Systems

IBM System/370 Model 165
Functional Characteristics

IBM

Systems

IBM System/370 Model 165 Functional Characteristics

This publication describes the organization and the functional characteristics of the IBM System/370 Model 165, an information-processing system designed for very high-speed, large-scale scientific and business applications.

The system components are described, and a detailed consideration is given to the functions of processor storage, the central processing unit, the input/output channels, and the operator-control and operator-intervention portions of the system control panel. In addition, certain coding and timing considerations are described.

The reader is assumed to have a knowledge of information-processing systems and to have an understanding of the System/360, as that system is described in the *IBM System/360 Principles of Operation,* Order No. GA22-6821. The *IBM System/370 Principles of Operation,* GA22-7000, should be used in conjunction with the *IBM System/360 Principles of Operation.*

Preface

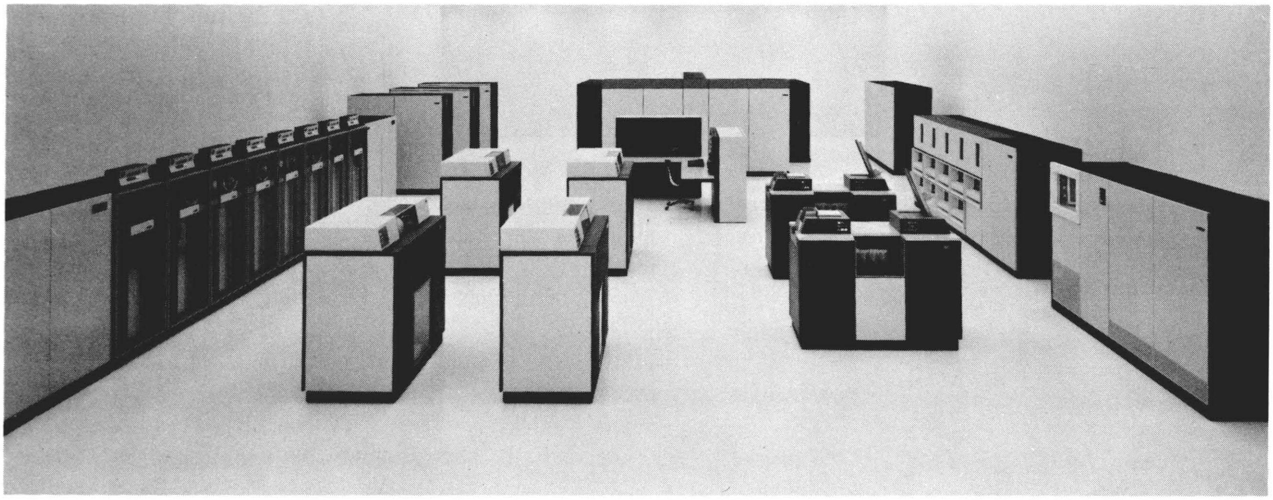
This manual is a general description of the capabilities and characteristics of the Model 165. It provides management, computer operators, systems engineers, customer engineers, programmers, and computer science students with a fundamental understanding of the System/370 Model 165. *IBM System/360 Principles of Operation,* GA22-6821, and *IBM System/370 Principles of Operation,* GA22-7000 are prerequisite reading.

First Edition (June 1970)

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

This manual has been prepared by the IBM Systems Developement Division, Product Publications, Dept. B98, PO Box 390, Poughkeepsie, N.Y. 12602. A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be sent to the above address.

Contents



IBM System/370 Model 165

The IBM System/370 Model 165 is an upwards compatible information-processing system designed for very high-speed, large-scale scientific and business applications. Its speed and power result primarily from the use of high-speed logic circuitry, the achievement of very fast access and execution times, the realization of a high degree of concurrency in operations, and the employment of highly efficient algorithms, particularly in fixed-point and floating-point operations.

Contributing greatly to the speed and power of the Model 165 are two standard features in the CPU: a high-speed buffer that holds currently used sections of processor storage for faster accessing, and an extended-precision floating-point feature that provides for arithmetic operations on 16-byte, floating-point operands including rounding from extended to long format and from long to short. Speed in the accessing of processor storage is further increased by the use of multiple, interleaved storage elements.

Increased performance is offered by the Model 165 through the addition of new instructions to handle character and variable length data, the expansion of the control capabilities of the PSW by means of control registers, and the use of a high resolution time-of-day clock.

The performance of the System/370 Model 165 is increased still further by high-speed multiply, an optional feature that allows both fixed-point and floating-point multiply instructions to be performed faster.

To provide more efficient control of instruction execution, conventional logic has been replaced by microprograms stored in the control storage. Control storage consists of a writable control storage (WCS) and a read-only storage (ROS) which define the state of the CPU at any given instant.

Reliability, availability, and serviceability are greatly enhanced by reducing unscheduled interruptions through the use of instruction retry, processor storage error checking and correction (ECC), and manual storage reconfiguration (isolation of an erratic storage unit while maintaining system operation). Programming support is provided by Operating System/360 (OS): the system is planned to take advantage of the performance gains possible because of this support, particularly when multiprogramming is used. The OS options, multiprogramming with a fixed number of tasks (MFT), and multiprogramming with a variable number of tasks (MVT), provide for efficient use of the large storage capacities available to the system.

In the Model 165, separate units, each mainly autonomous, may be operating concurrently: processor storage, the instruction preparation unit, the execution unit, and the channels. Data flow and system statistics are shown in Figure 1. Although processing operations are performed in the Model 165 in an overlapped fashion, no special optimization is required in preparing programs for CPU processing.

The Model 165 has a CPU machine-cycle time of 80 nanoseconds. Processor storage data flow is eight bytes (a doubleword) in parallel. Processor storage cycle time is two microseconds. However, through use of the high-speed buffer storage the effective system storage cycle is reduced to a fraction of the actual processor storage cycle.

Five capacities of processor storage are available: Model I (524,288 bytes), Model J (1,048,576 bytes), Model JI (1,572,864 bytes), Model K (2,097,152 bytes), and Model KJ (3,145,728 bytes). A special error checking and correction code is implemented in the storage units.

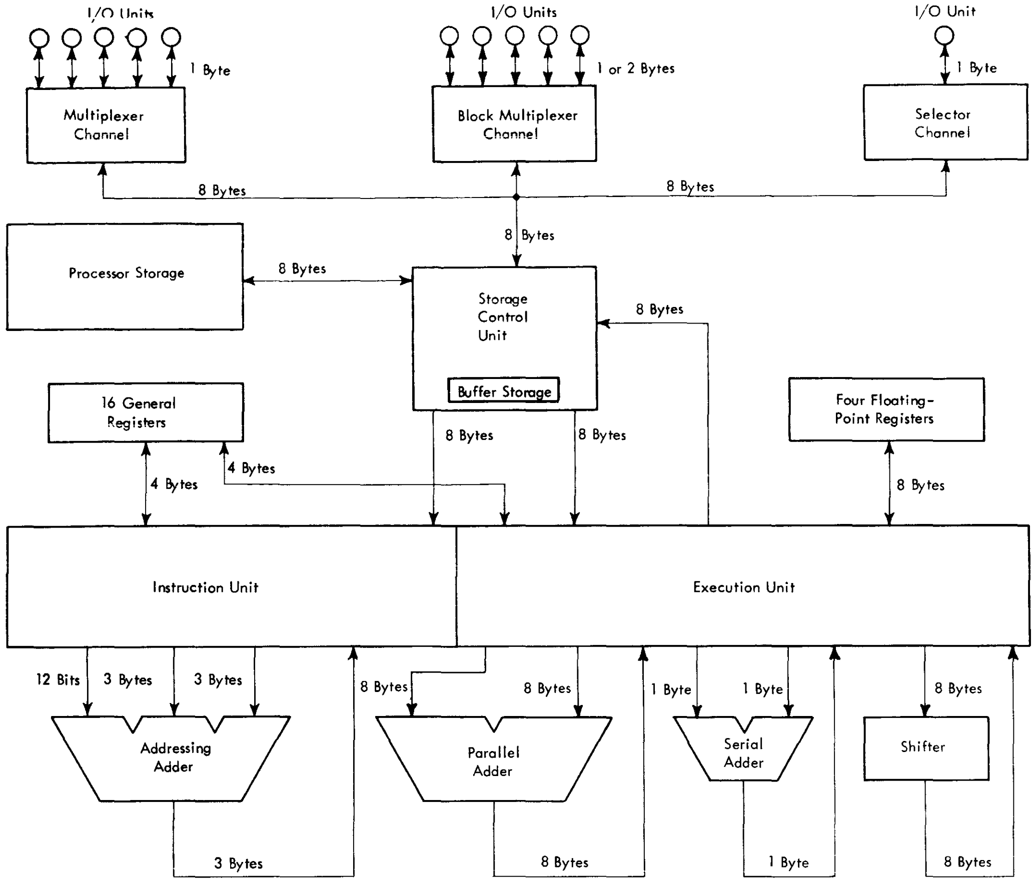
For input/output operations, the system may connect (with the extended channel feature) a maximum of 12 channels: one or two multiplexer channels (IBM 2870 Multiplexer Channel), as many as six selector channels (IBM 2860 Selector Channel), or up to 11 Block Multiplexer channels (IBM 2880 Block Multiplexer Channel). These channels use channel data buffers in the CPU to achieve increased aggregate data rates through the effective use of the four-way interleaved processor storage.

SYSTEM COMPONENTS

The major components of a Model 165 System are an IBM 3165 Processing Unit, Processor Storage, a System Console, either an IBM 2860 Selector Channel, or an IBM 2870 Multiplexer Channel, or an IBM 2880 Block Multiplexer Channel. Input and output (I/O) devices are attached to the channels by control units. (See Figure 2).

The standard features for an IBM System/370 Model 165 include:

Universal instruction set
Extended-precision floating-point feature
Byte-oriented operand feature
Buffer storage (8,192 bytes)
Direct control feature
2860 Selector Channel attachment
2870 Multiplexer Channel attachment
2880 Block Multiplexer Channel attachment
Writable control storage



Element	Data Width (Bytes)	Performance	Comments
Processor storage	8	Two-microsecond cycle (Note 1)	- - - -
Buffer storage	8	80-nanosecond cycle	8,192 byte capacity (Note 3)
Basic machine cycle	-	80 nanoseconds	- - - -
General registers	4	Once per machine cycle	16 general registers
Floating-point registers	8	Once per machine cycle	4 floating-point registers
Addressing adder	3	Once per machine cycle	- - - -
Parallel adder	8	Once per machine cycle	- - - -
Serial adder	1	Once per machine cycle	- - - -
2860 Selector Channel	1	1.3 million bytes per second	8 bytes to storage
2870 Multiplexer Channel	1	110 kb to 670 kb (aggregate)	8 bytes to storage
Burst mode	1	110 kb (Note 2)	- - - -
Multiplex mode	1	110 kb (Note 2)	- - - -
Selector subchannels 1-3	1	180 kb each	- - - -
Selector subchannel 4	1	100 kb (kb = 1000 bytes per second)	- - - -
2880 Block Multiplexer Channel	1	1.5 million bytes/second (Note 4)	8 bytes to storage

Notes:

1. Processor storage effective cycle time is greatly reduced by the buffer storage function.
2. Aggregate 192-subchannel rate for first or second 2870; reduced by concurrent selector subchannel operation.
3. May be expanded to 16,384 bytes.
4. Three million bytes per second when optional two-byte interface is used.

Figure 1. Model 165 Data Flow and System Statistics

System console with operator console including visual display
Interval timer (updated each 3.33 milliseconds)
Processor storage error checking and correction
Instruction retry
Time-of-day clock (updated each microsecond)
Control registers
Additional character and variable length data handling instruction

The extended-precision floating-point feature includes instructions designed to handle extended-precision (28-digit fraction) floating-point operands. Extended-precision operands may also be rounded to long-precision format, and long-precision operands may be rounded to short-precision format.

The byte-oriented operand feature allows the user to ignore, in part, the restriction that all operands in processor storage must be at addresses that are integral multiples of the operand length. The user that takes advantage of this feature can reference fixed-point, floating-point, and logical operands of unprivileged instructions on any byte boundary by RX and RS format instructions.

The operation performed when the byte-oriented operand feature is used is called boundary alignment.

Programming Note: Boundary alignment causes instruction processing to proceed at less than optimal speed. Performance degradation results when operand boundaries are unaligned (see "Instruction Times "). In addition, the System/360 Operating System Control Program continues to check for boundary alignment of operands passed in parameter lists to its various modules. Violation causes termination.

Details about the extended-precision floating-point and about the byte-oriented operand features are given in the *IBM System/370 Principles of Operation,* GA22-7000.

The time-of-day clock provides a precise measure of time suitable for accurate elapsed time measurements and time-of-day indication. Operation of the clock is continuous and is not affected by any activity in the system other than a power failure. Its binary value, updated each microsecond, may be interrogated or set by the appropriate instructions provided (see "Time- of-Day Clock" in *IBM System/370 Principles of Operation,* GA22-7000).

Four of sixteen control registers (CR's) are implemented in the Model 165. They are CR 0, 2, 14, and 15 having 32 bit capacities each. Their function is to extend the PSW-type of control over the system. The control function of each bit is defined in the *IBM System/370 Principles of Operation,* GA22-7000.

The new instructions provided are:

Shift and Round Decimal (SRP)
Insert Characters Under Mask (ICM)
Store Characters Under Mask (STCM)
Compare Logical Characters Under Mask (CLM)
Move Characters Long (MVCL)
Compare Logical Long (CLCL)
Load Control (LCTL)
Store Control (STCTL)
Set Clock (SCK)
Store Clock (STCK)
Store CPU ID (STIDP)
Store Channel ID (STIDC)
Start I/O Fast Release (SIOF)

A detailed definition of each instruction may be found in *IBM System/370 Principles of Operation,* GA22-7000.

The integrated operator console provides facilities through which the operator may enter data into the system, obtain visual output, be alerted by an audible alarm from the system, and present an attention signal to the system.

The integrated operator console links to the system console. The cathode-ray tube (CRT) display on the main control panel (Figure 3) is shared by the integrated operator console. Features of the integrated operator console are an alphameric keyboard (on the reading board in front of the main control panel), a character generator, a display buffer, and an audible alarm.

The optional features for an IBM System/370 Model 165 include:

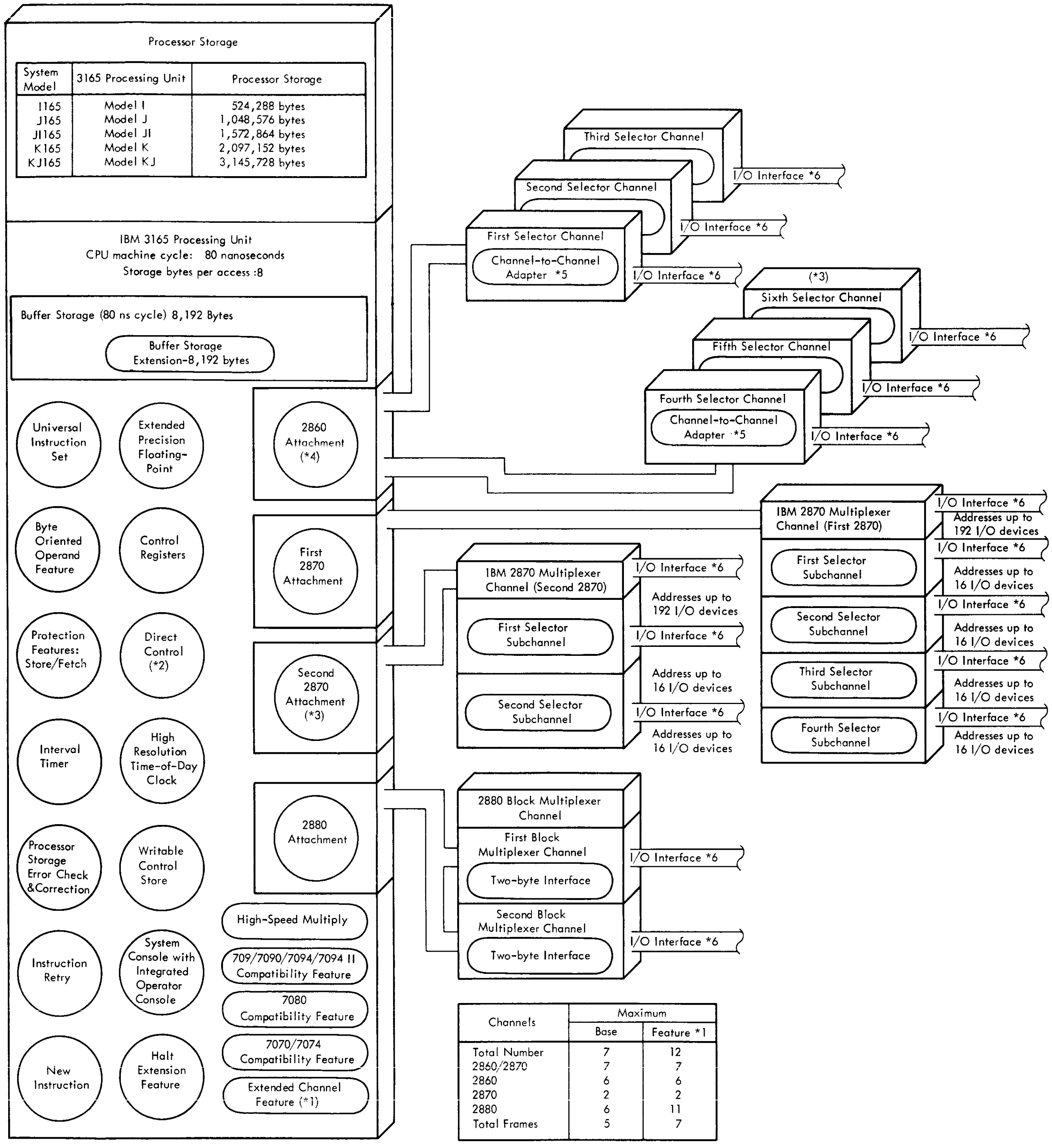
Buffer storage extension to 16,384 bytes
High-speed multiply feature
Extended channel feature
7070/7074 compatibility feature
7080 compatibility feature
709/7090/7094/7094II compatibility feature

Buffer storage extension expands the basic 8,192 byte buffer storage capacity by 8,192 bytes.

The use of the high-speed multiply feature allows both fixed - point and floating-point multiply instructions to be performed faster. The basic CPU does a floating-point long-precision multiply in about 1,870 nanoseconds and a fixed-point multiply in about 780 nanoseconds, with the high-speed multiply feature installed, the times required for the two operations are 610 and 420 nanoseconds, respectively.

The extended channel feature provides for attaching five additional channels to the 165 system; a total of 12 channels (selector, multiplexer, and block multiplexer) is permitted.

The compatibility features, 7070/7074, 7080, and 709/7090/7094/7094II, in conjunction with the appropriate emulator programs, allow the Model 165 to execute programs and programming systems originally written for other systems. The compatibility feature adds special instructions and internal logic. Emulator programs use these facilities and the universal instruction set to simulate 7070/7074,7080, or 7090-type instructions under the System/360 Operating System.


Processor Storage
IBM 3165 Processing Unit
CPU machine cycle: 80 nanoseconds
Storage bytes per access :8
Buffer Storage (80 ns cycle) 8,192 Bytes
Buffer Storage Extension-8,192 bytes
Universal Instruction Set
Extended Precision Floating-Point
Byte Oriented Operand Feature
Control Registers
Protection Features: Store/Fetch
Direct Control (*2)
Interval Timer
High Resolution Time-of-Day Clock
Processor Storage Error Check &Correction
Writable Control Store
Instruction Retry
System Console with Integrated Operator Console
New Instruction
Halt Extension Feature
2860 Attachment (*4)
First 2870 Attachment
Second 2870 Attachment (*3)
2880 Attachment
High-Speed Multiply
709/7090/7094/7094 II Compatibility Feature
7080 Compatibility Feature
7070/7074 Compatibility Feature
Extended Channel Feature (*1)
Third Selector Channel
Second Selector Channel
First Selector Channel
Channel-to-Channel Adapter *5
I/O Interface *6
(*3)
Sixth Selector Channel
Fifth Selector Channel
Fourth Selector Channel
Channel-to-Channel Adapter *5
IBM 2870 Multiplexer Channel (First 2870)
Addresses up to 192 I/O devices
First Selector Subchannel
Second Selector Subchannel
Third Selector Subchannel
Fourth Selector Subchannel
Addresses up to 16 I/O devices
IBM 2870 Multiplexer Channel (Second 2870)
Addresses up to 192 I/O devices
First Selector Subchannel
Address up to 16 I/O devices
Second Selector Subchannel
Addresses up to 16 I/O devices
2880 Block Multiplexer Channel
First Block Multiplexer Channel
Two-byte Interface
Second Block Multiplexer Channel
Two-byte Interface

System Model	3165 Processing Unit	Processor Storage
I165	Model I	524,288 bytes
J165	Model J	1,048,576 bytes
JI165	Model JI	1,572,864 bytes
K165	Model K	2,097,152 bytes
KJ165	Model KJ	3,145,728 bytes

Channels	Maximum	
	Base	Feature *1
Total Number	7	12
2860/2870	7	7
2860	6	6
2870	2	2
2880	6	11
Total Frames	5	7

Notes:

Indicates Standard Feature

Indicates Optional Feature

*1. The Extended Channel Feature permits connection of five additional/2880 channels; see chart above.

*2. Read/Write Direct and External Interrupt.

*3. A system with the first 2870 may attach up to six selector channels; a system with the first and second 2870 may attach up to five selector channels.

*4. Each selector channel addresses up to 256 I/O devices, one at a time. Operation is in burst mode with overlapped processing. The IBM 2860 Selector Channel is available in three models: Model 1 provides one channel, Model 2 provides two channels, and Model 3 provides three channels in one unit. The system attaches a maximum of two units and a minimum of one unit, or one 2870 with the optional selector subchannel feature installed, or one 2880 Block Multiplexer Channel.

*5. A Channel-to-Channel Adapter option (one per 2860 channel) permits interconnection of two channels. One channel position can connect to one channel position on any other IBM System/370 channel. Only one Channel-to-Channel Adapter is needed per connection; it counts as one control unit on each channel.

*6. Up to eight control units may be attached. Input/Output control units and devices are shown on the IBM System 360/370 Input/Output Configurator, GA22-6972

Figure 2. System/370 Model 165 Configurator

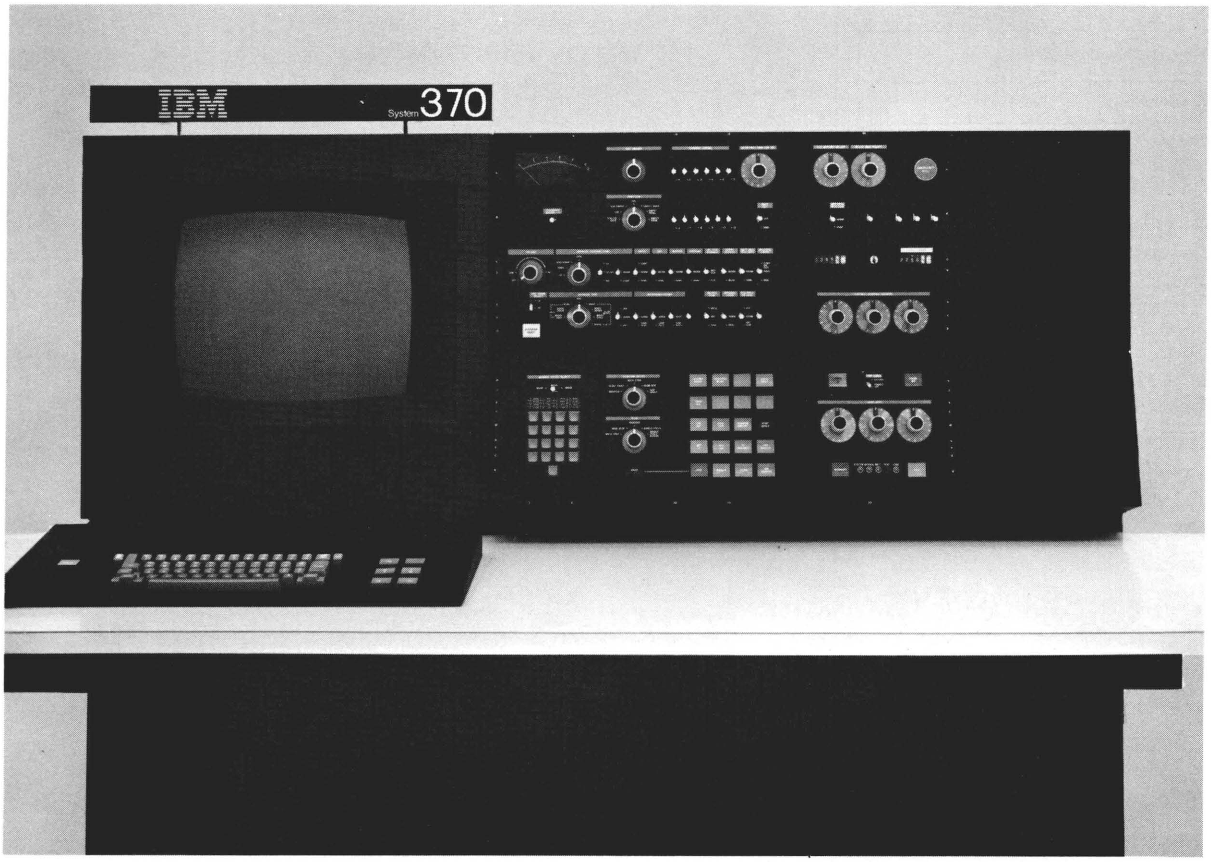


Figure 3. Model 165 System Console

PROCESSOR STORAGE

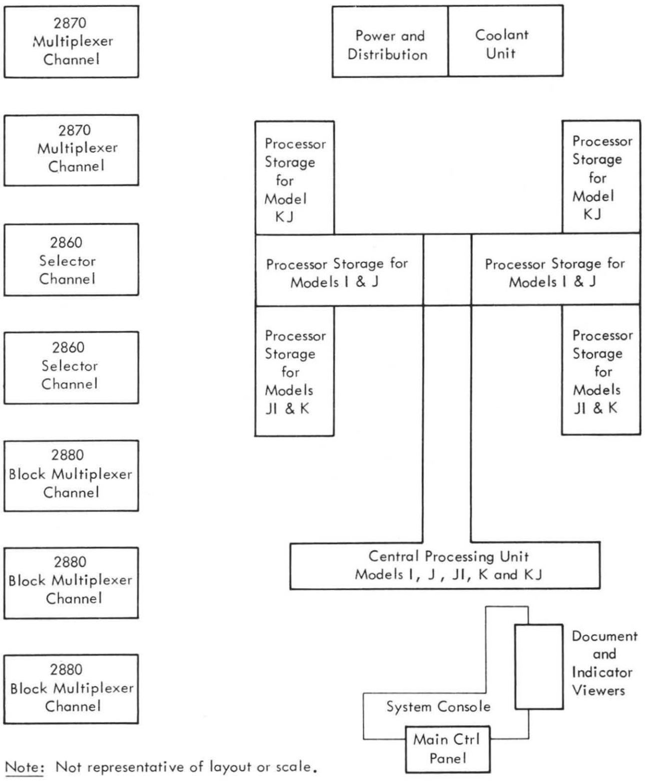
The Model 165 has five possible CPU/processor-storage combinations. These combinations vary in the number of core storage units used in the Processing Unit's processor storage. (In this publication, main storage and processor storage are used interchangeably.) See Figure 4.

CPU Model	*Processor Storage Capacity (bytes)*	*Interleaving*
3165I	524,288	Four-way
3165J	1,048,576	Four-way
3165JI	1,572,864	Four-way
3165K	2,097,152	Four-way
3165KJ	3,145,728	Four-way

Interleaving

Interleaving allows processor storage units to operate independently in an overlapped manner for effective reduction of the storage-cycle time. Storage performance is particularly improved in sequential-address accessing.

In four-way interleaving, four functionally independent storage units, each providing eight bytes per storage access, make up processor storage. Assume the four units to be: 0, 1, 2, and 3. Storage location 0 is in unit 0, location 8 is in unit 1, location 16 is in unit 2, and location 24 is in unit 3. Storage location 32 is in unit 0,


2870 Multiplexer Channel
2870 Multiplexer Channel
2860 Selector Channel
2860 Selector Channel
2880 Block Multiplexer Channel
2880 Block Multiplexer Channel
2880 Block Multiplexer Channel
Power and Distribution
Coolant Unit
Processor Storage for Model KJ
Processor Storage for Model KJ
Processor Storage for Models I & J
Processor Storage for Models I & J
Processor Storage for Models JI & K
Processor Storage for Models JI & K
Central Processing Unit Models I, J, JI, K and KJ
Document and Indicator Viewers
System Console
Main Ctrl Panel

Note: Not representative of layout or scale.

Figure 4. Model 165 System Elements

and the address-distribution sequence continues through all available storage locations (Figure 5).

An attempt to reference a processor storage location (unit) may be made during any CPU cycle. Actually, a storage reference can be made only during any CPU cycle during which the functionally independent storage unit containing the requested location is not busy. (A storage unit is defined as busy when it has not completed a storage cycle after being selected.) The storage unit, once selected for a storage reference, cannot be referenced again until the total storage-cycle time passes. The storage-cycle time corresponds to 25 CPU cycles. Therefore, a processor storage consisting of one storage unit would allow storage references to be made only once every 25 CPU cycles. However, the 165's processor storage with four-way interleaving allows up to four storage references each 25 CPU-cycle period: one access per storage unit.

Error Checking and Correction

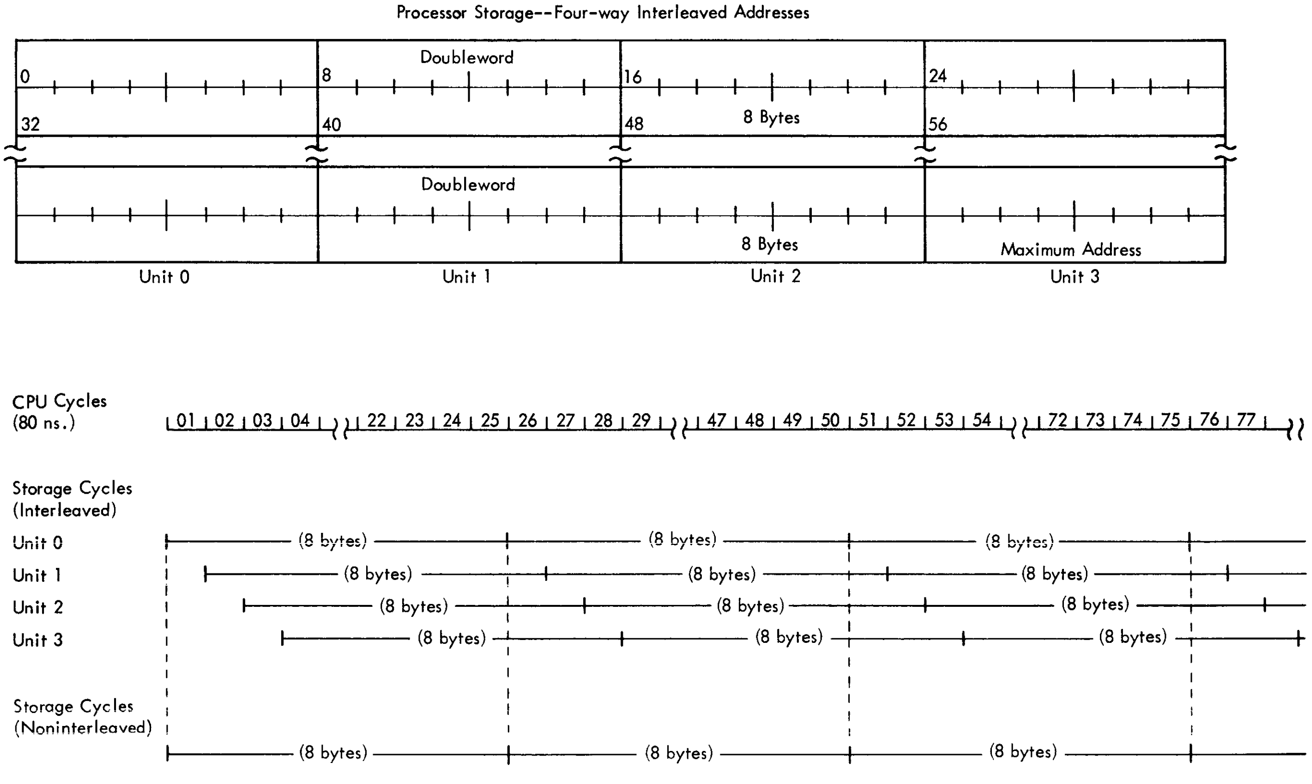
The error checking and correction function used by the 165 processor storage reduces the number of system interruptions that are due to single-bit parity errors.

When data is stored, an error-checking code is substituted for the parity bits. When data is fetched, a validity check is performed, and if a single- bit error is detected, it is corrected. Valid data is transmitted to the system, thereby allowing program execution to continue uninterrupted.

Most multiple-bit errors are signaled to the system for processing by the program. Optionally, single-bit errors may be counted and recorded only when a machine check interruption is taken, or each error may be allowed to cause an interruption for recording.

IBM 3165 CENTRAL PROCESSING UNIT

The IBM 3165 central processing unit (CPU) and the processor storage for the Model 165 are attached to form a single unit. The physical complex that contains the CPU also includes a stand-alone power and coolant distribution unit and the system console, which includes a main control panel with CRT display, a microfiche document viewer for examining reference material pertinent to system operation, and an indicator viewer consisting of a microfiche-type projection display. The system console also comprises an integrated operator console that includes a keyboard and shares the use of the CRT display with the maintenance function. A motor-generator set is also required and may be in an area other than the machine room.



Note: A single processor storage access (fetch or store) may be started during any CPU cycle in which the addressed processor storage unit is not busy. Data may be stored or fetched from processor storage at the maximum rate of 32 bytes per two-microsecond period (16 million bytes per second), as contrasted with 8 bytes per two-microsecond period (4 million bytes per second), noninterleaved.

Figure 5. Cycle Time in Four-Way Interleaved Processor Storage

Functionally, the CPU consists of an instruction unit, an execution unit, a storage control unit, processor storage, a high-speed buffer storage, and control storage.

The operations of the instruction unit and the execution unit are overlapped, allowing the execution of instructions to proceed while the instruction unit prepares for later operations.

The execution unit, controlled by microprograms, executes instructions one at a time in program sequence.

The instruction unit is controlled by logic circuits and can prepare several instructions concurrently.

Instruction Unit

The primary functions of the instruction unit are the fetching, decoding, and buffering of instructions, the calculation of addresses, the fetching of required operands, and the issuance of instructions to the execution unit. The required operands are fetched into two operand buffers in the execution unit.

The instruction unit contains two 16-byte instruction buffers, a four-byte instruction register, three instruction queue registers, a 24-bit three-input adder, four 24-bit address registers, an incrementer, and a decoder.

Instruction Buffering

Two 16-byte registers are provided in the instruction unit for buffering prefetched instructions. These registers are the main instruction buffer and the auxiliary instruction buffer.

Instruction Decoding

A four-byte instruction register is provided in the instruction unit to hold each instruction during decoding. The main instruction buffer provides the input. Each instruction processed by the instruction unit remains in the instruction register for at least one machine cycle. During this time, the instruction is identified and the availability of facilities needed for further processing is checked. If the facilities are not available, the instruction is held for another machine cycle and tried again (Figure 6).

Three queue registers are provided in the instruction unit. When decoded, an instruction is buffered in a queue register until it is transferred to the execution unit.

Certain instruction-stream-dependent conditions may delay decoding:

1. The nature of this instruction disallows overlap of the instruction and execution unit operation.

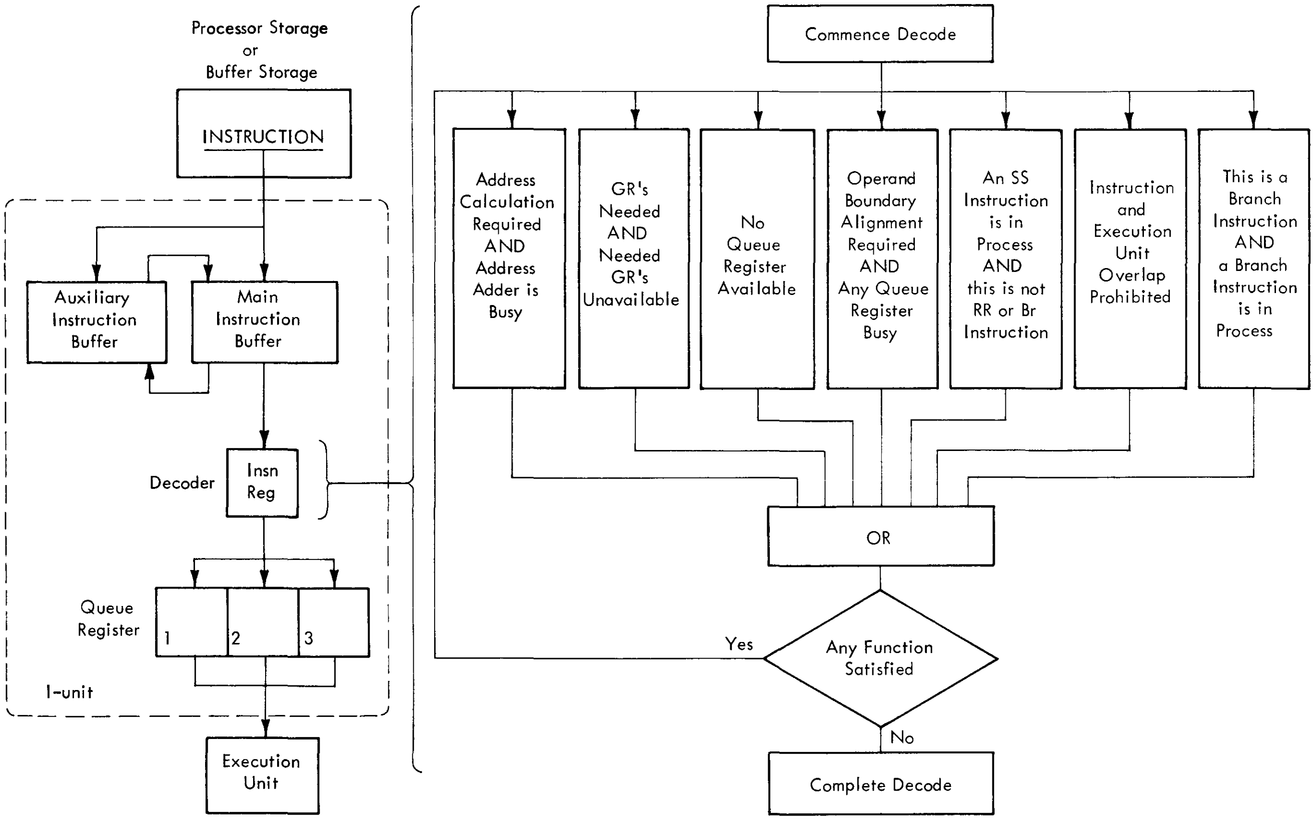

Processor Storage
or
Buffer Storage
INSTRUCTION
Auxiliary Instruction Buffer
Main Instruction Buffer
Decoder
Insn Reg
Queue Register
1
2
3
I-unit
Execution Unit
Commence Decode
Address Calculation Required AND Address Adder is Busy
GR's Needed AND Needed GR's Unavailable
No Queue Register Available
Operand Boundary Alignment Required AND Any Queue Register Busy
An SS Instruction is in Process AND this is not RR or Br Instruction
Instruction and Execution Unit Overlap Prohibited
This is a Branch Instruction AND a Branch Instruction is in Process
OR
Any Function Satisfied
Yes
No
Complete Decode

Figure 6. Instruction Decoding

2. A branch instruction cannot be in progress if this instruction is to be a branch.
3. If an SS-format instruction is in progress, only branch and RR-format instructions can be decoded.
4. The required registers, queue, or adder must be available. (Figure 6).

Address Calculations and Operand Fetches

A three-byte, three-input adder is provided in the instruction unit for performing address calculations. When an instruction is successfully decoded, the required data is sent to the adder and the calculation takes place on the next cycle.

For an instruction indicating a store, the result of the calculation is used as a destination address. For an instruction indicating a fetch, the result of the calculation is used as a source address. For shift and I/O instructions, the results of the calculation either specify the number of bit positions to be shifted or specify the channel and I/O device address to be selected.

For branch instructions, the calculated address is the address of an instruction stream. Instruction fetching begins, and the way the 16-byte instruction buffers are used depends on the estimate made during the decode cycle about the success of that branch.

For branches that are estimated to be unsuccessful, the target instruction and successive instructions in that stream are put into the auxiliary instruction buffer. Instructions from the current stream remain in the main instruction buffer, from which instructions continue to be processed, and additional requests for instructions are made from the current stream as required.

For branches that are estimated to be successful, the contents of the buffers for the current and target streams are switched. If the estimate is correct, when the branch is executed the only action required is to stop fetching instructions into the auxiliary instruction buffer, which contains instructions from the current stream. However, if the estimate is incorrect, the contents of the buffers must be switched again. The second switching, if required, is done during the cycle after the branch execution (Figure 7).

Unsuccessful branch estimates are made for:

Branch on condition (BC) where M is not equal to 0 or 15.
Branch on condition (BCR) where M is not equal to 0 or 15 and R2 is not equal to 0.

Successful branch estimates are made for:

Branch on condition (BC) where M equals 15.
Branch on condition (BCR) where M equals 15 and R2 is not equal to 0.
Branch and link (BAL).
Branch and link (BALR) where R2 is not equal to 0.
Branch on count (BCT).
Branch on count (BCTR) where R2 is not equal to 0.
Branch on index low or equal (BXLE).
Branch on index high (BXH).

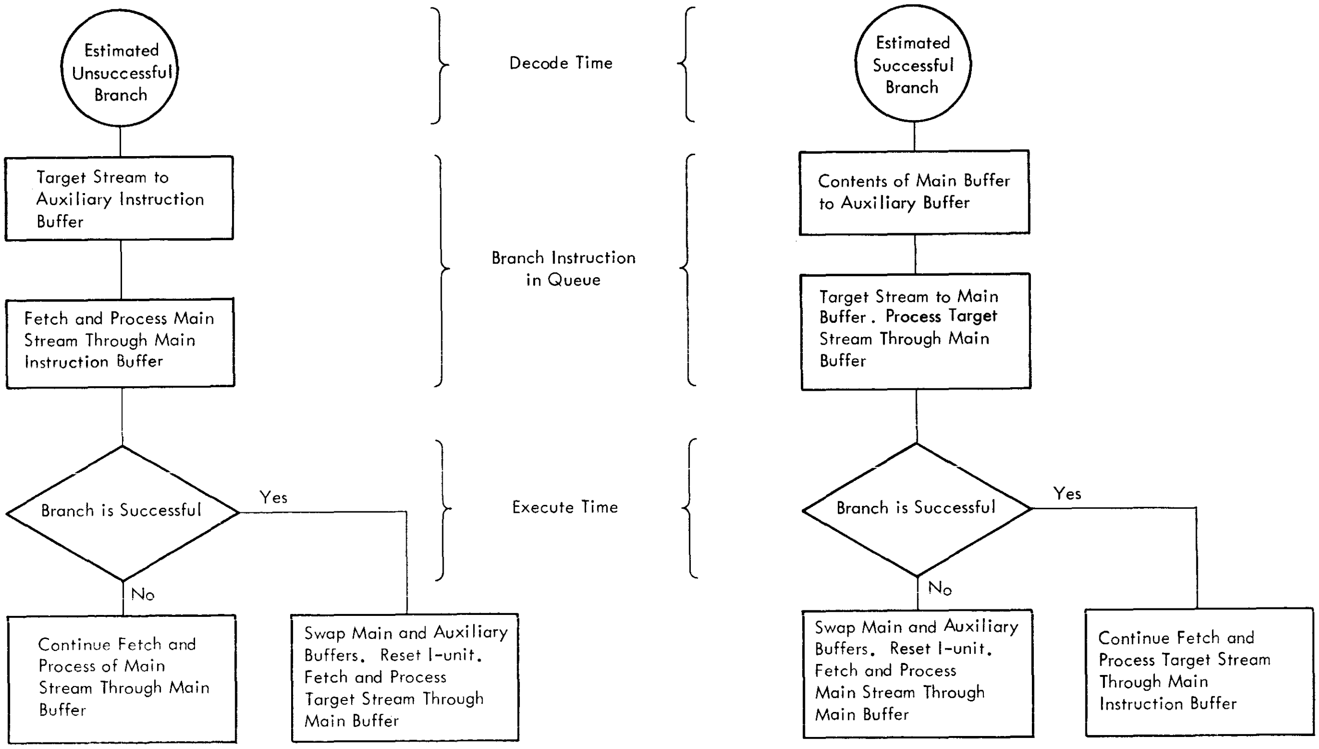

Estimated Unsuccessful Branch
Decode Time
Estimated Successful Branch
Target Stream to Auxiliary Instruction Buffer
Branch Instruction in Queue
Contents of Main Buffer to Auxiliary Buffer
Fetch and Process Main Stream Through Main Instruction Buffer
Target Stream to Main Buffer. Process Target Stream Through Main Buffer
Branch is Successful
Yes
Execute Time
Branch is Successful
Yes
No
No
Continue Fetch and Process of Main Stream Through Main Buffer
Swap Main and Auxiliary Buffers. Reset I-unit. Fetch and Process Target Stream Through Main Buffer
Swap Main and Auxiliary Buffers. Reset I-unit. Fetch and Process Main Stream Through Main Buffer
Continue Fetch and Process Target Stream Through Main Instruction Buffer

Figure 7. Processing a Branch

The following are treated as no-ops and are not recognized as branches:
Branch on condition (BC) where M equals 0.
Branch and link (BALR) where R2 equals 0.
Branch on count (BCTR) where R2 equals 0.

Instruction Handling for Execute

The execute (EX) instruction is handled entirely in the instruction unit. The subject instruction is treated as a branch that is estimated to be successful and found to be unsuccessful.

After all instructions that precede the execute instruction are processed by the execution unit, the instruction unit processes one instruction from the target stream. The current and target streams are then put back to normal and processing continues.

SS-Format Instruction Handling

In many respects, the instruction unit handles each SS-format instruction as two or three separate instructions. There are multiple decode cycles (three for logical and two for decimal) and multiple address calculation cycles. The instruction unit generates the starting addresses of both operands and fetches a doubleword from each address.

For the logical instructions, base and displacement are added to produce an address that points to the leftmost byte of the field. For the decimal instructions, base, displacement, and length are added to produce an address that points to the rightmost byte of the field.

When the execution unit begins the instruction, the two doublewords fetched are ingated to working registers. If the required data is not available, processing stops until the data becomes available.

For translate and translate and test instructions, the source fetch is not made.

Instruction Sequence Protection

When a store takes place, a check is made to determine whether any instructions that have been prefetched into the instruction unit are affected. A comparison is made between the destination address and each of the instruction address registers; a positive indication is given if the destination address plus 0, plus 16, or plus 32 equals the contents of the instruction address register. (The four low-order bits are not considered.) A positive indication, once found, is remembered until the end of the instruction. When the instruction is complete, the I-unit is reset and started again at the next available instruction. In this way, all instructions following the modifying store are fetched again after the store is completed.

Execution Unit Setup

When an instruction is passed from the instruction unit to the execution unit, the execution unit is set up with the values needed to begin the instruction. Setup includes making the instruction and required operands available to the execution unit, and readying the microprogram controls.

The conditions for passing a new instruction to the execution unit are that the instruction unit has an instruction ready and that the execution unit is not busy.

Instruction Unit Interruption Handling

If a fetch request is made to an invalid or protected processor storage address, the error (addressing or protection, respectively) is recognized during the storage operation. Because of overlapping instruction unit and execution unit operations, the interruption cannot be taken immediately. It is buffered by the instruction unit until the offending instruction or its operand is sent to the execution unit; a signal is then sent to the execution unit interruption controls. This is a precise interruption, that is, the instruction length code can be set (nonzero) because the instruction being executed is the one that caused the interruption.

A protection or addressing exception resulting from a store operation, however, produces an imprecise interruption. The error is detected during the storage operation, which is performed by the storage control unit. This occurs some time after the execution of the instruction requesting the store operation. The cause of the interruption is therefore not the instruction presently being executed, thus the instruction length code is unknown and must be set to zero.

When the instruction length code in the program old PSW is 0, 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 interruptions are referred to as imprecise exceptions. A program interruption associated with a nonzero instruction length code, and the corresponding exception, are referred to as precise.

By definition, an imprecise program exception can occur only on an attempt to store at an invalid address or at a protected location.

Execution Unit

The execution unit, which handles the execution of the instructions for the Model 165, has the capability of processing a new instruction every cycle.

The execution unit contains two eight-byte data buffers for prefetched operands, four eight-byte data registers, and an eight-byte result register. Arithmetic and logical functions are done with a 64-bit parallel adder, a 32-bit logical unit, a 64-bit shifter, and an eight-bit serial adder.

The shifter can perform a right or left shift of as many as 63 bit positions in one machine cycle. The serial adder is used to execute SS-format instructions and is also used in overlapped floating-point exponent calculations.

The basic data path within the execution unit passes eight bytes in parallel, with parity checking provided for each byte.

The execution unit is primarily controlled by microprograms. In several cases where data results determine the execution sequence, non-microprogram control is used. In addition, the instruction unit controls those portions of the execution unit required for establishing the initial conditions for instruction execution (such as those used for preferred operands and decoded operation codes).

The basic timing of the execution unit uses the 80-nanosecond machine cycle. The basic data transfer during execution consists of gating a register through the parallel adder, shifter, or serial adder, to a register in one cycle.

Use of Local Storage

Local storage is shared by the instruction unit and the execution unit to set up and execute instructions. Local storage contains 16 four-byte general registers and 4 eight-byte floating-point registers. To provide for complete instruction and execution unit overlap and the ability to process an instruction in one cycle, it is possible in the Model 165 to fetch from four general registers simultaneously and to store into a fifth general register in the same cycle. This condition allows the operand storing for an instruction to be overlapped with the next instruction execution, which may require one or two general registers. At the same time, the instruction unit is calculating an effective address, which may also require one or two general registers. In addition, two short-precision or long-precision, floating-point operands may be accessed simultaneously.

Storage Control Unit

The units of processor storage in the CPU are functionally connected to the system through the storage control unit. All references to storage from the CPU and channels are controlled by the storage control unit. The amount of information sent to or fetched from storage in one reference, sometimes called the physical storage word, is one doubleword (8 bytes).

The storage control unit can initiate a new storage cycle every machine cycle (80 nanoseconds) until all four processor storage units are busy.

Three different logical areas make up the storage control unit: the buffer storage control (which includes the high-speed buffer storage), the I/O channel control, and the processor storage control.

Buffer Storage Control

The buffer storage control handles all storage requests from the CPU for data stores or fetches. It also monitors all channel store operations so that the high-speed buffer storage can be kept updated.

For all CPU fetch operations, a check is made to determine whether the data referenced is in buffer storage. If the data is in buffer storage, buffer storage is accessed; if not, processor storage is accessed, and the addressed data is both transmitted to CPU and loaded into buffer storage.

For all store operations, the buffer storage control stores data into processor storage; the data is also stored into buffer storage if the referenced storage location also resides in the buffer storage.

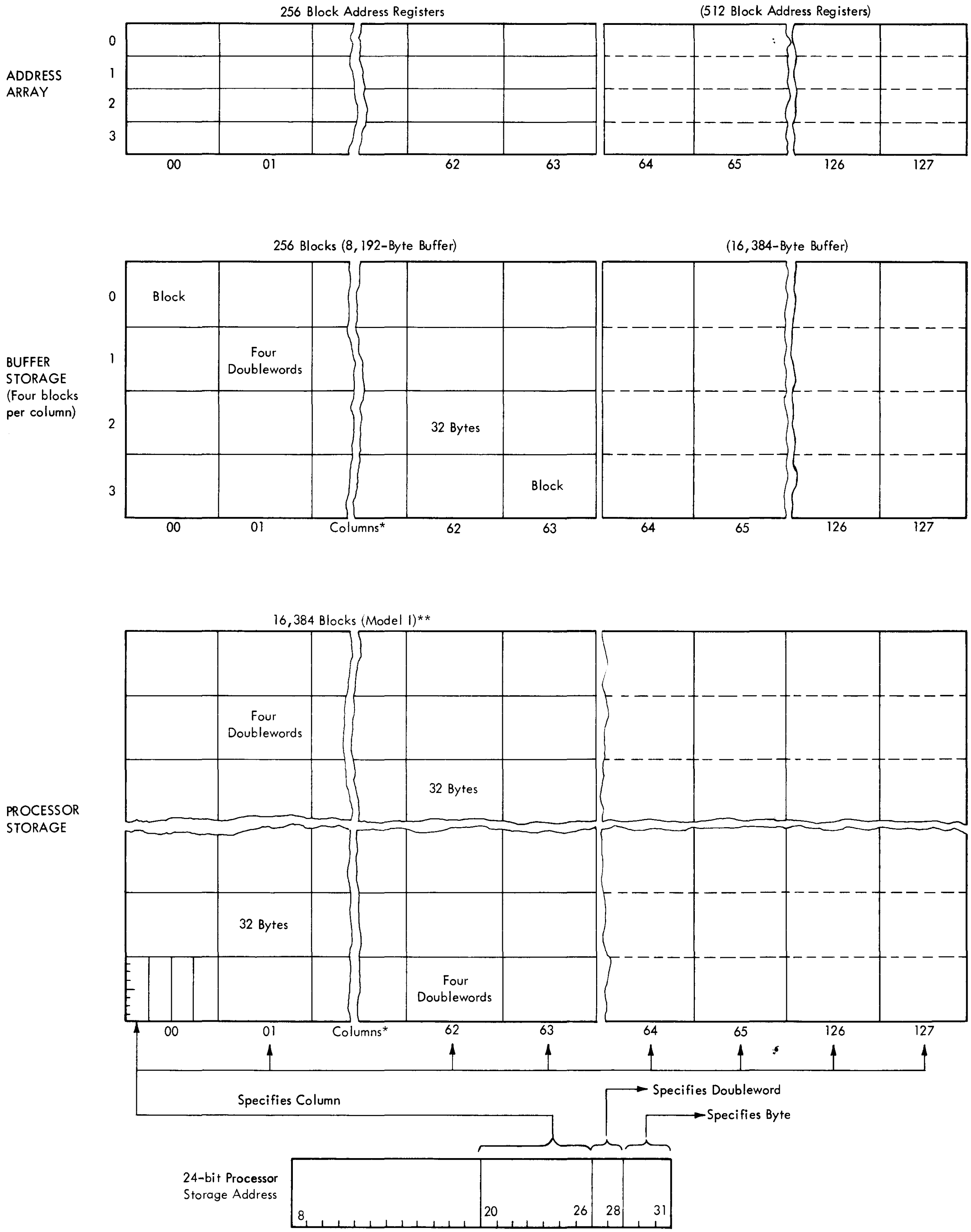
Processor storage and buffer storage are logically divided into a number of 32-byte blocks, on 32-byte boundaries (Figure 8). The number of blocks in the 8,192 byte buffer storage is 256, and these are logically divided into 64 columns, four blocks per column. Correspondingly, processor storage also is divided into 64 columns, the number of blocks per column varying with the size of processor storage.

During operation, a correspondence is set up that relates each block in buffer storage to a block in the corresponding column of processor storage. Each time the CPU makes a fetch, buffer storage control determines whether there is a buffer block corresponding to the addressed processor storage block. If none is found, one of the buffer blocks is automatically assigned to the block that the CPU addressed, the block address is placed in the buffer block's address register, and a buffer storage block load is called for.

When a CPU fetch dictates a block load, four 8-byte accesses to processor storage are required. The first processor storage location selected is the one containing the data addressed by the CPU. When it is available, this data is sent directly to the CPU and is also loaded into buffer storage. The three processor storage fetches needed to complete the block load are made one at a time on each succeeding CPU cycle that the required processor storage units are not busy or become not busy.

Channels store data into, but do not fetch data from, buffer storage. For a channel store operation, a check is made to determine whether the referenced data is in buffer storage; if so, the buffer storage data as well as the processor storage data is updated. If the referenced data is not in buffer storage, only processor storage is updated. Channel fetch requests are made only to processor storage.

Because buffer storage can contain only a portion of needed processor storage data at one time, any buffer block can be reassigned to any block in a corresponding column in processor storage. Priority of reassignment is based on usage. Each time data within a buffer block is referenced by a CPU fetch, that block is logically moved to the top of a logic-controlled activity list. Intervening blocks move down one position to fill the vacated slot. Note, however, that the logical movement of a block



* Number of columns depends on size of buffer storage.

** Number of blocks depends on size (model) of processor storage.

Figure 8. Buffer Storage

within the list involves no data transfer. When all four buffer blocks within a column are assigned and the CPU makes a fetch request to a corresponding storage location not yet in buffer storage, the buffer block lowest on the activity list is cleared and reassigned to the referenced processor storage block. (The buffer block at the bottom of a particular activity list is the one in that column that has gone the longest without being referenced by a CPU fetch.)

Store-type operations always update processor storage, but buffer storage is not updated unless the referenced processor storage block has a buffer storage block assigned to it. In summary, store operations cannot cause reassignment, or loading of a buffer storage block, or changing of the buffer storage block activity list.

I/O Channel Control

The I/O channel control receives and processes channel storage- requests. Each channel attached to the CPU has a fixed amount of channel control buffer (a buffer group) dedicated to its use. This buffering results in higher attainable channel data rates through maximum utilization of the four-way interleaved processor storage.

A channel buffer group provides two sections, each with control ability and data capacity for one doubleword inbound and one doubleword outbound (Figure 9). At installation time, the customer engineer assigns each buffer group to a channel by appropriate wiring of a matrix card. Thus, the channel requiring highest priority is assigned to buffer group 1, second highest priority to buffer group 2, etc. The resultant channel-channel buffer relationship may be viewed on the indicator viewer.

To transmit information to processor storage during an I/O read operation, the channels place data and control signals on a common channel data in-bus to the channel buffers in the I/O channel control. Channel buffer priority determines which channel may use the in-bus at any given time. As each channel buffer is loaded, it requests use of one of the four interleaved units of processor storage (and the high-speed buffer storage if the affected address is resident there). Here again, channel buffer priority determines which buffer section may use storage and transmit data via the storage data in-bus at any given time.

The loading of a channel buffer section frees the channel data in-bus for use by other channels vying for its use. Subsequent requests by the same channels are not allowed to contend for channel data in-bus use while the last loaded of its two buffer sections is busy, except for the highest priority channel. Requests from the highest priority channel, the channel assigned to buffer group 1, are inhibited from contending for channel data in-bus use only when both of its buffer sections are busy.

To access data from processor storage during an I/O write operation, a channel buffer section, upon receipt of a processor storage address from the channel, requests data through the processor storage control. Priority determines which buffer section may access processor storage at any given time. Once loaded from processor storage, the buffer section requests use of the data out-bus to the channel where channel buffer priority determines which buffer may use the channel data out-bus at any given time. The buffer section becomes non-busy upon reading out of its contents to the channel, and is again available to receive another channel request.

Processor Storage Control

The processor storage control handles the storage requests made by the buffer storage control and by the I/O channel control. Fetch, store and channel requests are recognized. For an active request, the processor storage control generates from-and-to information and makes the processor storage reference. During the latter part of the storage cycle, the processor storage control uses the from-and-to information to send the requesting unit the results of the storage request. These results consist of check and interruption information; and, when the operation is a fetch, of returning data as well.

Both store and fetch operations are modified by the error checking and correction logic in the storage units. Single-bit parity errors are detected and corrected, and double-bit parity errors are detected.

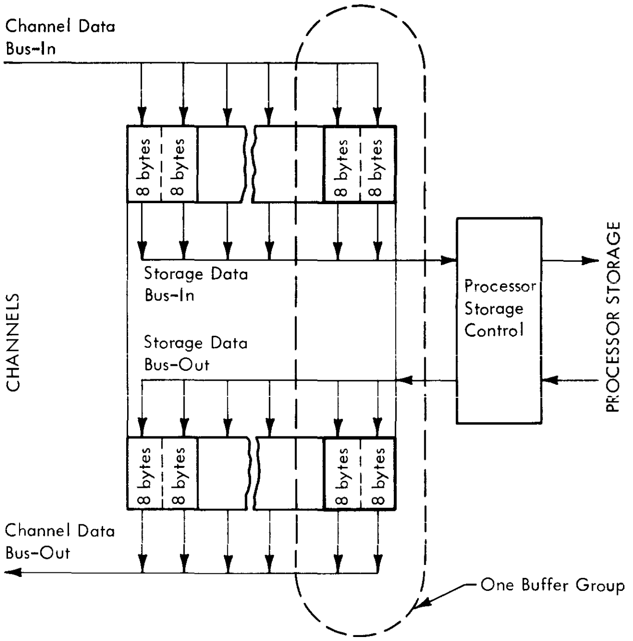


Figure 9. Channel Buffer

Control Storage

The control storage in the Model 165 consists of a combination of read-only storage (ROS) and writable control storage (WCS) plus associated logic. Internal transfers are parity-checked; a parity error causes a machine check. The cycle time for both types of control storage is 80 nanoseconds.

Control storage stores control information that is used to define the state of the CPU at any given time. The execution unit is controlled by microprograms in both read-only storage and writable control storage. The microprograms are arranged in a 108-bit control word format. More than 2,500 control word addresses are provided in the basic CPU: 2,048 in read-only storage and 512 in writable control storage.

One control word is accessed, decoded, and used during each machine cycle. Control is exercised over machine functions such as movement of the contents of a register to the input of an adder. Control words are used instead of conventional logic to control the operation of the CPU's execution unit.

The writable control storage is included to provide basic control as well as microdiagnostic capability. It is also used to contain the compatibility feature control words. When a compatibility feature is installed, 1,024 additional control word addresses in WCS are included in the CPU.

Instruction Retry

For most retryable instructions, retry involves restarting the instruction unit at the address of the instruction to be retried, and completely reexecuting the instruction.

For SS-format logical instructions, reexecution consists of restarting at the last byte successfully processed rather than reexecuting the entire instruction.

Instruction retry, as implemented in the Model 165, can be considered in three steps: normal instruction execution, error stop with retry decision, and retry restore.

Normal Execution

During normal execution of an instruction, data that would be needed for a retry of an instruction is saved in the maintenance control area. In addition, the address of the instruction being executed is maintained as a pointer to the instruction that will be retried if an error is detected. Retry controls update this value according to the length code of each instruction sucessfully executed. However, when a successful branch (or PSW load) is executed, the pointer address becomes invalid. The address of the next instruction to be executed is provided by the instruction unit, and this becomes the new pointer for retry control.

Error Stop

When an error is detected, a stop occurs (control is passed from the execution microprogram to error recovery circuitry). A status log is performed. The status log stores the contents of the maintenance control area into processor storage at location 256 (hex 100). This scan-out area in storage is referred to as the retry status table. Retry circuitry determines whether the error is retryable or unretryable. The decision involves the error type, the instruction type and the current retry mode. Circuitry subsequently initiates a CPU reset and returns control to the microprogram. If the error is retryable, the retry restore microprogram routine is entered. If the error is unretryable, a hard machine check interruption (MCI) is initiated and the microprogram interruption routine assumes control.

Retry Restore

The instruction to be retried is fetched from the retry instruction address in the status table. The instruction op code is examined by the retry decoder and a restore class identification is made. Operands that were changed during execution are replaced with data from the status table. When the restore operation is completed, the CPU status allows reexecution to begin. Reexecution restarts the instruction unit at the retry instruction address.

Maintenance Controls

The maintenance control area of the CPU executes the functions provided by the system control panel controls, provides diagnostic capabilities for the system, and implements the logic required to execute the system component portion of instruction retry.

The maintenance control data register (MCDR) is used during normal processing as an operand buffer for instruction retry. The maintenance control address register (MCAR) is used during normal processing to hold the current instruction counter (IC) value for instruction retry. The maintenance control retry register (MCRR) is used for operand buffering during instruction retry.

Other control and address registers, an arithmetic unit, decoders, count controls, and compare circuits are provided. Additional information on maintenance control registers is under "Operator Intervention Controls."

Machine Check Interruption

The machine check interruption is the primary way in which machine errors are signaled to the program. The errors may originate in the CPU, in processor storage operating with the CPU, in a channel operating with the CPU, or in processor storage operating with a channel. The channel error can occur during I/O instruction time or I/O interruption time. The processor storage- with-channel error can occur at any time.

When an error is signaled, enough information is provided to direct and allow subsequent program action. Two types of information are provided: program status and error information.

Program status is the data necessary to preserve the task for resumption later. Error information is the data necessary to help identify the source and severity of the error. Both types of information are provided in the machine check old PSW and in the logout field (status table). See Figure 10.

Machine Check Interruption Types

A machine check interruption (MCI) is classified as either hard or soft. These terms refer to the severity of the machine error that caused the interruption.

A hard MCI results from a critical (nonrecoverable) error and is characterized by an immediate termination of the CPU activity and a logout of machine status. Examples of errors that cause a hard MCI are:

1. An unretryable error in the CPU, such as a storage control unit check on a store operation.
2. An error that occurs while the CPU is in an unretryable state.
3. An uncorrectable storage error on a CPU store operation.
4. An error that is retried unsuccessfully.

A soft MCI is the result of a noncritical (recoverable) error and is a well-controlled interruption to the CPU. The soft MCI condition is not acted on immediately; the interruption is taken at the end of the operation during which the error is detected.

The errors that produce a soft MCI may or may not be associated with the program currently being executed in the CPU. At the time of the error, enough information is recorded in the logout area to allow error analysis later. Examples of errors that cause a soft MCI are:

1. An error that is successfully retried by the CPU.
2. A storage data error that is corrected by error-correction logic on a CPU or channel operation.
3. An uncorrectable storage error (address, key, mark, or data) on a channel operation.

Machine Check Interruption Code

The machine check interruption code contains information about the type and severity of the error, the validity of the various fields that are stored, and the validity and length of the extended logout. The machine check interruption code is a doubleword starting at location 232 (decimal); see Figure 10. (The interruption code in the machine check old PSW is all zeros.) The handling of a machine check interruption is defined in *IBM System/370 Principles of Operation,* GA22-7000.

176	Reserved
232	Machine Check Interrupt Code
240	Reserved
256	Fixed Logout Area (Retry Status Table)
352	Floating-Point Register Save Area
384	General Register Save Area
448	Control Register Save Area
512	

Figure 10. Fixed Logout Area in Processor Storage

Priority of Interruptions

The occurrence of a soft MCI condition does not cause an immediate interruption in the Model 165. A CPU operation in process when it occurs is completed before a soft MCI is taken. Therefore, the order in which the various types of interruptions are recognized and honored differs from that specified in the *IBM System/370 Principles of Operation,* GA22-7000.

Provided that the particular type of interruption is allowed, concurrent interruption requests are honored in the following priority:

Hard machine check
Program or supervisor call
Soft machine check
External
Input/Output

Soft and hard MCI's are disallowed when bit position 13 of the current PSW contains a 0. CPU errors are held pending. When MCI's are allowed again, only one MCI is taken for errors that occurred in the disabled state. Critical errors take precedence; other errors, if held pending, are eliminated.

Programming Note: Within the System/360 Operating System, the recovery management support (RMS) option has the ability to distinguish between hard and soft machine checks and to take appropriate action. RMS is available with MFT and MVT. Refer to appropriate Systems Reference Library (SRL) programming publications.

CHANNELS

The IBM 2870 Multiplexer Channel, the IBM 2860 Selector Channel, and the IBM 2880 Block Multiplexer Channel provide for attachment of I/O devices to the Model 165 system. The channel relieves the CPU of communication directly with I/O devices and permits data processing to proceed concurrently with I/O operations.

A standard channel-to-control-unit interface provides a uniform method of attaching control units to channels. Data is transferred a byte at a time between the I/O device and the channel. An optional two-byte-wide interface on the 2880 channel provides for attachment of devices with very high data rates. Data transfers between the channel and the SCU are eight bytes (one doubleword) in parallel for both selector and multiplexer channels.

The following descriptions include the maximum data rates attainable by the different channels. During systems operation, the actual rates may be less than the maximums, depending on (1) channel buffer priority. (see "I/O Channel Control"), (2) number of channels operating concurrently (3) the , speed of the devices operating on each channel, and (4) the type of channel programming used, e.g., single record vs chained records.

2860 Selector Channel

The 2860 Selector Channel provides for attachment and control of I/O control units and associated devices. At least one 2860 (any model) is required if no 2870 Multiplexer or 2880 Block Multiplexer Channel is attached. The 2860 is available in three models:

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

The 2860 Selector Channel permits data rates of up to 1.3 million bytes a second. I/O operations are overlapped with processing and, depending on the data rates and channel programming considerations, all selector channels can operate concurrently. A set of channel control and buffer registers permits each channel to operate with a minimum of interference.

Eight control units can be attached to each selector channel. Each control unit may have more than one I/O device connected to it, but only one device per channel may transfer data at any given time. A selector channel operates only in burst mode, and may be assigned addresses one through six only.

Note: A 2860 with a 2301 Drum Storage attached must be assigned either highest channel buffer priority and channel address 1, or second highest priority with channel address 2.

2870 Multiplexer Channel

The 2870 Multiplexer Channel provides for attachment of a wide range of low-to medium-speed I/O control units and associated devices. The basic 2870 Multiplexer Channel with 192 subchannels can attach eight control units and can address 192 I/O devices. The basic multiplexer channel can operate several multiplex-mode I/O devices concurrently or a single burst-mode device.

Two 2870's can be attached to the Model 165: the first one provides 196 subchannels including four optional selector subchannels; the second one provides 194 subchannels including two optional selector subchannels. The address of the first 2870 must be zero; the second 2870 may be assigned any address from one through six.

Selector subchannels are optional. Each selector subchannel can operate one I/O device concurrently with the basic multiplexer channel.

Each selector subchannel permits attachment of eight control units for certain devices having a data rate not exceeding 180 kilobytes (kb) a second. Regardless of the number of control units attached, a maximum of 16 I/O devices can be attached to a selector subchannel.

The maximum aggregate data rate for the multiplexer channel ranges from 110 kb to 670 kb, depending on the number of selector subchannels in operation. The first three selector subchannels may operate concurrently at up to 180 kb for each subchannel. When all four selector subchannels operate concurrently, the fourth 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 maximum data rates for concurrent selector subchannel operations are:

Basic	*Data Rates for Selector Subchannel*			
Multiplexer	*1st or 2nd 2870*		*1st 2870 Only*	
Channel	*lst*	*2nd*	*3rd*	*4th*
110 kb				
88 kb	180 kb			
66 kb	180 kb	180 kb		
44 kb	180 kb	180 kb	180 kb	
33 kb	180 kb	180 kb	180 kb	100 kb

Note: The 180-kb maximum data rate for 2870 selector subchannels pertains to attachment of magnetic tape devices; timing factors other than data rates may preclude attachment of direct-access storage devices that have lesser data rates. Also, note that when other channels in addition to the 2870 are in operation, the total system I/O data rate must be analyzed.

The maximum data rate for selector subchannels and for the basic multiplexer channel is a function of the channel buffer priority assigned to that 2870. The preceding table indicates the 2870 loading possible when assigned buffer priority 2.

2880 Block Multiplexer Channel

The functional use of the 2880 Block Multiplexer Channel closely parallels that of the 2860 Selector Channel; devices that attach to the Model 165 through a 2860 may also attach through a 2880 assigned one of the addresses one through six. The 2880 is capable of higher data rates than the 2860, however, and also offers a block multiplexing capability. Unlike the 2860 and 2870 channels, the 2880 performs a channel logout on occurrence of a channel data check condition. The logout occurs at the completion of the current command, and into locations 256 through 351.

Data check logouts may be eliminated during selector mode operation by inserting the bypass-logout circuit jumper.

Note: Only qualified maintenance personnel may insert or remove the bypass-logout circuit jumper.

At least one 2880 Block Multiplexer Channel is required if no 2860 Selector or 2870 Multiplexer channels are attached. The 2880 always transfers data in burst mode, and may be assigned any address from one through eleven.

Two models of the 2880 Block Multiplexer Channel are available:

Model 1 – provides one channel
Model 2 – provides two channels

The standard features of the 2880 are the high-speed data transfer mode (up to 1.5 million bytes per second) and the block-multiplexing capability (concurrent operation of up to 64 devices).

An optional feature is the two-byte interface that provides for a data transfer rate of up to 3.0 million bytes per second.

The basic 2880 Block Multiplexer Channel attaches to the standard one-byte I/O interface. The 2880 uses a high-speed data transfer mode that incorporates two additional interface tag lines, Data-In and Data-Out, to allow I/O data rates up to 1.5 million bytes per second. Data rates of 1.5 to 3.0 million bps are achieved through additional buffering and use of an optional two-byte I/O interface. All data transfers over the two-byte interface are in the high-speed data transfer mode.

The block multiplexing function provides 64 subchannels to be block multiplexed over the standard I/O interface. Block multiplexing is similar to the byte multiplexing capability of the 2870 Multiplexer Channel; the principal difference is the quantity of data per transmission. The 2880 multiplexes blocks (bursts) of data; that is, burst mode devices share the channel facilities and transmit data in burst mode. When the channel is multiplexing (interleaving) blocks of data on the single data path, it may also control the nondata transfer activities of multiple devices.

Eight control units can be attached to each 2880 Block Multiplexer Channel. Each control unit may have more than one I/O device connected to it, but only one device per channel may transfer data at any given time. However, as many as 64 channel programs may be in concurrent execution in each 2880 channel through use of the block multiplexing function.

To facilitate conversion, the 2880 also operates in selector channel mode, thus permitting operation under the current operating system. The selector or block multiplex mode is program selectable (See "Block Multiplexing Control" in *IBM System/370 Principles of Operation,* GA22-7000.)

Extended Channel Feature

This feature permits attachment of as many as 12 channels to the Model 165, in which case bit 6 in the system mask of the PSW masks channels 6-11 as a group rather than channel 6 alone. Bit 6 is a master bit; individual control is in control register 2, bits 6-11. (See "Extended I/O Masking" in *IBM System/370 Principles of Operation,* GA22-7000.)

Channels may be attached to the CPU as follows:

	Without Extended Channel Feature	*With Extended Channel Feature*
Maximum number channel frames	5	7
Maximum number channels	7	12
Maximum number 2860's and 2870's	7	7
Maximum Number 2860's	6	6
Maximum Number 2870's	2	2
Maximum Number 2880's	6	11

Note: Maximum channels per frame:
3 – 2860
2 – 2880
1 – 2870

Channel-to-Channel Adapter Feature

A channel-to-channel adapter is available as an optional feature on the 2860. The adapter provides a path for operations to take place between two channels and synchronizes those operations. It may be used in multiple-processor or single-processor systems: in a multisystem, to achieve rapid communications between the channels of two System/370 models, or a System/360 and a System/370; or in a single System/370 to move blocks of data from one processor storage area to another.

The adapter uses one control-unit position on each of the two channels, but only one of the two connected channels requires the feature. In the Model 165, one adapter may be installed per 2860 Selector Channel.

When the 2870 or 2880 channel is connected to a 2860 channel, the channel- to-channel adapter is installed on the 2860 channel, not on the 2870 or 2880.

For restrictions on channel attachments for another IBM System model used with the Model 165, refer to the Systems Reference Library (SRL) functional characteristics publication for that model.

SYSTEM CONTROL FUNCTIONS

System Reset

The system-reset function suspends all instruction processing and interval timer updating, resets the channels, on-line control units, I/O devices, all error-status indicators, and all pending interruptions. The system is placed in such a state that subsequent CPU operations can be initiated without causing a machine check due to a condition previously existing in the CPU.

Control is then passed to a microprogram routine which completes the system reset function, including the clearing of the PSW, under microprogram control. Also, if the clear lever-switch on the system console is in the clear position, the general registers, floating-point registers, processor storage, and the storage protection keys are also cleared. Subsequently the CPU is placed in the stopped state.

The reset state for a control unit or device is described in the appropriate Systems Reference Library (SRL) publication. A system-reset signal from a CPU resets only the functions in a control unit or device belonging to that CPU.

The system-reset function is performed when the system-reset key is pressed, when initial program loading is initiated, or when a system power- on sequence is performed.

Programming Note: If a system reset occurs in the middle of an operation, the contents of the PSW and of the result registers or storage locations are unpredictable. If the CPU is in the wait state when the system reset is performed and no I/O operation is in progress, this uncertainty is eliminated.

A system reset does not correct parity in registers or storage unless the clear lever-switch is set to the clear position. Because a machine check occurs when information with incorrect parity is used, the incorrect information should be replaced by loading new information when a system reset with clearing may not be performed.

Store and Display

The store-and-display function permits manual intervention in the progress of a program. The storing and/or displaying of data may be provided by a supervisor program in conjunction with appropriate I/O equipment and the interrupt key.

In the absence of a suitable supervisor program, the operator intervention controls allow direct storing and displaying of data. This is done by placing the CPU in the stopped state and subsequently storing and/or displaying information in processor storage, in general and floating-point registers, and in the instruction-address portion of the PSW. The CPU enters the stopped state when the stop key is pressed, when single instruction execution is specified and the instruction has been executed, or when a preset address is reached. The CPU completes the current instruction, and services pending interrupts before entering the stopped state. The store-and-display function is achieved through the use of the store, display, set IC, set PSW keys, and the hex data keys, the manual entry select and storage select switches, and the CRT mode select switch. When the desired intervention is completed, the CPU can be started again.

The normal stopping and starting of the CPU does not cause any alteration in program execution other than the time element involved in the transition from operating to stopped state.

Machine checks occurring during store-and-display operations do not log immediately but create a pending log condition that can be removed by a system reset, CPU reset, or a check reset. The error condition, when not disabled, forces a logout and a subsequent machine check interruption when the CPU is returned to the operating state.

Initial Program Loading

Initial program loading (IPL) is provided for initiation of processing when the contents of processor storage or the PSW are not suitable for further processing.

Initial program loading is initiated manually by selecting an input device with the load-unit switches and pressing the load key.

Pressing the load key causes a system reset, turns on the load light, turns off the manual light, and initiates a read operation at the selected input device. The system reset suspends all instruction processing and interval timer updating, resets the channels, and resets on-line nonshared control units and I/O devices. By setting the clear lever-switch in the clear position, the system reset portion of the IPL sequence may be modified to also provide clearing of general registers, floating-point registers, processor storage, and the storage protection keys.

When the read operation is completed satisfactorily, the load light is turned off, a new PSW is obtained, and the CPU starts operating.

When IPL is initiated, the selected input device starts transferring data. The first 24 bytes read are placed in storage locations 0-23. Protection, program-controlled interruption, and a possible incorrect length indication are ignored. The doubleword read into location 8 is used as the channel command word (CCW) for reading more than 24 bytes. When chaining is specified in this CCW, the operation proceeds with the CCW in location 16. Either command chaining or data chaining may be specified.

When the device provides channel end for the last operation of the chain, the I/O address is stored in bits 21-31 of the first word in storage. Bits 16-20 are made 0; bits 0-15 remain unchanged.

The CPU subsequently fetches the doubleword in location 0 as a new PSW and proceeds under control of the new PSW. The load light is turned off. No I/O interruption condition is generated. At this point, the loader portion of the first program is loaded.

When the I/O operations and PSW loading are not completed satisfactorily, the CPU idles, and the load light remains on.

Programming Notes: Initial program loading resembles a start I/O that specifies the I/O device selected by the load-unit switches and a 0 protection key. The CCW for this start I/O is simulated by the CPU circuitry and contains a read command, 0 data address, a byte count of 24, CC flag on, SLI flag on, PCI flag off, CD flag off, and skip flag off. The CCW has a virtual address of 0.

Initial program loading reads new information into the first six words of processor storage. Because the remainder of the IPL program may be placed in any desired section of storage, it is pôssible to preserve such areas of storage as the timer and PSW locations, which may be helpful in program debugging.

If the selected input device is a disk, the IPL information is read from track 0.

When the PSW in location 0 has bit 14 set to 1, the CPU is in the wait state after the IPL procedure (the manual, system, and load lights are off, and the wait light is on). Interruptions that are requested during IPL are taken, if allowed, before instruction execution begins.

When the load light goes off, the loader portion of the first program is loaded. The continuing procedure for IPL varies slightly (but is nearly, if not wholly, automatic) if stand-alone programs are being loaded. Program execution follows either immediately or, if the CPU is placed in the wait state at the end of IPL, on receipt of the signal to exit from the wait state. If the control program is being loaded, additional procedures for the operator are required.

SYSTEM CONSOLE

The system control panel (Figure 11) which includes the main control panel, contains the switches, keys, and indicator lights necessary to operate, control, and display the system. The system consists of the CPU, processor storage, channels, on-line control units, and I/O devices. Off-line control units and I/O devices, although part of the system environment, are not part of the system proper.

System controls are logically divided into three classes: operator control, operator intervention, and customer engineer control.

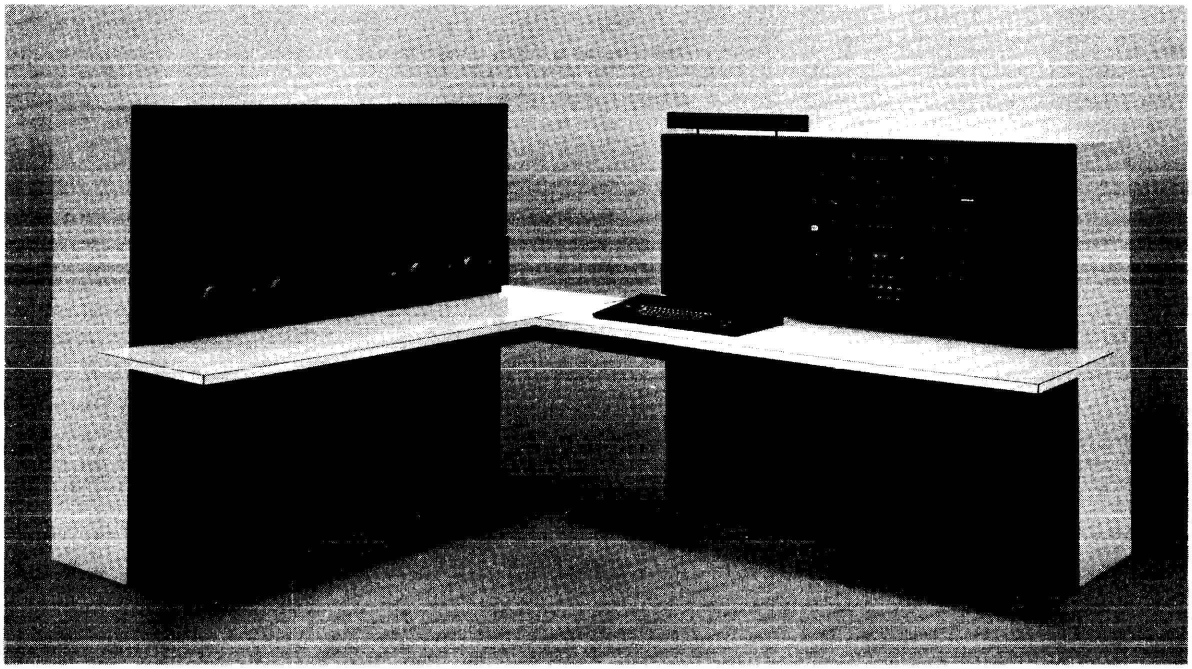


Figure 11. System Console Control Panels

Using the system control panel, the operator can reset the system; store and display information in storage, in registers, and in the PSW; perform initial program loading (IPL); and monitor activity in various system elements.

Operator Controls

The operator control section of the system control panel contains the controls and indicator lights required by the operator when the CPU is operating under full supervisor control.

Under supervisor control, a minimum of direct manual intervention is required because the supervisor performs operations similar to store and display.

Panel A5 on the main control panel (Figure 12) contains the operator control panel. Panel A1 contains the system activity monitor. The emergency-pull switch is on panel A2. The main functions provided by the operator controls are: the control and indication of power, the indication of system status, operator-to-machine communication, and initial program loading.

The following table lists (alphabetically) all operator controls and indicator lights and their implementation. (See Figure 12.)

Name	*Implementation*
Emergency pull	Pull switch
Interrupt	Pushbutton
Interval timer	Lever switch
Load	Pushbutton
Load	Indicator light
Load unit	Three rotary switches
Manual	Indicator light
Power off	Pushbutton
Power on	Pushbutton, backlighted
System	Indicator light
System activity meter	Meter
Test	Indicator light
Time-of-day clock	Lever switch
Wait	Indicator light

Emergency Pull (Panel A2)

Pulling the emergency-pull switch turns off all power beyond the power-entry terminal on every unit that is part of the system or that can be switched onto the system. This switch latches in the out position and can be restored to its normal position by maintenance personnel only. When the emergency-pull switch is in the out position, the power-on key is ineffective

Interrupt (Panel A5)

The interrupt pushbutton is pressed to request an external interruption. The interruption is taken when it is allowed and when the CPU is not stopped; otherwise, the interruption remains pending. Bit 25 in the interruption-code portion of the external old PSW is made 1 to indicate that the interrupt pushbutton is the source of the external interruption. The interrupt pushbutton is effective while power is on the system. (See definition of external interruption masking for the interrupt pushbutton in *IBM System/370 Principles of Operation,* GA22-7000.)

Interval Timer (Panel A4)

Placing the interval timer switch in the DSBL (disable) position prevents updating of the interval timer. When this is done, the test light is turned on.

Load (Pushbutton - Panel A5)

The load pushbutton is pressed to start initial program loading. The load key is effective while power is on the system.

Load (Light - Panel A5)

The load light is on during initial program loading; it is turned on when the load key is pressed and is turned off after the read operation and the loading of the new PSW are completed successfully.

Load Unit (Panel A5)

Three rotary switches provide the 11-bit address of the channel and unit to be used for initial program loading.

The leftmost rotary switch has sixteen positions (0-F) and is used to select the channel address. The other two are 16-position rotary switches (hexadecimal digits 0-F) and are used to select the subchannel, control unit, and device.

Manual (Panel A5)

The manual light is on when the CPU is in the stopped state. Several of the manual controls are effective only when the CPU is stopped (manual light on).

Power Off (Panel A5)

The power-off pushbutton is pressed to initiate the power-off sequence of the system. The contents of processor storage (but not the keys in storage associated with the protection feature) are preserved, provided the CPU is in the stopped state. The contents of local storage, buffer storage, and writable control storage are lost. The power-off pushbutton is effective while power is on the system.

Power On (Panel A5)

The power-on pushbutton is pressed to initiate the power-on sequence of the system. As part of the power-on sequence, a system reset is performed to reset all CPU, channel, on-line control unit, and I/O device circuitry to a neutral condition and to clear all general and floating-point registers and storage protection keys. The contents of processor storage are preserved.

The power-on pushbutton is backlighted white when power is on the entire system. The pushbutton is backlighted red during the power-on sequence. If there is a loss of power in some section of the CPU, processor storage units, or channels, the light changes from white to red. The power-on pushbutton is effective only when the emergency-pull switch is at the in position.

System (Panel A5)

The system light is on when either the central processing complex (CPC) usage meter or the customer engineer (CE) meter is running. These meters are on panel A5 of the main control panel.

The manual light and wait light indications function independently; the system light indication is a function of both the CPU and the I/O states. The following table shows possible conditions when power is on the system:

System Light	*Manual Light*	*Wait Light*	*CPU State*	*I/O State*
Off	Off	Off	*	*
Off	Off	On	Wait	Not Working
Off	On	Off	Stopped	Not Working
Off	On	On	Stopped, Wait	Not Working
On	Off	Off	Running	Undetermined
On	Off	On	Wait	Working
On	On	Off	Stopped	Working
On	On	On	Stopped, Wait	Working

*Abnormal condition

System Activity Meter (Panel A1)

The system activity monitor displays the average activity of the major system elements on the system activity meter. The meter provides a dynamic visual indication of the CPU or I/O activity.

System parameters are selected on a function selector control for meter display, and a monitor hub (not shown) permits external attachment of a stripchart recorder or time-base counter for precise recording of a selected function. The panel controls provide flexibility in performing simple combinatorial logic for setting up a particular measurement. Among the measurements, which are easily made on a one-at-a-time basis, are the following:

1. Compute time (total, supervisor state, or problem state).
2. CPU-channel overlap for selected channels.
3. Channel-to-channel overlap for selected channels.
4. Total channel time for selected channels.

Compute measurements can further be made for a specific PSW storage protection key when KEY SELECT is enabled.

The calibrate check switch is provided for use in verifying that individual functions within the device are in proper calibration.

Test (Panel A5)

The test light is on when a manual control is not in its normal position or when a maintenance function is being performed for the CPU, channels, or processor storage.

Any abnormal setting of a switch that is on the system control panel or on any separate maintenance panel for the CPU, processor storage, or channels that can affect the normal operation of a program, causes the test light to go on.

The test light may be on when certain diagnostic functions are activated or when certain abnormal circuit-breaker conditions occur. The test light is not an indication of the state of the marginal voltage controls.

The test light is on when any one of the following manual controls is not in its normal position:

Block retry buffer
Buffer
CE meter key switch
ECC
Forced repeat
Interval timer
Machine check
Microdiagnostics switches
(any one of the three)

Overlap
Rate
Repeat instruction
Retry
Storage test
Soft machine-check interrupt
Stop-on-compare
Store/display control

Time-of-Day Clock (Panel A5)

This two-position lever switch is placed in the set position to allow setting of the clock, and in the secure position to disallow setting of the clock by the control program.

Wait (Panel A5)

The wait light is on when the CPU is in the wait state. The wait state exists when bit position 14 of the current PSW contains a 1. The wait state can be changed to the running state only by loading a new PSW in which bit position 14 contains a 0; it cannot be changed by pressing the system- reset key. Normal exit is by an external or I/O interruption or an IPL.

Operator Intervention Controls

Panels A4 and A5 on the main control panel (Figure 12) contain most of the controls required for the operator to intervene in normal programmed operation. These controls are intermixed with CE controls in the same areas.

Operator intervention includes the system-reset and the store-and-display functions.

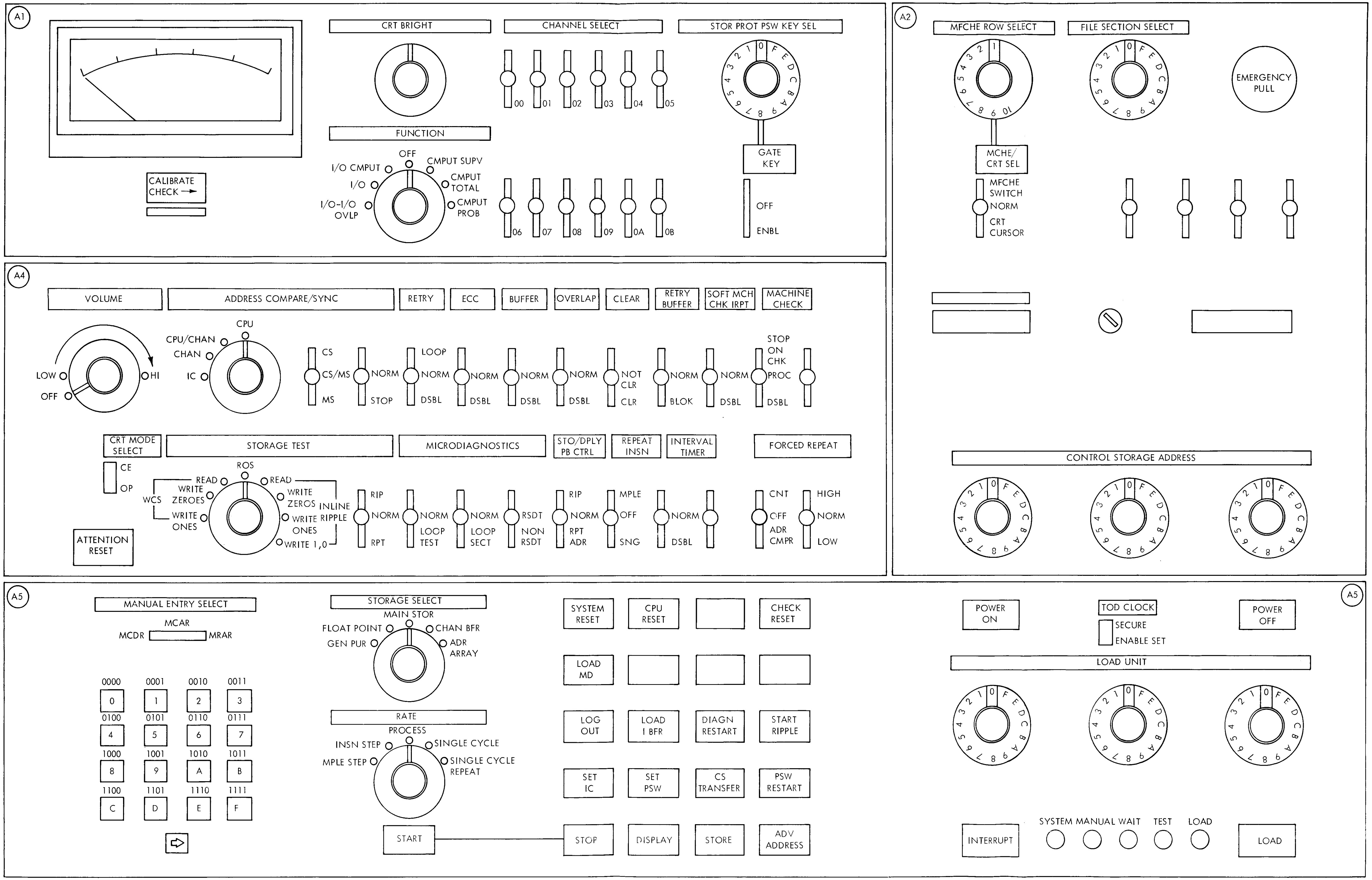


Figure 12. System Console – Main Control Panel

The operator intervention controls include (in alphabetical order):

Name	*Implementation*
Address compare/Sync	Rotary switch
Address compare stop	Lever switch
Advance address	Pushbutton
Attention reset	Pushbutton
Check reset	Pushbutton
Clear	Lever switch
CPU reset	Pushbutton
CRT brightness	Potentiometer
CRT mode select	Lever switch*
Cursor advance	Key
Data keyboard	Keys
Display	Pushbutton
Manual entry select	Lever switch
PSW restart	Pushbutton
Rate	Rotary switch
Set IC	Pushbutton
Set PSW	Pushbutton
Start	Pushbutton
Stop	Pushbutton
Storage select	Rotary switch
Store	Pushbutton
System reset	Pushbutton
Volume	Potentiometer*

*Operator console feature

Manual Entry for Store and Display (Panel A5)

Several program-unaddressable registers are used in conjunction with the switches and indicators on the system control panel during manual store-and- display operations. A description of these registers and their operation follows.

The maintenance control address register (MCAR) is a 32-bit register that holds the address of a storage location from which data is fetched or to which data is sent.

The maintenance control data register (MCDR) is a 64-bit register that holds storage data to be stored from the main control panel or displayed on the cathode-ray-tube (CRT) display.

The maintenance control entry register (MCER) is an eight-bit register that receives hexadecimal digits as they are entered from the data keyboard. After the MCER has received two hexadecimal digits, correct parity is generated and the byte is automatically transferred to the MCAR or the MCDR.

To manually enter data or addresses, the manual entry select switch and the cursor on the CRT display must be used. The manual entry select switch specifies the receiving register as the MCAR or MCDR and causes the cursor to appear under the selected register display on the CRT.

The cursor:

1. Is a bright line always displayed on the CRT.
2. Indicates the next byte into which data or addresses will be entered.
3. Can be moved from left to right by pressing the cursor advance key on the data keyboard.
4. Returns to byte 0 when the manual entry select switch position is changed.
5. Advances for the length of the register it is under and wraps around to (begins again at) byte 0 of that register.

If a hexadecimal digit has been entered into the first position of the MCER and a reset is desired, a momentary pressing of the cursor advance key allows the first digit to be reentered without the cursor being advanced. The MCER is not changed unless new data is entered.

Address Compare/Sync (Panel A4)

The address-compare/sync switch provides a means of stopping the CPU when a selected storage address is encountered during system operation.

With the stop-on-compare switch set to the stop position, the CPU stops on an equal comparison according to the mode selected by the address-compare rotary switch. The four modes refer to comparisons made to addresses selected by the CPU, the channels, the CPU or the channels, or the value of the instruction counter.

The address-compare switch can be manipulated without disrupting CPU operation other than by causing the address-comparison stop.

*Note;*The three-position CS/MS lever switch is used only by maintenance personnel. It must be at the MS position to allow a stop on main storage compare.

Advance Address (Panel A5)

The advance-address pushbutton causes the address in the maintenance control address register (MCAR) to be incremented by one doubleword. The advance- address pushbutton is active only when the CPU is in the stopped state.

Attention Reset (Panel A4)

The attention-reset pushbutton is part of the integrated operator console feature. Pressing the attention-reset pushbutton turns off the attention indicator that is used to alert the operator.

Check Reset (Panel A5)

Pressing the check-reset pushbutton resets all CPU and storage error checks. The check reset function can be considered a subset of the system reset. The check-reset pushbutton is active in all CPU states.

CPU Reset (Panel A5)

Pressing the CPU-reset pushbutton causes all CPU control triggers to be reset, forces the CPU into the stopped state, and activates the check-reset function. The CPU-reset pushbutton is active in all CPU states.

CRT Bright (Panel A1)

The CRT bright control on panel A1 provides for adjusting the CRT display for best viewing brightness.

CRT Mode Select (Panel A4)

The two-position CRT mode select switch on panel A4 of the main control panel is used to select either OP (operator display) or CE (maintenance display, for manual store-and-display functions).

Cursor Advance (Panel A5)

The cursor advance key (labeled with a right-going arrow) is pressed and held to move the cursor on the CRT continuously from left to right. When the key is momentarily pressed, the cursor moves one byte to the right. The use of the cursor is explained under "Manual Entry for Store and Display."

Data Keyboard (Panel A5)

The data keyboard consists of a 4 x 4 matrix of keys. The keyboard is used to enter data into the maintenance control data register (MCDR) and to enter addresses into the maintenance control address register (MCAR).

Data and addresses are entered one hexadecimal digit at a time, and only when the CPU is in the stopped state.

Display (Panel A5)

Pressing the display pushbutton causes the contents of the storage address specified by the MCAR and the storage-select switch to be transmitted to the MCDR and displayed on the CRT. The display pushbutton is effective only if the CPU is in the stopped state, and the display function is performed normally only if the store/display control switch (Panel A2) is in the normal (center) position.

Main Storage Display: If the storage-select switch is set to the main store position, pressing the display pushbutton causes the contents of the doubleword storage location specified by the contents of the MCAR to be displayed from the MCDR on the CRT. If the address specified is valid in buffer storage, the buffer storage data is displayed. If the address is not assigned in buffer storage, main storage is displayed on the first pressing of the display key. Buffer storage is displayed on subsequent pressings. If a system reset is performed before displaying, main storage is displayed first.

General Register Display: If the storage-select switch is set to GEN PUR, pressing the display pushbutton causes the contents of the general register specified by the high-order byte of the MCAR to be displayed on the CRT. The display occupies the right half of the MCDR.

Floating-Point Register Display: If the storage-select switch is set to FLOAT POINT, pressing the display pushbutton causes the contents of the floating-point register specified by the high-order byte of the MCAR to be displayed on the CRT. The display occupies all of the MCDR.

Manual Entry Select (Panel A5)

This three-position lever switch (MCDR,MCAR, and MRAR) selects the register for data entry and causes the cursor to be displayed with that register on the CRT display. The MRAR (maintenance ripple address register) position is used only by maintenance personnel.

PSW Restart (Panel A5)

Pressing the PSW-restart key causes the following actions (after completion of the current instruction and pending interruption):

1. The contents of the current PSW are stored in processor storage locations 8 through 15.
2. Loading of a new PSW from storage location 0.
3. Instruction fetching, starting at the new program location specified by the new PSW.
4. Execution of instructions as specified by the setting of the rate switch.

The PSW-restart pushbutton is effective in the running and stopped states; it is not necessarily effective in the check-stop state.

Rate (Panel A5)

The rate switch is a five-position rotary switch that defines the rate of instruction processing. A change in the rate of instruction processing should be made only when the CPU is in the stopped state, except in single- cycle operation; otherwise, results are unpredictable.

The test light is on if the rate switch is set to any position other than PROCESS.

Process: In the process position, the rate switch allows instructions to be processed at the normal operating rate when the start pushbutton is pressed. The decoding of instructions is halted by pressing the stop pushbutton.

Instruction Step: In the instruction-step position, the rate switch allows one instruction to be completely executed when the start pushbutton is pressed. If an interruption becomes pending before the end of the operation, it is serviced. I/O operations initiated in the instruction-step mode are completed. The execute instruction and its target instruction are considered a single instruction.

Multiple Step: In the multiple-step position, the rate switch allows instructions to be processed at the rate of one instruction every 80 milliseconds when the start pushbutton is held depressed. Instruction execution is halted when the start pushbutton is released. (If an interruption becomes pending before the end of the operation, it is serviced.) I/O operations initiated in the multiple-step mode are completed. The CPU enters the stopped state.

Note: The single-cycle and single cycle repeat positions are used only by maintenance personnel.

Set IC (Panel A5)

The set instruction counter pushbutton sets the instruction counter to bits 40-63 of the MCDR when the system is in the stopped state.

Set PSW (Panel A5)

Pressing the set-PSW pushbutton sets the PSW to the corresponding bits in the MCDR. The ILC (bits 32,33) and the interruption code (bits 16-31) are not updated. The set-PSW pushbutton is active only when the CPU is in the stopped state.

Start (Panel A5)

The start pushbutton is pressed to start instruction execution as defined by the setting of the rate switch.

Pressing the start pushbutton after a normal halt causes instruction processing to continue as if no halt had occurred, provided the rate switch is in the process, instruction-step, or multiple-step position.

Pressing the start pushbutton after system reset without first having introduced a new instruction address yields unpredictable results.

Pending interruptions that are allowed are honored before the first instruction is executed.

The start pushbutton is effective only when the CPU is in the stopped state.

Stop (Panel A5)

Pressing the stop pushbutton ends machine operation without destroying the machine environment. The CPU enters the stopped state after the current instruction has been executed and after all interruptions that become pending before the end of the current instruction are processed. I/O operations in progress are completed.

As the CPU goes into the stopped state, the manual light is turned on. No interruptions are processed after the CPU enters the stopped state. The stop pushbutton is effective when power is on the system.

Address Compare Stop (Panel A4)

Setting the stop-on-compare lever switch to the stop position causes the CPU to stop on an equal-address comparison, as defined by the position of the address-compare rotary switch.

Storage Select (Panel A5)

This five-position rotary switch specifies the data source (display) or data destination (store) for manual store and display operations.

Store (Panel A5)

Pressing the store pushbutton causes the contents of the MCDR to be stored into the storage address specified by the MCAR and the storage- select switch. Storage protection is ignored. A single byte can be stored by displaying the storage location, making the appropriate change to the MCDR, and pressing the store pushbutton. The store pushbutton is effective only when the CPU is in the stopped state, and the store function is performed normally only if the store/display control switch is in the normal (center) position.

Manual Store to Main Storage: If the storage-select switch is set to the main store position, pressing the store pushbutton causes the contents of the MCDR to be stored into main (processor) storage at the doubleword location specified by the contents of the MCAR. If the address specified is presently in buffer storage, buffer storage is also updated at the time of the store operation.

Manual Store to General Register: If the storage-select switch is set to GEN PUR, pressing the store pushbutton causes the contents of the right half of the MCDR to be put into the general register specified by the high- order byte of the MCAR.

Manual Store to Floating-Point Register: If the storage-select switch is set to FLOAT-POINT, pressing the store pushbutton causes the contents of the MCDR to be put into the floating-point register specified by the high- order byte of the MCAR.

System Reset (Panel A5)

Pressing the system reset pushbutton resets all CPU control triggers; resets all on-line channels, control units, and associated I/O devices; activates the CPU reset; activates the check reset and zeros the PSW. The CPU is placed in the stopped state, and all pending interruptions are eliminated. With the clear lever-switch in the clear position, the general and floating-point registers, processor storage and storage protection keys are cleared. The system reset key is active in all CPU states.

Volume (Panel A4)

The volume potentiometer is part of the integrated operator console feature and controls the loudness of the audible alarm used by the system to alert the operator.

CE Controls

All switches and indicators not described as operator controls or operator intervention controls are considered to be customer engineer (CE) controls.

The customer usage meter and the CE meter for the central processing complex (CPC) are both on panel A2 of the main control panel. The Model 165 CPC includes the processor, processor storage, the system console, the power and coolant distribution unit. Channels and I/O units contain individual usage meters. (See "Usage Metering.")

A key switch, between the customer and CE CPC meters, controls which meter is to be run. If the key switch is in the customer-operation position, the customer meter accumulates time when the system is in operation; that is, initiating, executing, or completing program instructions, including I/O or assignable unit operations. If the key switch is in the CE position, the CE meter accumulates time on the same basis, and conditioning signals are inhibited from other meters in the system.

USAGE METERING

Usage meters appear on the following units of the Model 165: the 3165 Central Processing Unit, the 2860 Selector Channel, the 2870 Multiplexer Channel, and the 2880 Block Multiplexer Channel. Meters also appear on individual I/O units.

On the 3165 Central Processing Unit, the customer usage meter and the CE meter are on panel A2 of the main control panel. The CE key switch controls which of these meters is to be run while the system is in operation; that is, initiating, executing, or completing instructions, including I/O and assignable unit operations. The system light on panel A5 indicates when the system is in operation. The test light on panel A5 may indicate when the key switch is in the CE meter position. See test light under "Operator Controls" for other conditions.

The 2860 Selector Channel Models 1-3, the 2870 Multiplexer Channel, and the 2880 Block Multiplexer Channel each have one usage meter mounted on their respective power controls panels.

When each meter runs depends on the general function performed by the unit to which it is attached. The function of those units, by category, and the conditions under which the meter runs are described in Figure 13.

Unit Categories	Category Description	When Meter Runs
Central Processing Complex (CPC) Base CPC Units: 3165 Central Processing Unit	Base units are essential in the operation of the CPC, and perform permanent functions in the CPC, which controls the system.	When the system is in operation, the CPC meter (base complex system's meter) records time during which the system is initiating, executing, or completing program instructions, including I/O and assignable unit operations. While the system is in operation, conditioning signals are supplied to all assignable and I/O unit meters. When the system is not operating, the CPC meter is not recording, and conditioning signals are not supplied if there are no I/O assignable units initiating, executing, or completing an instruction and if the CPU status is a stop or wait state.
Assignable CPC Units: 2860 Selector Channel 2870 Multiplexer Channel 2880 Block Multiplexer Channel Control Units	Assignable units are similar to base units in that they must consistently be available to the CPC and are essential in its operation for certain system applications. However, there may be significant periods of scheduled time when they are not required by the CPC; thus, customer control over their availability to the system is provided.	The assignable unit meter records time when it is enabled and the system is in operation. The assignable unit may be changed by the availability control switch from enabled to disabled or from disabled to enabled only when the CPU is in the stop or wait state and this assignable unit is not initiating, executing, or completing an operation across this interface. When the assignable unit is disabled, it is not available to the system.
Input/Output Units On-Line: Units physically interconnected to the CPC.	Input/output units are task-oriented units. Whole participation in a system operation normally can be predetermined and their initiation and termination anticipated. Availability to the system is controlled through the required normal servicing by the operator.	While conditioning signals are supplied, the input/output unit can be used, and its meter records time from its first operation until stopped, as defined below: Card Unit: From the first read or write command until cards are run out of all feeds. Printer: From the first write command until the carriage space key or restore key is pressed. Tape Unit: From the first read or write command until the end of rewind; that is, that period while the tape unit is ready and tape is not at load point.
Off-Line: Units that are never attached to a system.		The meter records time from the first operation until a runout occurs.
On-Line/Off-Line: Units that can be interconnected to the CPC and can also be operated independently off-line.		Time is recorded as previously indicated for on-line or off-line units, depending on the unit's mode of operation.

Note: A minimum of approximately 1 second is recorded for each CPU meter start. Unmetered units can be interposed between metered units without blocking conditioning signals.

Figure 13. Metering Table

Instruction Timing Considerations

Because of the complexity of the Model 165, it is not possible to provide simple timing formulas that exactly express the processor operation. The timings and formulas provided in this section are a reasonable approximation to the Model 165 timings. Because of the complex interaction of the various areas of the Model 165, variations from the listed times can result. For example, specific programs evaluated to date have shown the time computed from the list to vary from the actual time by as much as 28-1/2 percent.

The formulas were prepared by timing a sequence of each instruction until a repetitive pattern became evident; from this pattern, the average time was determined. Formulas for instructions with variable execution times were adjusted by including appropriate factors to account for the variations.

Timing Assumptions

For instructions whose execution time is data-dependent, random data has been assumed. Each branch instruction has been divided into several special cases appropriate to it, and separate times generated for each. In addition to timing a sequence of branches, each branch was timed for the situation during which it is preceded and followed by a sequence of fixed- point loads. Times quoted are the average of both approximations.

Multiply instruction timing formulas (except for multiply decimal) for both the basic machine and for the high-speed multiply feature are based on the following assumptions:

1. No prenormalization is required.
2. Postnormalization is required 20 percent of the time.
3. Fixed-point operands have the expanded-half-word format.
4. For multiply (MXR) instruction only, neither the high-order nor the low-order 56 bits of an extended-precision 112-bit fraction equal 0.

The timing formulas give the time for each instruction in microseconds. In addition, the following time must be allowed for:

1. The equations assume that all fetches are handled by reference to buffer storage. If the block must be fetched from the processor storage, add 1.3 microseconds for the first doubleword, plus .08 microseconds for each of the three following doublewords. (Processor storage is assumed to be four-way interleaved.)
2. The equations assume the general registers needed for address calculations are available when needed. If one instruction modifies a general register and a subsequent instruction uses it for an address calculation, at least 0.24 microsecond must be allowed for the intervening instructions.

Instructions marked in the table with an asterisk have special timing assumptions listed following the table.

Terms used in formulas are in "Legend for System/370 Model 165 Timings."

AVERAGE TIMING FORMULAS

Instruction	*Format*	*Mnemonic*	*Time(Microseconds)*
Add	RR	AR	0.08
Add	RX	A	0.16 + 2.54BA1
Add Decimal*	SS	AP	1.00 + 0.11N1 + 0.03N2
Add Halfword	RX	AH	0.16 + 2.28BA1
Add Logical	RR	ALR	0.08
Add Logical	RX	AL	0.16 + 2.54BA1
Add Normalized (Extended)	RR	AXR	1.57
Add Normalized (Long)	RR	ADR	0.30
Add Normalized (Long)	RX	AD	0.30 + 2.90BA1
Add Normalized (Short)	RR	AER	0.38
Add Normalized (Short)	RX	AE	0.38 + 2.46BA1
Add Unnormalized (Long)	RR	AWR	0.38

Instruction	*Format*	*Mnemonic*	*Time(Microseconds)*
Add Unnormalized (Long)	RX	AW	0.38 + 2.90BA1
Add Unnormalized (Short)	RR	AUR	0.46
Add Unnormalized (Short)	RX	AU	0.46 + 2.46BA1
AND	RR	NR	0.08
AND	RX	N	0.16 + 2.54BA1
AND	SI	NI	0.48

AND*	SS	NC	L2 = 0.08(10.2 + 1.375N)
			L3 = 1.82N
			L4 = 0.76

Branch and Link	RR	BALR	0.19 + 0.21F1
Branch and Link	RX	BAL	0.43
Branch on Condition	RR	BCR	0.32 + 0.08F1 - 0.08F3
Branch on Condition	RX	BC	0.32 + 0.10F1 - 0.07F3
Branch on Count	RR	BCTR	0.40 - 0.08F1
Branch on Count	RX	BCT	0.40-0.05F1
Branch on Index High	RS	BXH	0.48-0.06F1
Branch on Index Low or Equal	RS	BXLE	0.48-0.06F1
Compare	RR	CR	0.08
Compare	RX	C	0.16 + 2.54BA1
Compare Decimal*	SS	CP	0.81 + 0.09N1 + 0.03N2
Compare Halfword	RX	CH	0.16 + 2.28BA1
Compare Logical	RR	CLR	0.08
Compare Logical	RX	CL	0.16 + 2.54BA1
Compare Logical	SI	CLI	0.16
Compare Logical*	SS	CLC	0.51 + 0.08K2 + 05N
Compare Logical Characters Under Mask	RS	CLCM	0.16
Compare Logical Long	RR	CLCL	0.8 + 0.08 (22 + 6CL1 + K2) CL2
Compare (Long)	RR	CDR	0.88
Compare (Long)	RX	CD	0.24 + 2.90BA1
Compare (Short)	RR	CER	0.32
Compare (Short)	RX	CE	0.32 + 2.46BA1
Convert to Binary	RX	CVB	0.08 (5 + P9) + 3.06BA1
Convert to Decimal	RX	CVD	0.08 (8 + 2P10) + 1.42BA1
Divide	RR	DR	1.96
Divide	RX	D	1.96 + 2.46BA1
Divide Decimal*	SS	DP	2.26 + 0.81 (N1 - N2) (N2 - 1)
Divide (Long)	RR	DDR	2.65
Divide (Long)	RX	DD	2.65 + 2.82BA1
Divide (Short)	RR	DER	1.64
Divide (Short)	RX	DE	1.64 + 2.46BA1
Edit*	SS	ED	0.73 + 0.29N
Edit and Mark*	SS	EDMK	1.09 + 0.29N
Exclusive OR	RR	XR	0.08
Exclusive OR	RX	X	0.16 + 2.54BA1
Exclusive OR	SI	XI	0.48

Exclusive OR*	SS	XC	L2 = 0.08 (10.2 + 1.375N)
			L3 = 1.82N
			L4 = 0.76
			L5 = 1.28 + 0.05(N-1) + 0.08S19

Execute	RX	EX	0.56 + 3.46BA2
Halt I/O	SI	HIO	1.52 + 0.32B5
Halt Device	SI	HDV	1.52 + 0.32B5
Halve (Long)	RR	HDR	0.40

Instruction	*Format*	*Mnemonic*	*Time(Microseconds)*	
Halve (Short)	RR	HER	0.48	
Insert Character	RX	IC	0.16	
Insert Character Under Mask	RS	ICM	0.16 + 0.8M	
Insert Storage Key	RR	ISK	2.8	
Load	RR	LR	0.08	
Load	RX	L	0.16 + 2.54BA1	
Load Address	RX	LA	0.16	
Load and Test	RR	LTR	0.08	
Load and Test (Long)	RR	LTDR	0.08	
Load and Test (Short)	RR	LTER	0.08	
Load Complement	RR	LCR	0.08	
Load Complement (Long)	RR	LCDR	0.08	
Load Complement (Short)	RR	LCER	0.08	
Load Control	RS	LCTL	0.24 + 0.2 + 0.08LC2	
Load Halfword	RX	LH	0.16 + 2.36BA1	
Load (Long)	RR	LDR	0.08	
Load (Long)	RX	LD	0.16 + 2.90BA1	

Load Multiple	RS	LM	LM1 = 0.44 + 0.08GR
			LM2 = 2.42 + 52GR	

Load Negative	RR	LNR	0.08	
Load Negative (Long)	RR	LNDR	0.08	
Load Negative (Short)	RR	LNER	0.08	
Load Positive	RR	LPR	0.08	
Load Positive (Long)	RR	LPDR	0.08	
Load Positive (Short)	RR	LPER	0.08	
Load PSW	SI	LPSW	0.98	
Load Rounded (Extended to Long)	RR	LRDR	0.56	
Load Rounded (Long to Short)	RR	LRER	0.32	
Load (Short)	RR	LER	0.08	
Load (Short)	RX	LE	0.16 + 2.54BA1	
Move	SI	MVI	0.32	
Move*	SS	MVC	0.48 + 0.08(3N) 0.98 0.56 + 0.24N	
Move Long	RR	MVCL	1.36 + 0.08 MV2(29 + 0.5MV4) + 0.08 S19	

Move Numerics*	SS	MVN	L2 = 0.08(10.2 + 1.375N)
			L3 = 1.82N
			L4 = 0.76	

Move with Offset*	SS	MVO	0.64 + 0.17N1 + 0.01N2	

Move Zones*	SS	MVZ	L2 = 0.08 (10.2 + 1.375N)
			L3 = 1.82N
			L4 = 0.76	

			Basic	HS MPY
Multiply	RR	MR	0.78	0.42
Multiply	RX	M	0.78 + 2.46BA1	0.42 + 2.46BA1
Multiply Decimal*	SS	MP	0.17 + 0.18N2 + 0.4 (N1-N2)(N2 + 3.3)	
Multiply (Extended)	RR	MXR	9.90	3.74
Multiply Halfword	RX	MH	0.80 + 2.36BA1 0.48 + 2.36BA1	
Multiply (Long)	RR	MDR	1.87	0.61
Multiply (Long)	RX	MD	1.87 + 2.82BA1	0.61 + 2.90BA1
Multiply (Long to Extended)	RR	MXDR	2.11	0.78
Multiply (Long to Extended)	RX	MXD	2.11 + 2.82BA1	0.78 + 2.90BA1

Instruction	*Format*	*Mnemonic*	*Time(Microseconds)*	
Multiply (Short)	RR	MER	1.15	0.45
Multiply (Short)	RX	ME	1.15 + 2.46BA1	0.45 + 2.54BA1
OR	RR	OR	0.08	
OR	RX	O	0.16 + 2.54BA1	
OR	SI	OI	0.48	

OR*	SS	OC	L2 = 0.08 (10.2 + 1.375N)
			L3 = 0.76
			L4 = 0.76	

Pack*	SS	PACK	0.85 + 0.17N1	
Read Direct	SI	RDD	1.44 + ED	
Set Clock	SI	SCK	0.40	
Set Program Mask	RR	SPM	0.16	
Set Storage Key	RR	SSK	2.16(+ 0.5(N-1) + S19	
Set System Mask	SI	SSM	0.48	
Shift and Round Decimal	SS	SRP	0.32 + 0.08 (27 + 2N1)	
Shift Left Double	RS	SLDA	0.12	
Shift Left Double-Logical	RS	SLDL	0.12	
Shift Left Single	RS	SLA	0.12	
Shift Left Single-Logical	RS	SLL	0.12	
Shift Right Double	RS	SRDA	0.12	
Shift Right Double-Logical	RS	SRDL	0.12	
Shift Right Single	RS	SRA	0.12	
Shift Right Single-Logical	RS	SRL	0.12	
Start I/O	SI	SIO	1.52 + 0.32B5	
Start I/O Fast Release	SI	SIOF	1.52 + 0.32B5	
Store	RX	ST	0.32 + 3.16BA1	
Store Channel ID	SI	STIDC	1.52 + 0.32B5	
Store Character	RX	STC	0.32	
Store Character Under Mask	RS	STCN	0.16 + 0.08 (M + 1)	
Store Clock	SI	STCK	0.88	
Store Control	RS	STCTL	0.24 + 0.12 + 0.2 (SL2)	
Store CPU ID	SI	STIDP	0.64	
Store Halfword	RX	STH	0.32 + 2.89BA1	
Store (Long)	RX	STD	0.32 + 3.64BA1	
Store Multiple*	RS	STM	0.32 + 0.08 (2GR + S19) STM2 = 2.0 + 1.38GR	
Store (Short)	RX	STE	0.32 + 3.24BA1	
Subtract	RR	SR	0.08	
Subtract	RX	S	0.16 + 2.54BA1	
Subtract Decimal*	SS	SP	1.00 + 0.11N1 + 0.032N2	
Subtract Halfword	RX	SH	0.16 + 2.28BA1	
Subtract Logical	RR	SLR	0.08	
Subtract Logical	RX	SL	0.16 - 2.54BA1	
Subtract Normalized (Extended)	RR	SXR	1.57	
Subtract Normalized (Long)	RR	SDR	0.30	
Subtract Normalized (Long)	RX	SD	0.30 + 2.90BA1	
Subtract Normalized (Short)	RR	SER	0.38	
Subtract Normalized (Short)	RX	SE	0.38 + 2.46BA1	
Subtract Unnormalized (Long)	RR	SWR	0.38	
Subtract Unnormalized (Long)	RX	SW	0.38 + 2.90BA1	
Subtract Unnormalized (Short)	RR	SUR	0.46	
Subtract Unnormalized (Short)	RX	SU	0.46 + 2.46BA1	
Supervisor Call	RR	SVC	2.08	
Test and Set	SI	TS	3.2	
Test Channel	SI	TCH	1.52 + 0.32B5	

Instruction	*Format*	*Mnemonic*	*Time(Microseconds)*
Test I/O	SI	TIO	1.52 + 0.32B5
Test under Mask	SI	TM	0.16
Translate*	SS	TR	0.79 + 0.25N
Translate and Test*	SS	TRT	0.89 + 0.24B + 0.07N
Unpack*	SS	UNPK	0.95 + 0.08N1
Write Direct	SI	WRD	1.12
Zero and Add*	SS	ZAP	1.22 + 0.07N1 + 0.2N2

Special Timing Assumptions

Instructions marked with an asterisk in the average timing table assume the following conditions.

Add Decimal (AP)

1. Ten percent probability that the result is 0.
2. Twenty-five percent probability that the first operand is specified to be longer than the second operand.
3. Operand buffer loading time of 0.24 microsecond is included.

AND (NC)

1. L2 is used for the normal case and if N is greater than 1.
2. L3 is used for the word-overlap case.
3. L4 is used for one-byte operands.
4. L5 applies only to the exclusive OR instruction and is used if N is greater than 1, the low-order three bits of the source address are the same as the low-order three bits of the destination address, and both operands are in the same doubleword.
5. Fifty percent probability that N is greater than 8.

Compare Logical (CLC)

The operands are assumed to lie across three doubleword boundaries.

Compare Decimal (CP)

The same assumptions as for add decimal (AP)

Divide Decimal (DP)

The average divisor contains four bytes.

Edit (ED)

1. Twenty-five percent probability that a fill character will be stored.
2. Seventy-five percent probability that other than a fill character will be stored.
3. An average time of 0.29 microsecond is allowed per pattern character.

Edit and Mark (EDMK)

The same assumptions as for edit instruction as well as 90 percent probability that the address will be marked.

Exclusive OR (XC)

The same assumptions as for AND (NC).

Move (MVC)

1. V10 is used if N is greater than 8.
2. V11 is used if N is less than or equal to 8.
3. V12 is used for the word-overlap case.

Move Numerics (MVN)

The same assumptions as for AND (NC).

Move with Offset (MVO)

1. Overlap is not considered.
2. An average time of 0.08 microsecond is allowed for padding.
3. Whether an operand crosses a doubleword boundary is not considered.

Move Zones (MVZ)

The same assumption as for AND (NC).

Multiply Decimal (MP)

The second operand lies across (N2-1)/8 doubleword boundaries.

OR (OC)

The same assumptions as for AND (NC).

Pack (PACK)

1. Overlap is not considered.
2. Padding time is included.
3. Both the first and the second operands lie across a doubleword boundary.

Store Multiple (STM)

1. STM1 is used if boundary alignment is not necessary.
2. STM2 is used if boundary alignment is necessary.
3. The number of registers specified is equally likely over the range of 1-16.

Subtract Decimal (SP)

The same assumptions as for add decimal (AP).

Translate (TRT)

Assume no overlap.

Translate and Test (TR)

Seventy-five percent probability that nonzero function byte will be found.

Unpack (UNPK)

1 Overlap is not considered.
2. Both the first and the second operands lie across a doubleword boundary.

Zero and Add (ZAP)

The same assumptions as for add decimal (AP) and both operands lie across a doubleword boundary.

BYTE-ORIENTED OPERAND FEATURE TIMES

The byte-oriented operand feature allows addressing of operands of unprivileged RX and RS format instructions on any byte boundary of processor storage. However, significant performance degradation is possible when storage operands are not positioned at addresses that are integral multiples of the operand length. When a user takes advantage of the feature and does not observe the usual boundary constraints, instruction times are substantially modified according to the "BA" terms in the instruction timing formulas. To ensure optimum performance, storage operands should be aligned on integral boundaries and use of unaligned operands should be reserved for exceptional cases.

LEGEND FOR SYSTEM/370 MODEL 165 TIMINGS

B5 = 1 plus the greatest integer in (T/0.32), where T is the time in microseconds from the time that select channel leaves the CPU until release enters the CPU.

BA1 = 1.00 if operand does not begin on integral boundary appropriate to its length.
= 0 if operand is on integral boundary.

BA2 = 1 if operand of subject instruction requires boundary alignment.
= 0 otherwise.

Note: Boundary alignment (BA) refers to the functioning of the byte-oriented operand feature.

CL1 = 0 if either first operand length or second operand length is zero.

CL1 = 1 otherwise.

CL2 = 0 if both first operand length and second operand length are zero.

CL2 = 1 otherwise.

E = Time for the subject instruction that is executed by the execute instruction.

ED = External delay.

F1 = 1 if the branch is successful.
= 0 otherwise.

F3 = 1 if the mask equals F (hex).
= 0 otherwise.

GR = Number of general registers store or loaded.

K2 = Number of byte comparisons required until a no-compare is found.

K5 = 1 if a quadword boundary is crossed by the destination operand.
= 0 otherwise.

L2-L5 = Detailed under "AND (NC)" in special timing assumptions.

LC2 = No. of control registers loaded.

M = 2 if mark is 0.
= 7 otherwise.

MV2 = 0 if destructive overlap or first operand length = 0.

MV2 = 1 otherwise.

MV3 = No. of bytes processed up to the first doubleword boundary of the first operand.

MV4 = No. of bytes to be processed–MV3.

N = Total number of bytes in the first operand for those instructions with one length field.

N1 = Total number of bytes in the first operand (destination).

N2 = Total number of bytes in the second operand (source).

P9 = Number of significant decimal digits.

P10 = Number of significant hexadecimal digits.

S19 = Seven times the truncated value of the number of doublewords stored, divided by four. (If the number of doublewords stored is greater than four.)
= 0 otherwise.

SL2 = No. of control registers stored.

STM1,2 = Detailed under "Store Multiple (STM)" in special timing assumptions.

V10-V12 = Detailed under "Move (MVC)" in special timing assumptions.

Index

READER'S COMMENT FORM

System/370 Model 165
Functional Characteristics

GA22-6935-0

Your comments about this publication may be helpful to us. If you wish to comment, please use the space provided below, giving specific page and paragraph references.

Please do not use this form to ask technical questions about the system or equipment or to make requests for copies of publications; this only delays the response. Instead, make such inquiries or requests to your IBM representative or to the IBM Branch Office serving your locality.

Reply requested

Yes □

No □

Name ____________________

Job Title ____________________

Address ____________________

____________________ Zip __________

WTC users must add postage.

GA22-6935-0

YOUR COMMENTS, PLEASE . . . .

This manual is a reference source for systems analysts, programmers and operators of IBM systems. Your comments will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Note: Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

CUT ALONG THIS LINE

fold fold

FIRST CLASS
PERMIT NO. 419
POUGHKEEPSIE, N.Y.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . . .

IBM CORPORATION
P.O. BOX 390
POUGHKEEPSIE, N.Y. 12602

ATTENTION: CUSTOMER MANUALS, DEPT. B98

fold fold

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

GA22-6935-0

System/370 Model 165 Functional Characteristics   Printed in U.S.A.   GA22-6935-0

IBM

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
(USA Only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)