

Bus Structure

Single bus
Structure

Multiple bus
Structure

- A bus is a group of lines that serves as a connecting path for several devices
- The lines carry → data, address, or control signals

1) Single bus Structure:

- Only one transfer at a time
- Only 2 units can actively use bus (@ given time)
- Bus Control line are used to arbitrate multiple request for use of bus
 - Low Cost
 - Flexibility for attaching peripheral devices

2) Multiple bus Structure

- This system contains multiple buses achieve more concurrency in operation
- Two or more transfer can be carried out @ same time
 - Better performance
 - Increased cost
- devices connected to a bus vary widely in their

Operation Speed



In order to synchronize → Operation Speed

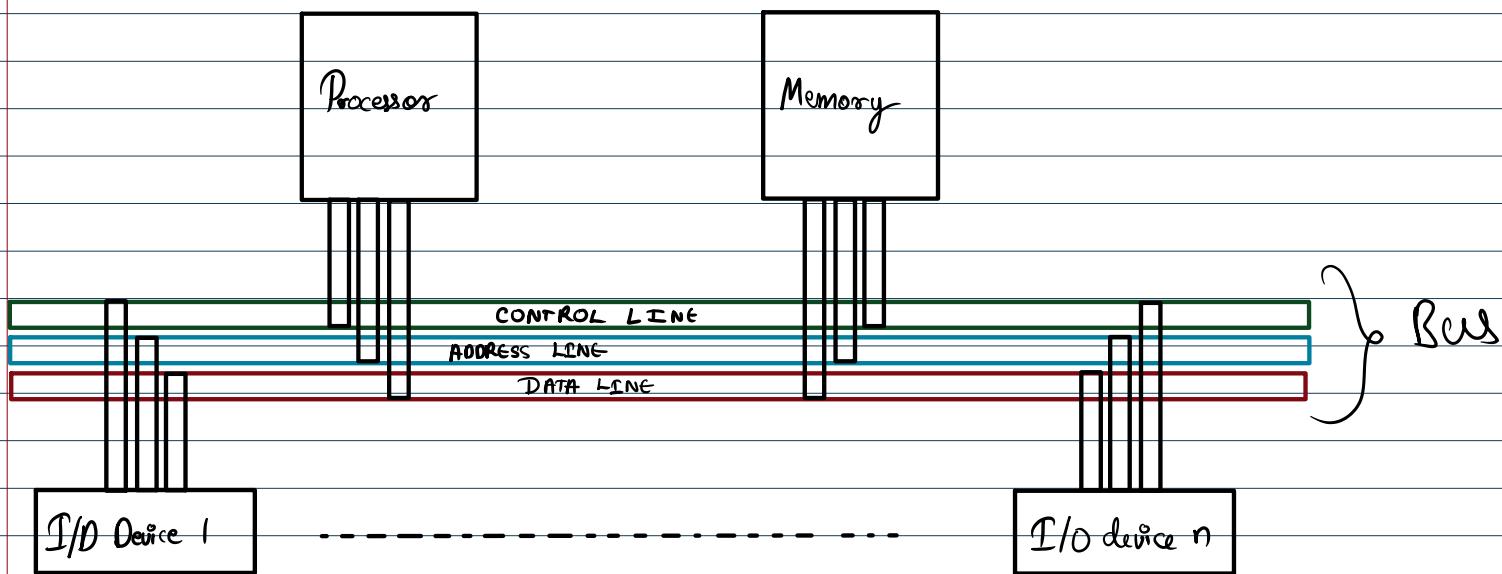


Use of Buffer Registers

① Buffer Register

→ These hold information during transfers

→ Prevents a high speed processor from being locked to slow I/O device during transfer



Bus Structure

① Data Lines

- Coordinated transfer of data
- No. of lines present in database defines **Bus width**
- Each data line transfers only one bit at a time

*The no. of data lines determine bits transferred/sec
performance of system depends on this*

② Address Lines

- Contents of address line of bus determines the source & destination of data present on database

Same points as data bus / width

- Width of address bus determines Memory capacity of the system

- Contents of address lines are used for I/O addressing

Higher order bits ; determine bus module

Lower order bits ; determines address of memory location & I/O port

③ Control Lines

- Address Line & Data Lines are shared by all Components of System

thus need of Control Lines

- These Control are to access of data & address lines

Control Signal → ① Command
② Timing Information

Command ; Specifies the operation that has to be performed

Timing Info ; Specifies till when data & address info. is valid

Control Line include one for ;

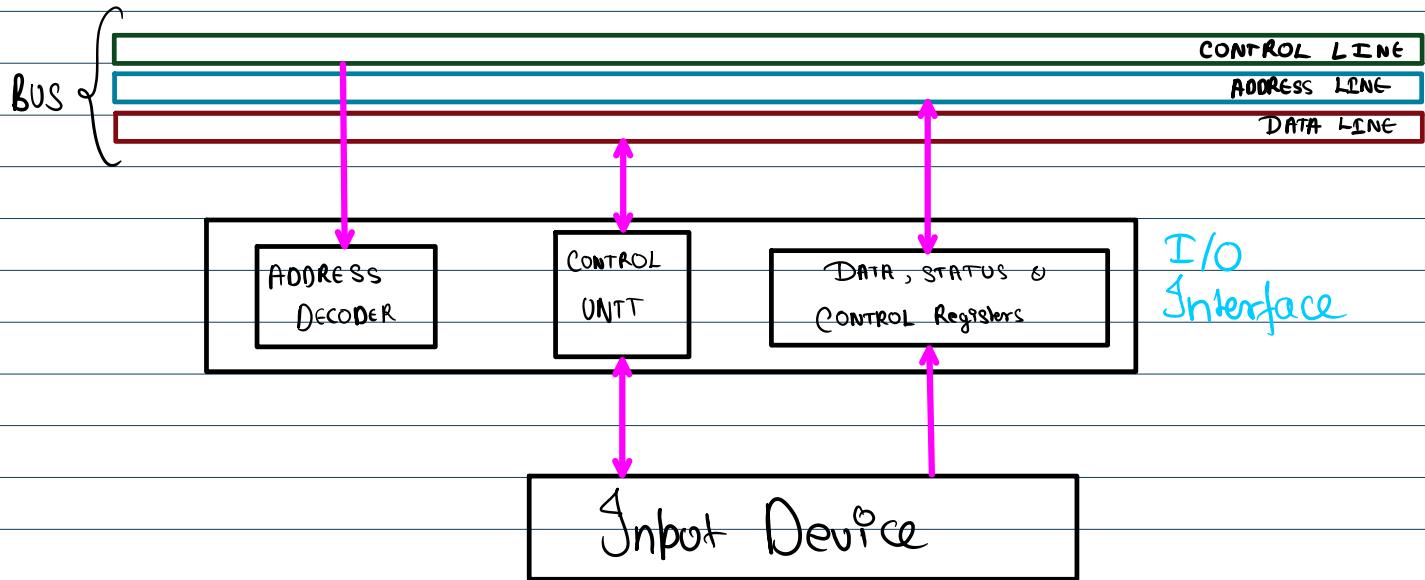
- Memory Read/write, I/O Read/Write
- Transfer ACK ; Transfer acknowledge
- Bus Request/Grant
- Interrupt Request / ACK
- Clock, Reset

"During transfer of data b/w two Components one acts as Master & other as Slave"

Device Initiating Request ; Master
other is ; Slave

- Timing*
- Synchronous*
Asynchronous
- High Speed transmission (@ const rate)
 - Synchronized by clock (bus clock)
alternating sequence of 1's & 0's
 - All device connected to bus can read the bus clock cycle events
 - Transmitting & receiving components are synchronized using clock
 - Can take more time
 - Not Synchronized by the clock
 - Data transfer controlled using "Hand Shake protocol"
 - Handshake protocol b/w Master & Slave
 - Data transfer is initialized by Master Component by activating Master ready-line & place address & Command info over bus
 - All connected components decode the address on address line to recognise addressed component
 - Now Addressed component notified the processor by activating slave-ready line

I/O Interface of Input device Connected to Bus



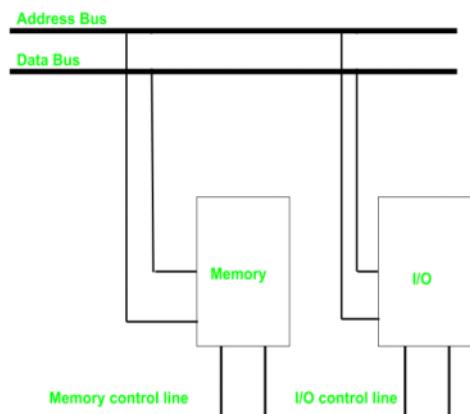
- o processor puts an address in address line if it is examined by all device on bus
- o whichever device recognizes to the address, responds to the Control operation on Control line
- o processor issues Read/write operation over Control line
 - o data corresponding to R/W is transmitted to data line
- o Control Register & Status Register of I/O device have info relevant to operation performed on/by the I/O device

Mapped I/O

Processor needs to communicate to various memory & I/O devices data flow;

Isolated I/O

- Common bus (data & address) for memory & I/O
- Separate Read write Control lines
- Address Space for memory & I/O is Isolated



- I/O addresses are called ports
- Complex logic

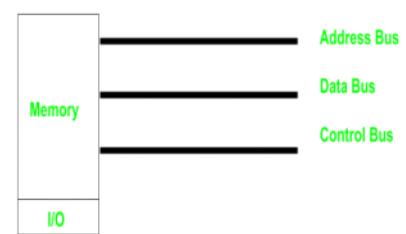
Memory-Mapped I/O

- In this case every bus is common



Same set of instruction work for I/O & Memory

- I/O & Memory have same address space



- Addressing capability of Memory becomes less, as some part is occupied by I/O

- Smaller size
- Less efficient

- Simple logic, I/O treated as memory

Modes of Transfer

Programmed I/O

Interrupt -
Initiated
I/O

Direct
Memory
Access

Programmed I/O ;

Each data transfer initiated by an instruction in the program

CPU stays in program loop until I/O unit indicates it is ready for data transfer

- Very time consuming process
(keeps processor busy)

Interrupt Initiated I/O ;

This doesn't check flag 0
Continues to perform task

Whenever device wants attention it sends
an interrupt signal

Then the CPU addresses the device

Direct Memory Access

This directly communicates with
the memory or peripheral devices

This involves peripheral devices to
read/write data without CPU intervention

