File No. S360-01
Form GA27-2719-2

Systems Reference Library

IBM System/360 Model 67

Functional Characteristics

This publication contains detailed information on the organization, characteristics, features, and functions unique to the IBM System/360 Model 67 Time Sharing System. Major areas described include time-sharing philosophy, system structure, new units, generalized information flow, standard and special features, instruction timings, and the system control panel.

Descriptions of specific input/output devices used with the Model 67 appear in separate publications. See the *IBM System/360 Bibliography*, Form GA22-6822 for a listing and a brief description of these publications.

The material in this publication is presented with the assumption that the reader has knowledge of System/360 as defined in the *IBM System/360 Principles of Operation*, Form GA22-6821 and the *IBM System/360 System Summary*, Form GA22-6810. The *IBM System/360 Model 67 Configurator*, Form GA27-2713 also may be of interest to the reader.

Third Edition (February, 1972)

This is a reprint of GA27-2719-1 incorporating changes released in the following Technical Newsletter:

GN27-2940 (dated September 11, 1970)
GN27-2944 (dated January 8, 1971)

Changes are periodically made to the specifications herein; before using this publication in connection with the operation of IBM systems, refer to the latest SRL Newsletter, Form N20-0360, for the editions that are applicable and current.

Text for this manual has been prepared with the IBM SELECTRIC ®Composer.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to: IBM Systems Development Division, Product Publications, Dept. 520, Neighborhood Road, Kingston, N.Y., 12401.

Contents

Illustrations and Tables

System/360 Model 67-2 Duplex Configuration

IBM System/360 is a versatile, all-purpose family of data processing equipment that can accommodate the wide range of applications which may be encountered in a diversified computing activity. The system is unique in its capacity to grow easily with the needs for increased computational capabilities. If particular input/output (I/O) equipment is outgrown, more and/or faster I/O equipment can be added. Also, processor storage capacity can be increased easily. The important point, however, is that any or all of this growth can be accomplished without changes, either in system programming or in problem programming.

The basic architecture of System/360 makes it ideally suited to operate in a multiprogramming and multiprocessing environment. Model 67 extends this basic architecture to provide the additional capabilities of a time-sharing system.

Four new units are introduced by the Model 67:

1. The IBM 2067 Processing Unit Models 1 and 2
2. The IBM 2365 Processor Storage Model 12
3. The IBM 2846 Channel Controller
4. The IBM 2167 Configuration Unit.

These new units, together with the IBM 2365 Processor Storage Model 2, 2860 Selector Channel, 2870 Multiplexer Channel, and the wide range of IBM System/360 control units and devices are the building blocks for Model 67 configurations.

There are two models of the Model 67, termed 67-1 and 67-2. The Model 67-1, a single-processor (simplex) system, uses the 2067-1 (Figure 1); the term Model 67-2 describes systems using the 2067-2 both in half-duplex (one-processor) and in duplex (two-processor) configurations (Figure 2). The Model 67-1 cannot be converted to a Model 67-2 in the field.

The outline configurations of the Model 67 produced by the various combinations of a 2365 Processor Storage and 2067 Central Processing Units (CPU's) shown in Figure 3 are only a few of the combinations possible.

The system control panel is located at one end of the 2067. An IBM 1052 Printer-Keyboard can be placed adjacent to the 2067 Processing Unit reading board.

Standard features for any System/360 Model 67 include:
Universal Instruction Set
High-Resolution Interval Timer
7-Bit Storage Protection
24-Bit Dynamic Address Translation

Optional features for System/360 Model 67 include:
Extended Dynamic Address Translation
709/7040/7044/7090/7094/7094II Compatibility
1052 Adapter

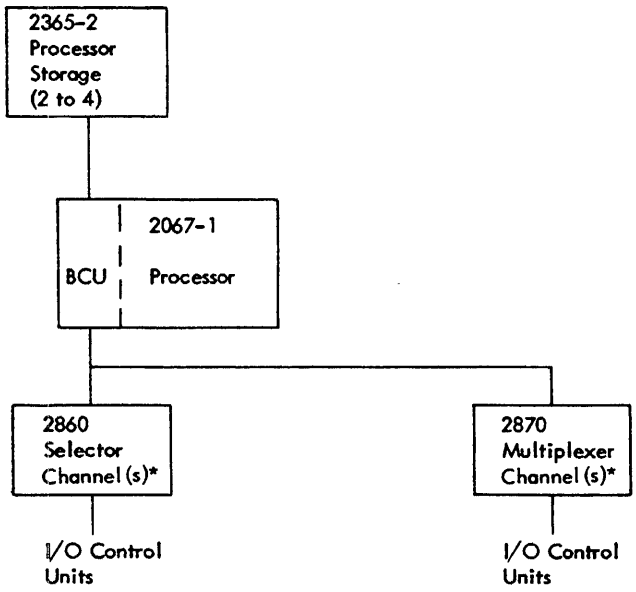


Figure 1. Model 67-1 Configuration

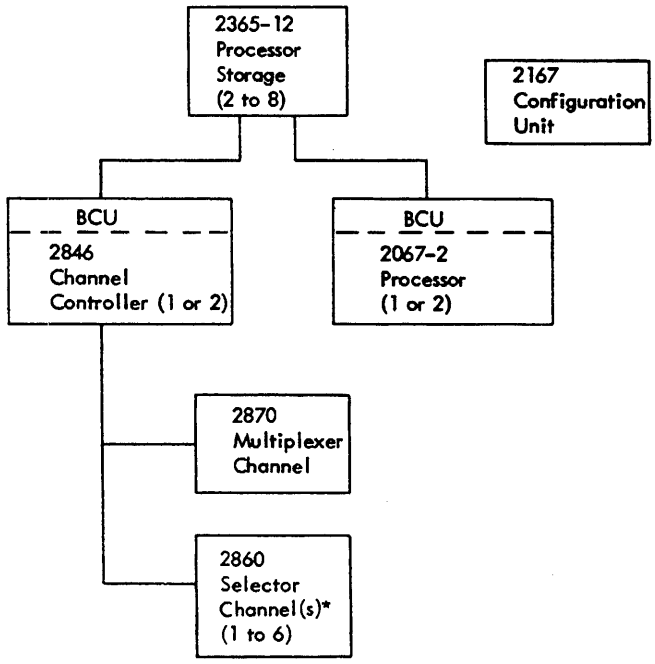


Figure 2. Model 67-2 Configuration

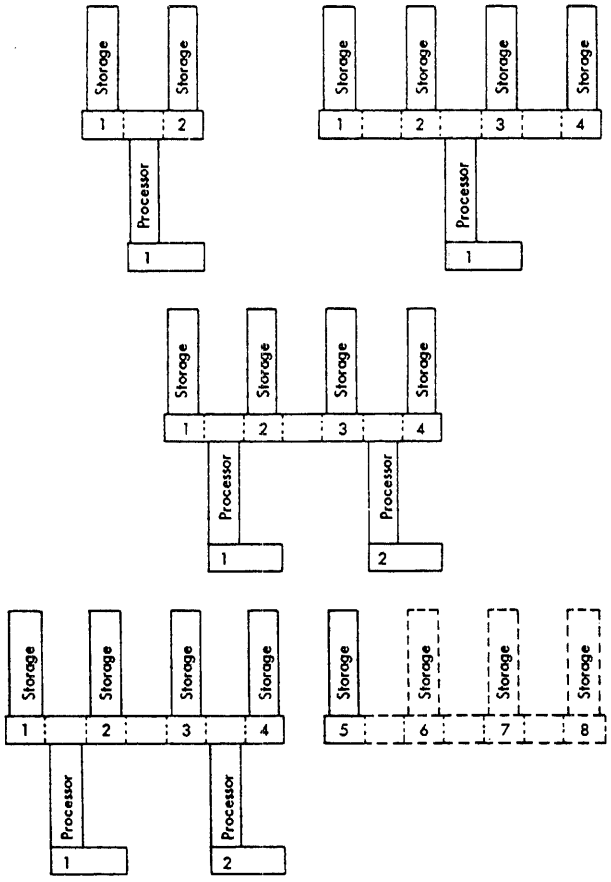


Figure 3. Model 67 Outline Configurations

Floating Storage Addressing (2067-1 only)
Direct Control (2067-1 only)
Extended Direct Control (2067-2 only)
Partitioning Sensing (2067-2 only)
Additional 2846 Attachment (2067-2 only)
Additional Addressing (2067-2 only)

A variety of control units and I/O devices are available for use with the Model 67. Descriptions of specific I/O devices appear in separate publications. Configurators for the I/O devices and systems components are also available. (See *IBM System/360 Bibliography,* Form A22-6822.)

Model 67 is a general-purpose computing system; it provides the user of an I/O device, called a "terminal", with the use of the system at times and places of his choosing.

A general-purpose computing system is one which can accommodate a variety of problems, expressed in several languages (including the language of the computer itself) and introduced from different sources.

Model 67 provides:

- On-Line Capability – The Model 67 couples users of computation facilities more closely to computer systems. This close user-system contact does not require physical proximity of the terminals; communication lines, dial connections, or similar apparatus of communications technology can be used in the system to convey computational power over any distance. The on-line capability makes possible the functions of console-type debugging techniques, conversational mode of operation, and the immediate utilization of many short-duration "shots" on the facility without significant delay. To enhance this on-line capability, the system employs large-capacity, direct-access storage devices to provide access both to the data bases and to programs of interest to the users.

- Multiple-Access Capability – Many users share the total system facility concurrently. One reason for providing multiple access in on-line systems is that, typically, one user does not continually require the total system facility. Without the multiple-access capability, a brief pause by a user would idle the entire system. With the multiple-access capability, pauses serve other uses. Since the cost of doing this is small compared with the facility time recovered, it is profitable, therefore, to have the capability of serving many users simultaneously.

- Multiprocessor Capability – For multiprocessor configurations, parallel operation is the normal mode of the system. During parallel processor operations, two CPU's can execute the same monitor simultaneously, independently executing programs, accessing storage, and controlling input/output. Since the monitor program is not changed by execution, simultaneous usage presents no problems.

- Time-Slicing Capability – Many different tasks are interleaved by using a timing mechanism to signal the end of the time interval given to one task and to permit the transfer of a CPU to another task. Each user receives a slice of computer time periodically, and, within a short cycle of time, all users are responded to in some sense.

Time-sharing is characterized by the time-slicing capability. Time-slicing, in conjunction with the supervisor program scheduling and dispatching routines, enhances the Model 67's ability to operate as a conventional batch processing system and as a conversational mode processor.

PROGRAM CHARACTERISTICS

Several highlights of Time Sharing System/360 (TSS/360) are described in the following paragraphs; TSS/360 is described in greater detail in the *IBM System/360 Time Sharing System Concepts and Facilities,* Form C28-2003.

Re-entrant Code

Much of the coding in the time-sharing system is re-entrant. Since a re-entrant program is not modified in any way by

execution, it can be shared by several other programs that can be executed by one or more processors. All parameters, data locations, and working storage spaces are passed to the re-entrant program. The integrity of each user's data is maintained regardless of the number of users who have entered the program.

Re-entrant coding of programs enhances system efficiency because only one copy of a multiple-usage program or subroutine need be present in processor storage.

Data Set Management

To the user, the computing system can be a file space as well as a computing device. The user need not keep programs (problem statements) or data files at his location. These can be introduced into the system, left in, and recalled and manipulated at will.

To maintain large sets of files with complete safety and security, the system contains a catalog of all data files within its storage devices and provides many modes of protection for these files. A user can specify that his files be accessible only to himself or be accessible to specific individuals other than himself. Various modes of access can be specified, for read-only purposes and for both reading and writing.

User Commands

Each terminal user can request the system to take certain actions in his behalf. Each action that he can specify is a command which, in the case of a typewriter-like terminal, is a one-line message. Commands are always acted upon by the Command-Language Interpreter, which receives the incoming message, scans it to determine the nature of the action requested, and fetches from the system files the command program that will carry out the action.

The command system (the set of actions the Command-Language Interpreter can recognize) provides the capability for a program to be:

1. Entered from the terminal line by line, with immediate diagnostic feedback from the system after each line.
2. Cataloged, stored permanently in the system for later manipulation, and made part of the user's prestored library.
3. Compiled or assembled.
4. Executed.

The user appears to the system as an active terminal presenting a series of tasks to be performed. As long as these tasks are system commands, the system fetches its own programs and executes them for the user. System commands are similar to user programs in that they are time-shared and multiprogrammed during execution. When the user initiates execution of his own program, he retains complete control over it. Thus, the command system also allows a program to be stopped, modified or displayed (for debugging purposes), restarted, and terminated.

Besides the basic capabilities of entering, executing, and controlling programs from terminals, the command system provides for system accounting and file maintenance.

System Accounting

A user must LOGON at a terminal before any other action can be taken, and LOGOFF when his terminal operations are complete. These commands identify the user to the system and initiate time charges to the user for services rendered. If he requests access to files in succeeding commands, his identity is matched against file-protection and security data to ensure file integrity and to prohibit unauthorized access to files.

File Maintenance

A user can request that his files be moved from one storage unit to another or be removed from the system. He can also modify them, combine them, or change their protection status. Commands can be initiated from terminals, as previously described, or they can in most cases be embedded in programs as the equivalent of subroutine calls. Commands can also be included as "control card" functions to specify the type of processing to be applied to batch jobs entering the system.

The command system allows the user to direct the system in its manipulation of files and in the solution to problems. Moreover, it is the universal interface between the system and the outside world. From the machine-room console, for example, installation personnel control recovery functions of the system, using special commands reserved for their use. Batch jobs entering the system from peripheral equipment or tape drives are controlled by the command system.

Paging

Every program using the system is treated as a sequence of 4096-byte units called "pages". By dividing programs into pages, processor storage can be allocated in page (4096-byte) increments. Program pages, therefore, can be located randomly throughout core storage and swapped in and out of processor storage, as pages are needed, commensurate with available space. Random location of pages for a given program necessitates the construction of tables (page tables) that reflect the processor storage location of the pages. The swapping of pages between auxiliary storage and processor storage is defined as "page turning". If a page of instructions refers to a program location not currently in processor storage, the system stops operating the program temporarily, makes arrangements to fetch the page from auxiliary storage (the disk and drum storage space reserved specifically for paging), and performs other operations in the interim.

"Page turning" has the following advantages:

1. The entire program need not be in core storage to operate. Parts of many programs can be present, and several may be ready for processing. Thus, the system has many opportunities to do useful work while a page is being swapped.
2. Program "swap time" is reduced as an overhead factor since only active pages of a program require movement between core storage and auxiliary storage.
3. Although written and executed as a classical set of contiguous instructions and working space, a program can exist in the machine as scattered active pages.

The programmer's concept of the program being executed is of a "virtual storage" rather than of an actual processor storage situation.

The user's virtual storage is the contiguous address space that would be needed to store the user's program. This virtual storage is not limited by actual processor storage size but is limited only by the available auxiliary (disk and drum) space. Theoretically, the programmer has 16 million bytes (24-bit addressing) or up to 4 billion bytes (optional 32-bit addressing) of virtual storage at his disposal.

Segmentation

System/360 24-bit addressing permits a 4096-page virtual storage. Virtual storage has been divided into 16 segments containing 256 pages each. A segment table with only 16 entries accounts for all virtual storage. When a page of the user's program is moved from virtual storage to processor storage, the location of the page is recorded in a page table. The entries in the page table are sequential by virtual address. The location of the page table is recorded in the segment table.

Since each segment contains 256 pages, the page table cannot contain more than 256 entries. If the program occupies more than one segment of virtual storage, a page table is established for each segment as its pages are moved to processor storage. Many segments can be used to organize a user program in virtual storage. Large working areas can be assigned virtual storage space. However, only an actual reference to a segment will result in the generation of a page table for that segment. This avoids reserving large areas in page tables for pages not currently in use.

The Extended Dynamic Address Translation feature provides 32-bit addressing by adding eight bits to the standard 24-bit addressing capability. This increased capability permits a 1,048,576-page virtual storage, which is divided into 4096 segments of 256 pages each.

EQUIPMENT CHARACTERISTICS

Dynamic Address Translation

Dynamic address translation is the process of converting virtual addresses into actual processor storage addresses during instruction execution. A bit in the program status word (PSW) determines whether translation is active or inactive. The translation feature is implemented in hardware in the processor. All processor storage addresses that originate from the processor, except for hardware generated addresses (PSW, etc.), are subject to translation. Storage addresses originating from the channels are not subject to translation. The basic feature provides for 24-bit addresses, whereas the Extended Dynamic Address Translation feature provides translation for either 24-bit or 32-bit addresses.

Prefixing as described in the *IBM System/360 Principles of Operation* manual is applied to each address after translation; i.e., if the translated address is in the range 0-4095, the address may be further modified under control of prefixing. Prefixing can be deactivated by a manual switch.

PROGRAMMING NOTE: Primary and alternate prefix areas must not be assigned within the first 102,399 bytes of physical storage.

Extended PSW Mode

The Model 67 can operate in Model 65 mode: under control of Operating System/360 (OS/360, standard PSW mode) or under control of TSS/360 (Extended PSW mode). Up to 14 channels can be attached in duplex configurations of the Model 67. Since this extended I/O capability requires additional interrupt masking capability, additional control registers have been implemented in the processor; TSS/360 utilizes this additional capability by operating the Model 67 in the Extended PSW mode. In addition, the PSW is redefined when the Model 67 is operating in Extended PSW mode.

High-Resolution Interval Timer

An interval timer with a high degree of resolution has been implemented in the processor. This timer supports the time-slicing technique and is used with the TSS/360 dispatching and scheduling routines to signal the end of the time interval allotted to a given task. The operation of this timer is fully compatible with the timer operation described in the *IBM System/360 Principles of Operation* manual.

Storage Protection Extension

Additional capability has been implemented in the storage protection circuitry. The extended field of the storage

protection key allows TSS/360 to evaluate the utilization of storage. If a reference is made to any location within a given block of storage, that fact is recorded in the protection key for that block. If the reference is a store operation, that fact is also recorded in the protection key. This recording is always active and is independent of the CPU state of the instruction or command being executed or of the manner in which the address is generated. The recorded information is accessible to the standard Insert Storage Key instruction. The increased field size affects only the Set Storage Key and Insert Storage Key instructions; the size of the PSW and channel address word (CAW) protection key fields are unchanged.

Partitioning

Partitioning is the division of a duplex configuration into two isolated subsystems. Partitioned units are made unavailable to part of the system for programmed control. This facility is implemented in units that have more than one interface to a processor-controlled subsystem. Each interface is equipped with a manual switch. This switch, together with a configuration-control bit in the processor and the wait state of the processor, control the partitioned status of all units in the system. Each processor can sense the status of all partitioning switches, regardless of the partitioned status of the processor.

Floating Addressing

Processor Storage

In a duplex configuration, the total bank of high-speed processor storage comprises several individual shared storage units. Each storage unit contains an increment of the total storage capacity in consecutive, unique addresses. In a time-shared system, this bank of storage is exposed to partitioning into several parts for use by independent subsystems. Thus, each unit can be reassigned any one of the address intervals possible in the given system. For example, configurations of four storage units providing a total of 1024K bytes of processor storage could be addressed as follows:

Configuration	Address Interval			
	Unit A	Unit B	Unit C	Unit D
Unpartitioned	000-256K	256K-512K	512K-768K	768K-1024K
Partitioned	000-256K	256K-512K	000-256K	256K-512K
Unit A Out for Maintenance	–	000-256K	256K-512K	512K-768K

Many different arrangements are possible, especially with a system that has as many as eight processor storage units. A single unit can be shared by two subsystems, or more than one unit can be shared by two subsystems. The example shows that more than one unit can contain the same address interval. Partitioning controls prevent a single address from accessing two units that contain the same address interval. In a single-processor configuration, the floating addressing capability is invoked when address reassignment is needed because a processor storage unit has failed.

Channel

In a duplex system, channels are controlled and attached to the system via one or two Channel Controllers. (See Figure 2.) A maximum of seven channels (one 2870 and two 2860s in any combination of models) can be attached to each Channel Controller. The channels attached to Channel Controller 0 are assigned addresses in the range of 0 to 6, as are the channels attached to Channel Controller 1.

When a CPU is in Extended PSW mode, channel addressing is defined by the Channel Controller to which a given channel is attached; that is, selection of a channel requires both an appropriate Channel Controller address and the channel address. Partitioning switches on the Configuration Unit control panel permit the assignment of both, either, or no Channel Controllers to each CPU.

When a CPU is in standard PSW mode, Channel Controller 0 must be addressed. Floating channel addressing permits the assignment of either Channel Controller (and associated channels), but not both, for use in standard PSW mode; for example, Channel Controller 1 can be assigned as Channel Controller 0 at the Configuration Unit control panel. If the other CPU is in Extended PSW mode, both Channel Controllers can be active for that CPU. In this case, Channel Controller 1 could appear to one CPU as Channel Controller 0 and to the other as Channel Controller 1.

Both CPUs can be in the same PSW mode or in different PSW modes. Therefore, OS/360 can be run on either subsystem, and TSS/360, OS/360, or a maintenance function can occupy the remaining subsystem.

Four new units are introduced by the Model 67.
1. IBM 2067 Processing Unit
2. IBM 2846 Channel Controller
3. IBM 2167 Configuration Unit
4. IBM 2365 Processor Storage Model 12

The following paragraphs contain detailed descriptions of these units.

IBM 2067 PROCESSING UNIT

The IBM 2067 Processing Unit is available in two models. The 2067-1 is used for single-processor configurations only. It is characterized by direct attachment of up to seven channels, storage control both for the processor and for channels, and a direct control feature capable of controlling and/or synchronizing special external devices. The 2067-2, designed for use in duplex configurations, is characterized by addressing capability for 14 I/O channels, attachment of a maximum of two Channel Controller units, processor storage control, and an extended direct control feature capable of communicating control information between two 2067-2s. When a 2067-1 is used, the configuration is called a Model 67-1 system; when one or two 2067 Model 2s are used, the configuration is called a Model 67-2 system. These systems are shown in Figures 1 and 2.

The 2067 contains the facilities for (1) addressing processor storage, (2) instruction decoding and execution, and (3) performing arithmetic and logical functions. Except where otherwise noted, the following descriptions apply to both models of the 2067 and to 24-bit addressing.

Processor Storage Addressing

The Bus Control Unit (BCU) located in the CPU controls processor storage addressing. Addresses are communicated to processor storage via a storage address bus (a multiplex bus since it is connected from the BCU to each Processor Storage Unit in a chain-like fashion). The BCU sends a select storage signal to each Processor Storage Unit; this signal requests a storage cycle and identifies the storage unit that is to sample the address bus.

One difference between the 2067-1 and the 2067-2 is in the functions performed by the BCU. In the 2067-1, the BCU provides access to processor storage both for processor references and for channel requests. The 2067-2 BCU accesses processor storage for the CPU only. In configurations using a 2067-2, channel requests are communicated to processor storage via the Channel Controller.

When handling CPU storage requests, the BCU stops the CPU clock when the channel has taken priority, when the referenced Processor Storage Unit is busy, or when the data required on a fetch cycle has not arrived in the time expected by the CPU.

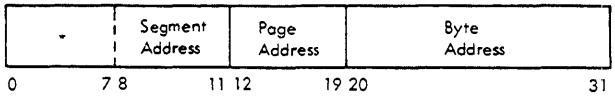
Floating Addressing

Floating storage addressing switches distributes select signals to the appropriate Processor Storage Unit. One eight-position switch is associated with each Processor Storage Unit. These positions correspond to the address intervals possible in any system. A three-bit field from the actual address is decoded into one of the eight possible selection signals (Figure 4). The unit associated with the switch selecting the interval (1) receives the request for a storage cycle and (2) samples the address from the address bus when the request is honored. This decoding and selection is accomplished in the BCU of the CPU and Channel Controller.

Dynamic Address Translation

Another factor affecting the addressing of processor storage is dynamic address translation, which records the relationship between the virtual address and the physical storage address used. This record is important to the segmentation and paging techniques previously described. The processor storage location of each user's page is recorded sequentially in a page table by virtual storage address. The page table's starting address is recorded in the user's segment table, which is loaded in processor storage. (See Table 1.) Each time the user is given control of the processor, the starting address of the segment table is loaded into control register 0 (the segment table register) by the control program.

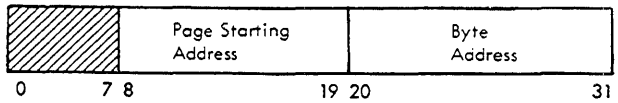
The virtual address operand is formed from the base address, index, and/or displacement, as determined by the instruction format. The resulting virtual operand address has three parts:

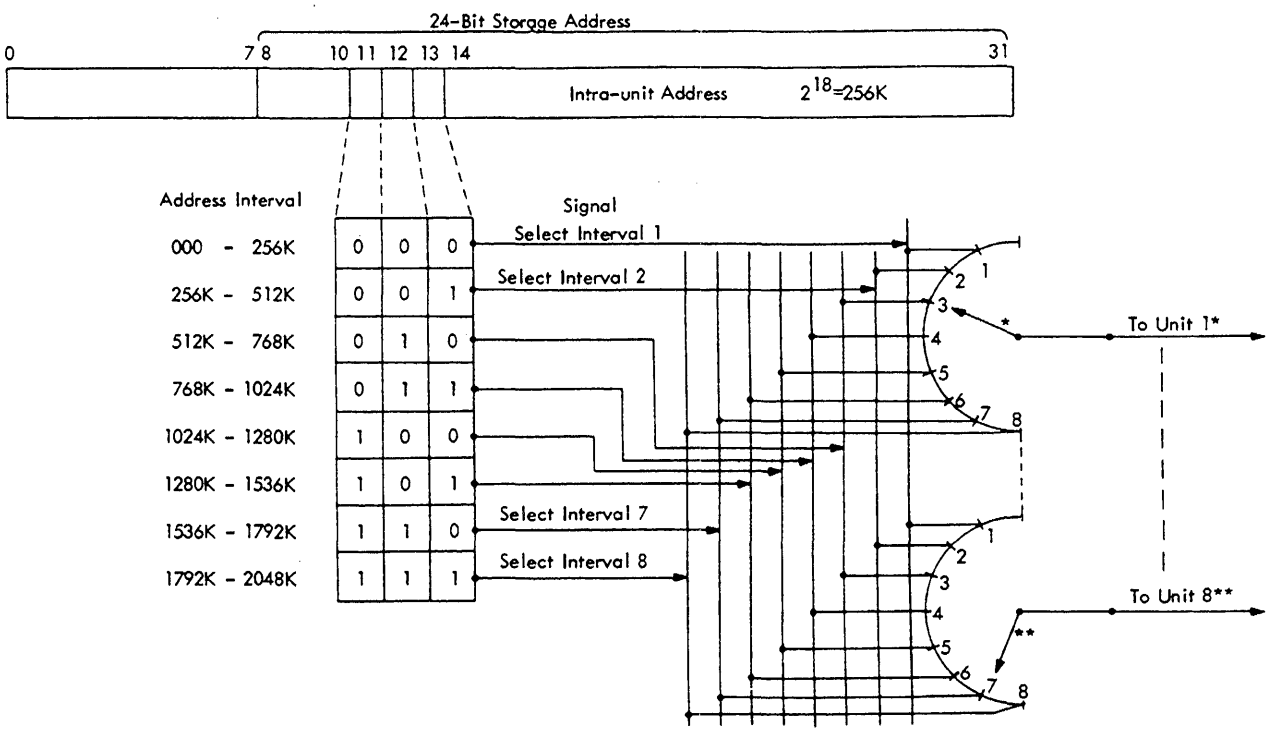


* For 32-bit addressing, bits 0-7 are part of the segment address; otherwise, these bits are not used.

Each part of this virtual address is used to retrieve the physical address of the operand (Figure 5).

The physical address of the operand consists of a byte address (12 low-order bits) and a page starting address (12 high-order bits):





* Storage unit 1 will be selected by addresses between 512K and 768K.

** Storage unit 8 will be selected by addresses between 1536K and 1792K.

Figure 4. Address Interval Assignment

Bits 20–31 of the virtual address provide the byte address, and bits 0–11 of a page table entry are the source of the page starting address. The segment address and page addresses in the virtual address are used to locate the appropriate page table entry.

The segment table register (control register 0) contains the address, in processor storage, of the segment table. The segment table contains one entry for each segment. (This entry is the address of the page table for that segment.) The segment address (from the virtual address) is added to the segment table origin to locate the page table origin for that segment. Then, the page address (from the virtual address) is added to the page table origin. This locates the physical starting address of the page in processor storage. (The page table contains a maximum of one entry for each page in the segment.) The byte address from the virtual address is then combined with the page starting address to form the physical operand address. (See Table 2.)

To avoid repeating this translation process for every memory reference by a user program, the page table entry (page starting address, bits 8–19) is recorded with, and identified by, its virtual address (segment and page address, bits 8–19) in an associative storage register. If a subsequent reference is within that virtual page, the virtual address accesses the associative register. The page starting address stored in the register is affixed to the byte address and forwarded to the BCU.

Eight associative storage registers are in the associative array; each stores an individual page address. (See Table 3.) Bits 36 and 37 of each register reflect the validity and the usage of that entry, respectively. Both bits are set to 1 when the register is loaded, and both are reset to 0 when the segment table register (control register 0) contents are changed. If, within a user's time slice, eight separate pages are referenced (i.e., all associative registers have been loaded and each bit 37 is a 1), all bit 37s are reset to 0. Subsequent references to a register will cause bit 37 for that register to be set. During subsequent translations, the result will be loaded into the lowest-numbered register that has a 0 in a bit 37. Bit 37 is set to a 1 as a result of this load. With this technique, the associative array always contains the most recent and/or most frequently used page addresses.

Since instructions frequently occur in sequence, an "IC register" is implemented to hold and provide the translated page address for use in serial instruction sequences; this avoids repeated accesses of the page address from the associative array.

Dynamic address translation applies only when the CPU is operating in Extended PSW mode. Bit 5 of the extended PSW controls whether translation is active (bit 5=1) or

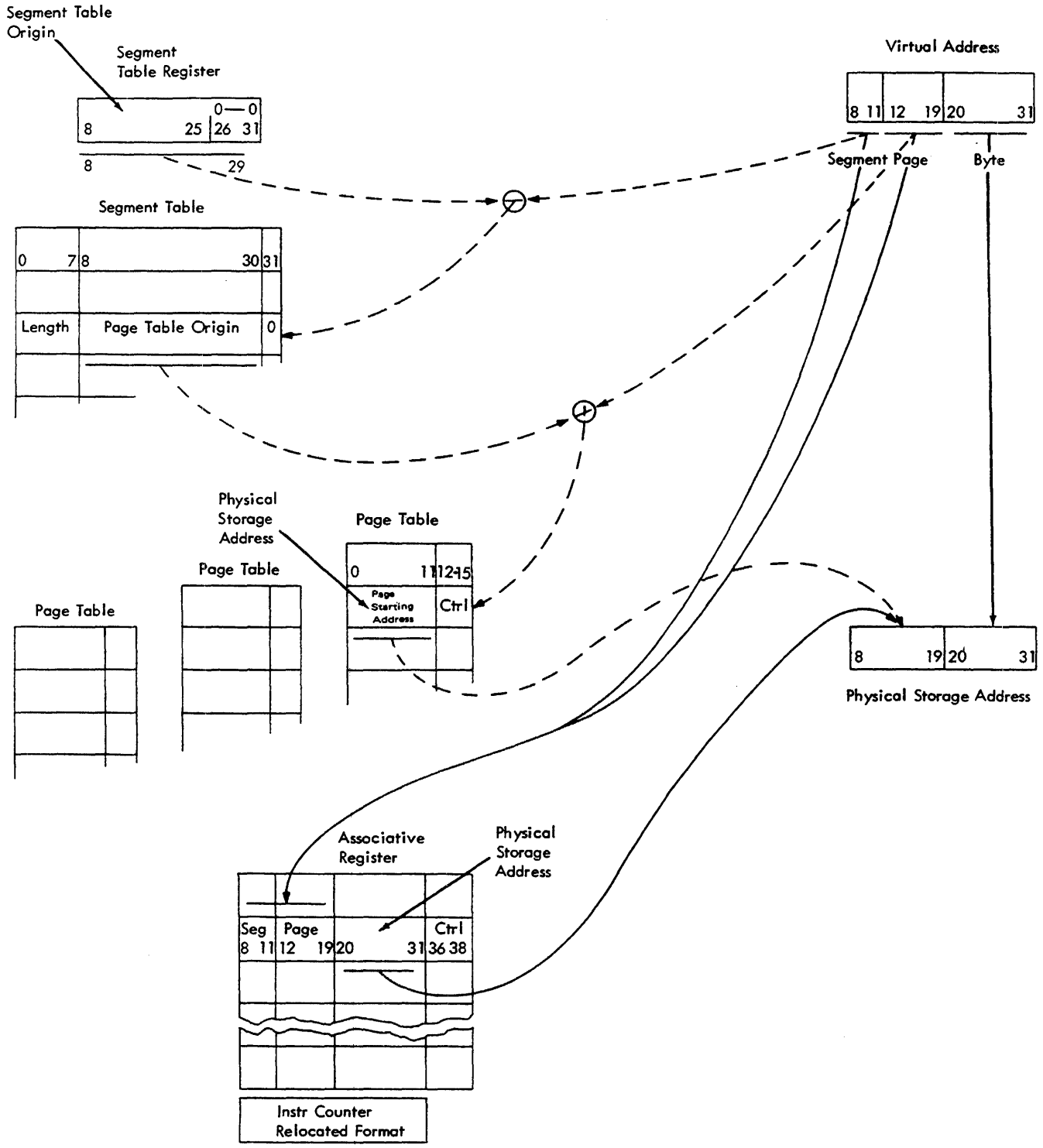


Figure 5. Data Flow for Dynamic Address Translation (24-Bit)

Table 1. Format of Registers and Entries

Bits	Meaning	Remarks
1. Segment Table Register Format:		
0–7	Segment Table Length*	Indicates the number of 16-entry groups in the segment table. All zero = One group.
8–31	Segment Table Origin	Since the segment table origin is located on a 64-byte boundary, bits 26–31 must be zero.
2. Segment Table Entry Format:		
0–7	Page Table Length	Indicates the number of entries in the page table. All zero = One entry.
8–30	Page Table Origin	The page table origin is located on a 2-byte boundary.
31	Page Table Availability	1 = Segment translation exception (program interrupt code 16).
3. Page Table Entry (Halfword):		
0–11	Physical Block Address	Starting address of page.
12	Page Availability	1 = Page translation exception (program interrupt code 17).
13–15	Control Bits, Reserved	Must be 000 or specification exception.

*Used only with CPU's that contain the 32-bit addressing feature.

Table 2. Bit Alignment of Address Arithmetic

1. Segment Table Entry Address

Bits (24-Bit Mode)	Meaning	Remarks
8–31	Table Origin	Bits 26–31 are considered zero.
8–11	Added to Logical Address	Aligned with bits 26–29 of segment table origin.
8–31	Yields Sum	Segment table entry address (bits 30–31 always zero).

Bits (32-Bit Mode)	Meaning	Remarks
8–31	Table Origin	Bits 26–31 are considered zero.
0–11	Added to Logical Address	Aligned with bits 18–29 of segment table origin.
8–31	Yields Sum	Segment table entry address (bits 30–31 always zero).

Table 2. Bit Alignment of Address Arithmetic (Cont)

2. Page Table Entry Address - Either 24- or 32-Bit Mode

Bits	Meaning	Remarks
8–31	Page Table Origin	Bit 31 is considered zero.
12–19	Added to Logical Address	Aligned with bits 23–30 of page table origin.
8–31	Yields Sum	Page table entry address (bit 31 is always zero).

3. Physical Address Result

Bits	Meaning	Remarks
0–11	Page Table Entry	The high-order portion.
20–31	Logical Address	The low-order portion.
8–31	Physical Address	Both portions taken together.

Table 3. Associative Register Format

24-Bit Addressing:

Bits	Content	Remarks
8–19	Virtual Address	
20–31	Physical Address	The page address from a previous translation which corresponds to the virtual address in bits 8–19.
32–35	Unassigned*	
36	Register Valid	Set to 1 upon loading the register.
37	Recent Usage - "Load"	Set to 1 upon loading the register and upon any use thereafter.
38	Disable	Set with special diagnose codes 8–15.

32-Bit Addressing: Same as above except for virtual address, which is in bits 0–19.

*For expansion; not physically implemented.

inactive. When translation is active, all program-generated processor storage addresses originating from the CPU are subject to translation; hardware-generated addresses (e.g., from timer, new PSW fetch, old PSW store, etc.) and storage addresses originating from the channels are not subject to translation.

Prefixing is applied to the address after translation. If the translated address is in the range of 0-4095, further address modification under control of prefixing might occur. A switch on the 2167 control panel can be used to manually deactivate prefixing. The basic prefixing feature provides translation for 24-bit addresses, whereas an optional feature provides translation for either 24-bit or 32-bit addresses (Figure 6). Bit 4 of the extended PSW indicates whether 24-bit or 32-bit address computation arithmetic should apply.

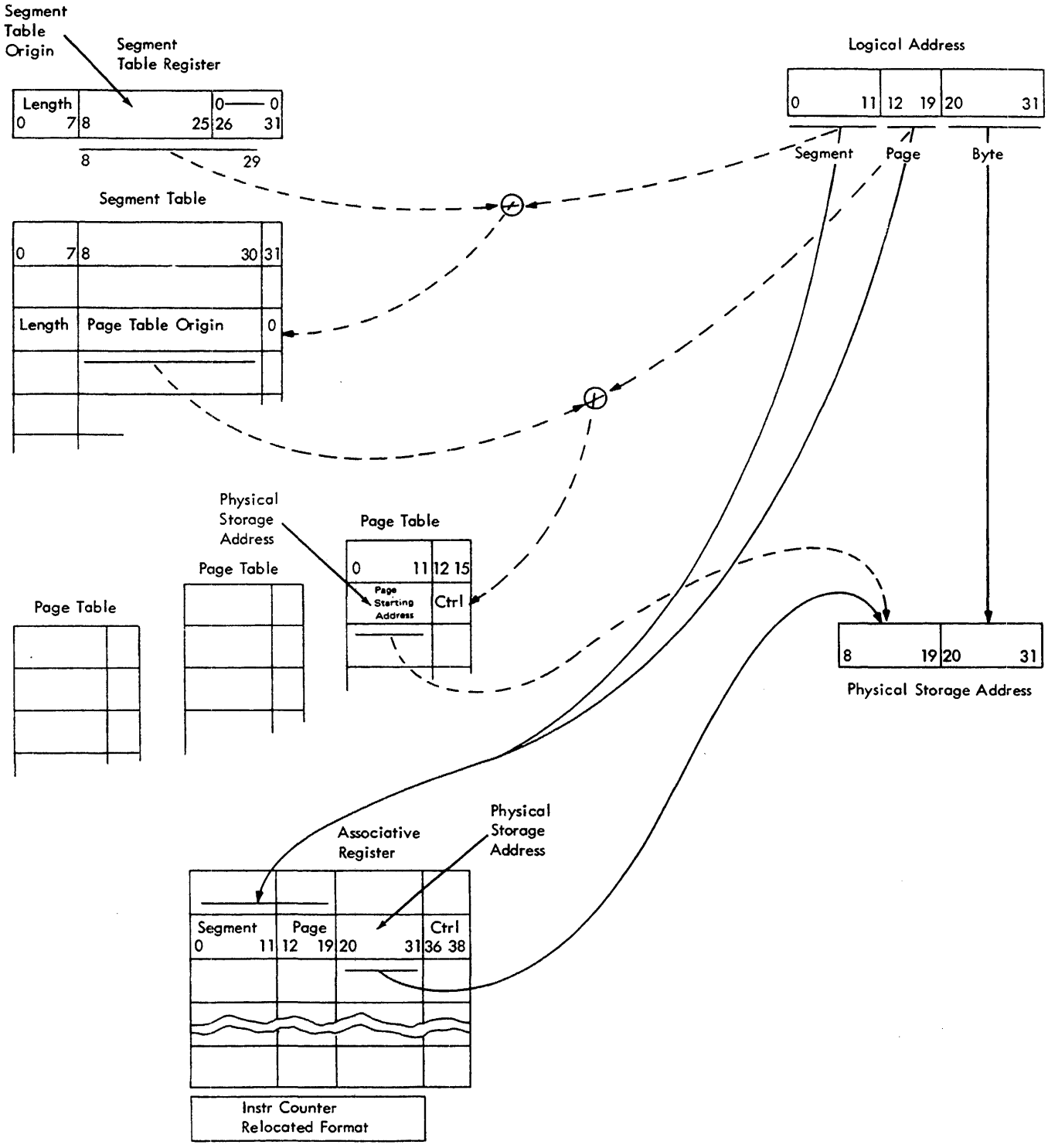


Figure 6. Data Flow for Dynamic Address Translation (32-Bit)

Timing

Instruction execution timing in the 2067 is affected by the dynamic address translation process as follows. When dynamic address translation is active and the translated address is in the associative array, no table references are required. The necessary physical address is made available in 150 ns, thus adding 150 ns to each storage reference for an operand. Consecutive instructions addressed sequentially incur the additional 150 ns on the first array reference only. Succeeding instructions in the sequence use the translated address from the instruction counter register, which adds no additional time to the operand reference.

Instructions which are located in a different page, or which are in the same page block but are reached by branching, require either the 150 ns (as above) or the longer lookup time described in the following paragraph.

When address translation is active and the physical address is not in the associative array, translation is obtained through two storage references. Thus, the approximate time required is (1) two storage references, one each for segment table and page table entries, plus (2) the storage reference for data, plus (3) 750 ns. The exact time depends on the other CPU or Channel Controllers requiring access to the same storage unit during the translation operation.

Channel Addressing

The 2067-1 can be attached to a maximum of seven channels in either of two configurations: (1) one 2870 and one 2860-3; or (2) one 2870, one 2860-2, and one 2860-3. The first 2870 can have up to four selector subchannels, whereas the second can have a maximum of two selector subchannels. Channel and device addressing is performed as described in the *IBM System/360 Principles of Operation* manual. There is little difference in channel address handling when the 2067-1 is operating in standard PSW mode or in extended PSW mode.

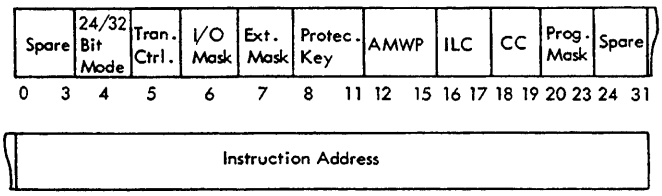
The 2067-2 can attach two Channel Controllers, each of which can attach up to seven channels. Channel addressing is preceded by selection of a Channel Controller. The channel address field of the effective address formed by an I/O instruction is expanded to provide for Channel Controller addressing. In the extended PSW mode of operation, bits 16–20 of the effective address are decoded to select one Channel Controller. Addresses 00100 through 11111 are decoded as invalid Channel Controller addresses. Instructions that specify these addresses are terminated without selection of a 2846, and condition code 3 is set in the PSW. Instructions that specify addresses 00010 and 00011 are also terminated without selection of a 2848; however, no condition code is set.

When the 2067 is operating in standard PSW mode, the channel controller compatibility addressing switch controls selection of a Channel Controller. The setting of this switch determines the Channel Controller to which the start I/O signal, I/O interrupt mask bits, channel and device addresses, etc. will be distributed. Since either Channel Controller can be selected for use by the CPU that is operating in standard PSW mode, I/O instructions in programs that operate in standard PSW mode must address Channel Controller 0 (i.e., bits 16–20 of the channel address field must be 0s).

When the system is in the standard PSW mode, the storage protection key is sent to the Channel Controller specified by the channel controller compatibility addressing switch.

Instruction Fetching and Execution

The PSW controls instruction sequencing and holds and indicates the status of the system relative to the program currently being executed. When operating in the standard PSW mode, the PSW format is identical with the format described in the *IBM System/360 Principles of Operation* manual. The PSW format is redefined as follows when operating in the extended PSW mode:



Bit	Meaning	
0-3	Spare (must be 0 s)	
4	24/32-Bit Address Mode	
5	Translation Control	
6	I/O Mask (Summary)	
7	External Mask (Summary)	
8-11	Protection Key	
12	ASCII-8 Mode (A)	
13	Machine Check Mask (M)	
14	Wait State (W)	
15	Problem State (P)	
16-17	Instruction Length Code (ILC)	
18-19	Condition Code (CC)	
20-23	Program Mask	
	20	Fixed Point Overflow Mask
	21	Decimal Overflow Mask
	22	Exponent Underflow Mask
	23	Significance Mask
24-31	Spare	
32-63	Instruction Address	

The content of several PSW fields is further specified, as follows. Unless bits 0–3 are 0s, a specification exception is generated. When the 32-bit address option is not installed, bit 4 must be 0. Specification exceptions are generated if this format is violated. These specification exceptions are recognized either during execution of the instruction that follows the loading of the PSW or following the Set System Mask instruction.

Fetching

Instruction fetching by the CPU (1) retrieves machine instructions from the storage units and (2) performs the operations common to many instructions. More specifically, the instruction fetching process ensures a correct value in the instruction counter, fetches instructions from storage, determines the format (RR, RX, RS, SI, and SS) of the fetched instructions, and loads the appropriate registers with operands and data fields. Then, control is transferred to the execution phase, in which specific machine instructions are decoded and executed.

Each instruction fetch retrieves one doubleword from storage. Since instructions vary from one to three halfwords in length, as few as 1-1/3 instructions (SS format) or as many as four complete instructions (RR format) can be retrieved from the processor storage unit during one fetch operation. The instruction counter is incremented by 8 each fetch operation.

Operands specified by RR format instructions are placed directly into the desired data flow registers during the instruction fetch operation. Operands specified by RX, RS, SI, and SS format instructions require effective address generation; when the effective address is calculated, control is passed to the operations necessary to complete execution of the instruction.

Execution

Instruction execution is a series of functional operations performed in a specific sequence. These operations are determined by the operation code of the instruction to be executed and by certain other environmental conditions; frequently, the relative importance of these environmental conditions is a function of the type of instruction currently controlling the CPU. Read-only storage (ROS), a fundamental device within the 2067, controls instruction execution. Each ROS address contains predetermined information that is used as required during instruction execution to control data flow. The program does not address ROS; instead, ROS operations are directed by conditions established by specific instructions and/or by their execution. Information in ROS is indestructible and can be modified only by physically changing the ROS element.

Interruptions

Since the Model 67 has extended I/O capability in respect to the standard System/360, it has expanded control over I/O interruptions. This control is accomplished by using additional control registers in the 2067. Both logic registers and switch registers (partitioning switches) are used as control registers (Table 4).

Table 4. Control Registers

Register	Bit Position Assignments	
0	Segment Table Register (for Dynamic Address Translation)	
1	Unassigned*	
2	Translation Exception Address Register	
3	Unassigned*	
4	Extended Mask Registers for I/O Channel Masks:	
	Bits 0–6:	I/O Masks for Channels 0–6.
	Bit 7:	Summary Bit; set to 1 if bits 0–6 not all 0s.
	Bits 8–14:	I/O masks for Channels 7–13.
	Bit 15:	Summary bit; set to 1 if bits 8–14 not all 0s.
	Bits 16–31:	Reserved
5	Unassigned*	
6	Bits 0, 1:	Machine Check Mask Extensions for Channel Controllers
	Bits 2, 3:	Reserved
	Bits 4–7:	Unassigned*
	Bit 8:	Extended Control Mode
	Bit 9:	Configuration Control Bit; defines when partitioning can take place.
	Bits 10–23:	Unassigned*
	Bits 24–31:	External interruption masking as follows:
	Bit Position	Interruption Source
	24	Timer
	25	Interrupt Key
	26	Malfunction Alert - CPU 1 (Ext. Sig. 2)
	27	Malfunction Alert - CPU 2 (Ext. Sig. 3)
	28	Reserved (Ext. Sig. 4)
	29	Reserved (Ext. Sig. 5)
	30	External Interrupt - CPU 1, 2 (Ext. Sig. 6)
	31	Reserved (Ext. Sig. 7)
7	Unassigned	
8–14	Partitioning Sensing Registers. See Table 5 for register layout.	
15	Unassigned*	

*For expansion; not physically implemented.

**Control register 6, bits 0, 1, and 8 are used in the basic 2067-2, whereas bits 9 and 24–31 are used only in a 2067-2 equipped with the extended direct control feature. Only bit 8 is used in a 2067-1.

The CPU distributes one machine check mask bit to each Channel Controller. Control register 6 (external machine check mask register) bits 0 and 1 are used for this function. Distribution of these mask bits is not affected by the PSW mode of operation and is independent of the channel controller compatibility addressing switch; PSW bit 13 is a summary mask for all machine check interrupts. The following situations are possible:

PSW Bit 13	Control Register 6 Bit 0	Bit 1	Remarks
0	x	x	Regardless of status of bits 0 and 1 of control register 6, all machine checks are masked off by PSW bit 13.
1	0	0	Only CPU machine checks will be recognized.
1	0	1	CPU and Channel Controller 1 machine checks will be recognized.
1	1	0	CPU and Channel Controller 0 machine checks will be recognized.
1	1*	1*	All machine checks will be recognized.

*These bits are set to 1 by system reset. Program systems, other than TSS/360, that do not contain instructions for modifying control register 6 must use PSW bit 13 for control over machine check interruptions.

Control register 6 also contains external interrupt mask bits. This field is added to control register 6 by the extended direct control feature. Bits 24 through 31 of control register 6 are always active as mask bits and are not affected by mode changes of the PSW. They are all set to 1 by system reset, power on reset, initial program load, IPL and external start. Bit 7 of either PSW is a summary mask for external interrupts; it must be a 1 for any external interrupts to be recognized.

The Model 67 is compatible with the interrupt capabilities described in the *IBM System/360 Principles of Operation* manual. In addition, the following program- and machine-check interruptions are recognized.

Program Interruptions

Specification Exception. A specification exception (interruption code 6) also is recognized:

1. When bit 4 of the extended PSW is a 1 when the 32-bit addressing option is not installed.
2. When bits 0–3 of the extended mode PSW are not 0s.
3. When bits 13–15 of the page table entry accessed during an address translation are not 0s.

Specification exceptions relating to the extended PSW are recognized during execution of the instruction that follows the loading of the PSW or follows the Set System Mask instruction. For the situation listed as item 3 above, the operation is suppressed. The instruction length code is 1, 2, or 3.

Data Exception. Data exception (interruption code 7) is also generated if bits 26–31 of the segment table register (control register 0) are not 0s. It is recognized following execution of the instruction that loads the segment table register. When data exception is recognized, the resulting interrupt request is handled as described in the *IBM System/360 Principles of Operation* manual. The operation is terminated, and the instruction length code is 2.

New Exceptions. Because of dynamic address translation, two entries are added to the list of program interruptions:

Interruption Code			Program Interruption Cause
Dec	Hex	Binary	
16	10	0001 0000	Segment Translation
17	11	0001 0001	Page Translation

A segment-translation exception is recognized:

1. When in 32-Bit Address mode (bits 0–7 of the segment table register indicate the segment table length) and the quantity in bits 0–7 of the logical address is greater than the segment table length, or
2. When the segment table entry is accessed, and bit 31 is a 1. (Bit 31 indicates the availability of a page table for that segment and is 0 when a page table has been established for that segment.)

A page-translation exception is recognized:

1. When the quantity in bits 12–19 of the logical address is greater than the page table length. (The quantity in bits 0–7 of the segment table entry is the page table length.)
2. When the page table entry is accessed and bit 12 is a 1. (Bit 12 of the page table entry, the availability bit, is 0 when the page represented by that entry is in the core address identified by the entry.)

Whenever a segment- or page-translation exception is recognized, the instruction is suppressed and storage is not modified. The virtual address that was to be translated is recorded in the translation exception address register (control register 2) and, if in 24-Bit Address mode, bits 0–7 are made 0s. The instruction length code for both exceptions may be 1, 2, or 3.

Machine Check Interruption

In addition to the machine check capability described in the *IBM System/360 Principles of Operation* manual, the 2067 performs machine checking on the dynamic address translation unit. Additional information about the machine check interruption is stored in the first byte (bits 0–7) of the translation exception address register (control register 2). Bits 50–54 in the maintenance control word (a diagnostic tool) indicate to the monitor whether a dynamic address translation unit machine check has caused the interruption; if it has, a 1-bit in the translation exception address register indicates one of the following conditions (24-bit and 32-bit addressing except where noted):

Bit	Condition
0	More than one associative register contains identical information, or one of the comparing circuits is at fault.
1	One of three conditions: a. Hardware error. A successful compare was achieved with a virtual address that was higher than the address in the segment table. b. Program interruption code 16 (32-bit addressing only). Virtual address bits 0–7 were greater than segment table register bits 0–7. c. Program interruption code 17. Virtual address bits 12–19 were greater than segment table entry bits 0–7.
2	The virtual address portion of the translated address just stored in the associative array does not compare with the virtual address that should have been stored.
3	A reset of the load-valid bits in the associative array was unsuccessful.
4	Parity of the adder sum is inconsistent with the predicted parity.
5	Parity of the virtual address was incorrect when received by the associative array.
6	Parity of the data word from storage was incorrect when received by dynamic address translation circuitry.
7	Parity of instruction bits 8–15 was incorrect when received by dynamic address translation circuitry.

When a machine check interruption is recognized, execution of the instruction is terminated, and the interrupt request is handled as described in the *IBM System/360 Principles of Operation* manual except as follows. When a machine check interrupt is caused by an external machine check (I/O machine check), the CPU (1) completes execution of the current instruction, (2) does not initiate an internal diagnostic procedure, and (3) does not perform a diagnostic scan-out into the storage area beginning at location 128. For both machine check and external machine check, the CPU stores the old PSW in location 48 and loads the new machine check PSW from location 112. Bit 17 of the old machine check PSW identifies the CPU mode of operation at the time of interruption. Bit 17 is stored as a 1 when the CPU is in Extended PSW mode and as a 0 in standard PSW mode. The high-order bit of the interrupt code (bit 16 of old machine check PSW if CPU is in standard PSW mode) indicates whether the machine check is an external machine check. This high-order bit is a 1 for external machine checks and a 0 for CPU machine checks.

Interrupt Codes

In Extended PSW mode, when the PSW is stored as the old PSW, the 16-bit interruption code is stored as a halfword in storage, as follows:

Interruption Type	Permanent Storage Address	
	Decimal	Hex
External	14–15	E-F
SVC	16–17	10–11
Program	18–19	12–13
Machine check	20–21	14–15
I/O	22–23	16–17

These bytes are not loaded when loading a PSW.

Extended Direct Control

The extended direct control feature enables direct communication of control information between two CPU's. Each CPU in a duplex system must be equipped with this feature. Extended direct control is a modification of the multisystem operation and the direct control feature described in the *IBM System/360 Principles of Operation* manual. Two major modifications have been implemented by this feature. The Read Direct instruction has been made an invalid instruction and the 'direct-out' lines are terminated within the CPU. Since the duplex configuration of the Model 67 is a shared storage system, there is no need to transfer a data byte over the 'direct-out' lines to the other CPU. Another CPU's attention can be directed to an area of shared storage via the external interruption signal lines.

The extended direct control feature uses the Write Direct instruction to externally interrupt or externally start another CPU. The following is a list of external interrupts received:

Interrupt Line	CPU 1	CPU 2
2	Not used	Machine check signal from CPU 1
3	Machine check signal from CPU 2	Not used
4, 5	Reserved for future use	
6	'Write direct' signal from CPU 1 or CPU 2	'Write direct' signal from CPU 1 or CPU 2
7	Not used	Not used

Line numbers correspond to external signal designations used with the interruption codes listed in the *IBM System/360 Principles of Operation* manual. Interrupt codes and mask bits for these lines are as follows:

Line	Control Register 6 External Interrupt Mask Bit	Interruption Code
2	26	00000000 nnInnnnn
3	27	00000000 nnnInnnn
6	30	00000000 nnnnnnIn

Write Direct is used in the Model 67 to generate the timing signal 'external interrupt request' (line 6) and the external start signals. The I_2 field of the instruction is divided, by function, into two fields: Bits 8 - 11 are assigned to the external interrupt function, and bits 12 - 15 are assigned to the external start function. Bit 8 is assigned to CPU 1, and bit 9 is assigned to CPU 2. Regardless of which CPU executed the Write Direct instruction, a 1 in bit 9 of the I_2 field will generate external interrupt signal on line 6 of CPU 2. A CPU can interrupt itself.

Bit 12 of the I_2 field is assigned to start CPU 1, and bit 13 is assigned to start CPU 2. Since there is a main 'external start' line and an alternate 'external start' line for each CPU, bit 7 (low-order bit) of the data byte addressed by the Write Direct instruction is used with the I_2 bit(s) to generate the appropriate 'external start' signal. For example, if bit 12 (of the I field) is a 1 and bit 7 (of the data byte) is a 0, a signal is generated on the main 'external start' line of CPU 1. When bit 12 is a 1 and bit 7 is a 1, a signal is generated on the alternate 'external start' line of CPU 1. Note that CPU 1 is started regardless of which CPU executed the instruction. "Main" and "alternate" correspond to the main and alternate prefix values prewired in each CPU. More than one CPU can be signaled by one Write Direct instruction. The 'external start' signal pre-empts current operations in the receiving CPU. The current operation is terminated, the current PSW is replaced by the contents of the new PSW at byte address 0000 (defined as the new external start PSW), and the current PSW is lost. Selection of the prefix value is a function of the line (main or alternate) on which the signal was received. A system reset of all control latches occurs in the receiving CPU, but the reset is not propagated to the I/O components. Therefore, the programmer must ensure, by suitable programming constraints, that the receiving CPU is not executing an I/O instruction and is not accepting an I/O interrupt. If either of these conditions occurs simultaneously with receipt of the external start signal, the results are generally unpredictable and the I/O channel and/or channel control unit may hang up.

The external start sequence is completed after 50 usec. Results are unpredictable for successive external starts issued to one CPU more frequently than one every 150 usec.

Extended direct control can be partitioned by switches on the Configuration Unit. One switch per CPU can be set to remove the CPU from the external direct control interface. When the CPU is partitioned off, machine checkout and timing signals are not generated, and external start and external interrupt-in signals are ignored (do not remain pending). When partitioned, a CPU cannot external-start itself and cannot interrupt itself, although manual IPL (load) and the interrupt pushbutton are still effective.

High-Resolution Interval Timer

An interval timer with a high degree of resolution is used in the 2067. Operation of this timer is fully compatible with that described in the *IBM System/360 Principles of Operation* manual.

The high-resolution timer provides approximately 13-usec resolution. This is accomplished with an 8-bit hardware register which contains the low-order byte of the timer. Each time the low-order byte counts to zero, the timer value at locations 80–82 is decremented at the end of the instruction currently being executed.

An operand fetch from location 80 will retrieve the three high-order bytes from location 80 plus the low-order byte from the hardware register. If the low-order byte has stepped through zero during the instruction, then before a fetch from location 80, zeros are inserted into the low-order byte instead of the contents of the hardware register. Any instruction that stores into location 80 also stores the low-order byte into the hardware register, as well as a full word into location 80. If the timer value at location 80 changes from positive to negative, an external interruption is requested.

PROGRAMMING NOTE: If the compatibility feature is installed and if the system is in Emulator mode, a machine check is not indicated by the machine time-out feature, and the machine-check interruption is lost.

Interruption Times

Interruption times vary for the class of interruption and for the type of instruction being executed at the time of the interruption.

External Interruption

External interruption time is 3.15 usec; it extends from the time the external interruption is discovered and honored to the time the next instruction is started.

Supervisor Call Interruption

Supervisor call interruption time is 3.95 usec (including supervisor call instruction time); it extends from the time the supervisor call is discovered and honored to the time the next instruction is started.

Program Interruption

Program interruption time is 3.15 usec; it extends from the time the program interruption is discovered and honored to the time the next instruction is started. Actual interruption occurs at the end of the current instruction.

Machine Check Interruption

Machine check interruption time is approximately 25 usec; it extends from the time the machine check interruption is discovered to the time the next instruction is started and includes scan-out and reset time.

I/O Interruption

The I/O interruption time is 4.65 usec; it extends from the time that the CPU takes a pending interruption from the channel to the time that the old PSW and CSW are stored.

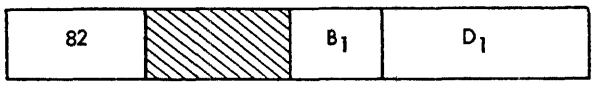
Modified Instructions

Bit 8 of control register 6 determines the 2067 mode of operation. When this bit is a 0, the PSW format is as defined in the *IBM System/360 Principles of Operation* manual (i.e., channels beyond 6 cannot be addressed and cannot cause interrupts). When this bit is a 1, the PSW is interpreted as previously described for Extended PSW mode. Upon system reset (generated by manual system reset, manual IPL, or external start), bit 8 of control register 6 is set to 0.

The following paragraphs describe the performance of existing instructions as modified in Extended PSW mode.

Load PSW Instruction

LPSW $D_1(B_1)$ [S1]



0 7 8 15 16 19 20 31

The doubleword at the location designated by the operand address replaces the PSW.

The operand address must have its three low-order bits 0 to designate a doubleword; otherwise, a specification exception results in a program interruption.

The doubleword that is loaded becomes the PSW for the next sequence of instructions. Bits 8–11 become the new protection key. Bits 32–63 of the doubleword become the new instruction address. The PSW is not checked for program interruptions during the load-PSW operation. These checks occur as part of the execution of the next instructions.

Bits 16 and 17 of the PSW are not retained upon loading. They will contain the instruction length code for the last-interpreted instruction when the PSW is stored (1) during a branch-and-link operation, or (2) during a program or supervisor call interruption.

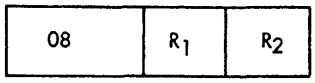
Condition Code. The code is set according to bits 18 and 19 of the new PSW loaded.

Program Interruptions.
Privileged operation
Protection (fetch violation)
Addressing
Specification

PROGRAMMING NOTE: The CPU enters the problem state when LOAD PSW loads a doubleword with a 1 to be consistent in bit position 15 and similarly enters the wait state if bit position 14 is 1.

Set Storage Key Instruction

SSK R_1, R_2 [RR]



0 7 8 11 12 15

The key of the storage block addressed by the register designated by R_2 is set according to the key in the register designated by R_1.

The storage block of 2048 bytes, located on a multiple of the block length, is addressed by bits 8–20 of the register designated by the R_2 field. Bits 0–7 and 21–27 of this register are ignored. Bits 28–31 of the register must be 0; otherwise, a specification exception causes a program interruption.

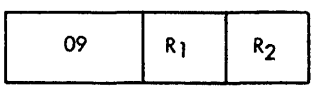
The 7-bit key is obtained from bits 24–30 of the register designated by the R_1 field. Bits 0–23 and bit 31 of this register are ignored.

Condition Code. The code remains unchanged.

Program Interruptions.
Privileged operation
Addressing
Specification

Insert Storage Key Instruction

ISK R_1, R_2 [RR]



0 7 8 11 12 15

The key of the storage block addressed by the register designated by R_2 is inserted in this register designated by R_1.

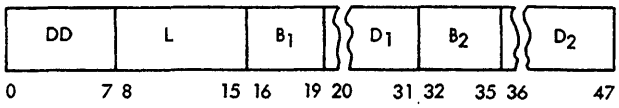
The storage block of 2048 bytes, located on a multiple of the block length, is addressed by bits 8–20 of the register designated by the R_2 field. Bits 0–7 and 21–27 of this register are ignored. Bits 28–31 of the register must be 0; otherwise, a specification exception causes a program interruption. The 7-bit key is inserted in bits 24–30 of the register specified by the R_1 field. Bits 0–23 of this register remain unchanged, and bit 31 is set to 0.

Condition Code. The code remains unchanged.

Program Interruptions.
Privileged operation
Addressing
Specification

Translate and Test Instruction

TRT $D_1(L, B_1)$, $D_2(B_2)$ [SS]



The eight-bit bytes of the first operand are used as arguments to reference the list designated by the second operand address. Each eight-bit function byte thus selected from the list is used to determine the continuation of the operation. When the function byte is a 0, the operation proceeds by fetching and translating the next argument byte. When the function byte is nonzero, the operation is completed by inserting the related argument address in general register 1 and by inserting the function byte in general register 2.

The bytes of the first operand are selected one by one for translation, proceeding from left to right. The first operand remains unchanged in storage. Fetching of the function byte from the list is performed as in the Translate instruction description. The function byte retrieved from the list is inspected for all-zero combination.

When the function byte is 0, the operation proceeds with the next operand byte. When the first operand field is exhausted before a nonzero function byte is encountered, the operation is completed by setting the condition to 0. The contents of general registers 1 and 2 remain unchanged.

When the function byte is nonzero, the related argument address is inserted in the 24 low-order bits of general register 1. This address points to the argument last translated. The eight high-order bits of register 1 remain unchanged. When operating in 32-Bit Address mode (PSW bit 4=1), the related argument address is inserted in all 32 bits of general register 1.

The function byte is inserted in the eight low-order bits of general register 2. Bits 0–23 of register 2 remain unchanged. The condition code is set to 1 when one or more argument bytes have not been translated. The condition code is set to 2 if the last function byte is nonzero.

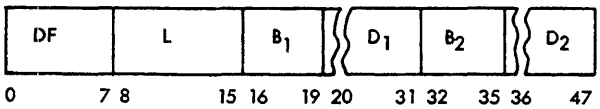
Resulting Condition Code.
0 All function bytes are zero
1 Nonzero function byte before the first operand field is exhausted
2 Last function byte is nonzero
3 —

Program Interruptions.
Protection (fetch violation)
Addressing

PROGRAMMING NOTE: Translate and Test is useful for scanning an input stream and locating delimiters. The stream can thus be rapidly broken into statements or data fields for further processing.

Edit and Mark Instruction

EDMK $D_1(L, B_1)$, $D_2(B_2)$ [SS]



The format of the source (the second operand) is changed from packed to zoned and is edited under control of the pattern (the first operand). The address of each first significant result digit is recorded in general register 1. The edited result replaces the pattern.

The operation is identical with edit, except for the additional function of inserting a byte address in general register 1. The use of general register 1 is implied. The byte address is inserted in bits 8–31 of this register. Bits 0–7 are not changed. When operating in 32-Bit address mode, however, the full 32-bit byte address is inserted in general register 1. For that situation, bits 0–7 may be changed.

The byte address is inserted each time the 'S' trigger is in the zero state and a nonzero digit is inserted in the result field. The address is not inserted when significance is forced by the significance-start character of the pattern, while inserting a fill character in the result field.

Resulting Condition Code.
0 Result is zero
1 Result field is less than zero
2 Result field is greater than zero
3 —

Program Interruptions.
Operation (if decimal feature is not installed)
Protection (store or fetch violation)
Addressing
Data

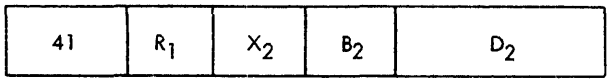
PROGRAMMING NOTES: Edit and Mark facilitates the programming of floating currency-symbol insertion. The character address inserted in register 1 is one more than the address where a floating currency-sign would be inserted. The Branch on Count, with zero in the R_2 field, may be used to reduce the inserted address by one.

The character address is not stored when significance is forced. Therefore, the address of the character following the significance-start character should be placed in register 1 prior to Edit and Mark.

When a single instruction is used to edit several numbers, the address of the first significant digit of each number is inserted in general register 1. Only the last address will be available after the instruction is completed.

Load Address Instruction

LA R_1D_2 (X_2B_2) [RX]



In 24-Bit Address mode, the address specified by the X_2, B_2, and D_2 fields is inserted in bits 8–31 of the general register specified by R_1. Bits 0–7 are set to zero. The address is not inspected for availability, protection, or resolution. The address computation follows the rules for address arithmetic. Any carries beyond the 24th bit are ignored, and no storage references for operands take place.

In 32-Bit Address mode, the address specified by the X_2, B_2, and D_2 fields is inserted in the general register specified by R_1. The address is not inspected for availability, protection, or resolution. The address computation follows the rules for address arithmetic. Any carries beyond the 32nd bit are ignored, and no storage references for operands take place.

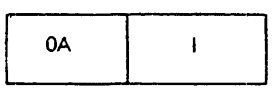
Condition Code. The code remains unchanged.

Program Interruptions. None

PROGRAMMING NOTE: The same general register may be specified by the R_1, X_2, and B_2 instruction fields, except that general register 0 can be specified only by the R_1 field. In this manner, it is possible to increment the contents of a general register, other than 0, by the contents of the D_2 field of the instruction. The register to be incremented should be specified by R_1 and by either X_2 (with B_2 set to 0) or B_2 (with X_2 set to 0).

Supervisor Call Instruction

SVC I [RR]



The instruction causes a supervisor call interruption, with the I-field of the instruction providing the interruption code.

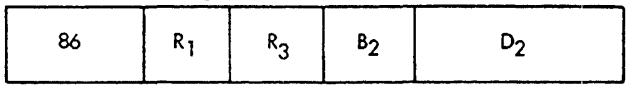
The contents of bit positions 8–15 of the instruction are placed in byte locations 16–17 of core storage in the course of the interruption. The old PSW is stored at location 32, and a new PSW is obtained from location 96. The instruction is valid in both problem and supervisor state.

Condition Code. The code remains unchanged in the old PSW.

Program Interruptions. None

Branch On Index High Instruction

BXH R_1, R_3, D_2 (B_2) [RS]



An increment is added to the first operand, and the sum is compared algebraically with a comparand. Subsequently, the sum is placed in the first operand location, regardless of whether the branch is taken. When the sum is high, the instruction address is replaced by the branch address. When the sum is low or equal, instruction sequencing proceeds with the updated instruction address.

The first operand and the increment are in the registers specified by R_1 and R_3. The comparand register address is odd and is either one larger than R_3 or equal to R_3. The branch address is determined prior to the addition and comparison.

Overflow caused by the addition is ignored and does not affect the comparison. Otherwise, the addition and comparison proceed as in fixed-point arithmetic. All 32 bits of the general registers participate in the operations, and negative quantities are expressed in two's-complement notation. When the first operand and comparand locations coincide, the original register contents are used as the comparand.

Condition Code. The code remains unchanged.

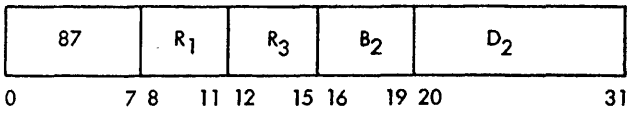
Program Interruptions. None

PROGRAMMING NOTE: Branch on Index High indicates that one of the major purposes of this instruction is the incrementing and testing of an index value. The increment may be algebraic and of any magnitude.

When Branch on Index High is used for address modification in 32-Bit Address mode and bit 0 of the first operand becomes a 1 as a result of the logical addition, the resulting address will be treated as a negative quantity during the compare process; this can result in an erroneous branch. For that magnitude of address manipulation, care must be taken in the use of this instruction.

Branch On Index Low or Equal Instruction

BXLE $R_1, R_3, D_2 (B_2)$ [RS]



An increment is added to the first operand, and the sum is compared algebraically with a comparand. Subsequently, the sum is placed in the first operand location, regardless of whether the branch is taken. When the sum is low or equal, the instruction address is replaced by the branch address. When the sum is high, normal instruction sequencing proceeds with the updated instruction address.

The first operand and the increment are in the registers specified by R_1 and R_3. The comparand register address is odd and is either one larger than R_3 or equal to R_3. The branch address is determined prior to the addition and comparison.

This instruction is similar to Branch on Index High, except that the branch is successful when the sum is low or equal compared with the comparand.

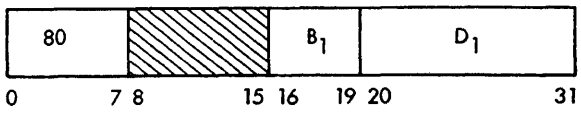
Condition Code. The code remains unchanged.

Program Interruptions. None

PROGRAMMING NOTE: The same exposure to erroneous branching exists for this instruction with 32-bit addressing during address modification as described for Branch on Index High.

Set System Mask Instruction

SSM $D_1 (B_1)$ [SI]



The byte at the location designated by the operand address replaces bits 0–7 of the current PSW.

Condition Code. The code remains unchanged.

Program Interruptions.
Privileged operation
Protection (fetch violation)
Addressing
Specification

PROGRAMMING NOTE: When issued in extended mode, this instruction no longer sets the system mask as defined in the *IBM System/360 Principles of Operation* manual because bits 0–7 of the current PSW are redefined as follows:

0–3	Spare (must be 0s)
4	24/32 Bit Address mode
5	Translation control bit
6	I/O system mask
7	External mask

The individual channel mask bits must be set with the Load Multiple Control instruction.

New Instructions

Five new instructions are provided by the 2067:

Instruction	Mnemonics	Type	Exceptions	Code
Load Multiple Control	LMC	RS	M,A,S,P	B8
Store Multiple Control	STMC	RS	M,P,A,S	B0
Load Real Address	LRA	RX	M,A,S	B1
Branch and Store	BASR	RR		0D
Branch and Store	BAS	RX		4D

Notes:
A Addressing Exception
M Privileged Operation Exception
P Protection Exception
S Specification Exception

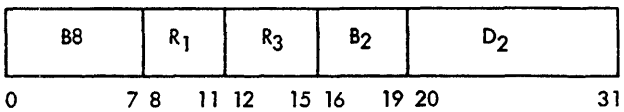
Privileged operation and specification exceptions cause instruction suppression. Addressing and protection exceptions cause instruction termination and leave the setting of the control registers unpredictable.

Load Multiple Control and Store Multiple Control change and inspect the control registers in the same manner as Load Multiple and Store Multiple change and inspect the general registers. Up to sixteen 32-bit registers may be provided; unassigned control registers and unassigned bits within registers are stored as 0s and are ignored when loaded. These instructions use the RS format. The storage address must be located on a word boundary (i.e., the two low-order bits of the address must be 0) or a specification exception is recognized.

Load Multiple Control and Store Multiple Control are privileged operations.

Load Multiple Control Instruction

LMC $R_1, R_3, D_2 (B_2)$ [RS]



The set of control registers starting with the register specified by R_1 and ending with the register specified by R_3 is loaded from the locations designated by the second operand address. Control registers 8 through 14 (the partitioning sensing registers) are not loaded by this instruction; these registers can be loaded only from manual controls.

The storage area from which the contents of the control registers are obtained starts at the location designated by the second operand address and continues through as many words as needed. The control registers are loaded in the ascending order of their addresses, starting with the register specified by R_1 and continuing up to and including the register specified by R_3, with register 0 following register 15. The second operand remains unchanged.

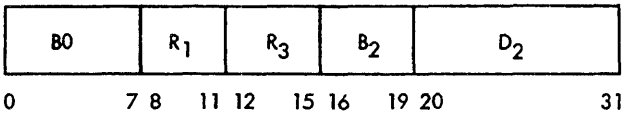
Condition Code. The code remains unchanged.

Program Interruptions.
Protection (fetch violation)
Addressing
Specification
Privileged operation

PROGRAMMING NOTE: All combinations of register addresses specified by R_1 and R_3 are valid. When the register addresses are equal, only one word is transmitted. When the address specified by R_3 is less than the address specified by R_1, the register addresses wrap around from 15 to 0.

Store Multiple Control Instruction

STMC $R_1, R_3, D_2 (B_2)$ [RS]



The set of control registers starting with the register specified by R_1 and ending with the register specified by R_3 is stored at the locations designated by the second operand address.

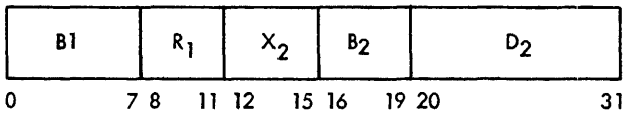
The storage area where the contents of the control registers are placed starts at the location designated by the second operand address and continues through as many words as needed. The control registers are stored in the ascending order of their addresses, starting with the register specified by R_1 and continuing up to and including the register specified by R_3, with register 0 following register 15. The contents of the control registers are unchanged.

Condition Code. The code remains unchanged.

Program Interruptions.
Protection (store violation)
Addressing
Specification
Privileged operation

Load Real Address Instruction

LRA $R_1, D_2 (X_2, B_2)$ [RX]



In 24-Bit Address mode, the translated address of the second operand is inserted in the 24 low-order bits of the general register specified by R_1. The remaining bits of the general register are made 0.

The address specified by the X_2, B_2, and D_2 fields is inserted in bits 8–31 of the general register specified by R_1. Bits 0–7 are set to 0. The address is not inspected for protection or resolution. Address computation follows the rules for address arithmetic. Any carries beyond the 24th bit are ignored.

In 32-Bit Address mode, the translated address of the second operand is inserted into the general register specified by R_1.

The address specified by the X_2, B_2, and D_2 fields is inserted in the general register specified by R_1. The address is not inspected for protection or resolution. Address computation follows the rules for address arithmetic. Any carries beyond the 32nd bit are ignored. Address translation takes place regardless of the status of the translation control bit.

Resulting Condition Code.
0 Translation was successful
1 Translation unsuccessful (unavailability bit encountered in the segment table, or virtual address bits 0–7 > segment table register bits 0–7)
2 Translation unsuccessful (unavailability bit encountered in the table, or virtual address bits 12–19 > segment table entry bits 0–7)
3 —

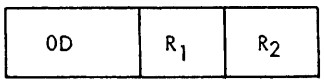
When the resulting condition code is 1 or 2, the real address of the unavailable or nonexistant segment or page table entry is placed in the general register specified by R_1. With 24-bit addressing, condition code 1 can occur only when the unavailability bit is encountered.

Program Interruptions
Addressing
Specification
Privileged operation

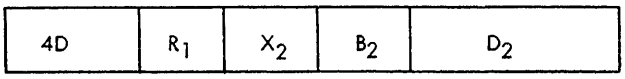
PROGRAMMING NOTE. The Load Real Address instruction does not cause translation exceptions.

Branch and Store Instruction

BASR R_1, R_2 [RR]



BAS R_1, D_2, (X_2, B_2) [RX]



If in standard PSW mode or if in extended PSW mode with 24-Bit Address mode specified (PSW bit 4=0), bits 40–63 of the PSW are stored in the general register specified by R_1. Bits 0–7 of the general register are made 0. If in extended PSW mode with 32-Bit Address mode specified (PSW bit 4=1), bits 32–63 of the PSW are stored in the general register specified by R_1.

The branch address is determined before the updated instruction address is stored. The instruction-length codes are 1 and 2 for the RR- and RX-format Branch and Store instructions, respectively.

Condition Code. The code remains unchanged.

Program Interruptions. None

PROGRAMMING NOTE: The updated instruction address is stored without branching when in the RR format and the R_2 field contains 0.

When Branch and Store is the subject instruction of an Execute instruction, the instruction length code is 2.

Arithmetic and Logical Functions

The 2067 can perform the full range of arithmetic and logical functions described in the *IBM System/360 Principles of Operation* manual.

The CPU operates on a basic internal cycle time of 200 ns. Two major working arithmetic registers in the data path allow high-speed and simplified implementation of the System/360 instruction set. A 60-bit parallel adder facilitates handling of long fractions in floating-point operations. An eight-bit serial adder enables simultaneous execution of floating-point exponent arithmetic and, also, handles the decimal arithmetic and variable field length (VFL) instructions.

Local Store

Local store is a bank of 16 full-word, general-purpose registers and four doubleword floating-point registers. All registers are implemented in transistor logic. They are used as index registers in address arithmetic and indexing and as accumulators in fixed-point, logical, and floating-point operations.

IBM 2846 CHANNEL CONTROLLER

The Channel Controller is used with a 2067-2 only. It provides the time sharing system with increased accessibility to auxiliary storage devices. In a duplex configuration, it provides communication interface between the CPUs and the channels controlling auxiliary storage devices. It also provides paths for control information and data transfers between the Processor Storage Unit and the channels. Relocation of these functions from the BCU of the 2067-2 to the Channel Controller allows them to be used by both CPUs.

The Channel Controller is contained in a single, stand-alone frame equipped with its own power supply. It contains a storage selection element (SSE), a processor interface element (PIE), and a scan/test element. The SSE provides the necessary controls and paths for data transfers between the attached I/O channels and the attached Processor Storage Units. The PIE provides a control path between the same I/O channels and attached CPUs.

The Channel Controller can interface:

1. Up to two 2067-2 CPUs. Additional 2846 attachment features must be added to each CPU that interfaces more than one Channel Controller.
2. Up to eight 2365-12 Processor Storage Units; a 2846 switching feature must be added to each Processor Storage Unit that interfaces a Channel Controller. An additional addressing feature must be added to each Channel Controller to permit the addressing of five to eight 2365-12s.
3. Up to seven physical channels (one 2870 Multiplexer Channel and up to two 2860 Selector Channels, in any combination of models not exceeding cabling limitations). The addressing prefixing feature must be added to each channel.
4. One 2167 Configuration Unit.

Storage Selection Element

The SSE provides the channels attached to the 2846 with unique data and control paths to processor storage. A multiplex bus from the SSE to all Processor Storage Units

provides the path for storage addressing and for I/O data. Simplex selection lines to each Processor Storage Unit communicate requests for storage references to the appropriate unit. The distribution of signals on these select storage lines is controlled by the set floating address switches on the 2167. The assignment of address intervals to storage units directs the select storage signals as described in the 2067 section. The speed at which these requests are serviced depends on the availability of the addressed storage unit and on the cable distance between the storage unit and the Channel Controller.

Simultaneous requests to a Processor Storage Unit by multiple Channel Controllers are handled by the storage switching element on the storage unit.

Each channel that needs a storage cycle sends a service request signal to the SSE. When not busy, the SSE continually scans for service requests. If only one channel requests service, that request is immediately recognized and processed; the SSE does not resume scanning until processing of the request is completed. If two or more channels request service, priority is granted in ascending channel number order (i.e., channel 1 has the highest priority, then channel 2, on up to channel 6, followed by channel 0). When the SSE finishes servicing a request, scanning resumes at channel 1. The high-speed I/O priority on channel 1 ensures immediate servicing at the Channel Controller for the 2301 drum.

The SSE examines the 12 high-order bits of the processor storage address for all 0s. If an all-zero condition exists, and if prefixing is not inhibited, the SSE loads the 12 high-order position with the appropriate prefix.

The prefix value is loaded into the address prior to decoding the set floating address switches.

Processor Interface Element

The PIE provides the communication link between the attached CPUs and channels.

The CPU/Channel Controller interface provides the operational lines used during normal program and interrupt operations of the channels. These lines include the diagnostic lines used by the CPU to gain diagnostic control of a channel or of a Channel Controller and the 'fault locating test' (FLT) lines required to control a channel when a CPU is performing fault-locating tests.

When two CPUs request access to the channel interface at the same time, CPU 1 has priority.

During the selection sequence, the parity of the channel address and of the unit address is checked in the Channel Controller. If a parity error exists for the channel address, the Channel Controller stacks the check status, returns a condition code 3 and releases the CPU, thereby terminating the selection sequence. If a parity error is detected in the unit address, the Channel Controller stacks the check status and continues with the selection sequence.

If parity of both addresses is good, the Channel Controller determines which channel is requested by the CPU. The available line from that channel is then checked. If the channel is not available, the Channel Controller returns condition code 3 and releases the selecting CPU. If the channel is available, the selection sequence is completed. The Channel Controller then waits for the selected CPU and the selected channel to finish the signaling sequence and keeps the other CPU logically disconnected until this sequence is completed. At completion, the Channel Controller disconnects the CPU and the channel and resumes polling both CPUs for another access request.

Channel Interruption

The Channel Controller receives, from the attached CPUs, the system mask bits associated with the attached channels (i.e., Channel Controller 0 receives mask bits 0–6 from control register 4, whereas Channel Controller 1 receives mask bits 8–14). Since each channel can be addressed by, and can interrupt, two CPUs, as many as two mask bits are received for each channel. Therefore, each I/O interrupt received by the Channel Controller from each attached channels can be gated to two CPUs. When a CPU has been interrupted, the Channel Controller provides that CPU with the address of the interrupting device and channel.

The CPU places the device address and the channel address into its PSW and stores it at the I/O PSW location. The Channel Controller then initiates another poll for CPU accesses to the channels.

Error Detection

The Channel Controller monitors the CPU, channel, and storage interfaces, and its internal circuitry for error conditions. When an error is detected, the Channel Controller causes an external machine check interrupt (if not masked off) at the CPU that initiated the operation during which the error is detected. When the interrupted CPU responds to the interrupt request, the error information is stored in the old machine-check PSW.

CHANNELS

Channels provide the data paths and direct control for I/O control units and for the I/O devices attached to the control units. Channels relieve the CPU of the task of communicating directly with the I/O devices and permit data processing to proceed concurrently with I/O operations.

Data is transferred one byte at a time between an I/O device and a channel. Data transfers between a channel and the storage control unit (Model 67-1) or between a channel, a Channel Controller, and a storage control unit (Model 67-2) are parallel (by eight bytes) for both selector and multiplexer channels (Figure 4).

A standard I/O interface provides a uniform method of attaching I/O control units to all channels, making the Model 67 adaptable to a broad range of applications.

The 2860 Selector Channel and the 2870 Multiplexer Channel are available for the Model 67. A maximum of seven channels can be attached to a 2067-1: (1) one 2870 and two 2860-3's; or (2) two 2870's, one 2860-2, and one 2860-3. At least one 2860 (any model) or one 2870 is required.

Up to fourteen channels can be attached in a Model 67-2 system: seven channels (including one 2870) to each 2846 Channel Controller.

2860 Selector Channel

The 2860 Selector Channel provides for the attachment and control of Burst mode I/O control units and associated devices. The 2860 is available in three models:

1. Model 1 provides one selector channel.
2. Model 2 provides two selector channels.
3. Model 3 provides three selector channels.

The selector channel permits data rates of up to 1.3 million bytes per second. I/O operations are overlapped with processing, and, depending on the data rate, all selector channels can operate concurrently. A full set of channel control and buffer registers permits each channel to operate with minimal interference.

A maximum of eight control units can be attached to each selector channel. However, each channel can perform data transfer operations with only one I/O device at any given time.

Channel-to-Channel Adapter Feature

A channel-to-channel adapter is available as an optional feature. This adapter permits communication between two System/360 channels and, thus, provides the capability of interconnecting two processing units within the System/360. The adapter uses one control unit position on each of the two interconnected channels. Only one of the two interconnected channels requires the feature. There can be a maximum of one channel-to-channel adapter per selector channel.

2870 Multiplexer Channel

The 2870 Multiplexer Channel provides for the attachment of a wide range of low- and medium-speed I/O control units and associated devices. Two 2870 Multiplexer Channels can be attached to the Model 67-1; one 2870 can be attached to each 2846 Channel Controller in a Model 67-2 system.

The first 2870 in a Model 67-1 system or each 2870 in a Model 67-2 system provides up to 196 subchannels, including four selector subchannels. The second 2870 in a Model 67-1 system provides up to 194 subchannels, including two selector subchannels. All selector subchannels are optional.

Each basic multiplexer channel has 192 subchannels; it can attach eight control units and can address 192 I/O devices. The basic multiplexer channel can overlap the operation of several I/O devices in Multiplex mode or operate a single device in Burst mode. Each selector subchannel can operate one I/O device concurrently with the basic multiplexer channel. Each selector subchannel permits the attachment of eight control units for devices having a data rate not exceeding 180 kb. Regardless of the number of control units attached, a maximum of 16 I/O devices can be addressed by a selector subchannel.

The maximum aggregate data rate for a multiplexer channel ranges from 110 kb to 670 kb, depending on the number of selector subchannels installed. Selector subchannels 1-3 may each operate concurrently at up to 180 kb; selector subchannel 4 has a maximum data rate of 100 kb. Each selector subchannel in operation diminishes the basic multiplexer channel's maximum data rate of 110 kb; the relationship to maximum data rates for concurrent selector subchannel operations is shown in the following table:

Basic Multiplexer Channel	Selector Subchannels 1st	2nd	3rd	4th	Aggregate Data Rate
110 kb					110 kb
88 kb	180 kb				268 kb
66 kb	180 kb	180 kb			426 kb
44 kb	180 kb	180 kb	180 kb		584 kb
30 kb	180 kb	180 kb	180 kb	100 kb	670 kb

Note: The 180-kb maximum data rate for selector subchannels pertains to attachment of magnetic tape devices; timing factors other than data rates may preclude attachment of direct access storage devices having lesser data rates. The data rates are theoretical maximums; the actual rates may be somewhat less, depending on program usage and on the number of channels operating in the system.

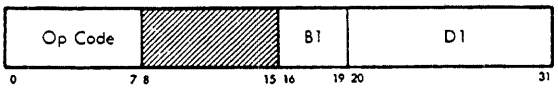
The 2870 may be connected to another system channel for channel-to-channel interconnection of two System/360 channels. The channel-to-channel adapter, however, is installed on the other channel, not on the 2870.

Channel Addresses and Priority

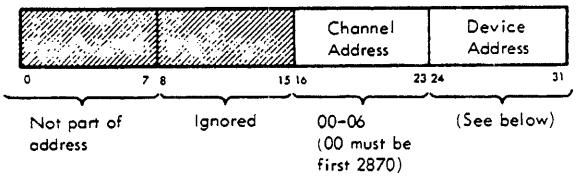
Each channel in a Model 67-1 system, or each channel attached to a Channel Controller in a Model 67-2 system, is assigned an address in the range of 0 to 6, as follows: the first 2870 attached must be assigned address 0; the second 2870 (Model 67-1 system only) and each 2860 Selector Channel are assigned a unique address in the range of 1 through 6, depending on the desired priority and on the number of channels in the system. Consecutive addresses must be assigned; for example, when four channels are

attached, (1) addresses 0, 1, 2, and 3 must be assigned if a 2870 is one of the channels, or (2) addresses 1, 2, 3, and 4 must be assigned if all four are selector channels.

All I/O instructions use the following SI format:



Bit positions 8–15 of the instruction are ignored. The contents of the B_1 field designates a register. The sum obtained by the addition of the content of register B_1 and content of the D_1 field identifies the channel and the I/O device. This sum has the format:



The range of device addresses is as follows:

1. If 2860: 00-FF (0 to 255 addresses per selector channel).
2. If 2870:
   00-BF: Basic multiplexer channel (192 addresses max)
   C0-CF: Selector subchannel 1
   D0-DF: Selector subchannel 2
   E0-EF: Selector subchannel 3
   F0-FF: Selector subchannel 4
   (16 addresses max per selector subchannel)

To allow the storage control unit or the Channel Controller to determine the action to be taken when simultaneous requests for service are made by the channels, each channel is assigned a priority. This priority is determined by the channel address assignments and on the System model, as follows:

Channel Priority	1	2	3	4	5	6	7
Channel Address (67-1)	1	2	0	3	4	5	6
Channel Address (67-2)	1	2	3	4	5	6	0

In a Model 67-1 system, when simultaneous requests for service are made, the storage control unit services channels in priority sequence; the processing unit, which has lowest priority, is serviced only when none of the channels require service. In a Model 67-2 system, the channel controller services the channels in priority sequence; then, the storage control unit services the channel controllers in address sequence (Channel Controller 0 first) and services the processing unit when neither Channel Controller requires service. The priority of a channel is a factor in determining the maximum data rate that can be sustained by that channel.

System Data Rates

The preceding 2860 and 2870 descriptions give the maximum data rates that can be achieved by the 2860 and 2870. During systems operation, the actual rates may be less than the maximums, depending on (1) channel priority, (2) the number of channels operating concurrently, (3) the speed of the devices operating on each channel, and (4) the type of programming used. If transfer-in-channel type programming (command-chaining or data-chaining) is used, the data rates must be reduced if a sufficient number of additional storage references are required.

The following examples show representative system data rates that can be achieved assuming (1) command-chaining is used, (2) Transfer in Channel commands are used, and (3) data-chaining is not used. If data-chaining or program controlled interrupts are used, the data rates must be lower. In the examples, all channels in the system are operating concurrently. The I/O units used are the IBM 2301 Drum Storage, the IBM 2314 Direct Access Storage Facility: "A" Series, and the IBM 2420 Magnetic Tape Unit Model 7. All data rates are in kilobytes (kb). Examples 1 through 3 apply to Model 67–1, whereas Example 4 applies to a Model 67–2, half-duplex system and Example 5 applies to a Model 67–2 duplex system.

Example Configuration 1 (Model 67-1 only):

Channel Address	Model	Attached I/O Units	Data Rates
1	2860	2301	1250
2	2870(2nd)	Basic mpxr chnl	66
		Two selector subchnls (180 kb each)	360
0	2870(1st)	Basic mpxr chnl	30
		Three selector subchnls (180 kb each)	540
		One selector subchnl	100
3-5	2860's	2314's (312 kb each)	936
6	2860	2420-7	320

Example Configuration 2 (Model 67-1 only)

Channel Address	Model	Attached I/O Units	Data Rates
1	2860	2301	1250
2	2860	2301	1250
0	2870(1st)	Basic mpxr chnl	30
		Three selector subchnls (180 kb each)	540
		One selector subchnl	100
3-4	2860's	2314's (312 kb each)	624
5	2860	2420-7	320
6*	2870(2nd)	Basic mpxr chnl	50
		Two selector subchnls (180 kb each)	360

*These rates also apply if the 2nd 2870 is at address 3, 4, or 5.

Example Configuration 3 (Model 67-1 only)

Channel Address	Model	Attached I/O Units
1	2860	2301
2	2860 or 2870 (2nd)	2301, 2314, or 2420-7
0	2870(1st)	Any
3	2860 or 2870 (2nd)	2314 or 2420-7
4		2314 or 2420-7
5		2314 or 2420-7
6		2314 or 2420-7

The data rates for the first 2870 (at address 0) are:

Basic Mpxr Chnl	Selector Subchannel 1	2	3	4	Aggregate Data Rates
110					110
80	180	-	-	-	260
60	180	180	-	-	420
44	180	180	180	-	584
30	180	180	180	100	670

The following data rates are for a second 2870 if at address 2:

Basic Mpxr Chnl	Selector Subchannel 1	2	3*	4*	Aggregate Data Rates
110	-	-	-	-	110
85	180	-	-	-	265
65	180	180	-	-	425

*The second 2870 can have up to two selector subchannels.

The following data rates are for a second 2870 if at address 3, 4, 5, or 6:

Basic Mpxr Chnl	Selector Subchannel 1	2	3*	4*	Aggregate Data Rates
110	-	-	-	-	110
70	180	-	-	-	150
50	180	180	-	-	410

*The second 2870 can have up to two selector subchannels.

Example Configuration 4 (Model 67-2 Half-Duplex):

Channel Address	Model	Attached I/O Units	Data Rates
1	2860	2301	1250
3-5	2860's	2314's (312 kb each)	936
6	2860	2420-7*	320
0	2870	Basic mpxr chnl	30
		Three selector subchnls (180 kb each)	540
		One selector subchnl	100

*2420-7's can be substituted for 2314's, and vice versa, with the data rates modified accordingly.

Example Configuration 5 (Model 67-2 Duplex):

Channel Address	Model	Attached I/O Units	Data Rates*
1	2860	2301	1250
2-4	2860's	2314's (312 kb each)**	936
5	2860	2420-7**	320
6	-	Spare	-
0	2870(1st)	Basic mpxr chnl	50
		Two selector subchnls (180 kb each)	360

*The total system burst data rate is 5,832kb per second (the example configuration is attached to each Channel Controller); this approaches the maximum data rate for a full-duplex system with all channels operating simultaneously.

**2420-7's can be substituted for 2314's, and vice versa, with the data rates modified accordingly.

IBM 2365 PROCESSOR STORAGE

Model 67-1 systems use IBM 2365 Model 2 Processor Storage Units. Model 67-2 systems use 2365 Model 12 Processor Storage Units. Up to four 2365-2s and eight 2365-12s can be used in the appropriate system configurations.

Both models of the 2365 have a basic 750-ns storage cycle and access eight bytes in parallel (a doubleword). A store function is possible on a byte basis, and any combination up to eight contiguous bytes can be stored in one storage cycle. Byte locations are consecutively numbered, starting with 0.

Each 2365 contains two independent storage arrays, each with its own address and storage buffer registers. Each array can access a doubleword and has 131,072 bytes of storage organized into 16,384 doublewords. All doublewords with odd-numbered addresses are in one array; doublewords with even addresses are in the other array.

Four independent storage buses can be attached to the 2365-12: one from each CPU and Channel Controller in a full duplex system. The unit consists of two logical elements, called the Storage Switch Unit (SSU) and the storage element.

The SSU contains appropriate priority circuitry, bus control circuits (partitioning), and a CE test and control panel. The SSU has independent power regulators and derives primary power from an independent customer-supplied power source.

The storage element contains the operational memory (two 128 K-byte arrays), common logic, and independent power regulators. It derives its primary power from an independent customer-supplied power source.

Priority

The SSU determines priority for simultaneous storage requests as follows. The general rule is that Channel Controllers have priority over CPUs. At installation time,

the relative priority between the two Channel Controllers and the relative priority between the two CPUs is established at each 2365-12. The Channel Controller that has first priority at one storage unit will not have it at every storage unit. The CPU priority also changes from storage unit to storage unit.

At any given storage unit, the Channel Controller given first priority is the Channel Controller closest to the storage unit. The same is true of CPU relative priority. The Channel Controller priority also applies to its priority select signal. This arrangement minimizes the waiting time accumulated during simultaneous selections and improves overall system throughput.

Another determinate in the establishment of priority for simultaneous selections is the odd/even criteria. The odd/even arrangement of doublewords into separate arrays permits overlapping of successive storage accesses. Each CPU and Channel Controller provides two selection lines to the SSU for odd and even selection of doublewords. To overlap storage accesses, a storage cycle on the array containing doublewords with odd-numbered addresses is overlapped with the highest-priority selection for access to the even array. For example, assume that (1) Channel Controller 0 is using the odd array, (2) Channel Controller 1 selects odd, and (3) CPU-1 selects even; the storage cycle for CPU-1 will be overlapped with the last half of the access by Channel Controller 0. When Channel Controller 0 finishes with the odd array, Channel Controller 1 is given access.

The SSU, upon receipt of a priority select signal, upgrades pending requests from that Channel Controller to the highest level of priority.

Access Time

The physical location of storage units in respect to CPUs contributes to variable storage access times. Access time for a doubleword from an adjacent storage unit is 750 ns. This access time assumes no interference from other users of the storage unit. In duplex systems with multiple units accessing processor storage, the access time for a given cycle depends on cable distance between the user and the storage element and on the degress of interference caused by simultaneous requests to storage from other users. In duplex systems, storage cycles may be interleaved efficiently with alternate requests from the combined set of CPUs and Channel Controllers. The minimum interleaved access time possible is 375 ns.

Storage Protection

The 2365-12 Processor Storage Unit contains a 128-key storage protect array. Each key is composed of seven bits plus odd parity and protects 2048 processor storage locations.

Each protection key is established by a Set Key instruction. When protection applies to a storage reference, the protection key is transmitted to storage; there, the contents of bits 1 through 4 are compared with the contents of the appropriate storage protect key.

Bits 0–3 are the standard four-bit protection keys. Bit 4 is the fetch-protection bit, bit 5 is the reference bit, and bit 6 is the change bit. The four-bit protection keys in the PSW and CAW remain unchanged in length. The reference bit (bit 5) is set to 1 each time the corresponding storage block is accessed for storing or fetching by a CPU or Channel Controller. The change bit (bit 6) is set to 1 each time data is stored in the corresponding storage block by a CPU or Channel Controller.

Reference and change recording is always active. It is independent of (1) the supervisor or masked state of the CPU, (2) the type of instruction or I/O command being executed, and (3) the manner in which the address is generated. Hence, references for updating or interruption purposes such as the timer, channel status word (CSW), or PSW locations are included in the reference and change recording.

The reference and change bits are updated on all store or fetch references made to processor storage, even if one of the following conditions should occur:

1. Address parity check
2. In key parity
3. Out key parity check
4. Key mismatch
5. Cancel instruction

In a 67-1 system, the seven-bit storage protect feature is required on each 2365-2 to provide the additional two bits for reference and change recording. This feature is not required on the 2365-12s in a Model 67-2 system because seven-bit storage protect is provided.

Partitioning

The Configuration Unit provides a remote location for the 2365-12 partitioning switches. Each CPU and Channel Controller interface at the SSU is controlled by a unique switch. If communication between the SSU and a given CPU or Channel Controller is to be prevented, the selection signals are blocked at the SSU. The available line to the specific user is also disabled, making the storage element unavailable to that specific user.

IBM 2167 CONFIGURATION UNIT

The 2167 is a free-standing console which provides a central location for the switches required for partitioning and miscellaneous control functions. It is required in a Model 67-2. Manual switches for the following functions are located on the 2167:

1. Partitioning
2. Prefix activation
3. Direct control activation
4. Floating addressing–Processor Storage Units
5. Floating addressing–Channel Controllers

The 2167 is available in four models:

2167 Model	Attachment Capability of 2067 Model			
	2067 CPU	2365 Storage	2846 CCU	I/O CU*
1	2	2	2	16
2	2	3	2	16
3	2	4	2	16
4	2	8	2	32

*I/O CU lists the number of dual-channel interface I/O control units that can be partitioned by switches on the 2167; i.e., 2167 Models 1, 2, and 3 have 16 pairs of partitioning switches for I/O control units, and 2167 Model 4 has 32 pairs of I/O control unit partitioning switches. This is not necessarily a limitation on the number of I/O control units possible in a given configuration.

Partitioning Switches

A manual partitioning switch is provided on the 2167 for each partitionable interface in the system. One switch is provided for each interface on a core storage unit, each CPU interface to a Channel Controller, and each channel interface on dual-channel-interface I/O control units. Partitioning switches are located in the form of a matrix on the panel. The rows and columns of the matrix are identified with units. The switch controlling a given interface is located at the intersection of the column and row identified with the two units sharing the partitionable interface (Figure 7).

The configuration status of all partitionable units is indicated by the contents of control registers 8 through 14. When the system has been placed in a reconfigurable state (bit 9, control register 6), partitioning switches can be used to change the configuration status.

Prefix Activation

One switch to deactivate prefixing is provided per CPU.

Direct Control Activation

One switch per CPU is provided to deactivate the CPU extended direct control interface; WRITE DIRECT is not deactivated.

Floating Addressing (2365-12)

A rotary switch is provided for each 2365-12 in the system. The setting of this switch indicates the starting address of the address interval contained in that unit. (A starting address of 0 for the interval from 0 to 262, 143; a starting address of 262, 144 for the interval from 262, 144 to 524, 287; etc.). These switches are not interlocked (to prevent assignment of the same starting address to more than one 2365-12).

Floating Addressing (Channel)

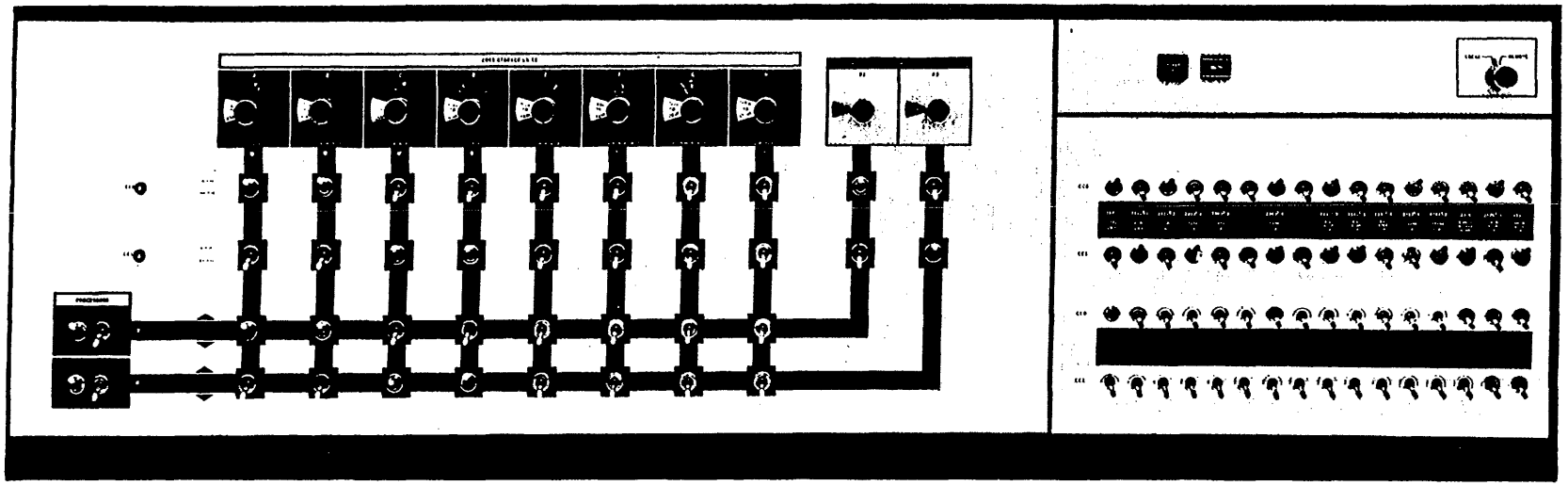
One rotary switch is provided for each CPU in the system. A switch can assign to the CPU with which it is associated either of two Channel Controllers (if Channel Controllers are in the system) for standard PSW mode operation.

Partitioning Sensing

The capability for sensing the status of the manual switches by each CPU is provided in the Configuration Unit. The switch status is assembled into 32-bit words, which are accessible to the Store Multiple Control instruction. A 1-bit indicates that the switch is such that the corresponding interface is connected. A 0-bit indicates that the interface is disconnected. Parity is provided for each 8-bit byte of the 32-bit word. Each CPU senses the partitioning switches independently.

Control Register Bit Assignments for Sensing

Of the 16 control registers that can be accessed by the Load/Store Multiple Control instructions, registers 8 through 14 reflect, by bit, the manual switch settings. These registers can be stored into processor storage but cannot be loaded from processor storage. Table 5 identifies each register and the significance of each bit.



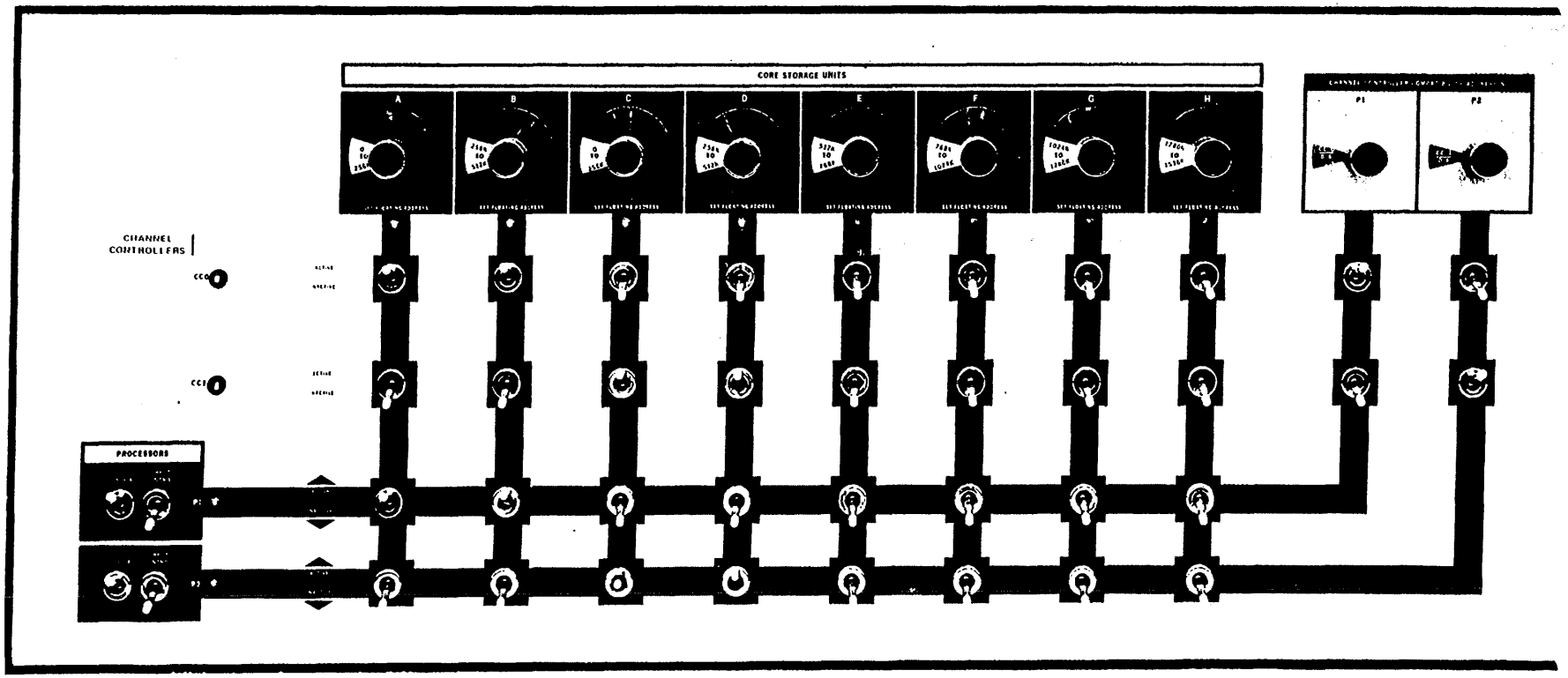


Figure 7. Configuration Unit Control Panel (Sheet 1 of 2)

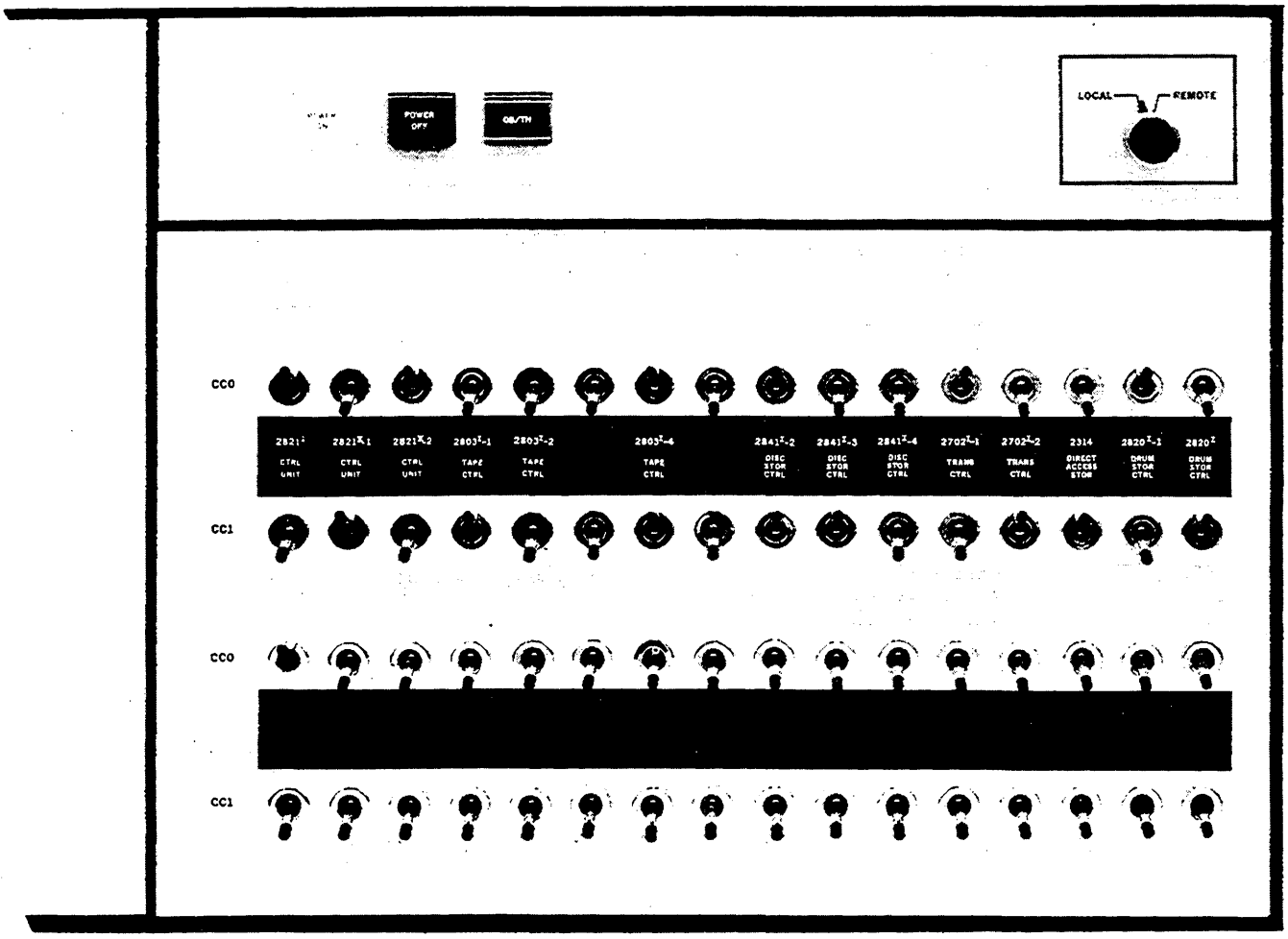


Figure 7. Configuration Unit Control Panel (Sheet 2 of 2)

Table 5. Control Register Bit Assignments

1. Control Register 8:

Bit Position*	Active Interface (Bit = 1)
0	Processor Storage Unit 1 to CPU 1
1	Processor Storage Unit 1 to CPU 2
2, 3	Unassigned
4	Processor Storage Unit 1 to CC 0
5	Processor Storage Unit 1 to CC 1
6, 7	Unassigned
8	Processor Storage Unit 2 to CPU 1
9	Processor Storage Unit 2 to CPU 2
10, 11	Unassigned
12	Processor Storage Unit 2 to CC 0
13	Processor Storage Unit 2 to CC 1
14, 15	Unassigned
16	Processor Storage Unit 3 to CPU 1
17	Processor Storage Unit 3 to CPU 2
18, 19	Unassigned
20	Processor Storage Unit 3 to CC 0
21	Processor Storage Unit 3 to CC 1
22, 23	Unassigned
24	Processor Storage Unit 4 to CPU 1
25	Processor Storage Unit 4 to CPU 2
26, 27	Unassigned
28	Processor Storage Unit 4 to CC 0
29	Processor Storage Unit 4 to CC 1
30, 31	Unassigned

*Bit equal to 1 for active interface.

2. Control Register 9:

Bit Position*	Active Interface
0	Processor Storage Unit 5 to CPU 1
1	Processor Storage Unit 5 to CPU 2
2, 3	Unassigned
4	Processor Storage Unit 5 to CC 0
5	Processor Storage Unit 5 to CC 1
6, 7	Unassigned
8	Processor Storage Unit 6 to CPU 1
9	Processor Storage Unit 6 to CPU 2
10, 11	Unassigned
12	Processor Storage Unit 6 to CC 0
13	Processor Storage Unit 6 to CC 1
14, 15	Unassigned
16	Processor Storage Unit 7 to CPU 1
17	Processor Storage Unit 7 to CPU 2
18, 19	Unassigned
20	Processor Storage Unit 7 to CC 0
21	Processor Storage Unit 7 to CC 1
22, 23	Unassigned
24	Processor Storage Unit 8 to CPU 1
25	Processor Storage Unit 8 to CPU 2
26, 27	Unassigned
28	Processor Storage Unit 8 to CC 0
29	Processor Storage Unit 8 to CC 1
30, 31	Unassigned

*Bit equal to 1 for active interface.

3. Control Register 10: Processor storage address assignment codes.

a. Assignments:

Bit Positions	Starting Address Code for:
0–3	Processor Storage Unit 1
4–7	Processor Storage Unit 2
8–11	Processor Storage Unit 3
12–15	Processor Storage Unit 4
16–19	Processor Storage Unit 5
20–23	Processor Storage Unit 6
24–27	Processor Storage Unit 7
28–31	Processor Storage Unit 8

b. Possible bit-code combinations for each 4-bit group:

Rotary Switch Setting	Bit Codes	Set Floating Address Switch
1	0000	0 to 256K
2	0010	256K to 512K
3	0100	512K to 768K
4	0110	768K to 1024K
5	1000	1024K to 1280K
6	1010	1280K to 1536K
7	1100	1536K to 1792K
8	1110	1792K to 2048K

Note: The 4-bit code stored by STORE MULTIPLE CONTROL represents address bits 11–14; i.e., bit 13 is on an address multiple of 256K bytes; thus, the 4-bit code represents the starting address of each unit.

4. Control Register 11:

Bit Position*	Active Interface	
0	CC 0 to CPU 1	
1	CC 0 to CPU 2	
2, 3	Unassigned	
4	CC 1 to CPU 1	
5	CC 1 to CPU 2	
6–15	Unassigned	
16	CPU 1 to only CC 0	See Note
17	CPU 1 to only CC 1	
18, 19	Unassigned	
20	CPU 2 to only CC 0	
21	CPU 2 to only CC 1	
22–31	Unassigned	

*Bit equal to 1 for active interface.

Note: Bit positions 16–31 set by channel controller compatibility addressing switches; bit positions 0–15 set by partitioning switches.

Table 5. Control Register Bit Assignments (Cont)

5. Control Register 12 (I/O Control Unit Partitioning Sensing):

Bit*	I/O Control Unit	Bit*	I/O Control Unit
0,1	1	16, 17	9
2, 3	2	18, 19	10
4, 5	3	20, 21	11
6, 7	4	22, 23	12
8, 9	5	24, 25	13
10, 11	6	26, 27	14
12, 13	7	28, 29	15
14, 15	8	30, 31	16

*Bit equal to 1 for active interface.

Note: For each control unit, the even-numbered bit defines the status of interface 1, the odd bit defines interface 2.

6. Control Register 13:

Bit*	I/O Control Unit	Bit*	I/O Control Unit
0, 1	17	16, 17	25
2, 3	18	18, 19	26
4, 5	19	20, 21	27
6, 7	20	22, 23	28
8, 9	21	24, 25	29
10, 11	22	26, 27	30
12, 13	23	28, 29	31
14, 15	24	30, 31	32

*Bit equal to 1 for active interface.

Note: For each control unit, the even-numbered bit defines the status of interface 1, the odd bit defines interface 2.

7. Control Register 14:

Bit Position*	Indication
0–21	Unassigned
22	2167 Power On
23	Unassigned
24	Direct Control, CPU 1
25	Direct Control, CPU 2
26, 27	Unassigned
28	Prefix, CPU 1
29	Prefix, CPU 2
30, 31	Unassigned

*Bit equal to 1 for active indication.

SYSTEM CONFIGURATIONS

Three basic configurations are available: the Model 67-1 for simplex applications, and the Model 67-2 either for half-duplex or duplex applications.

Figure 8 shows the minimum simplex system configuration. The minimum configuration is represented by solid blocks on the figure. Note that a 2067-1 Processing Unit and 2365-2 Processor Storage Units are used. The Model 67-1 configuration is not expandable to a half-duplex or duplex system. However, the minimum configuration can be expanded as shown by the dashed blocks on the figure. Additional I/O control units and devices are not shown. However, each selector channel can attach up to eight I/O control units and address up to 256 I/O devices. Each basic multiplexer channel can attach up to eight I/O control unit and address up to 192 I/O devices, and each selector subchannel can attach up to eight I/O control units and address up to 16 I/O devices. TSS/360 supports the 2870 selector subchannel for magnetic tape only. A single magnetic tape control unit can recognize all 16 device addresses, thereby excluding attachment of other control units to the selector subchannel for time sharing systems.

The minimum Model 67-2 half-duplex system is shown on Figure 9. Note that the 2067-2 Processing Unit, 2846 Channel Controllers, 2365-12 Processor Storage Units, and 2167 Configuration Units are used in Model 67-2 configurations. The half-duplex configuration can be expanded into a duplex system (Figure 10) by adding a second 2067-2 Processing Unit and a third 2365-12 Processor Storage Unit.

The minimum configurations reflect the requirements of the program system (TSS/360). Each installation must expand this to fit the requirements of its facility.

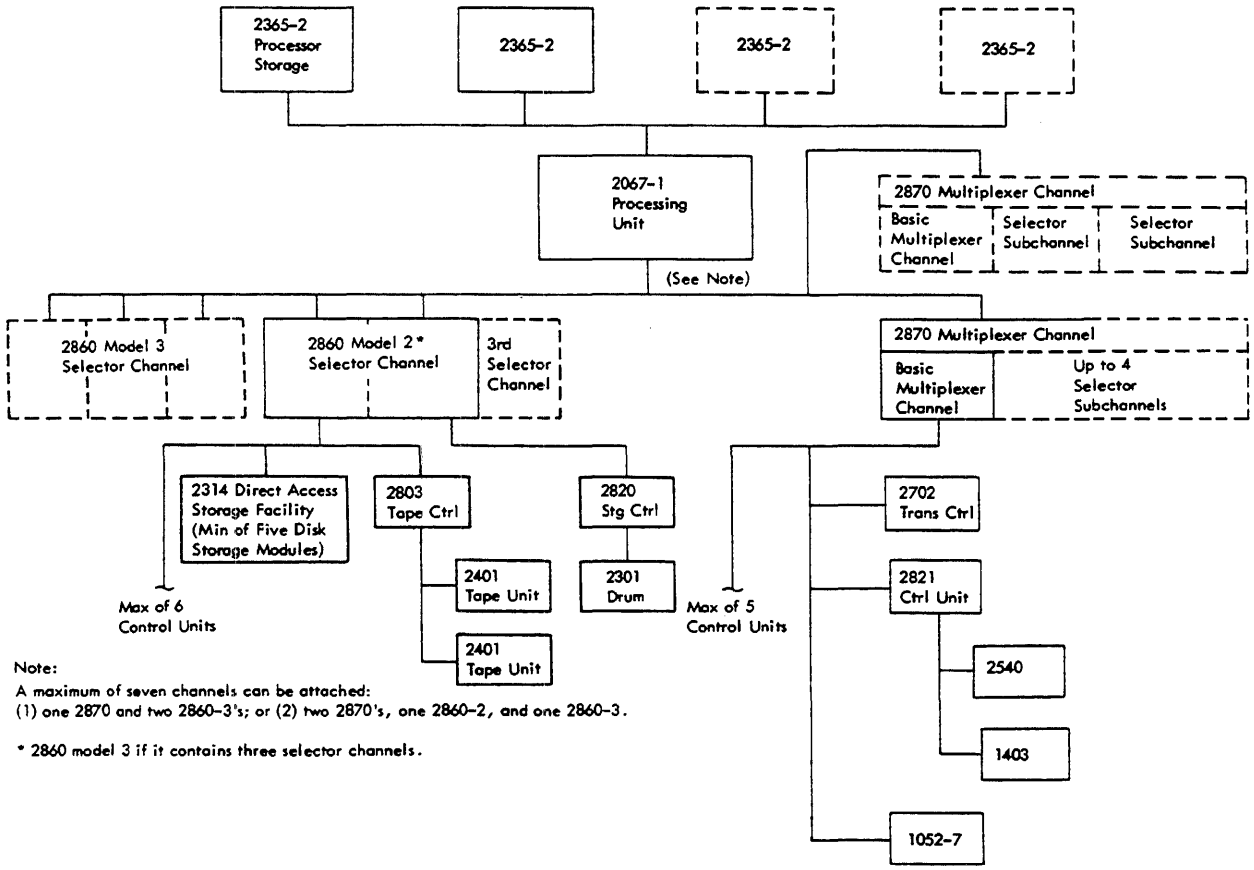


Figure 8. Model 67-1 Minimum Configuration

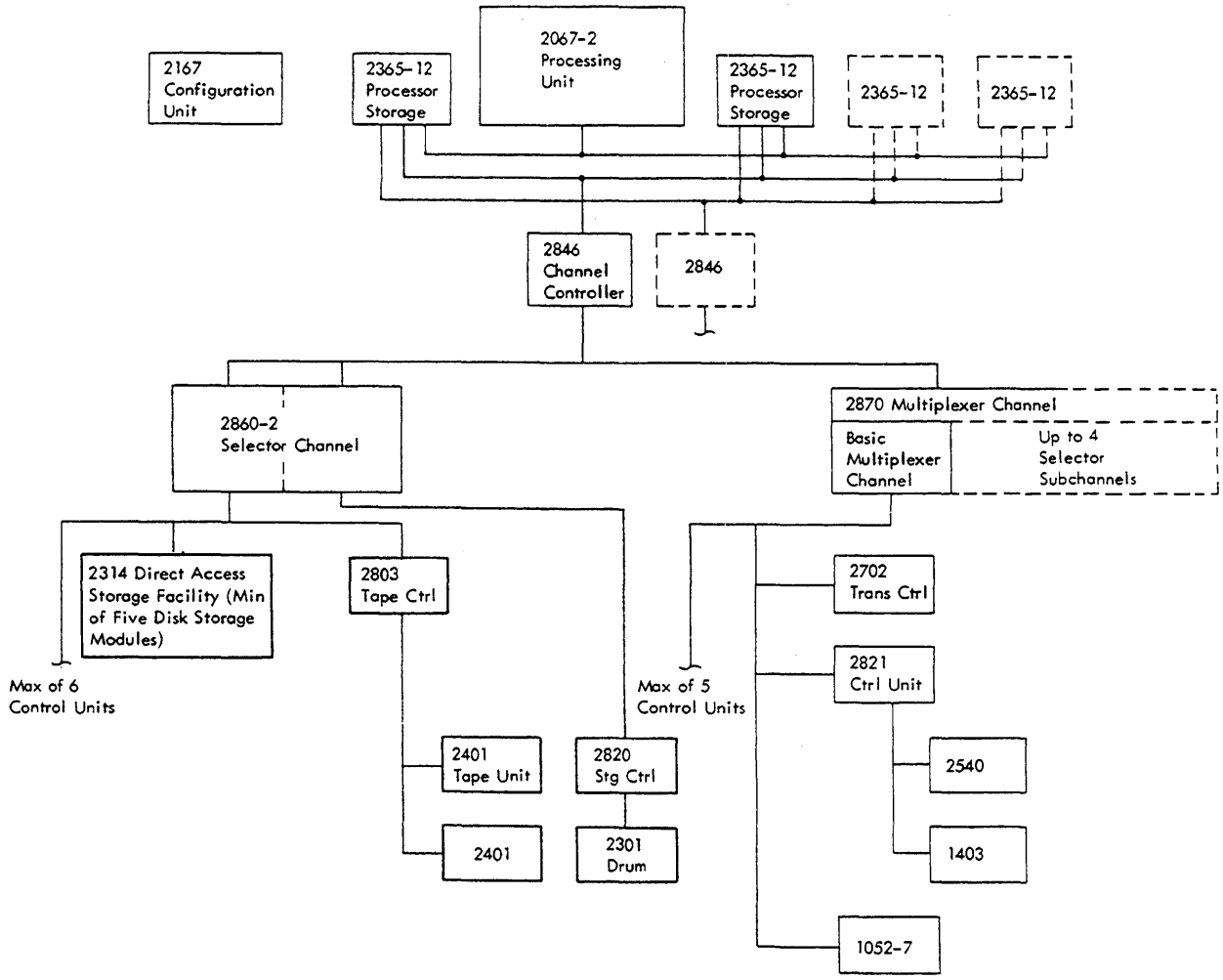


Figure 9. Model 67-2 Half-Duplex System, Minimum Configuration

SYSTEM DATA FLOW

Data is transmitted throughout the system along paths having number, width, and data-rate capabilities designed for efficient information flow and minimum interference. In general, paths are widest where data rates are highest. The number of paths between major elements of the system is a function of the requirement both for simultaneous operation and for availability.

Data is transmitted eight bytes at a time between processor storage, the CPUs, and the channels and is transmitted one byte at a time between the channels and I/O control units. Parity checking is performed throughout the system at the byte level.

Figures 11 and 12 are data flow diagrams of a Model 67-1 and a Model 67-2 system respectively. A major difference between these systems is the storage BCU function.

In a Model 67-2 system, each CPU and each Channel Controller is connected, with separate bus, to each storage unit in the system. For example, a system with two CPUs and two Channel Controllers has four buses. Each bus can connect to as many as eight storage units. Conflicts that occur among the several buses connected to each storage unit are resolved at the storage unit. Channel Controller requests for storage cycles are given priority over CPU requests. Each CPU has a BCU that controls the storage bus

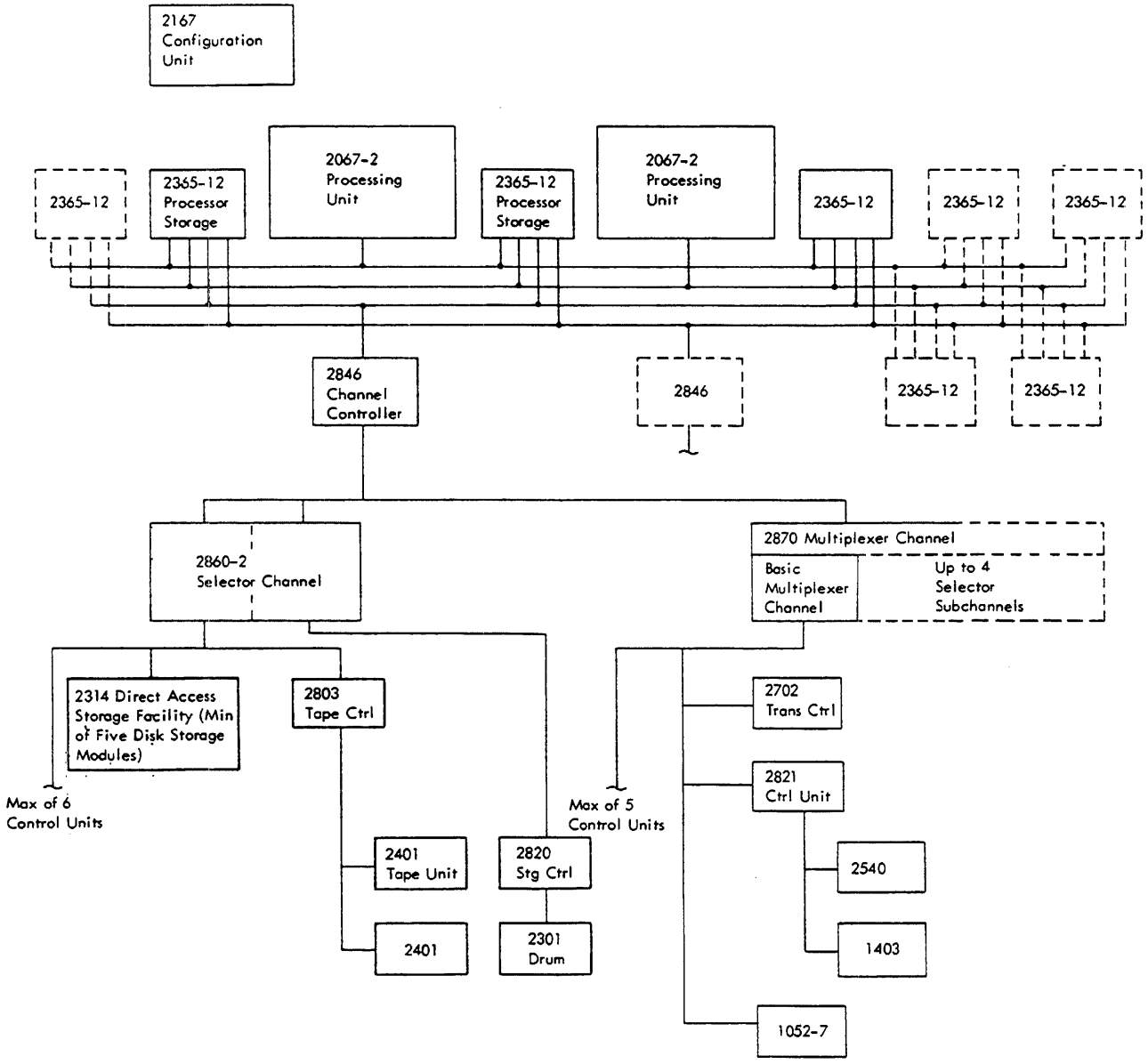


Figure 10. Model 67-2 Duplex System, Minimum Configuration

with which the CPU is associated. The Channel Controller performs two types of tasks in the Model 67-2 data transfer function: it communicates with the storage units, and it serves the requests from its channels. In a Model 67-1 system, both tasks are performed by the CPU BCU.

The channel data path provides for the transfer of data between the Channel Controller or CPU BCU and the attached channels. This path is eight bytes wide, including the associated parity bits. All data transfer between the BCU and the attached channels is accomplished over a single path; time-sharing of the path is controlled by the BCU. The definition of signals on this path (except signals required for establishing priority among channels) is the same as for signals on the storage bus.

All communication between channels and I/O control units is via I/O interface data buses (one in each direction), which are one byte wide. The connection between channels and control units is standardized. Further information is given in *IBM System/360 Interface-Channel to Control-Unit Original Equipment Manufacturers Information,* Form GA22-6843.

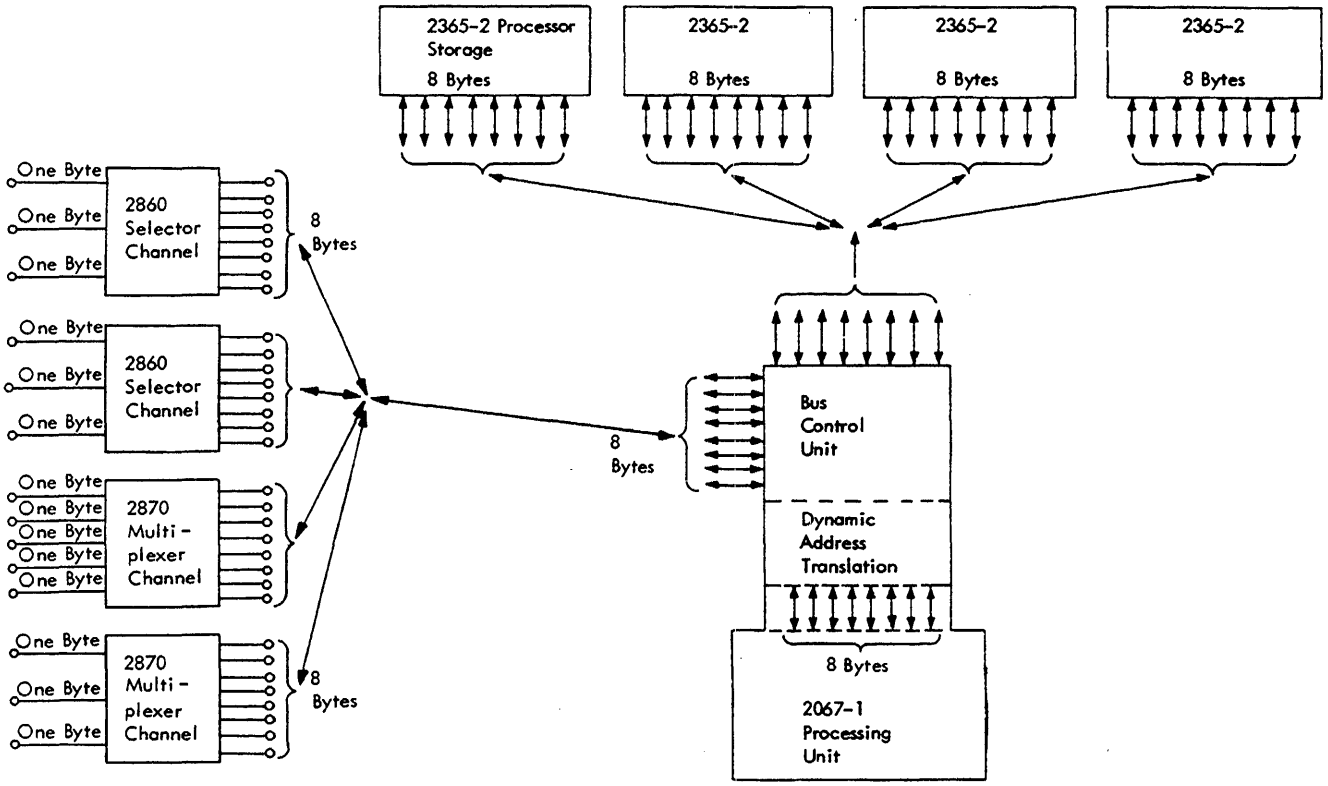


Figure 11. Model 67-1 Data Flow Diagram

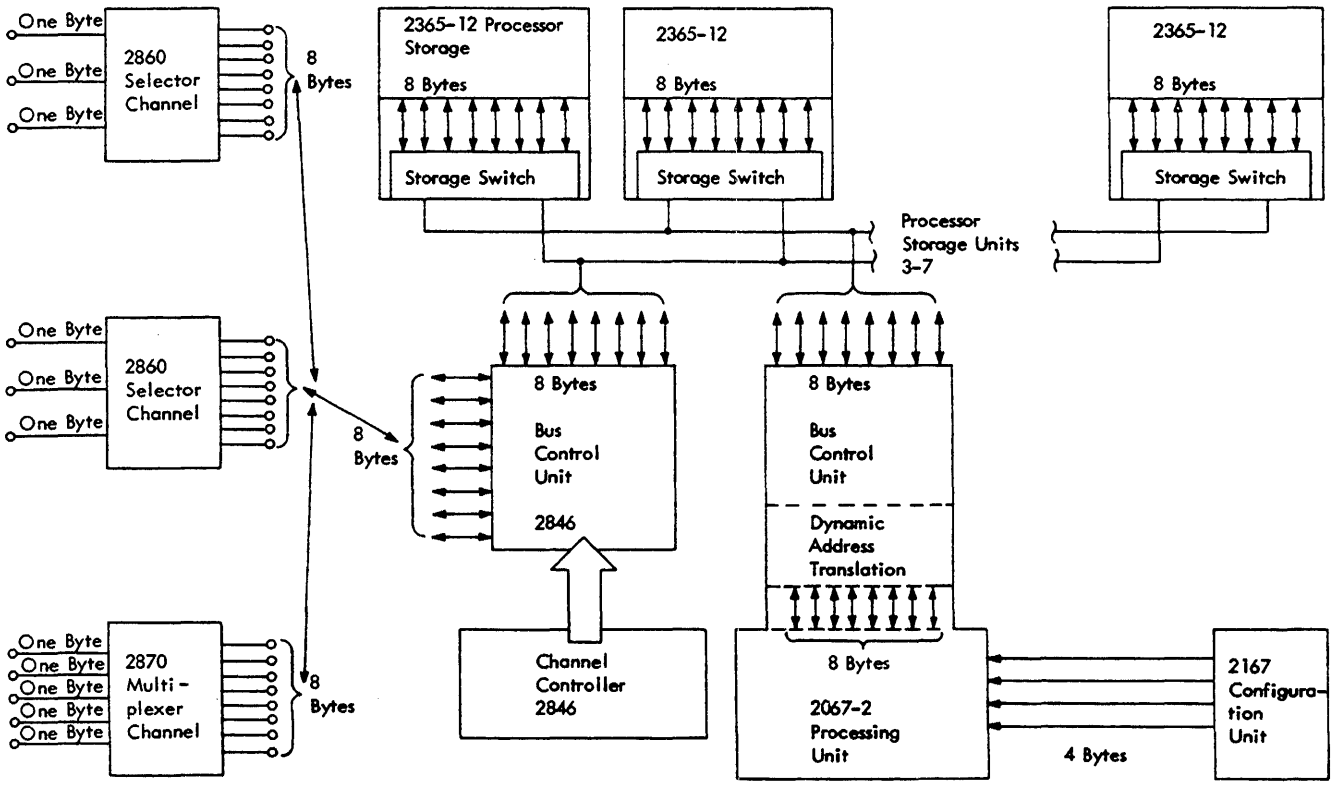


Figure 12. Model 67-2 Data Flow Diagram

The system control panel (Figure 13) contains the switches and indicators necessary to operate and control the system. Switches and indicators are also provided for operator intervention and for customer engineering operations.

The operator control sections (sections G and C) are located in the lower and upper-right portions of the system control panel. For monitor control operations, an identical control section can be provided on an IBM 2150 Operator's Console. Operator control sections are the same throughout the System/360 line, providing operations with compatibility between machines. They are described in the *IBM System/360 Principles of Operation* manual.

Operator intervention controls and indicators are located in sections E and F of the system control panel. These are described both in the *IBM System/360 Principles of Operation* manual and in the following paragraphs. Appendix A is a summary of all system control panel controls and indicators.

The customer engineer will use all controls and indicators; however, many of the controls and indicators located in sections A, B, E, and F are intended primarily for customer engineering use, and are not functionally described in this publication.

By the use of the control panel, the operator can perform these important system functions:

1. Reset the system.
2. Store and display information in storage, in registers, and in the PSW.
3. Load initial program information.

These functions are described in the *IBM System/360 Principles of Operation* manual.

OPERATOR INTERVENTION CONTROLS

Sections F and E of the system control panel contain the controls required for the operator to intervene in normal programmed system operations. These controls are intermixed with the customer engineering controls. Only operator intervention controls, on these panels, are described in this section.

Operator intervention controls provide the system reset and the store and display functions.

START Pushbutton

This pushbutton provides a means of starting the CPU in the Process, Instruction Step, Single Cycle, or Single Cycle Storage Inhibit mode, depending on the position of the RATE switch. The operation is as follows:

1. If this switch is pressed after a normal halt, instruction processing continues as though no halt had occurred. Interruptions pending are taken after execution of the first instruction.
2. If this switch is pressed after an abnormal halt or system reset, the results will not necessarily be predictable.
3. The type of operation executed by the START pushbutton depends on the position of the RATE switch (described later in this section).

STOP Pushbutton

This pushbutton provides the ability to completely terminate machine operations without destroying the machine environment. The operation is as follows:

1. The CPU proceeds to the end of the machine instruction being executed at the time the Stop command is recognized.
2. All waiting interruptions not masked off are executed.
3. All I/O operations in process are allowed to be completed.
4. The CPU is placed in the stopped state.
5. The operator can continue normal program operation by pressing the START pushbutton, or he can execute certain manual operations (e.g., an instruction-step operation).

RATE Switch

This rotary switch selects the rate that instructions are executed. It has four positions: PROCESS, INSN STEP (instruction step), SINGLE CYCLE, and SINGLE CYCLE STORAGE INHIBIT.

Pressing the START pushbutton with the RATE switch in the PROCESS position causes the system to operate at the normal clock speed of 200 ns.

With the RATE switch set to the INSN STEP position, the system executes one complete machine instruction for each depression of the START pushbutton. The high-resolution interval timer is disabled during Instruction Step mode. The operation is as follows:

1. Any machine instruction can be executed in this mode. Interruptions are executed after the instruction is completed.
2. The stop point is identical with that achieved by the STOP pushbutton.

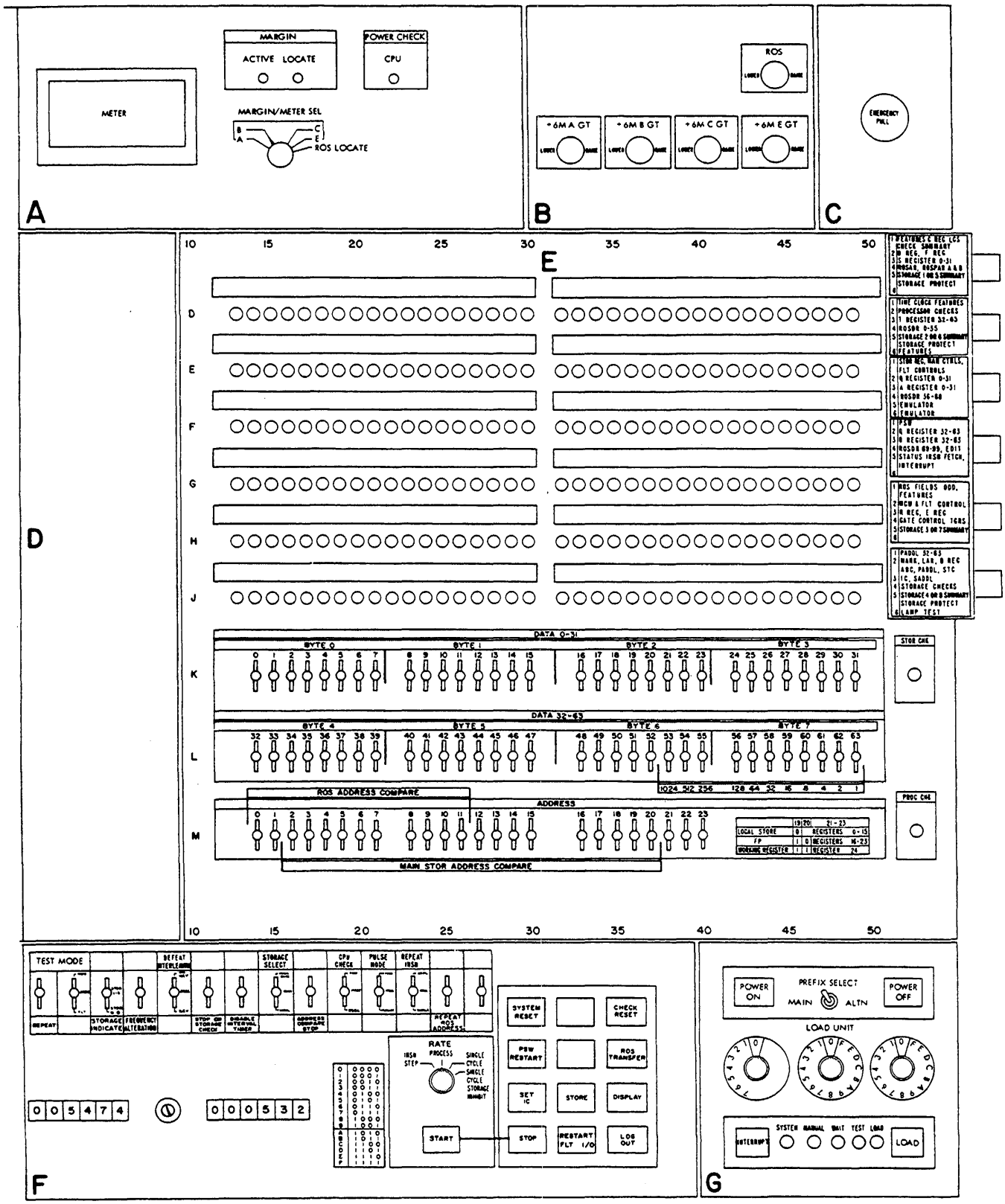


Figure 13. System Control Panel

3. When I/O operations are started, they are completed to the interruption point.
4. The test light is on in the INSN STEP position.

With the RATE switch set to the SINGLE CYCLE position, each depression of the START pushbutton advances the CPU one 200-ns machine cycle. The operation is:

1. When the instruction being single-cycled uses asynchronous devices, it single-cycles through all CPU functions of the instruction to the initiation point of the asynchronous operation. The asynchronous operation starts on the next depression of the START pushbutton and runs to the completion point in a normal manner.
2. If the asynchronous device initiates an interruption request during a single-cycle operation, it is not automatically executed. The interruption is divided into single operations. More than one depression of the START pushbutton is required to complete the transfer of PSWs.
3. During CPU storage cycles, more than one clock pulse is taken for each depression of the START pushbutton.
4. The test light is on in the SINGLE CYCLE position. The SINGLE CYCLE INHIBIT position is for customer engineering functions.

SYSTEM RESET Pushbutton

This pushbutton resets the on-line channels, control units, and CPU controls, including machine checks, to their initial state. The operation is:

1. The pushbutton is active in all modes of operation.
2. All check indicators are reset.
3. The data flow registers are not reset.
4. A system reset does not affect equipment in off-line channel operations.
5. The CPU is placed in the stopped state.
6. Since a system reset can occur in the middle of an operation, the contents of the PSW and of result registers and storage locations are unpredictable.

CHECK RESET Pushbutton

This pushbutton provides a means of resetting all check indicators in the CPU to the nonerror state. The operation is:

1. Pressing the pushbutton resets all CPU check triggers and latches to the no-check state (it is a subset of the SYSTEM RESET pushbutton).
2. This reset clears all CPU logic check indicators on the system control panel.
3. If the CPU is stopped because of a machine check, processing continues when the check indicators are reset. In this case, results are not predictable.

STORAGE SELECT Switch (Section F)

This toggle switch provides a means of selecting the storage unit that is to be addressed by the address keys when used with the DISPLAY or the STORE pushbutton. The operation is:

1. The MAIN position selects the main storage for addressing when storing or displaying.
2. The LOCAL position selects the local store for addressing when storing or displaying.
3. The MAIN BYTE position selects the main storage for addressing when storing or displaying, but causes only the byte addressed by the three low-order address keys to be stored.

ADDRESS Switches (Section E)

These 24 toggle switches provide a means of manually selecting an addressable location in storage. The operation is:

1. The 24 switches are arranged in hexadecimal groups to permit storage addressing.
2. Correct parity is generated automatically.
3. Switches 2 through 20 are used with the ADDRESS COMPARE switch to select an address for an address compare stop or an address compare sync.
4. Switches 0 through 11 are used for selection of ROS address and for a ROS compare sync. The sync pulse is provided whenever the ROS address compares successfully with the configuration placed in these switches.

DATA Switches (Section E)

These 64 toggle switches provide a means of manually entering data into the location selected by the STORAGE SELECT switch and the ADDRESS switches. The operation is:

1. The 64 switches are arranged in hexadecimal groups to permit data entry.
2. Correct parity is generated automatically.
3. Switches 53 through 63 are used as a count when the pulse mode count function is being performed.

STORE Pushbutton

This pushbutton provides a means of storing information into any address in the storage specified by the STORAGE SELECT toggle switch. The operation is:

1. The contents of the DATA switches are placed in the location specified by the ADDRESS switches and the STORAGE SELECT switch.
2. Correct parity is generated automatically.
3. If the STORAGE SELECT switch is in the MAIN position, the entire contents of the data keys are stored in the main storage.

4. If the STORAGE SELECT switch is in the LOCAL position, the five low-order address switches specify the local store location in which the contents of the right half of the configuration in the data switches will be stored. Address switch 19, when in the 0 position, permits storing in the general purpose registers; when in the 1 position, switch 19 permits storing into the floating point registers. The specific address is determined by the five low-order address switches. Address switches 19 and 20, when set to 1s, will address the working register (a local storage register not accessible under program control).
5. If the STORAGE SELECT switch is in the MAIN BYTE position, the byte of data in the data switches specified by the three low-order bits of the address switches will be stored in main storage at the address specified by the address keys.
6. The machine must be in a stopped state for this pushbutton to function.

DISPLAY Pushbutton

This pushbutton is pressed to display information in the location specified by the STORAGE SELECT switch and the ADDRESS switches. The operation is as follows:

1. If the STORAGE SELECT switch is in the MAIN BYTE position, the information in main storage at the address specified by the ADDRESS switches is displayed in the ST and AB registers. (See "Roller Indicators.")
2. If the STORAGE SELECT switch is in the LOCAL position, the information in the local store is displayed in the T-register. (See "Roller Indicators.")
3. The machine must be in a stopped state for this pushbutton to function.

ADDRESS COMPARE Switch

This toggle switch provides a machine stop on a CPU storage compare. The operation is as follows:

1. In the center (normal) position, a synchronizing pulse (for CE use) is provided whenever the storage address bus compares successfully with bits 2 through 20 of the address keys.
2. In the down (STOP) position, the machine stops at the end of the instruction in progress whenever the storage address bus compares successfully with bits 2 through 20 of the address keys.
3. The test light is on whenever this switch is in the down position.

PSW RESTART Pushbutton

This pushbutton switch provides a method to restart programs by loading a new PSW from the contents of storage location 0. The operation is as follows:

1. With the machine in the stopped or reset state, pressing this pushbutton causes a new PSW to be fetched from storage location 0.
2. The CPU continues processing after the new PSW is fetched if the RATE switch is in the PROCESS position.

SET IC Pushbutton (Instruction Counter)

This pushbutton is pressed to enter an address into the instruction address of the current PSW. The operation is as follows:

1. This pushbutton sets bits 40–63 of the current PSW to the value specified in the ADDRESS switches. The CPU is reset to the start of an I-fetch at that address. The instruction at the specified location is then fetched and loaded into the instruction buffer, and the instruction counter is updated. The machine then returns to the stopped state.
2. The machine must be in the stopped state for this pushbutton to function.

Note: The instruction address is displayed in the D register when the CPU is in the stopped state. (See "Roller Indicators".) The new address contained in the instruction counter is one or two doublewords more than the instruction address contained in the ADDRESS switches.

CPU CHECK Switch

This toggle switch provides a means of controlling the system when a machine check is encountered. The operation is as follows:

1. With this switch in the PROC position, if the machine check mask in the PSW is a 1 when a machine check is detected, the machine status is logged to storage and an interrupt trap is initiated. If the machine check mask in the PSW is a 0, the check is ignored, except that the check triggers are turned on.
2. With this switch in the STOP position, the check triggers are set when a machine check is detected. The CPU stops and no log-out occurs. If the CHECK RESET pushbutton is pressed, the operation is resumed, but the results are not predictable.
3. With this switch in the DSAB position the check triggers are set when a machine check is detected. Log-out, interruption, and termination do not occur. The check triggers can be reset by pressing the CHECK RESET pushbutton or by a system reset.
4. The test light is on whenever this switch is in the STOP or the DSAB position.

LOG OUT Pushbutton

This pushbutton is active when the RATE switch is not in the PROC position; it provides a means of logging the

machine status into storage. The operation is as follows:

1. Pressing this pushbutton causes the machine status to be stored in fixed locations in main storage.
2. This pushbutton is inactive under a normal processing condition. Log-out is the process of storing the status of most of the CPU indicators and registers in main storage. The CPU log-out area occupies 44 words (176 bytes) of main storage, starting at byte 128. The channel log-out area follows the CPU log-out area and occupies six words (24 bytes) of main storage, starting at byte 304.

STOP ON STORAGE CHECK Switch

This switch provides a means of inhibiting storage accesses when a storage check occurs so that the indicators will not be changed. Storage checks resulting from accesses by channels and other processors also cause a stop. The switchable indicators (rollers) are checked to determine the error and the address of the failing main storage word.

This switch should be operated with the CPU CHECK switch in the STOP position. The storage-stop state caused by a storage check is different from the stopped state.

The test light is on whenever this switch is in the down position.

STORAGE INDICATE Switch (2067-1 Only)

Information from a maximum of eight storage arrays (two 128 K-byte storage arrays per 2365) can be displayed by the roller indicators. Depending on the setting of this switch, information from storage arrays 1 through 4 or from storage arrays 5 through 8 is displayed.

ROLLER INDICATORS

Section E contains six rows of 36 indicator lights. Above each row of lights is an opening, and behind each opening is a roller that can be positioned to identify the significance of the related indicator lights for various operations. Each roller is manually placed in one of six positions by a positioning knob at the right side of panel B (see Figure 13). The significance of each roller position is identified by the printing located beside the related positioning knob on the face of the panel.

In the display main storage operation, the contents of the addressed main storage locations are displayed in the ST and AB registers. For this operation, the ST register is identified on roller 1, position 3 and roller 2, position 3. The AB register is identified on roller 3, position 3 and roller 4, position 3.

In the display local storage operation, the contents of the addressed register are displayed in the T register. The T register is identified on roller 2, position 3.

When the CPU is in the wait or stopped state, all but two parts of the current PSW are identified on roller 4, position 1. The two exceptions are (1) the instruction address, which is displayed in the D register (roller 1, position 2), and (2) the instruction length code (ILC), which is displayed in E register positions 0 and 1 (roller 5, position 3).

CUSTOMER ENGINEERING CONTROL

The ROS (read-only storage) TRANSFER and the RESTART FLT I/O pushbuttons, as well as all lever switches (except STORAGE SELECT) are principally for customer engineering use. All switches and lights and the meter in sections A and B are for customer engineering use only.

KEY SWITCH AND METERS

The usage meter and a customer engineering meter are installed in section F of the system control panel. A key switch controls the meter to be run when the machine is in process. When power is on and the key switch is in the customer operation position, the usage meter accumulates time while the SYSTEM light is on. If the key switch is in the customer engineer position and the TEST light is on, the CE meter accumulates time while the SYSTEM light is on.

MODEL-DEPENDENT FUNCTIONS

The compatibility rule of System/360 does not apply to a number of detail functions for which neither the frequency of occurrence nor usefulness of results warrants identical action on all models. These functions are concerned with the handling of invalid programs and machine malfunctions and are explicitly identified in *System/360 Principles of Operation,* Form A22-6821, in the section "Functions that May Differ Among Models." Whenever model dependency exists, the definition of System/360 allows choice in implementation or specifies that the operation is unpredictable. The intent is that the user should ignore results that are defined as unpredictable and should not base his program on any function where choice in implementation is permitted.

Considering any particular installation and operation, the operation normally is not truly unpredictable; the action may depend on the particular system components or on the input data. The purpose of this section is to describe how some of the model-dependent functions are performed on the Model 67.

Note, however, that writing a program on the basis of information contained in this section is in violation of the rules of System/360 compatibility. If a program relies on a function that is model-dependent, it may not run on another model of System/360. Even if the program takes into account the model-dependent operation of all other models of System/360, difficulties may be encountered if new models of System/360 are introduced. Furthermore, a mandatory engineering change may, in some instances, require a change in the execution of a model-dependent function in a machine installed in a customer's office; hence, changes may be required in a program making use of such model-dependent information.

INSTRUCTION EXECUTION

Modified Instructions

When the Model 67 is in Extended PSW mode, the operations performed by the ten instructions listed below are modified as described in the "System Components" section of this manual:

1. Load PSW
2. Set Storage Key
3. Insert Storage Key
4. Translate and Test
5. Edit and Mark
6. Load Address
7. Supervisor Call
8. Branch on Index High
9. Branch on Index Low or Equal
10. Set System Mask

When the Model 67 is not in extended PSW mode, these instructions operate as described in *System/360 Principles of Operation.*

The Read Direct and Write Direct instructions operate as described in *System/360 Principles of Operation* except when the Model 67 is equipped with the extended direct control feature; in this case, the operation of these instructions is as described in the "System Components" section of this manual.

New Instructions

Five instructions are unique to the Model 67 and, therefore, are not described in *System/360 Principles of Operation:*

1. Load Multiple Control
2. Store Multiple Control
3. Load Real Address
4. Branch and Store (RR)
5. Branch and Store (RX)

These instructions are described in the "System Components" section of this manual.

Diagnose Instruction

The Diagnose instruction is used for compatibility feature operation and for maintenance purposes. When the Model 67 is equipped with a compatibility feature, the I_2 field of the Diagnose instruction can be coded to allow entry into Emulator mode. When I_2=02, the Diagnose instruction becomes an Enter Emulator Mode instruction. Further information on use of the Diagnose instruction with the compatibility feature is described in *IBM System/360 Special Feature Description, 709/7040/7044/7090/7094II Compatibility Feature for System/360 Models 65 and 67,* Form A27-2715.

MACHINE-CHECK INTERRUPTION

For a machine-check interruption, the old PSW is stored at location 48 with a zero interruption code. The state of the CPU is scanned out into the CPU diagnostic scan-out area, which is 22 doublewords (176 bytes) in size and starts at location 128; the channel scan-out area is three doublewords in size, starting at location 304.

INSTRUCTION-LENGTH CODE

When the instruction-length code in the program old PSW is zero, the exception was not necessarily caused by the last instruction executed. Interruptions that cause a zero instruction-length code to be set in the program old PSW are referred to as *imprecise program interruptions,* and the exceptions causing such interrupts are referred to as *imprecise exceptions.* By contrast, a program interruption associated with a nonzero instruction-length code, and the corresponding exception, are referred to as *precise.* In the Model 67, an *imprecise* program exception can occur only when a protection check (store only) is encountered.

EXTENDED DIRECT CONTROL

The extended direct control feature, a modification of the direct control feature, is described in the "Systems Components" section of this manual. This description modifies the "Multisystem Operation" and "Direct Control Feature" descriptions in *System/360 Principles of Operation.*

INTRODUCTION

Two types of instruction times are presented in this section: (1) average times for all instructions executed by the Model 67, and (2) detailed times for all variable-field-length instructions executed by the Model 67. All symbols used to present Model 67 instruction times are defined in the "Legends" description that precedes the instruction times. Standard System/360 instruction timing formula legends are used.

Times are provided for instruction execution when instructions and data are located in main processor storage. All times are in microseconds unless otherwise noted. Complete information on each instruction is presented elsewhere in this publication or in the *IBM System/360 Principles of Operation* manual.

Timing Considerations

Unless otherwise noted, the following conditions were used in the development of instruction times in this section:

1. The time required for indexing by a base register is included in the times given. For those instructions that can be double-indexed (indicated by one or two asterisks in the instruction column on the table for average instruction times), an additional 0.15 usec (one asterisk) or 0.2 usec (two asterisks) must be added to the times given in the table.
2. In all arithmetic operations, positive and negative operands are equally probable.
3. Each bit location has equal probability of containing bit values 0 or 1, and each bit location is independent of other bit locations.
4. Addresses for unsuccessful branches are valid but unprotected storage locations.
5. Decimal data can contain digit values 0–9 in each digit position with equal probability. When either the multiplier of a Multiply Decimal instruction or a divisor of a Divide Decimal instruction contains the digits 5 or 6 in each position, a slower than average instruction time (worst case) will result. An incremental decrease in instruction time is realized as the digits descend from 5 to 0 or ascend from 6 to 9.
6. Instructions may start on even or odd halfwords with equal probability.
7. Interruptions are not reflected in these timings.
8. All timings provided include the storage access (S1 or S2 only), decoding, and execution times for instructions.
9. Instruction times for each format are modified as follows:
   a. 2067-1
      (1) RR – Add 10 ns for each instruction located in 3rd or 4th storage unit.
      (2) RX, RS, and SI – Add 25 ns for each instruction and 50 ns for each operand located in 3rd or 4th storage unit.
      (3) SS – Add 40 ns for each instruction and 50 ns for each operand located in 3rd or 4th storage unit.
   b. 2067-2
      (1) RR – For each instruction located in storage units 3–8, add one of the following:

S3 - 10 ns	S6 - 40 ns
S4 - 25 ns	S7 - 50 ns
S5 - 25 ns	S8 - 60 ns

      (2) RX, RS, and SI – For each instruction located in storage units 3–8, add one of the following:

S3 - 25 ns	S6 - 75 ns
S4 - 50 ns	S7 - 100 ns
S5 - 50 ns	S8 - 125 ns

      (3) SS – For each instruction located in storage units 3–8, add one of the following:

S3 - 40 ns	S6 - 110 ns
S4 - 75 ns	S7 - 150 ns
S5 - 75 ns	S8 - 190 ns

      (4) For each operand located in storage units 3–8, add one of the following:

S3 - 50 ns	S6 - 150 ns
S4 - 100 ns	S7 - 200 ns
S5 - 100 ns	S8 - 250 ns

      (5) Additional delay due to dynamic address translation is identified in the 2067 description in this publication.
10. Instruction times for STM, STMC, LM, LMC, VFL Decimal, and Logical-SS format instructions are modified as follows. When translation is indicated, the beginning and ending address of each field is tested by the translation process before instruction execution. Execution time is exposed to one of the following increments of additional time:

Translations per Field	Source of Translated Address	Additional Delay per Field (ns)
1	Array	1850
1	Lookup	2450 + 2 storage references
2	Array (2)	3050
2	Array & Lookup	3650 + 2 storage references
2	Lookup (2)	4250 + 4 storage references

Timing Assumptions

Unless otherwise noted, the following assumptions were used in the development of instruction times in this section:

1. For the Add Decimal (AP) and the Subtract Decimal (SP) instructions, the first operand (i.e., the destination field) is assumed to be equal to or greater than the length of the second operand (i.e., the source field).
2. In the Edit and Mark (EDMK) instruction, an address is stored once; i.e., this instruction is used with a single field, or a line with only one numeric field is employed rather than a complete print line.
3. In the Translate and Test (TRT) instruction, it is assumed that a nonzero byte from a translate and test table is found.
4. The instruction times for floating-point instructions depend on both the number of hexadecimal digits that are preshifted and postshifted and on the number of times the result is recomplemented. The floating-point instruction times given in this section are a weighted average of these variables.
5. For the Pack (PACK), Unpack (UNPK), and Move with Offset (MVO) instructions, it is assumed that no overflow field occurs.

LEGENDS

These legends are shorthand notations for data characteristics and computer conditions that were used to develop timing formulas for Model 67 instructions.

Legends for Instructions with Multiple Timing Formulas

Legends A_1 to A_4 are used for the Store Multiple or Load Multiple instruction, depending upon quantity of general registers and position of doubleword boundaries.

- A_1: Use if the number of registers is 2 and if the operand lies on doubleword boundaries.
- A_2: Use if the number of registers is > 2 and even and if the operand lies on doubleword boundaries.
- A_3: Use if the number of registers is even and if the operand does not lie on doubleword boundaries.
- A_4: Use if the number of registers is odd.

Legends E_5 and E_6 are used for the Execute instruction.

- E_5: Use when subject instruction is successful branch.
- E_6: Use when subject instruction is not a successful branch.

Legends V_1 or V_6 are used for Move instructions, depending upon the location of operand fields.

- V_1: Use if first and second operand fields start and end on doubleword boundaries.
- V_2: Use if first and second operand fields start at corresponding byte addresses within doublewords but do not lie on doubleword boundaries.
- V_3: Use if first and second operand fields do not start at corresponding byte addresses within doublewords or if $N < 8$.
- V_5: Use if $N_1 \leq N_2$.
- V_6: Use if $N_1 > N_2$.

Note: A byte address of a doubleword can have the value 0, 1, 2, 3, 4, 5, 6, or 7. A doubleword (8 bytes) must have an address that is a multiple of the number 8, but the four low-order bits of the binary address of a byte within a doubleword can have any value from 0 to 7.

Legends for Instructions with Single Timing Formulas

B	=	Total number of bytes of the first operand that are processed. It applies to instructions with a single-length field.
E	=	Time for the subject instruction that is executed by the Execute instruction.
ED	=	External delay.
F_1	=	1 if the branch operation is successful,
	=	0 otherwise.
G_1	=	1 if an overflow interruption occurs (PSW bit 36 = 1) or fixed point divide interruption occurs,
	=	0 otherwise.
G_3	=	0 if operand to be converted is positive,
	=	1 otherwise.
GR	=	Number of general registers loaded or stored.
M	=	Greater of N_1 or N_2.
MK	=	Number of times the mark address is stored in the Edit and Mark instruction.
N	=	Total number of bytes in the first operand for those instructions with a single-length field.
N_1	=	Total number of bytes in the first operand (destination).
N_2	=	Total number of bytes in the second operand (source).
N_3	=	Total number of bytes that overlap between the first and second operands. N_3 = 0 for nonoverlapping fields or for overlapping fields where the address of the second operand is greater than or equal to the first operand address.
N_6	=	Number of bytes of the field that lie outside of that part of the field bounded by doubleword.
$NWBB_1$	=	Number of word boundary crossovers for that part of the first operand processed.
$NWBB_2$	=	Number of word boundary crossovers for that part of the second operand processed.

$NWBL_1$ = Number of word boundary crossovers for the first operands.

$NWBL_2$ = Number of word boundary crossovers for the second operands.

$NWBL_1L_2$ = Number of word boundary crossovers for that part of the first operand that consists of N_2 bytes of high-order zeros.

q_4 = Quotient found by dividing by 4 the number of positions to be shifted.

QS = Smaller of $N_1 - 8$ or $N_1 - N_2$.

r_4 = Remainder after dividing by 4 the number of position to be shifted.

S_1 = 1 if $r_4 = 3$, or if $q_4 = 0$,
= 2 if $r_4 = 3$ and $q_4 = 0$,
= 0 otherwise.

S_2 = -1 if $r_4 = 0$,
= 1 if $r_4 = 1$ and $q_4 = 0$,
= 0 otherwise.

S_3 = 0 if $r_4 = 0$ and $q_4 \neq 0$,
= 1 if $r_4 = 0$ and $q_4 = 0$,
= 3 if $r_4 = 1$,
= 5 if $r_4 = 2$ or 3.

S_4 = 0 if $r_4 = 0$,
= 4 if $q_4 = 0$ and $r_4 = 1$, or if $q_4 \neq 0$ and $r_4 = 2$,
= 3 if $q_4 = 0$ and $r_4 = 2$, or if $q_4 \neq 0$ and $r_4 = 3$,
= 2 if $q_4 = 0$ and $r_4 = 3$,
= 5 if $q_4 \neq 0$ and $r_4 = 1$.

T_1 = 1 if the result field is recomplemented (i.e., changes sign),
= 0 otherwise.

T_2 = 1 if the result field is zero,
= 0 otherwise.

T_3 = 1 if $N_2 < 1/2 (N_1 + 1)$,
= 0 otherwise.

T_6 = 0 if $N_2 \leq 4$,
= 1 otherwise.

T_7 = 0 if $N_1 \leq 8$,
= 1 otherwise.

T_8 = 0 if fields do not overlap,
= 1 otherwise.

T_9 = 0 if any nonzero function byte is found,
= 1 otherwise.

T_{12} = 1 if R_1 field of Execute instruction is not zero,
= 0 otherwise.

U_1 = Select out delay + device delay.

U_2 = Device delay for halt I/O sequence.

W = Total number of doublewords in the first operand for those instructions with a single-length field.

EFFECT OF CHANGE TO FLOATING-POINT FEATURE

Installation of the floating-point change modifies the average time of eight instructions. Instruction times for floating-point instructions depend on the number of hexadecimal digits that are preshifted and post-shifted, as well as on the number of times the result is recomplemented. Each of the floating-point instruction times listed is an average of actual execution times. Although each value is the most accurate that can be given, the actual time is data-dependent.

		Times (in usec)	
Instruction	Form Mnem	Mod 67-1	Mod 67-2
Add Unnormalized Long	RR AWR	1.75	1.79
Add Unnormalized Long*	RX AW	2.50	2.73
Halve Long	RR HDR	1.2+0.4NC +0.6A	1.2+0.4NC +0.6A
Halve Short	RR HER	1.2+0.4NC +0.6A	1.2+0.4NC +0.6A
Multiply Long	RR MDR	7.65	7.69
Multiply Long*	RX MD	8.00	8.23
Subtract Unnormalized Long	RR SWR	1.75	1.79
Subtract Unnormalized Long*	RX SW	2.50	2.73

where: A = 1 if underflow occurs
= 0 otherwise
NC = number of normalization cycles

*Add 0.15 usec when double indexed.

Without the floating-point change, the times given under "Average Times" apply. The above times should be used when this change is installed.

AVERAGE INSTRUCTION TIMES

The average times for all instructions used by the Model 67 are as follows:

Instruction	Form	Mnemonic	Model 67-1 Times (usec)	Model 67-2 Times (usec)
Add	RR	AR	0.65	0.69
Add*	RX	A	1.40	1.63
Add Decimal	SS	AP	3.4+0.2M+ $0.2N_1$+$0.1N_2$+ $T_1(2.0+0.4N_1)$ +$1.2T_2$	3.52+0.2M+ $0.2N_1$+$0.1N_2$+ $T_1(2.15+0.4N_1)$ $1.35T_2$
Add Halfword*	RX	AH	1.80	2.03
Add Logical	RR	ALR	0.65	0.69
Add Logical*	RX	AL	1.40	1.63
Add Normalized Long	RR	ADR	1.72	1.76
Add Normalized Long*	RX	AD	2.45	2.68
Add Normalized Short	RR	AER	1.68	1.72
Add Normalized Short*	RX	AE	2.43	2.66
Add Unnormalized Long	RR	AWR	1.65	1.69
Add Unnormalized Long*	RX	AW	2.40	2.63
Add Unnormalized Short	RR	AUR	1.64	1.68
Add Unnormalized Short*	RX	AU	2.38	2.61
AND	RR	NR	1.25	1.29
AND*	RX	N	2.00	2.23
AND	SI	NI	1.73	1.96
AND	SS	NC	2.8+0.5N+ $0.2N_3$	2.92+0.5N+ $0.2N_3$
Branch and Link	RR	BALR	1.20	1.24
Branch and Link**	RX	BAL	1.20	1.43
Branch and Store	RR	BASR	1.43	1.47
Branch and Store*	RX	BAS	1.40	1.63
Branch on Condition	RR	BCR	$0.7+0.4F_1$	$0.74+0.55F_1$
Branch on Condition**	RX	BC	$0.8+0.3F_1$	$0.88+0.45F_1$
Branch on Count	RR	BCTR	$0.98+0.17F_1$	$1.02+0.32F_1$
Branch on Count**	RX	BCT	1.15	1.38
Branch on Index High	RS	BXH	$1.6-0.2F_1$	$1.68-0.05F_1$
Branch on Index Low or Equal	RS	BXLE	$1.6-0.2F_1$	$1.68-0.05F_1$
Compare	RR	CR	0.65	0.69
Compare*	RX	C	1.40	1.63
Compare Decimal	SS	CP	3.47+0.2M+ $0.13N_1$+$0.1N_2$	3.59+0.2M+ $0.13N_1$+$0.1N_2$
Compare Halfword*	RX	CH	1.80	2.03
Compare Logical	RR	CLR	0.65	0.69
Compare Logical*	RX	CL	1.40	1.63
Compare Logical	SI	CLI	1.40	1.63
Compare Logical	SS	CLC	2.9+0.4B	3.02+0.4B
Compare Long	RR	CDR	1.26	1.3
Compare Long*	RX	CD	2.00	2.23
Compare Short	RR	CER	1.24	1.28
Compare Short*	RX	CE	1.98	2.21
Convert to Binary*	RX	CVB	7.6+0.2G_3+ $0.2G_1$	7.83+0.2G_3+ $0.2G_1$
Convert to Decimal*	RX	CVD	8.65+0.4G_3	8.88+0.4G_3
Divide	RR	DR	8.45+0.15G_1	8.49+0.15G_1
Divide*	RX	D	8.70+0.15G_1	8.93+0.15G_1
Divide Decimal	SS	DP	5.95+4.5N_1- $4.7N_2$+$2.2N_2$ (N_1-N_2)+$0.6T_6$ +$1.2T_7$QS	6.07+4.5N_1- $4.7N_2$+$2.2N_2$ (N_1-N_2)+$0.6T_6$ +$1.35T_7$QS
Divide Long	RR	DDR	13.35	13.39
Divide Long*	RX	DD	14.10	14.33

*Add 0.15 usec when double-indexed
**Add 0.2 usec when double-indexed

Instruction	Form	Mnemonic	Model 67-1 Times (usec)	Model 67-2 Times (usec)
Divide Short	RR	DER	6.55	6.59
Divide Short*	RX	DE	7.30	7.53
Edit	SS	ED	3.20+0.63N+ $0.1N_2$	3.32+0.63N+ $0.1N_2$
Edit and Mark	SS	EDMK	3.20+0.63N+ $0.1N_2$+1.2MK	3.32+0.63N+ $0.1N_2$+1.35MK
Exclusive OR	RR	XR	1.25	1.29
Exclusive OR*	RX	X	2.00	2.23
Exclusive OR	SI	XI	1.73	1.96
Exclusive OR	SS	XC	2.8+0.5N+ $0.2N_3$	2.92+0.5N+ $0.2N_3$
Execute*	RX	EX	E_5=1.45+E E_6=2.6+E+0.4T_{12}	E_5=1.68+E E_6=2.83+E+0.4T_{12}
Halt I/O	SI	HIO	1.4+U_1+U_2	1.63+U_1+U_2
Halve Long	RR	HDR	1.25	1.29
Halve Short	RR	HER	1.05	1.09
Insert Character*	RX	IC	1.40	1.63
Insert Storage Key	RR	ISK	2.85	3.19
Load	RR	LR	0.65	0.69
Load*	RX	L	1.20	1.43
Load Address*	RX	LA	0.90	1.13
Load and Test	RR	LTR	0.65	0.69
Load and Test Long	RR	LTDR	1.05	1.09
Load and Test Short	RR	LTER	0.85	0.89
Load Complement	RR	LCR	0.65	0.69
Load Complement Long	RR	LCDR	1.05	1.09
Load Complement Short	RR	LCER	0.85	0.89
Load Halfword*	RX	LH	1.40	1.63
Load Long	RR	LDR	1.05	1.09
Load Long*	RX	LD	1.40	1.63
Load Multiple	RS	LM	A_1=1.40 A_2=0.8+0.3GR A_3=1.2+0.3GR A_4=1.0+0.3GR	A_1=1.63 A_2=0.88+0.38GR A_3=1.28+0.38GR A_4=1.08+0.38GR
Load Multiple Control	RS	LMC	1.6+1.4GR	1.68+1.48GR
Load Negative	RR	LNR	0.95	0.99
Load Negative Long	RR	LNDR	1.05	1.09
Load Negative Short	RR	LNER	0.85	0.89
Load Positive	RR	LPR	0.95	0.99
Load Positive Long	RR	LPDR	1.05	1.09
Load Positive Short	RR	LPER	0.85	0.89
Load PSW	SI	LPSW	2.20	2.43
Load Real Address	RX	LRA	2.3	2.53
Load Short	RR	LER	0.65	0.69
Load Short*	RX	LE	1.20	1.43
Move	SI	MVI	1.33	1.56
Move	SS	MVC	V_1=2.0+1.4W V_2=3.1+0.2N+0.2N_6 V_3=2.93+0.38N+0.2N_3	V_1=2.42+1.55W V_2=3.22+0.2N+0.2N_6 V_3=3.05+0.38N+0.2N_3
Move Numerics	SS	MVN	2.8+0.5N+ $0.2N_3$	2.92+0.5N+ $0.2N_3$
Move with Offset	SS	MVO	2.93+0.27N_1+ $0.3N_2$+0.4N_3 if $N_1 \geq N_2$	3.05+0.27N_1+ $0.3N_2$+0.4N_3 if $N_1 \geq N_2$
Move Zones	SS	MVZ	2.8+0.5N+ $0.2N_3$	2.92+0.5N+ $0.2N_3$
Multiply	RR	MR	4.45	4.49
Multiply*	RX	M	4.80	5.03
Multiply Decimal	SS	MP	3.77+3.4N_1- 2.9N_2+1.0N_2 (N_1-N_2)	3.89+3.4N_1 2.9N_2+1.0N_2 (N_1-N_2)

*Add 0.15 usec when double-indexed
**Add 0.2 usec when double-indexed

Instruction	Form	Mnemonic	Model 67-1 Times (usec)	Model 67-2 Times (usec)
Multiply Halfword*	RX	MH	5.00	5.23
Multiply Long	RR	MDR	7.25	7.29
Multiply Long*	RX	MD	7.60	7.83
Multiply Short	RR	MER	4.05	4.09
Multiply Short*	RX	ME	4.40	4.63
OR	RR	OR	1.25	1.29
OR*	RX	O	2.00	2.23
OR	SI	OI	1.73	1.96
OR	SS	OC	2.8+0.5N+ $0.2N_3$	2.92+0.5N+ $0.2N_3$
Pack	SS	PACK	$2.87+0.27N_1+$ $0.2N_2$, if $N_1 > \frac{N_2+1}{2}$	$2.99+0.27N_1+$ $0.2N_2$, if $N_1 > \frac{N_2+1}{2}$
Read Direct	SI	RDD	2.70+ED	–
Set Program Mask	RR	SPM	0.85	0.89
Set Storage Key	RR	SSK	1.8	2.14
Set System Mask	SI	SSM	4.16	4.54
Shift Left Double	RS	SLDA	$0.9+0.4q_4+$ $0.2S_3$	$0.98+0.4q_4+$ $0.2S_3$
Shift Left Double Logical	RS	SLDL	$0.9+0.4q_4+$ $0.2S_3$	$0.98+0.4q_4$ $0.2S_3$
Shift Left Single	RS	SLA	$0.7+0.2q_4+$ $0.2S_1$	$0.78+0.2q_4+$ $0.2S_1$
Shift Left Single Logical	RS	SLL	$0.7+0.2q_4+$ $0.2S_1$	$0.78+0.2q_4+$ $0.2S_1$
Shift Right Double	RS	SRDA	$0.9+0.4q_4+$ $0.2S_4$	$0.98+0.4q_4+$ $0.2S_4$
Shift Right Double Logical	RS	SRDL	$0.9+0.4q_4+$ $0.2S_4$	$0.98+0.4q_4+$ $0.2S_4$
Shift Right Single	RS	SRA	$0.9+0.2q_4+$ $0.2S_2$	$0.98+0.2q_4+$ $0.2S_2$
Shift Right Single Logical	RS	SRL	$0.9+0.2q_4+$ $0.2S_2$	$0.98+0.2q_4+$ $0.2S_2$
Start I/O	SI	SIO	$1.40+U_1$	$1.63+U_1$
Store*	RX	ST	0.93	1.16
Store Character*	RX	STC	1.33	1.56
Store Halfword*	RX	STH	1.73	1.96
Store Long*	RX	STD	0.93	1.16
Store Multiple	RS	STM	$A_1=1.33$ $A_2=0.53+0.2GR$ $A_3=1.33+0.2GR$ $A_4=0.93+0.2GR$	$A_1=1.56$ $A_2=0.61+0.28GR$ $A_3=1.41+0.28GR$ $A_4=1.01+0.28GR$
Store Multiple Control	RS	STMC	1.0+1.4GR	1.08+1.46GR
Store Short*	RX	STE	0.93	1.16
Subtract	RR	SR	0.65	0.69
Subtract*	RX	S	1.40	1.63
Subtract Decimal	SS	SP	$3.4+0.2M+$ $0.2N_1+0.1N_2+$ $T_1(2.0+0.4N_1)$ $+1.2T_2$	$3.52+0.2M+$ $0.2N_1+0.1N_2+$ $T_1(2.15+0.4N_1)$ $+1.35T_2$
Subtract Halfword*	RX	SH	1.80	2.03
Subtract Logical	RR	SLR	0.65	0.69
Subtract Logical*	RX	SL	1.40	1.63
Subtract Normalized Long	RR	SDR	1.72	1.76
Subtract Normalized Long*	RX	SD	2.45	2.68
Subtract Normalized Short	RR	SER	1.68	1.72
Subtract Normalized Short*	RX	SE	2.43	2.66
Subtract Unnormalized Long	RR	SWR	1.65	1.69
Subtract Unnormalized Long*	RX	SW	2.40	2.63
Subtract Unnormalized Short	RR	SUR	1.64	1.68

*Add 0.15 usec when double-indexed

**Add 0.2 usec when double-indexed

Instruction	Form	Mnemonic	Model 67-1 Times (usec)	Model 67-2 Times (usec)
Subtract Unnormalized Short*	RX	SU	2.38	2.61
Supervisor Call	RR	SVC	3.75	3.79
Test and Set	SI	TS	1.80	2.03
Test Channel	SI	TCH	$1.40+U_1$	$1.63+U_1$
Test I/O	SI	TIO	$1.40+U_1$	$1.63+U_1$
Test Under Mask	SI	TM	1.60	1.83
Translate	SS	TR	1.94+1.78N	2.06+1.78N
Translate and Test	SS	TRT	$4.1+1.2B-1.0T_9$	$4.22+1.2B-1.0T_9$
Unpack	SS	UNPK	$2.8+0.4N_1+0.1N_2+0.2T_3$ if $\frac{N_1+1}{2} \geq N_2$	$2.92+0.4N_1+0.1N_2+0.2T_3$ if $\frac{N_1+1}{2} \geq N_2$
Write Direct	SI	WRD	2.2	2.43
Zero and Add	SS	ZAP	$3.50+0.2M+0.1N_1+0.1N_2+0.72T_8+1.2T_2$	$3.62+0.2M+0.1N_1+0.1N_2+0.72T_8+1.35T_2$

*Add 0.15 usec when double-indexed
**Add 0.2 usec when double-indexed

DETAILED INSTRUCTION TIMES (VARIABLE FIELD-LENGTH INSTRUCTIONS)

The following timing formulas are for VFL instructions (i.e., those instructions that contain an "L" field). All times are given in terms of the word boundary crossovers and the operand addresses. The term "word boundary" specifies the boundary between two physical words. A physical word is the amount of information fetched in a single storage cycle (this is 64 bits for all Model 67s). Thus, the number of word boundary crossovers is one less than the number of doublewords spanned by the field.

Timing Formulas for VFL Instructions (Model 67-1)

Instruction	Mnemonic	Formula
Add Decimal	AP	$3.75+0.2M+1.6NWBL_1+0.8NWBL_2+T_1$ $(2.2+2N_1+1.6NWBL_1)+1.2T_2$
AND	NC	$3.16+0.2N+1.6NWBL_1+0.8NWBL_2+0.2N_3$
Compare Decimal	CP	$3.75+0.2M+1.00NWBL_1+0.8NWBL_2$
Compare Logical	CLC	$3.16+0.2B+0.8NWBB_1+0.8NWBB_2$
Note: The compare logical operation is terminated when an unequal condition is found.		
Divide Decimal	DP	$6.2+4.4N_1-4.8N_2+2.20N_2(N_1-N_2)+0.8NWBL_1+0.8NWBL_2+0.6T_6+1.2T_7QS$
Edit	ED	$3.57+0.43N+1.6NWBL_1+0.8NWBL_2$
Edit and Mark	EDMK	$3.56+0.43N+1.6NWBL_1+1.2NWBL_2+1.2MK$
Exclusive OR	XC	$3.16+0.2N+1.6NWBL_1+0.8NWBL_2+0.2N_3$
Move Characters	MVC	$V_1=2.0+1.4W$ $V_2=3.36+0.8NWBL_1+0.8NWBL_2+0.2N_6$ $V_3=3.16+0.6NWBL_2+0.8NWBL_2+0.2N_3+0.2N$
Move Numerics	MVN	$3.16+0.2N+1.6NWBL_1+0.8NWBL_2+0.2N_3$
Move with Offset	MVO	$V_5=3.2+0.4N_1+0.6NWBL_1+0.8NWBL_2+0.4N_3$ $V_6=3.2+0.4N_2+0.2(N_1-N_2)+0.6NWBL_1+0.8NWBL_2+0.4N_3$
Move Zones	MVZ	$3.16+0.2N+1.6NWBL_1+0.8NWBL_2+0.2N_3$
Multiply Decimal	MP	$4.12+3.4N_1-3.2N_2+1.0N_2(N_1-N_2)+0.8NWBL_2+1.6NWBL_1L_2$
OR	OC	$3.16+0.2N+1.6NWBL_1+0.8NWBL_2+0.2N_3$
Pack	PACK	$3.11+0.2N_1+0.1N_2+0.6NWBL_1+0.8NWBL_2$

Instruction	Mnemonic	Formula
Subtract Decimal	SP	$3.75+0.2M+1.6NWBL_1+0.8NWBL_2+T_1(2.2+0.6NWBL_1)+1.2T_2$
Translate	TR	$2.4+1.6N+1.4NWBL_1$
Translate and Test	TRT	$3.6+1.0N+1.8NWBB_1-1.0T_9$
Unpack	UNPK	$3.16+0.2N_1+1.6NWBL_1+1.6NWBL_2+0.2T_3$
Zero and Add	ZAP	$3.75+0.2M+0.8NWBL_1+0.8NWBL_2+0.72T_8+1.2T_2$

Timing Formulas for VFL Instructions (Model 67-2)

Instruction	Mnemonic	Formula
Add Decimal	AP	$3.87+0.2M+1.75NWBL_1+0.95NWBL_2+T_1(2.2+2N_1+1.75NWBL_1)+1.35T_2$
AND	NC	$3.28+0.2N+1.75NWBL_1+0.95NWBL_2+0.2N_3$
Compare Decimal	CP	$3.87+0.2M+1.15NWBL_1+0.95NWBL_2$
Compare Logical	CLC	$3.28+0.2B+0.95NWBB_1+0.95NWBB_2$
		Note: The compare logical operation is terminated when an unequal condition is found.
Divide Decimal	DP	$6.32+4.4N_1-4.8N_2+2.20N_2(N_1-N_2)+0.95NWBL_1+0.95NWBL_2+0.6T_6+1.2T_7QS$
Edit	ED	$3.69+0.43N+1.75NWBL_1+0.95NWBL_2$
Edit and Mark	EDMK	$3.68+0.43N+1.75NWBL_1+1.35NWBL_2+1.2MK$
Exclusive OR	XC	$3.28+0.2N+1.75NWBL_1+0.95NWBL_2+0.2N_3$
Move Characters	MVC	$V_1=2.42+1.55W$ $V_2=3.48+0.95NWBL_1+0.96NWBL_2+0.2N_3$ $V_3=3.28+0.75NWBL_1+0.95NWBL_2+0.2N_3+0.2N$
Move Numerics	MVN	$3.28+0.2N+1.75NWBL_1+0.95NWBL_2+0.2N_3$
Move with Offset	MVO	$V_5=3.32+0.4N_1+0.75NWBL_1+0.95NWBL_2+0.4N_3$ $V_6=3.32+0.4N_2+0.2(N_1-N_2)+0.75NWBL_1+0.95NWBL_2+0.4N_3$
Move Zones	MVZ	$3.28+0.2N+1.75NWBL_1+0.95NWBL_2+0.2N_3$
Multiply Decimal	MP	$4.24+3.4N_1-3.2N_2+1.0N_2(N_1-N_2)+0.95NWBL_2+1.75NWBL_1L_2$
OR	OC	$3.28+0.2N+1.75NWBL_1+0.95NWBL_2+0.2N_3$
Pack	PACK	$3.23+0.2N_1+0.1N_2+0.75NWBL_1+0.95NWBL_2$
Subtract Decimal	SP	$3.87+0.2M+1.75NWBL_1+0.95NWBL_2+T_1(2.2+0.75NWBL_1)+1.35T_2$
Translate	TR	$2.52+1.6N+1.55NWBL_1$
Translate and Test	TRT	$3.72+1.0N+1.95NWBB_1-1.0T_9$
Unpack	UNPK	$3.28+0.2N_1+1.75NWBL_1+1.75NWBL_2+0.2T_3$
Zero and Add	ZAP	$3.87+0.2M+0.95NWBL_1+0.95NWBL_2+0.72T_8+1.35T_2$

Appendix A. Summary of System Control Panel Controls and Indicators

Panel	Switch/Indicator Name	Type*	Function
A	All	-	For customer engineering use.
B	All	-	For customer engineering use.
C	EMERGENCY PULL	PS	When pulled, initiates emergency off in the system.
D	--	-	For expansion.
E	Roller Switches and Indicators	-	There are six 6-position roller switches and 36 indicators associated with each switch. The roller indicators are tested between switch positions. Position 6 of roller 6 is used to test the remaining indicators on the system control panel and on the 2150 console.
E	DATA 0–31, DATA 32–63	TGL	These 64 switches, in hexadecimal groups, permit manual entry of data. Correct parity is generated automatically.
E	ADDRESS	TGL	These 24 switches, in hexadecimal groups, select an addressable location in storage. Correct parity is generated automatically.
E	STOR CHK	IND	Indicates an error in the storage units.
E	PROC CHK	IND	Indicates an error in the CPU.
F	TEST MODE		
	REPEAT	TGL	For customer engineering use; should be in center position for normal CPU operation.
	ROS-PROC-FLT	TGL	For customer engineering use; the PROC position does not affect CPU operation and is the normal position of this switch.
F	STORAGE INDICATE	TGL	Used on 2067-1 only. Selects the storage arrays (1 through 4 or 5 through 8) that will have information displayed by roller switch indicators.
F	FREQUENCY ALTERATION	TGL	For customer engineering use; is effective only when CE key switch is in CE position.
F	DEFEAT INTERLEAVING	TGL	For customer engineering use; PROC is the normal position.
F	STOP ON STORAGE CHECK	TGL	For customer engineering use; should be in center position for normal operation.
F	DISABLE INTERVAL TIMER	TGL	For customer engineering use; should be in center position for normal operation.
F	DISABLE INTERVAL TIMER	TGL	For customer engineering use; should be in center position for normal operation.
F	STORAGE SELECT	TGL	
	MAIN	-	Normal position; selects main storage for storing or displaying data.
	LOCAL	-	Selects local storage for storing or displaying data.
	MAIN BYTE	-	Same as normal position except that the byte selected is the only byte affected by a manual store operation.
F	ADDRESS COMPARE STOP	TGL	Stops processing when storage address agrees with bits 2 through 20 of address switches. Should be in center position for normal operation.
F	CPU CHECK	TGL	
	PROC	-	This is the normal position. If the PSW machine mask is a 1, the CPU stops on detection of a CPU check, and the status is logged into main storage. If the mask is 0, the result is the same as if the switch is in the DSBL position.
	DSBL	-	The CPU does not stop on detection of a machine check, but the check trigger is set.
	STOP	-	The CPU stops on detection of a machine check, but there is no log-in of data.
F	PULSE MODE	TGL	For customer engineering use; should be in PROC position for normal CPU operation.
F	REPEAT INSN	TGL	For customer engineering use; should be in PROC position for normal CPU operations.
F	Rate	RTY	
	INSN STEP	-	CPU executes one machine instruction for each depression of the START pushbutton.
	PROCESS	-	Does not affect CPU operation; CPU operates at normal clock speed.

Panel	Switch/Indicator Name	Type*	Function
	SINGLE CYCLE	-	CPU advances by minimum clock amount for each depression of the START pushbutton; all CPU operations are as in PROC position.
	SINGLE CYCLE STORAGE INHIBIT	-	Same as SINGLE CYCLE position without storage references.
F	SYSTEM RESET	PB	Resets on-line channels, control units, and CPU controls (including machine checks) to their initial state.
F	CHECK RESET	PB	Resets all CPU and storage check triggers.
F	PSW RESTART	PB	Loads a PSW from main storage address zero and starts processing.
F	ROS TRANSFER	PB	For customer engineering use.
F	SET IC	PB	Enters an address from ADDRESS switches into the active (current) PSW.
F	STORE	PB	Enters data into the storage location specified by the STORAGE SELECT switch and the ADDRESS switches.
F	DISPLAY	PB	Displays data specified by the STORAGE SELECT switch and the ADDRESS switches.
F	START	PB	Starts the CPU operating in the mode selected by the RATE switch.
F	STOP	PB	Terminates CPU operation without changing the environment.
F	RESTART FLT I/O	PB	For customer engineering use.
F	LOGOUT	PB	Stores CPU status in fixed locations in main storage.
F	Elapsed Time Meters	-	Indicate CPU running time. The process meter shows customer elapsed time; the CE meter shows customer engineering elapsed time.
F	Key Switch	-	Determines which elapsed time meter will be used to record time.
G	POWER ON	PB/IND	Initiates power-on in the CPU and in selected system units. It is backlighted.
G	PREFIX SELECT	TGL	Selects either main or alternate prefix during manually-initiated program loading.
G	POWER OFF	-	Initiates power-off in the CPU and in selected system units.
G	LOAD UNIT	RTY	These three switches select the I/O units used by a load operation.
G	INTERRUPT	PB	Causes an external interruption in the system and sets bit 25 of the interruption code to a 1.
G	LOAD	PB	Resets the system and starts a load operation.
G	SYSTEM	IND	Indicates a CPU elapsed time meter is running.
G	MANUAL	IND	Indicates that the CPU as in the stopped state.
G	WAIT	IND	Indicates that the CPU is in the wait state.
G	TEST	IND	Indicates that a switch on panel F is not in the normal operating position or that a channel is in the test mode.
G	LOAD	IND	Indicates that a CPU load operation is in progress. A successful load turns off the indicator.

*Abbreviations:
IND - Indicator
PB - Pushbutton Switch
PS - Pull Switch
RTY - Rotary Switch
TGL - Toggle Switch

Address Translation. The process of replacing the 12 high-order bits (24-bit addressing) of a virtual address with 12 bits of a physical address. For 32-bit addressing situations, the 20 high-order virtual address bits are replaced.

Associative Array. An array of eight associative storage registers used in the dynamic address translation process. Each register contains both a virtual address and its equivalent current physical address.

Bus Control Unit. The logic used to control the flow of information over the several buses attached to processor storage. The BCU controls the address buses as well as the data buses to and from storage. Priority determinations in the use of the buses and associated control signals are handled by the BCU. A BCU is used for channel and CPU communications with processor storage.

Command. One or more uppercase alphabetic terms for which TSS/360 has a predetermined unique response.

Command Language. The set of commands that are recognized by TSS/360 and result in a predetermined set of actions.

Command System. The set of programs required to provide responses to commands.

Conversation. The interaction or dialog between a user and a time-sharing system, via a terminal device, whereby the user can control, interrogate, modify, and observe the processing of his task.

Data Management. A general description of the collective functions of a programming system that provide access to data sets, enforce data storage conventions, and regulate the use of I/O devices.

Data Set. A named collection of logically related data items, arranged in a prescribed manner and described by control information to which the programming system has access.

Direct-Access. A type of storage medium which allows information to be accessed by positioning the medium or accessing mechanism directly to the information required, thus permitting direct addressing of data locations.

Dynamic Address Translation. Address translation that occurs at instruction execution time.

Extended Direct Control. The direct control function as modified for IBM System/360 Model 67-2. The Read Direct instruction is dropped, performance of Write Direct has been modified, and some external signal lines have been assigned specific control functions.

Extended Dynamic Address Translation. Dynamic address translation for 32-bit addresses.

Extended PSW Mode. The CPU mode of operation in which some functions normally controlled by bits of the PSW are assigned to control registers, logically extending the PSW to include those control registers. The Extended PSW mode is controlled by bit 8 of control register 6.

Extended Storage Protection. The seven-bit storage protection feature for IBM System/360 Model 67; includes storage protection, fetch-protection, and two additional bits identified as reference (bit 5) and change (bit 6). Bit 5 of the protection key for the referenced block is set to 1; any store operation causes bit 6 to be set to 1.

High-Resolution Interval Timer. The timer feature used in IBM System/360 Model 67; this timer has a resolution time of 13 usec.

Language Processor. A general term for any assembler, compiler, or other routine that accepts statements in one language and produces equivalent statements in another language.

Library. In general, a collection of objects (for example, data sets or volumes) associated with a particular use, and the location of which is identified in some type of directory.

Linkage. The means by which communication is effected between two programs.

Linkage Editor. A program that produces an object program module by transforming other program modules, optionally combining separate control sections into a single control section, resolving symbolic cross-references among them, replacing, deleting, and adding control sections on request.

Load. The process of reading the beginning of a program into virtual storage and making necessary adjustments and/or modifications to the program so that it may have control transferred to it for the purpose of execution.

Main Storage. Storage which is directly addressed by the registers of a central processing unit; hence, storage from which program instructions may be fetched and executed.

Module (Programming). The input to, or output from, a single execution of an assembler, compiler, or linkage editor; a source or program module; hence, a program unit that is discrete and identifiable in respect to compiling, combining with other units, and loading.

Multiple Access. That characteristic of a system which allows many users to share the total system concurrently, each as though he has sole use of the system.

Multiprocessing. The simultaneous use of two or more processing units in the same computing system.

Object Program Module. The output of a single execution of an assembler, compiler, or link editor, which constitutes input to the dynamic loader or the linkage editor; an object module consists of one or more control sections in relocatable (though not executable) form and an associated program module dictionary.

Off-Line. A generic reference to a device which may not be directly controlled by a computing system with which it is being used.

On-Line. A generic reference to a device which can be directly controlled by a computing system to which it is electrically connected.

On-Line Storage. Storage devices, and, especially, the storage media which they contain under the direct control of a computing system; not the off-line or shelf-storage of these media.

One-Level Storage. A concept which treats all on-line storage, regardless of its physical characteristics, as having one level of appearance to a user; a technique which makes all on-line storage appear as main storage.

Page. A set of 4096 consecutive bytes; applied to main storage, a set of 4096 consecutive bytes, the first byte of which is located at a storage address that is a multiple of 4096 (an address whose 12 low-order bits are 0).

Page Boundary. A TSS/360 storage address that is a multiple of 4096; a System/360 address whose 12 low-order bits are 0.

Page Table. A table containing the physical starting addresses of pages (for one segment) that are in processor storage. The page table argument is the page table starting address plus the virtual page address.

Page Turning. See "Paging."

Paging. The process of transmitting pages of information between main storage and auxiliary storage, especially when done for a purpose of assisting the allocation of a limited amount of main storage among a number of concurrently executing programs.

Partitioning. The division of a multiprocessor configuration into two or more isolated subsystems.

Physical Addresses. Addresses in processor storage.

Privileged. (1) Privileged user: one who is entitled to execute certain system control commands from a terminal; (2) privileged module: a system module that is allowed to communicate with the TSS/360 supervisor; (3) privilege of access: an attribute of the level of allowable sharability of a shared data set (could be read-only access, read/write access, unlimited access, or no access).

Problem Program. Any routine that performs processing of the types for which a computing system is intended; includes routines that solve problems, monitor and control industrial processes, sort and merge records, perform computation, process transactions against stored records, etc; generally interpreted to be any nonsystem program.

Processing Programs. A generic description of programs that have the primary function of processing information and producing results, as compared with supervising and/or scheduling the flow of control through the system.

Program. (N) A generic reference to collections of instructions and data produced for the solution of some well-defined problem; (V) to create and/or produce these collections.

Protection Key. An indicator associated with a task that appears in the program status word (PSW) whenever the task is in control, and that is matched against the storage keys of all storage protection blocks that it is to use.

Real Time. The actual time during which a physical process transpires, especially if that process is monitored or controlled by a computing system.

Re-enterable. An attribute of a program that allows the program to be interrupted during execution, entered by another user, and, subsequently, re-entered at the point of interruption by the first user while producing the desired results for all users; a program with an intermediate state of execution that is totally restorable when it is re-entered after an interruption.

Relocation. The movement of a program from one place in storage to another, including the modification to the program required therefrom.

Response Time. The average time that a terminal user must wait to receive a response from the time-sharing system; a measure of the rate at which the dialog between man and machine takes place.

Routine. A sequence of machine instructions which carry out a well-defined function.

Secondary Storage. The on-line storage of a computing system that is not directly addressed by the registers of a central processing unit.

Segment. A logical grouping of data and procedure; on System/360 Model 67, an area of contiguous virtual storage equal to 256 pages, and on a 256-page boundary.

Segment Table. A table that contains the physical starting addresses of page tables. The segment table argument is the segment number plus the starting address of the segment table.

Storage Key. Indicators associated with storage protection blocks which require that tasks have matching protection keys to use the blocks.

Subroutine. A routine that performs a specific function which is part of the overall objective of the routine that uses it.

Supervisor. The program modules, supplied by IBM, that control and monitor the usage of the time-sharing system.

Supervisory Programs. A generic reference to the programs that have the primary function of scheduling, allocating, and controlling system resources (rather than processing data to produce results).

Task. Any action, and all work undertaken as a result of such action, performed by TSS/360 under the direction of a stream of commands from a system input unit and associated with a particular user between the receipt of the Logon and Logoff commands. Some background tasks are initiated for a user by the system and are terminated by the completion of the operation to be performed.

Throughput. A measure of the rate at which work can be performed by a computing system.

Time-Sharing. A method of using a computing system whereby a number of users can concurrently execute programs with which the users may interact during execution and be generally assured some minimum amount of program execution per unit time.

Time-Slice. The time during which each task executes before it is placed in a queue of tasks that are contending for the system resources.

TSS/360. The time-sharing operating system (a supervisory program) for use with the IBM System/360 Model 67.

Turnaround Time. The elapsed time between submission of a task to a computing center and the return of results.

User. Anyone who uses the services of a computing system.

Virtual. Conceptual or appearing to be, rather than actually being.

Virtual Address. An address generated by a program which references virtual or conceptual storage and must, therefore, be translated into a real storage address when it is to be used.

Virtual Storage. A conceptual form of main storage which does not really exist but is made to appear as though it exists (through the use of hardware and programming).

Wait Condition. As applied to tasks, the condition of a task that is dependent on events to enter the ready condition.

Index

READER'S COMMENT FORM

IBM System/360 Model 67

GA27-2719-2

- How did you use this publication?

  As a reference source ☐
  As a classroom text ☐
  As..... ☐

- Based on your own experience, rate this publication . . .

  As a reference source:

Very Good	Good	Fair	Poor	Very Poor

  As a text:

Very Good	Good	Fair	Poor	Very Poor

- What is your occupation? .....

- We would appreciate your other comments; please give specific page and line references where appropriate. If you wish a reply, be sure to include your name and address.

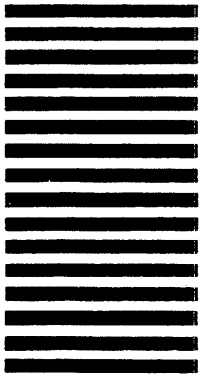
- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

Your comments, please . . .

Your comments, together with your answer to the questions on the back of this form will help us produce better publications for your use. Your comments will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Fold Fold

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

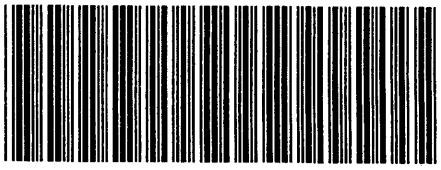
POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 520
Neighborhood Road
Kingston, New York 12401

Fold Fold

IBM

GA27-2719-02



Cut or Fold Along Line

IBM System/360 Model 67 Printed in U.S.A. GA27-2719-2

IBM System/360 Model c. Printed in U.S.A. GA27-2719-2

IBM