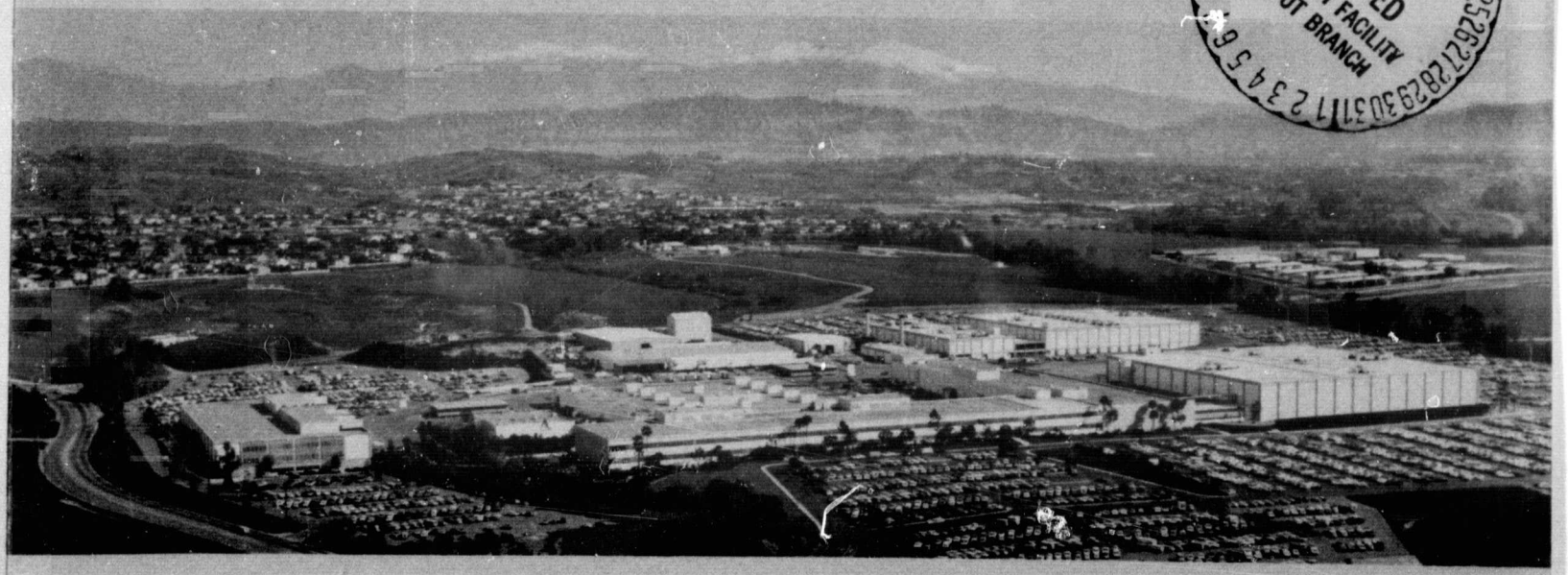
NASA CR-144038

3

HUGHES

HUGHES AIRCRAFT COMPANY
GROUND SYSTEMS GROUP



DESIGN OF A MODULAR DIGITAL COMPUTER SYSTEM

CDRL #D001

TECHNICAL REPORT, FINAL DESIGN PLAN

2 SEPTEMBER 1975

Design of a Modular Digital Computer System

CDRL #D001

Technical Report, Final Design Plan

September 2, 1975

Prepared under Contract NAS8-27926

by

HUGHES AIRCRAFT COMPANY
FULLERTON, CALIFORNIA

for

George C. Marshall Space Flight Center
National Aeronautics and Space Administration

FR 75-11-568

FORWARD

This report documents the engineering breadboard implementation developed during the logic design phases of contract NAS8-27926 from January 1, 1974 through September 1, 1975. This effort is a follow-on to the architecture study completed under the earlier phase of this contract on December 31, 1973 and documented in Hughes Report FR-73-11-998 (DRL 4 and 5) previously submitted. Hughes Data Processing Products Division in Fullerton, Calif has performed this design under direction of Dr. J.B. White of the Data Systems Laboratory of NASA's Marshall Space Flight Center in Huntsville, Alabama. The design applies the architecture concepts developed previously under this contract to MSFC's SUMC computer and is intended to fulfill all design requirements set forth in this contract's scope of work.

This report was prepared by R.A. Easton. Major contributors to the described implementation are R.A. Easton - system, CCE, memory, MPSE and CPE designs; G.H. Ullom and S.Y. Chung – CPE design, E.E. Erlandson – IOP design, and J.E. Bartling - mechanical packaging.

CONTENTS

1. INTRODUCTION

The ARMS Engineering Breadboard (EB) described in this report is designed to verify the concepts of a fault tolerant, automatically reconfigurable, modular version of the SUMC general purpose digital computer system developed under the architecture study phase of this contract. ARMS has a microprogrammed 32 bit word length, general register architecture and an instruction set consisting of a subset of the IBM System 360 instruction set plus additional fault tolerance firmware.

The ARMS EB consists of the following modules:*

1 – Central Control Element (CCE)
4 – Central Processing Elements (CPE)
4 – Memory Modules (MEM)
1 – Input/Output Processor (IOP)
1 – Maintenance/Status Panel and Electronics (MSPE)
1 – Interconnect Board (IC)

An overall block diagram of the ARMS EB is shown in figure 1-1. The detailed inter-module wire lists are discussed under each module later in this report. The functions of each module are as follows:

The CCE provides system control, configuration, and fault and status monitoring functions for the ARMS EB. The CCE automatically reconfigures ARMS when a fault has been detected. In addition it initially configures the breadboard with respect to which modules are active and passive, and to which physical memories will respond to given logical page assignments, and it provides for synchronized system clock distribution and for d.c. power control and power fault master clear signals to other ARMS modules. The CCE module is implemented in simplex in the ARMS EB to minimize cost. In a flight version it is recommended that 3 copies of the CCE be implemented with all control outputs voted. The CCE is described more fully in section 3 of this report.

The CPE modules for the ARMS EB are based at a register level on MSFC's 32-bit in-house breadboard version of the SUMC processor with modifications where necessary to meet ARMS fault tolerance demonstration objectives or to make cost effective

*Although all module designs are complete only 1-CCE, 1-CPE, 1-MEM, 1-MSPE and 1-IC have been fabricated in the initial fabrication phase. Fabrication of the IOP and remaining CPE and MEM modules is anticipated under future funding.

use of currently available MSI circuits. It's instruction set includes all 86 fixed point and 8 floating point instructions compatible with IBM System 360 and SUMC plus 2 special instructions for use in fault tolerance demonstration. Remaining system 360 floating point instructions can be easily added by adding additional ROMs to the CPE at a later date if desired. The CPE design includes fault tolerance features recommended in the ARMS architecture study including voting and parity logic to complement total system wide fault tolerance techniques. The CPEs are designed to operate in either simplex, duplex, or triply modular redundant (TMR) modes. The CPE module is described more fully in section 4 of this report.

Memory modules for the ARMS EB each consist of a commercial EMM micromemory 2000 series core memory in a 4096 word by 39 bit configuration with a custom interface providing for fault tolerance including Hamming-parity error detection and correction coding of data, and for control and processing of accessing requests from CPEs and IOPs on 4 sets of data buses in simplex, duplex, and TMR modes. The memory module is described in detail in section 5 of this report.

The IOP provides the communications link between ARMS and external devices. It contains two subchannels (one to interface with MSFC's DMS and one to interface with a TTY) which can operate independently once initiated by the CPE(s). The IOP accesses memory via buses shared with the CPEs but on which the IOP has first priority. Like the CPE it contains fault tolerant circuits complementing overall system fault tolerance techniques. The IOP for the ARMS EB is designed to simulate the effects of multiple IOPs operating in simplex, duplex, and TMR modes at system interfaces without incurring the costs associated with actual fabrication of multiple IOPs. The IOP module is described more fully in section 6 of this report.

The MSPE provides for operator access to and control of all modules in the ARMS breadboard including the controlled introduction of faults for study. The panel also provides for status scanouts in both binary and hexidecimal form of approximately 3500 key register and control bits within the various ARMS modules. Section 7 of this report describes the MSPE.

The interconnect board (IC) contains all inter-module wiring within the breadboard including the 4 sets of buses to and from memory. Each module physically connects to the IC by means of pigtail connectors for easy maintenance and expandability. A

discussion of all interface signals at each module follows the functional description of that module.

The ARMS EB is packaged in a single 6 1/2 foot rack using commercial TTL integrated circuits as described in section 2 of this report. Since no accurate reliability figures are available for these parts no overall reliability discussion is included in this report. However the final complexity level of the various modules compares favorably with that assumed in the reliability analysis of our previous phase report (except for the CPE which has more ROM than originally assumed since the original CPE concept was not System 360 compatible). Based on that analysis an ARMS flight prototype using an LSI implementation of the breadboard design and flight qualified memories instead of the commercial core memories included in the breadboard should have no difficulty meeting ARMS goal of 0.99 probability of successful operation over a 5 year mission despite the enlarged CPE. Discussion of specific module gate counts and reliability features are included later in this report in the sections dealing with each module.

Additional documentation pertaining to the ARMS breadboard but beyond the scope of this report is listed below:

Ref 1. IBM System/360 Principles of Operation. IBM document No. GA-6821-8 SDD Product Publications, Poughkeepsie N.Y. (1970)

Ref 2. Design of a Modular Digital Computer System. Final and Phase III Report FR73-11-998 Hughes Aircraft Co., Fullerton, Cal. (1973)

In addition an ARMS Breadboard Engineering Drawings and Documentation Package containing ARMS engineering notebook entries not incorporated into this present report will be submitted to MSFC in connection with this contract. The Package consists of:

1. CPE Microprogram Documentation Explanation
2. CPE Microprogram Flowcharts
3. CPE Microprogram Computer Listing
4. IOP PROM Contents Table
5. Logic Diagrams for all ARMS modules
6. Wire Lists for all ARMS modules* by physical sequence (connector sequence lists) and alphabetically by signal names (ADDS IC Maintenance Reports).

*not available for IOP module at this time

7. Integrated Circuit Layouts and IC lists by Function for all modules.
8. Maintenance Panel Scanout Maps listing both functions and signal mnemonics displayed as testpoints in all modules.
9. Documentation supplied with commercial core memories.
10. Cabinet Assembly Wiring and Detailed Intermodule Wiring Diagrams.
11. Explanations of Wire Plane Designation and Configuration Plan and of Interconnectboard special Wiring Requirements.
12. Mechanical Design Drawings.

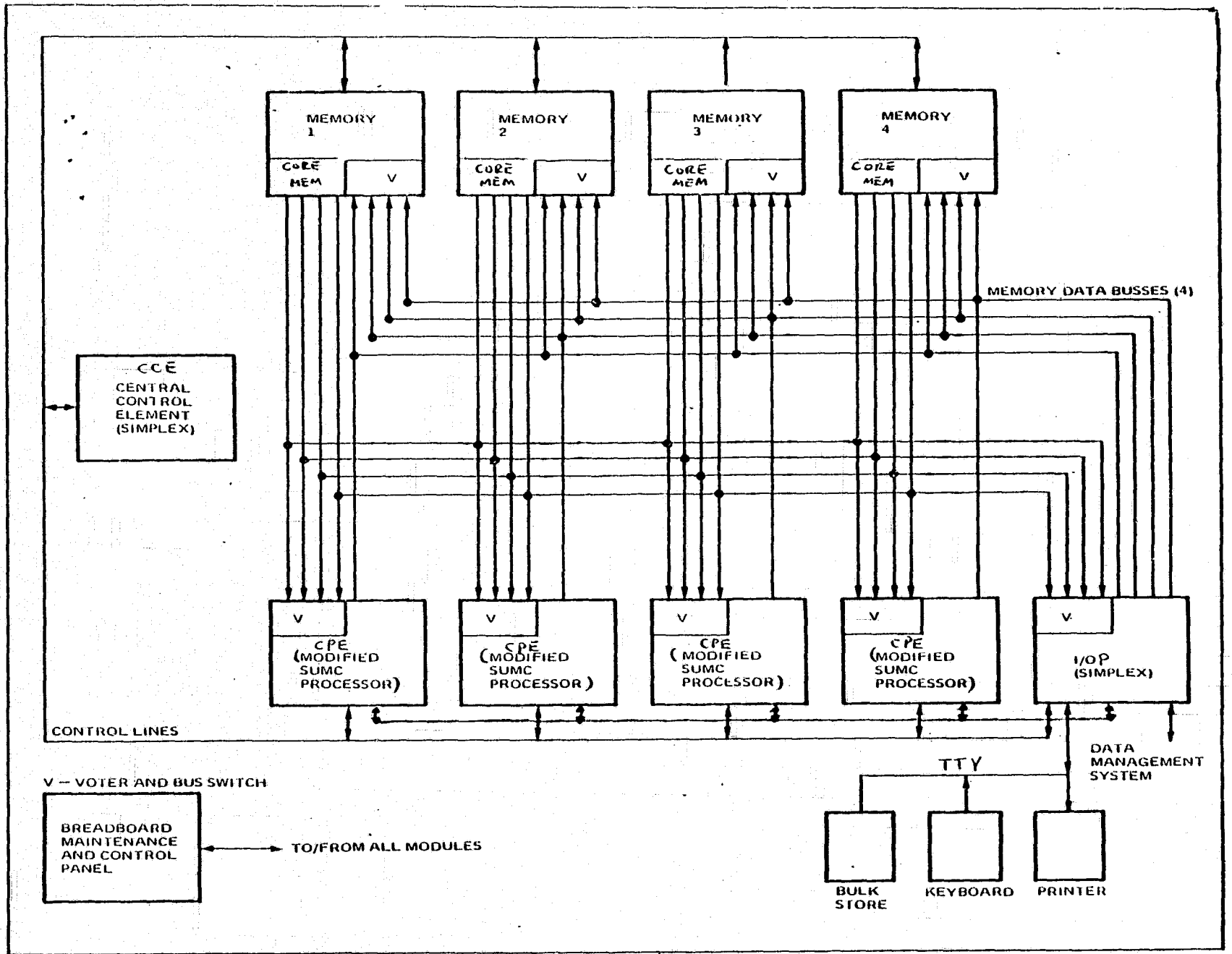


Figure 1-1
Automatically Reconfigurable Modular Computer

2. ARMS BREADBOARD PACKAGING

The ARMS breadboard is assembled using commercial techniques and readily available parts. The single string version uses approximately 2800 DIP packaged integrated circuits, predominantly of the 7400 series. The full breadboard uses nearly 7000 DIPs. These ICs are interconnected using AMP sockets with termipoint wiring mounted on EECO 2-D packaging frames.

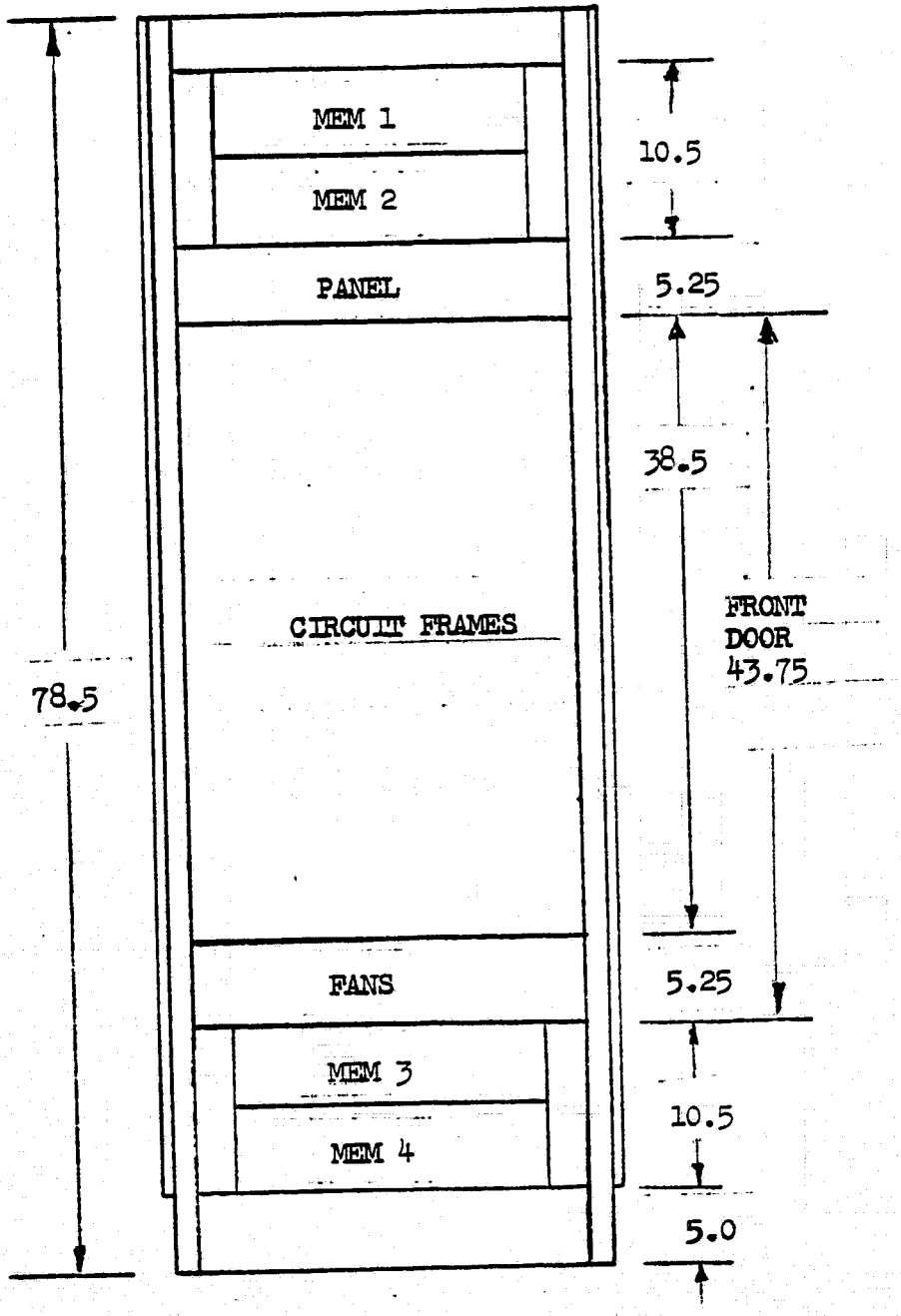
The breadboard packaging concept is shown in Figure 2-1. The logic except for the commercial memories are contained on up to 7 vertical frames with each having space for up to 1296 DIPs. The single string breadboard uses only 3 frames – one each for CPE and IOP modules and one shared between memory interface, maintenance electronics, and CCE. The full version of the breadboard contains 3 more CPE frames and one additional frame holding 3 more copies of the memory interface logic. All wiring between modules including pull-up resistors are contained on an interconnect board perpendicular to the DIP frames. All DIP frames are mounted on hinges allowing easy access to both IC and wiring sides of each frame. Forced air cooling is provided by fans mounted beneath the circuit frames as shown in Figure 2-2. This figure also shows the locations of the maintenance panel, above the circuit frames at eye level, and of the 4 commercial memory modules. All modules are contained in a 6 1/2 foot x 24 inch rack except for logic power supplies. The breadboard logic requires approximately 75A @ 5V for the single string and 190A @ 5V for the full version. Either version requires ±12V supplies at 300 ma for the TTY interface in addition to the main 5 volt supplies. The commercial core memory power supplies are self contained within the memory packaging.

TABLE 2-1. IC COUNT AND POWER REQUIREMENTS

Module	No. of* ICS	Amps @ +5V Required
CCE	254	4.56
CPE	949	30.25
Memory	351	7.11
IOP	1133	30.44
MSPE	46	2.26
Interconnect	58	-
	2791	74.62

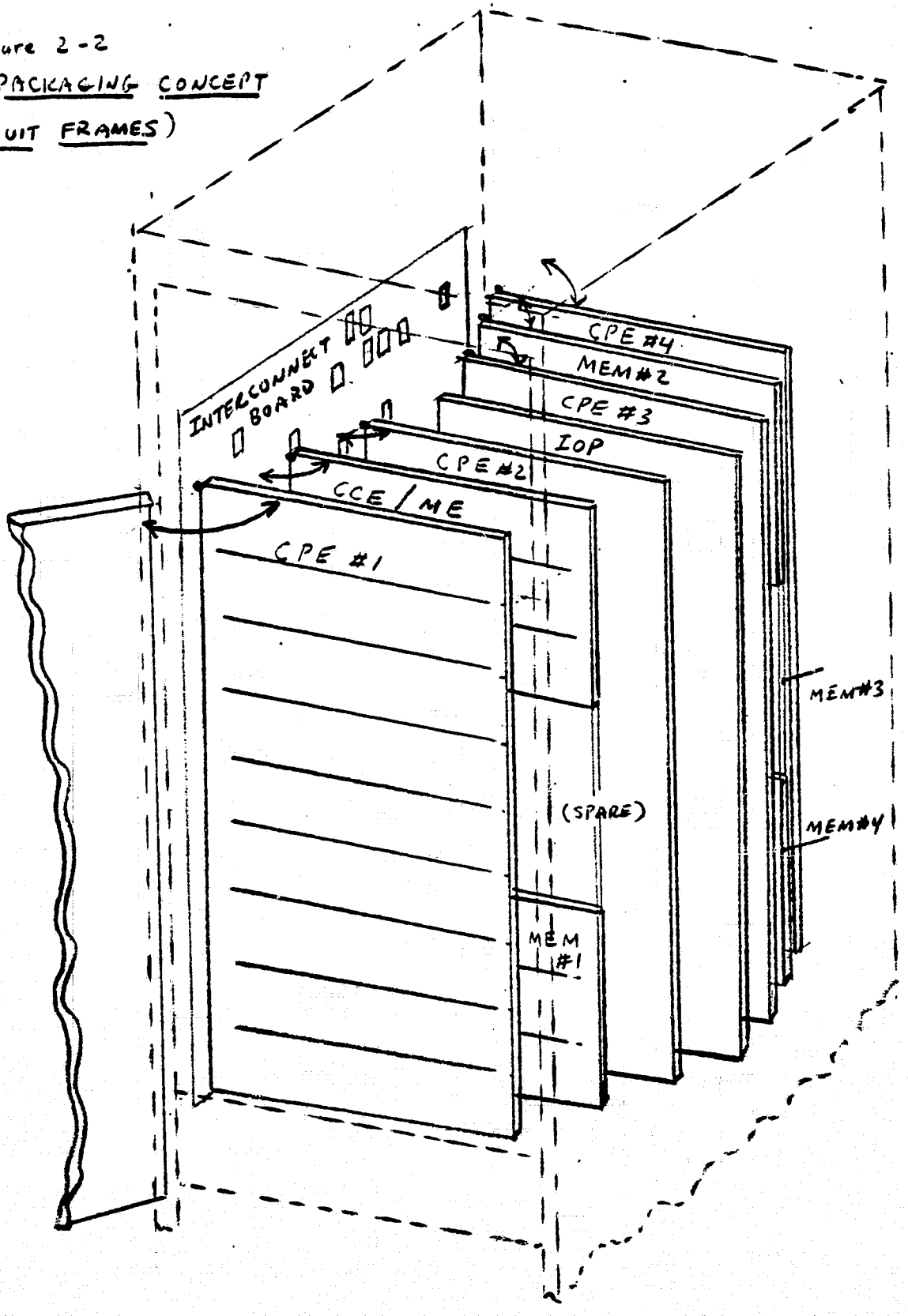
*includes flat-pack resistors

Figure 2-1
ARMS CABINET



UNIT LOCATIONS
(FRONT VIEW)

Figure 2-2
ARMS PACKAGING CONCEPT
(CIRCUIT FRAMES)



3. CENTRAL CONTROL ELEMENT (CCE)

The Central Control Element distributes power and clock signals to other ARMS modules and coordinates ARMS reconfiguration either due to new assignments from the maintenance/status panel or in response to fault interrupts from other ARMS modules. In order to minimize costs the breadboard CCE does not include redundancy that could be implemented in a flight version. For maximum reliability a TMR CCE with voting between the parts on all outputs would be desirable.

The CCE consists of individual status controllers for each ARMS module to be controlled, fault correlation logic, an overall program initiator and reconfiguration controller, switching logic for power supplied to other modules, a crystal controlled central clock source, and external interrupt routing logic. The CCE has no internal processing or main memory bus access capabilities but is capable of utilizing CPE software or hardware to enhance its own hardwired capabilities by means of interrupts. A block diagram of the CCE is shown in Figure 3-1. The following is a description of the specific embodiment of the CCE used in the ARMS breadboard:

CPE Module Status Controller. One CPE module status controller is required for each of the 4 CPE modules in the ARMS breadboard. Each controller keeps track of the CPE's status (spare, active normal, active abnormal, failed) outputting a stream assignment bit corresponding to that CPE's hardwired processor (to memory) bus. Together the 4 CPE module status controllers provide a 12 bit stream assignment to all CPE's identifying which CPE's are active and which are passive. When the CCE is powered initially, each CPE module status controller places its CPE in the spare state. A signal from the maintenance/status panel causes one or more of these controllers to place their CPE's in the "active normal" state. If a

fault interrupt from either a CPE or a memory module indicates that a specific CPE may have failed that CPE's status controller is placed in the "active abnormal" state. Figure 3-2a shows the various states that a module status controller may take on.

If the CPE is operating in the simplex mode when the fault was detected, or if it is operating in the duplex mode and the fault is detected by a memory module without being internally detected within the CPE, the CPE module status controller causes the Program Initiator and Reconfiguration Controller (PIRC) logic discussed in the next section to issue a stop CPE interrupt immediately. If the CPE is operating in the TMR mode when the fault was detected, or if it is operating in the duplex mode and the fault is internally detected within the CPE, the controller issues a stop CPE interrupt immediately following receipt of a CPE available/rollback pace signal from the CPE, or after a prescribed time interval, whichever is shorter. Once in the "active abnormal" state one of the following events occurs in the CPE module status controller:

a) If the CPE issues a CPE available/rollback pace signal prior to receipt of another fault interrupt concerning this CPE the status controller returns the CPE to the "active normal" state.

b) If another fault interrupt concerning the CPE is received prior to receipt of the CPE's available/rollback pace signal the controller enters the failure pending state. From this state the reconfiguration controller either replaces the faulty module if it has sufficient priority and a spare is available, transferring its assignment to the spare CPE and causing the CPE module status controller to place its CPE in the failed state, or otherwise the reconfiguration

b) continued

controller returns the CPE module status controller to the active abnormal state. Thus modules that cannot be immediately replaced continue to be retried, and ARMS continues to operate in the presence of maskable failures.

Fault interrupts from IOP or main memory modules cause issuance of a stop CPE interrupt immediately if the CPE is operating in the simplex mode or immediately following receipt of a CPE available/rollback pace signal from the CPE if the CPE is operating in the duplex or TMR mode. The CPE module status controller remains in the active normal state during this operation in the absence of a fault interrupt placing blame on the CPE. The CPE module status controller also issues stop CPE interrupts prior to any external command update of assignments from the maintenance/status panel or due to an emergency such as an impending power failure.

IOP Module Status Controller. The IOP module status controller design requirements are similar to those for the CPE module status controller with the following exceptions:

a) A stop IOP interrupt will not be issued unless the IOP does not stop within a prescribed time interval after all CPEs have halted.

b) An IOP will not be returned to the "active normal" state from the "active abnormal" state unless all active CPEs issue CPE available/rollback pace signals prior to the receipt of another fault interrupt concerning this IOP.

Main Memory Module Status Controller. One main memory module status controller will be required for each of the 4 main memory modules in the ARMS breadboard. These controller's design requirements will be similar to those for the CPE module status controller with the following exceptions:

a) A stop memory interrupt is not required.

b) A main memory module will not require stream assignment status bits but will require page address and output bus assignments. The output bus assignment determines if a memory module will transfer data to CPEs or IOPs on the lower, (middle), or upper numbered memory(to processor)bus paired with the processor(to memory)buses to which access was granted. An "essential/non-essential" memory status bit is also required internal to the main memory module status controller to determine the proper memory replacement algorithm for the reconfiguration controller in response to a memory fault interrupt. An essential memory contains programs and important data the loss of which could disable a stream. A non-essential memory contains working storage and other contents the loss of which would not disable a stream.

c) A main memory will not be returned to the "active normal" state from the "active abnormal" state unless all active CPEs issue CPE available/rollback pace signals prior to the receipt of another fault interrupt from this memory.

Program Initiator and Reconfiguration Controller. The program initiator and reconfiguration controller (PIRC) restarts the ARMS CPEs initially, or if they have been stopped for any reason, and controls the transfer of status assignments between individual module status controllers when ARMS reconfiguration is required.

The program initiator logic is activated whenever a load request is received from the maintenance/status panel, any faults are detected, or CPE available/rollback pace signals are not received from all CPEs within an interval timed by the PIRC logic. The various states that the PIRC logic can assume are shown in Figure 3-2b. Once activated, the program initiator logic issues stop interrupts to the CPEs as discussed in the previous section, issues a panic halt signal to all CPEs and IOP, waits for CPE available/rollback pace signals from all CPEs and an IOP available signal to stabilize in the available states,and then takes one of the following actions in descending priority:

a) In the case of an essential memory failure in the duplex or TMR mode the program initiator logic issues a clear memory interrupt to the questionable memory, forcing its output to "0" pending completion of initialization, followed by an initialize memory interrupt, along with control information specifying the memory page to be initialized, to the highest priority CPEs. These CPEs enter a program that alternately reads from and then writes into every word in that memory page duplicating data from the good memory(s) into the newly assigned memory. All zero output conditions from the memory being initialized shall be considered to be normal until this operation is completed as signalled

a) continued

by a rollback pace signal from the CPE in question. Upon receipt of this signal the program initiator logic issues start interrupts to any remaining active CPEs if more than one processing stream is used in ARMS and restores the newly initialized memory to normal operation. Upon completion the memory initialization program automatically returns to the appropriate rollback point of the program in progress at the time of the interrupt.

b) In the case of any other failure the program initiator logic issues start CPE interrupts to all active CPEs causing them to return to the appropriate rollback point(s) for the program(s) in progress at the time of the interrupt.

Figure 3-3 shows the PIRC logic necessary to respond to CPE rollback pace signals and to issue the interrupts discussed above. The reconfiguration controller controls the transfer of status assignments between individual module status controllers in response to commands from the breadboard's maintenance/status panel or to any of the individual module status controllers entering the failure pending state. Transfers of status assignments from failed active modules to newly activated spare modules occur once the program initiator logic verifies that the IOP and all CPEs are available (i.e., stopped) and prior to issuance of any interrupts by the program initiator logic with the following restrictions:

a) Only one module of any given type can be replaced at a time and a spare module of that type must be available. For example, one memory plus one CPE may be replaced but not two CPEs at one time. If two CPEs did fail at once, one would be retried a second time and if it still malfunctioned and an additional spare CPE was available it would then be replaced.

b) Essential main memory modules operating in simplex cannot be replaced by spares since no mechanism for initializing them is available. A permanent failure in such a memory module requires outside intervention for correction.

The logic for transferring assignments between status controllers is shown in Figure 3-4.

Fault Correlation Logic. The fault correlation logic allows the CCE to maximize the probability of correctly isolating a fault to a specific ARMS module within limitations dictated by a reasonable level of hardwired logic complexity and allows the CCE to determine that certain faults are maskable so that critical programs can continue to completion. The CCE correlates received fault interrupts from each CPE, IOP, and main memory module with appropriate status information from their status controllers as shown in Figure 3-5.

Many CPE and IOP faults may be isolated due to fault interrupts from the module in question. Single memory module fault interrupts indicate failures within the interrupting memory. In duplex and TMR modes simultaneous fault interrupts from two or more memories can isolate a failure to a CPE or IOP module whose identity is encoded in the interrupt. In the duplex mode these interrupts may only isolate the fault to one of two CPEs or IOPs in the absence of a direct fault interrupt from the offending module. However an arbitrary replacement of one of these modules provides 50% probability of success in cases that otherwise would result in an ARMS system failure.

In simplex mode detectable faults (other than maskable single bit failures within main memory modules) result in immediate rollback or replacement of the offending module. In the absence of a fault interrupt from the CPE or IOP the fault is blamed on non-essential memories or on the CPE or

IOP accessing an essential memory in the case of an ambiguous fault. If a fault is unambigiously isolatable to an essential memory the fault is insolvable since no mechanism exists for initializing a spare in this mode. Some faults may be undetectable in simplex mode.

In duplex mode virtually all faults are detectable and at least those detectable in simplex allow the program to continue to its next rollback point and then are correctable in real-time through reconfiguration so long as spare modules are available. In all modes ARMS breadboard is capable of continued computation in the presence of faults so long as these faults are maskable. The choice between rollback and continued computation is software determined in that it is dependent upon whether the program is stopped before or after the program status block is updated. If the block has been updated the next program is executed, if not, then the present program is repeated. Programs shall be constructed so that they can be repeated if necessary.

Power Switching Logic. The CCE distributes power to all other ARMS modules. The power switching logic provides power to each ARMS module whose individual status controller places it in either an "active normal", "active abnormal", or "failure pending" state.

Crystal Controlled Clock. The CCE contains a crystal controlled oscillator providing central clock signals to all ARMS modules to assure their synchronization.

External Interrupt Logic. The CCE holds external interrupts when they are received and routes them to the CPEs for which they were intended. When a CPE responds to a given interrupt it sends a response to the CCE which clears the interrupt once it receives response from a majority of the CPEs to which the interrupt was sent. As in the case of the power and clock distribution external interrupts are routed through the CCE since it is the only element in ARMS which remains stable throughout system reconfiguration. Clock Distribution and External interrupt logic is shown in Figure 3-6.

CCE Technology and Component Count. A CCE is being breadboarded out of T^2L small scale integrated circuit logic. For maximum reliability it should ultimately be implemented with CMOS LSI technology. Table 3-1 shows the number of gates and flip-flops required by each part of the CCE. Clearly the CCE complexity would increase for larger numbers of controlled modules but for ARMS it contains less than 1200 equivalent gates (60% of the complexity used for reliability calculations in Reference 2) and is simple enough to be readily implemented on 2 or 3 large scale integrated circuits if desired.

Relating the Functional Description to Logic Diagram. Table 3-2 relates states shown in the diagrams in this document to mnemonics used in logic diagrams for the CCE. Table 3-2a covers states shown in Figure 3-2a. Table 3-2b covers states shown in Figure 3-2b; Table 3-2c relates the module status controller mnemonics to paths in Figure 3-4 while Table 3-2d lists the 4 bit codes used in the memory fault interrupts followed by the mnemonics related to paths in Figure 3-5.

Table 3-3a lists CPE status controller stream assignment codes by status and priority and Table 3-3b lists memory status controller output mode codes by function. It should also be noted that for essential memories QESSM_ = 1.

Figure 3-1

ARM'S CCE BLOCK DIAGRAM

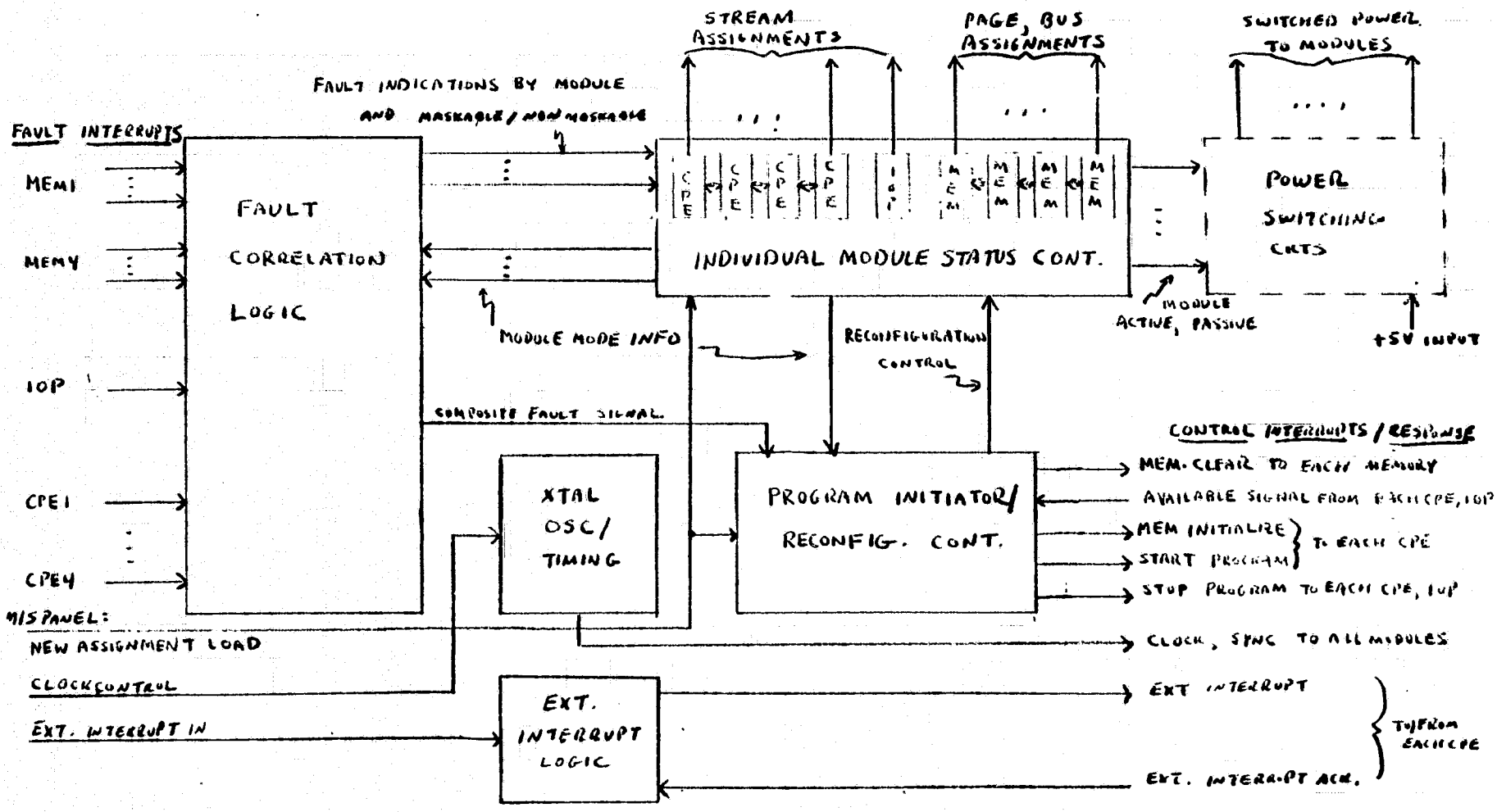
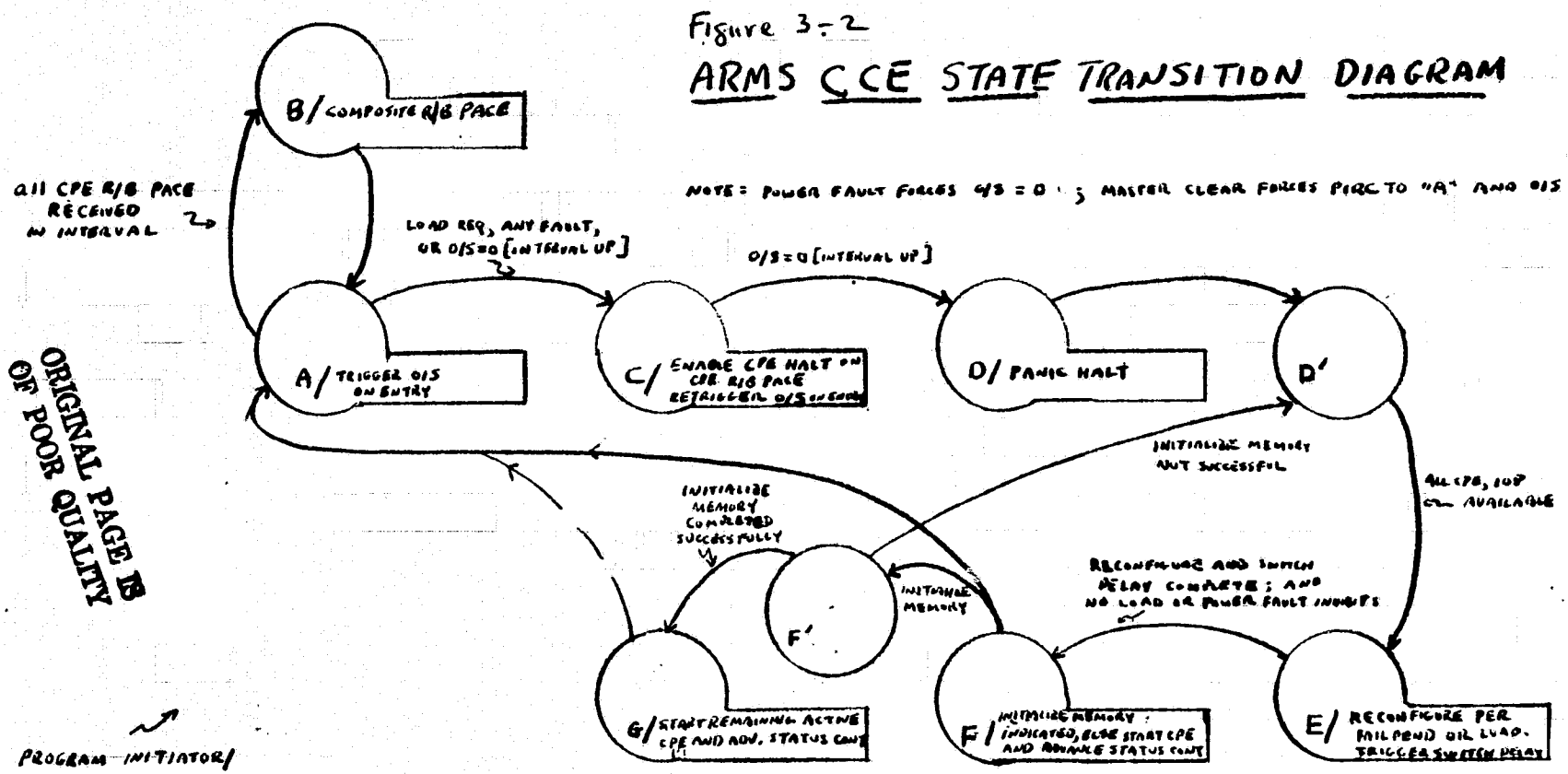


Figure 3-2

ARMS CCE STATE TRANSITION DIAGRAM



b. PROGRAM INITIATOR/ RECONFIGURATION CONTROL

NOTE: POWER FAULT FORCES O/S = 0 ; MASTER CLEAR FORCES PSC TO "A" AND O/S = 0.

a. INDIVIDUAL MODULE STATUS CONTROL

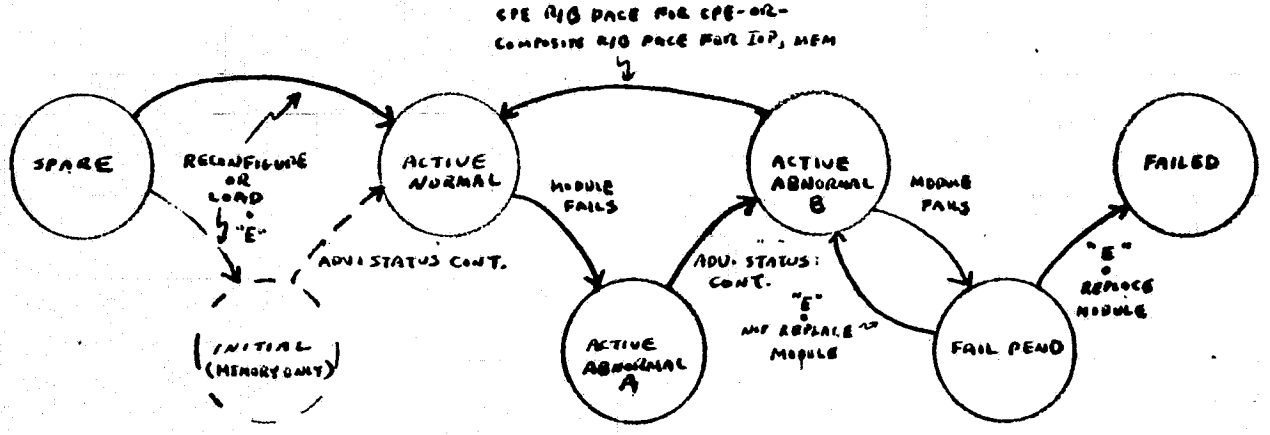


Figure 3-3

CCE PROGRAM INITIATOR/RECONFIGURATION CONTROL (PIRC) LOGIC

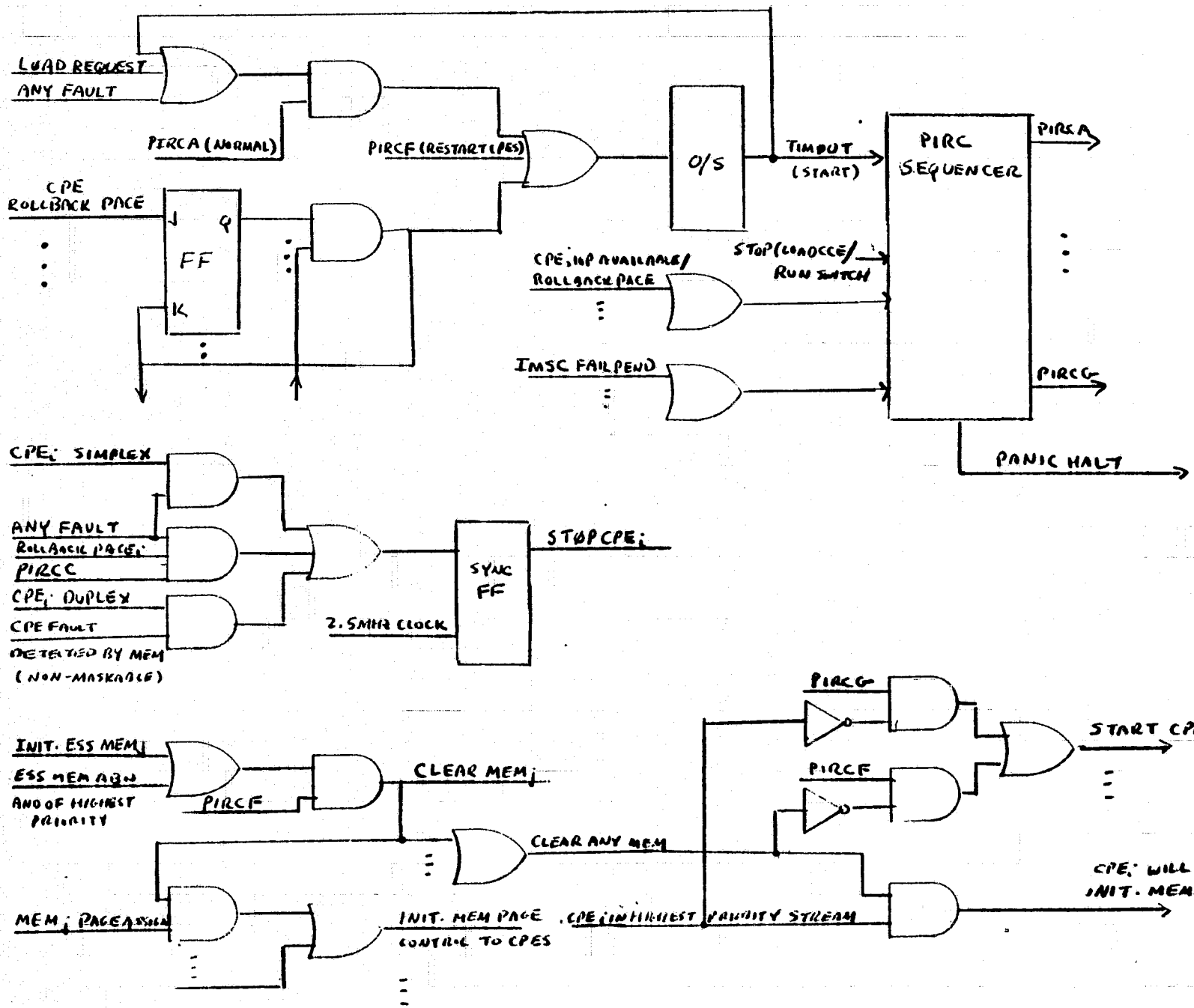


Figure 9-4

ECE MODULE STATUS CONTROLLER LOGIC

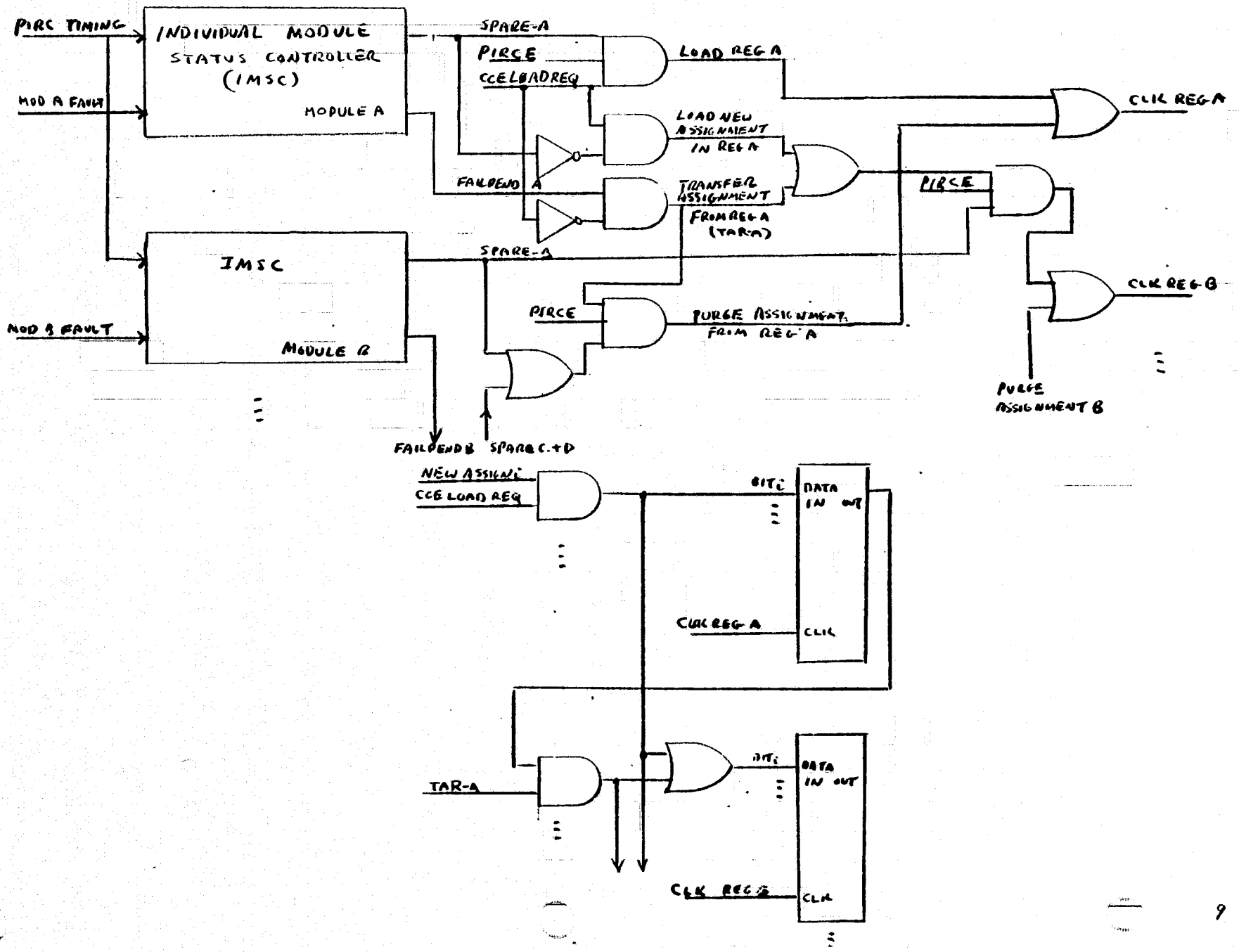


Figure 3-5

CCE FAULT CORRELATION LOGIC

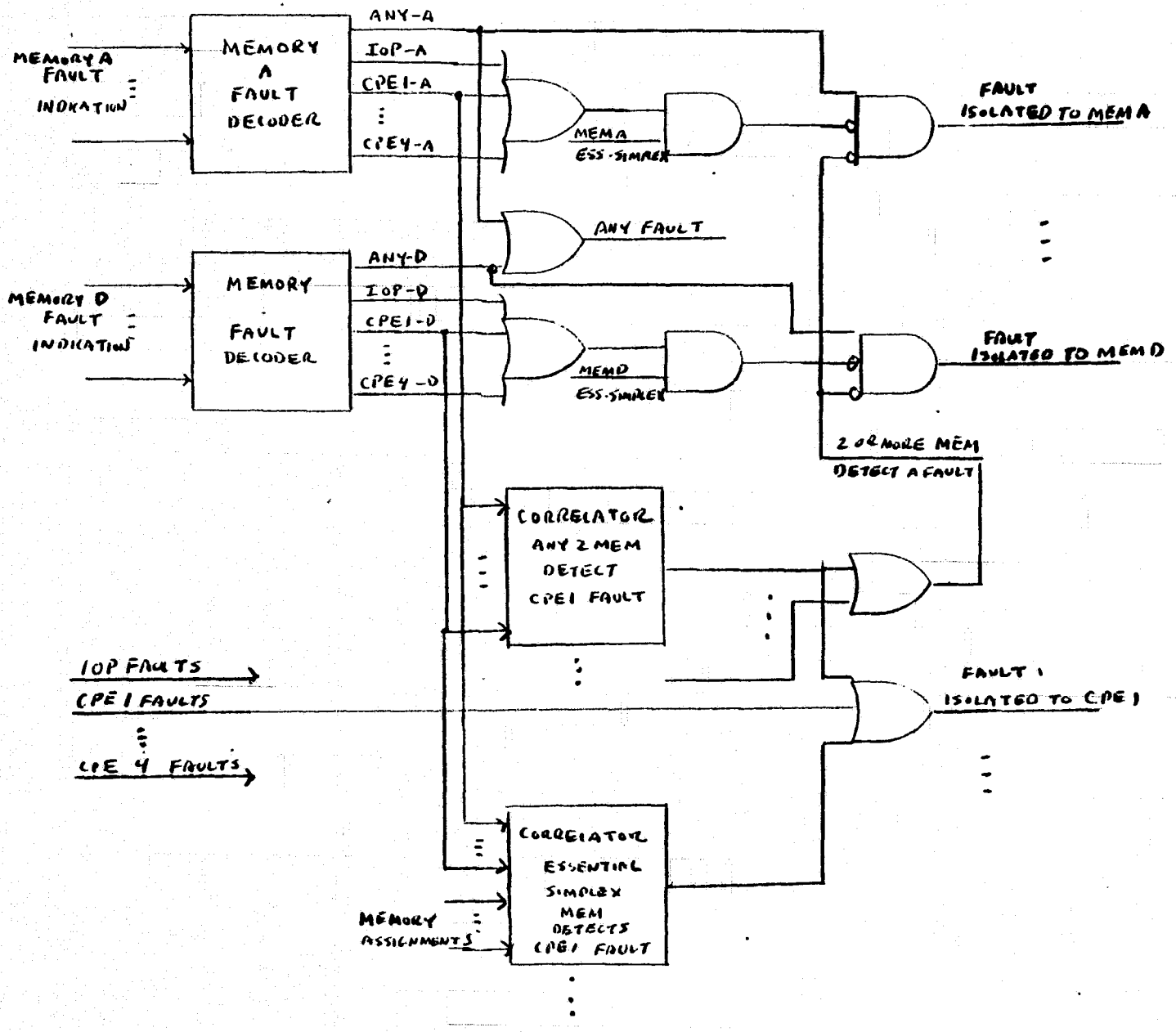


Figure 3-6

CCE CLOCK DISTRIBUTION AND EXTERNAL INTERRUPT LOGIC

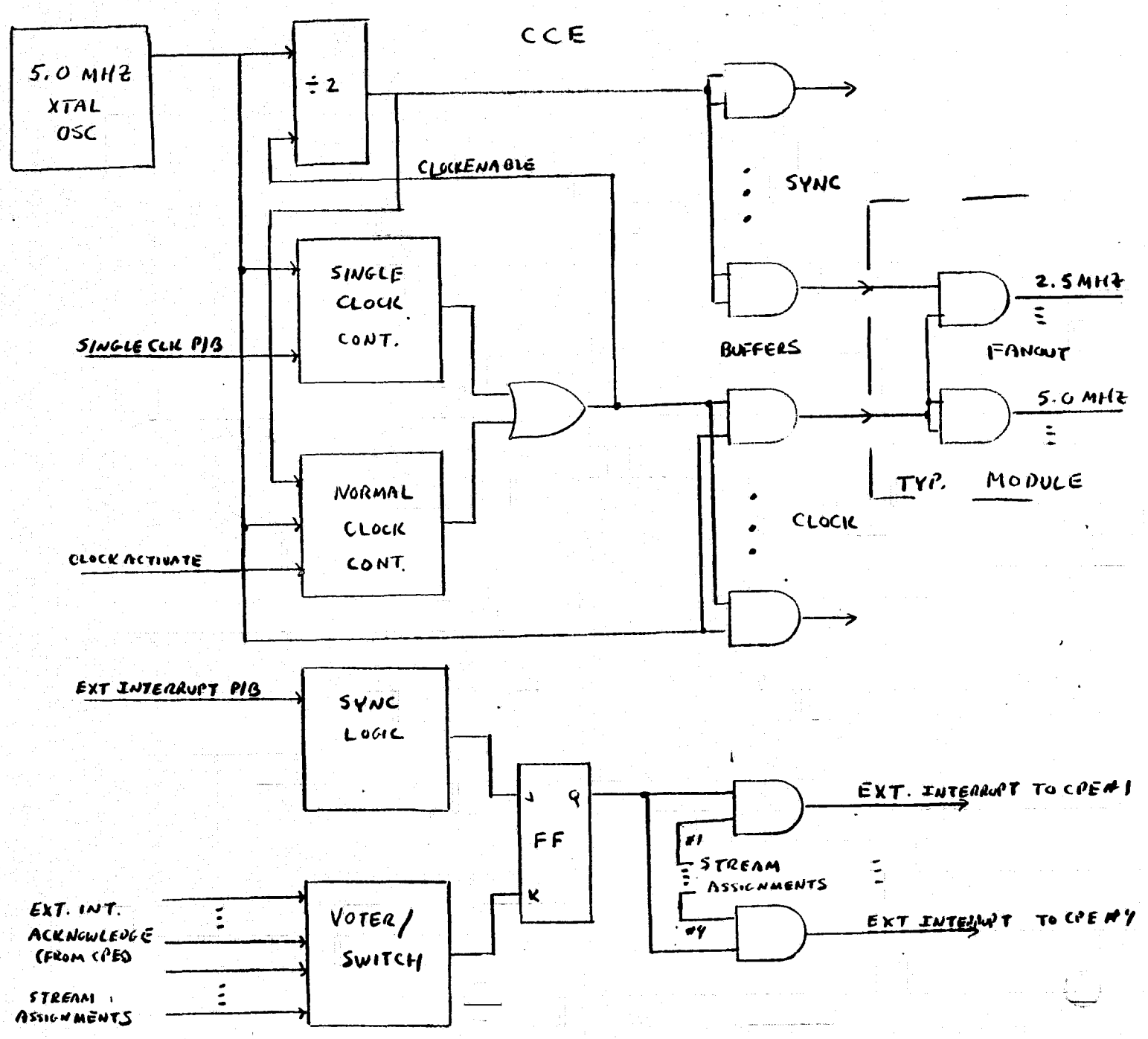


TABLE 3-1

CCE COMPONENT COUNT

Function	Gates	Flip/Flops	Total Equiv. Gate
1. CPE Status Controller	131	28	299
2. IOP Status Controller	51	8	99
3. Memory Status Controller	159	36	375
4. Program Initiator/Reconfiguration Control	119	13	197
5. Fault Correlation	134	0	134
6. Clock Control/Distribution	14	4	38
7. Ext. Interrupt Logic	20	2	32
	628	91	1,174

TABLE 3-2

A. INDIVIDUAL MODULE STATUS CONTROLLER STATES

STATE	QSCTLX1, 2, 3, 4
SPARE	0 0 0 0
INITIALIZE (Mem only)	0 0 1 1
NORMAL	0 0 1 0
ABNORMAL A	0 1 0 1
ABNORMAL B	0 1 0 0
FAIL PEND	1 0 0 1
FAILED	1 0 0 0

B. PROGRAM INITIATOR/RECONFIGURATION CONTROL STATES

STATE	QRCTLD, C, B, A, QTIMOUT
NORMAL (RECON A)	(0 0) 0 0 1
RECØN C	(0 0) X 1 0
PANICHALT (RECOND)	(0 0) 1 1 0
RECØN E	0 0 1 0 0
RECØN F	0 1 (1 0) 0
RECØN F'	1 0 (0 0) X
RECØN G	1 1 (0 0) 0

NOTE: In Figure 2a

RECON B = RBRCVA, B, C, D all = 1

RECON D' = RECON E prior to all RBPACi = 1

TABLE 3-2 - Continued

C. MODULE STATUS CONTROLLER LOGIC

1. A register may be clocked under up to 3 conditions:

   a. XLDNEW_ Load new assignment if module is highest priority spare

   b. XLDRPL_ Load assignment from highest priority register if module is highest priority spare and no new assignments are pending.

   c. XPURGE_ Purge current assignment if a spare is available and a XLDRPL is true

   All loads and purges are enabled by RECONE from PIRC
   Descending priorites for replacement are: 1, 2, 3, 4, or A, B, C. D

D. FAULT CORRELATION LOGIC

TMFLTX 1, 2, 3, 4

IOP	0	1	X	X	(if multiple IOPs were used this could be fully decoded)
CPE #1	1	0	0	0	
CPE #2	1	0	0	1	
CPE #3	1	0	1	0	
CPE #4	1	0	1	1	

NOTES

1. XDU_F signals signify 2 or 3 memories isolated a fault to one processor
2. XSI_F signals signify an essential simplex memory detected a fault that may be in the memory or may be in the processor.
3. Fault is placed on memory unless either any XDU_F or XSI_F pointing to that memory occurs. [XMFLT_] Any XDU_F or XSI_F pointing to a specific processor or a processor fault from that processor indicates a processor fault XCFLT_.

TABLE 3-3

A. Stream Assignment and Memory Priority Codes

QSASN_3, 2, 1	STATUS		ORDER OF PRIORITY
000	4 MR Spare/IO	used	1 - I/O (HI)
001	TMR/DU1	for TMR	4
011	TMR/DU1	used for	4
010	TMR/DU2	duplex	5
100	SIMPLEX 1		2
101	SIMPLEX 2		3
110	SIMPLEX 3		6
111	SIMPLEX 4		7 . (LO)

NOTES only Stream with 00X priority initializes memories in duplex or TM
CPEs interpret 010 and 011 codes as Duplex, 001 as TMR

B. Memory Output Mode Assignment Codes

QOTMD_2, 1		STATUS
00	=	SIMPLEX
01	=	HI
10	=	LO
11	=	MIDDLE

CCE INTERFACE

CCE - MEMORY

1. Page Assignment (2/memory)	CCE to Memory	-TPAGE_ _
2. Output Bus Assignment (2/memory)	CCE to Memory	-TOTMD_ _
3. Clear Memory Interrupt (1/memory)	CCE to Memory	TCLRMM_
4. Clock,Sync (2/memory)	CCE to Memory	TCLK, TSYNC
5. Power Protect Inhibit (1)	CCE to Memory	TPWRHLT
6. Memory Fault Interrupt (4/memory)	Memories to CCE	-TMFLT_ _

CCE - CPE

1. CPE Stream Assignment (12)	CCE to CPE	-TSASN_ _
2. Power Control (1/CPE)	CCE to CPE	-TPWRON_
3. Clock,Sync (2/CPE)	CCE to CPE	TCLK, TSYNC
4. Stop CPE Interrupt (1/CPE)	CCE to CPE	TSTOP_
5. Start CPE Interrupt (1/CPE)	CCE to CPE	TSTART_
6. CPE Available/Rollback Pace (1/CPE)	CPEs to CCE	TRBPAC_
7. Initialize Memory Interrupt (1/CPE)	CCE to CPE	TINITM_
8. Initialize Memory Control(2)	CCE to CPE	TPGCTL_
9. CPE Fault (1/CPE)	CPEs to CCE	-TCPFLT_
10. Panic Halt Interrupt (1)	CCE to CPE	TPANICH
11. External Interrupt (1/CPE)	CCE to CPE	-TEXTIN_
12. External Interrupt Acknowledge (1/CPE)	CPEs to CCE	TEXIAK_

CCE - IOP

1. IOP Stream Assignment (4)	CCE to IOPs	-TSASN_ _
2. Clock Sync (2/IOP)	CCE to IOP	TCLK, TSYNC
3. Panic Halt Interrupt (1)	CCE to IOPs	TPANICH
4. IOP Available (1/IOP)	IOPs to CCE	TRBPAC5
5. IOP Fault (1/IOP)	IOPs to CCE	-TIOFLT

CCE-MAINTENANCE PANEL/ELECTRONICS

1.	Fault Entry Switches (6)	Panel to CCE	SFLTIN_
2.	Scanout Source Select (2)	Panel to CCE	SSOC_ _
3.	Scanout Buss (66)	CCE to Electronics	TSBUS_ _
4.	Scanout Enable (1)	Electronics to CCE	TSOMS_ _
5.	Master Clear (1)	Electronics to CCE	-TCLR
6.	Panel Control (12)*	Electronics to CCE	-
7.	Clock, Sync (2)	CCE to Electronics	TCLK, TSYNC

* Not included in motherboard cabling.

CCE INTERFACE

CCE - MEMORY

1. Page Assignment (2/memory) - The CCE sends each memory a two (2)-bit code specifying the page assignment for that memory.
2. Output Buss Assignment (2/memory) - The CCE sends each memory a two (2)-bit code specifying which memory buss to output data on in the TMR and duplex modes.
3. Clear Memory Interrupt (1/memory) - The CCE sends each memory an interrupt line which causes the memory to output all "0's" on the memory bus during the execution of a software routine that cycles through and clears all locations. This is required when initializing essential memories.
4. Clock, Sync (2/memory) - The CCE sends each memory the system clock at the highest frequency required in the system and a sync signal at 1/2 this frequency.
5. Power Protect Inhibit (1) - The CCE sends each memory a Power Protect Inhibit Memory line which specifies the power is going out of tolerance or that system reconfiguration is about to occur and the memories should be shut down.
6. Memory Fault Interrupt (4/memory) - Each memory sends the CCE a 4-bit encoded fault line specifying what type of fault has occurred.

0 - no fault

4 - IOP 1 failure

5 - IOP 2 failure *

6 - IOP 3 failure *

7 - IOP 4 failure *

8 - CPE 1 failure

9 - CPE 2 failure

10 - CPE 3 failure

11 - CPE 4 failure

15 - memory internal

* Not used in the breadboard configuration.

CCE - CPE

1. CPE Stream Assignment (12) - The CCE sends all CPEs 12 lines specifying each CPEs stream assignments for voter switch and output bus control and memory access priority. Each group of 3 bits specifies a CPE in the stream.
2. Clock,Sync (2/CPE) - The CCE sends each CPE the system clock at the highest frequency used in the system and a sync signal at 1/2 this frequency.
3. Stop CPE Interrupt (1/CPE) - The CCE sends each CPE a line which causes the CPE to complete the instruction in progress and store the old PSW in a specified location in memory dedicated to this interrupt and then send the CCE the "CPE Available" signal and then wait in the Fetch cycle for an interrupt from the CCE.
4. Start CPE Interrupt (1/CPE) - The CCE sends each CPE an initialize signal which causes the CPE to fetch the new PSW and start processing. This insures synchronization of the CPEs during a Duplex or TMR mode of processing.
5. CPE Available/Rollback Pace (1/CPE) - Each CPE sends the CCE a signal which specifies that the CPE is halted in the Fetch cycle waiting to be activated or that the program progress alert instruction is being executed.

6. Initialize Memory Interrupt (1/CPE) - CCE sends each CPE an initialize memory interrupt. The CPE fetches a new PSW from a designated memory location for this interrupt and then initializes the specified memory contents to match the contents of the on-line memory it will be paired with in duplex or TMR.
7. Initialize Memory Control (2) - The CCE sends all CPEs a 2-bit code to designate which memory to initialize. These lines are used in conjunction with the Initialize Memory Interrupt.
8. CPE Fault (1/CPE) - Each CPE sends the CCE a signal indicating a fault condition has been detected.
9. Panic Halt Interrupt (1) - The CCE sends all CPEs a Panic Halt Interrupt which causes the CPE to terminate the operations in progress and stop.
10. External Interrupt (1/CPE) - The CCE receives the external interrupt and routes it to appropriate CPEs.
11. External Interrupt Acknowledge (1/CPE) - Each CPE sends an external interrupt acknowledge line to the CCE to indicate acceptance of the external interrupt for processing.

CCE - IOP

1. IOP Stream Assignment (4) - The CCE sends IOP 4 lines specifying the stream assignments for voter switch and output bus control. Each bit specifies an IOP in the stream.
2. Clock,Sync (2) - The CCE sends IOP the system clock at the highest frequency used in the system and a sync signal at 1/2 this frequency.
3. Panic Halt Interrupt (1) - The CCE sends each IOP a Panic Halt Interrupt which causes the IOP to terminate the operations initiated and stop.
4. IOP Available (1/IOP) - Each IOPs sends the CCE a signal which specifies the IOP has completed all outstanding operations and is quiescent.
5. IOP Fault (1/IOP) - Each IOP sends the CCE a line which specified that a fault condition has occurred in the IOP.

CCE-MAINTENANCE PANEL/ELECTRONICS

1. Fault Entry Switches (6) - The fault entry switches allow the insertion of switchable faults into any 6 desired points in any modules for fault tolerance studies.
2. Scanout Source Select (2) - The scanout source select switches allow selection of 2 different CCE scanout sources independently for each of the two 33 bit scanouts on the maintenance panel.
3. Scanout Buss (66) - The tri-state scanout bus multiplexes inputs from scanout sources in each module into the maintenance panel scanouts.
4. Scanout Enable (1) - The maintenance electronics decodes the scanout module select switch output to provide an enable to the scanout multiplexers in the CCE when selected.
5. Master Clear (1) - A master clear is provided to all modules when the master clear pushbutton on the maintenance panel is activated.
6. Panel Control (12) - Panel control lines to the CCE include Panel Data (5), Panel Function (2) and Load Initiate to allow loading new configuration assignments to the CCE from the panel; a STOP/run line forcing ARMS to stop for reconfiguration, single clock activate and initiate lines; and an external interrupt line.
7. Clock, Sync (2) - The CCE sends the maintenance electronics the system clock at the highest frequency used in the system and a sync signal at 1/2 this frequency.

4. CENTRAL PROCESSING ELEMENT (CPE)

The CPE provides general purpose computational capabilities for the ARMS EB. It's architecture is based on the MSFC's 32-bit in-house SUMC-I breadboard. Its instruction set (with the exception of 2 added instructions related to fault tolerance) is a subset of that for SUMC and consequently a subset of that for the IBM System 360.

The CPE is a high performance processor with a 400 nsec microinstruction cycle, two level microprogramming, multi-byte word lengths, fixed and floating point arithmetic, 16 fixed point plus 4 floating point multi-usage registers, a multi-level interrupt structure, and a high speed ALU. Two's complement number representation is used for fixed point data and sign plus absolute number representation for floating point. Addressing capabilities include use of both base and index registers. Sixteen bit Register to Register (RR), thirty-two bit Register and indexed storage (RI), Register and storage (RS), and storage and immediate operand (SI), and forty-eight bit storage to storage (SS) instruction formats operating on half word (16 bit), full word (32 bit) and double word (64 bit) operands are available providing compatibility with IBM System 360. The CPE's Program Status Word (PSW) is also compatible with that of the System 360.

The CPE instruction set is summarized in Table 4-1 and is discussed in detail in Ref. 1. This reference also provides the system 360 specification to which the CPE is expected to conform. The instructions and data formats are shown in Figure 4-1. It consists of the full system 360 standard instruction set (86 instructions) a representative subset of IBM's floating point instruction set (8 instructions) and two added ARMS instructions for use in establishing fault tolerance rollback points. Provisions have been made for incorporating the remainder of IBM's floating point instructions simply by adding more words to the CPE microprogram.

To enhance its fault tolerance the CPE module incorporates parity generating logic at the Processor bus interface at the output of the ALU, and at the outputs of the Memory Address Register and Multiply Quotient Register multiplexers. Parity checks are made at the outputs of the Memory Data Register, Indirect Address Format Control PROM (IAROM), ROM Address Register, Microprogram PROM (MROM), and Multi-Usage Registers (MUR), overall these parity checks detect odd numbers of bit errors in over 75% of the CPE logic, as summarized in the CPE Component Count Table 4-2,

but add less than 6% to the overall CPE gate count. The areas of CPE logic effected are shown cross-hatched in Figure 4-2. A voter/switch allows selection, comparison, or voting on Memory Bus Data on any combination of 1 to 3 memory buses in simplex, duplex, and TMR modes respectively. Internally detected faults force the CPE output to its processor bus to zero until the program is restarted by the CCE allowing a good CPE to over-ride a bad CPE in duplex mode. The fault tolerance design discussed in our architecture study Final Report (Ref 2) has been followed very closely both in the CPE implementation and in the overall system logic design. The only departure from this design has been the decision not to provide redundant ALUs in the ARMS EB in the interest of reduced cost since adding this redundancy to the breadboard would prove little that cannot be shown on paper.

The CPE provides address generation capabilities consistent with IBM system 360. The method of address generation during the fetch microprogram depends upon the instruction format involved. The effective address capabilities consist of base and indexing of a displacement field in the instruction format yielding a byte address internal to the CPE. Due to ARMS EB memory limitations the maximum effective byte address must not exceed 19 bits. The effective address calculation uses the entire 32 bit registers for base and/or indexing and it is the responsibility of the programmer not to exceed the maximum effective address value. Any half word or byte address calculated is presented to the ARMS memory module as a full word address by dropping the 2 lsbs. The CPE extracts the appropriate portion of the word for its use.

A priority interrupt structure is provided to start, stop, or modify its software program execution. Interrupts from the CCE module (stop, start, and initialize memory) have highest priority followed by the IOP interrupts. One external interrupt is provided, routed through the CCE. This interrupt has third priority. Supervisor call and other program interrupts have lowest priority. No machine check interrupts are provided since the CCE controls fault recovery in ARMS. Each of the interrupts listed have associated old and new PSW locations at fixed locations in main memory as shown in Table 4-3 which also lists word locations for communicating with the IOP module. The I/O, external, and program interrupts can be masked out by means of bits in the PSW. In a similar fashion the fixed point overflow, Decimal overflow, exponent underflow, and significant program interrupts can be masked. The remaining interrupts can not be masked by the program. Program interrupt code assignments are shown in Table 4-4. The CPE is only interruptible at a specific point in its fetch cycle.

Referring to Figure 4-2 the CPE logic will be lumped into the following functional blocks for discussion: 1) the processor-memory interface logic unique to ARMS, over and above SUMC's requirements consisting of the voter/switch, Memory Data Register (MR), Processor Bus Output Multiplexer, and Memory Interface control logic; 2) CPE sequence control logic consisting of the Instruction Address and Format Control PROMs (IAROM), the Microprogram Read Only Memory PROMs (MROM), the ROM address register, shift register (SR) and instruction registers (IR), the Program Status Word Register (PSW) and some discrete miscellaneous control and system interface logic in addition to the PROMs; 3) the Arithmetic Logic Unit (ALU) and Exponent ALU (EALU) including their input multiplexers, and exponent registers; 4) multiplexer register logic consisting of discrete product-remainder register and multiplexer (PRR), Memory Address Register and Multiplexer (MAR), Multiply-Quotient Register and Multiplexer (MQR) plus 48 words of RAM and 16 words of PROM making up an addressable multi usage register file.

MEMORY INTERFACE LOGIC

The memory interface operates asynchronously from the remainder of the CPE once a read or a write operation is initiated by the microprogram. The memory interface control logic has a clock rate twice that of the rest of the CPE (5.0 MHz vs 2.5 MHz) This allows maximum data rates on the buses to and from memory where there are no PROM delays in the control path. The state diagrams for the memory interface control logic are discussed in the section of this report dealing with the memory. Logic is included to generate as odd parity bit over 32 bits of output data prior to transmission of the data over the processor bus and to check odd parity of data loaded into the 32 bit plus parity memory Data Register. This register is loaded in 11 bit increments as the data is received over the selected memory buses through the voter/switch. This selection is determined by the stream assignment bits received from the CCE. The MR also can be loaded from Panel Data or Address switches or from MR data with half words swapped depending on the state of the MROMs MR SELECT field.

SEQUENCE CONTROL LOGIC

The CPE sequence control logic contains the registers, PROMs, and discrete logic necessary to control the requesting of instructions and data and the sequence of these instructions execution. It operates as follows:

The instruction register and its associated multiplexers allow the execution of 16, 32, and 48 bit instruction formats. The IR is configured in three 16 bit sections with each section having an associated load indicator flip-flop IRL 00, 01, 02. When a load indicator equals "1" the data in the associated IR section is valid. The 16 msbs of the IR (Bits 00,..., 15) can be loaded with MR data, MR data shifted left by 16 bits, the 8 lsbs of the ALU shifted left by 16 bits or IR data shifted left by 0, 16, or 32 bits depending upon the states of the 3 bits of the MROMs IR control field (IRS 0, 1, 2), the full word boundary indicator (MAR30), the state of this section of the IRs load indicator (IRL00), and the bits of the instruction up-code indicating whether or not the instruction has an RR (16 bit) format. The middle 16 bits of the IR can be loaded from the MR, the MR right shifted by 16 bits, or the IR left shifted by 0 or 16 bits depending upon the states of XIRS 0, 1, 2 and IRL00. The 16 lsbs of the IR can be loaded from MR right shifted by 16 bits or by unshifted IR data depending upon the state of IRS 0, 1, 2 and IRL00. The overall effects of loading the IR and its load indicators before and after execution of the IR load microinstruction in the fetch subroutine are shown in Table 4-5. An overview of IR interconnection is shown in Figure 4-3.

The Program Status Word is implemented according to the specifications given for the IBM system 360 (Ref 1) except for the Protect Key field which does not apply to the ARMS EB. It holds all information not contained in storage or registers but required for proper program execution. By storing the PSW the program can preserve the detailed status of the CPE for subsequent use when the program is interrupted or a CPE must be replaced during reconfiguration. The PSW format is shown in Figure 4-4. System mask bits 0, 1, 7 allow masking I/O channel 0, 1 or external interrupts respectively. Bit 11 allows masking the CCEs Halt interrupt until ongoing interrupt handling is completed to avoid confusion during the reconfiguration process. It is controlled by the Set Halt Mask instruction unique to ARMS. Bit 12 determines if ASCII or BCD results are generated during decimal instructions. Bit 14 is the CPE run flip-flop, Bit 15 determines if the CPE is in the problem state (bit on) or the supervisor state. Bits 32, 33, the instruction length code, indicates the number of half words in the last interpreted instruction when a program or supervisor call interruption occurs, Bits 34, 35 store the condition code according to system 360 specs (see Ref 1). Bits 36, ..., 39 (Program Mask) allow masking interrupts due to fixed point overflow, decimal overflow, exponent underflow, and significance respectively. The instruction address field is taken from the 24 lsbs of location 25 of the MURs which is the instruction address register. The interruption codes (bits 16-31) are from MUR constant storage

locations. The remaining bits of the PSW are identically equal to zero in the ARMS breadboard.

The shift register is used to control the sequencing of instructions which require an iteration of events such as shifts, and floating point alignment and normalization. This logic is shown in Figure 4-5. Shift register outputs related to byte, character, and bit shifts are decoded. Shift counting in various increments can be done by either the exponent adder (EALU) or the main ALU depending upon which multiplexer inputs are selected. The 6 lsbs of the ALU can be selected for general shift counting of up to 63 bits or the 2 lsbs of the ALU can be selected for fast half word shifts in certain instructions. The unshifted output of the exponent adder can be selected for control of the alignment of floating point mantisses by subtracting the exponents of each number in the EALU. The EALU output left shifted by 2 bits can be used for character manipulation instructions. Finally the output of the normalize logic can be selected to determine how many blocks of leading zeros remain in a floating point result during normalization. The SR=0 test is enabled when the MISC B field of the MROM is equal to 3. The SR multipexer is controlled by the 3 bit SR SELECT field of the MROM.

The miscellaneous control and interface logic performs control functions required over and above those performed directly by the microprogram and interfaces with other ARMS modules except for the memory. These functions include program wait and proceed, test and set, and carrysave, preprocessing of branch conditions, detection of instruction format errors such as R-field and branch exceptions and of arithmetic errors such as overflow, decoding of the PROM miscellaneous control fields, introduction of panel functions into the CPE and synchronization of CCE and IOP signals to and from the CPE.

The IAROM is addressed by the instructions op-code and provides an indirect address pointer to the starting location in MROM corresponding to that instruction along with format control flags specifying if the instruction will operate on half , full, or double word data, if an operand must be read from memory, if the operand is in floating or fixed point format, if tests must be made to determine if the R_1 or R_2 fields of an instruction are even (this is necessary when double words are being fetched from the MURs since double words can start only on even addresses), and finally to determine if an instruction is priveleged or non-priveleged (i.e. whether or not the CPE must be in the supervisor state to execute it). To save parts the IAROM uses three 512 word by

4 bit PROMs in a 256 word by 24 bit configuration by selecting 2 words out of each PROM for each op-code. Which half is selected is determined by the IARSEL flip-flop which is controlled by the ROMAR select field of the MROM. Once the address pointer has been accessed from the IAROM it is placed in the ROM Address Register (ROMAR). This register can also be loaded from the BRANCH ADDRESS field of the MROM or from the 11 lsbs of the ALU output depending upon the state of the ROMAR select field (1, 2, 3 respectively) allowing for both non-conditional and conditional program branches. When a multiplexer input is not selected (ROMAR SELECT=0) the register is connected as a binary counter allowing the MROM to step sequentially through the microprogram. The format and mnemonic definitions for the IAROM are shown in Figure 4-6.

The MROM contains the main CPE microprogram in a 1024 word x 96 bit configuration as shown in Figures 4-7 and 4-8. The first figure lists control field definitions, the second lists the associated mnemonics. The various branch conditions as a function of the BRANCH CONDITION field of the MROM and the MISCELLANEOUS CONTROL field functions are listed in our supplementary microprogram documentation. The sense and wait bits 76 ... 79 are decoded to cause a sense and wait condition to be dependant on nothing, proceed, memory data transfer complete, exponent adder=0, or I/O interrupt acknowledge for codes 0, ..., 5 respectively. The remaining control fields are discussed in connection with the CPE functional units that they control.

ARITHMETIC LOGIC

The main Arithmetic Logic Unit (ALU) in the ARMS EB consists of 74S181 MSI ALUs with external carry-look-ahead logic for maximum speed with a 4 input multiplexer at the ALU's "A" input and an 8 input multiplexer at its "B" input. The ALU is shown in Figure 4-9. This unit performs 32 bit, 2's compliment addition and subtraction plus logical AND, OR, Exclusive-OR and miscellaneous operations. The ALU function is controlled by the ALU FUNCTION CONTROL field of the MROM. The 6 bits of this field directly control 74S181 mode, forced carry, and function select S_3, ..., S_0 respectively from left to right except during the divide microprogram where the lsb of the MQ register forces function "A plus B" when it is "1" and "A minus B" when it is "0".

The adder "A" input multiplexer is a 32 bit wide multiplexer providing a choice of "all 0"; PRR_{0-31}; MAR_{0-31}; MUR_{0-31}; or ER_{0-7}, PRR_{8-31} inputs under control of the 3 bit MROM ALU "A" INPUT field. The latter code is used in floating point operation, the others are used in a variety of instructions.

The Adder "B" input multiplexer is a 32 bit wide multiplexer providing a choice of MR_{0-31}; MR_{16-31}, to $ALUB_{16-31}$ with MR_{16} to more significant ALU MUX inputs for half-word operations; MUR_{0-31}; IR_{20-31} to $ALUB_{20-31}$ with "0" to more significant ALU MUX inputs for operations involving the Displacement fields of an instruction; IR0-7 (op-code) to $ALUB_{0-7}$ and IR_{8-15} to $ALUB_{24-31}$ with "0" to other ALU MUX inputs for performing arithmetic or logic operations on bits 8, ..., 15 (I_2 or L fields) of an instruction; PSW_{0-15} to $ALUB_{0-15}$ and MUR_{16-31} to $ALUB_{16-31}$ for storage of the first half of the PSW and finally PSW_{33-39} to $ALUB_{0-7}$ and MUR_{8-31} to $ALUB_{8-31}$ for storage of the second half of the PSW. This multiplexer is controlled by the MROM ALU "B" INPUT field.

The Exponent ALU (EALU) in the ARMS EB also consists of 74S181 MSI ALUs with a 4 input multiplexer at the ALU's "A" input and an 8 input multiplexer at its "B" input. The EALU is shown in Figure 4-10. This unit performs 8 bit 2's compliment addition and subtraction on the exponent field of floating point number. These exponents are represented in excess 64 notation. The EALU function is controlled by the EALU FUNCTION CONTROL field of the MROM. The 6 bits of the field perform the same functions for the EALU as the corresponding bits in the ALU FUNCTION CONTROL field do for the ALU.

The EALU "A" input multiplexer is an 8 bit wide multiplexer providing a choice of "all 0", ER_{0-7} (for exponent manipulation), or SR_{0-7} (for shift register incrementation) under control of the MROM EALU "A" INPUT field. The EALU "B" input multiplexer is an 8 bits wide multiplexer providing a choice of zero sign plus MR_{1-7} to $EALUB_{1-7}$, zero sign plus MUR_{1-7} to $EALUB_{1-7}$ or ER_{0-7} all for exponent manipulation or SR_{0-5} to $EALUB_{2-7}$ with other bits zero for use in conjunction with the shift register for character manipulation instructions, or the 8 lsbs of the Branch Address field of the MROM when the MROM "PRR SR COUNT" field of the MROM equals "0" or the output of the SR count decode logic when "PRR SR COUNT"=1 for iterative and shift operations, or of the output of the normalize logic for floating point normalization.

The output of the EALU is routed directly to the SR multiplexer and is stored in the 8 bit Exponent Register when the MROM's ER Load bit is on. The ER register holds the result for further operations by the EALU when this bit is off.

MULTIPLEXER – REGISTER LOGIC

The Multiple Usage Registers (MUR) are shown logically in Figure 4-11. Their address assignments are shown in Figure 4-12. They are organized as 16 fixed point 32 bit multi usage registers, 4 floating point 64 bit multi usage register pairs, dedicated program counter and instruction address register, 16 temporary scratchpad 32 bit registers, (all from a 33 bit by 48 word RAM), plus 16 constant locations (contained in PROMs) which are used as masks in certain instructions and as interrupt codes in the PSW. These constants are listed by location in Table 4-6. The MURs are addressed in blocks of 16 locations by the R_1, R_2, B_1, B_2, or X_2 fields of the CPE instructions. These addresses are obtained from the instruction register and can be used direct or, in the case of even addresses, can be incremented by one in the MUR address multiplexer.

When a floating point instruction is to be executed (as designated by the FLTPT bit of the IAROM being on) the address specified by the instruction field is incremented by 16. In addition to instruction fields the MUR address can be obtained from the 6 lsbs of the Panel Data switches, or of the MQR, or from the MUR ADDRESS field of the MROM. The MUR address source is selected by the 3 bit MUR ADDRESS SELECT field of the MROM. The MUR write mode is enabled by the MUR WRITE bit of the MROM. When this bit is off the MURs are in this read mode. When valid data is to be read out of the MURs the MUR READ bit is turned on enabling a parity check to be made over the MUR outputs. The data source for MUR WRITE operations is selected by the MUR DATA SELECT field of the MROM. This data can come from the following sources: Panel Data switches, PRR, MAR, or MQR, PRR multiplexer, or $PAGE_{16,17}$ specifying which memory bank to initialize when the CPE processes an initialize memory interrupt from the CCE. The output stage of the MUR bank in a set of latches which can change state except during MUR WRITE operations. This allows MUR data to be either passed through or modified by the ALU or shift multiplexers and fed back to the same MUR location or to a new MUR location in a single micro-instruction. The flexibility as to address and data sources for the MURs significantly enhances CPE processing capabilities. By checking parity at the MUR output most

errors in the MURs or in its various data sources can be detected for a very small increase in complexity as noted in our architecture studies.

In addition to the MURs, the CPE contains 3 discrete hardware multiplexer-register assemblies: the PRR, MAR and MQR registers. This logic is shown in Figure 4-13. The first 2 registers are used respectively for holding memory data and address during accesses to main memory and are used together or separately for 64 and 32 bit manipulations during arithmetic and logical operations when the PRR DOUBLE bit of the MROM is on. The MQR is used to hold the multipler during multiply operations to develop the quotient during divide operations, and for general temporary storage outside the MURs during various other operations. MAR data can be shifted into the MQR and visa versa when the MQR DOUBLE bit of the MROM is on. Together this 3 register structure can handle 96 bit data manipulations. Data in any combination of the 3 registers can either be changed or left alone depending on the states of the PRR, MAR, and MQR LOAD bits of the MROM.

The PRR Multiplexer can perform either logical or arithmetic right shifts depending on the state of the PRR LOGICAL bit of the MROM. When this bit is a zero vacated bits are filled with the sign bit of the ALU. This multiplexer is 33 bits wide and has 8 inputs allowing shifting the main ALU output either left or right by 0, 1, 4 or 8 bits. It also allows for an all zero output. It is controlled by the MROM PRR SELECT field. The output of the PRR Multiplexer goes directly to the PRR register and also to the MAR and MQR multiplexers.

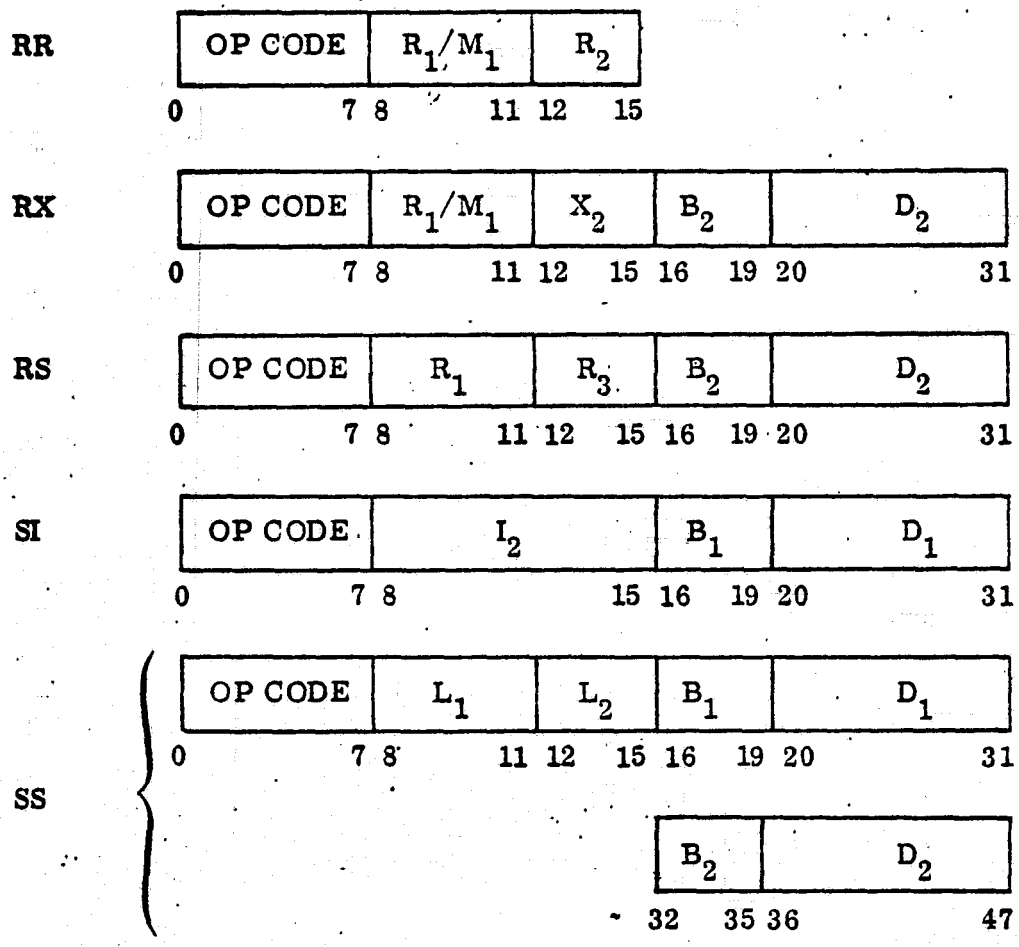
The 33 bit wide MQR multiplexer also allows selection of zeros, or MQR data left or right shifted by one bit or of quotient (ALU carry) bits into bit 31 during certain steps of the divide algorithm. This multiplexer is controlled by the MQR SELECT field of the MROM.

The MAR multiplexer is an 8 input 33 bit wide multiplexer controlled by the MAR SELECT field of the MROM. It allows selection of PRR multiplexer data of zeros, or of MAR data shifted either right or left by 1, 4, or 8 bits.

Parity bits are generated over the combinations of bits selected into each of the 3 registers and are stored whenever corresponding data is updated in a register in accordance with the design discussed in our architecture study (Ref 2). Parity is checked whenever stored data is retrieved from MURs or from main memory.

COMPONENT COUNT

The overall component count for the CPE module (Table 4-2) indicates that in an LSI version at least half of its complexity would lie in its read only memories (ROMs). The overall CPE equivalent gate count of 25, 843 is approximately 3 times that of the CPE proposed in our previous phase report. The bulk of this increase is in the ROM area where complexity increased tenfold due to the added requirements of the IBM system 360 instruction set which were not assumed in the earlier study. The remainder of the increase was due to the added performance provided in the instruction fetching, and multi-usage register areas, the requirements of the system 360 instruction set, and inefficiencies due to use of existing MSI parts. This latter reason would not be a factor in an LSI implementation of course. It is not anticipated that an LSI version of ARMS using this CPE design would have any difficulty in operating successfully over a 5 year mission despite the increase in performance and complexity.



OP CODE - Instruction Operation Code

R_1, R_2, R_3 - Operand register specification

X_2 - Index register specification

M_1 - Mask

L_1, L_2 - Operand length specification

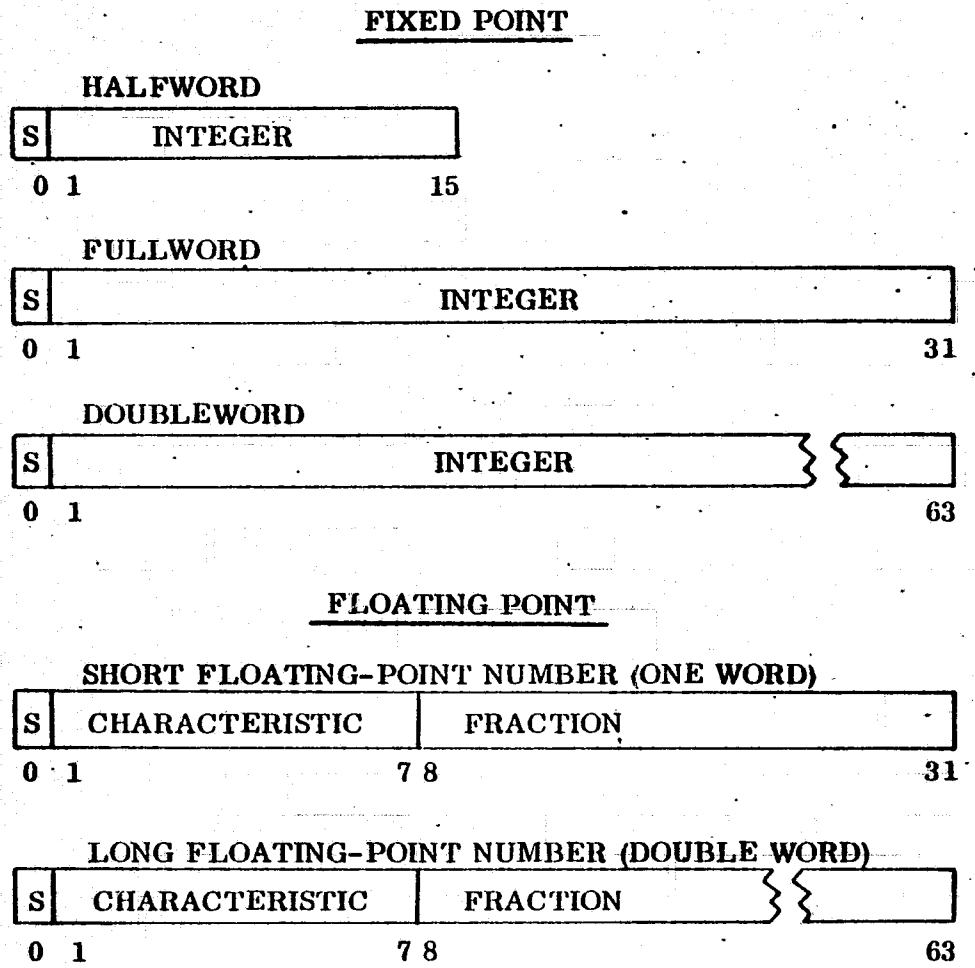
I_2 - Immediate data

B_1, B_2 - Base register specification

D_1, D_2 - Displacement

Figure 4-1a
INSTRUCTION FORMATS

Figure 4-1b
DATA FORMATS



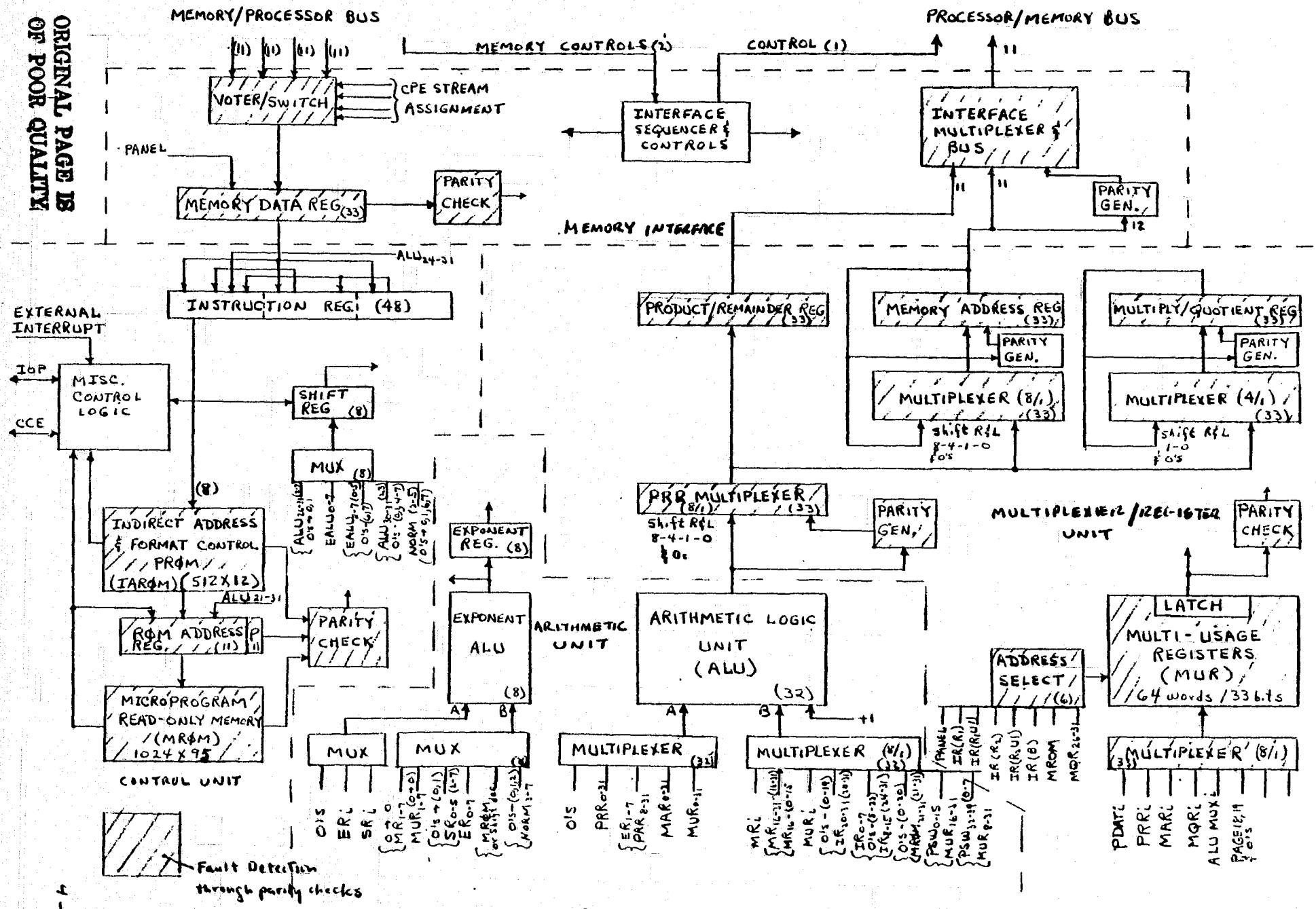


Figure 4-2

CENTRAL PROCESSING ELEMENT (CPE)

BLOCK DIAGRAM

Figure 4-3

INSTRUCTION REGISTER

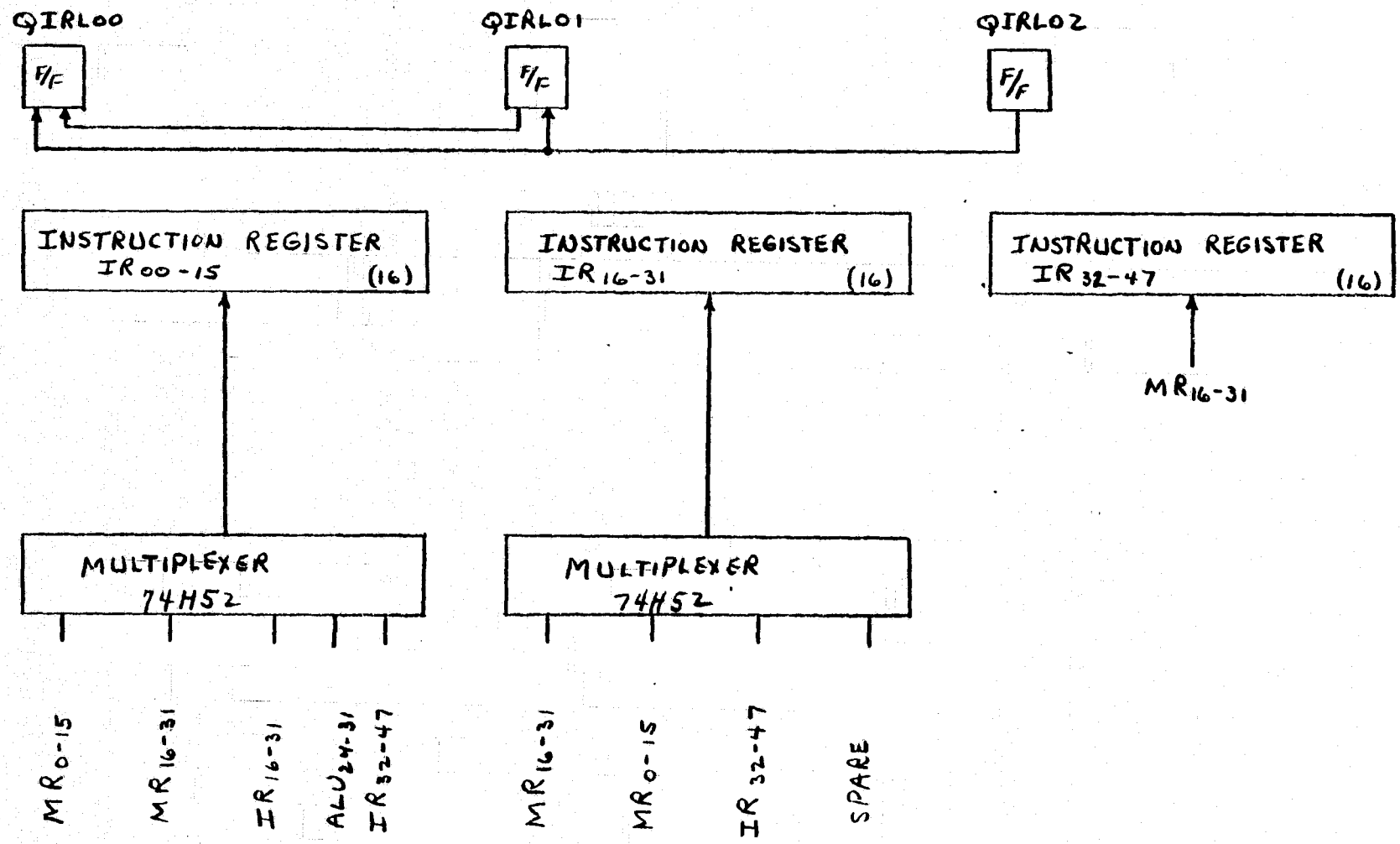
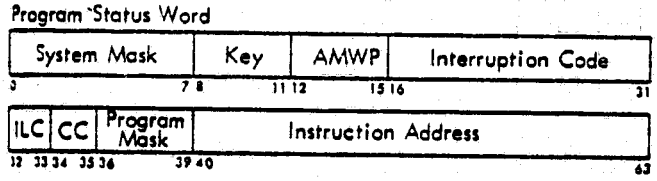


Figure 4-4

Program Status Word



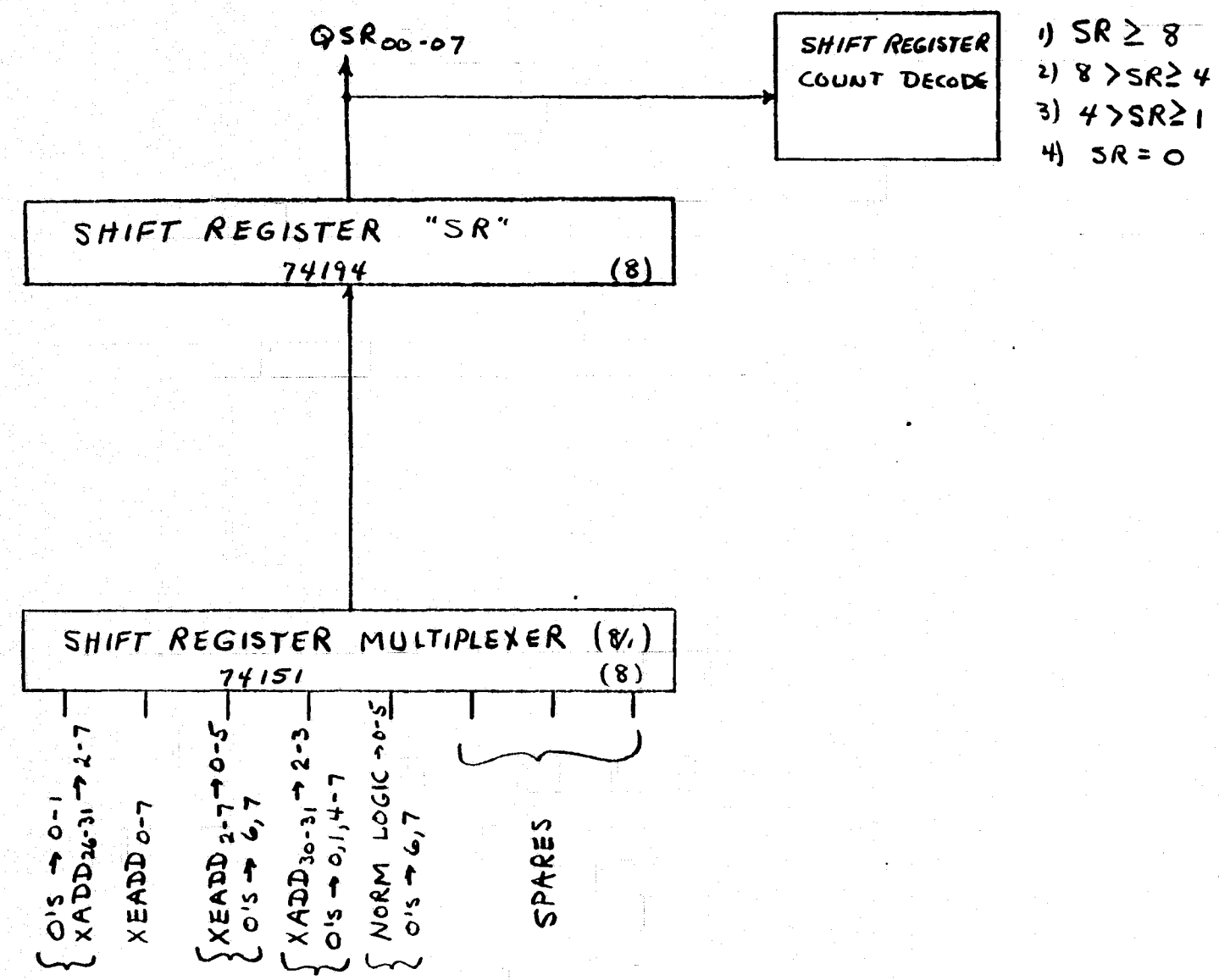
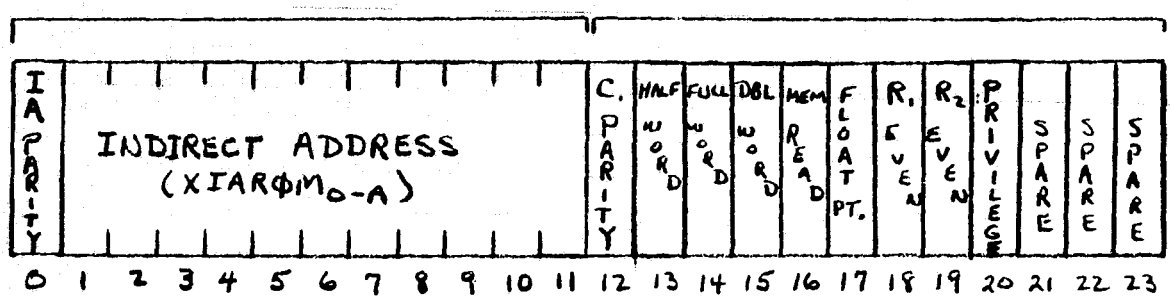


Figure 4-5

SHIFT REGISTER

Figure 4-6

INDIRECT ADDRESS & CONTROL PRØM FORMAT



BIT	MNEMONIC	FUNCTION
0	XIARØMP	INDIRECT ADDRESS PARITY [Bits 1-11] ODD
1-11	$XIAR\emptyset M_{0-A}$	INDIRECT ADDRESS POINTER TO MRØM
12	XIACPAR	CONTROL PARITY [Bits 13-23] ODD
13	XHAFWRD	HALFWORD DATA WORD
14	XFULWRD	FULLWORD DATA WORD
15	XDBLWRD	DOUBLE-WORD DATA WORD
16	XMEMRED	MEMORY DATA READ
17	XFLTPT	FLOATING POINT INSTRUCTION
18	XR1EVEN	R1 FIELD EVEN
19	XR2EVEN	R2 FIELD EVEN
20	XPRIV	PRIVILEGE INSTRUCTION
21-23	SPARES	SPARE BITS

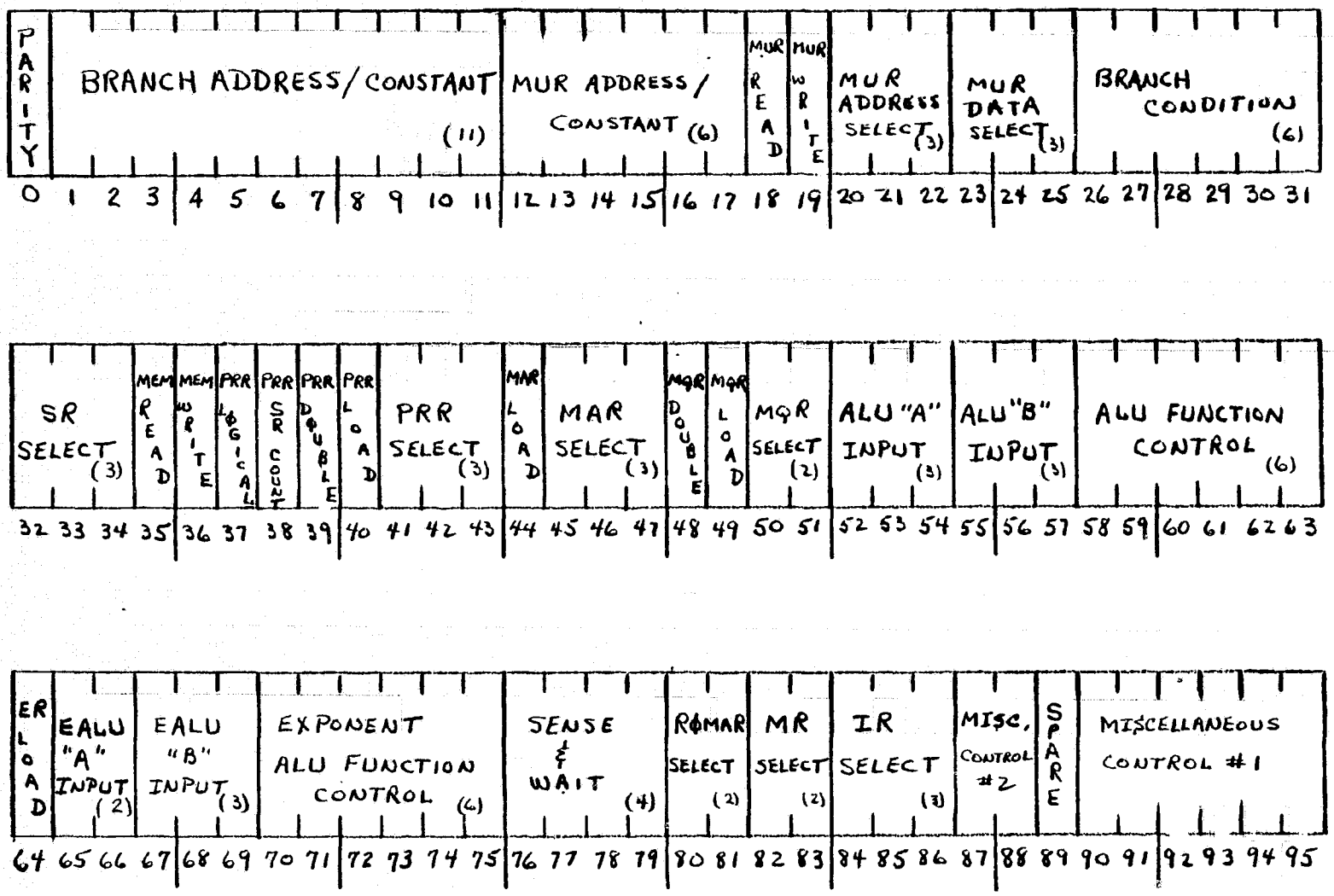


Figure 4-7

MICRO-PROGRAM DATA FORMAT

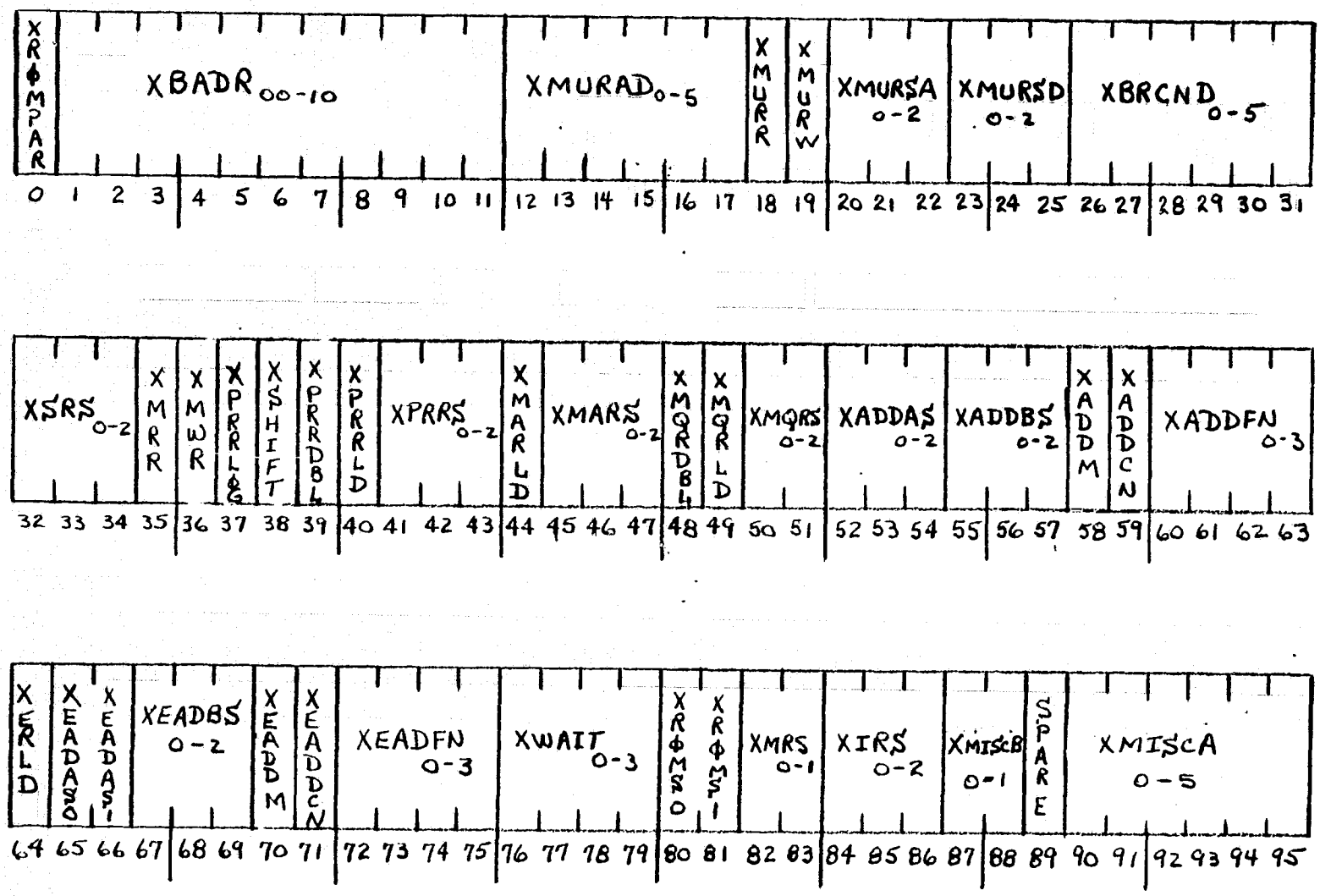
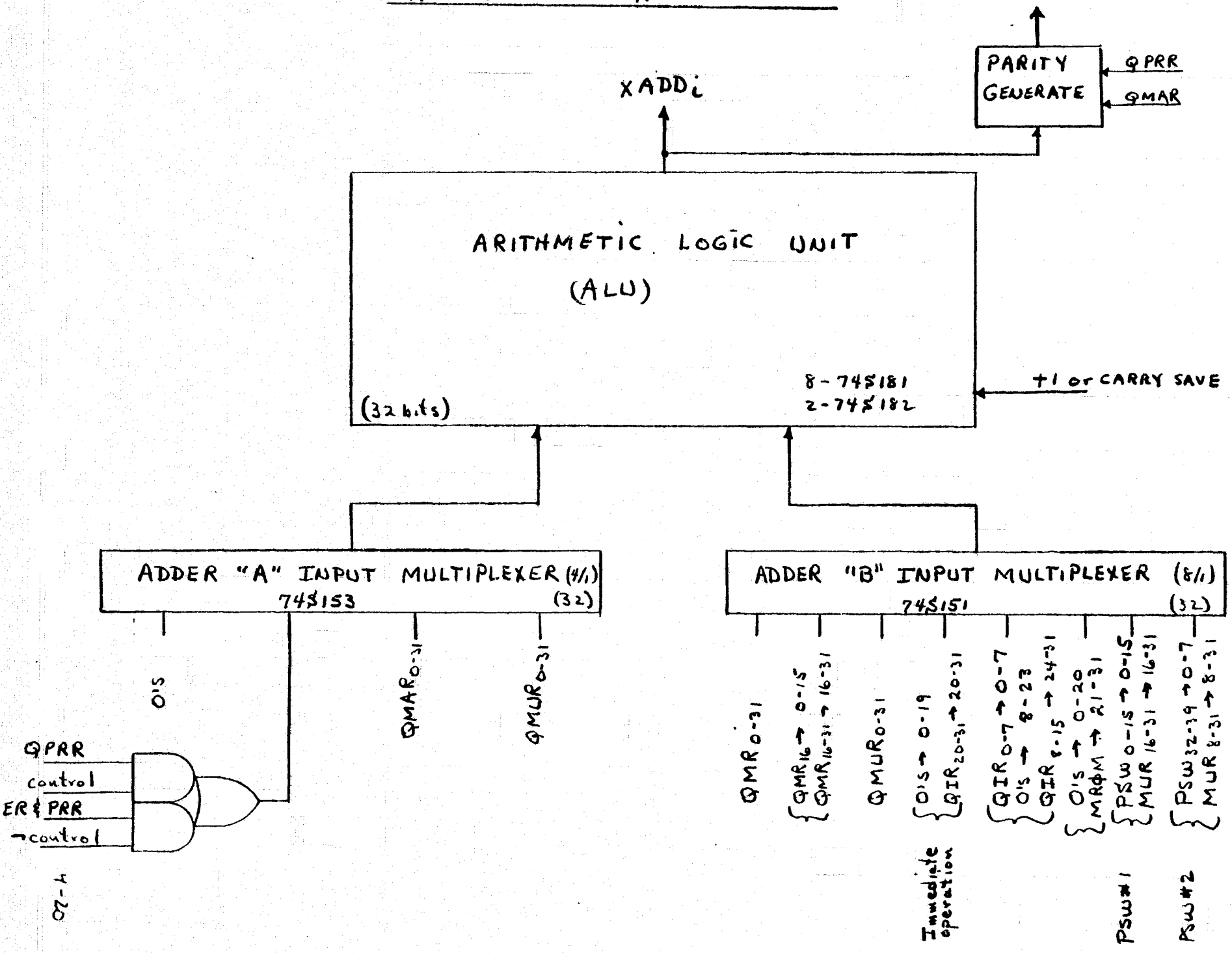


Figure 4-8

MICRO-PROGRAM DATA FORMAT & MNEMONICS

Figure 4-2

ARITHMETIC LOGIC UNIT



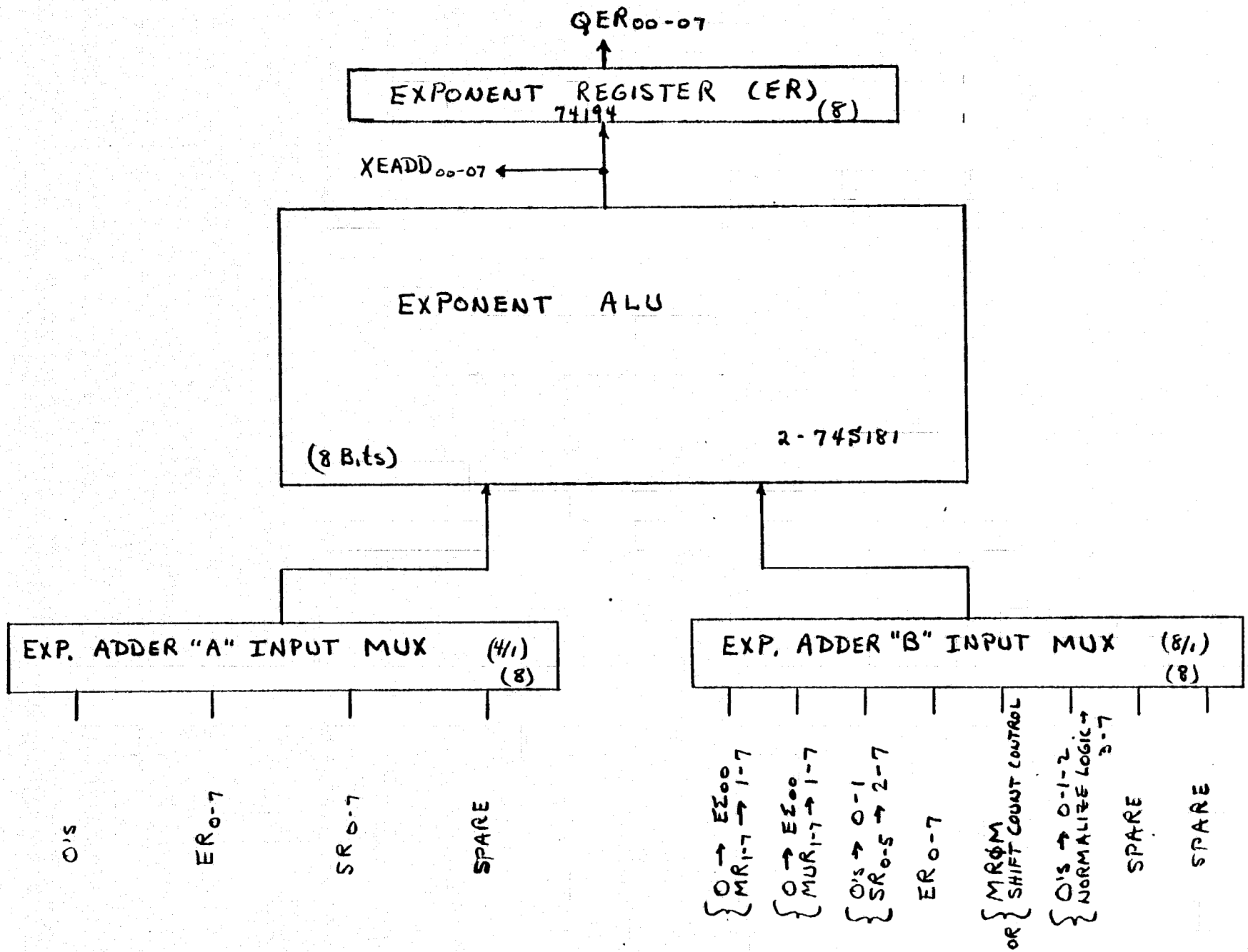


Figure 4-10

EXPONENT ARITHMETIC UNIT

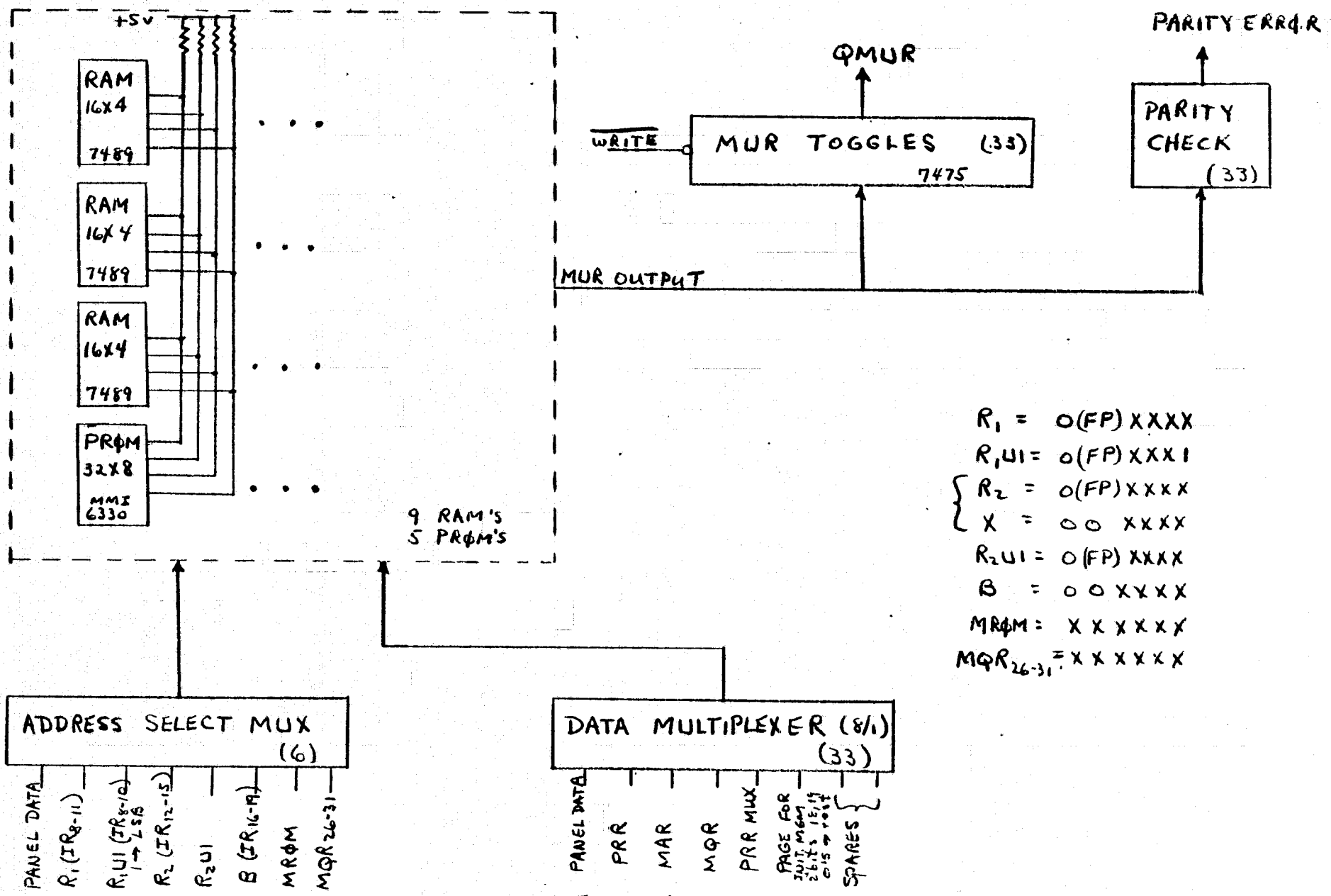


Figure 4-11

MULTIPLE-USAGE REGISTERS (MUR)

Figure 4-12

MULTI-USAGE REGISTER ASSIGNMENTS

ADDRESS	FUNCTION
0 - 15	MULTI-USAGE REGISTERS (16)
16 - 23	FLOATING POINT REGISTERS (4 - 64 BIT REGISTERS)
24	PROGRAM COUNTER
25	INSTRUCTION ADDRESS REGISTER
26 - 31	SPARE (6)
32 - 47	TEMPORARY REGISTERS (16)
48 - 63	CONSTANTS (16)

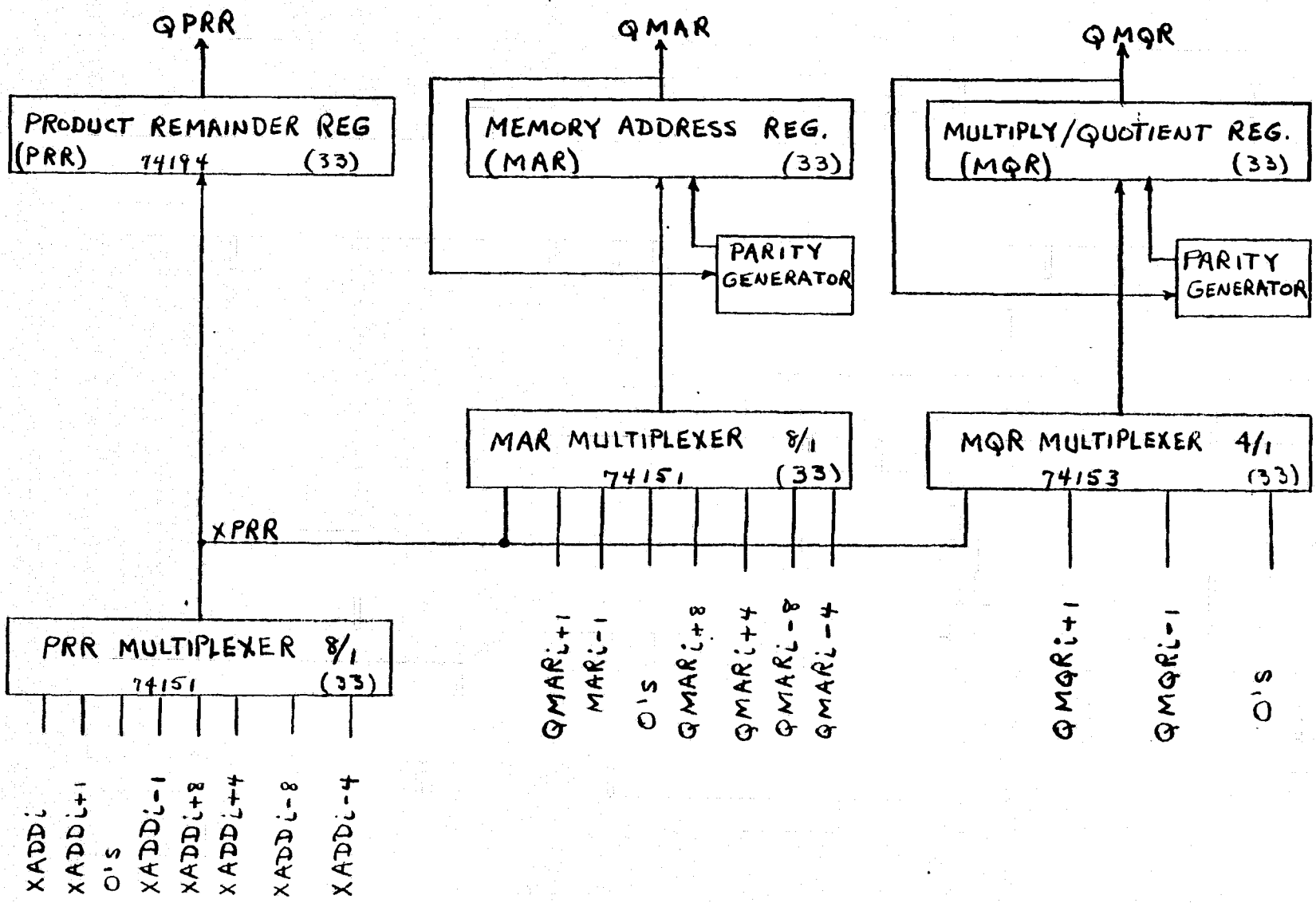


Figure 4-13
PRR - MAR - MQR REGISTERS

Table 4-1a

a) Standard Instruction Set

NAME	MNEMONIC	TYPE	CODE
Add	AR	RR	1A
Add	A	RX	5A
Add Halfword	AH	RX	4A
Add Logical	ALR	RR	1E
Add Logical	AL	RX	5E
AND	NR	RR	14
AND	N	RX	54
AND	NI	SI	94
AND	NC	SS	D4
Branch and Link	BALR	RR	05
Branch and Link	BAL	RX	45
Branch on Condition	BCR	RR	07
Branch on Condition	BC	RX	47
Branch on Count	BCTR	RR	06
Branch on Count	BCT	RX	46
Branch on Index High	BXH	RS	86
Branch on Index Low or Equal	BXLE	RS	87
Compare	CR	RR	19
Compare	C	RX	59
Compare Halfword	CH	RX	49
Compare Logical	CLR	RR	15
Compare Logical	CL	RX	55
Compare Logical	CLC	SS	D5
Compare Logical	CLI	SI	95
Convert to Binary	CVB	RX	4F
Convert to Decimal	CVD	RX	4E
Divide	DR	RR	1D
Divide	D	RX	5D
Exclusive OR	XR	RR	17
Exclusive OR	X	RX	57
Exclusive OR	XI	SI	97
Exclusive OR	XC	SS	D7
Execute	EX	RX	44
Halt I/O	HIO	SI	9E
Insert Character	IC	RX	43
Load	LR	RR	18
Load	L	RX	58
Load Address	LA	RX	41
Load and Test	LTR	RR	12
Load Complement	LCR	RR	13
Load Halfword	LH	RX	48
Load Multiple	LM	RS	98
Load Negative	LNR	RR	11
Load Positive	LPR	RR	10
Load PSW	LPSW	SI	82
Move	MVI	SI	92
Move	MVC	SS	D2
Move Numerics	MVN	SS	D1
Move with Offset	MVO	SS	F1
Move Zones	MVZ	SS	D3

Table 4-1b

NAME	MNEMONIC	TYPE	CODE
Multiply	MR	RR	1C
Multiply	M	RX	5C
Multiply Halfword	MH	RX	4C
OR	OR	RR	16
OR	O	RX	56
OR	OI	SI	96
OR	OC	SS	D6
Pack	PACK	SS	F2
Set Program Mask	SPM	RR	04
Set System Mask	SSM	SI	80
Shift Left Double	SLDA	RS	8F
Shift Left Single	SLA	RS	8B
Shift Left Double Logical	SLDL	RS	8D
Shift Left Single Logical	SLL	RS	89
Shift Right Double	SRDA	RS	8E
Shift Right Single	SRA	RS	8A
Shift Right Double Logical	SRDL	RS	8C
Shift Right Single Logical	SRL	RS	88
Start I/O	SIO	SI	9C
Store	ST	RX	50
Store Character	STC	RX	42
Store Halfword	STH	RX	40
Store Multiple	STM	RS	90
Subtract	SR	RR	1B
Subtract	S	RX	5B
Subtract Halfword	SH	RX	4B
Subtract Logical	SLR	RR	1F
Subtract Logical	SL	RX	5F
Supervisor Call	SVC	RR	0A
Test and Set	TS	SI	93
Test Channel	TCH	SI	9F
Test I/O	TIO	SI	9D
Test Under Mask	TM	SI	91
Translate	TR	SS	DC
Translate and Test	TRT	SS	DD
Unpack	UNPK	SS	F3

b) Floating Point Instructions

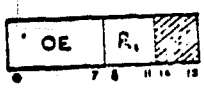
NAME	MNEMONIC	TYPE	CODE
Add Normalized (Short)	AE	RX	7A
Compare (Short)	CE	RX	79
Divide (Short)	DE	RX	7D
Load (Short)	LE	RX	78
Multiply (Short)	ME	RX	7C
Store (Long)	STD	RX	60
Store (Short)	STE	RX	70
Subtract Normalized (Short)	SE	RX	7B

Table 4-1c

c) Added ARMS Instructions

Set Halt Mask

SHM R_1 (RR)



Bits 8-11 of the general register specified by the R_1 field replace bits 8-11 of the current PSW.

Bits 0-7 and 12-31 of the register specified by the R_1 field are ignored. The contents of the register specified by the R_1 field remain unchanged.

The instruction permits the setting of the halt mask in either the problem or supervisor state.

Condition Code: The code remains unchanged.

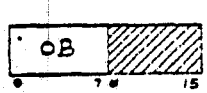
Program Interruptions: None.

Programming Note

Bit 11 of the general register may have been loaded from the PSW by BRANCH AND LINK. Bit 11 of the PSW is the halt mask.

Program Progress Alert

PPA (RR)



The instruction allows coordination of the CPE and CCE by informing the CCE of a convenient roll-back point and allowing the CCE to halt the CPE.

Bits 8-11 of the current PSW are set to ones, enabling the halt CPE interrupt. Bit positions 0-7 and 12-63 of the current PSW remain unchanged. Subsequently, the current PSW is stored in location 136. The available line for the CPE is used to inform the CCE of this instruction. If the CPE is not halted, normal instruction sequencing proceeds with the updated instruction address. The instruction is valid in both problem and supervisor state.

Condition Code: The condition code remains unchanged.

Program Interruptions: None.

TABLE 4-2. CPE COMPONENT COUNT

Function	Gates	Flip-Flop	RAM bits	ROM bits	Fail Rate* Equiv Gates	Simplex Coverage
1. Voter/Switch	289	0	-	-	289	1.0
2. Memory Data Reg (MR)	650	33			848	1.0
3. Proc Bus Output MUX	164	0			164	1.0
4. Product-Remainder Reg and MUX (PRR)	1,319	33			1,517	1.0
5. Memory Addr Reg and MUX (MAR)	893	33			1,091	1.0
6. Multiply-Quotient Reg (MQR)	488	33			686	1.0
7. Multiple Usage Registers (MUR)	865	38	1,584	556	2,747	0.95
8. IAROM	286	16		6,144	1,150	0.9
9. MROM	799	0		98,304	13,087	0.9
10. Instruction Register (IR)	551	51			857	0
11. Arithmetic Logic Unit (ALU)	1,456	0			1,456	0
12. Exponent ALU (EALU)	438	0			438	0
13. Memory Interface Control	42	14			126	0
14. Shift Register	191	8			239	0
15. Program Status Word	231	24			375	0
16. Misc Control Logic and Sys Interface	737	6			773	0
17. Scanout and Panel Functions	(1,424)				Not Applicable	
	9,399	289	1,584	105,004	25,843	0.775

*Assumes failure rate of gate = that of ram bit = that of 8 ROM bits = that of 1/6 flip-flop

Table 4-3

PERMANENT MEMORY ASSIGNMENTS

Address			Length	Purpose
0	0000	0000	double word	Initial program loading PSW
8	0000	1000	double word	Initial program loading CCW1
16	0001	0000	double word	Initial program loading CCW2
24	0001	1000	double word	External old PSW
32	0010	0000	double word	Supervisor call old PSW
40	0010	1000	double word	Program old PSW
48	0011	0000	double word	Machine check old PSW
56	0011	1000	double word	Input/output old PSW
64	0100	0000	double word	Channel status word
72	0100	1000	word	Channel address word
76	0100	1100	word	Unused
80	0101	0000	word	Timer
84	0101	0100	word	Unused
88	0101	1000	double word	External new PSW
96	0110	0000	double word	Supervisor call new PSW
104	0110	1000	double word	Program new PSW
112	0111	0000	double word	Machine check new PSW
120	0111	1000	double word	Input/output new PSW
128	1000	0000	double word	Stop CPE old PSW
136	1000	1000	double word	Start CPE new PSW
144	1001	0000	double word	Initialize Memory new PSW
152	1001	1000	word	Channel Instruction word

Table 4-4

PROGRAM INTERRUPT CODE ASSIGNMENTS

Interruption Code		Program Interruption
1	00000001	Operation
2	00000010	Privileged operation
3	00000011	Execute
4	00000100	Protection
5	00000101	Addressing
6	00000110	Specification
7	00000111	Data
8	00001000	Fixed-point overflow
9	00001001	Fixed-point divide
10	00001010	Unused
11	00001011	Unused
12	00001100	Exponent overflow
13	00001101	Exponent underflow
14	00001110	Significance
15	00001111	Floating-point divide

TABLE 4-5. INSTRUCTION REGISTER LOAD CONTROL

t						t+1			
XIRS0	XIRS1	XIRS2	IRL00	MAR30	RR Format	IRL00	IRL01	IRL02	Function
0	0	1	0	0		1	1	IRL02	$MR_{0-15} \rightarrow IR_{0-15}$ $MR_{16-31} \rightarrow IR_{16-31}$
0	0	1	0	1		1	IRL01	IRL02	$MR_{16-31} \rightarrow IR_{0-15}$ $X \rightarrow IR_{16-31}$
0	0	1	1	X		1	1	1	$MR_{0-15} \rightarrow IR_{16-31}$ $MR_{16-31} \rightarrow IR_{32-47}$
0	1	0			1	IRL01	IRL02	0	$IR_{16-31} \rightarrow IR_{0-15}$ $IR_{32-47} \rightarrow IR_{16-31}$ EXIT FROM RR INSTRUCTION
0	1	0			0	IRL02	0	0	$IR_{32-47} \rightarrow IR_{0-15}$ $X \rightarrow IR_{16-31}$ EXIT FROM $\overline{RR}$ INSTRUCTION
0	1	1				IRL00	IRL02	0	$IR_{0-15} \rightarrow IR_{0-15}$ $IR_{32-47} \rightarrow IR_{16-31}$ USE WITH SS FORMAT
1	0	0				IRL00	IRL01	IRL02	$XADD_{24-31} \rightarrow IR_{08-15}$
1	1	1				0	0	0	EXIT FROM A BRANCH OR EXECUTE
0	0	0				IRL00	IRL01	IRL02	Quiescent State

X = Don't Care

TABLE 4-5. INSTRUCTION REGISTER LOAD CONTROL (Continued)

XIRS0	XIRS1	XIRS2	Code Name	Definition
0	0	0	-	Quiescent State
0	0	1	XIRFMR	LOAD IR from MR
0	1	0	XIRSHFT	IR Shift for Instruction Exit
0	1	1	XIRFIR	$IR_{32-47} \rightarrow IR_{16-31}$
1	0	0	XIRFADD	LOAD IR from ADDER
1	0	1	-	UNUSED
1	1	0	-	UNUSED
1	1	1	XIROFF	EXIT from Branch or Execute

Table 4-6

MUR CONSTANTS LOCATIONS

SYMBOL	LOCATION	VALUE
K_0	48	00FFFFFF
K_1	49	FF00FFFF
K_2	50	FFFF00FF
K_3	51	FFFFFF00
K_4	52	0FFFFFFF
K_5	53	FF0FFFFF
K_6	54	FFFF0FFF
K_7	55	FFFFFF0F
K_8	56	F0FFFFFF
K_9	57	FFF0FFFF
K_{10}	58	FFFFF0FF
K_{11}	59	FFFFFFF0
K_{12}	60	0000FFFF
$K_{13} - K_{15}$	61 - 63	SPARES

CPE INTERFACE

CPE – MEMORY

1.	Processor Buss (11)	CPE to Memory	-TPMB_ _ _
2.	Memory Buss (11x4)=(44)	Memory to CPE	-TMPB_ _ _
3.	Memory Request (1)	CPE to Memory	-TCPREQ_
4.	Memory Request Acknowledge (4)	Memory to CPE	-TMRESP_
5.	Data Available (4)	Memory to CPE	-TDAVAL_

CPE – CCE

1.	CPE Stream Assignment (12)	CCE to CPE	-TSASN_ _
2.	Power Control (1)	CCE to CPE	-TPWRON_
3.	Clock, Sync (2)	CCE to CPE	-TCLK, TSYNC
4.	CPE Fault (1)	CPE to CCE	-TCPFLT_
5.	Start CPE Interrupt (1)	CCE to CPE	-TSTART_
6.	Stop CPE Interrupt (1)	CCE to CPE	-TSTOP_
7.	Initialize Memory Interrupt (1)	CCE to CPE	-TINITM_
8.	Initialize Memory Control (2)	CCE to CPE	-TPGCTL_
9.	CPE Available/Rollback Pace (1)	CPE to CCE	-TRBPAC_
10.	Panic Halt Interrupt (1)	CCE to CPE	-TPANICH
11.	External Interrupt (1)	CCE to CPE	-TEXTIN_
12.	External Interrupt Acknowledge (1)	CPE to CCE	-TEXIAK_

CPE – IOP

1.	IOP Memory Request Inhibits (1/IOP)	IOP to CPE	-TIOINH_
2.	CPE Memory Request Inhibit (1)	CPE to IOPs	-TCPINH_
3.	Initiate I/O Instruction (1)	CPE to IOP	-TIIOI_
4.	I/O Instruction Accepted (1/IOP)	IOP to CPE	-TIOIAK_
5.	I/O Interrupt (2/IOP)	IOP to CPE	-TIOINT_
6.	I/O Condition Code (2/IOP)	IOP to CPE	-TIOPCC_
7.	I/O Interrupt Acknowledge (2)	CPE to IOP	-TINTAK_

CPE – MAINTENANCE PANEL/ELECTRONICS

1.	Fault Entry Switches (6)	Panel to all	SFLTIN_
2.	Scanout Source Select (8)	Panel to all	SSOC_
3.	Scanout Buss (66)	All to Electronics	TSBUS_ _

4.	Scanout Enable (1 - tw)	Electronics to all	TSOMS_ _
5.	Master Clear (1 - tw)	Electronics to all	-TCLR
6.	Panel Data, Address (54)	Panel/Electronics to CPE	SPADR_ _, SPDAT_ _
7.	Panel Function Select (4)	Panel to CPE, CCE	SPANF_ _
8.	Panel Function Initiate (2 - tw)	Electronics to CPE	-TPFINT – TPMURQ-
9.	Breakpoint, Single Command Activate, Step (3 - tw)	Electronics to CPE	TBPTACT, TSCMACT, TCMSTEP
10.	Breakpoint Reached (1 - tw)	CPE to Electronics	-TBKPT-

NOTE: All lines are single-ended unless marked tw (twisted pair).

CPE – MEMORY

1. Processor Buss (11) – The CPE sends a buss to all memories. The buss is utilized to send memory address, memory data and controls to the memories.
2. Memory Busses (11/CPE) – Each memory sends the CPE (and IOP) 4 busses each containing 11 lines. The busses carry memory data. The CPE votes on this data processor busses.
3. Memory Access Request (1) – The CPE sends a Memory Access Request to all memories, the memory which is selected by the page being transmitted on the Output Buss will respond.
4. Memory Request Acknowledge (1/MB) – Each memory responds to the CPE via this line acknowledging that the memory request has been accepted and that the requesting CPE should transfer the remaining address and data (if a write) across the busses.
5. Data Available (1/MB) – Each memory sends the CPE the Data Available signal when memory data is available on the Buss.

CPE – CCE

1. CPE Stream Assignment (12) – The CCE sends the CPE a group of 12 lines which specifies the stream assignment of the CPEs at the present time. The stream assignment specifies which CPEs are processing in the simplex, duplex, or TMR mode. The CPE uses the lines to activate the appropriate ports into the voter switch on its memory to processor busses.

2. Power Lines (1) – The CCE sends the CPE a Power On Signal to signify if a CPE is on or off line.
3. Clock, Sync (2) – The CCE sends the CPE the system clock at the highest frequency required in the system, plus a sync signal at 1/2 this frequency.
4. CPE Fault (1) – The CPE sends the CCE a signal indicating a fault condition has been detected.
5. Start CPE Interrupt (1) – The CCE sends the CPE an initialize signal which causes the CPE to fetch a new PSW and start processing. This insures synchronization of the CPEs during a Duplex or TMR mode of processing.
6. Stop CPE Interrupt (1) – The CCE sends the CPE a line which causes the CPE to complete the instruction in progress and store the old PSW in a specified memory location dedicated to this interrupt. Then the CPE sends the CCE the "CPE Available" signal and then waits in the "fetch" cycle for an interrupt from the CCE.
7. Initialize Memory Interrupt (1) – The CCE sends the CPE a signal which requests initialization of a new memory. The CPE fetches a new PSW from the designated memory location for this interrupt and then proceeds processing to initialize memory via this program.
8. Initialize Memory Control (2) – The CCE sends the CPE a 2-bit code to designate which memory to initialize. These lines are used in conjunction with the Initialize Memory Interrupt.
9. CPE Available (1) – The CPE sends the CCE a line signalling that the CPE is quiescent and is waiting in the initial state of the "Fetch" cycle, or that the Program Progress Alert instruction being executed is signalling a system roll-back point.
10. Panic Halt Interrupt (1) – The CCE sends all CPEs a Panic Halt Interrupt which causes the CPE to terminate the operations in progress and stop.
11. External Interrupt (1) – The CCE routes the external interrupt to the appropriate CPE for execution.
12. External Interrupt Acknowledge (1) – The CPE sends an external interrupt acknowledge line to the CCE to indicate acceptance of the external interrupt for processing.

CPE – IOP

1. IOP Memory Request Inhibit (1/IOP) – Each IOP sends the CPE a signal which represents that a memory cycle by the IOP is being requested and the CPE should inhibit any memory request until this line is low. The fact that an IOP shares the

input and output busses with a CPE necessitates this communication line between the sharing IOP and CPE. The CPE must look at all IOP request inhibit lines when in a duplex or TMR mode of operation.

2. CPE Memory Request Inhibit (1) – The CPE sends a signal to each IOP in the system indicating that this CPE has requested a memory cycle and the memory has granted the request. In other words, the CPE is using the input buss and possibly the output buss while the signal is true. The IOP has a higher memory request priority than the CPE, so this signal must not be activated (high) until the memory has granted the request.
3. Initiate I/O Instruction (1) – The CPE sends the IOP a signal telling the addressed IOP that an I/O instruction is to be processed. The CPE must store the channel number, device number and subchannel number into a designated memory location, from which the signalled IOP can interrogate to determine what action is to be taken. It is the responsibility of the program to know when the I/O instruction can store in the designated memory location.
4. I/O Instruction Accepted (1/IOP) – Each IOP sends the CPE a signal which denotes the previously sent I/O instruction is completed or has progressed to the point to assure initialization of the requested function. The CPE must wait for the I/O instruction complete signal before completing the I/O instruction and setting the condition codes.
5. I/O Interrupt (2/IOP) – The IOP sends CPE an interrupt line for each channel which designates one of many conditions that may exist in the IOP which must be signalled the system. The IOP stores the interrupt information in a memory location (64) to be interrogated by the interrupt program. The CPE processes the interrupt if the I/O interrupt mask is off.
6. I/O Condition Code (2/IOP) – Each IOP sends the CPE two lines which represent the IOP condition, following the execution of an I/O instruction, which is to be loaded into the CPE condition code indicators. These lines are interrogated by the CPE while the I/O instruction accepted line is high.
7. I/O Interrupt Acknowledge (2) – The CPE acknowledges the 2 IOP channel's interrupts via these lines.

CPE – MAINTENANCE PANEL/ELECTRONICS

1. Fault Entry Switches (6) – The fault entry switches allow the insertion of switchable faults into any 6 desired points in any modules for fault tolerance studies.

2. Scanout Source Select (8) – The scanout source select switches allow selection of up to 16 different scanout sources independently for each of the two 33 bit scanouts on the maintenance panel.
3. Scanout Buss (66) – The tri-state scanout buss multiplexes inputs from scanout sources in each module into the maintenance panel scanouts.
4. Scanout Enable (1) – The maintenance electronics decodes the scanout module select switch output to provide an enable to the scanout multiplexers in the module selected.
5. Master Clear (1) – A master clear is provided to all modules when the master clear pushbutton on the maintenance panel is activated.
6. Panel Data, Address (54) – The Panel Data and Address Lines allow loading key CPE registers from the maintenance panel under microprogram control.
7. Panel Function Select (4) – The output of the panel function select switch is decoded within each CPE to allow activation of one of up to 16 panel functions for loading memory and key CPE registers.
8. Panel Function Initiate (2) – Each CPE receives separate panel function initiate lines allowing individual execution of activated panel functions for diagnosis and/or fault insertion. Panel switching allows initiation of activated functions in from 1 to 4 CPES simultaneously. One line provides a synchronized output, the other a direct output from the pushbuttons antibounce circuit for MUR operations.
9. Breakpoint, Single Command Activate, Step (3) – Panel switches allow activation of breakpoint and single command modes of operation within the ARMS CPE. The Breakpoint/Single Command Step switch allows stepping the program beyond the breakpoint in the breakpoint mode or stepping it ahead one instruction in the single command mode.
10. Breakpoint Reached (1) – When a CPE stops at a breakpoint, this line signals this condition to the maintenance panel indicator.

5. MEMORY

The memory interface contains all necessary logic for interfacing between the commercial core memory and the ARMS data bus structure and CCE module. This logic consists of:

a) memory timing and control logic
b) bus access request decoder and memory response generator
c) input voter switch
d) address register
e) data register
f) memory input Hamming - parity code logic
g) memory output Hamming - parity code logic
h) fault control logic
i) output multiplexer

Memory Timing and Control Logic. The memory timing and control logic controls the sequence of events in other portions of memory interface logic and provides necessary initiation and control signals to the core memory. This logic sequences read and write modes in response to access requests from CPE or IOP modules accepted by the bus access request decoder, and to the state of the read/write signals transmitted on the processor buses simultaneously with the access requests by the CPE or IOP modules.

The states taken on by the memory timing and control logic are shown in Figure 5-2, while Figure 5-3 shows corresponding memory access control logic states within the CPE. Figure 5-4 shows write cycle timing at the interface while Figure 5-5 shows read cycle timing. Figure 5-6 illustrates the memory interface timing and control logic path and Figure 5-7 does the same for the CPE memory interface sequencer.

Bus Access Request Decoder and Memory Response Generator. The bus access request decoder and memory response generator, as shown in Figure 5-8, consists of a page address comparitor, request priority ordering logic and voter/switch and output multiplexer control logic and holding registers. The page address comparitor determines if any bus access requests are directed to this memory by comparing the page addresses transmitted on the processor buses simultaneously with the access requests by the CPE or IOP modules with the page address assignment currently being received from the CCE module. When the addresses agree, and the memory is not busy as determined from the state of the memory timing and control logic, the highest priority request is granted and the memory timing and control logic initiates a new memory cycle. The request priority ordering logic always gives a new access request from an IOP priority over a new access request from a CPE. However if a CPE access request is already being sent the IOP will not be granted access until the CPE's access has been completed. Logic to order priority between competing CPE access requests in a multiple stream version of ARMS is also provided. The voter/switch control logic causes the voter/switch, by means of a holding register, to input on the processor buses corresponding to the CPE's or IOP's whose access requests have been granted. The holding register's contents are updated each time a new access request is granted. The output multiplexer control logic causes the output multiplexer, by means of a holding register, to output on a single memory bus according to the following algorithm:

a) if only one access request is granted (simplex mode) the memory outputs on the memory bus paired with the processor bus whose access was granted.

b) if two or three access requests were granted (duplex or TMR modes) the memory outputs on the memory bus paired with either the lowest, middle, or highest numbered processor bus depending upon the memory output bus assignment currently being received from the CCE module.

b) continued

This requires multiple memories assigned the same page address but different output bus assignments by the CCE to respond to duplex or TMR processor requests on different buses.

Input Voter/Switch. The input voter/switch is capable of operating in three modes:

a) the voter/switch passes data from the selected bus in the simplex mode. No fault detection is possible from the voter/switch in this mode.

b) the voter/switch compares data bit-by-bit logically ORing selected data in the duplex mode. A fault signal is issued for both selected buses upon disagreement in any of the bit positions unless all of the bits from one of the selected inputs are identically "0".

c) the voter/switch votes on selected data bit-by-bit outputting the majority decision in the TMR mode. A fault signal is issued for the disagreeing bus(es) upon disagreement in any of the bit positions.

The voter/switch is 11 bits wide and is capable of selecting between data from 4 processor buses. Figure 5-9 illustrates one bit of the memory's voter switch. The voter switch used in the CPE and IOP which is not capable of fault detection is also shown for comparison.

Address Register. The address register contains 13 bits (12 data, 1 parity) and is capable of addressing 4096 words in the core memory module.

Data Register. The data register contains 33 bits (32 data, 1 parity) and buffers data transferred through the voter/switch during write cycles.

Memory Input Hamming - Parity Code Logic. The memory input Hamming - parity code logic, as shown in Figure 5-10, performs the following functions:

a) it performs a 13 bit parity check on the address register output detecting odd numbers of errors.

b) It performs a 33 bit parity check on the data register output detecting odd numbers of errors.

c) it generates a 6 bit Hamming code over the 32 data bits at the output of the data register.

d) it generates an overall parity bit over the 32 data bits at the output of the data register plus the 6 Hamming code bits generated in item c). The parity bit is a "1" when all other bits are "0". The 6 Hamming code bits plus the overall parity bit are stored with the 32 data bits to provide single error detection and multiple error correction within the memory module.

Memory Output Hamming - Parity Code Logic. The memory output Hamming parity code logic, also shown in Figure 5-10, performs the following functions:

a) it performs a 38 bit Hamming check on the data output from the core memory. This logic outputs a non-zero syndrome when one or two Hamming code errors are detected.

b) it performs a 39 bit parity check on the data output from the core memory detecting odd numbers of errors.

c) it provides Hamming single bit error correction by inverting one of the 32 data bits outputted from the core memory according to the syndrome output of the Hamming check of item a) if and only if the parity check of item b) also indicates an error.

d) it indicates a multiple fault if either the Hamming checker of item a) outputs a non-zero syndrome or the parity checker of item b) indicates a fault but not when both indicate faults. These conditions signify two and three detectable (but not correctable) bit faults respectively. An all "0" output indication from the core memory also indicates a multiple fault except during memory initialization after a master clear interrupt.

e) it generates a parity bit over the 32 bit output of the Hamming correction logic of item c).

Fault Control Logic. The fault control logic as shown in Figure 5-11 performs the following functions:

a) it inhibits a memory write cycle due to fault indications from the address and data register parity check logic or from the voter/switch logic if the latter faults are not maskable through majority voting.

b) it forces the output multiplexer output identically to "0" on a read cycle due to fault indications from the address and data register parity check logic, from the multiple fault detection circuits of the memory output Hamming-parity check logic, or from the voter/switch logic if the latter faults are not maskable through majority voting.

c) it provides a fault interrupt to the CCE module due to fault indications from the address and data parity check logic, from the multiple fault-detection circuits of the memory output Hamming-parity check logic, or from the voter/switch. Except in the case of the Hamming check fault indication, the IOP(s) or CPE(s) accessing the memory at the time the error is detected are indicated to the CCE.

Output Multiplexer. The output multiplexer multiplexes the 33 bit output of the Hamming single bit error correction logic and its associated parity bit onto the memory bus selected by the output multiplexer holding register of the memory response generator, eleven bits at a time, unless inhibited from doing so by the fault control logic.

Data Bus Formats Between Memory, CPE, and IOP. Data transfers between memory, CPE, and IOP modules occur on 4 processor buses and 4 memory buses. Each bus contains 11 data lines. Thus address transfer requires 2 clock times, and data transfer 3. Address and data transfer formats are shown in Figure 5-12. Refer to Figure 5-4 and 5-5 for timing of these transfers.

Component Count

Table 5-1 shows the number of gates and flip-flops required by each part of the memory interface. The final design and complexity level of the memory interface is extremely close to that assumed for the reliability analysis contained in Reference 2.

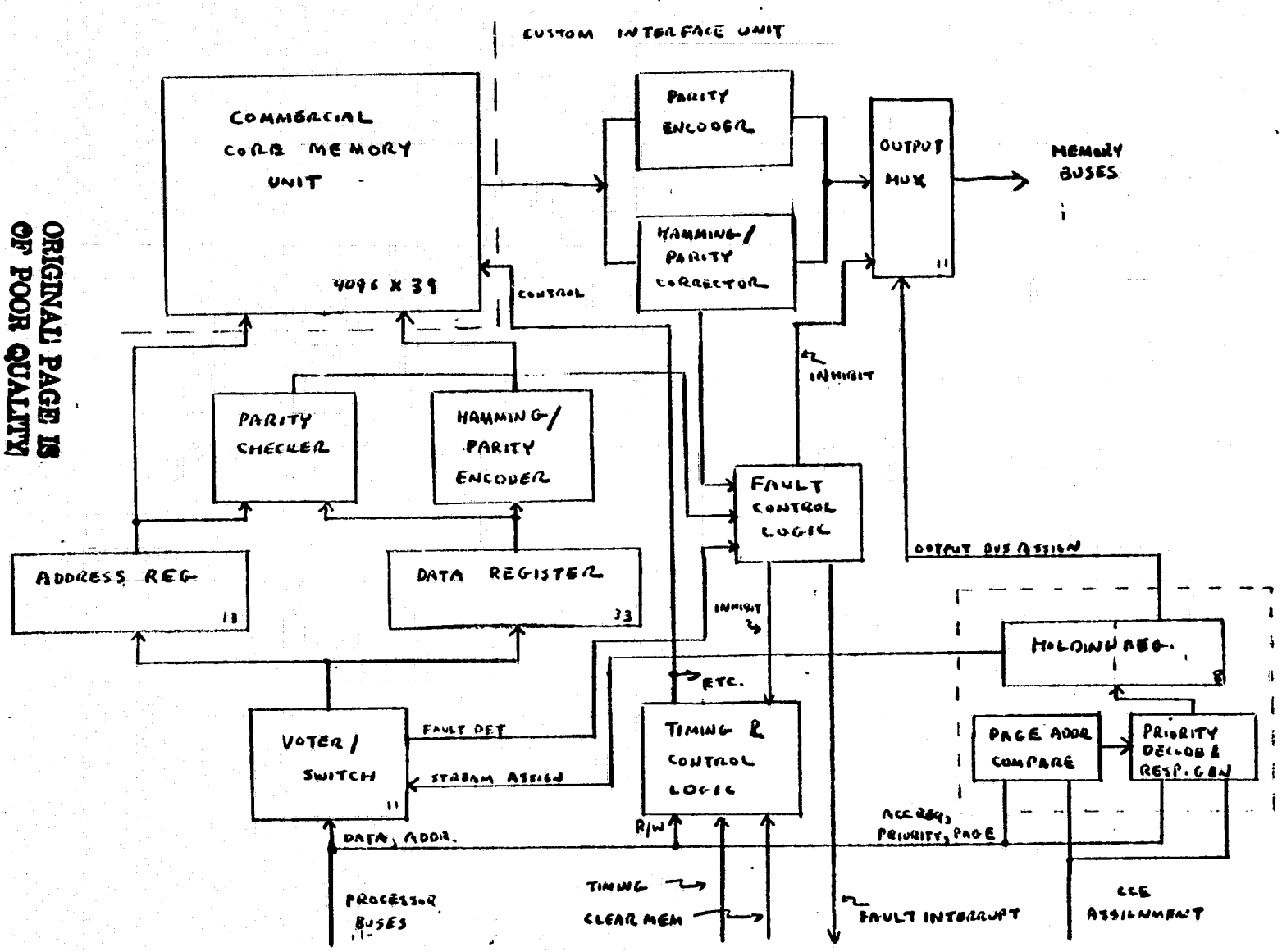
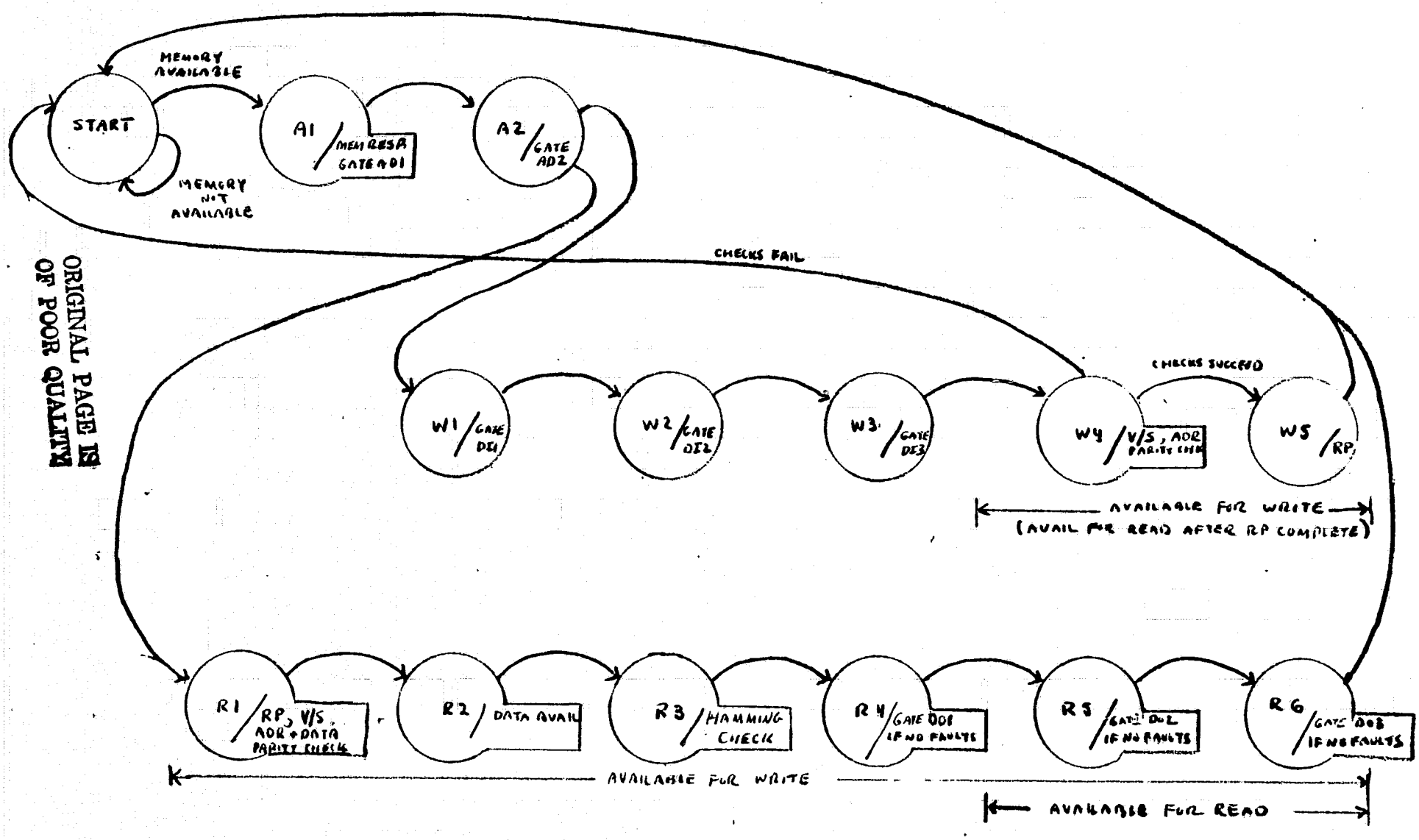


Figure 5-1
ARMS MEMORY MODULE

Figure 5-2

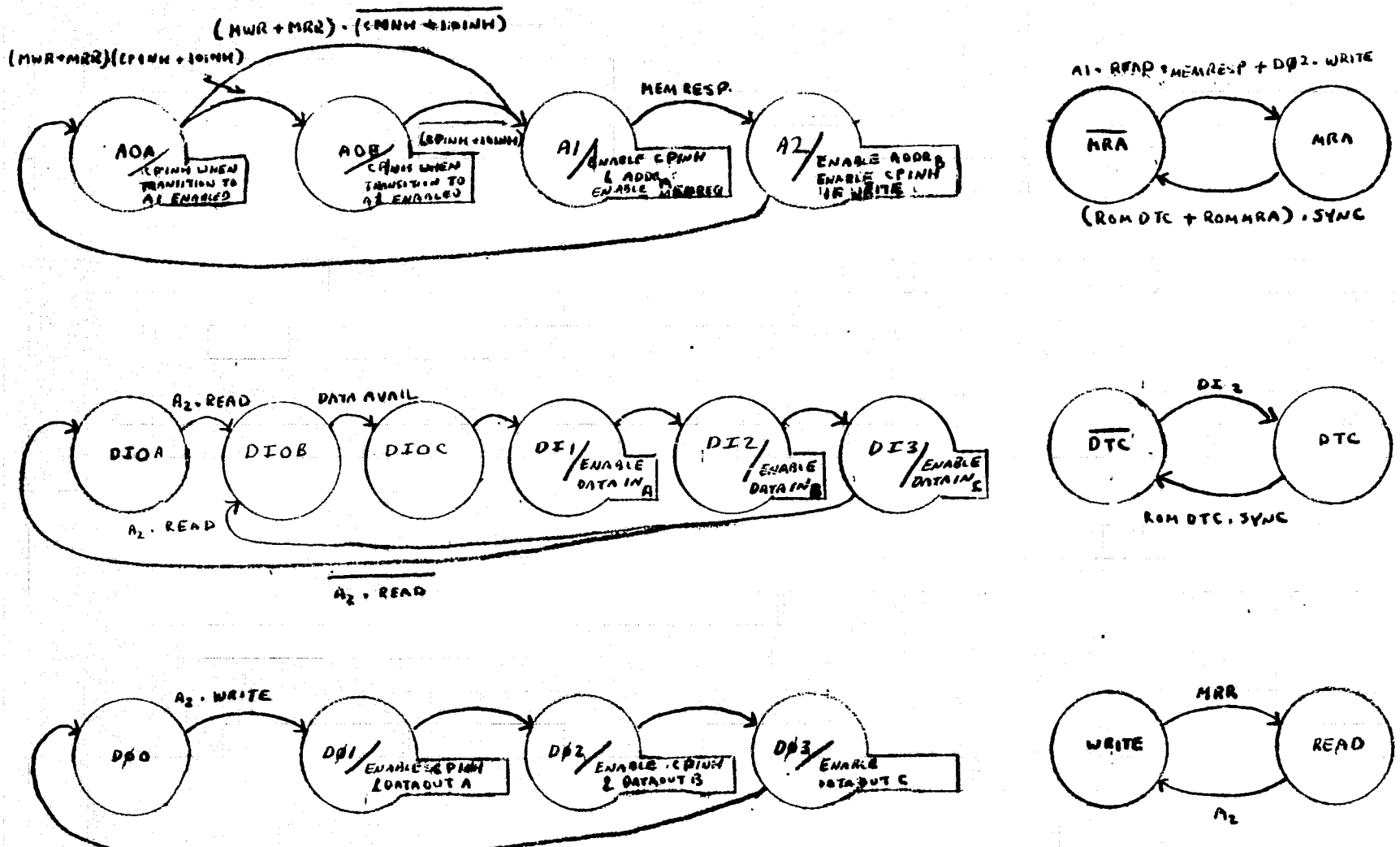
ARMS MEMORY INTERFACE STATE TRANSITION DIAGRAM



ORIGINAL PAGE IS OF POOR QUALITY

Figure 5-3

MEMORY ACCESS CONTROL LOGIC (CPE)



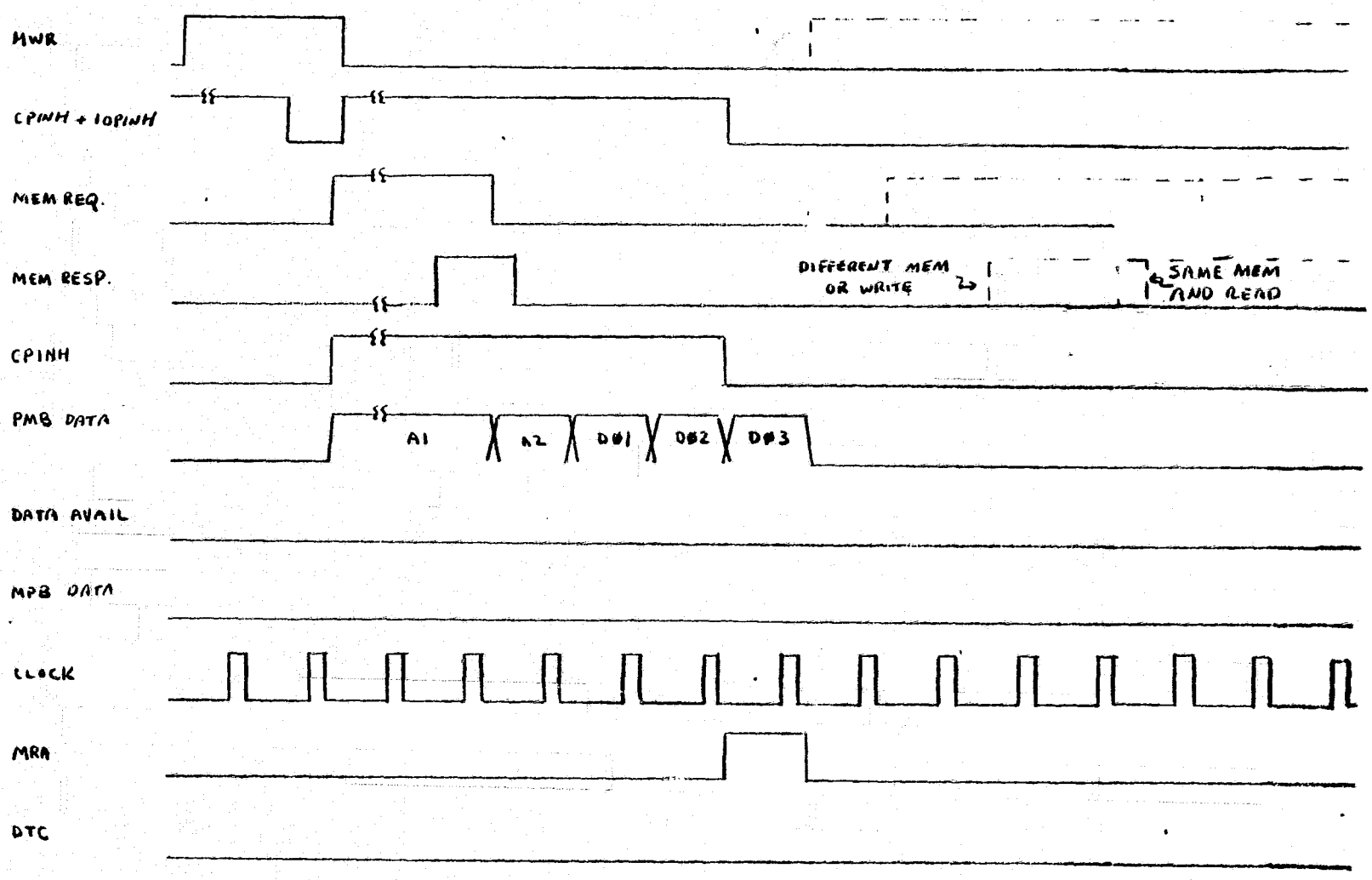


Figure 5-4

ARMS MEMORY INTERFACE TIMING – WRITE CYCLE (CPE)

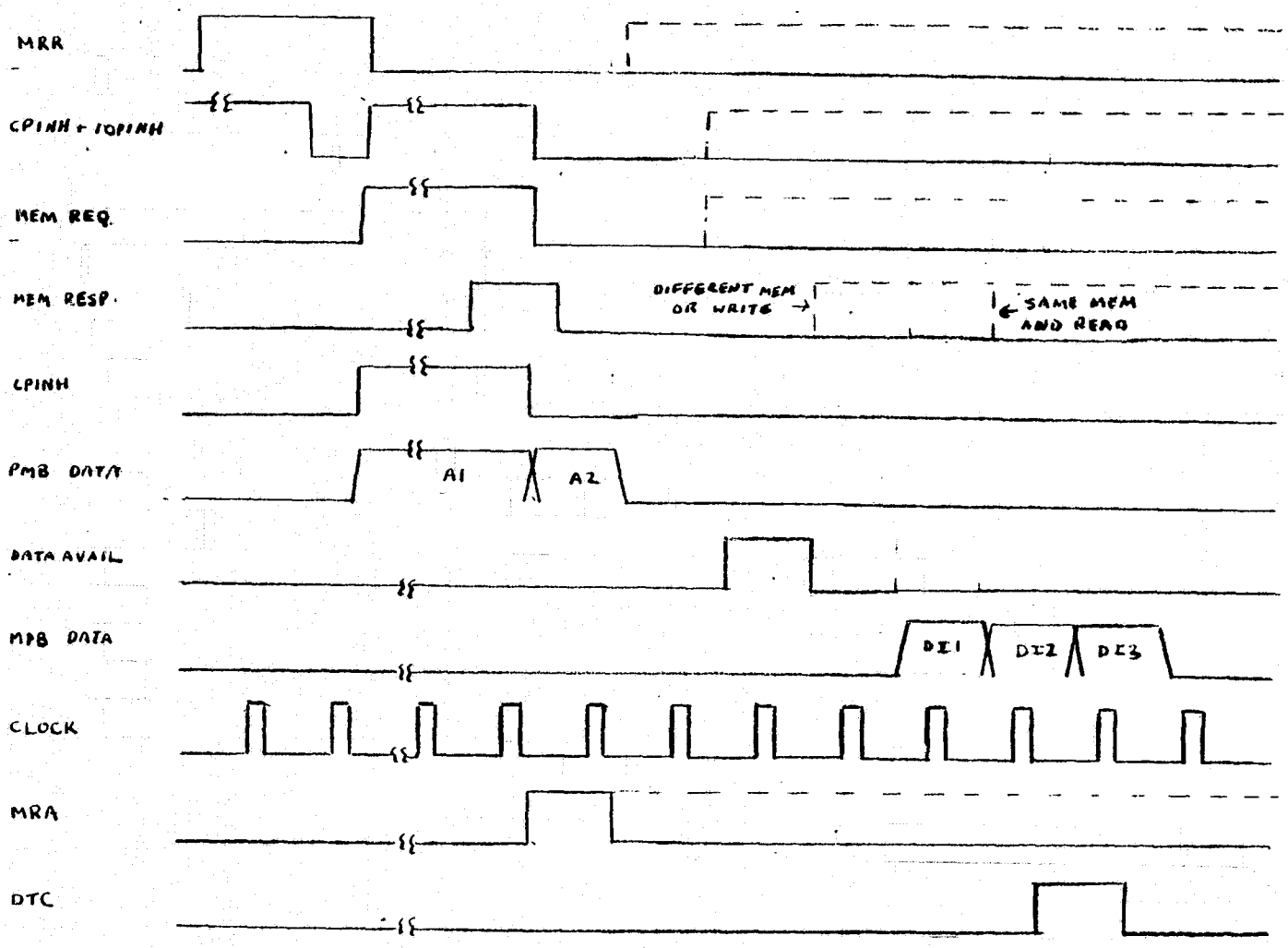


Figure 5-5
ARMSMEMORY INTERFACE TIMING — READ CYCLE (CPE)

Figure 5-6

MEMORY INTERFACE TIMING AND CONTROL LOGIC

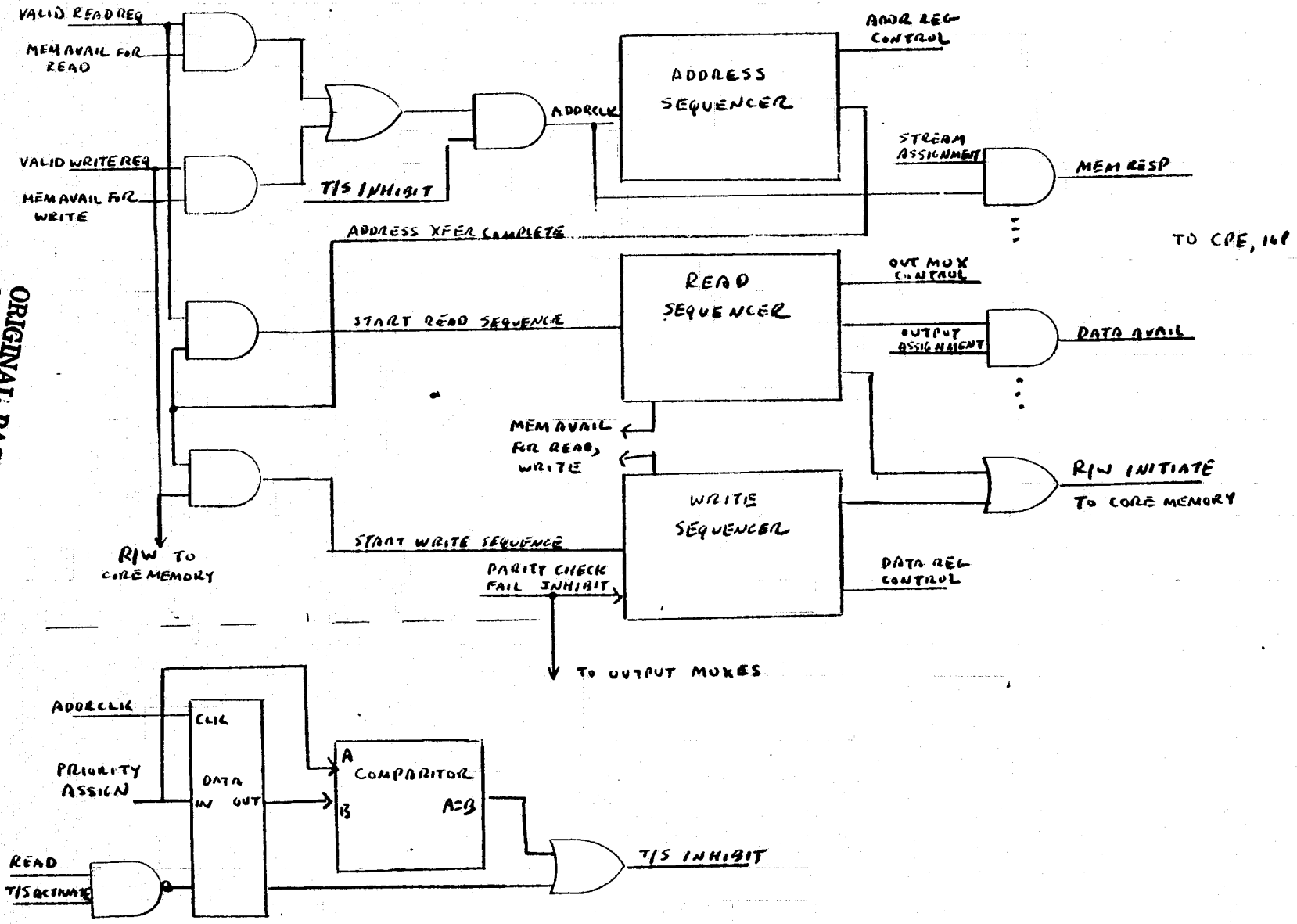
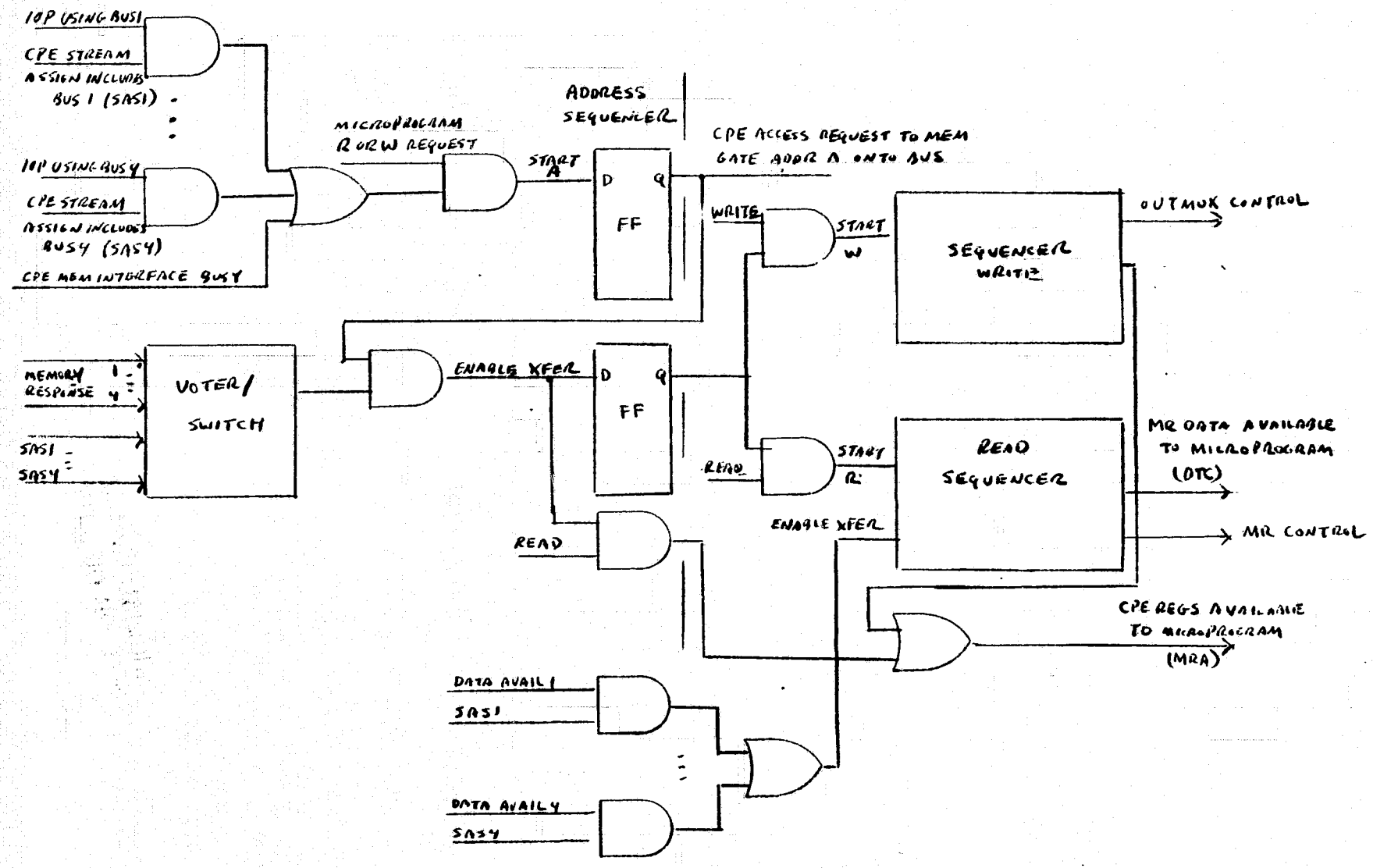


Figure 5-7

CPE MEMORY INTERFACE SEQUENCER



• IOP SEQUENCER SIMILAR

Figure 5

MEMORY INTERFACE BUS ACCESS REQUEST DECODE, MEMORY RESPONSE GENERATION

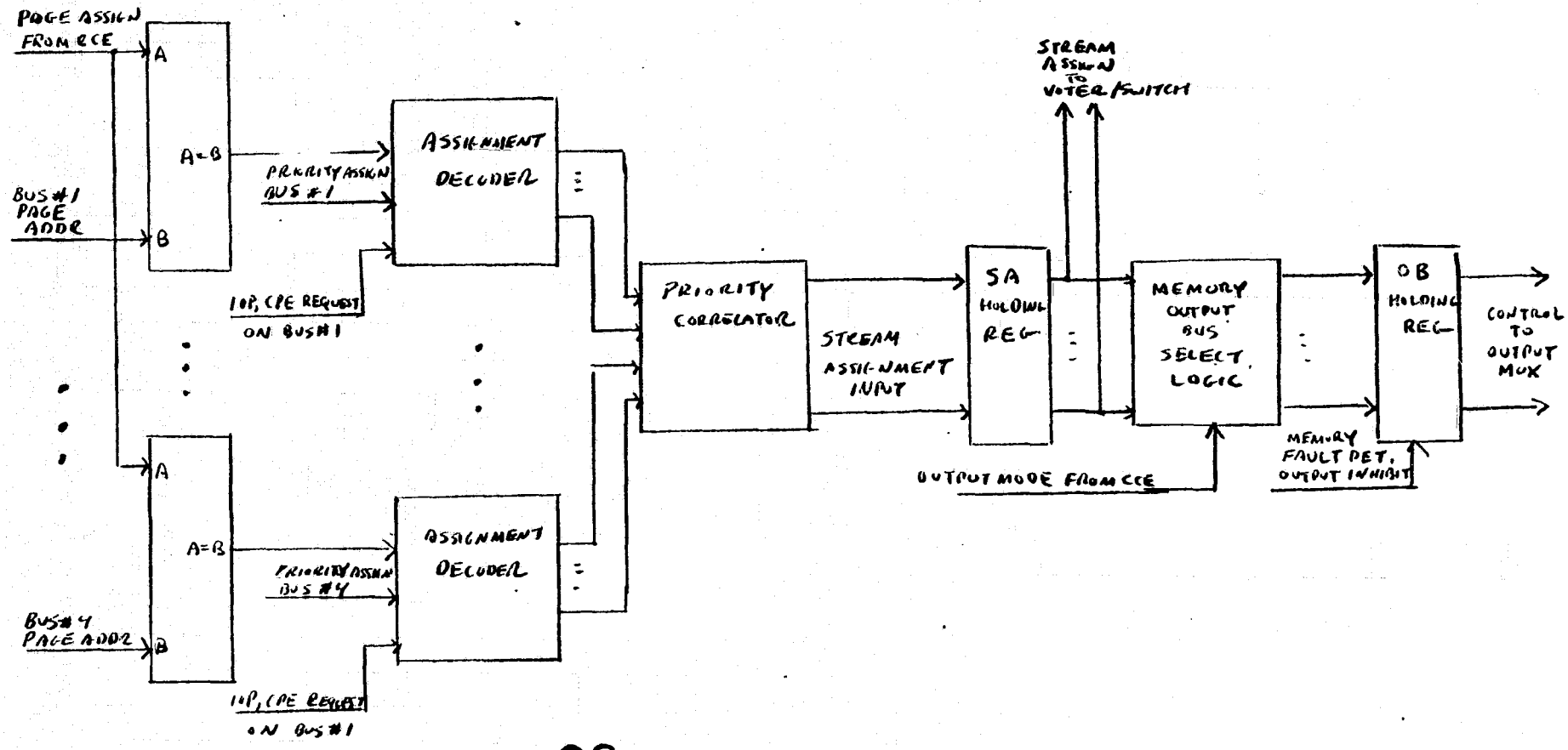
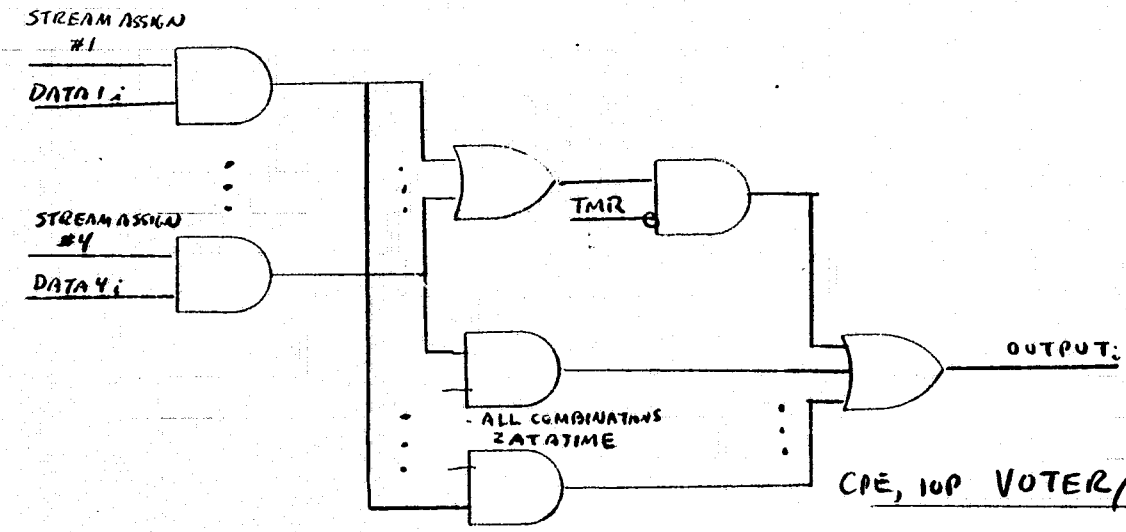
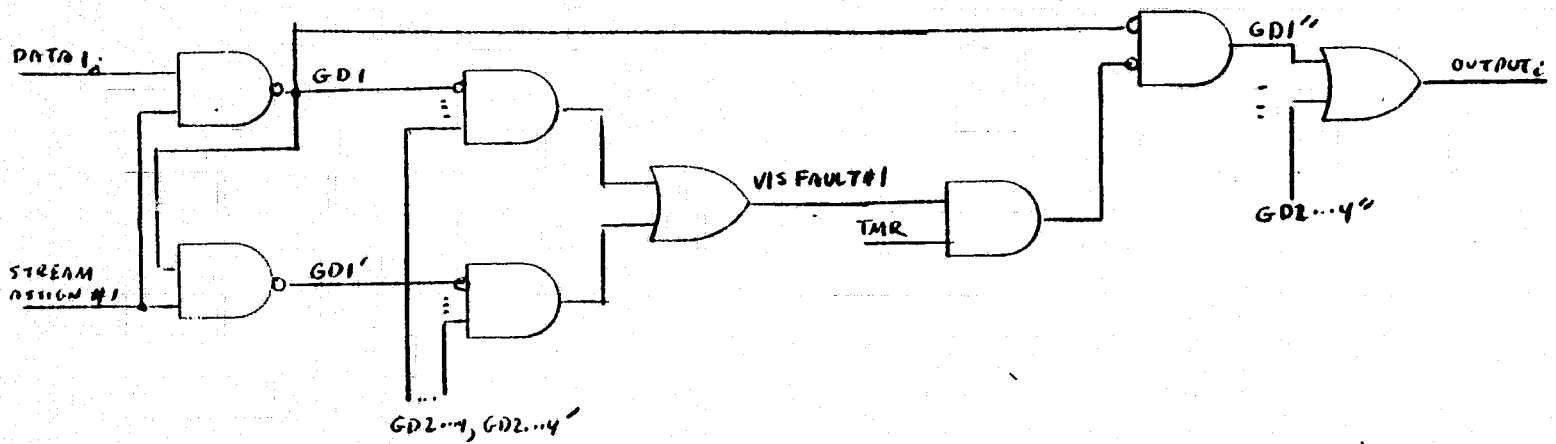


Figure 5-9

MEMORY INTERFACE VOTER/SWITCHES



CPE, IOP VOTER/SWITCH W/o FAULT DETECTION (5 DIPS/BIT)



MEMORY VOTER/SWITCH WITH FAULT DETECTION (10 DIPS/BIT)

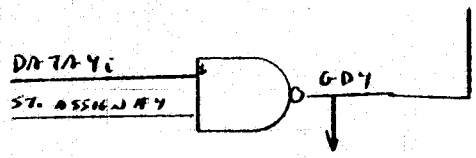


Figure 5

MEMORY INTERFACE HAMMING-PARITY CODE LOGIC

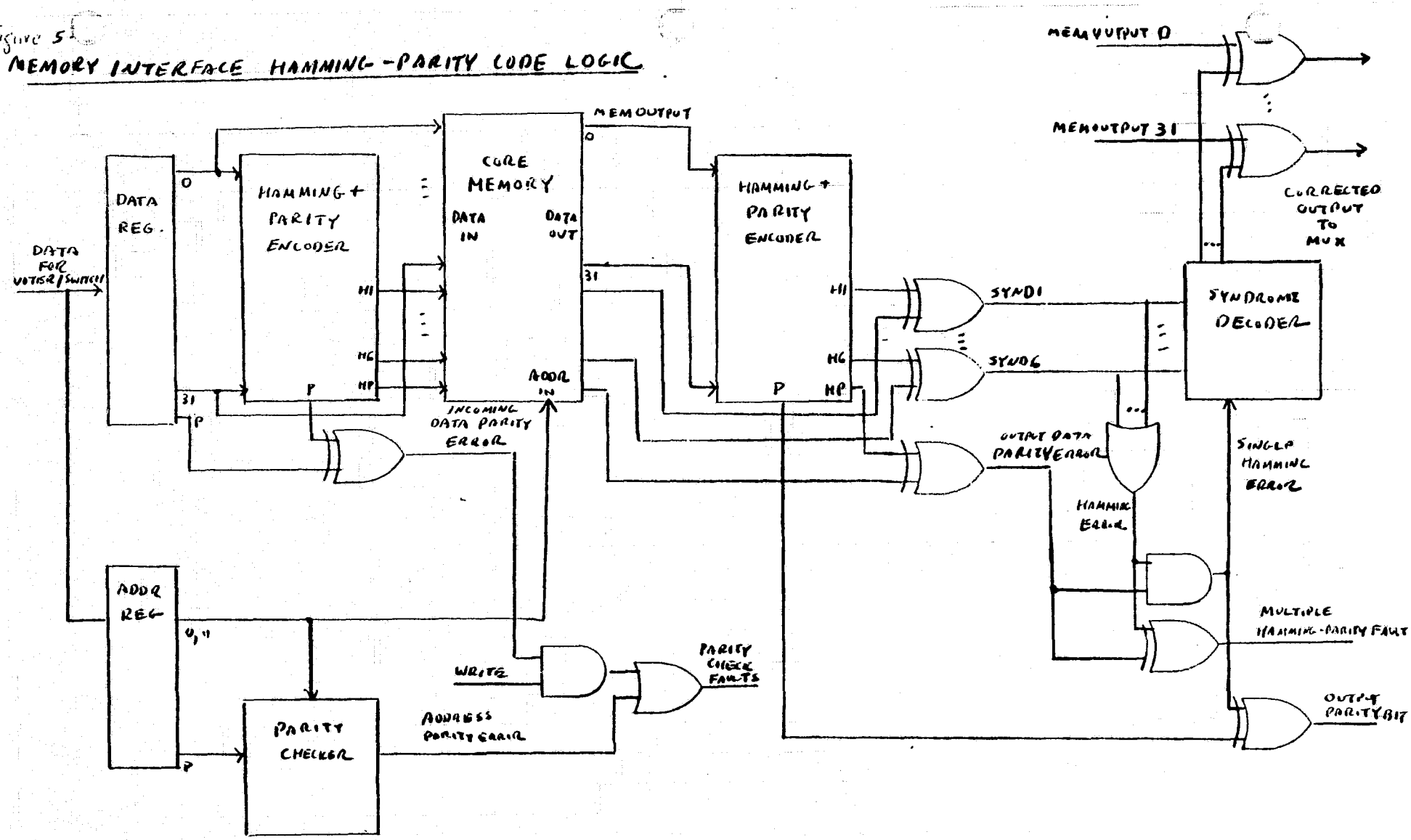
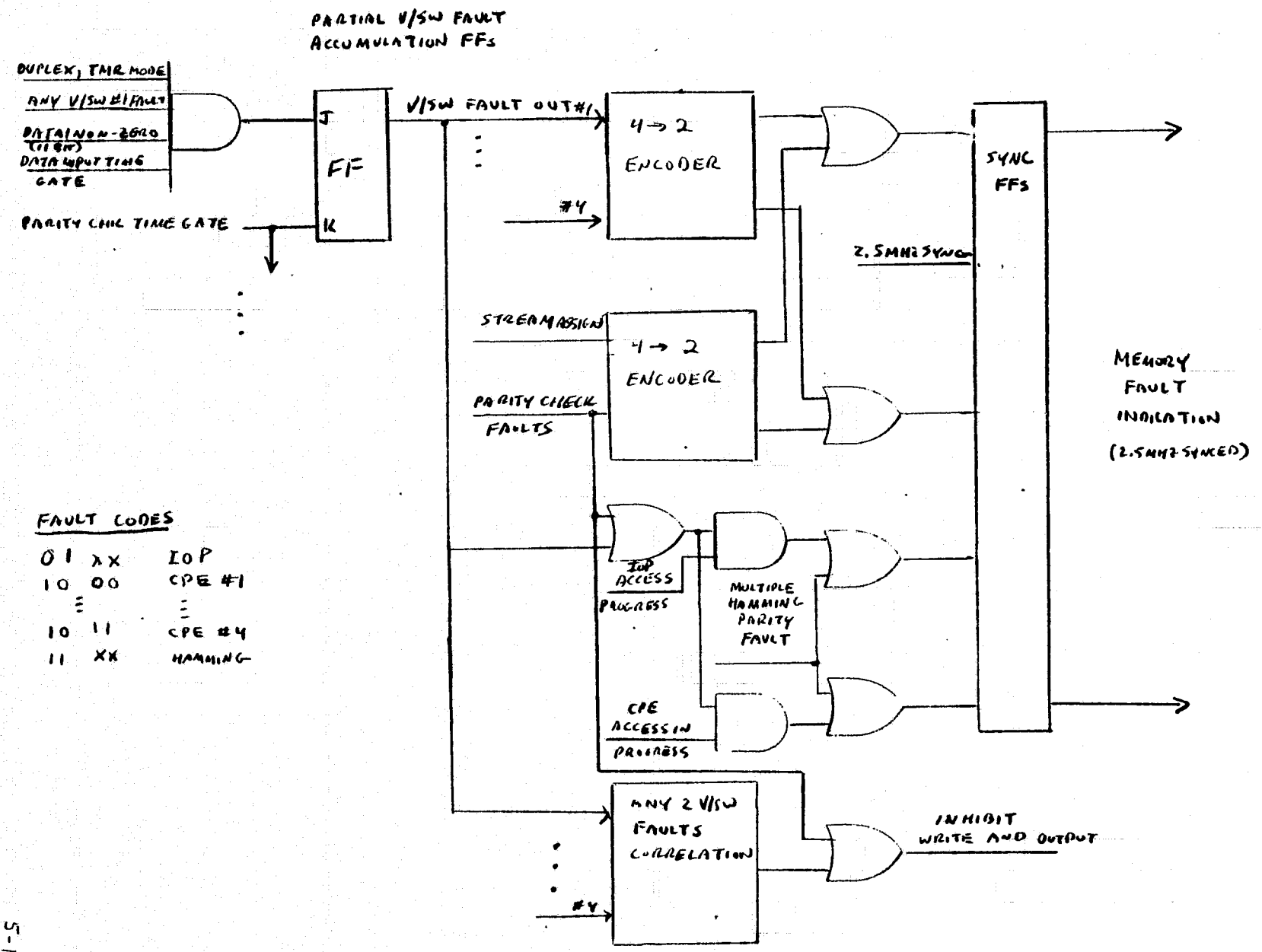


Figure 5-11

MEMORY INTERFACE FAULT CONTROL LOGIC



FAULT CODES

01	XX	IOP
10	00	CPE #1
=		=
10	11	CPE #4
11	XX	HAMMING

Figure 5-12

CPE/IOP ↔ MEMORY INTERFACE WORD FORMATS

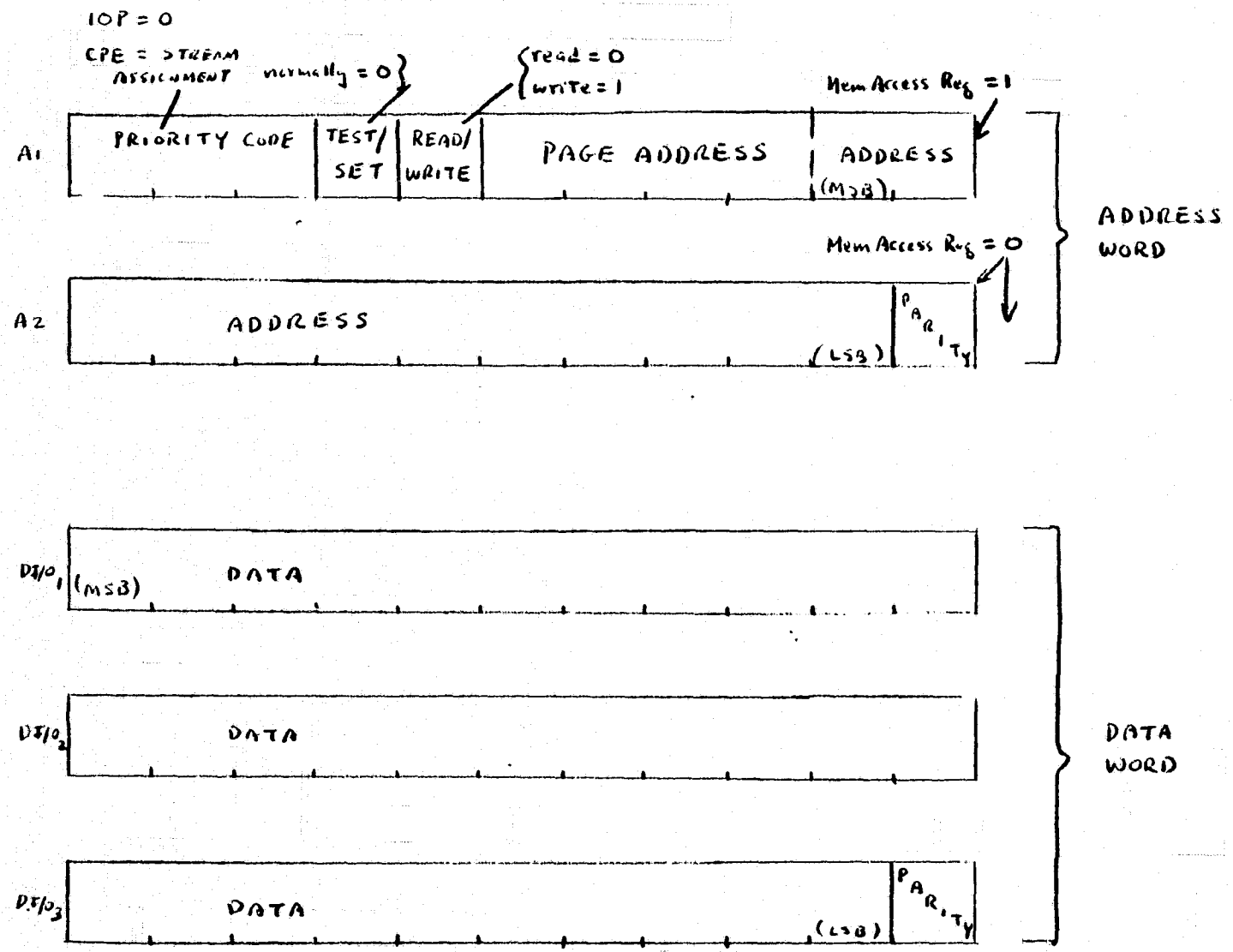


TABLE 5-1. MEMORY COMPONENT COUNT (Excluding Commercial Core Plane)

Function	Gates	Flip-Flops	Fail Rate Equiv Gates
1. Voter Switch/Output MUX	504	0	504
2. Addr and Data Reg	52	33	250
3. Hamming-Parity Logic	994	0	994
4. Access, Timing and Control	414	31	600
5. Fault Control	51	8	99
	2015	72	2447

MEMORY INTERFACE

MEMORY - CPEs - IOPs

1.	Processor Busses (11) x4=(44)	CPE to Memory	-TPMB_ _ _
2.	Memory Busses (11) x4=(44)	Memory to CPE and IOP	-TMPB_ _ _
3.	Memory Access Request (8)	CPE, IOP to Memory	-TCPREQ_
4.	Memory Request Acknowledge (4)	Memory to CPE or IOP	-TMRESP_
5.	Data Available (4)	Memory to CPE or IOP	-TDAVAL_

MEMORY - CCE

1.	Page Assignment (2 bits)	CCE to Memory	-TPAGE_ _
2.	Output Buss Assignment (2)	CCE to Memory	-TØTMD_ _
3.	Clear Memory Interrupt (1)	CCE to Memory	TCLRMM_
4.	Clock, Sync (2)	CCE to Memory	TCLK, TSYNC
5.	Power Protect Inhibit (1)	CCE to Memory	TPWRHLT
6.	Memory Fault Interrupt (4)	Memory to CCE	-TMFLT_ _

0	No fault	8	CPE1 failure
4	IOP1 failure	9	CPE2 failure
5	IOP2 failure*	10	CPE3 failure
6	IOP3 failure*	11	CPE4 failure
7	IOP4 failure*	15	Memory Internal

* not used in the breadboard configuration

MEMORY - MAINTENANCE PANEL/ELECTRONICS

1.	Fault Entry Switches (6)	Panel to Memory	SFLTIN_
2.	Scanout Source Select (2)	Panel to Memory	SSOC_ _ _
3.	Scanout Buss (66)	Memory to Electronics	TSBUS_ _
4.	Scanout Enable (1/Memory)	Electronics to Memory	TSØMS_ _
5.	Master Clear (1/Memory)	Electronics to Memory	-TCLR

MEMORY - INTERFACE - CORE MEMORY

1.	Address Input (12)	Memory Interface to Core Memory	-TMADR_ _
2.	Data Input (39)	Memory Interface to Core Memory	-TMDIN_ _
3.	Data Output (39)	Core Memory to Memory Interface	-TMDTA_ _
4.	Read/Write Level (1)	Memory Interface to Core Memory	-TRWL
5.	Initiate Cycle (1)	Memory Interface to Core Memory	-TRP

MEMORY INTERFACE

MEMORY - CPE - IOP

1. Processor Busses (11/CPE) - Each CPE (IOP combination) sends the memory an eleven (11) line buss. Control lines, address and data is multiplexed across the buss to accomplish memory read and write operations.

2. Memory Busses (11/CPE) - Each memory sends the CPEs and IOPs 4 busses each containing 11 lines. The busses carry memory data. The CPEs and IOPs vote on the output busses in the Duplex and TMR modes.

3. Memory Access Request (1/CPE and IOP) - Each CPE or IOP signals the memory for access by the memory access request line. The memory access request line will remain high until the memory signals acceptance of the request, (Memory Request Acknowledge).

4. Memory Request Acknowledge (1/M B) - The memory sends the requesting CPE or IOP a signal acknowledging that the memory request has been accepted and the requesting unit should transfer the remaining address and data (if a write) across the buss.

5. Data Available (1/M B) - The memory send the requesting (read) CPE or IOP a signal stipulating that the memory data is now selected on the appropriate output buss.

MEMORY - CCE

1. Page Assignment (2/Memory) - The CCE sends each memory a two (2)-bit code specifying the page assignment for that memory.

2. Output Buss Assignment (2) - The CCE sends each memory an encoded 2-bit line specifying which memory bus to use to output data on during Duplex or TMR modes.

3. Clear Memory Interrupt (1) - The CCE sends each memory an interrupt line which causes the memory to output all "0" on the memory bus during the execution of a software routine that cycles through and clears all locations. This is required when initializing essential memories.

MEMORY - CCE (Cont.)

4. Clock Sync (2) - The CCE sends the memory the system clock at the highest frequency required in the system and a sync signal at $\frac{1}{2}$ this frequency.

5. Power Protect Inhibit (1) - The CCE sends the memory a Power Protect Inhibit Signal. This line specifies that the power is going out of tolerance or that system reconfiguration is about to occur and the memory should be shut down.

6. Memory Fault Interrupt (4/Memory) - Each memory sends the CCE a 4 bit encoded fault line specifying what type of fault has occurred.

0	No fault	8	CPE 1 failure
4	IOP 1 failure	9	CPE 2 failure
5	IOP 2 failure*	10	CPE 3 failure
6	IOP 3 failure*	11	CPE 4 failure
7	IOP 4 failure*	15	Memory internal failure

* not used in the breadboard configuration

MEMORY - MAINTENANCE PANEL/ELECTRONICS

1. Fault Entry Switches (6) - The fault entry switches allow the insertion of switchable faults into any 6 desired points in any modules for fault tolerance studies.
2. Scanout Source Select (2) - The scanout source select switches allow selection of 2 different memory scanout sources independently for each of the two 33-bit scanouts on the maintenance panel.
3. Scanout Buss (66) - The tri-state scanout bus multiplexes inputs from scanout sources in each module into the maintenance panel scanouts.
4. Scanout Enable (1/Memory) - The maintenance electronic decodes the scanout module select switch output to provide an enable to the scanout multiplexers in the memory selected.
5. Master Clear (1/Memory) - A master clear is provided to all modules when the master clear pushbutton on the maintenance panel is activated.

MEMORY INTERFACE - CORE MEMORY

1. Address Input (12) - The Address Input lines control the location in the core memory to be accessed for read or write.
2. Data Input (39) - The Data Input lines transmit data to the core memory during write cycles.
3. Data Output (39) - The Data Output lines transmit data from the core memory during read cycles.
4. Read/Write Level (1) - This signal determines if a memory cycle will be a read cycle or a write cycle.
5. Initiate Cycle (1) - This pulse causes the core memory module to initiate a read or write cycle according to the state of the read/write level. Once the cycle is initiated the core memory module provides its own timing.

L-2

6. INPUT/OUTPUT PROCESSOR (IOP)

The ARMS IOP consists of 4 sections: 1) TTY channel logic, 2) TTY controller logic, 3) DMS channel logic, and 4) common IOP containing logic for servicing both channels and for interfacing with the rest of the ARMS computer. The first two sections are unique to the breadboard. A flight version of ARMS would have one or more DMS channels. Figure 6-1 is a block diagram of the IOP module. The ARMS IOP is functionally similar to 2 IBM system 360 selector channels with each operating in a burst mode during data transfer. Ref 1 describes the 360 I/O specification and operation.

The DMS channel is 16 bits wide. The TTY channel is 8 bits wide with the controller converting this to a serial stream at the TTY interface. The IOP can operate concurrently with the CPE once a channel has been started by the CPE. The CPE software initiates IOP action by use of the Initiate I/O Instruction line during Start I/O, Test I/O, Halt I/O, or Test Channel instructions. Information pertaining to these instructions is stored in the Channel Instruction Word (byte address 152_{10}) in main memory when the CPE executes the instruction and is fetched by the IOP when the Initiate I/O Instruction line is raised. The CIW format is shown in Figure 6-2. In the case of the DMS the CIU address replaces the device address in the figure.

Each IOP channel requires additional locations in memory to specify data address locations, word counts, and control and status flags. The contents of these words are used only by the channel and are not transferred to or from the I/O devices except for certain command and status flags. These formats are shown in Figures 6-3 and 6-4. Their memory byte address assignments are listed in Table 4-3 in the CPE module section of this report. The 32-bit Channel Address Word (CAW) points to the byte location in memory of the first Channel Command Word (CCW). Each CCW consists of 64 bits and specifies which command the IOP will execute (Read, Read Backward, Write, Control, Sense, and Transfer in Channel), the starting address in main memory from which data will be read or written, the number of bytes or bus words to be transferred under this CCWs control, and flags modifying the command (chain data, chain command, skip, program controlled interruption). The CCW for the DMS Channel also specifies DIU and channel addresses within the DMS. Once the IOP program is started the IOP sends its condition code to the CPE and signals it to proceed by means of its I/O Instruction Accepted line.

Upon completion of an I/O command or in response to an error or to a device interrupt the IOP requests permission to store a channel status word (CSW) in memory by raising its I/O interrupt line for the appropriate channel to the CPE. If the CPE program is finished with the data in the previous CSW and its PSW system mask bit is set equal to one it raises its I/O Interrupt Acknowledge line allowing the IOP to store the new CSW in memory byte addresses 64-71. The CSW format is shown in Figure 6-5. The command address field contains the last CCW address +8 bytes, the Device (or CIU/DIU) and channel status fields contain status information as of the time of the interrupt and the byte (or bus word) count field contains the residual byte or word count (if any) from I/O operations.

The channel/device interface operates as follows: The channel and device (TTY or DMS CIU) each may be in an available, busy, interrupt pending, or non-operational state when an I/O request is received from the CPE due to a Start I/O instruction. If the channel is available (or if an I/O interrupt is pending) it fetches a CAW and outputs the specified device address. If the device is in an available state and no faults are noted the device responds by returning its internal device address to the channel which then sends it the command code. The device then responds with status as to whether or not it will execute the command. If the status is affirmative the device remains connected, the channel fetches the first CCW from memory and executes it in connection with the selected device returning an appropriate condition code as it releases the CPE. If the status is negative, faults are noted, or either channel or device is unavailable different condition codes are returned to the CPE telling the CPE to either automatically request CSW storage or to obtain this by means of a "TEST I/O" instruction. The latter condition occurs if the channel was already busy at the time of the request.

The Halt I/O instruction terminates on-going I/O operation, placing the channel in an interrupt pending state and allowing the CPE to override on-going IOP action. The "TEST I/O" instruction allows the CPE program to obtain status and to clear pending interrupt conditions selectively by I/O device. The "Test in Channel" instruction allows CPE testing of channel status without disturbing on-going device operations. All instructions cause an appropriate condition code return to the CPE as summarized in Table 6-1.

Channel status bits returned in the CSW include 1) Program-Controlled Interruption (allowing the channel to interrupt the CPE upon fetching this CCW), 2) Incorrect Data Length transferred, 3) Program Check (invalid address, command, or count fields detected by IOP), 4) Channel Data Parity Error, 5) CCW parity error, 6) Invalid interface signals present, and 7) Input Data Overrun. Device status bits returned in the CSW are 1) attention, 2) status modifier, 3) Device Busy, 4) Channel End (transfer complete), 5) Device End, 6) Unit Check (identified by available sense data from the device) and 7) Unit Exception (unique to a given device). Status bit positions and meanings are compatible with the IBM system 360 CSW. The operation of the four sections comprising the IOP are described below.

DATA TERMINAL CONTROL UNIT

The Data Terminal Control Unit (DTCU) provides for serial to parallel conversion of the TTY output, parallel to serial conversion at the TTY input, and for buffering and parity checking at the channel/DTCU interface. It also provides for decoding and encoding of ASCII data and tape unit control characters by means of PROMS and for an optional direct binary mode bypassing the PROMS. This logic also converts the specific command and status signals within the TTY unit to a form compatible with the standard IBM 360 selector channel interface provided by the TTY channel section of the IOP.

IOP CHANNEL LOGIC

The logic found in the two channel sections of the IOP are essentially identical except for the differences due to the buss width of the 2 channels and the additional line drivers and line receivers and at the DMS interface. As summarized in Figure 6-7 the channel logic consists of 1) discrete control logic for read memory forward and backward, and write memory, 2) Read/write multiplexer and register logic for buffering and routing data both ways through the channel, 3) Priority routing logic for the various memory access requests that the channel can make, 4) Holding Registers for holding and operating on the channel Instruction, Command, and Status words including decoding the CCW op-code to obtain the proper IOP instruction, synchronized counters for data words address, and byte (or bus word) counts, a Channel Command Word Address Counter, Command Word and Data Word address parity generation, buffering and decoding of control flags (and of CIU and DIU address in the case of the

DMS channel) and buffering of device status bits for inclusion in the log-out of the CSW, 5) Device interface logic including logic for comparing device response to addresses sent and device interface parity checks, and, 6) I/O Interrupt Request control logic, 7) channel status and control logic to sequence the registers mentioned and to generate condition codes information and appropriate channel status bits for inclusion in the CSW. Multiplexers to log out CSW bits and decoder for CPE instructions to the IOP are found in the common IOP section.

COMMON IOP LOGIC

The remaining IOP logic is that common to both channels. Figure 6-8 summarizes this logic which involves sequence control including the 1) decoding of I/O instructions from the CPE, fetch control for the CIW, CAW, and CCW, encoding of IOP condition codes, and generation of and response to interface control signals with the CPE and CCE; 2) the voter/switch interface to the 4 memory buses including buffering and parity checking of the data; 3) memory access control logic including priority control between the 2 channels, and discrete sequencing logic driving PROM controlled selection of appropriate IOP/Memory output multiplexer sources; 4) IOP/Memory output multiplexer selecting between 36 possible output sources within the 2 channels followed by a buffer register, parity generation over this register and parity checking over the Output Multiplexer Control PROM, and CCE controlled interfaces to each of the 4 processor to memory buses; 5) scanouts of key register and control points throughout the IOP to the maintenance/status panel.

COMPONENT COUNT & RELIABILITY DISCUSSION

Component counts for the IOP logic (with the exception of scanout which would probably not be included in a reliability calculation) are given in Table 6-2. The overall complexity of the breadboarded IOP is approximately 14,000 equivalent gates. An IOP with only one DMS channel would require something under 8,000 equivalent gates since the common IOP logic would also be reduced somewhat if only one channel was implemented. An IOP with two DMS channels would have a complexity of approximately 12,500 equivalent gates. For comparison the baseline IOP discussed in our architecture study had a complexity of approximately 10,000 equivalent gates for one channel and a probability of successful operation over 5 years of 0.9997 in a TMR plus one spare configuration.

Since it is intended that the IOP be operated in a TMR configuration in a flight version of ARMS with voting on all outputs, fault tolerance add-ons in the breadboard version of the IOP are limited to parity checks on key registers and interfaces and to IBM compatible status checks as discussed earlier in this section. Although this combination of parity and likelihood coverage could be extended somewhat further it was felt that the only way to insure a high probability of fault correction in real time was through use of TMR IOPs.

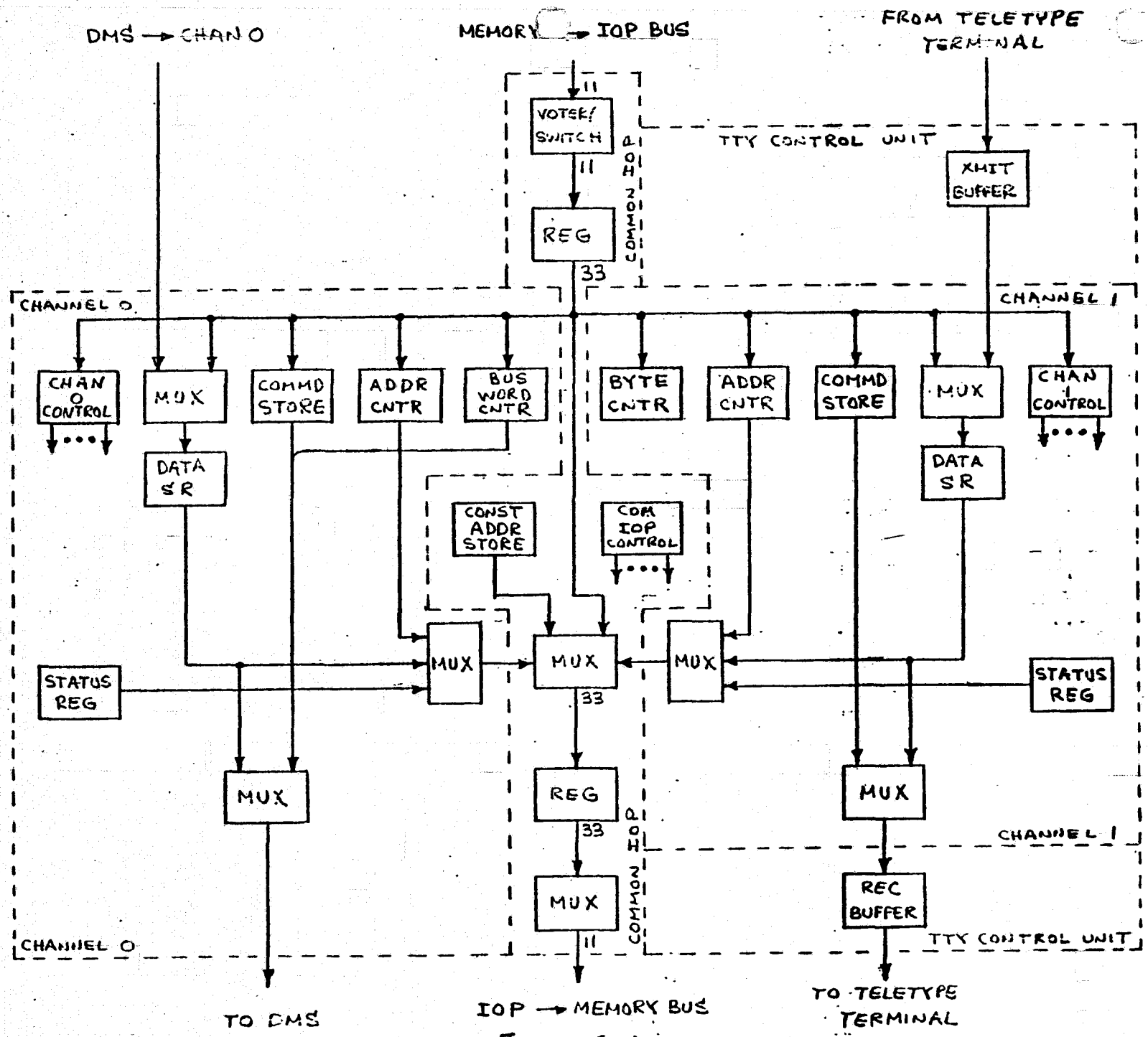


Figure 6-1

ARMS IOP BLOCK DIAGRAM

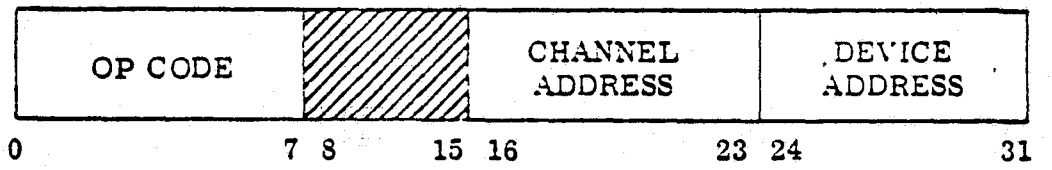


Figure 6-2 CIW Format (Channel 1)

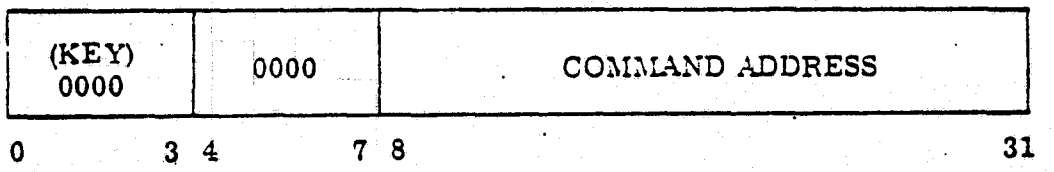
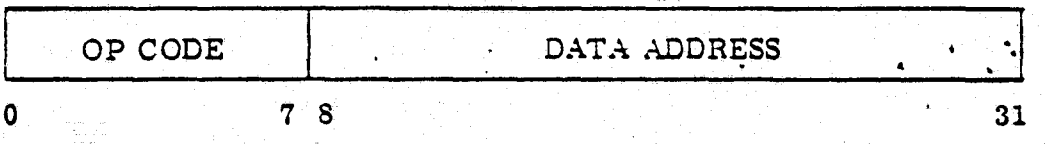


Figure 6-3. Channel Address Word (CAW)



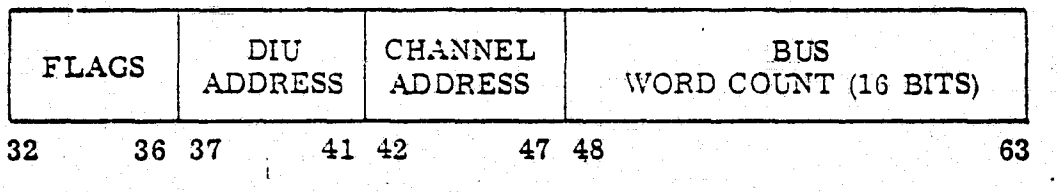
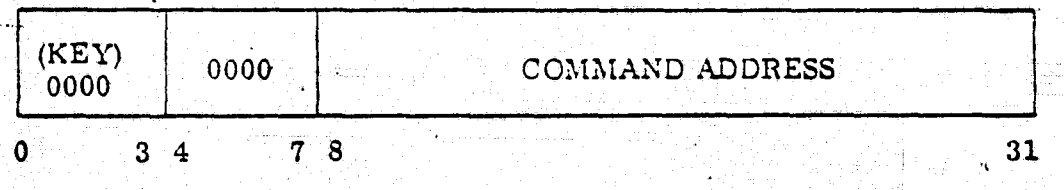


Figure 6-4 CCW Format (Channel 0)



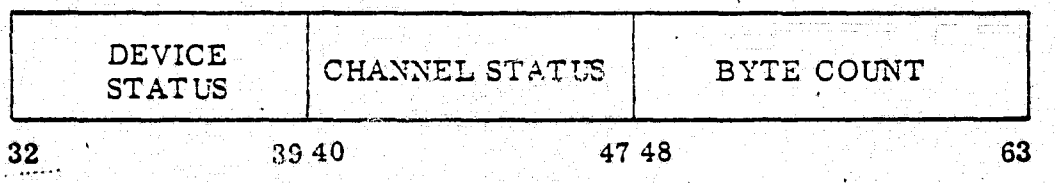


Figure 6-5 CSW Format

Figure 6-6
DATA TERMINAL CONTROL UNIT

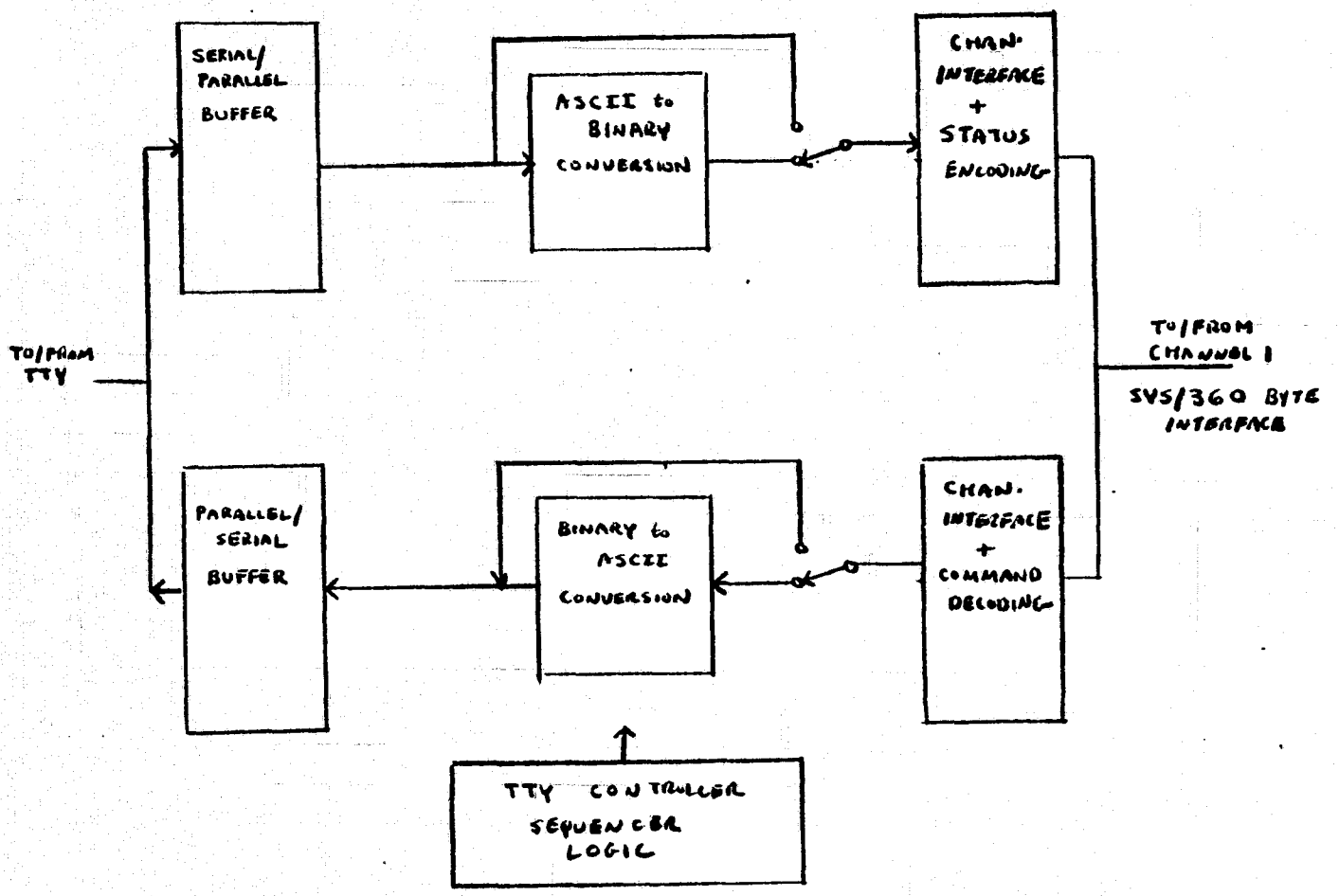


FIGURE G-7 DMS CHANNEL (CHANNEL 0)

2/26/79

6.1.2.0

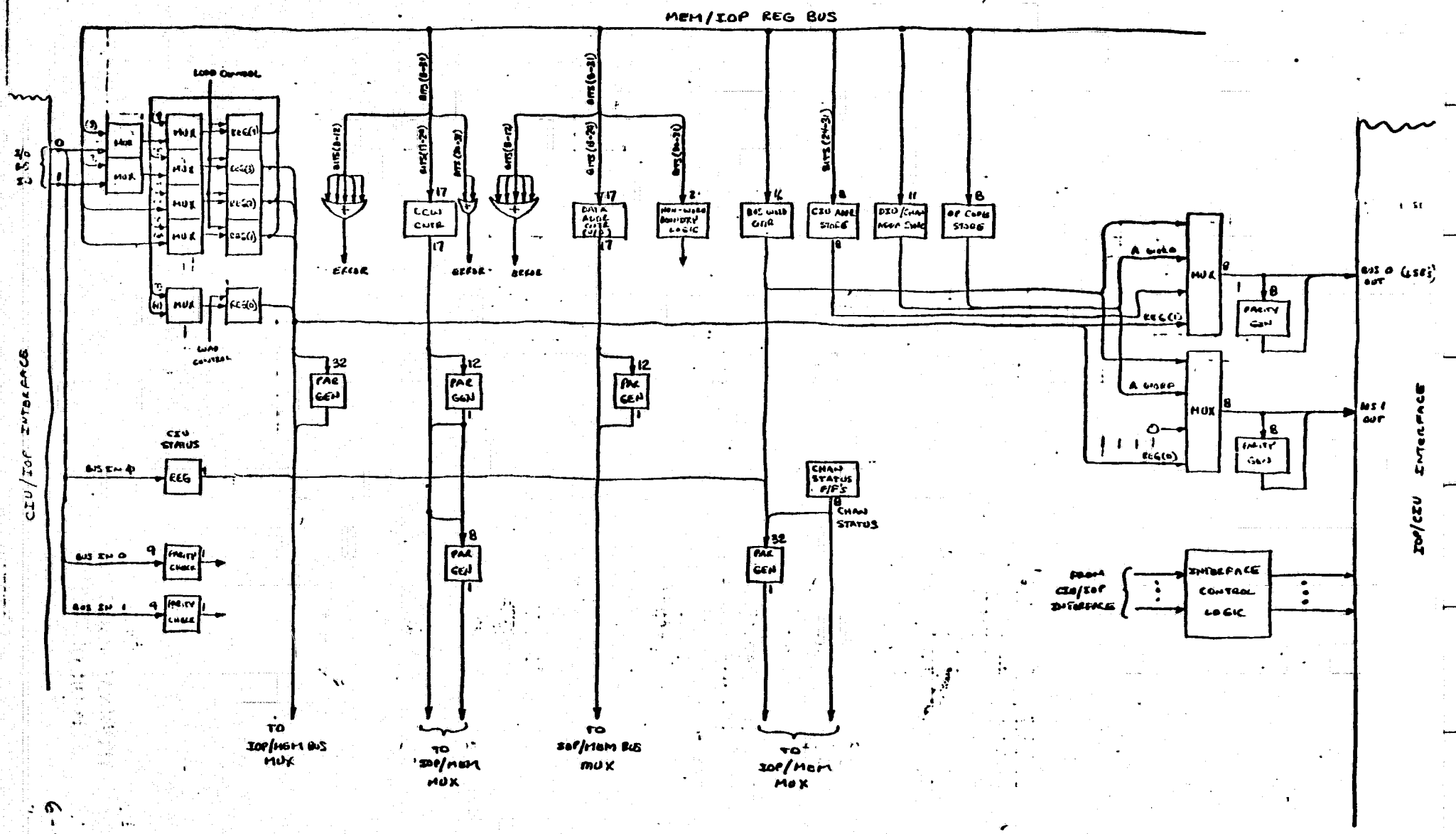


FIGURE G-8 COMMON IOP

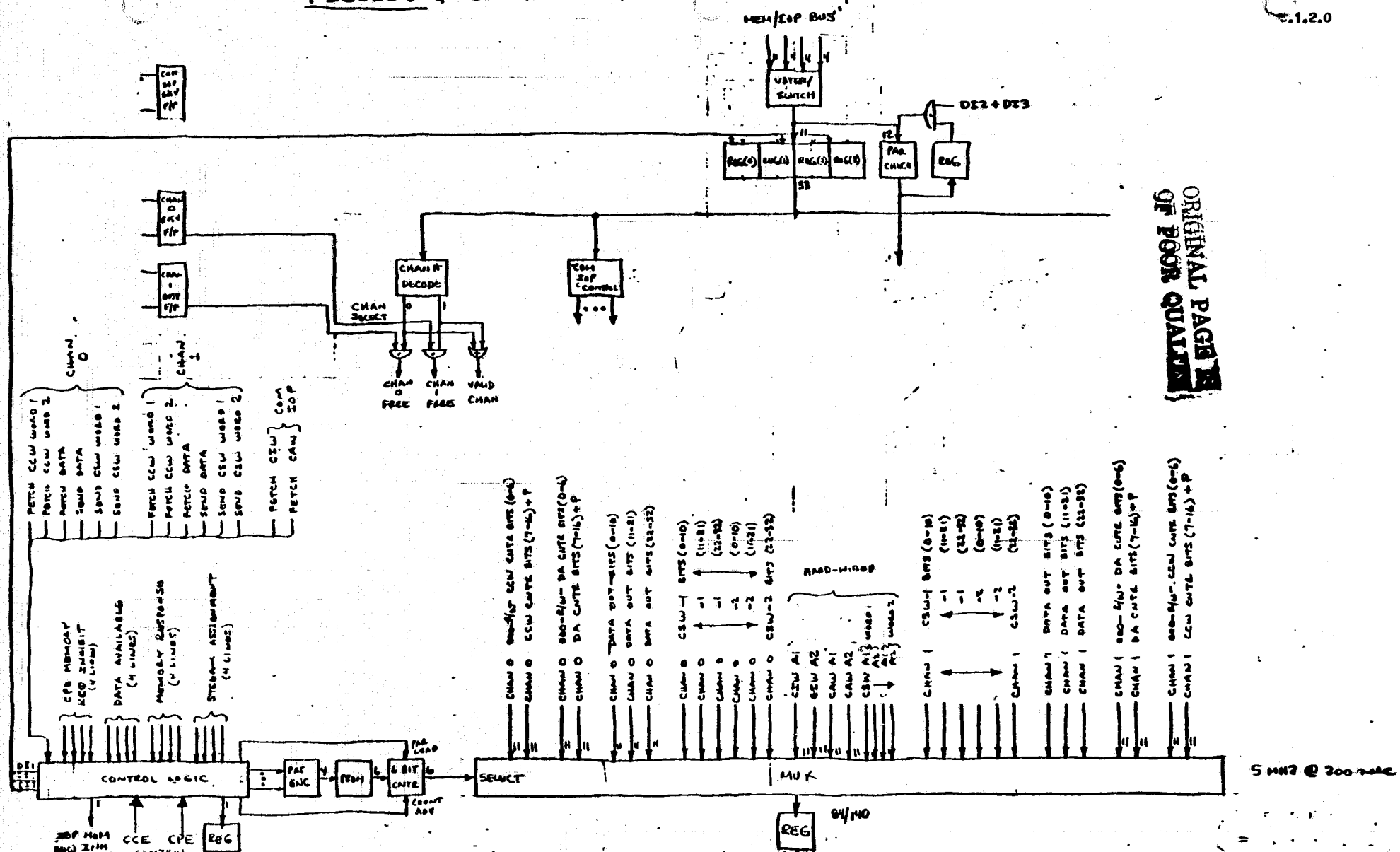


TABLE 6-1. I/O CONDITION CODES

CC \ Instr	Start I/O	Halt I/O	Test I/O	Test Ch
0	Channel Executing Requested Opr	I/O Interrupt Pending	Channel & Device Available →	
1	CSW Ready for Storage →			
2	Channel Busy →			
3	Not Operational →			

TABLE 6-2. IOP COMPONENT COUNT

Function	Gates	Flip-Flops	ROM bits	Fail Rate Equiv Gates*
1. Data Terminal Controller	770	92	6,400	2,106
2. TTY Channel (Ch 1)	2,346	252	256	3,874
3. Common IOP Logic	2,435	130	4,864	3,823
4. DMS Channel (Ch 0)	2,699	274	256	4,359
Total	8,250	748	11,776	14,162

*Assumes failure rate of a gate = failure rate of 8 ROM bits = failure rate of 1/6 Flip-Flop

IOP INTERFACE

IOP - MEMORY

1.	Processor Buss (11x4)	IOP to Memory	-TPMB___
2.	Memory Buss (11x4) = (44)	Memory to IOP	-TMPB___
3.	Memory Request (1)	IOP to Memory	-TIOREQ_
4.	Memory Request Acknowledge (4)	Memory to IOP	-TMRESP_
5.	Data Available (4)	Memory to IOP	-TDAVAL_

IOP - CCE

1.	IOP Stream Assignment (4)	CCE to IOPs	-TSASN__
2.	Clock Sync (2)	CCE to IOP	-TCLK, TSYNC
3.	Panic Halt Interrupt (1)	CCE to IOPs	-TPANICH
4.	IOP Available (1)	IOPs to CCE	-TRBPAC5
5.	IOP Fault (1)	IOPs to CCE	-TIOFLT

IOP - MAINTENANCE PANEL/ELECTRONICS

1.	Fault Entry Switches (6)	Panel to all	SSELTIN_
2.	Scanout Source Select (8)	Panel to all	SSOC___
3.	Scanout Buss (66)	All to Electronics	TSBUS__
4.	Scanout Enable (1)	Electronics to all	TSOMS__
5.	Master Clear (1)	Electronics to all	-TCLR

IOP - CPE

1.	IOP Memory Request Inhibits (1/IOP)	IOP to CPE	-TIOINH_
2.	CPE Memory Request Inhibit (1)	CPE to IOPs	-TCPINH_
3.	Initiate I/O Instruction (1)	CPE to IOP	-TIIOI_
4.	I/O Instruction Accepted (1/IOP)	IOP to CPE	-TIOIAK_
5.	I/O Interrupt (2/IOP)	IOP to CPE	-TIOINT_
6.	I/O Condition Code (2/IOP)	IOP to CPE	-TIOPCC_
7.	I/O Interrupt Acknowledge (2)	CPE to IOP	-TINTAK_

IOP - MEMORY

1. Processor Busses (44) - The IOP sends 4 busses to all memories. The buss is utilized to send memory address, memory data and controls to the memories.

2. Memory Busses (44) - Each memory sends the CPE (and IOP) 4 busses each containing 11 lines. The busses carry memory data. The IOP votes on this data in duplex and TMR modes.

3. Memory Access Request (4) - The IOP sends one Memory Access Request for each buss to all memories, the memory which is selected by the page being transmitted on the Processor Buss will respond.

4. Memory Request Acknowledge (1/MB) - Each memory responds to the IOP via this line acknowledging that the memory request has been accepted and that the requesting IOP should transfer the remaining address and data (if a write) across the buss.

5. Data Available (1/MB) - Each memory sends the CPE the Data Available signal when memory data is available on the Buss.

IOP - CCE

1. IOP Stream Assignment (4) - The CCE sends IOP 4 lines specifying the stream assignments for voter switch and output bus control. Each bit specifies an IOP in the stream.

2. Clock, Sync (2) - The CCE sends IOP the system clock at the highest frequency used in the system and a sync signal at 1/2 this frequency.

3. Panic Halt Interrupt (1) - The CCE sends each IOP a Panic Halt Interrupt which causes the IOP to terminate the operations initiated and stop.

4. IOP Available (1) - Each IOPs sends the CCE a signal which specifies that the IOP has completed all outstanding operations and is quiescent.

5. IOP Fault (1) - Each IOP sends the CCE a line which specified that a fault condition has occurred in the IOP.

IOP - MAINTENANCE PANEL/ELECTRONICS

1. Fault Entry Switches (6) The fault entry switches allow the insertion of switchable faults into any 6 desired points in the IOP for fault tolerance studies.

2. Scanout Source Select (8) The scanout source select switches allow selection of up to 16 different scanout sources independently for each of the two 33 bit scan-outs on the maintenance panel.

3. Scanout Buss (66) - The tri-state scanout buss multiplexes inputs from scanout sources in each module into the maintenance panel scanouts.

4. Scanout Enable (1) - The maintenance electronics decodes the scanout module select switch output to provide an enable to the scanout multiplexers in the module selected.

5. Master Clear (1) - A master clear is provided to all modules when the master clear pushbutton on the maintenance panel is activated.

IOP - CPE

1. IOP Memory Request Inhibit (1/IOP) - Each IOP sends the CPE a signal which represents that a memory cycle by the IOP is being requested and the CPE should inhibit any memory request until this line is low. The fact that an IOP shares the input and output busses with a CPE necessitates this communication line between the sharing IOP and CPE. The CPE must look at all IOP request inhibit lines when in a duplex or TMR mode of operation.

2. CPE Memory Request Inhibit (1) - The CPE sends a signal to each IOP in the system indicating that this CPE has requested a memory cycle and the memory has granted the request. In other words, the CPE is using the input buss and possibly the output buss while the signal is true. The IOP has a higher memory request priority than the CPE, so this signal must not be activated (high) until the memory has granted the request.

3. Initiate I/O Instruction (1) - The CPE sends the IOP a signal telling the addressed IOP that an I/O instruction is to be processed. The CPE must store the channel number, device number and subchannel number into a designated memory location, from which the signalled IOP can interrogate to determine what action is to be taken. It is the responsibility of the program to know when the I/O instructions can store in the designated memory location.

4. I/O Instruction Accepted (1/IOP) - Each IOP sends the CPE a signal which denotes the previously sent I/O instruction is completed or has progressed to the point to assure initialization of the requested function. The CPE must wait for the I/O instruction complete signal before completing the I/O instruction and setting the condition codes.

5. I/O Interrupt (2/IOP) - The IOP sends CPE an interrupt line for each channel which designates one of many conditions that may exist in the IOP which must be signalled to the system. The IOP stores the interrupt information in a memory location (64) to be interrogated by the interrupt program. The CPE processes the interrupt if the I/O interrupt mask is off.

6. I/O Condition Code (2/IOP) - Each IOP sends the CPE two lines which represents the IOP condition, following the execution of an I/O instruction, which is to be loaded into the CPE condition code indicators. These lines are interrogated by the CPE while the I/O instruction accepted line is high.

7. I/O Interrupt Acknowledge (2) - The CPE acknowledges the 2 IOP channels' interrupts via these lines.

7. MAINTENANCE STATUS PANEL AND ELECTRONICS DESCRIPTION

The maintenance/status panel and electronics (MSPE) allow displaying and controlling the ARMS breadboard's internal state, and inserting faults to test and demonstrate ARMS fault tolerance logic. It provides the following capabilities:

a) loading the memory register in the CPE from thumbwheel switches. From this register other CPE registers and scratchpad memory can be loaded, and data may be written into main memory modules with the function initiate button depending upon settings of the function control switches. Parity for data and address switches is generated by the maintenance electronics.

b) a breakpoint program halt and single-step capability.

c) "stuck at" and intermittant faults introduction by forcing IC pins to logical "0" (and hence their outputs to logical "1") by means of panel switches whose terminals may be temporarily connected to various appropriate points in ARMS prior to a test or demonstration.

d) simulation of ground command loading of new module status assignments into the CCE.

e) display of important register contents in ARMS CPE, IOP, main memory, and CCE modules through a multiplexer along with direct display of key points within these modules. Both binary and hexidecimal readouts are provided.

f) Master clear, External Interrupt, and Single Clock control

Figure 7-1 is a block diagram of the MSPE showing the interconnections between the panel switches and displays and the electronics assembly. Figure 7-2 shows the location of the various controls and displays on the panel. The controls operate as follows:

HEX DISPLAY SOURCE SELECT - Selects which binary display (32 msbs) will also be displayed hexidecimally.

FAULT INPUTS 1,...,6 - Each switch solidly (switch down) or momentarily (switch up) grounds a fault line that is routed to each ARMS module through the interconnect board. Within any module these lines can be connected as desired.

EXTERNAL INTERRUPT - Provides an external interrupt to the CPEs via the CCE.

STOP (LOAD CCE)/RUN - RUN (lighted) position allows normal operation. LOAD position allows loading new status assignments into the CCE with the FUNCTION INITIATE P/B as selected by the FUNCTION ACTIVATE DigiSwitches. Normal CCE reconfiguration action is inhibited until this switch is returned to the RUN position.

MASTER CLEAR - This button clears all control registers in all ARMS modules.

SINGLE CLOCK ACTIVATE - ACTIVATE (lighted) position inhibits all normal ARMS clock signals.

SINGLE CLOCK P/B - This button provides one 5.0 MHz clock pulse to all clock lines whenever it is pushed.

FUNCTION INITIATE - Executes the function selected by the FUNCTION ACTIVATE DigiSwitches.

SINGLE COMMAND ACTIVATE - ACTIVATE (lighted) position stops all CPE instruction execution on the first microinstruction of the instruction fetch sequence allowing stepping through a CPE program one instruction at a time.

BREAKPOINT ACTIVATE - ACTIVATE (lighted) position forces the CPE to stop on the first microinstruction of the fetch cycle in which the Program Counter contents equals the breakpoint address selected by the ADDRESS ENTRY DigiSwitches.

SINGLE COMMAND/BREAKPOINT STEP. Each time this button is pushed the CPE program will stop one instruction if the SINGLE COMMAND mode is activated or will advance from the breakpoint it stopped on if BREAKPOINT mode is activated.

ADDRESS AND DATA ENTRY DIGISWITCHES - These switches allow entering byte addresses and accompanying data to the CPE. The 5 lsbs of the address switches also allow loading module status assignments into the CCE.

FUNCTION ACTIVATE DIGISWITCHES - These switches select the module (lefthand switch) and function (righthand switch) to be controlled by the FUNCTION INITIATE button, according to Table 7-1.

SCANOUT SOURCE SELECT DIGISWITCHES - These switches select the module (lefthand switch) and function to be displayed on scanouts A (middle switch) and B (right hand switch). Module select 0-3 selects CPEs 1-4, Module Select 4-7 selects Memories 1-4, Module select 8 and 9 select the CCE and IOP respectively.

Figure 7-1

ARMS MAINTENANCE / STATUS PANEL / ELECTRONICS

BLOCK DIAGRAM

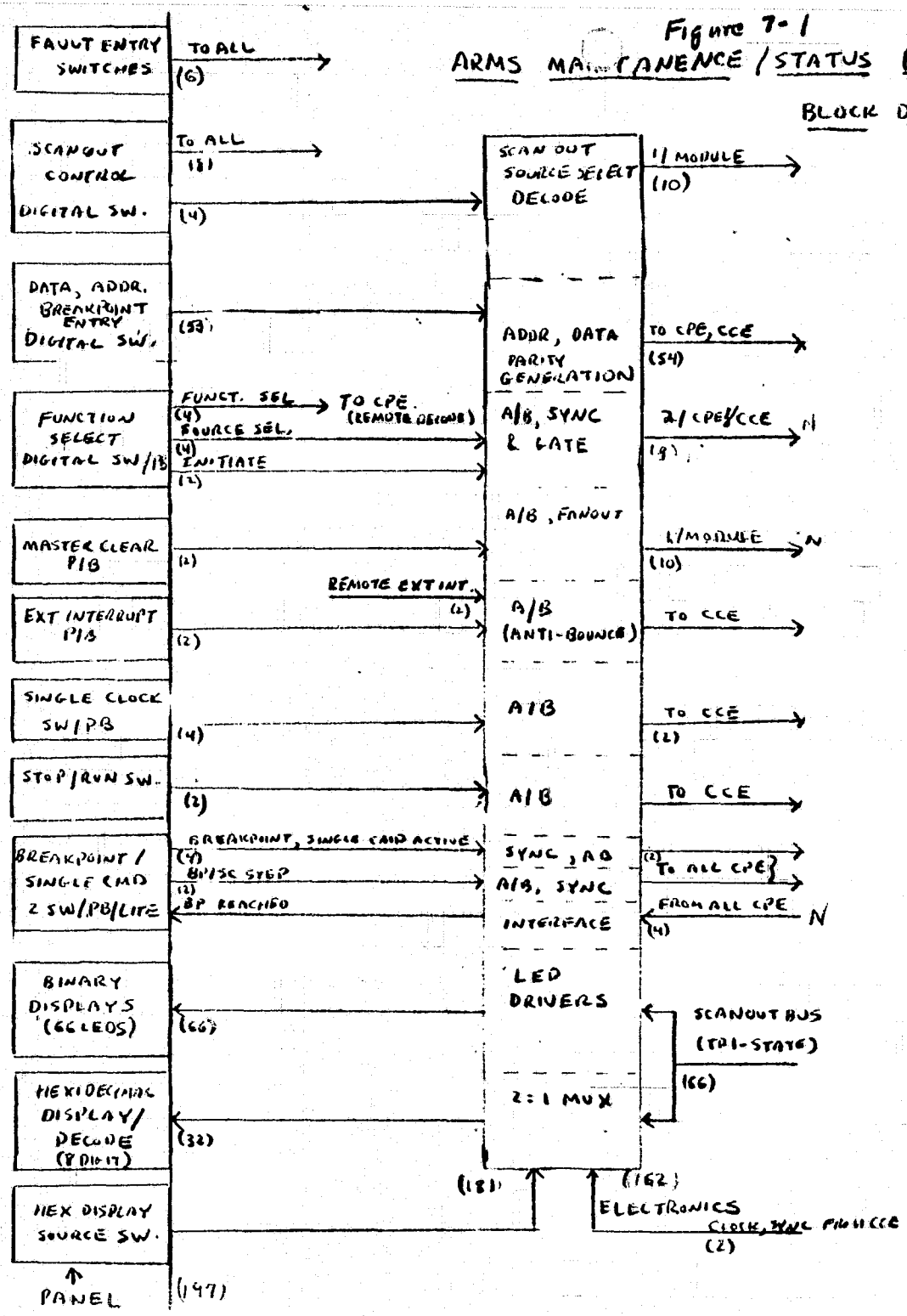
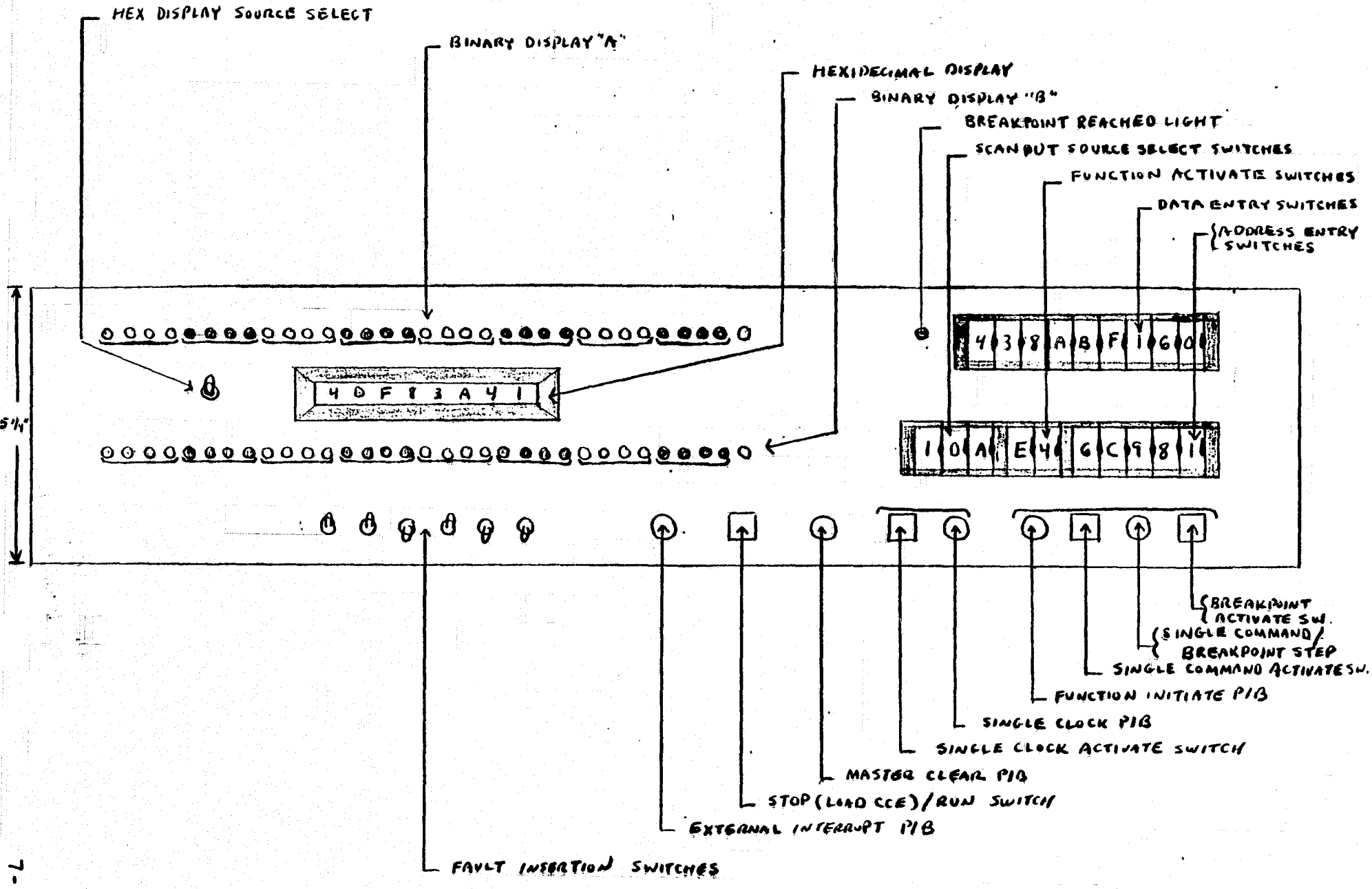


Figure 7-2

ARMS MAINTANENCE/STATUS PANEL LAYOUT



NOTE: EXTERNAL INTERRUPT AND IPL PIB AVAILABLE ON TTY

TABLE 7-1. FUNCTION ACTIVATE SWITCHES

Module Select (Left)	Modules To Execute Function					Function Select (Right)	Function To Be Executed
	CPE No. 4	CPE No. 3	CPE No. 2	CPE No. 1	CCE		
0	No	No	No	No	Yes		CCE SELECTED
1	↓	↓	↓	Yes	No	0	LOAD CPE STATUS
2	↓	↓	Yes	No	↓	1	LOAD MEMORY STATUS
3	↓	↓	↓	Yes	↓	2	LOAD IOP STATUS
4	↓	Yes	No	No	↓	Other	NO-OP
5	↓	↓	↓	Yes	↓		CPE SELECTED
6	↓	↓	Yes	No	↓	0	MUR READ
7	↓	↓	↓	Yes	↓	1	MUR WRITE
8	Yes	No	No	No	↓	2	MEMORY READ
9	↓	↓	↓	Yes	↓	3	MEMORY WRITE
A	↓	↓	Yes	No	↓	4	LOAD PSW_{0-31}
B	↓	↓	↓	Yes	↓	5	LOAD PSW_{32-63}
C	↓	Yes	No	No	↓	6	READ PSW_{0-31}
D	↓	↓	↓	Yes	↓	7	READ PSW_{32-63}
E	↓	↓	Yes	No	↓	8	LOAD IR
F	↓	↓	↓	Yes	↓	9	LOAD PRR
						A	LOAD MAR
						B	LOAD MQR
						C	LOAD ROM ADDR. REG.
						D	INITIAL PROG. LOAD
						E	NO-OP

MAINTENANCE PANEL/ELECTRONICS INTERFACE

MAINTENANCE PANEL/ELECTRONICS - ALL MODULES

1. Fault Entry Switches (6)	Panel to all	SFLTIN_
2. Scanout Source Select (8)	Panel to all	SSOC___
3. Scanout Buss (66)	All to Electronics	TSBUS__
4. Scanout Enable (1/module - tw)	Electronics to all	TSOMS__
5. Master Clear (1/module - tw)	Electronics to all	-TCLR

MAINTENANCE PANEL/ELECTRONICS - CCE ONLY

1. Panel Control (12)*	Electronics to CCE	_
2. Clock, Sync (2 - tw)	CCE to Electronics	TCLK, TSYNC

*Not included in motherboard cabling.

MAINTENANCE PANEL/ELECTRONICS - CPE ONLY

1. Panel Data, Address (54)	Panel/Electronics to CPE	SPADR__, SPDAT__
2. Panel Function Select (4)	Panel to CPE, CCE	SPANF__
3. Panel Function Initiate (2/CPE - tw)	Electronics to CPE	-TPFINT - TPMURQ-
4. Breakpoint, Single Command Activate, Step (3 - tw)	Electronics to CPE	TBPTACT, TSCMACT, TCMSTEP
5. Breakpoint Reached (1/CPE - tw)	CPE to Electronics	TBPTLED, - TBKPT -

NOTE: All lines are single-ended unless marked tw (twisted pair).

MAINTENANCE PANEL/ ELECTRONICS INTERFACE

MAINTENANCE PANEL/ELECTRONICS - ALL MODULES

1. Fault Entry Switches (6) The fault entry switches allow the insertion of switchable faults into any 6 desired points in any modules for fault tolerance studies.
2. Scanout Source Select (8) The scanout source select switches allow selection of up to 16 different scanout sources independently for each of the two 33 bit scanouts on the maintenance panel.
3. Scanout Buss (66) - The tri-state scanout buss multiplexes inputs from scanout sources in each module into the maintenance panel scanouts.
4. Scanout Enable (1/module) - The maintenance electronics decodes the scanout module select switch output to provide an enable to the scanout multiplexers in the module selected.
5. Master Clear (1/module) - A master clear is provided to all modules when the master clear pushbutton on the maintenance panel is activated.

MAINTENANCE PANEL/ELECTRONICS-CCE ONLY

1. Panel Control (12) - Panel control lines to the CCE include Panel Data (5), Panel Function (2) and Load Initiate to allow loading new configuration assignments to the CCE from the panel; a STOP/run line forcing ARMS to stop for reconfiguration; single clock activate and initiate lines; and an external interrupt line.
2. Clock, Sync (2) - The CCE sends the maintenance electronics the system clock at the highest frequency used in the system and a sync signal at 1/2 this frequency.

MAINTENANCE PANEL/ELECTRONICS - CPE ONLY

1. Panel Data, Address (54) - The Panel Data and Address Lines allow loading memory and key CPE registers from the maintenance panel under microprogram control.

2. Panel Function Select (4) The output of the panel function select switch is decoded within each CPE to allow activation of one of up to 16 panel functions for loading memory and key CPE registers.

3. Panel Function Initiate (2/CPE) - Each CPE receives separate panel function initiate lines allowing individual execution of activated panel functions for diagnosis and/or fault insertion. Panel switching allows initiation of activated functions in from 1 to 4 CPES simultaneously. One line provides a synchronized output, the other a direct output from the pushbuttons antibounce circuit for MUR operations.

4. Breakpoint, Single Command Activate, Step (3) - Panel switches allow activation of breakpoint and single command modes of operation within the ARMS CPE. The Breakpoint/Single Command Step switch allows stepping the program beyond the breakpoint in the breakpoint mode or stepping it ahead one instruction in the single command mode.

5. Breakpoint Reached (1/CPE) - When a CPE stops at a breakpoint, this line signals this condition to the maintenance panel indicator.