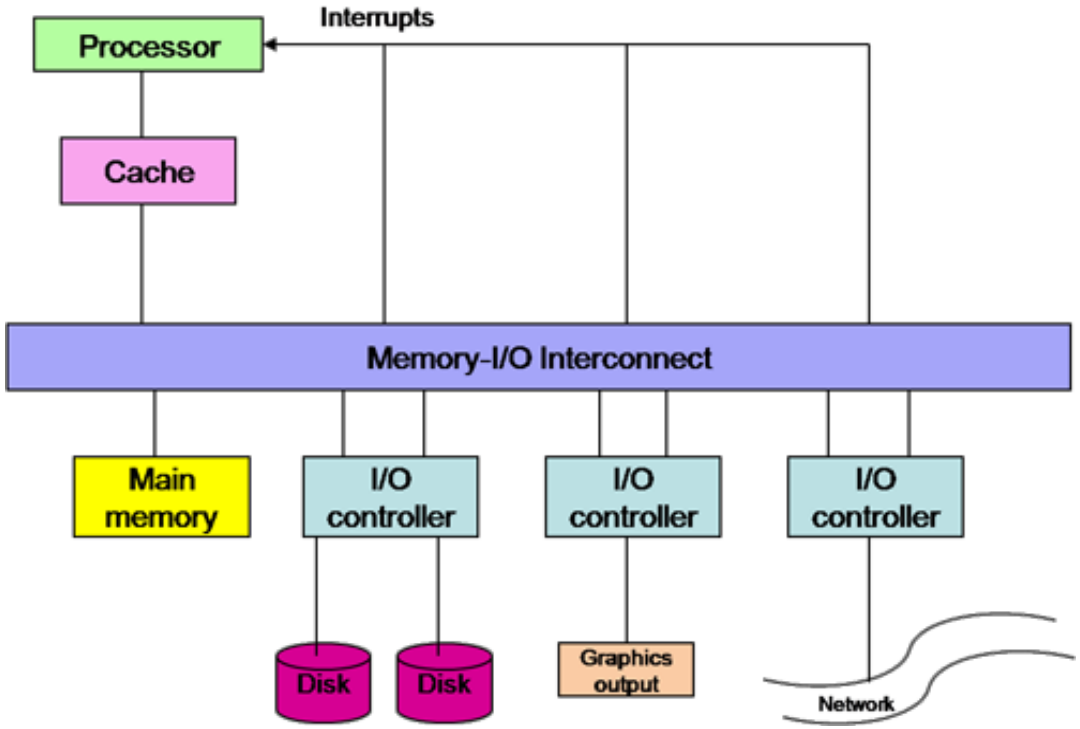
I/O System

I/O controllers manage communication between I/O devices and the CPU or main memory.



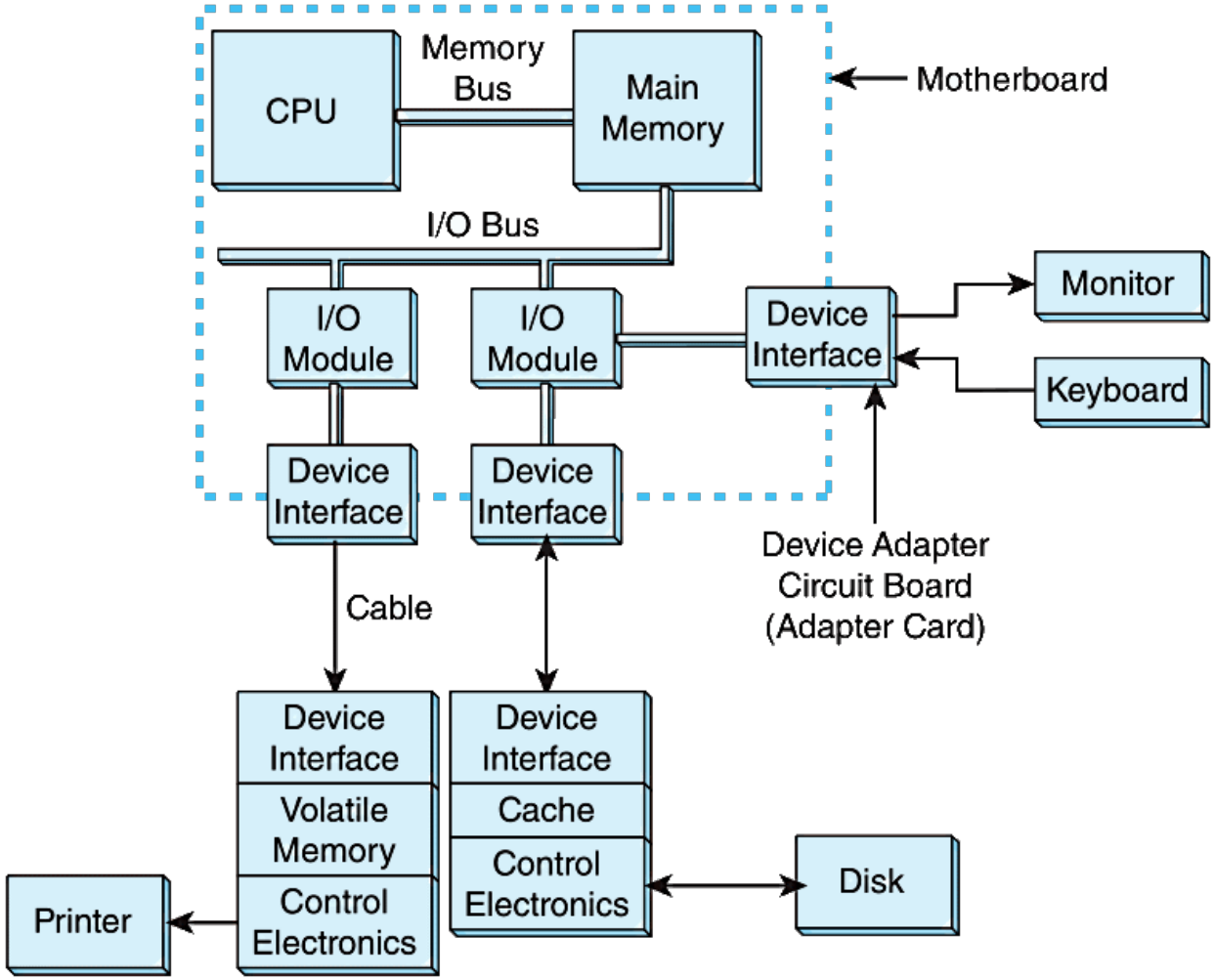
I/O transactions adhere to well-defined rules (*protocol*)

I/O System

I/O subsystems include:

- Blocks of main memory that are devoted to I/O functions.
- Buses that move data into and out of the system.
- Control modules in the host and in peripheral devices
- Interfaces to external devices such as keyboards and disks.
- Cabling or communications links between the host system and its peripherals.

I/O Architecture



Various types of buses allow for communication among components.

I/O Access Methods

- Devices can be accessed in one of two ways:

1 *Memory mapped*

  - One or more registers are assigned to each I/O device
  - The registers correspond to specified memory addresses
  - Part of the address space is thus reserved for I/O
  - Memory access instructions can also perform I/O
  - RISC systems tend to use this approach

2 *Port mapped or Isolated I/O*

  - Each device is assigned one or more port numbers
  - Special I/O instructions transmit data via the ports
  - Access must be identified as memory or I/O accesses
  - CISC systems tend to use this approach

I/O Access Examples

MIPS example:

```
    lui    $t0,0xFFFF    # base address for keyboard
    lw     $t1,4($t0)    # read the next input character
```

Pentium example:

```
KB_DATA    EQU   60H   # port number for keyboard

    in     AL,KB_DATA     # read character from port into AL register
```

Conclusion

Next we will examine more details of low level I/O

- Programmed I/O
- Interrupt Driven I/O
- DMA