

operations begin). The second reason is that interactions with high-speed I/O devices are provided by means of controller chips that perform many of the tasks (including temporary data storage) that would otherwise be performed by interrupt service routines. The master program is never concerned with multi-tasking or with close synchronization of I/O device operations. Note that while some input (e.g., keyboard entry), is not program controlled, its data entry rate is slow enough to allow temporary data storage by hardware while the processor is occupied.

In place of hardware interrupts, the hardware elements of the Intellec Series II development system use service requests (specified bits within status bytes) to determine when a device requires attention. The status bytes are returned to the master processor of the IPB/IPC on demand. The transfer of status bytes from the IOC or the PIO to the IPB/IPC constitutes a major segment of the interprocessor traffic. Although hardware interrupts are not widely used by Intel-supplied resident master programs, interrupt handling circuits exist within the Intellec Series II development system. These circuits establish priorities for the interrupt switches, for interrupts from the Multibus interface and for interrupts from internal hardware elements of the IPB/IPC, IOC, and PIO. The interrupt circuits also apply interrupt switch and internal hardware interrupts to the Multibus interface. Interrupt masking by resident programs inhibits sensing of interrupts by the IPB/IPC master processor. All interrupts generated by internal hardware elements are handled by a local interrupt controller that operates in the polled mode as a slave to the system interrupt controller. All internal interrupts are processed by the local interrupt controller and generate a level 7 interrupt to the system controller. Resident master programs must then poll the local interrupt controller to determine the source of the internal interrupt. An example of an interrupt routine used to service an interrupt originating from a device associated with the local interrupt controller is shown in Appendix A. Interrupt level assignments for the local interrupt controller are as follows (level 0 has the highest priority):

Level	Function
0	Serial I/O Channel 0 Input Data Ready
1	Serial I/O Channel 0 Output Data Ready
2	Serial I/O Channel 1 Input Data Ready
3	Serial I/O Channel 1 Output Data Ready
4	1ms Real Time Clock Interrupt
5	PIO Subsystem Interrupt
6	IOC Interrupt
7	Not Used

The system interrupt controller operates in the fully-nested mode and is initialized with a call address interval of eight and a base address of 0H to establish

the location of the vector address block. The vector addresses reserved for system interrupts are as follows:

Interrupt Level	Vector Address	Vector Usage
0	00H	Monitor
1	08H	ISIS-II
2	10H	Disk Controller
3	18H	
4	20H	ICE-80 Module
5	28H	
6	30H	ICE Modules
7	38H	Local Interrupt Controller



Reprogramming the 8259's call address interval from eight to four will cause undefined system operation.

The local and system interrupt controllers of the IPB and the system interrupt controller of the IPC cannot be programmed or polled by a non-resident master program (the local interrupt controller of the IPC can be accessed by another bus master). When another bus master assumes control of the bus, both the local and system interrupt controllers maintain any current interrupt request and latch any subsequent interrupt request (when the IPB/IPC regains bus access, any pending interrupt request is serviced). Since all system interrupts can be sensed by another bus master via the Multibus interface, all IPB/IPC local interrupts can also be sensed (any local interrupt causes a level 7 system interrupt).

2.4 MULTIBUS® PRIORITY LOGIC

To avoid conflicts that may arise when two or more bus masters simultaneously require bus access, the IPB/IPC includes parallel priority resolution logic. This logic accepts individual bus request inputs from up to nine bus masters that may be installed in the backplane (five available slots in the development system chassis and four slots in the expansion chassis) and returns an individual bus priority input to all but the bottom slot of the expansion chassis (the bottom slot has the highest priority and its bus priority input is permanently enabled). The parallel priority logic samples all of the bus request (BREQ) inputs and generates an individual bus priority in (BPRN) output to the highest priority bus master requesting the bus. Following Multibus interface protocol, when the requesting master receives its bus priority in signal, it examines the common bus busy (BUSY) bidirectional line to determine when the bus becomes available and, when the bus is available, activates bus busy to indicate to all other bus masters