Chapter- 6
Input – Output Organization

6.1 Input – Output Interface

Ref. Book Name : Computer System Architecture, M. Morris Mano

- I/O interface provides method for transferring information between internal storage and external I/O devices.

- Peripherals connected to computer need communication link for interfacing.

- Purpose of communication link is to resolve differences exist between central computer and each peripheral.

- The major differences are:

1. Peripherals are electro magnetic devices, while CPU and memory are electronic devices.
2. Data transfer rate of peripherals is slower than CPU.
3. Different data codes and format.
4. Different operating modes.

I/O bus and Interface Modules

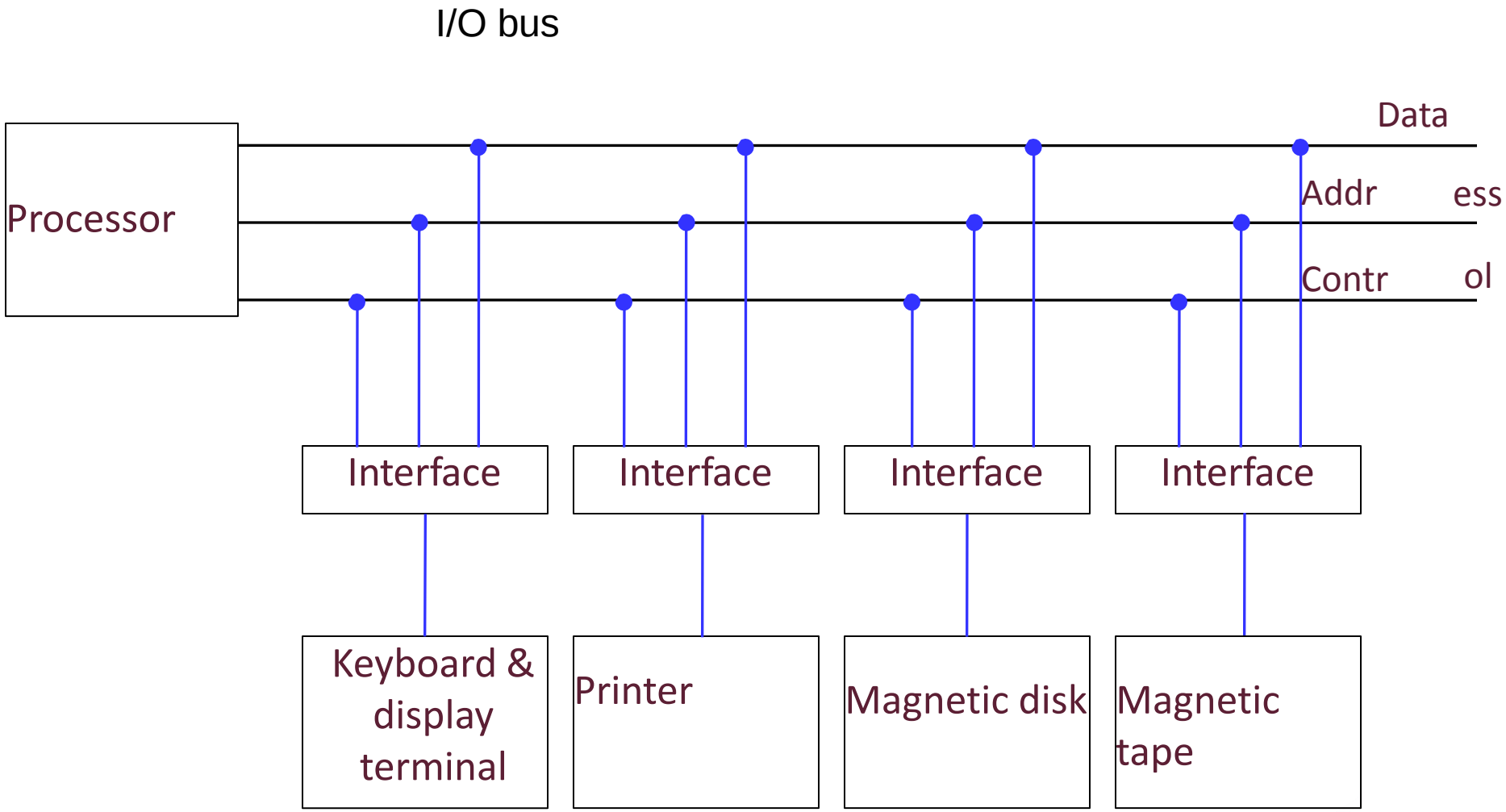


Figure: Connection of I/O bus to input-output devices

There are four types of commands that interface may receive They are:

1. **Control command**:
   It is issued to activate peripheral and to inform it what to do.

2. **Status command**:
   it is used to test various status conditions in interface and peripherals.

3. **Data output command**:
   it transfers data from processor to interface unit.

4. **Data input command**:
   it transfers data from interface unit to processor.

I/O versus Memory Bus

There are three ways that computer buses can be used to communicate with memory and I/O.

1. Use two separate buses, one for memory and other for I/O.

2. Use one common bus for both memory and I/O but have separate control lines for each.

3. Use one common bus for memory and I/O with common control lines.

Isolated versus Memory mapped I/O

Isolated I/O

- In isolated I/O configuration, CPU has distinct input and output instructions
- Each instruction is associated with the address of interface register.

Memory Mapped I/O

- In memory-mapped I/O organization there is no specific input or output instructions.
- There is common address space for both memory and I/O.
- Computers with memory mapped I/O can use memory type instructions to access I/O data.

Example of I/O interface

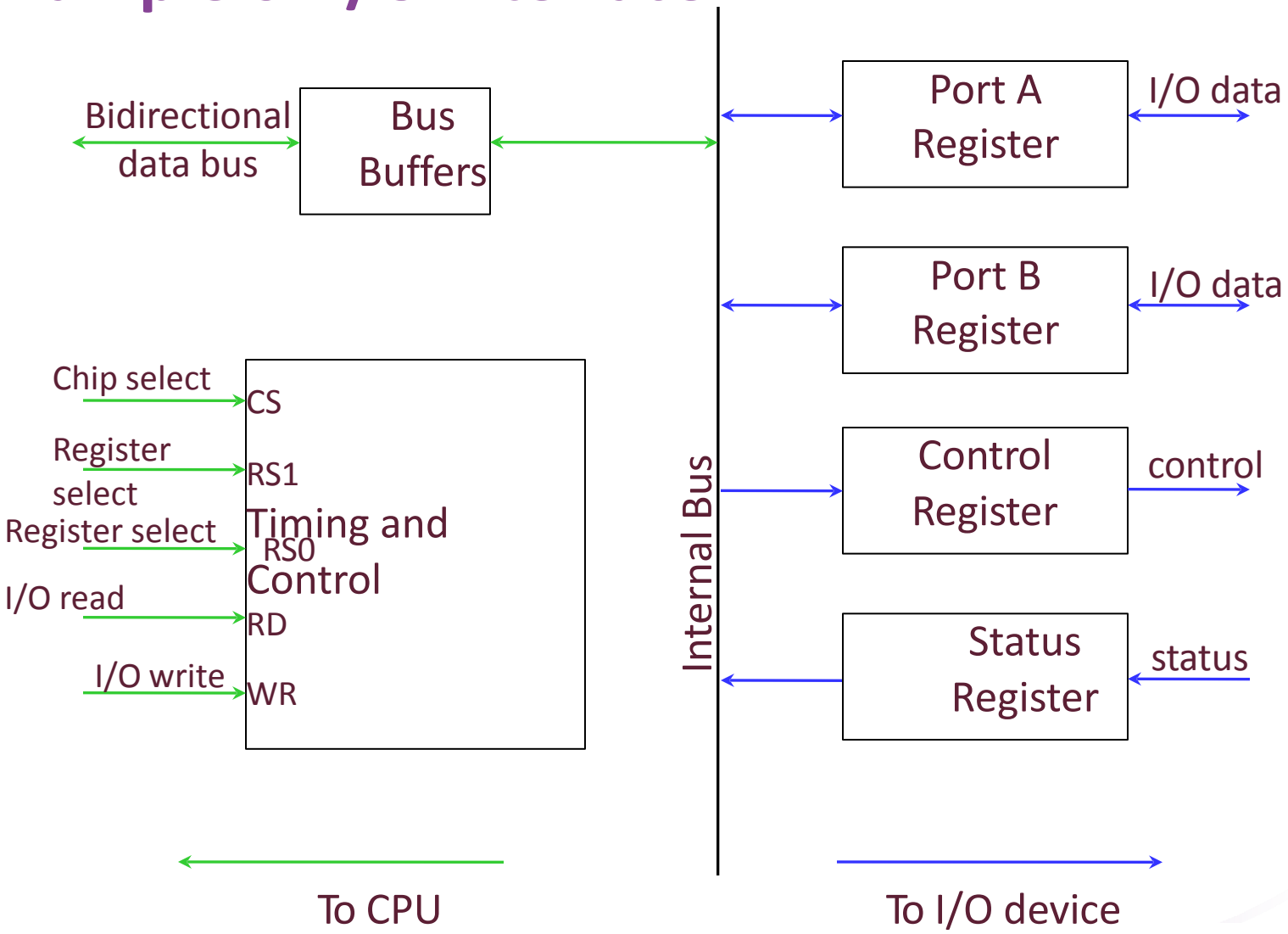


Figure: Example of I/O interface unit

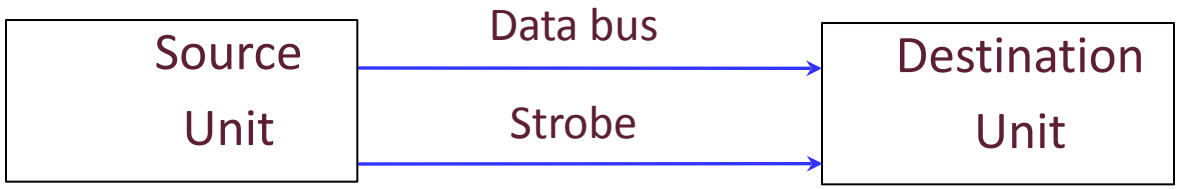
CS	RS1	RS0	Register select
0	X	X	None: data bus in high impedance
1	0	0	Port A register
1	0	1	Port B register
1	1	0	Control register
1	1	1	Status register

6.2 Asynchronous Data Transfer

Ref. Book Name : Computer System Architecture, M. Morris Mano

- Internal operations in a digital system are synchronized by means of clock pulses supplied by common pulse generator.
- Two units, CPU and I/O interface, are designed independently of each other.
- If interface and CPU share a common clock, then transfer is said to be synchronous.
- In most cases, internal timing in each unit is independent from each other that each uses its own private clock, the two units are said to be asynchronous.

6.2.1 Strobe control

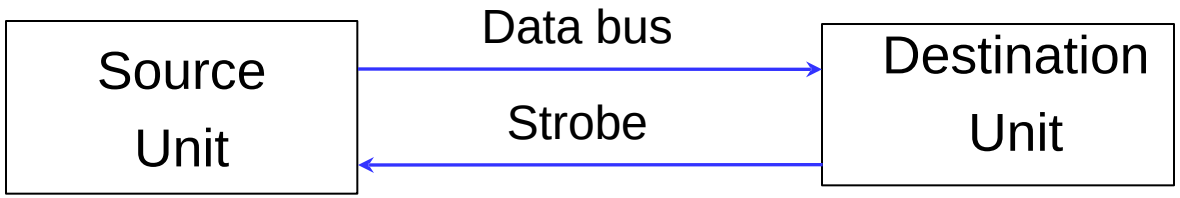


(a) Block diagram

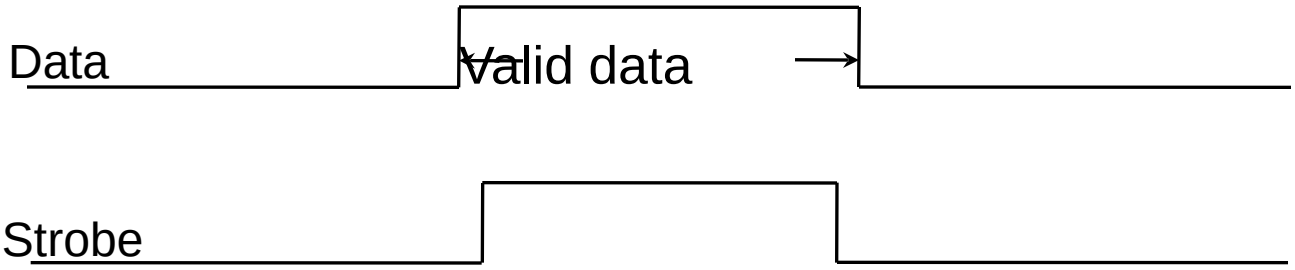
(b) Timing diagram

Figure: Source initiated strobe for data transfer

6.2.1 Strobe control



(a) Block diagram

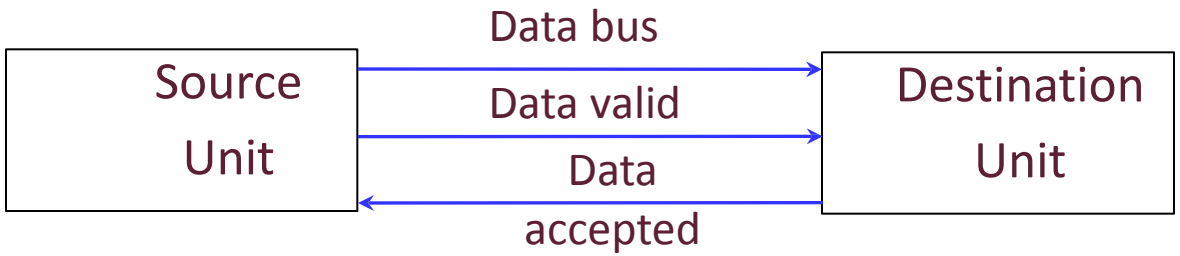


(b) Timing diagram

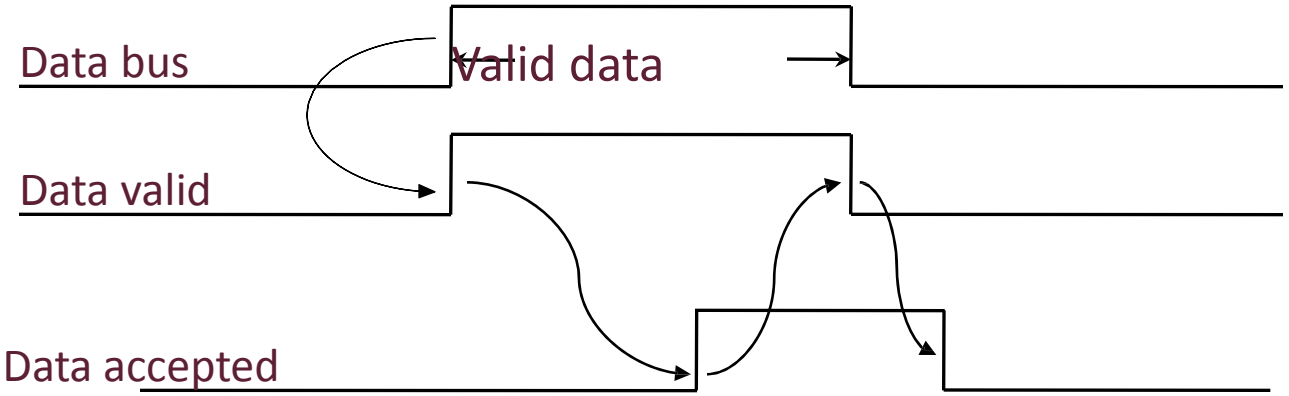
Figure: Destination initiated strobe for data transfer

- **Disadvantage**: source unit has no way of knowing whether destination unit has received data item placed in the bus or not.

6.2.2 Handshaking



(a) Block diagram



(b) Timing diagram

Figure: Source initiated transfer using handshaking

6.2.2 Handshaking

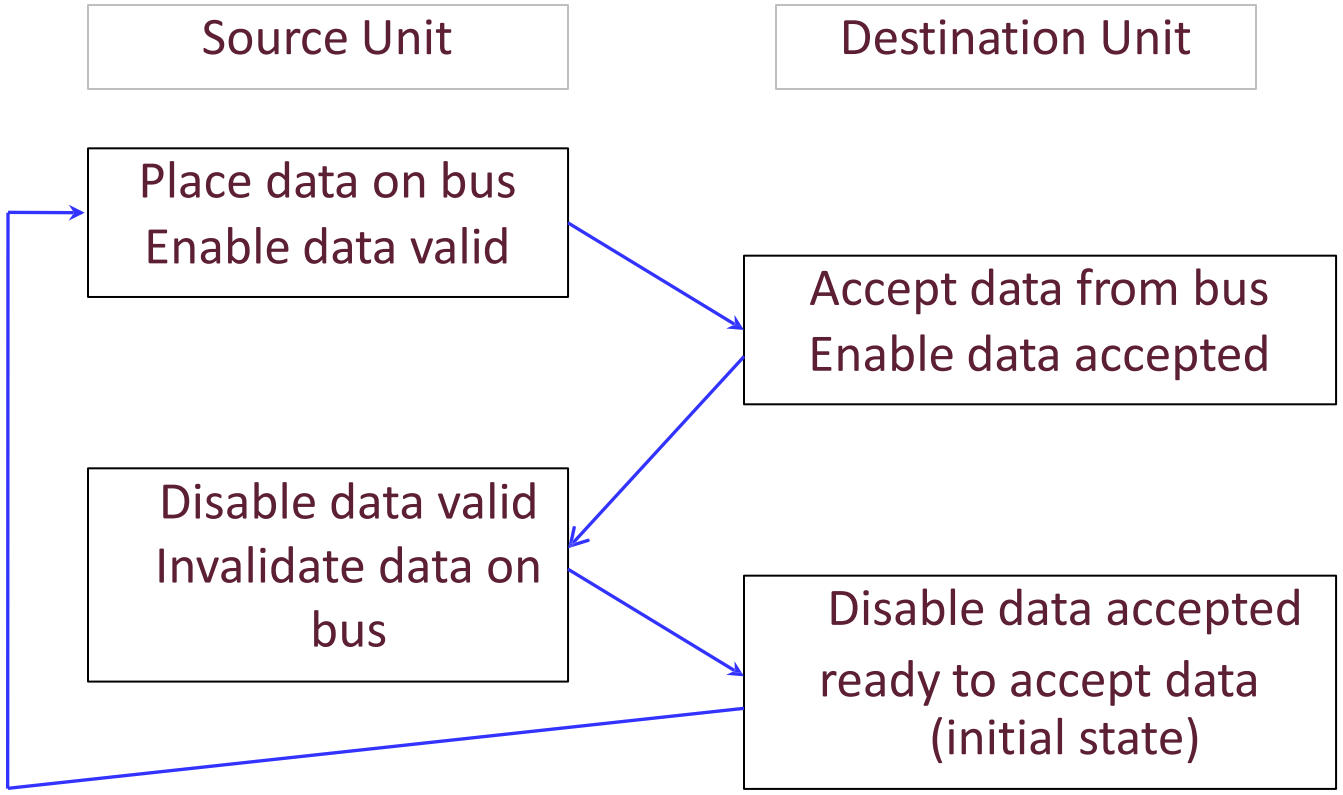
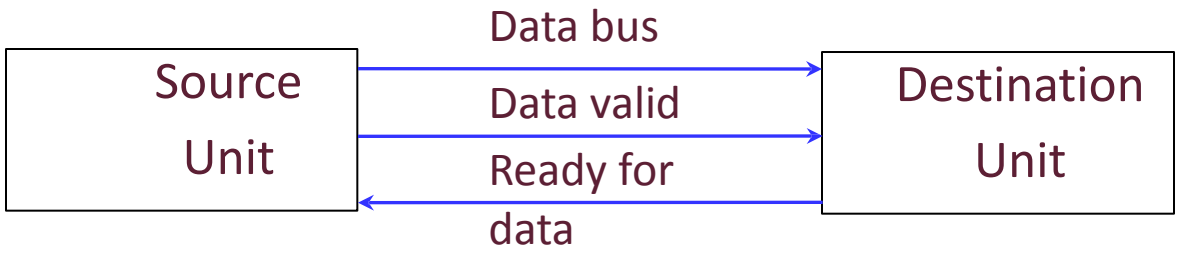
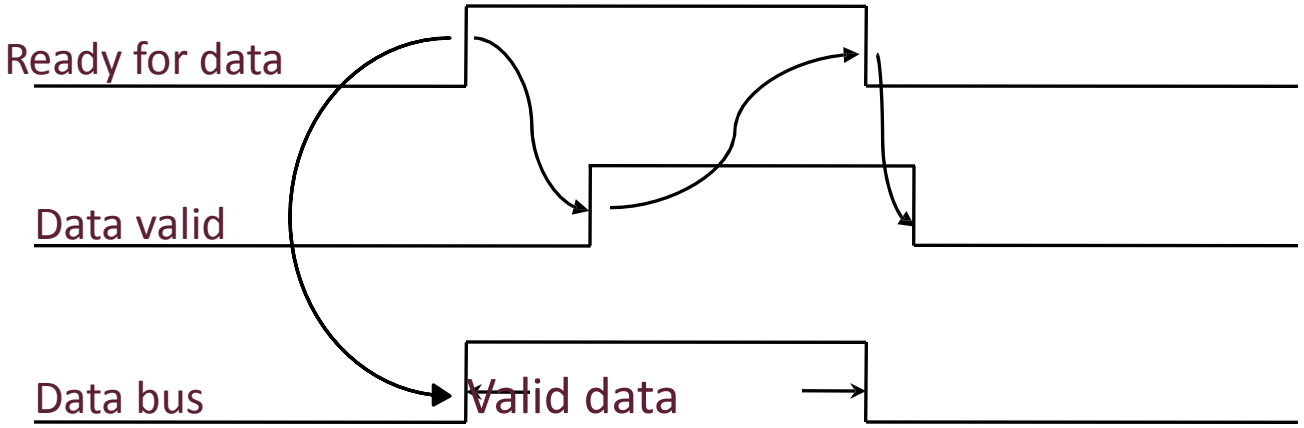


Figure: Sequence of events

6.2.2 Handshaking



(a) Block diagram



(b) Timing diagram

6.2.2 Handshaking

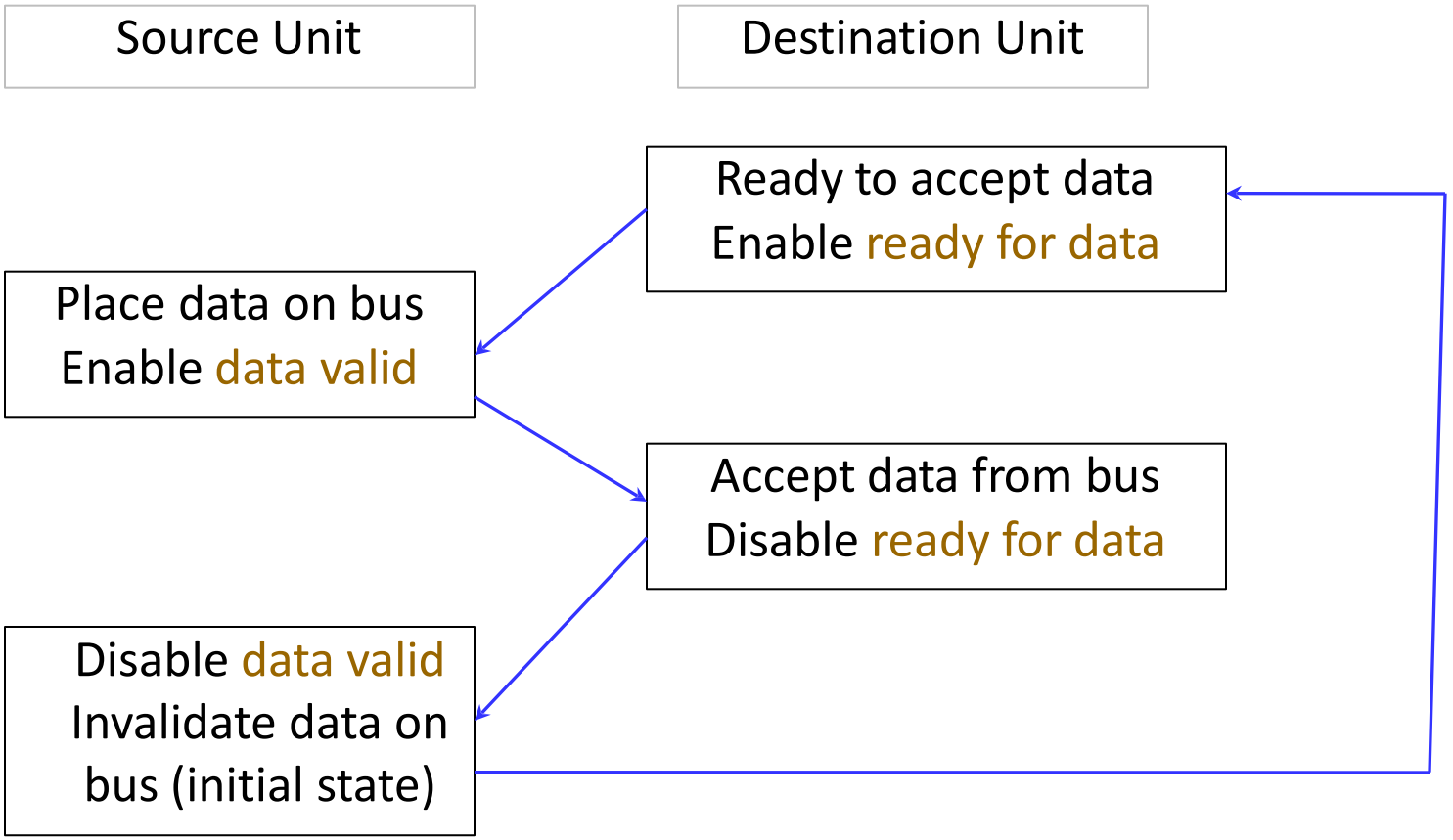
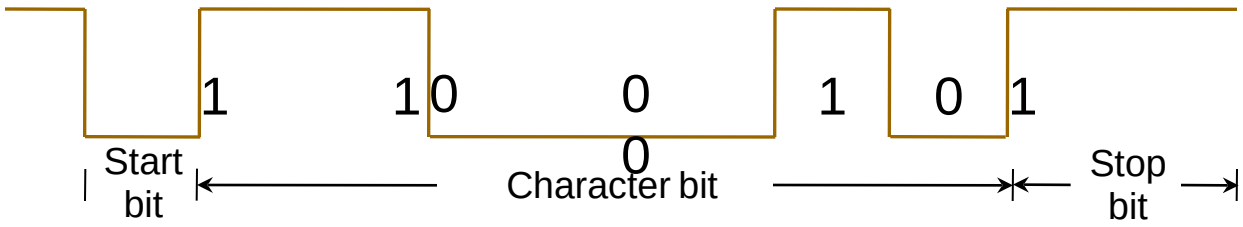


Figure: Sequence of events

6.2.3 Asynchronous Serial Transfer

- In asynchronous serial transfer, each character consists of three parts:
  - Start bit
  - Character bit
  - Stop bits

- For example:



6.2.3 Asynchronous Serial Transfer

- How receiver can detect a character?

1. When a character is not being sent , line is kept in state 1.
2. Initiation of character transmission is detected from start bit 0.
3. Character bits always follow start bit.
4. After last bit of the character is transmitted , stop bit is detected when the line returns to 1-state for one bit time.

6.3 Modes of Transfer

Ref. Book Name : Computer System Architecture, M. Morris Mano

- Data transfer to and from peripherals may be handled in one of three possible ways:

1. **Programmed I/O:**

- Programmed I/O instructions are result of I/O instructions written in computer program.

- Transfer is initiated by instruction.

- I/O device does not have direct access to memory.

6.3 Modes of Transfer

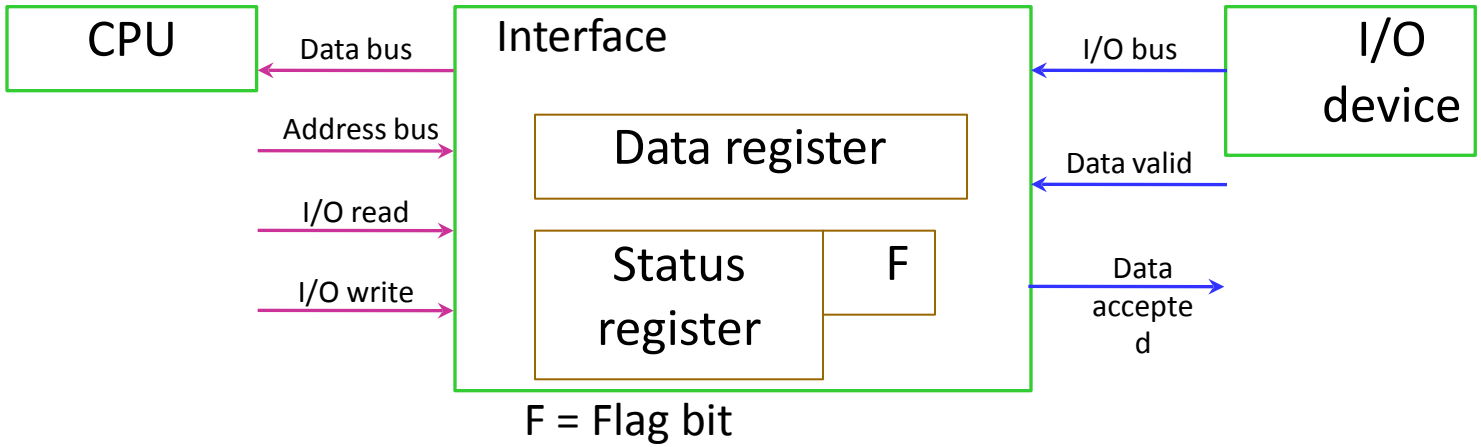
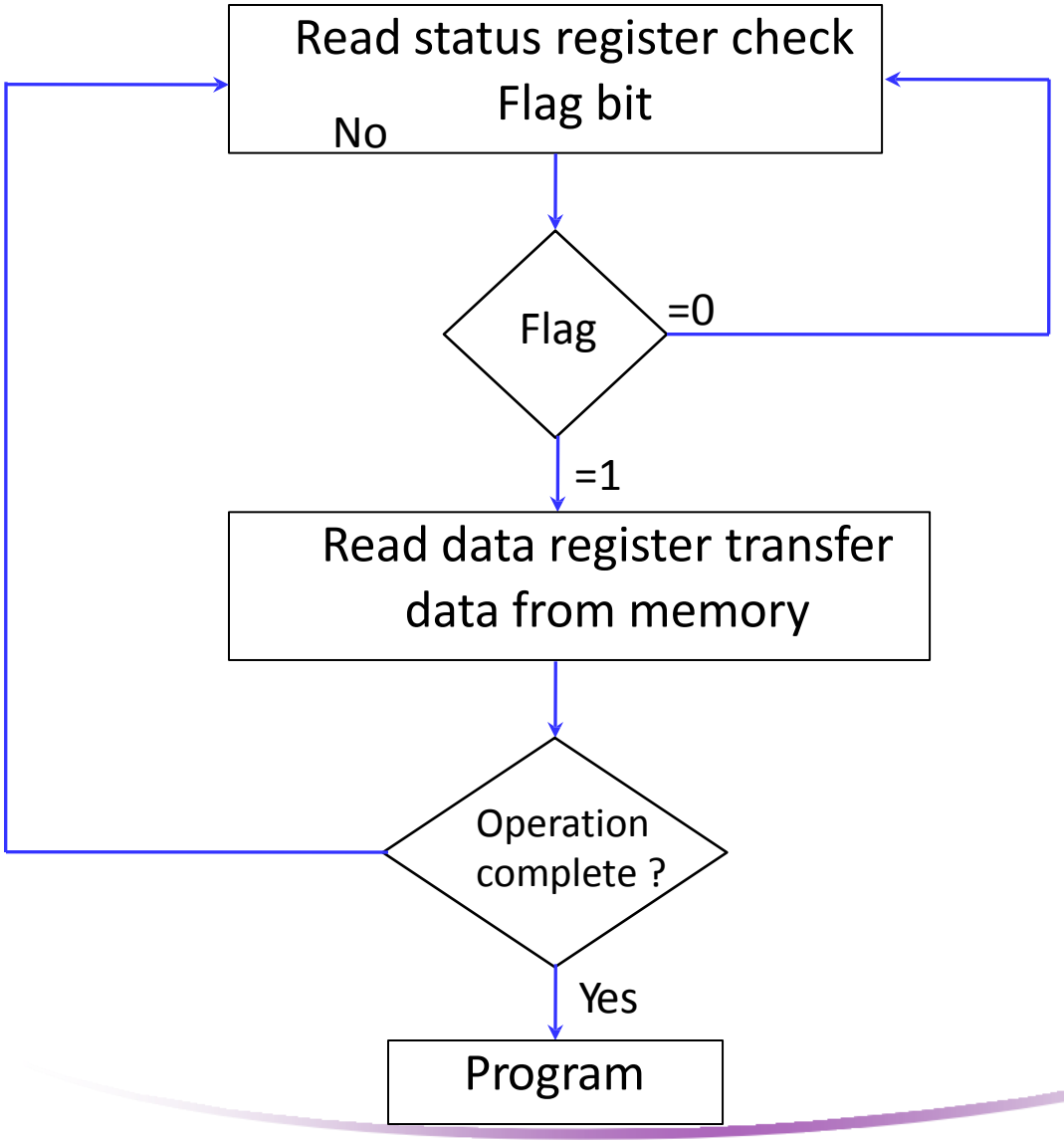


Figure: data transfer from I/O device to CPU

Flowchart of the program that must be written for CPU is shown below:



2. Interrupt Initiated I/O:

- When interface determines that device is ready for data transfer, it generates interrupt request to computer.

- Upon detecting external interrupt signal
  - CPU momentarily stops the task it is processing
  - Branches to a service routine to process I/O transfer
  - And then returns to task it was originally performing

- There are two methods to choose branch address:

1. Non vectored interrupt:

- Branch address is assigned to a fixed location in memory.

2. Vectored interrupt:

- The source that interrupts supplies branch information to the
- computer. This information is called interrupt vector.

3. Direct Memory Access

- In DMA, interface transfers data in and out of memory unit through memory bus.
- When transfer is made, DMA requests memory through memory bus.
- When request is granted by DMA controller, DMA transfers data directly into memory.
- **DMA controller:** it is an interface that provides I/O transfer of data directly to and from memory and I/O device.
- CPU initializes DMA controller by sending memory address and no. of words to be transferred.
- Actual transfer of data is done directly between device and memory through DMA controller.

6.4 Input-Output Processor (IOP)

Ref. Book Name : Computer System Architecture, M. Morris Mano

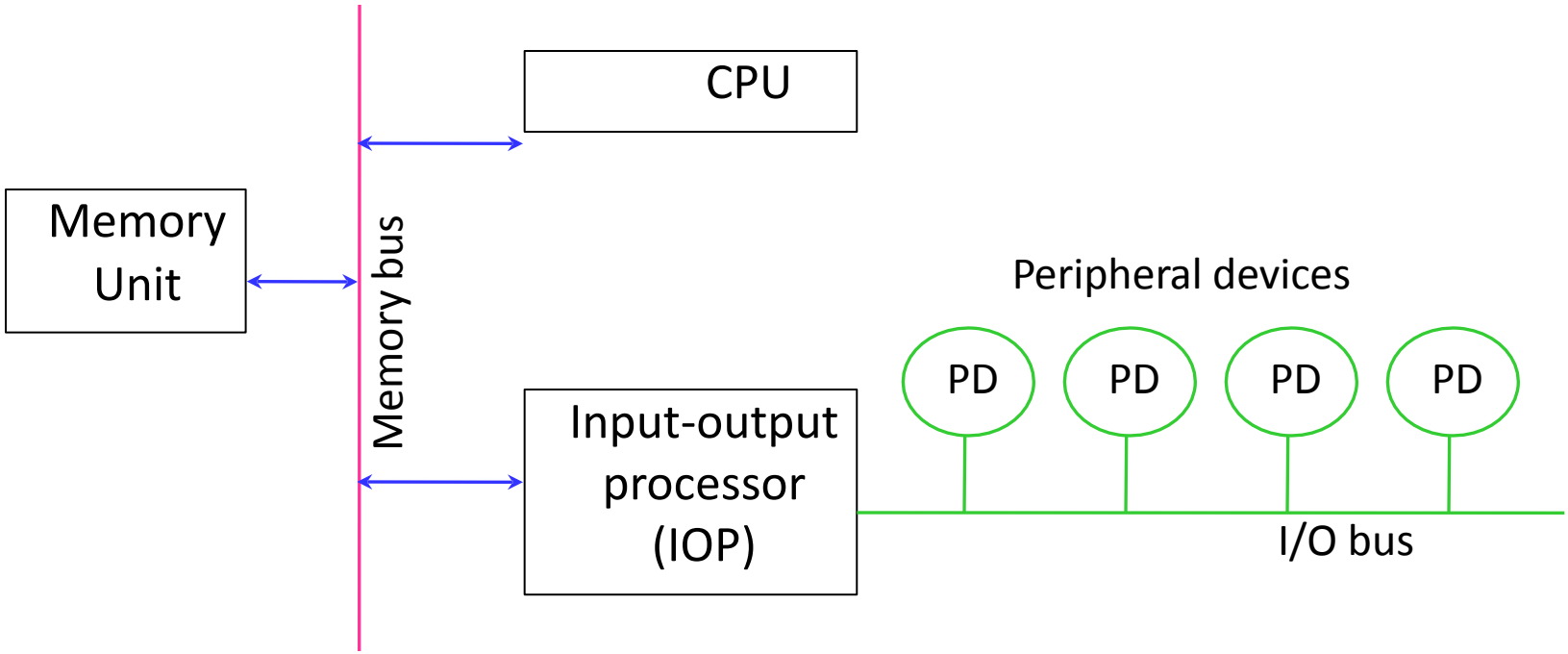


Figure: Block diagram of a computer with I/O processor

CPU-IOP Communication