Contents

This page intentionally left blank.

Interrupts and Cycle Steal Requests

INTERRUPTS

The control processor executes instructions one at a time; one instruction is followed by the next sequential instruction. This sequence of instructions can be changed by executing a branch or jump instruction. The sequence can also be changed by passing control from (1) one interrupt level to another interrupt level of higher priority, or (2) to the main program level.

The control processor uses the main program level, six levels of interrupts, and two levels of cycle steal requests to interrupt the control storage program; the priority is set by hardware. Interrupt level 0 has the highest priority. An I/O device that needs servicing generates either an interrupt request or a cycle steal request to the control processor. The control processor receives interrupt requests 1, 2, 3, and 4 through the port. Only the main storage processor uses interrupt request 5. An interrupt request 5 is sent directly to the control processor. Interrupt requests 1 through 4 interrupt normal instruction sequence, while cycle steal requests do not interrupt this sequence.

The priorities specified for interrupt requests and cycle steal requests are as follows:

Priority	Interrupt Level or Cycle Steal Request	Function
1	Interrupt 0	Machine check routines
2	Burst cycle steal	Disk data movement; diskette data movement (level 2 attachment)
3 (note)	Base cycle steal	Work stations or MLCA
4	Interrupt 1	Disk data I/O supervisor (62EH); diskette data movement (level 1 attachment)
5	Interrupt 2	Communication line 1; communication line 2 or MLCA; fixed interval timer
6	Interrupt 3	Reserved
7	Interrupt 4	Line Printer; Work Stations; diskette seek (level 1 attachment); all diskette data disk operations (level 2 attachment); seek (62EH); disk data I/O supervisor and seek (62PC); 1255 Magnetic Character Reader
8	Interrupt 5	Main storage processor
9	Main program level interrupt	Service that can be delayed

Note: Display stations that interrupt in a base-cycle-steal mode check the interrupt level 2 line as a prerequisite. A request for a base cycle steal is made under the following conditions:

- If interrupt level 2 is not busy, request a base cycle steal immediately.
- If interrupt level 2 is busy, start a display station timer and request a base cycle steal when the timer times out or when interrupt level 2 becomes not busy.

Base cycle steal has a priority of 3 but uses work registers 4 and 5 from interrupt level 4 (priority 7).

If the control processor is not working with interrupts, it is executing instructions on the main program level.

Each interrupt level and the main program level has a set of registers in the local storage register stack. Interrupt level 0 and the main program level use the same set of work registers.

Each interrupt level and the main program level has the following set of local storage registers:

- Eight 16-bit work registers.
- One microaddress register, which stores the address of the current instruction.
- One microaddress backup register, which stores the return address when a branch-and-link instruction is executed.

Having a specified set of registers for each interrupt level means that no data is lost when a higher level interrupts a lower level. The registers are saved and restored by program routines that rely on the function being performed.

A device sets an interrupt level by sending an interrupt request signal through its adapter. The control storage program performs the following functions:

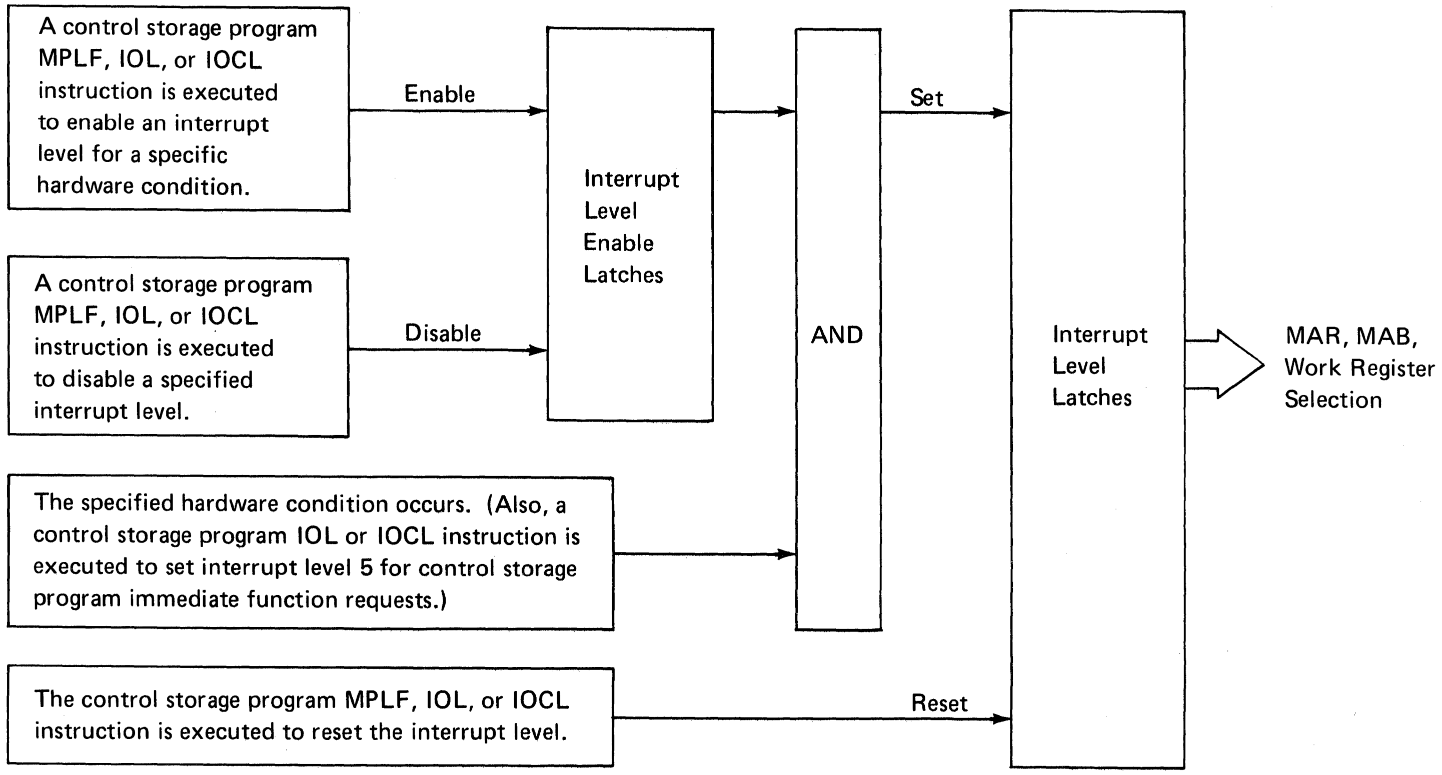
- Stores the processor condition register
- Initializes the processor condition register
- Services the interrupt
- Returns the contents of the processor condition register
- Resets the interrupt

All interrupt levels use the same processor condition register; therefore, the processor condition register must be saved when entering an interrupt level and then restored when leaving the interrupt level. This process ensures that the contents of the processor condition register are correctly associated with the interrupt level in process. (If the interrupt level program does not change the processor condition register, the save/restore process is not necessary.)

When the port senses an interrupt request, complete service may be performed on the interrupt level (immediate action), or the interrupt level program may store the necessary information for later processing on the main program level (delayed action). The level of interrupt requested does not rely on hardware only; it can also be changed by software to meet the needs of the program for different devices. Therefore, the processing of interrupts can be grouped into two classes:

- Immediate action–The system responds immediately to any device or system condition that has a critical response time. The routine services the interrupt request until the request is complete or until a critical response is complete. Program service continues when the system becomes available. Examples of immediate action processing are machine checks, disk and diskette data moves, and communications interrupt routines.
- Delayed action–After servicing an interrupt, the control storage program routine may determine that complete service can be temporarily delayed because response time for the device is not critical; that is, the device probably cannot cause another interrupt request during the time it takes to complete a routine. Therefore, the interrupt request is posted and service is delayed until the current routine is completed.

Local Storage Register Stack


Enable
Interrupt Level Enable Latches
Disable
AND
Set
Reset
Interrupt Level Latches
MAR, MAB, Work Register Selection

A control storage program MPLF, IOL, or IOCL instruction is executed to enable an interrupt level for a specific hardware condition.

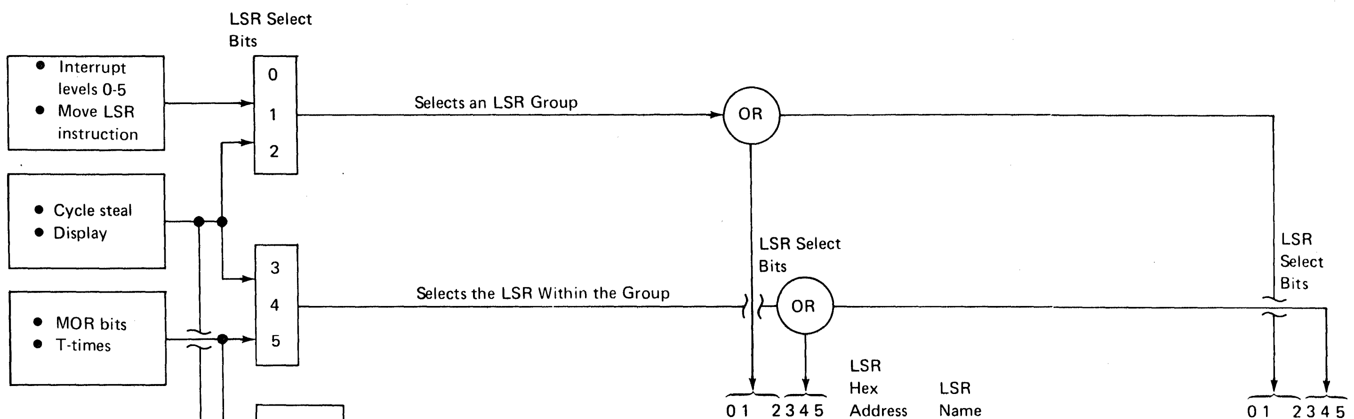
A control storage program MPLF, IOL, or IOCL instruction is executed to disable a specified interrupt level.

The specified hardware condition occurs. (Also, a control storage program IOL or IOCL instruction is executed to set interrupt level 5 for control storage program immediate function requests.)

The control storage program MPLF, IOL, or IOCL instruction is executed to reset the interrupt level.

Interrupt Level	Microaddress Register (Hex)	Microaddress Backup Register (Hex)	Work Register Physical (Hex)	Work Register Logical (Hex)
0	0A	0B	00–07	WO–W7
1	0C	0D	10–17	WO–W7
2	0E	0F	18–1F	WO–W7
3	28	29	20–27	WO–W7
4	2C	2D	30–37	WO–W7
5	2E	2F	38–3F	WO–W7
MPL	08	09	00–07	WO–W7

Note: Interrupt levels are shown in priority order.



	0 1	2 3 4 5	LSR Hex Address	LSR Name
Main Level or Machine Check 0	0 0	0 0 0 0	00	WR0
	0 0	0 0 0 1	01	WR1
	0 0	0 0 1 0	02	WR2
	0 0	0 0 1 1	03	WR3
	0 0	0 1 0 0	04	WR4/CS0
	0 0	0 1 0 1	05	WR5/CS1
	0 0	0 1 1 0	06	WR6/CS2
	0 0	0 1 1 1	07	WR7/CS3
MAR/MAB Stack 1	0 0	1 0 0 0	08	MAR
	0 0	1 0 0 1	09	MAB
	0 0	1 0 1 0	0A	MAR (MC)
	0 0	1 0 1 1	0B	MAB (MC)
	0 0	1 1 0 0	0C	MAR-1
	0 0	1 1 0 1	0D	MAB-1
	0 0	1 1 1 0	0E	MAR-2
	0 0	1 1 1 1	0F	MAB-2
Interrupt 1	0 1	0 0 0 0	10	WR0
	0 1	0 0 0 1	11	WR1
	0 1	0 0 1 0	12	WR2
	0 1	0 0 1 1	13	WR3
	0 1	0 1 0 0	14	WR4/CS0
	0 1	0 1 0 1	15	WR5/CS1
	0 1	0 1 1 0	16	WR6/CS2
	0 1	0 1 1 1	17	WR7/CS3
Interrupt 2	0 1	1 0 0 0	18	WR0
	0 1	1 0 0 1	19	WR1
	0 1	1 0 1 0	1A	WR2
	0 1	1 0 1 1	1B	WR3
	0 1	1 1 0 0	1C	WR4/CS0
	0 1	1 1 0 1	1D	WR5/CS1
	0 1	1 1 1 0	1E	WR6/CS2
	0 1	1 1 1 1	1F	WR7/CS3

	0 1	2 3 4 5	LSR Hex Address	LSR Name
Interrupt 3	1 0	0 0 0 0	20	WR0
	1 0	0 0 0 1	21	WR1
	1 0	0 0 1 0	22	WR2
	1 0	0 0 1 1	23	WR3
	1 0	0 1 0 0	24	WR4/CS0
	1 0	0 1 0 1	25	WR5/CS1
	1 0	0 1 1 0	26	WR6/CS2
	1 0	0 1 1 1	27	WR7/CS3
MAR/MAB Stack 2	1 0	1 0 0 0	28	MAR-3
	1 0	1 0 0 1	29	MAB-3
	1 0	1 0 1 0	2A	Spare
	1 0	1 0 1 1	2B	Spare
	1 0	1 1 0 0	2C	MAR-4
	1 0	1 1 0 1	2D	MAB-4
	1 0	1 1 1 0	2E	MAR-5
	1 0	1 1 1 1	2F	MAB-5
Interrupt 4	1 1	0 0 0 0	30	WR0
	1 1	0 0 0 1	31	WR1
	1 1	0 0 1 0	32	WR2
	1 1	0 0 1 1	33	WR3
	1 1	0 1 0 0	34	WR4/CS0
	1 1	0 1 0 1	35	WR5/CS1
	1 1	0 1 1 0	36	WR6/CS2
	1 1	0 1 1 1	37	WR7/CS3
Interrupt 5	1 1	1 0 0 0	38	WR0
	1 1	1 0 0 1	39	WR1
	1 1	1 0 1 0	3A	WR2
	1 1	1 0 1 1	3B	WR3
	1 1	1 1 0 0	3C	WR4/CS0
	1 1	1 1 0 1	3D	WR5/CS1
	1 1	1 1 1 0	3E	WR6/CS2
	1 1	1 1 1 1	3F	WR7/CS3

Interrupt Levels

Interrupt Level 0

Interrupt level 0 (machine check interrupt level) is set when the system finds a control processor parity check, address not valid, or control processor program check while running diagnostic programs. This interrupt can also be set by a port check while running diagnostic programs. Customer user programs set interrupt level 0 and cause an I/O immediate (MPLF) instruction (hex BEA3–processor check halt) to be executed by the control processor and stop the system.

Interrupt level 0 occurs when one of the following is found:

- A parity check (storage data register, control storage address register, storage gate, micro-operation register, arithmetic and logic unit gate, or MSAR during CP operation to MSP)

- An address not valid (control processor addressing main storage or control storage)

- A channel check

- A control processor instruction loop that is too long (7-second time-out)

Interrupt level 0 has priority over all other interrupts and cycle steal requests. The work registers used by interrupt level 0 are the same registers used by the main program level (local storage registers 0 through 7). During normal processing, a machine check results in a processor check.

Interrupt Level 1–Disk and Diskette

Interrupt level 1 and associated registers are shared between the disk and the diskette (level 1 attachment) for data movement. This interrupt request is the fourth highest priority.

Interrupt Level 2–Communications and Fixed Interval Timer

Interrupt level 2 and associated registers are used to support fixed interval timer and data communications. Data communications include binary synchronous communication and synchronous data link control. This interrupt request is the fifth highest priority.

Interrupt Level 3

Interrupt level 3 is reserved and is the sixth highest priority.

Interrupt Level 4

Interrupt level 4 is used by:

- Work stations

- Printers

- Disk seek operations

- Diskette seek operations (level 1 attachment)

- Diskette (level 2 attachment)

- Work stations (base cycle steal request)–work registers 4 and 5 are selected from interrupt level 4 for storage addressing

- 1255 Magnetic Character Reader

When interrupt level 4 is set, the control storage program executes a sense interrupt level status byte (SILSB) instruction to all attachments. Each device has an assigned bit in the sense interrupt level status byte on the 'data bus in' line. This bit is activated by the interrupt device during the sense interrupt level status byte (SILSB) instruction. This interrupt request is the seventh highest priority.

Interrupt Level 5

Interrupt level 5 is set by either the main storage processor or the control storage program as follows:

- Main storage processor:
  - Executes a supervisor call (SVC) instruction
  - For a main storage processor stop condition:
    a. When the Stop key on the CE subpanel is activated
    b. After each step (when in step mode)
    c. After an equal compare (when operating in address compare mode)
  - Tries to execute a nonexecutable instruction
  - When a processor error condition is active

- Control storage program:
  - Executes an I/O control load (IOCL) instruction to set interrupt level 5
  - Then, executes an I/O load (IOL) instruction to enable interrupt level 5
  - Executes a control processor load function (MPLF) that stops the main storage processor

This interrupt request is the eighth highest priority.

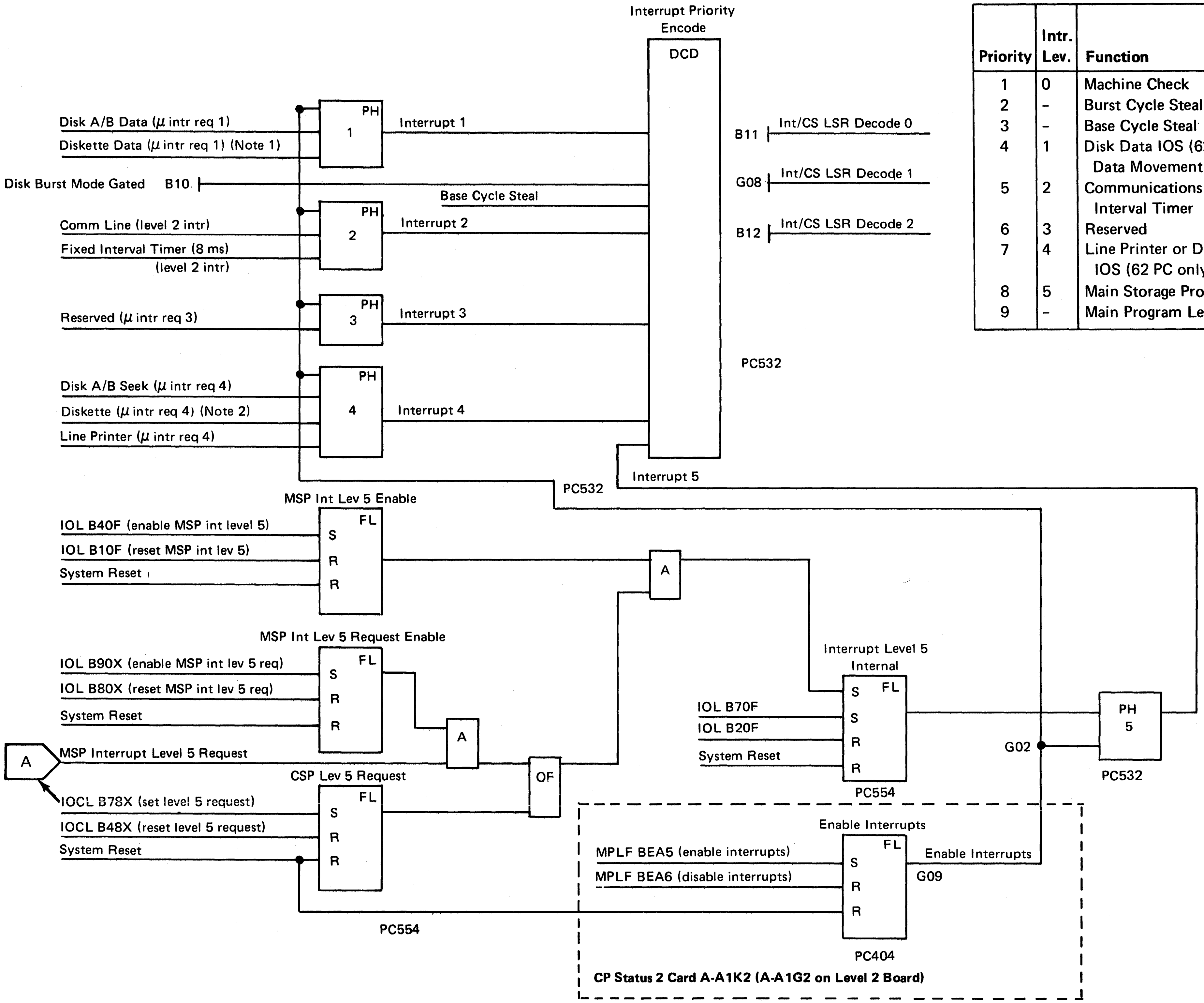
Main Program Level Interrupt

The main program level interrupt has the lowest priority level and includes all instructions or tasks in which the source can be delayed without affecting the results.

Note: Priority level 2 and priority level 3 are not described on this page. These are cycle steal requests and are described earlier in this section under *Interrupts.*

Interrupt/Cycle Steal Priority Encode and LSR Selection

CP Port Card A-A1L2 (A-A1H2 on Level 2 Board)


Interrupt Priority Encode
DCD
PC532
Disk A/B Data (μ intr req 1)
Diskette Data (μ intr req 1) (Note 1)
PH 1
Interrupt 1
Disk Burst Mode Gated B10
Base Cycle Steal
Comm Line (level 2 intr)
Fixed Interval Timer (8 ms) (level 2 intr)
PH 2
Interrupt 2
Reserved (μ intr req 3)
PH 3
Interrupt 3
Disk A/B Seek (μ intr req 4)
Diskette (μ intr req 4) (Note 2)
Line Printer (μ intr req 4)
PH 4
Interrupt 4
B11 Int/CS LSR Decode 0
G08 Int/CS LSR Decode 1
B12 Int/CS LSR Decode 2
Interrupt 5
PC532
MSP Int Lev 5 Enable
IOL B40F (enable MSP int level 5)
IOL B10F (reset MSP int lev 5)
System Reset
FL S R R
A
MSP Int Lev 5 Request Enable
IOL B90X (enable MSP int lev 5 req)
IOL B80X (reset MSP int lev 5 req)
System Reset
FL S R R
A
A
MSP Interrupt Level 5 Request
CSP Lev 5 Request
OF
IOCL B78X (set level 5 request)
IOCL B48X (reset level 5 request)
System Reset
FL S R R
PC554
Interrupt Level 5 Internal
IOL B70F
IOL B20F
System Reset
FL S S R R
PC554
G02
PH 5
PC532
Enable Interrupts
MPLF BEA5 (enable interrupts)
MPLF BEA6 (disable interrupts)
FL S R R
Enable Interrupts
G09
PC404
CP Status 2 Card A-A1K2 (A-A1G2 on Level 2 Board)

Priority	Intr. Lev.	Function	Work Register (hex)	MAR (hex)	MAB (hex)	LSR Decode 2	1	0
1	0	Machine Check	00-07	0A	0B	0	0	0
2	-	Burst Cycle Steal	14-17	—	—	0	1	0
3	-	Base Cycle Steal	34-37	—	—	1	1	0
4	1	Disk Data IOS (62 EH only) or Diskette Data Movement (Note 1)	10-17	0C	0D	0	1	0
5	2	Communications Lines or Fixed Interval Timer	18-1F	0E	0F	0	1	1
6	3	Reserved	20-27	28	29	1	0	0
7	4	Line Printer or Disk Seek or Disk Data IOS (62 PC only) Diskette Seek (Note 2)	30-37	2C	2D	1	1	0
8	5	Main Storage Processor	38-3F	2E	2F	1	1	1
9	-	Main Program Level	00-07	08	09	0	0	0

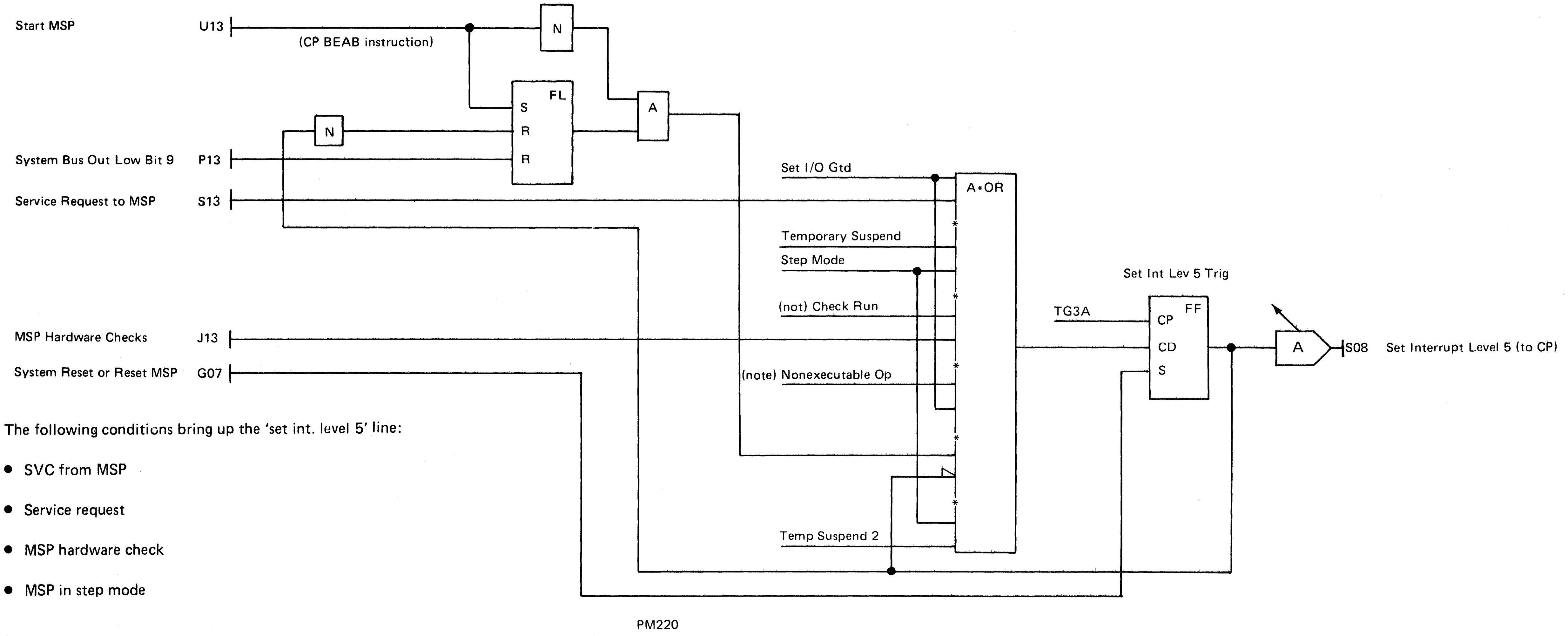
Notes:

1. Interrupt level 1 is used for diskette data movement if a level 1 diskette attachment is installed.*
2. Interrupt level 4 is used for all diskette operations if a level 2 diskette attachment is installed, and for diskette seek operations if a level 1 diskette attachment is installed.*

*See FSL page AC300 to determine which attachment is installed.

MSP Set Interrupt Level 5

MSP System Control Card A-A1N2 (A-A1J2 on Level 2 Board)



The following conditions bring up the 'set int. level 5' line:

- SVC from MSP
- Service request
- MSP hardware check
- MSP in step mode

Note: Nonexecutable op will be active for an MSP SVC as well as an invalid op.

CONTROL PROCESSOR AND MAIN STORAGE PROCESSOR COMMUNICATION

When the main storage processor needs to send data to or get data from an I/O device, a supervisor call instruction in the user program is executed. The supervisor call instruction stops the main storage processor and sets control processor interrupt level 5. The control processor determines that the main storage processor executed a supervisor call instruction. The control processor then gets the information needed from the main storage processor to determine the I/O command that is to be executed, saves it, and resets interrupt level 5. The control processor starts the I/O function later under control of the control storage main program level.

In order for the control processor to move data to or from the main storage processor, the main storage processor is stopped so that the control processor can access the main storage processor registers and main storage. The control processor stops the main storage processor clock. Only at specified times, while executing instructions, can the control processor stop the main storage processor. After stopping the main storage processor clock, the control storage program saves the contents of the main storage processor registers before moving any data. Before starting the main storage processor clock again, the control processor restores the contents of the main storage processor registers. After this, the control processor starts the main storage processor clock.

In effect, the main storage processor is an I/O device connected to the control processor. The control processor stops the main storage processor when it is necessary to stop the current task, sense registers, load registers, or to access main storage. When the control processor needs to stop the task that is running, it activates the 'service request' line.

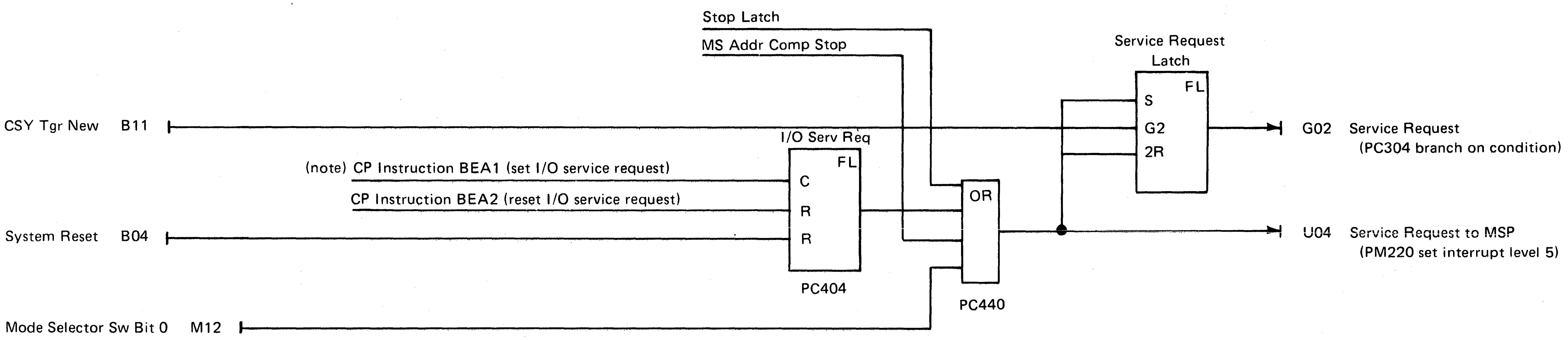
The main storage processor stops when it completes executing the current system instruction and requests an interrupt level 5. If the control processor needs access to main storage on main storage registers while the main storage processor is running, hardware control stops the control processor until the main storage program is at a point in its execution where it can be stopped ('temporary suspend' signal). The main storage processor clock stops and the control processor accesses the desired register or main storage location. The main storage processor then restarts at the point it was stopped.

Communication between the control processor and the main storage processor is controlled by the control storage program. How the control storage program controls data movement is described more fully in the *Control Storage Logic Manual*.

Service Request Line Generation

Generation of the 'service request' line is shown in the following diagram.

CP Status 2 Card A-A1K2 (A-A1G2 on Level 2 Board)


Stop Latch
MS Addr Comp Stop
Service Request Latch
S
FL
G2
2R
G02 Service Request (PC304 branch on condition)
CSY Tgr New B11
I/O Serv Req
FL
C
R
R
(note) CP Instruction BEA1 (set I/O service request)
CP Instruction BEA2 (reset I/O service request)
System Reset B04
OR
U04 Service Request to MSP (PM220 set interrupt level 5)
PC404
PC440
Mode Selector Sw Bit 0 M12

Note: When the control processor needs to stop the MSP task that is running, control storage issues a BEA1 instruction (set I/O service request) that brings up service request to MSP. The MSP will stop at the end of the current system instruction (temporary suspend).

CONTROL PROCESSOR AND CHANNEL COMMUNICATION

The I/O attachments and the controller communicate with the control processor through the channel by sending an 'interrupt request' signal or by sending a 'cycle steal request' signal. The disk or the display stations can communicate by sending either signal, but all other devices must send an interrupt request.

An interrupt request generates a signal to the control processor that sets the interrupt level. After the interrupt function is complete, the interrupt level is reset. A lower-level interrupt may be interrupted by a higher-level interrupt if the interrupt-enable condition ('enable interrupt' line) is active. The interrupt-enable condition is made not active when a lower-level interrupt must not be interrupted because of the danger of losing the sequence or integrity of data.

A cycle steal request generates a signal to the control processor that causes the control processor to stop executing instructions. A storage cycle is then taken between instructions in order to move the data into or out of control storage or main storage. The control processor stops the main storage processor clock if data is to be moved to or from main storage. After data movement is complete, the control processor permits the main storage processor clock to start again and instructions are executed normally.

Remember that the interrupt method of moving data causes the program routine to call another routine to control data movement. The cycle steal method of moving data stops instruction processing to get access to storage for data movement. The cycle steal method of data movement is divided into two types: base-cycle-steal mode and burst-cycle-steal mode.

Base-cycle-steal mode is for slow data transmit applications and causes the control processor to stop executing instructions long enough to move 1 byte of data to or from control storage or main storage. The control processor stops the main storage processor clock if data is to be moved to or from main storage. After data movement is complete, the control processor permits the main storage processor clock to start again and instructions are executed normally.

In burst-cycle-steal mode, the control processor stops executing instructions until a complete block of data is moved to control storage or main storage. The control processor stops the main storage processor clock if data is to be moved to or from main storage. After data movement is complete, the control processor permits the main storage processor clock to start again and instructions are executed normally.

This page intentionally left blank.