

Table 2-3. Dedicated and Reserved I/O Port Addresses

Device	I/O Port Addresses
ICE-80 In-Circuit Emulator	0E0H-0E3H
Other ICE Modules (ICE-85, ICE-86, ICE-88, etc.)	080H-083H
iSBC 80/05 Single Board Computer	00H-05H
iSBC 80/10A Single Board Computer	0E4H-0EFH
iSBC 80/20-4 Single Board Computer	0D4H-0DFH, 0E4H-0EFH
iSBC 80/30 Single Board Computer	0D8H-0DFH, 0E4H-0EFH
iSBC 86/12A Single Board Computer	0C0H-0CFH, 0D0H-0DFH
iSBC 544 Intelligent Communications Controller	0D0H-0DFH, 0E4H-0EFH
Disks	
First Floppy Diskette Controller	078H-07FH
Second Floppy Diskette Controller	088H-08FH
Hard Disk Controller	068H-06FH

Future Intel products may require I/O port addresses other than the addresses specified in table 2-3. To prevent possible incompatibility with future Intel products, all user-device I/O port addresses should be switch or jumper selectable.

The use of I/O port addresses to access user-designed hardware is the most common technique employed within Intellec Series II development systems. A second technique, called "memory-mapped I/O," may also be employed to access user hardware. With memory-mapped I/O, a block of memory addresses is assigned to the device and/or its controller. The major advantage of memory-mapped I/O is additional programming flexibility; any instruction that references memory can be used to access an I/O port located in the memory space. For example, the MOV (move) instruction can transfer data between any 8080/8085 register and a port or any of the logical instructions can be used to manipulate individual bits within I/O device registers. The simplest implementation of a memory-mapped I/O device is in a system that has unused address space (i.e., a Model 220 with 32k bytes of RAM) in which case the I/O device addresses would be assigned within the 32k to 62k address range. However, in systems containing 64k of RAM (i.e., the Models 225 and 230), the I/O device addresses must replace either RAM or ROM. RAM replacement is impractical as it would necessitate the physical removal of RAM. ROM replacement requires I/O device control logic that generates INH1/ (to inhibit RAM) when the I/O device is being used. Since most memory-mapped

I/O devices require very few addresses, ROM replacement is the most widely used method for memory-mapped I/O. However, programmers must be aware that the addresses assigned to an I/O device are no longer accessible in RAM.

## 2.3 INTERRUPT MECHANISMS

The standard definition of an interrupt is the process whereby an asynchronous device-generated signal causes program branching to a routine that services the needs of the device (usually the handling of I/O data). However, the reason for employing interrupts is that I/O devices (or other real-time system elements) often involve synchronous operations that cannot be delayed while the processor is performing a lengthy processing task. An interrupt service routine is used to quickly complete the I/O data transfer and then allow continuation of the interrupted process. At a later time, the program processes the data accepted by the service routine or prepares new data for output to the device. Without interrupts, input data could be lost or output data could be unavailable when required.

There are two reasons why the preceding real-time hardware interrupts are not required when an Intellec Series II development system is executing many of the Intel-supplied resident master programs (Monitor, ISIS, etc.). The first reason is that I/O operations are program controlled to be performed in sequence (e.g., if diskette data is to be printed, the diskette operations are completed before printing