amdahl

amdahl 470V/7

MACHINE REFERENCE MANUAL

amdahl®

amdahl 470V/7
MACHINE REFERENCE MANUAL

Publication No. G1003.0-01A

June 15, 1978

REVISION NOTICE

This publication is an original issue.

ABSTRACT

This manual describes the functional characteristics and model-dependent features of the Amdahl 470 V/7 computing system. It is intended for managers, system analysts, and programmers.

The topics covered include machine organization and configuration, operation of each unit, channel characteristics, subchannel assignment, machine-check conditions, and model-dependent instructions.

READER COMMENT FORM

A reader comment form is provided at the end of this publication. If this form is not available, comments and suggestions may be sent to Amdahl Corporation, Technical Publications Department, P.O. Box 5070, Sunnyvale, CA, 94086, U.S.A. All comments and suggestions become the property of Amdahl Corporation.

All specifications subject to change without notice.

CONTENTS

INTRODUCTION

The Amdahl 470V/7 computing system provides powerful, high-speed, general-purpose computing capabilities for sophisticated business and scientific applications. It has a pipeline that executes several instructions concurrently, a high-speed buffer for fast data access, and efficient execution algorithms. The 470V/7 also incorporates extensive error checking and recovery to optimize system reliability. The channels provided with the 470V/7 may be configured in any combination of selector, byte-multiplexer, or block-multiplexer channels.

The central processor and the channel logic are implemented by high-speed, large-scale-integration (LSI) circuits. Up to 100 of these circuits can be packed into a single chip. Up to forty-two chips fit into each 7.5-inch square multi-chip carrier (MCC). The central processor, storage control and channel logic together require only 59 MCCs. Because of this simplicity, the number of external connections in the system is small, and the system is consequently easy to service and maintain.

Reliability of the 470V/7 is enhanced by such features as hardware instruction retry, channel command retry, improved storage, error-correction and isolation hardware, including enhanced main-storage error checking and correction (ECC). ECC is capable of correcting any single-bit error and detecting any double-bit error.

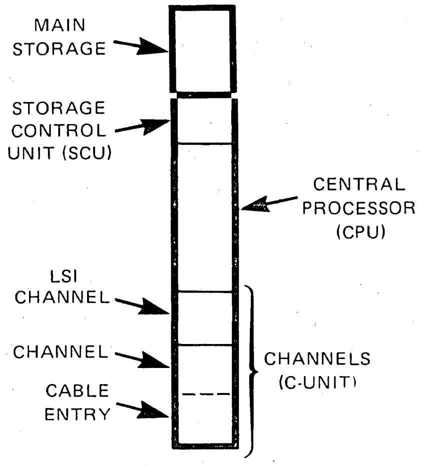
The 470V/7 console can determine and report the status of approximately 17,000 latches in the system. This information can be displayed at the console or preserved in extended logouts of error conditions. The machine can be reconfigured from the console, removing failing components from the system and leaving the remainder of the system operable.

The Amdahl 470V/7 system and the IBM System/370 are compatible within the constraints of the architecture defined in the ***System/370 Principles of Operation, GA22-7000, revision level 5*** (hereafter to be referred to as *"System/370 Principles of Operation"*). This specification requires machine compatibility in all but the following cases:

> Programs relying on model-dependent data such as the contents of logout areas
>
> Time-dependent programs that rely on instruction of CCW execution times
>
> Programs that cause deliberate machine checks.

The Amdahl 470V/7 system has four areas of model dependence: machine-check logouts, channel logouts, machine-check conditions, and the implementation of architecturally defined model-dependent instructions. These are all discussed later in this manual.

470V/7 SYSTEM



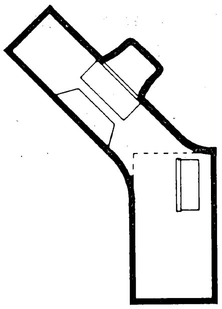
STANDARD FEATURES

SYSTEM/370 UNIVERSAL INSTRUCTION SET	SYSTEM/370 INSTRUCTION ENHANCEMENTS	BYTE-ORIENTED OPERANDS	32,768 BYTE HIGH SPEED BUFFER
CPU TIMER	TOD CLOCK	INTERVAL TIMER	CLOCK COMPARATOR
EXTENDED CONTROL	DYNAMIC ADDRESS TRANSLATION	PROGRAM EVENT RECORDING	DIRECT CONTROL
12 I/O CHANNELS	BYTE/BLOCK/SELECTOR	2048 MULTIPLEXER SUBCHANNELS	INDIRECT DATA ADDRESSING

OPTIONAL FEATURES

CHANNEL-TO-CHANNEL ADAPTER	2-BYTE INTERFACE 4 EXTRA I/O CHANNELS

CONSOLE



FEATURES

OPERATOR'S CONSOLE	3066 EMULATION	3215 EMULATION
SYSTEM CONSOLE WITH CRT DISPLAY	INDEPENDENT CONSOLE PROCESSOR	

POWER DISTRIBUTION UNIT

FEATURES

EMERGENCY POWER-OFF CONTROL	415 Hz and 60 Hz POWER DISTRIBUTION TO SYSTEM

FIGURE 1 470V/7 SYSTEM (8 MEGABYTE CONFIGURATION)

A524

SYSTEM OVERVIEW

CENTRAL PROCESSOR (CPU)

The Amdahl 470V/7 central processor (CPU) comprises three units: the Instruction Unit, the Execution Unit, and the Storage Unit (see figure 1). It includes these standard features:

STANDARD ARCHITECTURE. The Amdahl 470 V/7 follows standard System/370 architecture as specified in the *IBM System/370 Principles of Operation.* The standard, full System/370 Universal Instruction Set with extended-precision floating-point operations and System/370 instruction enhancements is implemented on the Amdahl 470 V/7. Direct control is also implemented.

INSTRUCTION PIPELINE. The 470V/7 Instruction Pipeline allows the CPU to process several instructions simultaneously and reduces the cycles lost in a program branch to three.

HIGH-SPEED BUFFER. The High-Speed Buffer (HSB) is a cache memory designed to maximize system throughput. It provides fast access to frequently used data.

TRANSLATION LOOKASIDE BUFFER. The 512-entry Translation Lookaside Buffer (TLB) provides high-speed storage of frequently used virtual address translations. A segment table origin stack, which associates a specific CPU state with each TLB entry, further enhances virtual address translation in the 470V/7.

TIMING FACILITIES. Standard System/370 timing facilities are provided. These include an interval timer, a time-of-day clock with 52 bit resolution, a 52-bit clock comparator, and a CPU timer.

SYSTEM CONSOLE

The 470V/7 system console not only acts as an operator's console but serves as an independent maintenance tool as well. It includes an operator's control panel, a keyboard and CRT display, and an independent console processor.

MAIN STORAGE

Main storage is available in configurations of 4, 6, 8, 12, and 16 megabytes. Main storage is interleaved 16 ways on a doubleword basis. Error checking and correction (ECC) corrects single bit errors and detects double-bit errors on a doubleword basis. Failing portions of main store can be configured out of the system in two-megabyte blocks. Access to main storage is controlled by the Storage Control Unit.

CHANNELS (C-UNIT)

The Amdahl 470V/7 system has physically inboard channels, with 12 standard and 4 more channels available as an option. These may all be installed in any combination of selector, byte-multiplexer, or block-multiplexer channels. The channels are implemented by the Channel Unit, and except for possible storage-access conflicts, they operate independently of the CPU. A total of 2048 subchannels may be assigned to the multiplexer channels in multiples of 32, up to 256.

POWER DISTRIBUTION UNIT

The power distribution unit distributes 415 Hz power to the 470V/7 system and provides emergency power-off and thermal monitoring. It also provides 60 Hz power for standard utility plugs. Where specified, machines may be manufactured for 50 Hz power.

OPTIONAL FEATURES

CHANNEL-TO-CHANNEL ADAPTER. This option provides the synchronization necessary to interconnect two channels. It may be attached to a selector or a block-multiplexer channel and uses one control unit position on each channel. When interconnecting an Amdahl 470V/7 system with another system, either may be equipped with the channel-to-channel adapter.

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

EXTRA CHANNELS. Four optional channels are available in addition to the 12 standard channels. All features and characteristics of the standard channels also apply to the optional channels.

INSTRUCTION UNIT

I-UNIT FUNCTIONS

The Instruction Unit (I-Unit) executes the instruction stream, updates the CPU timer, and processes interrupts and machine checks. It also contains the general-purpose registers, floating-point registers, control registers, and PSW.

To execute instructions, the I-Unit uses the facilities of the other 470 components. The E-Unit performs arithmetic and logical operations; the C-Unit performs input and output operations; the S-Unit writes and retrieves data and instructions in main storage. Because it controls the flow of instructions, the I-Unit directly or indirectly initiates the operations of all other units.

I-UNIT ORGANIZATION

The I-Unit functions (figure 2) can be summarized into the following major categories:

- Interrupt priority resolution and operation selection
- Instruction fetch and instruction buffer register control
- Instruction path selection in the case of branches or interrupts
- Pipeline processing of the instruction stream

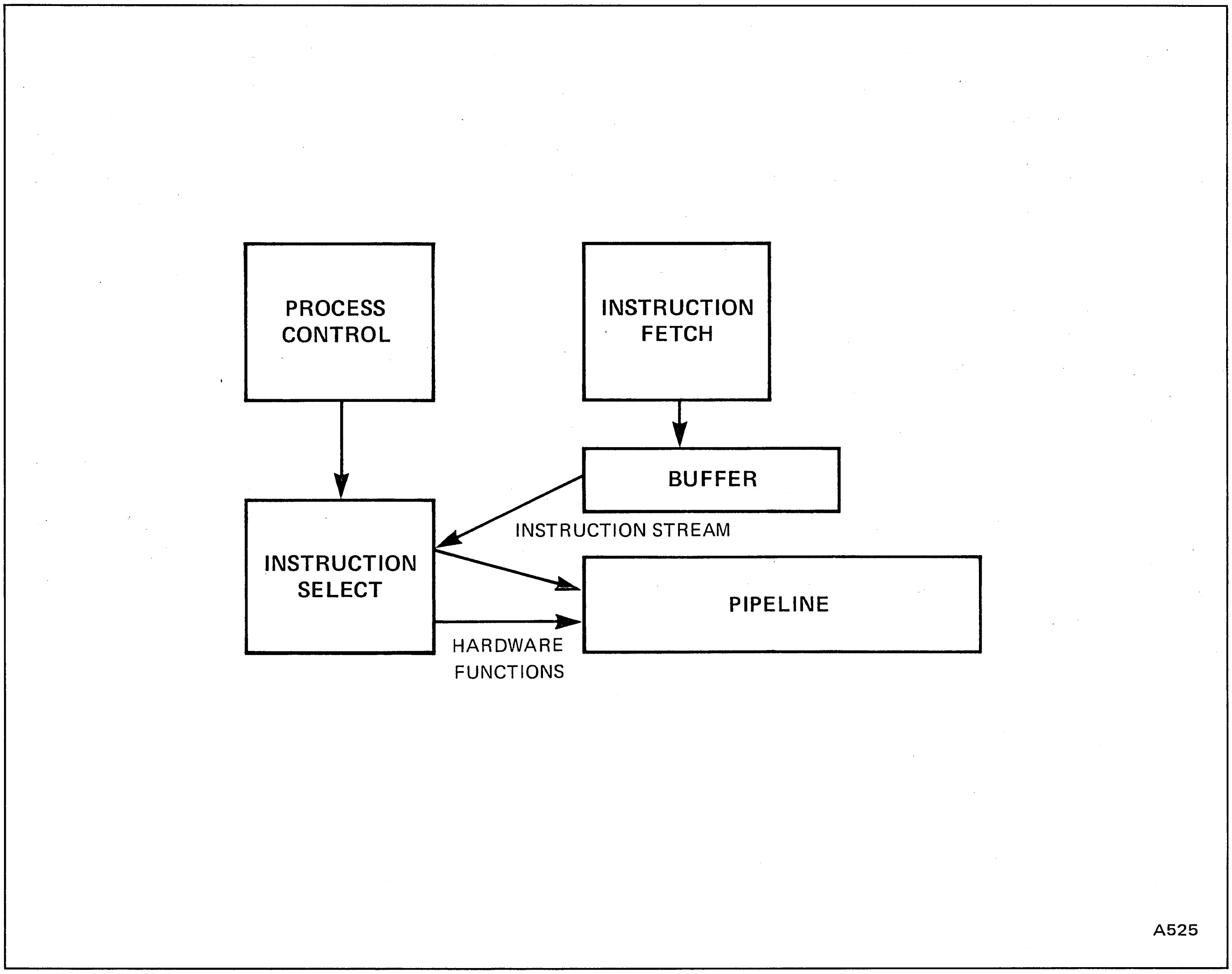


FIGURE 2 I-UNIT ORGANIZATION

The pipeline is a major factor in the high performance of the Amdahl 470V/7. It decodes instructions, reads general purpose registers (GPRs), computes operand addresses, requests operands, initiates operand modification, and checks and writes results. Modifying operands requires the facilities of the E-Unit. Fetching and storing operands requires the facilities of the S-Unit. By overlapping all these described functions, a large performance enhancement is achieved over non-pipelined processors. The purpose of the pipeline is to allow architectural instructions to proceed at the maximum speed of the execution hardware, rather than having to wait for auxiliary functions to be completed. Thus, the pipeline typically begins executing each instruction before it has finished executing the previous one. Because of this, the 470V/7 pipeline will often be processing several instructions simultaneously. Figures 3 and 4 illustrate the concept of pipeline overlap.

One of the problems inherent in pipeline design relates to the handling of branch instructions, or those conditions where prefetching and overlap of instructions is impossible because of multiple possible execution paths. Rather than resort to an expensive and massive duplication of hardware to follow multiple branch paths, Amdahl invented and implemented a unique, fast branch-resolution algorithm. The hardware to accomplish this function is primarily in the E-Unit, which tells the I-Unit the branch condition codes before the subject instruction has even completed execution. Thus the I-Unit is able to pick the correct branch path immediately, and the correct instruction stream proceeds down the pipeline with only a small "hole" (lost time) of three cycles. Even this relatively small degradation can be eliminated if the branch is not taken. Thus the 470V/7 branching algorithms complement an optimal pipeline organization to produce significant performance in the execution of the machine object-instructions.

HARDWARE INSTRUCTION RETRY

To enhance total system availability and reliability, almost all 470V/7 functions are retriable. This is accomplished within the total system design of the 470V/7 Instruction Unit by delaying any updates to architectural registers until the last cycle of instruction execution. Thus, the pipeline concept is extended to include enhancements to machine integrity, so that any errors that occur before registers are updated simply cause re-execution of the instruction. This method of instruction retry minimizes the hardware involved in error detection and therefore increases the effectiveness of the overall 470V/7 checking and correction mechanisms, while providing maximum recovery capability.

INTERRUPT HANDLING

All interrupts in the 470V/7 are precise. When an interrupt occurs, the I-Unit inserts an interrupt-handling routine into the pipeline. This mechanism provides for optimal status switching time, while preserving total system integrity. The interrupted instructon stream can be reinitiated by the I-Unit in the usual manner.

OPERATION	CYCLE	DESCRIPTION
COMPUTE INSTRUCTION ADDRESS	I	Request next sequential instruction from S-Unit
START BUFFER	B1	Start HSB in S-Unit
READ BUFFER	B2	Read instruction from HSB into I-Unit buffer
DECODE INSTRUCTION	D	Dispatch and decode instruction
READ GPR'S	R	Read base and index registers
COMPUTE OPERAND ADDRESS	A	Compute operand address in S-Unit
START BUFFER	B1	Start HSB in S-Unit to retrieve operand
READ BUFFER	B2	Read operand from HSB; access register operands
EXECUTE (ONE)	E1	Pass data to E-Unit; begin execution (LUCK)
EXECUTE (TWO)	E2	Complete execution in E-Unit
CHECK RESULT	C	Check E-Unit result for parity
WRITE RESULT	W	Write result to register

FIGURE 3 I-UNIT INSTRUCTION SEQUENCE A549

INSTRUCTION SEQUENCE	1	D	R	A	B1	B2	E1	E2	C	W				
	2			D	R	A	B2	B2	E1	E2	C	W		
	3					D	R	A	B1	B2	E1	E2	C	W
	4							D	R	A	B1	B2	E1	E2
	5									D	R	A	B1	B2
	6											D	R	A
		1	2	3	4	5	6	7	8	9	10	11	12	13
		CYCLES												

FIGURE 4 PIPELINE OVERLAP A550

EXECUTION UNIT

E-UNIT FUNCTIONS

The Execution Unit (E-Unit) performs all logical and arithmetic operations. It also sets condition codes and checks for errors.

E-UNIT ORGANIZATION

The E-Unit is divided into six subunits: Logical Unit and Checker, Adder, High-Speed Multiplier, Shifter, Byte Mover, and Table Lookup Unit.

LOGICAL UNIT AND CHECKER. The Logical Unit and Checker (LUCK) performs these functions:

- Logical operations: AND, OR, Exclusive-OR.
- Compares operands.
- Sets early condition codes: returns the condition code after one cycle for many operations.
- Checks parity of input, predicts parity of result.
- Checks decimal input for valid digits and sign.
- Counts leading zeros for normalization operations.
- Moves input data to E-Unit internal registers.

ADDER. The adder performs standard binary and decimal addition. It can add two single-word operands per cycle.

HIGH-SPEED MULTIPLIER. The multiplier multiplies an 8-bit multiplier with a 32-bit multiplicand and produces a 40-bit result every cycle.

SHIFTER. The shifter performs shift operations. A maximum of 68 bits can be input to the shifter. The operand can be shifted left or right, from 0 to 63 bit positions.

BYTE MOVER. The byte mover manipulates single-byte fields for such operations as EDIT, EDIT AND MARK, TRANSLATE, and TRANSLATE AND TEST.

TABLE LOOKUP UNIT. The Table Lookup Unit finds reciprocals of operands. These are used in division operations.

INSTRUCTION EXECUTION

The I-Unit presents instructions to the E-Unit and also provides intermediate scratch space for complex operations. The E-Unit accepts instructions at a maximum rate of one every two cycles. Data comes to the E-Unit from either the I-Unit or the S-Unit.

The E-Unit begins each instruction in the LUCK. The LUCK performs the appropriate functions, and, if possible, sets an early condition code in the first cycle of E-Unit execution. When it is finished, the LUCK moves the input data into four internal registers. The operands are now available to the adder, multiplier, shifter, or byte mover.

After the appropriate arithmetic is complete, the result is placed in the result register, where it is available to the I-Unit.

MULTIPLICATION

The multiplier multiplies a full-word first operand by one byte of the second operand and repeats this operation until each byte of the second operand has been used. Each iteration requires one cycle. At the end of the operation, the final result is placed into the result register. Refer to figure 5.

DIVISION

The 470V/7 performs division by multiplying the dividend by the reciprocal of the divisor. The Table Lookup Unit finds the inverse of the divisor and places it into the I-register. The multiplier then uses the inverse as an operand.

CONDITION CODES

The LUCK can set an early condition code for most operations that set a condition code. However, some operations are so complex that the condition code cannot be set until the operation is complete. In this case, the I-Unit branch handling waits for the E-Unit to finish. For some other operations, the E-Unit can set the condition code in the middle of the operation. In this case, the E-Unit signals the I-Unit when the condition code will be set.

ERROR CHECKING

All execution results are checked in the 470V/7. This checking includes parity checks for most operations, and a more comprehensive residue arithmetic check for multiplication and division.

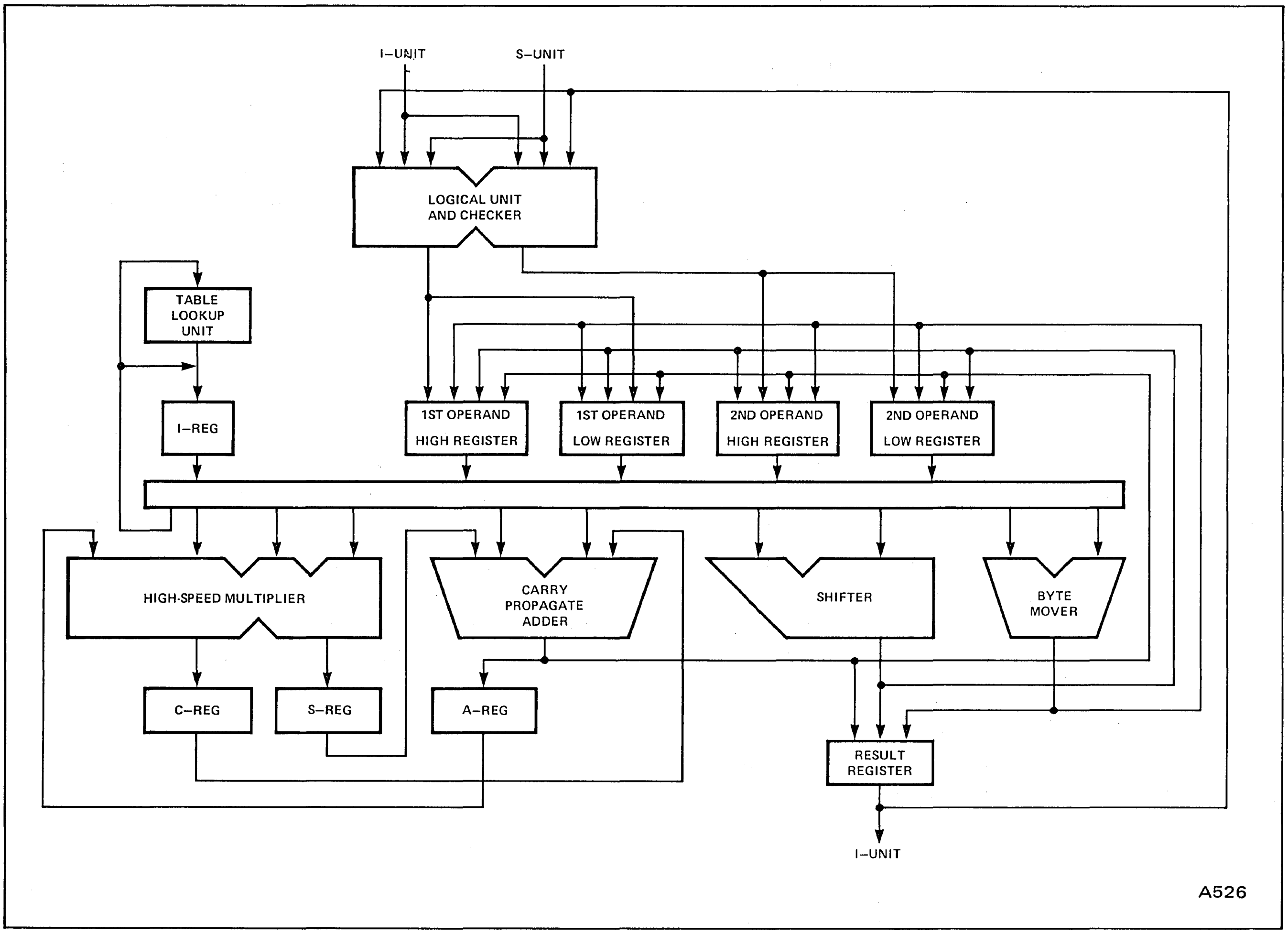


FIGURE 5 E-UNIT ORGANIZATION

CHANNEL UNIT

C-UNIT FUNCTIONS

The Channel Unit (C-Unit) implements the 470V/7 inboard channels. Except for occasional memory-access conflicts, these channels operate independently of the CPU. The channels may be configured in any combination of selector, byte-multiplexer, or block-multiplexer channels.

The C-Unit (figure 6) is implemented in large-scale-integration (LSI) technology. Associated with the C-Unit is the channel frame, which is implemented in non-LSI (third generation) technology. The channel frame contains the Remote Interface Logic (RIL), Channel Buffer Store (CBS), Subchannel Buffer Store (SBS), and other hardware used by the C-Unit. The C-Unit performs the I/O commands defined in the ***System/370 Principles of Operation*** and controls data movement to and from the S-Unit, data movement over the standard I/O interface, and communication with the I-Unit and S-Unit. The channel frame translates LSI signals to standard interface signals, drives and receives interface signals, and buffers I/O data.

C-UNIT ORGANIZATION

The C-Unit and channel frame together implement the channels. These channels share the same control logic. The Shifting Channel State (SCS) coordinates activities among the channels. Other parts of the C-Unit are the Controller Interface Control Logic (CICL), the Data Access Control Logic (DACL), and the Operations Control Logic (OCL).

SHIFTING CHANNEL STATE (SCS). The SCS maintains the current state of each channel. It is used by the OCL, DACL, and CICL. The status information for each channel rotates through the SCS by one step per cycle. Thus the OCL, DACL, and CICL can examine a different channel every cycle. The OCL, DACL, and CICL update the information in the SCS when appropriate; the SCS then forwards the new information.

CONTROLLER INTERFACE CONTROL LOGIC (CICL). The CICL moves data between the channel buffer store and the RIL and controls channel frame operations. The CICL has two ports into the SCS and two ports into the channel buffer store (CBS). Every two cycles, the CICL services two channels.

DATA ACCESS CONTROL LOGIC (DACL). The DACL moves and controls data between the S-Unit and C-Unit buffers. It examines each channel in the SCS once every 16 cycles. For an input operation, the data goes from the CBS to the S-Unit. The DACL is pipelined to overlap operations: while one section may be fetching data from the S-Unit, another may be posting results to the SCS. The DACL assigns each channel a dynamic priority based on the amount of data in its buffer. A priority change can occur while a fetch or store is in progress. The DACL attempts to select the highest-priority channel in the SCS to service.

OPERATIONS CONTROL LOGIC (OCL). The OCL sets up channel transfer sequences and coordinates channel program execution within the C-Unit. It sets up counts, flags, and data transfer addresses in the C-Unit buffers (normally the CBS), and it translates CCWs into CICL and DACL actions. The OCL obtains its control information directly from the I-Unit and S-Unit over an interface shared with the DACL.

CHANNEL BUFFER STORE (CBS). The Channel Buffer Store (CBS) contains a buffer for each channel. The CICL, DACL and OCL all communicate data and control information through the CBS. The CICL transfers one or two bytes per channel per access (two bytes are transferred per access on channels with the optional two byte interface). The DACL and OCL transfer on a one word basis.

CHANNEL OPERATION

The data path through the channel is shown in figure 6. The OCL interprets the channel program and indicates in the SCS the desired action for the appropriate channel. If the DACL sees an output request in the SCS, it fetches the data from the S-Unit and stores it in the CBS. The CICL then moves the data from the CBS to the RIL, which moves it to the external device. If the CICL sees an input request in the SCS, it fetches the data from the RIL and moves it to the CBS. The DACL then moves the data to the S-Unit.

MULTIPLEXING

The OCL coordinates subchannel activity for byte and block multiplexing. It stores inactive subchannel information in the Subchannel Buffer Store (SBS) and maintains subchannel status.

INDIRECT DATA ADDRESSING

Channel Indirect Data Addressing (IDA) as described in the ***System/370 Principles of Operation*** is fully implemented in the Amdahl 470V/7 system. IDA requires a control program to perform virtual-to-real address translations before a data-transfer command is executed by the channel.

CHANNEL TYPES

Any 470V/7 channel can be configured as a block multiplexer, byte multiplexer, or selector. Selector channels transfer only in burst mode and may address up to 256 I/O devices one at a time. Multiplexer channels execute several channel programs concurrently. Each channel program requires its own subchannel; therefore, the number of concurrent channel programs cannot exceed the number of allocated subchannels. For an explanation of subchannel assignment, see page 28.

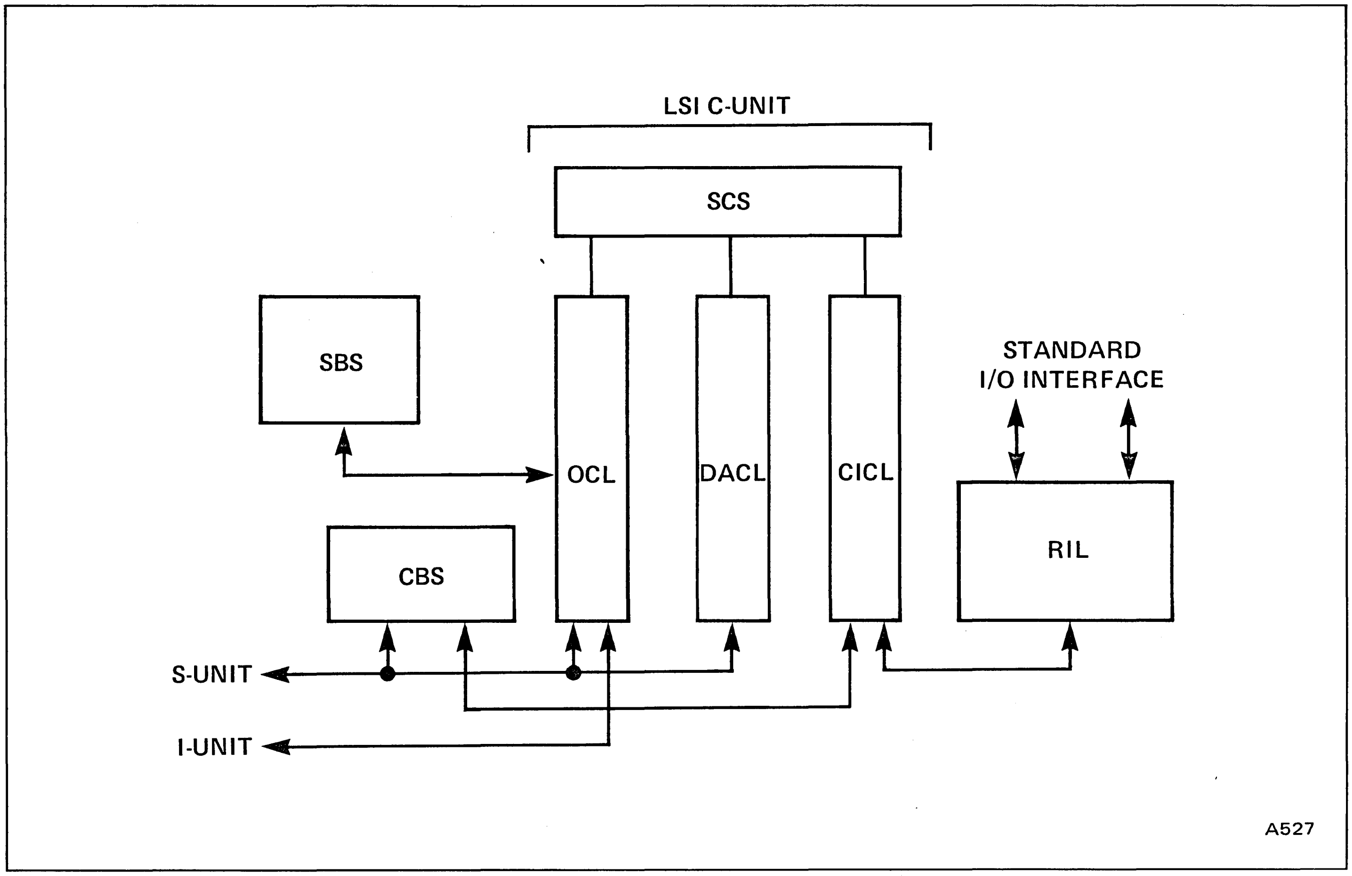


FIGURE 6 C-UNIT ORGANIZATION

CHANNEL BANDWIDTH

When allocating devices to channels, the channel bandwidth must be considered. Specific characteristics of certain high-speed devices can affect channel bandwidths; therefore, channel assignments for high-speed devices should be confirmed with an Amdahl representative.

SELECTOR. The maximum data rate on a selector channel is approximately two megabytes per second. An optional 2-byte interface doubles this rate.

BLOCK MULTIPLEXER. The maximum data rate on a standard, single-byte, block-multiplexer channel is approximately two megabytes per second. An optional 2-byte interface doubles this rate.

BYTE MULTIPLEXER. The maximum data rate for a byte-multiplexer channel in byte-multiplex mode is approximately 110 kilobytes per second.

NOTE. Chained CCWs will reduce the maximum data rate unless the chaining occurs during device gap time. Similarly, channel indirect data addressing can reduce the data rate. Chaining in a virtual environment therefore greatly increases the possibility of a channel overrun.

STORAGE UNIT

S-UNIT FUNCTIONS

The Storage Unit (S-Unit) performs all main storage requests from the I-Unit, E-Unit, and C-Unit. It also performs Dynamic Address Translation (DAT). See figure 7.

S-UNIT ORGANIZATION

Three features increase the speed of the S-Unit: the High-Speed Buffer (HSB), the Translation Lookaside Buffer (TLB), and the Segment Table Origin (STO) stack.

HIGH SPEED BUFFER. The HSB contains frequently used lines of memory. Because an HSB access is much faster than a main storage access, the S-Unit saves time by using the HSB to retrieve and write data.

TRANSLATION LOOKASIDE BUFFER. The TLB is a 512-entry table of frequently used virtual addresses with their real address translations. By using the TLB, the S-Unit can avoid translating most addresses.

SEGMENT TABLE ORIGIN (STO) STACK. A 128-entry STO stack saves the data from control registers 0 and 1 that define the current segment table. Each TLB entry is associated with an STO stack entry. Instead of purging the TLB whenever control registers 0 and 1 change, the S-Unit checks the STO ID of TLB entries to make sure they are valid with the current control register values. Up to 128 different virtual-address spaces can have currently valid identification information in the STO stack. Thus, at the same time, up to 128 different virtual-address spaces can have active translation information in the TLB.

S-UNIT OPERATION

When the S-Unit receives a virtual address, it starts the HSB and TLB simultaneously. While it uses the low-order (real) bits of the virtual address to create a pointer into the HSB, it uses the high-order virtual address bits to translate to a real address. It usually finds the real address in the TLB by the time it needs the high-order real address bits in the HSB. If the virtual address is not in the TLB, the S-Unit performs a complete translation and puts the address into the TLB. After using the real address to decide which bytes in the HSB were requested, the S-Unit forwards these bytes to the I-Unit, E-Unit, or C-Unit. If the requested bytes are not in the HSB, the S-Unit retrieves them from main storage, loads the 32-byte storage line containing the requested bytes into the HSB, and concurrently bypasses the data to the requesting unit. Refer to figure 7.

HIGH-SPEED BUFFER (HSB)

HIGH-SPEED BUFFER ORGANIZATION

The 470V/7 HSB is a 32,768-byte (32K) set-associative memory. It is divided into eight parts. Each part contains 128 32-byte lines.

TWO-KILOBYTE PAGING

For system control programs using 2K pages, it is necessary to operate the HSB in 16K mode. In this mode, each buffer part contains 64 32-byte lines.

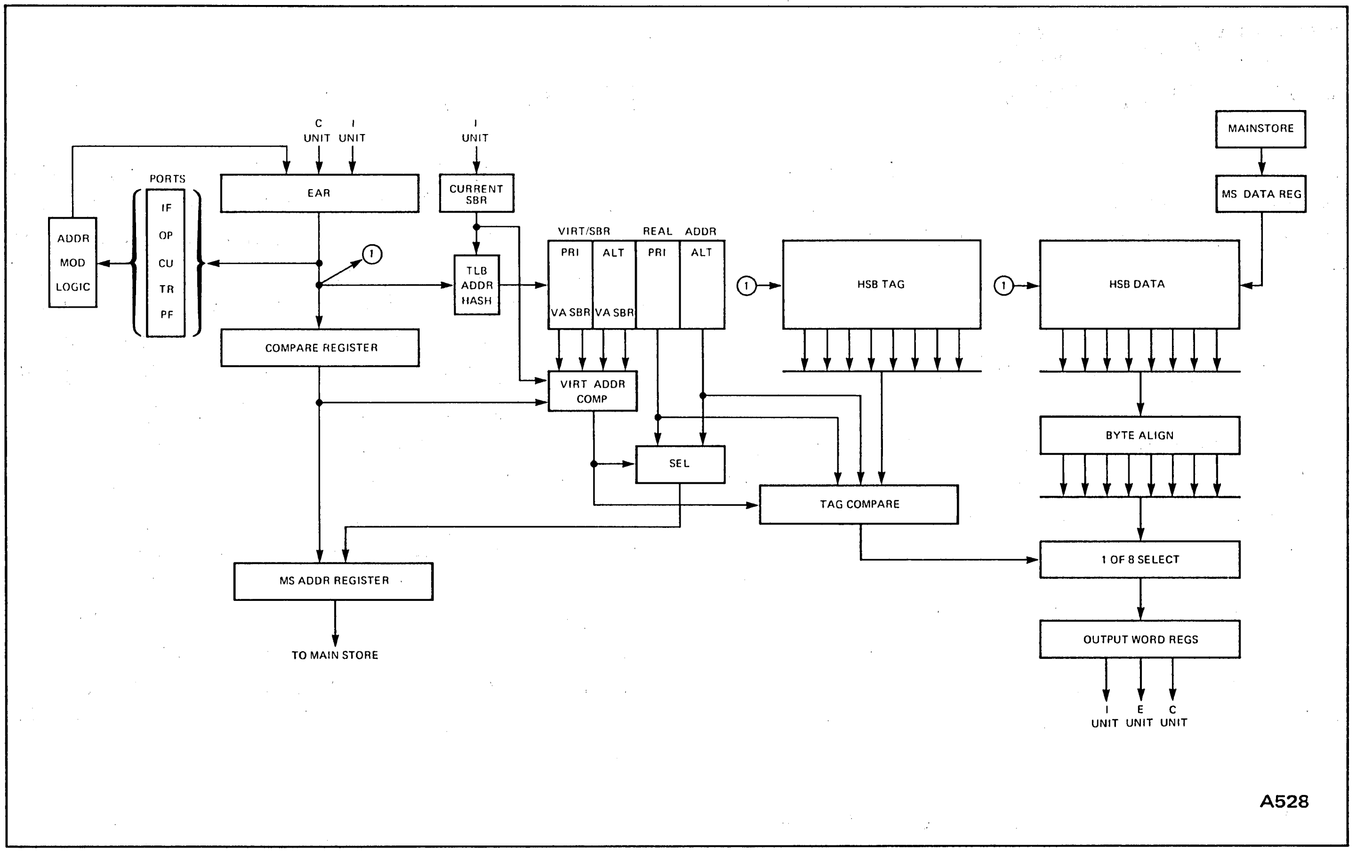


FIGURE 7 S-UNIT ORGANIZATION

HIGH-SPEED BUFFER TAG

Each line in the HSB has a tag associated with it. This tag contains three fields that identify and protect the data: tag-identifier, key, and status. Refer to figure 8. The tag-identifier field contains the real address. The key field contains five protection-key bits, a parity bit, and a check bit. The status field specifies whether the line is valid and unmodified, valid and modified, or invalid. It also specifies the type of modification: an ECC correction or a store made under program control. The status field can be recovered from a single-bit error.

TAG IDENTIFIER	KEY	STATUS
(REAL ADDRESS BITS)	0 1 2 3 4 PC	0 1 2 3

FIGURE 8 HIGH-SPEED BUFFER TAG A551

FINDING A LINE IN THE HSB

When the S-Unit references the HSB, it first forms a pointer into the buffer using bits 20–26 of the requested address. This pointer defines eight corresponding lines (refer to figure 9).

The S-Unit must decide which of these eight lines contains the requested bytes. To do this, it compares the tag-identifier fields of all eight tags with the corresponding real address. (These real address bits are determined by dynamic address translation (DAT)).

While it is performing the tag compare, the S-Unit simultaneously uses bits 27–31 of the requested address to decide which bytes of the 32 in the line were requested and aligns these bytes.

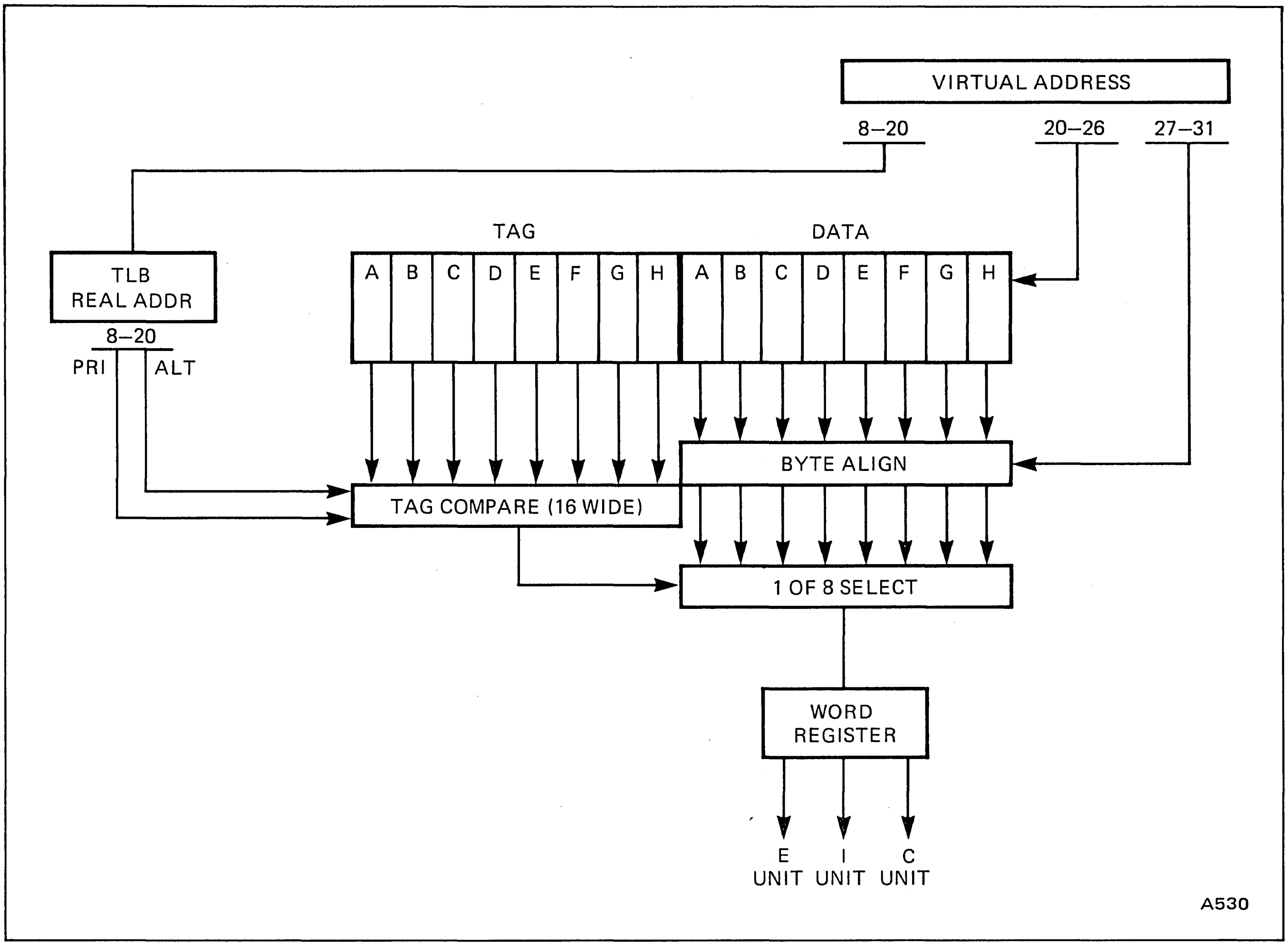


FIGURE 9 HIGH-SPEED BUFFER OPERATION

FETCHING A LINE FROM THE HSB

If the S-Unit is fetching a line from the HSB, it finds the eight possible lines and performs a tag compare. If one of the tags matches the real address bits, the resulting data selection forwards the desired bytes to the word registers, where they are available to the I-Unit, E-Unit, or C-Unit. If no tag matches the real address bits, the requested bytes are not in the buffer. In this case, the S-Unit moves the line into the HSB from Main Storage.

MOVING A LINE INTO THE HSB

To move a line into the HSB, the S-Unit fetches from main storage the 32-byte line containing the requested byte and creates a tag for the line, using the real address bits.

Because each storage line maps into eight specific HSB locations, the S-Unit must decide which of the lines already at the given locations to replace with the new lines. If a line is invalid, it is replaced immediately. If no line is invalid, the S-Unit replaces the least recently used line. If the line to be replaced is modified, it is written to main storage as the new line replaces it. (This write to main storage occurs in the background with no additional delay.)

STORING DATA IN THE HSB

When data is altered by a program, the S-Unit makes the change in the HSB. The change is not forwarded to main storage until the entire line is written back (such as when the buffer location is needed for another line).

To store data in the HSB, the S-Unit finds the appropriate line, updates the requested bytes, and sets the status field of the tag to show that the line is modified.

HSB RECONFIGURATION

If a buffer error occurs, the HSB is reconfigured by disabling the part in error. The rest of the HSB remains available to the system.

DYNAMIC ADDRESS TRANSLATION (DAT)

The 470V/7 can perform dynamic address translation (DAT) when in EC mode. Virtual addressing in the 470V/7 operates as defined in the *System/370 Principles of Operation*.

When the S-Unit performs an address translation using the segment and page tables, it saves the result in the Translation Lookaside Buffer (TLB).

STO STACK AND TLB ORGANIZATION

The segment table origin stack (STO) has 120 locations. It is addressed by the current segment table origin. Each STO stack entry records pertinent data from a recent value of control registers 0 and 1. One other bit, called the flipper bit, distinguishes old STO stack entries from new ones. The STO address and the flipper bit together constitute an STO ID.

The translation lookaside buffer (TLB) is divided into a primary half and an alternate half. Each half has 256 locations. The TLB address is a mapping of the current STO ID and the virtual address. The primary half and the alternate half are each addressed with a different mapping function. By making TLB address assignment more random, the mapping function reduces TLB address conflicts.

STO STACK ENTRIES

Whenever the value of control registers 0 and 1 changes, the S-Unit examines the STO stack entry addressed by the current segment table origin. If this location is empty, the S-Unit creates a new entry in the table. If there is already an entry, the S-unit compares it to the current value of control registers 0 and 1. If the entry and registers match, the entry is still valid. If they do not match, the S-Unit creates a new entry, writes it into the stack, and purges all TLB entries associated with the old STO ID.

SAVING A TRANSLATION

To save a virtual/real address translation in the TLB, the S-Unit finds the two TLB locations to which the virtual address maps and saves the new translation, along with the current STO ID, in the location less recently used. (The hot/cold bit determines which location was used more recently.)

RETRIEVING A TRANSLATION

When the S-Unit retrieves an address translation from the TLB, it first finds the two entries, primary and alternate, to which the presented virtual address maps. Then it compares bits 8–20 of the virtual address to these two entries to find the one that matches. Simultaneously, the S-Unit comapres the STO ID of both entries to the currently valid STO ID. If the presented virtual address and the current STO ID match one of the TLB entries, the associated real frame address is forwarded as the real address. If not, a full translation is performed and the new virtual/real pair is saved in the TLB.

PURGE TLB

To enhance performance, the TLB has two sets of valid bits. When the PURGE TLB instruction is executed, the S-Unit immediately switches to the other set of valid bits, which are all marked invalid. The S-Unit then resets the older set of valid bits in parallel with subsequent buffer accesses. Because PURGE TLB is issued relatively infrequently, the alternate set of valid bits will usually be reset by the time they are needed again, and the instruction will normally require only a few cycles.

ERROR CHECKING AND CORRECTION

The S-Unit stores an Error Checking and Correction (ECC) field with each 8 bytes of data in main storage. This field contains enough information to correct any single-bit error and detect any double-bit error within the 8 bytes. If the S-Unit detects a single-bit error while retrieving a line from main storage, it corrects the error in the HSB and flags the line as modified in the status field of the high-speed buffer tag.

SYSTEM CONSOLE

CONSOLE FUNCTIONS

The 470V/7 System Console provides communication with the 470, usage metering, diagnostic information on the hardware, and intermediate storage for machine-check logouts.

On the 470, most console input is entered on the keyboard rather than on toggle or rotary switches, and most console output appears as a formatted CRT display rather than a panel-light display.

COMMUNICATION. The console provides all standard communication between the 470V/7 processor and the operator. It emulates a 3066 or 3215 operator's console, performs functions such as IPL, reset, and clear, and displays diagnostic messages and the contents of registers, latches, and storage.

MACHINE-CHECK LOGOUT STORAGE. The 470 V/7 system console stores machine-check logout information on its attached disk. This makes it possible to save over a hundred scan pages at the time of a machine failure and to later display these pages at the console.

DIAGNOSTIC INFORMATION. The 470V/7 console provides formatted displays of approximately 17,000 latches within the 470 system. These displays are called "scan pages"; each scan page gives the current status of one area or function of the machine. The console also gives a continuous machine-status display at the top of the CRT screen. This display summarizes the current state of the 470.

USAGE METERING. Both a system meter and a maintenance meter reside in the console. The system meter accumulates time when the maintenance key switch is in the system position and the SYSTEM light is on. (The SYSTEM light in the operator's control panel will be on if the CPU is not in STOP, WAIT, or CHECK STOP state. It will also be on if a channel is active and the CPU is not in CHECK STOP state.) The maintenance meter accumulates time when the maintenance key switch is in the maintenance position.

CONSOLE COMPONENTS

The 470V/7 system console includes a CRT display screen, a keyboard, a standard channel interface, a computer-to-console interface, an independent processor, and a modem.

The standard channel interface is used when the 470V/7 is using the console to emulate a 3066 or a 3215. The computer-to-console interface is used when the console is reading scan information or issuing hardware commands to the 470.

The console processor is a minicomputer that allows the console to operate independently of the rest of the 470. The console can interrogate and diagnose the 470, whether it is running or stopped, even if the 470 is not operational. The console processor also performs 470 hardware functions such as Display Register or Alter Register. A disk and diskette are attached to the console processor.

The modem allows remote access to the 470V/7. Through the modem, the Amdahl central maintenance diagnostic facility, AMDAC, can diagnose problems from Amdahl headquarters.

CONSOLE OPERATION

The 470V/7 console operates in one of two modes: device support mode and hardware command mode.

DEVICE-SUPPORT MODE. In device-support mode, the console simulates the device support mode of an IBM 3066 or 3215 operator's console. This allows the operator to communicate with the system control program. In this mode, the console acts as a control unit and may be connected to either a selector or block-multiplexer channel.

HARDWARE-COMMAND MODE. In hardware-command mode, the console lets the operator communicate directly with the hardware, rather than with the system control program. This is the mode in which the console performs such commands as IPL, Reset, and Display Register. While the CRT and keyboard are used in hardware-command mode, device-support mode may continue in the background. The Amdahl field engineering staff uses maintenance mode to maintain and diagnose the 470V/7 hardware. In this mode, the computer can be connected to AMDAC and then can be used in either mode.

INSTRUCTION SET DIFFERENCES

Two instructions have important model-dependent results on the Amdahl 470V/7. They are: STORE CPU ID (STIDP) and STORE CHANNEL ID (STIDC).

STORE CPU ID

STIDP stores model-dependent data at the double word addressed by the second operand. Table 1 shows the information stored for the 470V/7.

STORE CHANNEL ID

STIDC stores channel-dependent data at decimal location 168. Because the 470V/7 channel model is implicit in the CPU model, zeros are stored in the channel model-number field. The remaining fields, channel type and IOEL length, follow standard conventions.

TABLE 1 STORE CPU ID

FIELD	BITS	VALUE STORED
Version Code	0–7	07
Serial Number	8–31	unique serial number
Model Number	32–47	0470
Maximum MCEL Length	48–63	0000

A552

MACHINE-CHECK CONDITIONS

The Amdahl 470V/7 system is continuously checking for valid data, instructions, arithmetic results, and legal control sequences. When an error is discovered, it can often be corrected without serious impact on machine performance.

Malfunctions causing machine-check interrupts (see figure 10) are grouped into two categories: repressible and exigent. These are defined in the ***System/ 370 Principles of Operation.***

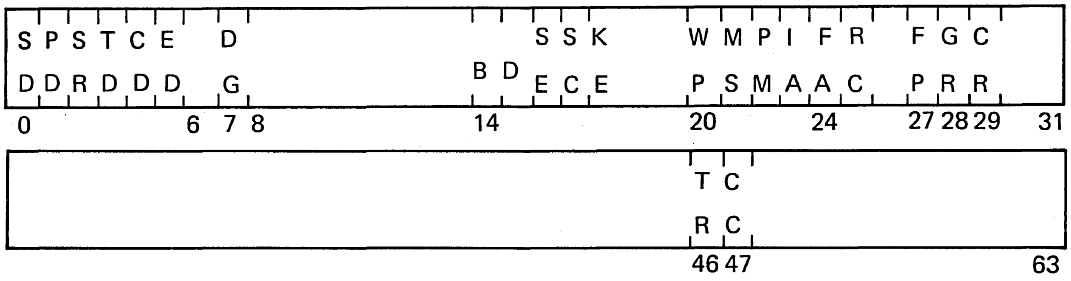
REPRESSIBLE CONDITIONS

Repressible conditions comprise system recovery conditions, timer damage conditions, time-of-day clock damage, external damage, and degradation of the segment table origin stack. These conditions do not terminate the current instruction or cause loss of interrupts.

A machine-check interrupt for a repressible condition occurs after an instruction, including any associated SVC interrupt or program interrupt, has completed. (This is the same point at which an I/O interrupt occurs.)

EXIGENT CONDITIONS

Exigent conditions comprise system damage conditions, multi-bit storage errors, protection-key parity errors, unretrievable internal data-transfer errors, move-out parity errors, and instruction processing



BIT	FUNCTION	BIT	FUNCTION
SD	System damage	KE	Storage Key error uncorrected
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	RC	Region code valid
B	Backed-up	FP	Floating point registers valid
D	Delayed	GR	General purpose registers valid
SE	Storage error uncorrected	CR	Control registers valid
SC	Storage error corrected	TR	CPU timer valid
		CC	Clock comparator valid

A532

FIGURE 10 MACHINE-CHECK INTERRUPTION CODE (MCIC)

damage conditions (if retry is unsuccessful or impossible).

A machine-check interrupt for an exigent condition immediately inhibits any updating of the machine state, including storage and registers, without waiting for an instruction to end. It points the instruction counter to the instruction farthest along in the pipeline, although any of the instructions in the pipeline may have caused the error.

SYSTEM RECOVERY CONDITIONS

The 470V/7 system has two facilities for error correction: Hardware Instruction Retry (HIR) and Error Checking and Correction (ECC). Any corrected error causes a system recovery condition.

HARDWARE INSTRUCTION RETRY. When an error is detected in the execution of an instruction, the HIR circuitry can usually retry the instruction. If the retry is successful, the machine check is repressible. If the retry is unsuccessful, the machine check is exigent.

ERROR CHECKING AND CORRECTION. Each 8-byte section of main storage has an ECC field associated with it. This field contains sufficient information to correct any single-bit error within the 8 bytes.

I/O ERRORS

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

When the C-Unit detects an external I/O equipment malfunction, it reports the error in a CSW as an I/O interrupt. The error is not handled as a machine check.

MACHINE-CHECK LOGOUTS

FIXED LOGOUT AREA

The 104-byte area starting at location 248 is reserved for machine-check logouts. The Amdahl 470 V/7 uses only the first 12 bytes of this area. The failing storage address (FSA) occupies the word starting at location 248; the region code occupies the three words starting at location 252. The rest of the area, locations 264—351, is reserved.

FAILING STORAGE ADDRESS (FSA). The FSA indicates the byte or block in which the error occurred. For a correctable storage error, bits 1—3 of the FSA contain the failing bit address; bits 8—31 contain the failing byte address. For an uncorrectable storage error, bits 8—28 of the FSA point to the failing 8-byte ECC block. For an uncorrectable protection-key error, bits 8—31 of the FSA may point anywhere within the 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 is set to zero.

REGION CODE. The region code specifies which part of the machine detected the error. Table 2 defines the region code bits.

MACHINE CHECK EXTENDED LOGOUT

The 470V/7 performs a machine-check extended logout (MCEL) when a machine check occurs and the mask bits of control register 14 are set to allow the logout. The logout on a 470V/7 includes a set of scan pages that record the state of approximately 17,000 latches in the system. These are the same scan pages that can be displayed at the console in hardware command mode. The console processor its memory or attached disk.

While the console processor performs the MCEL, the CPU suspends processing. When the logout is complete, the console restarts the CPU, which can then perform its own machine-check handling routines.

Machine-check handling software can access the console logout in two ways: through the channel or through the computer-to-console interface (CCI). Both methods will transfer the logout from the console to main storage.

CONTROL REGISTERS 14 AND 15

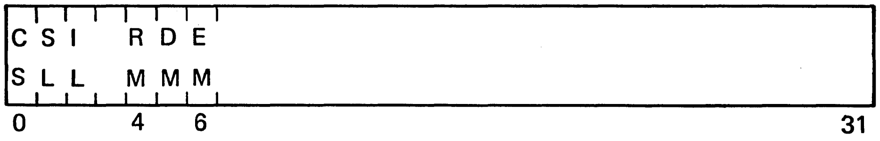
Because MCEL data is saved in the console, control register 15, which normally contains the MCEL address, is not implemented on the 470V/7 and stores as all zeros.

The significant bits of control register 14 appear in figure 11. These bits operate as defined in the *System/370 Principles of Operation*. Bit 4, recovery report mask, controls machine interrupts of both hardware instruction retry (HIR) and error checking and correction (ECC).

TABLE 2 470V/7 REGION-CODE BITS

Storage Location	Bit	Source	Storage Location	Bit	Source
252	0	I-Unit Pipeline Control Error	257	0	I-Unit Result Byte 0 Parity Error
	1	E-Unit Condition Code Error		1	I-Unit Result Byte 1 Parity Error
	2	E-Unit LUCK1 Byte 0 Parity Error		2	I-Unit Result Byte 2 Parity Error
	3	E-Unit LUCK1 Byte 1 Parity Error		3	I-Unit Result Byte 3 Parity Error
	4	E-Unit LUCK1 Byte 2 Parity Error		4	I-Unit EAG Parity Error (DA)
	5	E-Unit LUCK1 Byte 3 Parity Error		5	I-Unit EAG Parity Error (CI)
	6	E-Unit LUCK2 Byte 0 Parity Error		6	I-Unit Instruction Stream Entrance Parity Error
	7	E-Unit LUCK2 Byte 1 Parity Error		7	I-Unit Store Data Parity Error
253	0	E-Unit LUCK2 Byte 2 Parity Error	258	0	S-Unit TLB SBR ID Parity Error
	1	E-Unit LUCK2 Byte 3 Parity Error		1	S-Unit SBR Error
	2	E-Unit Multiplicand Byte 0 Parity Error		2	C-Unit 1 I/O Address Parity Error From I-Unit
	3	E-Unit Multiplicand Byte 1 Parity Error		3	Reserved
	4	E-Unit Multiplicand Byte 2 Parity Error		4	Reserved
	5	E-Unit Multiplicand Byte 3 Parity Error		5	Reserved
	6	E-Unit Adder High-Input Phase Error		6	Reserved
	7	E-Unit Adder Low-Input Phase Error		7	Reserved
254	0	S-Unit Compare Reg Parity Error	259	0	E-Unit Multiplier Residue Error
	1	S-Unit Tag Control Parity Error		1	E-Unit Adder Residue Error
	2	S-Unit Tag Key Parity Error		2	I-Unit Instruction Stream Exit Parity Error (DS)
	3	S-Unit Tag ID Parity Error		3	S-Unit Buffer LRU Error
	4	S-Unit Store Read Address Parity Error		4	I-Unit Control Register Bytes 0—1 Parity Error
	5	Main Store Read Address Parity Error		5	I-Unit Control Register Bytes 2—3 Parity Error
	6	Main Store Key Write Parity Error		6	I-Unit PSW Bytes 0—1 Parity Error
	7	Main Stroe Write Address Parity Error		7	Cycle Counter Parity Error
255	0	S-Unit LRC Error	260	0	S-Unit Execution Key Parity Error
	1	S-Unit Move Out Data Parity Error 0		1	S-Unit Encoded Buffer Bit 0
	2	S-Unit Byte Indication A		2	S-Unit Encoded Buffer Bit 1
	3	S-Unit Byte Indication B		3	S-Unit Encoded Buffer Bit 2
	4	S-Unit Reserved		4	S-Unit Compare Reg SBR ID Parity Error
	5	Multiple Byte Error		5	S-Unit Port ID A
	6	S-Unit Primary (1)/Alternate (0) TLB		6	S-Unit Port ID B
	7	S-Unit Translation Register Segment/Page Table Parity Error		7	S-Unit Port ID C
256	0	E-Unit Multiplier Byte Parity Error	261	0	Main Store Interface Error
	1	E-Unit Byte Adder Input 1 Parity Error		1	Main Store Configuration Reg Parity Error
	2	E-Unit Byte Adder Input 2 Parity Error		2	Main Store Reference/Change Address Parity Error
	3	E-Unit Byte Adder Input 3 Parity Error		3	Main Store Reference/Change Address OP Bus Parity Error
	4	S-Unit TLB Valid Error		4	Main Store Reference/Change Address Parity Error
	5	S-Unit TLB Key Parity Error		5	S-Unit RACR Parity Error
	6	S-Unit TLB Logical Address Parity Error		6	S-Unit Main Store Address Register Parity Error
	7	S-Unit RAR Parity Error		7	S-Unit General Word Register Parity Error

A533



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

A531

FIGURE 11 CONTROL REGISTER 14 – MACHINE-CHECK CONTROL REGISTER

CHANNEL LOGOUT

EXTENDED CHANNEL LOGOUT

I/O Extended Logout (IOEL) as defined in the *System/370 Principles of Operation* is fully implemented on the 470V/7 system. Figure 12 gives a diagram of the 470V/7 IOEL. The first four words are selected bits from the LSI Channel State, the next 12 words are Channel Buffer Store control information, and the last field is from a C-Unit storage area for subchannel state information. Because the number of subchannels varies from channel to channel, the length of this last field varies also. For selector channels, the length is zero; for multiplexer channels the length is four words for every 32 subchannels installed, up to 32 words. Figure 13 gives a detailed breakdown of IOEL words 0 to 3.

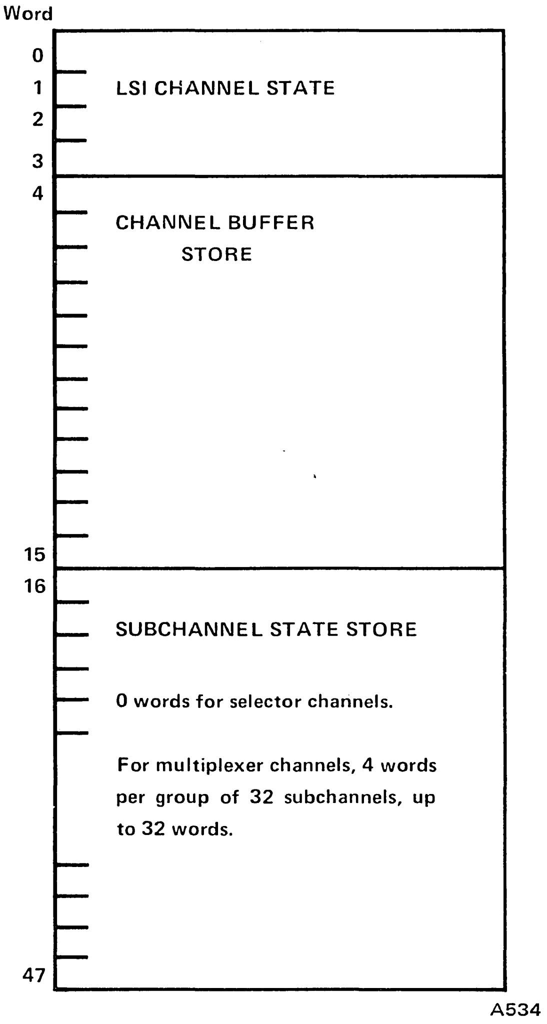

Word
0
1
2
3
4
15
16
47
LSI CHANNEL STATE
CHANNEL BUFFER STORE
SUBCHANNEL STATE STORE
0 words for selector channels.
For multiplexer channels, 4 words per group of 32 subchannels, up to 32 words.
A534

FIGURE 12 I/O EXTENDED LOGOUT

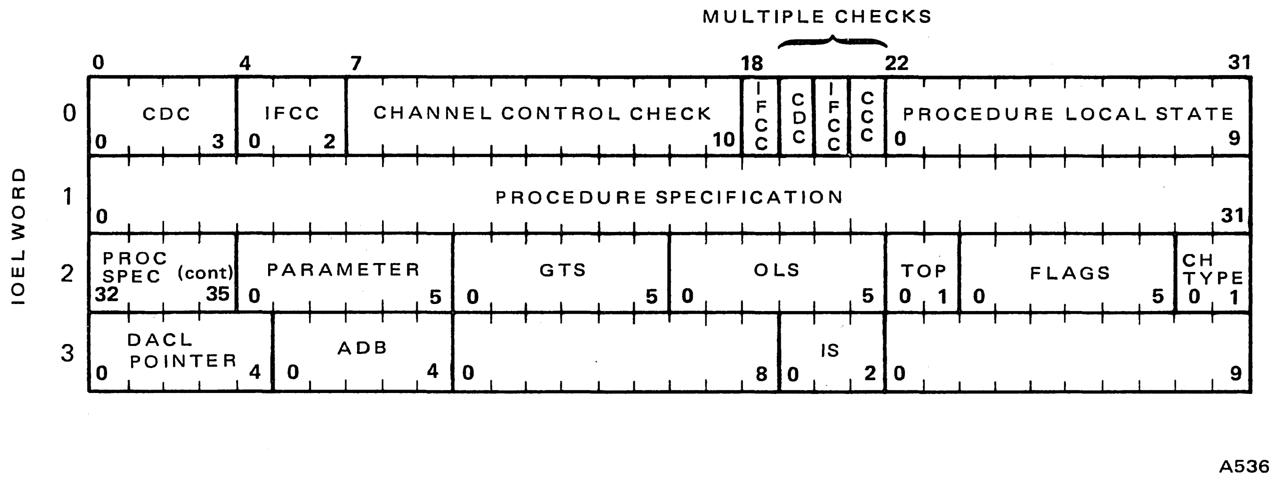


FIGURE 13 CHANNEL LOGOUT STATE

SUBCHANNEL ASSIGNMENT

470V/7 SUBCHANNELS

A total of 2048 subchannels may be assigned to the multiplexer channels on a 470V/7 system. They can be assigned in multiples of 32, up to a total of 256. The implicit subchannel of a selector channel is not part of the 2048 total. Selector subchannels are not available on 470V/7 byte-multiplexer channels.

I/O unit addresses are in the form "CUU", where "C" is the hexadecimal channel address and "UU" is the hexadecimal device address.

UNSHARED SUBCHANNELS

If a device address is less than the number of subchannels on the channel, then the subchannel address is equal to the device address. (For example, if there are 96 subchannels on the channel, device addresses 00 to 5F are assigned to subchannel addresses 00 to 5F.)

If a device address is greater than or equal to the number of subchannels on the channel, then the subchannel address for that device is determined by the low-order bits of the device address. (Bits are numbered 0 to 7, left to right.)

- On multiplexer channels with 128, 160, 192 or 224 unshared subchannels, the first 128, 160, 192, or 224 device addresses are assigned sequential subchannel addresses. The remaining device addresses are assigned to subchannel addresses modulo 128.

- On channels with 64 or 96 unshared subchannels, the first 64 or 96 device addresses are assigned to sequential subchannel addresses. The remaining addresses are assigned to subchannel addresses modulo 64.

- On channels with 32 unshared subchannels, all device addresses are assigned to subchannel addresses modulo 32.

For example, on a channel with 64 subchannels, the unit addresses 301, 341, 381, and 3C1 all map into subchannel address 01. Therefore, only one of these addresses can be assigned to a device.

On a channel with 256 subchannels, the subchannel address is always equal to the device address.

Figure 14 illustrates the device address groups associated with each subchannel address group. The device addresses within a group are unique, but all groups associated with the same subchannel addresses duplicate the same address range.

SHARED SUBCHANNELS

Subchannels can be shared on multiplexer channels with less than 256 subchannels. (Channels with 256 subchannels cannot have shared subchannels, because 256 gives each device its own subchannel.)

Depending on the total number of subchannels, either two, four, or eight subchannel addresses are shared.

- On a channel with 32 subchannels, subchannel addresses 00 and 01 can be shared.

- On a channel with 64 or 96 subchannels, subchannel addresses 00, 01, 02, and 03 can be shared.

- On a channel with 128, 160, 192, or 224 subchannels, subchannel addresses 00, 01, 02, 03, 04, 05, 06, and 07 can be shared.

If a device address is less than the number of subchannels on the channel, then the subchannel address is equal to the device address. (For example, if there are 64 subchannels, device addresses 00 to 3F are assigned to subchannel addresses 00 to 3F.)

32 SUBCHANNELS

SUBCHANNEL ADDRESSES	00 – 1F
DEVICE ADDRESSES	00 – 1F 20 – 3F 40 – 5F 60 – 7F 80 – 9F A0 – BF C0 – DF E0 – FF

64 SUBCHANNELS

SUBCHANNEL ADDRESSES	00 – 3F
DEVICE ADDRESSES	00 – 3F 40 – 7F 80 – BF C0 – FF

96 SUBCHANNELS

SUBCHANNEL ADDRESSES	00 – 1F	20 – 3F	40 – 5F
DEVICE ADDRESSES	00 – 1F 80 – 9F C0 – DF	20 – 3F 60 – 7F A0 – BF E0 – FF	40 – 5F

128 SUBCHANNELS

SUBCHANNEL ADDRESSES	00 – 7F
DEVICE ADDRESSES	00 – 7F 80 – FF

A540

FIGURE 14 SUBCHANNEL ASSIGNMENT: UNSHARED SUBCHANNELS (Part 1 of 2)

160 SUBCHANNELS

SUBCHANNEL ADDRESSES	00 – 1F	20 – 7F	80 – 9F
DEVICE ADDRESSES	00 – 1F	20 – 7F A0 – FF	80 – 9F

192 SUBCHANNELS

SUBCHANNEL ADDRESSES	00 – 3F	40 – 7F	80 – BF
DEVICE ADDRESSES	00 – 3F	40 – 7F C0 – FF	80 – BF

224 SUBCHANNELS

SUBCHANNEL ADDRESSES	00 – 5F	60 – 7F	80 – DF
DEVICE ADDRESSES	00 – 5F	60 – 7F E0 – FF	80 – DF

256 SUBCHANNELS

SUBCHANNEL ADDRESSES	00 – FF
DEVICE ADDRESSES	00 – FF

A541

FIGURE 14 SUBCHANNEL ASSIGNMENT: UNSHARED SUBCHANNELS (Part 2 of 2)

If the device address is greater than or equal to the number of subchannels on the channel, then the shared subchannel address will be equal to the value of either bit 3, bits 2 and 3, or bits 1, 2, and 3 of the device address, depending on the total number of subchannels. (Bits are numbered 0 to 7, left to right.)

- On a channel with 32 subchannels, bit 3 of the device address gives the shared subchannel address.
- On a channel with 64 or 96 subchannels, bits 2 and 3 of the device address give the shared subchannel address.
- On a channel with 128, 160, 192, or 224 subchannels, bits 1, 2, and 3 of the device address give the shared subchannel address.

For example, if there are 128 subchannels, device address B8 (1011 1000) will map into shared subchannel 03, because bits 1, 2, and 3 of B8 have the value 03. Note that device address 38 (0011 1011) will map to an **unshared** subchannel.

Figure 15 shows how device addresses are assigned to shared subchannel addresses.

Be sure to assign shared subchannel addresses to control units that share subchannels and to assign unshared subchannel addresses to control units that do not share subchannels. On a byte-multiplexer channel, only one control unit may be assigned to each shared subchannel. On a block-multiplexer channel, multiple control units can be assigned to a single shared subchannel, but the channel will act as a selector channel when servicing a device assigned to a shared subchannel.

32 SUBCHANNELS

	SHARED		UNSHARED
SUBCHANNEL ADDRESSES	00	01	02 – 1F
DEVICE ADDRESSES	00 20 – 2F 40 – 4F 60 – 6F 80 – 8F A0 – AF C0 – CF E0 – EF	01 30 – 3F 50 – 5F 70 – 7F 90 – 9F B0 – BF D0 – DF F0 – FF	02 – 1F

64 SUBCHANNELS

	SHARED				UNSHARED
SUBCHANNEL ADDRESSES	00	01	02	03	04 – 3F
DEVICE ADDRESSES	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	04 – 3F

96 SUBCHANNELS

	SHARED				UNSHARED
SUBCHANNEL ADDRESSES	00	01	02	03	04 – 5F
DEVICE ADDRESSES	00 80 – 8F C0 – CF	01 90 – 9F D0 – DF	02 60 – 6F A0 – AF E0 – EF	03 70 – 7F B0 – BF F0 – FF	04 – 5F

A546

FIGURE 15 SUBCHANNEL ASSIGNMENT: SHARED SUBCHANNELS (Part 1 of 2)

128 SUBCHANNELS

	SHARED								UNSHARED
SUBCHANNEL ADDRESSES	00	01	02	03	04	05	06	07	08 – 7F
DEVICE ADDRESSES	00 80 – 8F	01 90 – 9F	02 A0 – AF	03 B0 – BF	04 C0 – CF	05 D0 – DF	06 E0 – EF	07 F0 – FF	08 – 7F

160 SUBCHANNELS

	SHARED								UNSHARED
SUBCHANNEL ADDRESSES	00	01	02	03	04	05	06	07	08 – 9F
DEVICE ADDRESSES	00	01	02 A0 – AF	03 B0 – BF	04 C0 – CF	05 D0 – DF	06 E0 – EF	07 F0 – FF	08 – 9F

192 SUBCHANNELS

	SHARED								UNSHARED
SUBCHANNEL ADDRESSES	00	01	02	03	04	05	06	07	08 – BF
DEVICE ADDRESSES	00	01	02	03	04 C0 – CF	05 D0 – DF	06 E0 – EF	07 F0 – FF	08 – BF

224 SUBCHANNELS

	SHARED								UNSHARED
SUBCHANNEL ADDRESSES	00	01	02	03	04	05	06	07	08 – DF
DEVICE ADDRESSES	00	01	02	03	04	05	06 E0 – EF	07 F0 – FF	08 – DF

A547

FIGURE 15 SUBCHANNEL ASSIGNMENT: SHARED SUBCHANNELS (Part 2 of 2)

CONSOLE CHANNEL PROGRAMMING

CHANNEL COMMAND WORDS

Channel command words (CCWs) control the console in device support mode only. In this mode, the console can perform two functions: emulation of a 3066 or 3215 operator's console, or channel page passing.

3066 EMULATION

When emulating a 3066, the 470V/7 console responds to the CCWs defined in the ***370/168 Functional Characteristics*** manual (GA22-7010, revision level 4). These CCWs are summarized in table 3. The two bytes of console sense data for a 3066 are shown in table 4. The 35-line console-display area appears below he status display on the CRT screen.

TABLE 3 3066 CHANNEL COMMAND WORDS (CCW)

FUNCTION	HEX	EXPLANATION
NOP	'03'	No operation. This CCW sets the incorrect-length indication.
Sense	'04'	Reads two bytes of sense data (see Table 4).
Erase	'07'	Sets the entire screen to blanks, removes the cursor display, resets CRT buffer address and cursor address to zeros.
Alarm	'0B'	Sounds a one-second tone and lights the alarm key. This CCW sets the incorrect-length indication.
Set Buffer Address	'27'	Transfers a two-byte screen address to the console to designate the starting byte position for a subsequent Read or Write command.
Write	'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 2803 bytes are written. If position (34, 79) is reached, position (0, 0) is written next.
Read	'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 Address	'0F'	Transfers a two-byte address to the console to indicate the screen position at which the cursor should be displayed. If the keyboard was locked, this command unlocks it.
Read MI	'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 ('80' for "ENTER" and '40' for "CANCEL").
Lock Keyboard	'67'	Causes the cursor to be deleted from the screen and prevents keyboard entry upon the screen. This CCW sets the incorrect-length indication.

A542

TABLE 4 3066 CONSOLE 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	Channel-page-passing error
Byte 1		Reserved

A553

3215 EMULATION

When emulating a 3215, the 470V/7 console responds to the CCSs defined in the ***370/145 Functional Characteristics*** manual (GA24-3557, revision level 6). These CCWs are summarized in table 5.

The single byte of console sense data for a 3215 is shown in table 6.

The 470V/7 emulation of a 3215 differs from a standard 3215 in these respects:

- There is no hard copy on a 470. Output appears on the CRT screen below the status display.
- The 470 line-length is 80 characters rather than 132. Messages exceeding 80 characters will wrap to the next line.
- A backspace key is available on the 470.
- The Return key is implemented as the down arrow (↓) on the 470.
- A standard 3215 transmits data one byte at a time as each character is entered. A 470V/7 console transmits the entire line after the "ENTER" or "CANCEL" key is depressed. If the characters in the line exceed the byte-count in the channel program, the excess characters are truncated.
- If the CRT is switched to hardware command mode and a READ or WRITE to the console is attempted, the status returned will be Channel End, Device End, and Unit Check, and the sense returned will be Intervention Required.

TABLE 5 3215 CHANNEL COMMAND WORDS (CCW)

FUNCTION	HEX	EXPLANATION
Write	'01'	Writes without automatic carriage return.
NOP	'03'	No operation. This CCW sets the incorrect-length indication.
Sense	'04'	Reads one byte of sense data (see Table 6).
Write ACR	'09'	Writes with automatic carriage return.
Read	'0A'	Enables keyboard input.
Alarm	'0B'	Sounds audible alarm, lights console alarm indicator. This CCW sets the incorrect-length indication.

A543

TABLE 6 3215 CONSOLE SENSE DATA

Bit 0	Command reject
Bit 1	Intervention required
Bit 2	Bus out check
Bit 3	Equipment check
Bit 4	Unused
Bit 5	Unused
Bit 6	Unused
Bit 7	Channel-page-passing error

A554

FUNCTIONAL DIFFERENCES

The 470V/7 console performs several functions differently than both the 3066 and the 3215. Note these differences when emulating either console type:

- The 470V/7 console operates on a block-multiplexer or selector channel.

- The 470V/7 console may respond to initial selection with a Control Unit Busy Sequence and Status = hex 70. This can occur if the selection immediately follows a HALT I/O to the console or if the console is not emulating a 3215 or 3066 when selected.

- The 3215 and 3066 keyboards have both upper-case and lower-case alphabetic input. The 470V/7 console sends alphabetic input in upper case only.

- The 470V/7 console does not process immediate CCWs immediately: it does not return Channel End at the end of an initial selection. The console will return an incorrect length specification at the end of the above sequence. To suppress this indication, set CCW bit 34 (SLI).

CONSOLE SENSE DATA

Because the 470V/7 console performs the additional function of channel page passing, it uses bit 7 of byte 0 in the console sense data to indicate a channel-page-passing error. This bit is not used by a standard 3066 or 3215. All other sense data bits are as defined in the ***370/145*** and ***370/168 Functional Characteristics*** manuals (GA24-3557, revision level 4; respectively).

CHANNEL PAGE PASSING

Two extra CCWs are implemented on the 470V/7 console. These are used for transferring scan pages from the console memory to main memory during machine-check handling. Before these special CCWs can be issued, a Diagnose EB instruction must first enable channel page passing. The CCWs are summarized in table 7.

TABLE 7 CHANNEL-PAGE-PASSING CCWs

FUNCTION	HEX	EXPLANATION
Scan Page Control	'81'	Activates page-passing routine.
Scan Page Read	'82'	Transmits one scan page.

A555

INDEX

amdahl

READER'S COMMENT AND UPDATE-REQUEST FORM

If you would like to comment on this manual, or be placed on the mailing list for updates, please mail this form.

NAME ____________________

TITLE ____________________

ADDRESS ____________________

____

COMMENTS:

Do you wish to receive updates for this manual? YES □ NO □

Do you have any updates in your manual already? YES □ NO □

If yes, what is the revision date of the last one? __________

A544

YOUR COMMENTS, PLEASE...

This publication is one of a series describing the AMDAHL product line. 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 local 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
U.S.A.

ATTENTION: Department 734

Fold and staple

amdahl

AMDAHL CORPORATION
1250 EAST ARQUES AVENUE
SUNNYVALE, CALIFORNIA, 94086

PHONE 408-746-6000
TWX 910-339-9293