File No. S360-01
Form A22-6916-0

IBM Systems Reference Library

IBM System/360 Model 85

Functional Characteristics

This publication describes the organization and the functional characteristics of the IBM System/360 Model 85, 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 read the *IBM System/360 Principles of Operation*, Form A22-6821. Other related literature is referenced by form number and briefly described in *IBM System/360 Bibliography*, Form A22-6822.

First Edition

Specifications contained herein are subject to change from time to time. Any such change will be reported in subsequent revisions or Technical Newsletters.

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 Development Division, Product Publications, Dept. B98, P.O. Box 390, Poughkeepsie, N.Y. 12602. A form is provided at the back of this publication for reader's comments. If the form has been removed, comments may be sent to the above address.

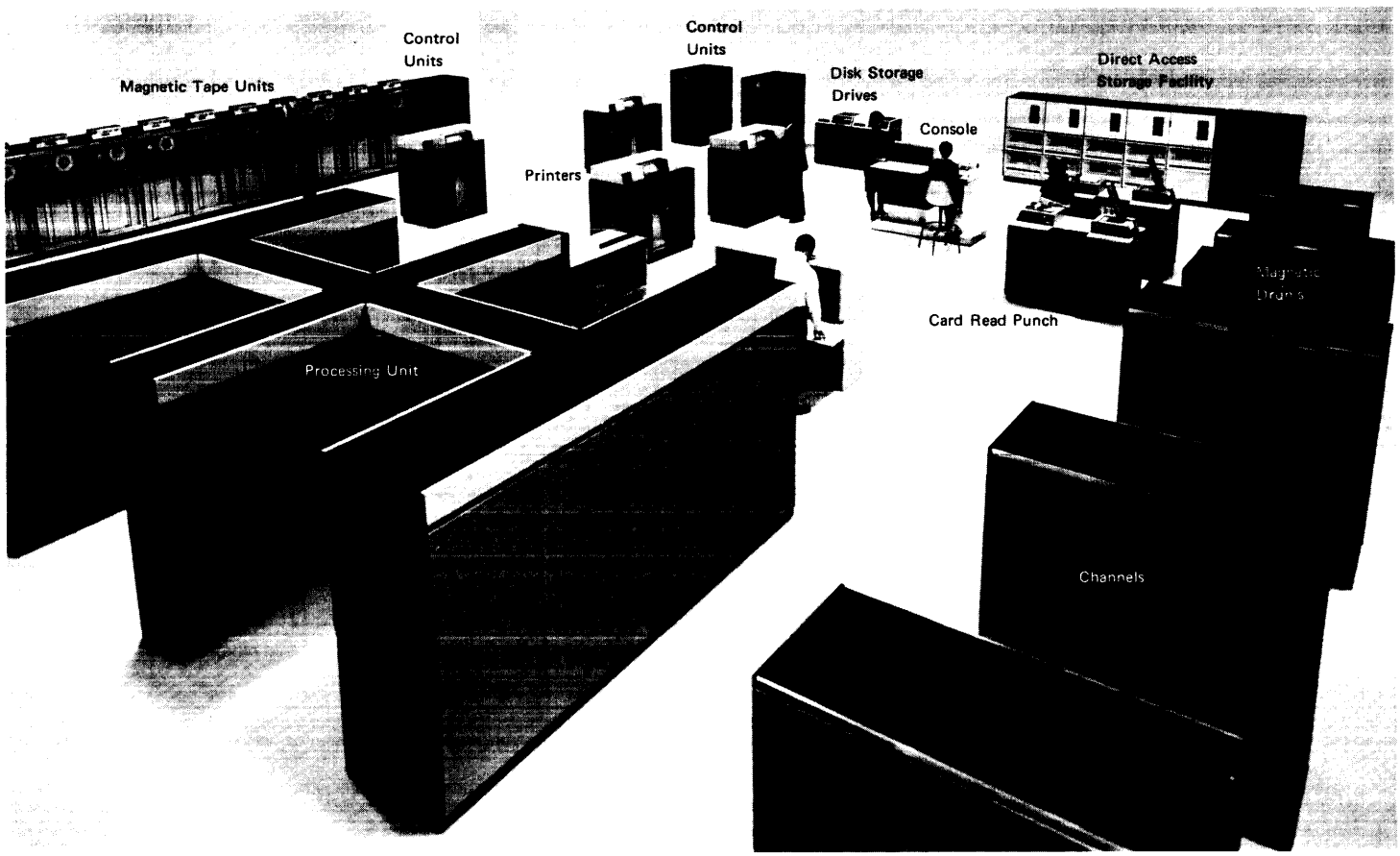
Contents

System Description 5
System Components 5
Main Storage 9
  Interleaving 10
2085 Processing Unit 10
  Instruction Unit 10
  Execution Unit 12
  Storage Control Unit 12
  Control Storage 14
  Instruction Retry 14
  Maintenance Controls 14
  Machine Check Interruption 14
  Priority of Interruptions 15
Channels 15
  2860 Selector Channel 16
  Channel-to-Channel Adapter Feature 16
  2870 Multiplexer Channel 16

System Controls 17
System Control Functions 17
  System Reset 17
  Store and Display 17
  Initial Program Loading 17
System Control Panel 18
  Operator Controls 18
  Operator Intervention Controls 20
  CE Controls 23
Usage Metering 24

Instruction Times 25
  Timing Considerations 25
  Timing Assumptions 25
Average Timing Formulas 25
  VFL Timing Assumptions 30
Byte-Oriented Operand Feature Times 32
Legend for System/360 Model 85 Timings 32

Index 34



IBM System/360 Model 85

The IBM System/360 Model 85 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 85 are two standard features in the CPU: a high-speed buffer that stores currently used sections of main storage for faster accessing, and an extended-precision floating-point feature that provides for arithmetic operations on 16-byte, floating-point operands and for rounding from extended to long format and from long to short. Speed in the accessing of storage is further increased by the use of multiple, interleaved main storage elements.

The performance of the System/360 Model 85 is increased even more by high-speed multiply, an optional feature that allows both fixed-point and floating-point multiply instructions to be performed faster.

Programming support is provided by the System/360 Operating System (OS), and the system is planned to take advantage of the performance gains possible because of this support, particularly when multiprogramming is used. The large storage capacities made available by the system allow the efficient use of the OS option called multiprogramming with a variable number of tasks (MVT).

In the Model 85, separate units, each mainly autonomous, may be operating concurrently: main storage, the instruction preparation unit, the execution unit, and the channels. Data flow and system statistics are shown on Figure 1. Although processing operations are performed in the Model 85 in an overlapped fashion, no special optimization is required in preparing programs for CPU processing.

The Model 85 has a major machine-cycle time of 80 nanoseconds. Main-storage data flow is 16 bytes (one quadword) in parallel. Main storage cycle time is either 960 or 1,040 nanoseconds, depending on the model. However, when high-speed buffer storage is used, the effective system storage cycle becomes one-third to one-quarter of the actual main storage cycle.

Four capacities of main storage are available: Model I (524,288 bytes), Model J (1,048,576 bytes), Model K (2,097,152 bytes), and Model L (4,194,304 bytes). A special error checking and correction code is implemented in the storage units.

For input/output operations, the system may have one multiplexer channel (IBM 2870 Multiplexer Channel) and as many as six selector channels (two units of IBM 2860 Selector Channel Model 3). Attachable input/output devices, generally the same as those for the Model 75, are given in the *IBM System/360 Input/Output Configurator,* Form A22-6843.

System Components

The major components of a Model 85 are an IBM 2085 Processing Unit, an IBM 2365 Processor Storage Model 5, an IBM 2385 Processor Storage Models 1 and 2, an IBM 2860 Selector Channel, and an IBM 2870 Multiplexer Channel. Note that the 2365 and 2385 storages may not be intermixed on a given system. Input/output (I/O) devices are attached to the channels by control units. (See Figure 2.)

The standard features for any IBM 2085 Processing Unit include:

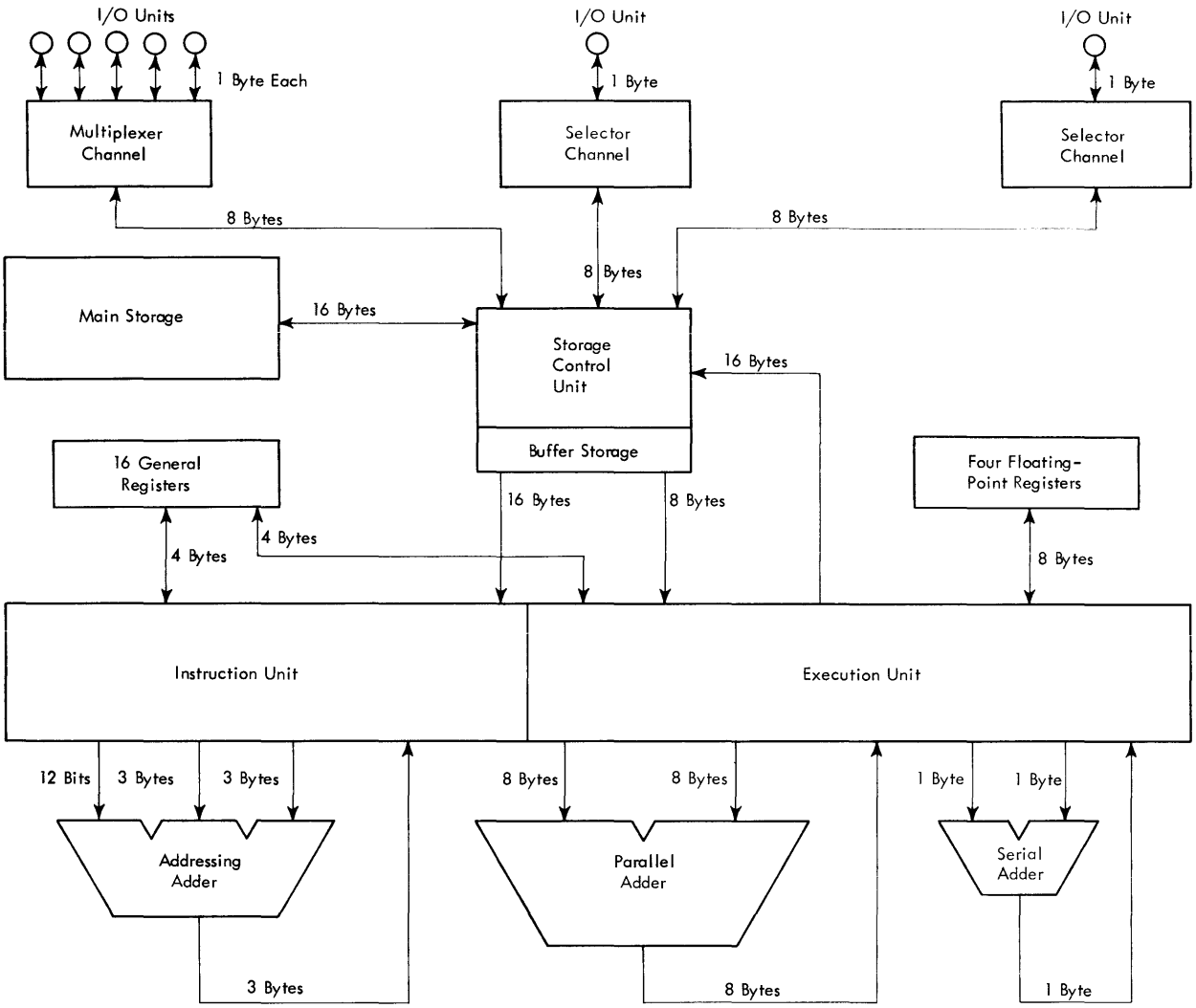
Universal instruction set
Extended-precision floating-point feature
Byte-oriented operand feature
Buffer storage (16,384 bytes)
Protection features (store and fetch protection)
Direct control feature
2860 Selector Channel attachment
2870 Multiplexer Channel attachment

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 main 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



Element	Data Width (Bytes)	Performance	Comments
2365-5 Processor Storage	16	1,040-nanosecond cycle (Note 1)	Models I and J
2385-1, 2 Processor Storage	16	960-nanosecond cycle (Note 1)	Models K and L
Buffer storage	16	80-nanosecond cycle	16,384 byte capacity
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 (I/O interface)	8 bytes to storage
2870 Multiplexer Channel	1	110 kb to 670 kb (aggregate)	8 bytes to storage (kb = 1,000 bytes/second)
Burst mode	1	110 kb (Note 2)	---
Multiplex mode	1	110 kb (Note 2)	---
Selector subchannels 1-3	1	180 kb each (I/O interface)	---
Selector subchannel 4	1	100 kb (I/O interface)	---

Notes:

1. Main storage effective cycle time is greatly reduced by the buffer storage function.
2. Aggregate 192-subchannel rate only; reduced by concurrent selector subchannel operation.

Figure 1. Model 85 Data Flow Diagram and System Statistics

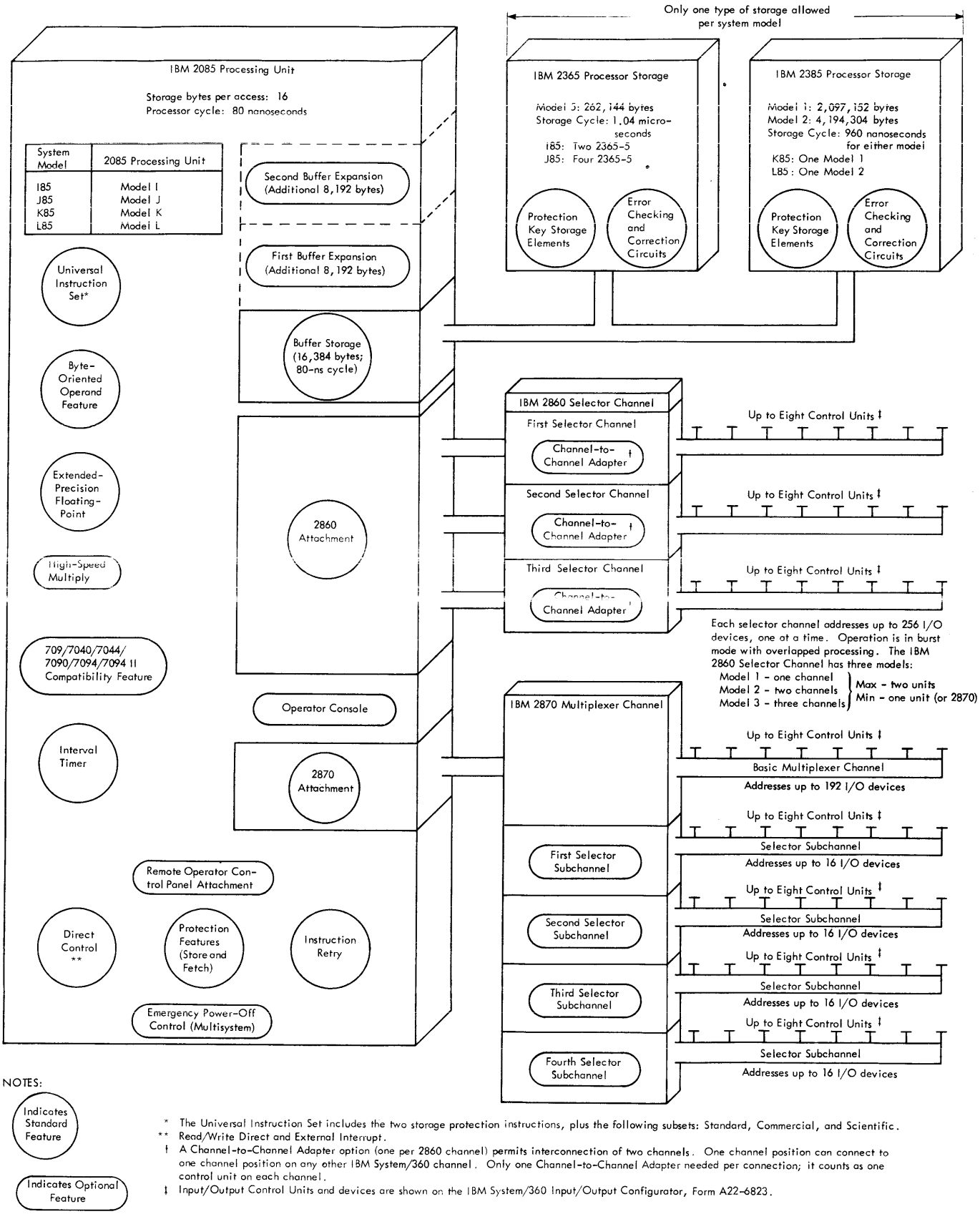


Figure 2. System/360 Model 85 Configurator

speed. Special consideration should be given to the information in "Byte-Oriented Operand Feature Times." Severe performance degradation may result when boundaries are unaligned. 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/360 Principles of Operation*, Form A22-6821.

The optional features for any 2085 Processing Unit include:

First buffer expansion
Second buffer expansion
High-speed multiply
Operator console
Emergency power-off (multisystem)
709/7040/7044/7090/7094/7094II compatibility
Remote operator control panel

The first and second buffer expansions each add 8,192 bytes to the capacity of the high-speed buffer storage. The first expansion is a prerequisite for installation of the second.

The use of the high-speed multiply feature allows both fixed-point and floating point multiply instructions to be performed faster. With the feature installed, a long-precision, floating-point multiply can be done in less than 600 nanoseconds; a fixed-point multiply in less than 450 nanoseconds.

The operator console links to the system control panel (on the maintenance console). The cathode-ray tube (CRT) display on the maintenance console (Figure 3) is shared by the operator console. Standard features for the operator console are an alphameric keyboard (on the reading board in front of the system control panel), a character generator, and a display buffer. It also has an audible alarm that may be used by the system to alert the operator.

Emergency power-off (EPO) control is required on only one of the processing units, normally on the largest, in any installation composed of more than one cable-connected processing unit and/or cable-connected units that can be operated off-line. The emergency power-off feature interconnects EPO switches to

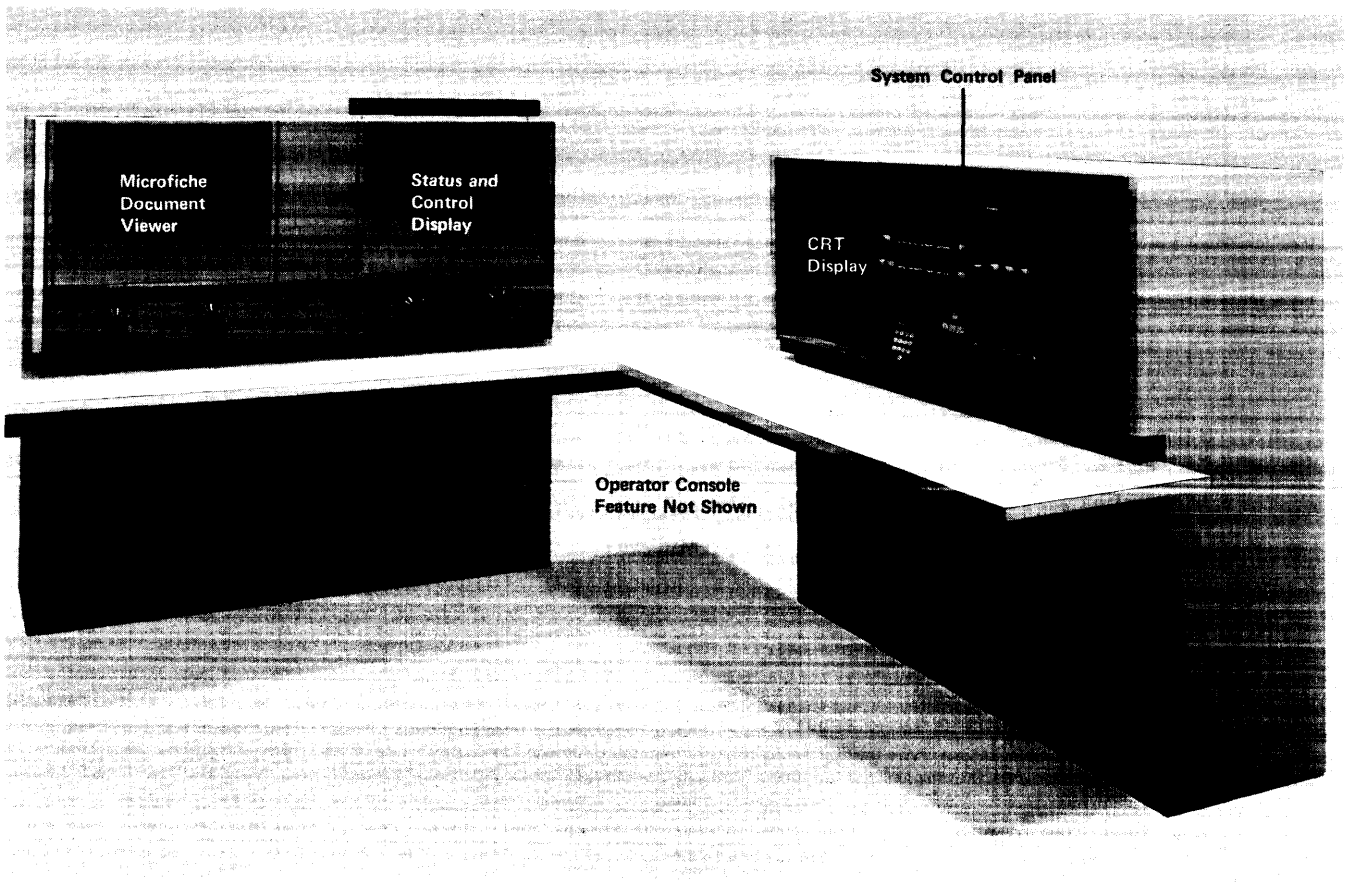


Figure 3. Model 85 Maintenance Console

provide, in effect, a single EPO switch in a room or area.

The 709/7040/7044/7090/7094/II compatibility feature, in conjunction with an emulator program, allows the Model 85 to execute programs and programming systems originally written for other systems. The compatibility feature adds special instructions and internal logic to the Model 85. Currently available emulator programs use these facilities and the universal instruction set to simulate either 7040/7044-type or 7090-type instructions.

The remote operator control panel attachment allows one set of the controls and indicator lights on the operator control panel to be duplicated as a remote panel on a stand-alone console such as the IBM 2150 Console or the IBM 2250 Display Unit Model 1.

A variety of control units and I/O devices are available for use with the Model 85. Descriptions of specific I/O devices are in separate publications. Configurators for I/O devices and for system components are also available. Refer to the *IBM System/360 Bibliography*, Form A22-6822.

Main Storage

Any of three different storages can be used in a Model 85: an IBM 2365 Processor Storage Model 5, an IBM 2385 Processor Storage Model 1, or an IBM 2385 Proc-

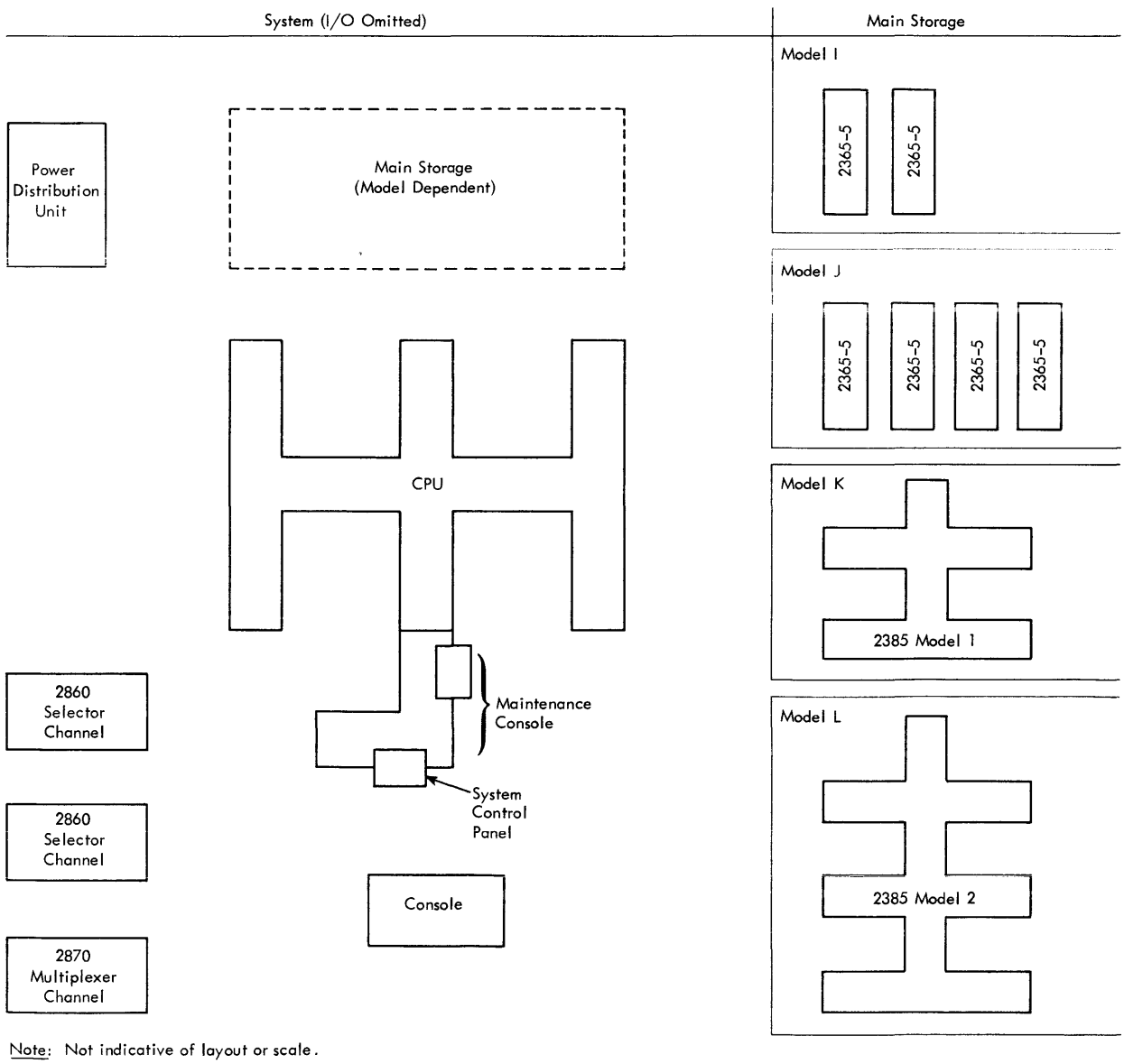


Figure 4. Model 85 System Elements

essor Storage Model 2. No intermixing is allowed; the use of one type of storage precludes the use of the others on the same system.

The four possible CPU/main-storage combinations in the Model 85 are: I85, J85, K85, and L85. They vary in the number and types of main storage units used with the 2085 Processing Unit (Figure 4).

CPU MODEL	MAIN STORAGE TYPE	CAPACITY (BYTES)	INTER-LEAVING
2085I	Two of 2365 Model 5	524,288	Two-way
2085J	Four of 2365 Model 5	1,048,576	Four-way
2085K	One of 2385 Model 1	2,097,152	Four-way
2085L	One of 2385 Model 2	4,194,304	Four-way

Interleaving

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

In four-way interleaving, four functionally independent storage units make up main storage. Assume the four units to be: A, B, C, and D. Storage location 0 is in unit A, location 1 is in unit B, location 2 is in unit C, and location 3 is in unit D. Storage location 4 is in unit A, and the address-distribution sequence continues through all available storage locations.

A storage unit, once selected for a storage reference, cannot be referenced again until the total storage-cycle time passes. A main storage consisting of one storage unit allows storage references to be made only once every storage-cycle time. With four-way interleaving, a storage reference can be made any time that the functionally independent storage unit containing the requested storage location is not busy (has not completed a storage cycle after being selected).

2085 Processing Unit

The IBM 2085 is the central processing unit (CPU) for the Model 85. The physical complex that contains the CPU also includes several power frames, a coolant distribution unit, the maintenance console (which includes a system control panel with CRT display, a status and control display consisting of a microfiche-type projection display coupled with incandescent indicators, and a microfiche document viewer for examining reference material pertinent to system operation), and a stand-alone power distribution unit. A motor-generator set is also required and may be in an area other than the machine room.

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

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.

Programming Note: Certain conditions must be satisfied before an instruction can be successfully decoded:

1. A branch instruction cannot be in progress if this instruction is to be a branch.
2. An SS format instruction cannot be in either the instruction unit or the execution unit if this instruction is to be a branch or RR format type.
3. There must be no instruction in either the instruction unit or the execution unit that will modify general registers to be used by this instruction.

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 the AND (NI), exclusive OR (XI), and OR (OI) instructions, the calculated address is used both as a source and a destination 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. When the branch is executed, if the estimate is correct, 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.

Programming Note: 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).

The following are treated as no-ops and are not recognized as branches:

Branch on condition (BC) where M equals 0.
Branch on condition (BCR) where M or R2 equals 0.
Branch and link (BALR) where R2 equals 0.
Branch on count (BCTR) where R2 equals 0.

When a fetch-type instruction follows a store-type instruction, it is possible that the fetch request made by the instruction unit will precede the store request made by the execution unit. To ensure that the data fetched is the changed data when both requests reference the same storage address, the out-of-sequence fetch is ignored, and a new fetch is made after the store is completed.

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 modified. 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 processor 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 main 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.

A protection or addressing exception resulting from a store operation, however, is sent to the execution unit interruption controls immediately. This results in an imprecise interruption (instruction length code equals 0). For an imprecise interruption, the instruction address in the old PSW does not represent the next instruction address.

Execution Unit

The execution unit handles the execution of the instructions for the Model 85, and it has the capability of processing a new instruction every cycle. Particular emphasis is given to optimization of the fixed-point and floating-point instructions.

The execution unit contains two eight-byte data buffers for prefetched operands, four eight-byte data registers, and a 16-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 85 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 main storage are physically separate from the CPU and are functionally attached to the system through the storage control unit. All references to storage from the CPU channels are controlled by the storage control unit. The amount of information sent to or fetched from storage on one reference, sometimes called the physical storage word, is one quadword (16 bytes).

The storage control unit can initiate a new storage cycle every machine cycle (80 nanoseconds) on successive store, fetch, and test-and-set requests. Successive storage references made by channels may be delayed for three to nine machine cycles.

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 main 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 requests during 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, main storage is accessed.

For store operations, the buffer storage control stores data into buffer storage only if the referenced data is in buffer storage. For *all* store operations, the buffer control stores data into main storage, regardless of whether buffer storage is accessed (updated).

Main and buffer storage are arbitrarily partitioned into a number of 1,024-byte pages, on 1,024-byte boundaries. The standard number of pages in buffer storage is 16, and the number in main storage is 500 or 1, 2, or 4 thousand, depending on size of main storage. During operation, a correspondence is set up that relates each page in the buffer storage to a page in main storage. Each time the CPU makes a fetch, the buffer storage control determines whether there is a buffer page corresponding to the addressed main storage page. If none is found, one of the buffer pages is automatically assigned to the page that the CPU addressed.

A number of references to main storage are required to load an entire page into buffer storage. To handle this loading more effectively, each page is divided into 16 blocks of 64 bytes each; and when a buffer page is newly assigned, the information loaded into the buffer is the particular block that the CPU fetch addressed. Subsequently, as the CPU issues fetch requests to other blocks in that page, those blocks are also loaded into buffer storage.

When a CPU fetch dictates a block load, the first main storage 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 additional main storage fetches needed to complete the block load are initiated every machine cycle thereafter if the main storage units are not busy and as soon as possible otherwise.

Channels store data into, but do not fetch data from, buffer storage. This condition allows maximum usage of the high-speed buffer storage by the CPU and minimizes main storage interference because of I/O transfers. 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 main storage data is updated. If the referenced data is not in buffer storage, only main storage is updated. Channel fetch requests are made only to main storage.

Because data in buffer storage must be current, any buffer page can be reassigned to any area in main storage. Priority of reassignment is based on usage. Each time data within a buffer page is referenced, that page is moved to the top of a logic-controlled assignment list. All other pages are moved one position lower on the list. Note that movement of a page in the list involves no storage data transfer. When all 16 buffer pages are assigned and the CPU makes a fetch request to a storage address not in the buffer, the buffer page longest unused and lowest on the assignment list is cleared and reassigned to the referenced main storage area.

I/O Channel Control

The I/O channel control receives and processes channel storage requests. When a channel buffer becomes available, the control resolves the priority among the outstanding channel requests and sends an acceptance to the channel that has top priority. This channel responds with appropriate information, which is then placed in one of the two channel buffers. When the information is in the channel buffer, a request is made to the main storage control. For a store operation, a request is made simultaneously to the buffer storage control. After the request is honored by the main storage control, the channel buffer just used is freed to handle another channel request.

Main Storage Control

The main 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 main storage control generates from-and-to information and makes the main storage reference. During the latter part of the storage cycle, the main 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.

For a store operation, data parity errors cause a check indication and allow the faulty data to be stored; control parity errors cause a check indication and inhibit the store.

For a fetch operation, data parity errors cause a check indication and allow the faulty data to be fetched; control parity errors cause a check indication and cause all 0s with valid parity to be fetched.

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.

Control Storage

The control storage in the Model 85 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 control word format. More than 2,500 control word addresses are provided in the basic CPU: 2,000 in read-only storage and 500 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.

The writable control storage is included to provide microdiagnostic capability. It is also used to contain the compatibility feature control words. When the compatibility feature is installed (factory installation only), 500 control words of writable control storage are included in the basic machine.

Instruction Retry

For most instructions, retry involves restarting the instruction unit at the address of the instruction to be retried, and completely re-executing the instruction.

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

Instruction retry, as implemented in the Model 85, can be considered in three steps: normal instruction execution, error stop with retry decision, and retry restore.

Normal Execution

During normal instruction execution, data that would be needed for a retry of that 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. The instruction unit provides the address of the next instruction to be executed after a successful branch or a PSW load. Retry controls update this value according to the length code of each instruction successfully executed. When sequential instruction processing is abandoned, the instruction unit again provides an instruction address.

Error Stop

When an error is detected, a stop occurs (control is passed from the execution microprogram) and a retry decision is made by the retry controls. The decision involves the error type, the instruction type, and the current retry mode. If a decision to retry is made, a status log is performed. The status log stores the contents of the maintenance control area into main storage at location 128. This storage area is called the status table. The retry-restore microprogram 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 re-execution to begin. Re-execution restarts the instruction unit at the retry instruction address.

Maintenance Controls

The maintenance control area of the CPU executes the functions provided on the operator and maintenance consoles, 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 storage operating with the CPU, in a channel operating with the CPU, or in storage operating with a channel. The channel error can occur during I/O instruction time or I/O interruption time. The storage-with-channel error can occur at any time.

When an error is signaled, enough information is provided to direct and allow subsequent program ac-

tion. Two types of information are required: 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).

Machine Check Interruption Types

A machine check interruption 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 CPU activity and a logout of machine status. Examples of errors that cause a hard MCI are:

1. An unretriable 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 unretriable state.
3. An uncorrectable storage error on a CPU operation.
4. An error that is retried unsuccessfully.
5. An error that causes a CPU hangup.

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 error in storage on a *channel* operation.

Machine Check Interruption Code

The 16-bit interruption code (PSW bit positions 16-31) is stored when a machine check interruption occurs. The interruption code is *not* all 0s in the Model 85; instead, the code bits contain information about the nature and severity of the failure and the validity of the stored status.

Bit positions 16-26 of the interruption code contain 0s.

Bit position 27 containing 1 indicates the presence of an uncorrectable (multiple-bit) storage data error. The bit is set for both CPU and channel references. The CPU response to this condition varies with the requesting unit. For a channel reference, a soft MCI is taken, and code 10 (intercepted error) is set in the result field (bit positions 28-31). For CPU references, the result field may show code 9 (successful retry), code 1 (unsuccessful retry), or code 4 (unretriable), based on the result of the retry operation that follows the error.

Bit positions 28-31 in the MCI code are the result field.

The bit codes defined for a hard MCI are:

Code 1 (0001)	Unsuccessful retry
Code 4 (0100)	Unretriable

The bit codes defined for a soft MCI are:

Code 8 (1000)	Error checking and correction circuitry corrected an error
Code 9 (1001)	Successful retry
Code 10 (1010)	Intercepted error

Priority of Interruptions

The occurrence of a soft MCI condition does not cause an immediate interruption in the Model 85. 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/360 Principles of Operation*, Form A22-6821.

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 MCIs are disallowed when bit position 13 of the current PSW contains a 0. Logic-controlled retry is disabled and CPU errors are held pending. When MCIs 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 Program (RMP) option has the ability to distinguish between hard and soft machine checks, and to take appropriate action. RMP is available with MFT or MVT but not with PCP. Refer to appropriate System Reference Library (SRL) programming publications.

Channels

In the Model 85, the 2860 Selector Channel and the 2870 Multiplexer Channel are available. The selector and multiplexer channels provide for the attachment of I/O devices to the system. The channel relieves the

CPU of the task of communicating directly with the I/O devices and permits data processing to proceed concurrently with I/O operations.

Data is transferred a byte at a time between the I/O device and the channel. A standard channel-to-control-unit interface provides a uniform method of attaching control units to all channels. Data transfers between the channel and the storage control unit are eight bytes (one doubleword), in sequential words, for both selector and multiplexer channels.

2860 Selector Channel

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

Model 1	Provides one selector channel
Model 2	Provides two selector channels
Model 3	Provides three selector channels

In addition to one 2870 Multiplexer Channel, a total of six selector channels in the following combinations may be attached:

NO. OF CHANNELS REQUIRED	RECOMMENDED COMBINATIONS
1	One of 2860-1
2	One of 2860-2
3	One of 2860-3
4	One of 2860-3 and one of 2860-1 or two of 2860-2
5	One of 2860-3 and one of 2860-2
6	Two of 2860-3

At least one 2860 (any model) is required if no 2870 Multiplexer Channel is attached. If only a 2870 is attached, the first selector subchannel feature is required.

The selector channel permits data rates 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 minimal interference.

A maximum of eight control units can be attached to each selector channel. A 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.

Channel-to-Channel Adapter Feature

A channel-to-channel adapter is available as an optional feature. The adapter provides a path for operations to take place between two channels and synchronizes those operations. It may be used in a system to move blocks of data from one area in main storage to another area in main storage.

The adapter uses one control-unit position on each of the two channels. Only one of the two connected channels requires the feature, and in the Model 85 one adapter may be installed for each selector channel.

2870 Multiplexer Channel

The 2870 Multiplexer Channel provides for the attachment of a wide range of low-speed to medium-speed I/O control units and associated devices. One 2870 can be attached to the Model 85.

The multiplexer channel provides as many as 196 subchannels, including four optional selector subchannels. The basic multiplexer channel has 192 subchannels; it 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 may be operated.

One to four selector subchannels are optional with a 2870. 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). One kilobyte is 1,000 bytes 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 (kb) for concurrent selector subchannel operations are:

BASIC MULTIPLEXER CHANNEL	DATA RATES FOR SELECTOR SUBCHANNELS			
	FIRST	SECOND	THIRD	FOURTH
110	–	–	–	–
88	180	–	–	–
66	180	180	–	–
44	180	180	180	–
30	180	180	180	100

Programming 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 having 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.

Channel-to-Channel Adapter Connection to 2870

The 2870 may be connected to another system channel. The channel-to-channel adapter, however, is installed on the other channel, not on the 2870.

System Control Functions

System Reset

The system-reset function suspends all instruction processing and timer updating, resets the channels, and resets on-line control units and I/O devices.

The CPU is placed in the stopped state, and all pending interruptions are eliminated. All error-status indicators are reset to 0.

In general, the system is placed in such a state that processing can be initiated without the occurrence of machine checks, except those caused by subsequent machine malfunction.

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 the PSW-restart 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. Because a machine check occurs when information with incorrect parity is used, the incorrect information should be replaced by loading new information.

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 main 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, and set PSW keys, the hex data keys, the manual entry register 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 main 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, clears all main storage locations attached to the system to 0s with valid parity, turns on the load light, turns off the manual light, and initiates a read operation at the selected input device. The clear-storage function may be inhibited by operating a switch on the system control panel. When reading is completed satisfactorily, a new PSW is obtained, the CPU starts operating, and the load light is turned off.

The system reset suspends all instruction processing and timer updating, resets the channels, and resets on-line nonshared control units and I/O devices. The contents of general and floating-point registers remain unchanged.

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 loca-

tion 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 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 storage. Because the remainder of the IPL program may be placed in any desired section of storage, it is possible 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, with program execution to follow 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 Control Panel

The system control panel, which is on the maintenance console, contains the switches, keys, and indicator lights necessary to operate, control, and display the system. The system consists of the CPU, 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. This section of the manual describes operator control and operator intervention.

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

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 A9 on the system control panel contains the operator control panel. Panel A3 contains the emergency-pull switch. (See Figure 5.) 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 5.)

NAME	IMPLEMENTATION
Emergency pull	Pull switch
Interrupt	Key
Load	Key
Load	Indicator light
Load unit	Three rotary switches
Manual	Indicator light
Power off	Key
Power on	Key, backlighted
System	Indicator light
Test	Indicator light
Wait	Indicator light

Emergency Pull

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

The interrupt key is pressed to request an external interruption. The interruption is taken when it is allowed and when the CPU is not stopped; otherwise,

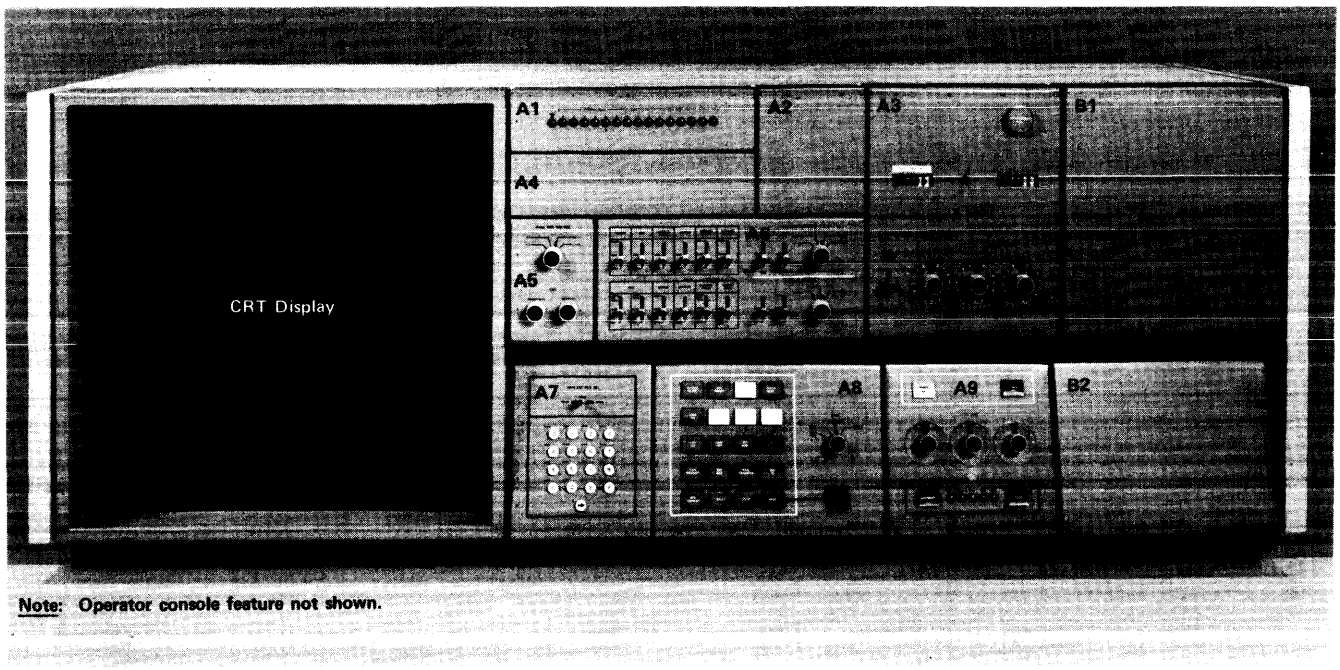


Figure 5. Model 85 System Control Panel

the interruption remains pending. Bit 25 in the interruption-code portion of the external old PSW is made 1 to indicate that the interrupt key is the source of the external interruption. The interrupt key is effective while power is on the system.

Load (Key)

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

Load (Light)

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

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 eight positions (0-7) 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

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

The power-off key is pressed to initiate the power-off sequence of the system. The contents of main 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 key is effective while power is on the system.

Power On

The power-on key is pressed to initiate the power-on sequence of the system. As part of the power-on sequence, a system reset is performed in such a manner that the system performs no instructions or I/O operations until explicitly directed. The contents of main storage, including the protection keys, are preserved.

The power-on key is backlighted white when power is on the entire system. The key is backlighted red during the power-on sequence and when any remote/local power control switch in the power system is in the local position. If there is a loss of power in some

section of the CPU, main storage units, or channels, the light will change from white to red. The power-on key is effective only when the emergency-pull switch is at the in position.

System

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 A3 of the system 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, Wait	Working

*Abnormal condition

Test

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 main storage.

Any abnormal setting of a switch that is on the system control panel or on any separate maintenance panel for the CPU, main 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 or thermal conditions occur. The test light is not an indication of the state of the marginal voltage controls.

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

- Buffer
- CE meter key switch
- ECC
- Forced repeat
- Interval timer
- Machine check
- Overlap
- Rate
- Retry
- Ripple mode
- ROS (check)
- ROS (ripple)
- ROS (stop-on-compare)
- Stop-on-compare

Wait

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, A5, A6, A7, and A8 of the system control panel contain most of the controls required for the operator to intervene in normal programmed operation. (See Figure 5.) These controls are intermixed with CE controls in the same areas.

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

The operator intervention controls include (in alphabetical order):

NAME	IMPLEMENTATION
Address compare	Rotary switch
Advance address	Key
Check reset	Key
CPU reset	Key
CRT intensity	Potentiometers
CRT mode select	Rotary switch
Cursor advance	Key
Data keyboard	Keys
Display	Key
Interval timer	Toggle switch
Logout	Key
Machine check	Toggle switch
Manual entry register select	Toggle switch
PSW restart	Key
PSW15	Indicator light
Rate	Rotary switch
Set IC	Key
Set PSW	Key
Start	Key
Stop	Key
Stop-on-compare	Toggle switch
Store	Key
System reset	Key

Manual Entry for Store and Display

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 system control panel or displayed on the cathode-ray-tube (CRT) display.

The maintenance entry register (MER) is an eight-bit register that receives hexadecimal digits as they are entered from the data keyboard. After the MER 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 register select switch and the cursor on the CRT display must be used. The register select switch specifies the receiving register as the MCAR or the MCDR and causes the cursor to appear under the selected register display on the CRT.

The cursor:

1. Is 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.
4. Returns to byte 0 when the register 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.

Digits entered from the keyboard are placed in the manual entry register (MER). When two digits have been entered, correct parity is generated, and the byte is transferred to the register specified by the register select switch. When a byte is entered into the selected register, the cursor advances to the next byte position of the selected register.

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

Address Compare

The address-compare 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 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.

Advance Address

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

Check Reset

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

CPU Reset

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

CRT Intensity

Two intensity control potentiometers are on panel A5 of the system control panel to provide for adjusting the CRT display for best viewing.

CRT Mode Select

The two-position CRT mode select switch on panel A4 of the system control panel is used to select either the operator display (if the operator console feature is installed) or the maintenance display (for manual store-and-display functions).

Cursor Advance

The cursor advance key (labeled with a right-going arrow) is on panel A7 of the system control panel. When the key is pressed and held in, the cursor on the CRT moves 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

The data keyboard is on panel A7 of the system control panel and 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

Pressing the display key 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 key is only effective if the CPU is in the stopped state.

Main Storage Display: If the storage-select switch is set to the main position, pressing the display key causes the contents of the 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 press-

ings until the buffer page containing the data is reassigned. If a system reset is performed before displaying, main storage is displayed first.

General Register Display: If the storage-select switch is set to GPR, pressing the display key 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 left half of the MCDR.

Floating-Point Register Display: If the storage-select switch is set to FP, pressing the display key 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.

Interval Timer

The interval timer toggle switch, on panel A6 of the system control panel, can be used by the operator to disable the updating of the interval timer. Setting the interval timer switch to the disable position turns on the test light on panel A9.

Logout

Pressing the logout key causes a logout of machine status into main storage, starting at location 128 (decimal). This logout takes place, regardless of the state of the CPU. The logout area normally contains 1,848 bytes (131 doublewords), unless the 709/7040/7044 7090/7094/7094II compatibility feature is installed, in which case the logout area contains 1,728 bytes (116 doublewords).

Machine status is also logged out when a machine check interruption takes place.

Machine Check

The machine-check switch is a three-position toggle switch that can be used to modify the handling of a machine check within the system.

In the stop-on-check position, all machine checks cause a hardstop. There is no logout and no interrupt is taken. If the switch is changed to the process position and the start key is pressed, a retry is attempted if the machine is in retry mode (retry switch in normal position). If the machine is not in retry mode, results are unpredictable.

In the process (normal) position, all machine checks cause a stop, a logout, and (if the retry switch is in the normal position) an instruction retry. If retry is switch-disabled, a machine check interruption and a logout occur.

In the disable position, all machine checks turn on the associated check triggers but no other action is taken. This position of the switch is normally used only by maintenance personnel. The machine-check switch is active in all CPU states.

Manual Entry Register Select

This three-position toggle 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 position is used only by maintenance personnel.

PSW Restart

The PSW-restart key is pressed to initiate the following actions in sequence:

1. System reset.
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 key is active only when the CPU is in the stopped state.

PSW15

The PSW15 indicator, on panel A1 of the system control panel, is on when the current PSW has bit 15 set to 1.

Rate

The rate switch is a four-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 key is pressed. The decoding of instructions is halted by pressing the stop key.

Instruction Step: In the instruction-step position, the rate switch allows one instruction to be completely executed when the start key is pressed. Any interruption that becomes pending before the end of the operation 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 key is held depressed. Instruction execution is halted when the start key is released. Any interruption that becomes pending before the end of the operation is serviced. I/O operations initiated in the multiple-step mode are completed. The CPU enters the stopped state.

Single Cycle: In the single-cycle position, the rate switch allows one machine cycle to be executed when the start key is pressed. Overlap conditions and sequences are identical to those when the rate switch is in the process position. If the rate switch is set to another position and the start key is pressed again, processing starts in the newly selected mode.

This switch position is used only by maintenance personnel.

Set IC

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

Set PSW

Pressing the set PSW key sets the PSW to the corresponding bits in the MCDR. The set PSW key is active only when the CPU is in the stopped state.

Start

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

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

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

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

Stop

Pressing the stop key 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 key is effective when power is on the system.

Stop-on-Compare

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

Store

Pressing the store key 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 change, and pressing the store key. The store key is effective only when the CPU is in the stopped state.

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

Manual Store to General Register: If the storage-select switch is set to GPR, pressing the store key causes the contents of the left 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 FP, pressing the store key 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

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; and activates the check reset. The CPU is placed in the stopped state, and all pending interruptions are eliminated.

The system reset key is active in all CPU states.

CE Controls

All switches and indicators not described as operator controls or operation intervention controls are considered to be CE controls.

The customer usage meter and the CE meter for the central processing complex (CPC) are both on panel A3 of the system control panel. The Model 85 CPC includes the processor, the maintenance console and system control panel, the CPU power supplies, and the power distribution unit. Main storage units, 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 85: the 2085 Processing Unit, the 2860 Selector Channel, the 2870 Multiplexer Channel, the 2365 Processor Storage Model 5, and the 2385 Processor Storage Models 1 and 2. Meters also appear on individual I/O units.

On the 2085 Processing Unit, the customer usage meter and the CE meter are on panel A3 of the system 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 A9, indicates when the system is in operation. The test light on panel A9 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 and the 2870 Multiplexer Channel each have one usage meter mounted on their respective power control panels.

Each 2365 Processor Storage Model 5 has one usage meter. The 2385 Processor Storage has one usage meter for the Model 1 and one usage meter for the Model 2.

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 will run are described on Figure 6.

Unit Categories	Category Description	When Meter Runs
Central Processing Complex (CPC) Base CPC Units: 2085 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 in a stop or wait state.
Assignable CPC Units: 2860 Selector Channel 2870 Multiplexer Channel 2365-5 Storage Unit 2385-1 Storage Unit 2385-2 Storage Unit 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 which are never attached to a system.		The meter records time from the first operation until a runout occurs.
On-Line/Off-Line: Units which 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 6. Metering Table

Timing Considerations

Because of the complexity of the Model 85, it is not possible to provide simple timing formulas which exactly express the processor operation. The timings and formulas provided in this section are considered to be a reasonable approximation to the actual Model 85 timings. The information was arrived at by timing a sequence of each instruction until a repetitive timing pattern became evident; from this pattern, the average time was determined. For those instructions with a variable execution time, some particular execution time (usually one cycle) was chosen for performing this procedure and the resulting time then adjusted by including appropriate factors to account for the variation.

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 where it is preceded and followed by a sequence of fixed-point loads. Times quoted are the average of both approximations.

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 the buffer. For the Model I85, 1.92 microseconds must be added for each block that must be loaded into buffer storage. For Models J85, K85, and L85, 0.88 microsecond must be added for each block that must be loaded into the buffer.

2. The equations assume that the general registers needed for address calculations are available when needed. If one instruction modifies a general register and a subsequent one uses it for an address calculation, at least 0.24 microsecond must be allowed for the intervening instructions.

Variable-field-length 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/360 Model 85 Timings."

Average Timing Formulas

INSTRUCTION	FORMAT	MNEMONIC	TIME (MICROSECONDS)
Add	RR	AR	0.08
Add	RX	A	0.16 + BA2
Add Decimal*	SS	AP	1.10 + 0.09N1 + 0.03N2
Add Halfword	RX	AH	0.16 + BA1
Add Logical	RR	ALR	0.08
Add Logical	RX	AL	0.16 + BA2
Add Normalized (Long)	RR	ADR	0.30
Add Normalized (Extended)	RR	AXR	1.58
Add Normalized (Long)	RX	AD	0.30 + BA3
Add Normalized (Short)	RR	AER	0.38
Add Normalized (Short)	RX	AE	0.38 + BA2
Add Unnormalized (Long)	RR	AWR	0.46
Add Unnormalized (Long)	RX	AW	0.46 + BA3

INSTRUCTION	FORMAT	MNEMONIC	TIME (MICROSECONDS)
Add Unnormalized (Short)	RR	AUR	0.54
Add Unnormalized (Short)	RX	AU	0.54 + BA2
AND	RR	NR	0.08
AND	RX	N	0.16 + BA2
AND	SI	NI	0.48
AND*	SS	NC	A1 = 0.08 (9.98 + 1.01N + 0.01N2) A2 = 0.64 + 0.56N A3 = 0.72
Branch and Link	RR	BALR	0.19 + 0.15F2
Branch and Link	RX	BAL	0.355
Branch on Condition	RR	BCR	0.19 + F1 (0.22 – 0.11F3)
Branch on Condition	RX	BC	0.21 + F1 (0.21 – 0.065F3)
Branch on Count	RR	BCTR	0.11 + 0.11F1
Branch on Count	RX	BCT	0.405 – 0.05F1
Branch on Index High	RS	BXH	0.485 – 0.13F1
Branch on Index Low or Equal	RS	BXLE	0.485 – 0.13F1
Compare	RR	CR	0.08
Compare	RX	C	0.16 + BA2
Compare Decimal*	SS	CP	0.84 + 0.07N1 + 0.03N2
Compare Halfword	RX	CH	0.16 + BA1
Compare Logical	RR	CLR	0.08
Compare Logical	RX	CL	0.16 + BA2
Compare Logical	SI	CLI	0.16
Compare Logical*	SS	CLC	0.56 + 0.12K2
Compare (Long)	RR	CDR	0.24
Compare (Long)	RX	CD	0.24 + BA3
Compare (Short)	RR	CER	0.32
Compare (Short)	RX	CE	0.32 + BA2
Convert to Binary	RX	CVB	[0.08 (5 + P1)] + BA4
Convert to Decimal	RX	CVD	[0.08 (5.5 + 2P2)] + BA5
Divide	RR	DR	1.92
Divide	RX	D	1.92 + BA2
Divide Decimal*	SS	DP	2.04 + 0.8 (N1 – N2) (N2 + 1)

INSTRUCTION	FORMAT	MNEMONIC	TIME (MICROSECONDS)
Divide (Long)	RR	DDR	2.65
Divide (Long)	RX	DD	2.65 + BA3
Divide (Short)	RR	DER	1.64
Divide (Short)	RX	DE	1.64 + BA2
Edit*	SS	ED	0.77 + 0.31N + 0.01N2
Edit and Mark*	SS	EDMK	1.14 + 0.31N + 0.01N2
Exclusive OR	RR	XR	0.08
Exclusive OR	RX	X	0.16 + BA2
Exclusive OR	SI	XI	0.48
Exclusive OR*	SS	XC	A1 = 0.08 (9.98 + 1.01N + 0.01N2) A2 = 0.64 + 0.56N A3 = 0.72 A4 = 1.04 + 0.025K5 (N − 17)
Execute	RX	EX	0.48 + E + BA10
Halt I/O	SI	HIO	0.96 + 0.32B5
Halve (Long)	RR	HDR	0.40
Halve (Short)	RR	HER	0.48
Insert Character	RX	IC	0.16
Insert Storage Key	RR	ISK	0.80
Load	RR	LR	0.08
Load	RX	L	0.16 + BA2
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 Halfword	RX	LH	0.16 + BA1
Load (Long)	RR	LDR	0.08
Load (Long)	RX	LD	0.16 + BA3
Load Microprogram	SI	LMP	0.8 + 0.16NCW
Load Multiple	RS	LM	0.44 + 0.08GR + BA9

INSTRUCTION	FORMAT	MNEMONIC	TIME (MICROSECONDS)
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	1.28
Load Rounded (Long to Short)	RR	LRER	0.32
Load Rounded (Extended to Long)	RR	LRDR	0.40
Load (Short)	RR	LER	0.08
Load (Short)	RX	LE	0.16 + BA2
Move	SI	MVI	0.32
Move*	SS	MVC	V3 = 1.34 + 0.028K5 (N − 17) V5 = 0.8 V6 = 0.56 + 0.18N + 0.04N2
Move Numerics*	SS	MVN	A1 = 0.08 (9.98 + 1.01N + 0.01N2) A2 = 0.64 + 0.56N A3 = 0.72
Move with Offset*	SS	MVO	0.64 + 0.17N1 + 0.01N2
Move Zones*	SS	MVZ	A1 = 0.08 (9.98 + 1.01N + 0.01N2) A2 = 0.64 + 0.56N A3 =0.72
Multiply	RR	MR	1.04
Multiply	RX	M	1.04 + BA2
Multiply Decimal*	SS	MP	0.40 + 0.16N2 + 0.4 (N1 − N2) (N2 + 3.3)
Multiply Halfword	RX	MH	0.80 + BA1
Multiply (Extended)	RR	MXR	(to be supplied)
Multiply (Long)	RR	MDR	2.08
Multiply(Long)	RX	MD	2.08 + BA3
Multiply (Long/Extended)	RR	MXDR	2.24
Multiply (Long/Extended)	RX	MXD	2.24 + BA3
Multiply (Short)	RR	MER	1.36
Multiply (Short)	RX	ME	1.36 + BA2

INSTRUCTION	FORMAT	MNEMONIC	TIME (MICROSECONDS)
OR	RR	OR	0.08
OR	RX	O	0.16 + BA2
OR	SI	OI	0.48
OR*	SS	OC	A1 = 0.08 (9.98 + 1.01N + 0.01N2) A2 = 0.72
Pack*	SS	PACK	0.38 + 0.01N1 + 0.17N2
Read Direct	SI	RDD	1.20 + ED
Set Program Mask	RR	SPM	0.32
Set Storage Key	RR	SSK	1.20
Set System Mask	SI	SSM	0.48
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	0.96 + 0.32B5
Store	RX	ST	0.32 + BA2
Store Character	RX	STC	0.32
Store Halfword	RX	STH	0.32 + BA1
Store (Long)	RX	STD	0.32 + BA3
Store Multiple*	RS	STM	STM1 = 0.40 + 0.1GR STM2 = 2.56 + 0.6GR
Store (Short)	RX	STE	0.32 + BA2
Subtract	RR	SR	0.08
Subtract	RX	S	0.16 + BA2
Subtract Decimal*	SS	SP	1.12 + 0.08N1 + 0.02N2
Subtract Halfword	RX	SH	0.16 + BA1
Subtract Logical	RR	SLR	0.08
Subtract Logical	RX	SL	0.16 + BA2

INSTRUCTION	FORMAT	MNEMONIC	TIME (MICROSECONDS)
Subtract Normalized (Extended)	RR	SXR	1.58
Subtract Normalized (Long)	RR	SDR	0.30
Subtract Normalized (Long)	RX	SD	0.30 + BA3
Subtract Normalized (Short)	RR	SER	0.38
Subtract Normalized (Short)	RX	SE	0.38 + BA2
Subtract Unnormalized (Long)	RR	SWR	0.46
Subtract Unnormalized (Long)	RX	SW	0.46 + BA3
Subtract Unnormalized (Short)	RR	SUR	0.54
Subtract Unnormalized (Short)	RX	SU	0.54 + BA2
Supervisor Call	RR	SVC	2.18
Test and Set	SI	TS	1.04
Test Channel	SI	TCH	0.96 + 0.32B5
Test I/O	SI	TIO	0.96 + 0.32B5
Test under Mask	SI	TM	0.16
Translate*	SS	TR	0.71 + 0.25N
Translate and Test*	SS	TRT	1.29 + 0.31N
Unpack*	SS	UNPK	0.48 + 0.09N1 + 0.01N2
Write Direct	SI	WRD	0.88
Zero and Add*	SS	ZAP	1.08 + 0.07N1 + 0.03N2

VFL Timing Assumptions

Variable-field-length instructions, marked by an asterisk in the average timing table, assume the following conditions.

Add Decimal (AP)

1. Twenty-five percent probability that recomplementation is required.
2. Average of three cycles required to load more SRC or DST from operand buffers.
3. Ten percent probability that the result is 0.
4. Twenty-five percent probability that L1 is greater than L2.

AND (NC)

1. A1 is used for normal case and if N is greater than 1.
2. A2 is used for word-overlap case.
3. A3 is used for one-byte operands.
4. A4 (exclusive OR only) is used if N is greater than 1, the low-order three bits of the source address are equal to the low-order three bits of the destination address, and both operands are in the same doubleword.
5. Average of 4.5 bytes processed before entering high-speed case.
6. Fifty percent probability that N is greater than 8, and the last doubleword to be stored is in the last half of a quadword.

Compare Logical (CLC)

Assume at least the SRC or DST crosses a doubleword boundary on an average of three times.

Compare Decimal (CP)

1. Average of three cycles required to load more SRC or DST from the operand buffers.
2. Twenty-five percent probability that L1 is greater than L2.
3. Ten percent probability that the result is 0.
4. Twenty-five percent probability that recomplementation is required.

Divide Decimal (DP)

1. The high number of iterations required to perform even a simple divide makes doubleword boundary crossings relatively insignificant; however, one operand crossing is assumed.
2. The average N2 is 4; therefore, the average for term (0.5N2) is 2.

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. Average of 3.6 cycles per pattern character.

Edit and Mark (EDMK)

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

Exclusive OR (XC)

The same assumptions as for AND (NC).

Move (MVC)

1. V3 is used if the first and second operand fields do not start at corresponding byte addresses.
2. V5 is used if the first and second operand fields start at corresponding byte addresses.
3. V6 is used if the source address is less than the destination address, or if the source address plus M is greater than or equal to the destination address, where M is the lesser of N − 1 or 31.
4. V3 is modified to include all cases where N is greater than 8. Normal and high-speed cases are averaged with the normal case weighted by a factor of 3.
5. Equal probability is assumed for all ways to process N equal to or less than eight bytes, averaging to four execution unit cycles. V5 is altered to handle all cases where N is equal to or less than 8.
6. Parameter (24 − N)/8 averages to 2.375 for all overlap cases.

Move Numerics (MVN)

The same assumptions as for AND (NC).

Move with Offset (MVO)

1. Overlap is not considered.
2. Assume one cycle for padding.
3. An average of 1 for the number of doubleword boundary crossovers for that part of the second operand processed, for the number of doubleword boundary crossovers for that part of the first operand processed before the second operand is depleted, and for the number of doubleword boundary crossovers for the first operand.

Multiply Decimal (MP)

1. Formula Portion:

$$5\,(N2 + 1)\,(N1 - N2) - 2.5\,(N1 - N2) + 14\,(N1 - N2 - 1)$$

Reduction as follows:

$$\begin{aligned} &5\,(N2 + 1)\,(N1 - N2) - 2.5\,(N1 - N2) + 14\,(N1 - N2) - 14 \\ &(N1 - N2)\,[15\,(N2 + 1) - 2.5 + 14] - 14 \\ &(N1 - N2)\,(5\,N2 + 5 - 2.5 + 14) - 14 \\ &5\,(N1 - N2)\,(N2 + 3.3) - 14 \end{aligned}$$

2. Assume as an average that:

$$\begin{aligned} NDWBL1L2 &= 1 \\ NDWBL2 &= 1 \end{aligned}$$

Move Zones (MVZ)

The same assumption as for AND (NC).

OR (OC)

The same assumptions as for AND (NC).

Pack (PACK)

1. Overlap is not considered.
2. Assume one crossing for DST = 6 (average).
3. Assume one crossing for SRC = 6 (average).
4. Assume no simultaneous crossings on average.
5. Add one cycle for any padding required times the probability of padding occurring at all.

Store Multiple (STM)

1. STM1 is used if boundary alignment is not necessary.
2. STM2 is used if boundary alignment is necessary.

Subtract Decimal (SP)

The same assumptions as for add decimal (AP).

Translate (TR)

Assume no overlap.

Translate and Test (TRT)

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

Unpack (UNPK)

1. Overlap is not considered.
2. SRC crossing one = 8.5.
3. DST crossing one = 8.5.

Zero and Add (ZAP)

1. Probability that L1 is less than L2 = 25 percent.
2. Assume one crossing for DST.
3. Assume one crossing for SRC.

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 main storage. When a user takes advantage of this feature and does not observe the usual boundary constraints, instruction times are modified as shown under the BA legend; or, in more detail, as follows:

INSTRUCTION TYPE		AVERAGE TIME (MICROSECONDS)
Halfword operand fetch		2.00 plus normal number of execution cycles if doubleword boundary is not crossed. 2.08 plus normal number of execution cycles if doubleword boundary is crossed.
Halfword store		2.16 if doubleword boundary is not crossed. 2.40 if doubleword boundary is crossed.
Fixed-point fullword and floating-point short		
Operand fetch		2.16 plus normal number of execution cycles if doubleword boundary is not crossed. 2.24 plus normal number of execution cycles if doubleword boundary is crossed.
Store		2.48 if doubleword boundary is not crossed. 2.56 if doubleword boundary is crossed.
Floating-point long		Add 0.32 to floating-point short times.
Store multiple	GR = 1	2.64 if doubleword boundary is not crossed.
	GR = 1	2.72 if doubleword boundary is crossed.
	GR = 2, 3	2.08 + 0.64GR
	GR = 4, 5	2.16 + 0.64GR
	GR = 6, 7	2.24 + 0.64GR
	GR = 8, 9	2.32 + 0.64GR
	GR = 10, 11	2.40 + 0.64GR
	GR = 12, 13	2.48 + 0.64GR
	GR = 14, 15	2.56 + 0.64GR
	GR = 16	2.64 + 0.64GR
Load multiple	GR = 1	2.72 if doubleword boundary is not crossed.
	GR = 1	2.80 if doubleword boundary is crossed.
	GR = 2, 3	2.24 + 0.56GR
	GR = 4, 5	2.32 + 0.56GR
	GR = 6, 7	2.40 + 0.56GR
	GR = 8, 9	2.48 + 0.56GR
	GR = 10, 11	2.56 + 0.56GR
	GR = 12, 13	2.64 + 0.56GR
	GR = 14, 15	2.72 + 0.56GR
	GR = 16	2.80 + 0.56GR
Execute (subject of)		For the subject instruction of an execute, add 2.56 microseconds to the times obtained from this table for the subject instruction.

Programming Note: Significant performance degradation is possible when storage operands are not positioned at addresses that are integral multiples of the operand length. 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/360 Model 85 Timings

A1 to A4 = Detailed under "AND (NC)" in VFL timing assumptions.

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 = 2.08 if boundary alignment is necessary.
= 0 otherwise.

BA2 = 2.24 if boundary alignment is necessary.
= 0 otherwise.

BA3 = 2.56 if boundary alignment is necessary.
= 0 otherwise.

BA4 = 2.72 if boundary alignment is necessary.
= 0 otherwise.

BA5 = 0.64 if boundary alignment is necessary.
= 0 otherwise.

BA6 = 2.40 if boundary alignment is necessary.
= 0 otherwise.

BA7 = 2.48 if boundary alignment is necessary.
= 0 otherwise.

BA8 = 2.32 if boundary alignment is necessary.
= 0 otherwise.

BA9 = 1.80 + 0.56GR if boundary alignment is necessary.
= 0 otherwise.

BA10 = 2.40 if subject instruction of execute contains an off-bounds operand.
= 0 otherwise.

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

DST = Destination.

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

ED = External delay.

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

F2 = 0 if the R2 field is 0, or the mask equals 0.
= 1 otherwise.

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

GR = Number of general registers stored 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.

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).

NCW = Value of word count field + 1.

NDWBL2 = Number of doubleword boundary crossovers for the second operand.

NDWBL1L2 = Number of doubleword boundary crossovers for that part of the first operand which consists of N2 bytes of high-order 0s.

P1 = Number of significant decimal digits.

P2 = Number of significant hexadecimal digits.

SRC = Source.

STM1, STM2 = Detailed under "Store Multiple (STM)" in VFL timing assumptions.

V3 to V6 = Detailed under "Move Character (MVC)" in VFL timing assumptions.

Index

Adapter, Channel-to-Channel ... 16
Address Calculation and Operand Fetches ... 11
Address-Compare Switch ... 21
Advance Address Key ... 21
Average Timing Formulas ... 25

Branch Instructions, Execution of ... 11
Branch Success Estimates ... 11
Buffer
  Storage ... 5
  Storage Control ... 12
  Storage Cycle Time ... 5
Byte-Oriented Operand Feature Times ... 32

CE (Customer Engineer)
  Controls ... 23
  Key Switch ... 23, 24
  Meter ... 24
Channel
  Attachment ... 5
  Metering ... 24
  Storage Operations ... 13
  2860 Selector ... 16
  2870 Multiplexer ... 16
Channel-to-Channel Adapter ... 16
Check Reset Key ... 21
Circuitry, Logic ... 5
Clear-Storage Function on IPL ... 17
Coding Considerations ... 10
Combinations, CPU/Main Storage ... 9, 10
Components, System ... 5
Configurator ... 7
Control Panel, System ... 18
Control Storage ... 14
Controls, System Control Panel ... 18
CPC (Central Processing Complex) ... 23
CPU (2085 Processing Unit) ... 10
CPU/Main Storage Combinations ... 9, 10
CRT (Cathode-Ray Tube)
  Display ... 8
  Intensity Potentiometers ... 21
  Mode Select Switch ... 21
Cursor ... 21
Cursor Advance Key ... 21
Cycle Time
  Buffer Storage ... 5, 12
  Main Storage ... 5, 9
  Major-Machine ... 5

Data Keyboard, System Control Panel ... 19, 21
Display
  CRT ... 20
  Function ... 21
  Key ... 21
  Status and Control ... 10
Document Viewer, Microfiche ... 10

Emergency-Pull Switch ... 18
Emulator ... 9
Entry, Manual ... 20
EPO (Emergency Power Off) ... 18
Error Checking and Correction ... 13
Execute Instruction Handling ... 11
Execution of Branch Instructions ... 11
Execution Unit ... 12
Extended-Precision Floating-Point ... 5

Features
  Optional ... 5
  Standard ... 5
Fetches, Operand ... 11
Floating-Point Registers ... 12

General Registers ... 12

High-Speed Multiply ... 8

IC (Instruction Counter) ... 14
Imprecise Interruptions ... 12
Initial Program Loading (IPL) ... 17
Instruction
  Buffering ... 10
  Counter (IC) ... 14
  Decoding ... 10
  Fetching ... 10
  Issuing ... 10
  Retry ... 14
  Sequence Protection ... 11
  Times ... 26
  Unit ... 10
Interleaving ... 10
Interrupt Key ... 18
Interruption
  Machine Check ... 15
  Priorities ... 15
Interruptions, Imprecise ... 12
Interval Timer (See Standard Features) ... 5
Interval Timer Switch ... 22
I/O Channel Control ... 13
I/O Devices Attachable to System ... 5
IPL (Initial Program Loading) ... 17

Key Switch and Meters ... 23, 24

Legend for Model 85 Instruction Timings ... 32
Load Key ... 19
Load Light ... 19
Load-Unit Switches ... 19
Local Storage ... 12
Logout Area ... 22
Logout Key ... 22
Logic Circuitry, System ... 5
Logical Elements of Model 85 ... 5

Machine Check Interruption ... 15
Machine Check Switch ... 22
Machine-Cycle Time, Major ... 5
Main Storage
  Configuration ... 9, 10
  Control ... 12
  Cycle Time ... 5
  Interleaving ... 10
  Metering ... 24
Maintenance
  Console ... 8, 14
  Control Address Register (MCAR) ... 20
  Control Data Register (MCDR) ... 20
  Control Retry Register (MCRR) ... 22
  Controls ... 14
  Entry Register (MER) ... 20
  Ripple Address Register (MRAR) ... 22
Major-Machine-Cycle Time ... 5
Manual Entry ... 22
Manual Entry Register Select Switch ... 22

Manual Light ... 19
MCAR (Maintenance Control Address Register) ... 20
MCDR (Maintenance Control Data Register) ... 20
MCI (Machine Check Interruption) ... 14
MCRR (Maintenance Control Retry Register) ... 22
MER (Maintenance Entry Register) ... 20
Metering Table ... 24
Metering, Usage ... 24
Meters and Key Switch ... 24
Microfiche Document Viewer ... 10
MFT (Multiprogramming with a Fixed Number of Tasks) 15
Mode Select Switch, CRT ... 21
MRAR (Maintenance Ripple Address Register) ... 22
Multiplexer Channel, 2870 ... 16
Multiplying, High-Speed ... 8
MVT (Multiprogramming with a Variable Number of Tasks) ... 15

Operator Console ... 8, 18
Operator Control Panel (OCP) ... 18
Operator Controls ... 18
Operator Intervention Controls ... 20
Optional Features ... 5

Physical Storage Width ... 5
Power Distribution Unit ... 10
Power-Off Key ... 19
Power-On Key ... 19
Priorities, Interruption ... 15
Processing Unit, 2085 ... 10
Program Interruptions ... 15
Programming Support ... 5, 6, 15
PSW Restart Key ... 22
PSW15 Light ... 22

Rate Switch ... 22
Read-Only Storage ... 14
Recovery Management Program (RMP) ... 15
Retry, Instruction ... 14
Remote Operator Control Panel (ROCP) ... 9
RMP (Recovery Management Program) ... 15
ROS (Read-Only Storage) ... 14
Rounding, Floating-Point ... 8

SCU (Storage Control Unit) ... 12
Selector Channel, 2860 ... 16
Set IC (Instruction Counter) Key ... 23
Set PSW Key ... 23
Special Features (Optional Features) ... 8
SS-Format Instruction Handling ... 11
Standard Features ... 5
Start Key ... 23
Stop-on-Compare Switch ... 23
Stop Key ... 23
Storage Control Unit ... 12
Storage Cycle Time ... 5
Storage Width, Physical ... 5
Store Key ... 23
Store-and-Display Function ... 17, 20
System Components ... 5
System Control Functions ... 17
System Control Panel ... 18
System Light ... 20
System Reset Function ... 23
System Reset Key ... 23

Test Light ... 20
Timer, Interval ... 7
Timing Assumptions, Instruction ... 25
Timing Considerations, Instruction ... 25
Unit
  Execution ... 12
  Instruction ... 10
  Power Distribution ... 10
  Storage Control ... 12
Usage Metering ... 24

VFL (Variable Field Length) ... 30
VFL Timing Assumptions ... 30

Wait Light ... 20
WCS (Writable Control Storage) ... 14
Width of Storage, Physical ... 5

READER'S COMMENT FORM

IBM System/360 Model 85
Functional Characteristics

Form A22-6916-0

- How did you use this publication?

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

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

	.....	.....	.....	.....	.....
As a reference source:	Very Good	Good	Fair	Poor	Very Poor
As a text:	Very Good	Good	Fair	Poor	Very Poor

- What is your occupation? .....

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

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

A22-6916-0

YOUR COMMENTS PLEASE . . .

This SRL bulletin is one of a series which serves as reference sources for systems analysts, programmers and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

fold fold

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

BUSINESS REPLY MAIL

NO POSTAGE 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

Printed in U.S.A. A22-6916-0

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]

A22-6916-0

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]