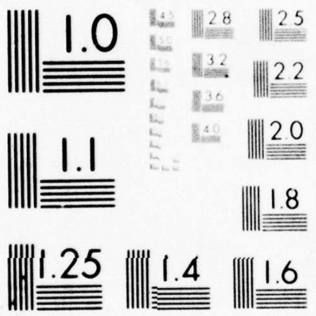
AD-A034 880 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 9/2
DESIGN OF THE BUS INTERFACE UNIT FOR THE DISTRIBUTED PROCESSOR/--ETC(U)
DEC 76 R C SIMPSON
UNCLASSIFIED GE/EE/76D-40 NL

1 OF 2
AD
A034880


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

GE/EE/76D-40

1

DESIGN OF THE BUS INTERFACE UNIT FOR THE DISTRIBUTED PROCESSOR/MEMORY SYSTEM

THESIS

GE/EE/76D-40 Robert C. Simpson
Major USAF

14 GE/EE/76D-40

6 # DESIGN OF THE BUS INTERFACE UNIT FOR THE DISTRIBUTED PROCESSOR/MEMORY SYSTEM.

THESIS 9 Master's thesis,

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

10 Robert C./Simpson, B.S.E.E.
Major USAF

Graduate Electrical Engineering

11 December 1976

12 178p.

Approved for public release; distribution unlimited

1473
012225
LB

Preface

This study was sponsored by AFAL/AAT in an effort to develop a Distributed Processor/Memory Avionics Interface System. The purpose of the study is to design the Bus Interface Unit (BIU) for the Processor/Memory Element, a standalone processor in a distributed network that is used to interface the avionics systems on board an aircraft. The BIU was originally designed as a hardware module but with the advances in LSI circuitry, it was decided to implement this system with a microprocessor. This change in the method of implementation and my limited knowledge in this area made this project very challenging since it is an attempt to emulate a complex hardware design with software.

I want to express my appreciation to Captain J. B. Peterson, my advisor, for his many hours of guidance and encouragement throughout this project. I also want to thank Deiter Schiller from AFAL/AAT who spent many patient hours helping me learn the BIU structure. Most of all, I want to thank my wife, Sheran, whose encouragement, support, and typing made this entire project possible.

Robert C. Simpson

Contents

List of Figures

List of Tables

Abstract

The Distributed Processor/Memory (DP/M) system is a concept developed by the Air Force Avionics Laboratory (AFAL). The DP/M system is a decentralized, software programable Processor Element (PE) which is used to interface the avionics on board an aircraft. Each PE or group of PEs is connected to an aircraft sensor and the PE's are interconnected via a global communication bus. The key characteristic of the DP/M is decentralization (i.e., each avionic interface module, PE, determines when it will access the global bus whereas present systems are controlled by a central computer).

This report is in support of a request from AFAL to design the Bus Interface Unit (BIU) of the PE. The BIU provides the communication link between a serial global bus and a parallel data PE processor. The original design concept was to implement the BIU in hardware, but with the advances in high-speed large scale integrated circuits, the system design and testing would be more flexible if a special purpose microprocessor was used.

This study, after defining the BIU requirements in detail, uses the AM2900 Bipolar Microprocessor chip set (Advanced Micro Devices, Sunnyvale, California) to implement the BIU. The hardware, microword format, and data reception microcode were developed.

The high-speed and flexibility of the AM2900 chip set as implemented shows that a microprocessor can be used to meet the requirements of a complex hardware system.

DESIGN OF THE BUS INTERFACE UNIT FOR THE DISTRIBUTED PROCESSOR/MEMORY SYSTEM

I. Introduction

Background

The complex high-speed aircraft designed for today's Air Force have resulted in many new concepts for integrating avionics systems. Initially, these systems were independent of each other and were implemented with analog devices. As aircraft became more complex and aircraft weight increased, better methods were needed to provide the information required to fly the high-speed missions.

The development of Large Scale Integrated (LSI) circuitry and small digital processors has lead to many new concepts for integrating aircraft avionics. These new processors provide fast, reliable, and light weight systems which are truly compatible with new and future aircraft. The computer systems presently operational (e.g., T-43, F-111) are highly centralized (because of size, weight, and power requirements) and usually rely on one central computer to process and distribute information to various sensors and displays. This concept relies heavily on this central computer and if it fails (or the communication interface systems fail) the mission must be aborted or drastically modified with highly degraded performance.

If the system could be decentralized by using a group of smaller processors (small, specialized minicomputers), the resulting system would not be highly dependent on one specialized central computer. This concept could be used on any new aircraft design. This would reduce the hardware cost of the avionics package tremendously. The only hardware modification required would be the interface module between the aircraft sensor and the processor. The software (usually the largest expense) would require modification only in very specialized areas (based on the requirements of the aircraft type).

These smaller processors (miniprocessors) or groups of miniprocessors would receive raw data from one aircraft sensor (e.g., airspeed, loran, radar, doppler), process it, format it, and make it available on a "global" bus to all the other miniprocessors that needed it.

This system concept would be highly reliable since the loss of one miniprocessor would not result in total system failure. System performance would be degraded but still highly effective and the mission could continue.

This concept was formalized in the Air Force Avionic Laboratory (AFAL) and a contract was released to Texas Instruments (TI) to determine if this type of decentralization was functionally feasible.

TI's Distributed Processor/Memory Architecture Study

The TI program consisted primarily of three major objectives: (1) functional design of the Distributed

Processor/Memory (DP/M) hardware and high level simulation analysis, (2) functional design of the DP/M executive software, and (3) fault tolerance analysis (Ref 1:14). This study was based on an estimate of the state-of-the-art of hardware and software in the 1980 time-frame.

<u>DP/M Functional Design</u>. The DP/M network concept consists of a dual-redundant global bus for communication between processor elements (PEs) or groups of processor elements (See Fig. 1). There is a local bus for communication within a group of PEs (Affinity Group). An Affinity Group (AG) would be required when a single PE could not process all the data available from one aircraft sensor or effector (e.g., hydraulic system, control surface). Both the global and local bus use a distributed time division multiplex (TDM) round-robin communication system where each PE or AG has a programmable predefined bus allocation time slot. The programmable bus allocation procedure provides limited super-commutation (multiple bus access time slots within one round-robin cycle) of bus transmissions. Each transmitted message begins with a message identification header which allows all recipient PEs to determine if the message is for them.

The global bus interface provides for dual-redundant TDM communication lines between PEs/AGs which allows each PE/AG to participate in primary bus communication and simultaneously monitor the secondary bus for a "switch-input-to-alternate-bus" command. The local bus interface

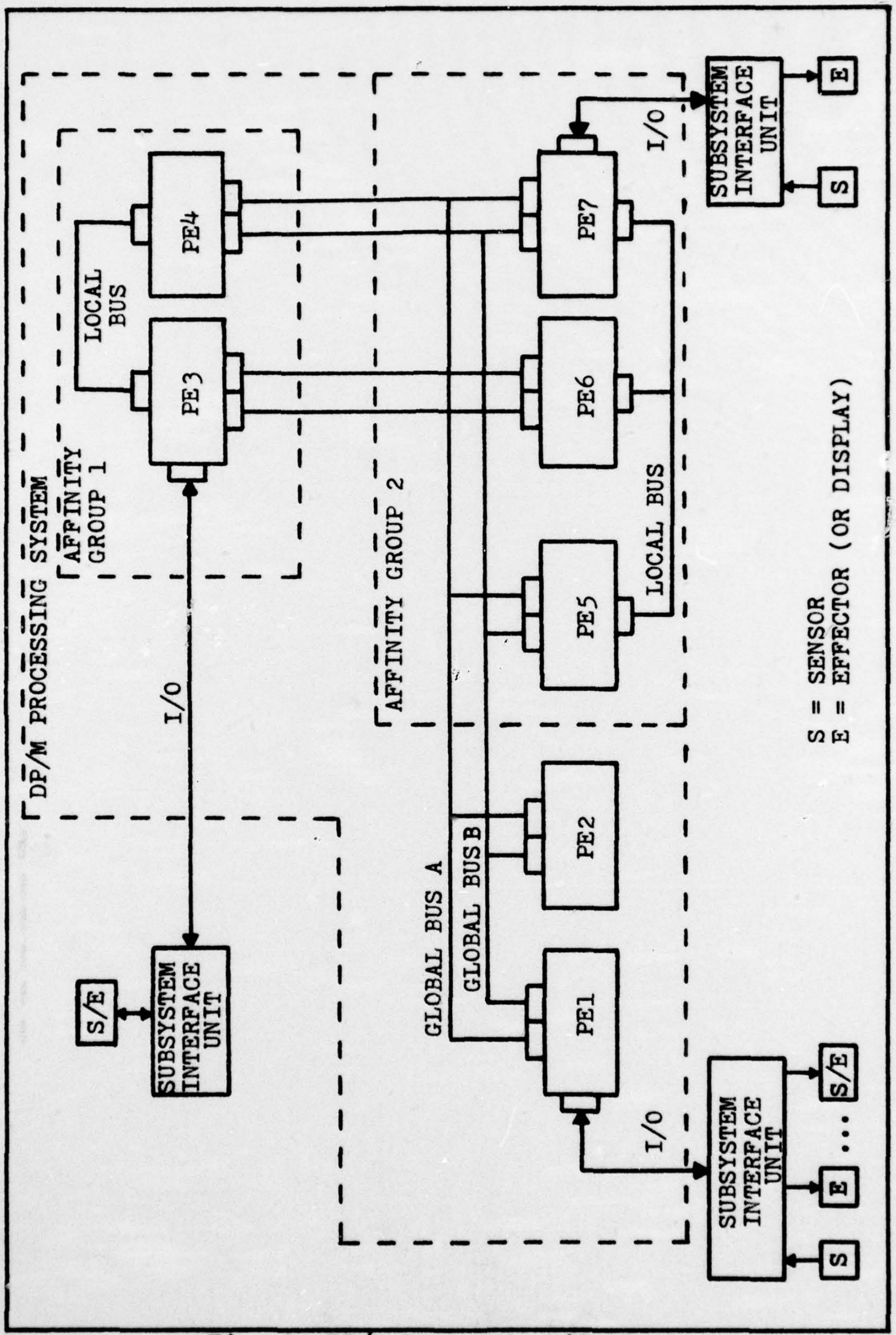


Fig. 1. DP/M System Architecture

allows PEs within an AG to communicate with each other using the global message format (Ref 1:15). This concept is depicted in Fig. 1.

Each PE consists of a Central Processor Unit (CPU), Memory, Local/Global Bus Interface Unit (BIU), and Input/Output Interface Unit (IOIU). These four modules are allowed to communicate through a shared internal bus (I-Bus). This structure is shown in Fig. 2.

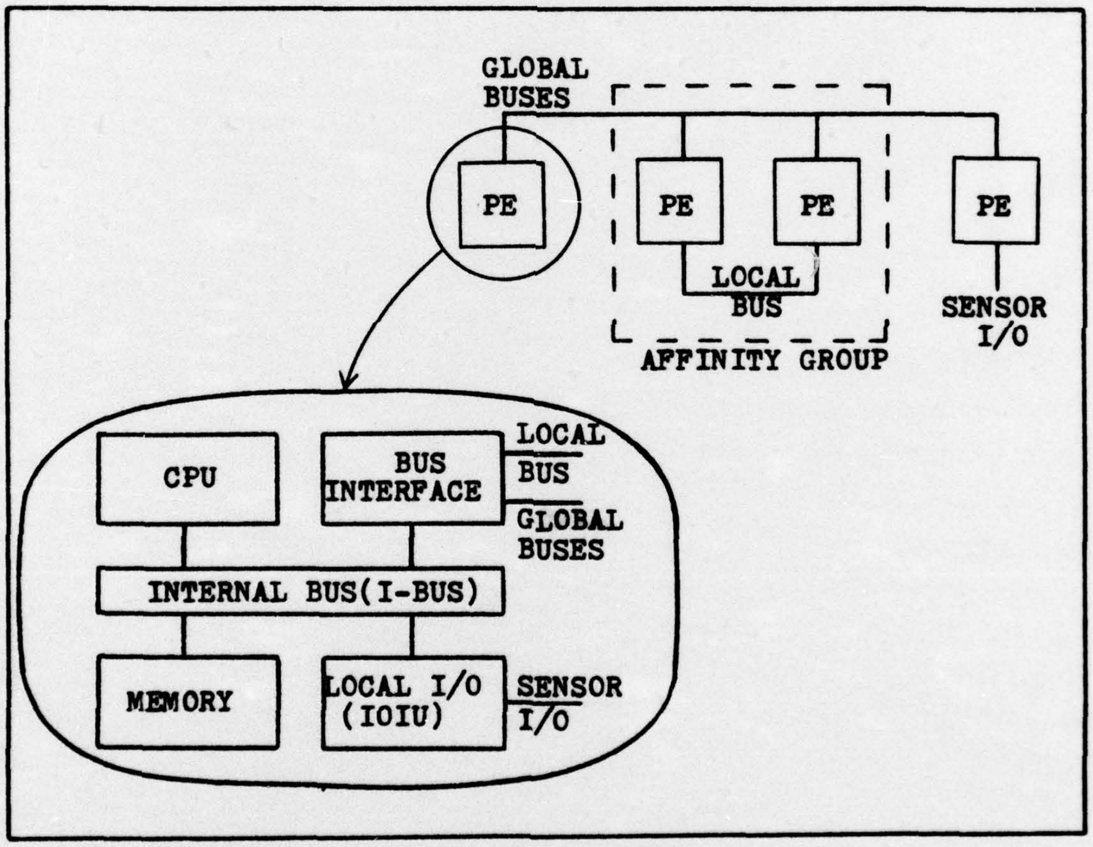


Fig. 2. PE Structure

The processor (CPU) design specifications require a 16-bit, 8-register file microprocessor under microprogram control. The processor is capable of addressing up to 65K words of memory but most avionic applications will require only 4K to 8K of memory. The processor uses forty 16-bit and 32-bit instructions (Ref 1:16).

The I-Bus structure consists of a standardized set of 80 control signals (including 20 for power and ground) which facilitate intra-PE data transfer and control. This standardization allows PE modules designed with different technologies to be intermixed.

The IOIU provides the basic PE external data transfer, command, and status information between aircraft sensor/ effector and the I-Bus.

The BIU was designed based on functional hardware modularity to provide the two levels of interfacing required (global and local). Thus, the BIU is divided into six functional modules for global bus management and five (of the six) for local bus management (Fig. 3). The six modules include: (1) Bus Interface Translation Unit (two required in global area), (2) Message Reception Control, (3) Message Transmission Control, (4) Redundant Bus Management (global only), (5) Bus Access Control, and (6) Processor/Memory Interface (for I-Bus).

The BIU modules contain the necessary functions to support a distributed round-robin bus scheme with message broadcast capability to multiple PEs/AGs (Ref 1:17). It is able to identify input message identification headers and automatically vector these messages into software selectable PE memory buffers. This ability to vector data

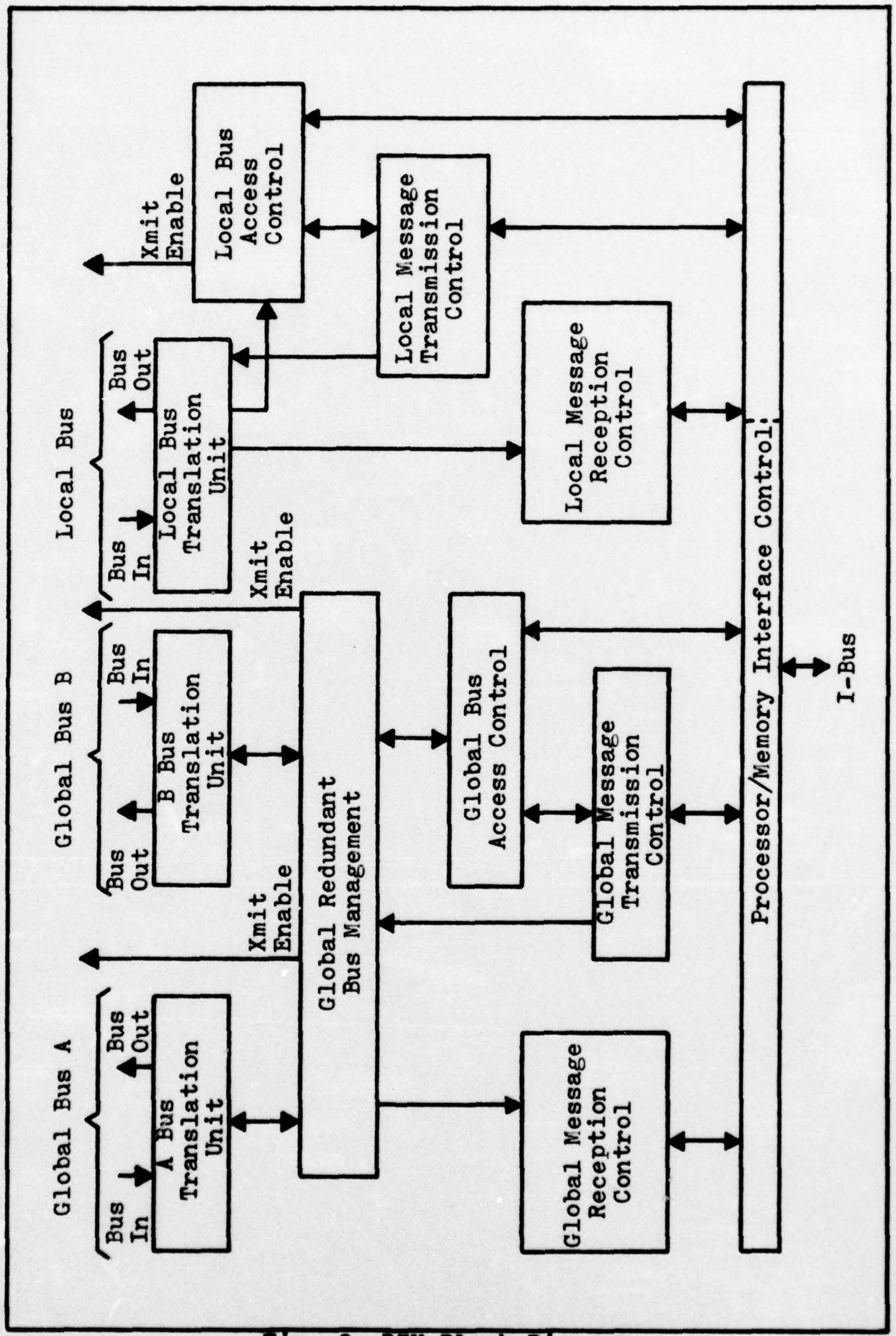

Global Bus A
Bus In
Bus Out
A Bus Translation Unit
Xmit Enable
Global Bus B
Bus Out
Bus In
B Bus Translation Unit
Xmit Enable
Local Bus
Bus In
Bus Out
Local Bus Translation Unit
Xmit Enable
Local Bus Access Control
Local Message Transmission Control
Local Message Reception Control
Global Redundant Bus Management
Global Bus Access Control
Global Message Transmission Control
Global Message Reception Control
Processor/Memory Interface Control
I-Bus

Fig. 3. BIU Block Diagram

(Direct Memory Access) relieves the processor from the task of continually setting up to service the BIU. This significantly increases response time for both global and local operations. This initial design concept envisioned the BIU implemented in hardware with more sophisticated hardware implementation as the state-of-the-art advanced (Ref 1:17).

TI designed the functional register transfer language description of the PE and bus interaction. Using this description, a System Network Simulator was designed to model the asynchronous operation of multiple PEs and distributed round-robin bus control on local and global PE bus interfaces. The PE simulator provided a register level emulation of the four DP/M sub-elements. The user is able to observe timing and control information transfers on global, local, and internal buses as well as those internal to the PE itself (Ref 1:16). This simulation demonstrated the feasibility of the DP/M concept and provided the data for the AFAL design study.

<u>Executive Software</u>. A functional specification of a two level control structure was also provided. The Global Executive (GEX) and Local Executive (LEX) were functionally designed. The GEX is responsible for system scheduling, system fault recovery procedures, and mode control (e.g., pilot request for data or change of flight profile which requires new data). The LEX provides the necessary control for tasks assigned to a given PE.

<u>Fault Tolerance Analysis</u>. The reliability of Large

Scale Integration (LSI) devices is expected to be much greater than that of avionic sensors and, therefore, DP/M is not expected to be a large source of faults. Physical fault-tolerance is achieved through redundancy techniques similar to those used in fly-by-wire flight-control systems. Functional fault-tolerance includes software techniques for achieving degraded modes of operation based on the type of error detected. Error detection features have been included in the PE processor, BIU, Memory, and IOIU to aid in graceful degradation and recovery procedures (Ref 1:20).

The Problem and Approach

The TI study was completed in February 1975. During 1975, AFAL modified and improved the TI study and issued a preliminary design draft (Ref 2). Presently, AFAL is working on the design of a laboratory test model to demonstrate the feasibility of this system.

The objective of this thesis is to design the Bus Interface Unit (BIU) module of the PE. Initially, the concept was to design the BIU strictly in hardware but the recent advancements in technology have provided new high-speed microprocessors which are capable of emulating the hardware functions of the BIU. Therefore, this thesis will modify the modular concept previously mentioned to provide the microprograms needed to implement the BIU functions in a microprocessor system.

Prior to any actual design, the basic functions of the BIU were studied thoroughly and a highly detailed microroutine flowchart developed. Once this had been done, a high-speed, flexible microprocessor that would be adaptable to a hardware design was needed. A new LSI group, the Advanced Micro Device's AM2900 Bipolar Family, was selected because it was specifically designed for control applications and provided a vectored interrupt concept that was adaptable to a hardware design. This LSI family was then studied and configured to provide the functions necessary to implement the microroutines. Based upon the microprocessor configuration and the BIU requirements, the microword was defined and used to code the message reception microroutine.

Thesis Outline

The body of this report represents a sequential description of the procedures used to develop the system. Chapter II and Appendices A and B provide a detailed description of the BIU functions which are comparable to the original hardware modules. Chapter III details the AM2900 components used and how they are configured to emulate the BIU. This chapter will also define the microword to be used to program the microroutines. Chapter IV will show how the first microroutine description (message reception routine) was translated into microcode and provide the mnemonic code and flowchart. Chapter V summarizes and discusses actual system construction considerations, coding techniques used, and possible trouble areas. Before beginning the detailed design, several design considerations and possible trouble areas should be discussed.

Design Considerations

The original design concept implemented the BIU in hardware. Several areas must be considered to transition from a hardware configuration to a software system that is implemented with a microprocessor:

(1) The modular BIU functions must be reconfigured to provide microroutines that perform the same functions.

(2) The large number of registers, register transfers, and register manipulations must be implemented with high-speed Random Access Memory (RAM) and an Arithmetic Logic Unit (ALU).

(3) The Bus Translation Unit (BTU) has been omitted from this thesis. This unit (which must meet Military Standard 1553) is presently being designed for AFAL by a contractor. The BTU will be implemented with hardware and the effort required indicates that it could be a thesis topic in itself.

(4) The simultaneous reception/transmission of global and local messages is of primary concern. The data information arrives serially and is converted to parallel data words with no spacing between serial words. This provides a 17 microsecond spacing between I-Bus transfers (1 bit/microsecond arrival rate, each word is 16-bits plus parity). Is this sufficient time for both global and local message reception/transmission routines to alternately receive/transmit data? If the timing requirements cannot be met as specified in the initial system concept, the lab-

oratory test model can be "slowed" down to demonstrate feasibility. As a possible alternative, two processors could be used to divide global and local reception/transmission responsibilities and duties. This alternative and others will be discussed in Chapter V.

(5) This design is primarily a feasibility study. It will determine if the speed associated with the hardware design can be accomplished using a single special-purpose microprocessor. If the AM2900 family will not meet the speed requirements of the BIU, it will help demonstrate the feasibility of the DP/M concept in the laboratory and, hopefully, the state-of-the-art will advance sufficiently to provide a microprocessor to meet all BIU requirements in the 1980 time-frame.

The use of a microprocessor also provides greater flexibility during initial system construction and testing. A hardware system is extremely difficult to modify after it is built, but, when using a microprocessor, the majority of the changes are in software. If Random Access Memory is used during the testing stage (to emulate Programable Read Only Memory), the required software changes are easily implemented.

II. BIU System Description

Introduction

The TI study provided a functional description of the BIU requirements. This chapter provides a general description of the BIU functions and includes the changes recommended by AFAL. A more detailed description is presented in Appendix B.

Communication Interfaces

Global/Local. Since the PE must be able to operate in a totally distributed system environment without centralized control, data transfers must be asynchronous (i.e., transmitting information in a manner which combines clocking information along with the actual binary data in a single signal) where PE transfer events are not strictly time-ordered. This is accomplished with a global/local serial data bus structure which conforms to Military Standard 1553 to ensure compatibility with present-day and future aircraft. Several studies have shown that present and future avionics systems information requirements can easily be met with a 1-Mbps serial bus data rate (Ref 1:35).

The global bus facility is the only central resource of the DP/M system. To alleviate this dependency, a dual-redundant global bus configuration is used. It is the responsibility of the BIU to provide fault tolerance control so that orderly system-software recovery procedures can be initiated. Therefore, the BIU must provide the necessary fault detection mechanisms required to monitor both buses for communication failures within other PEs and respond with the necessary actions to initiate removal of the faulty unit from the bus.

Because of bus length (up to 300 feet) and to provide the reliability required by Military Standard 1553, biphase level Manchester II binary data encoding is used for global/local communications. The information format and the message synchronization signals are shown in Fig. 4. To allow for bus skew (worst case transmission delay for a 300 foot bus), a minimum two microsecond gap-time is required between End Message Synchronization (EMS) and the next Message Header Synchronization (MHS). This technique prevents possible data overlap between PE transmissions.

Internal Bus. The Internal Bus (I-Bus) communication scheme is totally asynchronous in the sense that it is a request-acknowledge system. The BIU requires a parallel 16-bit data line, 16-bit address line, and 18-bit control line interface with the I-Bus.

Global/Local Bus Control

A decentralized control scheme (i.e., the bus access control logic located in each PE) is used to eliminate the dependency on one centralized control module and eliminate total system reliability on one module.

To accomplish decentralization, a bus assignment technique (predetermined and software controlled) which provides

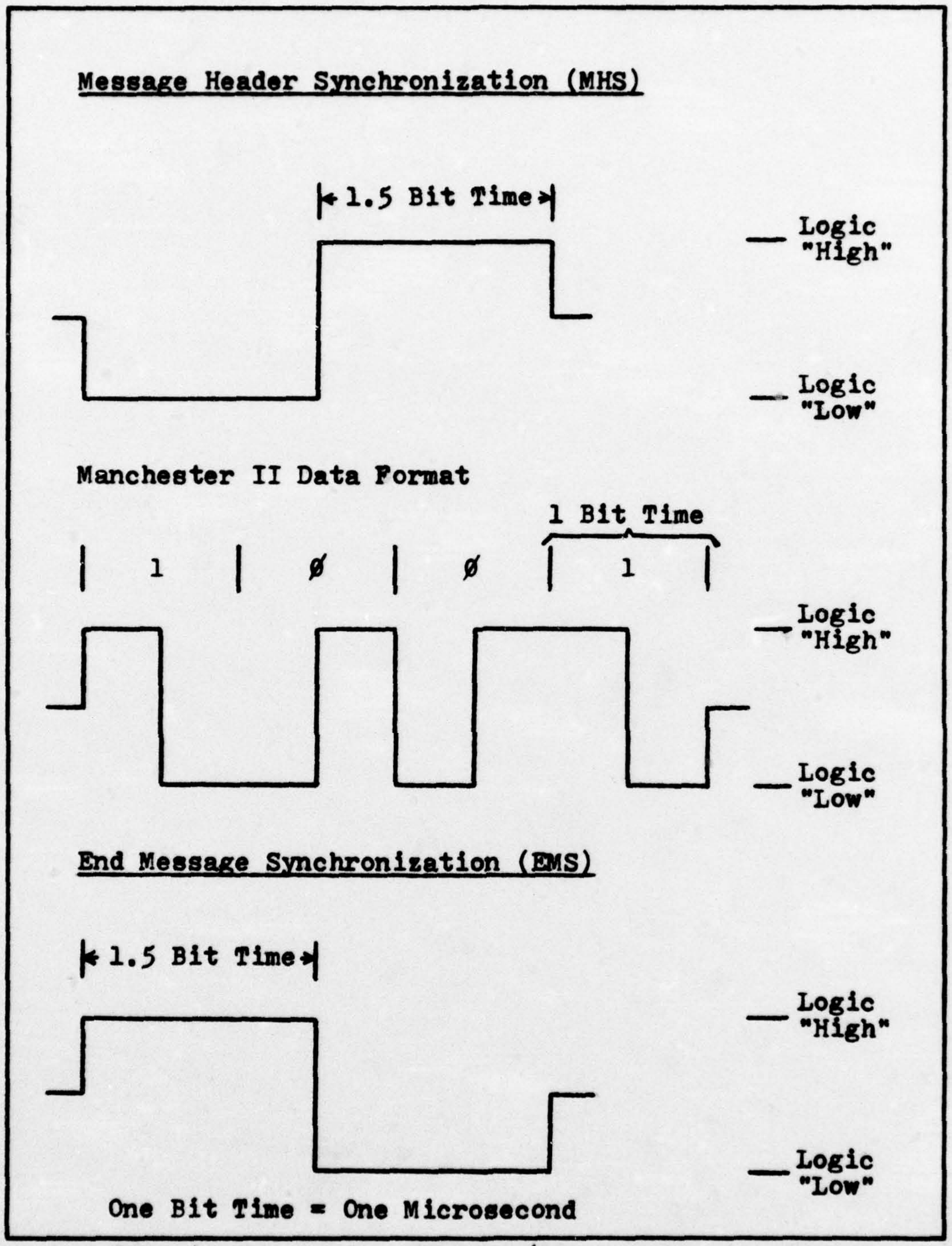

Message Header Synchronization (MHS)
1.5 Bit Time
Logic "High"
Logic "Low"
Manchester II Data Format
1 Bit Time
1
Ø
Ø
1
Logic "High"
Logic "Low"
End Message Synchronization (EMS)
1.5 Bit Time
Logic "High"
Logic "Low"
One Bit Time = One Microsecond

Fig. 4. Manchester II Data/Synchronization Signals

a modified round-robin time-slotting procedure is used. This method provides simple advancing of bus control from PE to PE in a predetermined order. A limited amount of super-commutation is provided to allow PEs with high priority data more frequent access to the bus. Each PE message transmission is terminated with an EMS. If a PE is not ready to transmit data, it immediately passes control to the next PE by transmitting only the EMS.

This method provides a nonresponsive asynchronous (imbedded clock) output demand communication system where each bus user transmits exclusive data messages under its own control after it has acquired control of the bus. Once bus access has been attained, the message transmitted can be a variable number of data words. The normal, maximum length is eight words but exceptions to this rule are possible through software control. When a PE is not transmitting, it remains in an active listening mode as a potential receiver of each transmitted message (Ref 1:45).

Global/Local Message Structure

The output demand message routing procedure requires that each transmitted message must be recognized by all PEs on the bus. Therefore, each message contains a message identification word at the "front" of each transmission. This message header (MH) is received by all listening PEs and decoded to determine if the data message is to be received. To accomplish the required decoding, a Message Identity Associative Address technique is used which can accommodate up to 1,024 unique message headers (data sets). The message format is shown in Fig. 5.

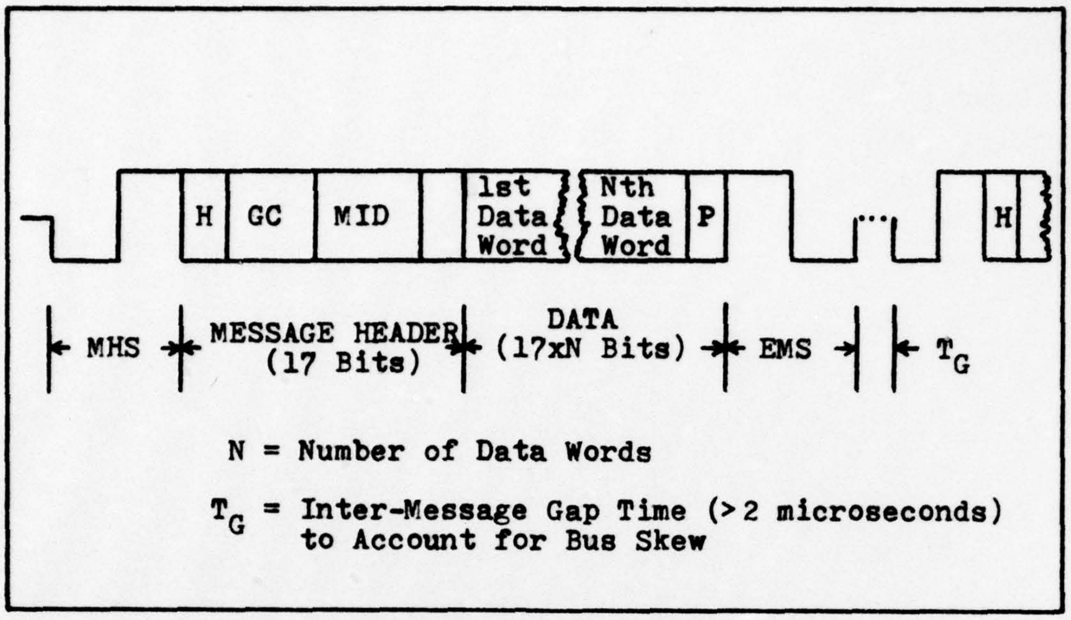


Fig. 5. Message Format

The H-field (defines a high priority message) is used to alert the message recipient(s) that the data requires immediate attention. the GC-field allows specification of special control/status information. At present, this field is being defined as a unique message source identification field for fault tolerance and error recovery procedures. The Message Identity (MID) field specifies the transmitter/receiver link during message data transmission. This field determines the source and type of message data. The MID is defined and interperated by system software. The P-field is used to determine odd-parity for the MH and data. The unique length of each message is placed in the user memory during system implementation. The message header fields are defined in Table I.

Table I

Message Header Fields

Field	Field Length	Field Function
H	1-bit	High Priority Message Tag
GC	5-bits	General-Purpose Control Field (Software Definable)
MID	1Ø-bits	Message Data Set Identity
P	1-bit	Parity Bit

BIU Functional Design

As mentioned in Chapter I, the BIU has been divided into six functional areas (Fig. 3). Each functional area as envisioned by TI is discussed below. This description includes changes recommended by the AFAL preliminary design (Ref 2) and those made as a result of further investigation since this study began.

Bus Translation Unit. The Bus Translation Unit (BTU) is not part of this design although it is functionally part of the BIU. Both AFAL and a contractor are presently designing this portion of the BIU.

The BTU is responsible for providing the interface between the global/local biphase-encoded serial data bus and the parallel-data oriented BIU. This unit provides two major subfunctions:

(1) During message input, it decodes the biphase-encoded data into 17-bit serial binary non-return-to-zero

(NRZ) data and bit-clock information. It scans each message for unique synchronization patterns (MHS and EMS) and notifies the BIU of an incoming message or of message termination. It also scans incoming data for Manchester II encoding errors (ENER).

(2) During message output, the BTU appends MHS and EMS to outgoing messages and converts NRZ 17-bit serial binary data into biphase-encoded format.

The BTU, therefore, provides five signals to the remainder of the BIU during message reception. These are MHS detection, NRZ data, NRZ clock, EMS detection and ENER detection. The signal format upon conversion is shown in Fig. 6.

Message Reception Unit. The message reception module of the BIU receives the MH from the BTU, converts it to 16-bit parallel binary data in the Input Shift Register (ISR). The 17th data bit is used to test odd-parity. When the MH is assembled, it is immediately placed into the Input Data Register (IDR). The reception unit then determines if the message is for this PE. This is accomplished by using a 6-bit field of the MID, appending a leading 1Ø-bit binary constant to generate a PE memory address which is placed into the Input Address Register (IAR). This address is sent to the PE memory (on a DMA cycle-stealing basis) to access a 64-word area defined as the Message Identity Associative Match Map (MIAMM). The MIAMM word addressed is returned to the BIU. This word consists of 16-bits and

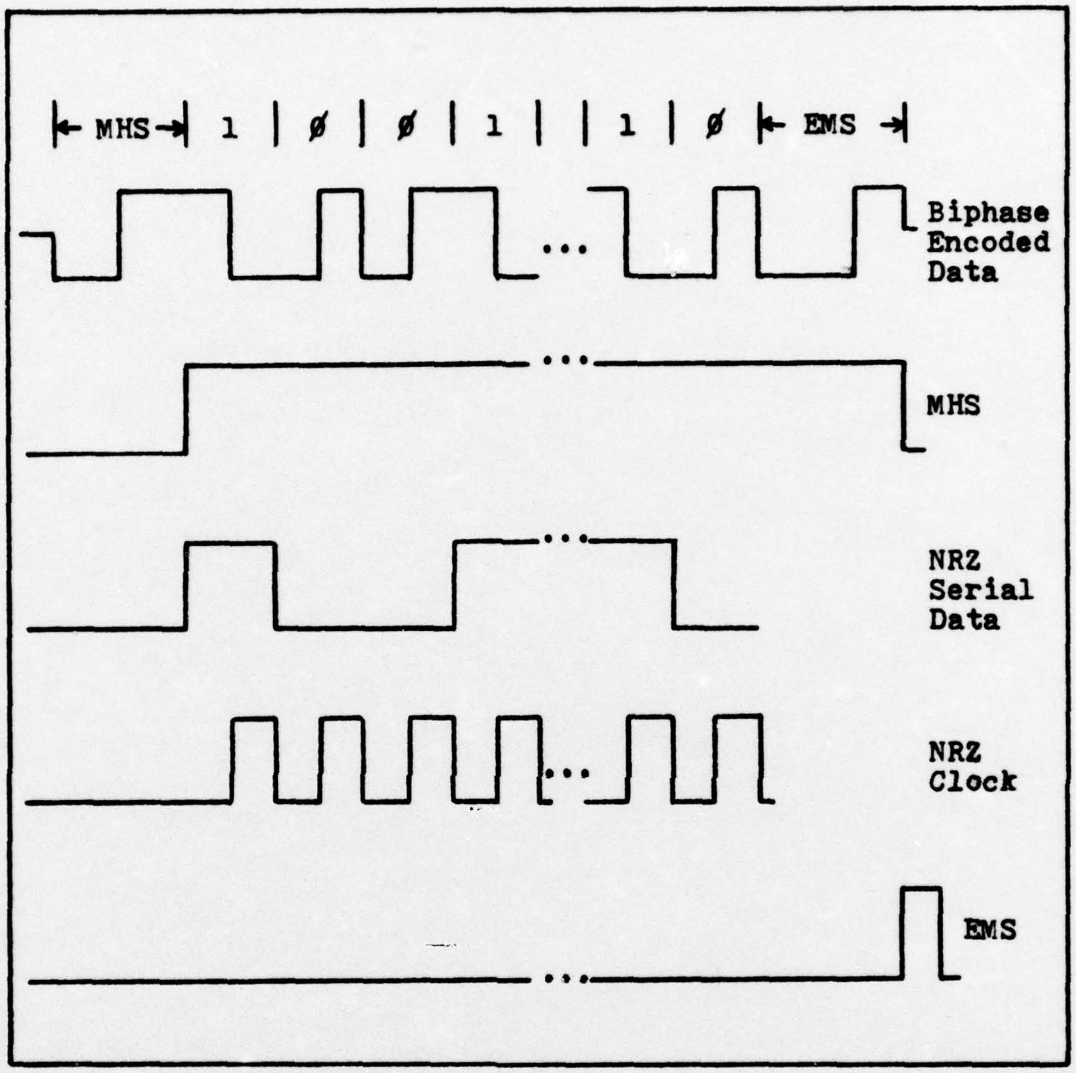


Fig. 6. Manchester II Biphase-Encoded Data Conversion

each bit is associated with a unique MH (64-words x 16-bits represent 1,024 unique messages). The appropriate response bit (RB) is determined by the binary value of the last four bits of the MID. The RB is tested and if "set", the message is to be received. If the RB is not "set", the message is not relevant to this PE and is discarded (Ref 1:64). Fig. 7 demonstrates this operation. The correct MIAMM word is

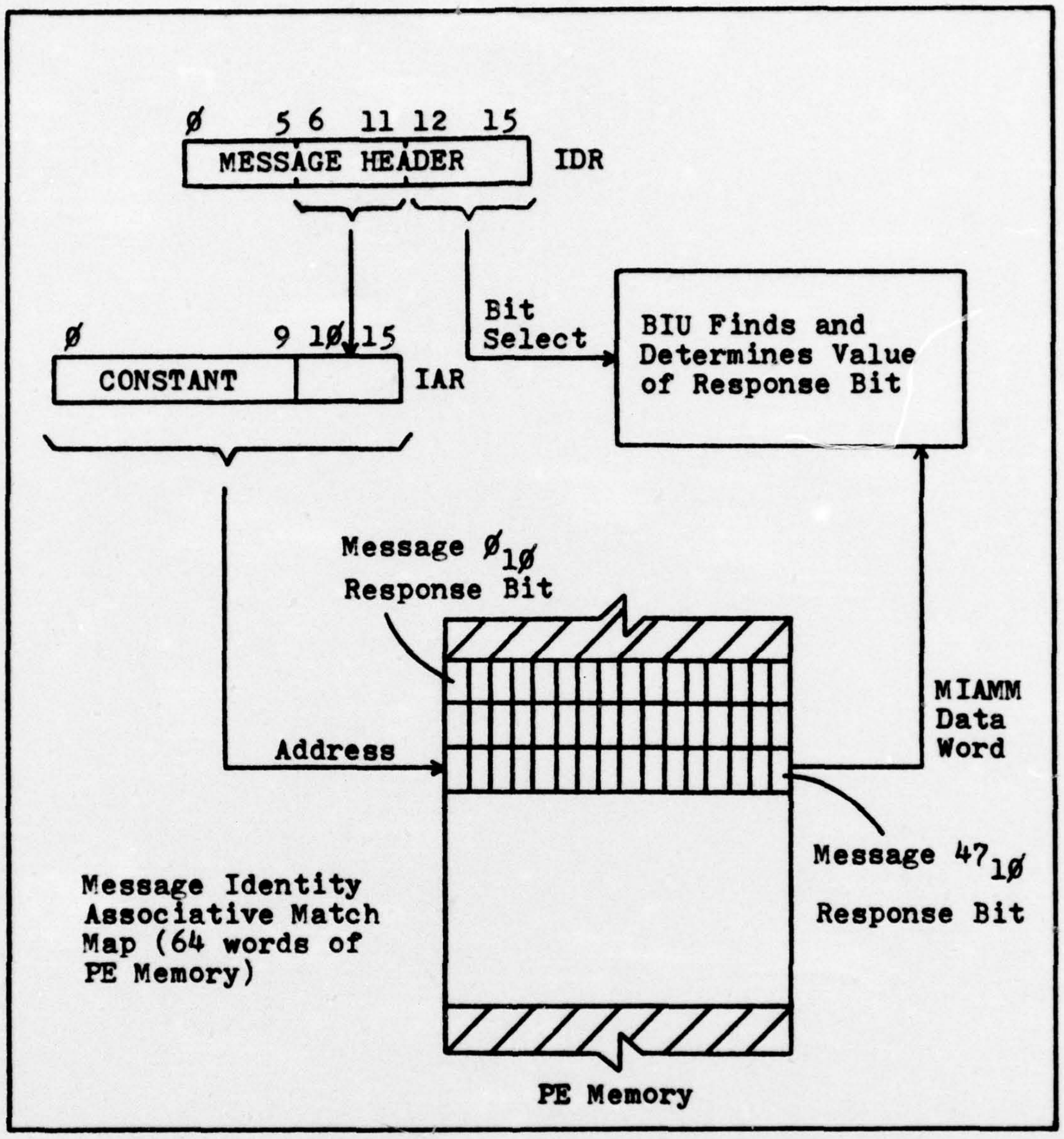

Ø 5 6 11 12 15
MESSAGE HEADER
IDR
Bit Select
BIU Finds and Determines Value of Response Bit
Ø 9 1Ø 15
CONSTANT
IAR
Message Ø10 Response Bit
MIAMM Data Word
Address
Message 4710 Response Bit
Message Identity Associative Match Map (64 words of PE Memory)
PE Memory

Fig. 7. Message Identity Associative Addressing

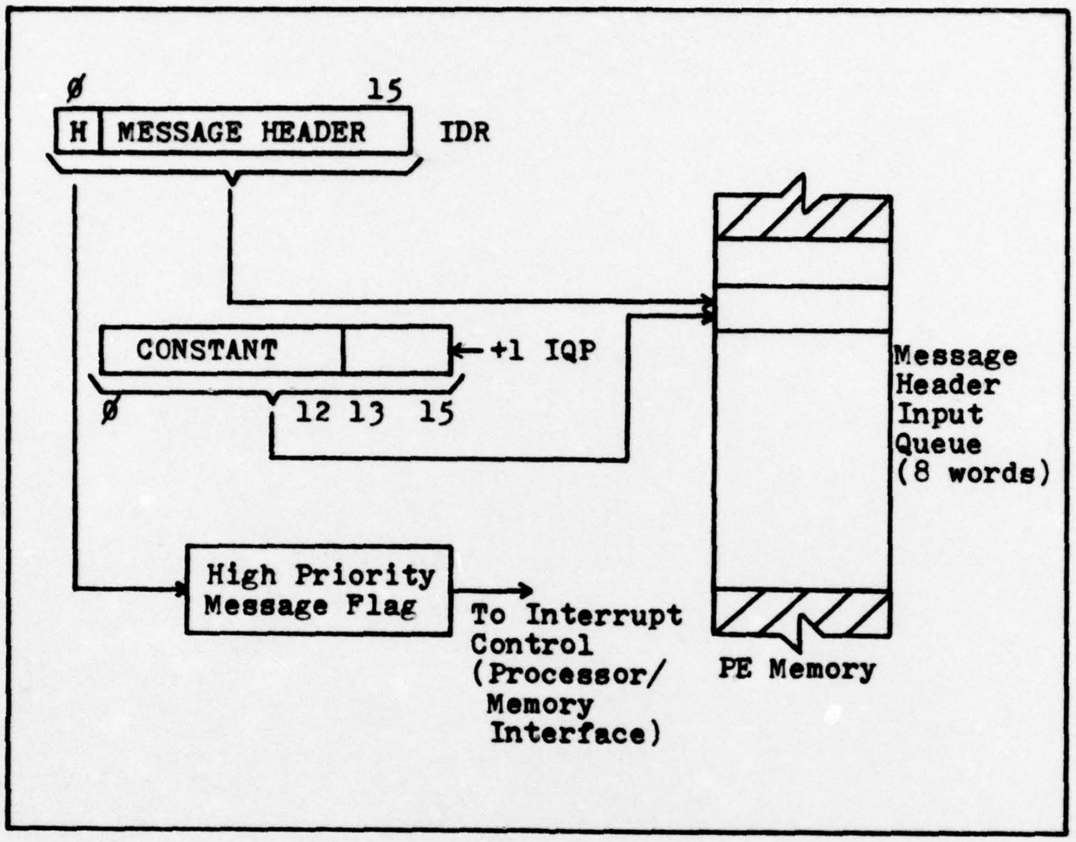
placed in PE memory during system implementation.

Message input (also referred to as message vectoring) is the procedure by which an input message, once recognized, is passed or "vectored" to an area in PE memory where it is stored for later use by an applications software process in the PE processor (Ref 1:52).

Because of the timing and nature of the message input from the global/local bus (continuous serial data input), the assembled data words are single-buffered (immediately transferred after assembly from the ISR to the IDR). This procedure allows the Message Reception Unit the time required to process the MH.

Following message recognition, the Message Reception Unit checks the H-field to determine if this is a high priority message and notifies the PE processor if it is. The BIU now places the MH into the Message Header Input Queue located in PE memory. This is a first-in-first-out modulo-8 queue. The proper address is obtained by using the Input Queue Pointer (IQP) which is a 13-bit constant appended with a 3-bit modulo-8 register (the IQP is incremented after each message input) to determine the queue address. This procedure is shown in Fig. 8.

The Message Reception Unit generates the indirect address of the message buffer space from the MID by appending the 10-bits of the MID to a 6-bit constant and placing the result in the IAR. This address is sent to PE memory (Memory Buffer Vector Space) to obtain the starting address of the Message Buffer Space (up to 1K addresses) where this particular message will be stored. This new address value is placed into the IAR which is then used to access the first word in the buffer. This first word contains the expected length of the incoming message (Fig. 9). This value is placed in the Input Length Register (ILR) and



Ø
15
H
MESSAGE HEADER
IDR
CONSTANT
+1 IQP
Ø
12 13 15
High Priority Message Flag
To Interrupt Control (Processor/ Memory Interface)
Message Header Input Queue (8 words)
PE Memory

Fig. 8. Message Header Input

is decremented after each data word is sent to PE memory. The IAR is now incremented, preparing the BIU for the first data word input.

At this point, the BIU is ready to transfer incoming data words and place them contiguously into PE memory, incrementing the IAR and decrementing the ILR as each word is stored.

Message Transmission Unit. Message transmission is similar to a conventional direct memory access (DMA) I/O channel. The output procedure is initiated by means of a Command Address Word (CAW) sent to the BIU which loads the

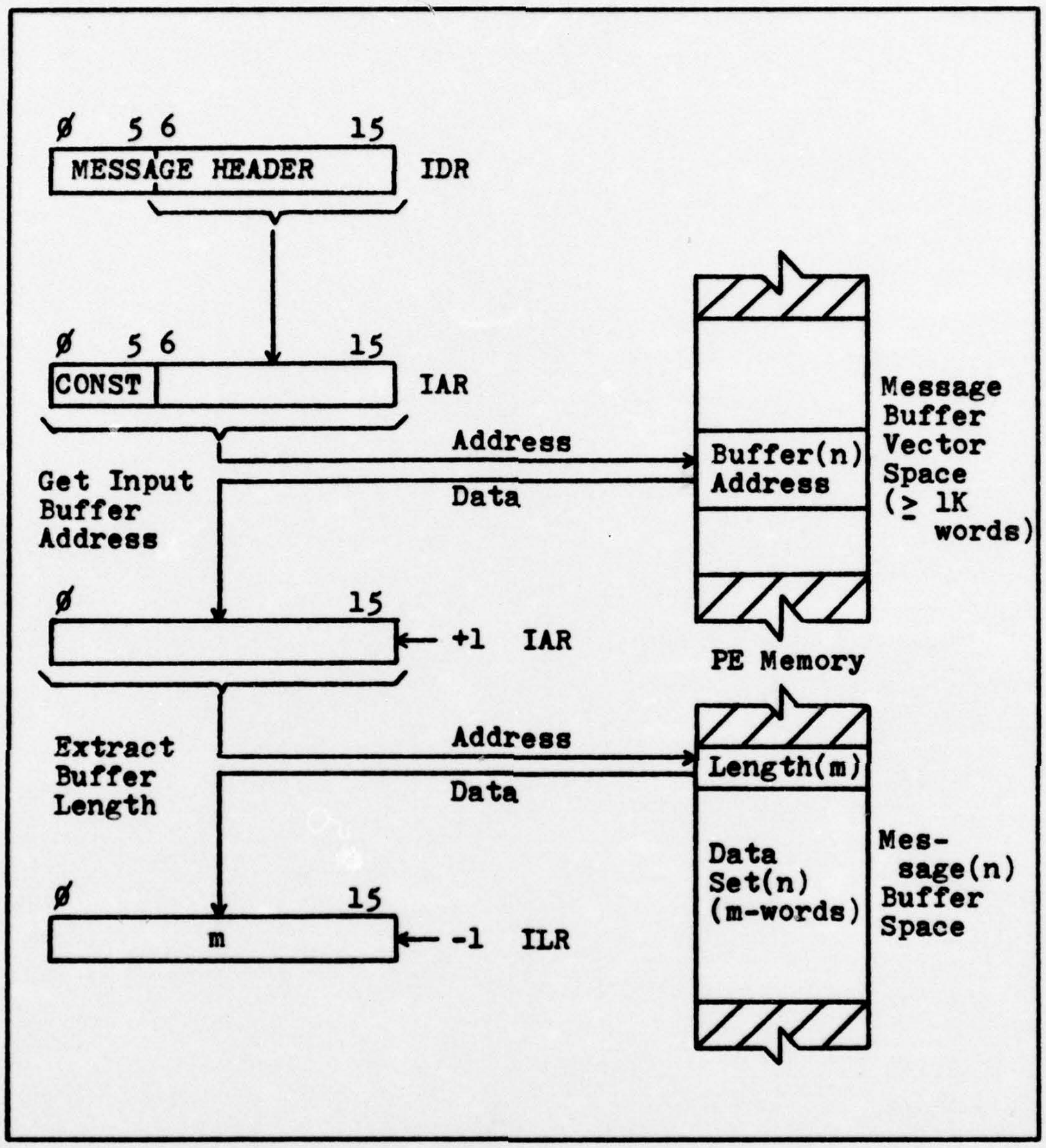


Fig. 9. Message Input Initialization

Output Address Register (OAR) with the beginning address of the block of data to be sent on the global/local bus. The first data word accessed is the message length which is placed into the Output Length Register and is decremented after each word is transmitted. The second word is the MH and each subsequent word is data. This operation is shown in Fig. 10. After each PE memory access, the OAR is incremented and the OLR is decremented.

The data output is single-word buffered to allow for continuous serial data transmission. Each word is placed in the Output Data Register (ODR). When the previous 16-bit data word is shifted out of the Output Shift Register (OSR), odd-parity is appended which provides a 17-bit serial data word to the BTU. The ODR is immediately transferred to the OSR and shifted out. The BIU obtains the next word from memory and places it in the ODR. The BTU must append the MHS and EMS to the beginning and end of the message (Ref 1:69).

Bus Access Control Unit. Global/local bus access for data transmission is based on a round-robin circular priority sequence. This allows each PE to control the bus when its allocation slot is detected. Bus access control is initiated by software Command Address Words (CAW) from the PE processor. A CAW tells the BIU to load two BIU 8-bit registers, the Bus Length Register and the Bus Position Register, with the values on the I-Bus. The Bus Length Register (BLR) defines the "length" of the bus in units of allocation time-slots to the PE's next access slot and the Bus Position Register (BPR) defines the current position of the PE's control slot in the bus control chain (Ref 1:69). The BPR is decremented after each EMS and when it reaches zero, it indicates that it is time for this PE to transmit.

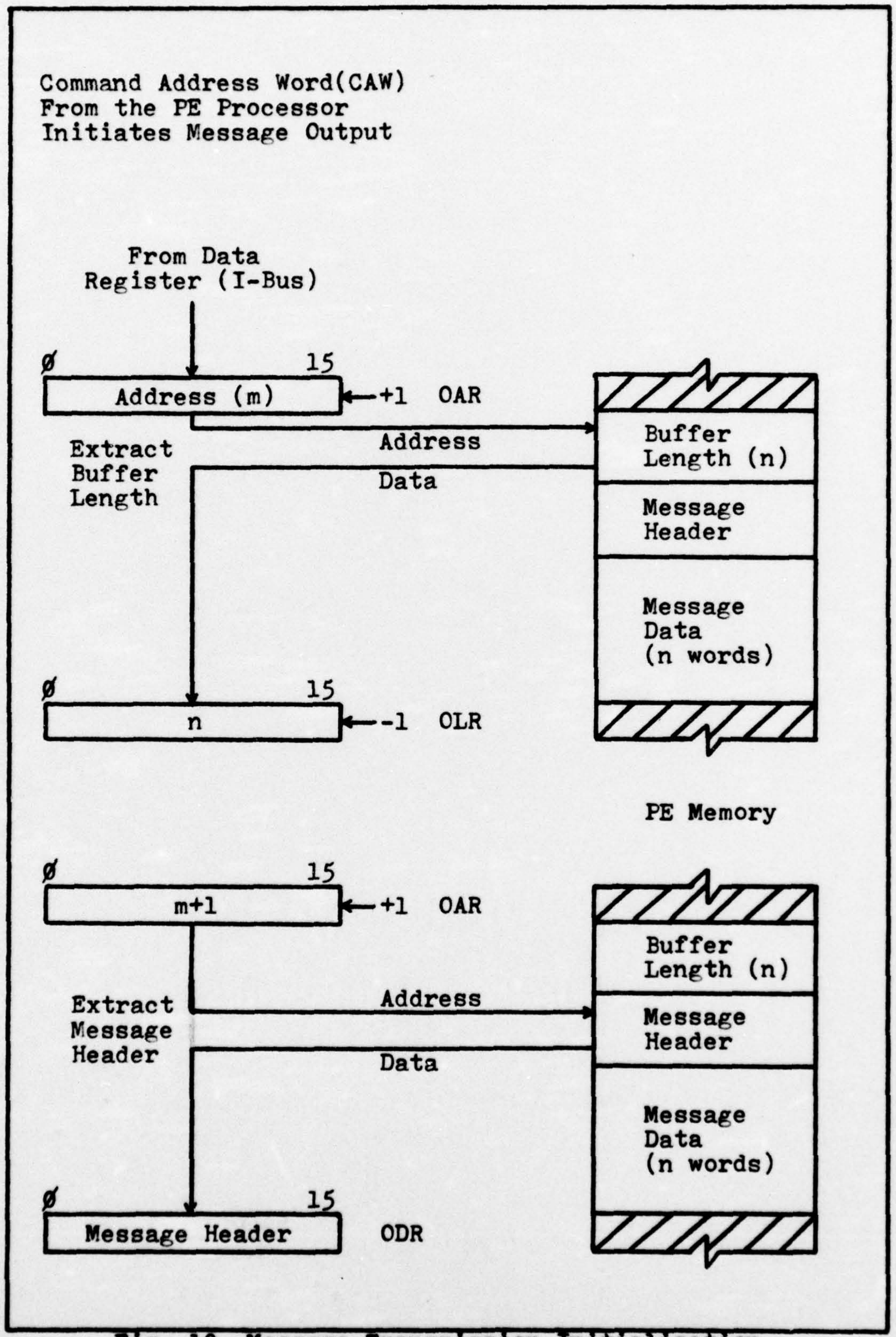


Fig. 10. Message Transmission Initialization

If the PE is not ready to transmit, it passes control by immediately issuing the EMS. The BPR is re-initialized with the value in the BLR and the whole procedure begins again.

With an 8-bit register, up to 256 bus allocation time-slots can be defined. This allows for some degree of super-commutation since most system applications will require less than 32 PEs/AGs (Ref 1:69).

Also included in this functional area, are two "watch-dog timer" mechanisms. The Bus Quiescence Watchdog Timer (BQWT) measures the time interval between bus activities. If a maximum "quiet-time" (no signal present) of 50 micro-seconds is exceeded, an interrupt stimulus is generated and bus access is disabled.

The Bus Dominance Watchdog Timer (BDWT) measures the length of each individual message transmission. If a PE attempts to transmit a message of more than eight words (normal, maximum message length), an interrupt stimulus is generated and bus access is disabled.

Processor/Memory Interface (for I-Bus). The four units of the PE are interfaced (Fig. 2) through the internal bus (I-Bus). All data and command communication between the PE processor, memory, IOIU, and BIU is accomplished via the I-Bus. This BIU module contains the logic required to recognize and decode CAW's from the PE processor and effect their execution. This unit also presents the BIU interrupt stimuli to the processor. Masking of each interrupt stimuli and compliance with PE processor generated interrupts is also accomplished here. Table II provides a list of the Command

Table II

Command Address Words

PE Processor Output (Write to BIU)	PE Processor Input (Read from BIU)
Load Global Bus Control Parameters (Bus Length and Bus Position)	Send Present Global Bus Position (GBPR)
Load Local Bus Control Parameters (Bus Length and Bus Position)	Send Present Local Bus Position (LBPR)
Activate Global Output (GOA)	Send Global Input Queue Pointer (GIQP)
Activate Local Output (LOA)	Send Local Input Queue Pointer (LIQP)
Load Global Interrupt Status (GISR)	Send Global High Priority Interrupts (GIFR(ØØ-Ø8))
Load Local Interrupt Status (LISR)	Send Local High Priority Interrupts (LIFR(ØØ-Ø8))
Load Global Primary Interrupt Masks (PL3/4M)	Send Global Low Priority Interrupts (GIFR(14-15))
Load Local Primary Interrupt Masks (PL5/6M)	Send Local Low Priority Interrupts (LIFR(14-15))
Enable Global Output Activity (GOEN)	Send Global Interrupt Status (GISR)
Enable Local Output Activity (LOEN)	Send Local Interrupt Status (LISR)
Disable Global Output Activity	Send Global Primary Interrupt Masks (PL3/4M)
Disable Local Output Activity	Send Local Primary Interrupt Masks (PL5/6M)
Switch Global Output to Alternate Bus (OSWCMD)	

Address Words (CAW) sent to the BIU from the PE processor (Appendix B, Table B-6 shows how the CAW's generated by the PE processor will be interperated by the BIU). Table III provides a list of the global/local interrupts generated by the BIU and sent to the PE processor.

<u>Redundant Bus Management Unit</u>. This functional area provides the interface control between the BIU and the dual-redundant global buses. This unit monitors the alternate or back-up bus and is waiting for a valid predetermined software "switch-input-to-alternate-bus" command for all PEs from the GEX. When this command occurs the BTU input is transferred to the alternate bus and all future message receptions are received on the newly established bus connection. This action automatically forces termination of any message input in progress and the GEX notities the PE processor of this action. The BIU output is under PE processor control and is switched by means of the appropriate CAW after GEX fault recovery procedures have been completed. This switching activity can be performed as many times as directed by the GEX. Thus, "primary" and "alternate" global bus identities can be interchanged as required (Ref 1:72). This procedure is used to remove a PE that has gone out of control from the "primary" bus.

<u>Fault Tolerance/Detection</u>

The distributed, decentralized network requires that each PE monitor all data communications between system PEs and alert the processor upon detection of data errors or

Table III

BIU Interrupts

G/LIFR Location	Global/Local High Priority	G/LIFR Location	Global/Local Low Priority
ØØ	High Priority Message Received (HP)	14	Message Received (MR)
Ø1	Bus Quiescence (BQ)	15	Message Output Complete (MOC)
Ø2	Bus Dominance (BD)		
Ø3	Output Message Length Violation (OMLV)		
Ø4	Message Header Error (MHE)		
Ø5	Message Word Count Error (MWCE)		
Ø6	Message Word Length Error (MWLE)		
Ø7	Message Data Parity Error (MDPE)		
Ø8	Message Data Encoding Error (MDEE)		

faults. This approach requires the existence of certain fault detection mechanisms within the BIU to facilitate system software recovery.

Bus Message/Data Errors. The biphase-encoded asynchronous data transmission procedure may cause such errors as: (1) improperly encoded data representations, (2) missing or added data bit, or (3) data content error (bit value error). The BTU must recognize these errors to ensure valid data input and notify the BIU of any such errors (Ref 1:55).

Biphase-encoding patterns received in the BTU are compared against legal Manchester II format. This includes data and synchronization signals (MHS and EMS).

Information content validity is determined by using an odd-parity check. When each 16-bit data word is transmitted, odd-parity is calculated and a 17th bit is added to each word.

Data content is checked upon reception for a modulo-17 length. This is a bit count to ensure that each word is an integral number of 17-bits.

Bus data errors of the types described above fall into two categories, message header errors and data errors. If a header error is detected, the message input is completely discarded. If an error is detected in a data word, the normal input sequence is continued and the PE processor is notified of the error. The decision to use or not use the data is left to the discretion of the applications program or Local Executive (LEX).

All data error detections in the BIU are under program control. These errors are recognized only when the processor has properly set an Interrupt Status Register (ISR) where each bit is associated with a particular message error (See Table III).

Message Protocol Errors. During message header reception, the BIU determines the length of the incoming message. If a discrepancy between the expected length and the actual length is detected, the message is terminated at the shorter length and the processor is notified.

During message transmission, maximum data word length is limited to eight words. Thus, any attempt by the PE to transmit a longer message (i.e., OLR value greater than eight), transmission is immediately aborted and the processor notified.

Bus Control Errors. Bus control errors consist of two basic forms: (1) bus quiescence (no activity) and (2) bus dominance (incessant activity). Bus quiescence can occur as a result of a faulty PE (functional fault) or a broken wire (physical discontinuity). These two faults are monitored by the BQWT and BDWT. If quiescence is detected, the system is forced into a disabled state to allow recovery activity to begin under a system software monitor (GEX PE). This period of quiescence is predetermined at system design (approximately 50 microseconds).

Bus dominance is usually the fault of a BIU output interface section that has gone out of control. The BDWT monitors messages for word count errors and if detected forces the PE into a known disabled state so that software recovery procedures can begin.

However, a PE which tries to access the bus in an unauthorized time-slot is harder to detect. This type of error is detected indirectly through parity errors, biphase-encoding errors, message length errors, and message synchronization errors. After a threshold of such errors is reached, the system can be forced into a known state to allow the GEX PE to identify the faulty PE and initiate recovery procedures.

Summary

This completes the functional description of the BIU (along with Appendices A and B). Using this detailed description, the next step is to define the hardware and microword required to implement the BIU functions. When this step is completed, the BIU microroutines must be coded.

III. BIU Design

Introduction

This study has redefined the BIU functional modules so that the implementation can be accomplished using a microprocessor to emulate the BIU hardware. This design has taken the approach that both global and local message transmissions/receptions can be properly handled by a common set of microprograms using one microprocessor (The TI design envisioned two hardware sections, one for global message management and one for local message management. Both consist of the same hardware modules except that the Redundant Bus Management module is not needed in the local message management section.).

The new modules are: (1) Bus Translation Unit (designed by AFAL), (2) Message Reception Routine, (3) I-Bus Access Routine, (4) Message Transmission Routine, (5) Global/Local Bus Access Routine (including Bus Quiescence and Bus Dominance), (6) Interrupt Routine, (7) Command Address Word Routine (including Clear/Reset), and (8) Global Redundant Bus Management Routine. Chapter I provided a list of functional hardware modules and it is very similar to the above list. Items three, six, and seven when combined; essentially perform the same functions as the Processor/ Memory Interface unit. Thus, even though the new system description is slanted toward software implementation, the same modular structure is maintained.

The high level block diagram and flow charts provided by the TI feasibility study and the AFAL preliminary design (Ref 2) were studied and a very detailed register transfer language (RTL) description was developed for each routine. Although this description is slanted toward software implementation, the specific microprocessor to be used was not looked at until this detailed RTL was complete. Thus, this description defines only the responsibilities of the BIU and does not include those of the microprocessor. This is in keeping with the top-down approach to design.

The results of this design effort are shown in Appendix B, Tables B-1 through B-7. Appendix A is a table of the mnemonics and their definitions which are used throughout the RTL description.

The RTL descriptions are organized in the following sequence: (1) Table B-1, Message Reception Routine, (2) Table B-2, I-Bus Access Routine, (3) Table B-3, Message Transmission Routine, (4) Table B-4, Global/Local Bus Access Routine, (5) Table B-5, Interrupt Routine, (6) Table B-6, Command Address Routine, and (7) Table B-7, Global Redundant Bus Management Routine. These routines were designed incorporating the changes recommended in the AFAL preliminary design (Ref 2) and changes which have been made since this study began.

AM2900 General Description

The AM2900 family is a new group of large scale integrated (LSI) circuits produced by Advanced Microdevices, Incorporated. These LSI, three-state, integrated circuits (IC) are extremely fast because they utilize Low-Power Schottky TTL technology. The concept used in developing the AM2900 series provides a basic group of "building blocks" rather than a fixed-instruction microprocessor. The fixed-instruction microprocessor may be difficult to use in some applications where the AM2900 family is very adaptable. The designer can choose the "building blocks" he needs to design his particular system and is not forced to adapt his system to a fixed-instruction machine.

The AM2900 family has been divided into eight specialty functions (or building blocks): (1) data manipulation, (2) microprogram control, (3) macroprogram control, (4) priority interrupt, (5) direct memory access, (6) I/O control, (7) memory control, and (8) front panel control (Ref 3:2). Almost any specialized control or computational application can be implemented (and with greater speed than fixed-instruction microprocessors) using various combinations of these speciality functions.

Since this study is a design using a new product, many of the ICs are not on the market at this time. The information available for many of the ICs is limited and a thorough analysis of their characteristics is difficult. With the data that is available, this design will provide a system configuration that can be used as a laboratory test model.

BIU Functional Description

The functional block diagram of the BIU using the AM2900 family is shown in Fig. 11. This diagram shows that the BIU is driven by interrupt stimuli which select the beginning address of the appropriate microroutine rather than by macroinstructions which is the conventional techinque.

When an interrupt is sent to the Vectored Priority Interrupt (VPI) logic, a 6-bit vector is generated along with an interrupt request (INTRQ). When the interrupt is recognized by the currently executing microinstruction, the Microsequencer Control (MSC) logic gates the starting address (generated from the 6-bit vector sent to the Mapping PROM) of the microroutine which services the interrupt to the Microprogram PROM (MPP) which outputs the first microinstruction to the Microword Register (MWR).

The Microsequencer (MS) has four address sources: (1) the direct inputs (from MWR), (2) the register inputs (from MPP), (3) a four-word push/pop stack (in the MS), and (4) the microprogram counter (MPC) also located in the MS. These address sources are controlled by the MSC logic which in turn, is controlled by the Arithmetic Logic Unit (ALU) status, INTRQ, and the currently executing microinstruction. Various test conditions are monitored by the MSC which determines the next microinstruction address based on the results of the test.

Associated with the BIU Processor (BIUP) is the Global/Local Input and Global/Local Output which interface with

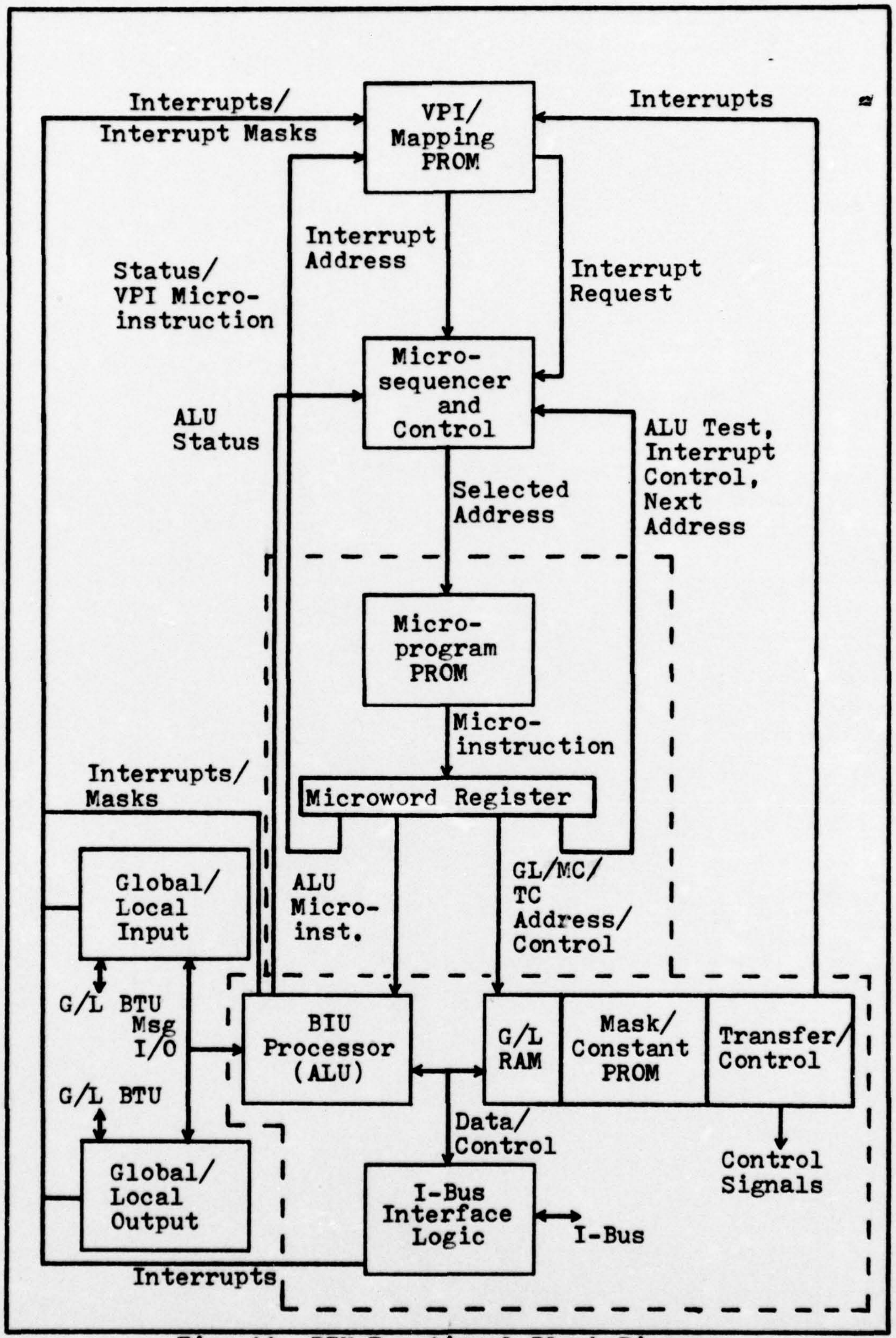


Fig. 11. BIU Functional Block Diagram

the global buses (through the BTU). Also, the I-Bus Interface Registers are considered part of the BIUP functional area. The Global/Local RAM (GLRAM) is used to store values for the various registers associated with message input/output (e.g., ILR, BPR, BLR, OAR). The Mask/Constant (MC) PROM contains the various masks needed to control and initialize the VPI. This PROM also contains the constants used to set message input/output interrupt stimuli and notify the PE processor of the BIU status. The Transfer/Control (TC) logic provides the signals to gate information from/to registers within the BIU, to set VPI interrupts, and to control various functions throughout the BIU.

Detailed System Description

The detailed system description will begin with the MPP and MWR then continue with the BIUP (including the GLRAM, MC, and TC), the VPI, the MS and MSC, and finally the Global/Local Input. The mnemonics, coding, memory registers, and control signals associated with the microword fields, GLRAM, MC, and TC are defined in Appendix C.

Microprogram PROM and Microword

Microprogram PROM. The MPP presently consists of 14/AM29761 PROM ICs. Each IC contains 256 words of 4-bits each. The configuration used provides a 256 word by 56-bit memory. The MPP is accessed by the MS which provides a 12-bit address. Presently only 8-bits are used to allow system growth and modification. The MPP provides a 56-bit, three-state output which is controlled by the Output Microprogram PROM (OMPP) signal from the TC control logic. A detailed description of both the hardware and software aspects of the MPP and MWR is shown in Fig. 12.

Microword Register. The MWR consists of 13/AM2918 4-bit registers and one AM25LS161 binary counter (modified and used as a register with a 4-bit input (DØ-D3) and two 4-bit outputs, standard (QØ-Q3) and three-state (YØ-Y3). Information is transferred into the MWR on the low-to-high clock transition. The data in the AM25LS161 is immediately available at the Q-outputs and is available at the Y-outputs when Output Word Register (OWR) from the TC field is active. This IC therefore, allows both synchronous and asynchronous execution.

The AM25LS161 is provided for system initialization. This IC is a 4-bit asynchronous counter which has been configured to force the MS to a known MPP address during system initialization. When power is applied to the system, the power-on logic (not designed) will issue a Reset (RSET) signal which forces the output of the AM25LS161 to all zeros. This address is sent to the MS PROM (in MSC) which in turn forces the address output of the MS to all zeros.

The microword contains ten field which consist of 54-bits. Two bits, D24 and D55 are not used. The D24-bit was originally used as an ALU test enable signal for the MSC but was found to be a duplication of the T/I control signal. Bit D55 was never defined and is present because

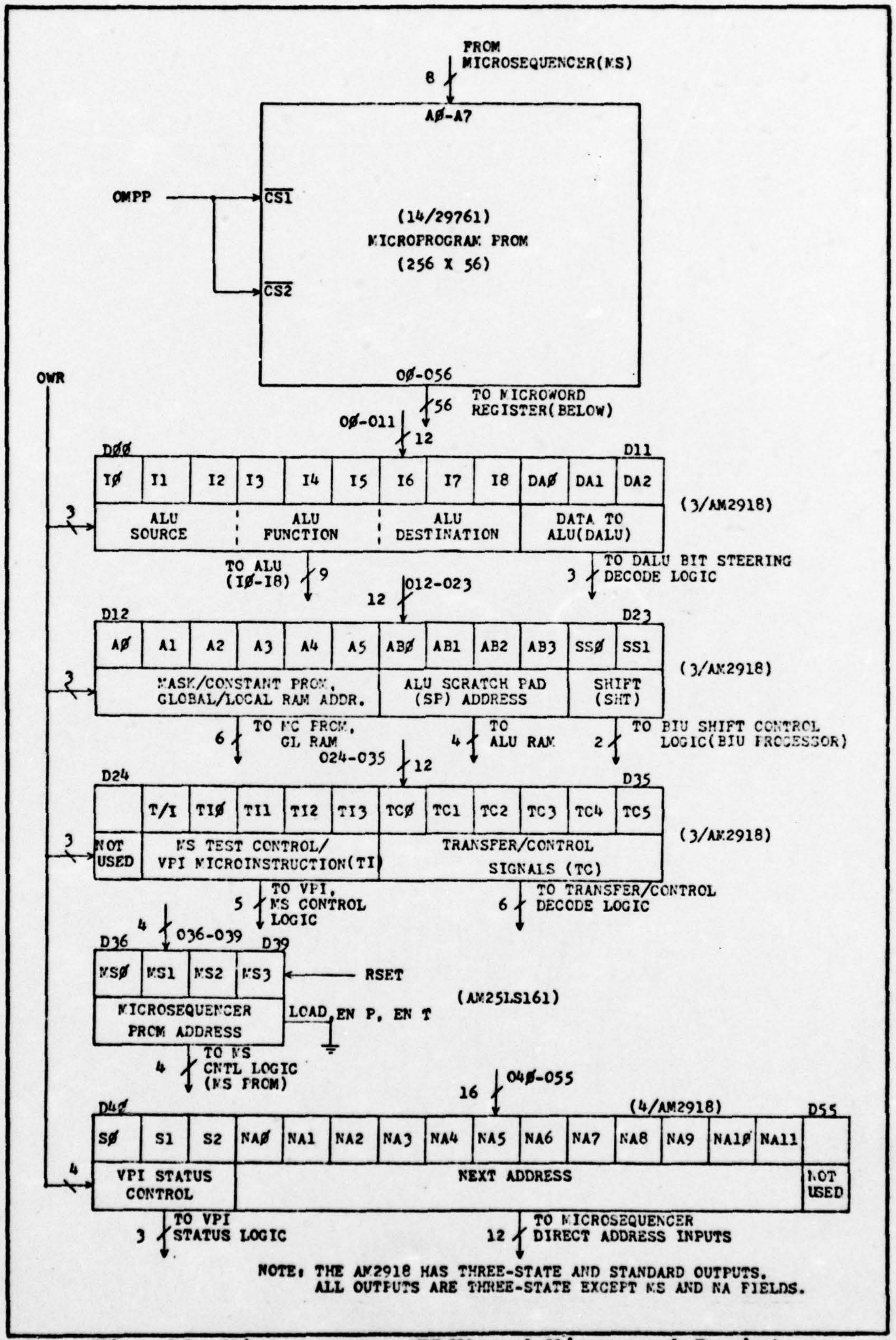

FROM MICROSEQUENCER(MS)
8
AØ-A7
OMPP
CS1
CS2
(14/29761) MICROPROGRAM PROM (256 X 56)
OØ-O56
TO MICROWORD REGISTER(BELOW)
56
OWR
OØ-O11
12
DØØ
D11
IØ I1 I2 I3 I4 I5 I6 I7 I8 DAØ DA1 DA2
ALU SOURCE
ALU FUNCTION
ALU DESTINATION
DATA TO ALU(DALU)
(3/AM2918)
TO ALU (IØ-I8) 9
3 TO DALU BIT STEERING DECODE LOGIC
12 O12-O23
D12
D23
AØ A1 A2 A3 A4 A5 ABØ AB1 AB2 AB3 SSØ SS1
MASK/CONSTANT PROM, GLOBAL/LOCAL RAM ADDR.
ALU SCRATCH PAD (SP) ADDRESS
SHIFT (SHT)
(3/AM2918)
6 TO MC PROM, GL RAM
4 TO ALU RAM
2 TO BIU SHIFT CONTROL LOGIC(BIU PROCESSOR)
O24-O35 12
D24
D35
T/I TIØ TI1 TI2 TI3 TCØ TC1 TC2 TC3 TC4 TC5
NOT USED
MS TEST CONTROL/ VPI MICROINSTRUCTION(TI)
TRANSFER/CONTROL SIGNALS (TC)
(3/AM2918)
5 TO VPI, MS CONTROL LOGIC
6 TO TRANSFER/CONTROL DECODE LOGIC
4 O36-O39
D36
D39
MSØ MS1 MS2 MS3
RSET
MICROSEQUENCER PROM ADDRESS
LOAD, EN P, EN T
(AM25LS161)
4 TO MS CNTL LOGIC (MS PROM)
16 O4Ø-O55
D4Ø
(4/AM2918)
D55
SØ S1 S2 NAØ NA1 NA2 NA3 NA4 NA5 NA6 NA7 NA8 NA9 NA1Ø NA11
VPI STATUS CONTROL
NEXT ADDRESS
NOT USED
3 TO VPI STATUS LOGIC
12 TO MICROSEQUENCER DIRECT ADDRESS INPUTS
NOTE: THE AM2918 HAS THREE-STATE AND STANDARD OUTPUTS. ALL OUTPUTS ARE THREE-STATE EXCEPT MS AND NA FIELDS.
3
3
3
4

Fig. 12. Microprogram PROM and Microword Register

a multiple of 4-bit registers were required. These bits can be defined if needed during system implementation. The ALU field (I∅-I8) is the first field in the MWR and provides the control signals for the AM2901 microprocessor (See Table C-1).

The second field, Data-to-ALU (DALU), is an encoded bit steering field which controls an AM25LS138 decoder (See Fig. 13) whose outputs are used to control addressing of the GLRAM, MC, the A and B ports of the AM2901 16-word RAM (Scratch Pad, SP), and loading of the Shift Counter. See Table C-2 for the DALU control signals.

The A field (A∅-A5) and AB field (AB∅-AB3) provide the addresses which are "steered" by the DALU field. The A field is the address for the GLRAM and MC. The AB field provides the address for the ALU SP input/output ports and the shift value for the Shift Counter.

The Shift Select (SS) field (SS∅-SS1) and I7 from the ALU field determine the type of shift operation the ALU will perform (See Table C-3).

The Test/Instruction (TI) field (ALU test and VPI microinstruction field) consists of T/I, bit steering control, and TI∅-TI3. When T/I is inactive, TI∅-TI3 becomes the microinstruction for the VPI. When T/I is active, the field determines the test to be performed by the MSC.

The Transfer/Control (TC) field provides the transfer and control signals that are used throughout the BIU. This field controls four 54LS154 decoders which provide 64 control lines (See Fig. 14). Each control line provides one or a combination of several control signals. The signals which have been defined are shown in Table C-4. How these signals are used and combined is described in Chapter IV.

The MS field, mentioned earlier, provides the address for the MS PROM which in turn provides the control signals for the MS.

The Status (S) field (SØ-S2) controls the status inputs to the VPI status encoder. This allows program control of the eight interrupts within each of eight levels of interrupts.

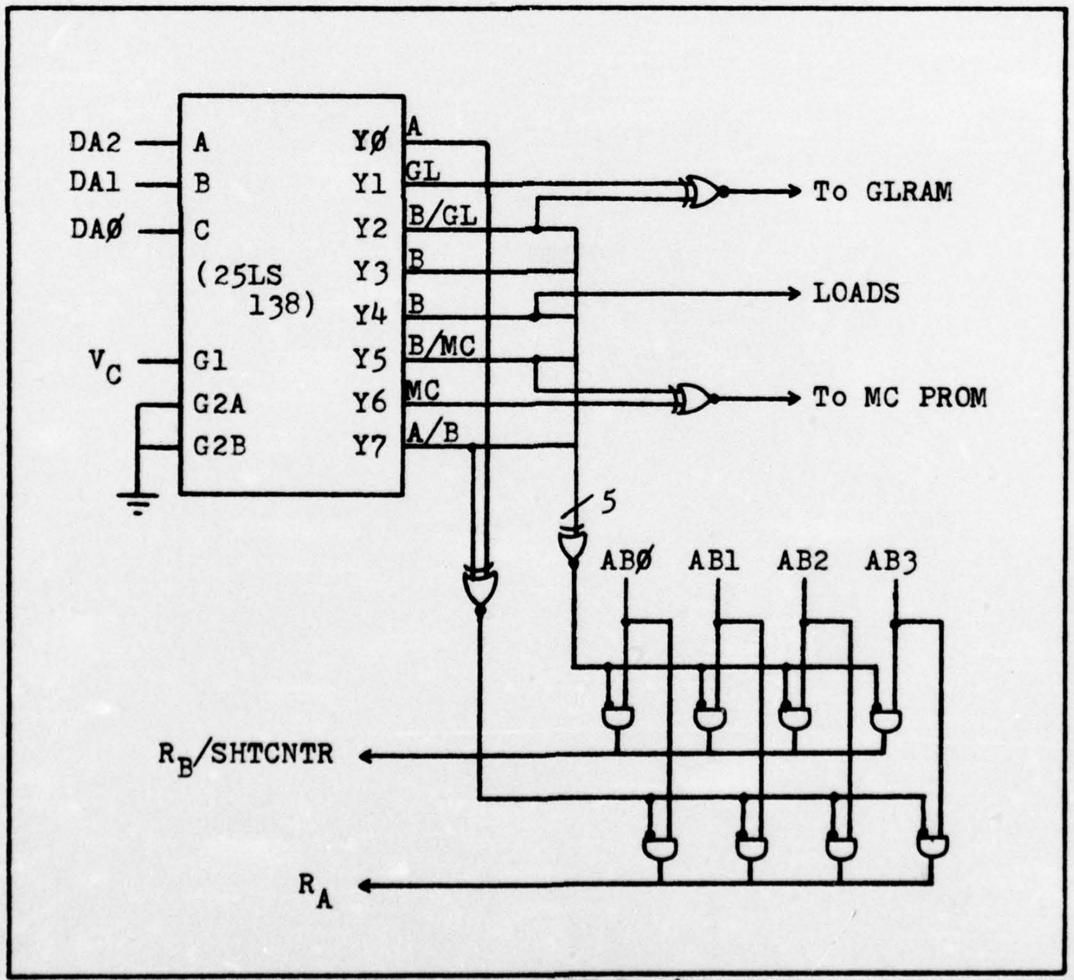


Fig. 13. DALU Logic

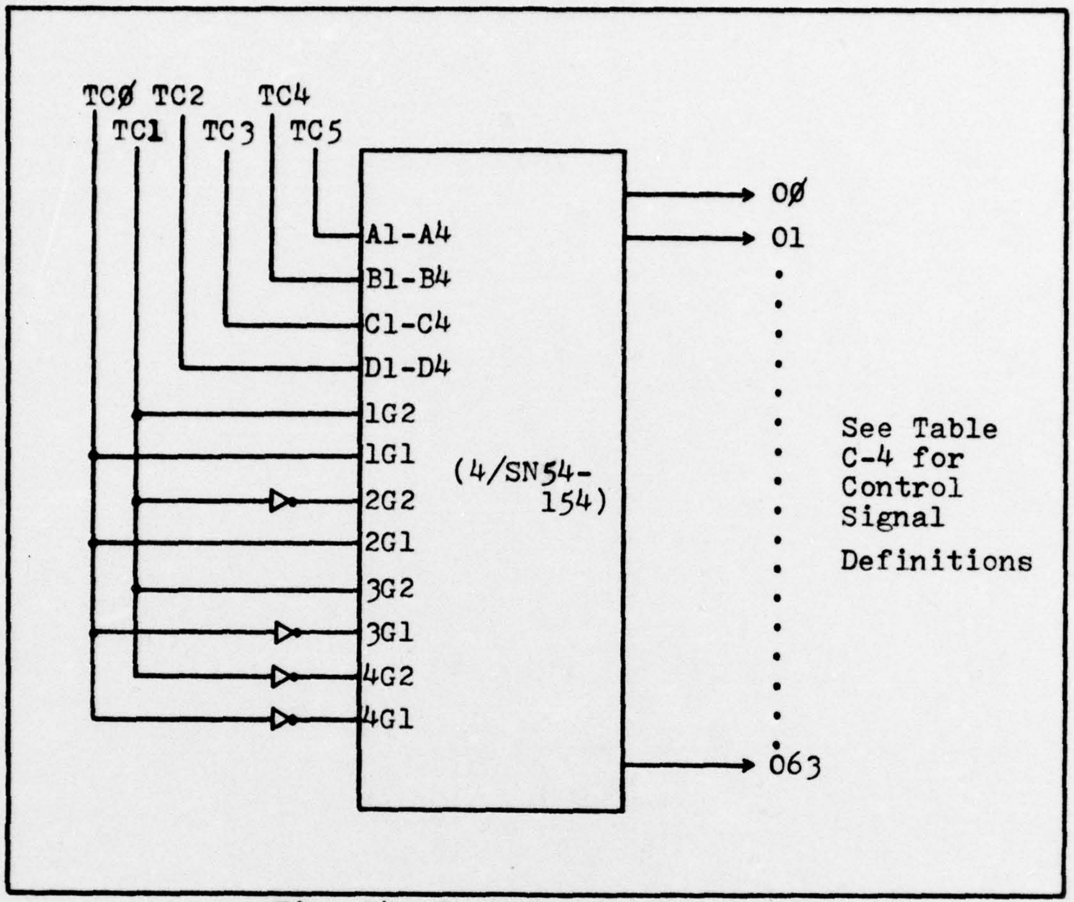


Fig. 14. Transfer Control Logic

The Next Address (NA) field provides the address for the MS direct inputs. This field allows program controlled branch operations based upon the results of MSC test results.

BIU Processor

The BIUP consists of the AM2901, AM2902, and the AM25LS253. The BIUP configuration is shown in Fig. 15. The AM2901 is a 4-bit slice and contains a two-port RAM, a high-speed ALU, a Q-register (accumulator) and associated shifting, instruction decoding, and multiplexing logic. The AM2901 is controlled by a 9-bit microinstruction field

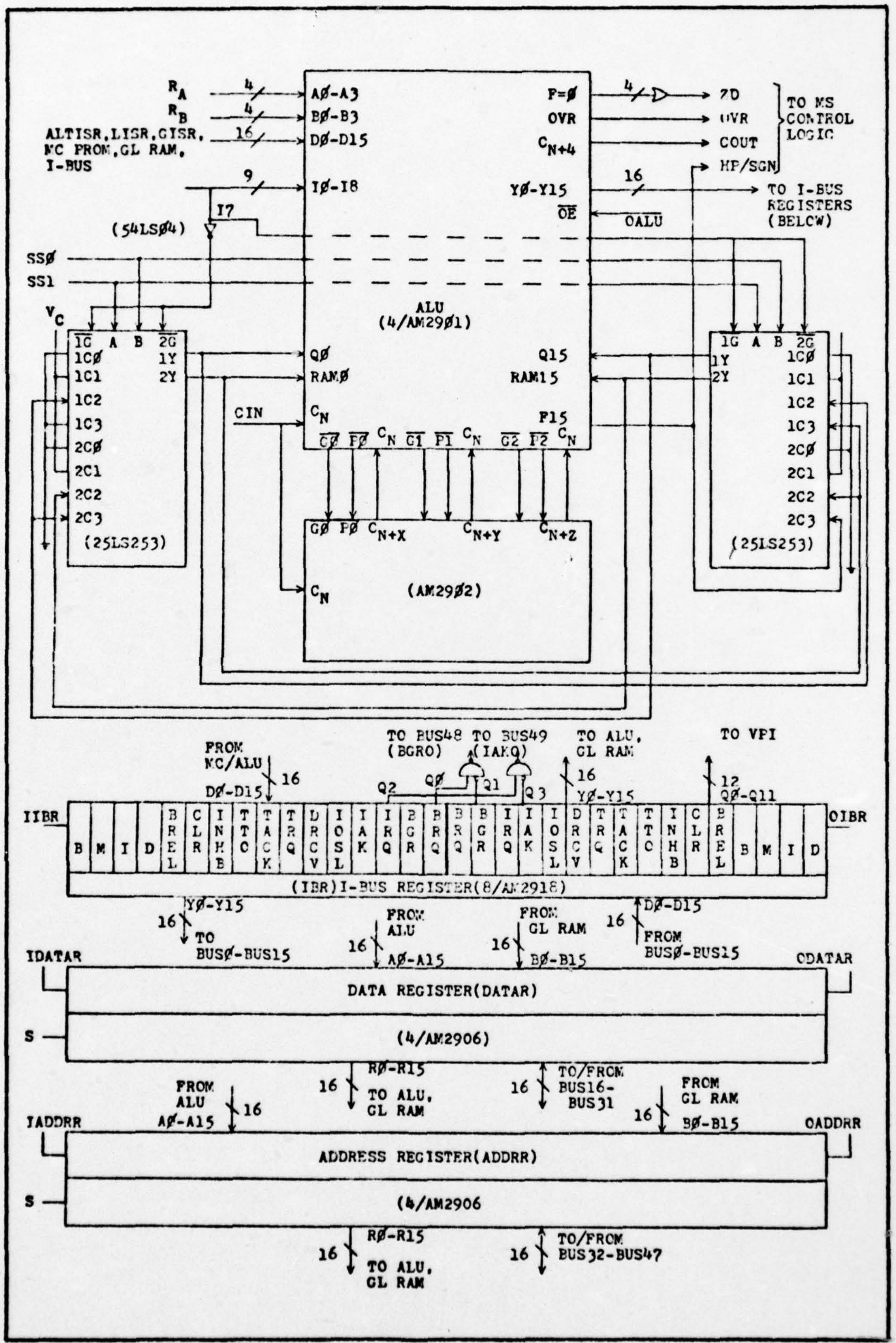


Fig. 15. BIU Processor and I-Bus Interface Logic

(IØ-18) which is divided into three groups of 3-bits each. These three groups select the ALU source operand, ALU function and ALU destination (Ref 3:5). These three functions were defined in Table C-1. Four AM2901 microprocessors are used to provide a 16-bit word. This word is defined with bit ØØ as the least significant bit and bit 15 as the most significant bit.

The 16-word RAM (SP) can be read from the A-latch or B-latch and is controlled by the R_A and R_B inputs. These inputs originate in the AB field of the MWR and are "steered" to the ALU by the DALU field. Data is written into the SP only with the address provided at the R_B inputs.

The ALU can perform three binary arithmetic and five logic operations with the two data input words, R and S, which are determined by the ALU source field. The R and S inputs come from four sources: (1) A-latch, (2) B-latch, (3) Q-register, or (4) direct inputs (DØ-D15). The ALU has carry propagate ($\overline{P}$) and carry generate ($\overline{G}$) outputs to provide a look-ahead carry configuration. There are four ALU status outputs: (1) F=Ø (zero detect, ZD), (2) OVR (overflow), (3) F3 (ALU sign bit, HP/SGN), and (4) C_{N+4} (carry out, COUT). The CN (CIN) input of the least significant AM2901 can be used to increment a source operand by one. The RAMØ, RAM3, QØ, and Q3 data lines are the shift I/O lines for cascaded AM2901 microprocessors.

The AM2902, High-Speed Look-Ahead Carry Generator, is used to provide faster ALU operation. One AM2902 can accomodate three pairs of $\overline{G}$, $\overline{P}$, and C_N to provide a 16-bit word length.

Two AM25LS253, Dual-Four-Input Multiplexers, are used to produce four shift operations which have been defined in Table C-3. The two multiplexers are controlled by I7 which defines a left or right shift operation and the SS field which defines the type of shift operation.

The ALU three-state outputs (YØ-Y15), controlled by Output ALU (OALU) are connected to three 16-bit registers (IBR, DATAR, and ADDRR) which provide the I-Bus interface (BUSØ-BUS49).

<u>I-Bus Interface</u>. The Data Register (DATAR) and Address Register (ADDRR) consist of eight AM2906 Quad two-input Bus Transceivers. The A-word input is from the ALU and the B-word input is from the GLRAM. The R output goes to the ALU and GLRAM inputs. The BUS I/O lines form the I-Bus. Data/address information from the bus is latched by IDATAR and IADDRR control signals from the TC field. Data/address information to the bus is controlled by ODATAR/OADDRR. The A or B source for the bus is selected by the S control signal.

The I-Bus Register (IBAR) consists of eight AM2918 Quad D-registers with standard and three-state outputs. The left half of the IBR places information onto the bus while the right half receives information from the bus. The three-state Y-outputs to and from the bus are controlled by OIBR and IIBR (from TC field). The IBR receives information from the ALU (left DØ-D15) and sends information to the ALU (right YØ-Y15). The right standard outputs to the BIU (QØ-Q11) provide interrupt stimuli to the VPI.

Bus Grant (BGR) and Interrupt Acknowledge (IAK) are threaded through each I-Bus device from highest priority device to lowest priority device. This procedure provides a priority acknowledge system with the BIU having highest priority. When the BIU issues a Bus Request (BRQ) or Interrupt Request (IRQ), BGRO or IAKO is blocked from other I-Bus devices allowing the BIU to obtain control of the I-Bus.

Memory. The BIUP requires a large number of registers and constant values to process message inputs and outputs. These registers and constants are located in the GLRAM, SP, and MC.

The GLRAM is shown in Fig. 16. This memory consists of 16 AM29701 64-bit RAMS and one AM25LS139 Dual One-of-Four Decoder. This RAM provides a 16-word by 4-bit high speed memory and is configured into four groups of four to provide 64 words of 16-bits each. The DALU field and A field provide access to the RAM. The DALU field activates the AM25LS139 then AØ and A1 provide the coded inputs. The decoded output activates one of the four memory groups and A2 through A5 provide the address for that group. To write into the GLRAM, the TC field provides a Write Global/Local (WGL) signal to activate the memory write cycle to place data from the D-inputs into memory.

The registers which have been defined are shown in Table C-5. The organization presented provides for the

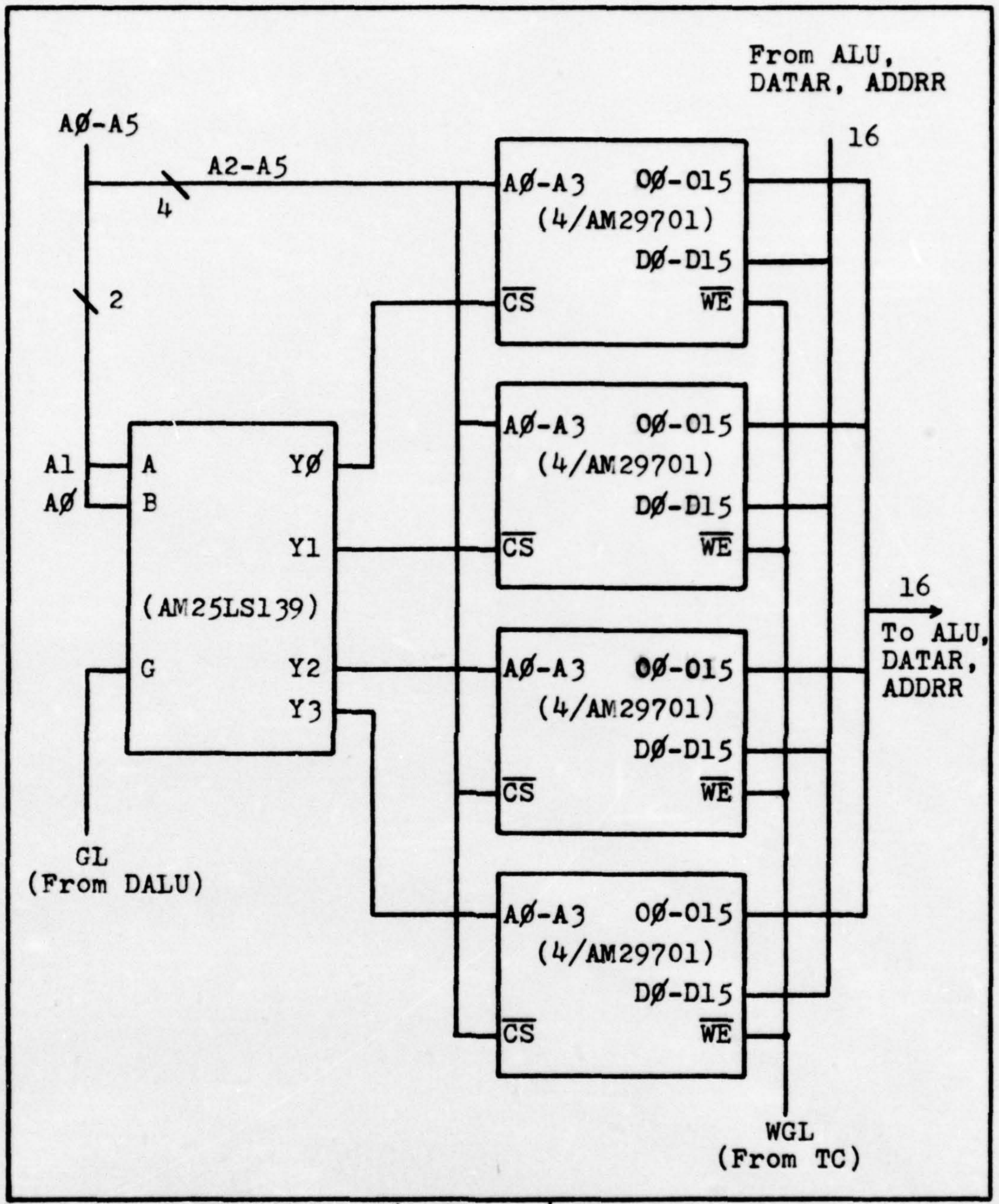


Fig. 16. Global/Local RAM

seperation of global and local registers which will be discussed in Chapter IV. The most active registers are also located in the SP which is divided into global and local areas (Table C-6). This structure allows data and address information to be loaded into DATAR and ADDRR simultaneously to provide faster execution speed.

The MC PROM consists of four AM29751 32 word by 8-bit PROMs. This structure provides 64 words of 16-bits each and is shown in Fig. 17. The DALU field and AØ are used to activate the memory and A1-A5 provide the address. The constants, masks, and other values needed are defined in Table C-7.

<u>Vectored Priority Interrupt Logic</u>

The VPI structure is shown in Fig. 18 and contains the AM2914, AM2913, AM2902, AM25LS138, 54LS139, and AM29761.

The AM2914 Vectored Priority Interrupt Encoder (VPIE) is a high-speed, 8-bit priority interrupt unit that is cascadable to handle any number of priority interrupt

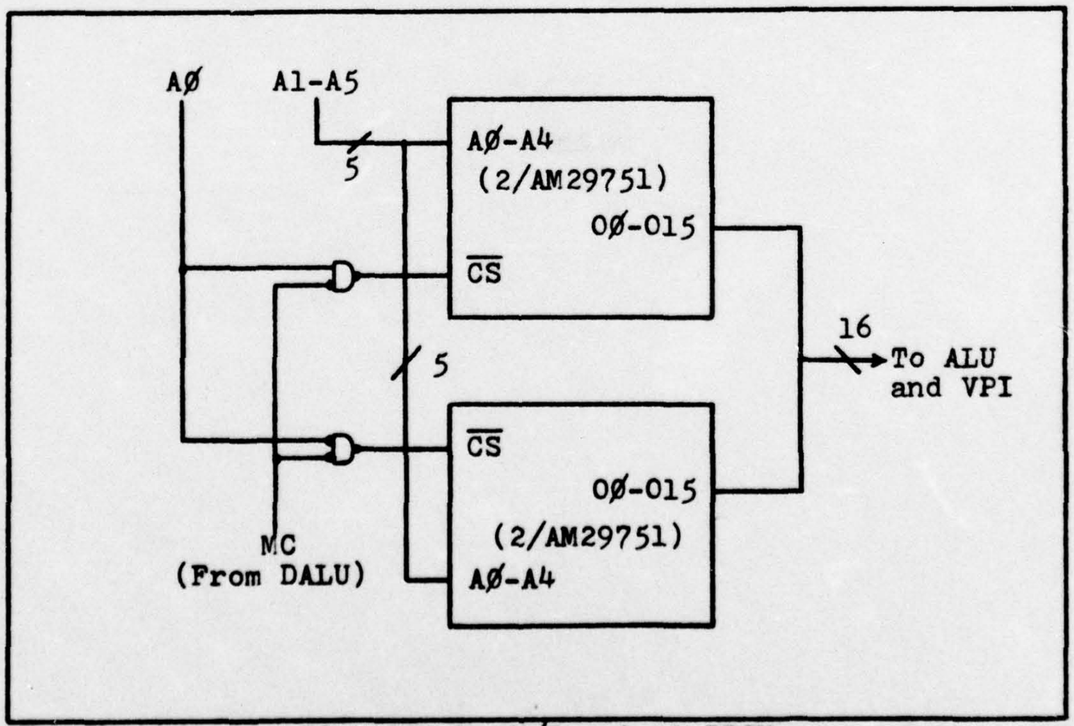


Fig. 17. Mask/Constant PROM

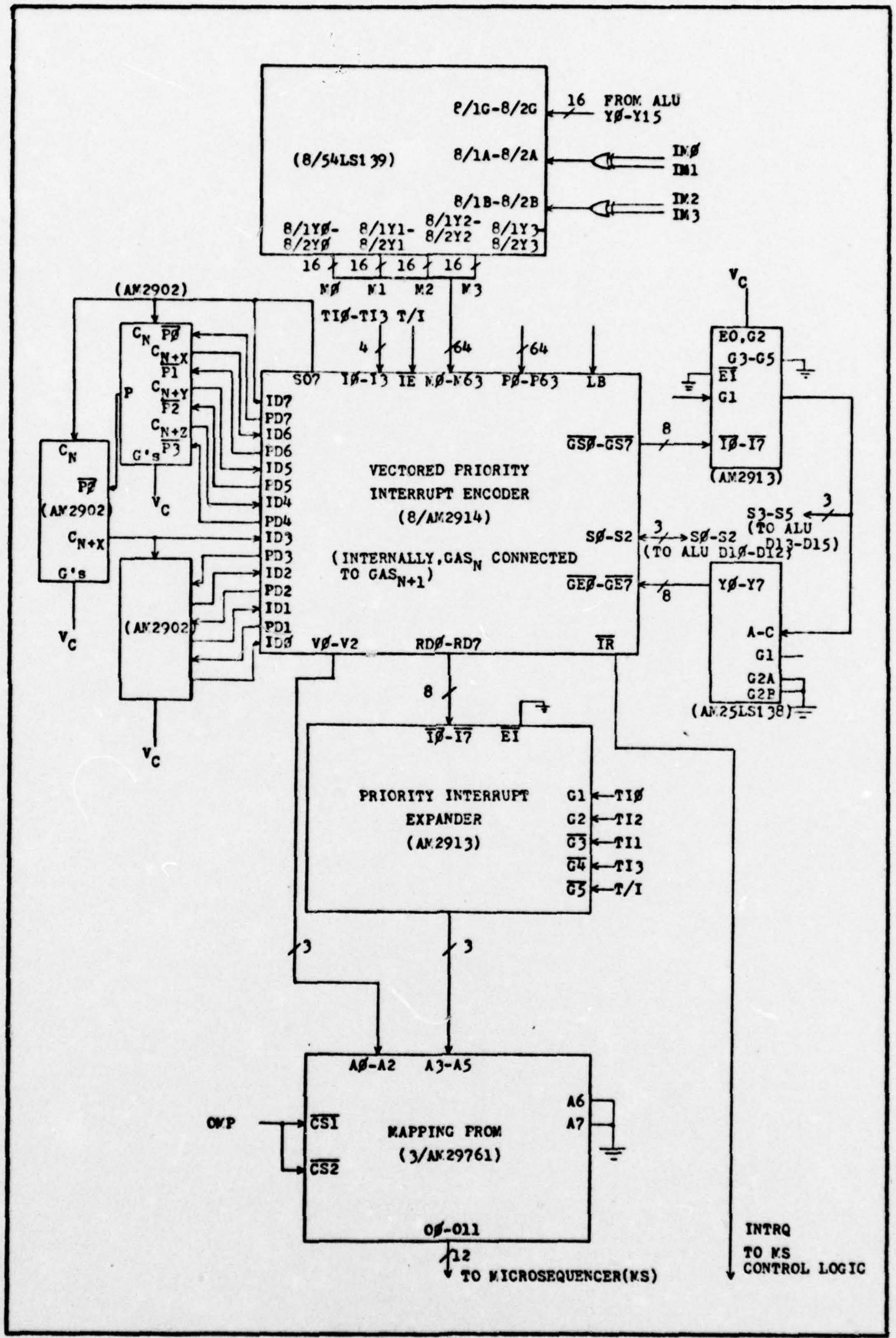


Fig. 18. VPI Structure

request levels (Ref 3:66). This design uses eight VPIEs to provide 64 levels of priority interrupts.

The VPIE receives interrupt requests on 64 interrupt inputs (PØ-P63). Each 8-bit interrupt register has an 8-bit mask register (MØ-M63) which is used to mask individual interrupts. Requests in the interrupt register are ANDed with the corresponding mask register and the results are sent to an 8-input priority encoder. The encoder produces a 3-bit encoded vector (VØ-V3) which represents the highest interrupt that is not masked (Ref 3:66) within a group-of-eight interrupts.

An internal programmable status register (SØ-S2) is used to point to the lowest priority at which an interrupt will be accepted. The contents of the status register are compared with the output of the priority encoder and an interrupt request (INTRQ) will occur if the vector is greater than or equal to the status (Ref 3:66). When a vector is read from the VPIE, the status register is automatically incremented to the next interrupt level. Interconnecting signals (GAS) are provided for moving the status across devices and for inhibiting lower priorities. The VPIE is controlled by a 4-bit microinstruction (TIØ-TI3) which is defined in Table C-8.

The eight AM2914s are connected to three AM2902s (PD1-PD7 and IDØ-ID7) to provide a high-speed, parallel disable of lower priority interrupts. This structure provides a faster reaction time similar to that of the look-ahead carry structure of the BIU processor.

The AM2913 Priority Interrupt Expander is an eight-to-three encoder which is specifically designed for use with the AM2914. The AM2913 that is used with the AM25LS138 three-to-eight decoder provides the status of the VPI (i.e., which interrupt level is active). The higher order bits (S3-S5) are generated from the Group Signal (GSØ-GS7) outputs (and indicate which VPIE chip is active) of the eight VPIEs. The low order bits (SØ-S2) are generated internally by each VPIE (all SØ-S2 outputs are connected in parallel). The AM25LS138 is used to decode S3-S5 and enable the group-of-eight that is active (Group Enable, GEØ-GE7). The low order status bits are program controlled and can be read or loaded by executing the appropriate VPIE microinstruction.

The AM2913 that is connected to the Ripple Disable (RDØ-RD7) outputs is used to encode the lowest group-of-eight interrupts that are active. This device provides a vectored output (V3-V5) only when the AM2914 Output Vector (OVECT) microinstruction is executed. When VØ-V5 are combined, the 6-bit code represents the active interrupt. This 6-bit vector is then sent to the Mapping PROM.

The output vector (VØ-V5) is sent to three AM29761s (256 word by 4-bit PROM) which are used as a Mapping PROM. This PROM contains the starting address of the microroutine that services the interrupt. Although, only an 8-bit address is needed at present, a 12-bit word is provided to allow for system expansion and modification. This allows the Microprogram PROM to grow with only minor design modifications.

Eight AM25LS139 dual one-of-four decoders are used to route the four 16-bit masks from the MC PROM to the mask registers of the VPIEs. This logic provides program control of all interrupts during system initialization and program execution.

The VPI is controlled by the Test/Instruction (TI) field, Transfer/Control (TC) field, and Status (S) field of the MWR.

The interrupts which have been defined and must be recognized are shown in Table C-9. The order shown does not represent a priority scheme since this cannot be established until all microroutines have been coded and a full understanding of program execution is developed. The occurance of one of these interrupts will activate INTRQ which is recognized by the MSC (under program control). Once recognized, a branch operation will occur in the MS to service the interrupt.

MS and MSC Logic

The primary function of this module is to control the source of the address of the next microinstruction to be executed. This decision is made by the MSC and executed by the MS (Fig. 19).

The MS consists of three AM2909 Microprogram Sequencers. This provides room for a 12-bit microprogram address (only

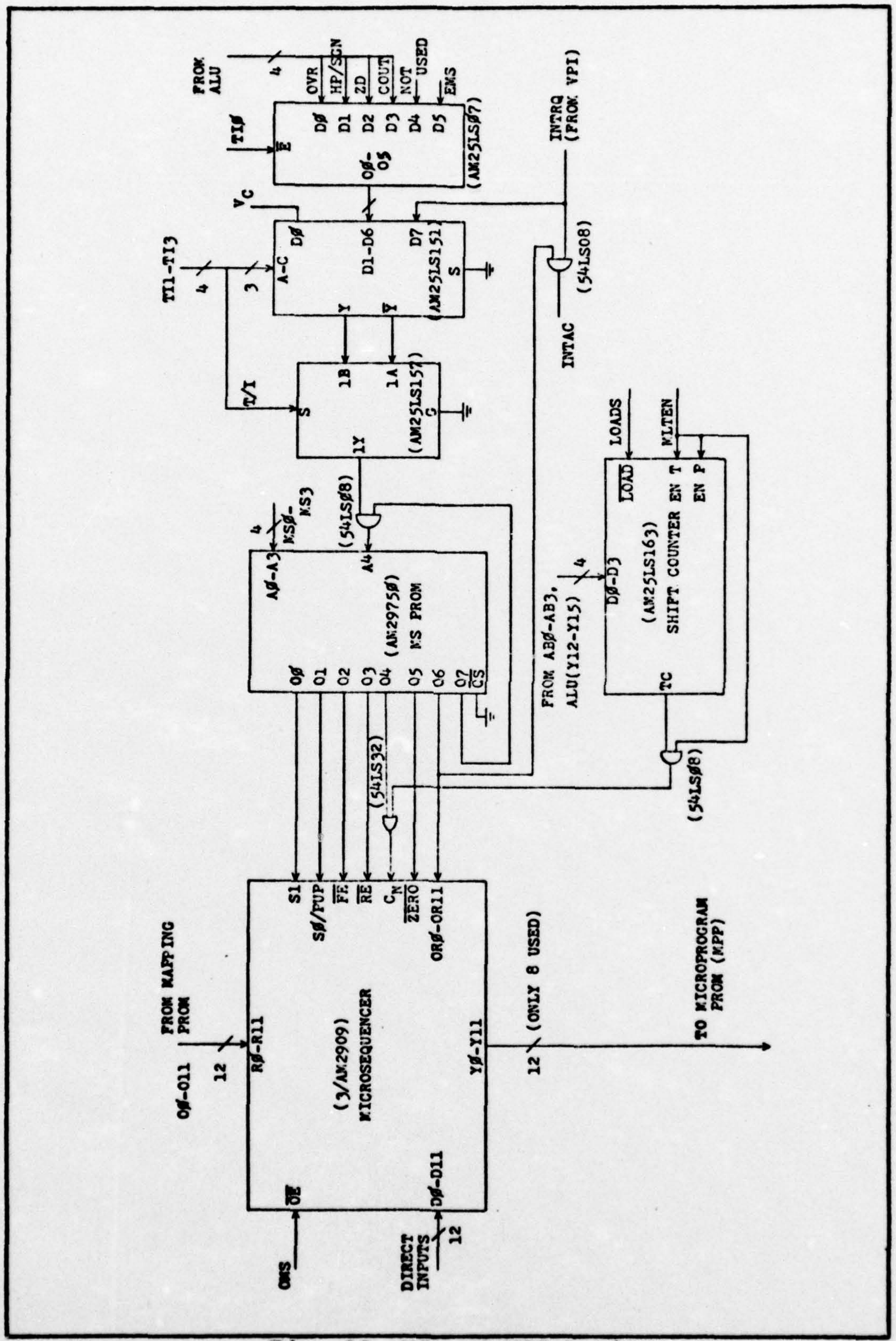


Fig. 19. MS and MSC Logic

8-bits are used at present). The MS can select an address from any one of four sources: (1) external direct inputs (DØ-D11) from the MWR, (2) external register inputs from the MPP (RØ-R11) which are stored in the internal R-register of the MS, (3) an internal four word push/pop stack, or (4) a microprogram count register (MPC) which usually contains the last address executed plus one. These four sources are sent to a four-input multiplexer which is controlled by SØ and S1. The source selected is passed to the three-state Y-outputs which are controlled by Output Microsequencer (OMS) from the TC field of the MWR.

The $\overline{\text{FE}}$ and PUP inputs provide control of the four word stack. This stack allows branching (push) and return (pop) operations to occur based upon the value of $\overline{\text{FE}}$ and PUP. The ORØ-OR11 and $\overline{\text{ZERO}}$ inputs force the output address to all ones or all zeros which can be used as the beginning address for a power-on routine or other highly used micro-instruction. The $\overline{\text{RE}}$ input gates the MPP address into the internal register.

The C_N input controls an internal adder which is connected to the outputs (YØ-Y11) and the MPC. This adder also has a carryout, C_{N+4}, which is used to cascade several AM2909s. The C_N input provides two modes of operation. When C_N is active, the address placed in the MPC is the output address plus one. This operation allows sequencing through a microroutine. When C_N is inactive, the Y-outputs are passed unmodified to the MPC. This procedure allows looping (executing the same microinstruction any number of times).

The MS is directly controlled by the MS PROM which is located in the MSC logic. The MSC consists of the AM25LSØ7, AM25LS151, AM25LS157, AM2575Ø, and AM25LS163.

The AM25LSØ7 is a 6-bit register which is loaded by TIØ from the TI field of the microword. This register holds the five status conditions, four from the ALU and the EMS signal from the message input logic. The output of this register is sent to the AM25LS151 eight input multiplexer. This multiplexer receives these six inputs plus INTRQ (from the VPI) and V_C (which provides an unconditional branch test). The TI1-TI3 signals select the status condition that is to be tested. The T/I signal (applied to the AM25LS157) allows selection of the test condition or its complement. All test condition signals are recognized only if the 06 output of the AM2975Ø (MS PROM) is active. If the test condition is true, the MS PROM address bit (A4) is activated and a branch to a new set of control signals for the MS occurs. The operations performed by the MS PROM that have been defined are shown in Table C-10.

The AM25LS163 is used as a program controlled shift counter. It is loaded from the AB field or ALU (Y12-Y15) outputs when LOADS (from the DALU field) is active. When the MLTEN signal (from the TC field) is active, the MS will perform a loop operation which will allow a defined number of shift operations. When the shift counter reaches zero, its output becomes active which in turn activates C_N and allows the MPC to be incremented.

Global/Local Input Logic

The logic required for message input and output is partially dependent upon how the microroutines are coded. Since only the Message Reception Routine has been coded, only the Global/Local Input logic has been defined (See Fig. 20 and Fig. 21).

The Global/Local Input logic receives data from the A and B Bus Translation Units. The "primary" and "alternate" bus are interchangable and are under program control. The switching function is controlled by a toggle flip-flop (54LS78) which is activated by the Input Switch Command (ISWCMD) from the TC field. The ten outputs from the "primary" bus are paired and then ORed together to provide the five message input signals which are sent to the "primary" bus control logic. The same procedure is used for the "alternate" bus.

During system initialization (power-on), $\overline{\text{REN}}$ (Receive Enable) is sent to the VPI to detect which bus is "primary" and switch to the A BTU if necessary (GIPR, Global Input Preset).

The GMHS, GENER, and GEMS signals are sent directly to the VPI to detect message start, data encode errors, and message termination.

The "primary" bus logic consists of the Input Data Bit Counter (IDBC) and the Input Shift Register (ISR).

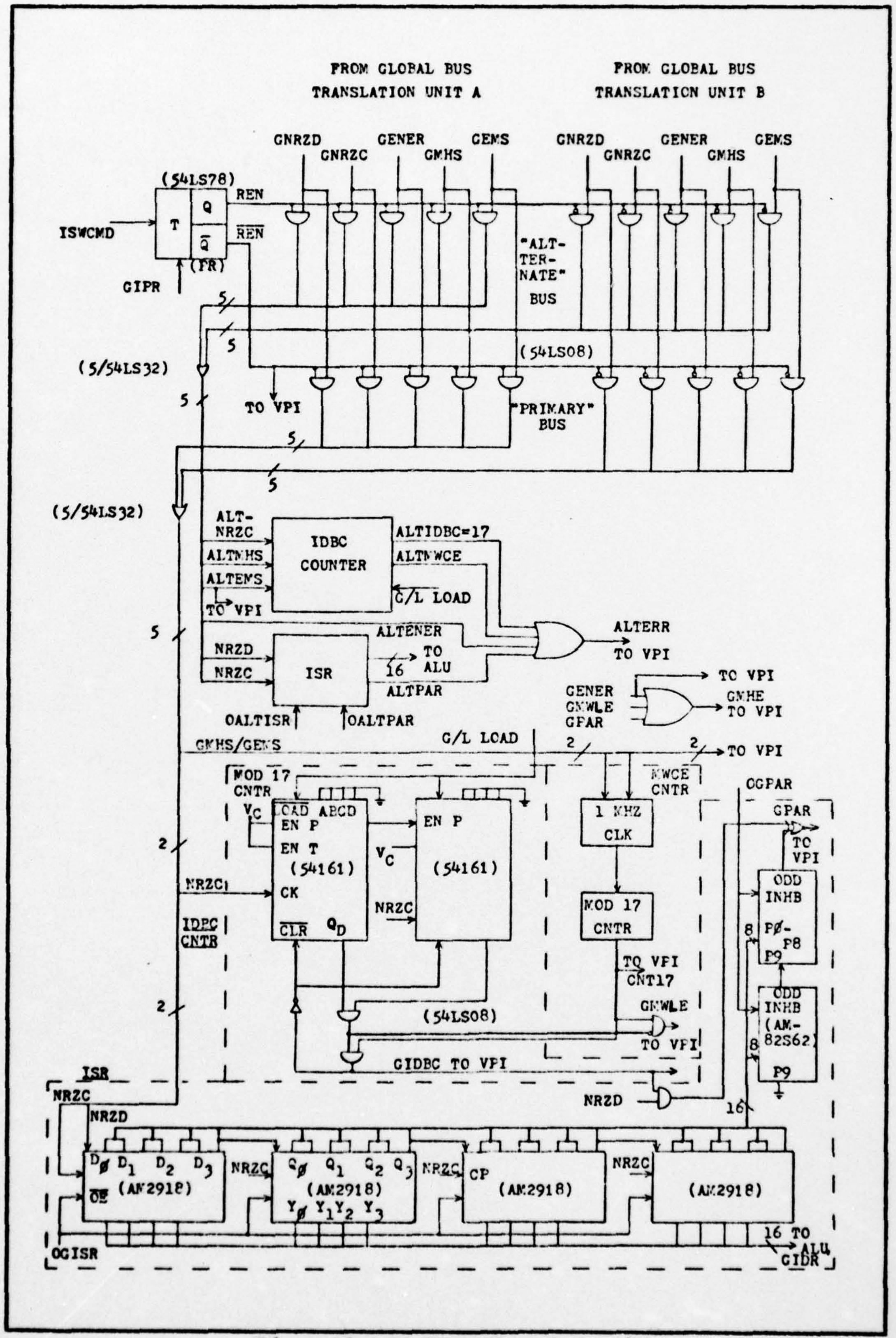


Fig. 20. Global Input Logic

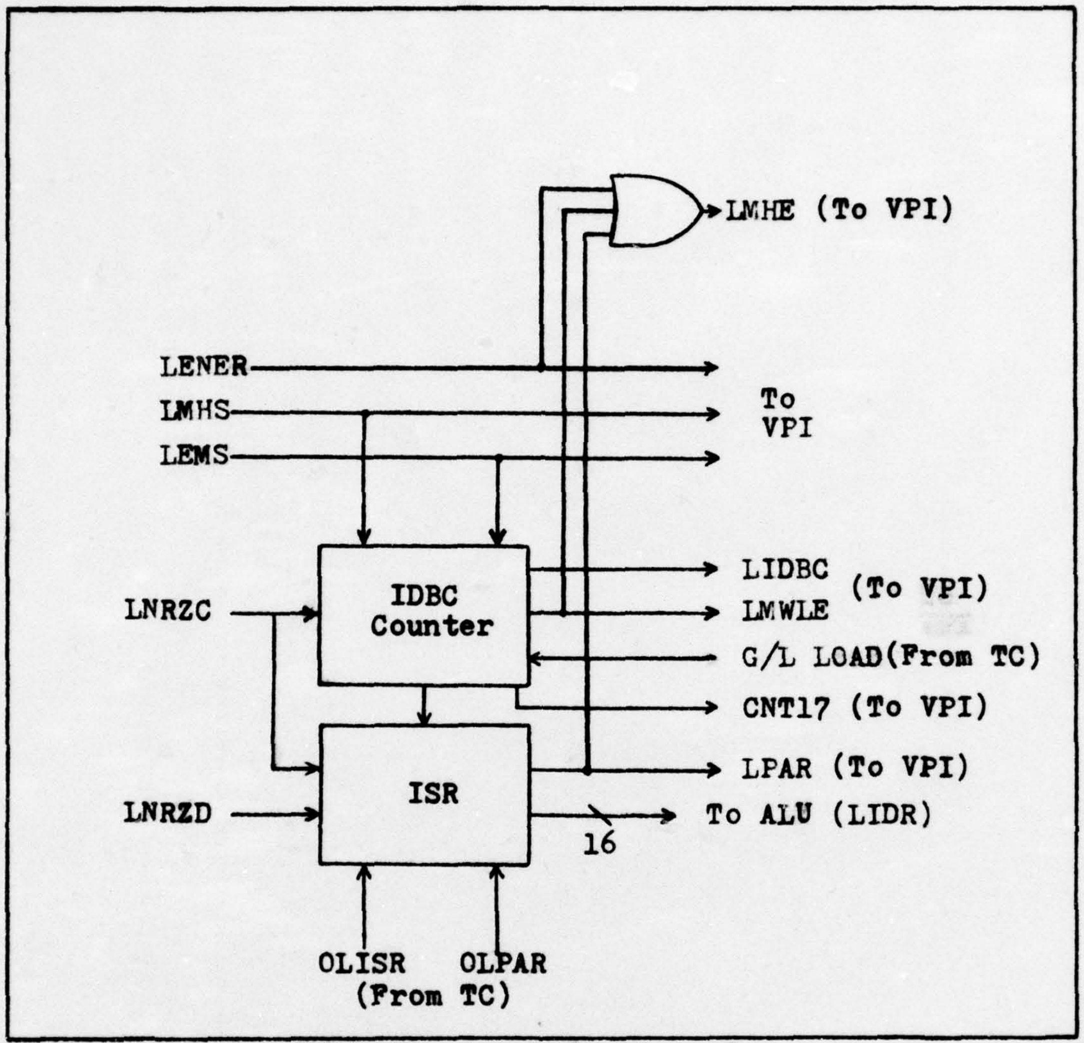


Fig. 21. Local Input Logic

During power-on, the IDBC is initialized by G/L LOAD. The IDBC contains the MOD-17 Counter and Message Word Count Error (MWCE) Counter. The MOD-17 Counter (2/54161) counts the incoming NRZ clock pulses. The MWCE Counter is an independent counter with its own 1-MHZ clock which counts modulo 17. This output (CNT17) is sent to the VPI. It is also compared to the MOD-17 output. If both outputs agree, GIDBC is sent to the VPI to indicate that an input word has been assembled. If the counters disagree, GMWLE is sent to the VPI to signify a data word error. If this error occurs during Message Header assembly, it is ORed with GENER and GPAR to indicate a Message Header Error (MHE) to the VPI. GIDBC is also compared with the 17th data bit to determine input parity and is also used to re-initialize MOD-17 and MWCE for the next data word input.

The NRZ clock and NRZ data are sent to the ISR (4/AM2918). The serial data is shifted into parallel format in the ISR. The ISR provides standard outputs which are used to generate odd-parity (2/AM82562). This parity is compared to the input parity and if they disagree, a parity error (GPAR) is sent to the VPI. The ISR also has three-state outputs (YØ-Y15) which are sent to the ALU and GLRAM (controlled by OGISR from the TC field) when a valid Message Header or a data word is assembled.

The "alternate" control logic performs the same function as the "primary" logic except that the VPI is only notified if the ALTEMS and the ALTERR signals occur. The "alternate" logic is waiting for a one word message, "switch-to-alternate-bus-command" (SOC). The VPI needs to know if this command occurs and if there is an error.

Summary

This chapter has presented a general description of the hardware used in the BIU design. A thorough understanding of each IC and its functions can be obtained only after studying Reference 3. With the hardware and microword defined, the next step is to code the microroutines. The Message Reception Routine has been coded to demonstrate the feasibility of the design concept.

IV. Microcoding

Coding Philosophy

To code the microroutines, the BIU register transfer language description, the interrupt service procedures of the AM2900 configuration, and the microword must be combined to provide the functions required for message input and output. The interrupt service philosophy allows each major routine to be broken into several smaller service routines which are accessed when that particular interrupt occurs. The ability to mask interrupts and control the status of groups-of-eight interrupts allows each major routine to control which interrupts it will recognize. Each routine will mask unwanted interrupts when it is accessed and during execution. It will also re-establish the desired interrupt structure after a return from a subroutine is executed. During execution, the program controlled VPI status allows the routine to shift the interrupt level up or down within a group-of-eight interrupts. This procedure lets the routine skip interrupts that are no longer applicable or to reactivate interrupts that have already occurred.

Prior to coding the Message Reception Routine (MRR), several general service routines were required. These routines and the MRR are presented in Appendix D. Appendix D provides an RTL description and the microcode which demonstrates the multiple TC control signal philosophy discussed in Chapter III. Table D-1 shows the RTL description for the general service routines and Table D-2 provides the microcode. Tables D-3 and D-4 provide the RTL and microcode description for the MRR. The combination of Appendices C and D provide a complete description of the function of each microinstruction.

The system concept used while coding the MRR routine was that all the microroutines (except GBMR) are shared by the global and local buses (e.g., MRR is used for global and local message reception). This concept will require additional control logic (not shown) to accomodate simultaneous global and local message reception or transmission. This procedure can be accomplished with additional hardware or an additional general service or executive routine. In either case this logic must be able to control the most significant address bit of the GLRAM and the BIU processor SP (both A and B ports). This logic must test the message type (global or local) after each data word is received/transmitted. If only one message type is active, the BIU will continue to service it. If either global and local reception or transmission are simultaneously active, the BIU will alternately service them. To accomplish this, a two-input AND gate is placed in each of the most significant address lines of the GLRAM and SP. These gates will be controlled by the logic just described. The most significant bit of the A field and AB field is always active and is passed to the GLRAM and

SP address inputs when the gate is active. This procedure divides the GLRAM and SP into global and local areas of responsibility as shown in Tables C-5 and C-6. When other routines are being executed, the most significant address bit gate can be controlled by the TC field so that the necessary locations in the GLRAM or SP can be accessed.

Service Routines

Three general service routines were designed: (1) Fetch Interrupt (FINT), (2) Return (RTN), and (3) Service Interrupt (SINT). The FINT routine is accessed immediately after system initialization or any time the system is idle (i.e., no message input or output activitey). After system initialization, FINTØØ (first instruction of the FINT microroutine) establishes a wait-for-interrupt-loop until the first interrupt occurs. When the system becomes idle, the last routine executed will perform an RTN instruction which will pop the MS stack and FINTØ3 will cause an unconditional branch to FINTØØ. The RTN instruction provides the link back to a routine from a subroutine. When RTN is executed, control is returned to the routine that was interrupted or that called the subroutine. The SINT routine is primarily used to provide fast service for interrupts that are program controlled (i.e., interrupts that are set by the microroutine that is presently executing to force a branch to a subroutine). This procedure is mainly used to provide the link between the executing routine and interrupt service routines.

MRR Microroutine

This routine was designed to provide one common MRR (discussed above) to handle global and local messages, therefore, no reference is made to global or local messages in the routine. The procedure used to code the MRR followed the RTL description combining the system characteristics and microword format with the MRR functions.

Since there is a 17 microsecond delay between the MHS and assembly of the MH, the MRR routine first prepares the system for message reception activity. This is accomplished by masking the appropriate interrupts and setting the VPI status. The coded routine (Table D-4) shows masking only one 16-bit mask. After defining the interrupt field (PØ-P63), several mask words may need to be initialized. A RTN is then executed to allow the BIU to continue processing other interrupts.

If ENER, MWLE, or PAR occur, there is an error in the MH and the MHE interrupt will be set. This will cause the MHE service routine to be executed and the remainder of the message will be ignored.

If there are no MH errors, the IDBC interrupt will generate the MRRØ3 address and message input will continue. The ISR (MH) is placed in the GLRAM and SP. A test is made for a high priority message (HP) and if true, the HP interrupt is set by the TC field and a branch is made to SINT which will allow the Interrupt Routine (IR) to notify the PE processor. The MIAMM address is generated and the

MIAMM word is obtained from PE memory. Then the Message Header/Message Length Input Queue address is generated and placed in the SP (IAR).

The response bit (RB) is located, shifted into the sign bit position, and tested. If the RB is true, message input is continued. If it is not true (message is not for this PE), a branch is made to the RB service routine where the message length is obtained from PE memory. If the message length is greater than eight words, the Bus Dominance (BD) interrupt is disabled (A toggle flip-flop with BDGATE from the TC field as the input signal controlling a two-input AND gate whose output is sent to the BD interrupt in the VPI. This hardware is not shown with the VPI logic in Fig. 18. If the message length is less than or equal to eight words, BD is left enabled.

If RB is true, message input continues and the MH is placed into the PE memory MH Input Queue. The address presently in the SP (IAR) is now used to obtain the beginning address of the Message Buffer Space. The first word in this space is obtained and placed in the ILR which specifies the length of the incoming message. After the message length is obtained, the IAR is incremented by one. The message length is tested to determine if it is greater than eight and BD is again disabled or left enabled depending upon message length.

At this point, the BIU is prepared to receive incoming data words and place them in the Message Buffer Space. The remainder of the routine follows the RTL description of the MRR with the exception of the EMS(1), EMS(2), PAR, and ENER which have been designed as seperate interrupt service routines. If a data word has an error, the PE processor is notified and message input continues (the decision to use or not use the data is made by the PE processor). These errors will cause an interrupt which will cause a jump to the appropriate service routine and a return to MRR. EMS(1) is used when the ILR is decremented to zero. EMS(2) is used when EMS occurs prior to an ILR value of zero being reached signifying a MWCE. The MRR27 through MRR38 micro-instructions are executed for each incoming data word. This process continues until either the ILR reaches zero and/or EMS occurs.

Other Microroutine Coding Concepts

The techniques used to code the remaining microroutines should follow the same basic procedures used for coding the MRR.

The Global/Local Output logic has not been developed but will essentially consist of a parallel-to-serial shift register (OSR), parity generator, gating logic to switch global buses and enable transmission, and the interface logic required with the Message Transmission Routine (MTR) and the Global Local Bus Access Routine (GLBAR). As the MTR and GLBAR are coded, a better understanding of the requirements for the output logic will be developed.

The MTR and GLBAR are unique in that they act as coroutines which will require some manipulation of the MS stack. A coroutine is where two routines (A and B) interact such that "When B returns to A, it jumps to the statement following the call to B...When A transfers control to B, it goes not to the beginning (except the first time) but to the statement following the most recent return--that is the most recent call of A..." (Ref 4:128).

The Interrupt Routine (IR) and Global Redundant Bus Management Routine (GBMR) should be relatively straight forward. The only major routine left with several unique characteristics is the Command Address Word Routine (CAWR). A CAW is initiated by the PE processor and all devices on the I-Bus must be able to recognize the CAWs that apply to it. When the PE processor issues TRQ (which will cause a VPI interrupt and a branch to CAWR), the BIU will mask the BMID and compare it to the Processor Identification (PID) in the MC PROM. If there is a match, the BIU will use the value in the ADDRR as a CAW trap address. The BIU must first test the trap address to determine if it is within the range of trap addresses that apply to the BIU. If it is, the trap address will be sent to the D-inputs of the MS where a branch will be made to provide the desired PE processor response.

The Clear/Reset (CLR/RST) routine is used by the PE processor to force the BIU into a known state so that error recovery procedures can begin.

Summary

This chapter has provided an example of the coding procedures used to combine the BIU functions and the AM2900 design. This was accomplished using the concept of common global/local routines with one microprocessor. There are several alternatives to this approach and a number of design problems not mentioned that are discussed in Chapter V.

V. Results and Conclusions

Design Limitations

Several areas necessary to provide a complete, detailed design were not included or analyzed in the system presented in Chapter III. These areas include skew, pulse width, open collector outputs, small segments of logic for various controls, large areas of logic not designed, and maintenance requirements.

Skew or timing considerations between incoming data signals and internal control signals were not analyzed. For example, the 17-Counter and MWCE Counter in the IDBC of the Global/Local Input Logic may not be synchronized after several data words are placed in PE memory. To alleviate this problem, the G/LIDBC signal can be used to re-scynchronize the 1-MHz clock after each word. Other areas within the BIU may require a similar analysis.

The pulse width required to set a VPI interrupt was not considered (no data available in preliminary specifications). Since the G/LIDBC signal is used to reset the 17-Counter, the length of this pulse may be extremely short. Therefore, a one-shot multivibrator may be necessary to provide a pulse width of sufficient length to set the VPI interrupt.

Open collector outputs are not shown in the design of Chapter III. These will have to be incorporated in the final, detailed system description.

Also, maintenance requirements have not been considered. The ability to manually control sequencing through routines and setting interrupts with visual acknowledgement (i.e., Interrupt Acknowledge, INTAC) must be added to the design.

In many cases, as coding continues, minor segments of hardware logic will be required (e.g., the BDGATE control logic was added while coding the MRR). As an example, the BIU is considerably faster than the PE processor and inter-communication is asynchronous. When data is passed to the I-Bus, it must be held there until the PE processor acknowledges receipt. This will require a flip-flop to control OIBR, OADDRR, and ODATAR which can be reset by TACK after the PE processor receives the data.

If the one processor, common routine concept is to be used, the GLRAM and SP control logic (whether hardware or software) must be designed.

There are two major areas which have not been included. These are: (1) the Global/Local Output Logic, and (2) the power-on logic. These need to be added before the design is complete.

Speed of Operation

With the present configuration, the fastest time possible from the occurance of an interrupt to MPP access is approximately 260 nS (preliminary specifications). Also, by sending the MP PROM output to the MS R-register requires an additional clock cycle to load the R register (the next clock cycle will pass the R register to the MS Y-outputs).

This additional delay can be eliminated by either connecting the MP outputs to the MS D-inputs or by using the AM2911 which is the same as the AM2909 except that the D and R inputs are internally tied together (and the OR input function is removed). Also, the AM2913 (PIE) has an $\overline{EO}$ control signal (not shown in Fig. 18) which is activated any time the VPIE RDØ-RD7 outputs are active. This control can be used as the OMP control signal to provide asynchronous control of the MP PROM output. These procedures will allow an approximate access time from VPI interrupt to MPP output of approximately 220ns.

Since the worst case cycle time for the AM2901 microprocessor is 120ns, the address access time becomes the critical cycle time. This means that a system clock of approximately 4.5 MHz can be used. Another alternative would be to force the AM2901 into an idle state (e.g., NOP instruction, or placed under control of the executive discussed below) for two cycle times any time INTRQ is recognized. This would provide the access time required and allow the microprocessor to operate at twice the speed during normal microroutine execution (approximately 9MHz clock). This should provide the speed required for the one processor, common routine concept.

Alternate Designs

If it is discovered that the one processor concept is not practical, two AM2901 16-bit microprocessors can be used to provide separate control of global and local responsibilities.

This procedure would require essentially two identical systems to handle simultaneous global and local messages.

A second design alternative alluded to in Chapter IV would be to provide a system software executive for overall system management. This would provide an interface between the interrupts and major routines which would monitor the interrupts which control the major routines while the major routines would monitor the interrupts that control the service routines. This programming concept is shown in Fig. 22. The general service routines would be available to all major routines.

Conclusions

This design does not provide the detail required to go directly to system implementation. It does provide a basic configuration and coding scheme which when expanded, will provide the detail necessary for implementation. Many areas could not be covered thoroughly because of the lack of detailed specifications and functional descriptions.

The modular BIU functional concept has been maintained and the detailed functional description provided for software design. The large number of registers and register manipulations can be implemented with the use of high-speed RAM which is divided into global and local areas of responsibility. This division plus the masking PROM provides the capability to test and manipulate register words or individual bits.

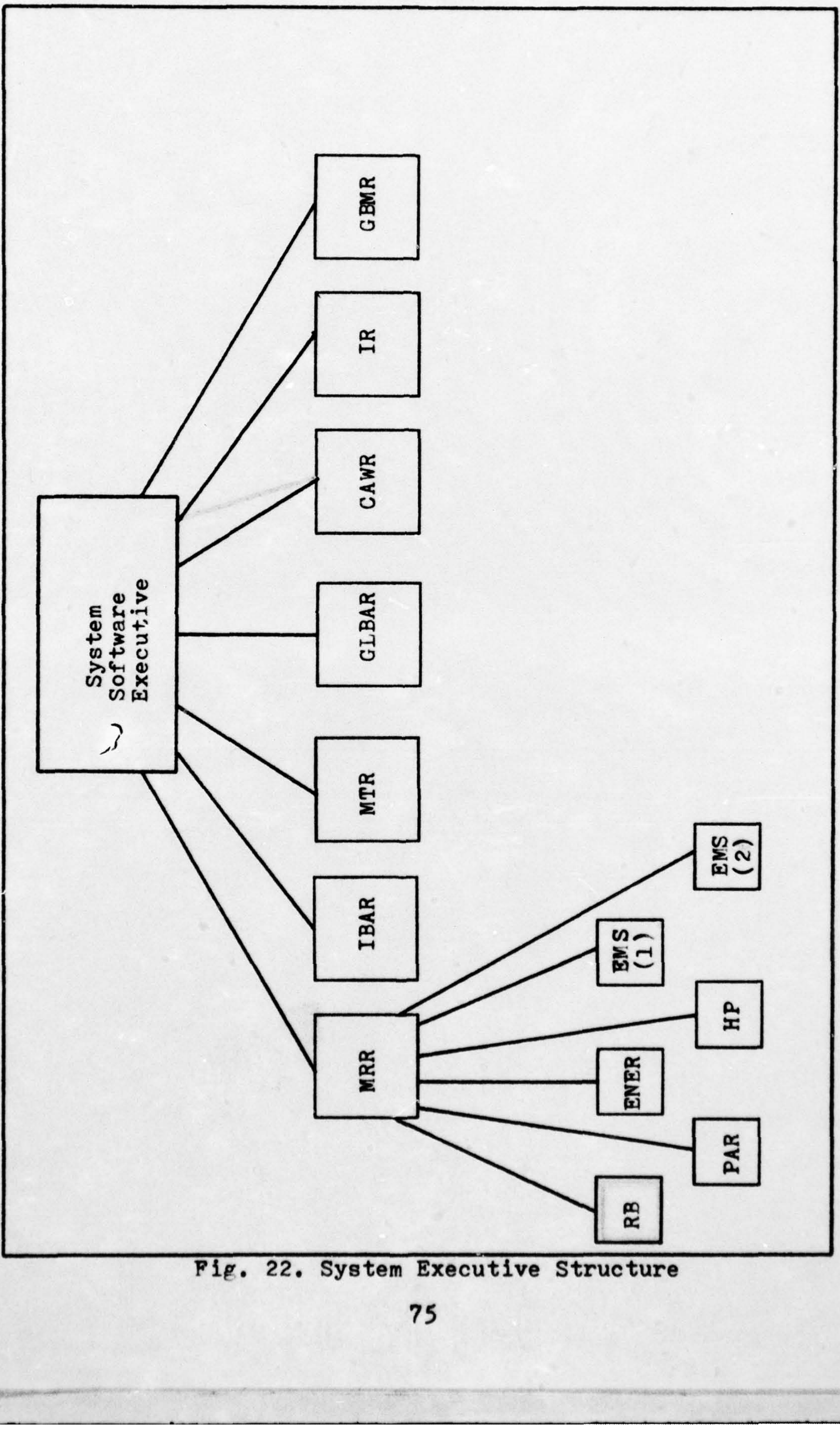


Fig. 22. System Executive Structure

Because of the speed and flexibility associated with this IC family, the BIU can be implemented with the design concept presented. The speed required for simultaneous messages can be achieved with the design concept given or the alternate design.

This design can be used in a laboratory test model to demonstrate DP/M feasibility and if the speed is not sufficient for actual implementation at the present time, it should be very close, and the state-of-the-art should provide the required speed in the 1980 time-frame.

Bibliography

1. Air Force Avionics Laboratory. Distributed Processor/ Memory Architectures Design Program. Report. Dallas, Texas: Texas Instruments Incorporated, February 1975.

2. DP/M Preliminary Design. Report Draft. Wright-Patterson Air Force Base, Ohio: Air Force Avionics Laboratory, December 1975.

3. Advances Micro Devices Incorporated. AM2900 Bipolar Microprocessor Family. Catalog. Sunnyvale, California: Advanced Micro Devices Incorporated, 1976.

4. Tanenbaum, Andrew S. Structured Computer Organization. Englewood Cliffs, New Jersy: Prentice-Hall, 1976.

Appendix A

<u>Definition of Mnemonics for BIU Register Transfer Language Description</u>

<u>ADDRR (Address Register)</u> - a 16-bit register in the BIU that interfaces with the I-Bus. The contents of this register is designated as ADDR.

<u>ALTBUS (Alternate Bus)</u> - refers to the global bus that is presently being used as the alternate bus.

<u>BD (Bus Dominance)</u> - a flag that signifies that a PE is out of control and is dominating the global primary bus. This flag is located in G/LIFR(Ø2) and is masked by G/LISR(Ø2). When BD occurs and if G/LISR(Ø2) is active, the BIU sends BD to the PE processor (See also PL3-6M).

<u>BDGATE (Bus Dominance Gate)</u> - controls BD. When it is desired to send a message greater than eight words, BDGATE is activated and the message length control logic is unable to set BD.

<u>BGR (Bus Grant)</u> - notifies a device connected to the I-Bus that it is in control of the I-Bus. This signal is threaded through each device from highest to lowest priority. It is called Bus Grant In (BGRI) and Bus Grant Out (BGRO). When a device requests the I-Bus, it blocks BGRO so that a lower priority device cannot access the I-Bus.

<u>BLR (Bus Length Register)</u> - an eight bit register that

contains the value to be loaded into the BPR. This value determines when the PE can access the global bus.

<u>BMIDR (Bus Master Identification Register)</u> - a four-bit register that interfaces with the I-Bus. The contents of this register is the BMID of the device currently in control of the I-Bus.

<u>BPR (Bus Position Register)</u> - an eight bit register that is loaded from the BLR. This register is decremented each time an EMS occurs on the global bus. When it reaches zero, that PE has control of the global bus for message transmission.

<u>BQ (Bus Quiescence)</u> - a flag that signifies that the primary global bus or local bus has had no traffic for 50 microseconds. This flag is located in G/LIFR(Ø1) and is masked by G/LISR(Ø1). When BQ occurs and if G/LISR(Ø1) is active, the BIU sends BQ to the PE processor (See also PL3-6M)

<u>BQTIMER (Bus Quiescence Timer)</u> - a modulo 50 counter that counts in one microsecond intervals. The BIU resets this timer to zero at regular intervals if there has been traffic on the primary global bus. If this timer reaches 50 microseconds, the BIU is notified which in turn notifies the PE processor (BQ).

<u>BREL (Bus Release)</u> - an I-Bus control signal which is issued by a device once it gains control of the I-Bus.

<u>BRQ(n) (Bus Request)</u> - a BIU control signal which is used to signify that the BIU is ready to send/receive data

on the I-Bus. The value of n (n=1-4) represents the priority of the request:

n=1, Global Message Reception request

n=2, Global Message Transmission request

n=3, Local Message Reception request

n=4, Local Message Transmission request

<u>CAW (Command Address Words)</u> - a command issued by the PE processor to an I-Bus device to request data, or to send data. The CAWs for the BIU are designated as ØØØ2-ØØØD (Hexidecimal).

<u>CLR/RST (Clear/Reset)</u> - an I-Bus control signal issued by the PE processor to all I-Bus devices. When issued, all devices halt activity and place themselves in a known state so that the PE processor and GEX can begin error recovery procedures.

<u>DATAR (Data Register)</u> - a 16-bit register in the BIU that interfaces with the I-Bus. The contents of this register is designated as DATA.

<u>DB(17) (Data Bit 17)</u> - represents the parity bit appended to each data word during message transmission activity.

<u>DRCVR (Data Receive Register)</u> - a one bit BIU register that interfaces with the I-Bus. The value (DRCV) of this register determines whether the device in control of the I-Bus is preparing to send data (DRCV=1) or receive data (DRCV=Ø).

EMS (End Message Synchronization) - a global bus control signal that notifies all PEs that message transmission has terminated. Also notifies the next PE in the round-robin cycle that it is his turn to transmit.

ENER (Encode Error) - a signal from the BTU that signifies a Manchester II encoding error has occured in an incoming Message Header or data word.

GT (Gap Timer) - a 2-5 microsecond clock that delays message transmission. This gap is between the EMS of the previous transmitting PE and the MHS of the next transmitting PE. It compensates for bus skew on a long global bus.

H/L TRAP (High/Low Priority Trap) - the address sent to the PE processor to service a high or low priority interrupt from the BIU. This word is used for global and local interrupts (a total of four trap addresses). Two for global high/low priority interrupts and two for local high/low priority interrupts.

HP (High Priority) - a one bit word, the first bit in the Message Header. When set, it signifies a high priority message which requires immediate attention. This information is sent to the PE processor by setting G/LIFR(ØØ). When G/LISR(ØØ) is set, the information will be passed to the PE processor.

IAK (Interrupt Acknowledge) - a one bit word issued by the PE processor to notify the I-Bus device that its interrupt request has been honored. This signal is threaded through each device from highest to lowest priority. The incoming signal is Interrupt Acknowledge In (IAKI) and the outgoing signal is Interrupt Acknowledge Out (IAKO). See BGR definition.

IAR (Input Address Register) - a 16-bit register that contains the address for the next memory access during message input (used in global and local message reception).

IDBC (Input Data Bit Counter) - a modulo 17 counter in the Global/Local input logic that determines the bit count in each data word.

IDR (Input Data Register) - a 16-bit register that contains the assembled data words prior to being sent to PE memory. There is a global and local IDR.

IFR (Interrupt Flag Register) - a 16-bit register that contains the interrupt flags which are sent to the PE processor. G/LIFR(ØØ-Ø8) are defined as high priority interrupts. G/LIFR(14-15) are low priority interrupts. G/LIFR(Ø1-Ø2,14) are sent to the PE processor only if G/LISR(Ø1-Ø2,Ø9) is set when the interrupt occurs (See also PL3-6M).

ILR (Input Length Register) - contains the length (number of data words) of the incoming message. Used in global and local message input.

IMDR (Input Mask Data Register) - contains the MIAMM word obtained from PE memory. Each bit in this word represents a particular message and if set, the message is for this PE.

INHB (Inhibit) - a maintenance test function which controls I-Bus access for G/LIFR interrupts.

IOSLR (Input/Output Select Register) - a one bit register that interfaces with the I-Bus and the value (IOSL) determines if the device being accessed is PE memory or another I-Bus device.

IQP (Input Queue Pointer) - an address register whose three least significant bits are a modulo 8 counter. This address is the location in PE memory where the incoming Message Header is placed.

IRQ (Interrupt Request) - an I-Bus control signal which is used to gain control of the I-Bus when a device has an IFR interrupt and must notify the PE processor.

ISR (Input Shift Register) - a 16-bit register used to convert incoming serial data to parallel format (used in global and local message reception).

ISR(n) (Interrupt Status Register) - a 16-bit register controlled by the PE processor and is used to mask G/LIFR interrupts. G/LISR(ØØ-Ø8) represent the high priority interrupts. G/LISR(Ø9-1Ø) represent the low priority interrupts. G/LISR(11,14-15) are used to control message output. All interrupts are also controlled by PL3-6M.

ISWCMD (Input Switch Command) - a BIU control signal which switches the BTU input from the "primary" bus to the "alternate" bus. This command is issued when the GEX realizes that a PE has gone out of control and the GEX begins fault recovery procedures by issuing a Switch Over Command on the "alternate" bus to all PEs.

<u>MDEE (Message Data Encode Error)</u> - a flag that signifies a Manchester II encoding error in an incoming data word. This flag is located in G/LIFR(ø8). The PE processor is notified when G/LISR(ø8) is set (See also PL3-6M).

<u>MDPE (Message Data Parity Error)</u> - a flag that signifies that a data word parity error has occured. This flag is located in G/LIFR(ø7). The PE processor is notified when G/LISR(ø7) is set (See also PL3-6M).

<u>MHE (Message Header Error)</u> - a BIU control signal that signifies that a parity, encode, or word length error has occured. If these errors occur in a Message Header, the message is ignored.

<u>MHS (Message Header Synchronization)</u> - a global bus control signal that notifies all PEs that a message transmission has begun.

<u>MR (Message Received)</u> - a flag that signifies message input has terminated. This flag is located in G/LIFR(14) and is sent to the PE processor only if G/LISR(ø9) is set when MR occurs (See also PL3-6M).

<u>MWC (Message Word Counter)</u> - a register that contains the numbers of data words in a message output. If it exceeds the authorized word length (usually 8 words), BD is set and message output is terminated.

<u>MWCE (Message Word Count Error)</u> - a flag that signifies that an incoming message does not contain the correct number of data words. This flag is located in G/LIFR(ø5) and is sent to the PE processor when G/LISR(ø5) is set (See also PL3-6M).

MWLE (Message Word Length Error) - a flag that signifies a data word has the incorrect number of bits. This flag is located in G/LIFR(Ø6) and is sent to the PE processor when G/LISR(Ø6) is set (See also PL3-6M).

NRZC (Non-Return-to-Zero Counter) - a modulo 17 counter that determines if each incoming data word contains 17-bits (counts the NRZ clock pulses).

OA (Output Active) - a flag located in G/LISR(14) controlled by the BIU and PE processor to keep each other informed of the status of message output. When set, message output is in progress.

OAR (Output Address Register) - same as IAR except it is used for global/local message output.

OC (Output Complete) - a flag that signifies message output has terminated with no known errors. This flag is located in G/LIFR(15) and is sent to the PE processor when G/LISR(1Ø) is set (See also PL3-6M).

ODBC (Output Data Bit Counter) - a modulo 17 counter which counts the number of bits in each outgoing data word to determine if the word has been shifted out of the OSR.

OEMS (Output End Message Synchronization) - a control signal to the BTU that signifies message output is complete and that EMS should be appended to the message.

OEN (Output Enable) - a flag controlled by the PE processor and when set, allows the BIU to continue message output.

OLR (Output Length Register) - same as ILR except it is used for global/local message output.

OMHS (Output Message Header Synchronization) - a control signal to the BTU that message output is ready to start and to place MHS on the bus.

OMLV (Output Message Length Violation) - a flag that signifies the value placed in the OLR is greater than eight. This flag is located in G/LIFR(Ø3). The PE processor is notified when G/LISR(Ø3) is set (See also PL3-6M).

OP (Output Pending) - a flag controlled by the BIU and PE processor, G/LIFR(11). When set, signifies that the BIU is ready to begin message output.

OPAR (Output Parity) - a BIU control signal that determines the value of DB(17). When set, DB(17) is set and appended to each data word.

OSR (Output Shift Register) - a 16-bit shift register which changes each outgoing data word from parallel to serial format.

OSWCMD (Output Switch Command) - a BIU control signal that switches the global output from the "primary" to the "alternate" bus. This signal is controlled by a CAW issued by the PE processor.

PAR (Parity) - a BIU control signal that signifies (when set) that the odd-parity of an incoming data word is correct.

PID (Processor Identification) - the PE processor identification. When the PE processor is in control of the I-Bus, this value will be in the BMIDR.

PL3-6M (Primary Level Interrupt 3-6 Mask) - a four-bit register that masks high/low priority interrupts:

PL3M masks all global high priority interrupts

PL4M masks all global low priority interrupts

PL5M masks all local high priority interrupts

PL6M masks all local low priority interrupts

If PL3-6M and/or G/LISR(Ø1-Ø2,Ø9) are not set when GIFR (Ø1-Ø2,14) occur, these interrupts are ignored and the PE processor is not notified. All other interrupts are remembered and the PE processor will be notified when PL3-6M and/or G/LISR(ØØ,Ø3-Ø8,1Ø) are set.

RB(Response Bit) - a bit located in the MIAMM word and refers to a particular global message. If set, the message is for that PE.

SOC (Switch Over Command) - a command issued by the GEX on the "alternate" bus. When issued, all PEs will switch their inputs to the "alternate" bus.

SPDR (Serial-to-Parallel Data Register) - a shift register located in the "alternate" bus control logic that converts the serial SOC to parallel format.

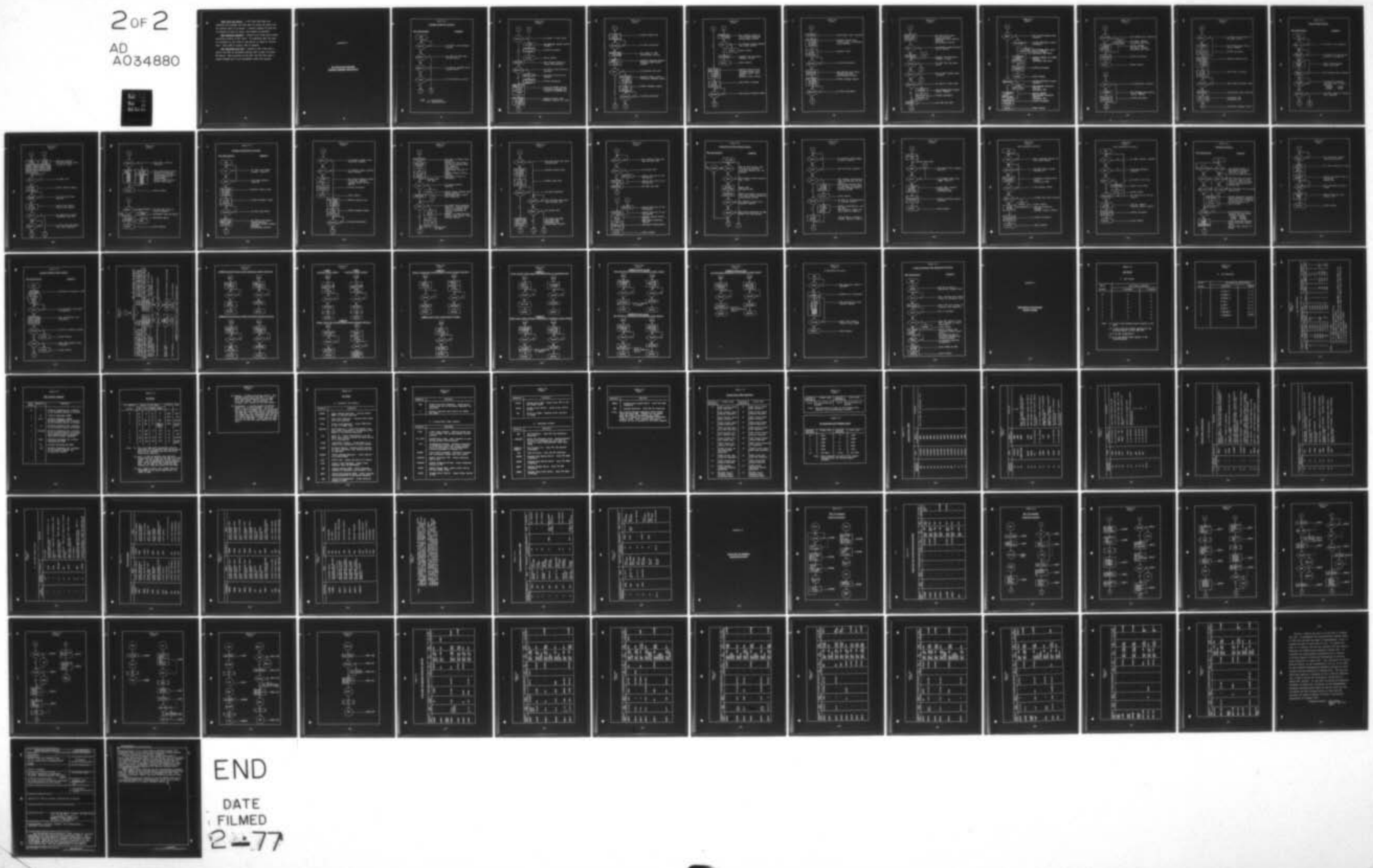
TACKR (Transfer Acknowledge Register) - a one-bit register that interfaces with the I-Bus. TACK is used to signify that data is available from a device sending data or that data has been received by a device that is receiving data.

TEN (Transfer Enable) - an I-Bus control signal issued by the PE processor to notify the BIU to begin message output and output MHS. When TEN is reset by the BIU, message transmission is complete.

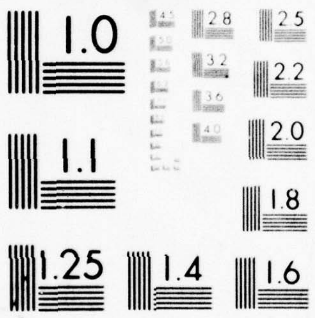
AD-A034 880   AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 9/2
DESIGN OF THE BUS INTERFACE UNIT FOR THE DISTRIBUTED PROCESSOR/--ETC(U)
DEC 76 R C SIMPSON

UNCLASSIFIED   GE/EE/76D-40   NL

2 OF 2
AD
A034880



END
DATE
FILMED
2-77


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

__TOER (Time Out Error)__ - a BIU flag that when set, signifies one attempt has been made to access PE memory but the address used is in error. A second attempt is made but if address is still in error, the request is aborted.

__TRQ (Transfer Request)__ - issued by an I-Bus device after receiving control of the I-Bus. It signifies that the data is available on the I-Bus or the device is ready to receive data. When TACK is issued, TRQ is removed.

__TTO (Transfer Time Out)__ - issued on the I-Bus when a device has sent an erroneous address when trying to access PE memory. When received by the BIU, the BIU will make a second attempt and if not successful abort the request.

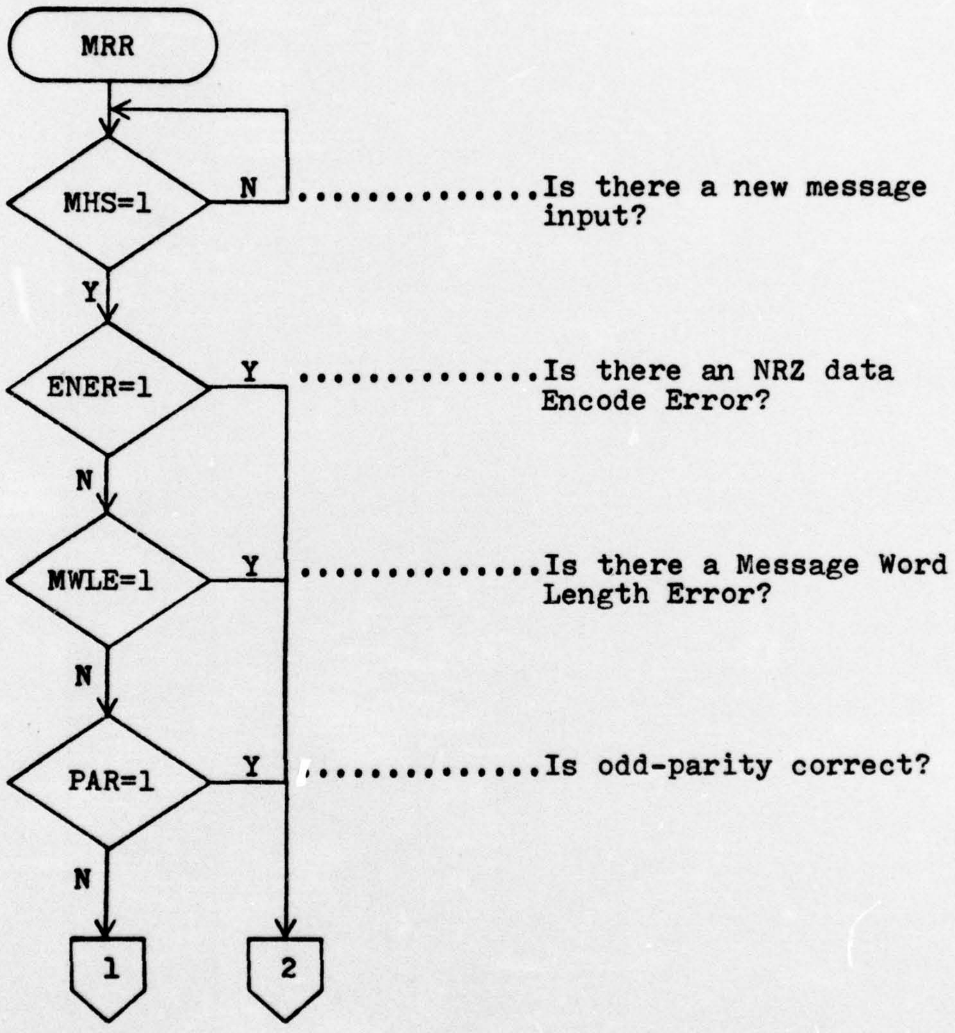
Appendix B

<u>BIU Functional Register Transfer Language Description</u>

Table B-1

Message Reception Routine

RTL Description | Comments



Is there a new message input?

Is there an NRZ data Encode Error?

Is there a Message Word Length Error?

Is odd-parity correct?

NOTE: 1 = True/Active
∅ = False/Inactive

Table B-1
(cont)

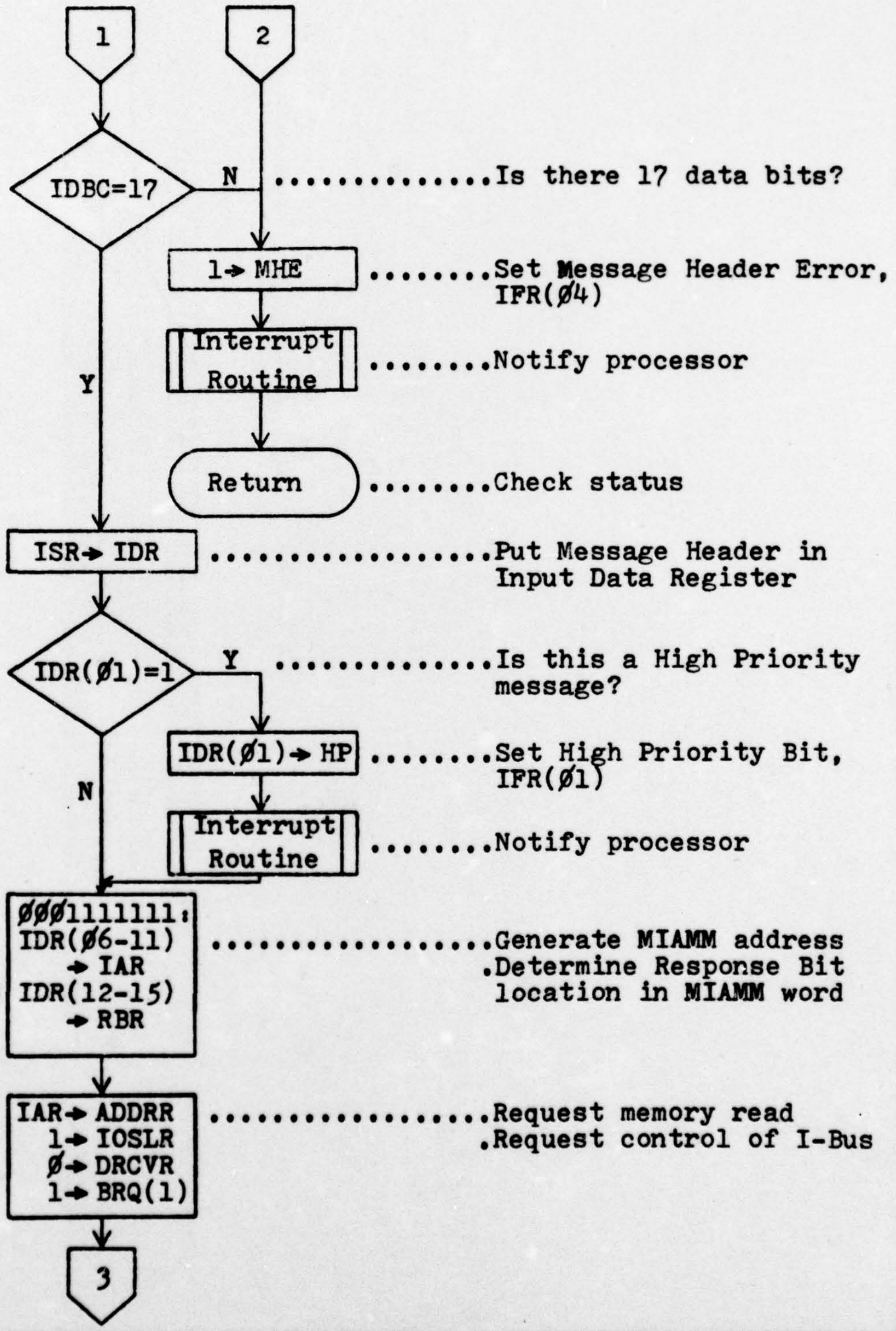
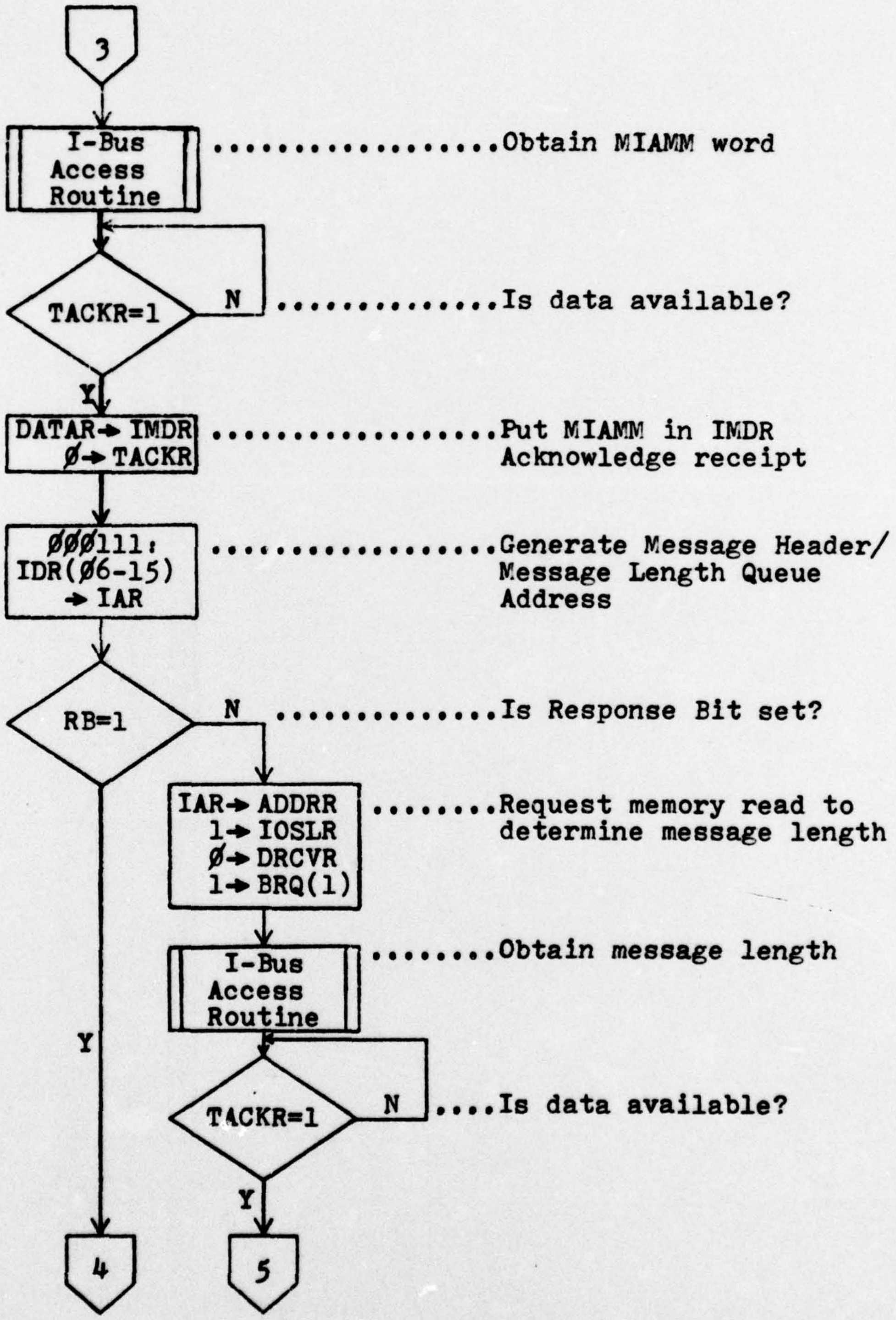
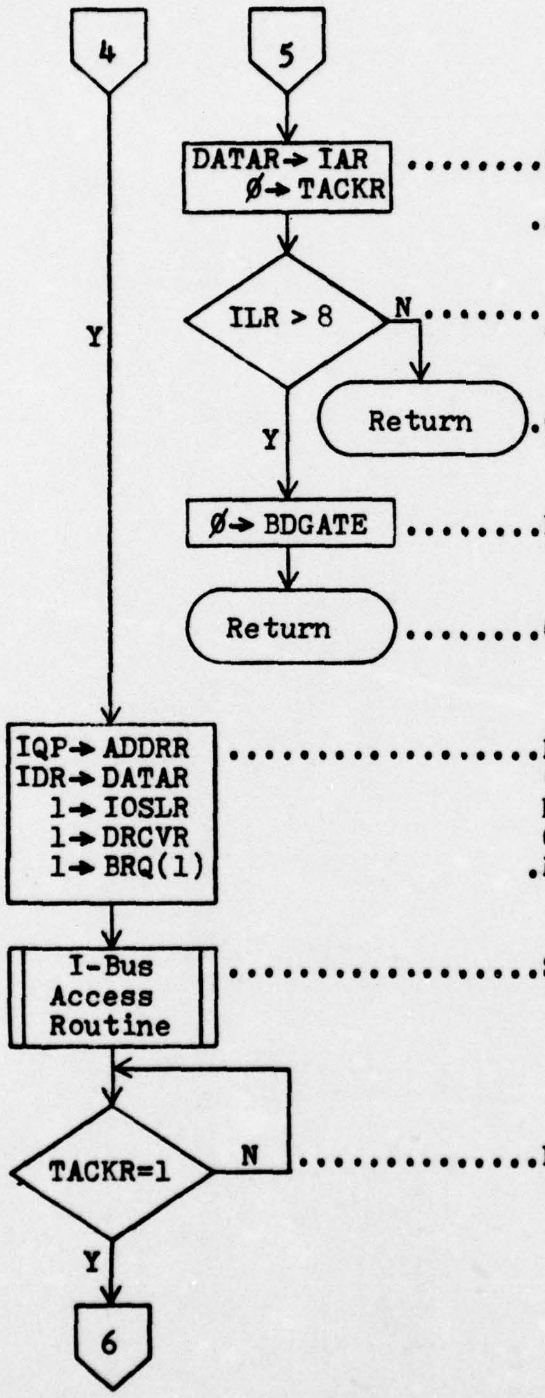


Table B-1
(cont)



3
I-Bus Access Routine
..................Obtain MIAMM word
TACKR=1
N
..............Is data available?
Y
DATAR→ IMDR
Ø→ TACKR
..................Put MIAMM in IMDR
Acknowledge receipt
ØØØ111:
IDR(Ø6-15)
→ IAR
..................Generate Message Header/
Message Length Queue
Address
RB=1
N
..............Is Response Bit set?
IAR→ ADDRR
1→ IOSLR
Ø→ DRCVR
1→ BRQ(1)
........Request memory read to
determine message length
I-Bus Access Routine
........Obtain message length
Y
TACKR=1
N
....Is data available?
Y
4
5

Table B-1
(cont)



........Put message length in Input Length Register
.Acknowledge receipt

........Is message length greater than eight words?

.Check status

........Disable Bus Dominance, ISR(Ø2), output

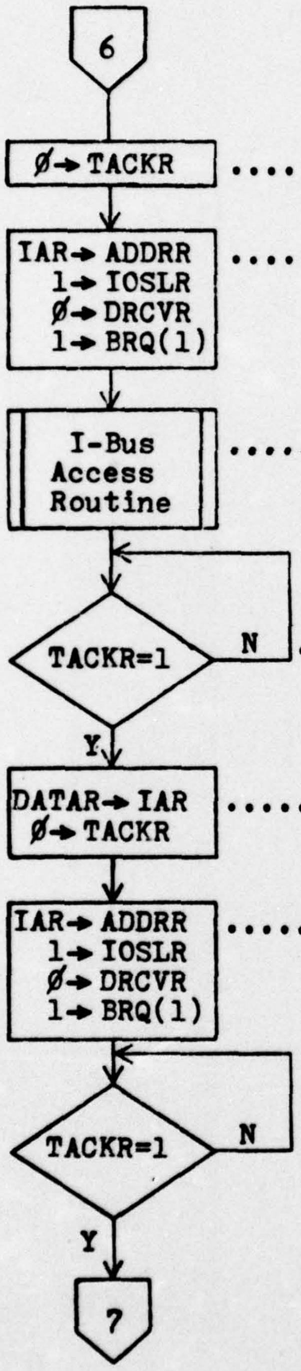
........Check status

.................Request memory write (Message Header into Message Header Input Queue)
.Request I-Bus control

.................Send data to memory

..............Has memory received data?

Table B-1
(cont)



..................Acknowledge send complete

..................Request memory read (obtain start of Message Input Buffer)

..................Request I-Bus

..............Is data available?

..................Put address into Input Address Register
.Acknowledge receipt

..................Obtain message length

.............Is data available?

Table B-1
(cont)

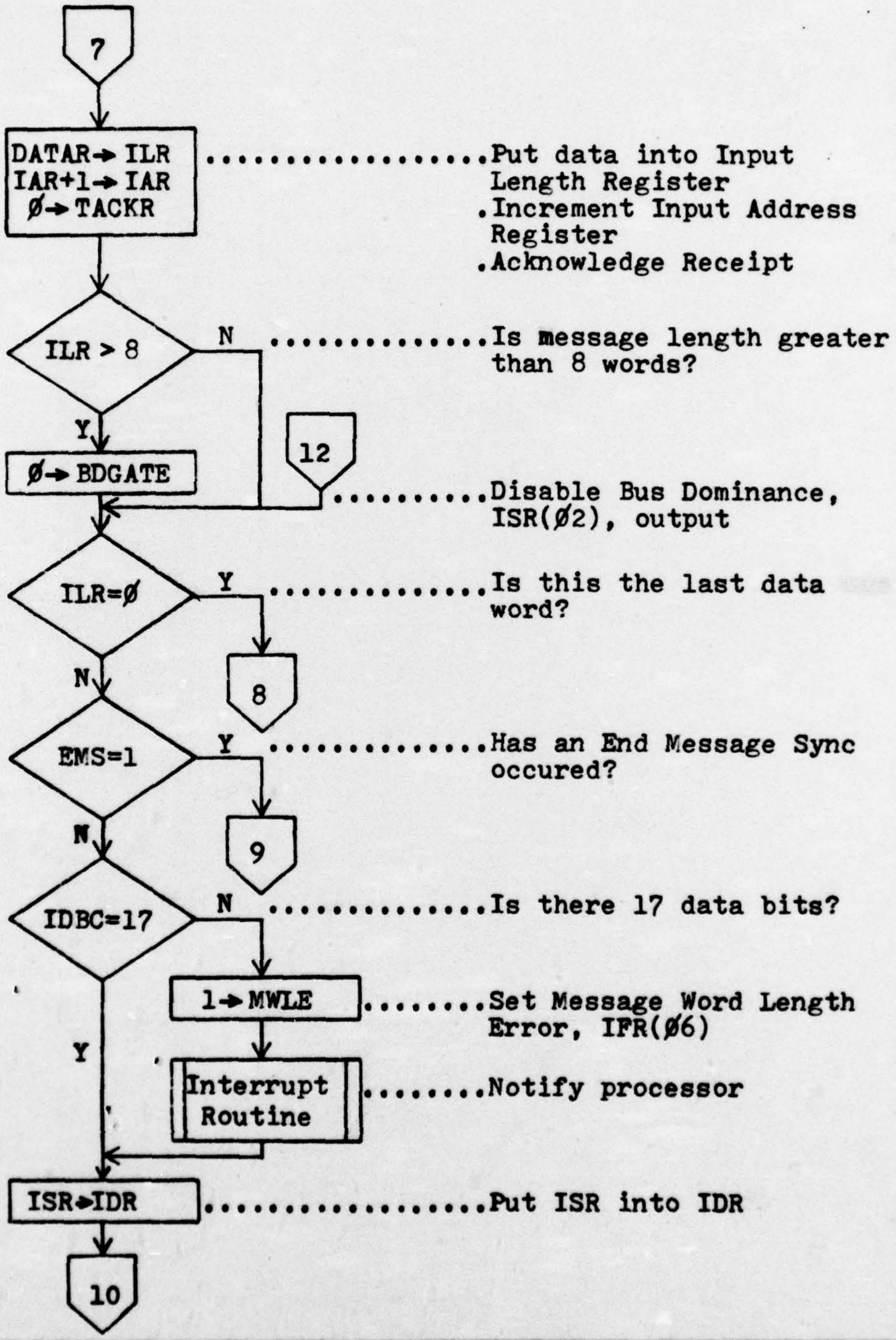
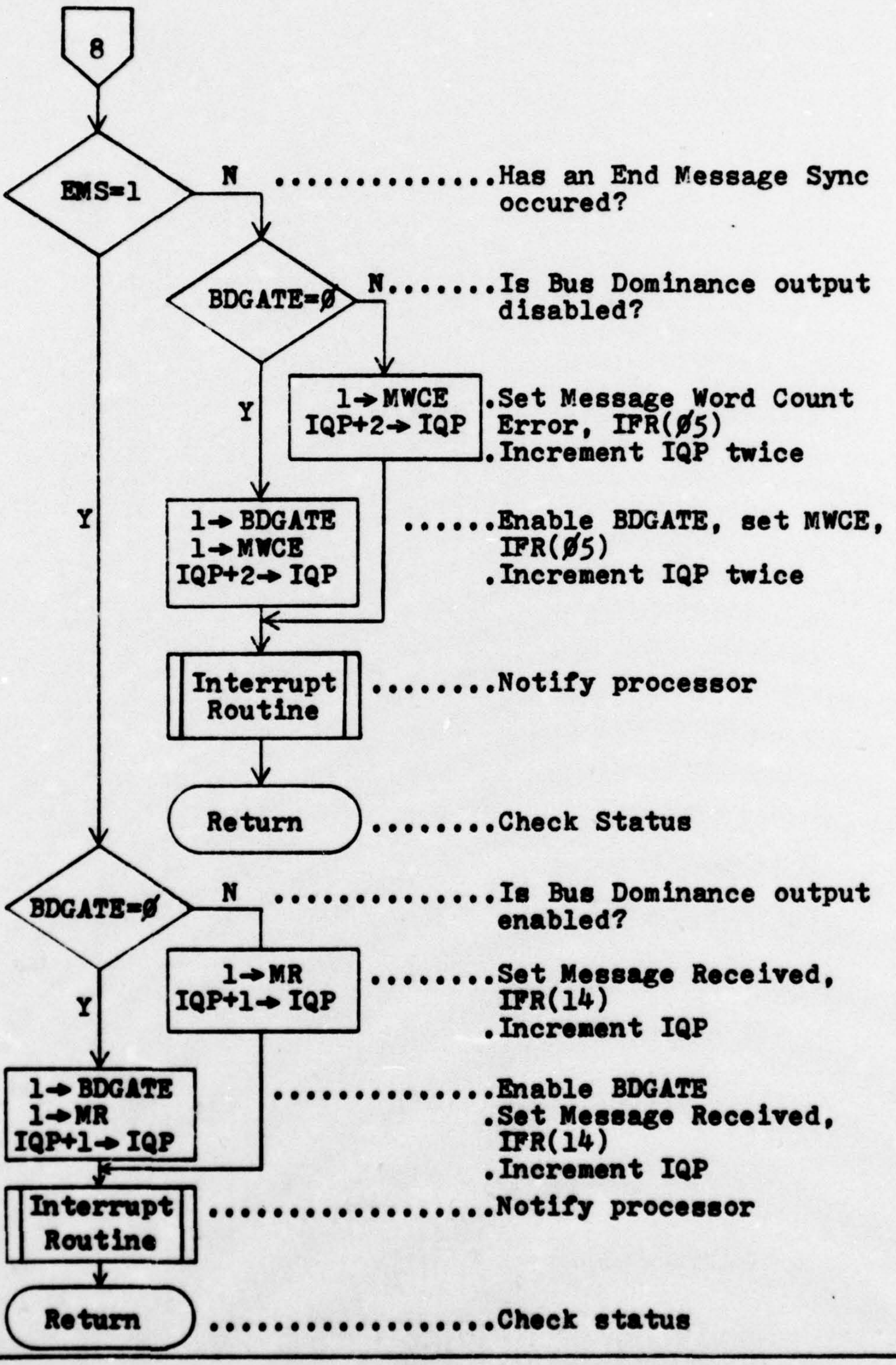

7
DATAR→ ILR
IAR+1→ IAR
Ø→ TACKR
.Put data into Input Length Register
.Increment Input Address Register
.Acknowledge Receipt
ILR > 8
N
Is message length greater than 8 words?
Y
Ø→ BDGATE
12
Disable Bus Dominance, ISR(Ø2), output
ILR=Ø
Y
Is this the last data word?
8
N
EMS=1
Y
Has an End Message Sync occured?
9
N
IDBC=17
N
Is there 17 data bits?
1→ MWLE
Set Message Word Length Error, IFR(Ø6)
Y
Interrupt Routine
Notify processor
ISR→IDR
Put ISR into IDR
10

Table B-1
(cont)



Has an End Message Sync occured?

Is Bus Dominance output disabled?

.Set Message Word Count Error, IFR(Ø5)
.Increment IQP twice

Enable BDGATE, set MWCE, IFR(Ø5)
.Increment IQP twice

Notify processor

Check Status

Is Bus Dominance output enabled?

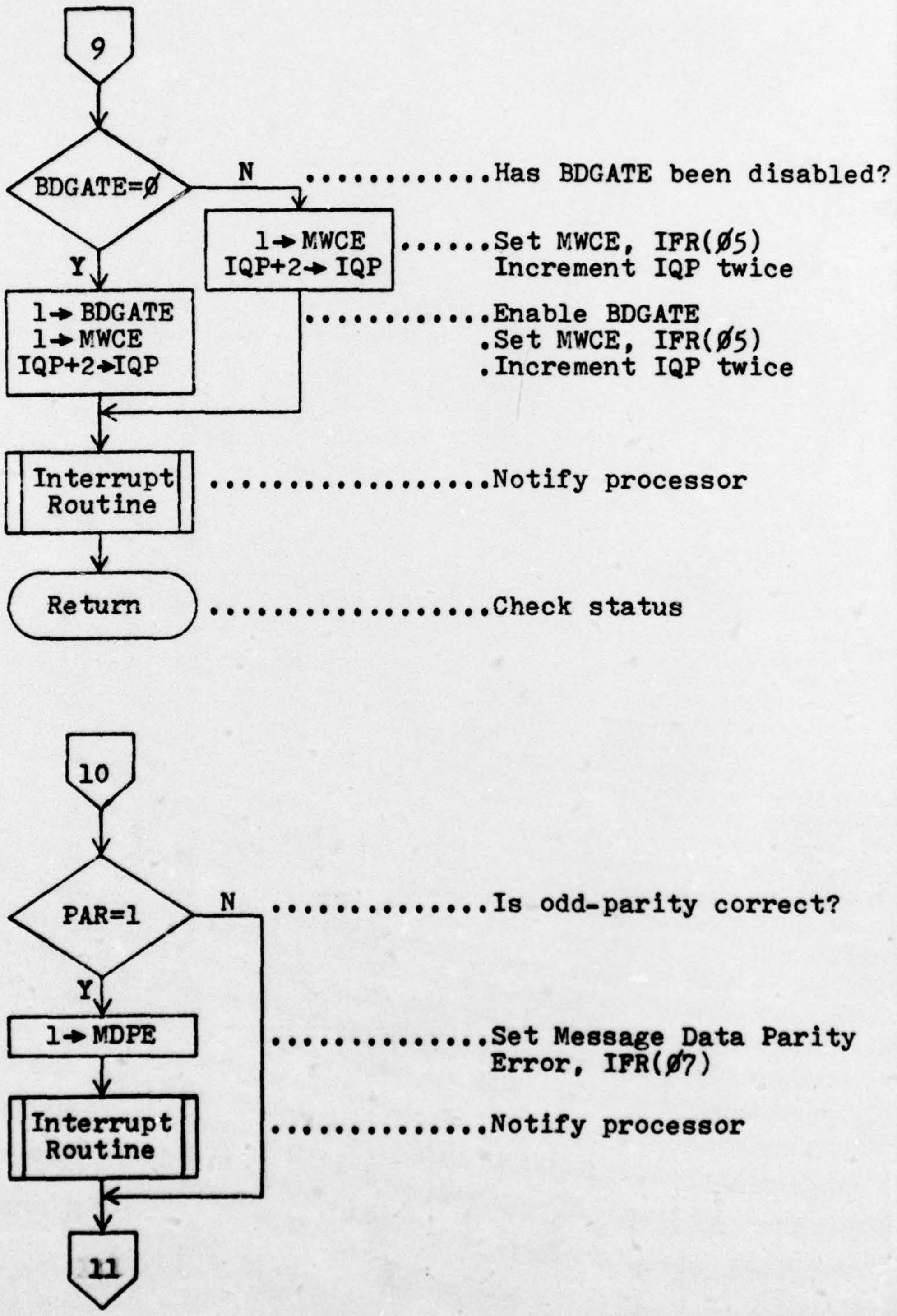
Set Message Received, IFR(14)
.Increment IQP

Enable BDGATE
.Set Message Received, IFR(14)
.Increment IQP

Notify processor

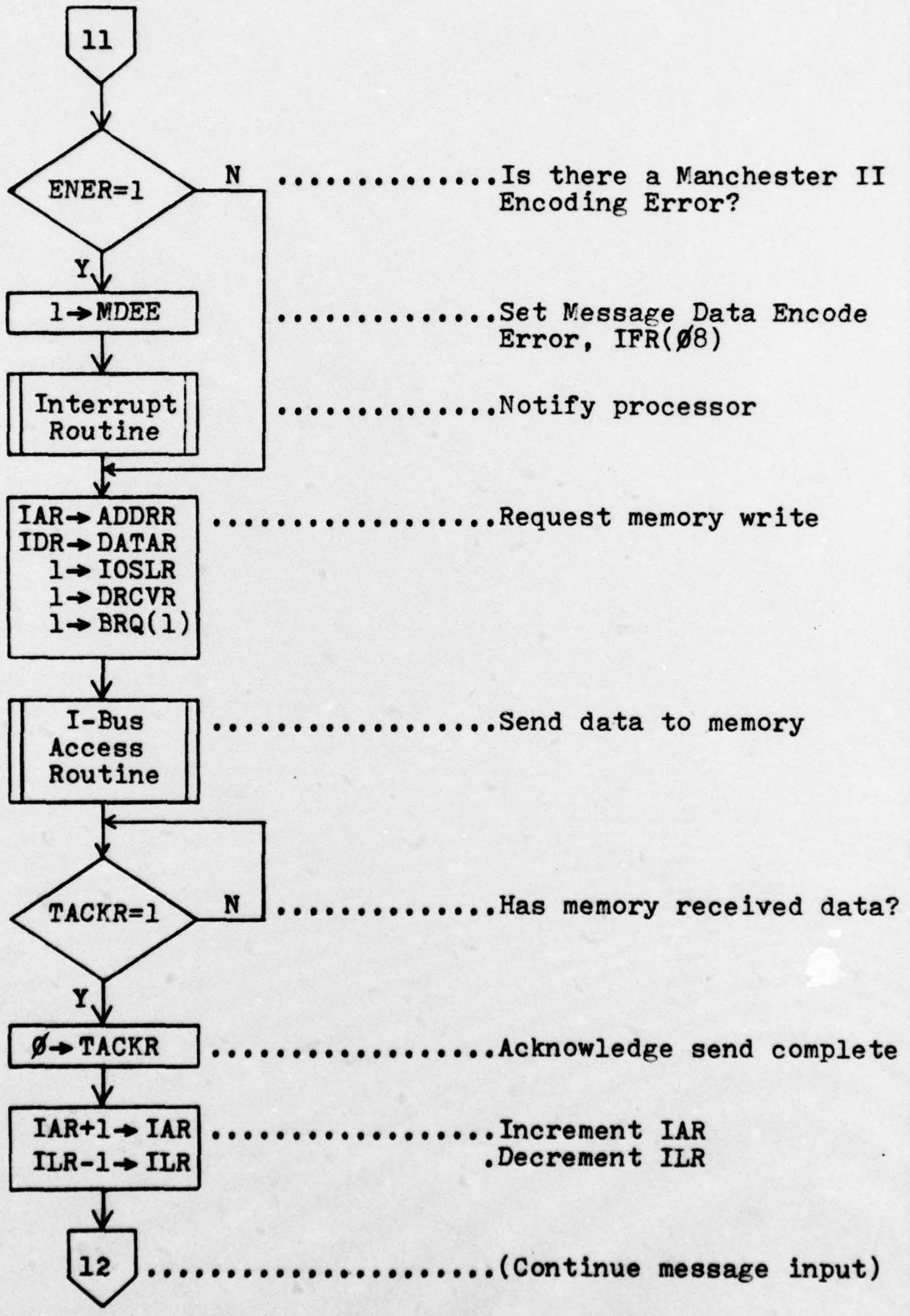
Check status

Table B-1
(cont)



............Has BDGATE been disabled?
......Set MWCE, IFR(Ø5)
Increment IQP twice
............Enable BDGATE
.Set MWCE, IFR(Ø5)
.Increment IQP twice
.................Notify processor
.................Check status
.............Is odd-parity correct?
.............Set Message Data Parity Error, IFR(Ø7)
.............Notify processor

Table B-1
(cont)



..............Is there a Manchester II Encoding Error?

..............Set Message Data Encode Error, IFR(Ø8)

..............Notify processor

..................Request memory write

..................Send data to memory

..............Has memory received data?

..................Acknowledge send complete

..................Increment IAR
.Decrement ILR

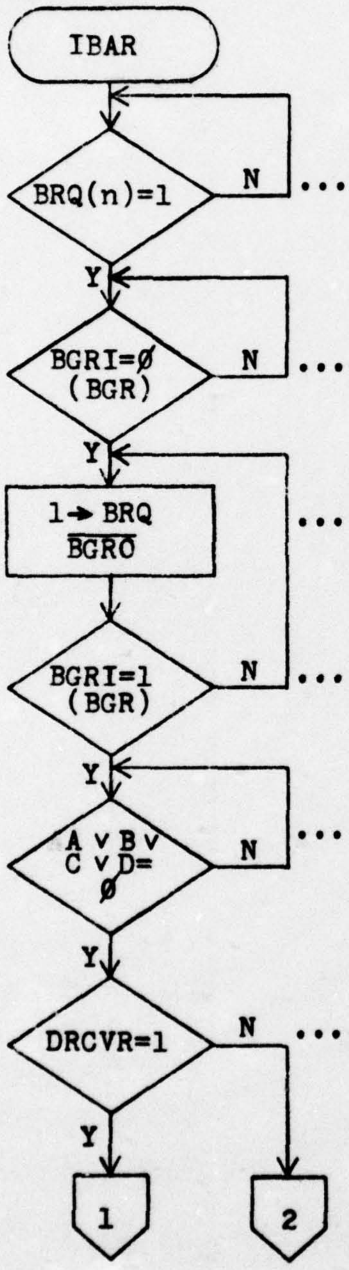
......................(Continue message input)

Table B-2

I-Bus Access Routine

RTL Description

Comments



..............Is there a Bus Request?

..............Is there an I-Bus Access request in progress?

..............Issue I-Bus Request
.Block Bus Grant

..............Has request been honored?

..............Is previous transfer complete (BIU or other device)?

A=TACKR	C=TRQ
B=TACK	D=IAK

..............Is this a send or receive data request?

Table B-2
(cont)

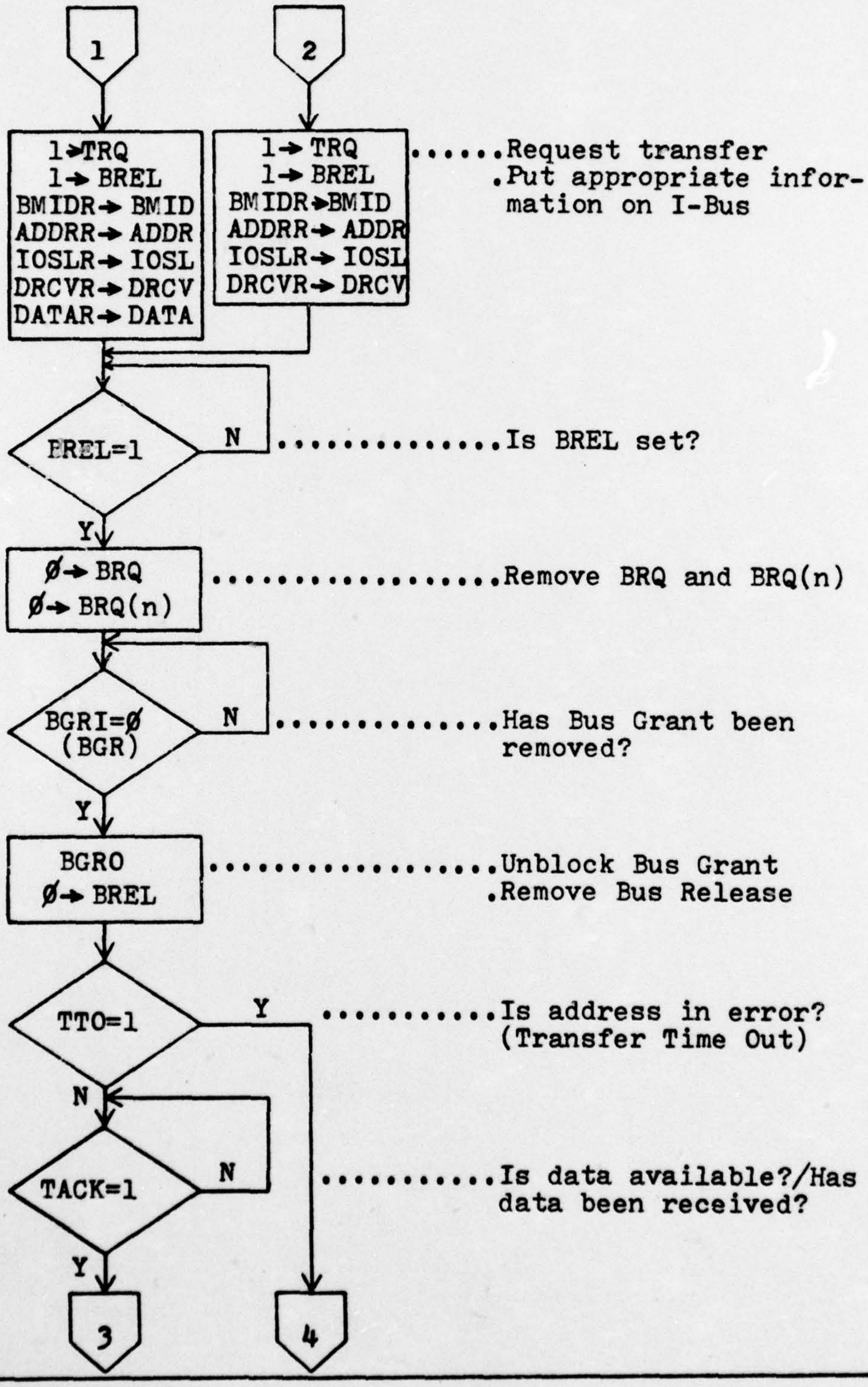

1
2
1→TRQ
1→BREL
BMIDR→BMID
ADDRR→ADDR
IOSLR→IOSL
DRCVR→DRCV
DATAR→DATA
1→TRQ
1→BREL
BMIDR→BMID
ADDRR→ADDR
IOSLR→IOSL
DRCVR→DRCV
......Request transfer
.Put appropriate information on I-Bus
BREL=1
N
Y
..............Is BREL set?
Ø→BRQ
Ø→BRQ(n)
.................Remove BRQ and BRQ(n)
BGRI=Ø (BGR)
N
Y
.............Has Bus Grant been removed?
BGRO
Ø→BREL
................Unblock Bus Grant
.Remove Bus Release
TTO=1
Y
N
...........Is address in error? (Transfer Time Out)
TACK=1
N
Y
...........Is data available?/Has data been received?
3
4

Table B-2
(cont)

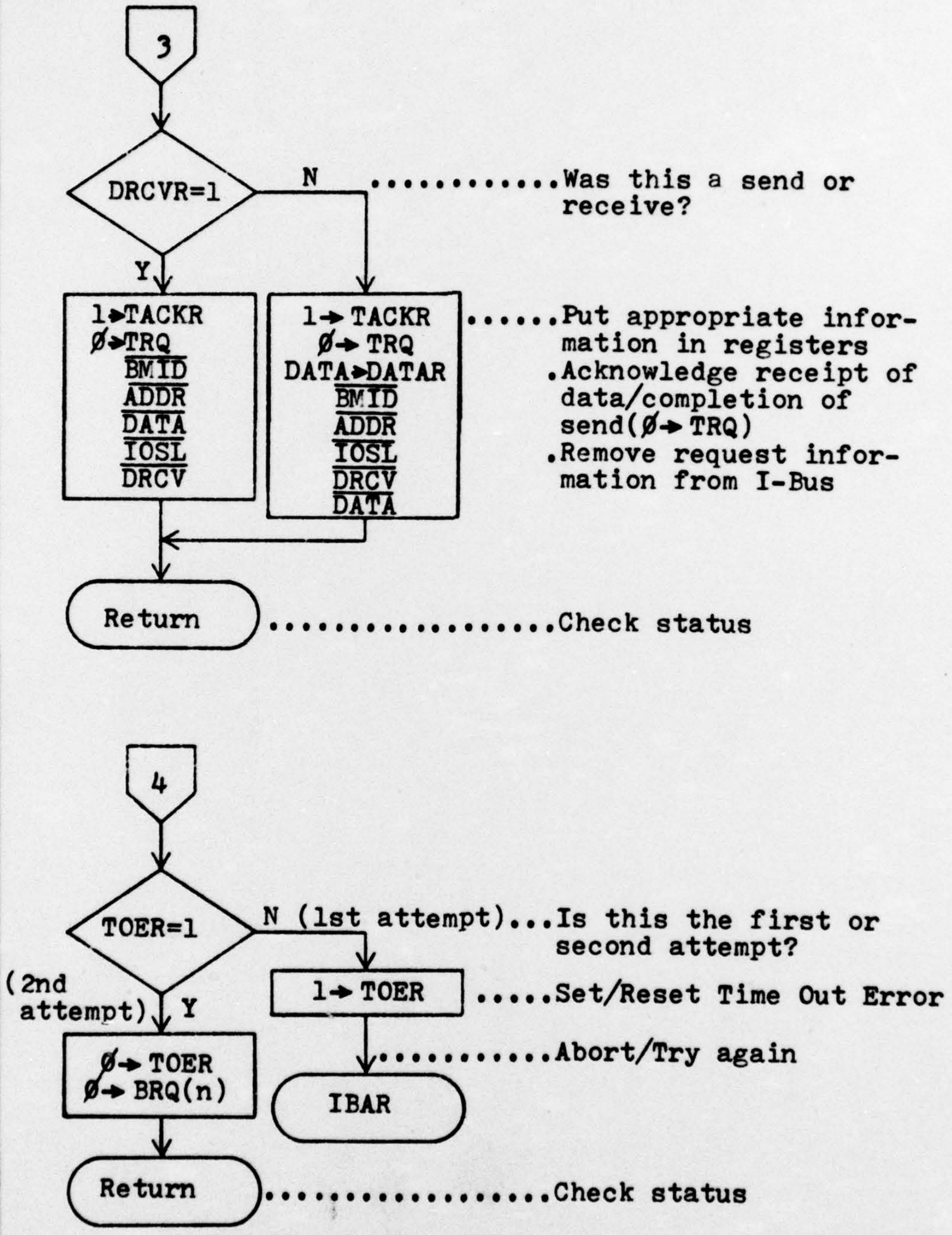
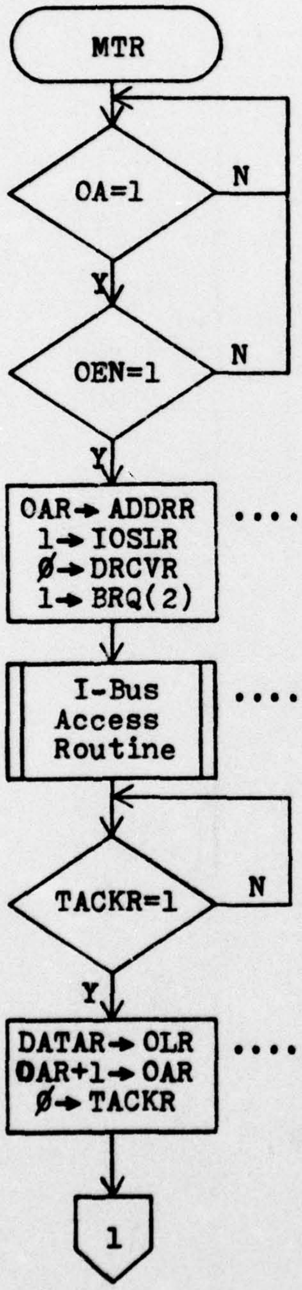

3
DRCVR=1
N
Y
............Was this a send or receive?
1→TACKR
Ø→TRQ
$\overline{\text{BMID}}$
$\overline{\text{ADDR}}$
$\overline{\text{DATA}}$
$\overline{\text{IOSL}}$
$\overline{\text{DRCV}}$
1→ TACKR
Ø→ TRQ
DATA→DATAR
$\overline{\text{BMID}}$
$\overline{\text{ADDR}}$
$\overline{\text{IOSL}}$
$\overline{\text{DRCV}}$
$\overline{\text{DATA}}$
......Put appropriate information in registers
.Acknowledge receipt of data/completion of send(Ø→ TRQ)
.Remove request information from I-Bus
Return
..................Check status
4
TOER=1
N (1st attempt)...Is this the first or second attempt?
(2nd attempt) Y
1→ TOER
.....Set/Reset Time Out Error
Ø→ TOER
Ø→ BRQ(n)
............Abort/Try again
IBAR
Return
..................Check status

Table B-3

Message Transmission Routine

RTL Description | Comments



..............Is there an output request, ISR(14)?

..............Is Output Enabled, ISR(15)?

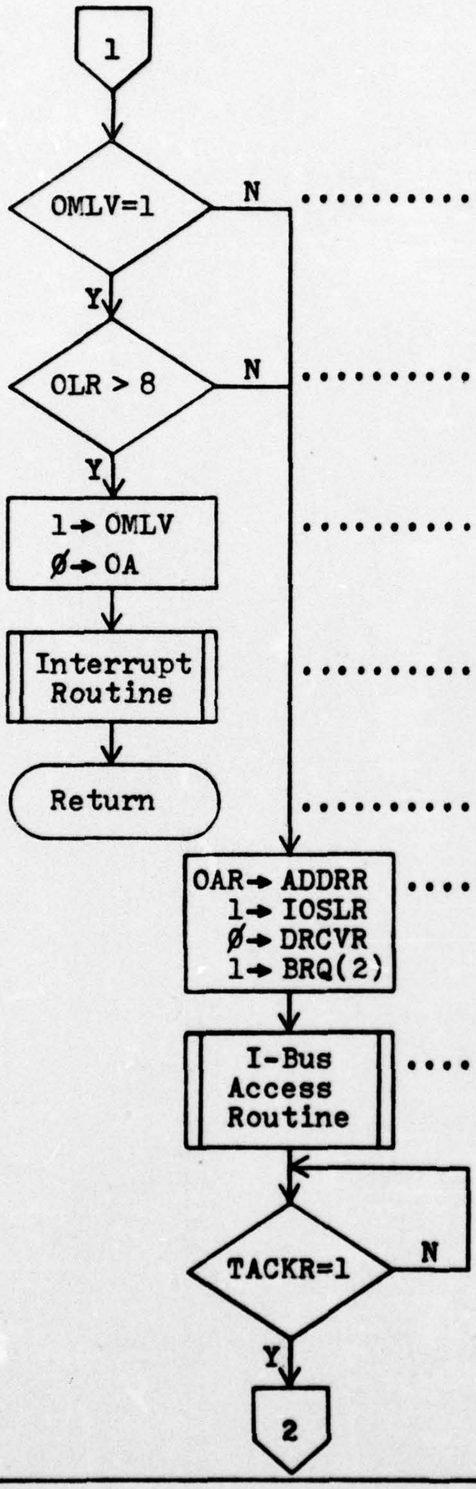
..................Request memory read

..................Obtain message length

..............Is data available?

..................Put data into Output Length Register
.Increment Output Address Register
.Acknowledge receipt

Table B-3
(cont)



..............Is message length being enforced, ISR(Ø3)

..............Is message length greater than 8 words?

..............Set Output Message Length Violation, IFR(Ø3)
.Disable Output Active, ISR(14)

..............Notify processor

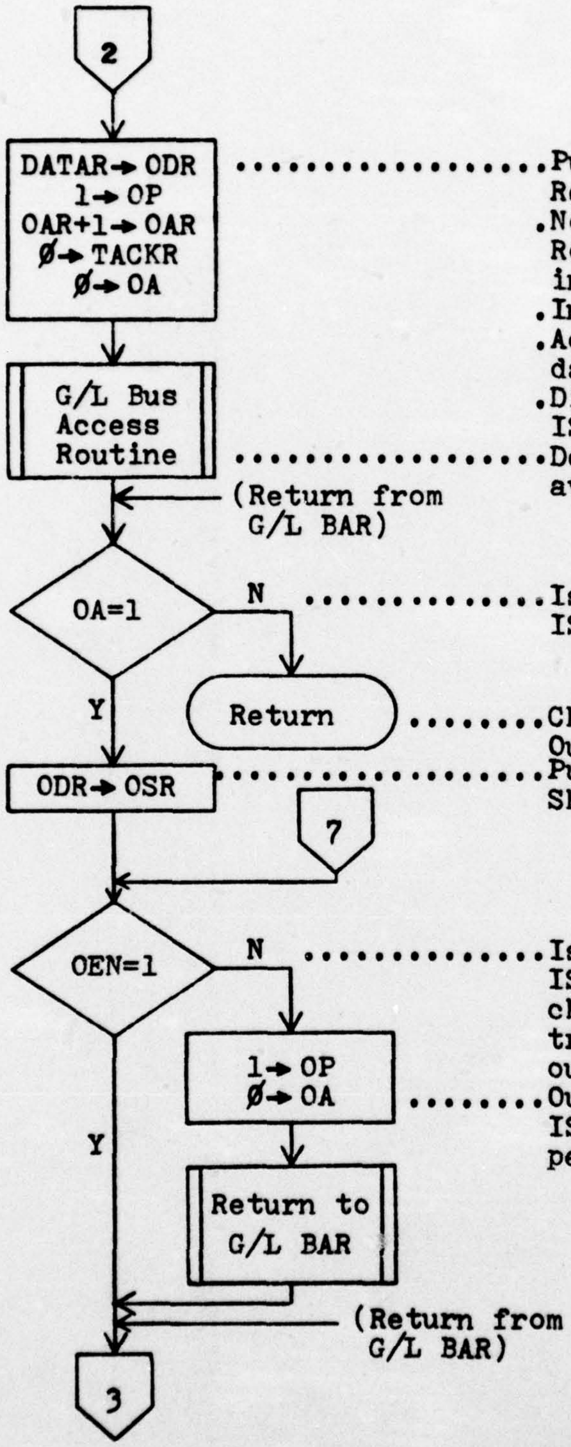
..............Check status

........Request memory read

........Obtain Message Header

....Is data available?

Table B-3
(cont)



.Put data in Output Data Register
.Notify G/L Bus Access Routine of Output Pending, ISR(11)
.Increment OAR
.Acknowledge receipt of data
.Disable Output Active, ISR(14)

.Determine if bus is available

.Is Output Active, ISR(14)?

.Check status (Return when Output Active)

.Put header into Output Shift Register

.Is output still enabled, ISR(15)? (Continuously check for program controlled disabling of output)

.Output no longer active, ISR(14) (But output still pending ISR(11)

Table B-3
(cont)

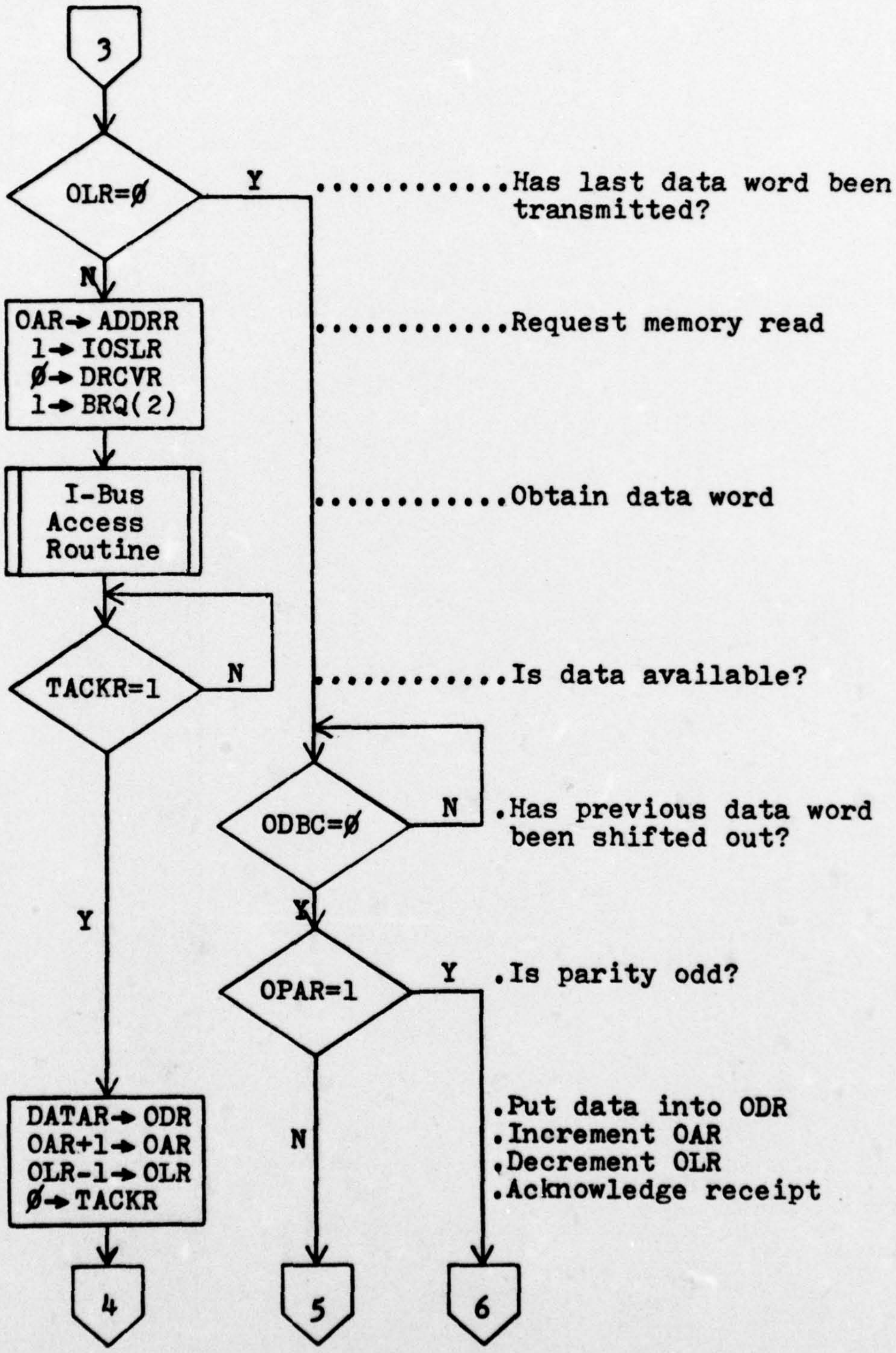
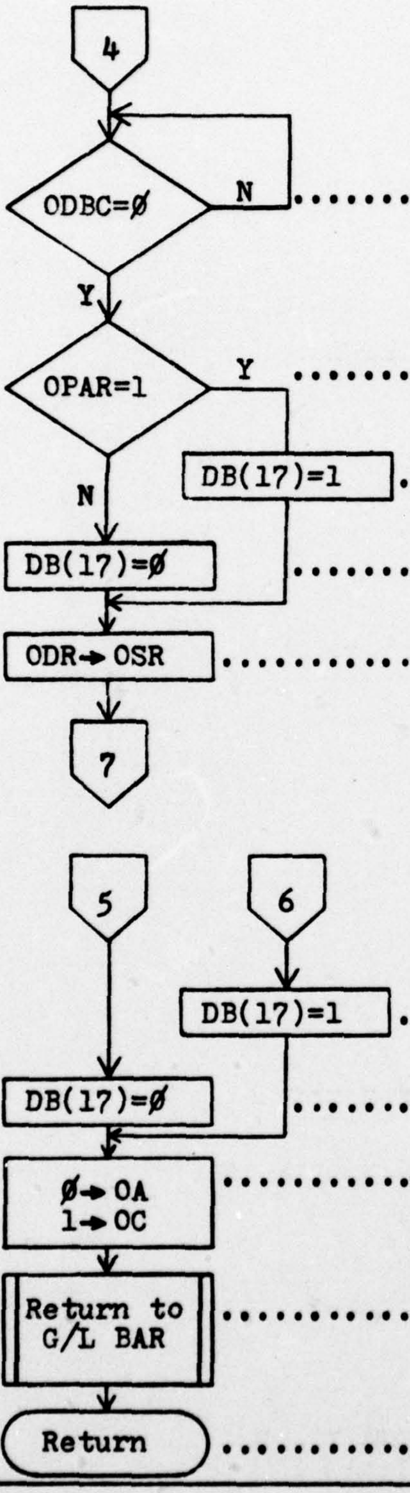


Table B-3
(cont)


4
ODBC=∅
N
Y
OPAR=1
Y
N
DB(17)=1
DB(17)=∅
ODR→OSR
7
5
6
DB(17)=1
DB(17)=∅
∅→OA
1→OC
Return to G/L BAR
Return

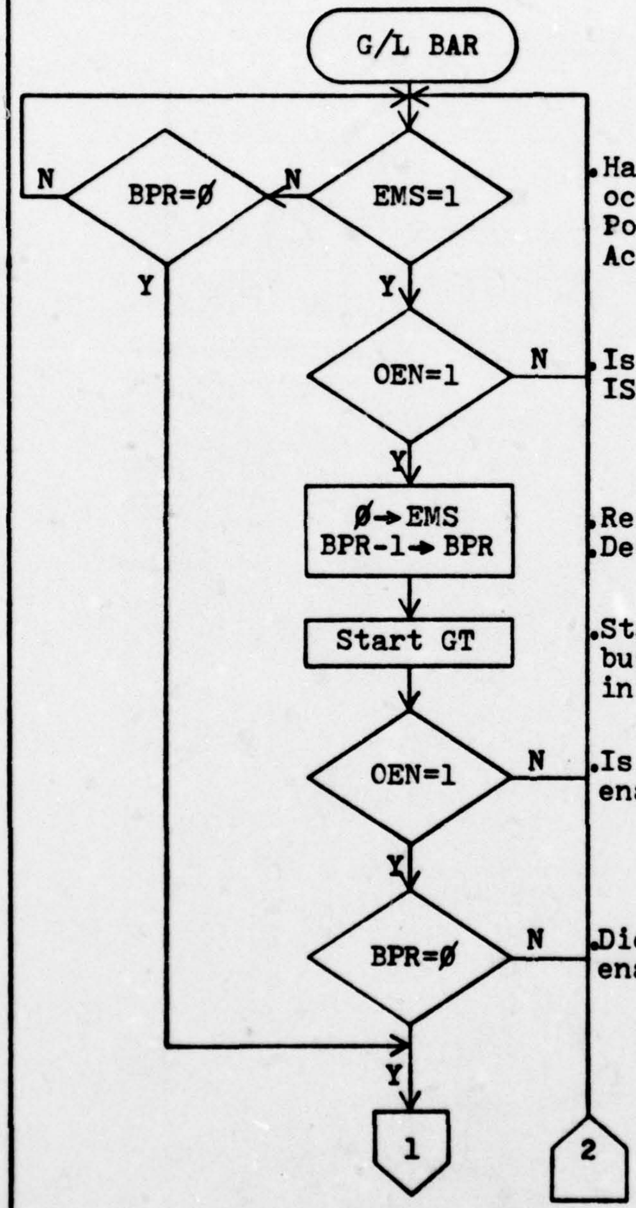
Has previous data word been shifted out?
Is parity odd?
Append Data Bit 17 for odd-parity
Append Data Bit 17 for odd-parity
Put ODR into OSR
Append Data Bit 17 for odd-parity
Append Data Bit 17 for odd-parity
Disable Output Active, ISR(14)
Set Output Complete, IFR(15)
Terminate Transmission
Check status

Table B-4

Global/Local Bus Access Routine

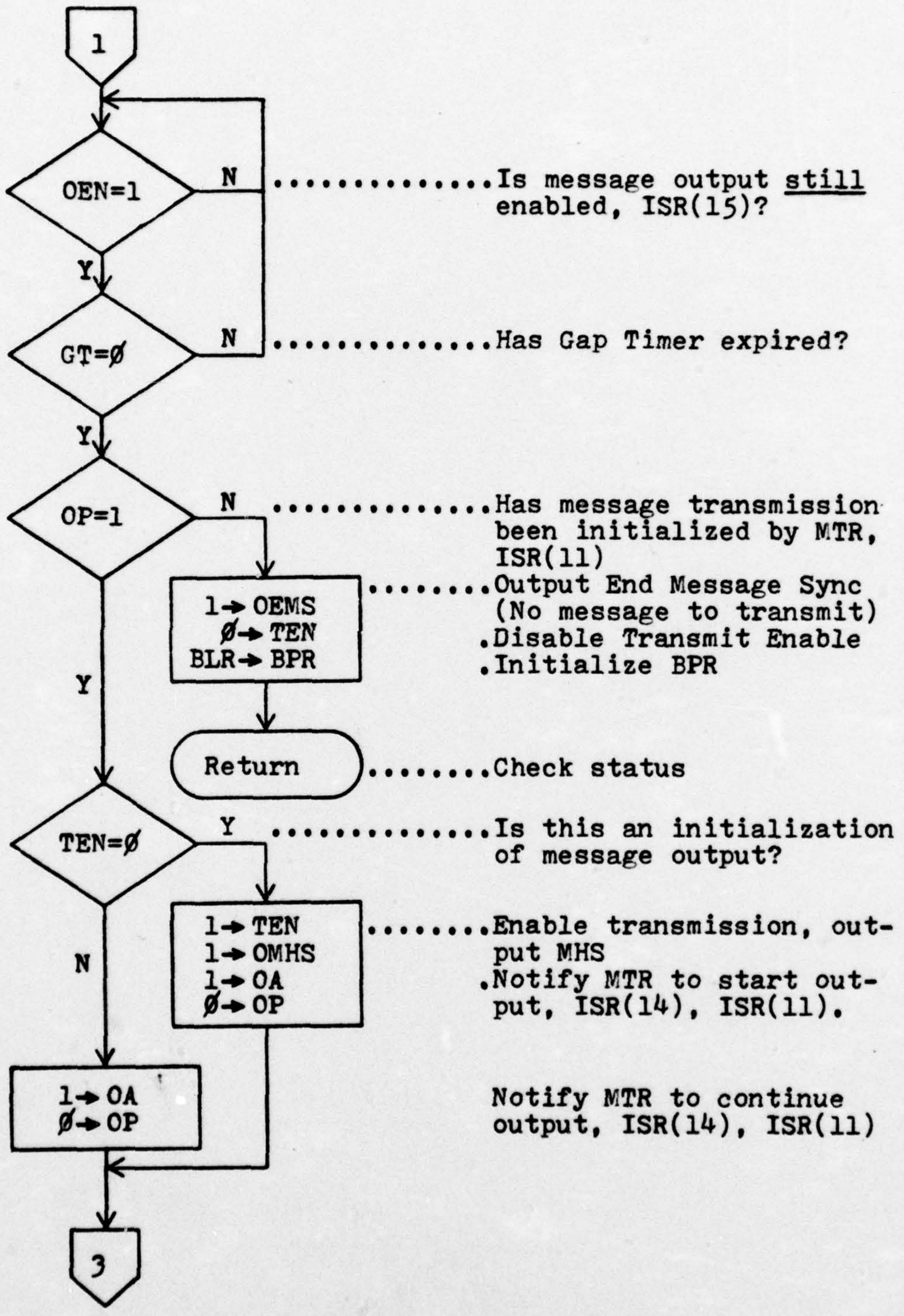
RTL Description

Comments



- Has an End Message Sync occurred or has Bus Position enabled G/L Bus Access?
- Is message Output Enabled, ISR (15)?
- Reset EMS
- Decrement BPR
- Start Gap Timer (delay for bus skew between messages), initialized by EMS
- Is message output still enabled, ISR(15)
- Did above decrement of BPR enable G/L Bus Access?

Table B-4
(cont)



..............Is message output <u>still</u> enabled, ISR(15)?

..............Has Gap Timer expired?

..............Has message transmission been initialized by MTR, ISR(11)

........Output End Message Sync (No message to transmit)

.Disable Transmit Enable

.Initialize BPR

........Check status

..............Is this an initialization of message output?

........Enable transmission, output MHS

.Notify MTR to start output, ISR(14), ISR(11).

Notify MTR to continue output, ISR(14), ISR(11)

Table B-4
(cont)

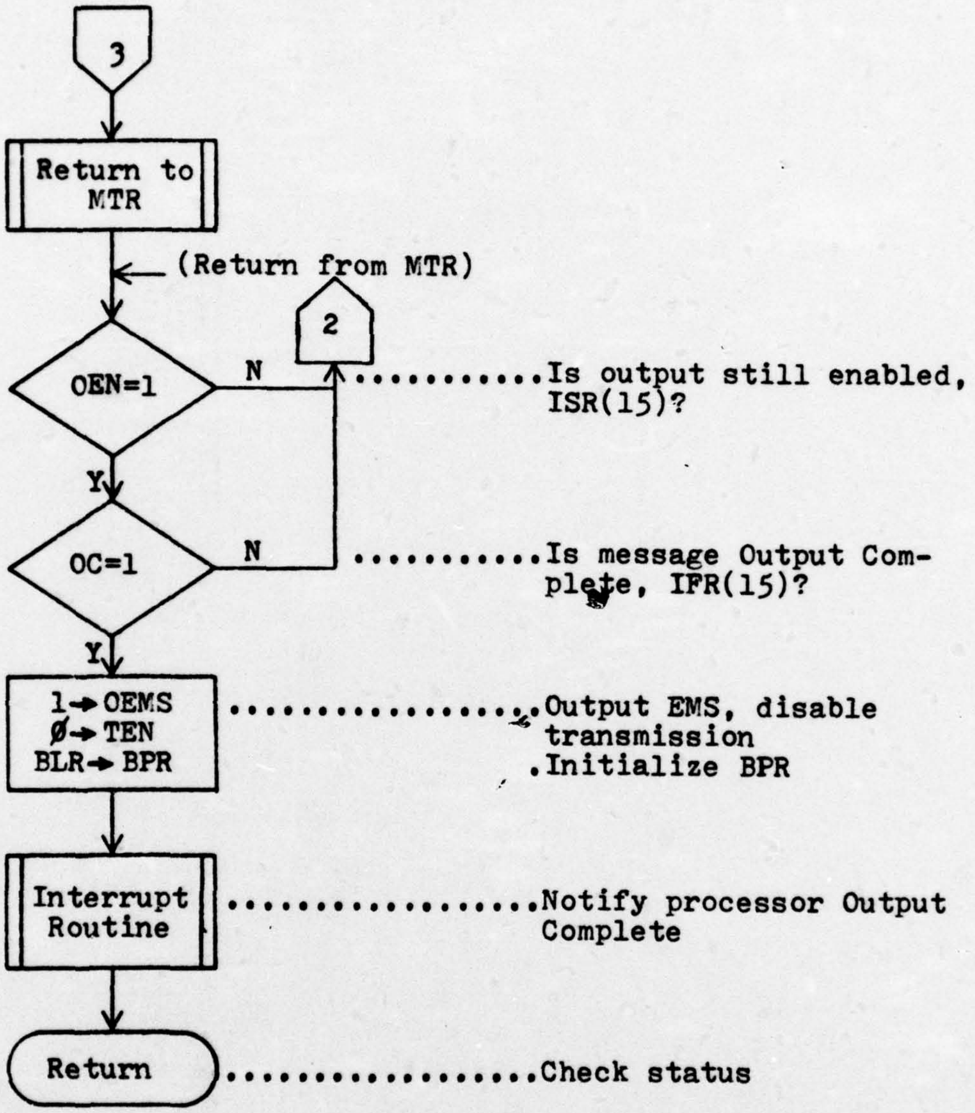
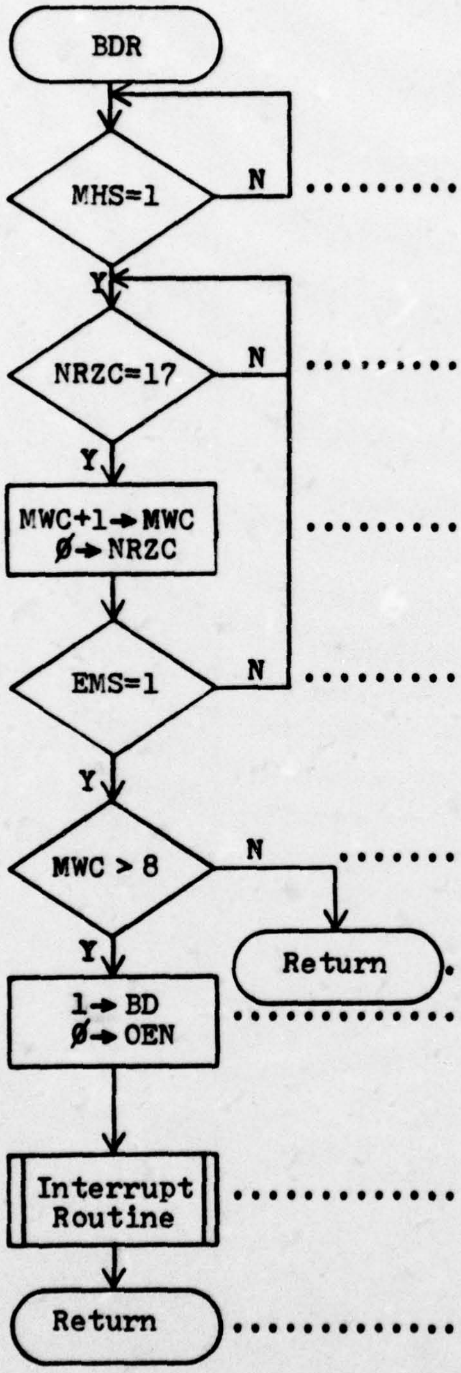

3
Return to MTR
(Return from MTR)
2
OEN=1
N
Y
OC=1
N
Y
1→OEMS
Ø→TEN
BLR→BPR
Interrupt Routine
Return
Is output still enabled, ISR(15)?
Is message Output Complete, IFR(15)?
Output EMS, disable transmission
Initialize BPR
Notify processor Output Complete
Check status

Table B-4
(cont)

(Bus Dominance Routine)



..............Has a message started on Bus (start NRZC)?

..............Has NRZ Clock counter reached 17?

..............Increment Message Word Count
.Initialize NRZC

..............Has message ended?

............Is MWC more than 8 words?

.....Check status

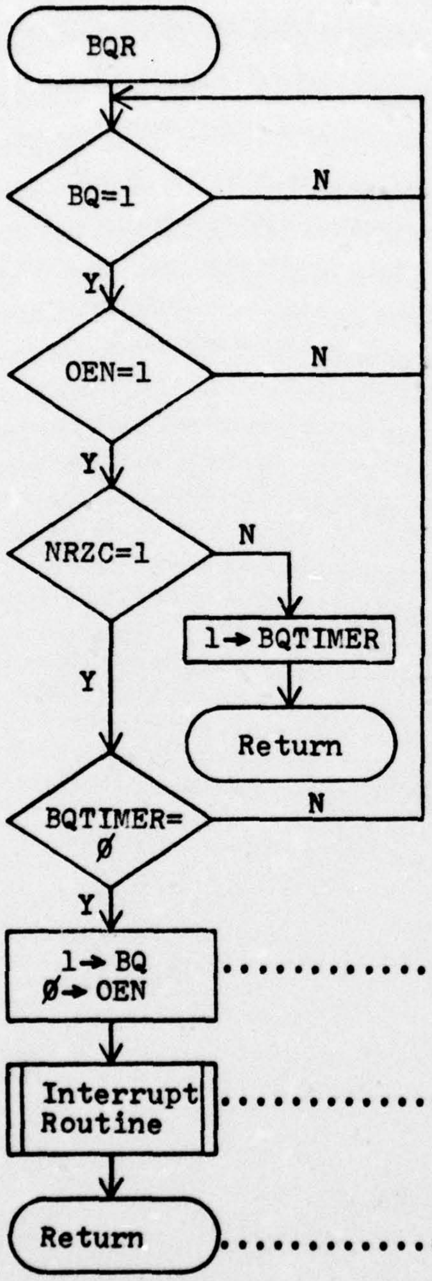
..................Set Bus Dominance, IFR(Ø2)
.Disable output, ISR(15)

..................Notify processor

..................Check status

Table B-4
(cont)

(Bus Quiescence Routine)



......Is BQWT enabled, ISR(Ø1)?

......Is Output Enabled, ISR(15)?

......Is data on G/L Bus?

......Reset 50 μs Timer

......Check status

......No data on G/L Bus for 50 μs?

..................Set BQ, IFR(Ø1)
.Disable output, ISR(15)

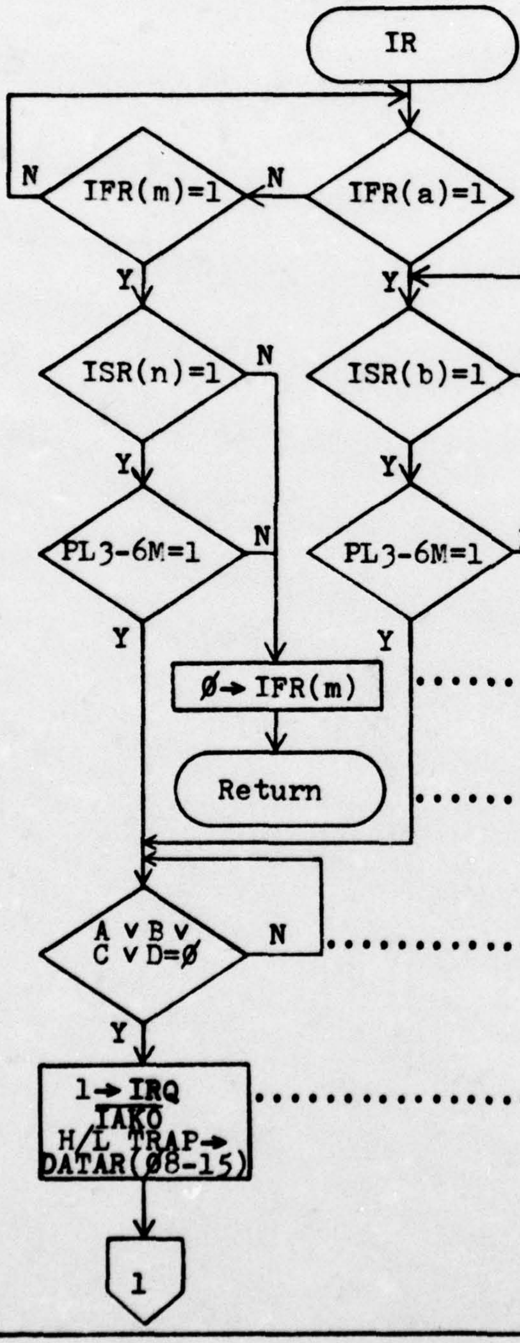
..................Notify processor

..................Check status

Table B-5

Interrupt Routine

RTL Description — Comments



...Has Enable/Disable or Arm/Disarm Interrupt Stimulus occurred? (m=1,2,14; a=Ø,3→8,15)

..Has processor set mask? (n=1,2,9; b=Ø,3→8,10). (Service other routines, return when ISR and/or PL3-6M set)

..Has processor set Priority Level Mask?

.........Remove interrupt stimulus (Enable/Disable remembered if and only if ISR and PL3-6M initially set)

.........Check status

.............Is Maintenance Inhibit enabled and last transmission complete?

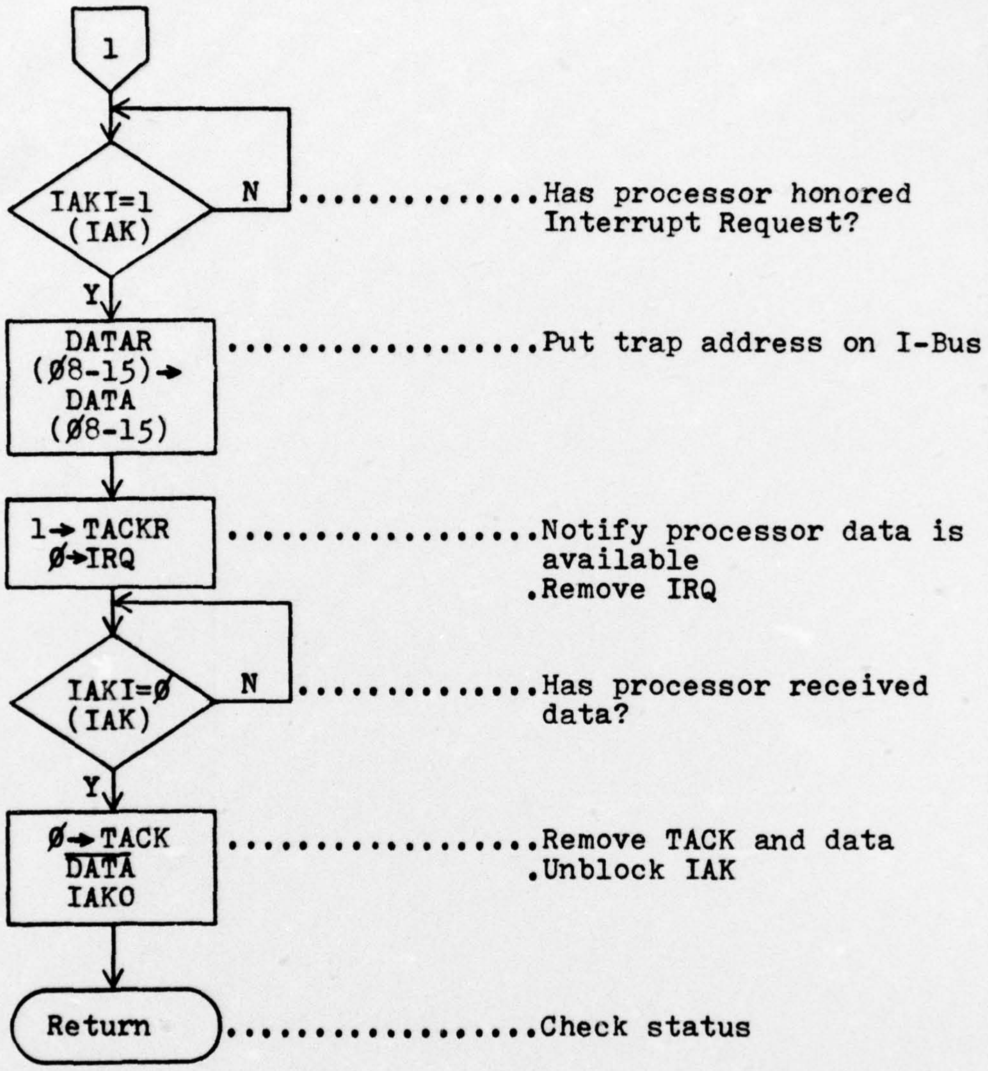
A=INHB	C=TACK
B=TRQ	D=TACKR

..................Issue Interrupt Request

.Block Interrupt Acknowledge

.Put H/L Trap address in DATAR

Table B-5
(cont)

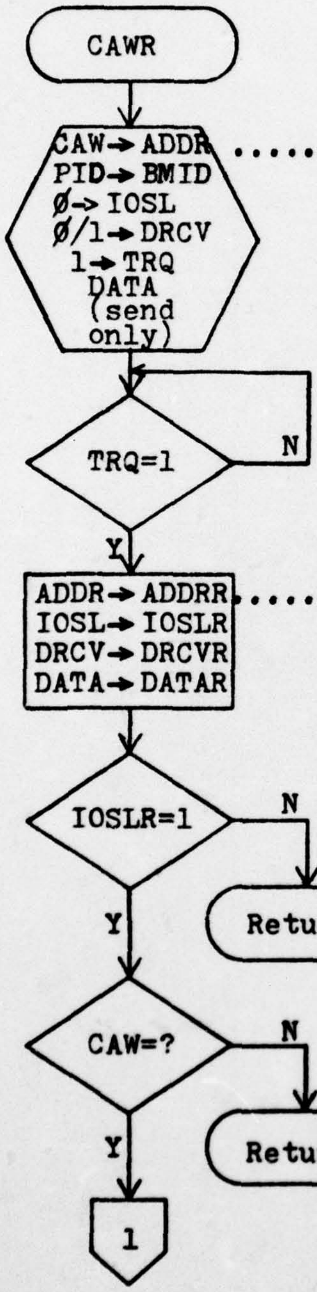


1
IAKI=1 (IAK)
N
Y
Has processor honored Interrupt Request?
DATAR (Ø8-15)→ DATA (Ø8-15)
Put trap address on I-Bus
1→TACKR
Ø→IRQ
Notify processor data is available
.Remove IRQ
IAKI=Ø (IAK)
N
Y
Has processor received data?
Ø→TACK
$\overline{\text{DATA}}$
IAKO
Remove TACK and data
.Unblock IAK
Return
Check status

Table B-6

Command Address Word Routine

RTL Description | Comments



..................Processor initiates a CAW

.............Has processor requested a transfer?

..................Put information into registers

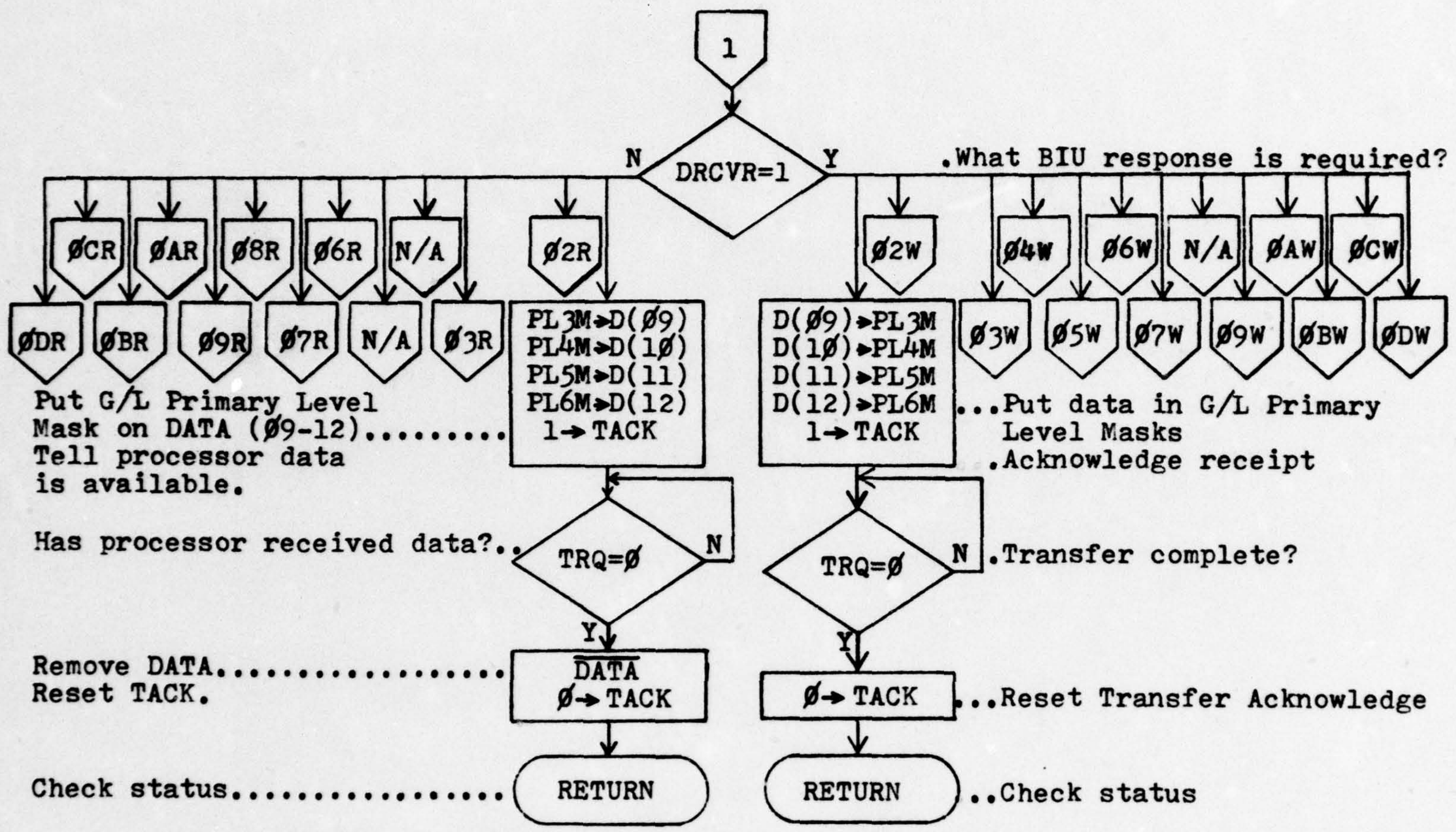
.............Is this a memory access?

........Check status

.............Does CAW require a BIU response?

........Check status

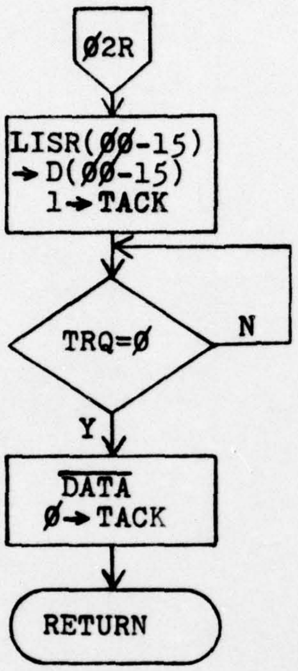
Table B-6
(cont)

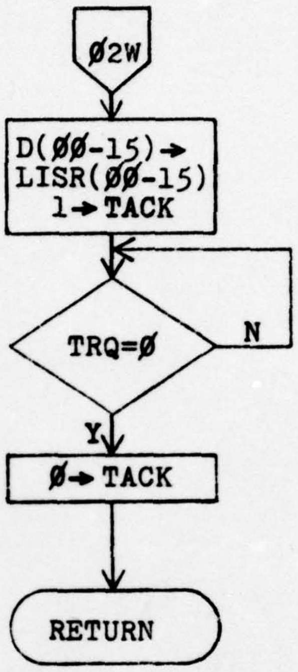


CAWØØR/W-Input/Output G/L Primary Level Interrupt Mask

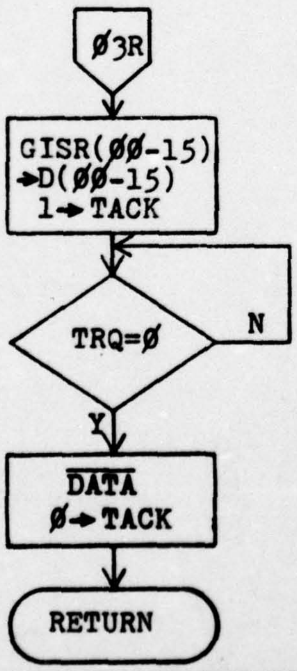
Table B-6
(cont)

CAWØ2R/W-Input/Output Local Interrupt Status Register


Ø2R
LISR(ØØ-15) → D(ØØ-15) 1 → TACK
TRQ=Ø
N
Y
$\overline{\text{DATA}}$ Ø → TACK
RETURN



CAWØ3R/W-Input/Output Global Interrupt Status Register


Ø3R
GISR(ØØ-15) → D(ØØ-15) 1 → TACK
TRQ=Ø
N
Y
$\overline{\text{DATA}}$ Ø → TACK
RETURN

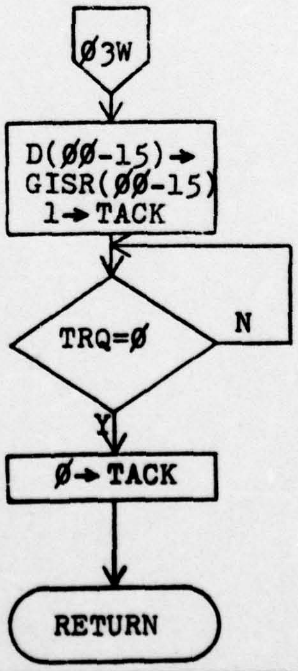
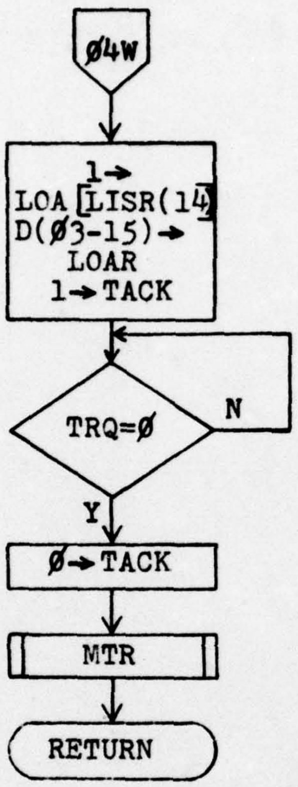
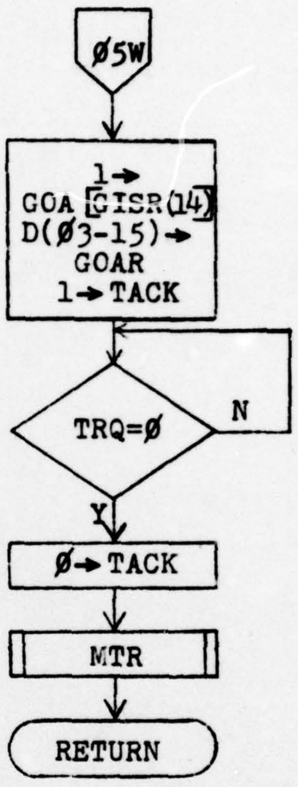


Table B-6
(cont)

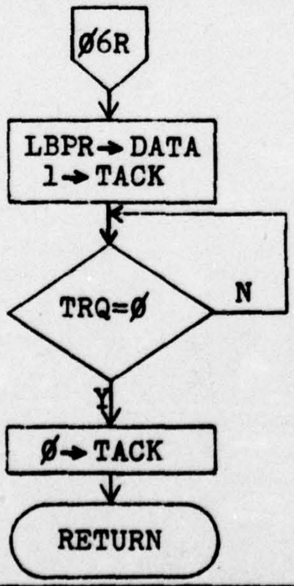
CAWØ4W
Activate Local Output



CAWØ5W
Activate Global Output



CAWØ6R/W
Output Local Bus Control/Input Local Present Position



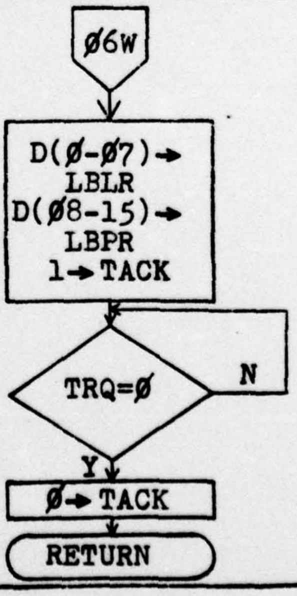
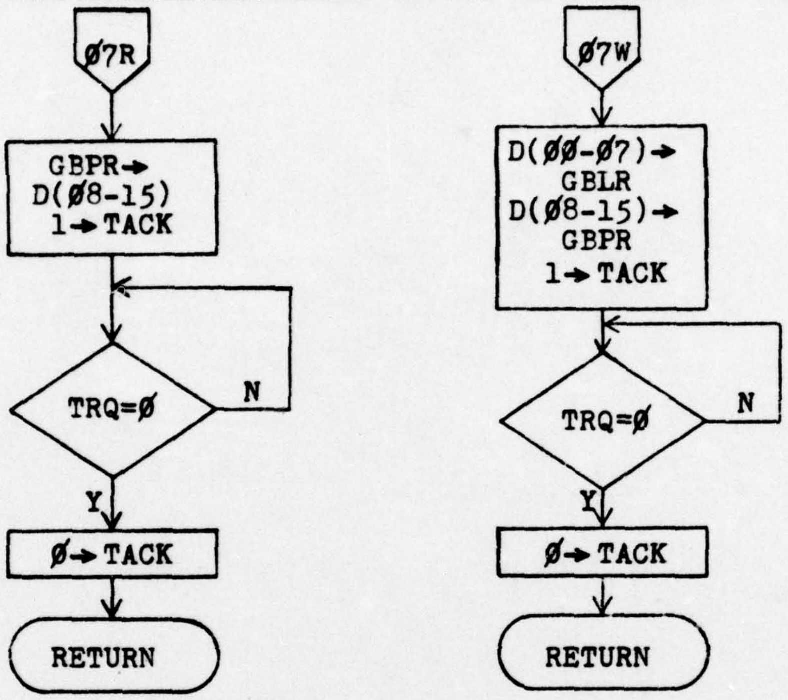


Table B-6
(cont)

CAWØ7R/W
Output Global Bus Control/Input Global Present Position



CAWØ8R-Input Local Input Queue Pointer

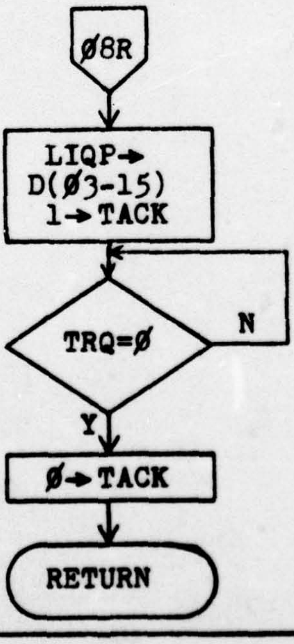
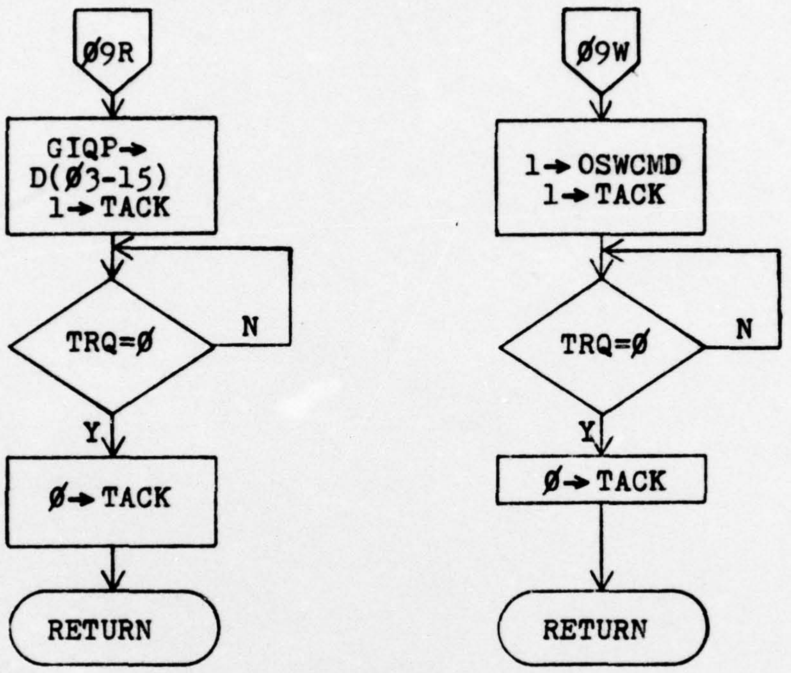


Table B-6
(cont)

CAWØ9R/W

Input Global Input Queue Pointer/Switch to Alternate Bus



CAWØAR/W

Input Local High Priority Interrupts (5)/Enable Local Output

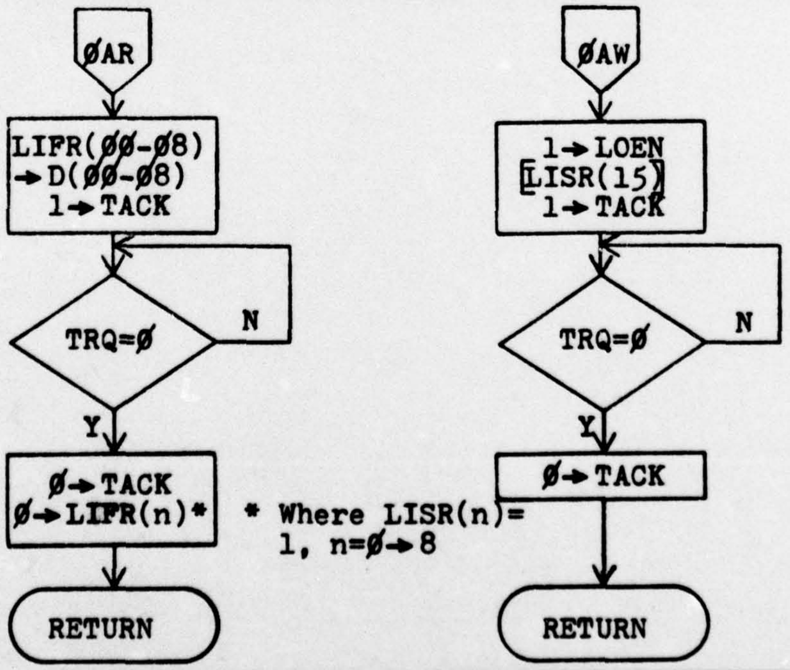
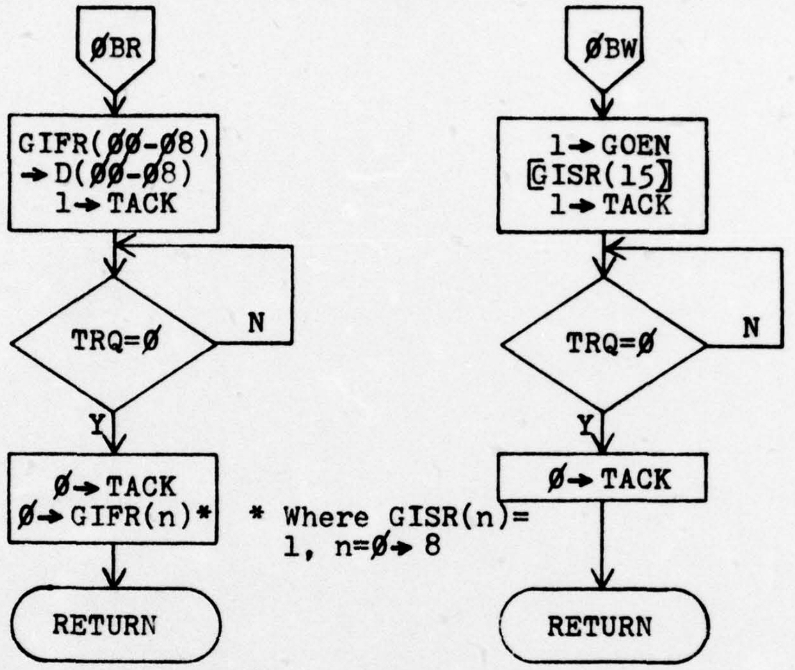


Table B-6
(cont)

CAWØBR/W-Input Global
High Priority Interrupts (3)/Enable Global Output



CAWØCR/W-Input Local
Low Priority Interrupts (6)/Disable Local Output

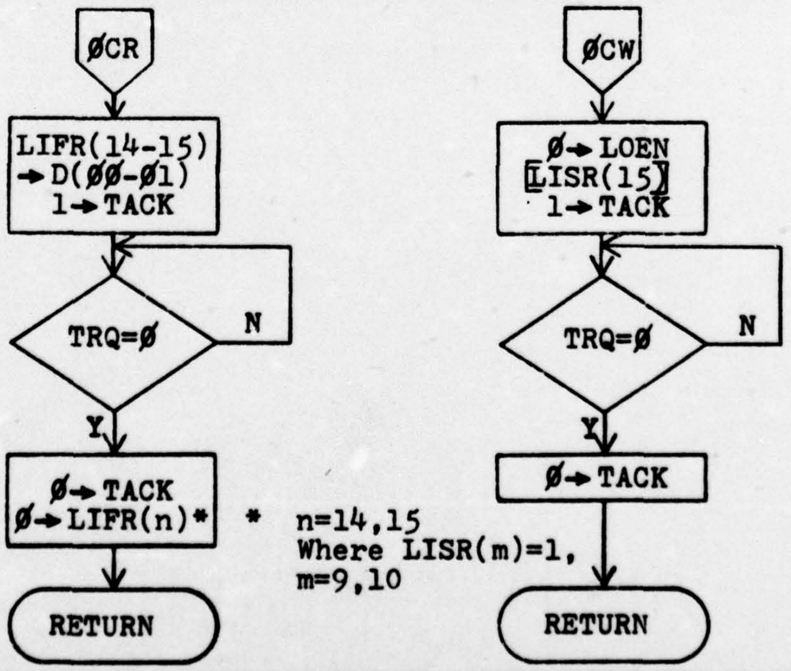


Table B-6
(cont)

CAWØDR/W-Input Global
Low Priority Interrupts (4)/Disable Global Output

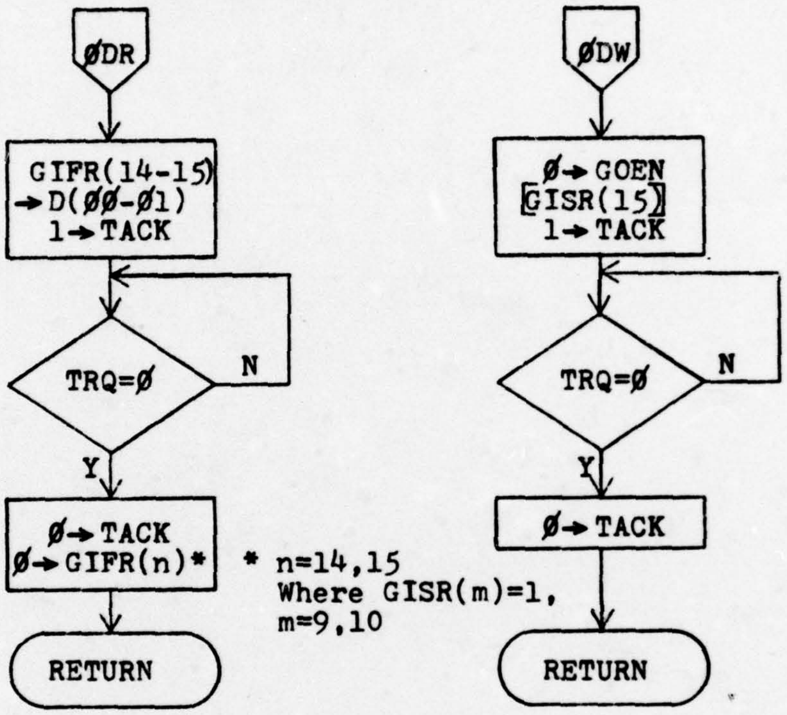
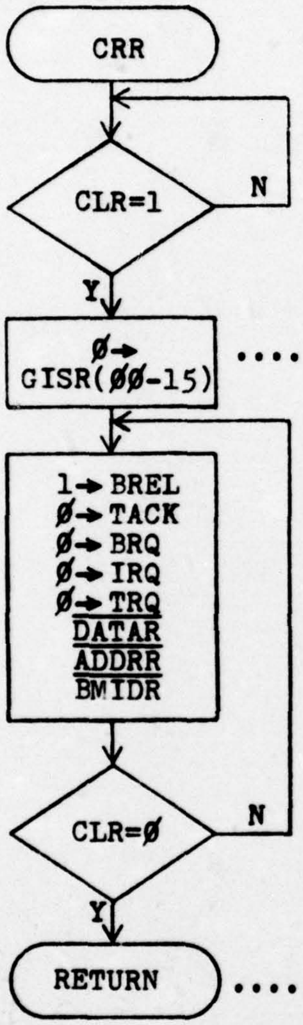


Table B-6
(cont)

(Clear/Reset Routine)



...............Has processor issued a CLK/RST?

..................Disable all interrupts

..............Set appropriate I-Bus control signals

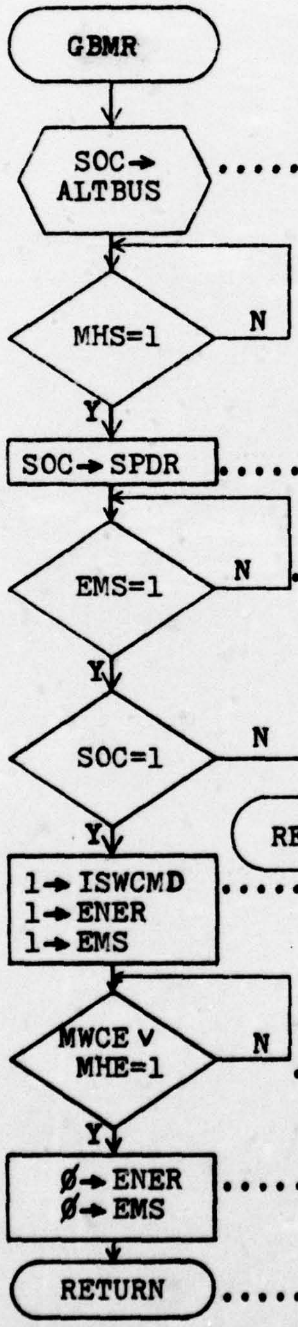
..............Hold I-Bus control signals until CLR=Ø

..................Check status

Table B-7

Global Redundant Bus Management Routine

RTL Description | Comments



GEX PE initiates a Switchover Command (SOC)

Has a message been transmitted on Alternate Bus?

Shift SOC into Serial to Parallel Data Register

End of message?

Does SOC apply to this PE? (compared to hard-wired Reg.)

Check status

Switch Primary and Alternate Buses (Input only)

Set ENER and EMS to terminate Message Reception Routine

Is message reception terminated?

Reset ENER and EMS

Check status

Appendix C

<u>Description of Microword Control Fields</u>

Table C-1

<u>**ALU Field**</u>

A. ALU Source

IØ-I2 (Octal)	ALU Source Operands	
	R Operand	S Operand
Ø	A	Q
1	A	B
2	Ø	Q
3	Ø	B
4	Ø	A
5	D	A
6	D	Q
7	D	Ø

Note: (1) R and S are operand source inputs to the ALU.

(2) A and B are the output latches for the microprocessor 16 word RAM (SP).

(3) Q is the Accumulator.

(4) D is the Direct Data Inputs to the microprocessor.

Table C-1
(cont)

B. ALU Function

I3-I5	ALU Function Definitions	
(Octal)	Function	Symbol
∅	R Plus S	$R + S$
1	S Minus R	$S - R$
2	R Minus S	$R - S$
3	R OR S	$R \vee S$
4	R AND S	$R \wedge S$
5	$\overline{R}$ AND S	$\overline{R} \wedge S$
6	R EX-OR S	$R \veebar S$
7	R EX-NOR S	$\overline{R \veebar S}$

Table C-1
(cont)

C. ALU Destination

16-17	RAM Function		Q-Reg Function		Y	RAM Shifter		Q Shifter	
(Octal)	Shift	Load	Shift	Load	Output	RAMØ	RAM15	QØ	Q15
Ø	X	None	None	$F \rightarrow Q$	F	X	X	X	X
1	X	None	X	None	F	X	X	X	X
2	None	$F \rightarrow B$	X	None	A	X	X	X	X
3	None	$F \rightarrow B$	X	None	F	X	X	X	X
4	Down	$F/2 \rightarrow B$	Down	$Q/2 \rightarrow Q$	F	FØ	IN15	QØ	IN15
5	Down	$F/2 \rightarrow B$	X	None	F	FØ	IN15	QØ	X
6	Up	$2F \rightarrow B$	Up	$2Q \rightarrow Q$	F	INØ	F15	INØ	Q15
7	Up	$2F \rightarrow B$	X	None	F	INØ	F15	X	Q15

Note: (1) X = Don't Care.
(2) B = RAM register addressed by R_B.
(3) A = Contents of RAM A-latch.
(4) INØ/IN15 = Value shifted into the RAMØ/RAM15 or QØ/Q15 (See Table C-3).
(5) Up is toward most significant bit; down is toward least significant bit.
(6) Y is microprocessor output.
(7) F is ALU output.

Table C-2

<u>DALU Control Signals</u>

DALU (Octal)	Mnemonic	Function
∅	A	AB-Field addresses SP, contents of AB location placed in A-Latch
1	GL	A-Field addresses GLRAM
2	B/GL	A-Field addresses GLRAM; AB-Field addresses SP, contents of AB location placed in B-Latch
3	B	AB-Field addresses SP, contents of AB location placed in B-Latch
4	B/LOADS	AB-Field addresses SP, contents of AB location placed in B-Latch; LOADS places value of AB-Field into Shift Counter (SHTCNTR)
5	B/MC	AB-Field addresses SP; A-Field addresses MC PROM
6	MC	A-Field addresses MC PROM
7	A/B	AB-Field addresses SP, contents of AB location placed into A-Latch and B-Latch

Table C-3

<u>SS Field</u>

I7, SSØ-SS1	Source of New Data				Shift	Type
(Octal)	QØ	Q15	RAMØ	RAM15		
Ø	Q1	Ø	F1	Ø	Down	Zero
1	Q1	1	F1	1	Down	One
2	Q1	QØ	F1	FØ	Down	Rotate
3	Q1	FØ	F1	RAM15= RAM14= F15	Down	Arith-metic
4	Ø	Q14	Ø	F14	Up	Zero
5	1	Q14	1	F14	Up	One
6	Q15	Q14	F15	F14	Up	Rotate
7	Ø	Q14	Q15	F14	Up	Arith-metic

Note: (1) The Q and RAM shift operations must be performed in conjunction with the AB-Field write to SP operation and the ALU-Field destination function.

(2) Zero - A LOW is shifted into the Most Significant Bit (MSB) of the RAM on a down shift. If the Q-register is also shifted, then a LOW is deposited in the Q-register MSB. If the RAM or both registers are shifted up, LOWs are placed in the LSBs.

(3) One - Same as zero, but a HIGH level is deposited in the Least Significant Bit (LSB) or MSB.

Table C-3
(cont)

(4) Rotate - A single precision rotate. The RAM MSB shifts into the LSB on a right shift and the LSB shifts into the MSB on a left shift. The Q-register, if shifted, will rotate in the same manner.

(5) Arithmetic - A double-length Arithmetic Shift if Q is also shifted. On an up shift a zero is loaded into the Q-register LSB and the Q-register MSB is loaded into the RAM LSB. On a down shift, the RAM LSB is loaded into the Q-register MSB and the ALU output MSB (F_n, the sign bit) is loaded into the RAM MSB. (This same bit will also be in the next less significant RAM bit.)

Table C-4

<u>TC Field</u>

A. Register I/O Control

Mnemonic	Function
IADDRR	Input Address Register: Latches BUS32-BUS47 in Address Register
IDATAR	Input Data Register: Latches BUS16-BUS31 into Data Register
IIBR	Input I-Bus Register: Gates BUSØ-BUS15 to ALU or GLRAM input
IMØ-IM3	Input Mask Ø-3: Loads VPI MaskØ-3 from ALU output (operates in conjunction with LMASK, a VPI Microinstruction)
CIN	Carry In: Least significant C_n of AM-2901; when active, allows ALU data source to be incremented
LOADS	Load Shift Counter: Allows data to be loaded into Q-Register of Shift Counter
MLTEN	Multiple Enable: Enables Shift Counter so that a multiple loop instruction can be executed
OADDRR	Output Address Register: Gates address onto BUS32-BUS47
OALU	Output ALU: Gates ALU data to Y-output
OIBR	Output I-Bus Register: Gates I-Bus Register data to BUSØ-BUS15
OMP	Output Mapping PROM: Gates beginning address of interrupt microroutine to MS
OMPP	Output Microprogram PROM: Gates selected microinstruction to Microword Register
OMS	Output Microsequencer: Gates selected address to MPP

Table C-4
(cont)

Mnemonic	Function
OWR	Output Microword Register: Gates micro-instruction out of Microword Register for execution
S	Select: Selects data source for DATAR and ADDRR

B. Global/Local Input Control

Mnemonic	Function
GIPR	Global Input Preset: Selects Global Bus A as "primary" bus during system initialization
G/L LOAD	Global/Local Load: Sets counters to zero during system initialization
GLS	Global/Local Select: Controls selection of GLRAM area (using the common Global/Local microroutine, one microprocessor concept discussed in Chapter IV)
ISWCMD	Input Switch Command: Switches "primary" and "alternate" buses when SOC occurs
OALTISR	Output Alternate ISR: Gates Alternate ISR to ALU
OALTPAR	Output Alternate Parity: Gates Alternate Parity to VPI
OGISR	Output Global ISR: Gates Global ISR to ALU and GLRAM (GIDR)
OGPAR	Output Global Parity: Gates Global Parity to VPI

Table C-4
(cont)

Mnemonic	Function
OLISR	Output Local ISR: Gates Local ISR to ALU and GLRAM (LIDR)
OLPAR	Output Local Parity: Gates Local Parity to VPI
WGL	Write to GLRAM: Enables write function to GLRAM

C. Interrupt Control

Mnemonic	Function
SBD	Bus Dominance: Sets VPI Bus Dominance interrupt
SBDGATE	Reset Bus Dominance Gate: Enable/disable BD when I/O messages are less than or equal to eight words/greater than eight words
SBRQ(1)-SBRQ(4)	Bus Request 1-4: Sets VPI Bus Request interrupts
SHP	High Priority: Sets VPI HP interrupt
SMDEE	Message Data Encode Error: Sets VPI MDEE interrupt
SMDPE	Message Data Parity Error: Sets VPI MDPE interrupt
SMHE	Message Header Error: Sets VPI MHE interrupt
SMWCE	Message Word Count Error: Sets VPI MWCE interrupt

Table C-4
(cont)

Mnemonic	Function
SMWLE	Message Word Length Error: Sets VPI MWLE interrupt
SMR	Message Received: Sets VPI MR interrupt

Note: The signals defined represent those needed after encoding MRR. Others will be needed when the other microroutines are coded. These T/C signals do not represent individual control lines. Each control line may be defined as one or several of the above signals.

Table C-5

Global/Local RAM Structure

Location (Decimal)	Global Area	Location (Decimal)	Local Area
∅	GIDR (Global Input Data Register)	32	LIDR (Local Input Data Register)
1	GIAR (Global Input Address Register)	33	LIAR (Local Input Address Register)
2	GIQP (Global Input Queue Pointer)	34	LIQP (Local Input Queue Pointer)
3	GILR (Global Input Length Register)	35	LILR (Local Input Length Register)
4	GODR (Global Out-put Data Register)	36	LODR (Local Output Data Register)
5	GOAR (Global Out-put Address Reg.)	37	LOAR (Local Output Address Register
6	GOLR (Global Out-put Length Reg.)	38	LOLR (Local Output Length Register
7	GRBR (Global Re-sponse Bit Reg.)	39	LRBR (Local Response Bit Register
8	GIMDR (Global In-put Mask Data Register)	40	LIMDR (Local Input Mask Data Register)
9	GBPR (Global Bus Position Register)	41	LBPR (Local Bus Position Register)
10	GBLR (Global Bus Length Register)	42	LBLR (Local Bus Length Register
11	GISR (Global Interrupt Status Register)	43	LISR (Local Interrupt Status Register)
12	PL3/4M (Global Primary Level Interrupt Mask)	44	PL5/6M (Local Primary Level Interrupt Mask)

Table C-5
(cont)

Location (Decimal)	Global Area	Location (Decimal)	Local Area
13-31	To be defined as needed	45-63	To be defined as needed

Note: This structure is used in the common microroutine, one microprocessor concept.

Table C-6

<u>SP Structure for Message Input</u>

Location (Decimal)	Global Area	Location (Decimal)	Local Area
Ø	GIDR	8	LIDR
1	GIMDR	9	LIMDR
2	GIAR	10	LIAR
3	GIQP	11	LIQP
4	GILR	12	LILR
5-7	Not Used	13-15	Not Used

Note: This sturcture is used in the common microroutine, one microprocessor concept.

Table C-7

Mask/Constant PROM

Location (Decimal)	Name	Value (Hexidecimal)	Function
ØØ	CONSTØØ	ØØØ1	Set or Mask IBR Interrupts, etc.
Ø1	CONSTØ1	ØØØ2	↓
Ø2	CONSTØ2	ØØØ4	
Ø3	CONSTØ3	ØØØ8	
Ø4	CONSTØ4	ØØ1Ø	
Ø5	CONSTØ5	ØØ2Ø	
Ø6	CONSTØ6	ØØ4Ø	
Ø7	CONSTØ7	ØØ8Ø	
Ø8	CONSTØ8	Ø1ØØ	
Ø9	CONSTØ9	Ø2ØØ	
10	CONST10	Ø4ØØ	
11	CONST11	Ø8ØØ	
12	CONST12	1ØØØ	
13	CONST13	2ØØØ	

Table C-7
(cont)

Location (Decimal)	Name	Value (Hexidecimal)	Function
14	CONST14	4ØØØ	↓
15	CONST15	8ØØØ	↓
16	MBMID/MRBL	ØØØF	Mask Bus Master Identification from IBAR/Mask Response Bit Location
17	MMHØ1	Ø3FØ	Mask Message Header bits Ø6-11
18	MMHØ2	ØØ7F	Mask Message Header bits Ø9-15
19	PID	To be defined	PE Processor Bus Master Identification
20	BBMID	To be defined	BIU Bus Master Identification
21	APPØ1	1FCØ	Appended constant to generate MIAMM Address
22	APPØ2	1CØØ	Appended constant to generate MH/ML Queue Address
23	HTRAP	To be defined	PE High Priority Interrupt Trap Address
24	LTRAP	To be defined	PE Low Priority Interrupt Trap Address

Table C-7
(cont)

Location (Decimal)	Name	Value (Hexidecimal)	Function
25	GIQP	To be defined	Input Queue Pointer for MH
26	LIQP	To be defined	Local Input Queue Pointer for MH
27	SEND	BBMID: 84Ø	IBR value for data send on I-Bus
28	RCV	BBMID: 86Ø	IBR value for data receive on I-Bus
29	SOCW	To be defined	Switch Over Command Word
30	MØ1	To be defined	Mask words for VPI during system initialization
31	MØ2	To be defined	
32	MØ3	To be defined	↓
33	MØ4	To be defined	
34-63	-	To be defined	—

Table C-8

<u>VPI/ALU Test Instructions</u>

A. VPI Instructions

T/I	TIØ-TI3 (Decimal)	Mnemonic	Function
Ø	Ø	IMCLR	Master Clear: Clears all interrupts, Mask Register, Status Register, Group Enable, Enables Interrupt Request
Ø	1	ICLR	Clear: Clears all interrupts
Ø	2	ICLRM	Clear M: Clears interrupts associated with data placed on M-Bus
Ø	3	ICLRMR	Clear Mask Register Interrupts: Clears interrupts from Mask Register data (using M-Bus to clear associated interrupts)
Ø	4	ICLRINT	Clear Interrupt: Clears Interrupts from M-Bus data
Ø	5	IOVECT	Output Vector: Read vector output to V outputs, load V+1 into Status Register, load V into Vector Hold Register and set Vector Clear Enable
Ø	6	IRS	Read Status: Read Status Register to S-Bus
Ø	7	IRMASK	Read Mask: Read Mask Register to M-Bus
Ø	8	ISM	Set Mask: Set Mask Register (inhibits all interrupts)

Table C-8
(cont)

T/I	TIØ-TI3 (Decimal)	Mnemonic	Function
Ø	9	ILS	Load Status: Load Status Register from S-Bus and Group Enable from GE input
Ø	10	IBCLR	Bit Clear Mask Register: Bit Clear Mask Register from M-Bus
Ø	11	IBSET	Bit Set Mask Register: Bit Set Mask Register from M-Bus
Ø	12	IMEN	Mask Enable: Enables all priority interrupts
Ø	13	IDINT	Disable Interrupt Request: Disables Interrupt Request to MS
Ø	14	ILMASK	Load Mask: Loads Mask Register from M-Bus
Ø	15	IENI	Enable Interrupt Request: Enables Interrupt Request to MS

Table C-8
(cont)

B. ALU Test Instructions

T/I (Decimal)	TIØ-TI3 (Decimal)	Mnemonic	Function
-	Ø-7	-	Not used
1	8	TUNB	Unconditional Branch: Provides an unconditional branch
1	9	TOVR	Overflow: Tests ALU for arithmetic overflow
1	10	THP/SGN	High Priority/Sign: Tests most significant bit of ALU
1	11	TZD	Zero Detect: Tests ALU output for a value of zero
1	12	TCOUT	Carry Out: Tests ALU for carry out from most significant bit
1	13	-	Not defined
1	14	TEMS	End Message Synchronization: Tests to determine if EMS has occured
1	15	TINTRQ	Interrupt Request: Tests to determine if an Interrupt Request from VPI has occured

Table C-9

VPI Interrupts

Interrupt	Name	Interrupt	Name
GCNT17	Global MWCE Counter	LCNT17	Local MWCE Counter
GEMS	Global End Message Synchronization	LEMS	Local End Message Synchronization
GENER	Global Encode Error	LENER	Local Encode Error
GIDBC	Global Input Data Bit Counter	LIDBC	Local Input Data Bit Counter
GMHE	Global Message Header Error	LMHE	Local Message Header Error
GMHS	Global Message Header Synchronization	LMHS	Local Message Header Synchronization
GPAR	Global Parity	LPAR	Local Parity
$\overline{\text{REN}}$	Receive Enable		
GBD	Global Bus Dominance	LBD	Local Bus Dominance
GBQ	Global Bus Quiescence	LBQ	Local Bus Quiescence
GHP	Global High Priority	LHP	Local High Priority

Table C-9
(cont)

Interrupt	Name	Interrupt	Name
GMDEE	Global Message Data Encode Error	LMDEE	Local Message Data Encode Error
GMDPE	Global Message Data Parity Error	LMDPE	Local Message Data Parity Error
GMHE	Global Message Header Error	LMHE	Local Message Header Error
GMWCE	Global Message Word Count Error	LMWCE	Local Message Word Count Error
GMWLE	Global Message Word Length Error	LMWLE	Local Message Word Length Error
GOC	Global Output Complete	LOC	Local Output Complete
GMR	Global Message Received	LMR	Local Message Received
GOMLV	Global Output Message Length Violation	LOMLV	Local Output Message Length Violation
GOA	Global Output Active	LOA	Local Output Active
GOEN	Global Output Enable	LOEN	Local Output Enable
GOP	Global Output Pending	LOP	Local Output Pending

Table C-9
(cont)

Interrupt	Name	Interrupt	Name
ALTERR	Alternate Error	BRQ	Bus Request
ALTEMS	Alternate End Message Synchronization	BGR	Bus Grant
		IRQ	Interrupt Request
BRQ(1)	Bus Request (Global Message Reception)	IAK	Interrupt Acknowledge
BRQ(2)	Bus Request (Global Message Transmission)	IOSL	Input/Output Select
BRQ(3)	Bus Request (Local Message Reception)	DRCV	Data Receive
BRQ(4)	Bus Request (Local Message Transmission)	TRQ	Transfer Request
SBDGATE	Signifies BDGATE Disables	TACK	Transfer Acknowledge
		TTO	Transfer Time Out
		INHB	Maintenance Inhibit
		CLR	Clear/Reset
		BREL	Bus Release

Table C-9
(cont)

Note: (1) The interrupts are presented in the following groups: Global Message Reception, Global High/Low Priority interrupts, Global Message Transmission, Alternate Bus Message Reception, Global/Local Bus Request, Local Message Reception, Local High/Low Priority interrupts, Local Message Transmission, and I-Bus control signals.

(2) This table defines 62 VPI interrupts. The present design provides for 64 interrupts. More than 64 interrupts may be required after coding the remaining microroutines. This does not present a major problem since the VPIE (AM2914) is easily cascadable to any number of interrupts and provision has already been made for a larger MPP microinstruction address field.

Table C-10

<u>Microsequencer Control PROM</u>

Location (Decimal)	Mnemonic	Name	Function	Location (Decimal)	Mnemonic	Name	Function
Ø	INIS	Initialize System	Ø→Y Y+1→MPC	16	-	-	Not defined
1	LPD	Loop on D	D→Y Y→MPC	17	-	-	Not defined
2	PBR	Prepare to Branch	MP→R MPC→Y Y+1→MPC	18	-	-	Not defined
3	SEQ	Sequence	MPC→Y Y+1→MPC	19	TSEQ	Jump to Sub-routine	MPC→STKØ D→Y Y+1→MPC
4	BRR	Branch on R	MPC→STKØ R→Y Y+1→MPC	20	-	-	Not defined
5	POP	Return from Sub-routine	STKØ→Y Y+1→MPC	21	-	-	Not defined
6	LPC	Loop on MPC (Wait)	MPC→Y Y->MPC	22	TLPC	Branch on Test	D→Y Y+1→MPC

Table C-10
(cont)

Location (Decimal)	Mnemonic	Name	Function	Location (Decimal)	Mnemonic	Name	Function
7	BRDR	Go to D	$D \rightarrow Y$ $Y+1 \rightarrow MPC$	23	TBRDR	Go to R	$R \rightarrow Y$ $Y+1 \rightarrow MPC$
8	BRD	Branch on D	$MPC \rightarrow STKØ$ $D \rightarrow Y$ $Y+1 \rightarrow MPC$	24	TBRD	Cont-inue	$MPC \rightarrow Y$ $Y+1 \rightarrow MPC$
9	REG	Go to R	$R \rightarrow Y$ $Y+1 \rightarrow MPC$	25	-	-	Not defined
10	CONT	Continue	$MPC \rightarrow Y$ $Y+1 \rightarrow MPC$	26	TCONT	-	$D \rightarrow Y$ $Y+1 \rightarrow MPC$
11	-	-	Not de-fined	27	TUNB	-	$MPC \rightarrow STKØ$ $D \rightarrow Y$ $Y+1 \rightarrow MPC$
12-15	-	-	Not de-fined	28-31	-	-	Not defined

Appendix D

<u>Microcode for Message Reception Routine</u>

Table D-1

RTL for General

Service Routines

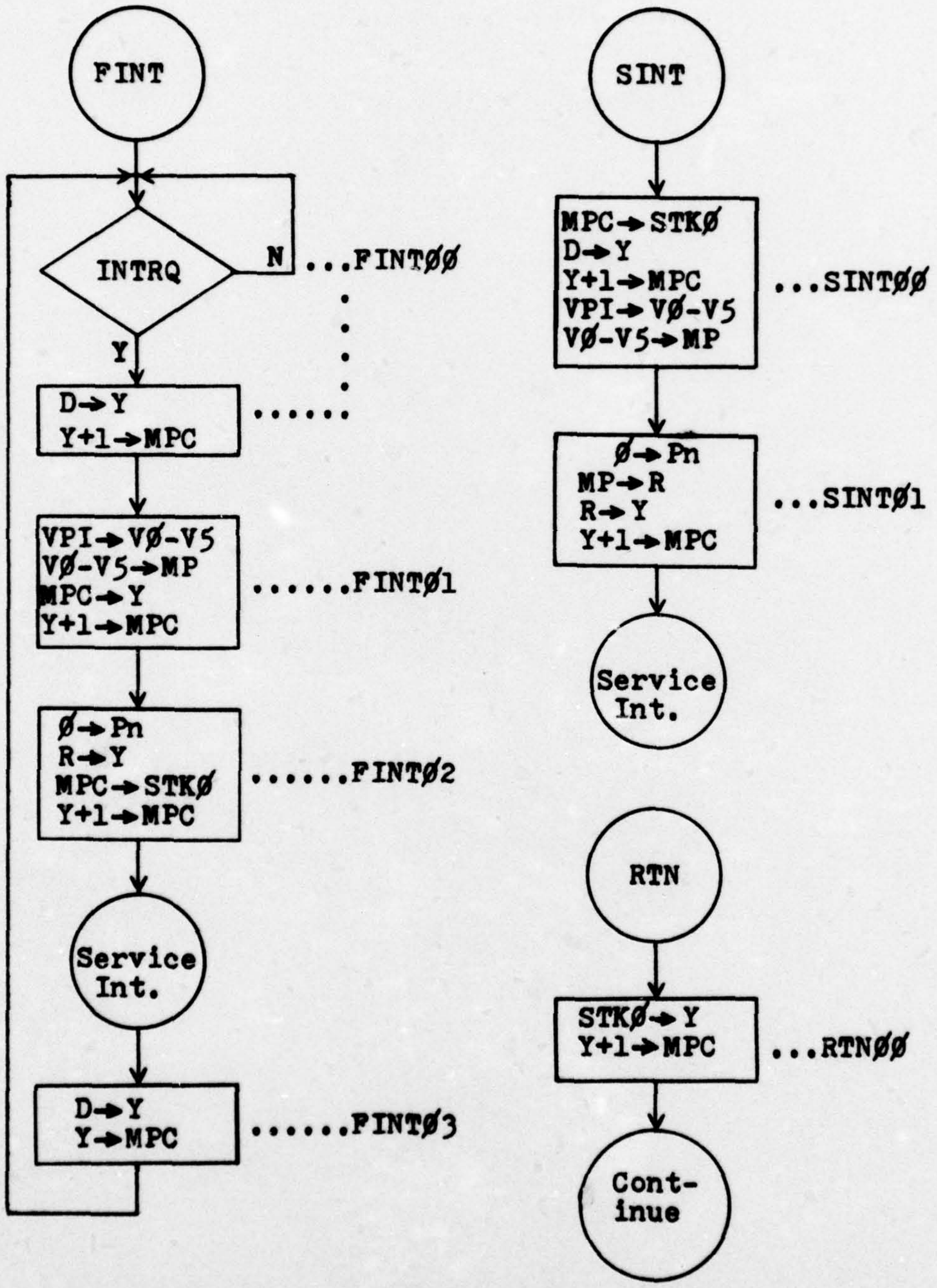


Table D-2

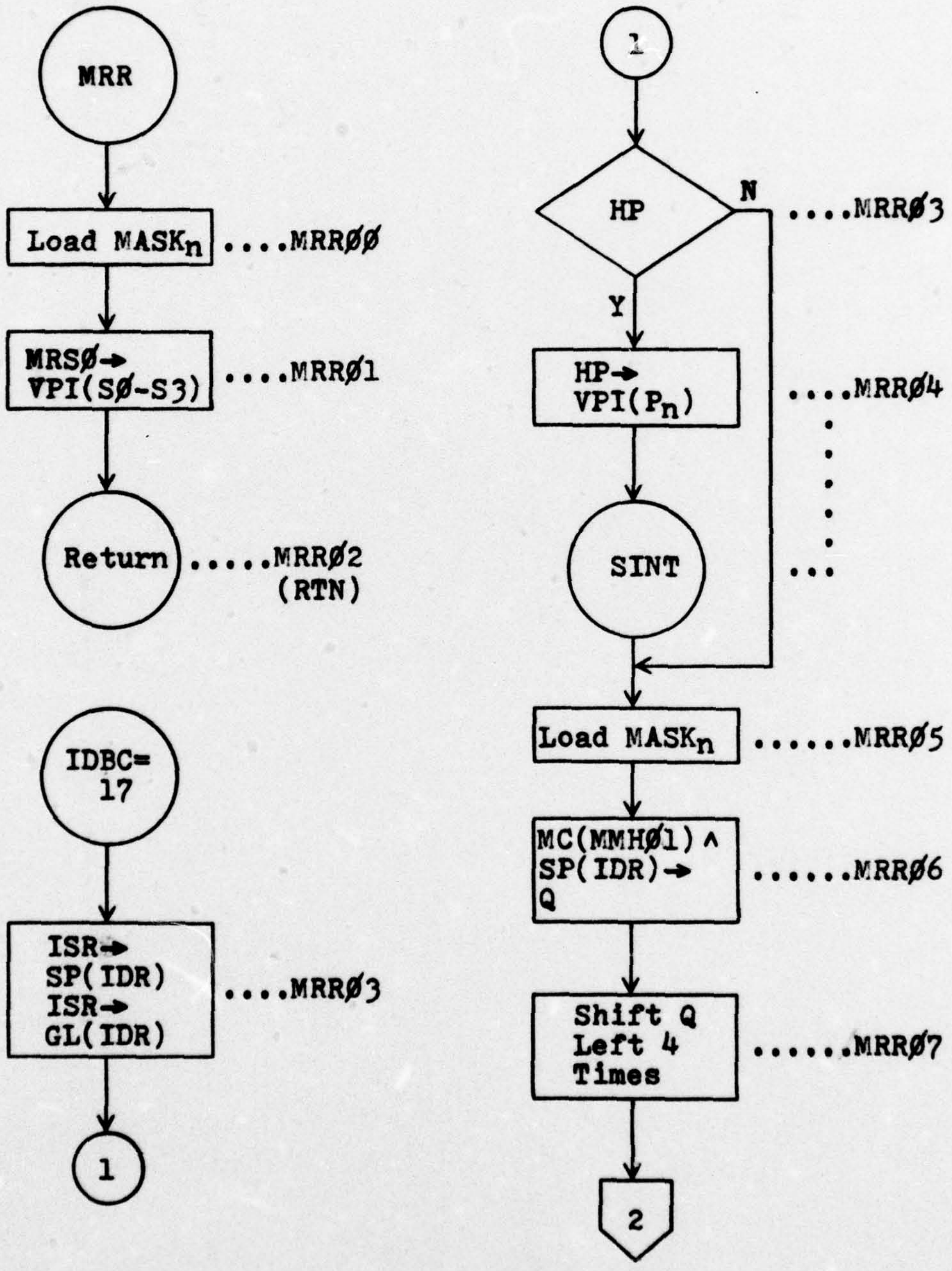
<u>Microcode for General Service Routines</u>

Instruction	ALU Field	DALU Field	A-Field	AB-Field	SS Field	TI Field	TC Field	MS Field	S Field	NA Field
FINTØØ	-	-	-	-	-	TINTRQ	OMPP, OMS OWR	LPC/ TLPC	-	FINT-Ø1
FINTØ1	-	-	-	-	-	IOVECT	OMP, OMPP OMS, OWR	SEQ	-	-
FINTØ2	-	-	-	-	-	ICLRINT	OMPP, OMS OWR	BRR	-	-
FINTØ3	-	-	-	-	-	-	OMPP, OMS OWR	LPD	-	FINTØØ
SINTØØ	-	-	-	-	-	IOVECT	OMP, OMPP OMS, OWR	BRD	-	SINTØ1
SINTØ1	-	-	-	-	-	ICLRINT	OMPP, OMS OWR	REG	-	-
RTN	-	-	-	-	-	-	OMS, OMPP OWR	POP	-	-

Table D-3

<u>RTL for Message</u>

<u>Reception Routine</u>



MRR
Load MASKn ....MRRØØ
MRSØ→ VPI(SØ-S3) MRRØ1
Return .....MRRØ2 (RTN)
IDBC= 17
ISR→ SP(IDR) ISR→ GL(IDR) MRRØ3
1
1
HP
N
....MRRØ3
Y
HP→ VPI(Pn) MRRØ4
SINT ...
Load MASKn MRRØ5
MC(MMHØ1) ^ SP(IDR)→ Q MRRØ6
Shift Q Left 4 Times MRRØ7
2

Table D-3
(cont)

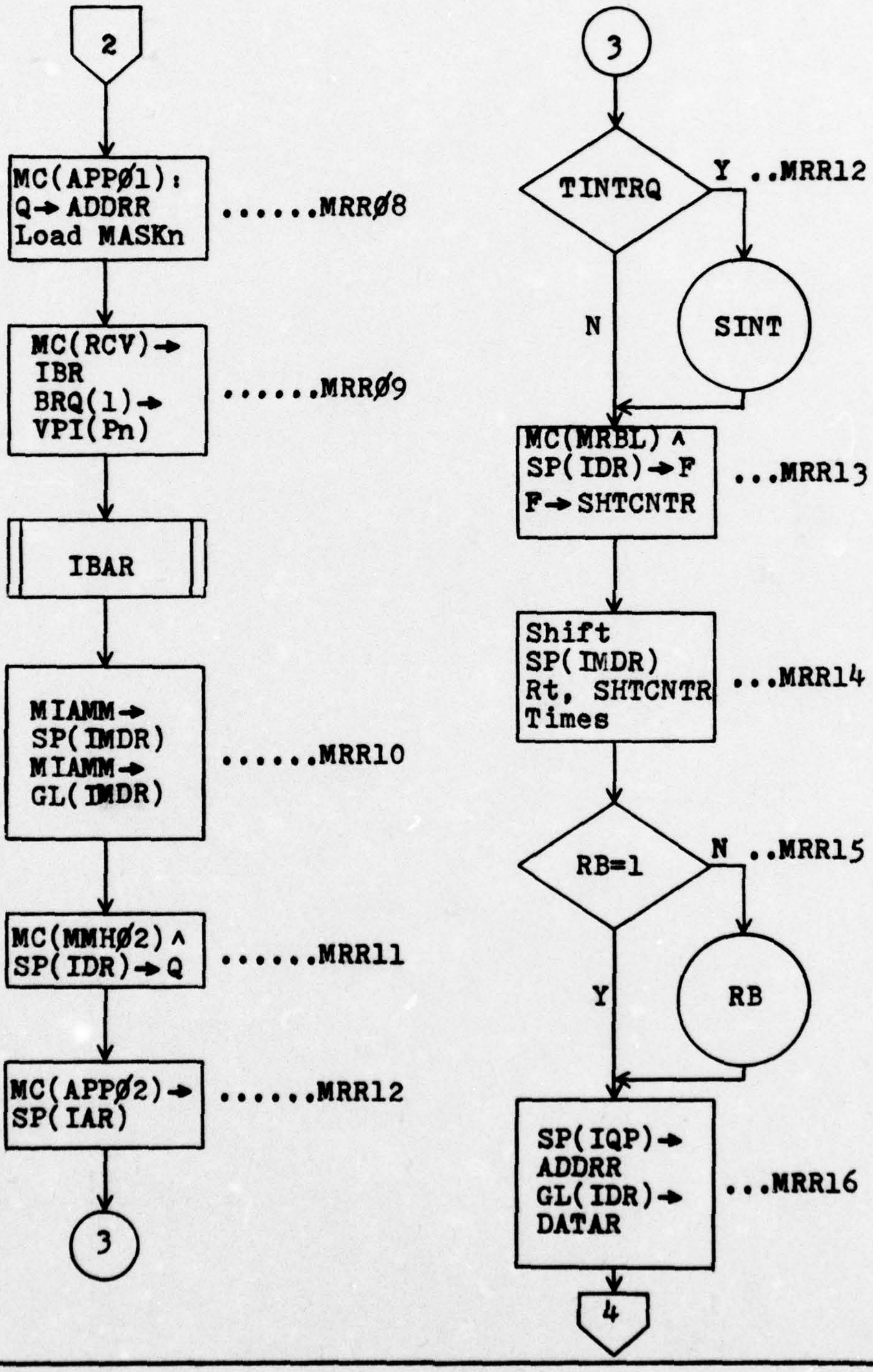


Table D-3
(cont)

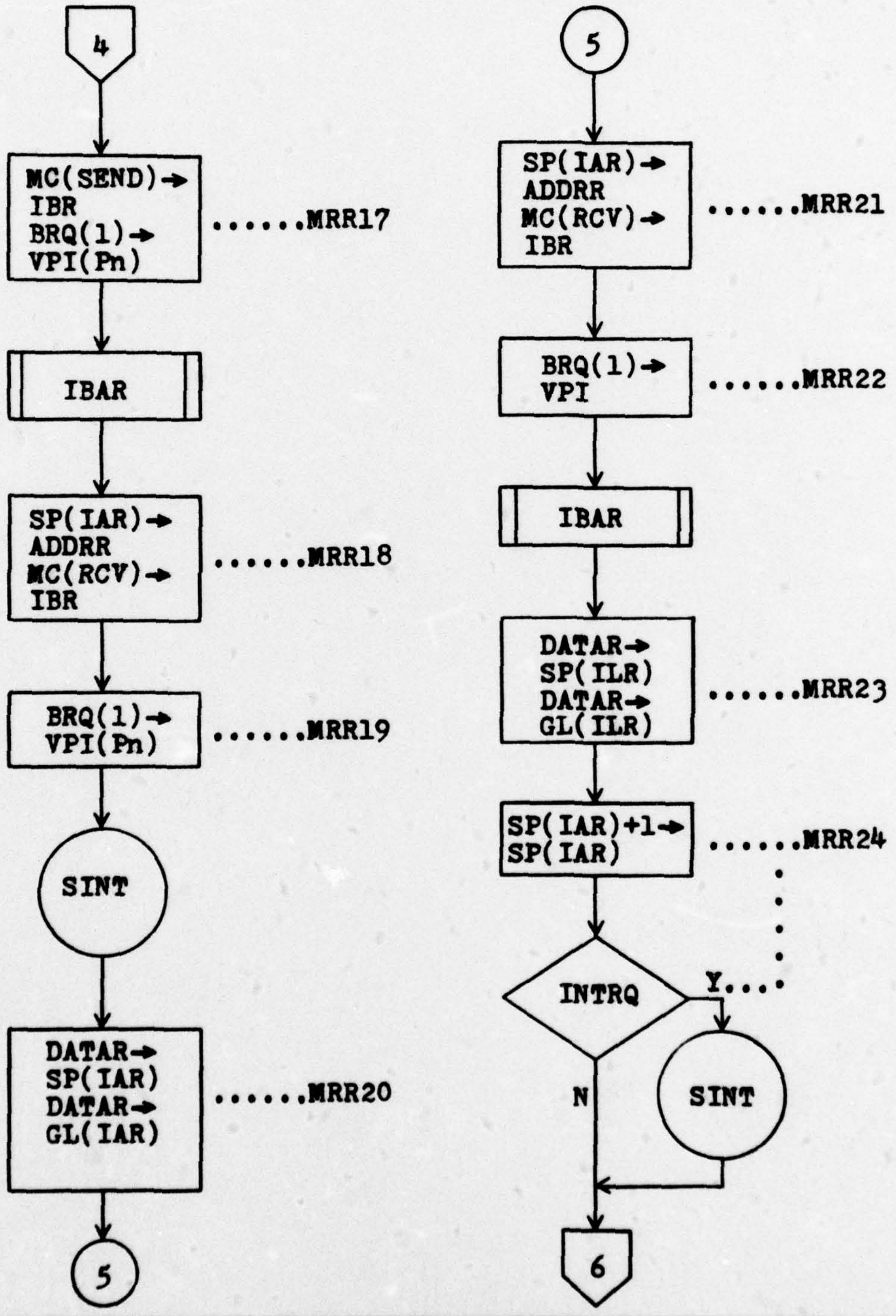


Table D-3
(cont)

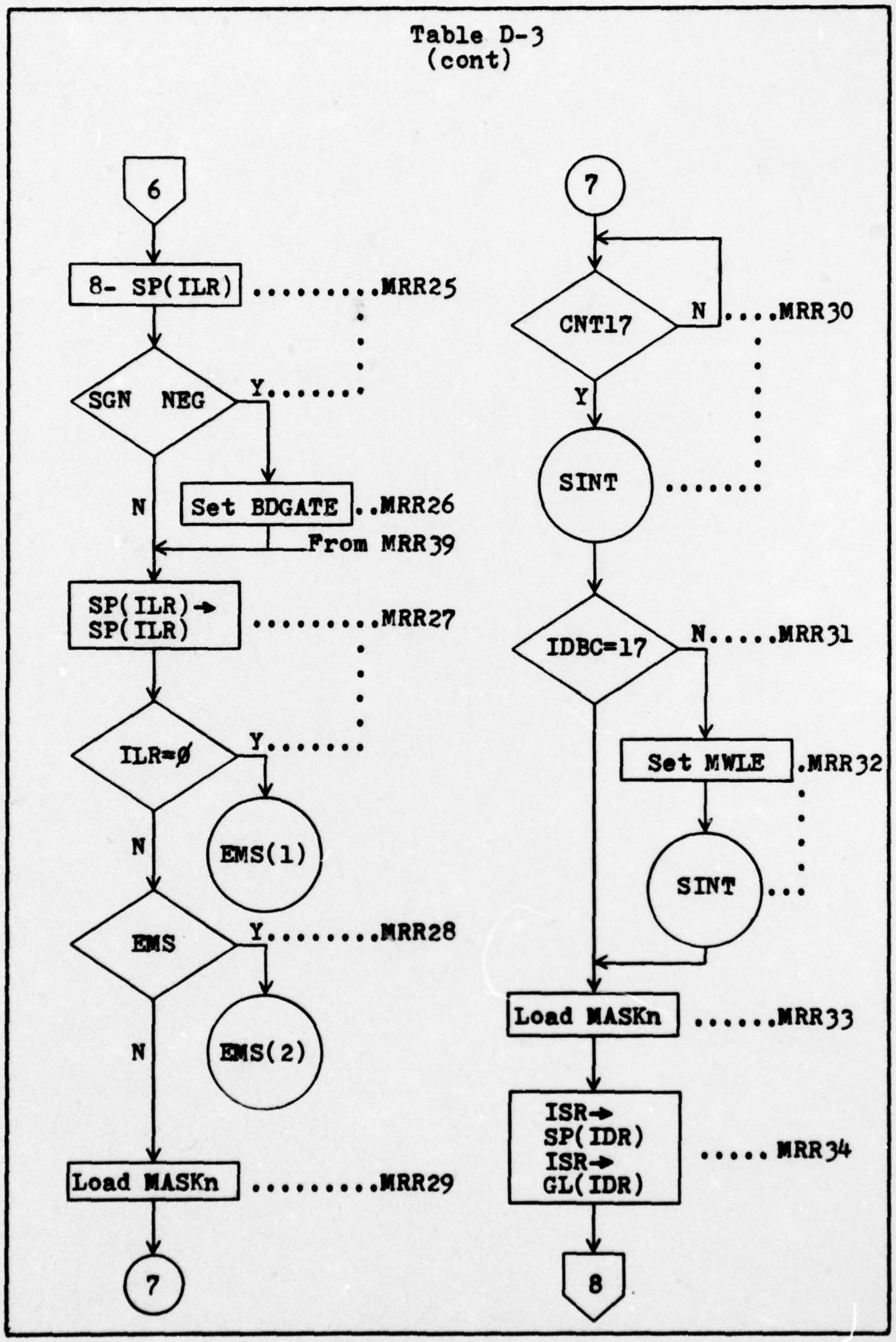


Table D-3
(cont)

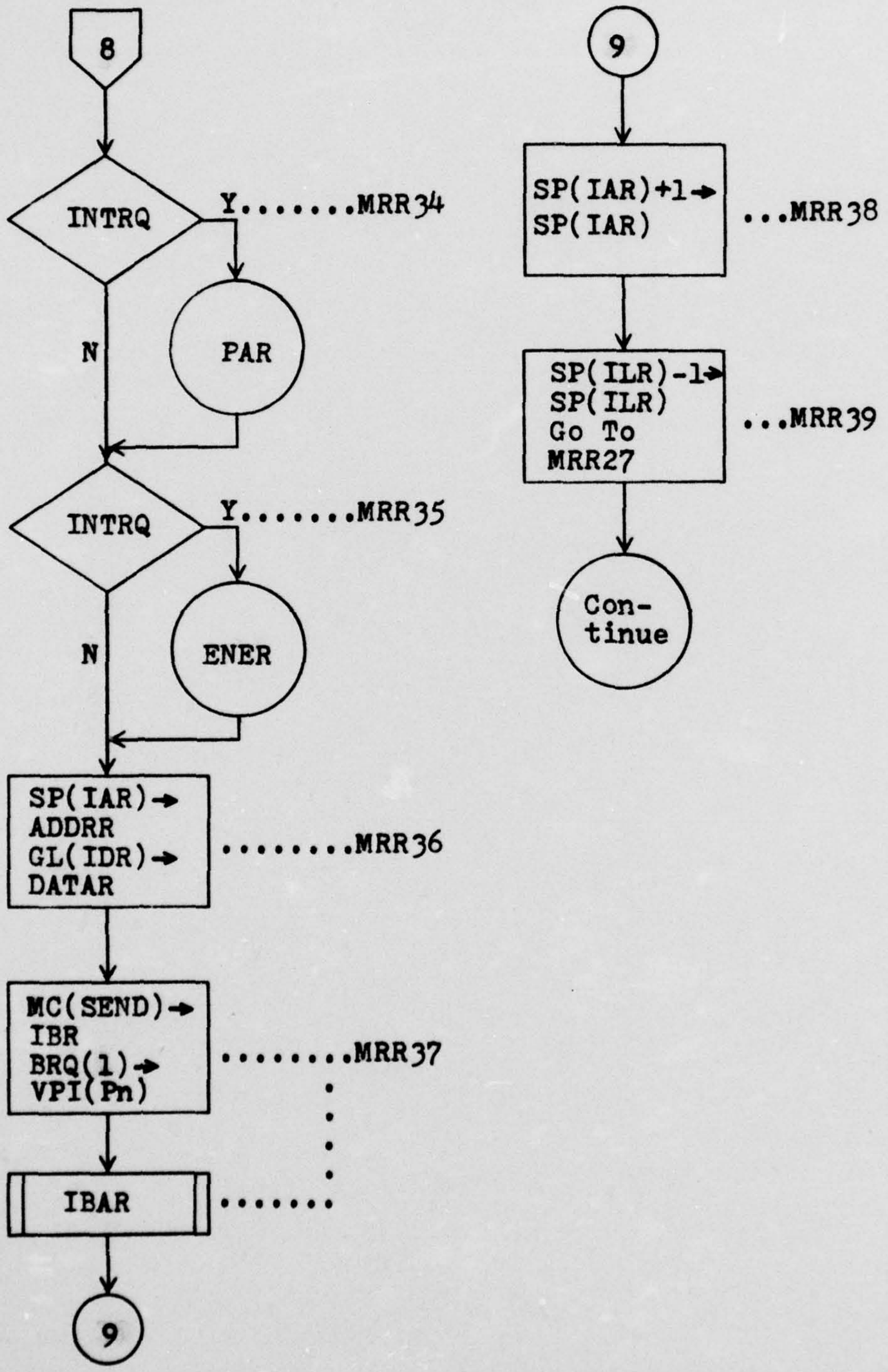


Table D-3
(cont)

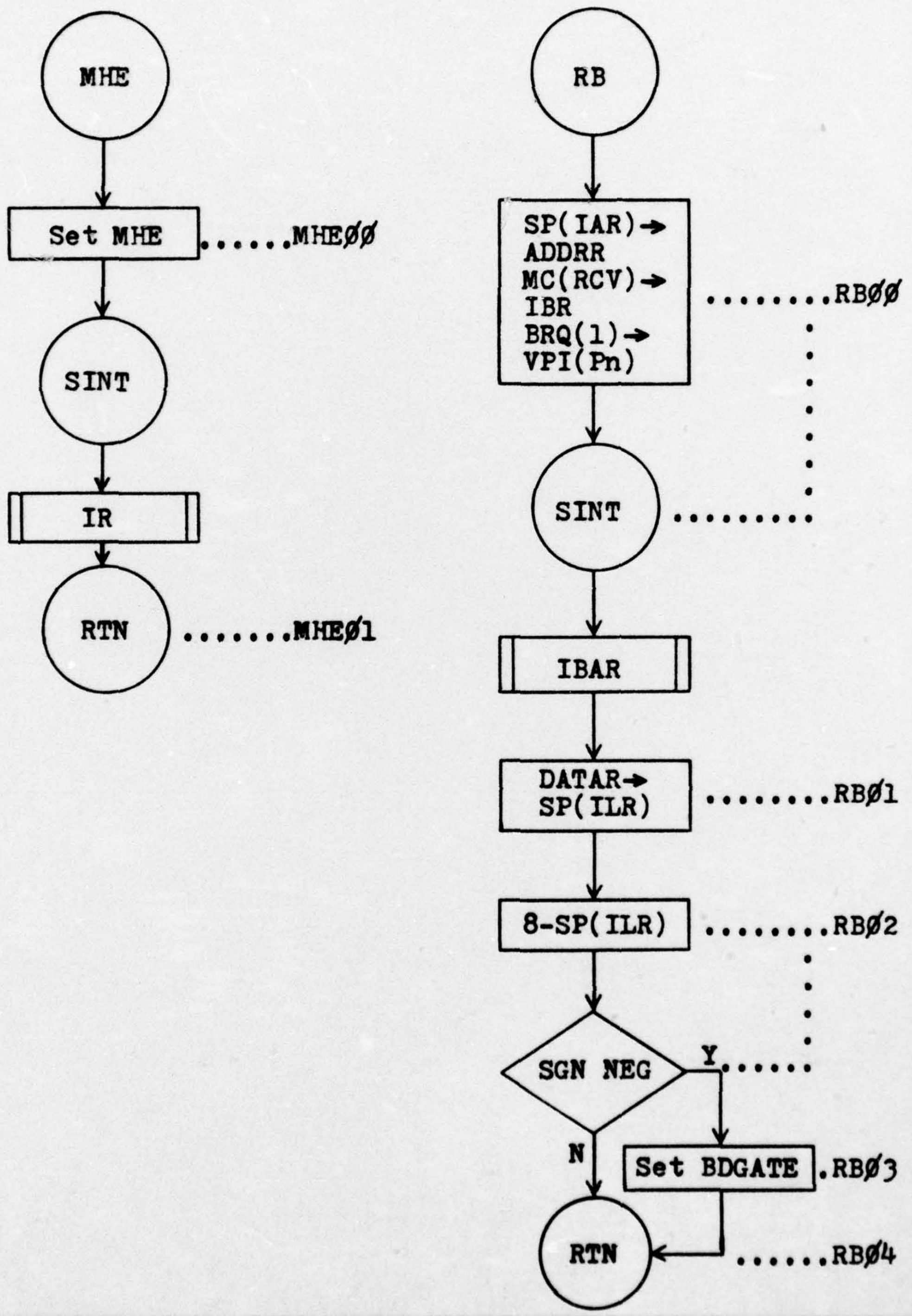


Table D-3
(cont)

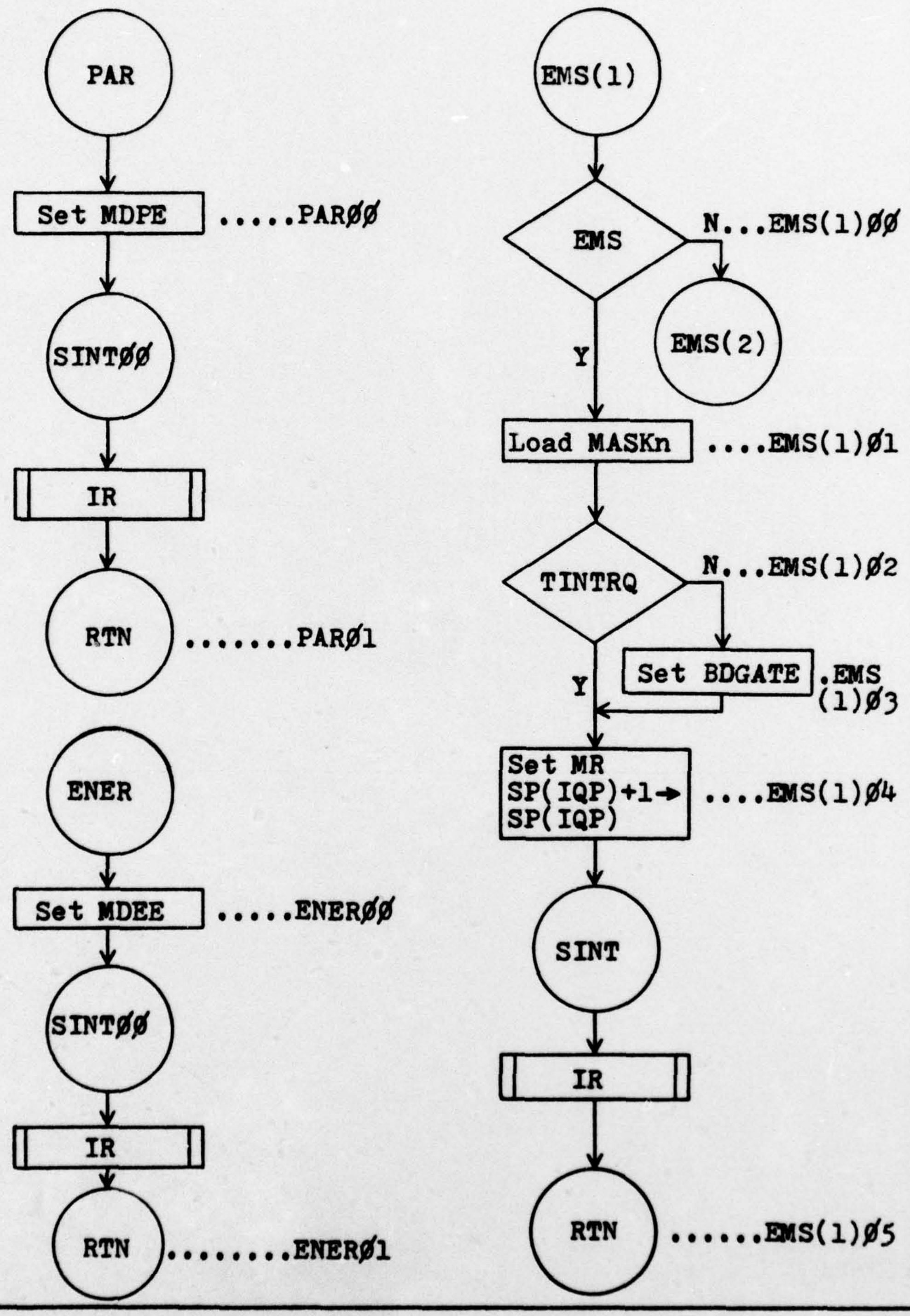


Table D-3
(cont)

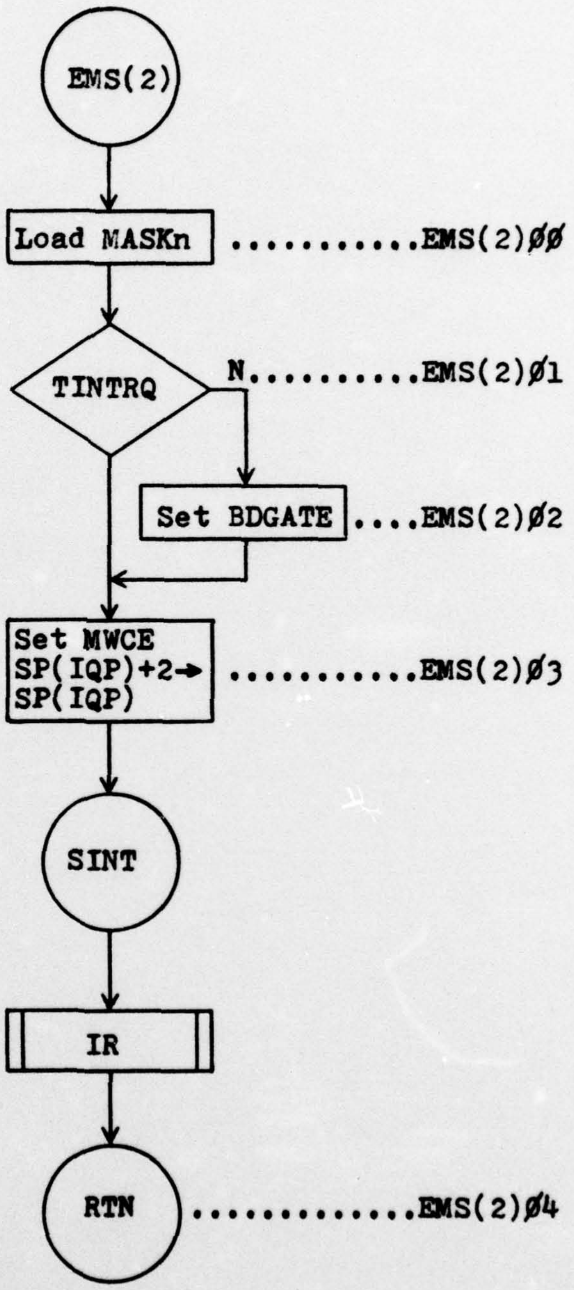


Table D-4

Microcode for Message Reception Routine

Instruction	ALU Field	DALU Field	A-Field	AB-Field	SS Field	TI Field	TC Field	MS Field	S Field	NA Field
MRRØØ	-	MC	MASKn	-	-	ILMASK	OMPP, OMS, OWR, IMn, GLS	SEQ	-	-
MRRØ1	-	-	-	-	-	ILS	OMPP, OMS, OWR	SEQ	MRSØ	-
MRRØ2 (RTN)										
MRRØ3	D→F F→GL/ SP(IDR)	B/GL	GL(IDR)	SP(IDR)	-	THP	OMPP, OMS, OWR, OISR, WGL	BRD/ TBRD	-	MRRØ5
MRRØ4	-	-	-	-	-	TINTRQ	OMPP, OMS, OWR, SHP	SEQ/ TSEQ	-	SINTØØ
MRRØ5	-	MC	MASKn	-	-	ILMASK	OMPP, OMS, OWR, IMn	SEQ	-	-
MRRØ6	D ∧ B→ Q	B/MC	MMHØ1	SP(IDR)	-	-	OMPP, OMS, OWR, GLS	SEQ	-	-

Table D-4
(cont)

Instruction	ALU Field	DALU Field	A-Field	AB-Field	SS Field	TI Field	TC Field	MS Field	S Field	NA Field
MRRØ7	Shift Left $Q \rightarrow Q$	B/LOADS	-	SP(11)	Zero	-	OMPP, OMS, OWR, MLTEN	SEQ	-	-
MRRØ8	$D \vee Q \rightarrow F$	MC	APPØ1	-	-	ILMASK	OMPP, OMS, OWR, IMn, OALU S(ADDRR)	SEQ	-	-
MRRØ9	$D \rightarrow F$	MC	RCV	-	-	TINTRQ	OMPP, OMS, OWR, BRQ-(1), OADDRR, OIBR	SEQ/ TSEQ	-	SINTØØ
MRR1Ø	$D \rightarrow B$	B/GL	IMDR	IMDR	-	-	OMPP, OMS, OWR, IDATAR, WGL	SEQ	-	-
MRR11	$D \wedge B \rightarrow Q$	B/MC	MMHØ2	SP(IDR)	-	-	OMPP, OMS, OWR	SEQ	-	-
MRR12	$D \vee Q \rightarrow B$	B/MC	APPØ2	SP(IAR)	-	TINTRQ	OMPP, OMS, OWR	SEQ/ TSEQ	-	SINTØØ

Table D-4
(cont)

Instruction	ALU Field	DALU Field	A-Field	AB-Field	SS Field	TI Field	TC Field	MS Field	S Field	NA Field
MRR13	D ∧ B→F	B/MC	MRBL	SP(IDR)	-	ILS	OMPP, OMS, OWR, OALU, LOADS	SEQ	MRS1	-
MRR14	B→B	B	-	SP(IMDR)	ZERO	-	OMPP, OMS, OWR, MLTEN	SEQ	-	-
MRR15	B→B	B	-	SP(IMDR)	-	SGN	OMPP, OMS, OWR	CONT/ TCONT	-	RBØØ
MRR16	B→F	B/GL	GL(IDR)	SP(IQP)	-	-	OMPP, OMS, OWR, OALU, S(ADDRR) S(DATAR)	SEQ	-	-
MRR17	-	MC	SEND	-	-	TINTRQ	OMPP, OMS, OWR, BRQ(1), OADDRR, ODATAR, OIBR	SEQ/ TSEQ	-	SINTØØ
MRR18	B→F	B/MC	RCV	SP(IAR)	-	-	OMPP, OMS, OWR, OALU, S(ADDRR)	SEQ	-	-

Table D-4
(cont)

Instruction	ALU Field	DALU Field	A-Field	AB-Field	SS Field	TI Field	TC Field	MS Field	S Field	NA Field
MRR19	-	-	-	-	-	TINTRQ	OMPP, OMS, OWR, BRQ(1) OADDRR, OIBR	SEQ/ TSEQ	-	SINTØØ
MRR2Ø	$D \rightarrow F$ $F \rightarrow B$	B/GL	GL(IAR)	SP(IAR)	-	-	OMPP, OMS, OWR, IDATAR	SEQ	-	-
MRR21	$B \rightarrow F$	B/MC	RCV	SP(IAR)	-	-	OMPP, OMS, OWR, OALU, S(ADDRR)	SEQ	-	-
MRR22	-	-	-	-	-	TINTRQ	OMPP, OMS, OWR, BRQ(1) OADDRR OIBR	SEQ/ TSEQ	-	SINTØØ
MRR23	$D \rightarrow F$ $F \rightarrow B$	B/GL	GL(ILR)	SP(ILR)	-	-	OMPP, OMS, OWR, IDATAR	SEQ	-	-
MRR24	$B+1 \rightarrow B$	B	-	SP(IAR)	-	TINTRQ	OMPP, OMS, OWR, CIN	SEQ/ TSEQ	-	SINTØØ

Table D-4
(cont)

Instruction	ALU Field	DALU Field	A-Field	AB-Field	SS Field	TI Field	TC Field	MS Field	S Field	NA Field
MRR25	D-B→F	B/MC	CONSTØ3	SP(ILR)	-	SGN	OMPP, OMS, OWR	CONT/ TCONT	-	MRR27
MRR26	-	-	-	-	-	-	OMPP, OMS, OWR, SBDGATE	SEQ	-	-
MRR27	B→B	B	-	SP(ILR)	-	TZD	OMPP, OMS, OWR, GLS	CONT/ TCONT	-	EMS-(1)ØØ
MRR28	-	-	-	-	-	TEMS	OMPP, OMS, OWR	CONT/ TCONT	-	EMS-(2)ØØ
MRR29	-	MC	MASKn	-	-	ILMASK	OMPP, OMS, OWR, IMn	SEQ	-	-
MRR3Ø	-	-	-	-	-	TINTRQ	OMPP, OMS, OWR	LPC/ TLPC	-	SINTØØ
MRR31	-	-	-	-	-	TINTRQ	OMPP, OMS, OWR	BRD/ TBRD	-	MRR33
MRR32	-	-	-	-	-	TINTRQ	OMPP, OMS, OWR, SMWLE	SEQ/ TSEQ	-	SINTØØ

Table D-4
(cont)

Instruction	ALU Field	DALU Field	A-Field	AB-Field	SS Field	TI Field	TC Field	MS Field	S Field	NA Field
MRR33	-	MC	MASKn	-	-	ILMASK	OMPP, OMS, OWR, IMn	SEQ	-	-
MRR34	D→F F→B	B/GL	GL(IDR)	SP(IDR)	-	TINTRQ	OMPP, OMS, OWR, OISR, WGL, OPAR	SEQ/ TSEQ	-	SINTØØ
MRR35	-	-	-	-	-	TINTRQ	OMPP, OMS, OWR	SEQ/ TSEQ	-	SINTØØ
MRR36	B→F	B/GL	GL(IDR)	SP(IAR)	-	-	OMPP, OMS, OWR, OALU, S(DATAR) S(ADDRR)	SEQ	-	-
MRR37	-	MC	SEND	-	-	TINTRQ	OMPP, OMS, OWR, BRQ(1), OADDRR, ODATAR, OIBR	SEQ/ TSEQ	-	SINTØØ
MRR38	B+1→B	B	-	SP(IAR)	-	-	OMPP, OMS, OWR, CIN	SEQ	-	-

Table D-4
(cont)

Instruction	ALU Field	DALU Field	A-Field	AB-Field	SS Field	TI Field	TC Field	MS Field	S Field	NA Field
MRR39	B-1→B	B	-	SP(ILR)	-	-	OMPP, OMS, OWR	BRDR	-	MRR27
MHEØØ	D→Q	-	-	-	-	TINTRQ	OMPP, OMS, OWR, SMHE	SEQ/ TSEQ	-	SINTØØ
MHEØ1 (RTN)										
RBØØ	B→F	B/MC	RCV	SP(IAR)	-	TINTRQ	OMPP, OMS, OWR, BRQ(1), OALU, S(ADDRR)	SEQ/ TSEQ	-	SINTØØ
RBØ1	D→F F→B	B	-	SP(ILR)	-	-	OMPP, OMS, OWR, IDATAR	SEQ	-	-
RBØ2	D-B→F	B/MC	CONSTØ3	SP(ILR)	-	SGN	OMPP, OMS, OWR	CONT/ TCONT	-	RBØ4
RBØ3	-	-	-	-	-	-	OMPP, OMS, OWR, SBDGATE	SEQ	-	-

Table D-4
(cont)

Instruction	ALU Field	DALU Field	A-Field	AB-Field	SS Field	TI Field	TC Field	MS Field	S Field	NA Field
RBØ4 (RTN)										
PARØØ	-	-	-	-	-	TINTRQ	OMPP, OMS, OMW, SMDPE	SEQ/ TSEQ	-	SINTØØ
PARØ1 (RTN)										
ENERØØ	-	-	-	-	-	TINTRQ	OMPP, OMS, OMW, SMDEE	SEQ/ TSEQ	-	SINTØØ
ENERØ1 (RTN)										
EMS(1)ØØ	-	-	-	-	-	TEMS	OMPP, OMS, OWR	CONT/ TCONT	-	EMS(2) ØØ
EMS(1)Ø1	-	MC	MASKn	-	-	ILMASK	OMPP, OMS, OWR, IMn	SEQ	-	-
EMS(1)Ø2	-	-	-	-	-	TINTRQ	OMPP, OMS, OWR	CONT/ TCONT	-	EMS(1) Ø4

Table D-4
(cont)

Insturc-tion	ALU Field	DALU Field	A-Field	AB-Field	SS Field	TI Field	TC Field	MS Field	S Field	NA Field
EMS(1)Ø3	-	-	-	-	-	-	OMPP, OMS OWR, SBDGATE	SEQ	-	-
EMS(1)Ø4	B+1→B	B	SP(IQP)	-	-	TINTRQ	OMPP, OMS, OWR, SMR, CIN	SEQ/ TSEQ	-	SINTØØ
EMSØ5 (RTN)										
EMS(2)ØØ	-	MC	MASKn	-	-	ILMASK	OMPP, OMS, OMW, IMn	SEQ	-	-
EMS(2)Ø1	-	-	-	-	-	TINTRQ	OMPP, OMS, OWR	CONT/ TCONT	-	EMS(2) Ø3
EMS(2)Ø2	-	-	-	-	-	-	OMPP, OMS, OWR, SBDGATE	SEQ	-	-
EMS(2)Ø3	D+B→B	B/MC	CONSTØ1	SP(IQP)	-	TINTRQ	OMPP, OMS, OWR, SMWCE	SEQ	-	SINTØØ
EMS(2)Ø4 (RTN)										

VITA

Robert C. Simpson was born on 14 May 1940 in Potsdam, New York. He graduated from Norwood-Norfolk Central School in 1959 and completed two years of undergraduate school at Clarkson College of Technology in 1961. He entered the USAF on 13 November 1961 and was accepted into the Airman's Education and Commissioning Program in August 1962. He completed his undergraduate requirements for a Bachelor of Science Degree in Electrical Engineering at Colorado State University in August 1964. Upon completion of Officer Training School, he was commissioned a 2nd Lt. in the USAF. He then attended Undergraduate Navigator Training at James-Connelly AFB and graduated in November 1965. The next six years were spent as a navigator in the C-130A at Lockbourne AFB, C-130B at Mactan AB, Phillipines, and KC-135A at Robins AFB, Georgia. In July 1971, he became an Instructor Navigator at Mather AFB, California in the Undergraduate Navigator Training School where he flew the T-29 and T-43. He entered the AFIT resident school in June 1975 and received the degree of Master of Science in Electrical Engineering in December 1976.

Permanent Address: Main Street
Norfolk, New York
13667

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER GE/EE/76D-40	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DESIGN OF THE BUS INTERFACE UNIT FOR THE DISTRIBUTED PROCESSOR/MEMORY SYSTEM		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert C. Simpson		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Avionics Laboratory (AFAL/AAT) Wright-Patterson AFB, Ohio 45433		12. REPORT DATE December 1976
		13. NUMBER OF PAGES 176
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

[illegible]

Jerral F. Guess, Capt, USAF
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Microprocessor, Avionics, Emulate, Bus Communication, Interface, Minicomputer

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Distributed Processor/Memory (DP/M) system is a concept developed by the Air Force Avionics Laboratory (AFAL). The DP/M system is a decentralized, software programable Processor Element (PE) which is used to interface the avionics on board an aircraft. Each PE or group of PEs is connected to an aircraft sensor and the PE's are interconnected via a global communication bus. The key characteristic of the DP/M is decentralization (i.e., each avionic interface module, PE, determines when it will access the global bus whereas present systems are controlled by a central computer).

This report is in support of a request from AFAL to design the Bus Interface Unit (BIU) of the PE. The BIU provides the communication link between a serial global bus and a parallel data PE processor. The original design concept was to implement the BIU in hardware, but with the advances in high-speed large scale integrated circuits, the system design and testing would be more flexible if a special purpose microprocessor was used.

This study, after defining the BIU requirements in detail, uses the AM2900 Bipolar Microprocessor chip set (Advanced Micro Devices, Sunnyvale, California) to implement the BIU. The hardware, microword format, and data reception microcode were developed.

The high-speed and flexibility of the AM2900 chip set as implemented shows that a microprocessor can be used to meet the requirements of a complex hardware system.