amdahl®

470V/5-I COMPUTING SYSTEM MACHINE REFERENCE MANUAL

amdahl®

AMDAHL 470V/5-I Computing System Machine Reference Manual

Publication Number G1012.0-01A
June 1979

REVISION NOTICE

This publication is a major revision of, and supersedes the 470 V/5 Machine Reference Manual, publication number A352-0005. Technical changes and additions to text and illustrations are indicated by a vertical bar to the left of the change.

ABSTRACT

This manual describes the functional characteristics and model-dependent features of the Amdahl 470V/5-I 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.

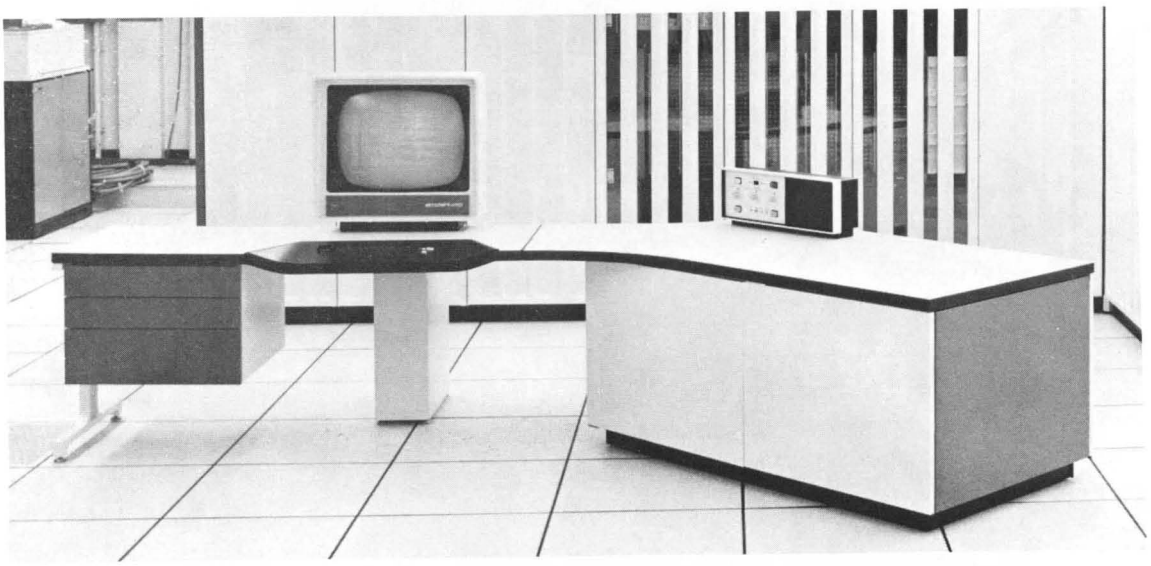
READER COMMENT FORM

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. All comments and suggestions become the property of Amdahl Corporation.

CONTENTS

FIGURES

TABLES



The Amdahl 470V/5-I Computing System

INTRODUCTION

The Amdahl 470V/5-I computing system provides powerful, high-speed, general-purpose computing capabilities for sophisticated business and scientific applications. It has a cycle time of 32.5 nanoseconds, a pipeline that executes several instructions concurrently, a high-speed buffer for fast data access, and efficient execution algorithms. The 470V/5-I also incorporates extensive error checking and recovery to optimize system reliability. Any of the channels provided with the 470V/5-I may be configured as a selector, byte-multiplexer, or block-multiplexer channel.

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 packaged on a single chip. Up to forty-two chips fit into each 7.5-inch square multi-chip carrier (MCC). The central processor and channel logic together require 51 MCCs. Because of this simplicity, the number of external connections in the system is small, and the system is easy to service and maintain.

Reliability of the 470V/5-I is enhanced by features such as hardware instruction retry, channel command retry, and main storage error checking and correction (ECC). ECC is capable of correcting any single-bit error and detecting any double-bit error.

The 470V/5-I console can determine and report the status of approximately 16,000 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.

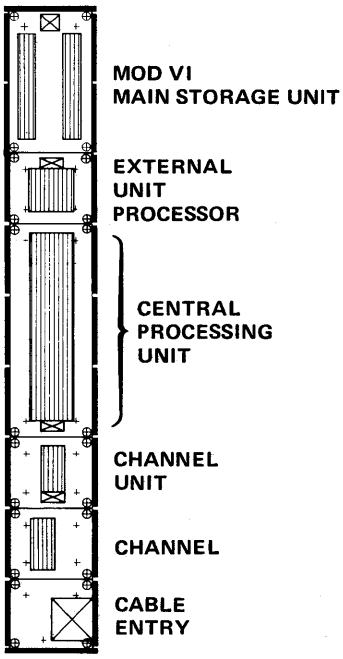
The Amdahl 470V/5-I system (see Figure 1) and the IBM System/370 are compatible within the constraints of the architecture defined in the IBM *Systems/370 Principles of Operation* manual, GA22-7000, revision level 5 (hereafter to be referred to as *System/370 Principles of Operation*). This specification 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 or Channel Command Word (CCW) execution times, and

Programs that cause deliberate machine checks.

The Amdahl 470V/5-I 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 in this manual.



STANDARD FEATURES

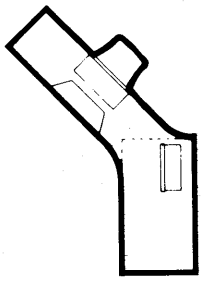
SYSTEM/370 UNIVERSAL INSTRUCTION SET	SYSTEM/370 INSTRUCTION ENHANCE-MENTS	BYTE ORIENTED OPERANDS	32,768 BYTE HIGH SPEED BUFFER
CPU TIMER	TOD CLOCK	INTERVAL TIMER	CLOCK COMPARATOR
EXTENDED CONTROL	DYNAMIC ADDRESS TRANSLATION	PROGRAM EVENT RECORDING	DIRECT CONTROL
8 I/O CHANNELS	BYTE/ BLOCK/ SELECTOR	1024 MULTIPLEXER SUBCHANNELS	INDIRECT DATA ADDRESSING

OPTIONAL FEATURES

CHANNEL TO CHANNEL ADAPTER	2-BYTE INTERFACE

A01399

Figure 1. 470V/5-I Standard Architecture



FEATURES

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

A01400

Figure 2. 470V/5-I System Console

FEATURES

EMERGENCY POWER-OFF CONTROL	POWER DISTRIBUTION

A01401

Figure 3. 470V/5-I Power Distribution Unit

SYSTEM OVERVIEW

CENTRAL PROCESSOR (CPU)

The Amdahl 470V/5-I 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 470V/5-I follows standard System/370 architecture as specified in the *System/370 Principles of Operation* manual. The standard, full System/370 Universal Instruction Set with extended-precision floating-point operations and System/370 instruction enhancements is implemented on the Amdahl 470V/5-I. Direct control is also implemented.

INSTRUCTION PIPELINE. The 470V/5-I 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 16,384 byte (16K) cache memory designed to maximize system throughput. It provides fast access to frequently used data.

TRANSLATION LOOKASIDE BUFFER. The 256-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/5-I.

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/5-I system console (see Figure 2) 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 two-megabyte increments, from four to eight megabytes. Interleaving is four-way in each two-megabyte unit. If an uncorrectable error develops in a two-megabyte section, that section can be configured out of the system, leaving the rest of main storage available to the CPU . Access to main storage is controlled by the Storage Unit (S-Unit).

CHANNELS (C-UNIT)

The Amdahl 470V/5-I system has 8 standard inboard channels which may be installed as a selector, byte-multiplexer, or block-multiplexer channel. Additional channels may be installed in 4-channel increments up to a maximum of 16 channels. The channels are implemented by the Channel Unit (C-Unit), and except for possible storage access conflicts, they operate independently of the CPU (see Figure 1). A total of 1,024 sub-channels may be assigned to the multiplexer channels in groups of 64, 128, or 256.

POWER DISTRIBUTION UNIT

The power distribution unit (see Figure 3) distributes 400 Hz power to the 470V/5-I system and provides emergency power off and thermal monitoring. It also provides 60 Hz power for standard utility plugs and fans.

OPTIONAL FEATURES

CHANNEL-TO-CHANNEL ADAPTER. This option provides the synchronization necessary to interconnect channels between two CPUs. It may be attached to a selector or a block-multiplexer channel and uses one channel control unit position on each CPU. When interconnecting an Amdahl 470V/5-I 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 all selector and multiplexer 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 program status word (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 the other units.

I-UNIT ORGANIZATION

The I-Unit functions (see Figure 4) are organized into the following parts:

- PROCESS CONTROL. The process control resolves interrupt priorities and selects the next operation to be performed.
- INSTRUCTION FETCH. The instruction fetch logic requests sequential bytes of the instruction stream from main storage (S-Unit). These bytes are then stored in a buffer, ready to be divided into discrete instructions by the instruction select logic.
- INSTRUCTION SELECT. The instruction select logic dispatches instructions to the pipeline. If the process control continues to execute the sequential instruction stream, instructions are requested from the instruction fetch buffer. If an interrupt or branch occurs, the instruction select logic selects the appropriate hardware instruction from the process control.
- PIPELINE. The pipeline is a major factor in the high performance of the Amdahl 470V/5-I. 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 require the facilities of the S-Unit.

PIPELINE OVERLAP AND BRANCHING

By overlapping all the functions described above, 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. The cycles required to execute a typical instruction, including instruction fetch, are summarized in Table 1. Most instructions use the sequence shown in the table, although an instruction may duplicate some cycles and omit others, depending on the type and complexity of the instruction. The pipeline begins executing each

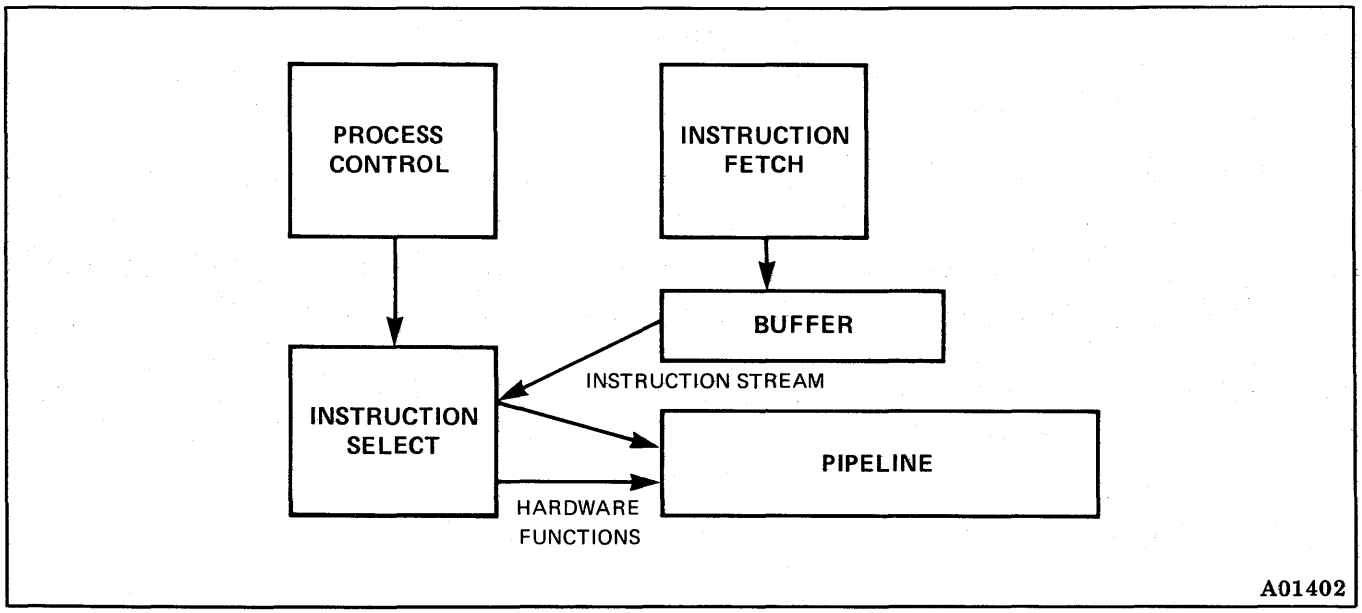


Figure 4. I-Unit Organization

Table 1. I-Unit Instruction Sequence

OPERATION	INSTRUCTION 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

A01416

Table 2. Pipeline Overlap

INSTRUCTION SEQUENCE	INSTRUCTION CYCLES
1	D R A B1 B2 E1 E2 C W
2	D R A B1 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
	MACHINE CYCLES

A01417

instruction before it has finished executing the previous one. Because of this feature, the 470V/5-I pipeline is capable of processing several instructions simultaneously. Tables 1 and 2 illustrate the concept of pipeline overlap.

To handle selection of branch instructions, or conditions where prefetching and overlap of instructions is impossible because of possible multiple execution 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 completed execution. Thus, the I-Unit is able to pick the correct branch path immediately, and the correct instruction stream proceeds down the pipeline, using only three machine cycles more than a sequential instruction.

The I-Unit continues to execute sequential instructions that are already in the pipeline if the branch is not taken. Rather than resort to an expensive and massive duplication of hardware to follow multiple branch paths, the 470V/5-I branching algorithms complement an optimal pipeline organization to produce significant performance in the execution of machine object instructions.

HARDWARE INSTRUCTION RETRY

To enhance total systems availability and reliability, most instructions are retriable. This is accomplished with the total system design of the 470V/5-I Instruction Unit by delaying any updates to architectural registers until the last cycle of instruction execution. Thus, 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/5-I checking and correction mechanisms, while providing maximum recovery capability. In addition, the pipeline concept is extended to include enhancements to machine integrity.

INTERRUPT HANDLING

All interrupts in the 470V/5-I are precise. When an interrupt occurs, the process control removes any instructions following the current instruction from the pipeline, and 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 I-Unit can reinitiate the interrupted instruction stream in the usual manner following execution of the interrupt-handling routine.

EXECUTION UNIT

E-UNIT FUNCTIONS

The Execution Unit (E-Unit) performs 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 (LUCK). The Logical Unit and Checker performs these functions:

- Executes 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. Output of the shifter is 36 bits.

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/5-I 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 has been set.

ERROR CHECKING

Execution results are checked in the 470V/5-I. This checking includes parity checks for most operations and a more comprehensive residue arithmetic check for multiplication and division operations.

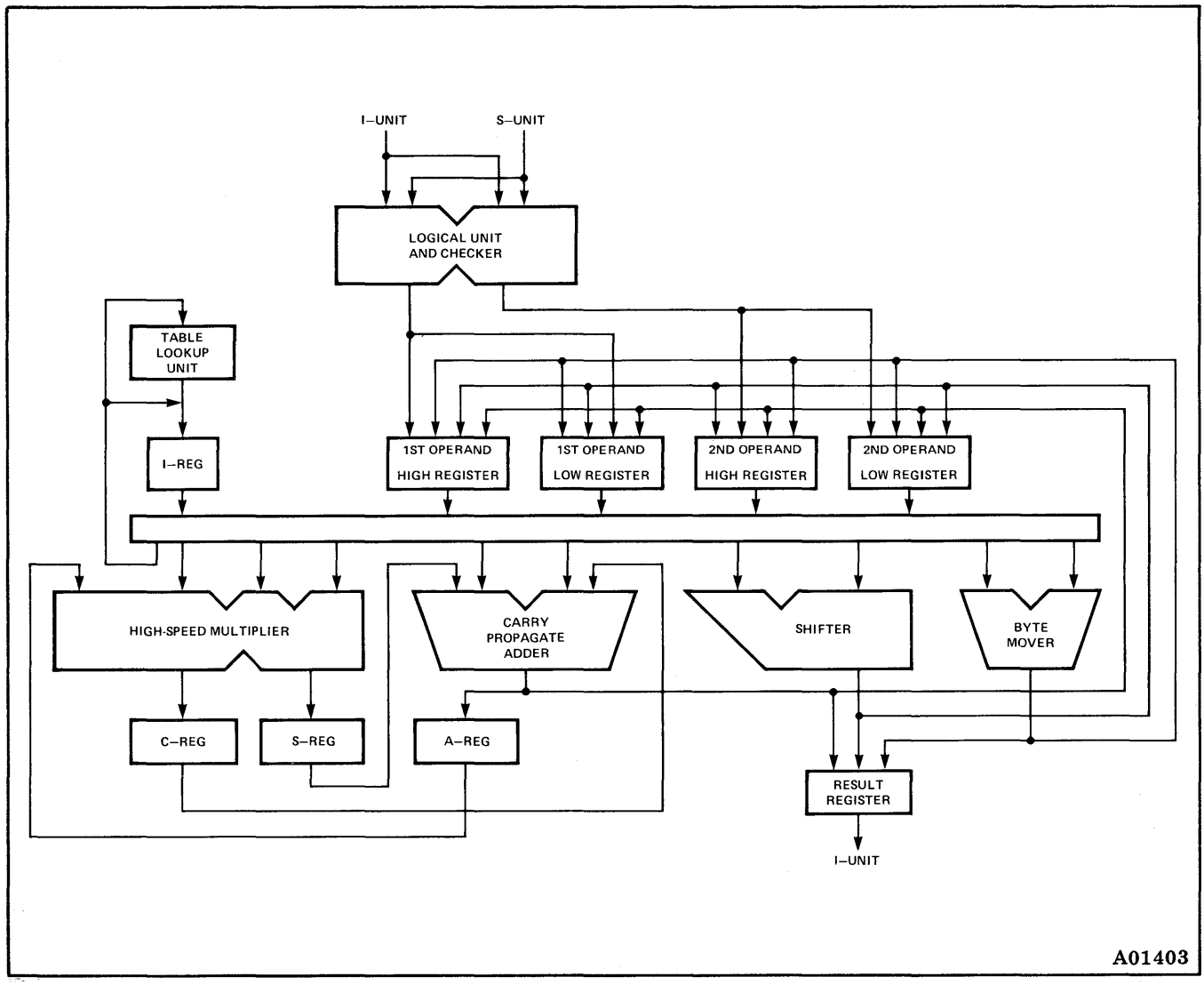


Figure 5. E-Unit Organization

CHANNEL UNIT

C-UNIT FUNCTIONS

The Channel Unit (C-Unit) implements the 8, 12, or 16 inboard channels of the 470V/5-I. Except for occasional memory-access conflicts, these channels operate independently of the CPU. The channels may be configured as a selector, byte-multiplexer, or block-multiplexer channel.

The C-Unit 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 (see Figure 6). The C-Unit performs the I/O commands defined in the *System/370 Principles of Operation* manual 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 inboard 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). Figure 6 illustrates the organization of these parts.

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 (CBS) and the Remote Interface Logic (RIL) and controls channel frame operations. The CICL has two ports into the SCS and examines two

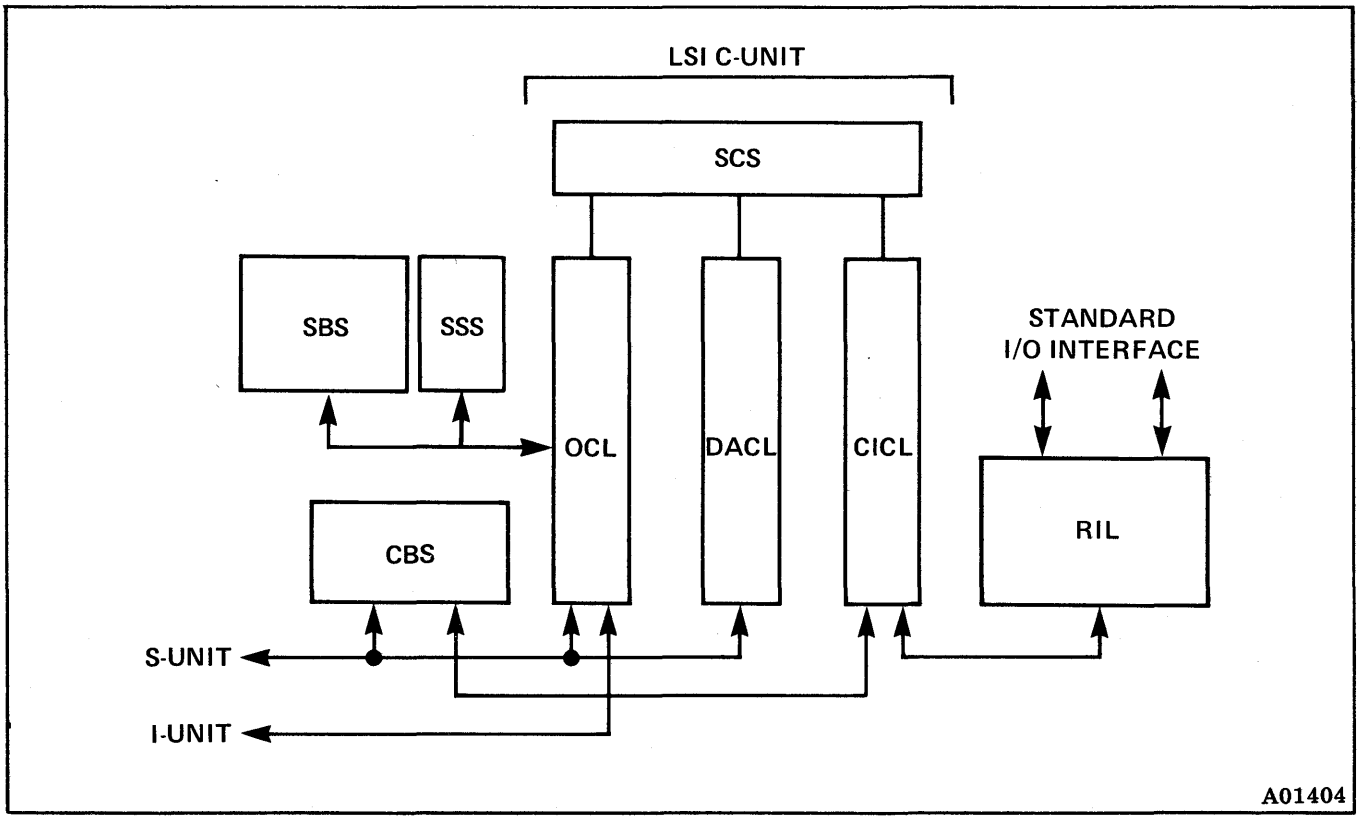


Figure 6. C-Unit Organization

channels every cycle. Every two cycles, it accesses the CBS for the highest-priority channel of the four it has just examined. The priority is determined internally by the CICL.

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 always selects 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 with the C-Unit. It sets up counts, flags, and data transfer addresses in the C-Unit buffers (normally the CBS), and it translates Channel Command Words (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 CBS contains a buffer for each channel. The CICL transfers data to or from the CBS on even cycles; the DACL and OCL share odd cycles. The CICL transfers one or two bytes per access; 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 in the Subchannel Status Store (SSS).

INDIRECT DATA ADDRESSING

Channel Indirect Data Addressing (IDA), as described in the *System/370 Principles of Operation* manual, is fully implemented in the Amdahl 470V/5-I computing 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/5-I channel can be configured as a block multiplexer, byte multiplexer, or selector channel. 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 the section entitled, "Subchannel Assignment."

CHANNEL BANDWIDTH

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

SELECTOR. The maximum data rate for a selector channel is approximately 1.9 megabytes per second. An optional two-byte interface doubles this rate.

BLOCK MULTIPLEXER. The maximum data rate for a standard, single-byte, block-multiplexer channel is approximately 1.9 megabytes per second. An optional two-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. In burst mode, the rate is approximately 1.9 megabytes per second.

CHANNEL QUADRANTS. Channels on the 470V/5-I are arranged into four quadrants. Because the channels within a quadrant share CICL and CBS resources, the total data rate for a quadrant may not exceed approximately 3.65 million transfers per second. (A transfer is one byte on a one-byte interface and two bytes on a two-byte interface.) See Figure 7 for an illustration of the channel quadrants for 8 channels.

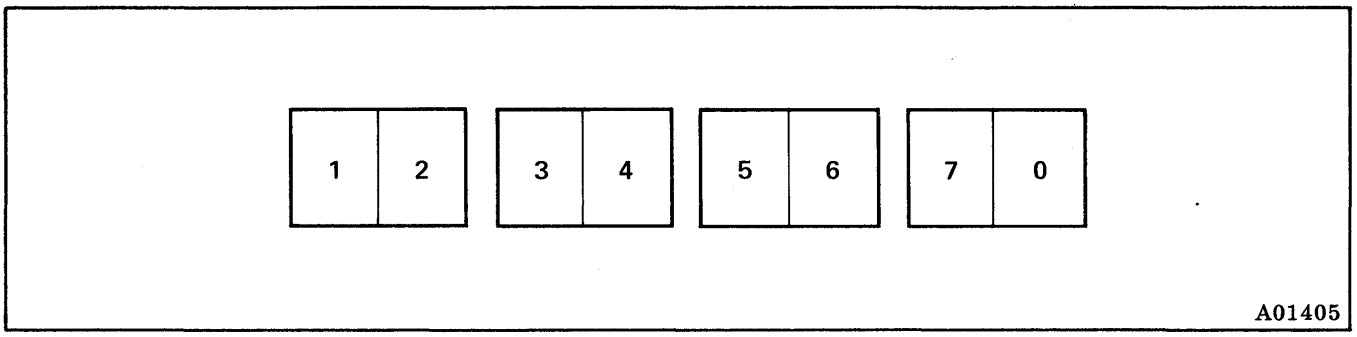


Figure 7. Channel Quadrants for Eight Channels

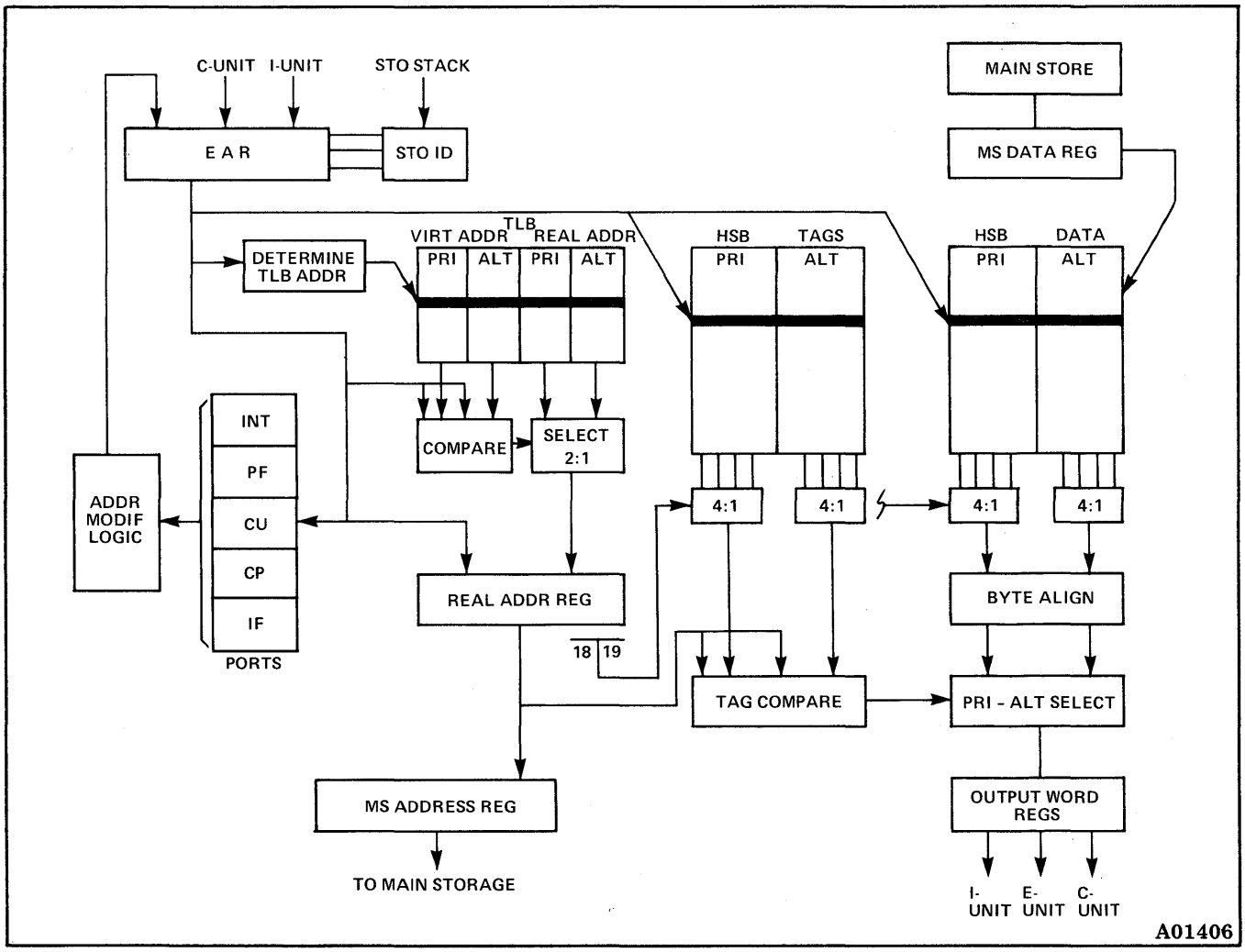


Figure 8. S-Unit Organization

STORAGE UNIT

S-UNIT FUNCTIONS

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

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 stack (STO stack). Figure 8 summarizes the organization of the S-Unit.

HIGH-SPEED BUFFER (HSB). 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 (TLB). The TLB is a 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. The STO stack saves the data from control registers 0 and 1 that define the current segment table. Each TLB entry is associated with a 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.

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 and loads the 32-byte storage line containing the requested bytes into the HSB. The HSB is then reaccessed. Refer to Figure 8.

HIGH-SPEED BUFFER (HSB)

HIGH-SPEED BUFFER ORGANIZATION

The 470V/5-I HSB is a 16,384 byte (16K) set-associative memory. It is divided into two parts: the primary half and the alternate half. Each half contains 256 32-byte lines.

The lines in each HSB half are divided into four groups. The high-order 3 bits of the buffer line-address uniquely define each group; the low-order 6 bits define corresponding locations within each group. Figure 9 provides an illustration of HSB organization.

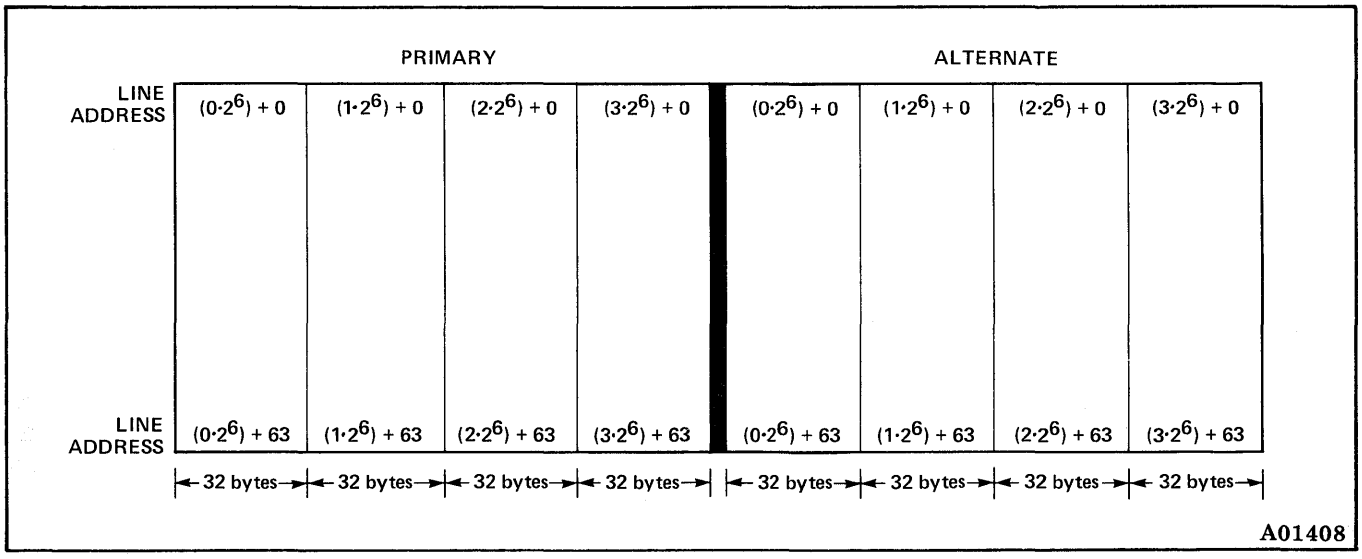


Figure 9. High-Speed Buffer Organization

TWO KILOBYTE PAGES

For system control programs using 2K pages, it is necessary to operate the HSB in 16K mode. In this mode, each buffer half contains 256 32-byte lines. The low-order six bits of the buffer line address define corresponding locations within each group, leaving an extra bit available for real-address translation.

HIGH-SPEED BUFFER TAG

A tag is associated with each line in the HSB. The tag contains four fields that identify and protect data. These fields are called the line identifier, key, reference, and control. Figure 10 illustrates these four fields.

The line identifier field contains bits 8-17, 19, and 20 of the real address. The key field contains five protection key bits, a parity bit, and a check bit. The reference field contains two bits: R1 and R2. R1 specifies whether a CPU or a channel access brought the line into the buffer. R2 specifies whether the CPU was in problem state or supervisor state when the line was brought in. The control 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. In the primary half of the buffer, there is one more bit that can be thought of as part of the tag: the hot/cold bit. It indicates which half of a given pair of buffer locations, primary or alternate, was referred to recently.

FINDING A LINE IN THE HSB

When the S-Unit references the HSB, it first forms a pointer into the buffer using virtual address bits 20-26. This pointer defines eight corresponding lines, four in the primary half and four in the alternate half (see Figure 9 and Figure 11).

As soon as the real address bits are determined by dynamic address translation (DAT), the S-Unit uses real address bits 18 and 19 to select one line of four in each half of the buffer. A real address with the bits 18-26 can be placed only in these two buffer line addresses. Because this line address points into both halves of the buffer, the S-Unit must decide which half contains the requested bytes. To do this, it compares the line identifier fields of both tags with the corresponding real address bits (8-17, 19, 20).

LINE IDENTIFIER	KEY	REF	CNTRL	H/C
(Real Address Bits)	01234PC	R1 R2	0 1 2	

Figure 10. High Speed Buffer Tag A01407

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

FETCHING A LINE FROM THE HSB

If the S-Unit is fetching a line from the HSB, it finds the two lines and performs a tag compare. If one of the tags matches the real address bits, the primary/alternate 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 specific HSB locations, the S-Unit must decide which of the lines already at a given location (in either the primary or alternate buffer half) to replace with the new line. If a line is invalid, it is replaced immediately. If no line is invalid, the S-Unit looks at the hot/cold bit in the primary tag to identify the least recently used line and replaces it. If the line to be replaced is modified, it is written to main storage before 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 control field of the tag to show that the line is modified.

HSB RECONFIGURATION

The S-Unit Operating State Registers (OSR's) control the HSB configuration. The system console initializes these registers. If a buffer error occurs, the HSB is reconfigured by changing the OSR's . The quadrant in error is disabled, and the rest of the HSB remains available to the system.

DYNAMIC ADDRESS TRANSLATION (DAT)

The 470V/5-I can perform Dynamic Address Translation (DAT) when in EC mode. Virtual addressing in the 470V/5-I operates as defined in the *System/370 Principles of Operation* manual.

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 (STO) stack contains 32 entries and 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, and each entry has a unique STO ID. One other bit, called the flipper bit, distinguishes old STO stack entries from new ones. The STO address and the flipper bit together constitute a STO ID.

Like the high-speed buffer, the translation lookaside buffer (TLB) is divided into a primary half and an alternate half. Each half has 128 locations. A virtual address will map into one location in the alternate half based on 7 bits of the address and into another location in the primary half based on a hash of the address. The TLB address is a mapping of the current STO ID and the virtual address.

STO STACK ENTRIES

Whenever the values of control registers 0 and 1 change, 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 into which the virtual address maps. The TLB has a hot/cold bit, similar to the one in the HSB, for deciding which entry — primary or alternate — to replace. The S-Unit saves the new translation with the STO ID of the current value of control registers 0 and 1 in the location least recently used.

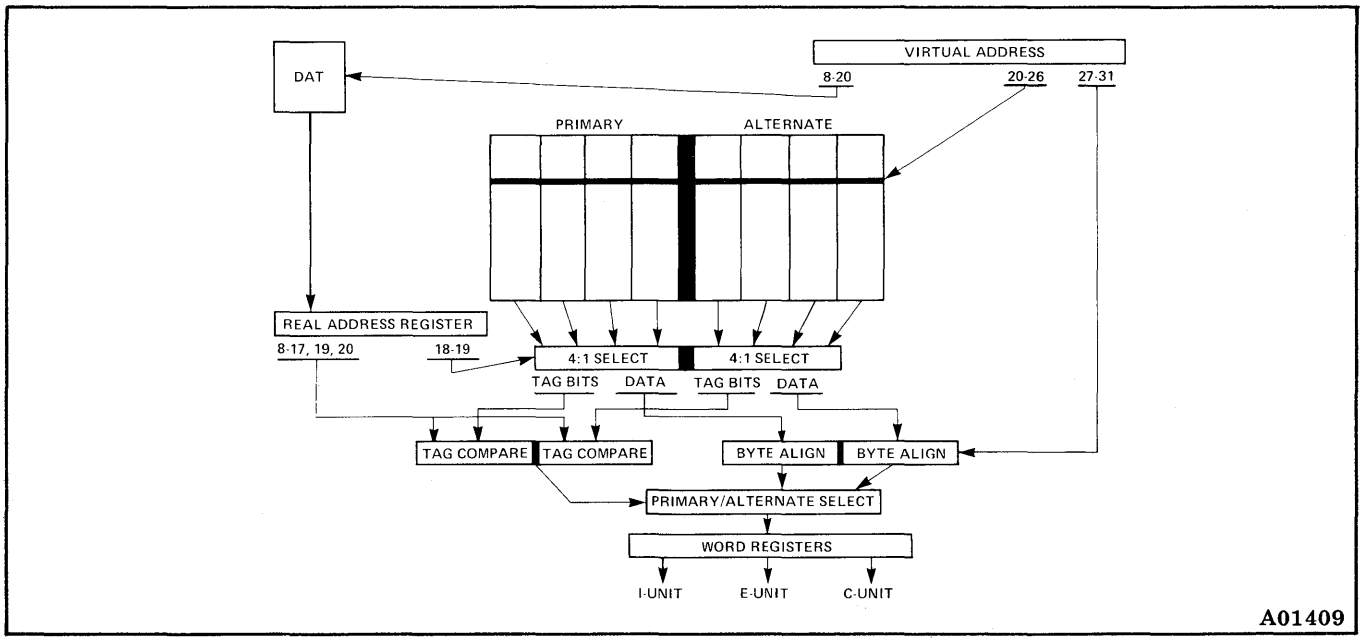


Figure 11. High-Speed Buffer Operation

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 compares 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 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 16 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 16 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 control field of the high-speed buffer tag.

SYSTEM CONSOLE

CONSOLE FUNCTIONS

The 470V/5-I System Console provides communication with the 470, diagnostic information on the hardware, intermediate storage for machine check logouts, and usage metering.

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. The console is shown in Figure 12.

COMMUNICATION. The console provides all standard communication between the 470V/5-I processor and the operator. It emulates a 3066 or 3215 operator's console, performs hardware functions such as IPL, reset, and clear, and displays diagnostic messages and the contents of registers, latches, and storage.

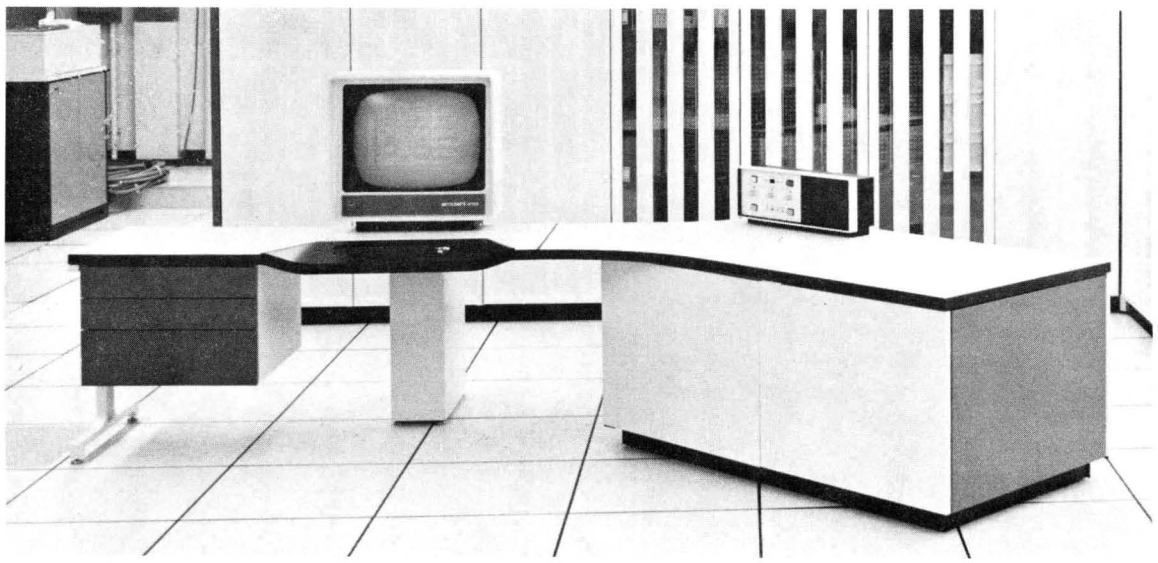


Figure 12. 470V/5-I System Console

DIAGNOSTIC INFORMATION. The 470V/5-I console provides formatted displays of approximately 16,000 latches within the 470V/5-I 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 provides a continuous machine status display at the top of the CRT screen. This display summarizes the current state of the 470V/5-I system.

MACHINE CHECK LOGOUT STORAGE. The 470V/5-I system console stores machine check logout information in its attached disk. This makes it possible to save several scan pages at the time of a machine failure and to later display these pages at the console.

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/5-I system console includes a CRT display screen, a keyboard, a standard channel interface, a computer-to-console interface (CCI), an independent processor, and a modem.

The standard channel interface is used when the 470V/5-I 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 470V/5-I.

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, 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/5-I. Through the modem, the Amdahl central diagnostic facility, AMDAC, can diagnose hardware problems from Amdahl headquarters.

CONSOLE OPERATION

The 470V/5-I console operates in one of three modes: device support mode, hardware command mode, and maintenance 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.

MAINTENANCE MODE. The Amdahl field engineering staff uses maintenance mode to maintain and diagnose the 470V/5-I 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 model-dependent results on the Amdahl 470V/5-I. They are: STORE CPU ID (STIDP) and STORE CHANNEL ID (STIDC).

STORE CPU ID (STIDP)

STIDP stores model-dependent data at the double word addressed by the second operand. Table 3 shows the data stored for the 470V/5-I.

STORE CHANNEL ID (STIDC)

STIDC stores channel-dependent data at decimal location 168. Because the 470V/5-I 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.

MACHINE CHECK CONDITIONS

The Amdahl 470V/5-I 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 13) are grouped into two categories: repressible and exigent. These are defined in the *System/370 Principles of Operation* manual.

REPRESSIBLE CONDITIONS

Repressible conditions include 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.)

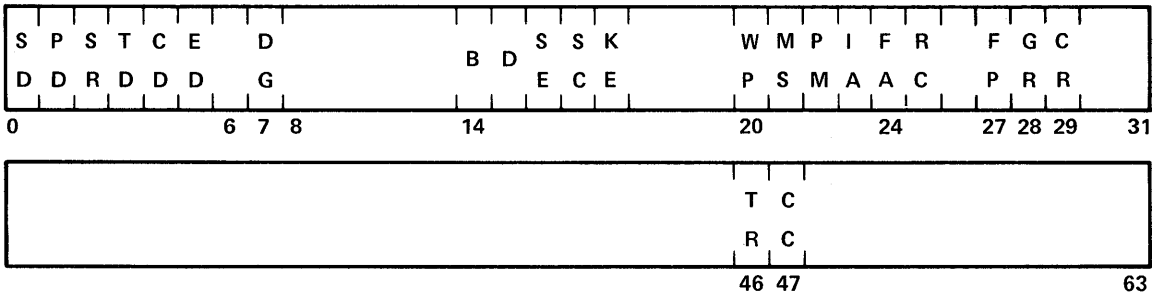
EXIGENT CONDITIONS

Exigent conditions include system damage conditions, multi-bit storage errors, protection key parity errors, unretrievable internal data transfer errors, move out parity errors, and instruction processing damage conditions (if retry is unsuccessful or impossible).

Table 3. Store CPU ID (STIDP) Instruction

FIELD	BITS	VALUE STORED
Version Code	0-7	05
Serial Number	8-31	unique serial number
Model Number	32-47	0470
Maximum MCEL Length	48-63	0000

A01418



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

A01411

Figure 13. Machine Check Interruption Code

A move out parity error is a parity error in data being moved from the high-speed buffer to main storage. A protection key parity error makes it impossible to establish whether protection applies to the 2048 storage bytes associated with the key.

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/5-I 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 (HIR). 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 (ECC). An ECC field is associated with each 16-byte section of main storage. This field contains sufficient information to correct any single-bit error within the 16 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 channel command word (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 (decimal) is reserved for machine check logouts. The Amdahl 470V/5-I 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 two words starting at location 252. The rest of the area, locations 260-351, is reserved.

FAILING STORAGE ADDRESS (FSA). The FSA indicates the byte or block in which the error occurred. For a correctable storage error, bits 0-3 of the FSA contain the failing bit address and bits 8-31 contain the failing byte address. For an uncorrectable storage error, bits 8-31 of the FSA point anywhere within the failing 16-byte ECC block. For an uncorrectable protection key error, bits 8-31 of the FSA point anywhere within the 2,048-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 interruption code is set to zero.

REGION CODE. The region code specifies which part of the machine detected the error. Table 4 defines the region code bits.

MACHINE CHECK EXTENDED LOGOUT (MCEL)

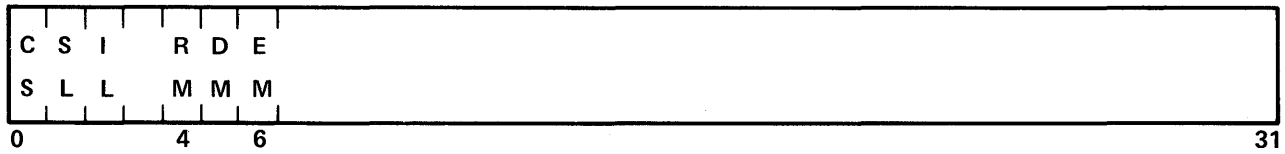
The 470V/5-I performs a machine check extended logout (MCEL) when a machine check occurs and the mask bits of control register 14 are set to allow the logout. Figure 14 defines the mask bits of control register 14. The logout on a 470V/5-I includes a set of scan pages that record the state of approximately 16,000 latches in the system. These are the same scan pages that can be displayed at the console in hardware command mode. The console processor performs the logout and saves the information in 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

Table 4. Region Code Bits

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 A01419



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

A01412

Figure 14. Control Register 14 — Machine Check Control Register

or through the computer-to-console interface. Either will transfer the logout from the console to main storage.

CHANNEL PAGE PASSING. To retrieve scan pages through the channel, issue the DIAGNOSE EB instruction x83EB to enable channel page passing. Then, issue the special console CCWs, x"81" and x"82" to retrieve scan pages. For detailed information, refer to the section on "Console Channel Programming."

COMPUTER-TO-CONSOLE INTERFACE. To retrieve scan pages through the CCI, issue the DIAGNOSE EB instruction x83EB with the appropriate parameters. This instruction retrieves three pages at a time; it must be repeated until all pages have been transferred.

DIAGNOSE EB. The operation of the DIAGNOSE EB instruction varies in accordance with the engineering revision level of the system. This instruction is explained in the section entitled, "Channel Page Passing CCWs."

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/5-I and stores as zeros.

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

CHANNEL LOGOUT

EXTENDED CHANNEL LOGOUT

I/O Extended Logout (IOEL), as defined in the *Systems/370 Principles of Operation*, is fully implemented on the 470V/5-I system. Figure 15 provides a diagram of the 470V/5-I 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 8, 16, or 32, depending on whether 64, 128, or 256 subchannels are assigned. Figure 16 diagrams IOEL words 0 to 3.

LIMITED CHANNEL LOGOUT

The Amdahl 470V/5-I channels only perform an extended logout of a channel error. Some programs require a Limited Channel Logout (LCL) for this function. This section describes how to create an LCL from an IOEL.

To determine whether an error has occurred, examine the three channel status word (CSW) error bits: Channel Control Check (CCC), Interface Control Check (IFCC), and Channel Data Check (CDC). If one of these bits is on, an error has occurred and an I/O Extended Logout has been written.

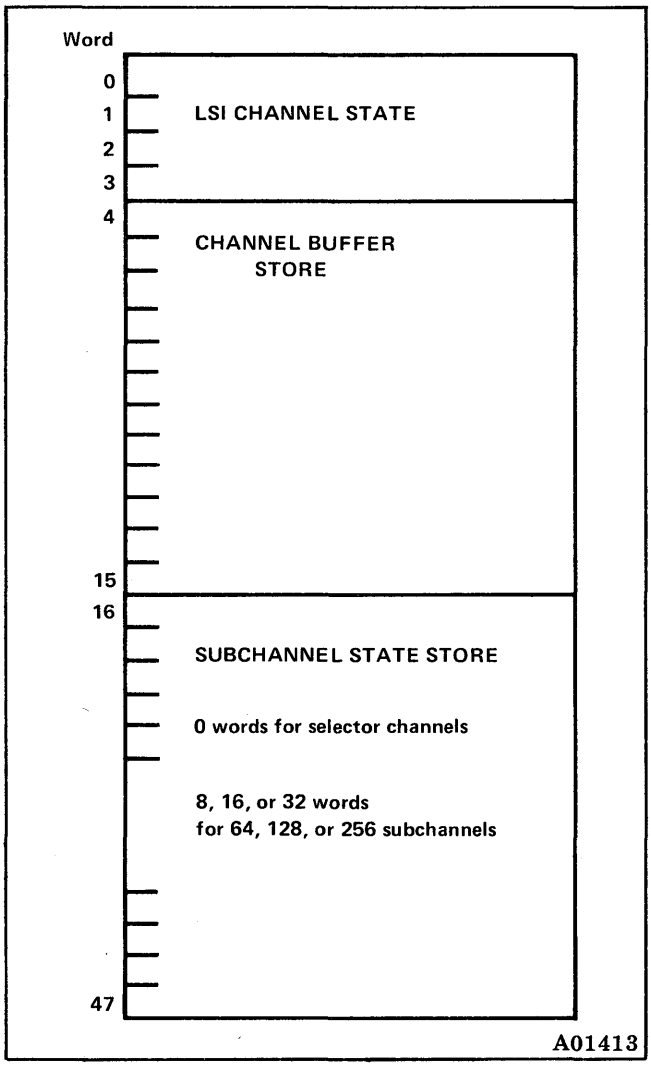


Figure 15. I/O Extended Logout

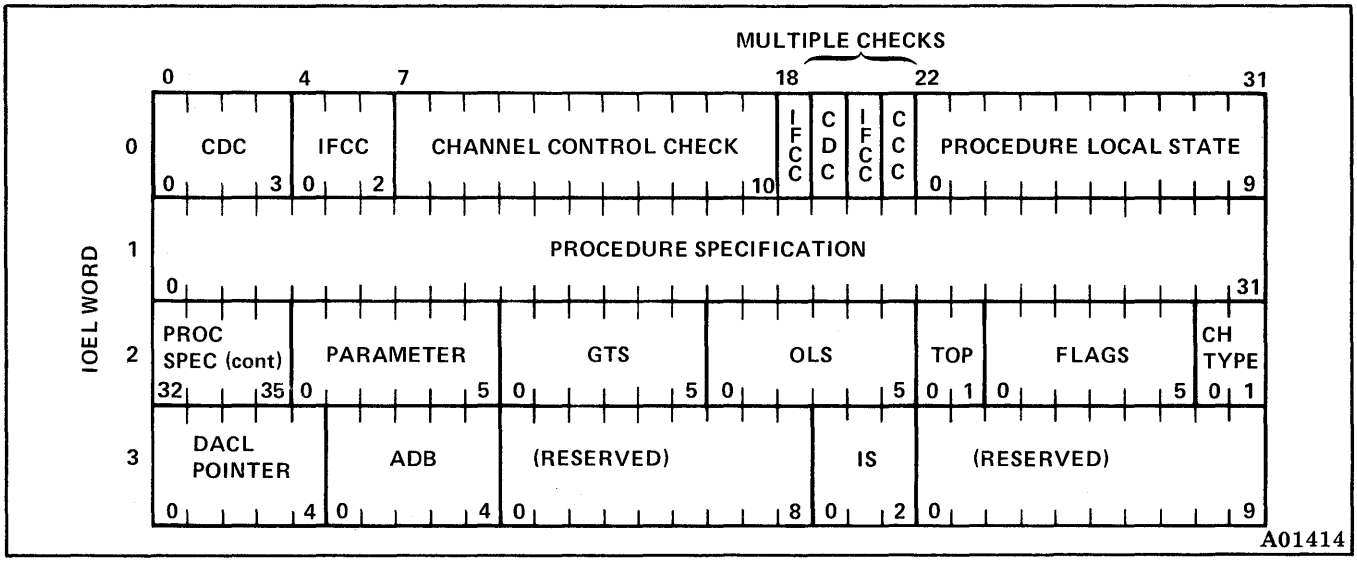
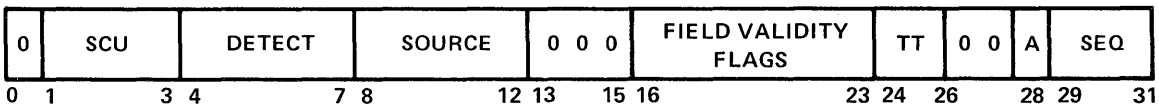


Figure 16. LSI Channel State — IOEL Words Zero to Three



FUNCTION	BITS	VALUE IF CCC	VALUE IF IFCC	VALUE IF CDC
Reserved	0	0	0	0
SCU ID	1-3	000	000	000
Detect				
CPU	4	0	0	0
Channel	5	C1|CCC(6)|CCC(8)|CCC(9)	1	¬CDC(0)
Storage Control Unit	6	C4|CCC(7)	0	CDC(0)
Storage Unit	7	0	0	0
Source				
CPU	8	0	0	0
Channel	9	C1|CCC(6)|CCC(8)|CCC(9)	0	¬[CDC(0)|TOP(0) & CDC(3)]
Main Storage Control	10	CCC(4)|CCC(7)	0	0
Main Storage	11	CCC(5) & ¬CCC(4)	0	CDC(0)
Control Unit	12	0	1	TOP(0) & CDC(3)
Reserved	13-15	000	000	000
Validity				
Interface Address	16	0	0	0
Reserved	17-18	00	00	00
Sequence Code Valid	19	¬CCC(6) & SC6	SC6	1
Unit Status	20	C9 & USV	USV	1
Command Address & Key	21	C1|C4|C9|[CCC(8) & CAV]	1	1
Channel Address	22	¬CCC(6)	1	1
Device Address	23	CCC(8) & ¬[PS(13)|PS(14)]|[C1|C4|C9]	¬[IFCC(1) & GTS=33]	1
Type of Termination	24-25	TERM	10	01
Reserved	26-27	00	00	00
I/O Error Alert	28	0	IFCC(2)	0
Sequence Code	29-31	SEQCODE	SEQCODE	011

A01415

Figure 17. Limited Channel Logout (LCL) Word

To determine if an LCL is valid, examine the IOEL. Table 5 indicates which check fields in the IOEL are valid. The notations and terms used in Table 5 are explained in Table 6 and Table 7, respectively.

After determining that an LCL is valid, use Figure 17 to calculate the value of each LCL bit. Figure 17 shows three formulas: one for Channel Control Checks, one for Interface Control Checks, and one for Channel Data Checks. Table 6 and Table 7 provide explanations for the notations and terms, and Table 8 lists the possible variables for each bit in an LCL word.

Table 5. Valid LCL

```
VCCC ← ¬MULTCCC & (CCC=1 | CCC=2 | CCC=4 |
          CCC=8 | CCC=16 | CCC=32 | CCC=64 |
          CCC=128 | CCC=256 | CCC=512 | CCC=1024)
VCDC ← ¬MULTCDC & (CDC=1 | CDC=2 | CDC=4 |
          CDC=8)
VIFCC ←¬MULTIFCC & (IFCC=1 | IFCC=2 | IFCC=4)

VALID LCL ← (VCC+VCDC+VIFCC)=1
```

A01420

Table 6. LCL Notations

SYMBOL	FUNCTION
+	Algebraic ADD
=	EQUAL
≠	NOT EQUAL
≥	GREATER THAN OR EQUAL TO
≤	LESS THAN OR EQUAL TO
¬	Boolean NOT
&	Boolean AND
|	Boolean OR
←	ASSIGN VALUE

These symbols are listed in hierarchic order (e.g., perform all AND operations in an expression before OR operations).

A01421

Table 7. LCL Terminology

TERM	DEFINITION
CAS	Command Accept Status.
CAV	Command Address Valid.
CCC	Channel Control Check (IOEL bits 7-17). CCC(X) represents bit x of the CCC field.
CDC	Channel Data Check (IOEL bits 0-3). CDC(x) represents bit x of the CDC field.
CSW	Channel Status Word. CSW(x) represents bit x of the CSW.
C1	Intermediate CCC variable.
C4	Intermediate CCC variable.
C9	Intermediate CCC variable.
DPP	Pointer variable. DPP has 5 bits.
GTS	IOEL field (Global Transfer State). Values for GTS are given in OCTAL.
IFCC	Interface Control Check (IOEL bits 4-6). IFCC(x) represents bit x of the IFCC field.
IS	IOEL field (Interrupt State).
MOD	Modulo function. MOD x, y is the remainder of x divided by y.
MULTCCC	Multiple CCC (IOEL bit 21).
MULTCDC	Multiple CDC (IOEL bit 19).
MULTIFCC	Multiple IFCC (IOEL bit 20).
OLS	IOEL field (OCL Local State).
PS	Procedure Specification field of IOEL. PS(x) represents bit x of the PS field.
SC0	SC0 through SC6 are intermediate variable.
SEQCODE	Sequence Code.
TOP	IOEL field (Transfer Operation).
TERM	Type-of-Termination intermediate variable.
UPP	Updated Pointer.
USV	Unit Status Valid.
VCCC	Valid CCC.
VCDC	Valid CDC.
VIFCC	Valid IFCC.

A01422

SUBCHANNEL ASSIGNMENT

470V/5-I SUBCHANNELS

A total of 1,024 subchannels may be assigned to the multiplexer channels on a 470V/5-I system. They can be assigned in multiples of 64, 128, or 256. The implicit subchannel of a selector channel is not a part of the 1,024 total. Selector subchannels are not available on 470V/5-I 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. On multiplexer channels with 128 subchannels, only the low-order 7 bits of the device address are significant. For multiplexer channels with 64 subchannels, only the low-order 6 bits of the device address are significant. For example, a multiplexer channel with 64 subchannels makes no distinction between the unit addresses 301, 341, 381, and 3C1. Therefore, the high-order address bits form redundant address groups.

Table 9 illustrates the device address groups assigned to each multiple of subchannels. The device addresses within a group are unique, but all groups associated with the same subchannel addresses duplicate the same address range.

Table 8. LCL Variables

```
VARIABLES
C1 ← CCC(0) I CCC(1) I CCC(2) I CCC(3)
C4 ← CCC(4) I CCC(5)
C9 ← CCC(7) I CCC(9)
USV ← GTS=11 I GTS=12 I GTS=13 I GTS=14 I GTS=17 I GTS=27 I GTS=35 I GTS=37 I GTS=65
CAV ← ¬[PS(0) I PS(2) I PS(6) I PS(7) I PS(12) I PS(15) I PS(16) I PS(35)]
CAS ← ¬[CSW(32) I CSW(34) I CSW(35) I CSW(38) I CSW(39)]
DPP ← MOD [(DACLP+ADB, 32]
UPP ← [TOP=01 & DPP≠0] I [TOP(0) & (DAC≠0 I ADB≠0)]

SEQUENCE CODE
SC0 ← TOP=00
SC1 ← GTS=63 I GTS=64 I GTS=65
SC2 ← GTS=35 & CAS
SC3 ← GTS=10 I [(GTS=66 I GTS≥70) & UPP]
SC4 ← [GTS=40 I GTS=44 I GTS=45 I 50≤GTS≤62]
      I [GTS=00 & OLS(0) & OLS(1) & ¬OLS(2) & ¬OLS(3)]
      I [GTS=35 & ¬CAS]
SC5 ← [GTS=06 I GTS=07 I GTS=36 I 20≤GTS≤34]
      I [GTS≠00 & OLS(0) & ¬OLS(1) & OLS(2)]
      I [(GTS=66 I GTS=67 I GTS≥70) & ¬UPP]
SC6 ← SC0 I SC1 I SC2 I SC3 I SC4 I SC5

SEQCODE ← 000 if SC0
SEQCODE ← 001 if SC1
SEQCODE ← 010 if SC2
SEQCODE ← 011 if SC3
SEQCODE ← 100 if SC4
SEQCODE ← 101 if SC5
SEQCODE   Valid if SC6

TYPE OF TERMINATION
TERM ← 11 if (IS=010) & VCCC
TERM ← 10 if (IS≠010) & VCCC
```

A01423

On unshared subchannels, the subchannel address is equal to the significant bits of the device address as defined above (the low-order 6, 7, or 8 bits). On shared subchannels, the subchannel address is calculated from the device address.

SHARED SUBCHANNELS

Subchannels can be shared on channels with 64 or 128 subchannels. Channels with 256 subchannels cannot have shared subchannels, because 256 gives each device its own subchannel. Table 10 shows how device addresses map into shared subchannels.

Table 9. Device Address Groups

NUMBER OF SUBCHANNELS	NUMBER 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

A01424

Table 10. Device Address Assignments — Shared Subchannels

TOTAL SUBCHANNELS	UNSHARED		SHARED		DEVICE ADDRESS GROUPS
	NUMBER OF SUBCHANNELS	DEVICE ADDRESS RANGE	NUMBER OF SUBCHANNELS	SUBCHANNEL ADDRESS	
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
				02	A0→AF
				03	B0→BF
				04	C0→CF
				05	D0→DF
				06	E0→EF
				07	F0→FF
256	256	00-FF	0	None	

A01425

On a channel with 64 subchannels, subchannels 00 to 03 are shared, and subchannels 04 to 3F are unshared. Device addresses 00 to 3F are synonymous with subchannel addresses 00 to 3F. Device addresses greater than 3F map into subchannels 00 to 03 according to the value of bits 2 and 3 of the device address. For example, 4F (01*00* 1111) will map into subchannel 0, and 5F (01*01* 1111) will map into subchannel 1.

On a channel with 128 subchannels, subchannels 00 to 07 are shared, and subchannels 08 to 7F are unshared. Device addresses 00 to 7F are synonymous with subchannel addresses 00 to 7F. Device addresses greater than 7F map into subchannels 00 to 07 according to the value of bits 1, 2, and 3 of the device address. For example, address F8 (*1111* 1000) will map into subchannel 7.

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.

CONSOLE CHANNEL PROGRAMMING

CHANNEL COMMAND WORDS (CCWs)

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 operator console, the 470V/5-I console responds to the CCWs defined in the IBM *370/168 Functional Characteristics* manual, GA22-7010, revision level 5. These CCWs are summarized in Table 11. The two bytes of console sense data for a 3066 are shown in Table 12.

The 35-line console display area appears below the status display on the CRT screen. Maximum data transfer includes up to 2,803 bytes, although the screen holds only 2,800 bytes.

Table 11. 3066 Channel Command Words (CCWs)

FUNCTION	HEX	EXPLANATION
NOP	'03'	No operation. This CCW sets the incorrect-length indication.
Sense	'04'	Reads two bytes of sense data (see Table 12).
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.

A01426

Table 12. 3066 Console Sense Data

BYTE	BIT	FUNCTION
0	0	Command reject
	1	Reserved
	2	Bus out check
	3	Equipment check
	4	Data check
	5	Reserved
	6	Buffer address check
	7	Channel-page-passing error
1	-	Reserved

A01427

3215 EMULATION

When emulating a 3215 operator console, the 470V/5-I console responds to the CCWs defined in the IBM *370/145 Functional Characteristics* manual, GA24-3557, revision level 9. These CCWs are summarized in Table 13. The single byte of console sense data is shown in Table 14.

The 470V/5-I 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 area.
- The 470 line length is 80 characters rather than 132. Messages exceeding 80 characters 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/5-I 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 is Channel End, Device End, and Unit Check, and the sense returned is Intervention Required.

Table 13. 3215 Channel Command Words (CCWs)

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 14).
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.

A01428

Table 14. 3215 Console Sense Data

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

A01429

FUNCTIONAL DIFFERENCES

The 470V/5-I console performs several functions differently from the 3066 and the 3215 operator consoles. Note these differences when emulating either console type:

- The 470V/5-I console operates on a block-multiplexer or selector channel; it does not operate on a byte-multiplexer channel.
- The 470V/5-I console operates in forced burst mode on a block multiplexer channel.
- The 470V/5-I 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 3066 or 3215 when selected.

- The 3066 and 3215 keyboards have both uppercase and lowercase alphabetic input. The 470V/5-I console sends alphabetic input in uppercase only.

- The 470V/5-I console does not process immediate CCWs; it does not return Channel End at the completion of a START I/O. Therefore, immediate CCWs to the console result in an incorrect length indication. To suppress this indication, set CCW bit 34 (SLI).

CONSOLE SENSE DATA

Because the 470V/5-I 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 operator console. All other sense data bits are defined in the *370/145* and *370/168 Functional Characteristics* manuals referenced previously.

CHANNEL PAGE PASSINGS CCWs

Two extra CCWs are implemented on the 470V/5-I 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 15. For more detailed information, refer to "Machine Check Logouts."

Table 15. Channel Page Passing CCWs

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

A01430

amdahl®

AMDAHL CORPORATION
1250 EAST ARQUES AVENUE
P.O. BOX 470
SUNNYVALE, CALIFORNIA 94086

READER COMMENT FORM

WE WOULD APPRECIATE YOUR COMMENTS AND SUGGESTIONS FOR IMPROVING THIS PUBLICATION.

Publication No.	Title	Current Date

How did you use this publication?
☐ Study ☐ Installation ☐ Sales
☐ Reference ☐ Maintenance ☐ Operations

What is your occupation?

What is your overall rating of this publication?
☐ Very Good ☐ Fair ☐ Very Poor
☐ Good ☐ Poor

Is the material presented effectively?
☐ Well organized ☐ Fully covered ☐ Clear
☐ Correct ☐ Well illustrated

Please enter your other comments below. If you were in any way dissatisfied with this publication, we would like to know why. Be specific, if possible; give page, column, and line number references where applicable.

All comments and suggestions become the property of Amdahl Corporation.

Your name & return address (include ZIP code):

AM2283 6/79

Thank you for your interest. Fold and fasten as shown on back. No postage necessary if mailed in U.S.A.

FOLD AND SCOTCH TAPE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 741 SUNNYVALE, CA.

POSTAGE WILL BE PAID BY ADDRESSEE

ATTENTION: TECHNICAL PUBLICATIONS
DEPARTMENT 734

amdahl®

AMDAHL CORPORATION
1250 EAST ARQUES AVENUE
P.O. BOX 470
SUNNYVALE, CALIFORNIA 94086

FOLD AND SCOTCH TAPE