Honeywell

TDC 4500 GENIE* BUS MAINTENANCE

ACPU1GBC-M
8/19/79

TABLE OF CONTENTS

*Trademark

TDC 4500 GENIE* BUS MAINTENANCE

1. INTRODUCTION

This publication provides maintenance-related information for the GENIE Bus Controller (GBC), including the Bus's Extender/Switches and the I/O interrupt capability for TDC 4500 process control systems.

2. OPTIONS

The GENIE Bus Controller is a functional component of the basic CPU. No options are associated with it except for related assemblies and functions, such as the chassis that house device controllers connecting to the GBC.

One I/O chassis is included as part of the CPU, model no. ACPU11. CPU model ACPU12 adds a second I/O chassis and a second power chassis. Models ACPU13 and ACPU14 add third and fourth I/O chassis.

3. REFERENCES

Test Program:

51103002 - GENIE Bus Test

Tables 1, 2, and 3 contain pin numbers for the GENIE Bus, CPU Bus, and console interface, respectively.

4. TEST EQUIPMENT

The general test equipment related to maintenance for the computer system is listed in Section 1 of this manual. In addition to this, the 4400AM131 GENIE GBC Test Aid is available for checkout of the GBC. The Test Aid is a PWA that plugs into a chassis of the GENIE Bus and can be exercised by program control (including the Test Program associated with the GBC) to verify the GBC's operation.

5. COMPONENT LOCATIONS

The GENIE Bus Controller occupies two card slots in the CPU chassis, slots 7 and 8.

6. PREVENTIVE MAINTENANCE

Preventive maintenance is not required for this function.

7. PERFORMANCE TESTS

Performance of the GENIE Bus Controller can be verified by use of the GENIE Bus Test 51103002, in conjunction with the 4400AM131 GENIE Bus Test Aid. Directions for use of the test are included on the related drawing of the same number; descriptions of instructions uaable on the Test Aid are included in Appendix A of this publication.

8. ASSEMBLY AND DISASSEMBLY

Assembly/disassembly of the GENIE Bus Controller is not applicable since the entire function is comprised of PWA's.

9. ADJUSTMENTS

Several pin-option adjustments are provided on the GBC PWA's. Refer to the ACPU1GBC-T theory publication for the GBC to determine their optional and normal settings.

10. TROUBLESHOOTING

This troubleshooting discussion covers the GENIE Bus Controller, GENIE Bus, Bus Extender/Switches and Automatic Program Interrupts. The discussion includes other system areas since the GENIE Bus interacts with virtually all aspects of a TDC 4500 process control system. In this discussion, the intent is to provide the user with insight into the GENIE Bus philosophy, including an overview of bus-related indications and their significances, and to offer troubleshooting approaches, references and test aids.

10.1 Error Indications

Indications of malfunctions within or related to the Bus can come from a variety of sources, such as:

- Software Messages (RTMOS Operating System or System Application Software)
- Alarm or Error Indicators (Programming and Maintenance Console, etc.)
- Observed Malfunctions (Incorrect outputs, lack of communication, etc.)
- Test Failures (Improper response to test programs)

10.1.1 Software Messages

The RTMOS Operating System provides indications to the user of malfunctions within the system, as detected by program status checks. Refer to the RTMOS Application Manual for Honeywell Process Computers, publication number PTS-038. Specific mention of RTMOS Error Detection and Alarming Messages is contained in Special Discussion 13 of that publication.

*Trademark

10.1.2 Alarm or Error Indicators

The extent and degree of alarm or error indications provided for the GENIE Bus varies according to system configuration. Fig. 10.1 provides an illustration of all bus-related error indications available through system options. The Programming and Maintennace Console is the CPU Alarm indicator and is a standard part of the CPU. Other indications of the GENIE Bus's performance can be obtained from the optional Bus Extender/Switches and the optional Device Emulator PWA.

10.1.3 Observed Malfunctions

In addition to the explicit indications of error messages and indicators, malfunctions can be implied from observation of system behavior. An incorrectly typed message (mispelled or incomplete), for example, or a lack of instrumentation response during periods when it should be operating, will signal a failure, although not necessarily GENIE Bus oriented. These non-explicit indications normally require a systems analyst or programmer to isolate the error source to a specific function or group of functions for troubleshooting. When this is not possible, it is sometimes best to build GENIE Bus confidence by running test programs until a malfunction is detected. (See "Troubleshooting Approaches".)

10.1.4 Test Failures

Malfunctions can be detected under test that do not show up under other circumstances because of the unique checks or conditions tested. The GENIE Bus Test provides testing for the basic GBC, the GENIE Bus, Bus Extender/Switches and the API's, virtually all GENIE Bus related areas. It does not check the individual devices and their controllers that connect to the bus, or their interaction, but uses the device emulator board to simulate a device controller under varying address, data and operations (GEN 2/TIM/TOM/DMA/Interrupts/Echo). Refer to the instructions within Honeywell PCD drawing 51103002, to determine specific tests used.

10.2 GENIE Bus Considerations

The GENIE Bus structure has some built-in factors that should be kept in mind while troubleshooting. Among these are:

- The bus has shared, common lines. A problem in a specific device controller can affect all operations. For example, a data line driver from a device controller that holds the line to ground will provide a "one" bit from all device controllers during bus operations.

- The GENIE Bus Controller time-shares operations on a priority basis. The GBC has a built-in priority network that governs its response to bus transfer requests from device controllers or the Processor. Refer to the GBC theory publication for service priority definitions.

  Requests from the Processor for GBC transfers are initiated by Processor-executed or interrupt-initiated GEN 2 instructions (IN/OUT/TIM/TOM, etc.). A request for an echo transfer notification to a device controller is also originated at the Processor when a DMT count reaches zero or TIM/TOM tables, etc. Device controllers request transfers between core memory and themselves (DMA operations) and request a transfer of interrupt information to the GBC during interrupts.

  A specific device controller, if malfunctioning, can hold down a DMA or interrupt request line and thereby create a continuous request situation. This would continue to be serviced by the GBC under the jurisdiction of the priority network. If no other requests were active, the GBC would provide continuous response, but if other, higher-priority requests occurred, these would interleave the response. Lower priority requests would be blocked, causing a Watchdog Alarm indication.

- Bus Extender/Switches and Device Controller Priorities. The physical placement of device controllers and Bus Extender/Switches on the bus, as well as their addresses, have no influence on their response priority. Priority of the extender/switches and device controllers is determined by priority switches on the PWA's themselves. (Refer to Theory publication ACPU1GBC-T.) Malfunctions in the priority network, such as an inadvertent setting up of two device controllers to the same priority or the shorting of a priority line would cause an OR'ing of response and the incorrect priority controller response, respectively, to the GBC acknowledge. With the OR'ing response to an acknowledge, one or more device controllers at the same priority would all respond, placing their respective DMA or interrupt-response information on the lines simultaneously. An incorrect controller response due to malfunctioning priority lines could permanently block response from certain controllers.

- Device Controller and Interrupt Response Addressing. Device controller addresses are determined by the setting of switches on the PWA's themselves. The Interrupt Response address for a device controller is a function of the selected address and information returned to the GBC in response to a GBC acknowledge to an interrupt request. (See Theory publication ACPU1GBC-T.)

TDC 4500 GENIE* BUS MAINTENANCE

1. INTRODUCTION

This publication provides maintenance-related information for the GENIE Bus Controller (GBC), including the Bus's Extender/Switches and the I/O interrupt capability for TDC 4500 process control systems.

2. OPTIONS

The GENIE Bus Controller is a functional component of the basic CPU. No options are associated with it except for related assemblies and functions, such as the chassis that house device controllers connecting to the GBC.

One I/O chassis is included as part of the CPU, model no. ACPU11. CPU model ACPU12 adds a second I/O chassis and a second power chassis. Models ACPU13 and ACPU14 add third and fourth I/O chassis.

3. REFERENCES

Test Program:

51103002 - GENIE Bus Test

Tables 1, 2, and 3 contain pin numbers for the GENIE Bus, CPU Bus, and console interface, respectively.

4. TEST EQUIPMENT

The general test equipment related to maintenance for the computer system is listed in Section 1 of this manual. In addition to this, the 4400AM131 GENIE GBC Test Aid is available for checkout of the GBC. The Test Aid is a PWA that plugs into a chassis of the GENIE Bus and can be exercised by program control (including the Test Program associated with the GBC) to verify the GBC's operation.

5. COMPONENT LOCATIONS

The GENIE Bus Controller occupies two card slots in the CPU chassis, slots 7 and 8.

6. PREVENTIVE MAINTENANCE

Preventive maintenance is not required for this function.

7. PERFORMANCE TESTS

Performance of the GENIE Bus Controller can be verified by use of the GENIE Bus Test 51103002, in conjunction with the 4400AM131 GENIE Bus Test Aid. Directions for use of the test are included on the related drawing of the same number; descriptions of instructions uaable on the Test Aid are included in Appendix A of this publication.

8. ASSEMBLY AND DISASSEMBLY

Assembly/disassembly of the GENIE Bus Controller is not applicable since the entire function is comprised of PWA's.

9. ADJUSTMENTS

Several pin-option adjustments are provided on the GBC PWA's. Refer to the ACPU1GBC-T theory publication for the GBC to determine their optional and normal settings.

10. TROUBLESHOOTING

This troubleshooting discussion covers the GENIE Bus Controller, GENIE Bus, Bus Extender/Switches and Automatic Program Interrupts. The discussion includes other system areas since the GENIE Bus interacts with virtually all aspects of a TDC 4500 process control system. In this discussion, the intent is to provide the user with insight into the GENIE Bus philosophy, including an overview of bus-related indications and their significances, and to offer troubleshooting approaches, references and test aids.

10.1 Error Indications

Indications of malfunctions within or related to the Bus can come from a variety of sources, such as:

- Software Messages (RTMOS Operating System or System Application Software)
- Alarm or Error Indicators (Programming and Maintenance Console, etc.)
- Observed Malfunctions (Incorrect outputs, lack of communication, etc.)
- Test Failures (Improper response to test programs)

10.1.1 Software Messages

The RTMOS Operating System provides indications to the user of malfunctions within the system, as detected by program status checks. Refer to the RTMOS Application Manual for Honeywell Process Computers, publication number PTS-038. Specific mention of RTMOS Error Detection and Alarming Messages is contained in Special Discussion 13 of that publication.

*Trademark

10.1.2 Alarm or Error Indicators

The extent and degree of alarm or error indications provided for the GENIE Bus varies according to system configuration. Fig. 10.1 provides an illustration of all bus-related error indications available through system options. The Programming and Maintennace Console is the CPU Alarm indicator and is a standard part of the CPU. Other indications of the GENIE Bus's performance can be obtained from the optional Bus Extender/Switches and the optional Device Emulator PWA.

10.1.3 Observed Malfunctions

In addition to the explicit indications of error messages and indicators, malfunctions can be implied from observation of system behavior. An incorrectly typed message (mispelled or incomplete), for example, or a lack of instrumentation response during periods when it should be operating, will signal a failure, although not necessarily GENIE Bus oriented. These non-explicit indications normally require a systems analyst or programmer to isolate the error source to a specific function or group of functions for troubleshooting. When this is not possible, it is sometimes best to build GENIE Bus confidence by running test programs until a malfunction is detected. (See "Troubleshooting Approaches".)

10.1.4 Test Failures

Malfunctions can be detected under test that do not show up under other circumstances because of the unique checks or conditions tested. The GENIE Bus Test provides testing for the basic GBC, the GENIE Bus, Bus Extender/Switches and the API's, virtually all GENIE Bus related areas. It does not check the individual devices and their controllers that connect to the bus, or their interaction, but uses the device emulator board to simulate a device controller under varying address, data and operations (GEN 2/TIM/TOM/DMA/Interrupts/Echo). Refer to the instructions within Honeywell PCD drawing 51103002, to determine specific tests used.

10.2 GENIE Bus Considerations

The GENIE Bus structure has some built-in factors that should be kept in mind while troubleshooting. Among these are:

- The bus has shared, common lines. A problem in a specific device controller can affect all operations. For example, a data line driver from a device controller that holds the line to ground will provide a "one" bit from all device controllers during bus operations.

- The GENIE Bus Controller time-shares operations on a priority basis. The GBC has a built-in priority network that governs its response to bus transfer requests from device controllers or the Processor. Refer to the GBC theory publication for service priority definitions.

  Requests from the Processor for GBC transfers are initiated by Processor-executed or interrupt-initiated GEN 2 instructions (IN/OUT/TIM/TOM, etc.). A request for an echo transfer notification to a device controller is also originated at the Processor when a DMT count reaches zero or TIM/TOM tables, etc. Device controllers request transfers between core memory and themselves (DMA operations) and request a transfer of interrupt information to the GBC during interrupts.

  A specific device controller, if malfunctioning, can hold down a DMA or interrupt request line and thereby create a continuous request situation. This would continue to be serviced by the GBC under the jurisdiction of the priority network. If no other requests were active, the GBC would provide continuous response, but if other, higher-priority requests occurred, these would interleave the response. Lower priority requests would be blocked, causing a Watchdog Alarm indication.

- Bus Extender/Switches and Device Controller Priorities. The physical placement of device controllers and Bus Extender/Switches on the bus, as well as their addresses, have no influence on their response priority. Priority of the extender/switches and device controllers is determined by priority switches on the PWA's themselves. (Refer to Theory publication ACPU1GBC-T.) Malfunctions in the priority network, such as an inadvertent setting up of two device controllers to the same priority or the shorting of a priority line would cause an OR'ing of response and the incorrect priority controller response, respectively, to the GBC acknowledge. With the OR'ing response to an acknowledge, one or more device controllers at the same priority would all respond, placing their respective DMA or interrupt-response information on the lines simultaneously. An incorrect controller response due to malfunctioning priority lines could permanently block response from certain controllers.

- Device Controller and Interrupt Response Addressing. Device controller addresses are determined by the setting of switches on the PWA's themselves. The Interrupt Response address for a device controller is a function of the selected address and information returned to the GBC in response to a GBC acknowledge to an interrupt request. (See Theory publication ACPU1GBC-T.)

	REQUESTS			ACKNOWLEDGES			SYNCS				
SIGNAL →	PR1	PR2	MRQ	IAK	ACK	MAK	RSN	MSN	GSN	SSN	Handshaking Signals
PIN NOS. →	A33	A83	A30	A82	A32	A81	A84	A80	A34	A79	
PURPOSE →	NON-INHIB REQ	NON-INHIB REQ	MEMORY REQ	PR2 ACK	PR1 ACK	MRQ ACK	RETURN SYNC	MASTER SYNC	GRANT SYNC	SLAVE SYNC	

	TRANSFER TYPE CODING										
SIGNAL →	MTR	SST	S05	S04	S03	S02	S01	SP4	SP2	SP1	Transfer Type
PIN NOS. →	A07	A10	A59	A09	A58	A08	A57	A60	A61	A11	
PURPOSE →	MEMORY TRANS	SENSE STAT	IN	OUT	OPR	ABT	ACT	SUB-CODING			

	DEVICE ADDRESS									
SIGNAL →	IDP	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	Interrupt ID/Device Controller Select
PIN NOS. →	A25	A75	A26	A76	A27	A77	A28	A78	A29	
PURPOSE →	PARITY	DEVICE CONTROLLER ID/SELECT								

	PARITY	GENIE BUS DATA											
SIGNAL →	BP	B23	22	21	20	19	18	17	16	15	14	13	B12
PIN NOS. →	A12	A63	A13	A64	A14	A65	A15	A66	A16	A67	A17	A68	A18

	GENIE BUS DATA											
SIGNAL →	B11	10	09	08	07	06	05	04	03	02	01	B00
PIN NOS. →	A69	A19	A70	A20	A71	A21	A72	A22	A73	A23	A74	A24

Bus Data

	STATUS			ALARMS					CLEARS		
SIGNAL →	EFL	BCA		ALM	CPU ALM	CPU ALM 1	CPU ALM 2	VMA	RSA	CLR	Status/Alarm/Clears
PIN NOS. →	A31	A62		A05	A89	B90	B40	A56	A55	A06	
	ECHO	BC	A	DEVICE ALM	CPU ALARM	BUS EXT. SW. ALM		VOLT ALM	RESET ALARMS	BUS CLEAR	

Parity Check Inhibit — GBC Parity Error

Table 3.1 GENIE Bus Signals

SIGNAL	HANDSHAKE CONTROL										INTERRUPT CONTROL			
	GREQ	GGNT	PGNT	ISNR	RID2	RID1	RID0	R/W	ISYN	ASYN	PAI	IAI2	INT	CONA
PIN NOS.	A34	A97	A96	A35	A08	A58	A09	A07	A05	A06	B88	B87	B15	B67
PURPOSE	INITIAL ACCESS REQ & GRANT			SYNC REQ	IDENTIFY RESPONDER			TRANSFER DIRECT	MBC SYNC	RESPONSE	PERMIT	INHIB LEVEL 2	GBC INT.	CONSOLE ATNT.

SIGNAL	ADDRESS LINES SA17-00																	
	SA17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	SA00
PIN NOS.	A10	A60	A11	A61	A12	A62	A13	A63	A14	A64	A15	A65	A16	A66	A17	A67	A18	A68
GEN 2 USE OF SIGNALS				S			DEVICE ADDRESS			DEVICE ADDRESS			S'			DEVICE ADDRESS		

SIGNAL	DATA LINES 23-12											
	DB23	22	21	20	19	18	17	16	15	14	13	12
PIN NOS.	A22	A72	A23	A73	A24	A74	A25	A75	A26	A76	A27	A77

SIGNAL	DATA LINES 11-00											
	DB11	10	09	08	07	06	05	04	03	02	01	DB00
PIN NOS.	A28	A78	A29	A79	A30	A80	A31	A81	A32	A82	A33	A83

SIGNAL	ALARMS/CLEARS								CLOCK TIMER	
	TRAP	RED & BLUE	PF	VMG	WTO	SCLR	CLRALM	WRWT	10VACH	10VACL
PIN NOS.	A86	B21	B32	B82	B66	A59	A38	B71	B34	B33
PURPOSE	MBC ALARMS		POWER FAIL		WATCHDOG TIMEOUT	SYSTEM CLEAR	ALMCL	WATCH. RESET	LINE FREQUENCY	

Table 3.2 CPU Bus Interface Signals

	CONTROL/STATUS								
SIGNAL →	ALARM	ALM RST	RESET	STLLO	APILO	RODMDF	ROPMCE	R1KRC7	CONA
PIN NOS. →	B97	B47	B83	B38	B37	B85	B31	B40	B86

	REG. SELECTION			KEYPAD SELECTION			SERVICE MODE		
SIGNAL →	RIDML1	RIDML2	RIDML4	RIKOM1	RIKOM2	RIKOM4	RISMC1	RISMC2	RISMC4
PIN NOS. →	B42	B92	B41	B43	B35	B30	B91	B89	B90

	DISPLAY DATA CODE					
SIGNAL →	CPU STB	CPU0	CPU1	CPU2	CPU3	CPU4
PIN NOS. →	B94	B46	B96	B45	B95	B44

Table 3.3 Console Interface Signals

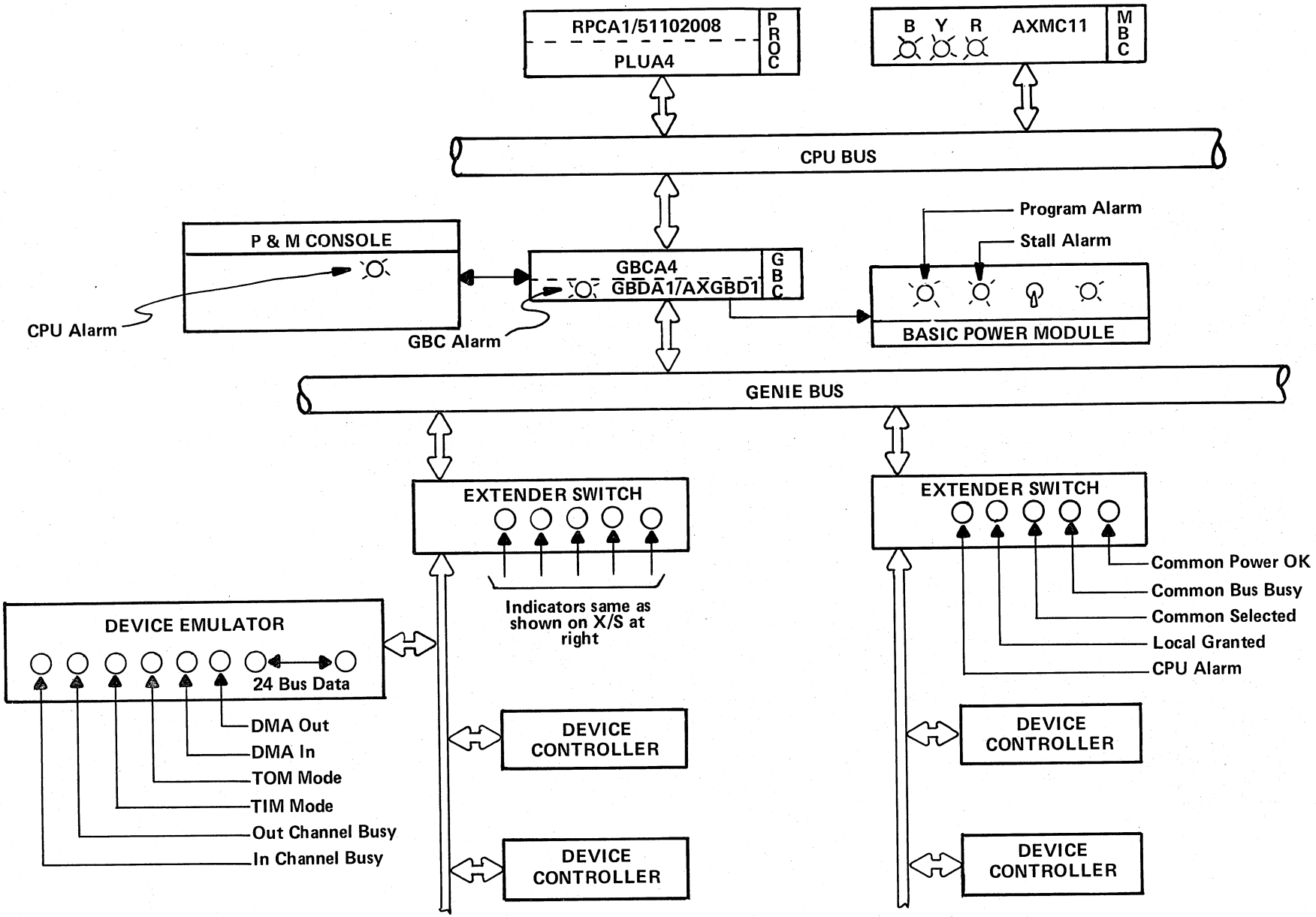


Fig. 10.1 GENIE Bus-Related Checks and Indications

10.3 Timeout Alarms

The GBC performs Timeout checks on the following GENIE Bus operating conditions:

1. During any bus transfer initiated by a GEN 2 instruction;
2. During an Interrupt Request ID cycle;
3. During a DMA ID cycle.

Special Timeout Alarm Notes

1. Timeout alarm indications, which are indicated by the Programming and Maintenance Console's "CPU Alarm" and the GBC's alarm indicator, and detectable by JNE test to the GBC, remain until cleared manually or by program control. (See Alarm Clearing.)
2. RTMOS clears Timeout alarms as they occur and records them by count for application software usage.
3. A common timeout alarm condition occurs from incorrect switch selections at the device controllers. Each device controller has selectable addresses and priorities. The switches have an accompanying parity bit to maintain odd parity. If incorrectly set (i.e., even parity), a timeout will occur during access.

10.3.1 Timeout Alarm Isolation

The particular approach used in isolating Timeout alarms will vary according to the conditions involved, such as the status of operation (on-line vs. off-line), the specific bus structure involved (master only vs. slave and/or switched bus segments), and the availability of reliable replacement PWAs. The following guidelines are offered for use in isolation of Timeout alarms.

If possible, determine which type of the three Timeout alarm checks yielded the alarm. This is generally identifiable if the test program is used or if a hand loop is being run. If Timeout alarms are experienced while running the operating system, any indication will be by application software message, which may or may not identify specifics. (See following Operating System discussion.)

10.3.2 Operating System Timeout Alarms

The operating system (RTMOS) can detect Timeout alarms and distinguish them from other GENIE Bus alarms detected by the device controllers; however, the Timeout and device controller alarms share a common indicator on the GBDA1 PWA (See following discussion of GENIE Bus device controller alarms.)

RTMOS does not provide overt indications, but does provide the GENIE Bus alarm information for use by application software. It also clears the alarm and there will be no detectable indicator lighting while the system is running. Indications of such an alarm must therefore come either from the application software itself or from someone able to interpret the actions of the software.

When there are no definite symptoms to indicate the source of the alarm, it may be necessary to resort to a T&D or a hand loop to isolate the alarm source. For GEN 2 Bus exchanges, the primary Timeout alarm failure suspects are the Master and Slave Sync signals.

If Bus/Extender Switches are used, the lighting of its Busy indicator notes that it is receiving a returned Slave Sync signal from a device controller, which implies that both the Master and Slave Sync signals for that bus segment are working.

An entire bus segment will yield Timeout alarms if it is not connected or if its power is off. Check the status of the associated alarm indicators on the X/S's if a bus segment is out. Also check for the Common Selected indicator, which indicates whether that bus segment is connected and selected.

Interrupt and DMA request cycles use more bus communication signals than the GEN 2 Bus exchanges. They additionally require that the Acknowledge signal be returned from a terminator board on the bus as RSN. (An incorrectly-pinned or missing terminator could provide Timeout alarms on the Interrupt and DMA requests, since the RSN signal would not return.) If only one bus segment is bad, check its far-end terminator. If all bad, check the Master Bus far-end terminator.

10.3.3 Clearing Timeout Alarms

A Timeout alarm indication remains until cleared by one of three methods:

1. Pressing the Alarm Reset button on the Programming and Maintenance console;
2. Pressing the Initialize function (Reset/0) on the Programming and Maintenance console;
3. Execution of a JNE or ABT command (ABT4000-GBC; ABT4070-GBC and all device controllers).

Although the alarm light remains after detecting an error until cleared manually or by program control, the GENIE Bus is released by the GBC's generation of a simulated Slave Sync signal at the end of the timeout period. The simulated Slave Sync is required to terminate the current busy function that did not complete and consequently produced the timeout alarm.

10.4 GBC/GENIE Bus-Related Indicators

Observable indicators related to GBC and GENIE Bus actions are located at several points in the CPU chassis, and optionally at the I/O chassis. Fig. 10.1 is a block diagram indicating the possible indicators and their functional locations. They are described in the following text in two groups: those that are standard and relate primarily to the GBC and other CPU functions (10.4.1) and those that are optional and relate primarily to the GENIE Bus (10.4.2).

10.4.1 GBC-Related Indicators

Fig. 10.1 shows both GBC and GENIE Bus related indicators. Those that relate primarily to the GBC and other CPU functions are located at the top side of the GENIE Bus. The physical location of these indicators is shown in Fig. 3 of the Processor maintenance publication, ACPU1PROC-M.

10.4.1.1 CPU ALARM

This indicator, labeled Alarm on the Programming and Maintenance console, is actually an ORing of all other CPU alarms. Some of these are cleared by Alarm Reset, while others require System Clear, as defined by their individual descriptions that follow.

10.4.1.2 GBC ALARM

The GBC alarm is an LED indicator located on the GBC's GBCA1 PWA. It lights whenever a GBC timeout condition or optionally connected device alarm has occurred. It resets following a GBC status check, alarm Reset, System Clear or GENIE Bus Clear. (See 10.4.1.7.)

The GBC timeout occurs whenever a GENIE Bus transaction that was initiated does not complete normally (by return of a Slave Sync signal from the affected device controller). This alarm may result following any Master Sync or Grant Sync transmission to the GENIE Bus. Master Sync is transmitted as a result of a Processor-initiated Input or Output GEN 2 command that affects a device controller. (Input includes all S=5 through S=7 coded GEN 2 commands; output includes all S=1 through S=4 coded GEN 2 commands.) Grant Sync is generated during the identification cycle of either an I/O interrupt request or a DMA request. (Refer to Timeout Alarms description, 10.3.)

Device controller alarms are optionally OR connected to the GENIE Bus from the device controllers. Refer to the specific device controller publications to determine the significance of the alarms. These alarms are typically reset by the assertion of the GENIE Bus Clear signal from the GBC. (See 10.4.1.7.)

10.4.1.3 STALL ALARM

The Stall alarm, when enabled, produces an alarm at a pin-option interval if not reset by a Reset Stall Time (RST) macro-instruction prior to the selected timeout interval. The Stall Alarm is disabled by the Stall Lockout switch at the Programming and Maintenance Console. It is reset by System Clear. Its LED is contained on the System Power Module.

10.4.1.4 PROGRAMMABLE ALARM

The Programmable Alarm is set by the SALM macro-instruction and reset by System Clear or by the RALM macro-instruction. Its indicator is also located on the System Power Module.

10.4.1.5 RED/BLUE MBC ALARMS

These alarms are indicated by LED's at the MBC PWA in the CPU chassis. The BLUE error indicates a timeout condition at the CPU Bus, implying a control failure in the bus handshaking exchange. The RED error indicates an error on data recovery from memory and therefore implies a failure on the data from memory itself. These alarms are related to the GBC since some device controllers require MBC and memory access for DMA transactions. Refer to ACPU1MBC-M Memory maintenance publication for specifics on their error detections and clearing conditions.

10.4.1.6 NON-INDICATING ALARMS

There are two additional GBC-related alarm conditions that are detectable, but not indicated by an LED. These are interrupt alarm checks that cause a Program relocation. One of these is the Permit Timer alarm, which causes a non-inhibitable I/O interrupt response. The other is the Watchdog Timer that causes an Internal Interrupt Trap response. The standard software provides response to these alarms, as defined under Special Discussion 13 of the RTMOS Application Manual, PTS-038.

The Permit Timer alarm causes a relocation to memory address 201_8. It is detected if an interrupt is left unserviced or if the PAI flip-flop is left reset longer than the timeout period selected. The timeout is 16 system clocks, which are generated from the ac input and can be optionally set to clock on every full cycle or half cycle. The Permit Timer Alarm is reset by a System Clear. (See 10.4.1.7.)

The Watchdog Timer Alarm causes an internal interrupt trap to memory location 24_8. It is detected if an interrupt is left unserviced or if the IAI_2 flip-flop is left reset longer than the timeout period of the Timer. The Timer, triggered by PGNT's (MBC grants to the Processor for CPU Bus access), can be pin set to alarm after 512 or 2048_{10} PGNT's. The Watchdog Timer is reset by a System clear and can be disabled by the Stall Lockout or API Lockout switches at the Programming and Maintenance Console.

10.4.1.7 GBC-RELATED ALARM CLEARING

There are three basic alarm clearing signals generated to reset alarms: System Clear, GENIE Bus Clear, and Alarm Reset. The previous alarms described can be cleared by one or more of these signals, as defined. These alarm lines are generated by the following conditions.

- System Clear (GBCA4 PWA, pin A59)

  System Clear is generated during a power up condition, or as a result of a Stall alarm or Reset function. The Reset function is the simultaneous depression of the Reset button and a keypad number at the Programming and Maintenance Console. It is also possible to program a System Clear using a Control Bus macro-instruction addressed to the GBC. See the Appendix of the 4500 General Description.

- GENIE Bus Clear (GBCA4 PWA, pin C01)

  The GENIE Bus Clear signal is asserted if System Clear is generated or if an ABT 4070 macro-instruction is executed.

- Reset Alarms (GBCA4 PWA, pins B47 and A38)

  The Reset Alarm signal is asserted if the Alarm Reset button is pressed at the Programming and Maintenance Console. This signal is passed to the device controller through the GBC from pin A55 on the GENIE Bus. The Reset Alarm line to the device controllers is also asserted if an ABT 4000 (GBC Clr.) macro-instruction is executed.

10.4.2 Extender/Switch Indicators

The Extender/Switch indicators are LED's, visible from the front side of the Extender/Switch PWA, that glow red when enabled. The X/S indicators and their enabling conditions are as follows: (Refer to Fig. 10.1)

- Common Power OK

  This indicator extinguishes if the +5V, +15V or -15V is lost from the supply powering the associated bus segment.

- Common Bus Busy

  This indicator lights and remains lit for 1/2 second following each Slave Sync sent to the X/S by a device controller on the Slave Bus (in response to MSN or GSN). If constantly lit, the indicator indicates a frequency of bus traffic of two transfers per second or greater. A blinking light indicates transfers of less frequency. An unlit light indicates no Slave Bus activity.

- Common Selected

  This indicator lights when the bus segment connected by the X/S is selected. If the X/S is pinned for "Extender only", selection is manual, as described by publication ACPU1GBC-T; if the X/S is pinned for X/S, selection can be either manual or by program. (Refer to Slave Bus Selection description in the theory writeup.)

- Local Granted

  This indicator lights when the X/S is in the Local mode of operation. Local mode is established either by switch selection at the X/S or by program selection (if the X/S is pinned for "switching" as well as "Extender" operation). In the Local mode, the X/S is unresponsive to program control. The X/S must be manually set to the "Remote" mode and program set via an OPR command before changing to the "Remote" mode. ("Remote" mode cannot be established if the X/S is pinned for "Extender Only" operation.)

- CPU Alarm

  The CPU alarm indication is available for system connections. When connected, detection of an alarm condition places the X/S in the Local mode of operation. If the X/S was in the "Remote" mode of operation when the alarm occurs, it automatically deselects the bus, cutting off communication.

  Normally the CPU alarm, when connected, indicates a memory parity error or a stall alarm, but can be connected for other conditions. See the specific System Logic for the system involved to determine if this alarm line is connected and to determine what alarm conditions are used.

10.4.3 Off-Line Timeout Alarms on GEN 2 Bus Exchanges

Fig. 10.2 provides a flow chart for isolating Timeout alarms detected from GEN 2-initiated bus exchanges. The following notes pertain to reference keys located on the flow chart.

⚠1 Attempt to determine if the Timeout alarm is common to the entire bus, to a specific bus segment, or just to one specific device controller. If it seems apparent that the entire bus, which includes slave segments, and does not consist of just a master bus, or if only one of the slave bus segments is affected, proceed to step 2; if not, proceed to step 6.

2 Note whether the POWER OK indicator on the Extender/Switch for the affected slave bus segment(s) is lit. Normally, the entire bus is on one power supply and all X/S POWER OK lights would be out in the event of a power supply failure. If more than one bus segment shares a common power supply, but one of the X/S's POWER OK indicator is out, either the X/S itself is failing or there is discontinuity from the power supply to that bus segment.

The POWER OK indicator is located on the Extender/Switch PWB assigned to a slave bus segment and lights if the +5V, +15V or -15V for a bus segment is lost.

3 Common Selected

Communication between the GBC and a slave bus section will not take place unless the X/S for the Slave Bus is selected. Selection can be accomplished by off-line, forced manual selection, by on-line manual selection, or by on-line automatic operation (program select). If the X/S is pinned for "Extender-Only" operation, selection is limited to on-line or off-line manual selection. (Refer to "Slave Bus Selection" in the theory publication for information on selection.)

If the bus is not selected, as noted by the indicator, attempt selection. The bus could

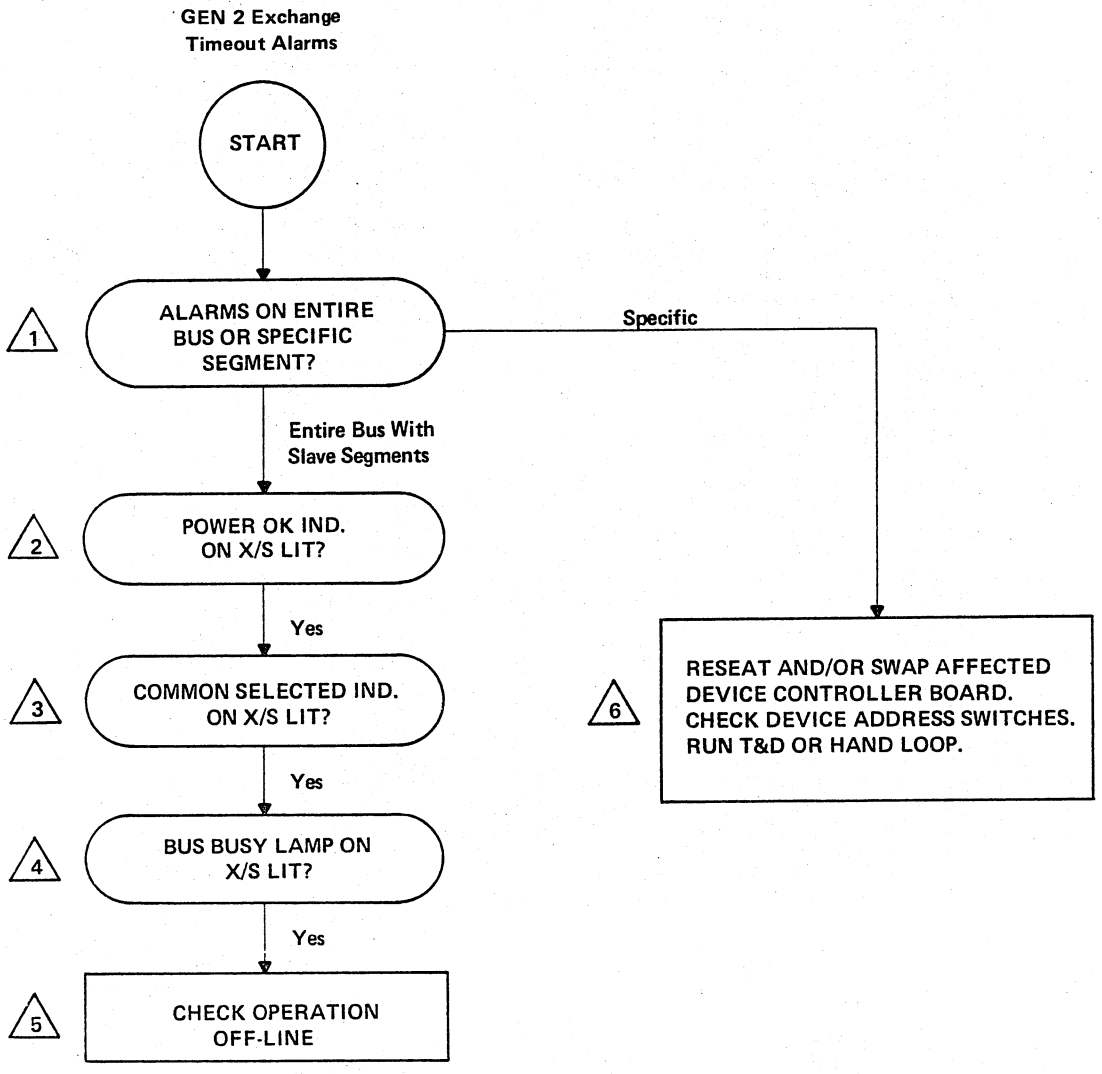


Fig. 10.2 Timeout Alarm Isolation

have been de-selected manually, by a program-executed ABT (if pinned for X/S), or by a power failure, CPU Voltage Monitor Alarm, or optional CPU alarm.

4 Bus Busy

Check whether the Bus Busy lamp is lighting at the affected Slave Bus X/S's. If not, the bus segment is not generating a Slave Sync response. This could be due to several causes. In order to generate a Slave Sync response, the bus segment must receive the Master Sync signal and a device controller on the bus segment must receive its device address. The fact that a Timeout error occurs on a GEN 2 operation indicates that the GBC issued a Master Sync; however, it is possible that the MSN signal is not being sent to the bus (connection problem, etc.) or that the device address is either incorrect or the device controller is not selecting or generating an SSN response. The former is more likely, if the problem is widespread to the entire bus, as implied by this branch of the flow chart; the latter (device controller not responding) is more likely if a specific section is affected.

Use an exerciser (see 10.6, Troubleshooting Aids) to check MSN generation, SSN response.

5 If all of the preceding was OK, it is necessary to go off-line to isolate the source of Timeout alarms. When off-line, swap or replace related PWBs; if necessary, put the Device Emulator in the bus and run a simple transfer loop to check the MSN out and Slave Sync response.

If it is not apparent which function produced the Timeout alarm, and if the preceding checks and loops did not indicate the error source, it is possible that the alarms were occurring from an interrupt request cycle or a DMA request cycle.

Use an exerciser (see 10.6, Troubleshooting Aids) to initiate these requests and check the bus transactions of the ACK, RSN, GSN and SSN signals.

6 This point in the flow chart is reached if it appeared that the Timeout alarm was coming from a single device controller. This could be due to the fact that the device controller was not responding to a bus transaction initiated by the GBC, or it could be the only one initiating a specific action, such as an interrupt or DMA request. Try re-seating the device controller board to eliminate a contact problem. If this does not correct the failure, and if sufficient spares are available, swap the board. If there are no good spares, find the function that produces the alarm, put in a short test loop and troubleshoot.

Use the Device Emulator in its place to see if it is the device controller or the bus leading to it that is producing the error.

10.4.4 Other GENIE Bus Alarms

In addition to the Timeout alarms, the GENIE Bus has parity generate/check circuitry and the individual device controllers have error checking circuits. The GENIE Bus Alarm lamp is designed to indicate any errors flagged by the individual device controllers, if they are designed to provide the alarm and are connected to the alarm line. Any parity errors detected on bus transfers are passed to the device controller for sensing and clearing. The exception is a non-compare of a device address's parity with that assigned by switches at the device controller. In the event of non-comparison, the device controller does not select and the GBC produces a Timeout alarm.

In normal operation, the GENIE Bus alarm is cleared by the operating system and a record made of the error, which then becomes the responsibility of the application software for alarm notification to the user. In the off-line mode, the user is usually aware of which device was being accessed prior to the alarm and would therefore know which device controller produced the alarm. If not, and the alarm condition was not cleared, the user could test each device controller with a JNE until the light extinguished or he could isolate bus segments to narrow it down that far before testing each controller.

PARITY CHECK/GENERATE NOTES

1. The IOC does not generate or check parity on information transfers between itself and the memory multiplexer or the AU.

2. On byte transfers using a TOM, the leading field (remainder of word other than byte) is masked by the GBC and is not a factor in the parity generated by the GBC. For OUT transfers to byte-oriented devices, however, parity is generated by the GBC using the entire 24-bit word from the AU. This may or may not be correct parity for the byte, depending upon the bit configuration of the leading field. (Most byte-oriented device controllers are designed to ignore this, by overlooking the parity on such transfers.)

3. Not all device controllers generate or check parity on transfers between themselves and the GBC. They can ignore parity sent by the GBC on output operations and can flag the GBC to ignore parity on inputs by asserting the Bit Check Disable line (BCD) to the GBC. (This line is dual-purpose; it is enabled by the GBC to indicate an error detected by the GBC on information received from the device controller or memory and it can be enabled by the device controller to inform the GBC to inhibit parity checking on the input data.)

10.5 Bus Isolation

Since the GENIE Bus is a shared set of communication lines, the effect of a single problem can affect the entire bus under certain circumstances. For example, it is possible for a device controller's communication lines (line drivers or line receivers) to short to ground under all, specific, or intermittent circumstances. Any line so tied to ground would pull down that line for the entire bus. This is not a frequent occurrence, but could result from failed driver/receiver circuits, or from exposed line shorts anywhere along the bus. This discussion on bus isolation provides assistance in isolating bus sections for the purpose of locating such shorts.

Fig. 10.3 illustrates a GENIE Bus example for the purpose of explaining each of the possible disconnect possibilities. This example is by no means typical, but does provide a reference for explaining the general rules that can be followed for isolating any GENIE Bus. At first appearance, the example may seem complex, especially if the reader is unfamiliar with the GENIE Bus structure. If this is the case, refer to the GENIE Bus Structure discussion within the theory writeup to see how such a bus goes together.

Fig. 10.3 includes reference notes that explain the symbols and isolation possibilities. The following rules should also be kept in mind when isolating the bus.

10.5.1 Bus Isolation Rules

- Any time the Master Bus or a Slave Bus segment is disconnected, the bus section's far-end terminator is automatically removed by the disconnection. The terminator should then be moved in to the furthest point on the remaining bus section so that the terminator's pull-up resistors and the RSN signal are still available to the bus. (On a short bus, exchanges will still work with only the GBC's near-end terminators, although the far-end terminators are required for DMA and Interrupt request cycles.)

- Removing a device controller PWA by unplugging it from the bus also disconnects it from the associated priority bus. Any effect from the device controller on the common GENIE Bus or associated priority bus would therefore be removed. Disconnecting an X/S isolates an entire Slave Bus segment.

- X/S's can be de-selected by a switch at the PWB to isolate its bus segment from the Master Bus. De-selecting the X/S does not, however, electrically isolate its line driver/receivers from transferring a short condition to the bus.

- Power should be removed before unplugging or re-inserting PWBs.

10.6 Troubleshooting Aids

The following test aids are included to provide assistance where the test program cannot be loaded. These aids are test loops that exercise the GENIE Bus using the Device Emulator in the basic operations of GEN 2 transfers, DMA transfers, and interrupt generation. These tests assume Device Emulator PWB inserted on GENIE Bus. Each of the loops included in the following text contains the instruction sequence and related instructions for executing it. Following these loops are additional aids for getting the test program loaded and operating.

10.6.1 GEN 2 Transfers - OUT

1. Load the following "OUT" data program:

```
Location 100   25050000   RCS
         101   25047707   OUT, 7707 (bus add /377)
         102   26200000   NOP
         103   14000100   BRU, 100
```

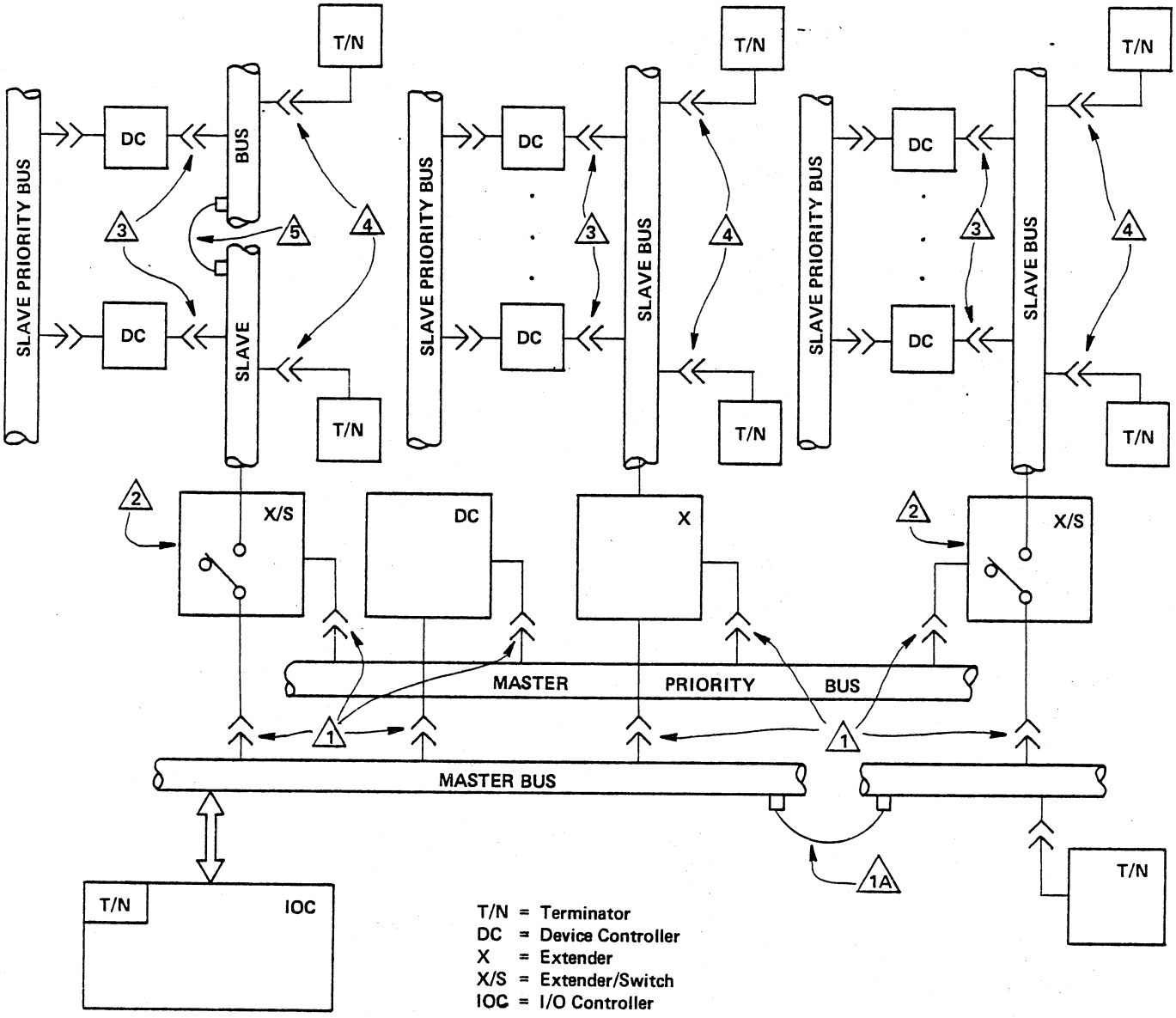
2. Enter a 25252525 data pattern in the console switches. Branch to Location 100 and STEP through the program. On the OUT command, verify the following:

   a. No CPU Alarm

   b. Test PWA data lamps have assumed the test pattern in the console switches.

> NOTE
>
> See Appendix for a description and control of the test PWA GTBA1.

3. Place computer in AUTO. Enter various test patterns in the console switches ranging from 00000000 to 77777777.



NOTES

1. Device controllers on the Master Bus can be disconnected by unplugging them. See following note.

1A. If the Master Bus is configured on more than one chassis, the bus is connected by "waterfall" ribbon cable connections. If connected in this manner, the bus can be segmented by removing the ribbon cable. (See discussion on bus isolation to note the effect of loss of far-end termination when bus is segmented.)

2. The X/S and associated Slave Bus can be isolated by manual switching at the X/S, or automatically in the event of program deselect, power failure or the optional CPU Alarm connection. (See descriptions on Bus Selection and on Bus Isolation.)

3. Device controllers on slave busses can be physically disconnected by unplugging them, in the same manner as those connected on the Master Bus.

4. Slave Bus terminators can be disconnected from the bus by disconnecting them, but they should be replaced to provide the bus with pull-up resistors and RSN generation.

Fig. 10.3 GENIE Bus Isolation Example

4. Check the test board data lamps as the data is being changed.

> NOTE
>
> For troubleshooting, replace the NOP at Location 102 with 25034070 (ABT).

> NOTE
>
> The Device Emulator's "I/O Bus" lamp should be lit brightly indicating continuous bus traffic.

10.6.2 GEN 2 Transfers - IN

1. Assuming "Data OUT" movement is OK, load the following short loop program to exercise the IN command:

```
Location 100   25050000   RCS
         101   32000111   STA  111 (save it)
         102   25047707   OUT  7707 (TX)
         103   05000000   LDZ
         104   25057706   IN 7706 (RX)
         105   10000111   ERA 111 (compare)
         106   05004670   TZE (test)
         107   34000100   BTS 100 (OK)
         110   14040000   BRU* comp. error
```

2. Enter a /25252525 pattern in the console switches. Branch to Location /100 and step through the program.

3. On execution of the IN 7706 command, verify the following:

   a. No CPU Alarm

   b. "A" Register contains console switch pattern.

4. If an error has been detected, a short IN 7706, BRU*-1 loop will be necessary for troubleshooting. First, set up the data pattern on the test PWB by doing an OUT 7707, then go to the short loop.

5. If the CPU Alarm lamp remains lit following the IN (or OUT) loop, it will be necessary to include an ABT 4070 in the loop. This will provide a good sync point on the bus.

6. With the loop program running, change the console switches to various data patterns ranging from /00000000 to /77777777. There should be no error halts.

7. The following loop can be used for a scope check of the IN command:

```
Location 100   25057706   IN 7706
         101   14077777   BRU*-1
```

10.6.3 Direct Memory Access (DMA) Full-Word Read

1. With all other drive boards removed from the GENIE Bus, install the Device Emulator in the first or dedicated bus.

2. Load the following test program:

```
Location 100   25050000   RCS
         101   32000777   STA 777
         102   05010000   CPL
         103   32001000   STA 1000
         104   25034070   ABT
         105   00000111   LDA 111
         106   25027727   OPR TX, S' = 2
         107   25057706   IN RX
         110   14040000   HALT
         111   00000777   CON (Packing &
                               Mem. Add.)
```

3. Put an alternate data pattern (e.g. 70707070) in the console switches.

4. Branch to Location 100 and step through the program until the HALT command. (The A-register should contain the value of the console switches.)

5. Examine the light bank of the test board which should contain the complement of the console switch values.

   The test patterns were previously set up by the program at /777 and at /1000. The transmitter was then OPERATED to put it in the DMA (output) mode with the "A" register giving it its packing mode (full word) and the starting address of /777. This triggered a DMA request for the contents of /777 and transferred the whole word out to the test board. At the completion of this DMA cycle, the test board's memory address register was incremented to /1000.

   The IN command read the data pattern in (contents of 777) and at the same time triggered off another DMA output cycle, this time for Location 1000.

6. Change the BRU* at Location 110 to a BRU 100 for cycling on the program if it is necessary for troubleshooting.

10.6.4 DMA Byte Read

1. Using the same program as that used for DMA Full-Word Read, change Location 111 to 01000777. This will specify the most significant byte for transfer.

2. Change the value of /111 in the test program to 01000777. (19, 18 saying M. S. byte.)

3. Enter test pattern 00125377 into the console switches.

   This makes the most significant 8 bits all zeros, the middle 8 bits alternate one's and the least significant 8 bits all one's.

4. Branch to Location 100 and step through the program. At the halt, the "A" register should contain all zero's and the test board's data register 00000252.

   Since the test board is now acting like an 8-bit oriented device, the GBC will put the 8 data bits on data lines 0-7 regardless of what byte position it came out from memory.

   The "A" register is holding the most significant 8 bits of Location 777 in its ("A" register) 0-7 positions, and the test board is holding the middle 8 bits of Location 777 (alternate one's), due to the normal byte increment and the second DMA cycle, triggered by the IN command.

5. Without initializing, enter another IN 7706 in the next Instruction register and depress step. This will initiate a third DMA output cycle directed toward the least significant 8 bits of Location 777.

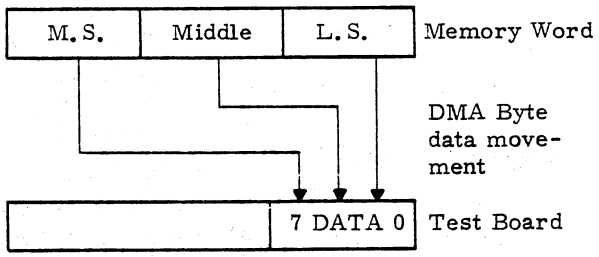
   The "A" register should contain 00000252 and the test board 00000377.

   Since the last DMA operation was directed toward the least significant byte of the current memory address (777), the memory address of the test board is incremented by one (1000) and its byte packing "address" is reset to point to the most significant byte again.

   Subsequent IN commands would continue to transfer the M.S., middle and L.S. bytes of Location 1000 (complemented pattern of the console switches) in the same manner just described.

6. Although the program allows only two DMA cycles as it is written, you may change the Starting Byte Only by altering the values of bits 18 and 19 of Location 111 as follows:

19	18	
0	1	Most significant byte
1	0	Middle byte
1	1	Least significant byte
0	0	Full word (no byte transfer)



10.6.5 DMA Write (Full Word)

1. Load the following test program:

```
Location 100  25034070  ABT Clear
         101  00000107  LDA, 107
         102  25027726  OPR RX, S' = 2
         103  25050000  RCS
         104  25047707  OUT TX Trigger DMA
         105  26200000  NOP
         106  14040000  BRU*
         107  00001000  CON (DMA Packing,
                          Mem. Add.)
```

2. Put an alternate "ONE" pattern in the console switches.

   The OUT command will transfer this data pattern out to the test board's data register and also trigger the DMA input function to transfer this data into memory at the "starting" address of /1000.

3. Step through the program until the BRU* hang-up Examine the contents of core Location 1000 for the data pattern entered into the console switches (alternate one's).

10.6.6 DMA Write (Byte Mode)

1. Change Location 107 in the preceding program (10.6.5.1) to 01001000. (This acts as the starting byte transfer to the most significant 8 bits.)

2. Store a minus one value in Location 1000.

3. Perform the following step to observe the results of byte transfers:

   Most Significant Byte

   a) Clear the console switches and step through the next Instruction register.

   b) At the halt, examine the contents of Location 1000. It should be 00177777.

Middle Byte

c) Enter a second OUT command to /7707 in the Next Instruction register

d) Set the "A" register to 00000252.

e) Depress Execute. (Do not initialize.)

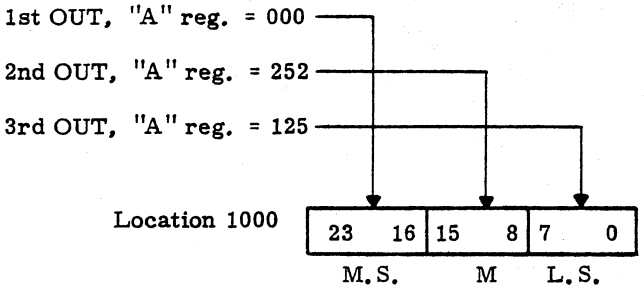
f) Examine Location 1000. (It should be 00125377.)

Least Significant Byte

g) Enter a third OUT command in the Next Instruction register, with the "A" register containing 00000125.

h) Depress Execute and examine Location 1000. (It should contain 00125125.)

4. In each of the byte transfer operations in step 3, only the affected byte position of the memory location should have been altered. The sequence of events for the data movements would have appeared as indicated by the following figure:



5. For a troubleshooting program loop, change the BRU* at /106 to a BRU /100.

In Location 107, set bits 18 and 19 to the byte under question as follows:

19	18	
0	1	Most significant byte
1	0	Middle byte
1	1	Least significant byte
0	0	24 bit transfer, (full word)

For byte operations, the data pattern is entered in "A" register bits 0-7 for the OUT command, regardless of what byte is under test.

10.6.7 Non-Inhibitable APIs (Non-TIM/TOM)

1. Load the following short loop:

```
Location 100  25034070  ABT
         101  26200000  NOP
         102  25017706  ACT S' = 0
         103  26200000  NOP
         104  26200000  NOP
         105  14000100  BRU /101
```

2. Store into the following interrupt vector locations:

```
Location 1174 = 14001174  RX, Type 0
         1175 = 14001175  RX, Type 1
         1176 = 14001176  TX, Type 0
         1177 = 14001177  TX, Type 1
```

The above interrupt vectors were calculated from the test board's receiver and transmitter addresses of 776 (RX) and 777 (TX):

```
RX = 2 x 776 + 200 = 1174  Type 0
                     +1
                    1175  Type 1

TX = 2 x 777 + 200 = 1176  Type 0
                     +1
                    1177  Type 1
```

3. Start the above program and lower the API lockout switch. Computer should halt at Location 1174 (BRU /1174), the RX's Type 0 vector.

4. Change Location 102 of test program to 25017716 (S' = 1).

5. Restart the test program and lower API lockout switch. This time, the computer should halt at Location 1175. RX's Type 1 vector.

6. Repeat the above procedure, except use the transmitter's address for the ACT command:

With S' = 0 CPU halts at 1176
With S' = 1 CPU halts at 1177

7. Command an address change to the test board to 4002 and 4003 as follows:

Next Instruction Register = 25027746
"A" Register = 00004002

8. Depress Execute. Do not initialize the computer until this portion of the test is complete because the test board will revert back to 776/777.

> NOTE
>
> It will be necessary be bypass LOC/100 (ABORT). See 10.6.7, Step 1.

The interrupt vectors now will be as follows:

```
RX = 2 x 2 + 200 = 204  Type 0
                   +1
                  205  Type 1
```

TX = 2 x 3 + 200 = 206 Type 0

+1

207 Type 1

9. Store into the Interrupt Vectors:

Location 204 = 14000204 RX Type 0
205 = 14000205 RX Type 1
206 = 14000206 TX Type 0
207 = 14000207 TX Type 1

10. Change Location 102 of test program to 25014002 (ACT, RX).

11. Repeat steps 3 through 6, but this time use the new addresses for the receiver and transmitter.

10.6.8 Inhibitable APIs

1. Restore the program (see 10.6.7.1) at Location 100, except change Location 102 to 25017726. (S' = 2, request Inhibitable API)

2. Start the program and lower API lockout switch.

   API 1174 should NOT occur because the PAI light is off.

3. Stop the program and change Location 101 to PAI (25020000).

4. Restart the program and lower the API lockout switch. This time the computer should halt at Location 1174.

10.6.9 TIM/TOM APIs

1. Set up a TIM/TOM control word into the transmitter's Type 0 interrupt vector as follows:

Location 1176 = 00740777 (full word from /1000)

Location 1174 = 76741777

2. Set up a data pattern in Location 1000 of alternate one's. (25252525)

3. Load the following test loop:

Location 100 25034070 ABT
101 25020000 PAI
102 25027717 OPR S' = 1 Trigger TOM
103 25027716 Trigger TIM
104 26200000 NOP
105 14040000 BRU* Halt

4. Lower API lockout switch and start the above program.

   When program halts, examine the test board's data register for 25252525 pattern.

   Location 1176 should contain 01741000, indicating the TOM control word updated properly.

   Location 2000 should contain the 25252525 data pattern and the TIM control word at /1174 should be 77741000.

   The program loop given may be used for trouble-shooting if the BRU* at 105 is changed to BRU 100. However, due to the type of failure, the TOM control word may have to be reset on each loop which will require additional programming.

APPENDIX A

4400AM131 I/O TEST AID

This appendix section lists and describes the operation of the commands used in exercising the model 4400AM131 GENIE* I/O Test Aid. The test aid, referred to in this description as a Device Emulator, is a printed wiring board (PWB) that can be plugged into any GENIE Bus card slot for the purpose of emulating a device controller. It can be used in conjunction with the GENIE Bus test program to verify the operation of the GENIE Bus. It can also be used in off-line troubleshooting of the GENIE Bus with single-step or continuous looping instructions. In contains a bank of indicators that provides the user with a visible indication of the actions and data transfers taking place when the emulator is accessed.

The following actions can be performed or initiated by program control of the Device Emulator:

- Interrupt Generation

  Non-inhibitable TIM/TOM/ECHO/EOR interrupts, as well as inhibitable event interrupts, can be initiated by program control of the emulator.

- Direct Memory Access (DMA) IN or DMA OUT

  DMA IN operations write information to core memory; DMA OUT operations read information from memory.

- GEN 2 Commands

  IN/OUT/OPR/ACT/ABT

- Force the following GBC conditions:

  a) Address parity errors

  b) Data parity errors

  c) Timeout alarms

  d) Enabling of "ready" and "error" lines

ADDRESS AND PRIORITY

The Device Emulator contains a switch bank that permits selectable priority in the same manner as other GENIE Bus device controllers. (Refer to theory publication for description on priority selection.)

There is no address switch bank, as other GENIE Bus device controllers have; instead, the Device Emulator can be assigned an address by a program command. It retains the address until re-programmed or initialized.

Any Abort command addressed to the Device Emulator automatically sets its receiver address to /7706 and its transmitter address to /7707. These two addresses are the two highest possible GENIE Bus addresses. When programmed, the receiver must always be assigned an even address and the transmitter portion is automatically assigned an odd address one higher than the receiver. (Add bit 1 to the receiver address to determine the transmitter address.) All data outputs from the Processor are gated to the transmitter address. Data Input transfers are from the receiver address.

*Trademark

CMD	S' (A)	ADDRESS (B)	"A" REGISTER (C)	ACTION
ABT (S=3)	0	GBC	--	Clears GBC. Does not affect Device Emulator.
	0	TX/RX	--	Initializes test board, address register.
	7	GBC	--	Initializes GBC, test board control F/F's, address registers (376, 377).
OUT (S=4)	0	TX	Data Pattern	Transfers data out for readback by TIM. Processor's "A" Register's contents from GBC to Device Emulator's register. (Displayed at Device Emulator's lamps.)
IN (S=5)	0	RX	--	Contents of Device Emulator's register to GBC to Processor.
OPR (S=2)	0	TX	--	Set output channel busy. (Generates API.) Turns off "TX RDY" lamp. (1)
	0	RX	--	Set input channel busy. (No API generated.) Turns off "RX RDY" lamp.
	1	TX	--	Set output channel busy and set TOM; force TOM API. (2)
	1	RX	--	Set input channel busy and set TIM mode. Turns off "RX RDY" lamp. (TIM API if followed by OUT cmd.)
	2	TX	0 - 17 mem. adr. 18, 19 pack mode	Set DMA OUT mode (memory read); force memory request; set output channel busy. Data to Device Emulator. (3)
	2	RX	0 - 17 mem. adr. 18, 19 pack mode	Set DMA IN mode (memory write); set input channel busy. (4)
	3	X	$A_0 = 1$ → Force GBC Timeout. $A_1 = 1$ → Force data parity error. $A_2 = 1$ → Force address parity error. $A_3 = 1$ → Force JNE error.	(5)
	4		4002 thru 7706 octal	Assign device address. (6)
ACT (S=1)	0	X	--	Force Type 0 non-inhibitable interrupt.
	1	X	--	Force Type 1 non-inhibitable interrupt.
	2	X	--	Force Type 0 inhibitable interrupt. (7)
	3	X	--	Force Type 1 inhibitable interrupt. (8)

NOTES

(A) S' = instruction bits 3, 4, 5

(B) TX = output channel; RX = input channel; X = either channel

(C) -- = don't care

(D) Byte packing is specified by bits 18 and 19 from the "A" register in the following manner:

19	18		
0	1	= Byte transfer to/from M.S.	8 bits
1	0	= Byte transfer to/from Middle	8 bits
1	1	= Byte transfer to/from L.S.	8 bits
0	0	= Word transfer to/from	24 bits

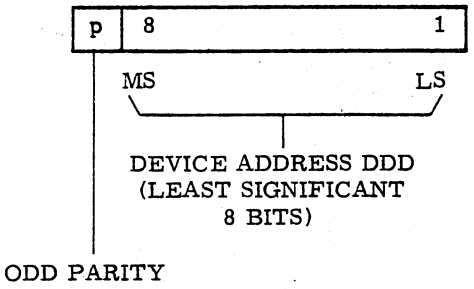
Table A.1 Device Emulator Commands

NOTES KEYED TO TABLE A.1

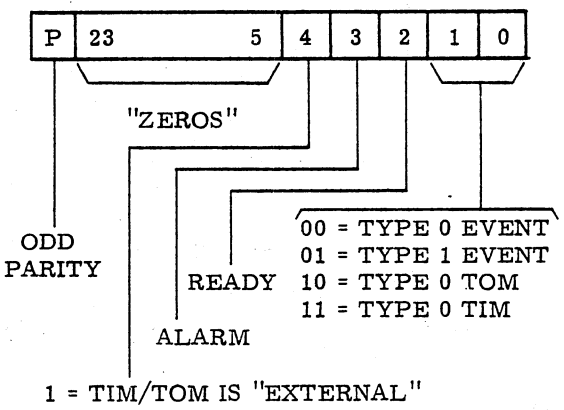
(1) A data ready Interrupt request is generated as a result of the setting of the channel busy. This will, when serviced, result in an access of the interrupt vector corresponding to the address being used. The interrupt vector should have an SPB to an output data routine. Each subsequent output (see OUT, S = 4, TX desc. in table) transfers data and sets the receiver data ready, generating a Type 0 Interrupt from the receiver address. The response routine may also contain an input command (see IN, S = 5, RX) to read data back to the Processor. With the output channel busy "true" (set "true" by the OPR, S' = 0), the receiver channel will go busy with each receipt of an IN command, triggering an interrupt from the transmitter. The OUT and IN commands, accessed via interrupts from the transmitter and receiver addresses, can, therefore, be used in a form of "ping-pong" cycle to transfer data out to the Device Emulator and then retrieve it for comparison.

(2) In forcing a TOM API, the test board tells the GBC that the interrupt is a TOM type by the information it returns in response to the GBC's acknowledge. The interrupt information and the device address are sent to the GBC in the following formats:

I/O DEVICE-ADDRESS BUS IMAGE



I/O DATA BUS IMAGE



TYPE 0 IS TIM, TOM, DMT OR $EVENT_0$
(HIGHER PRIORITY API OF EVENT DOUBLET)

TYPE 1 IS ECHO, EOR OR $EVENT_1$
(LOWER PRIORITY API OF EVENT DOUBLET)

(3) The packing mode bits used will determine whether the read operation from memory is to be a full word or only a byte that is output to the Device Emulator. Should this output be followed by an IN command (S = 5), another DMA output function will be initiated.

(4) This command is used to initiate a memory write operation from the Device Emulator. Of the information transferred from the "A" register during the OPR's execution, bits 0 through 17 will be the starting memory address and bits 18 and 19 will specify the packing mode. Once set up by this command, the Device Emulator can operate from GEN 2 OUT or TOM commands to perform continuous DMA write operations. (If TOM's are used, they must be preceded by an OPR, S' = 1 to trigger a TOM API from the transmitter.) The OUT or TOM command transfers to the Device Emulator the data to be written to memory.

Upon execution of the OUT or TOM command, the Device Emulator will request memory to transfer the information received. If the transfer is either full-word or from the most-significant byte position, the Device Emulator memory address register will increment one following the transfer. This will prepare for a subsequent write operation triggered by another OUT or TOM command.

On each memory write operation, the Acknowledge from memory increments the Device Emulator's packing mode flip-flops. Thus, if the operation is byte-oriented, the transfer will step from least to middle to most significant positions. Each time a transfer concludes at the most significant position, the Device Emulator's memory address register increments once.

The memory write operations from the Device Emulator can be halted using the ABT command, which initializes the Emulator PWB.

(5) This command can be sub-coded, using the four least significant bits of the "A" register, to force one of four internal GBC conditions, as specified under the "Action" column in Table A.1. The following descriptions explain the forced actions:

1) "A" Register = 00000001
(Force GBC Timeout Alarm)

The execution of the OPR (S' = 3) command with this code in the "A" register prevents a Slave Sync signal from being returned to the GBC, thereby permitting the GBC's timeout single-shot to time out and cause an alarm.

2) "A" Register = 00000002
(Force Data Parity Error)

The execution of the OPR (S' = 3) command with this code in the "A" register causes any subsequent IN command to send back an incorrect parity bit level, creating a parity alarm. The condition can be cleared by issuing an ABT command to the Emulator.

3) "A" Register = 00000004
(Force Address Parity Error)

The execution of the OPR (S' = 3) command with this code in the "A" register will cause the parity bit associated with the Emulator's device address at that time to reverse. On the next command addressed to the Emulator, the GBC's calculated parity bit and the Emulator's reversed parity bit will not agree. Slave Sync will not be returned, since there will be no address correspondence and a GBC Timeout Alarm will occur.

4) "A" Register = 00000010
(Force JNE line true)

The execution of the OPR (S' = 3) command with this code in the "A" register will cause Device Emulator's error flip-flop to set.

(6) The execution of the OPR (S' = 4) command can be used with the device address in the "A" register to assign a new address to the Emulator. The Emulator's address register can be set to any valid bus device address and it will retain the address for any future transfers. A subsequent console initialize or a program ABT, S' = 7 directed to the GBC (address 4000) will cause the Emulator's device address to be returned to /376. Only even addresses (e.g. 402, 404, etc.) can be used and they will be assigned to the Emulator's receiver; the Emulator's transmitter will automatically receive the next higher address.

(7) The execution of an ACT command with S' = 2 will force a Type 0, inhibitable API. Servicing of the API requires that the PAI flip-flop be set.

(8) The execution of an ACT command with S' = 3 forces a Type 1, inhibitable API. Servicing of the API requires that the PAI flip-flop be set.

READER COMMENTS

In order to improve future editions of this and other publications, Honeywell's Process Management Systems Division solicits your comments. You may direct them to the writer through this form, by letter, or telephone. The address and phone number are provided below. Some of the factors contributing to the usefulness of this publication are listed below. Please explain any "no" responses in the COMMENTS section.

Writer: Richard F. Hoover

Honeywell Inc.
Publications - 540
2222 W. Peoria Ave.
Phoenix, AZ 85029

(602) 997- 3380
HVN 364-3380

- Publication No. ACPU1GBC-M

- How is this publication used:

Familiarization	☐	Reference	☐
Training	☐	Maintenance	☐

Other (Explain) _____

	YES	NO
• Does this publication meet your requirements?	☐	☐
• Is the material:		
1) Presented in clear text?	☐	☐
2) Conveniently organized?	☐	☐
3) Adequately detailed?	☐	☐
4) Adequately illustrated?	☐	☐
5) Presented at appropriate technical level?	☐	☐
6) Technically accurate?	☐	☐

- Please provide specific text references (page number, line, etc.) with your comments.

NAME _____ DATE _____

TITLE _____

COMPANY NAME _____

ADDRESS _____

COMMENTS

No postage necessary if mailed in the U.S.A.

Staple

Communications concerning technical publications should be directed to:

Technical Publications - 540
Process Management Systems Division
Honeywell Inc.
2222 West Peoria Avenue
Phoenix, Arizona 85029

Fold Fold

FIRST CLASS
Permit No. 4091
Phoenix, Arizona

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

PROCESS MANAGEMENT SYSTEMS DIVISION
HONEYWELL INC.
2222 West Peoria Avenue
Phoenix, Arizona 85029

Attention: Manager, Technical Publications - 540

Fold Fold

Cut line

Additional Comments: