MCDONNELL DOUGLAS TECHNICAL SERVICES CO.
HOUSTON ASTRONAUTICS DIVISION

SPACE SHUTTLE ENGINEERING AND OPERATION SUPPORT

1.3-DN-C0504-037

General Purpose Computer (GPC) to GPC Systems Interface Description

AVIONICS SYSTEMS ENGINEERING

20 SEPTEMBER 1976

This Design Note is Submitted to NASA Under Task Order No. C0504 in Fulfillment of Contract NAS 9-14960.

PREPARED BY: B. C. Breyer
B. C. Breyer
Technical Specialist
488-5660 x 285

APPROVED BY: T. G. Poiitte
T. G. Poiitte
Senior Group Engineer
488-5660 x 285

APPROVED BY: L. R. Blanke
L. R. Blanke
Technical Manager
488-5660 x 257

APPROVED BY: R. F. Pannett
R. F. Pannett
Project Manager
488-5660 x 258

1.0 SUMMARY

This note describes the General Purpose Computer (GPC) "subsystem" of the Orbiter Data Processing System (DPS). Two interface areas are described. One is the area of GPC intraconnections and intracommunications involving the hardware/software interface between the Central Processing Unit (CPU) and the Input/Output Processor (IOP). The other is the area of GPC interconnections and intercommunications and involves the hardware/software interface between the five Orbiter GPC's.

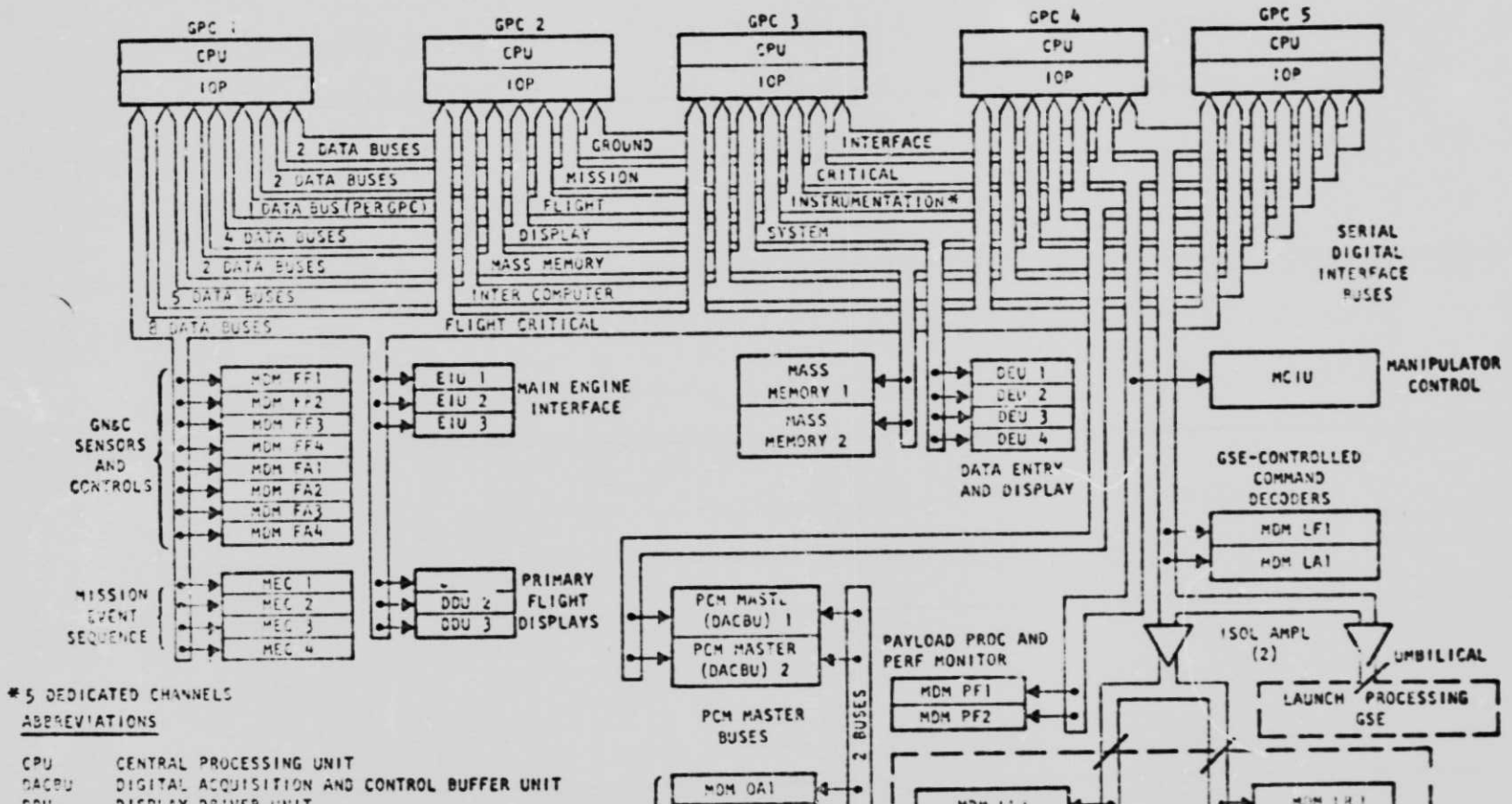
This note is one of a series of Orbiter DPS interface descriptions that will be published on specific data buses and their respective Bus Terminal Units. (BTU's). The purpose of each design note is to thoroughly describe the hardware/software configuration and interface to a level of detail sufficient for the integrated systems operation to be understood.

Based on the detailed GPC interface description developed in Section 3.0, it is felt that the basic CPU to IOP interface and the GPC to GPC interface have the potential for trouble free operation. However, due to the complexity of the interface and the criticality of GPC synchronization to overall avionics performance, the GPC to GPC interface should be carefully evaluated when attempting to resolve test anomalies that may involve GPC timing and synchronization errors.

2.0 INTRODUCTION

The Space Shuttle DPS design requires that the five Orbiter GPC's have the ability to communicate certain vital information. This information enables the GPC's to operate in a coordinated manner. The exchange of information between GPC's is accomplished by using five dedicated Intercomputer Communication (ICC) data buses and selected Discrete Input (DI) and Discrete Output (DO) lines. Each GPC is in command of one ICC data bus which is used by that GPC to send specially formatted serial digital ICC messages to the other GPC's. The selected DO lines are set or reset by each GPC to be input on the DI lines of the other GPC's. The set or reset condition of these discrete lines is used to communicate status or operating modes between the GPC's. The discrete lines provide GPC information exchange that is best communicated by on/off bit configurations and is available to be read continuously during program execution.

A block diagram of the total Orbiter Avionics DPS (extracted from reference C) is shown in Figure 1. Only those DPS elements previously discussed are described by this design note.



BLOCK DIAGRAM OF AVIONICS DPS SYSTEM

Figure 1

3.0 DISCUSSION

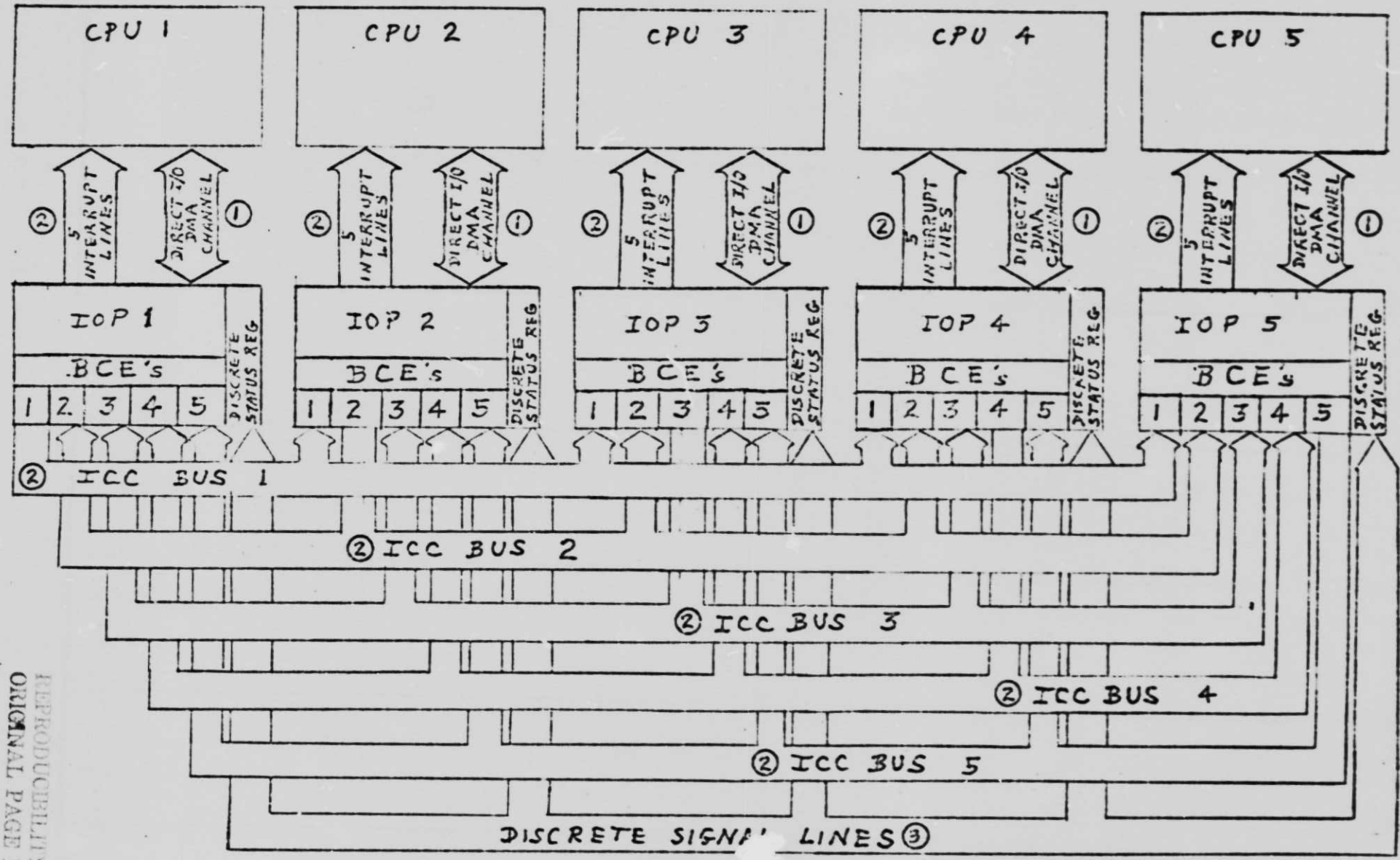
The reference material used during this effort is listed in Section 5.0.

3.1 GPC System Overview

The hardware and software interfaces within the Orbiter Avionics System are essential to the successful operation of the multiprocessor GPC configuration which is designed for simplex and redundant computer operation. The DPS hardware and associated software must operate in a cooperative relationship which includes hardware Built-In-Test Equipment (BITE) and software that check each other to prevent error or failure conditions from going undetected. The interfaces between the five GPC's are referred to as intercommunication and interfaces between the CPU and IOP of a single GPC are referred to as intra-communication. Figure 2 shows these hardware interfaces.

The intercommunication interface utilizes both serial digital data buses (to transmit twenty-eight bit words containing sixteen data bits) and discrete signal lines (to convey certain discrete information). Five ICC data buses are used to send and receive the serial digital data essential to the operation of GPC's as common or redundant set members. Each GPC has a dedicated ICC data bus to transmit messages to the other GPC's, while it receives messages from the other GPC's on the remaining ICC data buses. All GPC's exchange their ICC messages simultaneously at scheduled intervals, a 1 MHz serial bit rate.

Intracommunication involves the bi-directional exchange of information between the CPU and the IOP that is essential to operation of the GPC as



(1) 32 BIT PARALLEL BI-DIRECTIONAL DATA BUS, SHIELDED TWISTED PAIRS
(2) SHIELDED, TWISTED PAIR
(3) 18 SHIELDED, TWISTED PAIRS

BLOCK DIAGRAM OF GPC INTERCONNECTS
Figure 2

a unit. This is the first level of interface and must be operating in order for the individual GPC to initiate and sustain the intercommunication processes. Intracommunication between the CPU and IOP utilizes a bi-directional thirty-four bit (includes two parity bits) parallel data channel and five interrupt lines. The data channel allows the CPU to communicate, through the use of program controlled Input/Output (I/O) commands, with the processors in the IOP to read, set and reset the discrete signal line registers associated with the GPC. The IOP processors communicate with the CPU by using the data channel as a Direct Memory Access (DMA) channel to access main CPU and IOP memory. This path is used to read instructions and read/write data from/to memory. The IOP also communicates with the CPU via interrupts to indicate I/O completion or to indicate certain error conditions or change of status. The CPU reads the IOP interrupt status register with a program controlled input command to determine the cause of the interrupt.

3.2 GPC Intracommunication

GPC Intracommunication involves the communication between the CPU and the IOP and is conducted over a bi-directional thirty-four bit parallel data bus which is utilized as an I/O channel between the CPU and IOP. All information transfers between the CPU and IOP are managed by the Channel Controller (CC).

3.2.1 Direct I/O

The CPU utilizes the I/O channel for direct I/O operations controlled by the CC. The CC consists of differential drivers and receivers for control signal transfer between IOP and CPU. Specifically, an eighteen bit DMA

address register, a DMA address register odd parity generator, a thirty-two bit DMA input data register, an eighteen bit PCI/O command register, a parity generator/checker, a thirty-two bit output data register, a DMA input data register, and other logic. The thirty-six bit bi-directional data consists of thirty-two bits of data, an odd parity bit and a storage protect bit for each half word. Under CPU control for direct I/O operations, the I/O channel operates at a worst case rate of eighty-four KHz (derived from reference G).

The CPU communicates with the Master Sequence Controller (MSC) and Bus Control Elements (BCE's) by use of program controlled input (PCI) and program controlled output (PCO) instructions. The PCI/O instructions are two thirty-two bit words. The first word is the command, followed by a data word for the PCO instruction. For the PCI instruction, an IOP data word to the CPU follows. The PCI/O instructions are used to control the CC, the Control Monitor (CM), the Redundancy Management (RM) logic, the Data Flow (DF) logic, and the BCE/MSC local store (LS). The CPU controls the operation of MSC programs by loading the MSC program counter (PC) through use of the PCO LS instruction which starts the MSC at a particular program; the BCE's register can be loaded by the CPU but the MSC must start them with an SIO instruction. The MSC can control which program the BCE will execute by loading the BCE PC and/or base register using the MSC 'LBP' and 'LBB' instructions. The CPU can monitor the activity of the MSC and BCE's by PCI instructions to read the IOP status registers that are set by the hardware and firmware MSC/BCE logic. The MSC can monitor BCE activity by executing a 'repeat until' instruction which waits for the specified BCE(s) to enter the wait state by monitoring the associated bits in the Busy/Wait or BCE-MSC indicator register status. The

BCE indicates to the CPU and MSC that it is finished with a program by executing the 'Wait' (WAT) instruction which resets the Busy bit in IOP status Register 4.

3.2.2 Direct Memory Access

The IOP utilizes the CC for DMA capability to access CPU and IOP memory. The DMA capability is under microcode control to acquire MSC and BCE program instructions. The DMA may also read Multiplex Interface Adapter (MIA) output data from memory and write MIA input data into main memory. The requests for CPU or IOP memory access by the MSC and BCE are queued in a First-in-First-out (FIFO) DMA request stack that can accomodate up to sixty-four requests. The DMA goes into a 'burst' mode of operation when eight or more memory access requests have been stored in the FIFO stack. The 'burst' mode empties the stack, allowing no CPU I/O channel usage during this period. When the I/O channel is operating under IOP DMA, the channel operates at 250 KHz in the normal mode, and 714 KHz in the 'burst' mode. The DMA rate will vary with other processor channel usage which may degradate access speed.

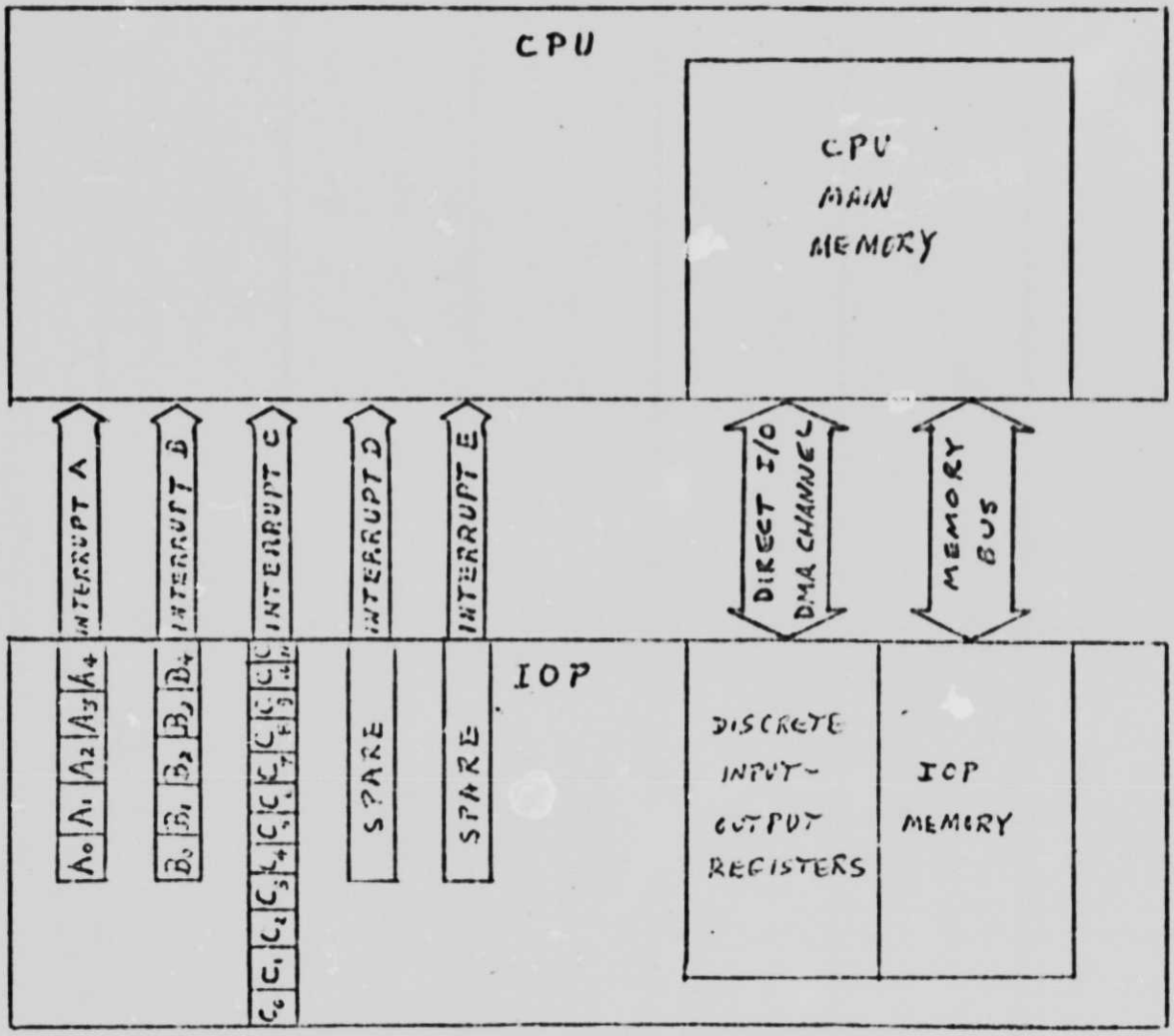
3.2.3 Interrupts

There are five interrupt lines between the IOP and the CPU. The interrupt lines are used to communicate certain IOP hardware detected error conditions and program controlled interrupt requests. Each interrupt line is connected to the wire ORed output of a six bit interrupt register. Any bit set in the register will cause the interrupt, if the interrupt is enabled. The first two interrupt registers (A and B) have their most significant six bits

set by RM error detection hardware. The third interrupt register has twelve programmable bits that can be set by the MSC '@INT' instruction, which includes a mask of bits to be set or reset. The CPU interrupt service routine can determine the cause of the interrupt by examining the proper IOP interrupt status register by use of a PCI command, which will also reset the register bits. The CPU program reads the programmable interrupt register C with a PCI command and interprets the twelve bits as an Effective Interrupt List (EIL), as indicated in reference H. Each bit corresponds to a unique IOP interrupt program number, which usually indicates that a MSC program is finished executing. The last two interrupt registers (D and E) are spares. The interrupts and I/O channel between the CPU and IOP is graphically depicted in Figure 3.

3.3 GPC Intercomputer Discrete Lines

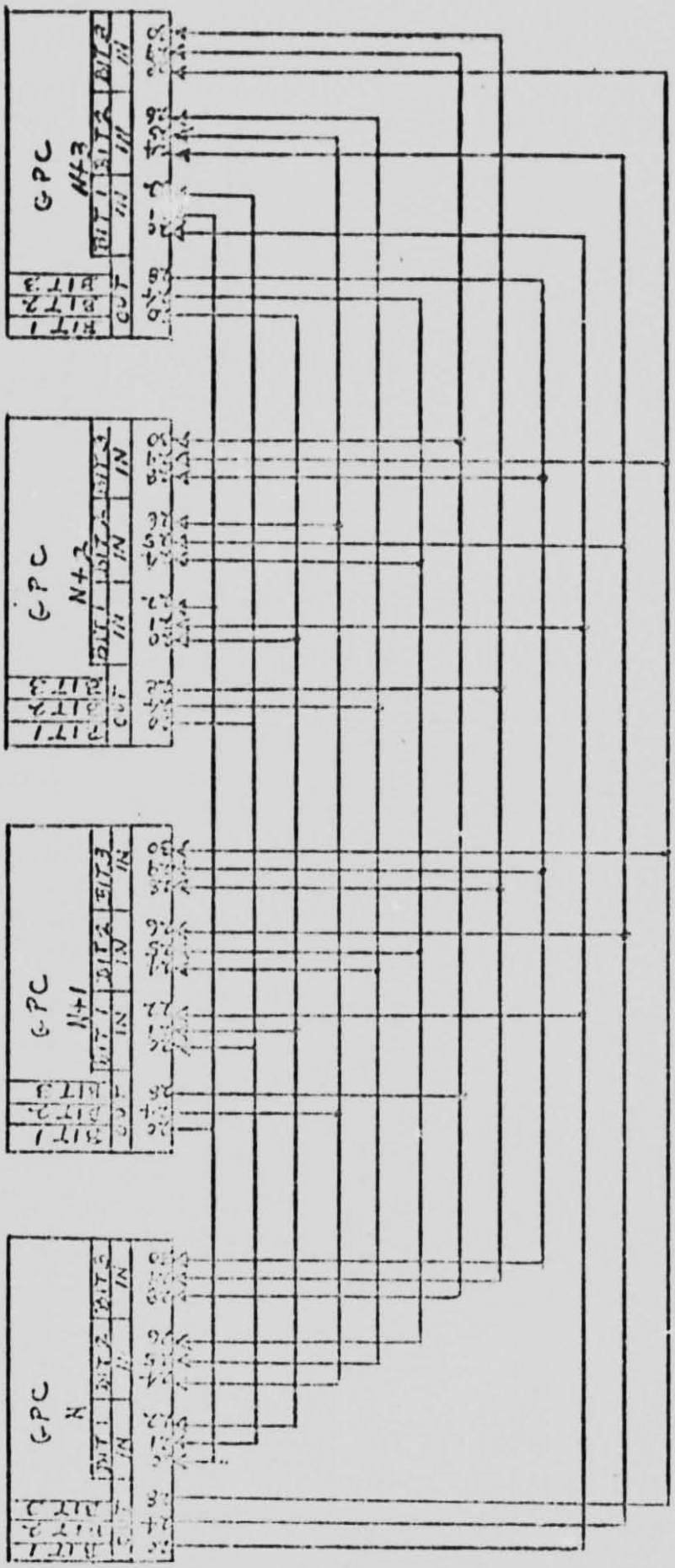
There are forty discrete input and thirty-two discrete output signal lines. Eighteen of these lines are interconnected between the five GPCs and are used to communicate synchronization codes (twelve lines) and RM failure vote status (six lines) to the GPC's (see Figures 4 and 5 respectively). The discrete signal lines are terminated in thirty-two bit registers in each IOP. The IOP Control Monitor controls the loading of the DO register and reading of the two DI registers through CPU issued PCI/PCO direct I/O commands. The discrete input lines are split between two registers, A and B. The first thirty-two bits are in DI register A and the last eight bits are in DI register B. The remaining bits in register B are not used. DI bit positions Ø - 7 and 12-13 are hardwired to the Crew Control Panel (CCP) to indicate Halt, Standby, Run and IPL switch positions. Hardwired bits also indicate Mass Memory Unit (MMU) status, Backup Flight Control System (BFCS) engage, and TERMINATE/NORMAL switch positions.



A_0	GC/NGC TIMER	B_0	PCI/PCO PARITY ERROR
A_1	IOP FAIL LATCH	B_1	DMA INSTRUCTION PARITY ERROR
A_2	CONTROL/MONITOR IDLE	B_2	DMA DATA PARITY ERROR
A_3	READ ONLY STORAGE ERROR	B_3	DMA BURST ERROR
A_4	IOP FAULT	B_4	SPARE
A_5	SPARE	B_5	SPARE

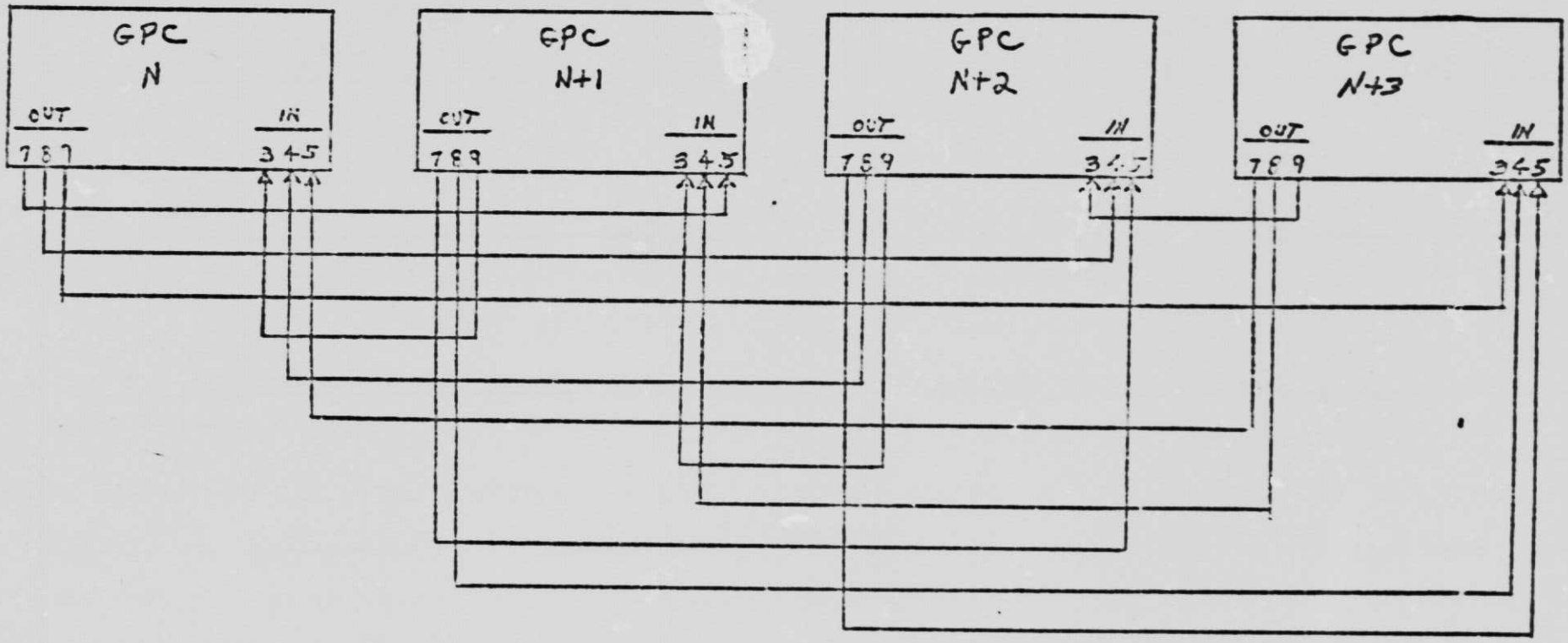
GPC INTRACOMMUNICATION CONNECTS

Figure 3



SYNCHRONIZATION DISCRETE SIGNAL LINES

Figure 4



REDUNDANCY MANAGEMENT FAILURE VOTE DISCRETE SIGNAL LINES

Figure 5

The synchronization discrete signal lines (see Figure 4) are arranged so that GPC 'N' sync bit DO lines 1, 2 and 3 are connected to the DI lines of GPC 'N+1', 'N+2', and 'N+3'. The DI sync lines are grouped by sync bit instead of by GPC. The RM discrete signal lines (see Figure 5) are arranged so that GPC 'N' failure vote 'N+1', 'N+2' and 'N+3' DO signal lines are connected to the failure vote DI lines of GPC 'N+1', 'N+2' and 'N+3', as indicated in Reference J. The RM failure vote discretes are used every minor cycle. The synchronization discretes are used at the slowest rate every minor cycle and the fastest rate is directly related to the use of I/O and Supervisory Calls (SVC's) by the application programs.

3.4 GPC Intercomputer Communication Overview

GPC intercommunication is required to operate a multicomputer, multiprogramming configuration in a coordinated, cooperative manner. The GPC's operate in a coordinated manner by exchanging similar information and by cross checking each other. The GPC's that are executing the same programs are operating in a redundant configuration and should be executing the same instructions at the same time. The GPC's that are operating as members of a redundant configuration must exchange information that allows them to synchronize their program execution. This exchange of information is accomplished by use of dedicated intercomputer communication data channels and discrete signal lines. The cross checking among GPC's by ICC data exchange also allows the GPC's to detect an errant GPC and remove it from the redundant set in order to keep the Data Processing System operating correctly.

3.4.1 ICC Data Buses

The main interconnection between the GPC's is the five ICC Data buses. The ICC data buses consist of twisted, shielded pairs. Each GPC commands a

dedicated ICC data bus that is interfaced to the other four GPC's. The ICC data is transmitted in serial Manchester II code at a rate of one million bits/second. Each ICC data bus interconnects an IOP MIA of each of the five GPC's. GPC's transmit serial data over the data bus and can process simultaneous data transmissions on the ICC data bus set. The GPC's communicate over the ICC data buses by transmitting twenty-eight bit command words, command data words, or response data words. Each IOP has five BCE's that are dedicated to the five ICC data buses. Each GPC is commander of one of the five buses and is a listener on the other four buses. The commander preceeds each message transmission with a "Listen Command" which is a command word with an interface unit address (IUA) of 01000 binary. The transmitting BCE sends its ICC message when the receiving BCE goes into the 'receive' mode of operation.

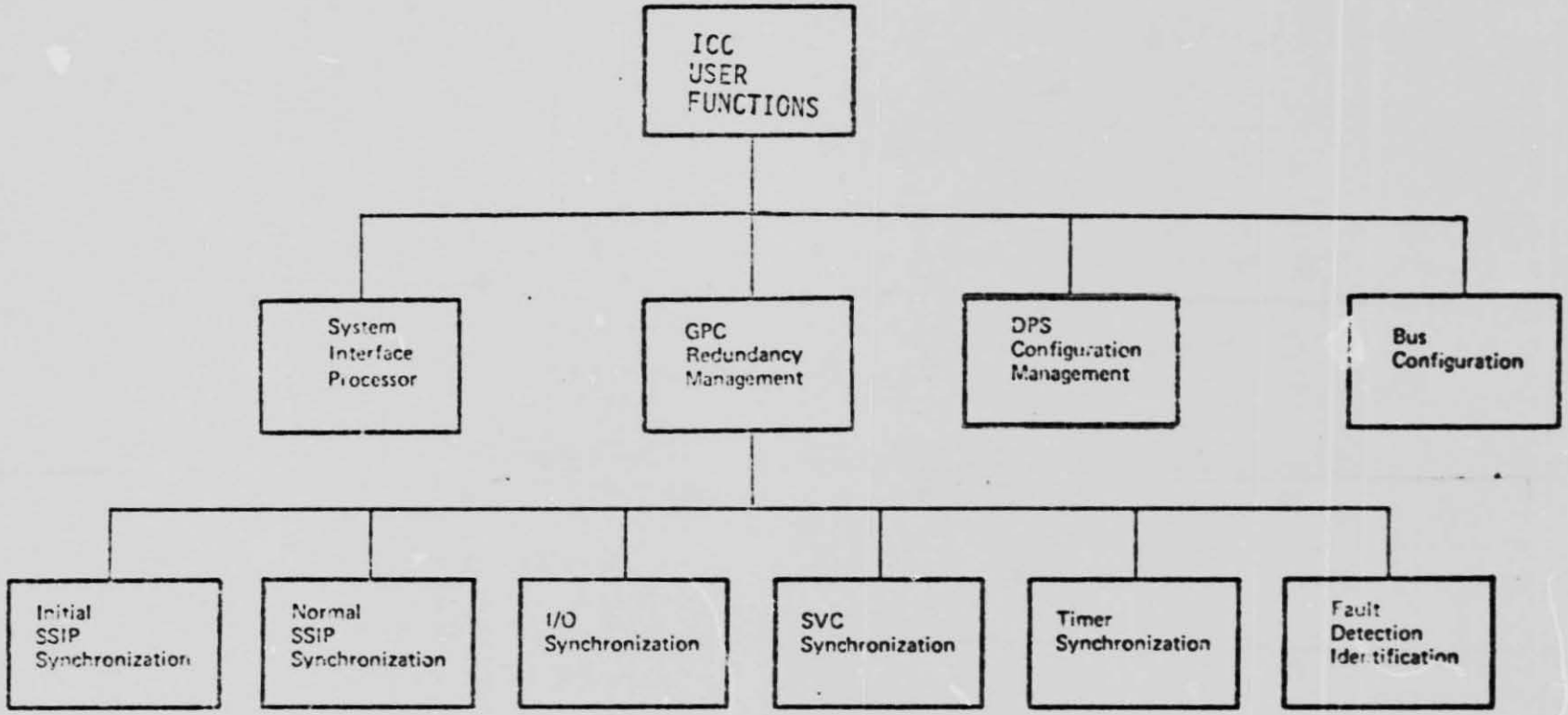
The transmission process begins by either the CPU or MSC selecting the BCE associated with the ICC channel. The BCE sends a 'listen' command with a common address that will start the other BCE's on the bus that are executing a Wiat-for-Index (WIX) instruction. The index is used to calculate the address of the instruction that will process the data sent by the command BCE.

3.4.2 ICC Traffic Categories and User Functions

There are two distinct categories of GPC intercommunication-serial digital and discrete. These modes of GPC intercommunication can be used separately or in conjunction with each other. The serial digital intercommunication is effected by use of ICC data bus messages which are converted from parallel twenty four bits words (sixteen data bits plus overhead), transmitted using a serial modulation scheme, and re-converted to a parallel twenty-four bit data word at the receiver.

The serial digital ICC traffic is used to convey messages containing one or more words of data between the GPC's. The forty DI and thirty-two DO signals lines are used to indicate the status of switch positions or other set/reset, enable/disable, on/off two-state conditions that can best be intercommunicated using single bits or groups of bits. The ICC traffic is used for communication between GPC's that are members of a common set (CS) or a redundant set (RS) of GPC's. The CS traffic is intended for exchange between active GPC's that are in either the Standby or Run mode and includes messages that relate to GPC processing that is not controlling flight critical buses and related functions. The RS traffic is intended for exchange between GPC's that are cooperating and coordinating the control of the flight critical data buses and other related control functions. The CS includes the RS, but the RS does not necessarily include the CS.

Figure 6 shows the ICC user functions. One of the major users of ICC communication traffic is GPC Redundancy Management, which is composed of GPC synchronization processing for both the CS and RS and Fault Detection Identification (FDI) processing. There are four basic synchronization procedures, which include System Software Interface Processing (SSIP) synchronization (CS), and Timer, I/O and SVC synchronization (RS). Synchronization is required only if two or more GPCs are active, and establishes a common starting point in the same program being executed by different GPC's. FDI processing, part of I/O completion processing, votes the RM sum words generated by each member of the RS. The System Interface Processor uses the ICC channel to exchange information relative to CS member GPC's. DPS configuration management uses the ICC channel to communicate changes necessary to add or delete members of the common and redundant set. Bus configuration changes



ICC USER FUNCTIONS

Figure 6

are communicated on the ICC channel among RS members whenever there must be a change of GPC primary or secondary bus command.

3.4.3 ICC Traffic Initialization

The ICC traffic initiation for the RS and CS are derived from the same programmable counter but, due to the queuing stack for the clock interrupts, occur at different times. The CS ICC traffic, for one or more GPC's, occurs at every SSIP synchronization point and is scheduled by the GPC Locator program to occur every forty milliseconds by placing a Timer Queue Element (TQE) in the timer queue stack. The RS traffic is initiated at every Programmable Timer 2 (PC2) interrupt, whereas the CS traffic is initiated only if the top active TQE is a SSIP synchronization request.

The ICC I/O traffic is initiated by a call to the Flight Computer Operating System (FCOS) SVC service routine which dispatches the I/O request to the MSC control program to initiate needed BCE programs. BCE programs are designed for a specific interface unit address. The BCE program executes in either the commmand or listen mode and the command BCE sends a 'listen' command to 'listening' BCE's. BCE programs expect a fixed number of ICC message words to be transmitted. The index from the 'listen' command determines the BCE Program to execute which in turn determines the number of words to receive.

3.4.3 ICC Message Handling

ICC messages are handled by three software functional groups-message collection,message routing, and message dispatching.

The collector and router are part of the User Interface (UI) software. The ICC message collector provides support and buffering of individual ICC messages from System Control (SC) programs, such as GPC Reconfiguration. The individual messages are collected into a single message block comprised of coded modules. The ICC message buffer is a 128 x 16 bit buffer in the common pool part of GPC memory. There are five copies of the buffer in each GPC; one for each GPC in the common set. The ICC message buffer for GPC 'N' is used to store messages input from other GPC's, whereas, ICC message buffers 'N+1', 'N+2', 'N+3', and 'N+4' (modulo 5) are used to store messages to be output to those GPC's. The first seven data slots in each buffer are fixed while the remainder are used for variable length messages. After the message is transfered to the ICC buffer, indices are updated so that the process will accept additional messages, if there is room.

Once the ICC messages have been exchanged between the common set GPC's, the ICC message router is invoked to distribute the received message, from the four input buffers, to the specified processes for which they are destined. The ICC Message Table (IMT) contains control data the router uses for processing the packed messages. The input ICC buffers are scanned and a comparison made on message destination and type codes. Depending on the IMT entry, the message may be processed immediately, scheduled for later processing, withheld from moving, or moved to a buffer invoking no processing. To avoid redundant processing, redundant messages can be filtered out. During interface processing, the ICC collector is called to finish processing any message. Next, ICC messages are built for each common pool (COMPOOL) ICC flag that is set and the ICC message collector is called to place the

the ICC messages in the ICC message buffer. The flags are reset if the ICC message collector accepts (has room in ICC buffer) the message.

After all messages are built, a SVC is issued to invoke ICC output and input traffic. The I/O SVC Handler (F1OSVC) attempts to start I/O operations by passing an I/O Queue Element (IOQE) to the IOP Dispatcher (FIODISP) which in turn initiates the request by passing a mask of buses needed in the request to the MSC Control Program (F1OMCNTL). FIOMCNTL starts the proper BCE's to executing BCE programs based on the input I/O monitor mask. The BCE programs are designed for a specific interface unit address and execute in either command or listen mode. The command BCE sends a 'listen' command to listening BCE's. The 'listen' indices are used to determine transfer word count for ICC requests. When the ICC traffic exchange is completed, the ICC message router is called to distribute the received ICC messages.

3.5 GPC ICC Synchronization Features

Synchronization provides system wide, time coordinated sampling of BTU's, use of the same data sets for RS GPC processing, coordinated commands to initiate subsystem actions, and coordinated GPC information exchanges and associated processing. The synchronization process involves the intercomputer exchange of three-bit binary sync codes over the discrete signal lines. GPC 'N' outputs its discrete code by loading the DO register bits 20,24 and 28. The sync codes from the other GPC's are input on the DI signal lines. The sync bit from GPC's N+1, N+2, and N+3 (modulo 4) are input on DI lines 20-22, likewise sync bit 2 on DI lines 24-26 and sync bit 3 on DI lines 28-30.

The synchronization processes have a priority order. Timer sync and SSIP have highest priority, I/O sync is next, and SVC sync has lowest

priority. This priority structure allows the I/O sync process to interrupt the SVC sync process, and SSIP or timer sync to interrupt either I/O or SVC sync processing. The priority structure is used to prevent sync failure due to different GPC's being at different sync points at the same time. This could occur due to a processing time skew which would allow one GPC to arrive at a higher priority sync slightly ahead of the other RS GPC's. Lower level sync processes must enable higher priority sync process interrupts when there is a sync code discrepancy.

The detailed synchronization functions are discussed in the following subsections. All functional flow charts (Figures 7-15) are extracted from Reference D and are included in this design note for completeness.

3.5.1 Initial SSIP Synchronization

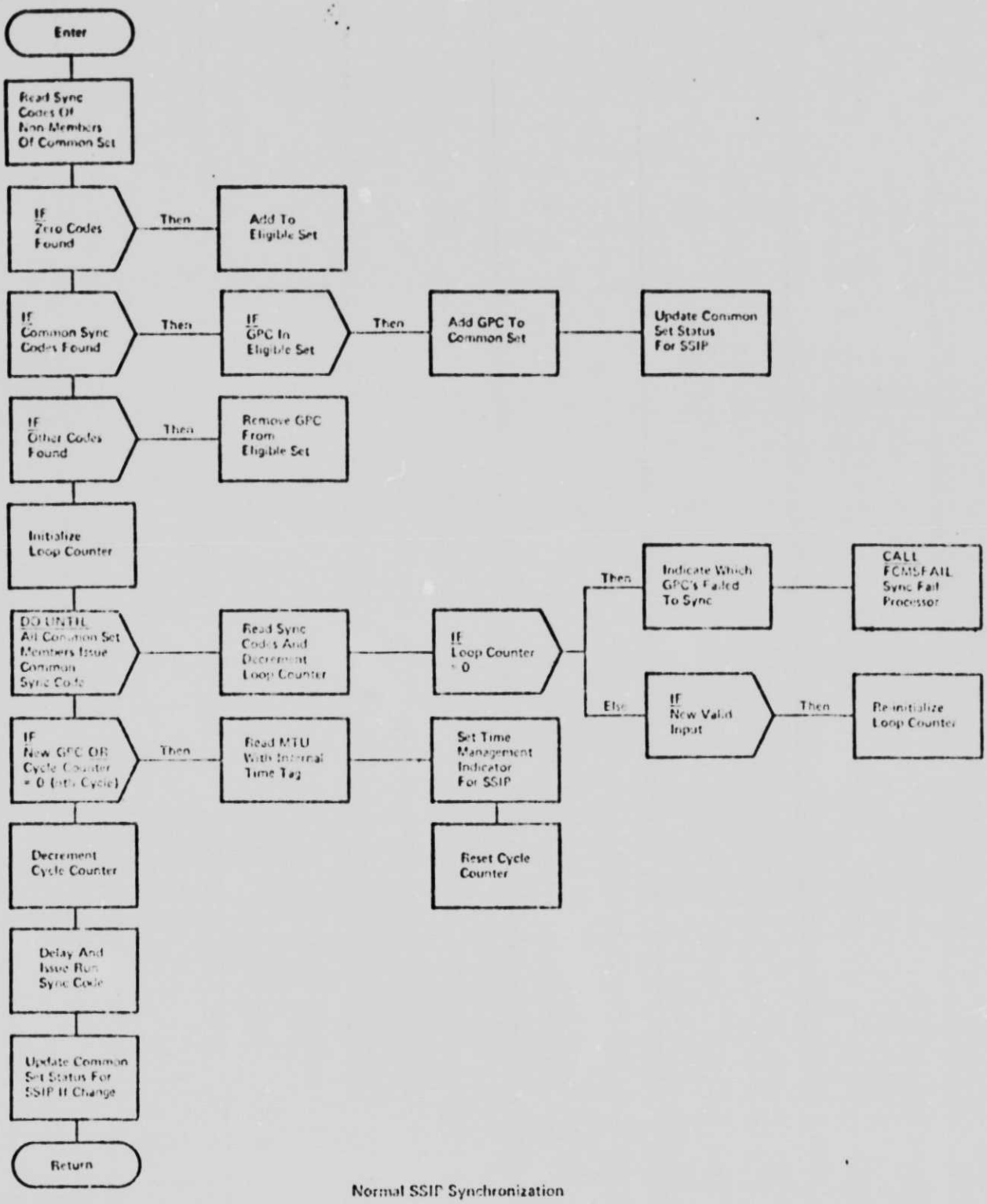
Initial SSIP synchronization is part of the GPC initialization SSIP processing and is used to determine active GPC's and add them to the common set (see Figure 7). The process involves letting the common set sync codes, read from the sync discrete lines, dissipate. The sync discretes are then read for a period of fifty milliseconds or until a GPC sends a common sync code. If no common sync code is detected, the discrete sync code lines are read for ten additional milliseconds to allow other common set members to report. Any additional common set members determined are added to the common set sync mask.

3.5.2 Normal SSIP Synchronization

Normal SSIP synchronization (see Figure 8) is scheduled to execute at every minor cycle. SSIP sync processing gains CPU control when its TQE time expires. The PC2 interrupt service routine counts out the scheduled time interval. The sync codes of non-common set members are read and these GPC's are allowed to join the CS from the HALT state. The sync code discrete lines

Initial SSIP Synchronization

Figure 7



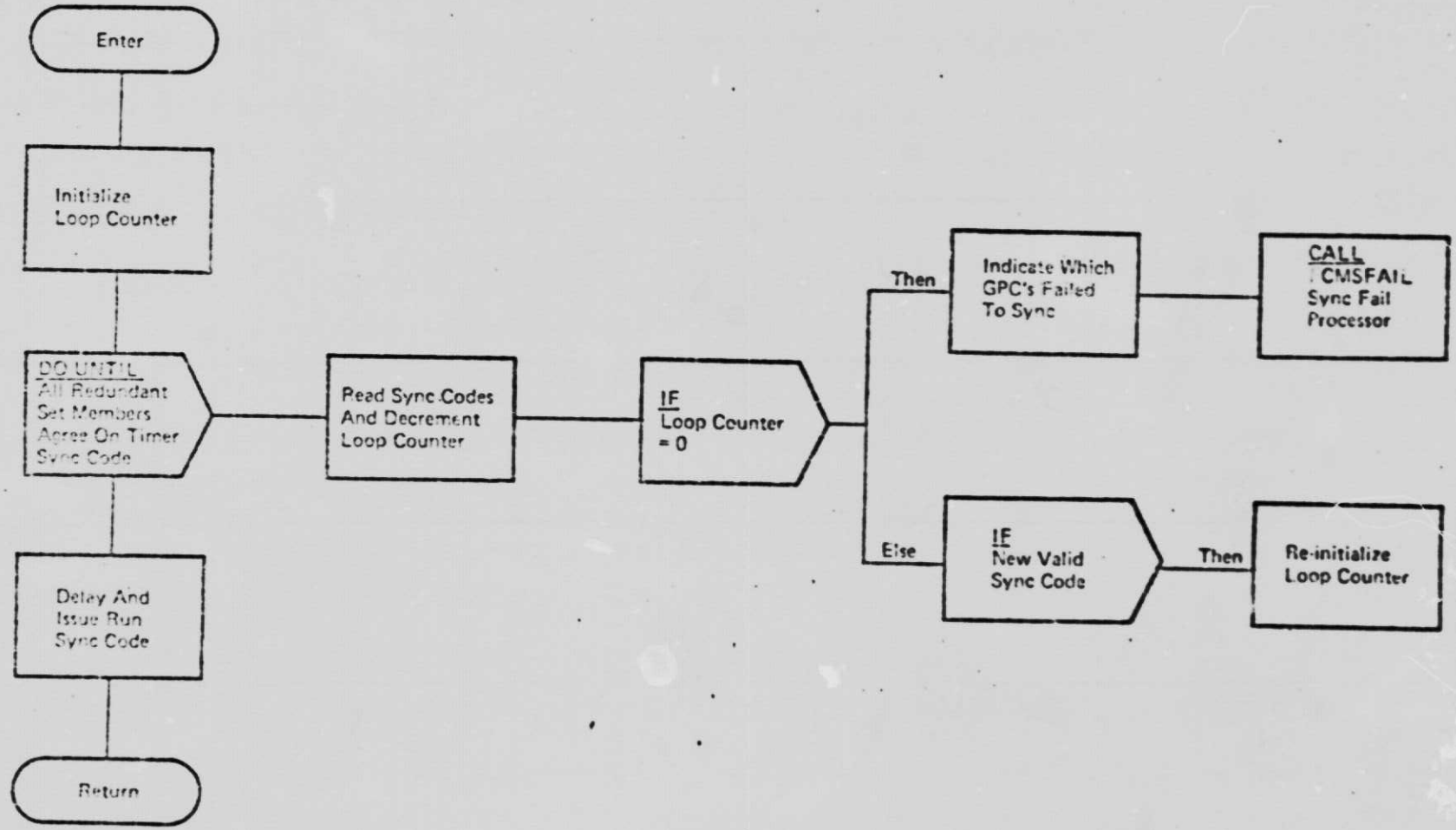
Normal SSIP Synchronization

Figure 8

of all CS members are read and checked for the common sync code (100), or until a Time Counter (TC), set to a COMPOOL value, times out the sync check process. Any CS member failing to issue the common sync code in the alloted time, is removed from both the common and redundant sets by the Sync Fail Processor. Since the TC is re-initialized each time a previously missing CS member issues the common sync code, worst case time for CS sync would be (TC value)x(number of CS members). When a CS member has failed a CS sync, it is ignored at all further CS and RS sync points. Data bus re-configuration may be necessary when a RS member controlling certain data buses is removed from the RS. Resetting the TC value during synchronization assures that all CS members leave the sync point together. If a new GPC is added to the CS or if the minor cycle counter was decremented to zero on the previous cycle, then the Master Timing Unit (MTU) is read using a pre-initialized IOQE issued via an SVC, the Time Management Processor (TMP) flag is set for SSIP, and the minor cycle counter is reset to maximum value. Before exiting the normal SSIP process, the minor cycle counter is decremented and the null sync code (111) is issued on the sync DO lines.

3.5.3 Timer Synchronization

Timer synchronization (see Figure 9) occurs at every PC2 interrupt except when the interrupt expires the SSIP TQE. The time sync code (101) is issued from the TQE expiration routine. The DI sync lines of the RS members are read and checked for the timer sync code or until a time counter, which has been set to a COMPOOL value, is decremented to zero by the loop execution time. Any RS member that does not issue a timer sync code within the allotted time is deleted from the RS by the sync fail processor (FCMSFAIL).



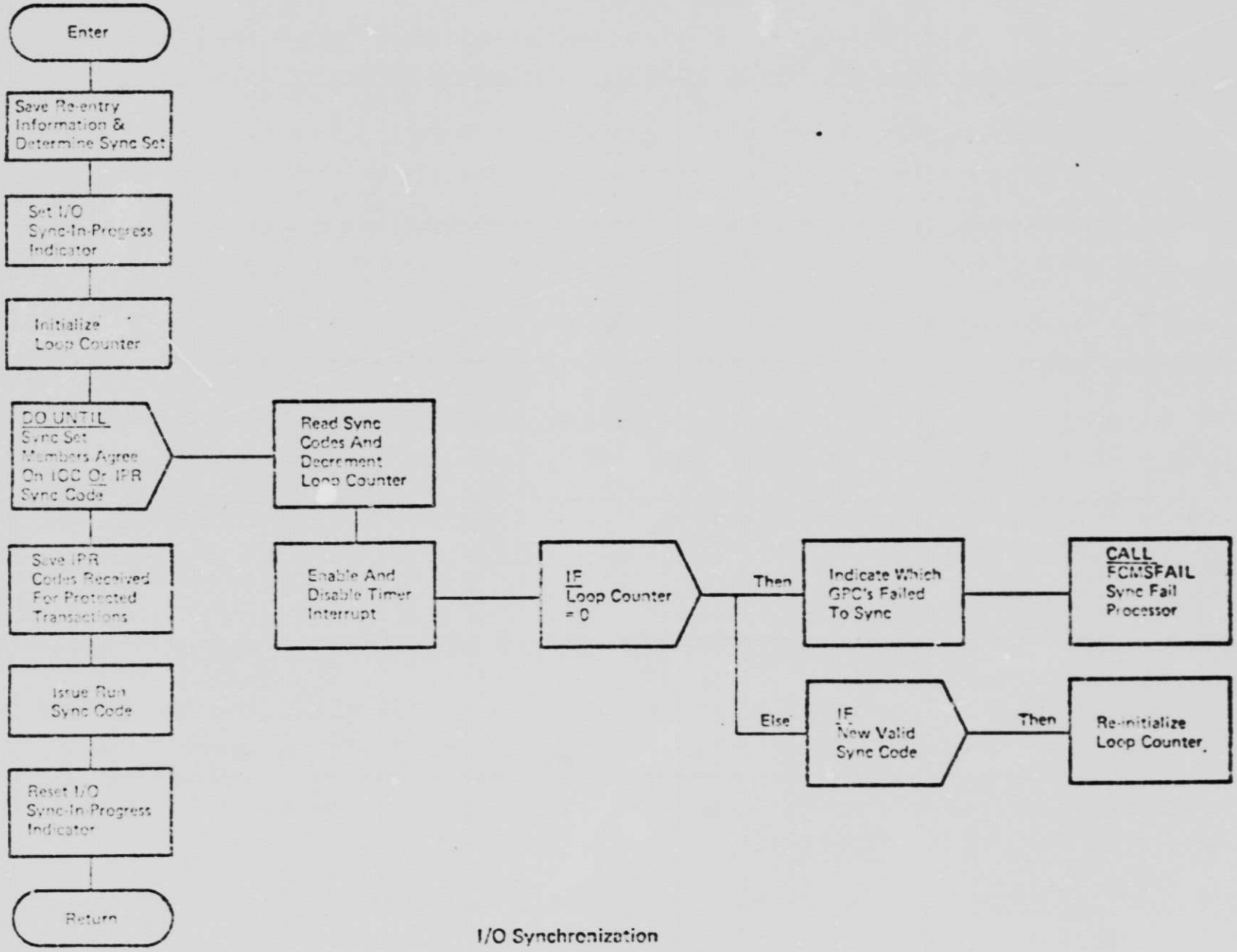
Timer Synchronization

Figure 9

When all RS members have issued the timer sync code on their DO lines, the DI register is read and the sync lines checked again for stability. If a timer sync code is detected from a GPC previously issuing a invalid sync code, then the time counter for the sync read loop is reset, thus allowing the GPC's to leave the sync point together. A short delay is processed to allow sync code detection by other GPC's. The sync fail processor will initiate updating of bus/data path masks and CALL the Bus Reconfiguration routine for buses commanded by a fail-to-sync GPC. A flag is set in the ICC Flag COMPOOL area to indicate that the RS mask has been updated and a message should be generated on the next ICC message exchange cycle to communicate the RS mask change to the other GPC's. The null sync code (111) is issued on the DO sync lines before the time sync routine returns through the PC2 interrupt handler.

3.5.4 Input/Output Synchronization

I/O Synchronization Processing is used to synchronize I/O completion (IOC) interrupts between the GPC's (see Figure 10). The interrupt is initiated by execution of the MSC '@INT' instruction. The programmable interrupt instruction is executed by the MSC when it has detected that an I/O task has been completed by a monitored BCE. The I/O synchronization process involves synchronization between either the CS or RS depending on the sync type bit set in a half word of the IOQE, which provides the control information for a single I/O request. The I/O completion processor gains control of the CPU by a Program Status Word (PSW) swap, which in turn calls the I/O synchronization process.



I/O Synchronization

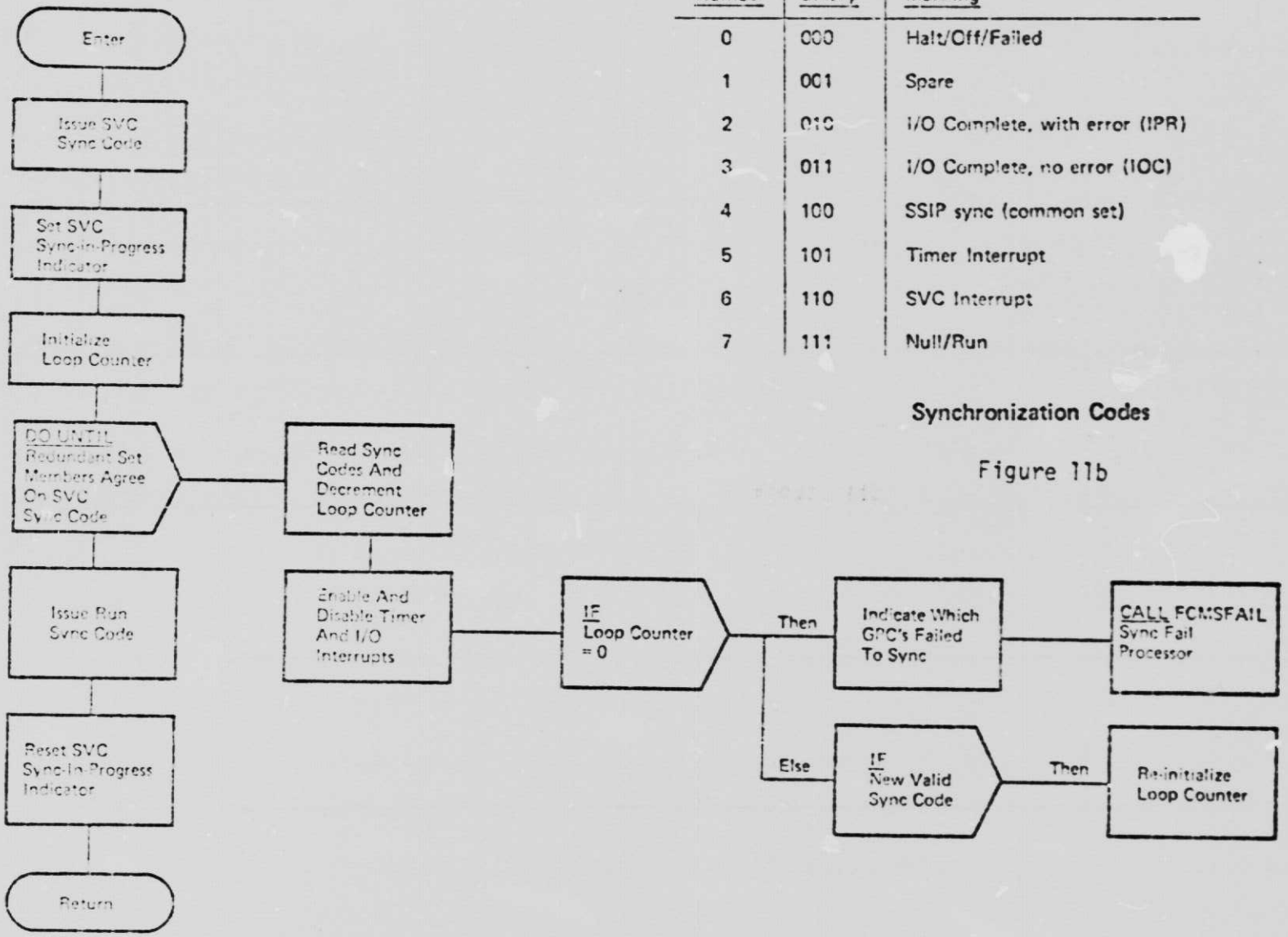
Figure 10

An IOQE flag bit indicates if the completed transaction had errors. The IOC sync code (010) is issued if no I/O errors occurred or the Input Problem Report (IPR) sync code is issued on the DO sync lines (I/O errors are indicated). The process goes into a loop reading the DI register and checking for IOC or IPR sync codes sent from the members of the syncing set (CR or RS). A time counter decremented by the loop execution time, is set to an FCOS COMPOOL value to time the syncing process. If the correct sync codes are not read in the alloted time, the sync fail processor (FCMSFAIL) is called to remove the fail-to-sync GPC from the syncing set and perform any necessary bus reconfiguration. The DI register is read again to check the sync codes for stability. If the two reads are not identical or the sync codes are not IOC or IPR then the timer interrupt is enabled, then disabled, to allow a possible RS time synchronization process to gain CPU control. If no interrupt occurs, then the last sync codes input are checked for a correct sync code from a GPC previously issuing an incorrect sync code. A new correct sync code will reset the time counter and start the sync code input loop from the beginning. If after two identical inputs, the sync codes are IOC or IPR, then a delay is processed to give the other GPC's time to receive the correct sync code. The sync codes are again input for a third time and if the sync codes are IOC or IPR, then the process exits the input loop. Next, the null sync code (111) is issued by loading the DO register. IPR codes received from any of the GPC's are saved for later protected transaction processing. If the IOQE device ID indicates ICC for IPR, then an ICC/IPR IOQE is queued. The I/O sync in progress flag is reset to zero before control is returned to the I/O completion processor.

3.5.5 SVC Synchronization

The Supervisor Call synchronization process is the lowest priority synchronization process and is invoked at SVC interrupts which request data gathering or queue manipulation functions (see Figure 11a). The interrupt causes a PSW swap that puts the CPU in the supervisory state from which program controlled I/O (PCI/O) can be performed.

The SVC synchronization process begins by setting the SVC sync-in-progress flag so that if higher order synchronization processes (I/O and timer) interrupt, the SVC may be re-issued. The DI register is read and the sync code lines of all RS members checked for the SVC sync code (110). If the SVC sync code is not received, a time counter, set to a UI COMPOOL value, will terminate the synchronization process. If the SVC synchronization process times out, the fail-to-sync GPC's are removed from the RS. The DI register is input twice to check for stability. If the two inputs are identical and all the sync codes are SVC, then a short delay is processed and the DI register is input and checked a third time. If sync codes are correct, after the third read, the null sync code (111) is loaded into the DO register, the SVC sync-in-progress indicator is reset, and the program exits. If the two input reads were not identical or if all the sync codes were not SVC, then the timer and I/O interrupts are enabled and then disabled. If no interrupt occurs, then the last DI register input is checked for new correct sync codes reported from previously incorrect GPC's. If a new correct sync is detected, then the input timer counter is reset to maximum value and the DI register input loop is started from the beginning, thereby enabling the GPC's to exit the input loop together. The synchronization codes are summarized in Figure 11b.



SVC Synchronization

Figure 11a

Number	Binary	Meaning
0	000	Halt/Off/Failed
1	001	Spare
2	010	I/O Complete, with error (IPR)
3	011	I/O Complete, no error (IOC)
4	100	SSIP sync (common set)
5	101	Timer Interrupt
6	110	SVC Interrupt
7	111	Null/Run

Synchronization Codes

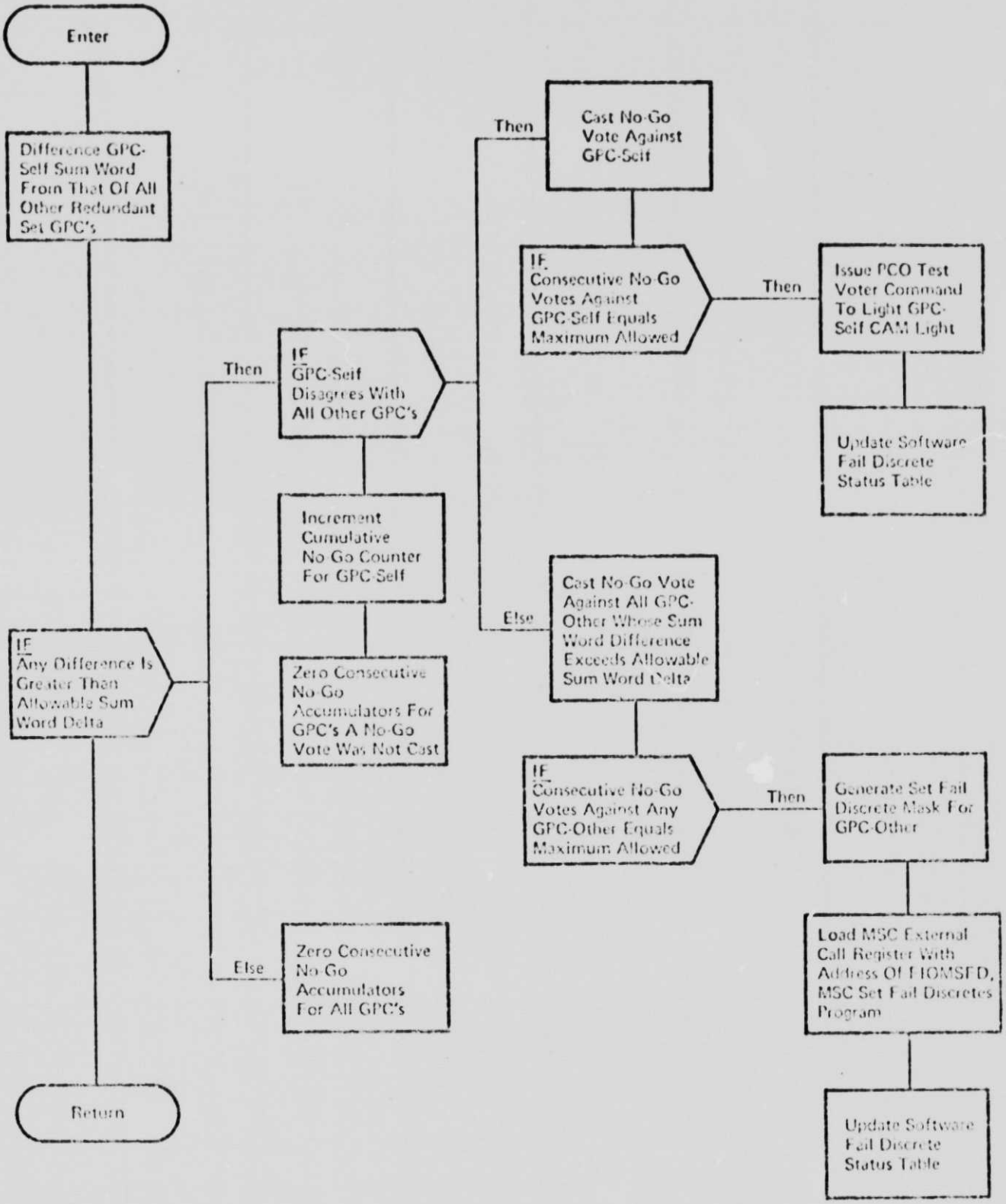
Figure 11b

3.6 Fault Detection Identification

Another form of RM is Fault Detection Identification (FDI) processing (see Figure 12). FDI votes on the RM checksum words, which are calculated by the Guidance, Navigation and Control (GN&C, Fast Executive module, for specified critical output commands. These sum words are transmitted on the ICC data buses between RS members so that every GPC receives one less sum word than the number of GPC's in the RS. Each RS GPC compares its sum word with the other GPC sum words and sets a NOGO vote, against itself or the other GPC's, whenever there is a sum word difference. GPC counters are transferred to other GPC's by ICC message once per second. Nominally, three consecutive NOGO opinions of a GPC are voted before the associated GPC DO register fail vote bit is set via the appropriate bit set in the Set Fail Discretes Mask of the Communications Vector Table (CVT). The RM sum word is the first thirty-two bit word in each ICC Buffer.

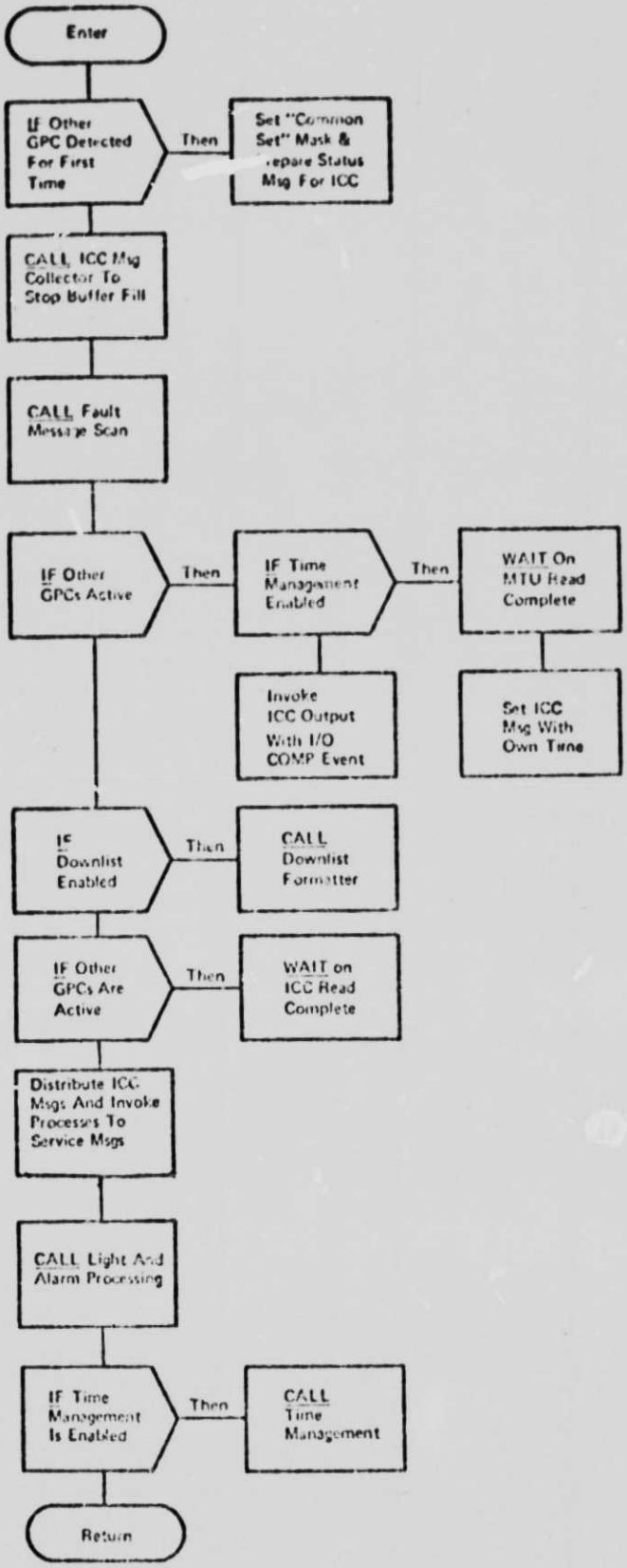
3.7 System Interface Processor

Another major user of ICC is the System Interface Processor (SIP) which provides control for system-wide processes which are required to operate simultaneously in a coordinated manner in the CS member GPC's (see Figure 13). The SIP is scheduled by the GPC Location process to execute every minor cycle when critical applications processes are not active, unless the GPC is in the HALT mode or turned off. The CS mask is used to determine if other GPC's are active, and if so, a wait for ICC I/O is processed. The RM sum word is moved to the ICC buffer, then the ICC message collector is forced to terminate processing of any message interrupted by SIP. An ICC message is built for each active GPC for every ICC flag bit that is set, indicating an updated value or status. The bits are set by applications and control processes between minor cycle points. Once the messages are



Fault Detection Identification

Figure 12



System Interface Processor

Figure 13

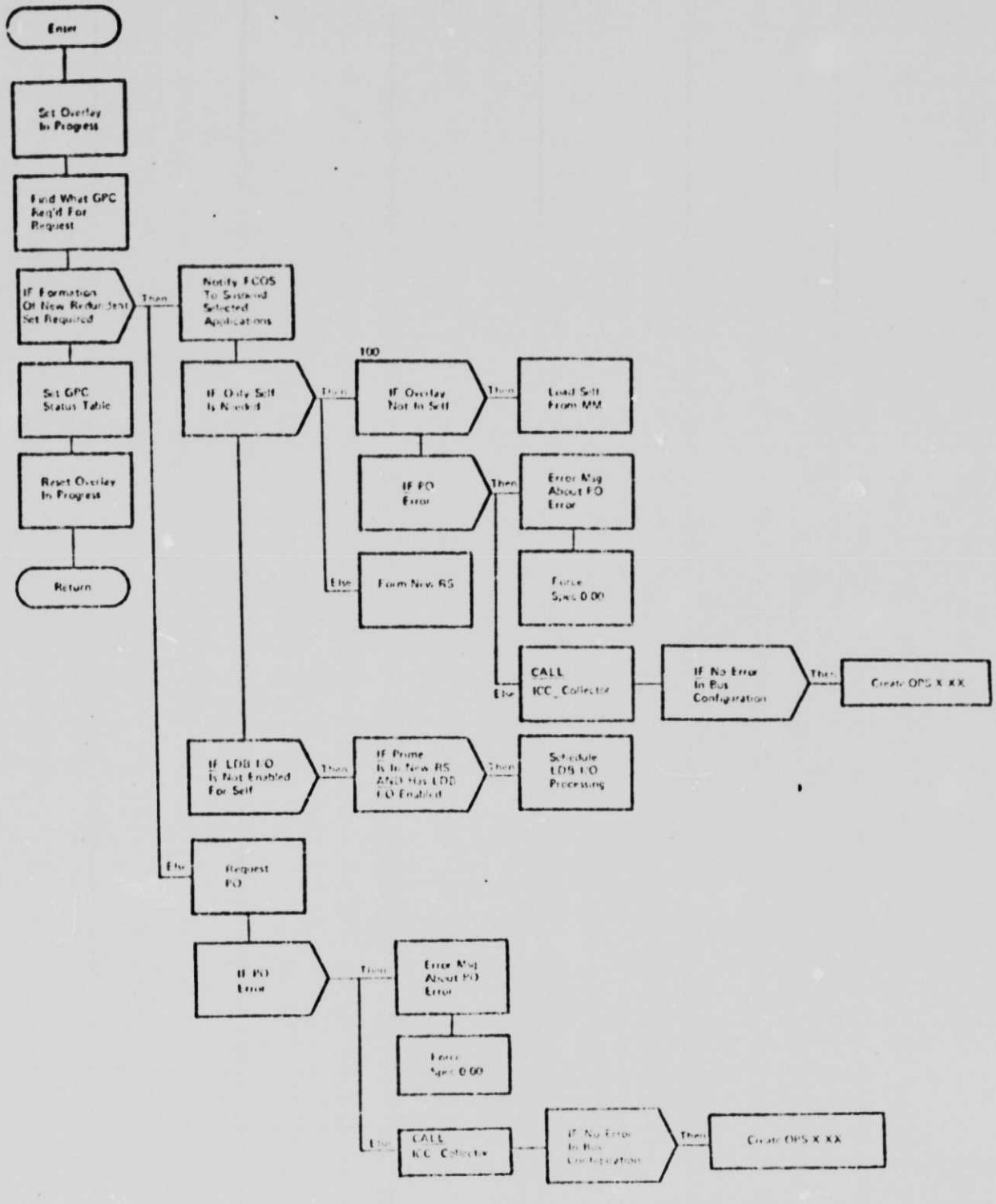
built, an FCOS SVC is issued to initiate the ICC message transmission. The data transferred can be self status for other GPC's, internal clock time (when time management function is invoked after sync routine), keyboard input configuration change, systems status data for display, system software logic control parameter status, annunciation common for all memory configurations, and mass memory I/O contention coordination data. The ICC router processes the GPC's own ICC buffer and any ICC buffers received from other active GPC's when ICC message transfer terminates. The ICC buffer controls are reinitialized for the next minor cycle.

3.8 DPS Reconfiguration

DPS reconfiguration also utilizes ICC and is composed of GPC Reconfiguration and Data Bus Reconfiguration. Reconfiguration is initiated by user inputs which specify the state to which the DPS and its elements must be modified. The prime GPC controls the reconfiguration of GPC's while any other GPC involved in the process operates under the direction of the prime GPC. GPC reconfiguration is necessary when a major function Operational Sequence (OPS) transition requires a change in the GPC RS to support the OPS.

3.8.1 GPC Reconfiguration

The GPC reconfiguration processor services requests to overlay main memory with new memory configurations and controls the forming of new sets of redundant GPC's (see Figure 14). GPC's in the CS which are needed in the new RS for the new OPS receive an ICC message from the prime GPC communicating the impending changes in the DPS configuration. Messages transfered via ICC to initiate and control reconfiguration in secondary GPC's are of five types. The first type



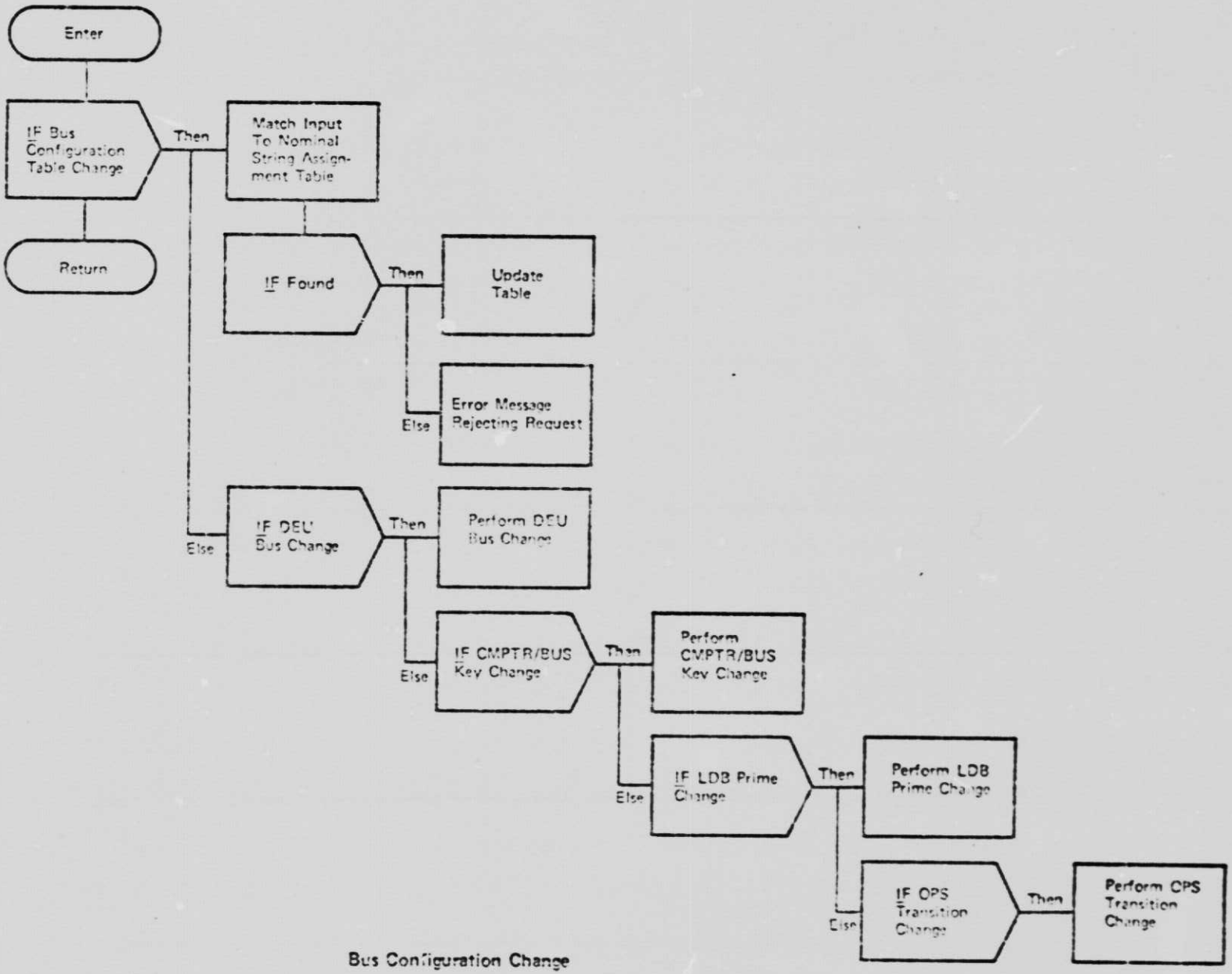
GPC Reconfiguration

Figure 14

tells new GPC's for the RS being formed that reconfiguration is about to take place. The second type message tells the new GPC's for the RS being formed that a new OPS is to be initiated. The third type message tells new GPC's for the RS being formed that the overlay for the new OPS is to proceed now. The fourth type message is the reconfiguration MMU overlay completion status, and the fifth type message sends application process initialization data to new GPC's in the RS. ICC messages are also built to control RS initiation and GPC sync for memory overlay initiation. The new RS of GPC's is synchronized after which the MMU read is requested of all GPC's. The prime GPC commands the MMU load and transmits MMU 'listen' instructions via ICC channel message to the secondary GPC's. After the MMU load, system initialization ICC messages are transmitted to the secondary GPC's. The messages are lists of processes that must continue operation across the OPS transition and are resident in the major function (MF) overlay.

3.8.2 Data Bus Reconfiguration

The data bus configuration change process provide a means for changing command of a data bus from one GPC to another (see Figure 15). One GPC releases command of a bus, while another GPC assumes command of that bus. The bus configuration change process causes the appropriate BCE to be set to 'listen' or 'command'. A bus reconfiguration message is passed to the ICC Message Collector and issued at CS synchronization time to all other active GPC's. Upon completion of the ICC, the ICC Message Router distributes the bus reconfiguration message to the Bus Configuration Change Processor.



Bus Configuration Change

Figure 15

The ICC buffer number and array index point to the message to be processed. The DI register A data in the ICC buffer is used to determine the GPC RUN/ STBY status.

The bus configuration messages can originate from major function switch change, computer/CRT key input, DPS Configuration Monitor Display (formally executed by the computer/bus key input), Launch Data bus (LDB) commander or bus change, new OPS request, OPS terminate request, Force OPS 0 request, or fail-to-sync request. Bus configuration tables, the DPS Status table, the GPC Status table, and bus/data path masks and counters are updated and this information is transfered to all GPC's via ICC message so they can update their corresponding tables at the conclusion of the processing.

4.0 CONCLUSION

The Orbiter Data Processing System intercomputer and intracomputer processes have been described in sufficient detail to be understood without reference to other Shuttle documents. Based upon this description, it is noteworthy to emphasize that intercomputer communication processing is clearly one of the most complex areas of this multicomputer, multiprogramming system and serves the most critical function of redundant set GPC operation. Synchronization and the associated ICC message traffic require extremely complicated and time critical logic, which is by its nature subject to potential errors. The basic design has the potential of correct operation but should be considered suspect if any timing problems occur during testing.

The timing of ICC message traffic is critical to the operation of the redundant set. Variations in response to interrupts, the processing time between the interrupt and the resulting I/O operation, may provide sufficient time skew between RS GPC's to cause a transmitting BCE to send a message before the receiving BCE is initialized to receive it. More analysis and some laboratory measurements will be needed to verify that no timing problems exist.

Frequent synchronization of the redundant set (500 to 600 synchronizations per second) may cause a CPU execution time overload. The possibility of combining applications "calls" for I/O to reduce the number of redundant set synchronizations should be considered as a means of reducing the CPU overhead associated with I/O and synchronization.

5.0 REFERENCES

A. Computer Program Development Specification Volume I, Book I, System Level Description, Hardware, Level A (SS-P-ØØØ2-11ØA) Dated June 26, 1975

B. Computer Program Development Specification, Volume I, Book 2, System Level Requirements, Software, Level A, (SS-P-ØØØ2-12ØB), dated September 26, 1975

C. Data Processing Subsystem Description and Performance Document (SD74-SH-Ø23ØB), dated February 27, 1976

D. Approach and Landing Test, Functional Design Specification, Volume II - Systems Software (75-SS-0714), dated 7/21/75

E. Approach and Landing Test, Detailed Design Specification, FCOS, User Interface, and System Control (76-SS-0929), dated 1/30/76

F. Space Shuttle Advanced System 4/PI Model AP-101 Central Processor Unit, Technical Description, (75-A97-001), dated 3/31/75.

G. Space Shuttle Advanced System/4 Pi Prototype Input/Output Processor (IOP), Functional Description (74-A31-016), dated 10/25/74

H. Space Shuttle Advanced System/4 Pi Prototype Input/Output Processor (IOP), Principals of Operation for PCI/PCC, MSC, and BCE (6246556), dated 12/20/74.

I. IBM System/4 Pi Model AP-101 C/M Principals of Operation, IBM No. 6246156, dated 12/15/75.

J. A study of Discrete Control Signal Fault Conditions in the Shuttle DPS prepared by W. W. Gaertner Research, Inc., Contract NAS 9-14703, Project No. 4052, dated 6/30/76.