cause movement (up or down) in priority but class interruptions do not (i.e., they remain on the same priority level as the interrupted program).

Lights on the operator console indicate the cause of a class interruption during machine stop state.

Program Checks

Program-check interruptions occur upon detecting a machine instruction that is invalid for any of the following reasons:

1. Invalid operation-code.
2. Invalid modifier field contents (hexadecimal values D through F) in instructions with operation code of 11111.
3. Invalid address (i.e., a main-storage address that exceeds the limits of the storage installed in the system).
4. Invalid function field in an execute I/O instruction.
5. Invalid shift count value in shift instructions.

Operator console lights indicate (when the machine is in the stop state) the cause of a program check, which can occur on one or more of the four priority interruption levels.

When a program-check interruption occurs, the IAR contents (address of the next instruction in the interrupted program) are saved automatically in the first word of the appropriate level table. The starting address of the program-check servicing routine is loaded into the IAR from storage location 9 (its fixed address) and the servicing routine is given control. (Refer to "Reserved Storage Locations.")

Machine Checks

Machine-check interruptions occur when machine malfunctions are detected by error-checking circuitry. Malfunctions can be caused by:

1. A parity error in a register.
2. More than one type of storage cycle requested or taken simultaneously.
3. More than one interruption level being executed simultaneously.

Operator console lights indicate the cause of a machine check when the machine is in the stop state. Only one machine check can occur at any one time in the system.

When a machine-check interruption occurs, the IAR contents (address of the next instruction in the interrupted program) are saved automatically in storage location 13. The starting address of the machine-check servicing routine is loaded into the IAR from storage location 11 (its fixed address) and the servicing routine is given control. (Refer to "Reserved Storage Locations.")

Power/Thermal Failure

Power/thermal failure interruptions result when the power control system senses that power has fallen to below 90 percent of nominal voltage. This condition is caused by voltage or temperature changes that border on the operating limits of the system. (Refer to "Power Failure and Thermal Warning.")

When a voltage or temperature condition causes a power failure interruption, the IAR contents (address of the next instruction in the interrupted program) are saved automatically in storage location 12. The starting address of the power failure servicing routine is loaded into the IAR from storage location 10 (its fixed address) and the servicing routine is given control. (Refer to "Reserved Storage Locations.")

RESERVED STORAGE LOCATIONS

In main storage, an area is reserved for:

1. The restart instruction.
2. Addresses required for processing class interruptions.
3. Address pointers to the level tables.

These storage locations should not be used for purposes other than those for which they are intended; otherwise, required system operating data can be destroyed. (See Figure 3.)

	Storage location (decimal)	Contents
Level vectors	17	Level 3 start instruction address pointer
	16	Level 2 start instruction address pointer
	15	Level 1 start instruction address pointer
	14	Level 0 start instruction address pointer
Class vectors	13	Machine-check old instruction address
	12	Power failure old instruction address
	11	Machine-check start instruction address
	10	Power failure start instruction address
	9	Program-check start instruction address
	8	Reserved
	7	Unassigned
	6	
	5	
	4	
	3	
	2	
	1	Restart instruction
	0	

BR0838

Figure 3. Reserved Storage Locations

Variable-size level tables (one table for each priority interruption level specified by the user) are also necessary, but need not be placed in fixed locations in lower storage. (See Figure 4.)

	Table word (decimal)	Contents (decimal)
Ascending storage locations ↑	n + 16	Level 0 sublevel 15 interrupt routine address
	n + 15	Level 0 sublevel 14 interrupt routine address
	n + 14	Level 0 sublevel 13 interrupt routine address
	n + 13	Level 0 sublevel 12 interrupt routine address
	n + 12	Level 0 sublevel 11 interrupt routine address
	n + 11	Level 0 sublevel 10 interrupt routine address
	n + 10	Level 0 sublevel 9 interrupt routine address
	n + 9	Level 0 sublevel 8 interrupt routine address
	n + 8	Level 0 sublevel 7 interrupt routine address
	n + 7	Level 0 sublevel 6 interrupt routine address
	n + 6	Level 0 sublevel 5 interrupt routine address
	n + 5	Level 0 sublevel 4 interrupt routine address
	n + 4	Level 0 sublevel 3 interrupt routine address
	n + 3	Level 0 sublevel 2 interrupt routine address
	n + 2	Level 0 sublevel 1 interrupt routine address
	n + 1	Level 0 sublevel 0 interrupt routine address
	n	Level 0 program-check old instruction address

BR0839

Figure 4. Priority Level Table Format

The reserved storage locations are described in detail as follows:

Storage Location	*Contents*
0-1	The restart instruction. After loading the IPL routine from the operator station, the machine automatically branches to location 0 to initiate processing.
2-7	Unassigned.
8	Reserved.
9-11	The addresses of the first instructions in the routines that service program-check, power failure, and machine-check interruptions, respectively. When a class interruption occurs, the appropriate start instruction address (SIA) is loaded into the IAR and a branch is made to the servicing routine.
12-13	The IAR contents (address of the next instruction in the interrupted program), after a power failure or machine-check interruption, respectively. This is the old instruction address (OIA). Control can then be returned to the correct instruction in the interrupted program after successful completion of the interruption servicing routine. Only one word is required to save the interrupted program address for each type of interruption because only one power failure or machine check can occur at any one time (it is not possible to have a machine check or power failure at more than one interruption level at the same time).
14-17	The addresses of the first word in the level displacement tables for interruption levels 0-3, respectively. Following a priority I/O interruption, this area is accessed to find the start instruction address of the level table assigned to the interruption level that is to be serviced.

Locations 0-17 are fixed as to size and function of contents. However, additional storage is required for the level displacement tables assigned to each of the priority interruption levels (0, 1, 2, and 3). These tables are variable both in number (from 1 to 4) and individual size (from 2 to 17 words). They can be located elsewhere in main storage. (See Figure 4 for the typical format of a priority level table.)

The storage space required for the level displacement tables is dependent, firstly, upon the number of priority interruption levels needed, i.e., the number of different priorities assigned to I/O devices by the user. If only two different priority levels are used (e.g., interruption levels 0 and 1), only two level displacement tables need be established in storage.

Secondly, storage space required for each of the two tables, in the preceding example, depends upon the number of sublevels assigned to I/O devices on the same priority level. For instance, six sublevels could be assigned to devices that are on priority level 0, while 12 sublevels could be assigned to devices that are on priority level 1. This would require seven words for the level 0 table and 13 words for the level 1 table.

Each interrupting source in the system must be assigned both a level and a sublevel, so that the system can locate the starting address of its servicing routine. The system accesses the correct branch address by adding the sublevel to the address of the second word in the table corresponding to the particular level.

The first word in any defined level table is reserved to hold the contents of the IAR if a program-check interruption occurs on that particular level. Each table can then have from 1 to 16 additional words, one for each unique sublevel assigned to interrupting sources on that level.

One table must be established in storage (need not be contiguous) for each priority level assigned. The words must be in ascending order of storage locations, i.e., if word 1 is at storage location 123, word 17 is at storage location 123 + 16, or location 139. Each table need be only as long as required by the current assignment of sublevels.

Every System/7 configuration has a data processing element that fetches and executes instructions in order to perform operations on data (such as testing, arithmetic, movement, and logical decision operations).

In the System/7, the processor is the data processing element of the system. It is a compact, 16-bit binary computer with a set of machine instructions that can manipulate data in a wide variety of ways.

INSTRUCTION CLASSES

For purposes of discussion and clarification, the instruction set is divided into eight classes that describe, in general, the type of data operation that is performed.

Mnemonics are used to abbreviate the instruction names when programming in assembler language. All mnemonics begin with P to indicate that the instructions are for execution by the System/7, although the instructions may be assembled on a different computer. The System/7 instructions, along with their mnemonics, operation codes, and execution times, are:

Instruction	*Mnemonic*	*Operation Code*	*Execution Time in Nanoseconds*
Load and store			
Load accumulator	PL	11000	800
Load and zero	PLZ	11001	1200
Load immediate	PLI	01100	400
Load index long	PLXL	10001	1200
Store accumulator	PST	11010	800
Store index	PSTX	01101	800
Arithmetic			
Add	PA	10000	800
Subtract	PS	10010	800
Add register	PAR	11111	400
Subtract register	PSR	11111	400
Complement register	PCR	11111	400
Add immediate	PAI	01110	400
Logical			
AND	PN	11100	800
OR	PO	11101	800
Exclusive OR	PX	11110	800
AND register	PNR	11111	400
OR register	POR	11111	400
Exclusive OR register	PXR	11111	400
Shifting			
Shift left logical	PSLL	00010	400 + 50N
Shift left circular	PSLC	00010	400 + 50N
Shift right logical	PSRL	00010	400 + 50N
Shift right arithmetic	PSRA	00010	400 + 50N

(N is the number of bits shifted.)

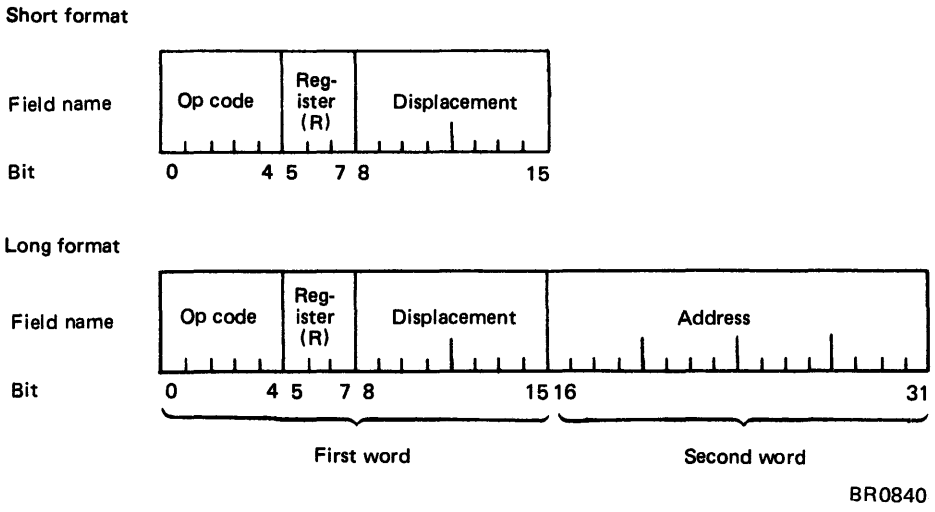
Instruction	*Mnemonic*	*Operation Code*	*Execution Time in Nanoseconds*
Branching			
Branch	PB	00111	400
Branch and link	PBAL	01011	400
Branch and link long	PBALL	01010	800
Branch on condition	PBC	01000	400–no branch 800–branch taken
Skip on condition	PSKC	01001	400
Add to storage and skip	PAS	01111	1200
No operation	PNOP	11111	400
Register-to-register			
Store to register	PSTR	11111	400
Load from register	PLR	11111	400
Interchange register	PIR	11111	400
Load processor status	PLPS	11111	400
Inspect IAR backup	PIIB	11111	400
AND to mask	PNM	10110	400
OR to mask	POM	10110	400
Sense level and mask	PSLM	11111	400
State control			
Level exit	PLEX	00110	400
Input/output			
Execute I/O	PIO	00001	800 + I/F

(I/F is the internal interface delay time. It varies from 450 to 1800 ns depending on the physical location of the I/O module concerned.)

INSTRUCTION FORMATS

Two instruction formats are used in the System/7. A short-format instruction is 16 bits in length (one word); a long-format instruction is 32 bits in length (two words).

Both instruction formats are divided into several fields whose contents specify the operation to be performed and the location of the data to be processed. Most format fields have the same meaning from one instruction to the next; some, however, have special meanings dependent upon the instruction. These exceptions are discussed in the detailed description of the instruction.



The operation-code (Op code) field is five bits that specify which instruction is to be performed.

The register (R) field is three bits that indicate whether the accumulator, instruction address register, or one of the index registers is to take part in the instruction execution. An R field of 000 specifies either the accumulator or the instruction address register (IAR) depending on the instruction. The index registers are specified by the following R fields:

001–Index register 1
010–Index register 2
011–Index register 3
100–Index register 4
101–Index register 5
110–Index register 6
111–Index register 7

The contents of the register specified in the R field can be used for generating an effective address to locate the operand, as an instruction operand, or as data sent to an I/O device.

The displacement field is eight bits long. Its function is determined by the instruction to be performed, as indicated in the operation-code field. The displacement field can contain data that is to be manipulated in some fashion by the instruction. Bits may also be set in the displacement field to modify the operation code so that a particular instruction is executed. This is necessary because some instructions (such as the shifting instructions) have the same operation code. The most common use of the displacement field is to hold data that takes part in generating an effective address to locate the operand.

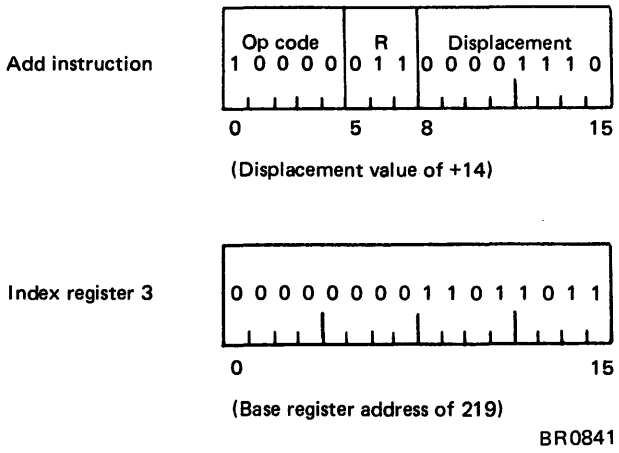
The address field is 16 bits long. Its contents are used in generating an effective address to locate the operand in three of the four long-format instructions (load index long, branch and link long, and branch on condition). With execute I/O, the fourth long-format instruction, the address field is used to specify the I/O instruction modifier and the address of the I/O device.

EFFECTIVE ADDRESS GENERATION

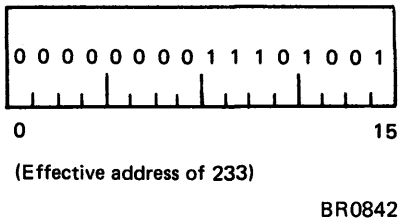
Storage addresses are expressed internally in 16-bit binary numbers, ranging from zero (16 bits set to 0) to a maximum value determined by the storage size of the System/7 configuration. Any attempt to address a location that is beyond the limits of the storage installed in the system causes a program-check interruption due to the invalid storage address.

Most instructions refer to a storage address to locate the data that is to be used. This storage address is called the effective address (EA). The System/7 operates on a base register-displacement principle for storage addressing.

In a short-format instruction, the effective address is derived by adding algebraically the contents of the displacement field to the contents of the register indicated by the R field. The R field specifies the contents of the instruction address register (R = 000) or one of the index registers (R = 001 to 111) for this purpose. For example:



The algebraic sum of the displacement field in the instruction and the contents of index register 3 is the effective address of the data that is to be used by the add instruction. In this case:



The displacement can be either positive or negative, with bit 8 of the instruction word as the sign bit. If this high-order bit of the displacement field is a 0, the displacement is positive with a maximum value of +127 (decimal). If the high-order bit of the displacement field is a 1, the displacement is negative with a maximum value of -128. The negative number is represented in twos-complement form.

The use of the address field varies for each long-format instruction. (Refer to the detailed descriptions of the four long-format instructions—load index long, branch and link long, branch on condition, and execute I/O.)

INSTRUCTION DESCRIPTIONS

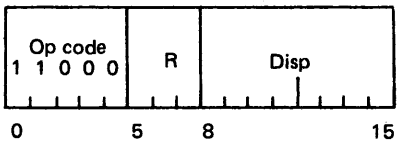
Each machine instruction is discussed in detail, in the order listed in "Instruction Classes." Following each instruction name is its mnemonic, execution time in nanoseconds (ns), instruction format, and description.

LOAD AND STORE INSTRUCTIONS

Six instructions are provided to move data between storage and the index registers/accumulator in the processor module.

Load Accumulator

PL–800 ns



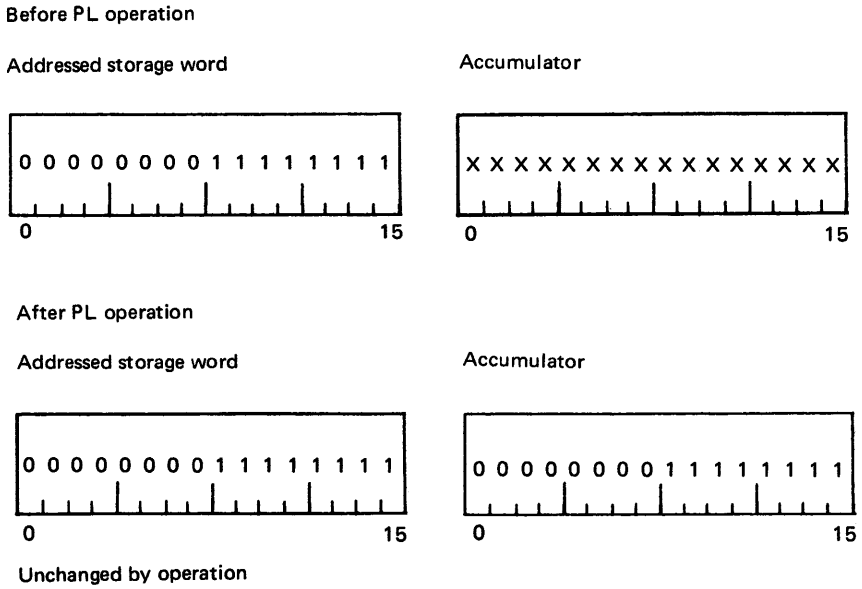
BR0843

The contents of the storage location specified by the effective address replace the contents of the accumulator. The storage contents are unchanged.

An R field of 000 means that the effective address is formed using the IAR (instead of an index register) as a base register.

The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the accumulator.

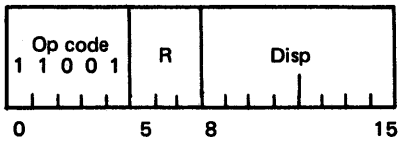
Example:



BR0844

Load and Zero

PLZ—1200 ns



BR0845

The contents of the storage location specified by the effective address replace the contents of the accumulator. The storage contents are set to zero.

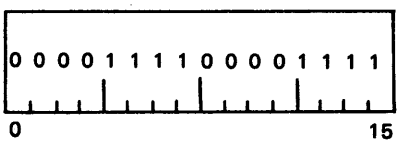
An R field of 000 means that the effective address is formed using the IAR (instead of an index register) as a base register.

The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the accumulator.

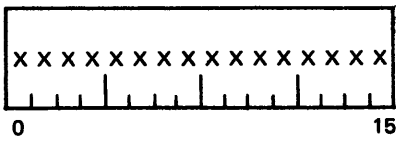
Example:

Before PLZ operation

Addressed storage word

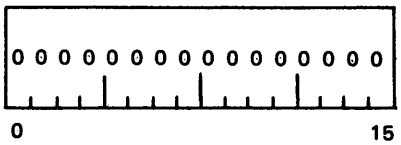


Accumulator

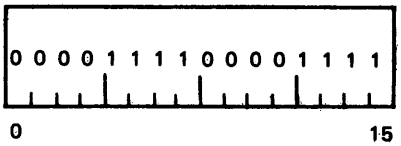


After PLZ operation

Addressed storage word



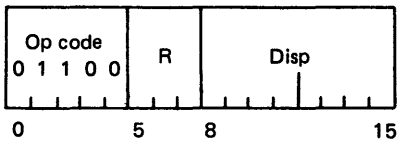
Accumulator



BR0846

Load Immediate

PLI–400 ns



BR0847

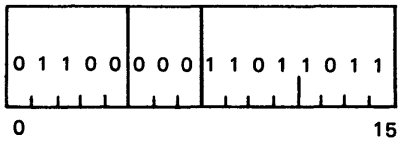
The contents of the index register specified by the contents of R, or the accumulator if R = 000, are replaced by the contents of the displacement field. Before loading, the 8-bit displacement field is expanded to 16 bits by propagating the displacement sign-bit (bit 8) through the high-order bits (i.e., bits 0-7 take on the value of the sign bit). The instruction word is unchanged.

The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the accumulator or index register.

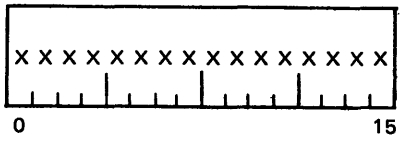
Example:

Before PLI operation

PLI instruction

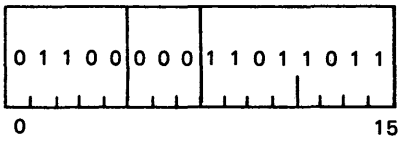


Accumulator

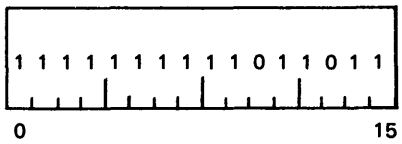


After PLI operation

PLI instruction



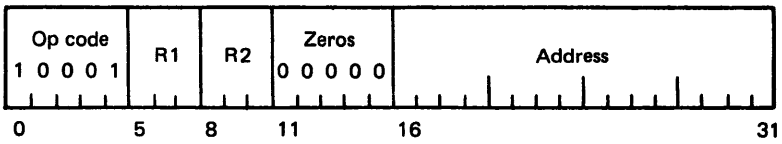
Accumulator



BR0848

Load Index Long

PLXL–800 ns



BR0849

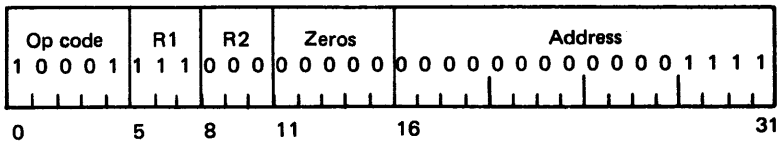
The contents of the storage location specified by the effective address replace the contents of the index register specified by the R1 field. R1 = 000 specifies the accumulator. The storage contents are unchanged.

If R2 = 000, the address field contains the effective address. If R2 ≠ 000, the effective address is the algebraic sum of the contents of the index register (specified by R2) and the address field.

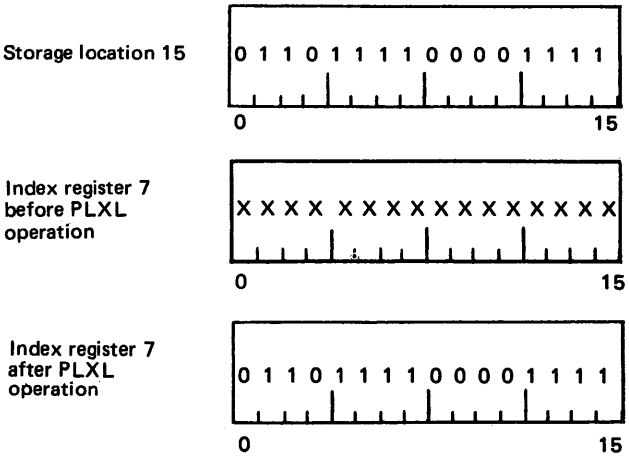
Bits 11-15 of the instruction must be 0's.

The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the accumulator or index register.

Example:



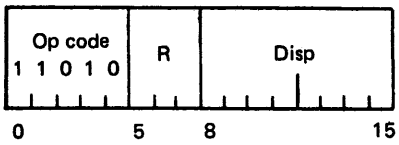
Since R2 = 000, the data is in storage location 15 (value of address field)



BR0850

Store Accumulator

PST—800 ns



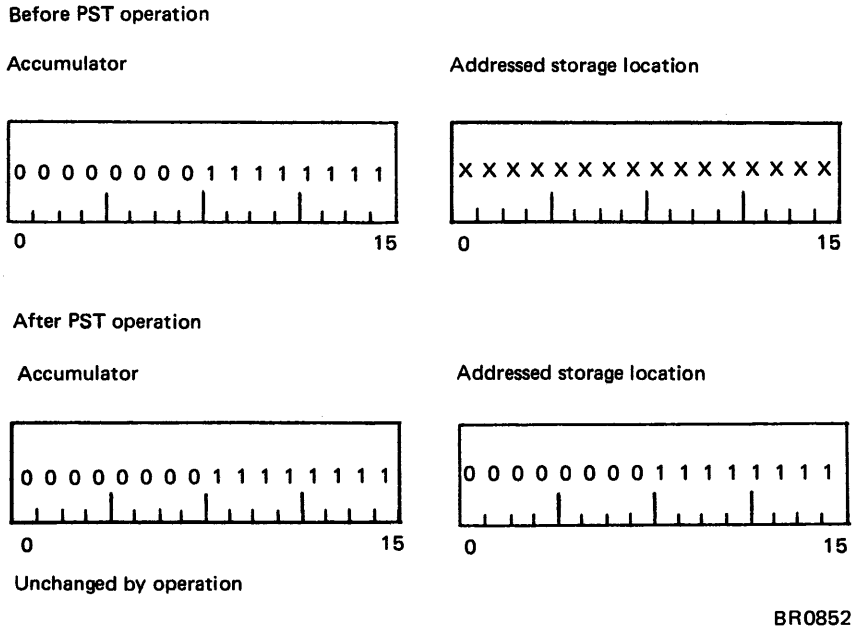
BR0851

The contents of the accumulator replace the contents of the storage location specified by the effective address. The contents of the accumulator are unchanged.

If R = 000, the effective address is formed using the IAR (instead of an index register) as a base register.

The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the storage location.

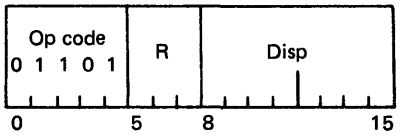
Example:



BR0852

Store Index

PSTX–800 ns



BR0918

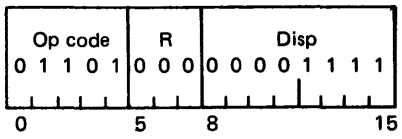
The contents of the index register (R) replace the contents of the storage location specified by the effective address. The index register contents are unchanged. If R = 000, the storage location contents are replaced by 0's.

The effective address of the storage location is always generated using the IAR as the base register.

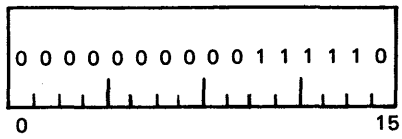
The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the storage location.

Example:

PSTX instruction (assumed to be at storage location 61)

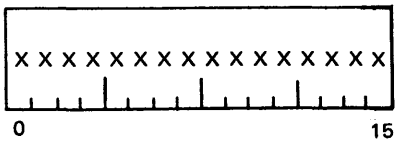


Update IAR (contains address of next instruction at storage location 62)

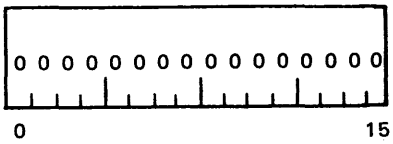


IAR	0 0 0 0 | 0 0 0 0 | 0 0 1 1 | 1 1 1 0	Decimal 62
Displacement (expanded to 16 bits)	+0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1	Decimal 15
Effective address of storage location	0 0 0 0 | 0 0 0 0 | 0 1 0 0 | 1 1 0 1	Decimal 77

Storage location 77 before PSTX operation



Storage location 77 after PSTX operation



Replaced by 0's because R = 000 in the PSTX instruction

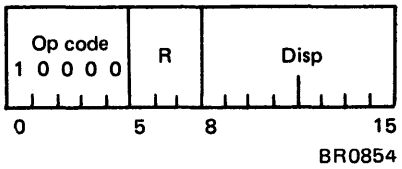
BR0853

ARITHMETIC INSTRUCTIONS

Six instructions are provided to perform addition, subtraction, or complementing of data residing in storage and/or the index registers/accumulator.

Add

PA–800 ns



The contents of the accumulator are added algebraically to the contents of the storage location addressed by the effective address. The result replaces the contents of the accumulator; the contents of the addressed storage location remain unchanged.

If the R field = 000, the effective address of the data in storage is generated by using the IAR as the base register.

The carry, overflow, and result indicators are set to reflect the result in the accumulator. When an overflow occurs, the accumulator contains the correct low-order 16 bits of the sum and the carry indicator contains the sign bit.

The result of the addition is either positive or negative, depending upon the magnitude of the values used and whether the signs of the two operands are the same:

\+ plus a + = +
\- plus a - = -
\+ plus a - = sign of the larger operand
\- plus a + = sign of the larger operand

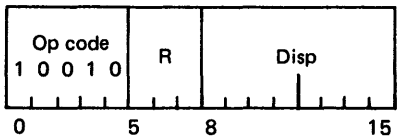
The value in the accumulator is positive if the leftmost bit is a 0; the value in the accumulator is negative if the leftmost bit is a 1. Negative numbers are in twos-complement form.

Example:

```
 0 000 0000 1001 1101  Contents of accumulator
+0 000 0010 0011 0101  Contents of storage location addressed by PA instruction
 0 000 0010 1101 0010  Result in accumulator
```

Subtract

PS–800 ns



BR0855

A 16-bit operand in storage is subtracted from the 16-bit operand in the accumulator. The operand in storage and the operand in the accumulator can have the same or different signs; negative numbers are in twos-complement form. The result of the subtraction is in the accumulator at the end of the operation. The operand addressed in storage by the PS instruction is unchanged by the operation.

An R field of 000 means that the effective address of the data in storage is formed using the IAR as the base register.

The carry indicator is set on if a borrow occurs out of the high-order bit in the accumulator.

Occurrence of an overflow condition sets on the overflow indicator. Under these circumstances, the correct low-order 16 bits of the answer are in the accumulator and the carry indicator contains the correct sign bit.

The result indicators are set to reflect the result in the accumulator.

The sign of the result is dependent upon the signs and magnitudes of both operands. Possible combinations (where operand B is always numerically greater than operand A, regardless of signs) are:

Operand in Accumulator		*Operand in Storage*		*Sign of Result in Accumulator*
+B	-	+A	=	+
+B	-	-A	=	+
-B	-	+A	=	-
-B	-	-A	=	-
+A	-	+B	=	-
+A	-	-B	=	+
-A	-	+B	=	-
-A	-	-B	=	+

Example 1:

```
 0 000 0000 0000 0011  Operand in accumulator
-0 000 0000 0000 0010  Operand addressed by subtract instruction
 0 000 0000 0000 0001  Result in accumulator
```

(In decimal: 3 - 2 = 1)

Example 2:

```
 1 000 0000 0000 0011  Operand in accumulator
-0 000 0000 0000 0010  Operand addressed by subtract instruction
 1 000 0000 0000 0001  Result in accumulator
```

(In decimal: -32,765 - 2 = -32,767)

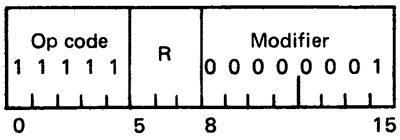
Example 3:

```
 1 000 0000 0000 0011  Operand in accumulator
-1 000 0000 0000 0000  Operand addressed by subtract instruction
 0 000 0000 0000 0011  Result in accumulator
```

(In decimal: -32,765 - (-)32,768 = -32,765 + 32,768 = +3)

Add Register

PAR–400 ns



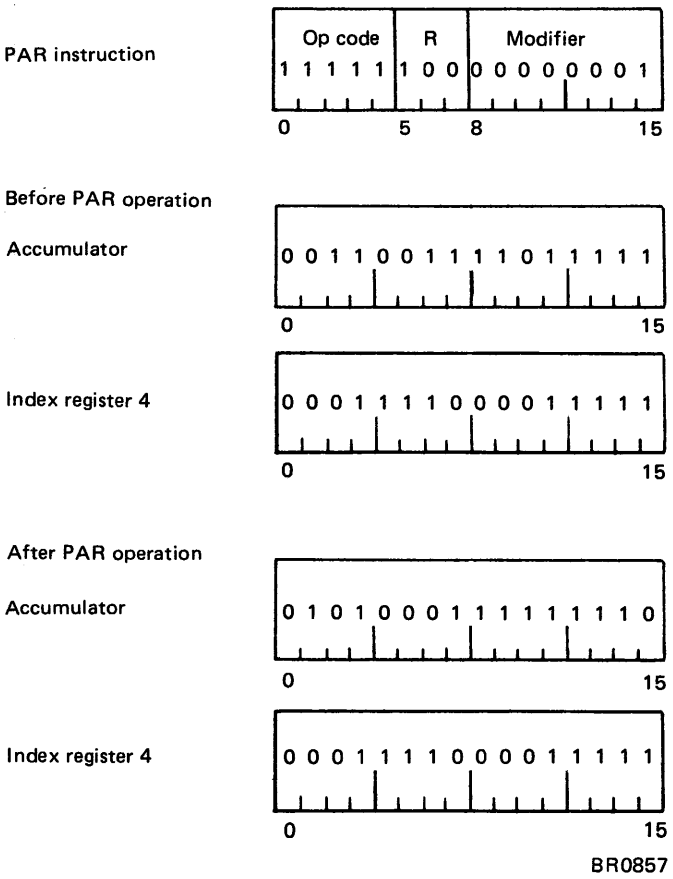
BR0856

The contents of the index register (R) are added algebraically to the contents of the accumulator. The resulting sum replaces the contents of the accumulator. The index register contents are unchanged.

An R field of 000 specifies the accumulator. This means that the accumulator can be added to itself, giving a simple method of doubling the value of the accumulator contents.

The carry, overflow, and result indicators are set to reflect the result sum in the accumulator. When an overflow condition occurs, the accumulator contains the correct low-order 16 bits of the sum while the carry indicator contains the correct sign bit.

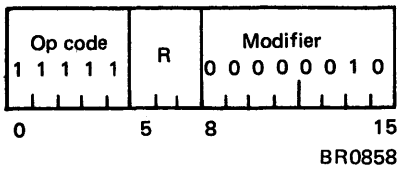
Example:



BR0857

Subtract Register

PSR—400 ns



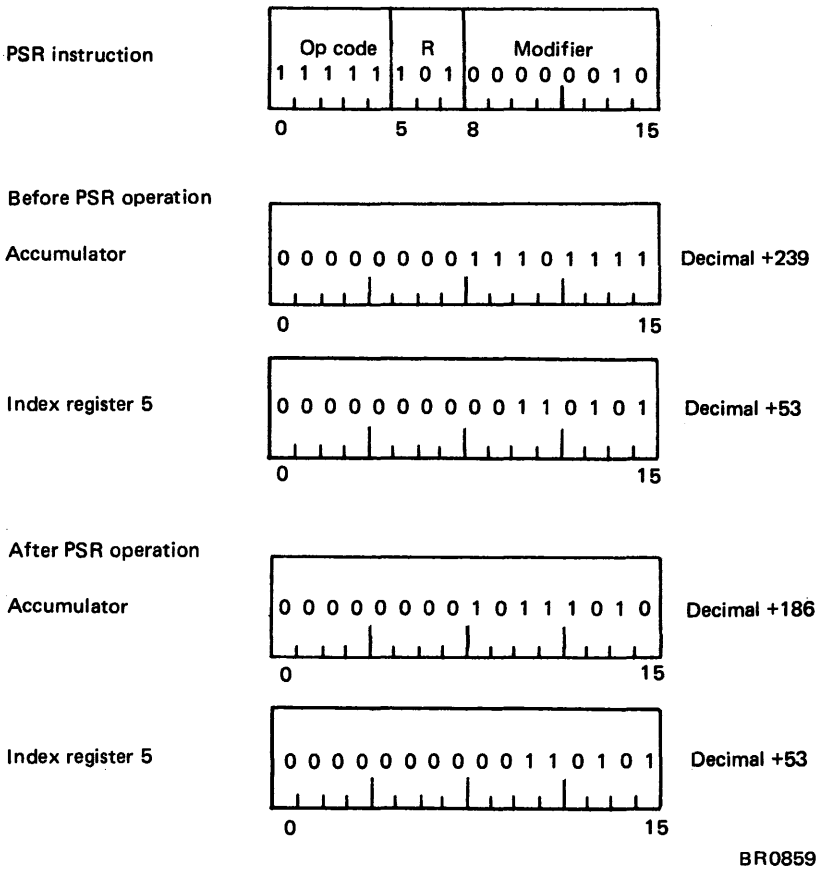
BR0858

The contents of the index register (R) are subtracted algebraically from the contents of the accumulator. The resulting difference replaces the contents of the accumulator. The index register contents are unchanged.

An R field of 000 specifies the accumulator. This means that the accumulator can be subtracted from itself, giving a simple method of setting the accumulator contents to a zero value.

The carry, overflow, and result indicators are set to reflect the result in the accumulator. The carry indicator is set on by a borrow out of the high-order bit in the accumulator. When an overflow condition occurs, the accumulator contains the correct low-order 16 bits of the result and the carry indicator contains the correct sign bit.

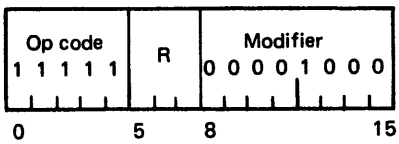
Example:



BR0859

Complement Register

PCR–400 ns



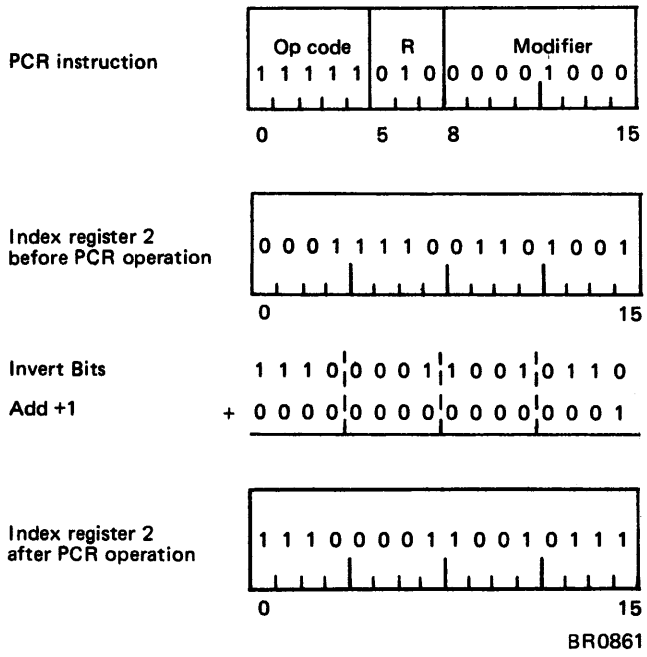
BR0860

The contents of the index register (R) are twos complemented. The result replaces the contents of the register specified by R. An R field of 000 specifies the accumulator.

The carry indicator is not changed. The overflow indicator is set on if the number to be complemented is the maximum negative number representable; i.e., a 1 followed by fifteen 0's. The result indicators are changed to reflect the final contents of the accumulator or index register.

The twos complement of a number is obtained by inverting each bit of the number and then adding a binary 1 to the result.

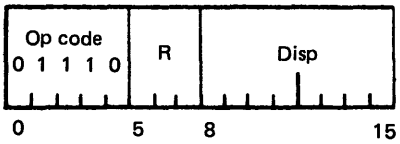
Example:



BR0861

Add Immediate

PAI—400 ns

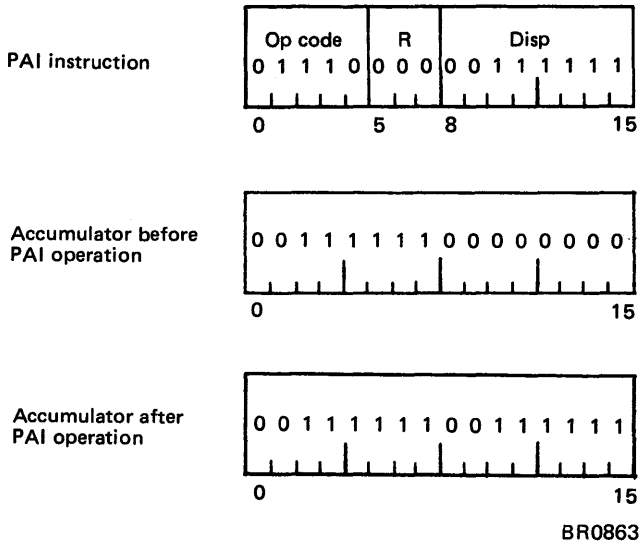


BR0862

The contents of the displacement field in the instruction are added algebraically to the contents of the index register (R), or the accumulator if R = 000. Before the addition takes place, the 8-bit displacement field is expanded to 16 bits by propagating the displacement sign-bit (bit 8 of the instruction word) value through the high-order bits (i.e., bits 0-7 take on the value of the sign bit). The resulting sum replaces the contents of the index register or accumulator. The instruction word is unchanged.

The carry, overflow, and result indicators are set to reflect the result in the index register or accumulator. When an overflow condition occurs, the index register or accumulator contains the correct low-order 16 bits of the result and the carry indicator contains the correct sign bit.

Example:



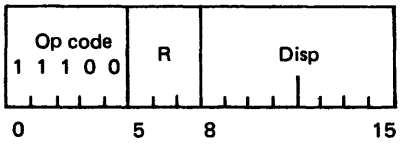
BR0863

LOGICAL INSTRUCTIONS

Six instructions are provided to perform logical operations with data in storage and/or the index registers/accumulator. Two sets of data are used in order to produce a logical result.

AND

PN–800 ns



The contents of the accumulator are ANDed, bit by bit, with the contents of the storage location addressed by the effective address. The result replaces the contents of the accumulator. The contents of the addressed storage word are not changed as a result of the operation.

An R field of 000 means that the effective address of the storage data is generated using the IAR as the base register.

The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the accumulator.

ANDing occurs only between corresponding bits in the accumulator and the storage word: bit 0 is ANDed only with bit 0, bit 1 only with bit 1, and so on. The four possible ANDing results are:

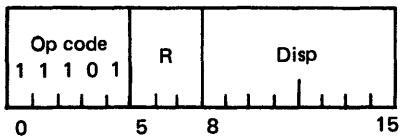
Bit Value from Storage Word	*Bit Value from Accumulator*	*Result in Accumulator*
0	0	0
0	1	0
1	0	0
1	1	1

Example:

```
0101 0000 1111 1010  Word in accumulator
1010 1111 1010 1111  Word from storage
0000 0000 1010 1010  Result in accumulator
```

OR

PO–800 ns



The contents of the accumulator are ORed, bit by bit, with the contents of the storage location addressed by the effective address. The result replaces the contents of the accumulator. The contents of the addressed storage word are not changed as a result of the operation.

An R field of 000 means that the effective address of the storage data is generated using the IAR as the base register.

The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the accumulator.

ORing occurs only between corresponding bits in the accumulator and the storage word: bit 0 is ORed only with bit 0, bit 1 only with bit 1, and so on. The four possible ORing results are:

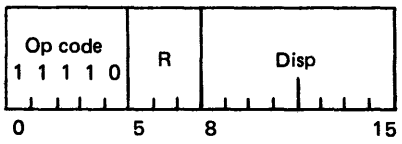
Bit Value from Storage Word	*Bit Value from Accumulator*	*Result in Accumulator*
0	0	0
0	1	1
1	0	1
1	1	1

Example:

```
0011 0101 1111 1010  Word in accumulator
0101 0001 1010 0000  Word from storage
0111 0101 1111 1010  Result in accumulator
```

Exclusive OR

PX–800 ns



The contents of the accumulator are exclusive ORed, bit by bit, with the contents of the storage location addressed by the effective address. The result replaces the contents of the accumulator. The contents of the addressed storage word are not changed as a result of the operation.

An R field of 000 means that the effective address of the storage data is generated using the IAR as the base register.

The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the accumulator.

Exclusive ORing occurs only between corresponding bits in the accumulator and the storage word: bit 0 is exclusive ORed only with bit 0, bit 1 only with bit 1, and so on. The four possible exclusive-ORing results are:

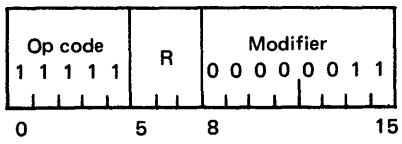
Bit Value from Storage Word	*Bit Value from Accumulator*	*Result in Accumulator*
0	0	0
0	1	1
1	0	1
1	1	0

Example:

```
0000 1111 0000 1111  Word in accumulator
1111 0000 0000 1111  Word from storage
-------------------
1111 1111 0000 0000  Result in accumulator
```

AND Register

PNR–400 ns



BR0867

The contents of the accumulator are ANDed, bit by bit, with the contents of the index register (R). The result replaces the contents of the accumulator. The contents of the addressed index register are not changed as a result of the operation.

An R field of 000, specifying the accumulator, is, in effect, a no-operation because the final contents of the accumulator are unchanged from the original contents. Any data ANDed with itself has a final result equal to the original data.

The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the accumulator.

ANDing occurs only between corresponding bits in the accumulator and the index register: bit 0 is ANDed only with bit 0, bit 1 only with bit 1, and so on. The four possible ANDing results are:

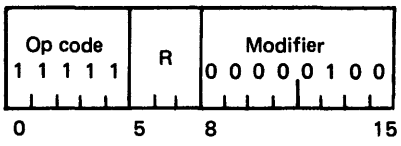
Bit Value from Index Register	*Bit Value from Accumulator*	*Result in Accumulator*
0	0	0
0	1	0
1	0	0
1	1	1

Example:

```
0101 0000 1111 1010  Word in accumulator
1010 1111 1010 1111  Word in index register
0000 0000 1010 1010  Result in accumulator
```

OR Register

POR–400 ns



The contents of the accumulator are ORed, bit by bit, with the contents of the index register (R). The result replaces the contents of the accumulator. The contents of the addressed index register are not changed as a result of the operation.

An R field of 000, specifying the accumulator, is, in effect, a no-operation because the final contents of the accumulator are unchanged from the original contents. Any data ORed with itself has a final result equal to the original data.

The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the accumulator.

ORing occurs only between corresponding bits in the accumulator and the index register: bit 0 is ORed only with bit 0, bit 1 only with bit 1, and so on. The four possible ORing results are:

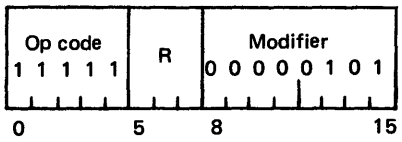
Bit Value from Index Register	*Bit Value from Accumulator*	*Result in Accumulator*
0	0	0
0	1	1
1	0	1
1	1	1

Example:

```
0011 0101 1111 1010 Word in accumulator
0101 0001 1010 0000 Word in index register
---------------------------------------
0111 0101 1111 1010 Result in accumulator
```

Exclusive OR Register

PXR–400 ns



The contents of the accumulator are exclusive ORed, bit by bit, with the contents of the index register (R). The result replaces the contents of the accumulator. The contents of the addressed index register are not changed as a result of the operation.

An R field of 000, which specifies the accumulator, is a simple method of clearing the accumulator. Any data that is exclusive ORed with itself results in setting all the bits to 0's.

The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the accumulator.

Exclusive ORing occurs only between corresponding bits in the accumulator and the index register: bit 0 is exclusive ORed only with bit 0, bit 1 only with bit 1, and so on. The four possible exclusive-ORing results are:

Bit Value from Index Register	*Bit Value from Accumulator*	*Result in Accumulator*
0	0	0
0	1	1
1	0	1
1	1	0

Example:

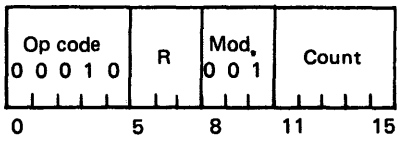
```
0000 1111 0000 1111  Word in accumulator
1111 0000 0000 1111  Word in index register
-------------------
1111 1111 0000 0000  Result in accumulator
```

SHIFTING INSTRUCTIONS

Four instructions are provided to displace, to either the left or the right, data residing in an index register or the accumulator.

Shift Left Logical

PSLL—400 ns + 50 times the number of bits shifted



BR0870

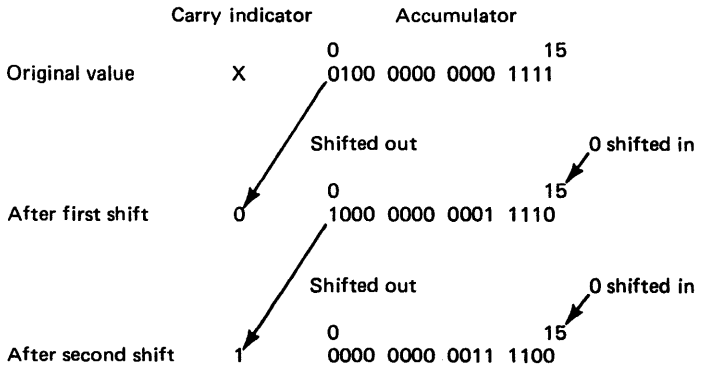
All 16 bits in the index register (R), or the accumulator if R = 000, are shifted left by the number of bits specified in the count field.

All bits shifted out of the high-order bit (bit 0) are lost, except the last one which sets the carry indicator. Vacated low-order bits are set to 0's.

The shift count field can have any decimal value from 0-16. Shift counts greater than 16 are invalid and cause a program-check interruption. A shift count of zero is valid and serves a useful purpose. Although no shifting takes place, this is a simple method of setting the result indicators to reflect the current contents of R. A shift count of 16 puts 0's in all bits of the R register.

The overflow indicator is not changed. The carry indicator is set on or off by the last bit shifted out of bit 0. The result indicators are changed to reflect the final contents of the R register. For a shift count greater than zero, the result-even indicator is always set on, because vacated bits are automatically set to 0's.

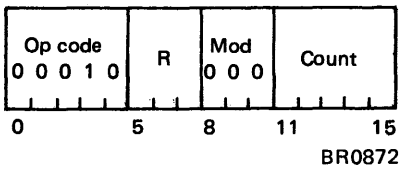
Example: Assume a left shift of two bits in the accumulator.



BR0871

Shift Left Circular

PSLC—400 ns + 50 times the number of bits rotated

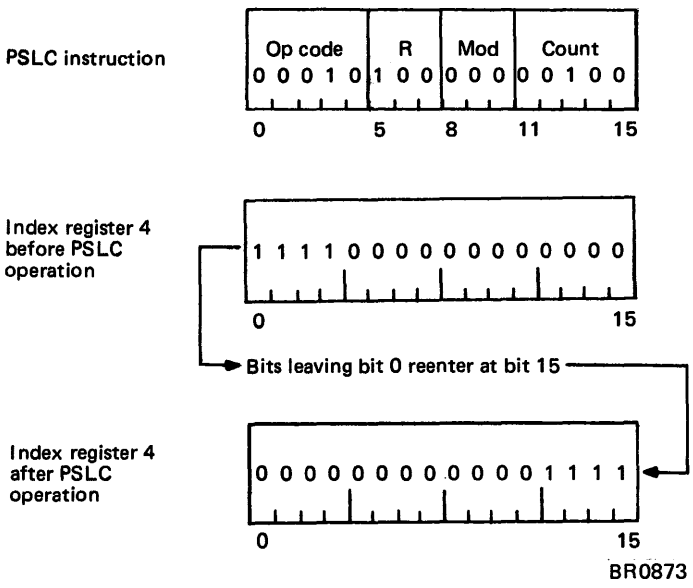


All 16 bits in the index register (R), or the accumulator if R = 000, are rotated left by the number of bits specified in the count field. Rotated left means that bits shifted out of the high-order bit (bit 0) reenter at the low-order bit (bit 15). Therefore, no bits are lost, just rearranged.

The shift count field can have any decimal value from 0-16. Shift counts greater than 16 are invalid and cause a program-check interruption. A shift count of zero is valid and serves a useful purpose. Although no rotating takes place, this is a simple method of setting the result indicators to reflect the current contents of the R register. A shift count of 16 has the same effect because the bits are rotated until they return to their original setting.

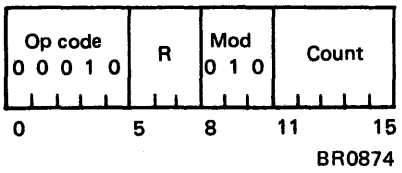
The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the R register.

Example: Assume a left circular shift of four bits in index register 4.



Shift Right Logical

PSRL—400 ns + 50 times the number of bits shifted



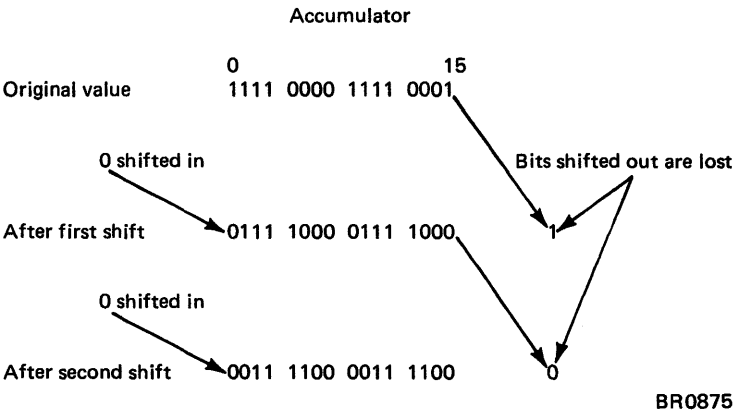
All 16 bits in the index register (R), or the accumulator if R = 000, are shifted right by the number of bits specified in the count field.

Vacated high-order bits are set to 0's. All bits shifted out of the low-order bit (bit 15) are lost.

The shift count field can have any decimal value from 0-16. Shift counts greater than 16 are invalid and cause a program-check interruption. A shift count of 16 sets the entire contents of the R register to a zero value. A shift count of zero is valid and serves a useful purpose. Although no shifting takes place, this is a simple method of setting the result indicators to reflect the current contents of the R register.

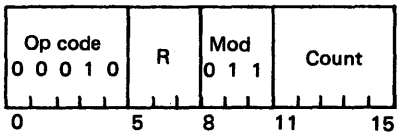
The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the R register. For a shift count greater than zero, the result-positive indicator is always set on.

Example: Assume a right shift of two bits in the accumulator.



Shift Right Arithmetic

PSRA—400 ns + 50 times the number of bits shifted



BR0876

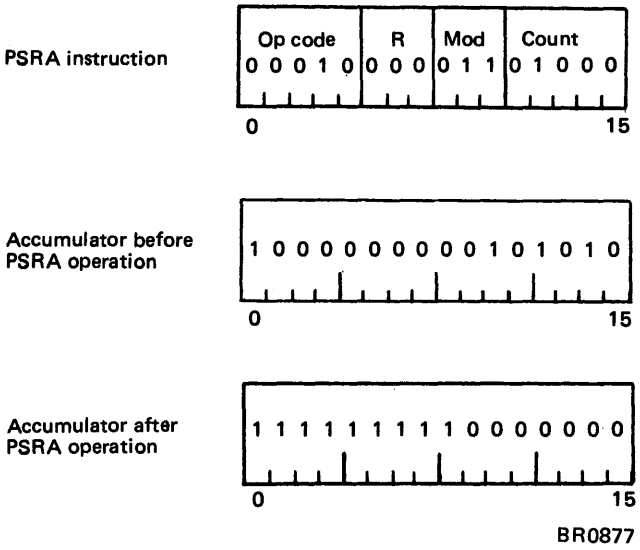
All 16 bits in the index register (R), or the accumulator if R = 000, are shifted right by the number of bits specified in the count field.

Vacated high-order bits are set to the value of the sign bit (bit 0). All bits shifted out of the low-order bit (bit 15) are lost.

The shift count field can have any decimal value from 0-16. Shift counts greater than 16 are invalid and cause a program-check interruption. A shift count of zero is valid and serves a useful purpose. Although no shifting takes place, this is a simple method of setting the result indicators to reflect the current contents of the R register.

The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the R register.

Example: Assume a right arithmetic shift of eight bits in the accumulator.



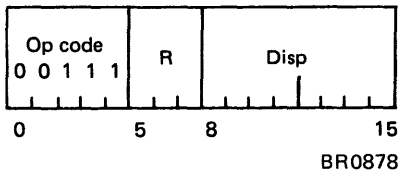
BR0877

BRANCHING INSTRUCTIONS

Seven instructions are provided to alter the sequence of instruction execution in a program. Branches can occur unconditionally or based upon the results of a test.

Branch

PB–400 ns



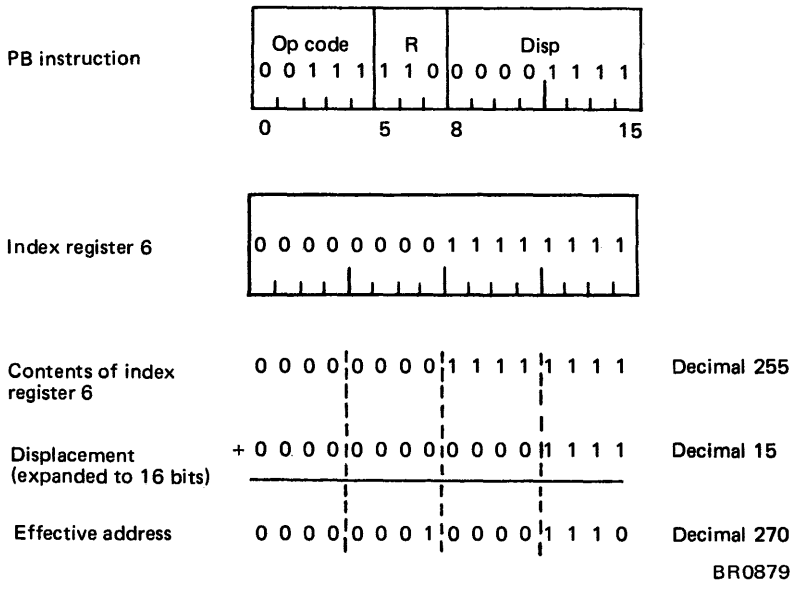
The execution of a sequential series of instructions is terminated by branching to another location in storage. The PB instruction is unconditional; i.e., the branch is always taken because it is not based on the result of some condition or test.

Location of the next instruction to be executed (the location branched to) is determined by the effective address as computed from the register and displacement fields in the PB instruction word. The contents of the displacement field are added algebraically to the contents of the specified index register (R) to form this effective address. If R = 000, the IAR is used in this calculation.

Since the IAR always contains the address of the next instruction to be executed, the PB instruction causes the calculated effective address to replace the contents of the IAR.

The PB instruction does not change the carry, overflow, and result indicators.

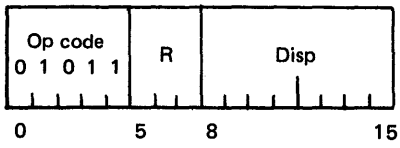
Example:



Thus, the next instruction to be executed is at storage location 270.

Branch and Link

PBAL–400 ns



BR0880

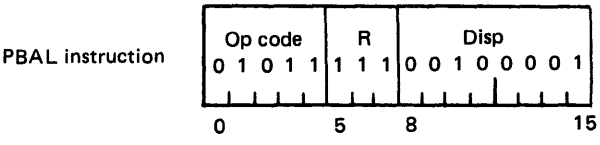
When executing a sequential series of instructions, it is sometimes necessary to branch to another storage location and then return to the original sequence. The PBAL instruction provides this capability. It is an unconditional instruction; i.e., the branch is always taken because it is not based upon the result of some condition or test.

The IAR always contains the address of the next sequential instruction. In this case, that is the address of the instruction following that of the PBAL instruction. Execution of the PBAL instruction causes the contents of the IAR to be stored (thus saved for future use in returning to the original sequence) in the index register specified by the R field. If R = 000, the accumulator is used for this purpose.

The location of the next instruction to be executed (the location branched to) is determined by the effective address derived by adding algebraically the contents of the displacement field in the instruction word to the IAR contents. Thus, the IAR contains the effective address of the instruction being branched to.

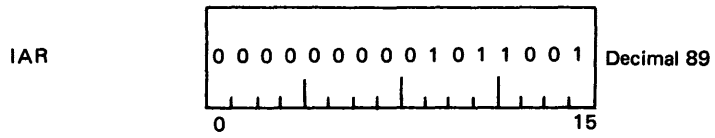
The PBAL instruction does not change the carry, overflow, and result indicators.

Example: Assume that the PBAL instruction is at storage location 88.



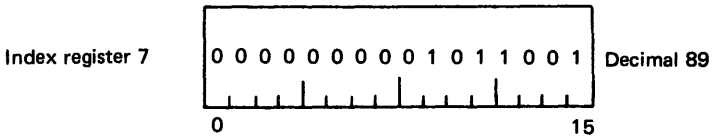
BR0881

When execution of the PBAL instruction begins, the IAR contains the address (89) of the next sequential instruction.



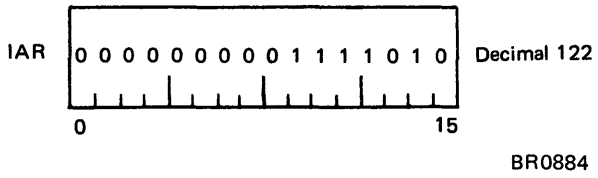
BR0882

This address is stored in index register 7.



BR0883

The displacement field is added to the IAR with this result.

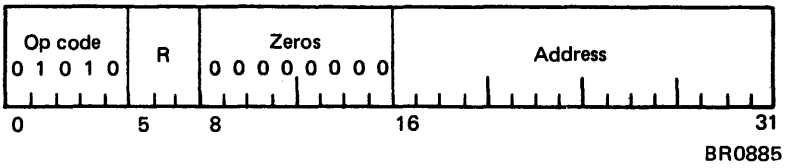


The IAR contains the effective address of the next instruction to be executed; i.e., the instruction branched to, which is at location 122. The branch to location 122 is taken and execution begins.

When it is desired to return to the original sequence (at location 89), the contents of index register 7 must be moved to the IAR. A branch (PB) instruction, with zero displacement and R = 111 (specifying XR7), can be used to accomplish this.

Branch and Link Long

PBALL–800 ns



When executing a sequential series of instructions, it is sometimes necessary to branch to another storage location and then return to the original sequence. The PBALL instruction provides this capability. It is an unconditional instruction; i.e., the branch is always taken because it is not based upon the result of some condition or test.

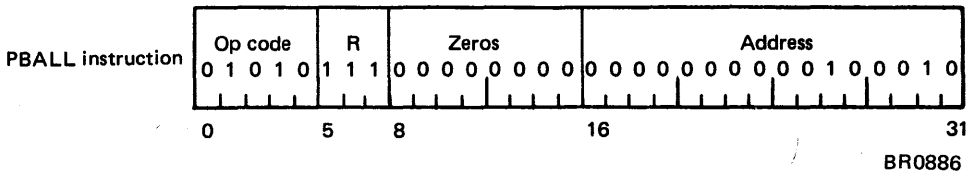
The IAR always contains the address of the next sequential instruction. In this case, this is the address of the instruction following that of the PBALL instruction. Execution of the PBALL instruction causes the contents of the IAR to be stored (thus saved for future use in returning to the original sequence) in the index register specified by the R field. If R = 000, the accumulator is used for this purpose.

The address field in the instruction contains the effective address of the location to be branched to. This effective address is loaded into the IAR and the branch is made to that address.

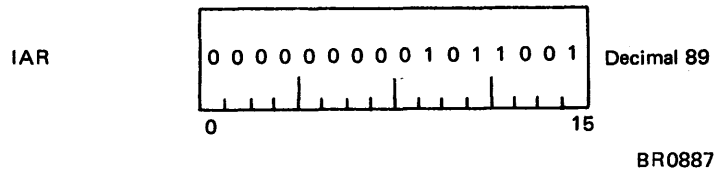
Bits 8-15 of the instruction are always set to 0's.

The branch and link long instruction does not change the carry, overflow, and result indicators.

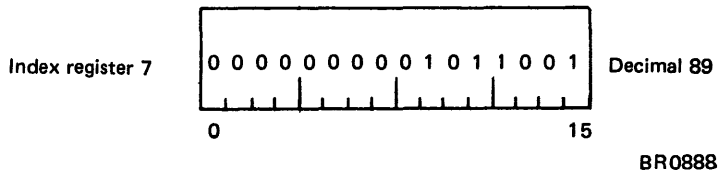
Example: Assume that the PBALL instruction is at storage locations 87 and 88.



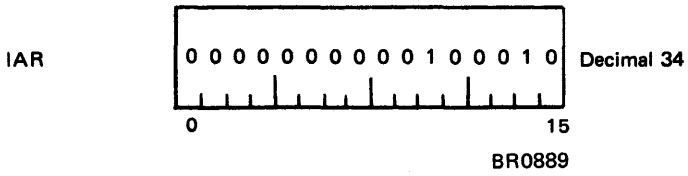
When execution of the PBALL instruction begins, the IAR contains the address (89) of the next sequential instruction.



This address is stored in index register 7.



The address field is loaded into the IAR.

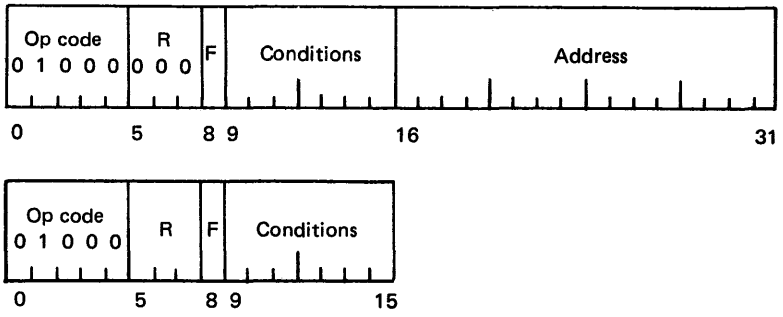


The IAR contains the effective address of the next instruction to be executed; i.e., the instruction branched to, which is at location 34. The branch to location 34 is taken and execution begins.

When it is desired to return to the original sequence (at location 89), the contents of index register 7 must be moved to the IAR. A branch (PB) instruction, with zero displacement and R = 111 (specifying XR7), can be used to accomplish this.

Branch on Condition

PBC—800 ns if branch is taken; otherwise, 400 ns



BR0890

When executing a sequential series of instructions, the PBC instruction provides the facility to branch and begin executing at another storage location. The PBC instruction is conditional in that the result of a test determines whether or not the branch is taken.

There are two ways of determining the effective address of the storage location that is to be branched to. When R equals 000, the effective address is in the address field of the instruction. When R does not equal 000, the effective address is contained in the specified index register. In either case, if test results cause the branch to be taken, the effective address is loaded into the IAR.

The programmer determines what tests are conducted to decide whether or not a branch is taken. This is done by setting bits on or off in the condition field of the instruction. The bits and their meanings are:

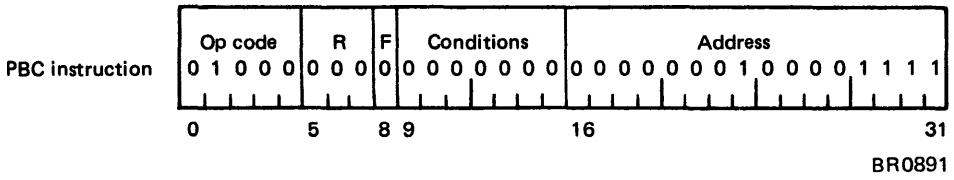
Bit 9—Carry and overflow indicators both off
Bit 10—Result-zero indicator on
Bit 11—Result-negative indicator on
Bit 12—Result-positive indicator on
Bit 13—Result-even indicator on
Bit 14—Carry indicator off
Bit 15—Overflow indicator off

Setting a condition field bit to 1 requests that the state of its associated indicator be tested. If no testing is requested (all condition field bits are set to 0's), the PBC functions as an unconditional branch. Any combination of testing can be requested.

If none of the specified test conditions is true, the branch is taken. If any are true, the branch does not occur and program execution proceeds with the next sequential instruction.

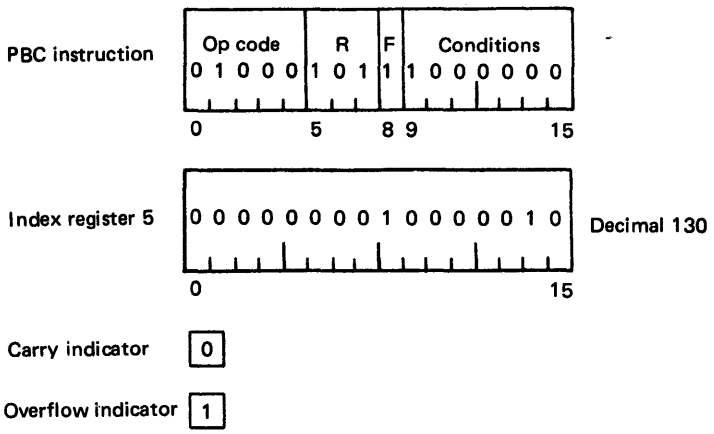
The branch on condition instruction does not change the carry and result indicators. Testing of the overflow indicator always sets it off unless bit 8 (overflow save flag) of the instruction is set on.

Example 1:



No testing is requested since all condition field bits are 0's. Because R = 000, the effective address is in the address field. Therefore, this instruction is an unconditional branch to storage location 271.

Example 2:

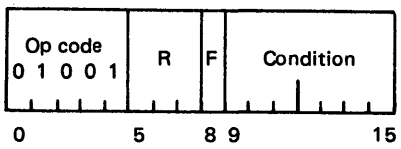


The only test requested (bit 9 set on) is that both the carry and overflow indicators are off. This is not true because the overflow indicator is on. Therefore, this instruction results in a branch to location 130, the effective address in index register 5. The overflow indicator remains on because the overflow save flag, in bit 8 of the instruction, is on.

Example 3: In example 2, if the overflow indicator is off, the test requested is true. In that case, no branch occurs. Program execution proceeds with the next sequential instruction following the short-format branch on condition instruction.

Skip on Condition

PSKC—400 ns



BR0893

The next word in the program sequence is skipped over (not executed). Since only one word is skipped, this instruction should never be followed by a long-format instruction.

The PSKC instruction is conditional in that the result of a test determines whether or not the skip is made. The programmer determines what tests are conducted. This is done by setting bits on or off in the condition field of the instruction. The bits and their meanings are:

Bit 9—Carry and overflow indicators both off
Bit 10—Result-zero indicator on
Bit 11—Result-negative indicator on
Bit 12—Result-positive indicator on
Bit 13—Result-even indicator on
Bit 14—Carry indicator off
Bit 15—Overflow indicator off

Setting a condition field bit to a 1 requests that the state of its associated indicator be tested. If no testing is requested (all condition field bits are set to 0's), no skip occurs. Any combination of testing can be requested.

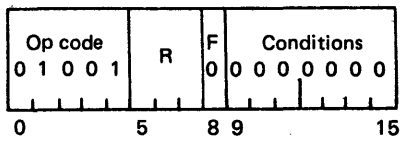
If none of the specified test conditions is true, no skip occurs. If any are true, the skip takes place by incrementing the IAR by one before the next instruction is fetched.

The R field is ignored in this instruction.

The PSKC instruction does not change the carry and result indicators. Testing of the overflow indicator always sets it off unless bit 8 (overflow save flag) of the instruction is set on.

Example 1:

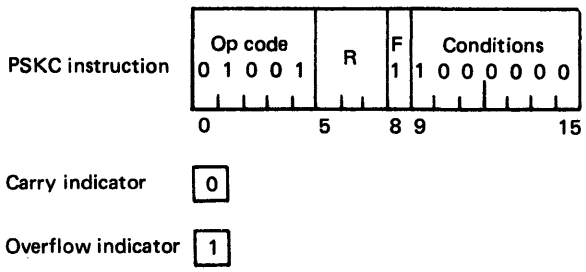
PSKC instruction



BR0894

No testing is requested since all condition field bits are 0's. No skip occurs.

Example 2:

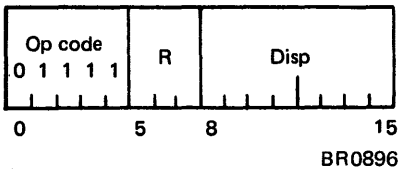


The only test requested (bit 9 set on) is that both the carry and overflow indicators are off. Since the overflow indicator is on, the test is false and no skip occurs. The overflow indicator remains on because the overflow save flag, bit 8 of the instruction, is on.

Example 3: In example 2, if the overflow indicator is off, the test requested is true and the next word is skipped.

Add to Storage and Skip

PAS—1200 ns



The contents of the storage location specified by the effective address are incremented by one. If the result is zero, the next sequential word is skipped. Since only one word is skipped, this instruction should never be followed by a long-format instruction.

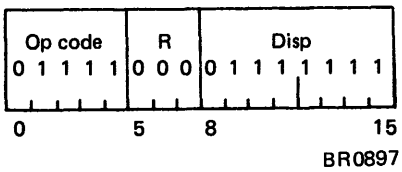
The effective address is generated by adding the contents of the displacement field to those of the index register (R). If R = 000, the IAR contents are used in this calculation.

The carry and overflow indicators are not changed. The result indicators are changed to reflect the final contents of the addressed storage location.

In writing a program, it can be desirable to set up a counter field (initialized to some negative value) that is to be incremented each time a particular job, function, or routine is accomplished. The counter value becoming zero can be the condition for termination and a branch to another portion of the program.

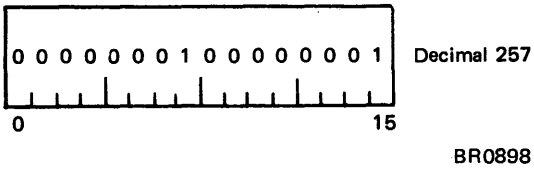
Example: Assume that the PAS instruction is at address 256.

PAS instruction

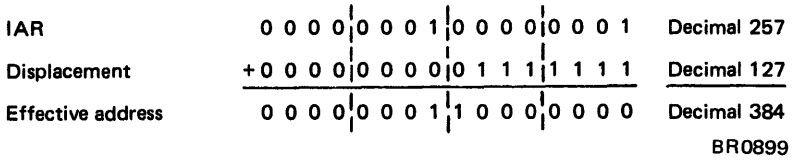


When the PAS instruction is executed, the IAR holds the address of the next sequential instruction.

IAR

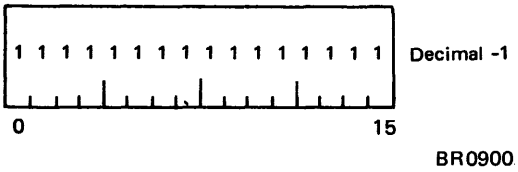


The effective address of the storage location (counter) that is to be incremented is found by adding.

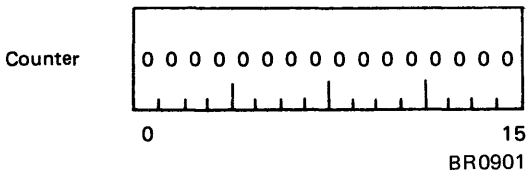


At location 384, assume that the counter has a value of -1 (in twos-complement form).

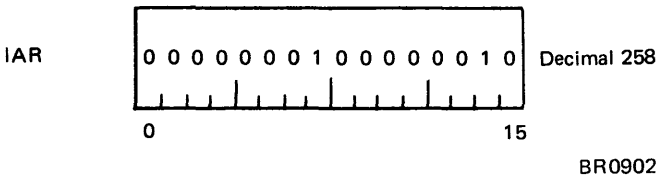
Counter



The PAS instruction increments the counter by +1, with the following result:



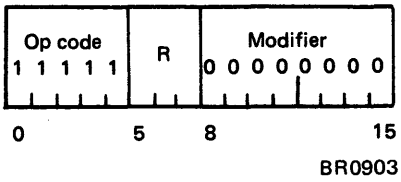
The result-zero and result-even indicators are set on because of the zero value. The PAS instruction tests the result-zero indicator, finds it on, and thus increments the IAR by one.



Upon completion of the PAS, the next instruction fetched and executed is at location 258. The instruction at location 257 is skipped.

No Operation

PNOP–400 ns



No operation is performed. The value in the R field is ignored.

This instruction is inserted into the program to reserve space for future instructions, or to overlay and prevent the operation of unwanted instructions during program debugging.

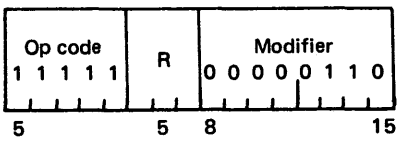
The PNOP instruction does not change the carry, overflow, and result indicators.

REGISTER-TO-REGISTER INSTRUCTIONS

Eight instructions are provided to move data between the index registers, control registers, and accumulator.

Store to Register

PSTR—400 ns



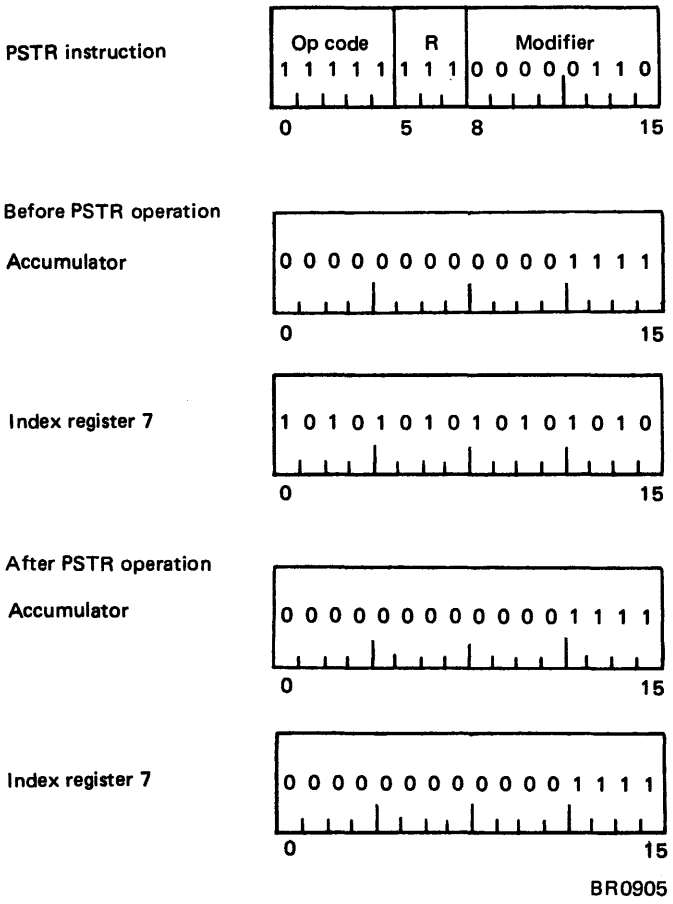
BR0904

The accumulator contents replace the contents of the index register specified by the R field. The contents of the accumulator are unchanged.

If R = 000, the accumulator contents replace those in the IAR, thus providing a branch to the address specified by the accumulator contents.

The PSTR instruction does not change the carry and overflow indicators. The result indicators are changed to reflect the final contents of the R register.

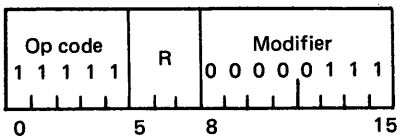
Example:



BR0905

Load from Register

PLR–400 ns

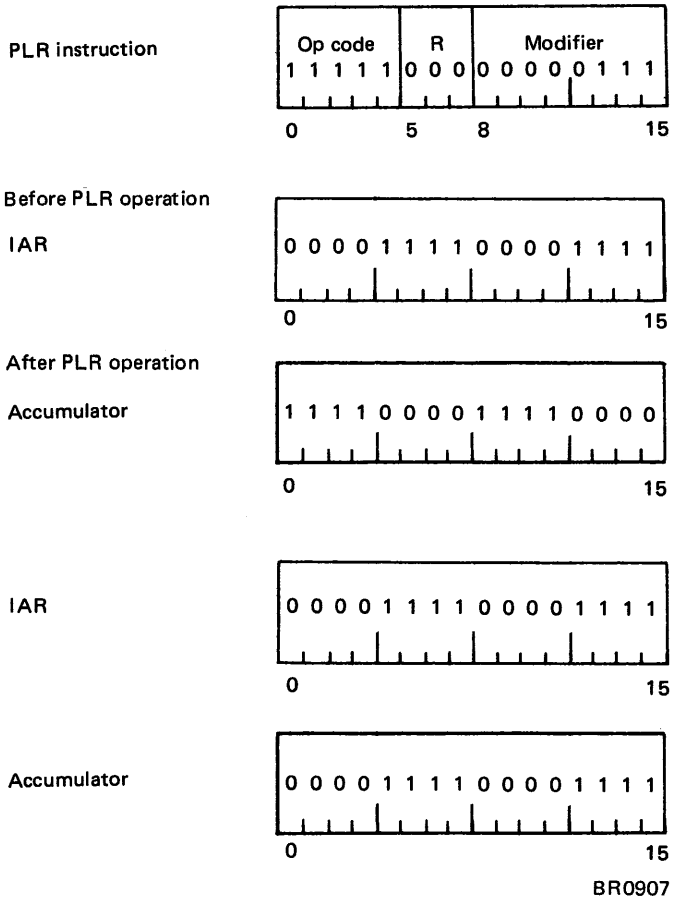


BR0906

The contents of the index register specified by the R field replace the accumulator contents. The contents of the index register are unchanged. If R = 000, the contents of the IAR replace the contents of the accumulator.

The PLR instruction does not change the carry and overflow indicators. The result indicators are changed to reflect the final contents of the accumulator.

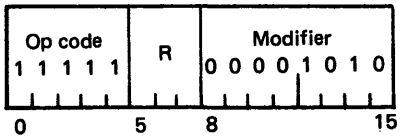
Example:



BR0907

Interchange Register

PIR–400 ns



BR0908

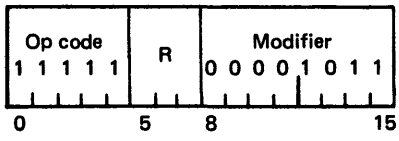
The contents of the accumulator and the register specified by the R field are interchanged. The initial contents of the accumulator are placed in the specified register, and the initial contents of the register replace the accumulator contents.

An R value of 000 in the instruction specifies the accumulator. This results in a no-operation because the contents of the accumulator are interchanged with themselves and, effectively, no change takes place.

The PIR instruction does not change the carry and overflow indicators. The result indicators are changed to reflect the final contents of the accumulator.

Load Processor Status

PLPS–400 ns



BR0909

Sixteen bits of processor status information are stored into the index register (or accumulator if R = 000) specified by the R field. This information has the format:

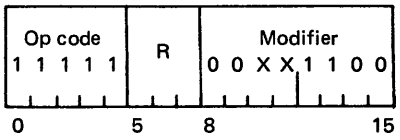
Bit	*Indicator*
0	Invalid shift count
1	Invalid storage address
2	Invalid op-code
3	Local storage parity check
4	SDR parity check
5	Control check
6	I/O check
7	Sequence indicator
8	Power warning
9	Thermal warning
10-15	Must be 0's

Execution of this instruction resets the processor status indicators, except for bits 8 and 9. A system reset does not reset any of these indicators.

This instruction does not change the carry and overflow indicators. The result indicators are changed to reflect the contents of the R register.

Inspect IAR Backup

PIIB–400 ns



BR0910

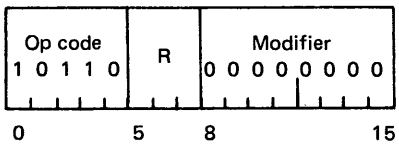
The contents of an IAR (associated with a particular priority level) are stored into the index register (or accumulator if R = 000) specified by the R field in the instruction. The contents of the selected IAR are unchanged.

The IAR is selected by the binary encoded level value in bits 10 and 11 of the instruction.

This instruction does not change the carry and overflow indicators. The result indicators are changed to reflect the contents of the R register.

AND to Mask

PNM–400 ns



BR0911

The four-bit interruption mask register is ANDed, bit by bit, with bits 0-3 of the index register specified by the R field. The result replaces the contents of the interruption mask register, where a bit value of 1 means that an interruption is permitted.

The contents of the index register are not changed as a result of the operation. If R = 000, bits 0-3 of the accumulator are ANDed with the interruption mask register.

The PNM instruction does not change the carry, overflow, and result indicators.

ANDing occurs only between relative corresponding bits in the interruption mask register and the R register: bit 0 is ANDed only with bit 0, bit 1 only with bit 1, and so on. The four possible ANDing results are:

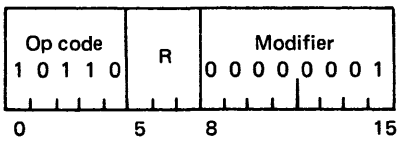
Bit Value from R Register	*Bit Value from Interruption Mask Register*	*Result in Interruption Mask Register*
0	0	0
0	1	0
1	0	0
1	1	1

Example:

1101 Register bits 0-3
0111 Interruption mask register
0101 Result in interruption mask register

OR to Mask

POM–400 ns



BR0912

The four-bit interruption mask register is ORed, bit by bit, with bits 0-3 of the index register specified by the R field. The result replaces the contents of the interruption mask register, where a bit value of 1 means that an interruption is permitted.

The contents of the index register are not changed as a result of the operation. If R = 000, bits 0-3 of the accumulator are ORed with the interruption mask register.

The POM instruction does not change the carry, overflow, and result indicators.

ORing occurs only between relative corresponding bits in the interruption mask register and the R register: bit 0 is ORed only with bit 0, bit 1 only with bit 1, and so on. The four possible ORing results are:

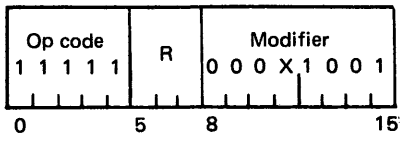
Bit Value from R Register	*Bit Value from Interruption Mask Register*	*Result in Interruption Mask Register*
0	0	0
0	1	1
1	0	1
1	1	1

Example:

1101 Register bits 0-3
0101 Interruption mask register
1101 Result in interruption mask register

Sense Level and Mask

PSLM–400 ns



BR0913

The currently active interruption level designation and the contents of the interruption mask register are stored in the index register specified by the R field.

The interruption level is stored as a binary number in bits 12-15 of the register; the mask is stored intact in bits 0-3. Bits 4-11 are set to 0's.

If R = 000, the interruption level and mask are stored in the accumulator.

The PSLM instruction does not change the carry and overflow indicators. The result indicators are changed to reflect the final contents of the R register.

If bit 11 of the instruction is a 1, the interruption mask register remains unchanged in value but is logically presented as 0000 to the system. During this time, no I/O interruptions can occur until the next AND-to-mask or OR-to-mask instruction is executed.

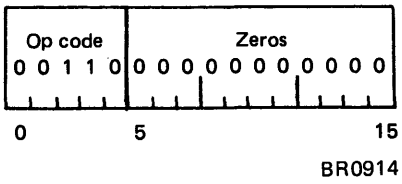
A system reset has somewhat the opposite effect: it sets on all four bits of the interruption mask register so that I/O interruptions can occur on any level.

STATE CONTROL INSTRUCTION

One instruction is provided to alter the state of the system. Either pending interruptions of lower priority are processed or the wait state is entered.

Level Exit

PLEX—400 ns



The processor exits the current priority interruption level. If lower-level interruptions are pending, they are serviced and the machine remains active. If no other interruptions are pending, the machine enters the wait state.

The PLEX instruction does not change the carry, overflow, and result indicators.

INPUT/OUTPUT INSTRUCTION

One instruction is provided to control communications with System/7 I/O devices and other processors, and for the setting of interruptions.

Execute I/O

PIO—800 ns + I/F

where I/F = internal interface delay time. It varies from 450 to 1800 ns depending on the physical location of the addressed I/O module.

This instruction is described fully under "I/O Instruction Commands." The execute I/O instruction does not change the result indicators, but does alter the carry and overflow indicators.

The direct control channel (DCC) is the link between the processor and the I/O modules. The direct control channel:

1. Controls the internal interface to the I/O modules.
2. Interprets all I/O commands.
3. Houses two 16-bit interval timers, operating on a fixed 50-μs time base.
4. Contains the standard attachment for the operator station.
5. Processes interruption requests through its interruption buffers.
6. Establishes interruption requests as directed by the set interrupt command.
7. Contains the attachment for the asynchronous communications control.
8. Houses the attachment to an 1130 host processor.

INTERNAL INTERFACE

The internal interface is the common communication link between the direct control channel and all attached I/O modules. In the configuration with only one I/O module (i.e., model A02 enclosure), the internal interface consists of three cables connecting the direct control channel and the I/O module.

In configurations having multiple I/O modules, the internal interface cables are distributed in a star-shaped manner by an interface multiplexer unit. This unit is directed by the direct control channel to select I/O modules, and to poll them for interruption requests.

In a model C03 or C06 enclosure, the interface multiplexer connects the direct control channel to each (maximum of five) of the I/O module positions. An additional connector is available for attaching an expansion enclosure (model D03 or D06).

An interface multiplexer unit in a model D03 or D06 expansion enclosure is driven by coaxial cables from the interface multiplexer in the model C03 or C06 enclosure.

The internal interface has data buses and indicators for common internal communication between the direct control channel and all attached I/O devices.

I/O INSTRUCTION COMMANDS

The execute I/O instruction is used to service I/O devices. In this instruction, bit combinations in the function field determine which of the following five basic I/O commands is to be performed:

- Immediate Write. Sends one word of data from a processor register to an output device.
- Immediate Read. Receives one word of data into a processor register from an input device.
- Prepare I/O. Sends one word of interruption information from a processor register to an I/O device.
- Halt I/O. Resets an I/O device.
- Set Interrupt. Sends one word of data to establish an interruption request, either on a priority level or to a host processor.

Only one word is sent or received each time a command is issued.

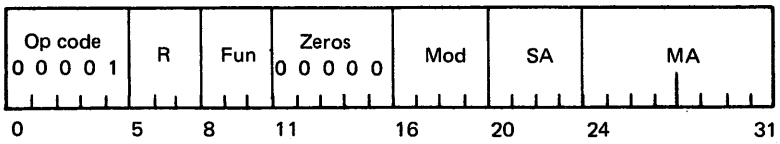
Each time an I/O command is issued, it is determined whether (1) the command was executed successfully, (2) an error occurred, (3) the device was busy, (4) an interruption was pending, or (5) the device was not attached in the system. This is done by examining the two-bit condition code, whose settings are determined (and made available to the program) by the results of the I/O operation.

The result indicators are not affected by I/O commands. The settings of the condition code and their meanings are discussed with each of the commands. The two bits used for condition codes are the same two bits that serve as carry and overflow indicators for other machine instructions. Therefore,

Condition code 00 = carry and overflow indicators both off.
Condition code 01 = carry indicator off, overflow indicator on.
Condition code 10 = carry indicator on, overflow indicator off.
Condition code 11 = carry and overflow indicators both on.

I/O INSTRUCTION FORMAT

The execute I/O (PIO) instruction has the long format.



BR0915

The instruction fields have the following significance:

Op Code: An operation-code field with a value of 00001 specifies an execute I/O instruction.

R: This three-bit field identifies which of the seven index registers (001 to 111) or accumulator (000) is involved in the I/O operation. Data read from an I/O device is stored in the register or accumulator, and data written to an I/O device is transferred from the register or accumulator.

Fun: The three-bit function field determines the type of I/O operation performed. The values and their meanings are:

000–Invalid
001–Immediate write
010–Immediate read
011–Prepare I/O
100–Halt I/O
101–Set interrupt
110–Invalid
111–Invalid

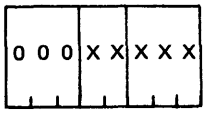
Invalid functions cause program-check interruptions. The five valid I/O operations are discussed in greater detail under "I/O Commands" (direct control channel).

Zeros: This field is not used, but the value must be set to zero.

Mod: The four-bit modifier field, used only with immediate read or write, further defines the operation to be performed. The contents of the modifier field are indicated in the format of each I/O command.

SA: The four-bit subaddress field specifies a point, group, or device within a module. The assigned subaddresses are discussed separately for each module.

MA: The contents of the module address field byte are used to select the enclosure and module being addressed by the execute I/O instruction. The module address byte is subdivided into three fields as follows:

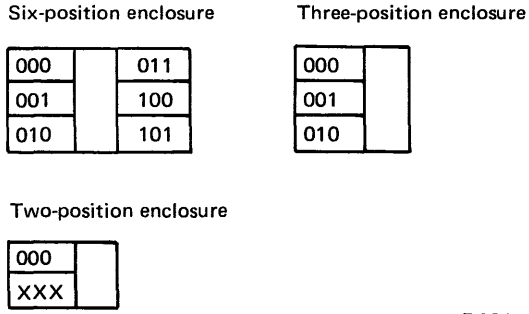


BR0916

Bits 0-2 of the word must be 0's.

System configurations have a maximum of two enclosures. The main enclosure (in which the processor module is housed) is addressed by 00 in bits 3 and 4. The expansion enclosure is addressed by 01 in bits 3 and 4.

Bits 5-7 are used to locate the module position within an enclosure (values of 110 and 111 are not available in the system):



BR0917

With the two-position enclosure, any value in bits 5-7 other than 000 addresses the single I/O module position. However, because of the possibility of future upgrading of the configuration, it is advisable to specify a value of 010 or 101. With this designation, the I/O module will maintain the same relative position in a larger configuration, i.e., at the bottom of the enclosure. Thus, the amount of program redesign and rewiring will be minimized when changing to a three- or six-position enclosure.

The processor module occupies the first position in the main enclosure; therefore, its module address is always 00000000.

I/O COMMANDS

The contents of the three-bit function field in the execute I/O instruction determine which of the five I/O operations is performed. These operations, listed with the function field value that invokes them, are described in the following paragraphs.

Immediate Write (Fun = 001)

An immediate write command transfers one word (16 bits) of data from a specified processor register to the addressed device. The condition code settings are:

00—Transfer of data was successfully completed.
01—An error was detected and recorded in the device DSW.
10—The device is busy or has an interruption pending.
11—The device is not in the system or is currently off-line.

Immediate Read (Fun = 010)

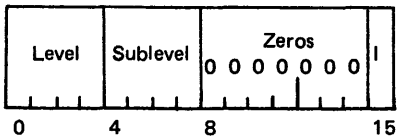
An immediate read command transfers one word (16 bits) of data from an addressed device to a specified processor register. The condition code settings are:

00—Transfer of data was successfully completed.
01—An error was detected and recorded in the device DSW.
10—The device is busy or has an interruption pending. This condition code does not occur on the read DSW command.
11—The device is not in the system or is currently off-line.

Prepare I/O (Fun = 011)

The prepare I/O command transfers one word (16 bits) of data from a specified register to the addressed device. The difference between this command and the immediate write command is the data and its purpose. The immediate write command transfers data for recording on the output medium, or for control of attached sensor-based I/O. The prepare I/O command transfers information advising the device whether it is allowed to interrupt and, if so, on what priority level and sublevel.

The data sent to the device on prepare I/O has the following format and meaning:



BR0919

The data fields have the following significance:

Level: This four-bit field specifies the priority interruption level assigned to the interrupting source. The value of bits 2 and 3 indicates the priority level (0, 1, 2, or 3). Bits 0 and 1 are ignored.

Sublevel: This four-bit field assigns a sublevel (from 0000 to 1111) to the interrupting source. When requesting an interruption, the source presents its sublevel so that the system can locate the starting address of the servicing routine in the appropriate level table. A sublevel of 0000 accesses the second word in the table; a sublevel of 1111 accesses the seventeenth (last) word. The first word in each level table is reserved for the old instruction address associated with a program check.

Zeros: This field is not used, but the value must be set to zero.

I: This bit determines whether the device is allowed to cause an interruption condition. An I-bit value of 1 permits the device to do so, and a value of 0 prevents it. (The interruption mask determines whether the device can actually interrupt the processor.)

If the device is not permitted to cause an interruption condition, the issuing of an interruption-causing command results in:

1. Condition code 1 being set.
2. The command reject indication being set in the DSW.
3. The command not being executed.

The device stores the data, presenting the priority level and sublevel to the system each time the device has an interruption condition (and interruptions are enabled). Stored data at the device is reset only on a system reset. The condition code settings for a prepare I/O instruction are:

00—Transfer of data was successfully completed.
01—An error was detected and recorded in the device DSW.
10—The device is busy or has an interruption pending.
11—The device is not in the system or is currently off-line.

Halt I/O (Fun = 100)

The halt I/O command resets the addressed I/O module or control, with the exception of sensor-based output points and interruption information sent by prepare I/O commands. All other controls, device status, and pending interruptions are reset.

Halt I/O to the processor module (zero module address in the instruction) causes a program-check interruption.

If the device has an outstanding request in the interruption buffer, the request is not reset by halt I/O. Therefore, an interruption request can be honored by the system even though the interrupting source is reset.

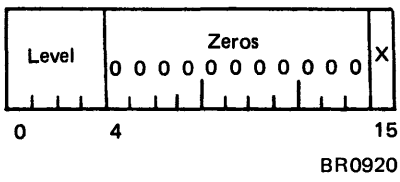
The condition code settings for halt I/O are:

00—The I/O module is on-line and reset was successfully completed.
01—Does not occur with halt I/O.
10—Does not occur with halt I/O.
11—The I/O module is not attached to the system.

Set Interrupt (Fun = 101)

The set interrupt command is issued to the direct control channel. One word of data is sent, directing the channel to establish an interruption request, either on a specified priority level or to an attached 1130 processor. The module address field (bits 4-14) in the set interrupt command must be all 0's, signifying that the processor module is addressed.

The data word transmitted from a processor register to the direct control channel has the following format:



BR0920

Bit 15 set to 0 requests that an interruption request be established in the buffer reserved for the priority level indicated by the level field contents (in this case, the sublevel value is zero).

Two buffers are used for each interruption level: one for general interruptions and one expecially for set interruptions. If a set interruption is pending in either buffer, condition code 2 is returned. If not, the set interrupt request is entered into the general interruption buffer (if free) or the special set interruption buffer.

Condition code settings for this use of the set interrupt command are:

00—Interruption request was successfully entered into the appropriate level buffer.
01—Does not occur with this use of the command.
10—A set interrupt request is already pending on the specified level.
11—Does not occur with this use of the command.

Bit 15 set to 1 requests the direct control channel to establish an attention interruption to the 1130 processor. The contents of the level field are ignored.

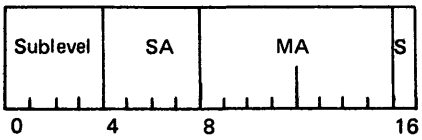
The 1130 attachment within the direct control channel generates an 1130 storage access channel interruption request, and sets on the attention status bit in the attachment.

Condition code settings for this use of the set interrupt command are:

00—Interruption request was made and attention status bit was set on in the 1130 attachment (does not indicate whether or not the 1130 has received and/or recognized interrupt yet).
01—Does not occur with this use of the command.
10—Attention status bit was already set on so this command is not honored.
11—1130 processor is off-line or not in the System/7 configuration.

INTERRUPTION MECHANISM

The direct control channel contains an interruption buffer for each priority level. All interruptions must pass through these buffers, except class interruptions (program check, machine check, and power failure) and interruptions to the 1130 processor (if present). Each buffer consists of a request signal and holding register for 17 bits of interruption identification in the following format:



BR0921

The holding register contains values for sublevel, subaddress (SA), and module address (MA). The S bit is a summary of the interrupting status; a 0 value indicates normal interruption condition and a 1 value indicates that an exception or error condition is present in the interrupting source.

Presenting an Interruption Request

When an interrupting source has an interruption request, it signals the processor module with its priority level and S bit, and specifies an interruption request. What happens next is determined by the availability of the buffer for the requested level.

If the interruption buffer for the priority level is not in use, the direct control channel signals the interrupting source. In response, the interrupting source sends its sublevel, subaddress, and module address to the buffer. Receipt of the information is acknowledged by the direct control channel and the interruption request latch for the source is reset.

On the other hand, if the interruption buffer for the priority level is already in use, the interruption request is stacked in the interrupting source (i.e., the interruption waits until the buffer is available and the direct control channel polls the devices for the next interruption).

When the processor is ready to accept the interruption currently in the buffer, the S bit is moved into the carry indicator and the remainder of the buffer contents are moved to the accumulator. The buffer is then available for another request.

To process stacked interruption requests, the direct control channel sends a signal to all devices through the internal interface, accompanied by the priority level of the available buffer.

This information is kept active for 2 μs so that all devices in the system can be polled. Devices with the same level number can interrupt: the first one performs the sequence detailed at the beginning of this section and any others are stacked in the manner just described.

Set Interrupt Command Requests

A set interrupt command can be used to direct that an interruption request be established in a particular interruption buffer. If the interruption buffer is currently occupied, however, the command request is placed in a separate set interruption buffer. This buffer has a signal for each interruption level to indicate that a program interruption is pending.

When the interruption requested by the set interrupt command is placed in the interruption buffer, either directly from the command or from the set interruption buffer, the displacement and subaddress fields are set to zero with a module address field of 00000111.

Issuing a set interrupt command when a set interruption is pending (in either the general interruption buffer or the set interruption buffer) results in condition code 2.

Host Attachment Interruptions

Interruptions coming from the host attachment are assigned a priority of level 3 displacement 0 in the direct control channel. A module address of zero, and a subaddress of 0111, are automatically assigned. Interruptions coming from the host processor are treated in the same manner as those coming from other I/O sources.

DEVICE STATUS WORD (DSW)

There is one device status word (DSW) for each module. Bits in the 16-bit DSW are set on as a result of errors occurring during the execution of an I/O instruction, i.e., before the condition code is set. No interruptions result from setting on a status bit because the presence of recorded errors is indicated to the program by condition code 1.

Thus, once an error is recorded in the DSW, any subsequent command (except halt I/O and read DSW) to the device is rejected and the condition code is set to 1, until the DSW is reset by a system reset, or a halt I/O or read DSW command.

A general DSW has the following format, with modules implementing bits pertinent to their operation:

Significant Bits	*Meaning*
1	CR—command reject. The addressed module cannot execute the command issued, e.g., an interruption-causing command to a device disabled for interruptions.
3-7	Device dependent. (See individual devices.)
9	ICC—interface control check. A parity error is detected on the interface involving a control field (i.e., module address, interruption request, or prepare I/O data). The operation that caused this condition may be retried; however, the error may have already caused inconsistent results.
14	DCK—data check. A parity error involving data is detected on the interface. The operation that caused this condition can usually be retried successfully if the error condition is intermittent.
15	INSA—invalid subaddress.

Any error encountered during an immediate read DSW or ISW is recorded in the DSW, replacing the original contents, and sets the condition code to 1.

INTERRUPT STATUS WORD (ISW)

There is one interrupt status word (ISW) for each interrupting source. Bits in the 16-bit ISW are set when errors are detected after completion of immediate commands and while the device is busy. Any ISW bit set on (except the device-busy bit) causes the device to request interruption. Each interruption source in a module has its own ISW.

An ISW can be read by the immediate read ISW command. If this instruction is successfully executed, the ISW is reset as long as its interruption request has been accepted by the system. The ISW is also reset by halt I/O, system reset, or the first new selection of the device by a command (not necessarily immediate read ISW) after its pending interruption has been accepted by the system.

A general ISW has the following format, with interrupting sources implementing bits pertinent to their operation:

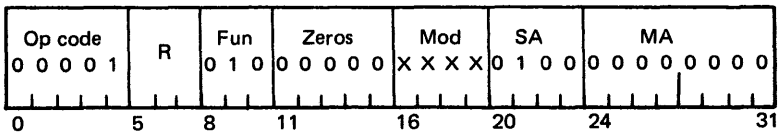
Significant Bits	*Meaning*
1	CR—command reject. The addressed module cannot execute the issued command, e.g., an interruption-causing command to a device disabled for interruptions.
3-7	Device dependent. (See individual devices.)
9	ICC—interface control check. A parity error involving a control field (i.e., module address, interruption request, or prepare I/O data) is detected on the interface. The operation that caused this condition may be retried; however, the error may have already caused inconsistent results.
12	Device busy. The device is in the working state, either busy or with an interruption pending.
13	Device end. The device has terminated an operation. This is not an error condition.
14	DCK—data check. A parity error involving data is detected on the interface. The operation that caused this condition can usually be retried successfully if the error condition is intermittent.
15	INSA—invalid subaddress. An input point or group was selected which was not installed.

DIRECT CONTROL CHANNEL STATUS WORD

Certain bits of the direct control channel status word are set on or off as a result of errors detected during the execution of immediate commands to devices that are directly attached to the processor module (timers, operator station, and asynchronous communications control). The function and format of the status word are similar to those of the device status word (DSW) for each I/O module.

Any bit set on in the status word causes condition code 1 to be returned in response to the current I/O command addressed to the processor module. Any subsequent I/O commands to these interruption sources receive the same condition code until the status word is reset.

The status word is read by the following immediate read command:



BR0922

This command stores the status word in the index register specified by the R field, or the accumulator if R = 000.

The status word is reset by either the aforementioned read command or system reset.

Significant bits of the status word are:

Significant Bits	*Meaning*
1	CR—command reject. An invalid command is addressed to the processor module.
3	A punch, read, or print character command was issued to the operator station when it was not ready.
4	ACC check. A command that could not be performed at the time was addressed to the asynchronous communications control. This is usually caused by a control/character sequence error.
15	INSA—invalid subaddress. An immediate command was issued to the processor module but the subaddress is for devices not available in the processor module. This also occurs when a command is issued to the asynchronous communications control, but either this feature is not installed or the required data set is not on-line.

INTERVAL TIMERS

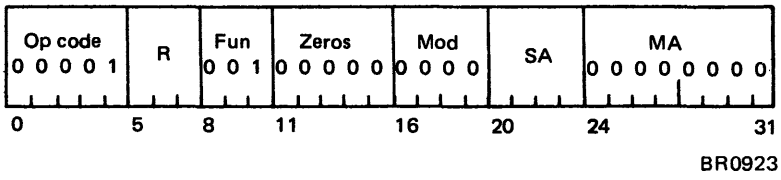
The direct control channel contains two 16-bit interval timers. Each timer is a one-word binary counter which decrements the value of its contents once each 50-μs interval. Upon reaching a value of zero (having completed counting the requested number of time intervals), the timer interrupts the system. Unless the timer is stopped, it continues to count and interrupts again whenever a value of zero is reached (timer value going -1, -2, . . . , -32,768, +32,767, +32,766, . . . , +1, 0, etc.).

Though timers are automatically decremented by the machine, they are separately controlled by programming. Timers can be started, stopped, read, or set to a value by I/O commands. A timer that is running can be read without disturbing its operation, but a timer must be stopped before it can be set to a new value.

I/O Commands

Timer commands must have module addresses (MA) of zero to specify the processor module, with subaddresses (SA) of 0000 for timer 0 and 0001 for timer 1. Operations on the timers are performed by immediate read and write commands; specific operations are further defined by the setting of the modifier bits in the commands.

Set Timer

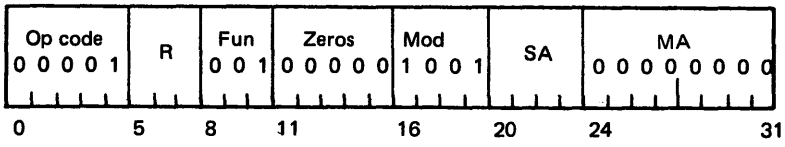


BR0923

A timer is set to an initial value by the immediate write command with a modifier field of 0000. The addressed timer (specified by the subaddress field) is loaded with the value residing in the index register (R), or the accumulator if R = 000. This value is the number of 50-μs intervals to be counted (after a start timer command is issued) before the timer interrupts the system.

Condition code 2 is set if the addressed timer is already running or has an interruption pending.

Start Timer

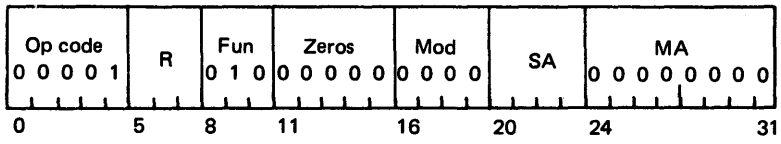


BR0924

An addressed timer (specified by the subaddress field) is started by the immediate write command with a modifier field of 1001. The R field is ignored. The start timer command results in an interruption when the timer completes counting the specified number of 50-μs time intervals.

Condition code 2 is set when this command is issued to a timer that has an interruption pending, or is already on and counting.

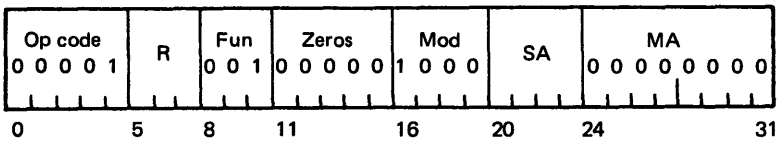
Read Timer



BR0925

A timer is read by the immediate read command with a modifier field of 0000. The addressed timer (specified by the subaddress field) is stored into the index register (R), or the accumulator if R = 000. No interruption occurs and the timer continues to count.

Stop Timer



BR0926

Timers must be stopped before being set to new values. An addressed timer (specified by the subaddress field) is stopped by the immediate write command with a modifier field of 1000. The R field is ignored.

Timer Interruptions

Interruption requests from timers use the same interruption mechanism as the I/O devices. Because no errors can occur during timer operations and timer interruptions have only one meaning, a read ISW command to the timers always records an interrupt status word whose contents are zero.

Both timers are prepared simultaneously (i.e., given the same priority level and sub-level) by a single prepare I/O command, with a module address of zero and a subaddress field of either 0 or 1.

The operator console has the switches, keys, and lights (valid in machine stop state only) needed to operate and control the system. The need for operator manipulation of manual controls is minimized by system design. To provide security from unauthorized use, the operator console may be disabled. It can be enabled by a key switch.

The main functions of the operator console are to turn power on and off, reset the system, load the initial program information, and store/display information in storage/registers. Additional facilities exist for program debugging and hardware testing.

The operator console is shown on Figure 5. Each of the console functions is discussed in turn, from top to bottom, starting at the upper left.

Tag Lights

These three lights indicate, in binary notation, the index register being used by the current instruction. If all lights are off, the instruction either does not require an index register or is using the accumulator or IAR (specified by a zero value in the register field of the instruction).

Level Lights

These lights indicate which one of the four interruption levels (0, 1, 2, or 3) is currently active.

LS Pty (Local Store Parity) Light

A local store parity error (detected in the accumulator, the IAR, or one of the index registers) turns on this indicator. A machine-check interruption occurs if the check-control switch is in the process position.

SDR Pty (SDR Parity) Light

A parity error detected in the storage data register turns on this indicator. A machine-check interruption occurs if the check-control switch is in the process position.

Ctl Chk (Control Check) Light

This indicator turns on when the processor has requested or taken more than one type of storage cycle simultaneously, or more than one interruption level is being executed simultaneously. A machine-check interruption occurs if the check-control switch is in the process position.

I/O Chk (I/O Check) Light

If a machine error occurs on the internal interface that prevents further communication with the sensor-based I/O modules, the condition turns on this light and causes a machine-check interruption. (Refer to "Seq (Sequence) Light.")

Seq (Sequence) Light

This light operates in conjunction with the I/O check light. The sequence light is on if the machine error occurred during an interruption, or off if the machine error occurred during an execute I/O instruction.

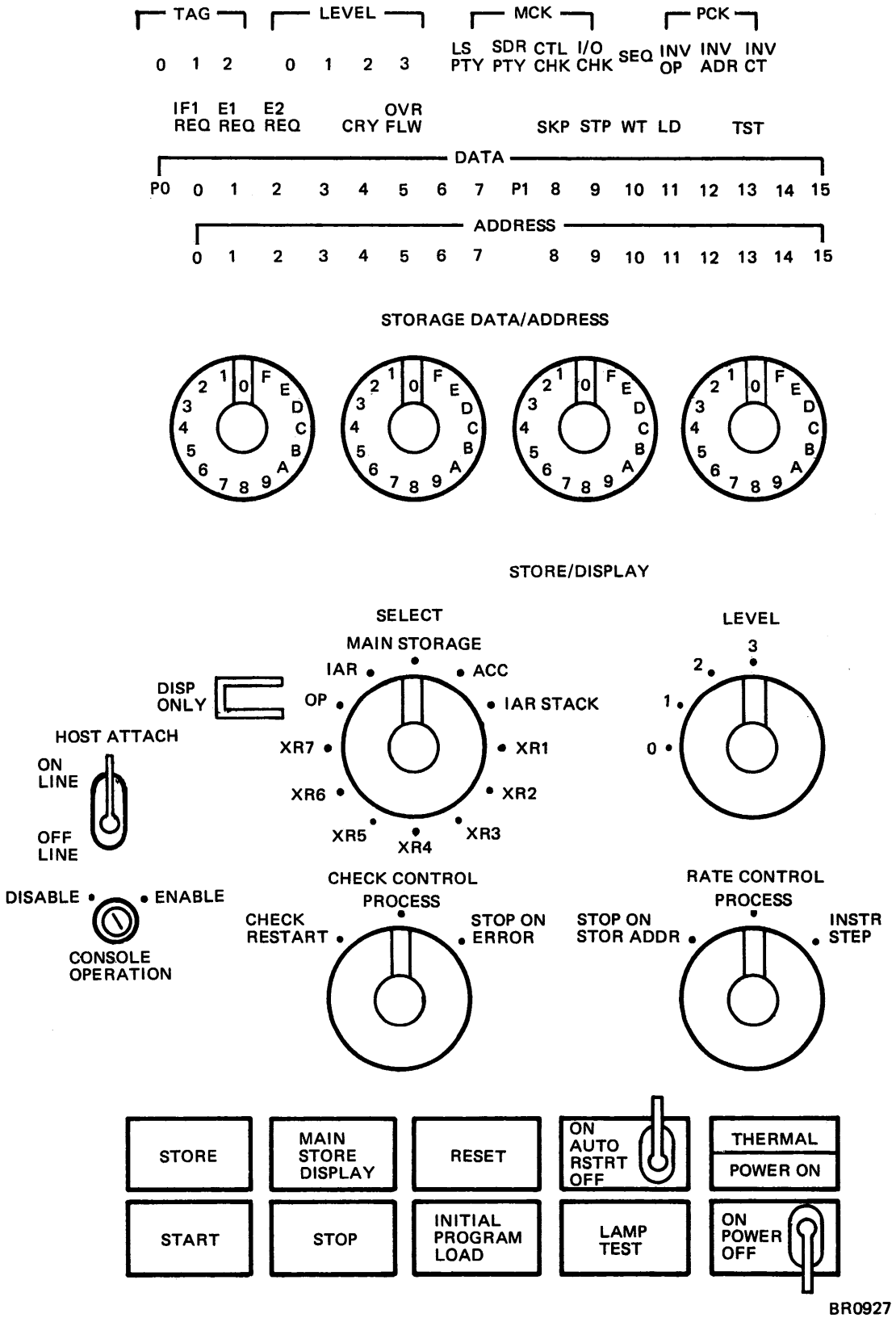


Figure 5. Operator Console

Inv Op (Invalid Operation) Light

This light turns on when an instruction has an invalid value in the operation-code or modifier field. This results in a program-check interruption.

Inv Adr (Invalid Address) Light

When the storage address register contains a main-storage address that exceeds the limits of the storage installed in the system, and storage is accessed, this light is lit and a program-check interruption occurs.

Inv Ct (Invalid Count) Light

This light turns on when an invalid count field is detected in a shift instruction, causing a program-check interruption.

IF1 Req (Instruction Fetch Request) Light

This light indicates that an instruction is about to be taken from storage (i.e., the machine will next be in the instruction fetch cycle).

E1 Req and E2 Req (Execution Request) Lights

These lights show whether the machine will next be in the first or second execution cycle, respectively.

Cry (Carry) Light

This light is lit when the carry indicator for the active interruption level is on.

Ovrflw (Overflow) Light

This light is lit when the overflow indicator for the active interruption level is on.

Skp (Skip) Light

This light is on when conditions required for a successful skip instruction are satisfied.

Stp (Stop) Light

This light is lit when the machine stops at the end of its current cycle. To continue processing, the start key must be pressed. The machine can be brought to a stop state and the stop light lit by any one of the following:

1. Pressing the stop key.
2. The occurrence of an error when the check-control switch is in the stop-on-error position.
3. The rate-control switch is in the instruction step position, or the stop-on-storage-address position, and the conditions applying to that position are met. (Refer to "Rate Control Switch.")

Wt (Wait) Light

This light is on when the machine is in the wait state. This happens when any of the following occurs:

1. A level exit instruction is executed and no interruptions are pending at other levels.
2. The system is reading the initial program load routine into the first 64 words of main storage.
3. The system is reset.

The term "wait state" means that state in which the processor is not processing instructions but is waiting for either a manual start, program-controlled start, or I/O interruption.

Ld (Load) Light

This light turns on when the initial program load key is pressed. The load light remains on until 64 words have been read from the paper tape reader into main storage (starting at location 0000) and a branch to location 0000 occurs to begin executing (on priority level 3) the program just loaded.

Tst (Test) Light

This light turns on and the machine operates in a nonstandard manner if either the check-control switch or the rate-control switch is not set to the process position.

Data Lights

These 18 lights consist of 16 data lights and the 2 associated parity lights. The lights display data at a point in time determined by the setting of the check and rate control switches. The displayed data is selected by the store/display switches.

Address Lights

These 16 lights display the last storage address that was accessed for data or an instruction.

Storage Data/Address Switches

These four rotary switches set up data for entry into the system via the operator console, or establish an address for use with the rate control switch. Each rotary switch has settings from 0 to F so that the data word or address to be established is selected by its hexadecimal value.

Data set up by the rotary switches is stored into the storage medium indicated by the select and level switches. The storing occurs when the store key is pressed.

Host Attach Switch

This switch is used (and in the on-line position) only when the system configuration is a System/7 coupled to an 1130 via the 1130 storage access channel. When the switch is in the off-line position:

1. No interruption or cycle-steal requests go to the 1130.
2. All 1130 instructions are ignored by the System/7. Condition code 3 is set if the System/7 issues a set interrupt command to the 1130.

Store/Display Select Switch

This 12-position rotary switch governs which storage medium is addressed from the console for storing and/or displaying information. The information to be displayed from the selected storage medium is indicated by the console data lights. The information to be stored in the selected storage medium is indicated by the setting of the storage data/address switches. Storing occurs when the store key is pressed.

The storage mediums selected at each position of the rotary switch are:

1. Op: Operation register, which contains the first word of the current instruction. This register can be selected only for the display of information.
2. IAR: Instruction address register (the main incrementing IAR, not the IARB associated with each priority level).
3. Main Storage: Main storage location, as specified by the IAR. The contents of the storage location are displayed when the main storage display key is pressed. To store information, the location to be stored into must first be stored in the IAR.

   Note: The storage mediums in positions 4-12 exist on each priority level. This switch selects the storage medium and the store/display level switch selects the particular level.

4. Acc: Accumulators (ACC0-ACC3).
5. IAR stack: IARB0-IARB3.

6-12. XR1-XR7: Index registers 1-7, respectively.

Store/Display Level Switch

This four-position switch is marked from 0-3 so that the appropriate priority level can be selected. The level switch is used in conjunction with the select switch. Each priority level has an accumulator, an IAR, and a set of seven index registers. The level switch determines what priority level of register is selected by the setting of the select switch.

Console Operation Switch

When this switch is in the disable position, all console functions are disabled except power on/off, host attach, and auto restart. To enable the console, a key must be used to turn the switch to its enable position.

Check Control Switch

This three-position switch has these options:

1. Process: Normal switch position, where a machine-check condition causes an interruption.
2. Stop on Error: A machine check causes the system to stop at the end of the machine cycle in which the error is detected. The stop light is then lit and the error condition is displayed.
3. Check Restart: A machine check causes system reset and the restart of program execution at address 0000. This provides a looping facility for the customer engineer.

Rate Control Switch

The three positions of this switch have the following effects:

1. Process: Normal position; permits continuous program execution.
2. Stop on Storage Address: Whenever the contents of the storage address register match the address designated by the storage data/address rotary switches, the machine stops with the stop light on. The stop occurs whether the address in the storage address register is to be used for fetching an instruction or an operand.
3. Instruction Step: Operation of the start key causes one instruction to be executed. The machine then stops and the stop light turns on.

Store Key

Pressing this key stores the data represented by the storage data/address rotary switches into the register specified by the settings of the store/display select and store/display level switches.

If main storage is being selected, the location to be stored into must first be set into the IAR by using the storage data/address rotary switches. Sequential depressions of the store key automatically increment the address if main storage is selected. The store key is operative only when the machine is stopped.

Main Store Display Key

Pressing this key causes the data lights to display the data located at the main storage address specified by the IAR. Sequential depressions of this key automatically increment the address.

This key is operative only when the machine is stopped and the store/display select switch is in the main storage position.

Register contents are automatically displayed by the data lights when the machine is stopped. No key need be pressed. The register that is to be displayed is determined by the settings of the select and level switches.

Reset Key

The following actions occur when the reset key is pressed:

1. The entire System/7 is reset, with the exception of digital and analog outputs.
2. The System/7 enters the wait state and all internal registers are cleared (set to zero). Program addressable registers (accumulator and index registers) are not reset.
3. The priority interrupt mask is set so that all interruptions are enabled.

Auto Rstrt (Automatic Restart) On/Off Switch

This is a two-position toggle switch. In the off position, operator intervention is necessary to restart the system after a shutdown due to a power failure or thermal warning.

When set to the on position, an attempt is made to restart the power system automatically after a power failure. If power is restored, the System/7 resets automatically and performs the IPL function if the IPL tape is loaded and positioned correctly in the operator station. If the automatic restart fails to restart the power system, the operator must intervene.

Thermal/Power On Lights

The red thermal light indicates an overtemperature or undertemperature condition in the system. A white backlight is on while power is in the system.

Start Key

When the machine is stopped, pressing the start key causes processing to resume. The amount of processing accomplished depends upon the setting of the rate control switch.

Stop Key

Pressing the stop key stops the machine, after completion of the current instruction, and turns on the stop light.

Initial Program Load (IPL) Key

Pressing this key results in a reset similar to that caused by the reset key. In addition, after the reset, 64 words are read from paper tape and stored into main storage starting at location 0000. The load light remains on for the entire procedure.

When the 64 words are stored, the load light goes off, interruption level 3 is activated, and instructions are executed beginning at location 0000.

Lamp Test Key

This key tests the console indicator lights. Pressing the key turns on all the lights in the console indicator section.

Power On/Off Switch

Placing this toggle switch in the on position applies line power to the power system and turns on the power-on light. In the off position, line power is removed and the power-on light is turned off.

The 5028 Operator Station (Figure 6) is included in every System/7 configuration. It attaches to an adapter in the processor module.

BR1031

Figure 6. IBM 5028 Operator Station

The operator station has a paper tape reader and a keyboard to input data and control information. To receive output from the system, the operator station includes a paper tape punch and a printer.

The operator station serves as the I/O device for operator communication with the system. In addition, the initial program load (IPL) of the system and the preparation of programs can also be accomplished through the operator station.

Information is transmitted at the rate of 10 characters per second. ASCII code is standard for the graphics on the operator station. However, all 256 binary combinations can be punched on the paper tape and read from it.

The printer uses 8-1/2-inch-wide roll paper. Characters are printed 10 per inch and 72 per line. Vertically, 6 lines per inch are printed.

A switch mounted on the front of the keyboard determines whether the operator station is in the on-line or off-line mode of operation. When the operator station is on-line, it is connected for communicating with the System/7. When the operator station is off-line, it is disconnected from the system. The operator station can then be tested or used as a keypunch machine to prepare program tapes for subsequent use.

A character (with a bit structure of 10000111), transmitted from the System/7 to the operator station, activates a bell in the keyboard to provide an audible alarm to the operator.

OPERATOR STATION ADAPTER ISW

The 16-bit interrupt status word (ISW) for the operator station adapter has only three significant bits. The remaining 13 bits are not used.

Bit 0—Attention
Bit 12—Busy
Bit 13—End

The attention bit is set on by pressing the operator station request key, which causes an interruption request. If the operator station adapter is not busy and no interruption is pending, turning on the attention bit causes the adapter to present an interruption request. If the adapter is busy or has an interruption pending, both the attention and end bits are on when the interruption occurs and the program issues a read ISW to the adapter.

The busy bit is on during motor-on timeout, and during the time between setting condition code 0 to any command resulting in an interruption request, and the occurrence of the end interruption.

The end bit is on from the time an interruption request is presented until the ISW is reset.

The ISW is read by the read ISW command. If successfully read, the ISW is reset if its interruption request is accepted by the system. The ISW is also reset by a system reset, or by the first new selection of the operator station adapter by a command (not necessarily read ISW) after its pending interruption has been accepted by the system.

INITIAL PROGRAM LOAD (IPL)

When the IPL button on the operator console is pressed, a system reset occurs and the operator station adapter turns on the motor, locks the keyboard, disables the printer, and feeds tape until a hole is detected (i.e., a non-zero character is read).

The adapter then starts to transfer characters for storing. This is done two characters (two bytes) at a time in order to fill each 16-bit storage location word. Storing begins with location 0 and continues consecutively until location 63 is filled.

During this data transfer the System/7 cannot be interrupted. Any error detected during this portion of the IPL procedure requires operator intervention.

Following the data transfer, a branch is made automatically to location 0 and the system begins executing instructions on priority level 3.

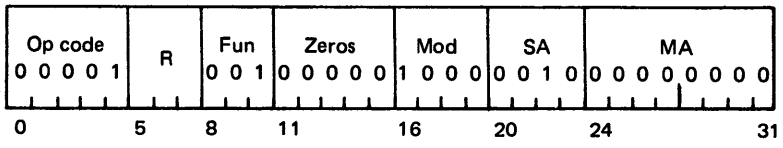
I/O COMMANDS

The operator station is programmed by immediate read and write commands. Modifier bits further define the operation to be performed. Interruption requests can be presented to the system by the operator station after the execution of a prepare I/O command.

All I/O commands to the operator station must have a subaddress (SA) of 0010 and a module address (MA) of 00000000.

Characters are stored into, or transmitted from, bits 8-15 of an index register or accumulator; bits 0-7 are not affected.

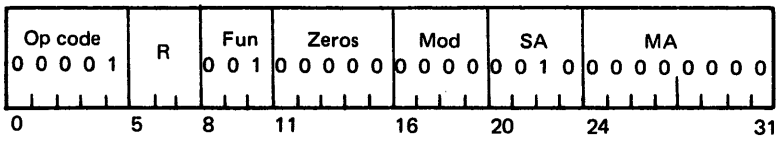
Turn On Motor



BR0928

This command starts the operator station motor. After approximately 1 second, the motor attains the proper operating speed and an interruption request is presented to the system. Condition code 2 is returned if the operator station adapter is busy or has an interruption pending. The contents of the R field are ignored.

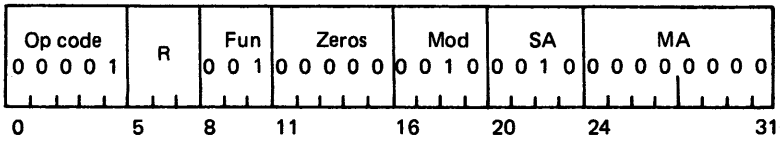
Turn Off Motor and Lock Keyboard



BR0929

This command stops the operator station motor and locks the keyboard. No interruptions occur as a result of this command. The contents of the R field are ignored.

Print Only

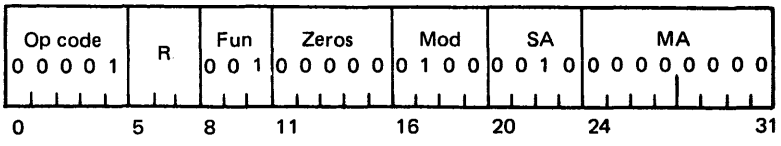


BR0930

One character is sent to the operator station for printing as a result of this command. The single character is obtained from bits 8-15 of the index register specified by the R field, or from the accumulator if R = 000. After the character is printed, the operator station adapter presents an interruption request.

If the operator station motor is not running, condition code 1 is returned and bit 3 of the direct control channel status word is set on. Condition code 2 is returned if the operator station is busy or has an interruption pending.

Punch Only

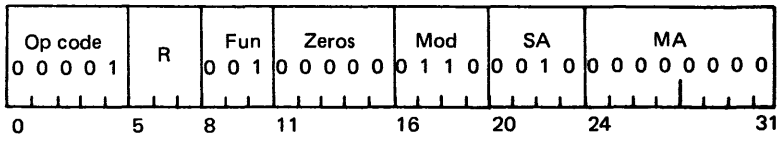


BR0931

One character is sent to the operator station for punching on tape as a result of this command. The single character is obtained from bits 8-15 of the index register specified by the R field, or from the accumulator if R = 000. Tape movement through the punch mechanism is automatic, stopping after each character is punched. The operator station adapter presents an interruption request after the character is punched.

If the operator station motor is not running, condition code 1 is returned and bit 3 of the direct control channel status word is set on. Condition code 2 is returned if the operator station is busy or has an interruption pending.

Print and Punch

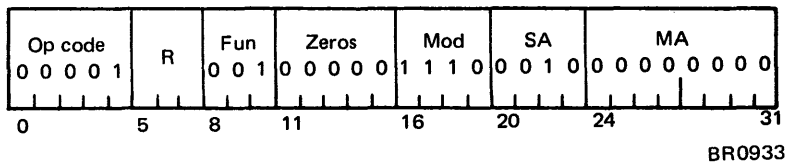


BR0932

One character is sent to the operator station, for both punching on tape and printing, as a result of this command. The single character is obtained from bits 8-15 of the index register specified by the R field, or from the accumulator if R = 000. Tape movement through the punch mechanism is automatic, stopping after each character is punched. The operator station adapter presents an interruption request after the character has been punched and printed.

If the operator station motor is not running, condition code 1 is returned and bit 3 of the direct control channel status word is set on. Condition code 2 is returned if the operator station is busy or has an interruption pending.

Feed Tape and Print

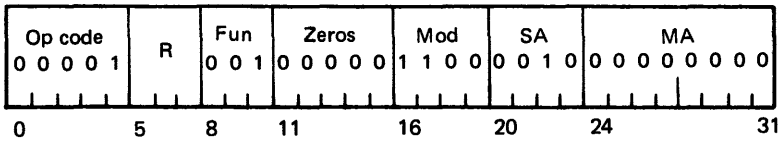


BR0933

This command enables the printer and causes the tape to move to the next character. The character is printed and transmitted to the operator station adapter, causing an interruption request. To obtain the character from the adapter, a read command must be given after the interruption request is presented. The contents of the R field are ignored in the feed-tape-and-print command.

If the operator station motor is not running, condition code 1 is returned and bit 3 of the direct control channel status word is set on. Condition code 2 is returned if the operator station is busy or has an interruption pending.

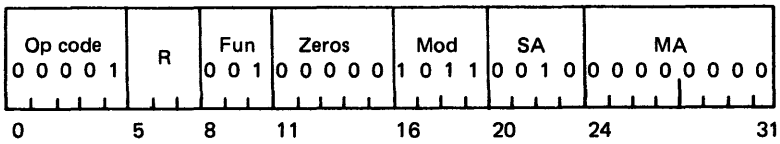
Feed Tape and No Print



This command performs the same function as the feed-tape-and-print command, except that the character is not printed. The feed-tape-and-no-print command moves the tape to the next character. This character is transmitted to the operator station adapter, causing an interruption request. To obtain the character from the adapter, a read command must be given after the interruption request is presented. The contents of the R field are ignored in the feed-tape-and-no-print command.

If the operator station motor is not running, condition code 1 is returned and bit 3 of the direct control channel status word is set on. Condition code 2 is returned if the operator station is busy or has an interruption pending.

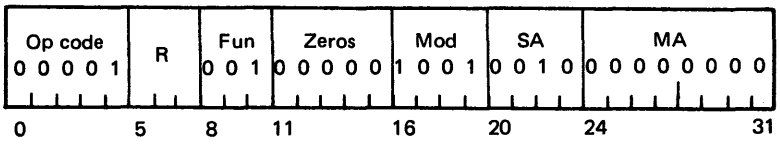
Keyboard Entry and Print



This command unlocks the keyboard, enables the printer, and turns on the operator station proceed light. When the operator enters a character from the keyboard, it is printed and transmitted to the operator station adapter. The keyboard is then locked, the printer is disabled, the proceed light is turned off, and the operator station adapter presents an interruption request. To obtain the character from the adapter, a read command must be used after the interruption request is presented. Following this read command, another keyboard-entry-and-print command is normally issued to accept the next character from the keyboard. The contents of the R field are ignored in the keyboard-entry-and-print command.

If the operator station motor is not running, condition code 1 is returned and bit 3 of the direct control channel status word is set on. Condition code 2 is returned if the operator station is busy or has an interruption pending.

Keyboard Entry and No Print

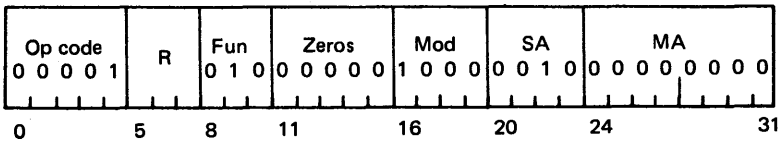


BR0936

This command performs the same function as the keyboard-entry-and-print command, except that the character is not printed. The command unlocks the keyboard and turns on the operator station proceed light. When the operator enters a character from the keyboard, it is transmitted to the operator station adapter. The keyboard is then locked, the proceed light is turned off, and the adapter presents an interruption request. To obtain the character from the adapter, a read command must be used after the interruption request is presented. Following this read command, another keyboard-entry-and-no-print command is normally issued to accept the next character from the keyboard. The contents of the R field are ignored in the keyboard-entry-and-no-print command.

If the operator station motor is not running, condition code 1 is returned and bit 3 of the direct control channel status word is set on. Condition code 2 is returned if the operator station is busy or has an interruption pending.

Read Character With Tape Feed

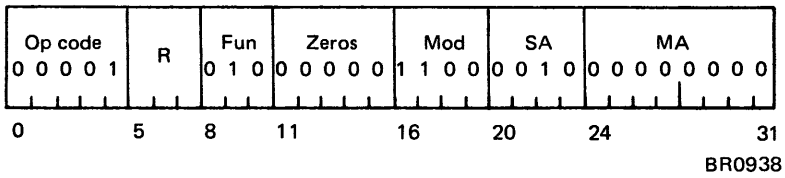


BR0937

This command stores a single character, from the operator station adapter, into bits 8-15 of the index register specified by the R field, or the accumulator if R = 000. Tape is then moved to the next character, which is transmitted to the operator station adapter, and an interruption request is presented. Whether this character is printed or not depends on whether a feed-tape-and-print command or a feed-tape-and-no-print command was issued prior to this read-character-with-tape-feed command.

If the operator station motor is not running, condition code 1 is returned and bit 3 of the direct control channel status word is set on. Condition code 2 is returned if the operator station is busy or has an interruption pending.

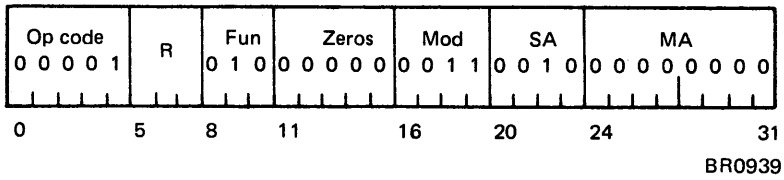
Read Character Without Tape Feed



This command stores a single character, from the operator station adapter, into bits 8-15 of the index register specified by the R field, or the accumulator if R = 000. No interruptions occur as a result of this command.

If the operator station motor is not running, condition code 1 is returned and bit 3 of the direct control channel status word is set on. Condition code 2 is returned if the operator station is busy or has an interruption pending.

Read ISW



The read ISW command stores the operator station adapter ISW into the index register specified by the R field, or the accumulator if R = 000.

The asynchronous communications control is a special feature attachment residing in the processor module, mutually exclusive with the adapter required for the 1130 attachment. The asynchronous communications control provides the required logic for controlling transfer of serial data to and from a host processor at a remote location, via a data set/communications line facility. This feature can be used for interfacing with telecommunication equipment or other processors that have compatible adapters. One or more System/7 configurations (each with an asynchronous communications control) can function as satellite processors to a System/360 (models 25 and up), a System/370, or an 1800 Data Acquisition and Control System.

Data transmission is serial-by-bit, using the start/stop method of character and bit synchronization. The code used is the IBM Paper Tape Transmission Code/Extended Binary Coded Decimal, so nine bits are needed to form each character (start-B-A-8-4-2-1-C-stop). Start and stop are data signal bits, C is the parity bit, and the remainder are data bits.

The ACC is programmed by System/7 I/O commands, with data being transmitted or received one byte at a time. On output, bits 8-15 of the register specified in the I/O command are transmitted. On input, the contents of bits 8-15 of the register specified in the I/O command are replaced by the incoming data. In either case, bits 0-7 of the register are unchanged. This feature operates on an interruption request basis similar to that used by other System/7 I/O devices.

Data rates, selected by machine pluggable options, are 14.8 or 66.7 characters per second. These character rates correspond to bit rates of 134.5 and 600 bits per second, respectively.

The ACC operates in half-duplex mode; i.e., the System/7 or the host processor can either transmit or receive data, but cannot transmit and receive simultaneously.

LINE CONTROL CHARACTERS

The ACC presents the same line control interface to the communications line as does the IBM 2740 Communications Terminal Model I. The following six characters are used to control line functions. They are control characters, not data characters. However, the last three can also be presented to the program as data if they are received within the message text.

	Bit Structure						
Character Function	*B*	*A*	*8*	*4*	*2*	*1*	*C*
End of transmission			8	4	2	1	C
End of address			8		2	1	
End of block		A	8	4	2		C
Positive response	B	A	8		2	1	
Negative response	B						
Start of address		A	8		2	1	C

IBM COMMUNICATION LINE ADAPTERS

The System/7 asynchronous communications control can communicate with the host processor over private lines, leased common-carrier facilities, or switched voice-grade common-carrier lines.

The following IBM line adapters are available as features with the ACC:

Type	*Rate (Bits per Second)*	*Line Type*	*Maximum Line Length*
2B limited distance	600 maximum	2-wire	8.25 miles
1A leased line	600 maximum	2-wire	No limit
1B leased line	600 maximum	4-wire	No limit

An attachment interface option is also available for attachment of the following IBM World Trade modems:

Type	*Rate (Bits per Second)*	*Line Type*	*Type of Service*
IBM 3976-I	200 maximum	2- or 4-wire	Private or leased
IBM 3976-II	200 maximum	2-wire	Switched
IBM 3976-III	1200 maximum*	2- or 4-wire	Private, leased, or switched
IBM 3977-I or IBM 3977-II	1200 maximum*	2- or 4-wire	Private, leased, or switched

*System/7 supports 600 bps maximum.

An attachment interface option is also available to provide the following six interface signals for non-IBM modems in conformance with Electronic Industries Association (EIA) Standard RS 232C within the United States, and Consultive Committee on International Telephone and Telegraph (CCITT) V24 Standards outside the United States:

1. Transmitted data
2. Received data
3. Request to send
4. Clear to send
5. Data terminal ready
6. Data set ready

INTERRUPT STATUS WORD (ISW)

Bits in the ISW are set on to indicate operating status of the ACC and notification of errors detected. Interruptions result from setting on any ISW bits except serial receive data (bit 1) and device busy (bit 12).

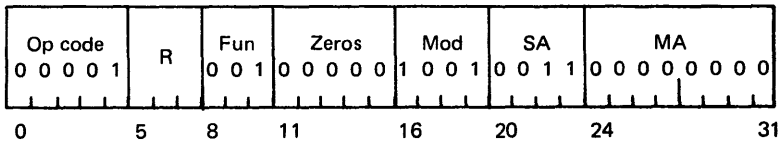
The significant bits of the ACC ISW and their meanings are:

Significant Bits	*Meaning*
0	Sel—selected. The ACC recognizes a poll or address from the host processor. This bit can be active only when the station control option is installed.
1	SRD—serial receive data. This bit reflects the condition of the serial receive line at all times. The bit is used as a diagnostic aid for the receive circuits of the ACC. No interruption occurs when this bit is set on.
2	Ovr—overrun. This bit signifies that a character has been received before the last character is read by the System/7.
3	EOB—end of block. The ACC recognized an end-of-block control character followed by an LRC character.
4	CHR—character received. This bit is set on by receipt of any character other than end-of-block (EOB), end-of-transmission (EOT), or the first end-of-address (EOA). The ACC must be in receive mode to activate this bit.
5	TO—time out. This bit is set on if any of the following operations do not occur within a 15-second time limit: 1. ACC responds to a transmit control command. 2. System/7 issues a transmit character command when in transmit mode. 3. ACC receives a character when in receive mode.
6	RT—ready to transmit. This bit is set on when either of the following occurs: 1. The ACC is ready to accept a new character for transmission. 2. A transmit control command is issued and the ACC senses that the data set is ready to send.
7	DSE—data set error. This bit is set on whenever the data set becomes inactive while the ACC is communicating with it.
11	Pol—device polled. This bit works in conjunction with bit 0. If the Pol bit is on, it means that the ACC was polled by the host processor. If the Pol bit is off, the ACC was addressed by the host processor.
12	Busy. The device busy bit reflects the actual operating condition of the ACC at all times. It is on whenever the ACC is performing any transmit or receive operation. The status of this bit does not cause interruptions, and a read ISW command does not reset the bit.
13	End. In receive mode, the device end bit is set on when the host processor has finished transmitting. In transmit mode, the end bit is set on if the host processor responds to a transmitted message with another message instead of a positive or negative response.
14	DCK—data check. This bit is set on by a negative response, a synchronization error, or either a longitudinal or vertical redundancy check error.

The asynchronous communications control is programmed by immediate read and write commands. Modifier bits further define the operation to be performed. Interruption requests can be presented to the system by the ACC after the execution of a prepare I/O command.

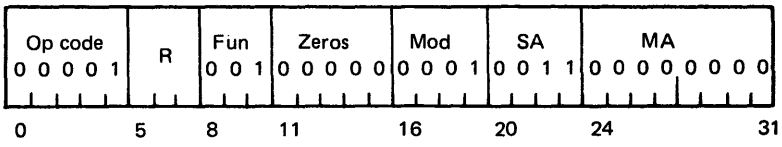
All I/O commands to the ACC must have a subaddress (SA) of 0011 and a module address (MA) of 00000000.

Transmit Control



This command puts the ACC into the transmit mode of operation (R field is ignored). The ACC attempts to establish a data link with the remote processor, during which time the ACC remains in busy status. When the data link is established, the ACC requests an interruption after setting on the ready-to-transmit bit in the ISW (bit 6). A time limit of 15 seconds is allowed to establish a data link. If no link is established, a timeout interruption (ISW bit 5 set on) request is presented.

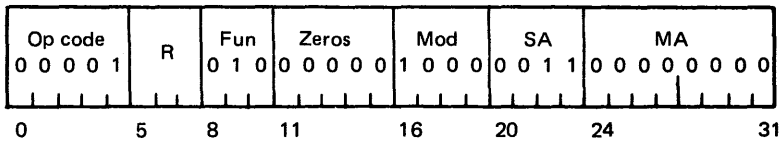
Transmit Character



This command transmits bits 8-15 of the index register (R), or the accumulator if R = 000, to the remote processor. When transmission is complete, the ACC requests an interruption after setting on the ready-to-transmit bit in the ISW (bit 6).

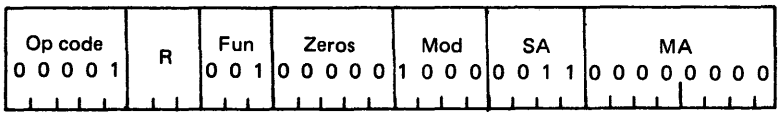
Transmit character commands must always be preceded by a single transmit control command, to determine if the data set and transmission lines are in working condition. If this is not done, transmit character commands are rejected, condition code 1 is returned, and the ACC error indicator (bit 4) is set on in the direct control channel status word.

Read Character



This command stores the character received from the host processor into bits 8-15 of the index register (R), or the accumulator if R = 000. Bits 0-7 are not changed. This command normally is given following a character received interruption (ISW bit 4) from the ACC.

Reset ACC Attachment

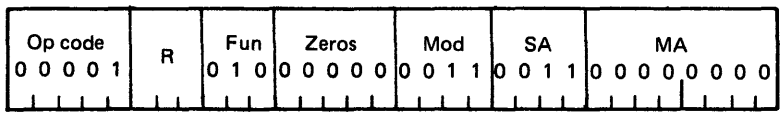


BR0943

This command performs a halt I/O function on the ACC. A pending interruption, device status, and all other controls are reset, with the single exception of data sent by a prepare I/O command. The R field in the command is ignored.

Following this reset command, the ACC is in the standby mode of operation if interruptions are allowed. Incoming messages can be received, or transmit operations can begin, under control of a System/7 program. If interruptions are prevented, the ACC is held in the reset condition.

Read ISW

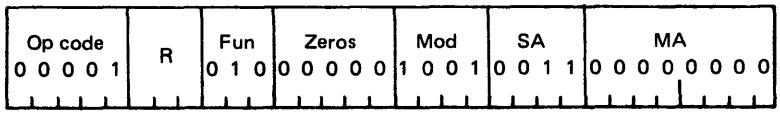


BR0944

The read ISW command stores the 16-bit interrupt status word associated with the ACC into the index register (R), or the accumulator if R = 000.

When the interruption that results from setting an ISW bit is serviced by the program, subsequent execution of any command to the ACC resets the ISW. The ISW is also reset by system reset.

Read LRC



BR0945

The read LRC command stores the ACC LRC (longitudinal redundancy check) character into bits 8-15 of the index register (R), or the accumulator if R = 000. Bits 0-7 are not changed and bit 8 is always 0.

The LRC character provides a horizontal parity check applied to the group of certain bits from each character in a message block. At the end of a message, the LRC character is stored and checked to determine if a complete message was received. The received LRC character is compared with an LRC character from the received message. If the two characters do not agree, the message was not received correctly.

If System/7 is directly attached to an 1130 host processor, a special adapter is required in the direct control channel to permit direct two-way storage-to-storage communication and information transfer through the 1130 storage access channel (SAC). This adapter precludes using the System/7 as a satellite processor by means of the asynchronous communications control.

The 1130 host attachment can be made to either the 1130 SAC or the 1130 SAC II. The attachment transfers data or instructions at the 1130 storage speed on either storage access channel (i.e., 454 kc/s on the 2.2-μs 1130 or 277 kc/s on the 3.6-μs 1130). Although the attachment requests cycles so quickly that it prevents instruction execution in the 1130 CPU (except for interruption level 0 or 1), the System/7 storage speed permits execution of System/7 instructions during a data transfer. The reason that interruption level 0 or 1 instructions are executed in the 1130 during a data transfer is that the attachment obeys the inhibit-cycle-steal-request line from the 1130. For further details, refer to the Storage Access Channel Original Equipment Manufacturers' Information Manual, Order No. GA26-3645.

With the 1130 host processor configuration, System/7 storage access conflicts between the System/7 and 1130 can be resolved. Use of this storage by the two systems requires both machine and program controls. The machine provides an interruption mechanism that enables either system to alert the other for service. To determine the nature of the service required, programming must maintain current control parameters in a System/7 storage area.

All data transfers are initiated by the 1130, and all end or error conditions are signalled to the 1130 by means of interruptions. No direct communication exists between the 1130 and the System/7 I/O modules, which are used solely by the System/7.

Although no provision is made for parity within the 1130 SAC, the attachment generates the necessary parity for data destined for System/7 storage.

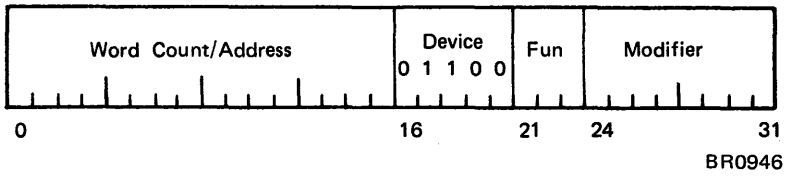
An on-line/off-line switch for the 1130 attachment is located on the System/7 operator console. When in the off-line position, interrupt and cycle-steal requests cannot go to the 1130, all 1130 instructions are ignored, and condition code 3 is returned if a System/7 set interrupt command is addressed to the attachment.

I/O CONTROL COMMANDS (IOCC)

Standard 1130 I/O control commands are used to program the attachment. An 1130 execute I/O (XIO) instruction is issued to address the appropriate IOCC. An IOCC specifies the operation to be performed and the device to which the operation is directed. Also, on data transfer operations, an IOCC specifies the word count or address of the data to be transferred. For a general description of the XIO instruction and IOCCs, refer to *IBM 1130 Functional Characteristics,* Order No. GA26-5881.

An IOCC must start at an even word address in 1130 storage. The contents of the 1130 accumulator are destroyed during execution of the XIO instruction.

An 1130 IOCC for the System/7 has the following format:



The IOCC fields have the following significance:

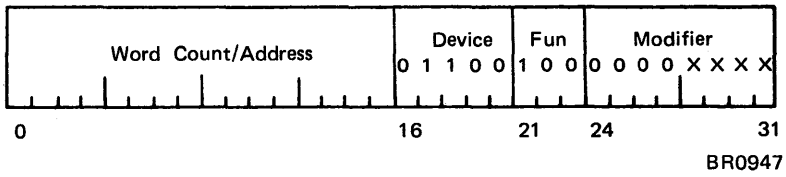
Word Count/Address: This field is used to pass the three values required by the attachment in order to perform a data transmission: a transmission word count, a System/7 storage address, and an 1130 storage address. The count specifies the amount of data to be transferred. The addresses establish the beginning of the data tables between which the data transfer is to occur.

Device: This five-bit field identifies the I/O device to which the IOCC is directed. In this case, a binary 01100 is the code signifying the System/7.

Function (Fun): The three-bit function code determines the specific I/O operation performed.

Modifier: An eight-bit field provides additional information, when necessary, for the function specified.

Control



An XIO with a control IOCC loads the 1130 attachment in the System/7 with one of two parameters necessary for performing a data transfer operation between the System/7 and the 1130. The two parameters are:

1. System/7 storage address at which the data transfer is to begin.
2. Count of the number of 16-bit data words to be transferred.

The 1130 storage address involved in the data transfer is indicated subsequently in either the 1130 initiate read or initiate write IOCC.

Since only one of the two parameters can appear in a single control IOCC, two control IOCC's must be programmed for each data transfer.

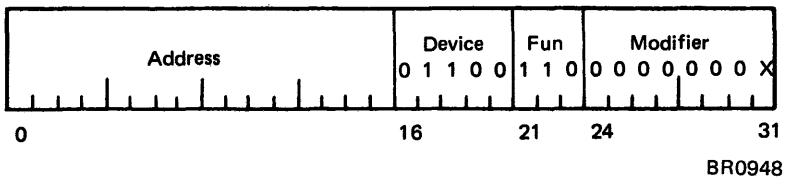
Modifier bits 30 and 31 are used to signal which, if either, of the two parameters is contained in the control IOCC:

Bit 30 = 0: Ignore bit 31.
Bit 30 = 1: Perform function specified by bit 31.
Bit 31 = 0: Transfer word count to attachment.
Bit 31 = 1: Transfer System/7 storage address to attachment.

Modifier field bits 28 and 29 of the control IOCC are used to control interruptions as further described in "1130 Attachment Interruptions":

Bit 28 = 0: Ignore bit 29.
Bit 28 = 1: Perform function specified by bit 29.
Bit 29 = 0: Permit interruptions caused by attention signals and power/thermal warnings.
Bit 29 = 1: Prevent interruptions caused by attention signals and power/thermal warnings.

Initiate Read

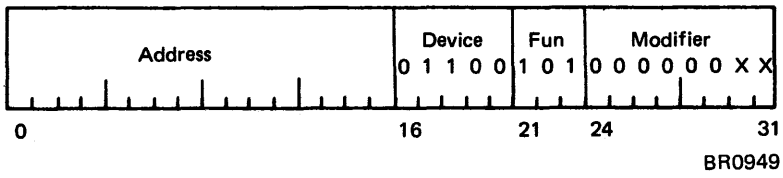


An XIO with an initiate read IOCC sends a block of contiguous data from System/7 storage to 1130 storage. The starting location of the System/7 data, and the number of words transferred, are established by previously executed control IOCCs. The address field of the initiate read IOCC contains the starting location, in 1130 storage, of the data to be received.

Modifier field bit 31 is used to control interruptions as further described in "1130 Attachment Interruptions":

Bit 31 = 0: Do not interrupt System/7 for operation end.
Bit 31 = 1: Interrupt System/7 for operation end.

Initiate Write



An XIO with an initiate write IOCC sends a block of contiguous data from 1130 storage to System/7 storage. The number of words transferred, and the System/7 starting location into which they are stored, are established by previously executed control IOCCs. The address field of the initiate write IOCC contains the starting location, in 1130 storage, of the data to be transmitted.

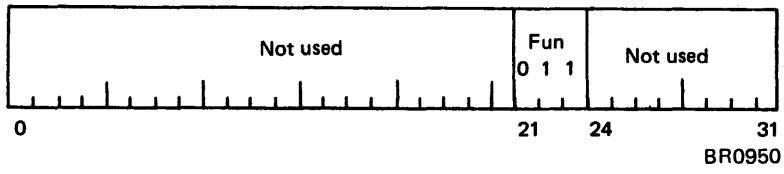
Modifier field bit 31 is used to control interruptions as further described in "1130 Attachment Interruptions".

Bit 31 = 0: Do not interrupt System/7 for operation end.
Bit 31 = 1: Interrupt System/7 for operation end.

See "Electronic Initial Program Load (EIPL)" for the special use of modifier field bit 30:

Bit 30 = 0: No EIPL.
Bit 30 = 1: EIPL.

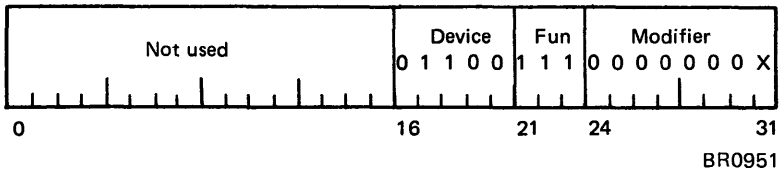
Sense Interrupt



An XIO with a sense interrupt IOCC loads the 1130 accumulator with the interruption level status word (ILSW) associated with the highest-priority interruption level that is on, in order to determine the interrupting device. This command is common to all 1130 I/O devices; therefore, no device-code field is needed.

When the System/7 interrupts the 1130, bit 4 in the ILSW is on for interruption level 3.

Sense Device



An XIO with a sense device IOCC loads the 1130 accumulator with the device status word (DSW) from the 1130 attachment adapter. The 1130 can determine the cause of an interruption by analyzing these DSW bits.

Modifier field bit 31 has the following special meaning:

Bit 31 = 0: Do not reset DSW bits.
Bit 31 = 1: Reset DSW bits.

DEVICE STATUS WORD (DSW)

The 16-bit device status word (DSW) associated with the 1130 attachment has various bits set on to indicate program operating status and detected errors. The DSW is read by an 1130 XIO with the sense device IOCC. The DSW can be reset (except bit 14) by the same IOCC if the System/7 interruption request is on and the reset bit (bit 31 of the IOCC) is also on. Reset can also be accomplished by turning on System/7 power or by the reset line in the 1130 storage access channel.

The significant bits in the DSW for the 1130 attachment have the following meanings:

Significant Bits	*Meaning*
0	Atten–attention. This bit is set on by a System/7 set interrupt command directed to the 1130. This is not an error.
1	Op End–operation end. This bit is set on when the word count equals zero, or a power/thermal warning or error is detected during a data transfer operation.
2	ISA–invalid storage address. This bit is set on when the 1130 attempts to address a storage location outside the installed capacity of the System/7. This is an error.
3	DCK–data check. This bit is set on when a parity error is detected by the 1130 attachment when fetching words from System/7 storage. This is an error.
4	CT=0–count=0. This bit is set on when the word count register in the 1130 attachment is equal to zero. This is not an error.
5	PTW–Power or thermal warning. This bit is set on when the appropriate condition occurs. Refer to "Power Failure and Thermal Warning."
14	Ready. This bit is set on when the System/7 is on-line and power is good. This is the only bit not reset by the sense device IOCC. This is not an error.
15	Busy. This bit is on while the 1130 attachment performs data transfer operations, but is turned off as soon as the operation end bit is set on. Any 1130 command (except sense device) to the 1130 attachment is ignored if the busy bit is on. This is not an error.

1130 ATTACHMENT INTERRUPTIONS

The 1130 attachment presents interruptions to the 1130 on its priority level 3. Interruption requests are made to the 1130 on any of the following conditions:

1. DSW attention bit is set on because a System/7 set interrupt command is directed to the 1130 (if the interruption controls do not inhibit the interruption request).
2. DSW operation end bit is set on. (The count=0 bit, the power/thermal warning bit, or an error bit is also on for this interruption request.)
3. DSW power or thermal warning bit is set on by the appropriate condition (if the interruption controls do not inhibit the interruption request). The DSW ready bit is not affected (i.e., remains on) but is set off when the power actually fails. If a data transfer operation is in progress when the power/thermal warning bit comes on, the operation end bit is also set on.

If the interruption that occurs at the "operation end" time of a data transfer is to be directed to the System/7 as well as the 1130, bit 31 in the modifier field of the initiate read or initiate write IOCC must be on. However, if an error is detected during the data transfer, the operation is terminated and the interruption is directed to the 1130 only (count=0 DSW bit may not be set on in this case).

When the System/7 directs a set interrupt command to the 1130 attachment, the DSW attention bit is set on unless it is already on. In this case, condition code 2 is returned to the System/7. If the 1130 attachment is busy, the attention bit is set on and indicated to the 1130 at the time an operation-end interruption occurs for the operation in progress. If the 1130 attachment is not busy, an interruption request is made to the 1130 on its priority level 3 unless attention interruptions are inhibited by the attachment interruption controls.

The attachment interruption controls, which permit or prevent attention and power/thermal interruptions to the 1130, are determined by the setting of modifier field bits 28-30 in the control IOCC.

Modifier bits 28-29 are the basic interruption controls; bit 30 is a temporary interruption control. Attention and power/thermal interruptions are permitted only if the basic status is "permit" and there is no temporary "prevent."

To establish the basic interruption control status, modifier field bits 28 and 29 are as follows:

Bit 28 = 0: Ignore bit 29.
Bit 28 = 1: Perform function specified by bit 29.
Bit 29 = 0: Permit attention and power/thermal interruptions.
Bit 29 = 1: Prevent attention and power/thermal interruptions.

Once such interruptions are prevented by this method, they are not permitted again until another control IOCC is executed with bit 28 set on and bit 29 set off, to alter the basic interruption control status.

To establish the temporary interruption control status, modifier field bit 30 is as follows:

Bit 30 = 0: No effect and basic control status prevails.
Bit 30 = 1: Prevent attention and power/thermal interruptions.

Once such interruptions are prevented by this method, they are not permitted again until one of the following occurs:

1. The DSW is reset.
2. Another control IOCC with modifier bit 30 set off is executed.
3. An initiate read or initiate write IOCC is executed.

At that time, the basic interruption control status prevails.

Attention and power/thermal interruptions to the 1130 are also prevented by either a System/7 power-on reset or an 1130 storage access channel reset. They are not permitted again until a control IOCC is executed with bit 28 on and bit 29 off.

ELECTRONIC INITIAL PROGRAM LOAD (EIPL)

An XIO with this IOCC is used to IPL the System/7 from the 1130. EIPL is simply an initiate write IOCC with modifier field bit 30 set on. When the 1130 attachment recognizes this command, the System/7 does a system reset and enters the wait state. The 1130 attachment sets the System/7 address to zero and proceeds with the EIPL as if it were a normal initiate write.

For an error-free termination, the System/7 instruction address register is set to a zero value, interruption level 3 is activated, and System/7 begins to execute instructions starting at location 0. A standard operation-end interrupt is generated in the 1130.

For an error termination, the System/7 is not informed of the IPL and a standard operation-end interrupt is generated in the 1130.

I/O modules are physical structures housing the total logic and signal processing functions for sensor-based I/O devices. Logic contained within the I/O module is divided into two parts:

1. Common controls.
2. Device logic and special controls that vary depending upon the type of sensor-based I/O device.

The common control logic governs data transfers between the I/O module and the processor module. Each I/O module interfaces to the direct control channel through the standard internal interface. All transfer of data and control information is accomplished by two-byte transfers using immediate commands. Upon reaching the end status of an I/O operation, or upon detecting an error condition, a priority interruption request can be set by the I/O module controls.

The 5014 Analog Input Module Model B01 converts analog signals into binary values of 14 bits plus sign. These analog signals can be within the range of ±10 mV full scale to ±5.12V full scale.

In addition to logic controls, this I/O module consists of an analog-to-digital converter (ADC), an amplifier, and a point-select relay multiplexer capable of operating at a maximum of 200 points per second. As many as 128 two-wire differential analog input points can be accommodated, in eight groups of 16 points each.

Systems can be configured with as many as 11 independent model B01 analog input modules. (See Figure 7.)

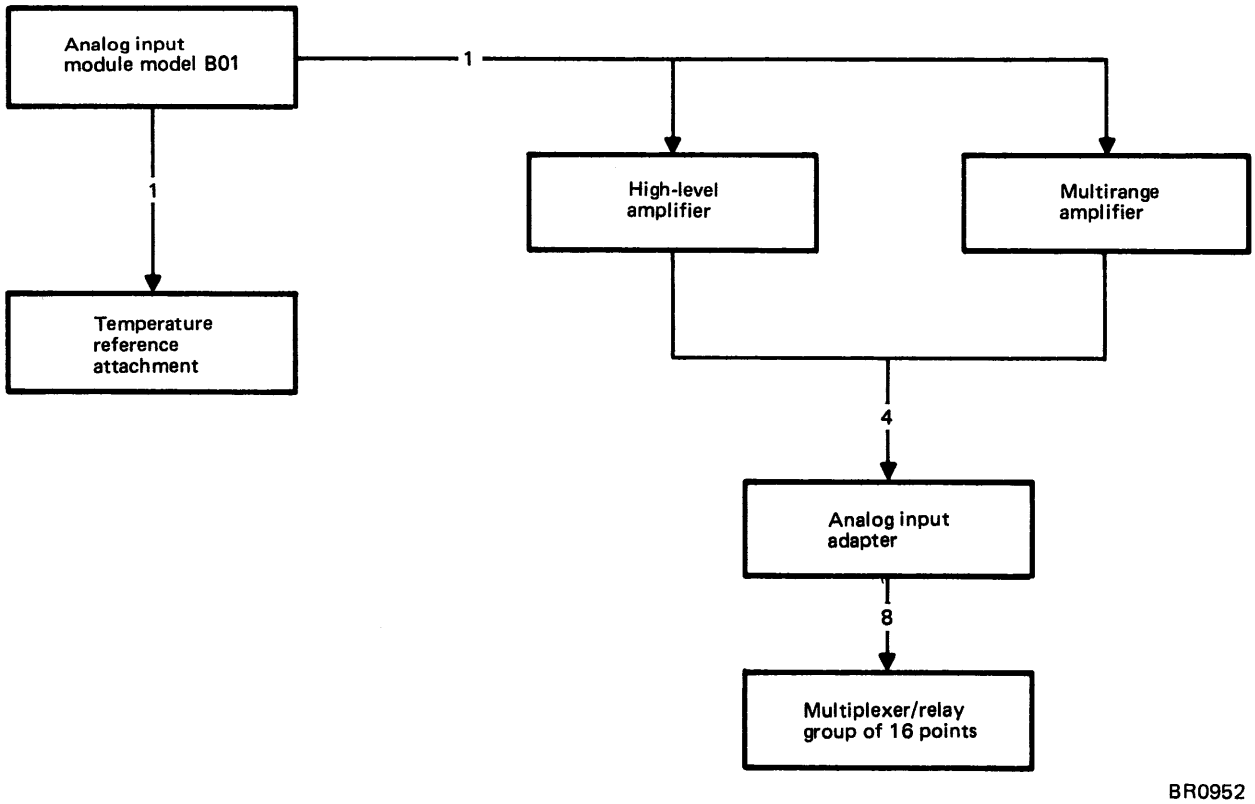


Figure 7. IBM 5014 Analog Input Module Model B01

One amplifier is required to service all of the analog input points. If all analog inputs are at the ±5.12V level, the high-level-only amplifier is used. If it is necessary to accommodate a voltage range, a multirange amplifier is used. This amplifier can be program-controlled to one of seven different ranges, or to automatically select the appropriate amplifier gain. With automatic gain selection, analog signal resolution is reduced to 12 bits plus sign because the selected gain is returned (binary encoded in three bits) to the program with the data. The multirange amplifier allows conversion of signals in the ±10 mV, ±20 mV, ±40 mV, ±80 mV, ±160 mV, ±640 mV, and ±5.12V ranges.

The analog-to-digital converter furnishes an overload indication to the control logic, and subsequently to the processor, when the signal level falls outside of a range that can be converted.

A temperature reference attachment is available that is capable of providing temperature reference for use in the cold junction compensation of thermocouple inputs.

STATUS WORDS

The analog input module model B01 has two status words for the identification of device conditions. Bits in the device status word (DSW) are set by error conditions occurring during the execution of an immediate command; bits in the interrupt status word (ISW) are set by conditions occurring after the immediate command is completed but while the device is busy. Therefore, the setting of the DSW bits is indicated to the program by the condition code but the setting of the ISW bits is indicated to the program by an interruption request. Interruptions are requested for errors detected in the analog-to-digital conversion cycles.

Device Status Word (DSW)

The bits in the 16-bit device status word are set as a result of detected error conditions occurring during the execution of an immediate command to the I/O module. The operating program is notified of these error conditions by returning a condition code of 1.

The program can examine the DSW to determine which error condition exists by issuing an immediate read DSW command, after which the DSW bits are reset. If other commands are attempted before resetting the DSW bits, a condition code of 1 continues to be returned to the program.

The meanings of the significant bits in the DSW are:

Significant Bits	*Meaning*
1	CR—command reject. The addressed module cannot execute the command, e.g., an interruption-causing command issued to a device disabled for interruptions.
9	ICC—interface control check. A parity error is detected on the interface involving a control field (i.e., module address, interruption request, or prepare I/O data). The operation that caused this condition can be retried, but the error may have already caused inconsistent results.
14	DCK—data check. A parity error involving data is detected on the interface. The operation that caused this condition usually can be retried successfully if the error condition is intermittent.

Interrupt Status Word (ISW)

The bits in the 16-bit interrupt status word are set by conditions occurring after an immediate write command is completed but while the device is still busy. Since these errors cannot be indicated to the program by the condition code, interruptions result from setting on any ISW bit except the busy bit.

The program can examine the ISW to determine which error condition exists by issuing an immediate read ISW command.

After the interruption is accepted, executing any command other than immediate read DSW resets the ISW.

The meanings of the significant bits in the ISW are:

Significant Bits	*Meaning*
3	ADCI—analog-to-digital converter inoperative.
5	MRE—multiple relays selected. A multiplexer failure because more than one input was selected during the conversion cycle.
6	NRE—no relays selected. A multiplexer failure because no input was selected during the conversion cycle.
7	OLD—overload. The voltage read by the ADC was greater than the input voltage range selected by the amplifier.
12	Device busy.
13	Device end. This bit is set on when the I/O operation is completed. If error conditions occur during the execution of such operations as analog-to-digital conversions, the device-end bit is also set on along with other error-identifying bits.
15	INSA—invalid subaddress. An input point was addressed that was not installed.

I/O COMMANDS

The prepare I/O and halt I/O commands are used as described under "I/O Commands" (direct control channel).

Data transfers between the analog input module and the processor module are accomplished by immediate read and write commands. The setting of the modifier field bits in the command further defines the operation to be performed.

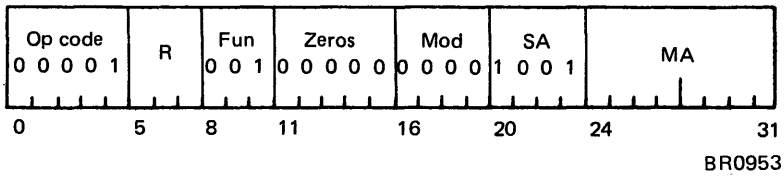
In all I/O commands (except read ID and read DSW) addressed to the analog input module, the subaddress (SA) field contents must be 1001. The module address (MA) varies depending upon the physical location of the I/O module.

Immediate Write Commands

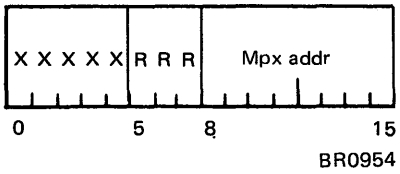
Usually, a complete immediate operation is performed within the time limit required for execution of the immediate commands, leaving no residue or aftereffects within the I/O module. Exceptions to this rule are two convert commands which start the conversion cycle and then permit the program to resume normal operation while awaiting the interruption request presented by the module at the end of the conversion cycle. This type of immediate command is indicated in discussions by a suffix of (int) which signifies "interruption."

Executing either of these two commands sets on the ISW device-busy bit which remains on during the conversion time. At the completion of conversion, an interruption occurs which turns off the device-busy bit and turns on the ISW device-end bit. The analog input module presents a busy signal to all immediate commands (except read ISW) issued while the device is busy or has a pending interruption.

Convert Normal Input (Int)



This command transmits data in the index register (R), or the accumulator if R = 000, to the analog input module. This data has the format:



The three RRR bits specify the input voltage range and amplifier gain desired:

RRR Bits	*Input Voltage Range*	*Amplifier Gain*
000	–––	Automatic gain
001	±10 mV	512
010	±20 mV	256
011	±40 mV	128
100	±80 mV	64
101	±160 mV	32
110	±640 mV	8
111	±5.12V	1

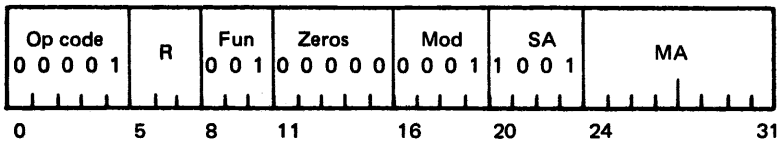
When automatic gain is specified (RRR bits = 000), the optimum gain (one of the seven amplifier gains) is selected automatically based upon the input signal voltage. The selected gain can be returned to the program, along with the transferred data, by issuing a read ADC command. Since the indication of gain requires three bits, the input data itself is truncated to 12 bits plus sign.

The range control bits (RRR) have no significance if the high-level-only gain amplifier is installed.

Convert normal input (int) starts an analog-to-digital conversion cycle of the analog input point specified by the multiplexer address (mpx addr). Interruption occurs at the end of the conversion cycle. Results are obtained by issuing an immediate-read-ADC or immediate-read-ADC-extended-precision command.

With the immediate-read-ADC command, the data format is in short form (12 bits plus sign) if automatic gain was specified or standard form (14 bits plus sign) if a programmed range was specified. The immediate-read-ADC-extended-precision command always returns the data in standard form.

Convert Normal Input With External Synchronization (Int)



BR0955

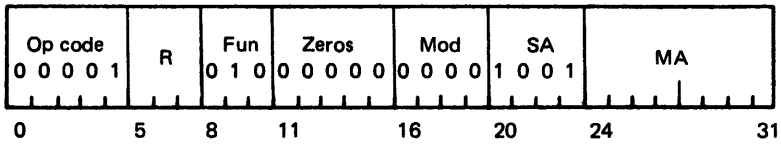
This command functions in the same manner as the convert normal input (int) command except that the start of conversion depends upon receipt of an external synchronization input pulse.

The user provides the external synchronization input pulses to control the speed of analog-to-digital data conversion. (Refer to the *System/7 Installation Manual–Physical Planning,* Order No. GA34-0004.)

Immediate Read Commands

Immediate read commands are used with the analog input module to obtain the binary output of the analog-to-digital converter, status information pertaining to the I/O device, and information identifying the I/O module.

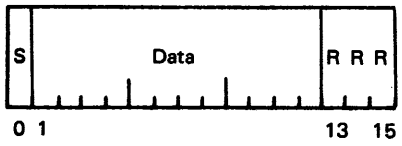
Read ADC



BR0956

The read ADC command stores the binary output value from the ADC into the index register (R), or accumulator if R = 000.

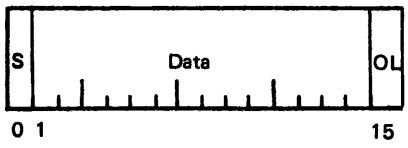
The data transferred to the register can have two formats, depending upon whether the last convert command requested automatic gain or selected a particular amplifier gain. If automatic gain was requested, the 16 bits of data have the format:



BR0957

Twelve bits of binary data plus sign (S bit) are returned. The RRR field indicates which amplifier gain was automatically selected by the amplifier. (Refer to the table in the "Convert Normal Input (Int)" description.) If RRR = 000, overload occurred with the automatic gain selection.

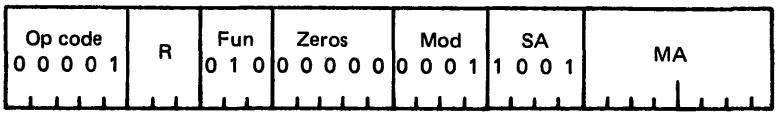
If a particular amplifier gain was selected by the last convert command, the 16 bits of data have the format:



BR0958

Fourteen bits of binary data plus sign (S bit) are returned. The single-bit OL field is set on if overload occurred with the programmed gain selection.

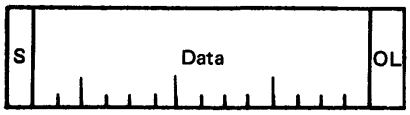
Read ADC Extended Precision



BR0959

The read-ADC-extended-precision command is used to store 14 bits of binary data plus sign when automatic amplifier gain was selected by the last convert command.

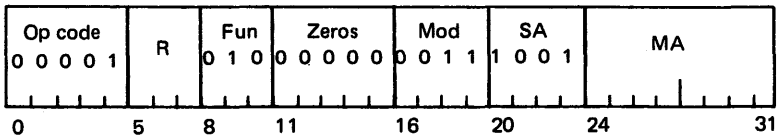
The ADC output value resulting from the last ADC cycle is placed in the index register (R), or accumulator if R = 000. The data transferred to the register has the format:



BR0960

Fourteen bits of binary data plus sign (S bit) are returned. The single-bit OL field is set on if overload occurred with the automatic amplifier gain selection. Note that the automatically selected gain is not returned with the data. This gain can be determined by issuing a read ADC command before or after the read-ADC-extended-precision command.

Read ISW

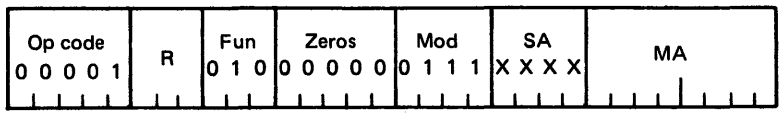


BR0961

The read ISW command stores the 16-bit interrupt status word (ISW) associated with the analog input module into the index register (R), or accumulator if R = 000.

If the interruption that results from setting an ISW bit is accepted, subsequent execution of any command (except read DSW) directed to the module resets the ISW.

Read DSW

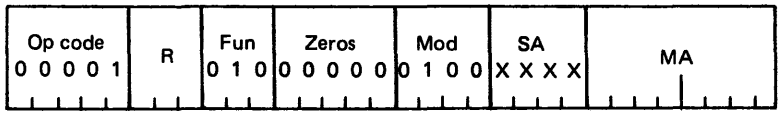


BR0962

The read DSW command stores the 16-bit device status word (DSW) associated with the analog input module into the index register (R), or accumulator if R = 000.

Execution of this command resets the bits in the DSW.

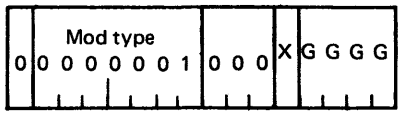
Read ID



BR0963

The read identification command stores 16 bits of I/O module identifying information into the index register (R), or accumulator if R = 000.

The identifying information has the format:



BR0964

Normally, bit 0 is used to indicate that additional module identifying information exists in an ID extension word. However, this bit is always set to 0 because the analog input module model B01 does not require an ID extension word.

The type of I/O module is indicated by the mod type field of 0000001, which is the coded designation of the analog input module model B01.

Bits 8-10 are always set to 0's.

If bit 11 is set on, it indicates the presence of the multirange amplifier in the module.

Bits 12-15 contain the binary number of analog input groups installed in the module. Each group has 16 points, and a maximum of eight groups can be accommodated. Groups must be installed sequentially; thus, a binary 0011 specifies that groups 1, 2, and 3 are installed.

I/O modules are physical structures housing the total logic and signal processing functions for sensor-based I/O devices. Logic contained within the I/O module is divided into two parts:

1. Common controls.
2. Device logic and special controls that vary depending upon the type of sensor-based I/O device.

The common control logic governs data transfer between the I/O module and processor module. Each I/O module interfaces to the direct control channel through the standard internal interface. All transfer of data and control information is accomplished by two-byte transfers using immediate commands. Upon reaching the end status of an I/O operation, or upon detecting an error condition, a priority interruption request can be set by the I/O module controls.

The 5014 Analog Input Module Model C01 converts analog signals into binary values of 14 bits plus sign. These analog signals can be within the range of ±10 mV full scale to ±5.12V full scale. In addition to logic controls, this I/O module consists of an analog-to-digital converter (ADC), an amplifier, and a differential solid-state multiplexer.

The solid-state multiplexer can have as many as 128 two-wire differential analog input points, in eight groups of 16 points each. With high-level inputs (±5.12V), a speed of 20,000 points per second can be attained. Input speed of 14,000 points per second can be achieved with low-level inputs (10-640 mV). Use of automatic gain selection reduces input speeds to a maximum of 7,000 points per second.

Systems can be configured with as many as 11 independent model C01 analog input modules. (See Figure 8.)

One amplifier is required to service all of the analog input points. If all analog inputs are at the ±5.12V level, the high-level-only amplifier is used. If it is necessary to service a voltage range, a multirange amplifier is used. This amplifier can be program-controlled to one of seven different ranges, or to automatically select the appropriate amplifier gain. With automatic gain selection, analog input resolution is reduced to 12 bits plus sign because the selected gain is returned (binary encoded in three bits) to the program with the data. The multirange amplifier allows conversion of signals in the ±10 mV, ±20 mV, ±40 mV, ±80 mV, ±160 mV, ±640 mV, and ±5.12V ranges.

The analog-to-digital converter furnishes an overload indication to the control logic, and subsequently to the processor, when the signal level falls outside of a range that can be converted.

A temperature reference attachment is available that is capable of providing temperature reference for use in the cold junction compensation of thermocouple inputs.

STATUS WORDS

The analog input module model C01 has two status words for the identification of device conditions. Bits in the device status word (DSW) are set by error conditions occurring during the execution of an immediate command; bits in the interrupt status word are set by conditions occurring after the immediate command is completed but while the device is still busy. Therefore, the setting of the DSW bits is indicated to the program by the condition code but the setting of the ISW bits is indicated to the program by an interruption request. Interruptions are requested for errors detected in the analog-to-digital conversion cycles.

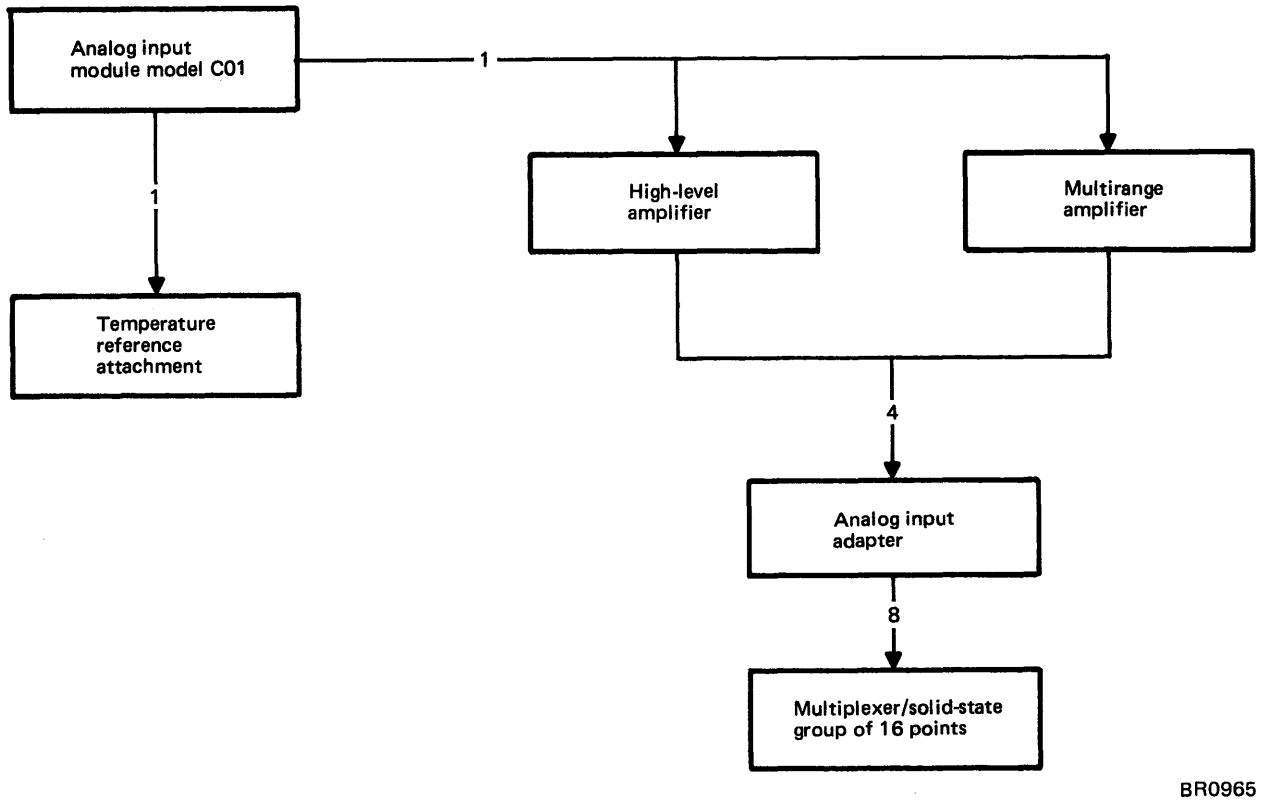


Figure 8. IBM 5014 Analog Input Module Model C01

Device Status Word (DSW)

The bits in the 16-bit device status word are set as a result of detected error conditions occurring during the execution of an immediate command to the I/O module. The operating program is notified of these error conditions by returning a condition code of 1.

The program can examine the DSW to determine which error condition exists by issuing an immediate read DSW command, after which the DSW bits are reset. If other commands are attempted before resetting the DSW bits, a condition code of 1 continues to be returned to the program.

The meanings of the significant bits in the DSW are:

Significant Bits	*Meaning*
1	CR–command reject. The addressed module cannot execute the command, e.g., an interruption-causing command issued to a device disabled for interruptions.
9	ICC–interface control check. A parity error is detected on the interface involving a control field (i.e., module address, interruption request, or prepare I/O data). The operation that caused this condition can be retried, but the error may have already caused inconsistent results.
14	DCK–data check. A parity error involving data is detected on the interface. The operation that caused this condition usually can be retried successfully if the error condition is intermittent.

Interrupt Status Word (ISW)

The bits in the 16-bit interrupt status word are set by conditions occurring after an immediate write command is completed but while the device is still busy. Since these errors cannot be indicated to the program by the condition code, interruptions result from setting on any ISW bit except the busy bit.

The program can examine the ISW to determine which error condition exists by issuing an immediate read ISW command.

After the interruption is accepted, executing any command other than immediate read DSW resets the ISW.

The meanings of the significant bits in the ISW are:

Significant Bits	*Meaning*
3	ADCI—analog-to-digital converter inoperative.
5	IAD—invalid analog data. The input voltage polarity changed states during the conversion.
7	OLD—overload. The voltage read by the ADC was greater than the input voltage range selected by the amplifier.
12	Device busy.
13	Device end. This bit is set on when the I/O operation is completed. If error conditions occur during the execution of such operations as analog-to-digital conversions, the device-end bit is also set on along with other error-identifying bits.
15	INSA—invalid subaddress. An input point was addressed that was not installed.

I/O COMMANDS

The prepare I/O and halt I/O commands are used as described under "I/O Commands" (direct control channel).

Data transfers between the analog input module and the processor module are accomplished by immediate read and write commands. The setting of the modifier field bits in the command further defines the operation to be performed.

In all I/O commands (except read ID and read DSW) addressed to the analog input module, the subaddress (SA) field contents must be 1001. The module address (MA) varies depending upon the physical location of the module.

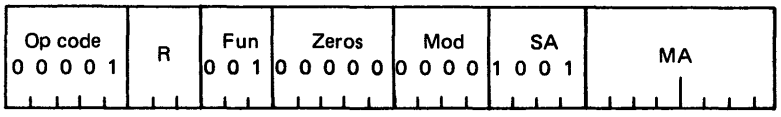
Usually, a complete immediate operation is performed within the time limit required for execution of the immediate commands, leaving no residue or aftereffects within the I/O module. Exceptions to this rule are three convert commands which start the conversion cycle and permit the program to resume normal operation while awaiting the interruption request presented by the module at the end of the conversion cycle. This type of immediate command is indicated in discussions by a suffix of (int) which signifies "interruption."

Executing any of these three commands sets on the ISW device-busy bit which remains on during the conversion time. At the completion of conversion, an interruption occurs which turns off the device-busy bit and turns on the ISW device-end bit. The analog input module presents a busy signal to all immediate commands (except read ISW) issued while the device is busy or has a pending interruption.

Immediate Write Commands

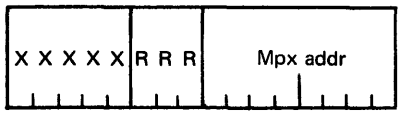
Immediate write commands are issued to start analog-to-digital conversion cycles of specifically addressed analog input points.

Convert Normal Input (Int)



BR0966

This command transmits data in the index register (R), or accumulator if R = 000, to the analog input module. This data has the format:



BR0967

The three RRR bits specify the input voltage range and amplifier gain desired:

RRR Bits	*Input Voltage Range*	*Amplifier Gain*
000	---	Automatic gain
001	±10 mV	512
010	±20 mV	256
011	±40 mV	128
100	±80 mV	64
101	±160 mV	32
110	±640 mV	8
111	±5.12V	1

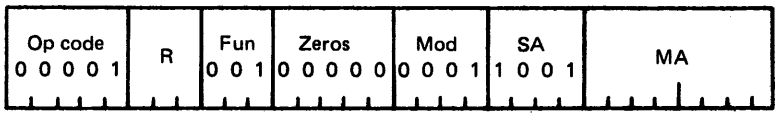
When automatic gain is specified (RRR bits = 000), the optimum gain (one of the seven amplifier gains) is selected automatically based upon the input signal voltage. The selected gain can be returned to the program, along with the transferred data, by issuing a read ADC command. Since the indication of gain requires three bits, the input data itself is truncated to 12 bits plus sign.

The range control bits (RRR) have no significance if the high-level-only gain amplifier is installed.

Convert normal input (int) starts an analog-to-digital conversion cycle of the analog input point specified by the multiplexer address (mpx addr). Interruption occurs at the end of the conversion cycle. Results are obtained by issuing an immediate-read-ADC, an immediate-read-ADC-extended-precision, or a read-and-convert-ADC (int) command.

With either of the latter two commands, the format of the returned data is in short form (12 bits plus sign) if automatic gain was specified or standard form (14 bits plus sign) if a programmed range was specified. The immediate-read-ADC-extended-precision command always returns data in standard form.

Convert Normal Input With External Synchronization (Int)



BR0968

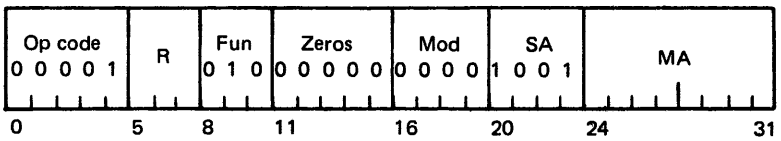
This command functions in the same manner as the convert normal input (int) command except that the start of conversion depends upon receipt of an external synchronization input pulse.

The user provides the external synchronization input pulses to control the speed of analog-to-digital data conversion. (Refer to the *System/7 Installation Manual–Physical Planning,* Order No. GA34-0004.)

Immediate Read Commands

Immediate read commands are used with the analog input module to obtain the binary output of the analog-to-digital converter, status information pertaining to the I/O device, and information identifying the I/O module.

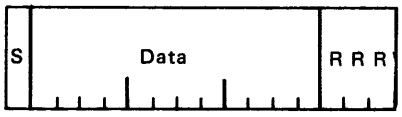
Read ADC



BR0969

The read ADC command stores the binary output value from the ADC into the index register (R), or accumulator if R = 000.

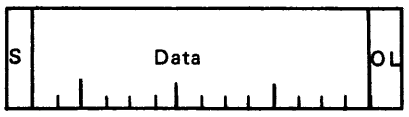
The data transferred to the register can have two formats, depending upon whether the last convert command requested automatic gain or selected a particular amplifier gain. If automatic gain was requested, the 16 bits of data have the format:



BR0970

Twelve bits of binary data plus sign (S bit) are returned. The RRR field indicates which amplifier gain was automatically selected by the amplifier. (Refer to the table in the "Convert Normal Input (Int)" description.) If RRR = 000, overload occurred with the automatic gain selection.

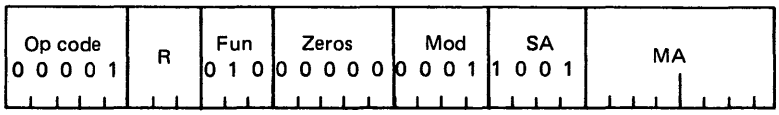
If a particular amplifier gain was selected by the last convert command, the 16 bits of data have the format:



BR0971

Fourteen bits of binary data plus sign (S bit) are returned. The single-bit OL field is set on if overload occurred with the programmed gain selection.

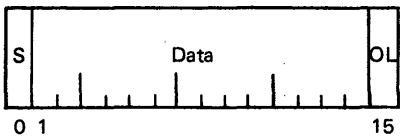
Read ADC Extended Precision



BR0972

The read-ADC-extended-precision command is used to store 14 bits of binary data plus sign when automatic amplifier gain was selected by the last convert command.

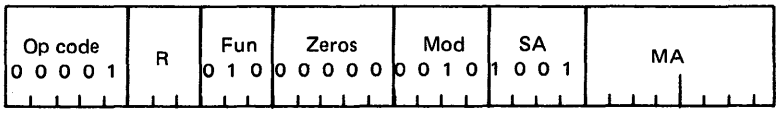
The ADC output value resulting from the last ADC cycle is placed in the index register (R), or accumulator if R = 000. The data transferred to the register has the format:



BR0973

Fourteen bits of binary data plus sign (S bit) are returned. The single-bit OL field is set on if overload occurred with the automatic amplifier gain selection. Note that the automatically selected gain is not returned with the data. This gain can be determined by issuing a read ADC command before or after the read-ADC-extended-precision command.

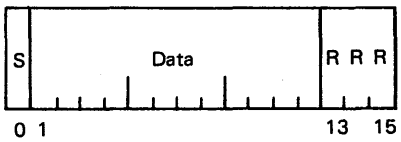
Read and Convert ADC (Int)



BR0974

This command is used for repetitive operations on a single analog input point. The read-and-convert-ADC (int) command stores the binary output value from the ADC into the index register (R), or accumulator if R = 000.

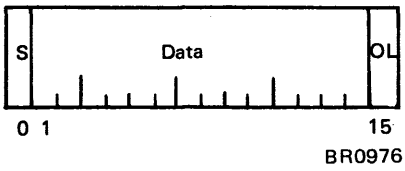
Data transferred to the register can have two formats, depending upon whether the last convert command requested automatic gain or selected a particular amplifier gain. If automatic gain was requested, the 16 bits of data have the format:



BR0975

Twelve bits of binary data plus sign (S bit) are returned. The RRR field indicates which amplifier gain was automatically selected by the amplifier. (Refer to the table in the "Convert Normal Input (Int)" description.) If RRR = 000, overload occurred with the automatic gain selection.

If a particular amplifier gain was selected by the last convert command, the 16 bits of data have the format:

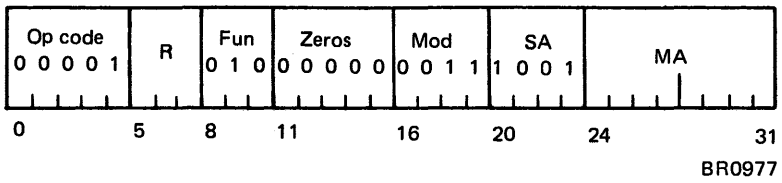


Fourteen bits of binary data plus sign (S bit) are returned. The single-bit OL field is set on if overload occurred with the programmed amplifier gain selection.

The convert portion of the read-and-convert-ADC (int) command converts the analog input point specified by the last convert command, utilizing the same amplifier gain. Thus, if automatic gain was specified by the last convert command, then automatic gain is also used by each subsequent execution of the read-and-convert-ADC (int) command.

Since this command performs repetitive operations on a single analog input point, the solid-state multiplexer is monopolized during this time. To terminate this repetitive read sequence and free the multiplexer, a read ADC command must be issued.

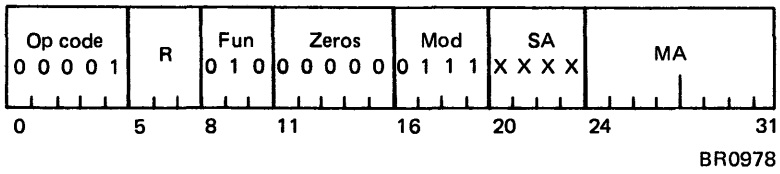
Read ISW



The read ISW command stores the 16-bit interrupt status word (ISW) associated with the analog input module into the index register (R), or accumulator if R = 000.

If the interruption that results from setting an ISW bit is accepted, subsequent execution of any command (except read DSW) directed to the module resets the ISW.

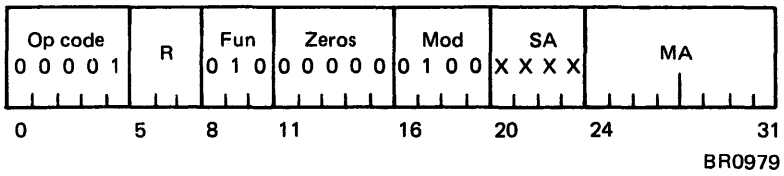
Read DSW



The read DSW command stores the 16-bit device status word (DSW) associated with the analog input module into the index register (R), or accumulator if R = 000.

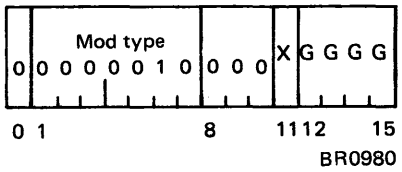
Execution of this command resets the bits in the DSW.

Read ID



The read identification command stores 16 bits of I/O module identifying information into the index register (R), or accumulator if R = 000.

The identifying information has the format:



Normally, bit 0 is used to indicate that additional module identifying information exists in an ID extension word. However, this bit is always set to 0 because the analog input module model C01 does not require an ID extension word.

The type of I/O module is indicated by the mod type field of 0000010, which is the coded designation of the analog input module model C01.

Bits 8-10 are always set to 0's.

If bit 11 is set on, it indicates the presence of the multirange amplifier in the module.

Bits 12-15 contain the binary number of analog input groups installed in the module. Each group has 16 points, and a maximum of eight groups can be accommodated. Groups must be installed sequentially; thus, a binary 0011 specifies that groups 1, 2, and 3 are installed.

I/O modules are physical structures housing the total logic and signal processing functions for sensor-based I/O devices. Logic contained within the module is divided into two parts:

1. Common controls.
2. Device logic and special controls that vary depending upon the type of sensor-based I/O device.

The common control logic governs data transfer between the I/O module and processor module. Each I/O module interfaces to the direct control channel through the standard internal interface. All transfer of data and control information is accomplished by two-byte transfers using immediate commands. Upon reaching the end status of an I/O operation, or upon detecting an error condition, a priority interruption request can be set by the I/O module controls.

The 5012 Multifunction Module Model A01 groups the following functions within one physical enclosure:

- As many as 32 analog input points (points installed must be either all model B01 or all model C01).
- As many as 128 digital input points.
- As many as 64 digital output points.
- As many as two analog output points.
- One 2790 control to attach the IBM 2790 Data Communication System.

To suit installation requirements, any combination of the above functions can be chosen for inclusion within each multifunction module. However, programming support exists for only one 2790 control in a System/7 configuration.

Systems can be configured with as many as 11 independent multifunction modules. (See Figure 9.)

Analog Input Control Model B01: Consists of an analog-to-digital converter, an amplifier, and a point-select relay multiplexer capable of operating at a maximum of 200 points per second.

As many as 32 two-wire differential analog input points can be accommodated, in eight groups of four points each.

Analog Input Control Model C01: Consists of an analog-to-digital converter, an amplifier, and a differential solid-state multiplexer.

The solid-state multiplexer can have as many as 32 two-wire differential analog input points, in eight groups of four points each. With high-level inputs (±5.12V), a speed of 20,000 points per second can be attained. Input speed of 14,000 points per second can be achieved with low-level inputs (10-640 mV). Use of automatic gain selection reduces input speeds to a maximum of 7,000 points per second.

Digital Input: As many as 128 digital input points can be accommodated, in eight groups of 16 points each. These points are completely isolated two-terminal inputs that can be activated by either contact or voltage sources provided by the user equipment.

The first two digital input groups can each be provided with a special feature that permits them to interrupt the System/7 processor. A 16-point group is compared with a 16-bit reference register, and interruptions can be initiated on the basis of either an equal or unequal comparison.

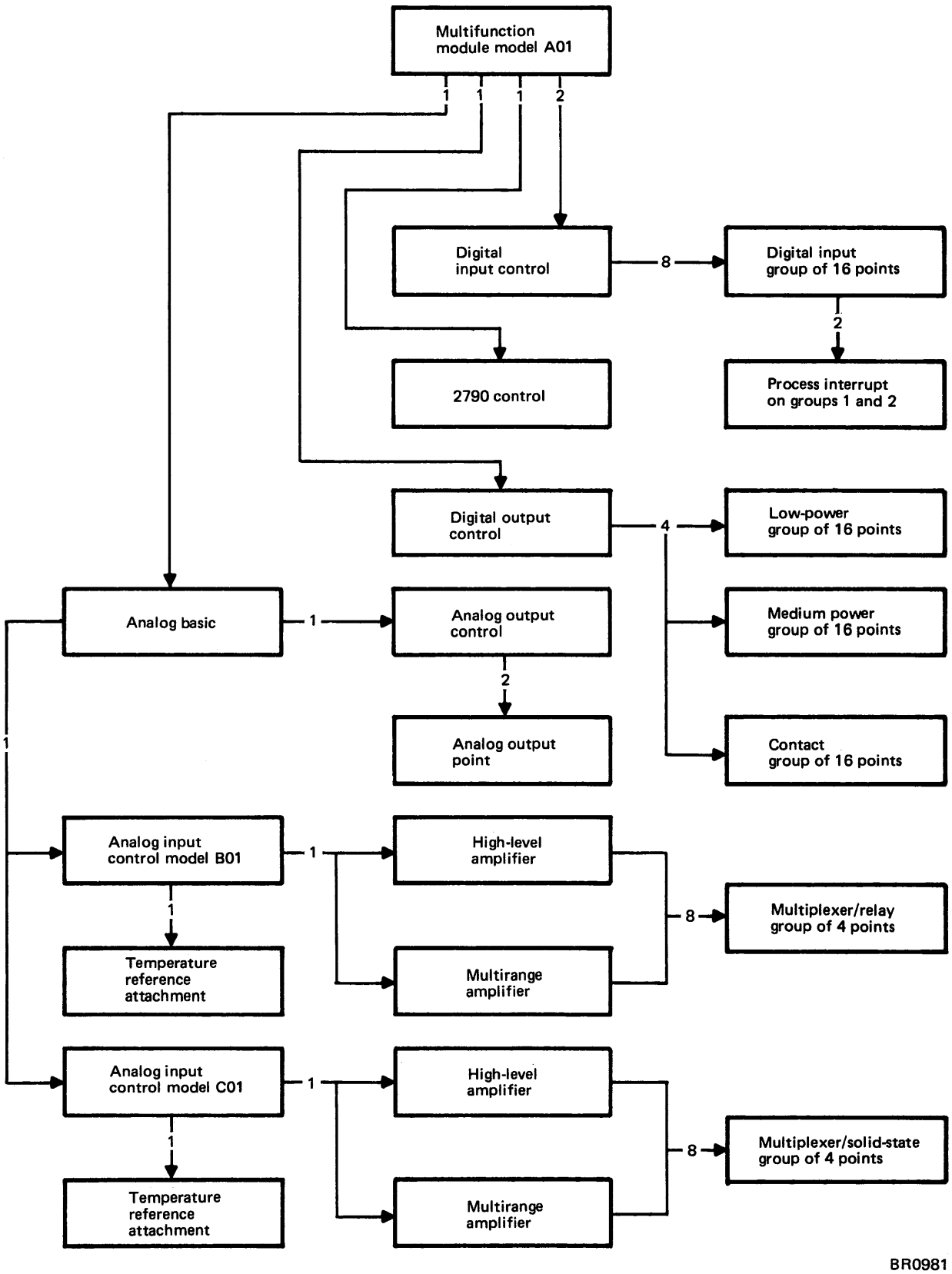


Figure 9. IBM 5012 Multifunction Module Model A01

Digital Output: As many as 64 digital output points can be accommodated, in four groups of 16 points each. Each group can be configured to use one type of the three optional output circuits available (DO low-power group, DO medium-power group, and DO contact group).

Analog Output: One or two isolated analog output voltage points can be accommodated within the multifunction module. Each point provides a voltage output to user terminals.

2790 Control: Permits attachment of a single loop of the IBM 2790 Data Communication System to the System/7. This configuration allows the System/7, or an 1130 host processor, to act as system controller to the 2790 devices (2791/2793 Area Stations and 2795/2796 Data Entry Units). Input to the 2790 system is through the card reader, badge reader, or key entry units. Output is by visual display or a printer.

DEVICE STATUS WORD (DSW)

The multifunction module has a single DSW for storing device status information. The bits in the 16-bit DSW are set as a result of detected error conditions occurring during the execution of immediate commands to the multifunction module. Notice of these error conditions is given to the operating program by returning a condition code of 1.

The program can examine the DSW to determine which error condition exists by issuing an immediate-read-DSW command, after which the DSW bits are reset. If other commands are attempted before resetting the DSW bits, a condition code of 1 continues to be returned to the program.

The meanings of the significant bits in the multifunction module DSW are:

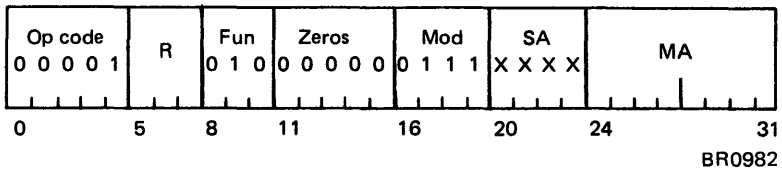
Significant Bits	*Meaning*
1	CR—command reject. The multifunction module cannot execute the command, e.g., an interruption-causing command issued to a device disabled for interruptions.
3	OPF—open fuse. This bit is set on when a digital input fuse is open and a digital input subaddress is selected.
9	ICC—interface control check. A parity error is detected on the interface involving a control field (i.e., module address, interruption request, or prepare I/O data). The operation that caused this condition can be retried, but the error may have already caused inconsistent results.
14	DCK—data check. A parity error is detected on the interface involving data. The operation that caused this condition usually can be retried successfully if the error condition is intermittent.
15	INSA—invalid subaddress.

COMMON I/O COMMANDS

The multifunction module is programmed by immediate I/O commands. Those that read or write program data are discussed individually with each of the functions performed by the multifunction module. However, there are three commands that read control information that is applicable to the entire module, such as device status and module identification.

In these three commands, the contents of the subaddress (SA) field are ignored but the module address (MA) is determined by the physical location of the multifunction module.

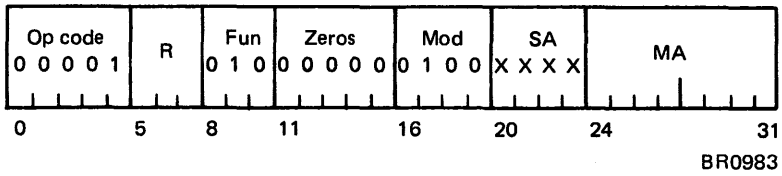
Read DSW



The read DSW command stores the 16-bit device status word (DSW) associated with the multifunction module into the index register (R), or accumulator if R = 000.

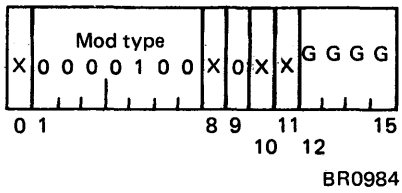
Execution of this command resets the bits in the DSW.

Read ID



The read identification command stores 16 bits of I/O module identifying information into the index register (R), or the accumulator if R = 000.

The identifying information has the format:



Bit 0 is set on if additional identifying information exists in an ID extension word, which can be obtained by issuing an immediate-read-ID-extension command.

The type of I/O module is indicated by the mod type field of 0000100, which is the coded designation of the multifunction module.

Bit 8 is set on if the 2790 control function is included in the multifunction module.

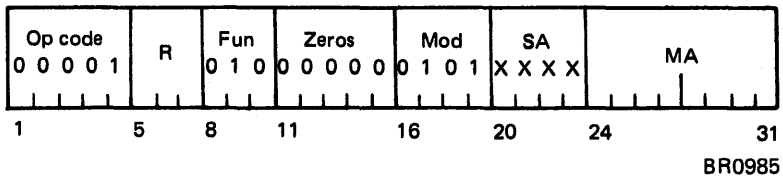
Bit 9 is not used but is set to 0.

Bit 10 is set on if the analog input control model C01 is included in the multifunction module. If this bit is off, and GGGG (bits 12-15) is not equal to 0000, analog input control model B01 is included.

Bit 11 is set on when a multirange amplifier is used with the analog input control.

Bits 12-15 contain the binary number of analog input groups (either all model B01 or all model C01) installed in the multifunction module. Each group has four points, and a maximum of eight groups can be accommodated. Groups must be installed sequentially; thus, a binary 0011 specifies that groups 1, 2, and 3 are installed.

Read ID Extension



The read-ID-extension command stores 16 bits of multifunction module identifying information into the index register (R), or accumulator if R = 000.

Each bit is set on or off to indicate the presence or absence, respectively, of input or output groups. The following list indicates the meaning of each bit when it is on:

Bit	*Meaning*
0	Digital input group 1 has interrupt feature
1	Digital input group 0 has interrupt feature
2	Digital output group 3 is present
3	Digital output group 2 is present
4	Digital output group 1 is present
5	Digital output group 0 is present
6	Analog output point 1 is present
7	Analog output point 0 is present
8	Digital input group 7 is present
9	Digital input group 6 is present
10	Digital input group 5 is present
11	Digital input group 4 is present
12	Digital input group 3 is present
13	Digital input group 2 is present
14	Digital input group 1 is present
15	Digital input group 0 is present

SUBADDRESSES

The four-bit subaddress field in I/O commands is used to specify the particular multifunction module feature being addressed. The subaddress bit assignments for each feature are:

Subaddress (SA)	*Multifunction Module Feature*
0000	Digital input group 0 with or without interrupt feature
0001	Digital input group 1 with or without interrupt feature
0010	Digital input group 2
0011	Digital input group 3
0100	Digital input group 4
0101	Digital input group 5
0110	Digital input group 6
0111	Digital input group 7
1000	2790 control
1001	Analog input
1010	Analog output point 0
1011	Analog output point 1
1100	Digital output group 0
1101	Digital output group 1
1110	Digital output group 2
1111	Digital output group 3

The prepare I/O command functions as described under "I/O Commands" (direct control channel). However, the prepare I/O command directed to the multifunction module must use the following subaddresses to perform the indicated function, i.e., prepare an input source to interrupt:

Subaddress (SA)	*Function*
0000	Prepare digital input group 0 with interrupt feature
0001	Prepare digital input group 1 with interrupt feature
1000	Prepare 2790 control
1001	Prepare analog input

ANALOG INPUT CONTROL MODEL B01

In the multifunction module, model B01 and C01 analog inputs are mutually exclusive.

The analog input control model B01 converts analog signals into binary values of 14 bits plus sign. These analog signals can be within the range of ±10 mV full scale to ±5.12V full scale.

The analog input function is accomplished by an analog-to-digital converter (ADC), an amplifier, and a point-select relay multiplexer capable of operating at a maximum of 200 points per second.

As many as 32 two-wire differential analog input points can be accommodated, in eight groups of four points each.

One amplifier is required to service all of the analog input points. If all analog inputs are at the ±5.12V level, the high-level-only amplifier is used. If it is necessary to accommodate a voltage range, a multirange amplifier is used. This amplifier can be program-controlled to one of seven different ranges, or to automatically select the appropriate amplifier gain. With automatic gain selection, analog signal resolution is reduced to 12 bits plus sign because the gain selected is returned (binary encoded in three bits) to the program with the data. The multirange amplifier allows conversion of signals in the ±10 mV, ±20 mV, ±40 mV, ±80 mV, ±160 mV, ±640 mV, and ±5.12V ranges.

The analog-to-digital converter furnishes an overload indication to the module control logic, and subsequently to the processor, when the signal level falls outside of a range that can be converted.

A temperature reference attachment is available that is capable of providing temperature reference for use in the cold junction compensation of thermocouple inputs.

Interrupt Status Word (ISW)

The analog input control model B01 has its own interrupt status word (ISW) for identifying interruption conditions. Bits in the ISW are set by conditions occurring after an immediate write command is completed but while the device is still busy (after starting a convert operation). The setting of the ISW bits (except the busy bit) is indicated to the operating program by an interruption request. Interruption requests are presented for errors detected in analog-to-digital conversion cycles.

The program can examine the ISW to determine which error condition exists by issuing an immediate-read-ISW command.

After the interruption is accepted, the execution of any command directed to the analog input control resets the ISW.

The meanings of the significant bits in the ISW are:

Significant Bits	*Meaning*
3	ADCI—analog-to-digital converter inoperative.
5	MRE—multiple relays selected. A multiplexer failure because more than one input was selected during the conversion cycle.
6	NRE—no relays selected. A multiplexer failure because no input was selected during the conversion cycle.
7	OLD—overload. The voltage read by the ADC was greater than the input voltage range selected by the amplifier.
12	Device busy.
13	Device end. This bit is set on when the input/output operation is completed. If error conditions occur during the execution of such operations as analog-to-digital conversions, the device-end bit is also set on along with other error-identifying bits.
15	INSA—invalid subaddress. An input point was addressed that was not installed.

I/O Commands

The prepare I/O and halt I/O commands are used as described under "I/O Commands" (direct control channel).

Data transfers between the analog input control and the processor module are accomplished by immediate read and write commands. The setting of the modifier field bits in the command further defines the operation to be performed.

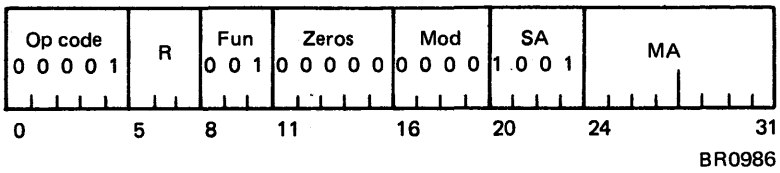
In all I/O commands addressed to analog input, the subaddress (SA) field contents must be 1001. The module address (MA) varies depending upon the physical location of the multifunction module.

Immediate Write Commands

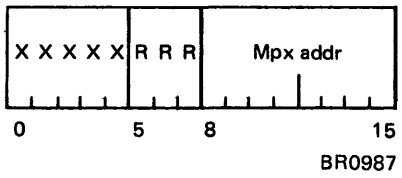
Usually, a complete immediate operation is performed within the time limit required for executing the immediate commands, leaving no residue or aftereffects within the I/O module. Exceptions to this rule are two convert commands which start the conversion cycle and then permit the program to resume normal operation while awaiting the interruption request presented by the module at the end of the conversion cycle. This type of immediate command is indicated in discussions by a suffix of (int) which signifies "interruption."

Executing either of these two commands sets on the ISW device-busy bit which remains on during the conversion time. At the completion of conversion, an interruption occurs which turns off the device-busy bit and turns on the ISW device-end bit. The analog input function presents a busy signal to all immediate commands (except read ISW) issued as long as the device is busy or has a pending interruption.

Convert Normal Input (Int)



This command transmits data in the index register (R), or accumulator if R = 000. This data has the format:



The three RRR bits specify the input voltage range and amplifier gain desired:

RRR Bits	*Input Voltage Range*	*Amplifier Gain*
000	----	Automatic gain
001	±10 mV	512
010	±20 mV	256
011	±40 mV	128
100	±80 mV	64
101	±160 mV	32
110	±640 mV	8
111	±5.12V	1

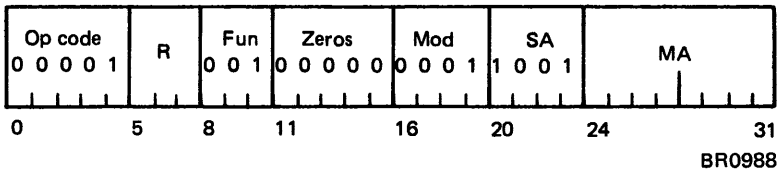
When automatic gain is specified (RRR bits = 000), the optimum gain (one of the seven amplifier gains) is selected automatically based upon the input signal voltage. The selected gain can be returned to the program, along with the transferred data, by issuing a read ADC command. Since the indication of gain requires three bits, the input data itself is truncated to 12 bits plus sign.

The range control bits (RRR) have no significance if the high-level-only gain amplifier is installed.

Convert normal input (int) starts an analog-to-digital conversion cycle of the analog input point specified by the multiplexer address (mpx addr). Interruption occurs at the end of the conversion cycle.

Results are obtained by issuing an immediate-read-ADC or immediate-read-ADC-extended-precision command. With the immediate-read-ADC command, the data format is in short form (12 bits plus sign) if automatic gain was specified or standard form (14 bits plus sign) if a programmed range was specified. The immediate-read-ADC-extended-precision command always returns the data in standard form.

Convert Normal Input With External Synchronization (Int)



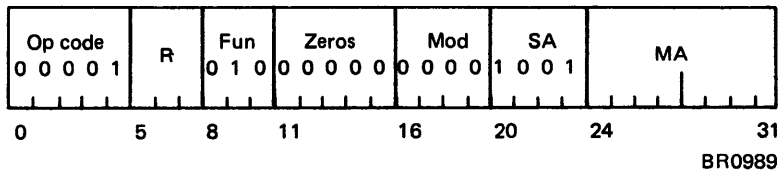
This command functions in the same manner as the convert-normal-input (int) command except that the start of conversion depends upon receipt of an external synchronization input pulse.

The user provides the external synchronization input pulses to control the speed of analog-to-digital data conversion. (Refer to *System/7 Installation Manual–Physical Planning,* Order No. GA34-0004.)

Immediate Read Commands

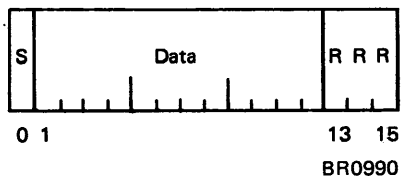
Immediate read commands are used with the analog input function to obtain the binary output of the analog-to-digital converter and status information pertaining to the I/O device.

Read ADC



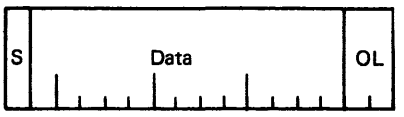
The read ADC command stores the binary output value from the ADC into the index register (R), or accumulator if R = 000.

The data transferred to the register can have two formats, depending upon whether the last convert command requested automatic gain or selected a particular amplifier gain. If automatic gain was requested, the 16 bits of data have the format:



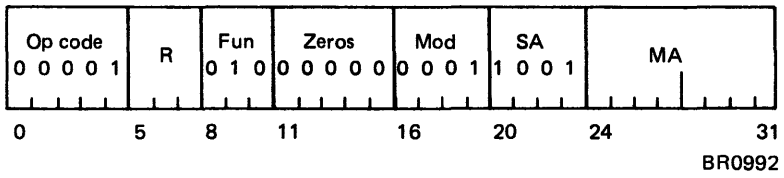
Twelve bits of binary data plus sign (S bit) are returned. The RRR field indicates which amplifier gain was automatically selected by the amplifier. (Refer to the table in the "Convert Normal Input (Int)" description.) If RRR = 000, overload occurred with the automatic gain selection.

If a particular amplifier gain was selected by the last convert command, the 16 bits of data have the format:



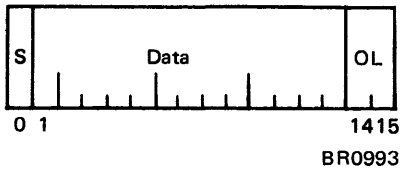
Fourteen bits of binary data plus sign (S bit) are returned. The single-bit OL field is set on if overload occurred with the programmed gain selection.

Read ADC Extended Precision



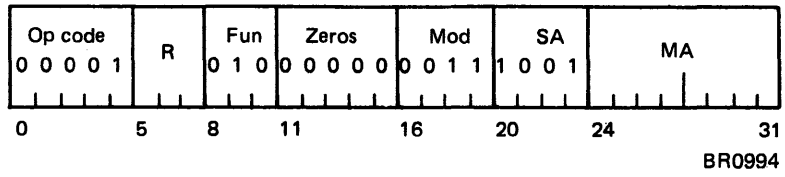
The read-ADC-extended-precision command stores 14 bits of binary data plus sign when automatic amplifier gain was selected by the last convert command.

The ADC output value resulting from the last ADC cycle is placed in the index register (R), or accumulator if R = 000. The data transferred to the register has the format:



Fourteen bits of binary data plus sign (S bit) are returned. The single-bit OL field is set on if the overload occurred with the automatic amplifier gain selection. Note that the automatically selected gain is not returned with the data. This gain can be determined by issuing a read ADC command before or after the read-ADC-extended-precision command.

Read ISW



The read ISW command stores the 16-bit interrupt status word (ISW) associated with the analog input control into the index register (R), or accumulator if R = 000.

If the interruption that results from setting an ISW bit is accepted, subsequent execution of any command directed to the analog input control resets the ISW.

ANALOG INPUT CONTROL MODEL C01

In the multifunction module, model B01 and C01 analog inputs are mutually exclusive.

The model C01 analog input control converts analog signals into binary values of 14 bits plus sign. These analog signals can be within the range of ±10 mV full scale to ±5.12V full scale.

The analog input function is accomplished by an analog-to-digital converter (ADC), an amplifier, and a differential solid-state multiplexer.

The solid-state multiplexer can have as many as 32 two-wire differential analog input points, in eight groups of four points each. With high-level inputs (±5.12V), a speed of 20,000 points per second can be attained. Input speed of 14,000 points per second can be achieved with low-level inputs (10-640 mV). Use of automatic gain selection reduces input speeds to a maximum of 7,000 points per second.

One amplifier is required to service all the analog input points. If all analog inputs are at the ±5.12V level, the high-level-only amplifier is used. If it is necessary to service a voltage range, a multirange amplifier is used. This amplifier can be program-controlled to one of seven different ranges, or to automatically select the appropriate amplifier gain. With automatic gain selection, analog input resolution is reduced to 12 bits plus sign because the gain selected is returned (binary encoded in three bits) to the program with the data. The multirange amplifier allows conversion of signals in the ±10 mV, ±20 mV, ±40 mV, ±80 mV, ±160 mV, ±640 mV, and ±5.12V ranges.

The analog-to-digital converter furnishes an overload indication to the control logic, and subsequently to the processor, when the signal level falls outside of a range that can be converted.

A temperature reference attachment is available that is capable of providing temperature reference for use in the cold junction compensation of thermocouple inputs.

Interrupt Status Word (ISW)

The analog input control model C01 has its own interrupt status word (ISW) for identification of interruption conditions. Bits in the ISW are set by conditions occurring after an immediate write command is completed but while the device is still busy (after starting a convert operation). The setting of the ISW bits (except the busy bit) is indicated to the operating program by an interruption request. Interruption requests are presented for errors detected in analog-to-digital conversion cycles.

The program can examine the ISW to determine which error condition exists by issuing an immediate-read-ISW command.

After the interruption is accepted, the execution of any command directed to the analog input control resets the ISW.

The meanings of the significant bits in the ISW are:

Significant Bits	*Meaning*
3	ADCI—analog-to-digital converter inoperative.
5	IAD—invalid analog data. The input voltage polarity changed states during the conversion.
7	OLD—overload. The voltage read by the ADC was greater than the input voltage range selected by the amplifier.
12	Device busy.
13	Device end. This bit is set on when the I/O operation is completed. If error conditions occur during the execution of such operations as analog-to-digital conversions, the device-end bit is also set on along with other error-identifying bits.
15	INSA—invalid subaddress. An input point was addressed that was not installed.

I/O Commands

The prepare I/O and halt I/O commands are used as described under "I/O Commands" (direct control channel).

Data transfers between the analog input control and the processor module are accomplished by immediate read and write commands. The setting of the modifier field bits in the command further defines the operation to be performed.

In all I/O commands addressed to analog input, the subaddress (SA) field contents must be 1001. The module address (MA) varies depending upon the physical location of the multifunction module.

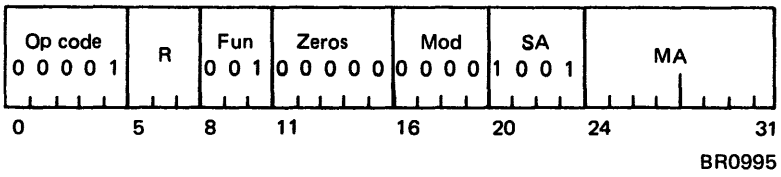
Usually, a complete immediate operation is performed within the time limit required for execution of the immediate commands, leaving no residue or aftereffects within the module. Exceptions to this rule are three convert commands which start the conversion cycle and permit the program to resume normal operation while awaiting the interruption request presented by the module at the end of the conversion cycle. This type of immediate command is indicated in discussions by a suffix of (int) which signifies "interruption."

Executing any of these three commands sets on the ISW device-busy bit, which remains on during the conversion time. At the completion of conversion, an interruption occurs which turns off the device-busy bit and turns on the ISW device-end bit. The analog input control presents a busy signal to all immediate commands issued (except read ISW) as long as the device is busy or has a pending interruption.

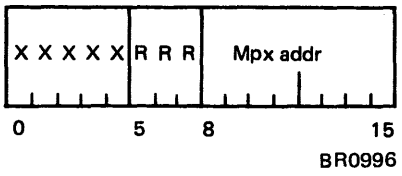
Immediate Write Commands

Immediate write commands are used with the analog input control to start analog-to-digital conversion cycles of specifically addressed analog input points.

Convert Normal Input (Int)



This command transmits data in the index register (R), or accumulator if R = 000, to the analog input control. This data has the format:



The three RRR bits specify the input voltage range and amplifier gain desired:

RRR Bits	*Input Voltage Range*	*Amplifier Gain*
000	——	Automatic gain
001	±10 mV	512
010	±20 mV	256
011	±40 mV	128
100	±80 mV	64
101	±160 mV	32
110	±640 mV	8
111	±5.12V	1

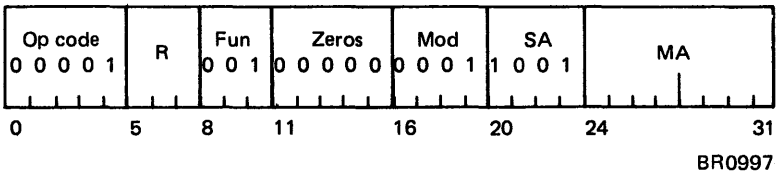
When automatic gain is specified (RRR bits = 000), the optimum gain (one of the seven amplifier gains) is selected automatically based upon the input signal voltage. The selected gain can be returned to the program, along with the transferred data, by issuing a read ADC command. Since the indication of gain requires three bits, the input data itself is truncated to 12 bits plus sign.

The range control bits (RRR) have no significance if the high-level-only gain amplifier is installed.

Convert normal input (int) starts an analog-to-digital conversion cycle of the analog input point identified by the multiplexer address (mpx addr). Interruption occurs at the end of the conversion cycle.

Results are obtained by issuing an immediate-read-ADC-extended-precision, an immediate-read-ADC, or a read-and-convert-ADC (int) command. With either of the latter two commands, the format of the data returned is in short form (12 bits plus sign) if automatic gain was specified or standard form (14 bits plus sign) if a programmed range was specified. The extended precision command always returns data in standard form.

Convert Normal Input With External Synchronization (Int)



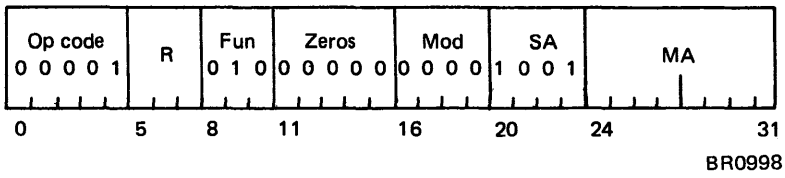
This command functions in the same manner as the convert normal input (int) command except that the start of conversion depends upon receipt of an external synchronization input pulse.

The user provides the external synchronization input pulses to control the speed of analog-to-digital data conversion. (Refer to *System/7 Installation Manual–Physical Planning,* Order No. GA34-0004.)

Immediate Read Commands

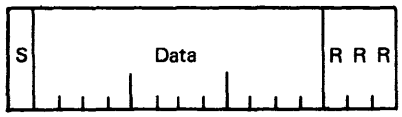
Immediate read commands are used with the analog input function to obtain the binary output of the analog-to-digital converter and status information pertaining to the I/O device.

Read ADC



The read ADC command stores the binary output value from the ADC into the index register (R), or accumulator if R = 000.

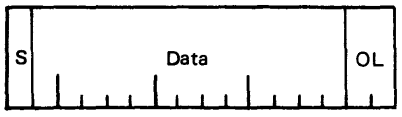
The data transferred to the register can have two formats, depending upon whether the last convert command requested automatic gain or selected a particular amplifier gain. If automatic gain was requested, the 16 bits of data have the format:



BR0999

Twelve bits of binary data plus sign (S bit) are returned. The RRR field indicates which amplifier gain was automatically selected by the amplifier. (Refer to the table in the "Convert Normal Input (Int)" description.) If RRR = 000, overload occurred with the automatic gain selection.

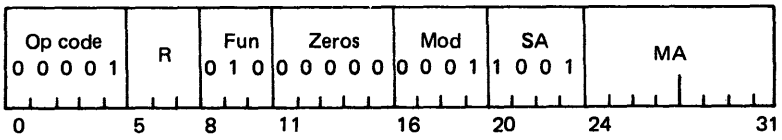
If a particular amplifier gain was selected by the last convert command, the 16 bits of data have the format:



BR1000

Fourteen bits of binary data plus sign (S bit) are returned. The single-bit OL field is set on if overload occurred with the programmed gain selection.

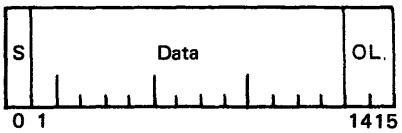
Read ADC Extended Precision



BR1001

The read-ADC-extended-precision command stores 14 bits of binary data plus sign when automatic amplifier gain was selected by the last convert command.

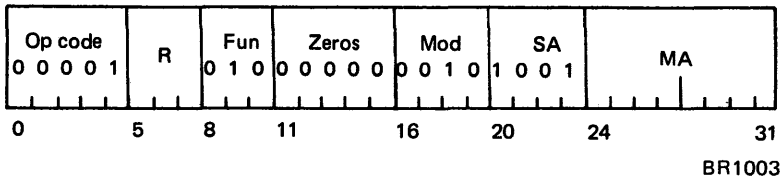
The ADC output value resulting from the last ADC cycle is placed in the index register (R), or accumulator if R = 000. The data transferred to the register has the format:



BR1002

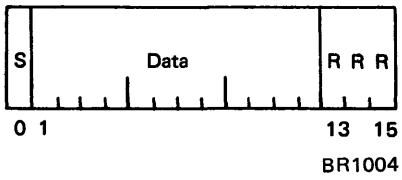
Fourteen bits of binary data plus sign (S bit) are returned. The single-bit OL field is set on if overload occurred with the automatic amplifier gain selection. Note that the automatically selected gain is not returned with the data. This gain can be determined by issuing a read ADC command before or after the read-ADC-extended-precision command.

Read and Convert ADC (Int)



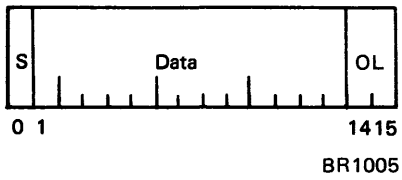
This command is used for repetitive operations on a single analog input point. The read-and-convert-ADC (int) command stores the binary output value from the ADC into the index register (R), or accumulator if R = 000.

Data transferred to the register can have two formats, depending upon whether the last convert command requested automatic gain or selected a particular amplifier gain. If automatic gain was requested, the 16 bits of data have the format:



Twelve bits of binary data plus sign (S bit) are returned. The RRR field indicates which amplifier gain was automatically selected by the amplifier. (Refer to the table in the "Convert Normal Input (Int)" description.) If R = 000, overload occurred with the automatic gain selection.

If a particular amplifier gain was selected by the last convert command, the 16 bits of data have the format:

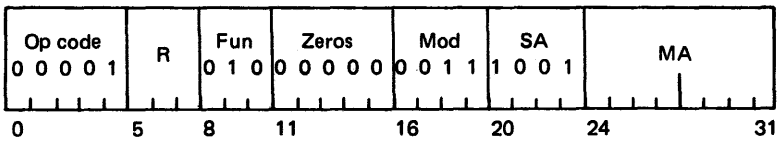


Fourteen bits of binary data plus sign (S bit) are returned. The single-bit OL field is set on if overload occurred with the programmed amplifier gain selection.

The convert portion of the read-and-convert-ADC (int) command converts the analog input point specified by the last convert command, also utilizing the same amplifier gain. Thus, if automatic gain was specified by the last convert command, automatic gain is also used by each subsequent execution of the read-and-convert-ADC (int) command.

Since this command performs repetitive operations on a single analog input point, the solid-state multiplexer is monopolized during this time. To terminate this repetitive read sequence and free the multiplexer, a read ADC command must be issued.

Read ISW



The read ISW command stores the 16-bit interrupt status word (ISW) associated with the analog input into the index register (R), or accumulator if R = 000.

If the interruption that results from setting an ISW bit is accepted, subsequent execution of any command directed to the analog input control resets the ISW.

DIGITAL INPUT CONTROL

The digital input control feature of the multifunction module can accommodate as many as 128 digital input points, in eight groups of 16 points each. These points are completely isolated two-terminal inputs that can be activated by either contact or voltage sources provided by the user equipment.

A digital input contact point has a 1 value in its corresponding bit when the contact is closed, and a 0 value when the contact is open. A digital input voltage point has a 1 value in its corresponding bit when the input is from +2V to +48V, and a 0 value when the input is from +0.8V to - 48V. Inputs between +0.8V and +2V have an indeterminate result.

Each group of digital input points can be program-controlled to provide either latched or unlatched input. In the latched mode of operation, each bit in the group that is set on remains in this state (regardless of any further input signal or switch fluctuations) until the group is reset or unlatched. In the unlatched mode of operation, each bit in the group is set on or off dynamically as determined by the input fluctuations.

The digital output features are compatible with the digital input circuits. This gives the ability to do "wraparound" testing and multiplexing of user contacts with digital output for digital input sensing applications. Since all of the points are isolated, either positive or negative voltage sources can be used through proper connection of input sources to the input terminals.

Process Interrupt Feature

A process interrupt feature can be obtained for one or both of the first two digital input groups (group 0 and/or group 1).

With this feature, a 16-point group is compared with a 16-bit reference register (which can be set to any value by the program), and interruption requests are initiated on the basis of either an equal or unequal comparison. The choice of comparisons is under program control and can be changed at any time.

The process interrupt feature can be used with groups 0 and/or 1 when they are operating in either the latched or unlatched mode of operation. Each of the two groups is a separate interrupting source and, therefore, each group can be assigned its own priority level and sublevel by a prepare I/O command.

Even though the process interrupt feature is installed, it can be disabled, and enabled, at any time. Program control of the process interrupt feature is provided by the set-digital-input-group-control command.

Interrupt Status Word (ISW)

No ISW is associated with the digital input function unless the process interrupt feature is installed. For each process interrupt feature (maximum of two), there is a separate ISW (one for each group with interruption capability).

The setting of ISW bits (except the busy bit) is indicated to the operating program by an interruption request. Interruption requests are presented because of errors detected in digital input operations as well as for the occurrence of a process interruption.

The program can examine the ISW to determine which interruption condition exists by issuing an immediate-read-ISW command, with a subaddress field of 0000 (for group 0 ISW) or 0001 (for group 1 ISW). This command resets the ISW bits.

Interruption control is provided by the set-digital-input-group-control command.

The 16-bit ISW for each digital input group has only three significant bits. The remaining 13 bits are not used.

Bit 3–Open fuse
Bit 12–Busy
Bit 13–End

The open-fuse bit is set on when a digital input fuse is open when a digital input group is busy. This condition results in an interruption request.

The busy bit is on whenever the group is in an interruption-enabled state. An interruption sets this bit off and turns the end bit on.

The end bit is set on by the process interruption when the compare on equal or unequal is successful, and the group is in an interruption-enabled state. This causes an interruption request and sets the group in an interruption-disabled state.

I/O Commands

The prepare I/O command is used as described under "I/O Commands" (direct control channel). However, it is applicable only for groups that have the process interrupt feature. The prepare I/O subaddress field must be 0000 (for group 0) or 0001 (for group 1).

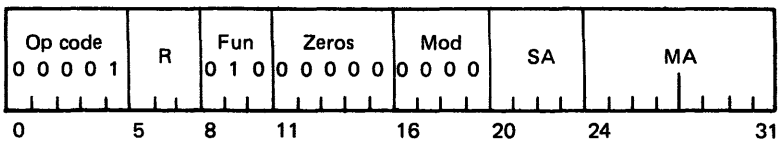
Data transfers between the digital input control and the processor module are accomplished by immediate read and write commands. The setting of the modifier field bits in the command further defines the operation to be performed.

All I/O commands addressed to the digital input control must have the following subaddress field contents in order to address each digital input group separately:

Subaddress (SA)	*Digital Input Group*
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

The module address (MA) in the I/O command varies depending upon the physical location of the multifunction module.

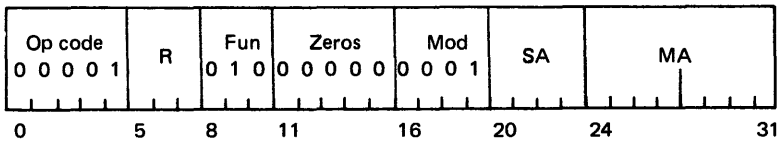
Read Digital Input Register



BR1007

The read-digital-input-register command stores the 16 bits of input data from a group (identified by the subaddress field) into the index register (R), or accumulator if R = 000.

Read/Reset Digital Input Register

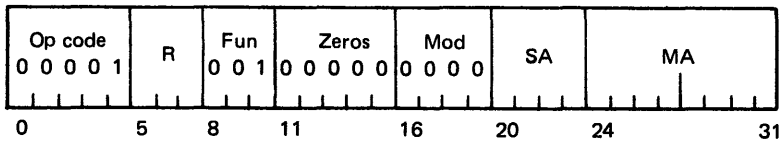


BR1008

This command stores the 16 bits of input data from a group (identified by the subaddress field) into the index register (R), or accumulator if R = 000.

If the group is operating in latched mode, its digital input register is then reset to zero values (unless the input is still active).

Set Digital Input Reference Register



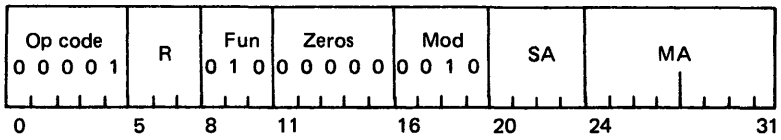
BR1009

This command stores a 16-bit value in the reference register associated with group 0 or group 1. The choice is determined by the subaddress field in the command: 0000 for group 0 or 0001 for group 1.

The value set into the reference register is obtained from the index register (R), or accumulator if R = 000.

The reference registers are part of the process interrupt feature for groups 0 and 1, and are used for comparison with the 16-point group input register. Interruption requests can be initiated for either an equal or unequal comparison; the choice is under control of the program. (Refer to "Set Digital Input Group Control.")

Read Digital Input Reference Register



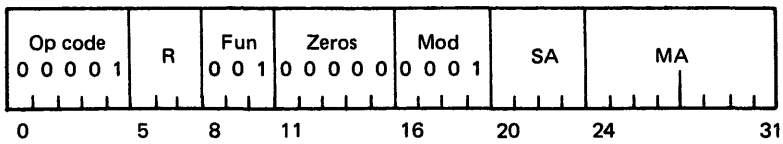
BR1010

This command stores the contents (16 bits) of the reference register associated with group 0 or group 1. The choice is determined by the subaddress field in the command: 0000 for group 0 or 0001 for group 1.

The reference register contents are stored into the index register (R), or accumulator if R = 000.

The reference registers are part of the process interrupt feature for groups 0 and 1, and are used for comparison with the 16-point group input register. Interruption requests can be initiated for either an equal or unequal comparison; the choice is under control of the program. (Refer to "Set Digital Input Group Control.")

Set Digital Input Group Control



BR1011

This command sends control information to the digital input group identified by the subaddress and module address fields. The control information is indicated by the value of bits 13-15 of the index register (R), or accumulator if R = 000.

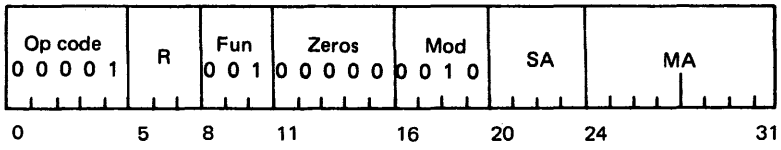
Control information in bits 13-14 should be directed only to digital input group 0 or 1, and is meaningful only when the group has the process interrupt feature.

Interruption control is provided by the setting of bit 13. If bit 13 is on, the process interrupt feature for the selected group functions normally, i.e., interruption requests can be presented. If bit 13 is off, interruption requests are inhibited.

Control of comparisons (between the digital input group register selected and its reference register) is provided by the setting of bit 14. If bit 14 is on, an interruption request is presented to the system if the comparison is equal. If bit 14 is off, the interruption request occurs if the comparison is unequal.

Control information in bit 15 can be directed to any of the eight possible digital input groups (groups 0-7) in a multifunction module. The setting of bit 15 determines whether the selected group is to operate in the latched or unlatched mode. A 1 value means the latched mode, and a 0 value indicates the unlatched mode.

Set Test Signal

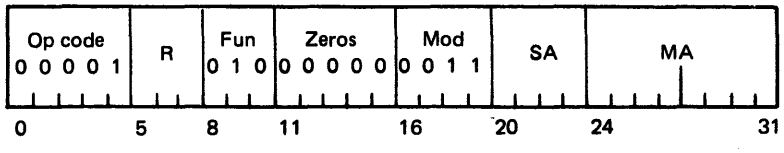


BR1012

The set-test-signal command tests a digital input register by establishing a predetermined value in the register. Selection of the desired digital input group is made by the subaddress (SA) and module address (MA) field contents in the command.

The setting of bit 15 in the index register (R), or the accumulator if R = 000, indicates the value to be set into the digital input register. If bit 15 is a 0, the 16-bit digital input register is set to all 0's. If bit 15 is a 1, the digital input register is set to all 1's. The register can be reset to all 0's by issuing a read/reset-digital-input-register command.

Read ISW



BR1013

The read ISW command stores a 16-bit interrupt status word (ISW) into the index register (R), or accumulator if R = 000.

A separate ISW is used for each process interrupt feature. The subaddress field contents determine which ISW is to be stored (0000 for the group 0 ISW, or 0001 for the group 1 ISW).

The read ISW command also resets the addressed ISW.

DIGITAL OUTPUT CONTROL

The digital output control feature of the multifunction module can accommodate as many as 64 digital output points, in four groups of 16 points each. Each group can be configured to use one of the three optional output circuit types available (DO low-power group, DO medium-power group, and DO contact group).

The digital output points can be operated by either of two methods: (1) Data can be sent directly to a selected digital output group register which, in turn, operates the output points. (2) Data can be sent to a holding register associated with the selected group; the data can then be read back and checked for accuracy before transferring it to the digital output group register.

The digital output features are compatible with the digital input circuits. This gives the ability to do "wraparound" testing and multiplexing of user contacts with digital output for digital input sensing applications. Either positive or negative voltage sources can be used through proper selection of the input terminals when the DO medium-power group or DO contact group is used.

I/O Commands

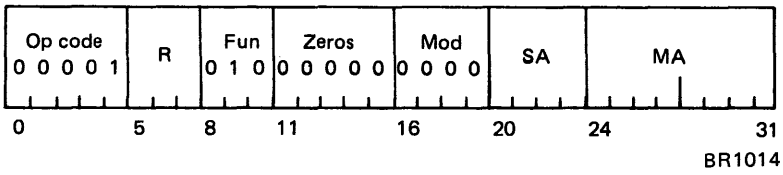
Data transfers between the digital output function and the processor module are accomplished by immediate read and write commands. The setting of the modifier field bits in the command further defines the operation to be performed.

All I/O commands addressed to the digital output function must have the following subaddress field contents in order to address each digital output group separately:

Subaddress (SA)	*Digital Output Group*
1100	0
1101	1
1110	2
1111	3

The module address (MA) in the I/O command varies depending upon the physical location of the multifunction module.

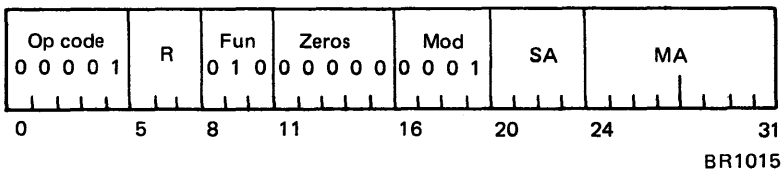
Read Digital Output Register



BR1014

The read-digital-output-register command stores the 16 bits of output data from a group register (identified by the subaddress field) into the index register (R), or accumulator if R = 000.

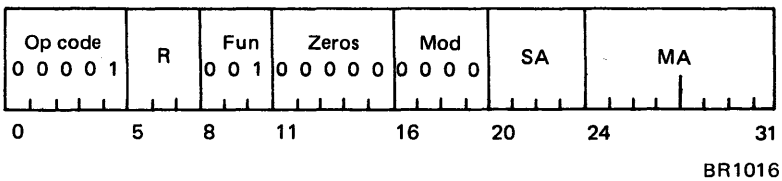
Read Digital Output Holding Register



BR1015

This command stores the 16 bits of output data from a holding register (associated with the group identified by the subaddress field) into the index register (R), or accumulator if R = 000.

Write Digital Output Register

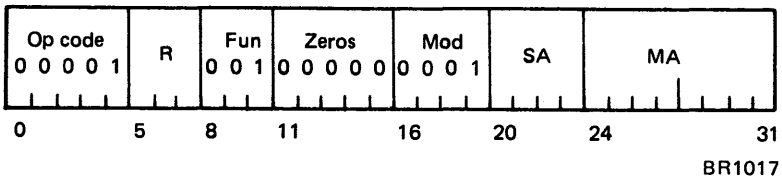


BR1016

The write-digital-output-register command stores 16 bits of output data from the index register (R), or accumulator if R = 000, into the digital output group register identified by the subaddress field.

The status of the 16-bit digital output register is reproduced at the corresponding terminals for its associated 16-point group. A 1-bit in the register produces a closed or "on" condition at its corresponding output point; a 0-bit in the register produces an open or "off" condition at its corresponding output point.

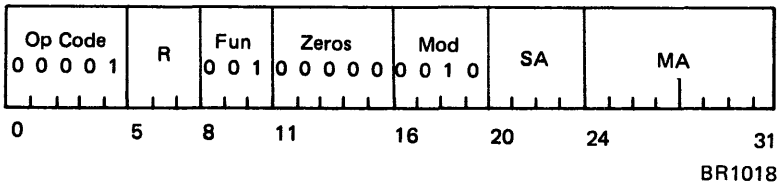
Write Digital Output Holding Register



This command stores 16 bits of output data from the index register (R), or accumulator if R = 000, into the holding register associated with the digital group identified by the subaddress field.

The holding register provides a "buffer" which allows the data to be verified by the program prior to transferring it to the output register (and the corresponding terminals).

Set Digital Output Register



This command transfers the 16 bits of output data in the holding register (identified by the subaddress field) to its associated digital output group register. The R field is ignored.

ANALOG OUTPUT CONTROL

The analog output control feature of the multifunction module has either one or two isolated analog output voltage points. Each point provides a voltage output to user terminals.

The analog output points can be operated by either of two methods: (1) Data can be sent directly to a selected analog output register which, in turn, operates an output point. (2) Data can be sent to a holding register associated with the selected point; the data can then be read back and checked for accuracy before transferring it to the analog output register.

The analog output function can be used as an input to the analog input function. This gives the ability to do "wraparound" testing and multiplexing of analog output points with user contacts for analog input sensing applications. Caution must be used when wrapping around directly because the analog output function has a voltage output capability that is higher than the maximum input voltage to the analog input function. For this reason, a 2:1 divider is recommended to prevent overload and possible damage to the analog input subsystem.

I/O Commands

The analog output function is programmed by I/O commands which read or write 16 bits of program data. However, analog output data has a resolution of only 10 bits. Therefore, in all registers (which have 16 bits), the analog output data resides in bits 1-10 and the sign resides in bit 0.

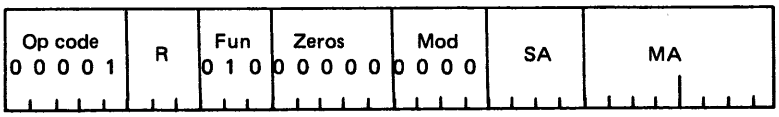
Data transfers between the analog output function and the processor module are accomplished by immediate read and write commands. The setting of the modifier field bits in the command further defines the operation to be performed.

All I/O commands addressed to the analog output function must have the following subaddress field contents in order to address each analog output point separately:

Subaddress (SA)	*Analog Output Point*
1010	0
1011	1

The module address (MA) in the I/O command varies depending upon the physical location of the multifunction module.

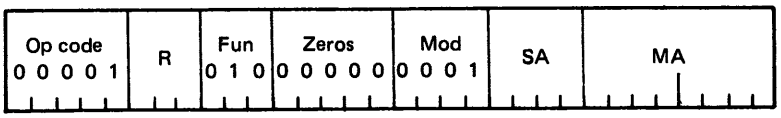
Read Analog Output Register



BR1019

The read-analog-output-register command stores the output data from an analog output register (identified by the subaddress field) into the index register (R), or accumulator if R = 000. Bits 11-15 of the R register are set to 0's.

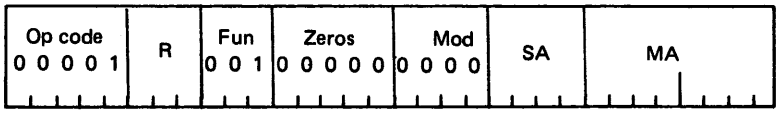
Read Analog Output Holding Register



BR1020

This command stores the output data from a holding register (associated with the point identified by the subaddress field) into the index register (R), or accumulator if R = 000. Bits 11-15 of the R register are set to 0's.

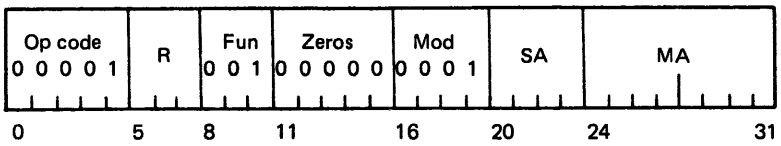
Write Analog Output Register



BR1021

The write-analog-output-register command stores the output data from the index register (R), or accumulator if R = 000, into the analog output point register identified by the subaddress field.

Write Analog Output Holding Register

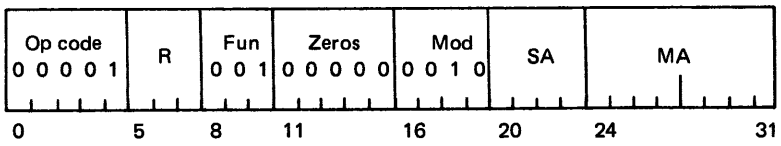


BR1022

This command stores the output data from the index register (R), or accumulator if R = 000, into the holding register associated with the analog point identified by the subaddress field.

The holding register provides a "buffer" which allows the data to be verified by the program prior to transferring it to the analog output register (and the corresponding analog output point).

Set Analog Output Register



BR1023

This command transfers the output data in a holding register (identified by the subaddress field) to its associated analog output point register. The R field is ignored.

2790 CONTROL

The 2790 control feature of the multifunction module has the necessary control logic for attaching a single loop of the IBM 2790 Data Communication System to the System/7 internal interface. This feature provides the user with a flexible plant communication and information collection system.

Either the System/7 itself, or an attached 1130 host processor, can act as the system controller for the 2790. Data is entered through the 2790 devices by card reading, badge reading, or key entry units. Output is through visual display or a printer.

The 2790 Data Communication System is connected by a two-wire, high-speed loop capable of operating at about 500,000 bits per second. This loop is a serial connection that starts and ends at the 2790 control feature in the System/7.

Multiple area stations (2791 and/or 2793) are connected serially at various points on the loop. Multiple 2795/2796 Data Entry Units, 1035 Badge Readers, and 1053 Printers can be connected to each area station. Area stations are separated from each other, and the first and last area stations on the loop are separated from the 2790 control feature, by a maximum of 1,000 feet.

The actual number and configuration of the area stations, and their associated input and output units, that can be attached to the loop are limited by programming capabilities and the storage required to support the configuration. Programming support is provided for one 2790 control feature in a System/7 configuration.

For more detailed information on the capabilities, components, features, and applications of the 2790 Data Communication System, refer to *IBM 2790 Data Communication System Component Description,* Order No. GA27-3015, and *IBM 2790 System Summary,* Order No. GA27-3016.

Data Transmission

All communication between the System/7 and the area stations is accomplished by a sequence of five information bytes. The first byte (start byte) is generated by the 2790 control; the remaining four bytes are generated by the system controller. The five information bytes are followed by 25 synchronization bytes, with the total of 30 bytes making up what is called an active frame. All synchronization bytes are generated by the 2790 control.

The active frames are transmitted by the 2790 control at intervals of 524 μs. Active frames are separated by a series of inactive frames which consist of a start byte and 29 synchronization bytes. This separation is used primarily to allow time for the processing of input frames and the preparation of output frames. The synchronization bytes, together with the start byte, give continuous byte synchronization information.

The functions of the five active frame information bytes are:

Start Byte: The start byte (hexadecimal 39) contains coded information that signals the start of an active frame. The end of an active frame is indicated by the selected area station reaching a byte count of five. The start byte, generated by the 2790 control, is the only byte that requires even parity.

Area Station Address Byte: Coded information in this byte can be used to select a unique area station, any area station, or all area stations.

There are 128 unique area station addresses, which must be indicated by codes within the range of hexadecimal 80 to hexadecimal FF. All area station address codes outside this range are invalid. Due to the characteristics of the system, it is recommended that area station addresses be assigned sequentially starting with hexadecimal 80. This does not restrict the physical placement of area stations on the loop.

An any-area-station address is designated by a hexadecimal 11 code. Whenever an area station requires service, it searches for frames having this code. Upon finding one, the area station selects the frame by inserting its own address in this byte.

A code of hexadecimal 09 sends data to all of the area stations. Every area station executes the instruction specified in the control byte, if it can, without modifying the area station address byte.

Device Address Byte: A frame addressed to an area station is also addressed to a particular device (e.g., 2795 Data Entry Unit) on a specific adapter connected to that area station. Bits 0-1 of the device address byte identify the particular adapter on the area station; bits 2-7 identify the device attached to that adapter. If an area station adapter is selected, or an area station captures an "any address" frame, the area station inserts into the frame its own address and the address of a particular device.

Control Byte: The control byte is divided into two hexadecimal digits. The low-order digit (bits 4-7) indicates the command issued to an area station by the 2790 control. The command digit is further subdivided into an operation code (bits 4-5) and a modifier (bits 6-7).

The high-order digit (bits 0-3) contains the response from an area station, which always responds to a valid command. All responses from area stations are sent to the 2790 control.

Command frames are easily distinguished from response frames because the response digit is zero for all commands, but non-zero for all responses. The 2790 control can issue several commands in each of the three major command categories (read, write, and control), and there is at least one valid response for each of the commands.

Bit 0 of the response digit is always set to 0. Bit 1 is the data mode bit, and is set on when the area station decodes one of the diagnostic commands, or when the addressed adapter decodes a valid operation code and is in the data mode. Bits 2-3 are status bits, and are encoded to reflect the stage of operation in the area station adapter.

The control byte codes are established for two-way communication between the area stations and the system controller at every step during the various operations.

Data Byte: The data byte contains either data or status information, depending upon the contents of the control byte.

A data transfer caused by a read command always involves transmitting the data byte twice, once in each of two frames. Thus, input data can be checked for accuracy without special checking logic in the area station. Data transferred as a result of a write command is also checked for accuracy by transmitting it to an area station and then receiving the data back from the area station. If the transmitted data is the same as the received data, the area station is then directed to print the data or turn on the specified operator guidance light.

Interrupt Status Word (ISW)

The 16-bit interrupt status word associated with the 2790 control function has bits indicating device status operating information and detection of parity errors.

The program can examine the ISW to determine which interruption condition exists by issuing an immediate-read-ISW command with a subaddress field of 1000. The ISW bits are reset on the first selection of the 2790 control function after the interruption request is successfully presented to the system controller.

The 2790 control ISW has the following significant bit assignments:

Significant Bits	*Meaning*
3	Parity error detected on the area station address byte.
4	Parity error detected on the device address byte.
5	Parity error detected on the control byte.
6	Parity error detected on the data byte.
7	No parity errors detected on the active frame.
12	Device is busy.
13	Device has completed operation.

I/O Commands

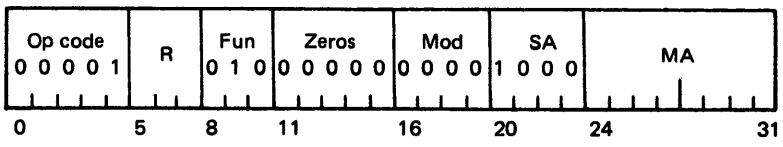
The 2790 control function is programmed by I/O commands. The 2790 control has two 4-byte buffer registers, one for input frames and one for output frames. Immediate read commands are used to read the frame in the input buffer register, while immediate write commands are used to fill the output buffer register with an outgoing frame. The start byte (first byte of a frame) is generated by the 2790 control itself, and is not read into the input buffer or written from the output buffer.

The prepare I/O command is used as described under "I/O Commands" (direct control channel).

With immediate read and write commands, the setting of the modifier field bits in the command further defines the operation to be performed.

All I/O commands addressed to the 2790 control function must have 1000 in the subaddress (SA) field. The module address (MA) in the I/O command varies depending upon the physical location of the multifunction module.

Read Address

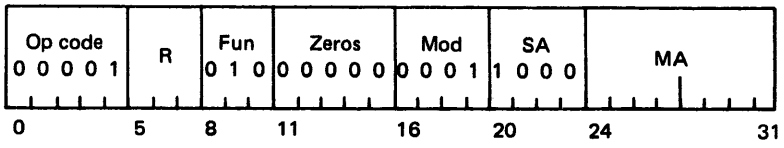


BR1024

Following a 2790 adapter interruption, four bytes of a frame reside in the input buffer register. The read address command stores the first two bytes of the buffer (area station address and device address) into the index register (R), or accumulator if R = 000.

These addresses are obtained by the program so that the contents of the incoming frame can be compared with the contents of the outgoing frame for accuracy. The addresses are also placed, by the program, in the next outgoing frame that is generated.

Read Control/Data

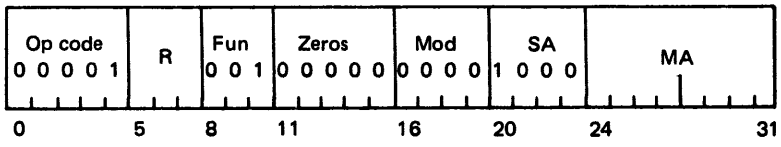


BR1025

Following a 2790 adapter interruption, four bytes of a frame reside in the input buffer register. The read control/data command stores the last two bytes of the buffer (control and data bytes) into the index register (R), or accumulator if R = 000.

This information is obtained by the program to compare the control/data bytes for accuracy every time a frame is transmitted by an area station. The information is also used to determine the control and/or data that is placed in the next outgoing frame.

Write Address



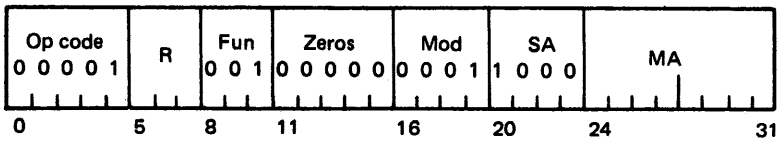
BR1026

The output buffer register must contain the data that is to be transmitted to the loop in the next active frame.

The write address command stores the area station address and device address into the first two bytes of the output buffer register in the 2790 control. These two address bytes are transferred from the index register (R), or accumulator if R = 000. The 2790 control generates the start byte (first byte of frame) itself.

A write address command must be issued once to establish the desired addresses in the output buffer register. Any number of write control/data commands can then be issued. Another write address command must be issued only when the addresses require change.

Write Control/Data

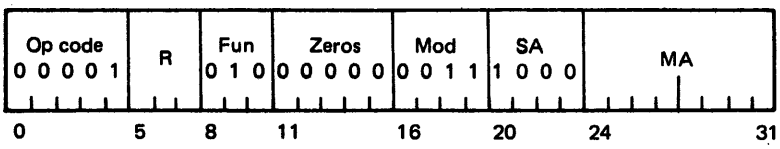


BR1027

The output buffer register must contain the data that is to be transmitted to the loop in the next active frame.

The write control/data command stores the control and data bytes into the last two bytes of the output buffer register in the 2790 control. These two bytes are transferred from the index register (R), or accumulator if R = 000.

Read ISW



BR1028

The read ISW command stores the 16-bit interrupt status word (ISW) into the index register (R), or accumulator if R = 000.

Main page references are those listed first. Throughout the index, alphabetic entries are listed first, then numeric entries.

READER'S COMMENT FORM

IBM System/7 Functional Characteristics — Order No. GA34-0003-0

- Your comments, accompanied by answers to the following questions, help us produce better publications for your use. If your answer to a question is "No" or requires qualification, please explain in the space provided below. Comments and suggestions become the property of IBM.

	Yes	No
• Does this publication meet your needs?	☐	☐
• Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

- What is your occupation? _____.
- How do you use this publication?

As an introduction to the subject?	☐	As an instructor in a class?	☐
For advanced knowledge of the subject?	☐	As a student in a class?	☐
For information about operating procedures?	☐	As a reference manual?	☐

Other _____.

- Please give specific page and line references with your comments when appropriate. If you wish a reply, be sure to include your name and address.

COMMENTS:

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

YOUR COMMENTS, PLEASE. . .

Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Note: 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.

Fold Fold

FIRST CLASS
PERMIT NO. 110
BOCA RATON, FLA
33432

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation
General Systems Division
Boca Raton, Florida 33432

Attention: Product and Programming Publications, Department 707

Fold Fold

Cut Along Line

System/7 Functional Characteristics Printed in U.S.A. GA34-0003-0

IBM

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

System/7 Functional Characteristics Printed in U.S.A. GA34-0003-0

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, New York 10601
(USA only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)