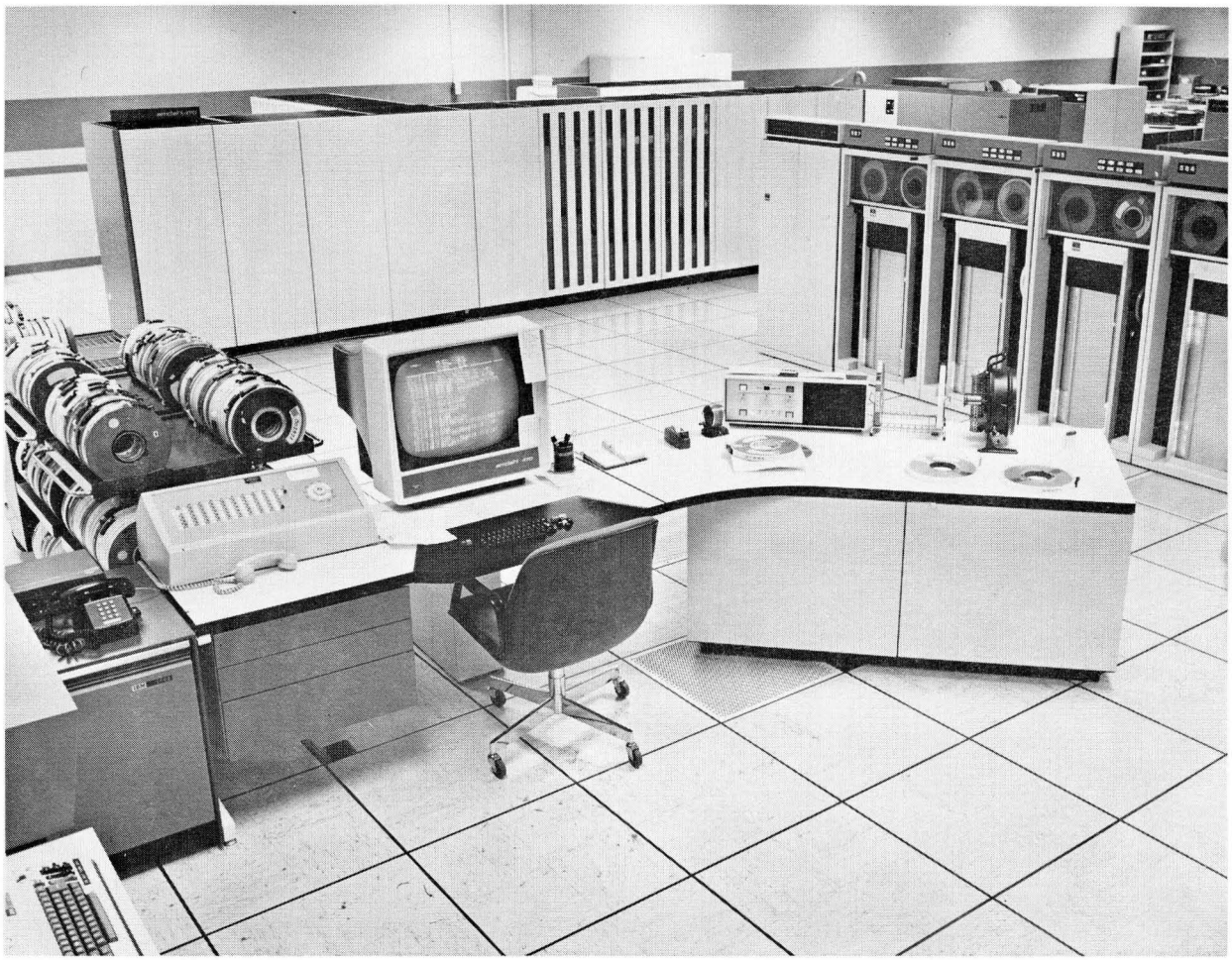
amdahl

470 V/6
MACHINE REFERENCE MANUAL

PREFACE



This manual describes the organization and functional characteristics of the Amdahl 470V/6, a very high-speed, large scale general purpose computing system. It provides machine reference data of fundamental interest and value to management, systems analysts, programmers and operations personnel.

Details on machine organization, performance and configuration are provided. Supplementary data on characteristics of machine check conditions, instruction timing formulas, specific channel characteristics and console operation is provided as appendices.

SECOND EDITION:

Amdahl Corporation Machine Reference Manual, Form MRM 1000–1.

This manual has been prepared by Amdahl Corporation Technical Publications.

The Amdahl 470V/6 computing system provides extremely powerful, high-speed, general-purpose computing capabilities for sophisticated business and scientific applications. The high performance of the Amdahl 470V/6 results primarily from a newly-designed high-speed circuitry with internal speeds measured in picoseconds, packing densities with up to 100 circuits on a central processor chip, a sophisticated architecture that incorporates the pipeline concept for a high concurrency of operations, efficient algorithms for high-speed internal functions, and a high-speed buffer coupled with monolithic main store for fast data access times. A highly-flexible input/output channel scheme is also provided for the Amdahl 470V/6. Sixteen inboard I/O channels are standard and may be configured in any combination of byte or block multiplexer or selector channels. These inboard channels operate independently from the central processor.

The central processor and most of the channel unit are implemented with high-speed large scale integration (LSI) circuitry, which enables a six-to-eight times reduction in overall size and the number of external connections. The reduction in size and complexity becomes immediately obvious in that the central processor and the LSI portion of the channels occupy only 51 cards. Each card, or multichip carrier (MCC), is a ten-layer printed circuit board with 42 chip positions in a 7½" square area.

Reliability is greatly enhanced by the large reduction in components and connections. Further improvement is provided with hardware instruction retry of central processor instructions and channel command retry of I/O instructions. A significant fault recovery capability is thus provided without software assistance. An independent console processor is an integral part of the system console and provides formatted displays of any of approximately 16,000 internal latches, extended logouts to record error conditions, and the capability of configuring failing components out of the system. The high-speed buffer may thus be configured around most failures, and the system can continue running.

Main storage has error checking and correction (ECC) that corrects all single bit errors and detects all double and most multiple bit errors. If an uncorrectable error develops in a section of main storage (each megabyte is an independent section), that section may be configured out of the system.

TABLE OF CONTENTS

INTRODUCTION ii

SYSTEM OVERVIEW 3
- Central Processor 5
- Main Storage 5
- Channels 5
- System Console 5
- Power Distribution Unit 5
- Optional Features 5

CPU–MAIN STORAGE SYSTEM 6
- Instruction Unit 7
- Execution Unit 8
- Storage Control Unit 10
  - High-Speed Buffer 10
  - Buffer Operation 10
  - Virtual Address Operation 11
  - TLB Operation 12
  - STO Stack Operation 13

CHANNEL FACILITIES 14
- Central LSI Channel 15
- Modes of Channel Operation 17

SYSTEM CONSOLE 19
- Usage Metering 20
- Operator's Console Programming 21

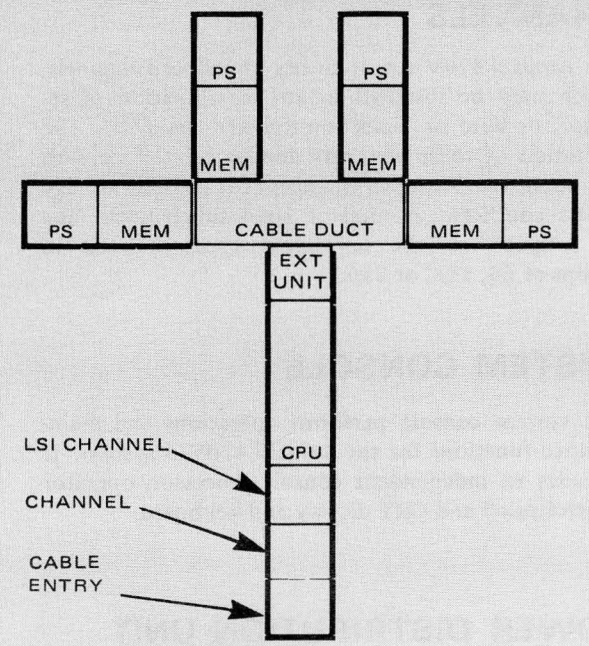
APPENDIX A: MODEL-DEPENDENT FUNCTIONS 23
- Instruction Set Differences 24
- CPU Logout Areas 24
  - Fixed Logout Area 24
  - Extended CPU Logout 26
- Machine Checks 26
  - Machine Check Conditions 26
  - Machine Check Interrupts 27
  - Hardware Diagnostic Mode 28

APPENDIX B: INSTRUCTION TIMING ESTIMATES 29
- Early Condition Code Settings 30
- Retryable Instructions 30
- Timing Formulas 31
- Legend for Timing Formulas 35
- Special Timing Formulas 37

APPENDIX C: MODEL-DEPENDENT CHANNEL FUNCTIONS 41
- Subchannel Assignment 42
- Extended Channel Logout 44
- Limited Channel Logout 44

APPENDIX D: CONSOLE PROCESSOR SYSTEM 49
Modes of Operation 50
Amdahl 470V/6 Power Up and Power Down 50
Initialization 51
Initial Program Load 51
Status Display 52
Errors 54
Control Mode Operation 55
Control Mode and Device Support Mode Commands 56
Strings 64

Major components of the Amdahl 470V/6 System are the central processing unit, main storage, channels, system console, and power distribution unit.



SYSTEM/370 UNIVERSAL INSTRUCTION SET

SYSTEM/370 INSTRUCTION ENHANCE-MENTS

BYTE ORIENTED OPERANDS

16,384 BYTE HIGH SPEED BUFFER

CPU TIMER

TOD CLOCK

INTERVAL TIMER

CLOCK COMPARATOR

EXTENDED CONTROL

DYNAMIC ADDRESS TRANSLATION

PROGRAM EVENT RECORDING

DIRECT CONTROL

16 I/O CHANNELS

BYTE/ BLOCK/ SELECTOR

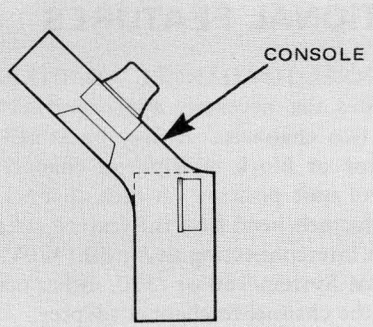
1024 MULTIPLEXER SUBCHANNELS

INDIRECT DATA ADDRESSING

OPTIONAL FEATURES

CHANNEL TO CHANNEL ADAPTER

2–BYTE INTERFACE



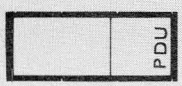
OPERATOR'S CONSOLE

3066 EMULATION

3215 EMULATION

SYSTEM CONSOLE WITH CRT DISPLAY

INDEPENDENT CONSOLE PROCESSOR



EMERGENCY POWER-OFF CONTROL

FIGURE 1 SYSTEM CONFIGURATION

CENTRAL PROCESSOR

The Amdahl 470V/6 central processor includes the following standard features:

STANDARD ARCHITECTURE. The Amdahl 470V/6 follows the architecture of System/370 as specified in the *IBM System/370 Principles of Operation, GA22–7000* (henceforth referred to as *Principles of Operation).* The standard full System/370 Universal Instruction Set with extended precision floating point operations, direct control and System/370 instruction enhancements are implemented on the Amdahl 470V/6.

DIRECT CONTROL. This facility provides a direct control path into the central processor from external devices and operates independently of the channels.

16K-BYTE HIGH-SPEED BUFFER. The standard buffer size provides 16,384 bytes of high-speed bipolar memory to maximize system throughput. Memory transfer width to the buffer is 32 bytes and CPU transfer width from the buffer is 4 bytes.

TRANSLATION LOOKASIDE BUFFER. To enhance virtual performance the Amdahl 470V/6 provides a 256-entry translation lookaside buffer (TLB). This TLB provides high-speed storage of most recently used virtual address translation information. Operation of the TLB is further enhanced with a 32-entry segment table origin stack to identify specific CPU states with TLB entries.

TIMING FACILITIES. Standard timing facilities of System/370 architecture are provided. These include an interval timer, time-of-day (TOD) clock (a 64-bit field with 52-bit resolution), a clock comparator (provides a 52-bit comparison field for the TOD clock) and CPU timer.

MAIN STORAGE

Main storage is available in one megabyte increments, up to eight megabytes. Interleaving is four-way for each two megabyte storage unit and two-way if one megabyte is installed in a storage unit. Thus, a three megabyte system is four-way interleaved for the first two megabytes and two-way interleaved for the last odd megabyte.

CHANNELS

The Amdahl 470V/6 system has 16 inboard channels, which may be installed in any combination of selector, or byte or block multiplexer channels. The operation of these channels does not interfere with the operation of the CPU, except for possible storage access conflicts. A total of 1024 subchannels may be assigned to the multiplexer-type channels in groups of 64, 128, or 256.

SYSTEM CONSOLE

The system console performs operations and maintenance functions for the Amdahl 470V/6 system. It includes an independent console processor, operator control panel and CRT display and keyboard.

POWER DISTRIBUTION UNIT

The power distribution unit (PDU) distributes 400 Hz and 60 Hz power to the Amdahl 470 V/6 system. Emergency power off and thermal monitoring facilities are also provided.

OPTIONAL FEATURES

CHANNEL-TO-CHANNEL ADAPTER. This option provides the necessary synchronization to interconnect two channels. It may be attached to either a selector or block multiplexer channel and uses one control unit position on each channel. Only one of the channels need have the feature installed, however. When interconnecting an Amdahl 470V/6 system with an IBM System/360 or /370, either may be equipped with the channel-to-channel adapter.

2–BYTE INTERFACE. A standard channel interface provides a one-byte wide data path between controllers and a channel. The addition of the two-byte interface effectively doubles bandwidth for control units that support this feature. The two-byte interface option is available on all selector and multiplexer channels.

The Amdahl 470V/6 consists of four logical, independent units physically implemented in 51 MCCs. These four units are the instruction unit (I-Unit), which implements the pipeline; the execution unit (E-Unit), which performs arithmetic and logical instructions; the storage control unit (S-Unit), which controls the high-speed buffer and main storage requests; and the channel unit (C-Unit) which executes channel commands. Because of the distinct functional difference of the C-Unit, it is discussed separately in "Channel Facilities".

INSTRUCTION UNIT

The instruction unit pipeline performs instruction fetching and decoding and coordinates the execution of an operation with other units. Since the I-Unit is principally devoted to controlling instruction execution, it must interface with most of the rest of the machine and resolve priorities. It handles interrupts; status switching; CPU-channel control interface requirements; and scratch, general purpose and floating point register requests from the E-Unit.

A typical instruction execution sequence is illustrated in Figure 2. Each individual instruction's execution is divided into six phases plus an I-fetch phase.

The I-fetch phase begins with the I-Unit requesting the S-Unit priority in the I-cycle. A high-speed buffer access is initiated in the B1 cycle, and the instruction word becomes available in the B2 cycle. The I-Unit then begins instruction interpretation.

Phase A is the instruction decode and general purpose register (GPR) read cycle. Since most instructions have an operand address consisting of the contents of at least one register plus some modification, the register reads are done at this time. Phase B then generates the operand address using the effective address generator and initiates a buffer request for this data. This request may be for either four or eight bytes, depending on the instruction. In a branch instruction

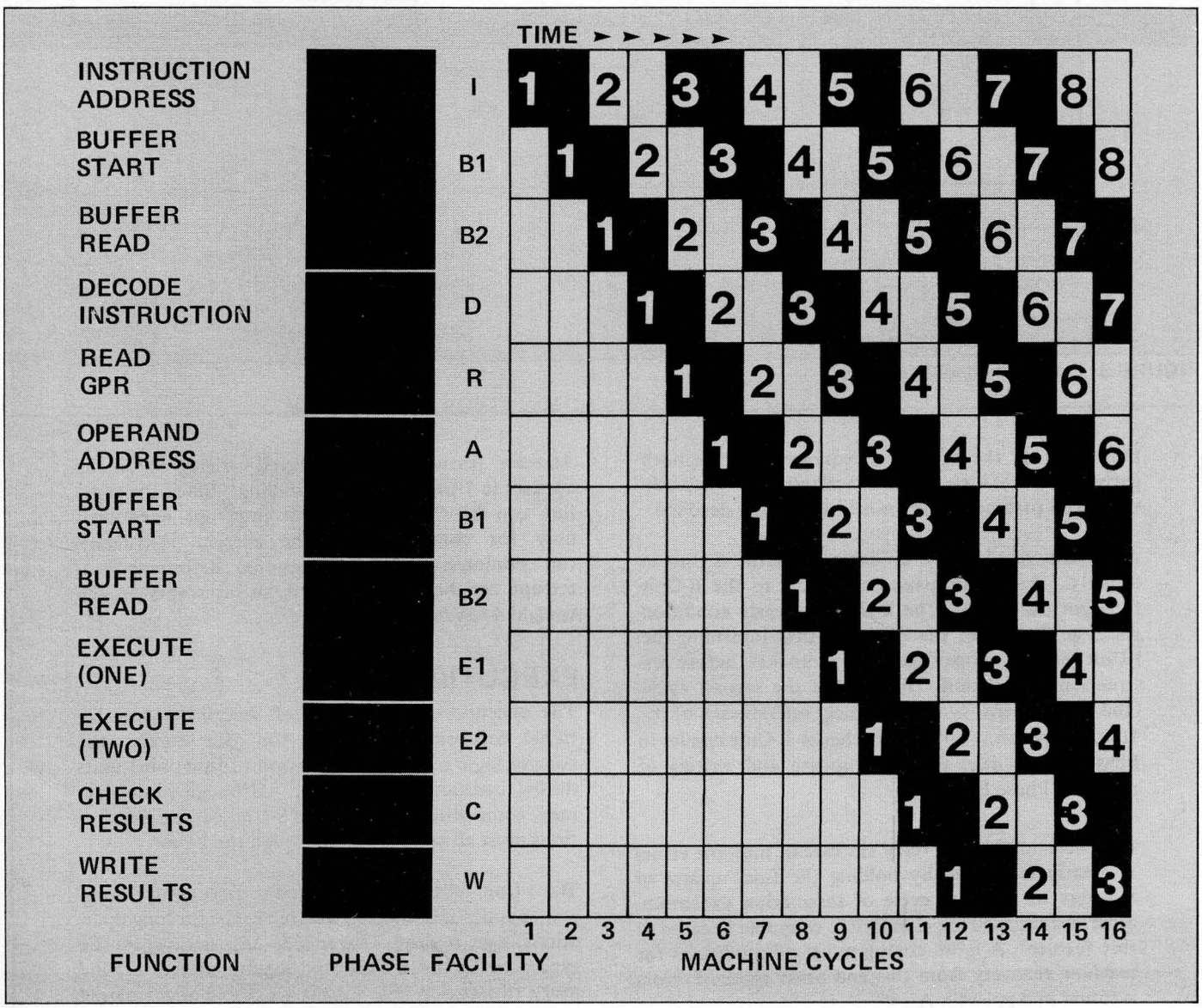


FIGURE 2 I-UNIT INSTRUCTION SEQUENCE

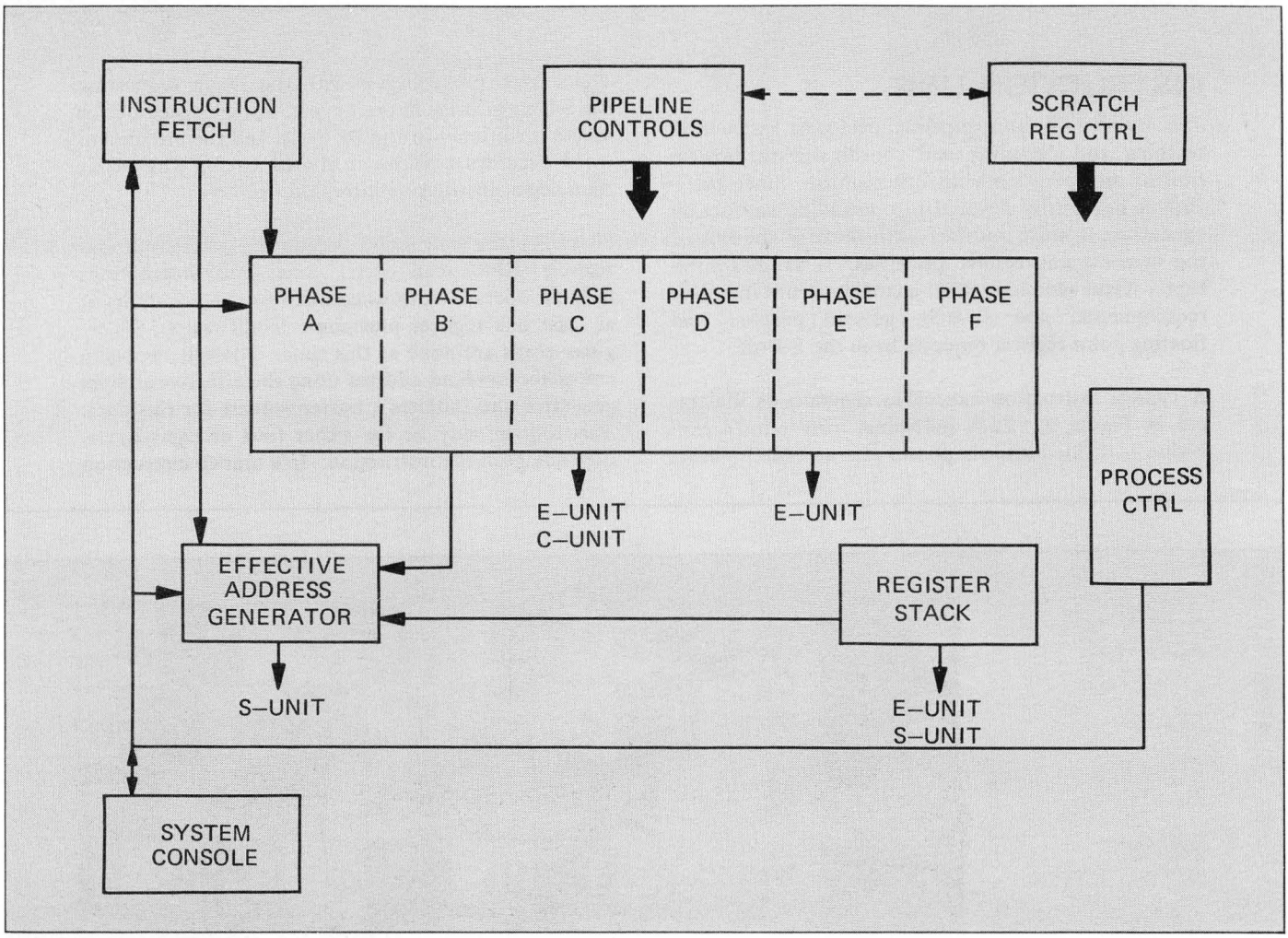


FIGURE 3 I-UNIT ORGANIZATION

the target of the branch is requested so that both paths of the branch will be available. (The branch-not-taken path continues to be fetched and decoded).

After data has become available from the S-Unit in Phase C, the I-Unit passes this data to the E-Unit to begin execution. The E-Unit sets early condition codes at the end of the E1 cycle, thus informing the I-Unit of the proper path for branches before instruction completion. Phase D is the second cycle (and subsequent cycles for long operations) of E-Unit execution. The I-Unit checks E-Unit results in Phase E and does the final update and writing of results in Phase F.

Hardware instruction retry (in case of machine error) is greatly facilitated by holding the final update of registers to the last cycle of instruction execution. Most instructions are completely retryable because of this feature. A more complete list of conditions for machine recovery from this and other types of errors is found in Appendix A.

Another feature of the delayed writing of results appears in I-Unit interrupt handling. Since the pipeline can be "backed up" an interrupt need wait only for completion of the current instruction (on noninterruptable instructions) before the interrupt can be taken; hence, all interrupts on the Amdahl 470V/6 are precise.

EXECUTION UNIT

The execution unit consists of several major functional components, registers for data storage, and control logic for the coordination of functional units during instruction execution. Although some data may come directly from the S-Unit, actual instructions must all be presented through the I-Unit.

The I-Unit tells the E-Unit when to start an operation and supplies intermediate scratch space for long operations, e.g., Multiply Extended. Instructions can be presented to the E-Unit at a maximum rate of one every two cycles; the E-Unit is capable of generating

results on single cycle boundaries thereafter.

Data is presented from the I-Unit to the E-Unit through the logical unit and checker (LUCK). The LUCK scans operands for any information that can be obtained before execution actually begins by performing the following functions:

- LOGICAL OPERATIONS
  AND, OR, Exclusive OR.
- OPERAND COMPARISON
  Magnitude of two operands is compared where possible.
- SETTING EARLY CONDITION CODES
  Condition code is returned after the first cycle on many operations.
- PARITY CHECK
  Input parity is checked and predicted.
- DECIMAL DIGIT CHECK
  Input data is checked for valid digits and sign.
- BIT COUNTER
  Several methods of counting leading zeros are performed for use in shifting and normalization.

The LUCK is placed first in the E-Unit data path to provide a "head-start" for the rest of the E-Unit. Data output from the LUCK is to several intermediate registers. These registers are each one word wide and are divided into higher and lower order words and first and second operands (hence, "1H" and "2L" in Figure 4). Data from each of these registers may then be sent to a functional unit.

The principal E-Unit functional units are multiplier, adder, shifter and byte mover. The multiplier is a carry-save adder used to multiply two operands to produce a 40-bit result; the adder performs a standard binary add or a decimal add, depending on which of several modes is requested; the shifter shifts input operands; the byte mover is used in manipulating single-byte fields for arithmetic or logical operations. Each functional unit performs its function in a single cycle. Error checking is done in all functional units. Output data is checked for good parity, and internal functional checks are made using residue arithmetic.

Output registers from the functional units store intermediate results. The S- and C-registers contain

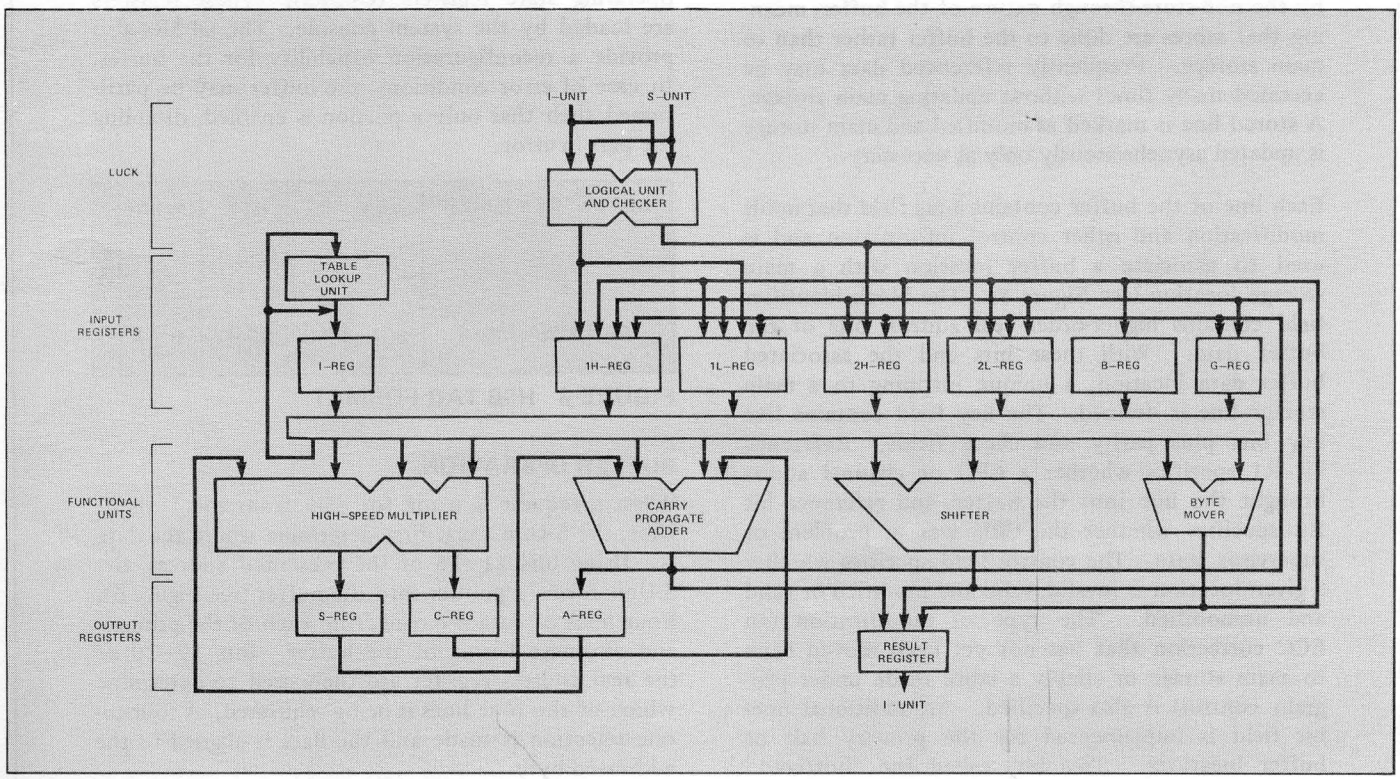


FIGURE 4 EXECUTION-UNIT COMPONENTS

the sum and carry outputs of the multiplier; the A-register contains output from the adder. The I-register is used in generation of 10-bit inverses, which are created by the table-lookup unit, or division operations. The B- and G-registers are used for intermediate storage of fields. When the E-Unit functions are complete, results are staged in the result register for transmission to the I-Unit.

STORAGE CONTROL UNIT

All storage requests from the CPU and channels are processed by the storage control unit. Priorities are resolved and facilities are provided based on the S-Unit's internal priority structure and the priority of the request from the I- or C-Unit. The S-Unit also determines whether requested data comes directly from the high-speed buffer or must be read in from main storage. All virtual-to-real address translations are performed by the S-Unit.

HIGH-SPEED BUFFER

The 470V/6 high-speed buffer (HSB) is organized as a 16,384-byte set-associative memory consisting of primary and alternate halves. Each half contains 256 32-byte lines that can be accessed on a double or single word basis. Because of the local nature of most programs, execution time should be related to buffer speed rather than main storage speed. This significant improvement in speed is further enhanced by the non-store-through nature of the buffer; meaning that stores are done to the buffer rather than to main storage. Frequently referenced data may be accessed many times without updating main storage. A stored line is marked as modified and main storage is updated asynchronously only as necessary.

Each line of the buffer contains a tag field that holds modification and other control information and is used to associate a buffer location with a main storage location (see Figure 5). The block identifier field contains higher-order real address bits of the buffer data. With these bits and the associated buffer data location, a unique mapping to a main storage line is defined. The key field contains five key bits plus parity and check fields. Reference bit R1 specifies whether a CPU or channel access brought the line into the buffer, and reference bit R2 specifies whether the CPU was in problem or supervisor state. The control field specifies whether a given location is invalid, valid and modified or valid and unmodified. The type of modification (an ECC correction that has not yet been moved back to main storage or simply a store made under program control) is also specified. An additional one-bit field is implemented for the primary half of buffer locations. This bit, called the "hot/cold" bit, indicates that the primary location has been referenced more recently than the associated alternate location.

The above factors are variables used by the S-Unit to determine where to put new data in the buffer. If data is not present in the buffer when requested, it may be moved to either the primary or alternate half of the buffer from main storage. The S-Unit can check to see if either the associated primary or alternate location is invalid (and thus may be loaded immediately) or which was used more recently. A facility is also provided to distinguish between CPU and channel data and between supervisor and problem state. The ability to change buffer algorithms further enhances buffer tuning and provides for the most efficient operation under a wide variety of circumstances.

The S-Unit may also use a prefetch function to fetch data from main storage in advance of need. This prefetch may occur after either the first or second access to any quarter-line segment of the buffer. Prefetch may be selectively enabled for the channel, instruction fetch or operand fetch port. Prefetch and buffer change algorithms are preset by Amdahl to optimal values for most applications.

All the above changes to S-Unit algorithms are accomplished by changing the contents of the S-Unit operating state registers (OPSRs). These registers are loaded by the system console. The OPSRs also provide a reconfiguration capability for the buffer. In case of error conditions, the buffer may be partitioned such that only a portion is enabled, disabling the part in error.

BLOCK IDENTIFIER	KEY	REF	CNTRL	H/C
(Real Address Bits)	0 . . . 4 P C	R1 R2	0 1 2	

FIGURE 5 HSB TAG FORMAT

BUFFER OPERATION

When a request is made for data from the I- or C-Unit, the S-Unit must first determine where the data is. Using bits 21—26 of the presented address, the S-Unit forms a pointer into the buffer (see Figure 6). Four lines of data are read from each of the primary and alternate halves of the buffer. Bits 19—20 of the real address register are then used to determine which of the four lines is being requested. A four-to-one selection is made and the data is aligned to the addressed byte.

With the associated tag information now available from both the primary and alternate lines, a tag comparison may be performed. The real address bits that correspond to tag information are compared to both of the tags from the primary and alternate reads, and a match is signaled for the appropriate half of the buffer. This match is then used to select which of the aligned words (byte alignment has already been accomplished) will be loaded into the output word register. This operation takes two cycles. Status information, such as protection exception, is available at the same time.

Requests to the S-Unit can be pipelined at the rate of one per cycle; data is returned at the end of two machine cycles. If data is not in the buffer a main storage request is generated. Main storage requests are four-way interleaved for each two megabytes of memory. Four requests, therefore, may be active simultaneously.

VIRTUAL ADDRESS OPERATION

To provide the advantages of virtual memory to users, the Amdahl 470V/6 system can perform dynamic address translation when in EC mode. The virtual address implementation in the 470V/6 follows the System/370 architecture. Performance is further enhanced with a translation lookaside buffer (TLB), which provides high-speed storage of most recently translated addresses, and a segment table origin (STO) stack, which identifies the environment of different TLB entries.

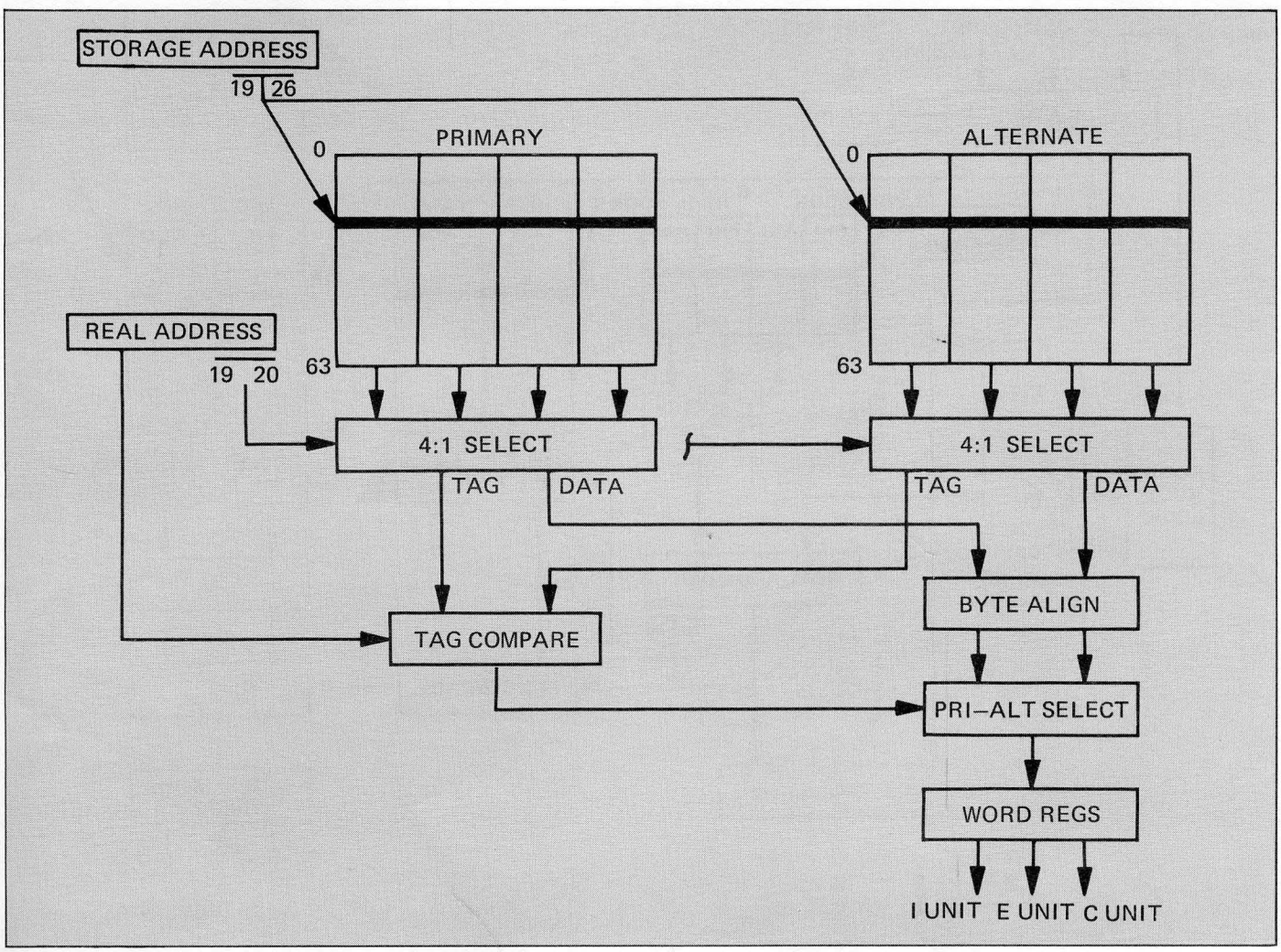


FIGURE 6 HIGH-SPEED BUFFER OPERATION

TLB OPERATION

The TLB provides a fast source of translation information to minimize virtual operation time and to allow overlap of translations and HSB accesses. The TLB is implemented very similarly to the HSB, but consists of 128 virtual-real address pairs in each of the primary and alternate halves. Thus, a correspondence is set up in the TLB between virtual and real address space whenever a translation is performed.

When a virtual address is presented to the S-Unit through the effective address register (EAR), a TLB and an HSB access are both started (see Figure 7). The virtual address maps to a given TLB location. When the associated real address is obtained from the TLB (assuming that an entry was there), a selection is performed based on matching the virtual address in the TLB with the virtual address presented. The real address register (RAR) is then loaded with the effective real line address, and bits 19–20 are used for buffer data selection as described above. If data is not present in the buffer, the real byte address is loaded in the main store address register (MSAR) to obtain data from main storage. S-Unit ports are provided for each of the possible requesting units (instruction fetch, operand, C-Unit, pre-fetch and internal ports) to allow the capability of having requests from the various units active simultaneously.

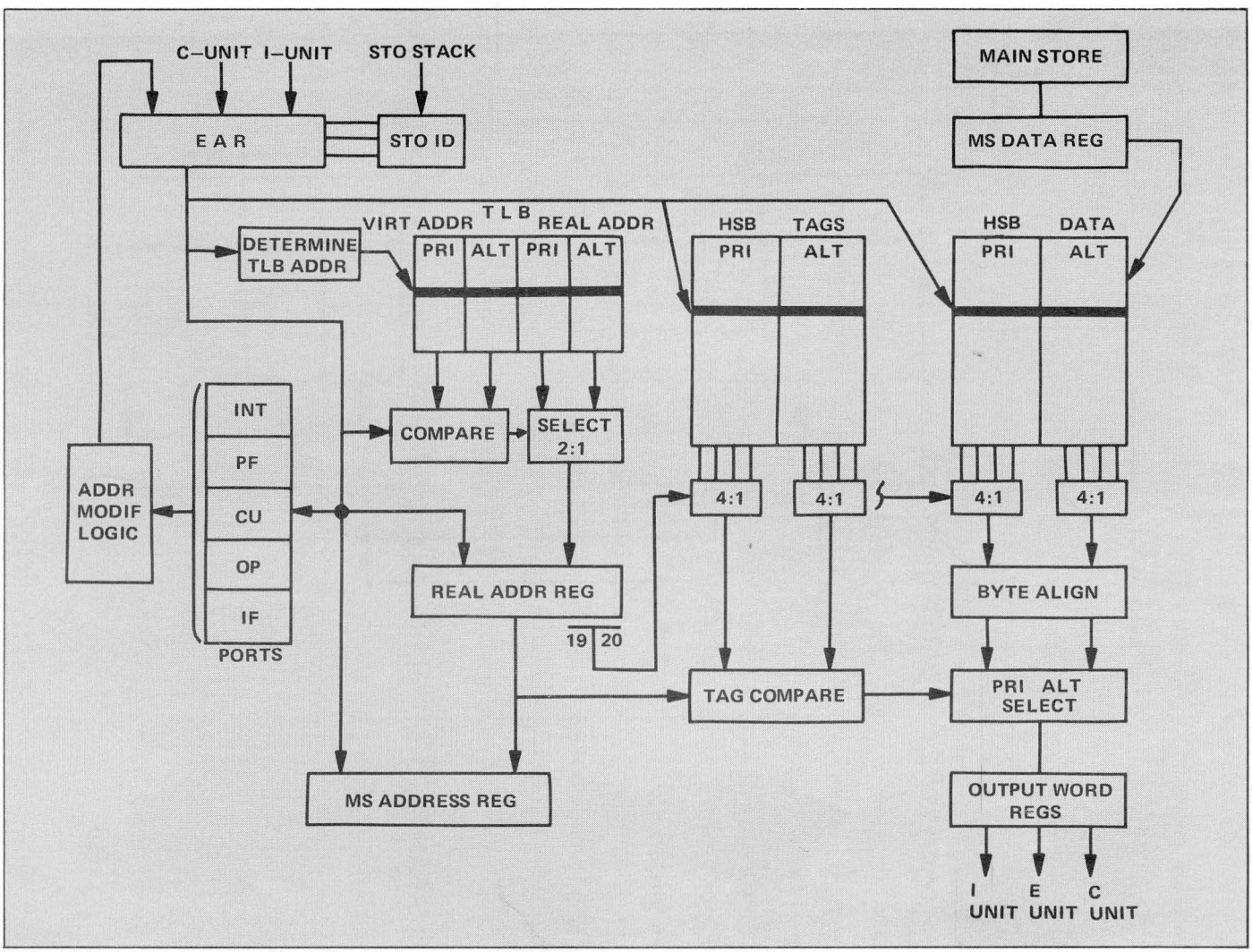


FIGURE 7 S-UNIT OPERATION

If the virtual-real address translation has been performed and information resides in the TLB, no overhead for the TLB access occurs since its access is overlapped with the buffer; thus, data is returned in two cycles for both real and virtual operations. If the virtual-real address pair is not currently valid in the TLB a full translation must be made, which requires two additional storage (either buffer or main store) references. The new translation is placed in the TLB using an algorithm similar to the buffer replacement algorithm. A hot/cold (H/C) bit feature similar to the feature of the HSB is implemented in the TLB.

TLB addresses may be made invalid by a system reset, a Purge TLB instruction, or by changing the state of control registers 0 and 1. To enhance performance, two sets of valid bits are kept in the TLB. When a Purge TLB instruction is executed, the S-Unit automatically switches the set of valid bits it is currently using to the other set, which are all marked "invalid". The S-Unit resets the older set of valid bits during spare cycles to be ready for the next Purge instruction; thus a single Purge TLB instruction may be executed, incurring only the minimum two cycle instruction overhead.

STO STACK OPERATION

The STO stack stores information about control registers 0 and 1 so that TLB entries can be reused if these control registers are reloaded with previous values. This is done by associating a STO identification field (ID) with translation information from control registers 0 and 1. The STO ID is then stored with the TLB and STO stack entries. Valid TLB entries must match the current STO stack entry (which matches current values in control registers 0 and 1).

When a translation is performed, control registers 0 and 1 determine segment table size and location in main storage. Thus, when these registers are changed, the current virtual-real address correspondence becomes invalid. Rather than invalidate all current TLB entries, the S-Unit assigns a new STO ID to be stored with all subsequent TLB entries and enters information from control registers 0 and 1 into the STO stack location associated with the new ID. If control registers 0 and 1 are changed to previous values, any entries remaining in the TLB from old translations need not be made again. The STO stack consists of 32 slots; when the 32nd entry is made, the S-Unit purges the oldest entry from the STO stack and all associated TLB entries during spare cycles. The system thus maintains up to 31 active address spaces. All STO stack entries are invalidated by a Purge instruction.

The Amdahl 470V/6 has 16 standard inboard channels. These channels are independent of the CPU and, with the possible exception of memory access conflicts, do not affect CPU performance. The channels may be configured in any combination of selector channels or byte or block multiplexer channels. A total of 1024 subchannels are available for allocation to multiplexer channels in groups of 64, 128 or 256.

The channel unit (C-Unit) is implemented in both LSI technology and non-LSI, third-generation technology. The LSI portion of the C-Unit provides interface sequencing, data movement to and communication with the S-Unit, and an operational interface to the CPU that obeys the System/370 architecture as defined in the Principles of Operation. The non-LSI portion (channel frame) performs buffering, translation from LSI logic levels to standard interface levels, and signal conditioning (driving and receiving interface signals).

CENTRAL LSI CHANNEL

The LSI portion of the C-Unit is shown in Figure 8. It is functionally divided into three major parts. The central interface control logic (CICL) controls interface sequences and data buffering to and from the external devices; the data access control logic (DACL) updates buffer pointers and counts and moves data between the S-Unit and C-Unit buffers; the operations control logic (OCL) sets up channel transfer sequences and coordinates channel program execution with the operation of the rest of the C-Unit. The shifting channel state (SCS) is a communications area used by each of these major parts.

Data path information is also shown in Figure 8. On an output operation the DACL fetches data a word at a time from the S-Unit and stores it in the channel buffer store (CBS); the CICL moves it to the non-LSI remote interface logic (RIL) for transmission to the control unit. The above sequence is reversed for an input operation.

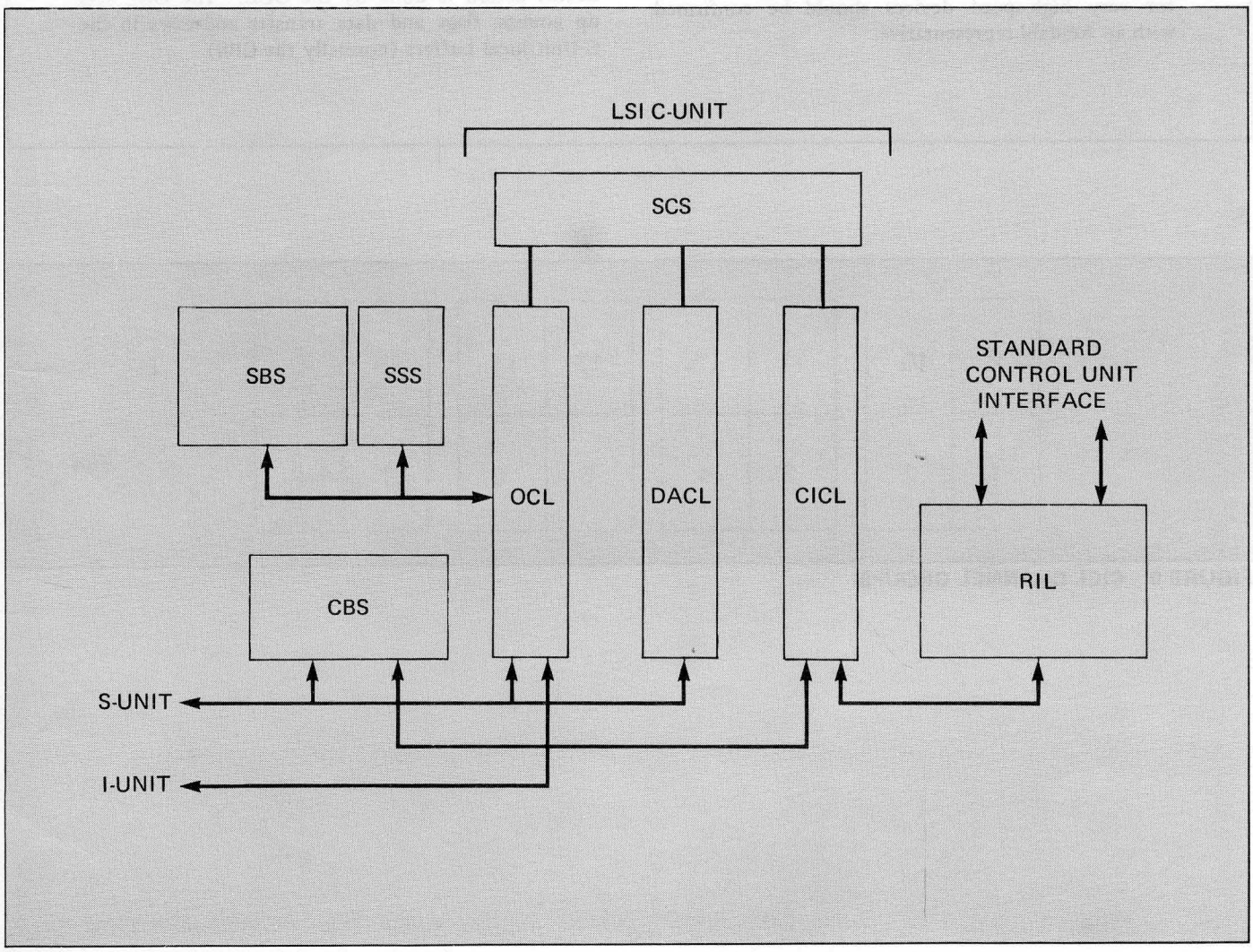


FIGURE 8 C-UNIT

The CICL transfers data to or from the CBS on even cycles; the OCL and DACL share odd cycles. The CICL transfers one byte or two bytes at a time; the DACL and OCL transfer a word at a time. The CICL examines every channel in the SCS once every eight cycles for data transfer requests. Figure 9 illustrates how this property of the CICL may affect channel configuration. A group of four channels as illustrated shares certain resources of the C-Unit buffers. With a machine cycle time of 32.5 nanoseconds, the maximum bandwidth of each group is approximately 3.85 million bytes per second.

Therefore, configuration of devices on channels must insure that this total transfer rate is not exceeded for each group. Specific channel assignments for very high-speed devices should be confirmed with an Amdahl representative.

The DACL is pipelined to optimize throughput. Each channel is examined every 16 cycles. Thus, while one section of the DACL may be fetching data from the S-Unit, another may be posting results to the SCS, and another may be examining the incoming SCS for channels that need service. Each channel is assigned a dynamic priority based on the amount of space it has remaining in the C-Unit buffers. If a channel has less than half of its buffer remaining, the DACL will change its S-Unit request from a low to a high priority. This change can occur even if a fetch or a store is in progress. The highest priority channel is marked from the SCS and serviced next by the DACL.

The translation of channel commands to CICL and DACL action is done by the OCL. The OCL sets up counts, flags and data transfer addresses in the C-Unit local buffers (normally the CBS).

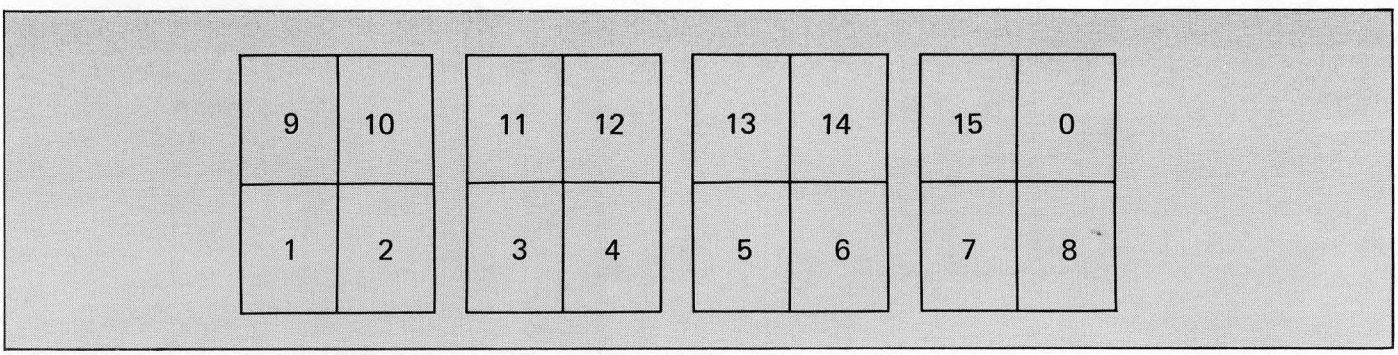


FIGURE 9 CICL CHANNEL GROUPS

To perform byte or block multiplexing, the OCL activates a subchannel (the portion of a channel that actually "executes" a channel program) when appropriate status information is received from the CICL. The OCL uses the subchannel buffer store (SBS) for temporary storage of inactive subchannel information. The OCL can monitor subchannel state and perform subsequent decisions with the information from the subchannel state store (SSS). The OCL obtains its control information directly from the I-Unit and S-Unit over an interface shared with the DACL.

Channel indirect data addressing as described in the *Principles of Operation* is fully implemented on the Amdahl 470V/6 by the OCL and DACL. Indirect data addressing (IDA) requires that a control program perform any necessary virtual-to-real address translations before a data transfer command is executed by the channel.

MODES OF CHANNEL OPERATION

Because of the relative independence of the OCL, DACL and CICL, any channel can be configured as a byte or block multiplexer or selector. Selector channels transfer only in burst mode and may address up to 256 I/O devices, one at a time. Selector channels contain a single implicit subchannel, which does not require allocation from that pool of 1024 subchannels. The total possible data rate per selector channel is the same as the rate for a block multiplexer.

Multiplexer-type channels perform device multiplexing and execute several channel programs simultaneously. The level of concurrency is determined by the number of subchannels allocated to the multiplexer channel; one subchannel is necessary for each concurrent channel program execution. Subchannels are allocated in groups of 64, 128 or 256. With either 64 or 128 assigned subchannels, a channel may be configured for shared subchannel operation. Details on shared subchannel allocation and device address assignment are contained in Appendix C.

Block multiplexer channels perform the basic multiplexer function, but always transfer in burst mode. It is the more efficient type of multiplexer channel since a considerable amount of data is transferred each time a device logically connects to a channel. Maximum data rate on a standard single-byte block multiplexer channel is the same as for any channel operating in burst mode, or approximately two million bytes per second. The optional two-byte bus doubles this available channel bandwidth.

Byte multiplexer channels usually multiplex a very few bytes at a time. The Amdahl 470V/6 allows a byte multiplex-mode bandwidth of approximately 110 kilobytes per second. The byte multiplexer bandwidth for single-byte bus devices running in burst mode is approximately two million bytes per second. Selector subchannels are not implemented in the Amdahl 470V/6 byte multiplexer channels.

Extended channel logout is fully implemented on the Amdahl 470V/6. If control register 14 bit 2 is set, extended logouts will be made when certain errors occur. No logout will be made if this bit is off. More information on extended channel logout fields is found in Appendix C.

NOTE: Maximum burst mode data rates cited are maintainable with chaining only when the chaining occurs during device "gap time". The use of IDA flags in a virtual environment has a similar effect on data bandwidth. The possibility of overrun, therefore, greatly increases when chaining in a virtual environment.

FIGURE 10 SYSTEM CONSOLE

The Amdahl 470V/6 system console serves as an operator's console and as a system maintenance console (Figure 10). It consists of a CRT and keyboard, an independent console processor, a disk for the console processor, a diskette unit to provide the Amdahl field engineer with a diagnostic test loading facility, and a modem to allow remote access to the Amdahl diagnostic facility. A standard channel interface allows the console processor to emulate a 3066 or 3215 operator's console. A standard operator's control panel is also provided (Figure 11).

The Amdahl 470V/6 system console can operate in three modes. Device support mode provides emulation of standard consoles for data transmission to and from the operator using the system console and a channel; hardware command mode allows display and storage of registers and memory, reset, clear, and other hardware functions; and maintenance mode allows operation of diagnostic programs and provides formatted displays of some 16,000 latches within the Amdahl 470V/6. Device support mode can emulate either a 3066 CRT display or a 3215 printer/keyboard using the CRT display instead of a printer. In device support mode the system console functions as a control unit on either a selector or a block multiplexer channel.

A direct computer-to-console interface is provided between the system console and the central processor. This separate path is independent of the channel and is used by hardware command and maintenance modes to permit static readout of system latches and setting of certain control and data registers. This provides the system console with increased diagnostic and maintenance flexibility.

Two functions of the Amdahl 470V/6 system console differentiate it from most other similar consoles presently available:

- Most console input functions are performed using the keyboard rather than toggle or rotary switches.

- Most scan-out and console output functions appear as formatted displays on the CRT rather than as panel lights.

To increase system availability and reliability, the system console can perform initial program loading using only the console processor and the diskette unit. For further information on functions, features and operations of the system console, see Appendix D.

USAGE METERING

System and maintenance meters are located in the system console next to the console processor. Which of these meters runs is a function of the maintenance key switch. Time is accumulated on the system meter when the switch is in the system position and the SYSTEM light is on. Time is accumulated on the maintenance meter when the switch is in the maintenance position. If there is no check condition in the CPU and the CPU is not in the STOP or WAIT state, or if the channels are active, then the SYSTEM light will be on.

FIGURE 11 OPERATOR'S CONTROL PANEL

OPERATOR'S CONSOLE PROGRAMMING

For 3066 emulation the system console CRT screen will display 35 lines of 80 characters each. Data may be read from or transmitted to any of these positions on the CRT under program control. Characters typed on the keyboard can be placed in any of these positions on the CRT, depending upon the location of the cursor.

The screen is addressed by a two-byte buffer field and a two-byte cursor address field. The first byte in both address fields gives the line number (0 to 34); the second byte gives the position within the line (0 to 79). This form of address is used to control where information will be written on the screen from the program, which locations on the screen are to be transferred in response to a read from the program, and where the cursor will be placed for keyboard entry. A two-byte address is also returned to the program when the cursor position is requested.

The following commands control the console:

NOP	X'03'	No operation.
Sense	X'04'	Reads two bytes of sense data (see Table 1).
Erase	X'07'	Sets the entire screen to blanks, removes the cursor display, resets CRT buffer address and cursor address to zeros.
Alarm	X'0B'	Sounds a one-second tone and lights the alarm key.
Set Buffer Address	X'27'	Transfers a two-byte screen address to the console to designate the starting byte position for a subsequent Read and Write command.
Write	X'01'	Transmits EBCDIC data to be displayed, starting from the current value of the CRT buffer address, and advances this address by one for each byte transferred. The operation stops when the CCW count is exhausted or when 2800 bytes are written. If position (34, 79) is reached, position (0, 0) is written next.
Read	X'06'	Transfers data from the screen to the program, starting from the current CRT buffer address, and continues until either the CCW count is exhausted or the byte at the current cursor position is transferred.
Set Cursor	X'0F'	Transfers a two-byte address to the console to indicate the screen position at which the cursor should be displayed. If the cursor was not previously displayed and the keyboard was disabled, it can now be used to enter data onto the screen.
Read MI	X'0E'	Usually issued in response to an attention interruption caused by either the "ENTER" or the "CANCEL" keys, this command returns three bytes of information to the program. The first and second bytes are the current cursor address; the third byte indicates which key was pressed (X'80') for "ENTER" and X'40' for "CANCEL").
Lock Keyboard	X'67'	Causes the cursor to be deleted from the screen and prevents keyboard entry upon the screen.

Sense Data:		
Byte 0	Bit 0	Command reject
	Bit 1	Reserved
	Bit 2	Bus out check
	Bit 3	Equipment check
	Bit 4	Data check
	Bit 5	Reserved
	Bit 6	Buffer address check
	Bit 7	Reserved
Byte 1		Reserved

TABLE 1 SYSTEM CONSOLE SENSE DATA

There are a few functional differences between the operation of the Amdahl 470V/6 system console in 3066 emulation as explained above and the operation of the 3066 system console. The following should be noted if exact compatibility is a requirement:

The 3066 keyboard has both upper and lower case alphabetic input even though the CRT displays only upper case. The Amdahl 470V/6 system console has only upper case alphabetic input.

The system console may respond to initial selection with a Control Unit Busy Sequence and Status = 70_{16}. This sequence can occur if the selection was immediately preceded by a Halt I/O to the Console or if the console is selected while not in either 3066 or 3215 emulation mode.

MODEL-DEPENDENT FUNCTIONS

The Amdahl 470V/6 System and the IBM System/370 are compatible within the constraints of System/370 architecture as specified in the *Principles of Operation*. This specification requires machine compatibility except in the following cases:

Programs relying on model-dependent data, such as the contents of logout areas

Time-dependent programs that rely on instruction or CCW execution times

Programs that cause deliberate machine checks

The Amdahl 470V/6 System has four areas of model dependence: the implementation of two instructions, the contents of CPU logout areas, machine check conditions, and channel logout. Only the first three are discussed here; channel logout is discussed in Appendix C.

INSTRUCTION SET DIFFERENCES

Two instructions have model-dependent results on the Amdahl 470V/6. They are:

STORE CPU ID (STIDP). This instruction causes certain model-dependent data to be stored at the double word addressed by the second operand address. In the Amdahl 470V/6 the version code field is reserved (bits 0-7). A unique serial number is stored in bits 8-31. The hexadecimal number 0470 is stored in the model number field (bits 32-47). Bits 48-63 are stored as zeros since the Amdahl 470V/6 performs the standard machine check extended logout to the memory of the console processor.

STORE CHANNEL ID (STIDC). Channel dependent data is to be stored at location 168. Since the 470V/6 channel type is implicit in the CPU model, zeros are stored in the channel model number field. The remaining fields (type and length) follow standard conventions.

CPU LOGOUT AREAS

Fixed CPU logout areas are assigned for storage of machine check logout information. Machine check logouts are the result of or associated with a machine error and result in storing model-dependent information.

FIXED LOGOUT AREA

The 104-byte area starting at location 248 is the fixed logout area. The Amdahl 470V/6 uses only the first 12 bytes of this area for fixed logout information. The failing storage address occupies the word starting at location 248; the region code is assigned to the two words starting at location 252. The remainder of the area, locations 260—351, is reserved.

The failing storage address (FSA) indicates the storage location or block that caused a correctable storage, an uncorrectable storage, or an uncorrectable key error. For a correctable storage error, bits 0—3 of the FSA contain the failing bit address; bits 8—31 contain the failing byte address. When an uncorrectable storage error is detected, bits 8—31 of the FSA may point anywhere within the failing 16-byte ECC block. When an uncorrectable key error is detected, bits 8—31 of the FSA may point anywhere within the failing 2048-byte protection block. In the case of multiple errors, the FSA may point to any one of the failing locations. In some cases an FSA cannot be stored. When this occurs, the FSA valid bit in the machine check interrupt code (MCIC) is set to zero. (The MCIC is discussed under machine checks).

Bytes 252—259 contain the region code that specifies where the error was detected. A summary of the region code bits is given in Table 2.

Storage Location	Bit	Source	Storage Location	Bit	Source
252	0	I–Unit Pipeline Control Error	256	0	E–Unit Multiplier Byte Parity Error
	1	E–Unit Condition Code Error		1	E–Unit Byte Adder Input 1 Parity Error
	2	E–Unit LUCK1 Byte 0 Parity Error		2	E–Unit Byte Adder Input 2 Parity Error
	3	E–Unit LUCK1 Byte 1 Parity Error		3	E–Unit Byte Adder Input 3 Parity Error
	4	E–Unit LUCK1 Byte 2 Parity Error		4	S–Unit Bypass Error
	5	E–Unit LUCK1 Byte 3 Parity Error		5	S–Unit TLB Key Parity Error
	6	E–Unit LUCK2 Byte 0 Parity Error		6	S–Unit TLB Logical Address Parity Error
	7	E–Unit LUCK2 Byte 1 Parity Error		7	S–Unit RAR Parity Error
253	0	E–Unit LUCK2 Byte 2 Parity Error	257	0	I–Unit Result Byte 0 Parity Error
	1	E–Unit LUCK2 Byte 3 Parity Error		1	I–Unit Result Byte 1 Parity Error
	2	E–Unit Multiplicand Byte 0 Parity Error		2	I–Unit Result Byte 2 Parity Error
	3	E–Unit Multiplicand Byte 1 Parity Error		3	I–Unit Result Byte 3 Parity Error
	4	E–Unit Multiplicand Byte 2 Parity Error		4	I–Unit EAG Parity Error (DA)
	5	E–Unit Multiplicand Byte 3 Parity Error		5	I–Unit EAG Parity Error (CI)
	6	E–Unit Adder High-Input Phase Error		6	I–Unit Instruction Stream Entrance Parity Error
	7	E–Unit Adder Low-Input Phase Error		7	I–Unit Store Data Parity Error
254	0	S–Unit Search Error	258	0	S–Unit Address Translation Error
	1	S–Unit Compare Register Parity Error		1	S–Unit Channel Request
	2	S–Unit Tag Key Parity Error		2	C–Unit I/O Address Parity Error From I–Unit
	3	S–Unit Tag ID Parity Error		3	Reserved
	4	S–Unit Store Data Parity Error		4	C–Unit Error on CSW Store
	5	Main Store Read Address Parity Error		5	Reserved
	6	Main Store Key Write Parity Error		6	Reserved
	7	Main Store Write Address Parity Error		7	Reserved
255	0	S–Unit Tag Control Parity Error	259	0	E–Unit Multiplier Residue Error
	1	S–Unit Move Out Data Parity Error 0		1	E–Unit Adder Residue Error
	2	S–Unit Move Out Data Parity Error 1		2	I–Unit Instruction Stream Exit Parity Error (DS)
	3	S–Unit Move Out Data Parity Error 2		3	S–Unit TLB Valid or ID Error
	4	S–Unit Move Out Data Parity Error 3		4	I–Unit Control Register Bytes 0 - 1 Parity Error
	5	S–Unit Primary (1)/Alternate (0) Buffer		5	I–Unit Control Register Bytes 2 - 3 Parity Error
	6	S–Unit Primary (1)/Alternate (0) TLB		6	I–Unit PSW Bytes 0 - 1 Parity Error
	7	S–Unit Translation Register Segment/Page		7	Reserved

TABLE 2 REGION CODE

EXTENDED CPU LOGOUT

A machine check extended logout (MCEL) is performed by the console processor based on the setting of control register 14 bit 1, the synchronous MCEL mask. If the bit is on and a condition occurs that would normally cause an extended logout, the logout information is stored in the console processor.

The console processor detects the MCEL situation and performs a logout by scanning out all pertinent CPU latches to its own memory. This information can later be examined by a program in the main processor by issuing a special Diagnose instruction. After the console logout, the main processor is restarted to allow the CPU to continue its normal operation. If the console is not operational, the CPU is restarted by a time-out function and the MCEL is not performed. The CPU is informed of this action on the subsequent Diagnose.

Because MCEL data is saved in the console processor, the machine check extended logout area (location 512 and above) is not used by the Amdahl 470V/6. Control register 15, which normally contains the MCEL address, is not implemented and stores as all zeros.

The Diagnose instruction (operation code x'83') provides a means of communication between the CPU and the console processor for diagnostic and maintenance purposes. Because the operation of the Diagnose instruction may differ depending on the engineering change level of a particular system, documentation of this instruction is not provided in this manual.

MACHINE CHECKS

Extensive checking facilities are implemented in the Amdahl 470V/6. These include parity checking of instructions, registers and data, arithmetic checking of execution functions and effective address generation, and checking for certain illegal control sequences. When processing damage is detected, an attempt can usually be made to retry the instruction. (Details on which instructions are retryable are contained in Appendix B). If the retried instruction is successful, machine operation can continue and a system recovery condition is recognized. A similar retry facility is implemented in the 470V/6 channels. Additional correction facilities exist in the S-Unit for main storage error correction with automatic error checking and correction (ECC). The ECC field provides sufficient information to correct any single bit error for an associated group of 16 bytes. This type of correction is also recognized as a system recovery condition.

MACHINE CHECK CONDITIONS

A machine check condition is recognized for an uncorrectable error. A machine check condition is generated whenever an instruction or data with invalid parity is fetched, an arithmetic function is improperly performed, or invalid parity on the protection key makes it impossible to establish whether protection applies. Certain invalid control sequences can also cause a machine check condition. A machine check condition can be generated when instructions or data with invalid parity are fetched and before they are used.

If storage and registers contain valid information a machine check condition will be caused only by a machine malfunction, and never by an invalid instruction or data. If an unavailable component is specified the appropriate program, I/O interrupt or condition is set; a machine check condition is not generated.

A malfunction detected by the S-Unit during an I/O operation causes an external damage machine check condition. When the I/O operation is a fetch of a CCW or data, the malfunction is reported in the channel status word (CSW). When data is stored by the channel and a malfunction is detected in the S-Unit after status has been returned to the C-Unit, the CSW does not report the error. When the channel detects bad parity on an input operation, good parity is forced to the S-Unit and a channel data check is reported in the CSW.

An external I/O equipment malfunction detected in the channel is indicated in a CSW and not treated as a machine check condition. The error is reported as an I/O interrupt.

Machine malfunctions that cause machine check interrupts are grouped into two classifications, soft and hard machine checks. These correspond to repressible and exigent conditions defined in the *Principles of Operation*.

Soft machine checks comprise system recovery conditions, timer damage conditions, time-of-day clock damage, external damage and degradation conditions (associated with the S-Unit STO stack). Retryable errors and storage errors resulting from I/O or prefetch operations cause soft machine checks. These conditions do not terminate the current instruction or cause loss of interrupts.

Hard machine checks comprise instruction processing damage conditions (if retry is unsuccessful or impossible) and system damage conditions. Multiple bit

storage errors, storage key parity errors, and internal CPU data transfer errors that are unretryable are considered hard machine checks. These conditions may result in termination of the current instruction or loss of interrupts.

MACHINE CHECK INTERRUPTS

The machine check interrupt reports an equipment malfunction and supplies information about the location and nature of the malfunction. Actions taken by the Amdahl 470V/6 System in machine check condition are summarized in Table 3.

A soft machine check occurs after an instruction, including any associated SVC or program interruption, has completed. This is the same point at which an I/O interrupt occurs. Two classes of soft machine checks are retryable errors and corrected errors. A retryable error starts re-execution of the problem instruction; a corrected error is handled as a soft machine check (so far as point of interruption is concerned).

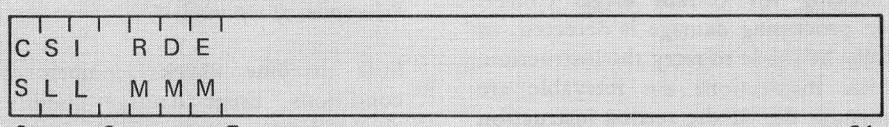
A hard machine check immediately inhibits further updating of machine state, including storage and registers, and points the instruction counter to the instruction farthest along in the pipeline. This action is taken immediately, before the end of an instruction, but still could have been caused by a part of the execution of any of the instructions in the pipeline (up to six).

With the exception of processing of machine check extended logouts and the associated nonimplementation of control register 15, the Amdahl 470V/6 fully implements standard machine check handling as defined in the *Principles of Operation*. The significant bits of control register 14 and the machine check interrupt code (MCIC) for the Amdahl 470V/6 are detailed in Figures 12 and 13.

	MCIC STORED	SYS DMG	PROC DMG	SYS REC	BACKUP	GPRs STORED	FSA STORED	REG CODE STORED
SOFT MACHINE CHECK								
CORRECTED	YES	0	0	1	0	**	**	NO
RETRIED	YES	0	0	1	0	**	**	**
HARD MACHINE CHECK								
UNCORRECTED	YES	0	1	0	†	**	**	**
UNRETRYABLE	YES	*	*	0	0	**	**	**

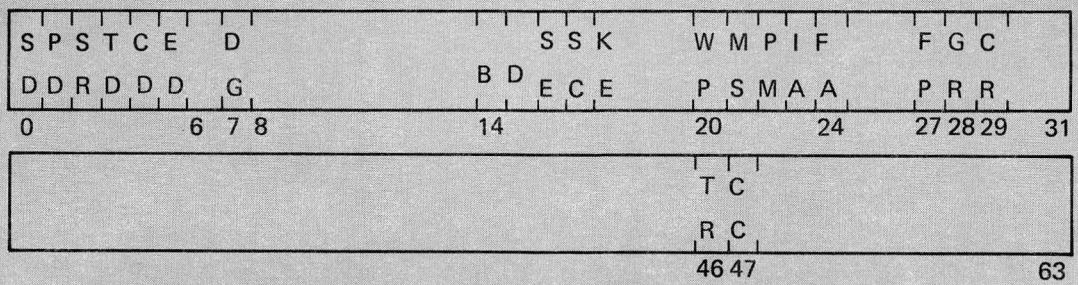
* = 1 AS APPROPRIATE
† = 1 IF NOT PAST RETRY THRESHOLD; ELSE = 0
** IF AVAILABLE

TABLE 3 AMDAHL 470V/6 MACHINE CHECK ACTION



BIT	FUNCTION
CS	Check stop
SL	Synchronous machine check extended logout channel
IL	I/O extended logout mask
RM	Recovery report mask
DM	Degradation report mask
EM	External damage report mask

FIGURE 12 CONTROL REGISTER 14 — MACHINE CHECK CONTROL REGISTER



BIT	FUNCTION	BIT	FUNCTION
SD	System damage	KE	Key in storage error corrected
PD	Processor damage	WP	PSW EMWP valid
SR	System recovery	MS	PSW masks and key valid
TD	Timer damage	PM	Program masks and CC valid
CD	Timing facilities damage	IA	Instruction address valid
ED	External damage	FA	Failing storage address valid
DG	Degradation	FP	Floating point registers valid
B	Backed-up	GR	General purpose registers valid
D	Delayed	CR	Control registers valid
SE	Storage error uncorrected	TR	CPU timer valid
SC	Storage error corrected	CC	Clock comparator valid

FIGURE 13 MACHINE CHECK INTERRUPT CODE (MCIC)

HARDWARE DIAGNOSTIC MODE

When a check stop or some other condition occurs that requires manual intervention, a diagnostic mode (hardware command or maintenance mode) may be entered from the operator's console. Data may then be stored or displayed, and the machine state may be otherwise changed from the console. Further details on these procedures are found in Appendix D.

INSTRUCTION TIMING ESTIMATES

The timing formulas given below were derived through engineering analysis of system documentation. Because the Amdahl 470V/6 is complex and has many overlapping functions, the formulas make certain assumptions and are not completely comprehensive. While efforts were made to assure accuracy of data included, it is possible that errors have been made and these timings are not warranted to be correct. Future system measurement and analysis are planned to further modify and refine the formulas. In addition, Amdahl Corporation reserves the right to make changes and enhancements that may alter the timings as herein presented.

The following table lists the number of cycles required for execution of each of the instructions of the Amdahl 470V/6. Included with some of the formulas are variables that affect execution; these are explained at the end of the table. In addition, those instructions indicated by an asterisk are sufficiently complex that only imprecise estimates of their execution times are given in the table (see "Special Timing Formulas"). Because of the parallel, complex architecture of the Amdahl 470V/6, care should be exercised in the use of these formulas. As is readily determined from an analysis of the special terms in many instructions, the execution times for a given instruction can often vary greatly.

For most instructions, all data needed to assemble and execute the instruction is assumed to be in the buffer. If it isn't, a main storage request must be made. Main storage access times will vary according to the level of interleaving, contending requests, etc.

Information on early condition code setting and retryability of certain instructions is also included in the table.

EARLY CONDITION CODE SETTINGS

The timing estimate for the conditional branch instruction depends on how early the condition code becomes valid. Condition code setting instructions have a term in the "CC Set" column that specifies the number of cycles to add to the execution of an associated conditional branch instruction. An "N" indicates that the speed of the condition code depends on whether the instruction operands are normalized. This term is further explained under "Special Timing Formulas".

RETRYABLE INSTRUCTIONS

The given digits and symbols indicate thresholds beyond which the instruction becomes nonretryable. The notation "SI" indicates that the instruction is not retryable after a store in the instruction stream. The absence of any digit or symbols indicate the instruction is completely retryable. Other cases of nonretryable instructions include threshold numbers. The BAL instruction has a threshold of two cycles. If the error is detected during the first two cycles of BAL processing, the instruction is retryable; otherwise, it is not. A CLCL instruction has a threshold code of "E8", denoting that it cannot be retried if an error is detected in the last eight cycles of processing (End-8). For decimal operations, like Add Decimal (AP), the symbol is E-2R, denoting that the end factor is instruction completion (End) less than two cycles for each result word to be stored. The Branch on Condition (BC and BCR) has a threshold of zero when the branch is taken (T).

TIMING FORMULAS

INSTRUCTION	FORMAT	MNEMONIC	CC SET	RTRY	PERFORMANCE
Add	RR	AR	0		2
Add	RX	A	0		2
‡Add Decimal	SS	AP	0	E—2R	13→47 + S4
Add Halfword	RX	AH	0		2
Add Logical	RR	ALR	0		2
Add Logical	RX	AL	0		2
Add Normalized (Extended)	RR	AXR	0	2	15 + 3*RC + AL + NP + DWD
‡Add Normalized (Long)	RR	ADR	N		7 + 2*RC + RRR
‡Add Normalized (Long)	RX	AD	N		7 + 2*RC + RRR
‡Add Normalized (Short)	RR	AER	N		6 + RC
‡Add Normalized (Short)	RX	AE	N		6 + RC
‡Add Unnormalized (Long)	RR	AWR	N		7 + 2*RC + RRR
‡Add Unnormalized (Long)	RX	AW	N		7 + 2*RC + RRR
‡Add Unnormalized (Short)	RR	AUR	N		6 + RC
‡Add Unnormalized (Short)	RX	AU	N		6 + RC
AND	RR	NR	0		2
AND	RX	N	0		2
AND	SI	NI	0	2	4 + S1
AND	SS	NC	0	4	6L + S1
Branch and Link	RR	BALR		2	8 + DWD
Branch and Link	RX	BAL		2	8 + DWD
Branch on Condition	RR	BCR		0(T)	2 + BT*(3—S3) + CCS + S3 + 16*BCR
Branch on Condition	RX	BC		0(T)	2 + BT*(3—S3) + CCS + S3
Branch on Count	RR	BCTR		2	4 + S3 + BT*(3—S3)
Branch on Count	RX	BCT		2	4 + S3 + BT*(3—S3)
Branch on Index High	RS	BXH		2	6 + S3 + BT*(3—S3)
Branch on Index Low or Equal	RS	BXLE		2	6 + S3 + BT*(3—S3)
‡Clear I/O	S	CLRIO	0	2	26 + CURT
Compare	RR	CR	0		2
Compare	RX	C	0		2
Compare and Swap	RS	CS			24
‡Compare Decimal	SS	CP	0		13→47 + S4
Compare Double & Swap	RS	CDS			32
Compare Halfword	RX	CH	0		2
Compare Logical	RR	CLR	0		2
Compare Logical	RX	CL	0		2
Compare Logical	SI	CLI	0	2	2
Compare Logical	SS	CLC	0		4*W + 4*X + 4*Y
Compare Logical Characters Under Mask	RS	CLM	+2		2
Compare Logical Long	RR	CLCL	0	E8	36 + 4*W + 4*X + 4*Y
Compare	RR	CDR	+2		3 + 3*UN + RRR
Compare	RX	CD	+2		3 + 3*UN + RRR
Compare	RR	CER	+2		3 + UN
Compare	RX	CE	+2		3 + UN
Convert to Binary	RX	CVB			5 + N + 2*D + RRR
Convert to Decimal	RX	CVD		SI	42 + S2 + N1

INSTRUCTION	FORMAT	MNEMONIC	CC SET	RTRY	PERFORMANCE
Divide	RR	DR			50
Divide	RX	D			50
‡ Divide Decimal	SS	DP		E–2R	31→2036 + S4
Divide (Long)	RR	DDR			64 + PN1 + PN2 + DPS
Divide (Long)	RX	DD			64 + PN1 + PN2 + DPS
Divide (Short)	RR	DER			27
Divide (Short)	RX	DE			27
Edit	SS	ED	0	4	6*L + S1
Edit and Mark	SS	EDMK	0	4	6*L + 6
Exclusive OR	RR	XR	0		2
Exclusive OR	RX	X	0		2
Exclusive OR	SI	XI	0	2	4 + S1
Exclusive OR	SS	XC	0	4	6*L + S1
Execute	RX	EX			24 + TGE
‡ Halt I/O	S	HIO	0	2	26 + CURT
‡ Halt Device	S	HDV	0	2	26 + CURT
Halve (Long)	RR	HDR			6 + RRR
Halve (Short)	RR	HER			4
Insert Character	RX	IC			2
Insert Character Under Mask	RS	ICM	0		9
Insert PSW Key	S	IPK			2
‡ Insert Storage Key	RR	ISK			23→75
Load	RR	LR			2
Load	RX	L			2
Load Address	RX	LA			2
Load and Test	RR	LTR	0		2
Load and Test (Long)	RR	LTDR	+1		2 + RRR
Load and Test (Short)	RR	LTER	0		2
Load Complement	RR	LCR	0		2
Load Complement (Long)	RR	LCDR	+1		2 + RRR
Load Complement (Short)	RR	LCER	0		2
Load Control	RS	LCTL		2	6 + 2*R + 16*RIF
Load Halfword	RX	LH			2
Load (Long)	RR	LDR			2 + RRR
Load (Long)	RX	LD			2 + S3 + RRR
Load Multiple	RS	LM		2	2*R + 2
Load Negative	RR	LNR	0		2
Load Negative (Long)	RR	LNDR	+1		2 + RRR
Load Negative (Short)	RR	LNER	0		2
Load Positive	RR	LPR	0		2
Load Positive (Long)	RR	LPDR	+1		2 + RRR
Load Positive (Short)	RR	LPER	0		2
Load PSW	S	LPSW		2	28
‡ Load Real Address	RX	LRA			17→70
Load Rounded (Extended to Long)	RR	LRDR			7 + RN
Load Rounded (Extended to Short)	RR	LRER			4 + RRR
Load (Short)	RR	LER			2
Load (Short)	RX	LE			2
Monitor Call	SI	MC			6 + ME

INSTRUCTION	FORMAT	MNEMONIC	CC SET	RTRY	PERFORMANCE
Move	SI	MVI		SI	2 + S1
Move	SS	MVC		6	6 + MV + S1 + 4*WB
Move Long	RR	MVCL	0	2	45 + 4*W + 4*BPE + 16*PE
Move Numerics	SS	MVN		4	6*B + S1
Move with Offset	SS	MVO		4	2 + MVB + S1
Move Zones	SS	MVZ		4	6*B + S1
Multiply	RR	MR			7
Multiply	RX	M			7
‡ Multiply Decimal	SS	MP		E–2R	21→409 + S4
Multiply	RR	MXR		2	94 + PN1 + PN2 + DWD
Multiply Halfword	RX	MH			8
Multiply (Long)	RR	MDR			20 + PN1 + PN2 + RRR
Multiply (Long)	RX	MD			20 + PN1 + PN2 + RRR
Multiple (Long to Extended)	RR	MXDR		2	28 + PN1 + PN2/2 + DWD
Multiply (Long to Extended)	RX	MXD		2	28 + PN1 + PN2/2 + DWD
Multiply (Short)	RR	MER			8
Multiply (Short)	RX	ME			8
OR	RR	OR	0		2
OR	RX	O	0		2
OR	SI	OI	0	2	4 + S1
OR	SS	OC	0	4	6*L + S1
‡ Pack	SS	PACK		4	4*B + S1
‡ Purge TLB	S	PTLB			2 (Nominal)
Read Direct	SI	RDD			88 + HRT
‡ Reset Reference Bit	S	RRB	0	2	23→75
Set Clock	S	SCK	0		6 + TODRT
Set Clock Comparator	S	SCKC			6 + TODRT
Set Program Mask	RR	SPM			6
Set CPU Timer	S	SPT			6 + TODRT
Set PSW Key from Address	S	SPKA			18
‡ Set Storage Key	RR	SSK		SI	156→384
Set System Mask	S	SSM		2	8 + EC
Shift & Round Decimal	SS	SRP	0	E–2R	40 + S2
Shift Left Double	RS	SLDA	+2		4 + RRR
Shift Left Double Logical	RS	SLDL			3 + RRR
Shift Left Single	RS	SLA	+2		2
Shift Left Single Logical	RS	SLL			2
Shift Right Double	RS	SRDA	+2		3 + RRR
Shift Right Double Logical	RS	SRDL			2 + RRR
Shift Right Single	RS	SRA	+2		2
Shift Right Single Logical	RS	SRL			2
‡ Start I/O	S	SIO	0	2	26 + CURT
‡ Start I/O Fast Release	S	SIOF	0	2	26 + CURT
Store	RX	ST		SI	2 + S1 + DWD
‡ Store Channel ID	S	STIDC	0	2	26 + CURT
Store Character	RX	STC		SI	2 + S1 + DWD
Store Character Under Mask	RS	STCM		SI	4 + S1
Store Clock	S	STCK	0	SI	22 + TODRT
Store Clock Comparator	S	STCKC		SI	6 + S2 + TODRT
Store Control	RS	STCTL		SI	2*R + S2 + DWD

INSTRUCTION	FORMAT	MNEMONIC	CC SET	RTRY	PERFORMANCE
Store CPU ID	S	STIDP		SI	4 + S2 + DWD
Store CPU Timer	S	STPT		SI	6 + S2 + TODRT
Store Halfword	RX	STH		SI	2 + S1 + DWD
Store (Long)	RX	STD		SI	4 + S2 + DWD
Store Multiple	RS	STM		SI	2*R + S2 + DWD
Store (Short)	RX	STE		SI	2 + S1 + DWD
Store Then AND System Mask	SI	STNSM		SI	24
Store Then OR System Mask	SI	STOSM		SI	24
Subtract	RR	SR	0		2
Subtract	RX	S	0		2
‡ Subtract Decimal	SS	SP	0	E2	13→47 + S4
Subtract Halfword	RX	SH	0		2
Subtract Logical	RR	SLR	0		2
Subtract Logical	RX	SL	0		2
Subtract Normalized (Extended)	RR	SXR	0	2	15 + AL + 3*RC + NP + DWD
‡ Subtract Normalized (Long)	RR	SDR	N		7 + 2*RC + RRR
‡ Subtract Normalized (Long)	RX	SD	N		7 + 2*RC + RRR
‡ Subtract Normalized (Short)	RR	SER	N		6 + RC
‡ Subtract Normalized (Short)	RX	SE	N		6 + RC
‡ Subtract Unnormalized (Long)	RR	SWR	N		7 + 2*RC + RRR
‡ Subtract Unnormalized (Long)	RX	SW	N		7 + 2*RC + RRR
‡ Subtract Unnormalized (Short)	RR	SUR	N		6 + RC
‡ Subtract Unnormalized (Short)	RX	SU	N		6 + RC
Supervisor Call	RR	SVC			44
Test and Set	S	TS	0	0	18
‡ Test Channel	S	TCH	0	2	26 + CURT
‡ Test I/O	S	TIO	0	2	26 + CURT
Test Under Mask	SI	TM	0		2
Translate	SS	TR		4	5* $\lceil B/4 \rceil$ + 2 + 4*B + S1 ‡‡
Translate and Test	SS	TRT	0	2	5* $\lceil B/4 \rceil$ + 4*B + 4*M + 4 ‡‡
Unpack	SS	UNPK	4		5* $\lceil (B-1)/2 \rceil$ + 4 + UPK * (B–1) ‡‡
Write Direct	SI	WRD		2	86
Zero and Add	SS	ZAP	0	E–2R	13 + 5*WL + S4

‡ More precise timing may be obtained for these instructions by referring to the section entitled "Special Timing Formulas" in this appendix.

‡‡ The term in brackets is the "greatest integer" function; hence, if B = 1 in $\lceil B/4 \rceil$ the term = 1.

LEGEND FOR TIMING FORMULAS

SYMBOL	EXPLANATION
AL	Alignment exponent difference $<8 = 9$; $\geq 8, <16 = 8$; $\geq 16 = 7$
B	Number of bytes moved, packed/unpacked, or translated.
BCR	If B1 = F and R2 = 0 then BCR = 1; otherwise, BCR = 0.
BPE	The number of bytes (less than 4) remaining to be accessed by either the fetch or store field for a given page.
BT	Branch taken = 1; otherwise = 0.
CCS	Condition code setting factor. The value is given in CC Set column.
CURT	Channel unit release time. For detailed explanation of this term, see the "Special Timing Formulas" section of the Appendix.
D	Number of significant decimal digits.
DPS	= 1 if dividend preshift required, else = 0.
DWD	DWD = 1 if a double word result instruction precedes subject instruction.
EC	In EC mode, EC = 16.
HRT	Hold response time. The value ranges from 0 to 33 with equal probability; the average is 16.5.
L	In SS-type instructions L = the length in bytes of the first operand.
L2	In SS-type instructions L2 = the length in bytes of the second operand.
M	equals 1 if any other than last byte to be tested is nonzero; otherwise, M = 0.
ME	ME = 46 cycles if the event is monitored.
MV	equals: No overlap or overlap >32 bytes or both operands on word boundary — 4*W 3 bytes $<$ overlap ≤ 32 bytes or both operands not on word boundary — 5*W 1 byte $<$ overlap ≤ 3 bytes — 4*B overlap = 1 byte — 6*B
MVB	equals: No overlap or overlap $\neq 1$ byte — 4*B overlap = 1 byte — 6*B
N	= 1 if sign negative; otherwise = 0.
NP	Post normalization. Leading zero digits $\leq 6 = 6$; $>6 \leq 14 = 10$; $>14 = 7$
N1	= 1 if first operand's sign is negative; else = 0
PE	When the fetch or store field reach a page boundary, PE = 1.
PN1	= 2 if prenormalization required on operand one, else = 0.
PN2	= 2 if prenormalization required on operand two, else = 0.
R	Number of registers to be loaded or stored.
RIF	RIF = 1 if instruction modifies CR0 or CR1; or if PER is enabled and instruction modifies CR9, CR10 or CR11. Otherwise, RIF = 0.
RC	Recomplement. = 1 if recomplementation required; otherwise = 0.
RN	= 1 if rounding required, else = 0.
RRR	If the instruction immediately following the subject instruction has a single word operand, then RRR = 0 if $E \geq 4$ 1 if E = 3 2 if E = 2 where E is the number of execution cycles in that that instruction. If the instruction immediately following the subject instruction has a double word operand, then RRR = 0 if $E \geq 5$ 1 if E = 4 2 if E = 3 3 if E = 2
S1	S1 = 2 if M = 6 = 1 if M = 7 = 0 if M = 8 where M equals the number of execution cycles attributable to the three words of the instruction stream following the instruction of interest.

SYMBOL	EXPLANATION
S2, S3	S2 = 0 if P = 8 = 1 if P = 7 = 2 if P = 6 = 3 if P = 5 = 4 if P = 4 S3 = 0 if P = 6 = 1 if P = 5 = 2 if P = 4 where P equals the number of execution cycles attributable to the two words of the instruction stream following the instruction of interest.
S4	Use S3 if both operands total less than five bytes; otherwise, use S2.
TGE	Target instruction execution time.
TODRT	Time-of-day response time. The value ranges from 1 to 30 with equal probability; the average is 15.5.
UN	= 1 if either operand unnormalized; otherwise, = 0.
UPK	equals 1 if overlap present.
W	Number of words compared or moved.
WB	For L $\geq$ 64, WB equals the number of bytes which must be moved to have the store field aligned on a word boundary. For L $<$ 64, WB = 0
WL	Number of words in the longer operand.
X	In logical comparison (CLC, CLCL), X = 1 if an inequality is detected in other than the last word compared; otherwise, X = 0.
Y	In logical comparison (CLC, CLCL), Y = 1 if an inequality is detected in other than the last two words compared; otherwise, Y = 0.

SPECIAL TIMING FORMULAS

Several instructions were flagged in the main list, since only a wide variation in execution times could be given there. More precise estimates of execution times can be determined using the tables and formulas below.

ADD DECIMAL (AP)

	Length in Bytes of the Longer Operand			
	1–4	5–8	9–12	13–16
SIGNS ALIKE	13	23	33	43
SIGNS DIFFERENT	15	25	36	47

COMPARE DECIMAL (CP)

	Length in Bytes of the Longer Operand			
	1–4	5–8	9–12	13–16
SIGNS ALIKE	15	25	36	47
SIGNS DIFFERENT	13	23	33	43

CURT

Channel response time depends on broad variety of conditions. The formulas given below assume no interference of other channels with an I/O instruction issued to a channel. All values are machine cycles except values of variable C.

	Channel Idle/Subchannel Idle		
Instruction	Minimum	Maximum Selector	Maximum Byte & Block Multiplexer
SIO	246 + BB	261 + BB	261 + BB
SIOF	36	51	51
TIO	230 + BB	245 + BB	245 + BB
TCH	4	19	19
STIDC	20	35	35
HIO	278 + BB	293 + BB	293 + BB
HDV	326 + BB	341 + BB	341 + BB
CLIO	20	35	35

Channel Idle/Subchannel Working

Instruction	Minimum	Maximum Selector	Maximum Byte & Block Multiplexer
SIO	20	35	35
SIOF	20	35	35
TIO	20	35	35
TCH	4	19	19
STIDC	20	35	35
HIO	274 + BB	289 + BB	289 + BB
HDV	274 + BB	289 + BB	337 + BB
CLIO	20	35	153

Channel Idle/Subchannel Interrupt Pending

Instruction	Minimum	Maximum Selector	Maximum Byte & Block Multiplexer
TIO	36	51	99

Channel Working

Instruction	Minimum	Maximum Selector	Maximum Byte Mpx in Burst Md	Maximum Block Mpx
SIO	4	19 + A	19 + C	19 + A
SIOF	4	19 + A	19 + C	19 + A
TIO	4	19 + A	19 + C	19 + A
TCH	4	19 + A	19 + C	19 + A
STIDC	4	19 + A	19 + C	19 + A
HIO	20	35	35 + C	35
HDV	20	35	35 + C	83
CLIO	4	19 + A	19 + C	19 + A

Where:

A = delay due to reselection during chaining of a working subchannel.
Min = 0 cycles Max = 210 cycles + BB

C = 120 to 135 microseconds

BB = extra delay caused by tag turnaround time to device. Let T equal time in microseconds from issuance of out tag to receiving of in tag.

$$BB = 8 \times \left\lceil \left(\frac{T}{32.5 \times 10^{-3}} - 11 \right) / 8 \right\rceil^{*}$$

* The term in brackets is the "greatest integer" function.

DIVIDE DECIMAL (DP)

If L2 < 3, formula is 20 + (5 + 4Q)N + 2RW
If 3 < L2, formula is 27 + (9 + 6Q)N + 2RW

Where Q equals the average value of the quotient digits plus 1.
N equals the number of quotient digits.
RW equals the number of words of result.

FLOATING POINT OPERATIONS – EARLY CONDITION CODE SETTING FACTOR

	CC Set for Branch	
	Taken	Not Taken
Operands Normalized, no recomplementation	−1	0
Operands Normalized, recomplementation	−2	0
Operands Unnormalized, no recomplementation	+2	+2
Operands Unnormalized, recomplementation	+2	+2

INSERT STORAGE KEY (ISK)

Minimum 23 cycles, maximum 75 cycles. Execution time is dependent upon whether the main storage or the pre-fetch port to main storage is busy when ISK is executed.

LOAD REAL ADDRESS (LRA)

Minimum 17 cycles; assumes segment page table entries in high-speed buffer. Maximum 70 cycles; assumes no table entries in high-speed buffer and maximum main storage access.

MULTIPLY DECIMAL (MP)

The cycle estimate assumes a random distribution of multiplier digits. If the multiplier data is not random, the performance can vary greatly. The number of cycles required increases as the multiplier digits go from a value of zero to a value of four, or go from a value of nine to a value of five. The cycles shown in the table below represent an average.

MULTIPLIER LENGTH IN BYTES

MULTIPLICAND LENGTH IN BYTES	1	2	3	4	5	6	7	8
1–4	21	39	97					
5–8	31	61	91	121	145	165	185	
9–12	42	86	130	174	210	240	270	300
13–16	36	114	172	230	277	321	365	409

PACK (PACK)

No overlap assumed. From 1 to 16 cycles must be added for overlap.

PURGE TLB (PTLB)

Minimum 2 cycles. This assumes that PTLBs are executed with low enough frequency such that invalidating TLB entries can be done in spare CPU cycles and at PTLB execution time only switching from one set of "TLB entry valid" bits to another is required. This is expected to be the nominal case. If PTLBs are executed close together then the second PTLB can take up to 384 cycles.

RESET REFERENCE BIT (RRB)

Minimum 23 cycles, maximum 75 cycles. Execution time is dependent upon whether the main storage or the prefetch port to main storage is busy when RRB is executed.

SET STORAGE KEY (SSK)

Minimum 156 cycles, maximum 384 cycles. Execution time is dependent upon how many lines associated with key block are in high-speed buffer when SSK is executed.

SUBTRACT DECIMAL (SP)

	Length in Bytes of the Longer Operand			
	1 – 4	5 – 8	9 – 12	13 – 16
SIGNS ALIKE	15	25	36	47
SIGNS DIFFERENT	13	23	33	43

APPENDIX C:

MODEL-DEPENDENT CHANNEL FUNCTIONS

SUBCHANNEL ASSIGNMENT

The Amdahl 470V/6 channels allow assignment of up to 1024 subchannels for multiplexer-type channels (byte or block). Subchannels may be assigned in groups of 64, 128, or 256.

I/O unit addresses are of the form CUU, where C is the channel address and UU is the hexadecimal device address. For multiplexer channels with 128 subchannels, only the low order 7 bits of the device address are significant; for 64 subchannels, only the low order 6 bits are significant. For example, on a multiplexer channel with 64 subchannels, no distinction is made among addresses 301, 341, 381, and 3C1. Figure 14 illustrates the subchannel addressing uniqueness limitations.

Shared subchannel assignment may be made for channels with either 64 or 128 subchannels. When 64 subchannels are assigned to a channel, 4 are shared and 60 are unshared; for 128 subchannels, 8 are shared and 120 are unshared.

The shared subchannel addresses for 64 subchannels are 00 to 03, with all device addresses 64 and above sharing this group of 4 subchannels. The shared subchannels above 63 map to these 4 by taking the higher order 4 bits of the device address modulo 4. Device addresses in the range of 04 to 3F (hex) map to unshared subchannels.

The shared subchannel addresses for 128 subchannels are 00 to 07, with all device addresses 128 and above sharing these 8 subchannels. The shared subchannels above 127 map to this group of 8 by taking higher order 4 bits of the device address modulo 8. Device addresses in the range of 08 to 7F (hex) map to unshared subchannels.

Figure 15 diagrams the above relations. Control units permitting shared subchannels are assigned to one of the groups of shared subchannels. Control units using unshared subchannels must be assigned device addresses from the unshared section of subchannels. On byte multiplexer channels, one control unit (at most) may be assigned addresses within a block of addresses mapping to a shared subchannel. On block multiplexer channels, block multiplexing is inhibited on all devices that have device addresses associated with a shared subchannel. Multiple control units can be assigned to the same shared subchannel.

NO. OF SUBCHANNELS	NO. OF UNIQUE DEVICE ADDRESSES	SIGNIFICANT BITS	ADDRESS GROUPS
64	64	– –XX XXXX	00–3F 40–7F 80–BF C0–FF
128	128	– XXX XXXX	00–7F 80–FF
256	256	XXXX XXXX	00–FF

FIGURE 14 DEVICE ADDRESS ASSIGNMENT, UNSHARED SUBCHANNELS

	UNSHARED		SHARED	
TOTAL SUBCHANNELS	NO. SUBCH	DEV ADR RANGE	NO. SUBCH.	DEVICE ADDRESS GROUPS
64	60	04–3F	4	00 40→4F, 80→8F, C0→CF 01 50→5F, 90→9F, D0→DF 02 60→6F, A0→AF, E0→EF 03 70→7F, B0→BF, F0→FF
128	120	08–7F	8	00 80→8F 01 90→9F . . . . . . 07 F0→FF
256	256	00–FF	0	None

FIGURE 15 SHARED SUBCHANNELS, DEVICE ADDRESS ASSIGNMENT

EXTENDED CHANNEL LOGOUT

The Amdahl 470V/6 channels perform an extended logout (IOEL) when any of several severe errors (defined in the *Principles of Operation*) occur and control register 14 bit 2 is set. A severe error is normally classified as a channel control check, an interface control check or a channel data check.

A diagram of the Amdahl 470V/6 IOEL is given in Figure 16. The first 4 words are selected bits from the LSI channel state, the next 12 are from the Channel Buffer Store control information, and the last field is from the subchannel state store (SSS). Since the number of subchannels varies from channel to channel, the length of this last field also varies. For selector channels it has length zero, and for multiplexer channels it has a length of 8, 16, or 32 words depending on whether 64, 128 or 256 channels are installed, respectively. A further breakdown of the channel states (IOEL words 0–3) is given in Figure 17. Undefined fields generally contain information of interest only to field engineering (for detailed error analysis).

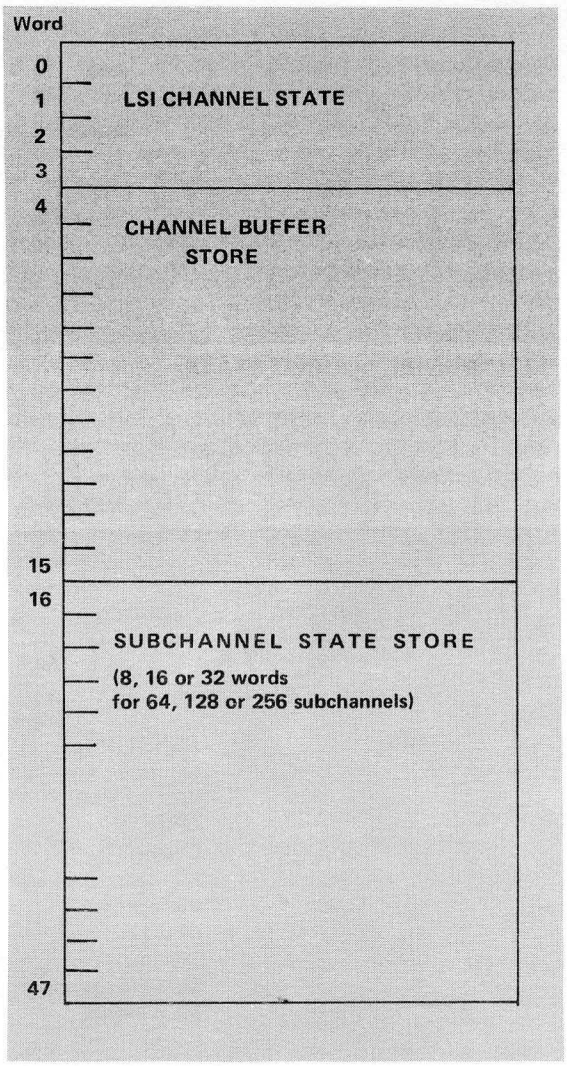


FIGURE 16 I/O EXTENDED LOGOUT FORMAT

LIMITED CHANNEL LOGOUT

The Amdahl 470V/6 channels perform an extended channel logout of error conditions only. Some programs require a limited channel logout format. The following is provided for information purposes only.

Since the 470V/6 channels do not perform a limited channel logout (LCL), tables are provided to diagram how this information can be generated from the extended logout data. A standard limited channel logout word is diagrammed in Figure 18.

Table 4 defines several global variables, which are used in Tables 5 and 6. Normal Boolean algebra notation is used in all the tables.

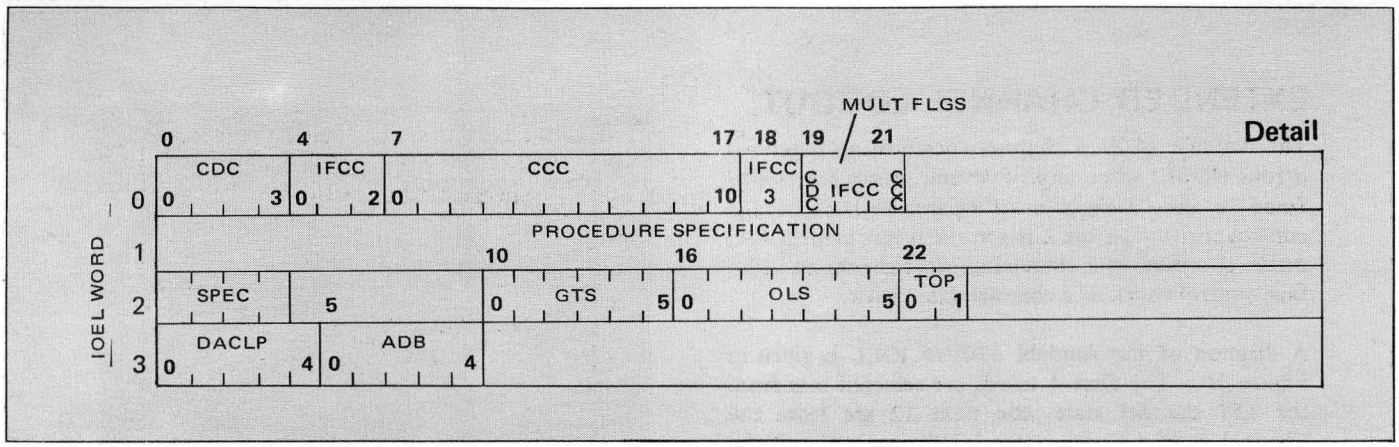
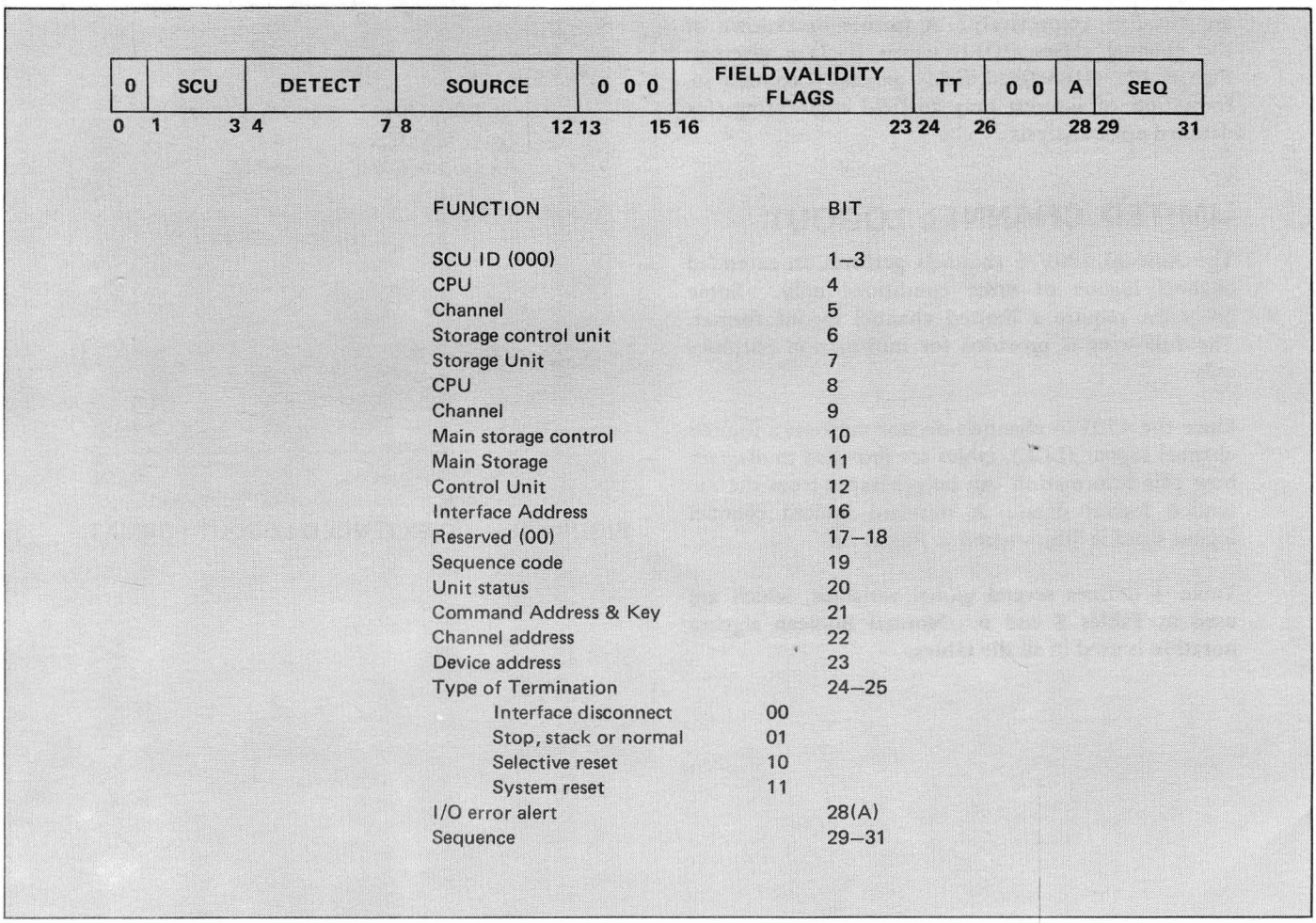


FIGURE 17 CHANNEL LOGOUT STATE



FUNCTION		BIT
SCU ID (000)		1–3
CPU		4
Channel		5
Storage control unit		6
Storage Unit		7
CPU		8
Channel		9
Main storage control		10
Main Storage		11
Control Unit		12
Interface Address		16
Reserved (00)		17–18
Sequence code		19
Unit status		20
Command Address & Key		21
Channel address		22
Device address		23
Type of Termination		24–25
Interface disconnect	00	
Stop, stack or normal	01	
Selective reset	10	
System reset	11	
I/O error alert		28(A)
Sequence		29–31

FIGURE 18 LIMITED CHANNEL LOGOUT WORD

To process an IOEL to form the LCL, the CSW stored is first examined. An LCL should be generated if any of the three error bits, channel control check (CCC), interface control check (IFCC), or channel data check (CDC) is on. A valid LCL, the type of error, some miscellaneous variables, and the sequence code and validity bit are generated from Table 4. Then, either Table 5 or 6 is used to generate the remainder of the LCL word. The LCL word contains information about type of error, the source, and how much of the diagnostic information stored may be valid.

* CCC=$\overline{\text{MULT CCC}}$ •[CCC(0+1+2+3+4+5+6+7+8+9+10)]

IFCC=$\overline{\text{MULT IFCC}}$ [IFCC(0+1+2+3)]

CDC=$\overline{\text{MULT CDC}}$ [CDC(0+1+2+3)]

INVALID LCL=MULT CCC + MULT IFCC + MULT CDC

** USV= [GTS=65+35+27+37+11+12+13+14+17]

*** $\overline{\text{CAV}}$=EL01:(00+02+06+07+12+15+16)+EL02:03

† CAS=CSW(32–39)=0X00XX00

†† DPP=MOD(DACLP+ADB,32)

UPP=(TOP=01) (DPP≠0)+(TOP=1X) [DACLP≠0)+(ADB≠0)]

USV2= [GTS=65+35+27+37+11+12+13+14+17] $\overline{\text{IFCC(1)}}$

SEQUENCE CODE

000 ← TOP=00

001 ← GTS=63+64+65

010 ← (GTS=35) CAS

011 ← (GTS=10)+(GTS=66+67+7X) UPP

100 ← (GTS=40+44+45+5X)+(GTS=60+61+62)+
(GTS=00) (OLS=1100XX)+(GTS=35) $\overline{\text{CAS}}$

101 ← GTS=2X+30+31+32+33+36)+(GTS=34+06+07)+
(GTS=00) (OLS=101XXX)+(GTS=66+67+7X) $\overline{\text{UPP}}$

If none of the above, sequence validity bit (LCL 19) is set to zero.

* The global functions CCC, IFCC, and CDC are indicated by the unqualified appearance of the identifier. Individual bits within a field are denoted by an integer within parentheses, e.g., bit 0 of the CCC field is denoted by CCC(0). The ". " and " + " are the normal Boolean operators AND and OR, and the bar (a) denotes negation.

** USV is unit status valid. CAV is command address valid; CAS is command accept status, and UPP is updated pointer. The GTS field only is always given in octal; hence, a GTS field of 37 would be 011 111 in binary. The "x" character implies that a given digit (binary or octal) is a "don't care" condition and should not be considered.

*** IOEL fields not otherwise identified are addressed as ELww:bb, where EL stands for extended logout, ww for the word displacement in Figure 17, and bb for the bit number.

† CSW (32–39) is the channel status word bits 32–39 stored on occurence of the error.

†† The operation specified here is "the sum of DACLP and ADB modulo 32". The " + " is not a Boolean operator here only.

TABLE 4 GLOBAL VARIABLE TABLE

C1=CCC(0)+CCC(1)+CCC(2)+CCC(3)
C4=CCC(4)+CCC(5)
C8=CCC(8)+CCC(0)

LCL Bits	Detect			
4		CPU=0		
5		CH=C1+CCC(6)+CCC(9)+C8		
6		MSC=C4+CCC(7)		
7		MS=0		
	Source			
8		CPU=0		
9		CH=C1+CCC(6)+C8+CCC(9)		
10		MSC=CCC(4)+CCC(7)		
11		$MS=\overline{CCC(4)}\cdot CCC(5)$		
12		CU=0		
	Validity			
16		INT ADR=0		
19		SEQ CODE=CCC(6)·SEQ CODE TAB		
20		UNIT STAT=[CCC(7)+CCC(9)]·USV		
21		CMD ADR=C1+C4+CCC(7)+CCC(9)+C8·CAV		
22		CHAN ADR=$\overline{CCC(6)}$		
23		DEV ADR=C1+C4+CCC(7)+CCC(9)+C8·$\overline{(EL1:(13+14))}$		
	Type of Termination			
24,25		00	I/F Disc	← 0
		01	Stop, STCK, NOR	← 0
		10	Sel Rst	← $\overline{CCC(6)}$
		11	Sys Rst	← CCC(6)
28	I/O Error Alert = 0			
29–31	Sequence Code (See Table)			

TABLE 5 CHANNEL CONTROL CHECK (CCC)

LCL Bits	Detect	
4		CPU=0
5		CH=IFCC+CDC·$\overline{CDC(0)}$
6		MSC=CDC(0)
7		MS=0
	Source	
8		CPU=0
9		CH=$\overline{(CDC(0) + (TOP=1X)\cdot CDC(3))}$
10		MSC=0
11		MS=CDC(0)
12		CU=IFCC+(TOP=IX)·CDC(3)
	Validity	
16		INT ADR=0
19		SEQ CODE=CDC+IFCC·SEQ CODE TAB
20		UNIT STAT=CDC+IFCC·USV
21		CMD ADR=1
22		CHAN ADR=1
23		DEV ADR=CDC+$\overline{(IFCC(1)\ GTS=33)}$·IFCC
	Type of Termination	
24,25	00	IF/DSC
	01	Stop, STCK, NOR ← CDC
	10	Sel Res ← IFCC
	11	Sys Res
28	I/O Error Alert = IFCC (2)	
29–31	Sequence Code	If CDC ← 011
		If IFCC See Table

TABLE 6 CHANNEL DATA CHECK (CDC) OR INTERFACE CONTROL CHECK (IFCC)

CONSOLE PROCESSOR SYSTEM

This appendix outlines the functions of the Console Processor System (CPS). It describes the procedures necessary to load, initialize and operate the console in its various modes of operation. For additional information, consult the Amdahl 470V/6 Operations Guide and appropriate IBM documentation.

The Console Processor System operator interface is via a keyboard/CRT display. The keyboard/CRT display responds as an IBM 3066 or 3215 operator console and as a CPS command/display console.

MODES OF OPERATION

CPS has three modes of operation, which are controlled from the console keyboard.

- Device support mode is the normal mode of operation. The keyboard/CRT display emulates an IBM 3066 or 3215 operator console so that the operator can establish communications with the Amdahl 470V/6.

- Control mode of operation provides the operator with the capability to alter and monitor CPS and hardware operations. Since the Amdahl 470V/6 does not have the display lights, buttons and switches that many computers provide for displaying and altering hardware registers and conditions, these functions are provided via keyboard commands.

  The control mode of operation is independent of the device support mode of operation except for the use of keyboard/CRT display. That is, device support communication with the system continues in the background while the keyboard/CRT display facilities are being used in control mode.

- Maintenance mode of operation provides the capability to use the console processor as a vehicle to perform diagnostic maintenance activities by Amdahl field engineering personnel.

AMDAHL 470V/6 POWER UP AND POWER DOWN

The procedure to power up the 470V/6 follows.

Set the console processor switches, located behind the front cover, as below and put the console processor key in lock position.

Switch	0	1 2 3	4 5 6	7 8 9	10 11 12	13 14 15
0 = DOWN 1 = UP	1	0 1 0	0 0 1	1 0 0	0 1 0	0 0 0

If the 470V/6 is off, press POWER ON. Power status messages will be displayed. At completion of power on, the message SYSTEM POWERED ON will be displayed. The operator may continue with an IPL.

For power off, set the console processor switches as above and press the POWER OFF button.

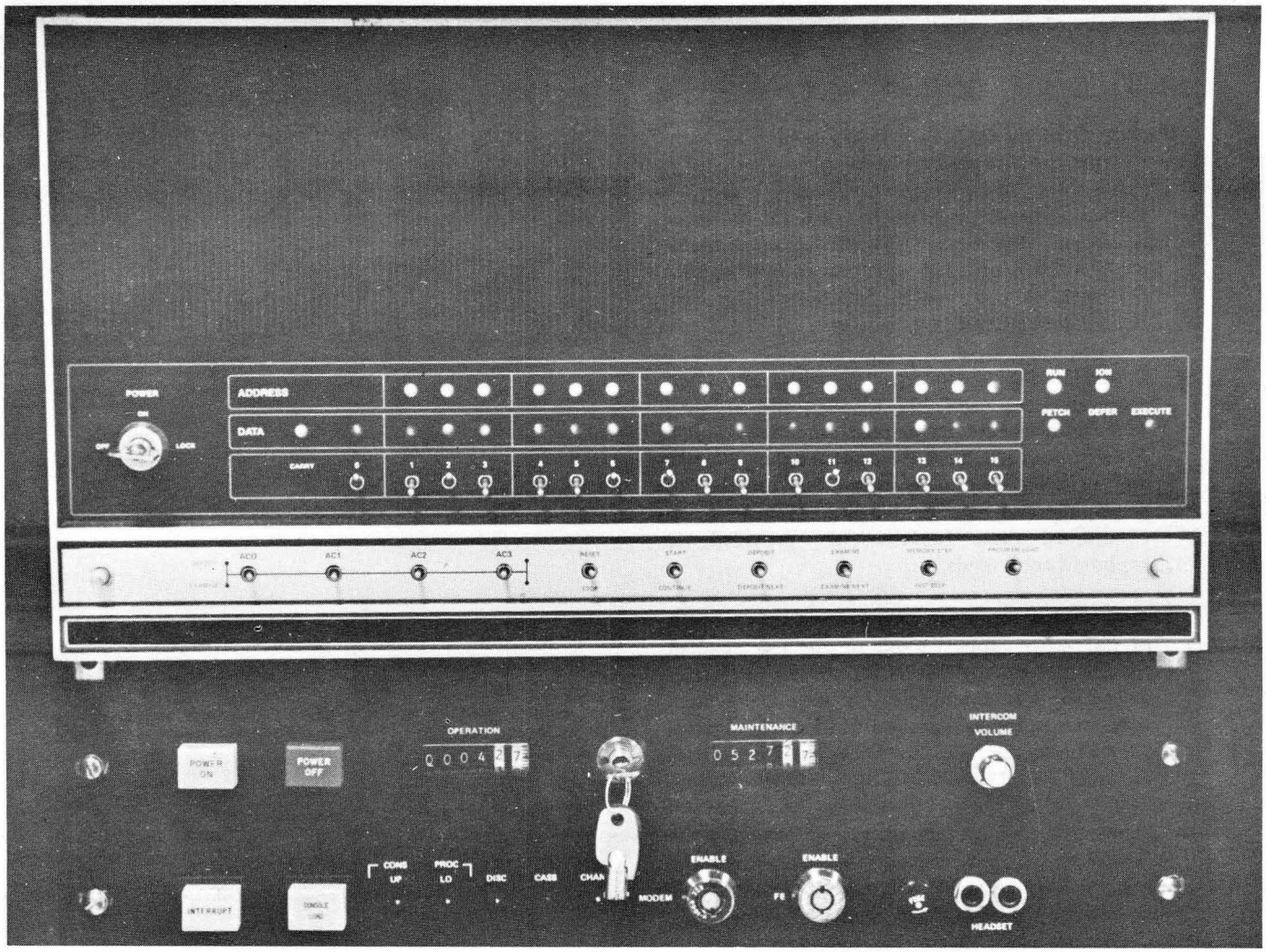


FIGURE 19 CONSOLE PROCESSOR

INITIALIZATION

Initialization of CPS is accomplished by depressing either the POWER ON or LOAD button on the operator's control panel. In addition to the power on and initial program load (IPL) functions of these switches, they also bring an initialized copy of CPS to console processor memory from the console's library device and cause it to begin execution.

INITIAL PROGRAM LOAD

The IPL procedure for the Amdahl 470V/6 is as follows.

Set the console processor switches as above. Set the LOAD UNIT dials to the address of the desired I/O device. Press the LOAD button.

NOTE: IPL may also be accomplished with the IPL keyboard command. This IPL command is described later.

STATUS DISPLAY

The status display occupies the top five lines of the console display CRT regardless of mode of operation. The status display is designed to give an overall snapshot of CPS and Amdahl 470V/6 operation and to notify the operator of abnormal or unusual conditions. The status display presentation has a number of fields that are periodically updated to reflect changes. Each field is discussed in terms of indication, meaning and cause.

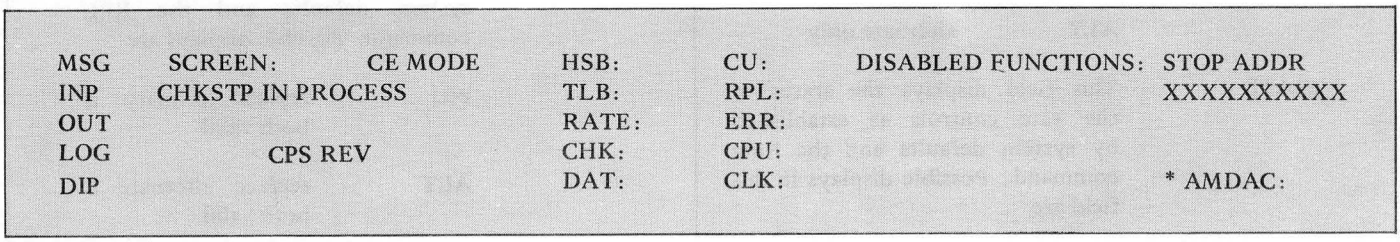


FIGURE 20 FULL STATUS DISPLAY

MSG — The MSG indicator flashes continuously whenever a message has been generated by CPS. Issuing the SE command will cause the message(s) to be displayed on the console display CRT, clear the message stack and turn off the MSG indicator.

INP — The notice INP appears whenever operator action is required by a user subsystem and the console keyboard/CRT is in control mode. The notice flashes until the request is honored by shifting to device support mode and responding to the request.

OUT — The notice OUT appears whenever output is pending from a user subsystem and the console keyboard/CRT is in control mode. The notice flashes until the output is allowed by shifting to device support mode.

LOG — Displays selected machine check log number

DIP — A Diagnose instruction is being handled by the system console.

SCREEN — In command mode, the digit displayed indicates which scan screen is currently being displayed.

CHKSTP IN PROCESS — A 470V/6 check stop is being handled by the system console

CPS REV — Characters displayed represents the change level of CPS currently being used in the system.

CE MODE — The CE key has been enabled.

HSB — This field shows the present configuration of the high-speed buffer (HSB). HSB configuration is determined by system defaults and the HSB command. Possible displays in this field are

NORM	full buffer
PRI HI	primary, upper half only
PRI LO	primary, lower half only
PRI BOTH	primary, both halves
ALT HI	alternate, upper half
ALT LO	alternate, lower half
ALT BOTH	alternate, both halves

TLB — This field shows the present configuration of the translation look-aside buffer (TLB). TLB configuration is determined by system defaults and the TLB command. Possible displays in this field are

NORM	both
PRI	primary only
ALT	alternate only

RATE — This field displays the status of the rate controls as established by system defaults and the Rate command. Possible displays in this field are

INST	single instruction mode
CYCLE	single cycle mode
INST CYCLE	both single instruction and single cycle
PROCESS	process state

CHK — This field displays the status of machine check stop enables as established by system defaults and the CHK command. Possible displays are

NORM	checks are logged
DISABLE	checks are disabled
STOP CPU	stop on CPU checks only
STOP CU	stop C–Unit on channel checks only
STOP ALL	stop on all checks

DAT — This field shows the status of dynamic address translation as determined by system defaults and the DAT command. For normal operation, the display is

NORM	translation controlled by machine registers

CU — This field indicates whether the channel unit is on line. Possible displays are

ON	online
OFF	offline

RPL — This field displays high-speed buffer replacement control as specified by system defaults and the RPL command. Possible displays are

PRI	replace primary if both valid
ALT	replace alternate if both valid
NORM	replace according to replacement algorithm

ERR — For normal operation, the display is

ERR	NORM

CPU — This field displays the CPU process status dynamically. Possible displays are

STOP	the CPU is stopped
WAIT	the CPU is in wait state
PROC	the CPU is in process state

CLK — This field displays the status of CPU clocks dynamically. Possible displays are

OFF	CPU clocks are off
OFFC	channel unit clocks only are off
ON	all clocks on
*OFF	clock error condition

DISABLED FUNCTIONS

This field itemizes the processor functions that are allowed by system defaults and disabled by the NO command. Possible items in this list are

BYP	high-speed buffer bypass
CTMR	console timeout facility
ECC	error checking and correction
MOD	all lines modified
MS	main storage
MSER	main storage error checks
OVLP	overlap
PF	prefetch
PROT	main storage protection checks
PRTY	storage unit priority
REPL	HSB replace on valid and hot/cold only
RTRY	retry
SBR	segment buffer register
SOFT	soft machine checks
SUER	storage unit error checks
TMR	interval timer

STOP ADDR

This field displays the stop address register contents and the stop conditions. Possible displays in this field are

STOP ADDR

XXXXXXXX	stop address
IA	stop on instruction
F	stop on operand fetch
ST	stop on operand store
CF	stop on channel check
CS	stop on channel store

*AMDAC This field shows the status of the remote assistance feature. Possible displays are:

AMDAC C	control mode; AMDAC site has control
AMDAC M	monitor mode; AMDAC site is only monitoring

ERRORS

There are two types of errors, recoverable and fatal, that can occur while CPS is in operation. As the category titles imply, fatal errors imply CPS must be initialized by pressing the LOAD button. Recoverable errors signify only an operator error or the failure of a function or operation in the system.

The detection of a non-fatal error is signaled by the flashing of the MSG indicator on the status display. The operator should select the message screen. The error message is then displayed at the bottom of the screen. Depending on the nature of the error, the operator initiates recovery actions. If the error persists, a field engineer should be called.

Fatal errors are of such a serious nature that CPS abruptly halts after displaying a PANIC HALT message on the CRT. Fatal errors occur when unrecoverable errors prevent CPS from continuing. A CPS reinitialization is necessary. If the error persists, a field engineer should be called.

CONTROL MODE OPERATION

Control mode can be entered as a CPS initialization option or when SHIFT/STOP is depressed when in device support mode. Control mode is recognized by the appearance of the CRT image. On CPS initialization entry to control mode, CRT lines 1 through 5 will show the status display; lines 6 through 40 will be blank except for the appearance of the keyboard input cursor in the lower lefthand corner. On SHIFT/STOP entry to CPS, again, lines 1 through 5 show the status display, but lines 6 through 40 will redisplay whatever was formerly displayed in control mode.

In control mode, all keyboard entries are handled by CPS. Certain key entries are processed immediately, regardless of context, but normally CPS commands are formatted on the keyboard echo line of the CRT display, and the ENTER key initiates command interpretation and processing by CPS.

The following keys are processed immediately by CPS when in control mode regardless of context.

ALARM	Clears the console processor alarm lamp.
BACKSPACE	Blanks the current input cursor position and moves the input cursor one position to the left.
CANCEL	Clears the keyboard entry line and resets the input cursor.
ENTER	Causes CPS to process the current input. On normal completion, the input line is rolled up one line, the input line cleared and the cursor reset. On abnormal termination, the input line remains and MSG flashes on the status display.
*HOME	No operation.
*REQUEST	No operation.
START	The Amdahl 470V/6 is started unless the rate is cycle or both, in which case a single cycle is executed.
STOP	The Amdahl 470V/6 is stopped
SHIFT/STOP	The keyboard/CRT display is switched to device support mode.
* →	Moves the input cursor one position right.
* ←	Moves the input cursor one position left.
* ↑	No operation.
* ↓	Repeats the last command entered.

*Use of these keys depends upon operation mode. For more information, refer to the Amdahl 470V/6 Operations Guide.

Commands entered in control mode may start anywhere on the keyboard entry line, providing they are preceded only by spaces. A space is required between a command and its operand(s). Spaces or commas may be used to separate operands. Multiple spaces are automatically compressed to one or none if accompanied by a comma separator. Commands are issued by depressing the ENTER key. The following forms are equivalent:

COMMANDbARG1,ARG2,ARG3 ⓔ

COMMANDbARG1bARG2bARG3 ⓔ

bbCOMMANDbbbARG1b,bARG2bbbARG3 ⓔ

Note: ⓔ denotes the ENTER key

b denotes a space

Multiple commands may be entered together, separating them with a slash (/).

Note that if the left cursor key is used to backspace (say, to correct an error) all information to the right of the cursor key is ignored, even though it still appears on the console display CRT, unless the cursor is repositioned before depressing the ENTER key.

In control mode, CPS divides the CRT screen into three logical areas. Lines 1–5 continuously display a snapshot of the system status. Lines 6–38 display an optional screen presentation selectable by operator command. Finally, lines 39–40 are reserved as the keyboard echo/command response area.

CPS maintains a number of independent CRT screen images, which can be selected for presentation by keyboard command. With the exception of the fixed screen display, which requires the system to be stopped, any of the following screen images can be selected without disturbing background device support mode activities.

- Message Screen

  The message screen is selected via the SE command. CPS maintains a log of all CPS messages. Whenever the message screen is selected, the current message screen is restored so that previously displayed messages will appear again. Then, any new messages in the log are displayed in roll-up fashion and log is reset to accept new messages. The status display will flash MSG when there are messages logged and will go off when the messages are displayed.

- Fixed Screen

  The fixed screen provides a fixed format snapshot of significant operational registers and selected areas of Amdahl 470V/6 memory. The system must be stopped whenever the fixed screen is selected. There are a number of commands that cause the fixed screen display to be presented. The alter and display registers and memory commands cause a partial fixed screen display, with only the applicable sections displayed. Issuing the SF command or depressing the STOP key twice causes a full fixed screen display.

- Scan Screens

  The scan screens display formatted pages of alphanumeric representation of internal system conditions. The system need not be stopped to display scan pages; however, many internal conditions change frequently when running so that the conditions displayed could be inconsistent. There are nine independent sets of scan screens selectable by commands S1, S2, . . . , S9. Each set presents 6 individual 11-line by 40-character pages. The pages in a set are displayed according to the area of the system they present or other logical grouping. CPS also provides a means for arbitrary grouping of pages via the Scan command. By use of the Scan command, any combination of 6 pages can be arranged on any of the 9 scan screens. CPS thereafter associates the new definition with a scan screen selection. For example, by entering

  S1

  SCAN 4,A,20,B,2,C,51,D,59,E,70,F

  CPS will display pages 4, 20, 2, 51, 59 and 70 whenever scan screen 1 is selected via the S1 command. This holds true until a new set of pages is associated with the S1 screen by either another Scan command or a CPS reinitialization.

CONTROL MODE AND DEVICE SUPPORT MODE COMMANDS

Control mode and device support commands, with definitions, syntax, and explanations, are listed alphabetically for easy reference. String commands are described and listed following the control mode and device support mode commands.

The symbols [ ] and { } are used here to define the required syntax. These symbols are not entered; they are a convention used to indicate the requirements of each command. They have the following meanings:

[ ] The operand enclosed in the brackets is optional. If more than one value appears in brackets, one or none of them may be coded. If there is a default value for the operand, it is underlined.

{ } A choice must be made of one of the values for operands appearing in braces.

AC: ALTER CONTROL REGISTER

Syntax:

Command	Operand(s)
AC	register number, data

Explanation: Alters and displays the contents of the designated control register with the specified value. The control register number must be from 0 through F; the data is in hexadecimal form and right adjusted and from 0 through FFFFFFFF. The 470V/6 must be stopped; clocks must be on.

AF: ALTER FLOATING POINT REGISTER

Syntax:

Command	Operand(s)
AF	{0, 2, 4, 6}, data, data

Explanation: Alters and displays the contents of the designated floating point register with the specified values. Both parameters 2 and 3 must be in hexadecimal form, right adjusted and be from 0 through FFFFFFFF. Parameter 2 is the upper 32 bits; parameter 3 is the lower 32 bits. The system must be stopped; clocks must be on.

AG: ALTER GENERAL PURPOSE REGISTER

Syntax:

Command	Operand(s)
AG	register number, data

Explanation: Alters and displays the contents of the designated general purpose register with the specified value. The register number must be from 0 through F. Data must be in hexadecimal form, right adjusted, and from 0 through FFFFFFFF. The system must be stopped; clocks must be on.

AGO: RELEASE ADDRESS STOP

Syntax:

Command	Operand(s)
AGO	none

Explanation: Releases address stop control on the Operating State Registers. See the ASTOP command. The system must be stopped; clocks must be on.

AHSB: HSB INVALID

Syntax:

Command	Operand(s)
AHSB	none

Explanation: The content of the high-speed buffer is marked invalid. Any data in the buffer that is not in main store is lost. The system must be stopped; clocks must be on.

AI: ALTER INSTRUCTION ADDRESS

Syntax:

Command	Operand(s)
AI	address

Explanation: Alters and displays the instruction address field of the PSW. The instruction address must be any even value from 0 through FFFFFE.

AM: ALTER MEMORY (REAL ADDRESS)

Syntax:

Command	Operand(s)
AM	address, data

Explanation: Alters and displays the contents of memory beginning at the designated byte address with the specified data. The memory address may be from 0 to the memory limit. Parameter 2 must be in hexadecimal form, left adjusted and from 0 up to 64 digits. The system must be stopped; clocks must be on. NOTE: Groups of one or more digits of parameter 2 may be separated with spaces for clarity.

AP: ALTER PROGRAM STATUS WORD

Syntax:

Command	Operand(s)
AP	data, data

Explanation: Alters and displays the contents of the program status word with the specified value. Parameters 1 and 2 must be in hexadecimal form, right adjusted and from 0 through FFFFFFFF. Parameter 1 is the upper 32 bits; parameter 2 is the lower 32 bits. The system must be stopped; clocks must be on.

AS: ALTER SCRATCH REGISTER

Syntax:

Command	Operand(s)
AS	register number, data

Explanation: Alters and displays the contents of the designated scratch register with the specified value. Parameter 1 must be from 8 through F. Parameter 2 must be in hexadecimal form, right adjusted, and from 0 through FFFFFFFF. The system must be stopped; clocks must be on.

ASTOP: ADDRESS STOP CONTROL

Syntax:

Command	Operand(s)
ASTOP	address [{ALL, IA, F, ST, CF, CS}]

Explanation: The address stop register is set to the value specified, and the Amdahl 470V/6 will stop when an equal address comparison is made for the appropriate condition. The conditions that must be met can be specified in any combination and are discussed below.

ALL: The address stop encompasses all references to the address.

IA: Instruction address references to the specified address cause a stop.

F: The reference must be an operand fetch for the address stop to take effect.

ST: An operand store is the condition that causes a stop.

CF: Channel fetch causes a stop.

CS: Channel store causes a stop.

When an address stop occurs, the MSG indicator will flash in the status display and the message screen should be selected (SE command) to verify the occurrence of the address stop. The system must be stopped; clocks must be on.

AV: ALTER MEMORY (VIRTUAL ADDRESS)

Syntax:

Command	Operand(s)
AV	address, data

Explanation: Same as for AM except that parameter 1 is a virtual address. The system must be stopped; clocks must be on.

CHK: CHECK SWITCH CONTROL

Syntax:

Command	Operand(s)
CHK	{NORMAL, STOP, DISABLE} [{CPU, ALL, CU}]

Explanation: This command specifies the action to be taken when a machine check occurs. The operands are positional and their meanings are discussed below. This command can be used without stopping the system.

NORMAL: The operating system may recover from machine checks when possible. The occurrence of the machine check may be logged by the operating system and pertinent information recorded.

STOP: All machine checks cause the Amdahl 470V/6 to check stop. No attempt is made to recover from the condition.

DISABLE: All machine checks are ignored in the specified unit.

CPU: The action specified in operand 1 applies only to machine checks in the CPU.

ALL: The action specified in operand 1 applies to all machine checks.

CU: The scope of the action specified in operand 1 is limited to the channel unit.

CLRSW: CLEAR SWITCHES

Syntax:

Command	Operand(s)
CLRSW	none

Explanation: This command is used to return all process conditions to normal. It is equivalent to issuing the following commands in the order given:

RATE PROC

YES OVLP, RTRY, ECC, SOFT, PROT, MS,
MSERR, SUERR, PRTY, TMR, BYP,
SBR, PF, RPL, MOD, CTMR

CHK NORM, ALL

DAT NO

ERR

RPL NORM

AGO

HSB NORM

TLB NORM

The system must be stopped; clocks must be on.

DA: DISPLAY ACTIVE AND GATED CCI

Syntax:

Command	Operand(s)
DA	none

Explanation: Displays the active and gated states of the CCI on the fixed screen.

DCHN: DISPLAY CHANNEL BUFFER

Syntax:

Command	Operand(s)
DCHN	channel/device address

Explanation: Displays the contents of the channel buffer for channel/device. Parameter 1 must consist of CUU where C equals the channel and UU equals the device. The system must be stopped; clocks must be on.

DLOG: DISPLAY MACHINE CHECK LOG KEYS

Syntax:

Command	Operand(s)
DLOG	none

Explanation: Displays the machine check log keys. The DSP subsystem must be active.

DM: DISPLAY MEMORY (REAL ADDRESS)

Syntax:

Command	Operand(s)
DM	[address]

Explanation: This command displays 64 bytes of memory beginning at the designated byte address. The memory address must be from 0 to the memory limit; if no address is specified, the last byte address displayed is incremented by one to generate an address for the current request. If the current request is also the first request and no address is specified, memory locations beginning at address 0 are displayed.

The system must be stopped; clocks must be on.

DM2: DISPLAY MEMORY (REAL ADDRESS)

Syntax:

Command	Operand(s)
DM2	[address]

Explanation: This command displays 32 bytes of data, according to the address conditions as specified for DM. The DM2 command is used in conjunction with the DM command to display disjunctive areas of main storage. The 32 bytes displayed by this command overlay the second 32 bytes of the 64 displayed by the DM command.

DM3: DISPLAY MEMORY (REAL ADDRESS)

Syntax:

Command	Operand(s)
DM3	[address]

Explanation: This command displays 256 bytes of data, beginning at the real hexadecimal address specified. If no address is specified, the address for the last byte displayed is incremented by one to generate an address for the current request. If the current request is also the first request and no address is specified, memory locations beginning at address 0 are displayed.

DO: DISPLAY OPERATING STATE REGISTERS

Syntax:

Command	Operand(s)
DO	none

Explanation: Displays the contents of certain of the operating state registers.

DR: DISPLAY REGION CODE ERROR REGISTER

Syntax:

Command	Operand(s)
DR	none

Explanation: Displays the contents of the region code error register. The system must be stopped; clocks must be on.

DSP: DEVICE SUPPORT SYSTEM

Syntax:

Command	Operand(s)
DSP	[3066 / 3215] [, NS]

Explanation: The Device Support System is activated. The operands and their meanings are:

* 3066: 3066 support is activated; this is default.

* 3215: 3215 support is activated, rolling lines in the CRT from the bottom of the screen.

NS: Do not start system.

HSB: HSB CONFIGURATION

Syntax:

Command	Operand(s)
HSB	{PRI / ALT / BOTH / NORM} {BOTH / LO / HI}

Explanation: HSB configuration. The operands and their meanings are as follows:

PRI: Use primary only.

ALT: Use alternate only.

BOTH: Use both.

NORM: Use normal configuration.

BOTH: Use both hi and low half.

LO: Use lo half only.

HI: Use hi half only.

* If no device support mode is specified, a system default is used.

If the system is not stopped; clocks must be on.

IPL: INITIAL PROGRAM LOAD

Syntax:

Command	Operand(s)
IPL	[[unit address] [, [CLEAR] [, [3066 / 3215 / MCH] [, NS]]]]

Explanation: This command initiates an initial program load. It is an alternative to depressing the LOAD button and functions in the same way. Unlike the LOAD button, this command provides the option of overriding IPL defaults. The IPL command will cause the CPU to resume execution after a stop, but may be issued without stopping the system. The parameters used with this command are positional and their meanings are discussed below.

unit address: Hexadecimal channel/device address of IPL device; default to the value in the console switches if omitted.

CLEAR: Main store is cleared; no clear is default.

* 3066: 3066 mode.

* 3215: 3215 mode.

* MCH: The machine check handler interface is initialized, but the console remains in control mode.

NS: System is not started after IPL; start is default.

KEY: ASSIGN KEY TO LOG

Syntax:

Command	Operand(s)
KEY	high, low, [log no.]

Explanation: Causes a key to be assigned to the current, valid unkeyed log. Parameters are positional and their meanings are discussed below. The DSP

subsystem must be active.

high: High order 32 bits of key

low: Low order 32 bits of key

log number: Log number which is to be keyed

LOGK: EXTENDED LOGOUT

Syntax:

Command	Operand(s)
LOGK	high, low, [log number]

Explanation: Causes an extended logout and keys it with the value specified. Parameters are positional and are as described below. The DSP subsystem must be active to execute this command.

high: High order 32 bits of key

low: Low order 32 bits of key

log number: Specifies which log position (1–6) the logout is to occupy. If none specified, the next available log position is used.

MCH: MACHINE CHECK HANDLER

Syntax:

Command	Operand(s)
MCH	[NS]

Explanation: Initializes the machine check handler interface. The NS operand overrides automatic starting on entry.

NO/YES: DISABLE/ENABLE

Syntax:

Command	Operand(s)
NO YES	function

Explanation: The NO command will disable a processor function which is otherwise enabled by system defaults; if the system defaults do not allow the function, the NO command is not effective. The YES command reverses the effect of a previous NO command if any. Any number of the listed arguments may be entered with a single YES/NO command. Order is unimportant.

RTRY: Instruction retry

ECC: Error checking and correcting

SOFT: Soft machine checks

PROT: Main store protection checks

MS: Main store fetching

MSERR: Main store error checking

SUERR: Storage unit error checking

PRTY: Storage unit priority

TMR: Interval timer

BYP: Buffer

SBR: Segment buffer register

PF: Prefetch

RPL: HSB replace on valid and hot/cold only

MOD: Mark all HSB lines as modified

CTMR: Console timeout facility

OVLP: Instruction overlap

The system must be stopped; clocks must be on.

PHSB:

Syntax:

Command	Operand(s)
PHSB	none

Explanation: The contents of the high-speed buffer are written back to main store and all locations marked invalid. No data is lost. The system must be stopped; clocks must be on.

PLOG: PURGE ALL LOGS

Syntax:

Command	Operand(s)
PLOG	none

Explanation: Purges all machine check logs currently retained by the DSP subsystem. The DSP subsystem must be active.

PROC: RESTART CPU OR C-UNIT

Syntax:

Command	Operand(s)
PROC	{CPU / CU}

Explanation: Restarts CPU or C–Unit after an error stop. Operand 1 meanings are as follows:

CPU: CPU clocks are started.

CU: Channel Unit clocks are started.

PSBR: CLEAR STO STACK

Syntax:

Command	Operand(s)
PSBR	none

Explanation: All segment table origin (STO) stack entries are invalidated by this command. The STO stack enables the translation lookaside buffer (TLB) entries associated with a given set of values in control registers 0 and 1 to be reused if these registers are reloaded with the same values. If previous values are restored to control registers 0 and 1, any entries remaining in the TLB from old translations need not be made again. This command, therefore, invalidates the STO stack and implicitly the TLB. The system must be stopped; clocks must be on.

PTLB: PURGE TRANSLATION LOOKASIDE BUFFER

Syntax:

Command	Operand(s)
PTLB	none

Explanation: This causes all valid bits in the translation lookaside buffer (TLB) to be turned off, thus invalidating all entries. As a result, until valid entries are made in the TLB, all main storage references will go through the complete translation process before real addresses can be determined. PTLB also includes the functions of PSBR command.

RATE: SET RATE

Syntax:

Command	Operand(s)
RATE	{I / C / B / P} [,nnn] [,START]

Explanation: Establishes the instruction processing mode of the CPU and displays status of the rate controls. The operands and their meanings are as follows:

I: When started, nnn instructions are executed.

C: The CPU performs nnn machine cycle of processing.

B: The CPU performs nnn cycles or one instruction, whichever occurs first.

P: Instruction execution proceeds at processor speed, which is the normal mode of operation.

nnn: If parameter 1 equals I, the number of instructions. If parameter 1 equals C or B, the number of cycles. If parameter 1 equals P, not applicable.

START: If present, a start processing is issued after set up.

RPL: HSB REPLACEMENT

Syntax:

Command	Operand(s)
RPL	{PRI / ALT / NORM}

Explanation: Forces high-speed buffer replacement according to parameter 1, which meanings follow:

PRI: Force replacement on primary.

ALT: Force replacement on alternate.

NORM: Use normal replacement.

The system must be stopped; clocks must be on.

RPSW: PSW RESTART

Syntax:

Command	Operand(s)
RPSW	none

Explanation: This command causes the current PSW to be stored at absolute location 8 and a restart interrupt to be generated. The restart interrupt loads the new PSW from absolute location 0. The system need not be stopped to issue this command.

RS: RESET

Syntax:

Command	Operand(s)
RS	{IU, CU, SU, CHK, CPU, SYS} [, {INITIAL, CLEAR}]

Explanation: Performs CPU reset to an initialized state according to values given. The operands and their meanings are as follows:

IU: Instruction unit only

CU: Channel unit(s) only

SU: Storage unit only

CHK: I and S checks reset

CPU: Equivalent to IU, SU, CHK.

SYS: Equivalent to CPU, CU

INITIAL: Initialize all registers, configure main store. Valid only with SYS or CPU.

CLEAR: Clear main store (valid only with SYS)

S1 – S9: SCAN SCREEN

Syntax:

Command	Operand(s)
S1 – S9	none

Explanation: Keyboard commands to select one of nine independent sets of scan screens. A scan screen, as defined by the CPS default or by a previous Scan command, is formatted and displayed once if Redisplay is disabled or continuously if Redisplay is enabled (see SN and SR, below).

SA: START CPU

Syntax:

Command	Operand(s)
SA	none

Explanation: The CPU is started.

SCAN: SELECT SCAN SCREEN

Syntax:

Command	Operand(s)
SCAN	$\{P_1, A_1, P_2, A_2, \ldots P_n, A_n\}$

Explanation: This command arranges 1 to 6 scan pages on the currently selected scan screen (or on scan screen 1 if none currently selected).

P_x: page number x is to be displayed

A_y: scan screen address y is area of screen where page x is to be displayed, according to the following scheme:

A	B
C	D
E	F

SE: ERROR SCREEN

Syntax:

Command	Operand(s)
SE	none

Explanation: A keyboard command to select the message screen. The last displayed message screen is displayed and any pending error messages are rolled in from the bottom, with the oldest messages rolling off the top. Any new messages that occur while the message screen is selected are automatically displayed.

SF: FIXED SCREEN

Syntax:

Command	Operand(s)
SF	none

Explanation: A keyboard command to select the fixed screen. The system is stopped, and the contents of its operational registers and the area of memory that was specified in the last entered Display Memory (Real) command are displayed in a full fixed screen display.

SLOG: SCAN MACHINE CHECK LOG COMMANDS

Syntax:

Command	Operand(s)
SLOG	machine log 1 – 7

Explanation: Causes all scan commands to scan from the machine check log specified by P1. P1 = machine log 1 – 7. 0 = scan from the 470 (normal scan). The DSP subsystem must be active to scan from logs or execute this command.

SN: DISABLE REDISPLAY

Syntax:

Command	Operand(s)
SN	none

Explanation: Causes once only display of scan screens 1 – 9 whenever selected.

SR: ENABLE REDISPLAY

Syntax:

Command	Operand(s)
SR	none

Explanation: Causes continuous refresh of scan screens 1 – 9 whenever selected.

STSTS: STORE OPERATING REGISTERS

Syntax:

Command	Operand(s)
STSTS	none

Explanation: Causes CPU operating registers to be stored in permanently assigned locations on the base page of memory and causes CPU to resume execution after a stop. The following table itemizes the result:

Field	Address	Length
CPU Timer	216	8
Clock Comparator	224	8
Current PSW	256	8
Prefix	264	4
Floating-Point Registers	352	32
General-Purpose Registers	384	64
Control Registers	448	64

Clocks must be on.

SU

Syntax:

Command	Operand(s)
SU	none

A keyboard command to select device support mode screen. The last device support mode screen that was active is retrieved from disk and the screen is given to the active user.

TLB: TLB CONFIGURATION

Syntax:

Command	Operand(s)
TLB	{ NORM | PRI | ALT }

Explanation: Establishes translation lookaside buffer configuration. The system must be stopped; clocks must be on. Operand meanings are:

NORM: Use normal configuration

PRI: Use primary only.

ALT: Use alternate only.

YES

See NO/YES command. Reverse the effect of a previous NO command, if any.

STRINGS

CPS allows the operator to put multiple commands on a command line by separating the individual commands with slashes. The operator can also store away command lines for later execution; these are called strings. The following command set defines the legal string operations; initialization (CSTR),

definition (STR), display (DSTR), and execution (GOTO and EXIT).

Strings are stored on the console processor disk as strings of characters up to 63 characters long. The strings are defined by execution of the STR command, which inserts a string into the proper place on disk according to the string number. String numbers then identify the string and order the strings.

Strings are executed by the GOTO command, which causes a string of commands to execute with one of the following results:

The commands in a string are all executed so the next highest numbered defined string is then executed.

A GOTO command is encountered in the string, which causes a different string to begin execution.

An EXIT command is encountered, which causes string execution to stop.

If strings are executing in a continuous loop, execution can be stopped by depressing the REQUEST key and entering the word EXIT.

CPS contains 26 variables labeled A through Z. The variables may be used where parameters are normally used in CPS commands. The variables are assigned values via the LET command, which allows assignment of constants or arithmetic expressions of variables or constants to a variable. The form of an expression is:

⟨VAR1⟩ ⟨OPERATOR⟩ ⟨VAR2⟩ ⟨CONSTANT⟩

The rules for evaluation are as follows:

If ⟨VAR1⟩ is defaulted then ⟨VAR1⟩ and ⟨OPERATOR⟩ are ignored; otherwise, ⟨VAR1⟩ and ⟨OPERATOR⟩ form the left half of the expression.

If ⟨VAR2⟩ is defaulted then the right half of the expression is ⟨CONSTANT⟩ . ⟨CONSTANT⟩ defaults to 0.

If ⟨VAR2⟩ is present then ⟨CONSTANT⟩ is ignored.

Division by 0 returns an error.

The LET command allows assignment of an expression to a variable. The IF command allows conditional execution of a string based on comparison of variables.

String operations are defined below:

CSTR

Clear the string area. No parameters. This operation clears the string table. This must be done as part of the CPS installation procedure.

DSTR ⟨STRING #⟩

Display the strings. STRING #can be any hexadecimal number from 0_{16} to 77_{16}. If STRING # is not entered in the command line, then the default, which is 0_{16}will be used.

EXIT

Stop string execution.

GOTO ⟨STRING #⟩

Execute a string. STRING # is optional and has a default value of 0. GOTO allows the operator to initiate execution of strings at any specified string as indicated by the STRING # . The specified string in a GOTO command must not be a null string or an error will result.

STR ⟨STRING #⟩ ⟨STRING⟩

Define a string. STRING # is the same as above except that it is a required parameter. STRING can be any character string from the following character set:

A – Z

0 – 9

! " # $ % & ' ( ) * + , – . /

: ; ⟨ = ⟩ ? @ [ ] ∿

If a null string is entered, the string with the specified STRING # is deleted. The system has a capacity of 60 strings, each up to 63 characters long.

2-Byte Interface 5
3066 Emulation 21, 22
3215 Emulation 22

Address Space 13
Architecture 5, 11, 23
Arithmetic Checking 26

Block Multiplexer 5, 17, 38
Branch 8
Branch Instruction 7
Buffer 5, 7, 10, 11, 12, 13
Byte Multiplexer 5, 17, 38

C-Unit 12, 15, 25, 26
CBS 15
Central Interface Control Logic 15
Central Processor 5
Channel 5, 15, 17, 19, 37, 38, 42, 44
Channel Assignment 15
Channel Buffer Store 15, 44
Channel Indirect Data Addressing 17
Channel-To-Channel Adapter 5
CICL 15, 17
Clock Comparator 5
Codes 8
Command Mode 52
Computer-To-Console Interface 19
Condition Code Setting 30
Console 5, 21, 28
Console Processor 5, 19, 20, 26, 49, 50, 51
Console Processor System 49
Control Logic 15
Control Mode 50, 52, 55, 56
Control Panel 5
CPS 49, 50, 51, 52, 54, 55, 56
CPU 26
CPU Logout Areas 24
CPU Timer 5
CRT 5, 19, 21, 22, 49, 50, 52, 54, 55, 56
Cursor 21

DACL 15, 17
Data Access Control Logic 15
Device Address Assignment 43
Device Support Mode 19, 50, 56
Diagnose 26
Diagnostic Mode 28
Direct Control 5
Dynamic Address Translation 11
Dynamic Priority 15

E-Unit 7, 8, 9, 25
Early Condition Code 8, 9
EC Mode 11
ECC 10, 26
Emergency Power Off 5
Error Checking and Correction 26
Execution Unit 8
Extended Channel Logout 17
Extended Logout 44

Fixed Logout Area 24
Fixed Screen 56

Hard Machine Check 26, 27
Hardware Command Mode 19, 28
High-Speed Buffer 5, 7, 10, 11, 39, 40
HSB 10, 12, 13

I-Unit 7, 8, 9, 25
IDA 17
Indirect Data Addressing 17
Initial Program Load 19, 51
Instruction Retry 8
Instruction Set 5
Instruction Unit 7
Interleaving 5, 11
Interrupt Handling 8
Interval Timer 5
IOEL 44, 46
IPL 51

Keyboard 5, 19, 21, 49, 50, 51, 52, 55

LCL 44, 46
Limited Channel Logout 44
Line 10
Logical Unit and Checker 9
Logout 26
LUCK 9, 25

Machine Check Extended Logout 26, 27
Machine Check Interrupt Code 24, 27
Machine Check Logout 24
Machine Checks 26
Main Storage 5, 10, 11, 12, 13, 25, 39
Maintenance Mode 19, 28, 50
MCEL 26
MCIC 24, 27
Message Screen 56
Meters 20
Model-Dependent Data 23
Multiplexer 17, 42, 44

OCL 15, 17
Operations 15
Operator's Control Panel 19, 51
Operator's Console 5, 19, 21, 28, 49, 50

Parity Checking 26
PDU 5
Pipeline 8, 25, 27
Power Distribution Unit 5
Power Down 50
Power Up 50
Processing Damage 26
Protection Key 26

Region Code 24, 25
Remote Interface Logic 15
RIL 15

S-Unit 7, 8, 10, 11, 12, 13, 15, 17, 25, 26
SBS 17
Scan Screens 56
Scan-Out 19
SCS 15
Segment Table Origin 11

Selector 5, 17, 44
Selector Subchannels 17
Shared Subchannel 42, 43
Shared Subchannel Operation 17
Shifting Channel State 15
Soft Machine Check 26, 27
SSS 17, 44
Standard Channel Interface 19
Status Display 52
STO Stack 11, 13, 26
Storage Control Unit 10
Subchannel 5, 17, 38, 42, 43

Subchannel Assignment 42
Subchannel Buffer Store 17
Subchannel State Store 17, 44
System Console 5, 19, 20, 21, 22, 52
System Maintenance Console 19
System Meter 20
System Recovery Conditions 26

Thermal Monitoring Facilities 5
Time-of-Day Clock 5
TLB 11, 12, 13, 25
TOD 5, 40
Translation Lookaside Buffer 5, 11

Universal 5
Usage Meter 20

Virtual Memory 11

READER'S COMMENT FORM

AMDAHL 470V/6
Machine Reference Manual
Form MRM 1000–1

Name _____

Address _____

Job Title _____

Possible topics for comment are:

Accuracy Clarity Completeness Examples Figures Index Legibility Organization

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

YOUR COMMENTS, PLEASE.....

This publication is one of a series describing the AMDAHL 470V/6. Your comments on the reverse side will be carefully reviewed and will enable us to produce better publications.

If you require an immediate reply, please contact your Amdahl representative.

Fold and staple

First Class Permit
Number 741
Sunnyvale, CA

BUSINESS REPLY MAIL

No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

AMDAHL CORPORATION
1250 EAST ARQUES AVENUE
SUNNYVALE, CA 94086

Attention: AMDAHL Publications

Fold and staple

amdahl

AMDAHL CORPORATION
1250 EAST ARQUES AVENUE
SUNNYVALE, CALIFORNIA, 94086

PHONE 408-735-4011
TWX 910-339-9293