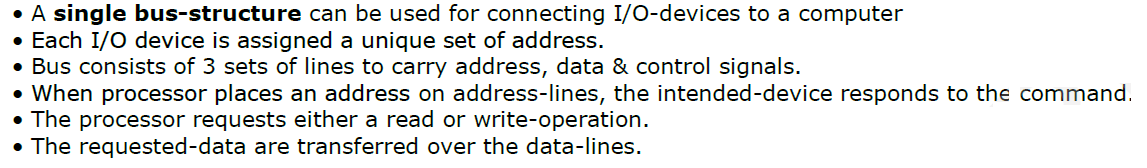
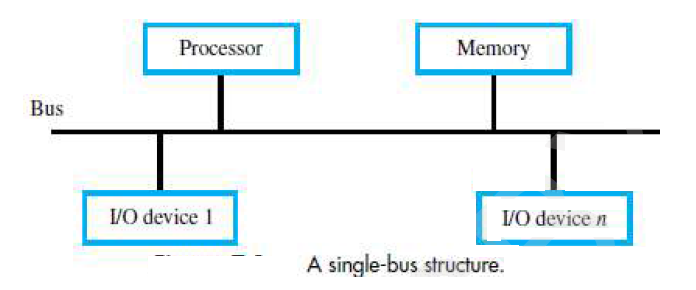
**INPUT / OUTPUT ORGANIZATION**

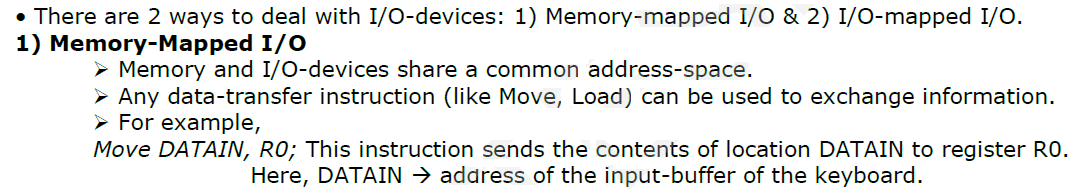
UNIT-II

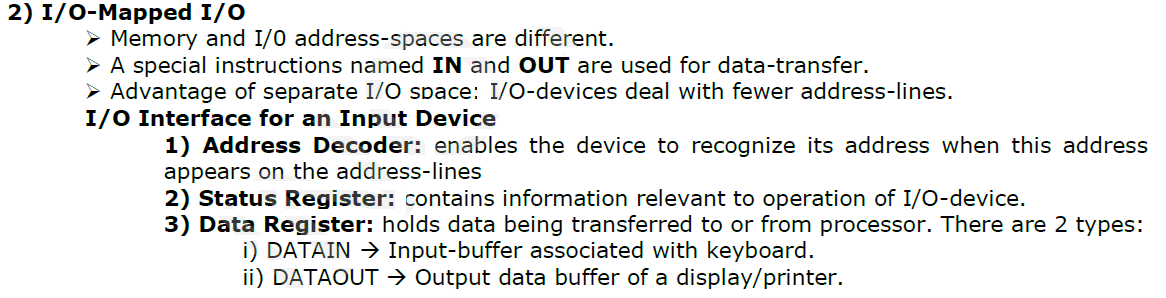
**Input/output Organization:** Accessing I/O Devices, Interrupts, Processor Examples, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces.

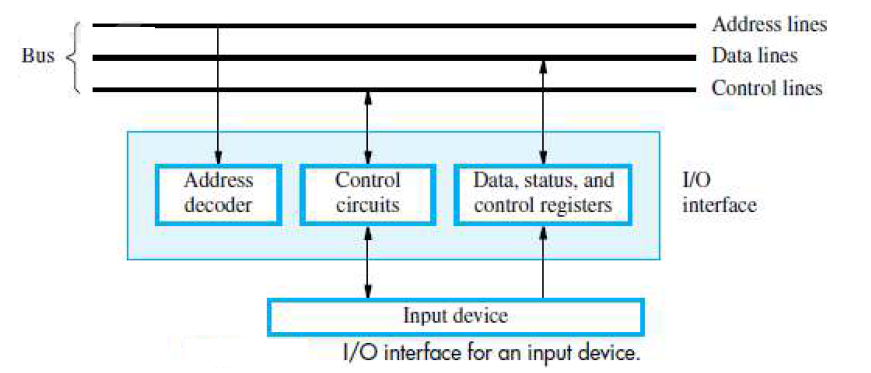
**ACCESSING I/O DEVICES:**



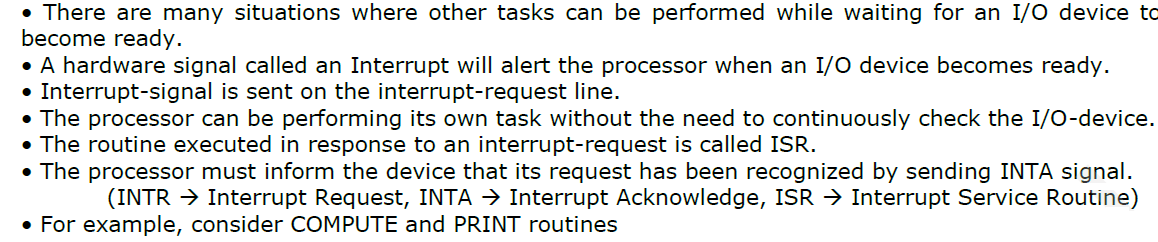


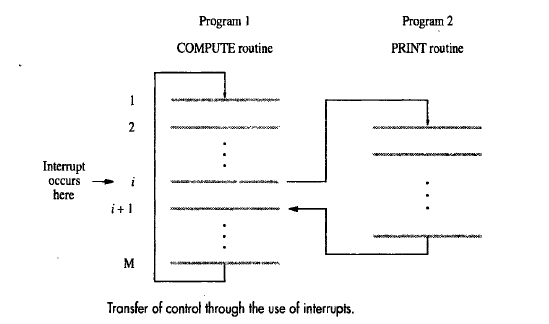


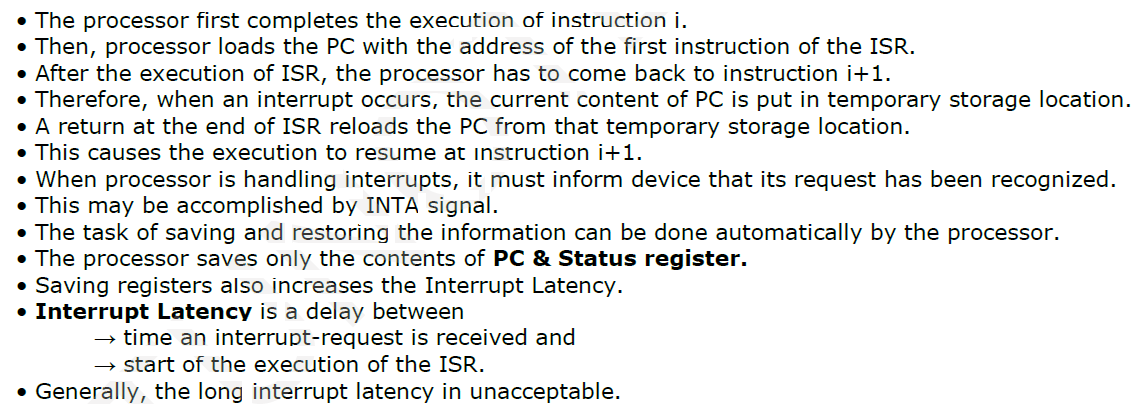


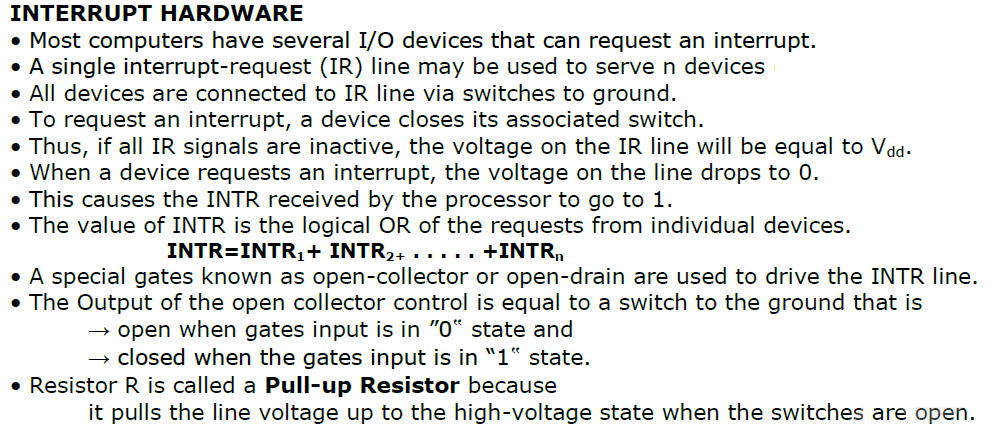


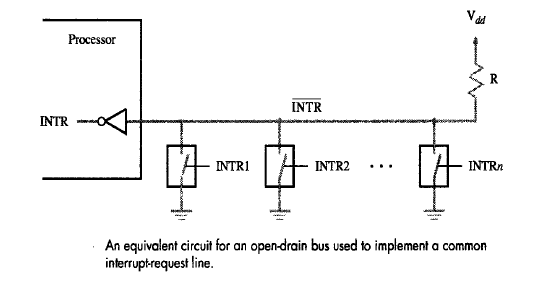
**INTERRUPTS:**

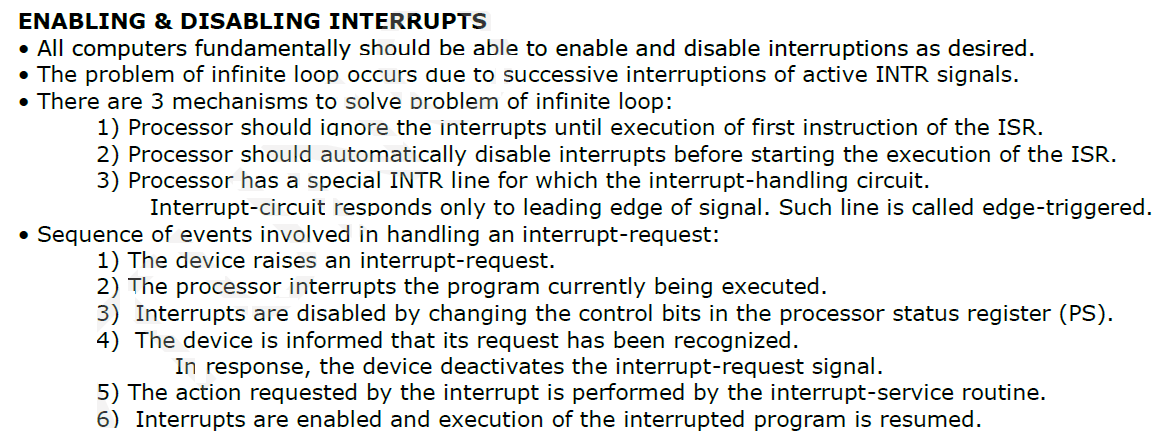
****

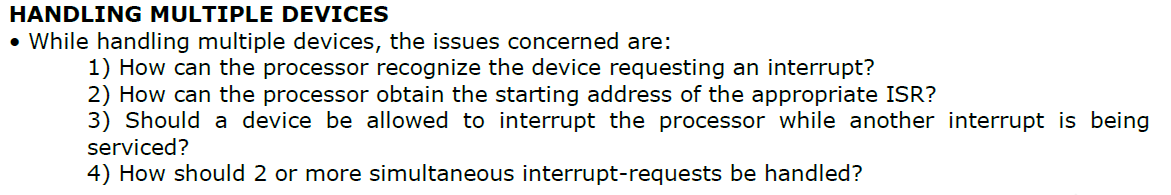
****

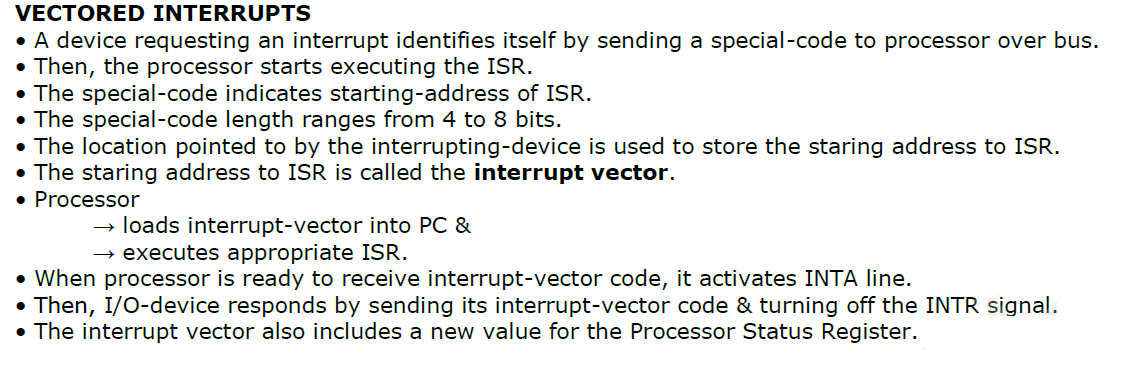
****

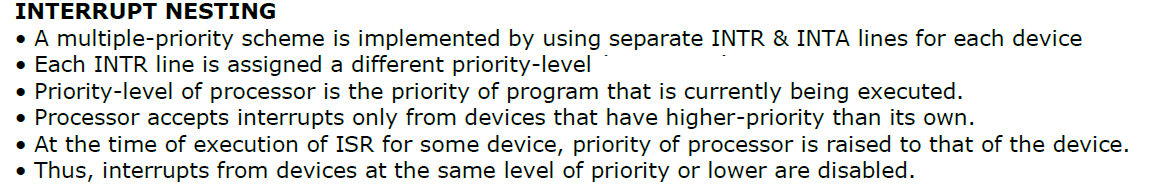


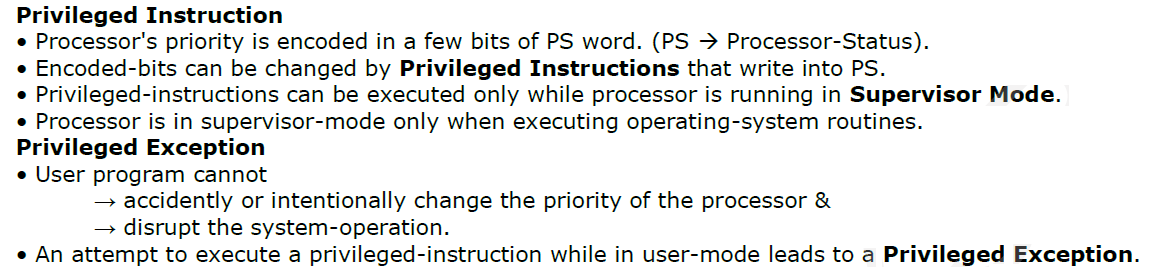


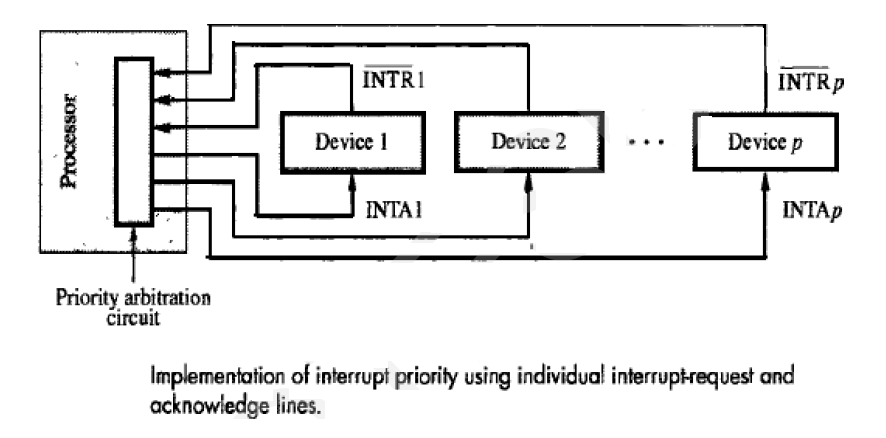


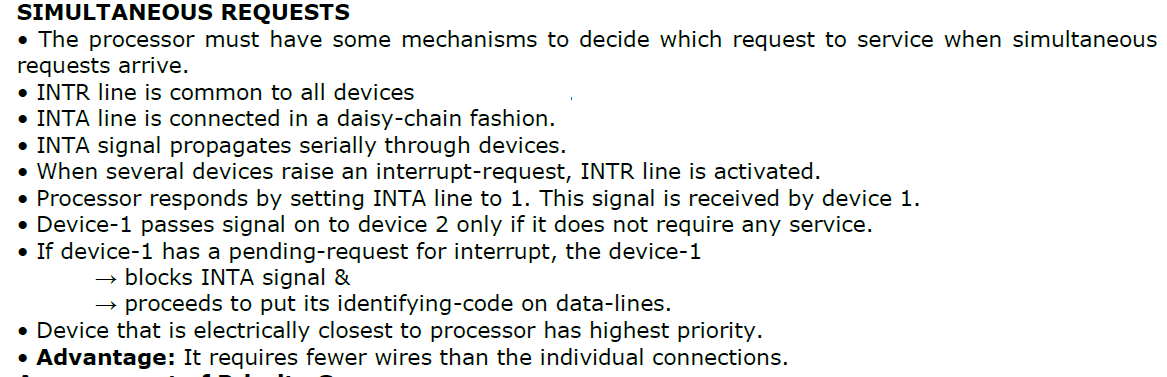


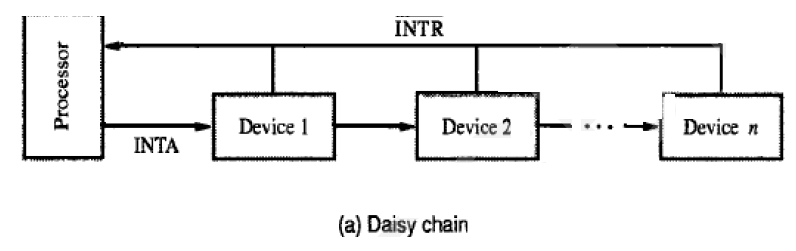


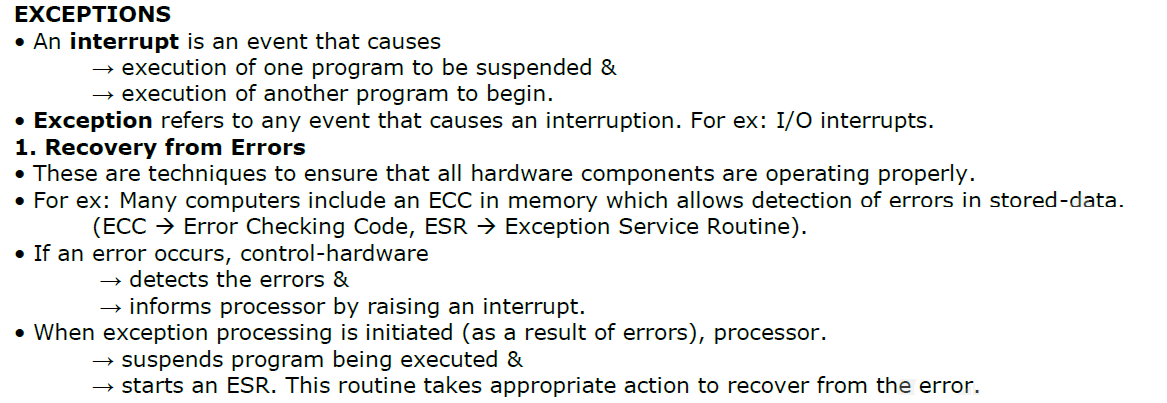


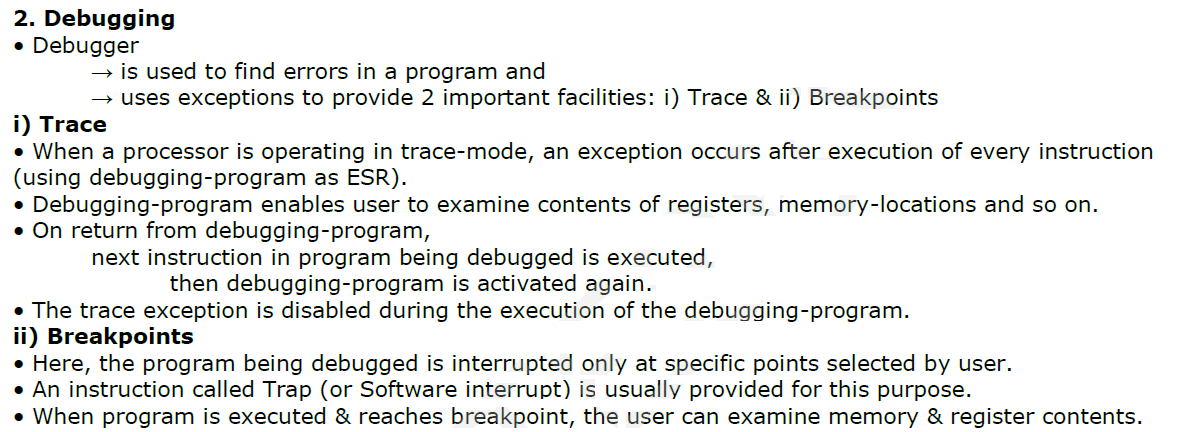


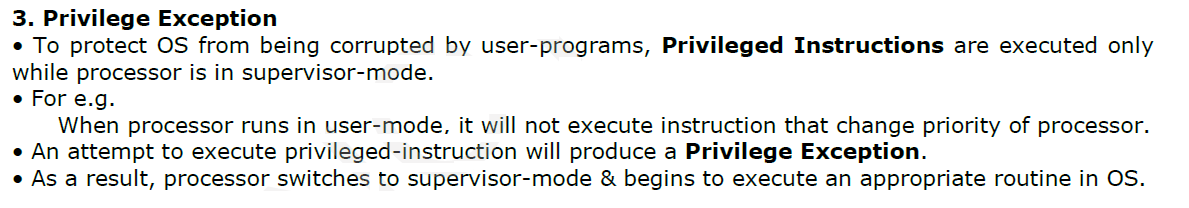




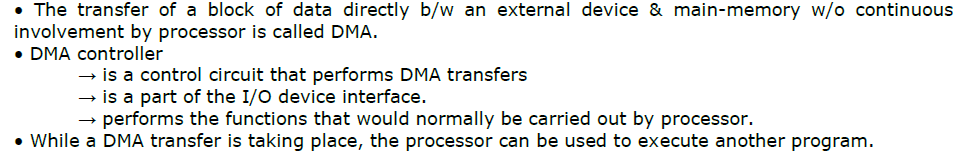


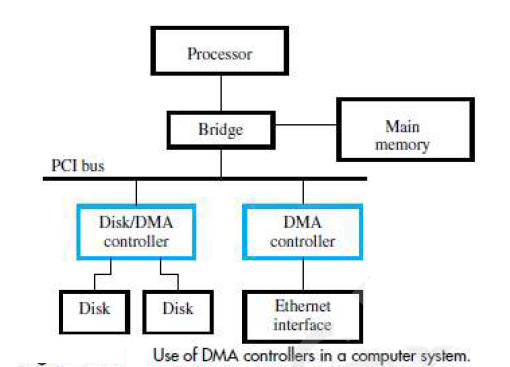


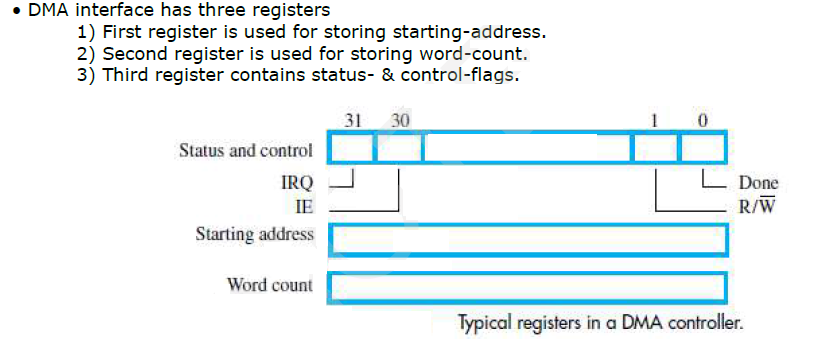


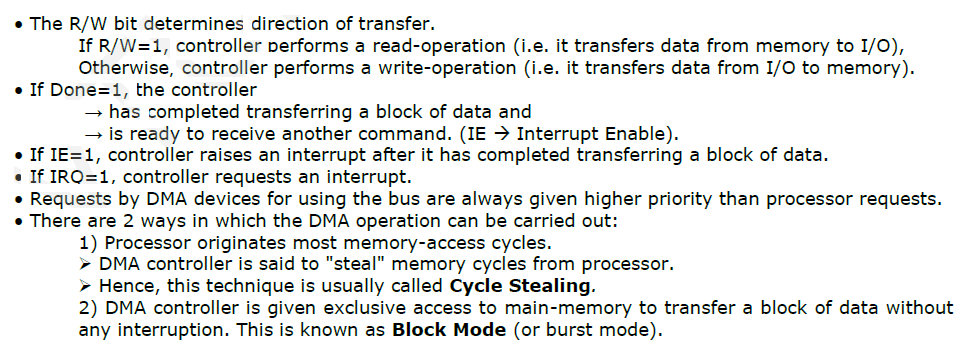


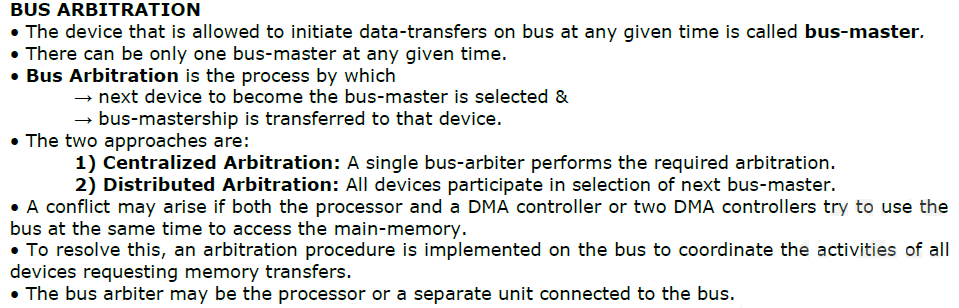
**DIRECT MEMORY ACCESS (DMA):**

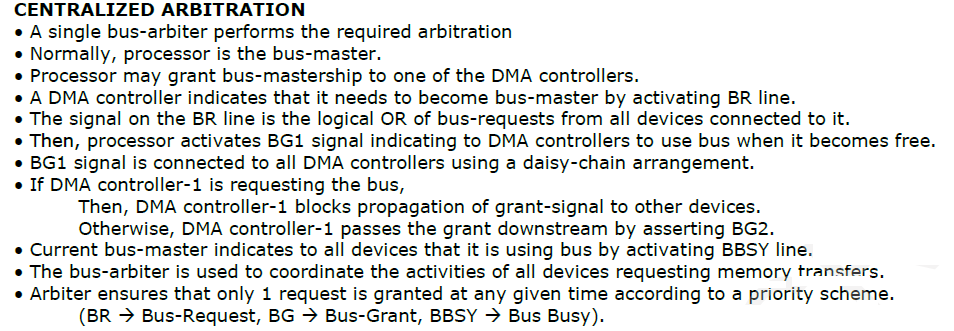
****

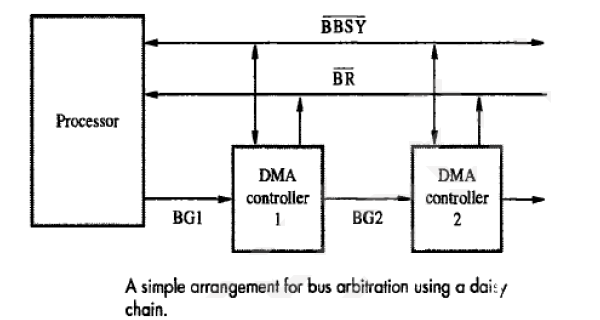
****

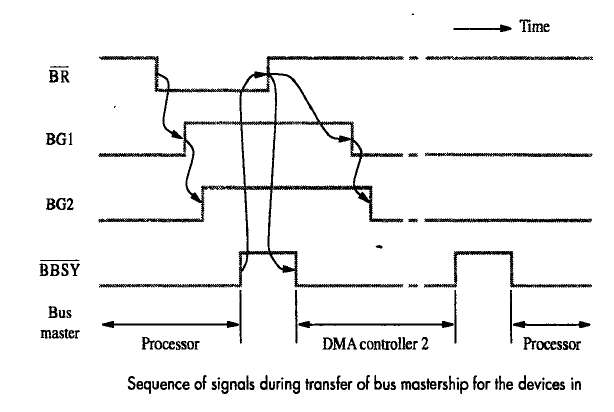
****

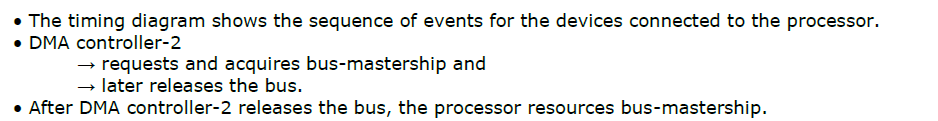
****

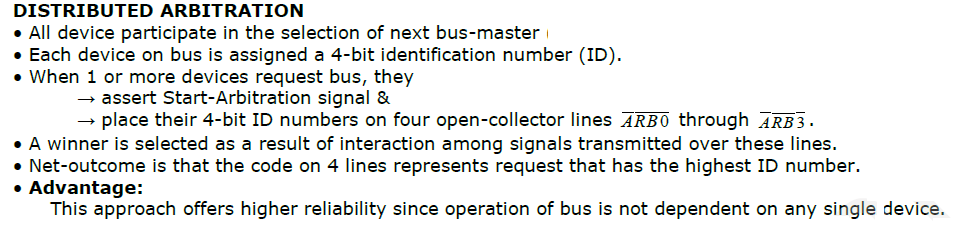
****

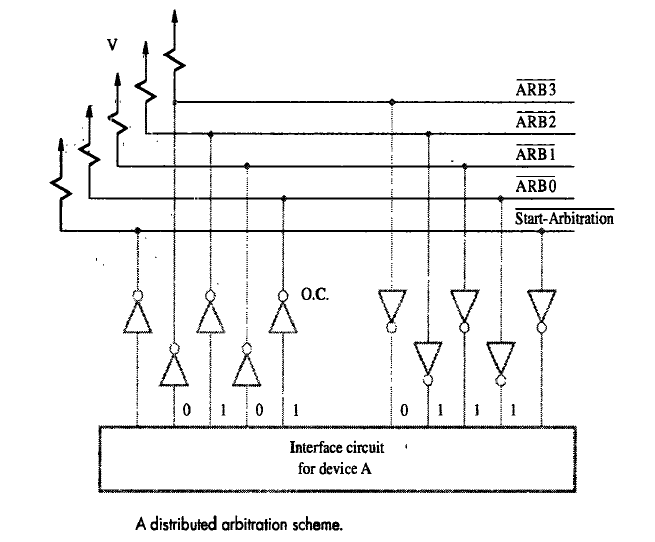
****

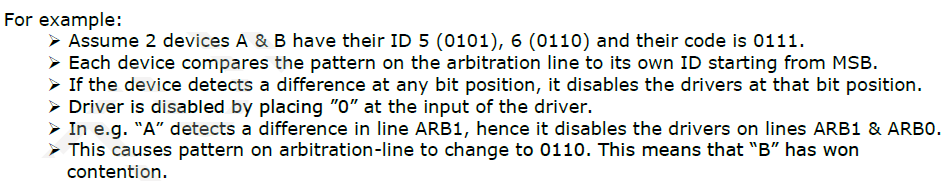
****

****

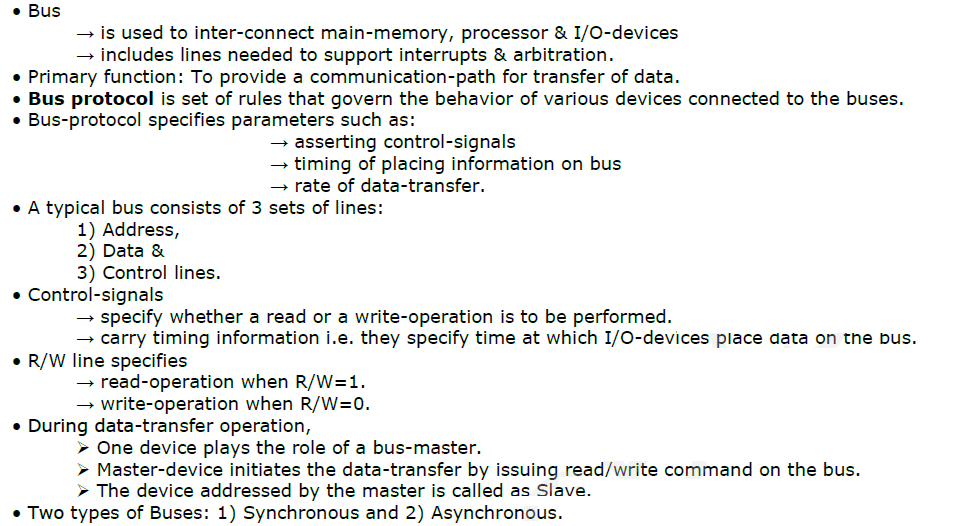


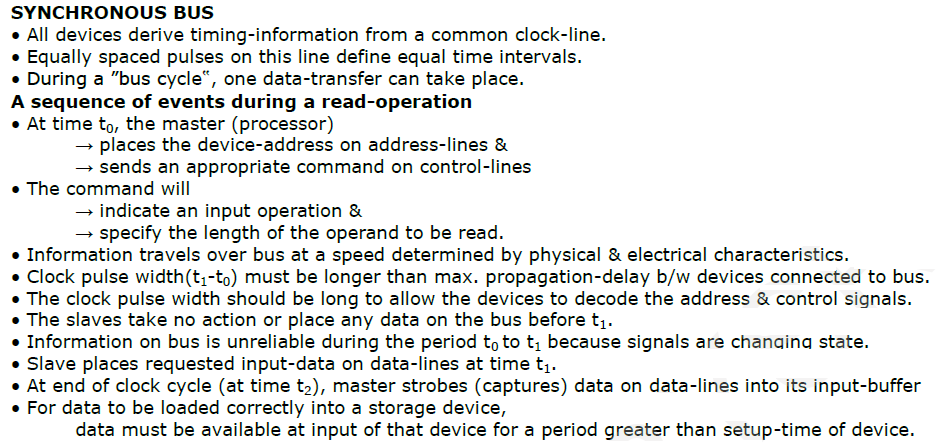


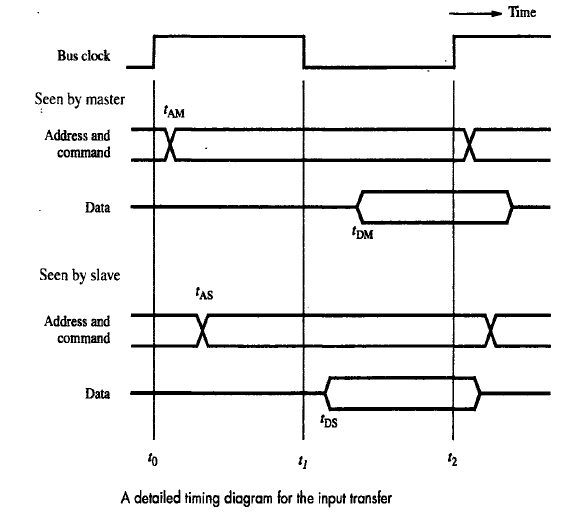


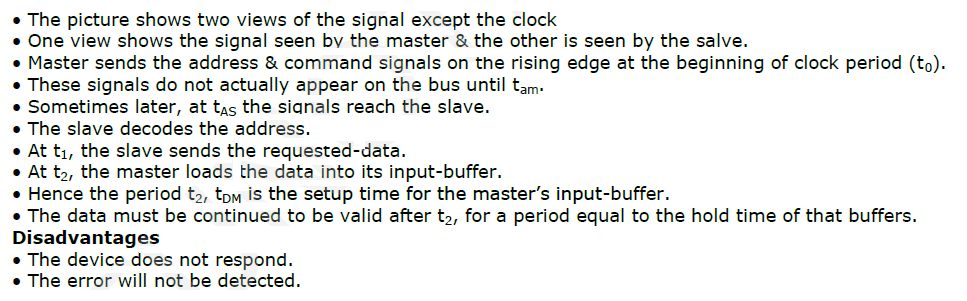


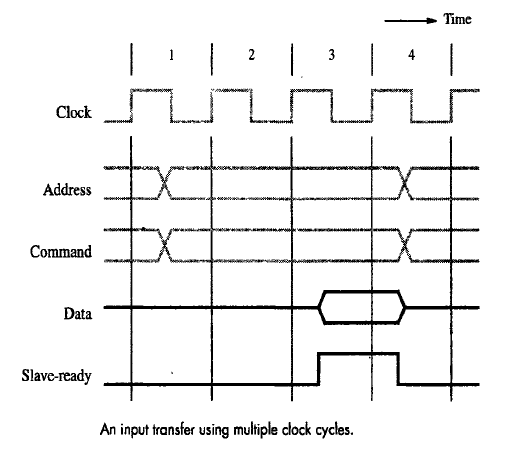
**BUS:**

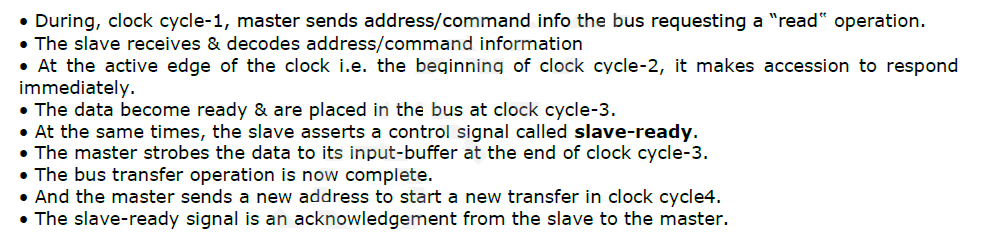
****

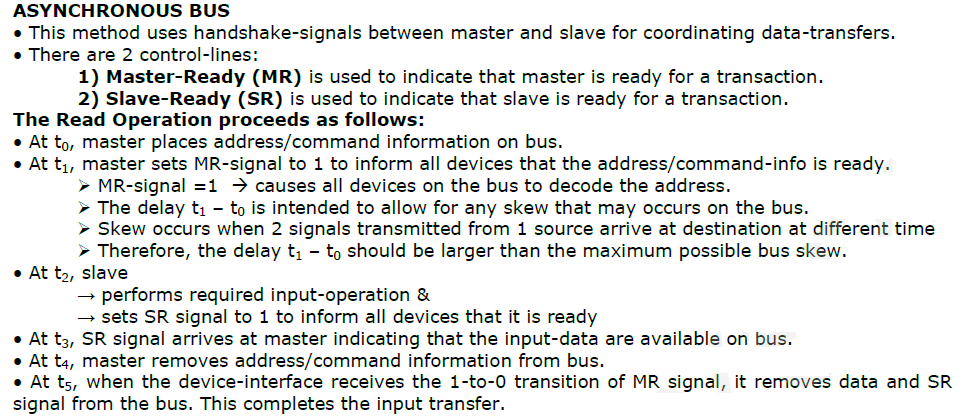


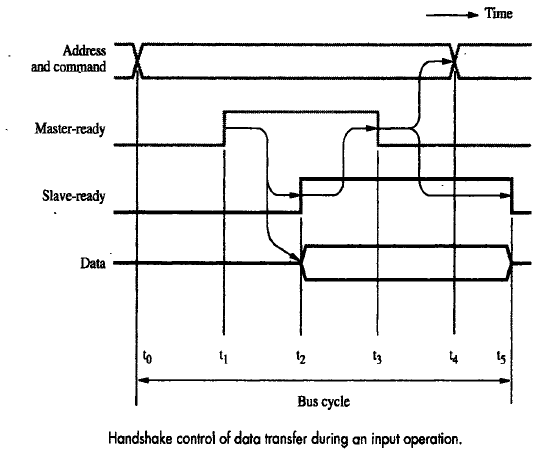




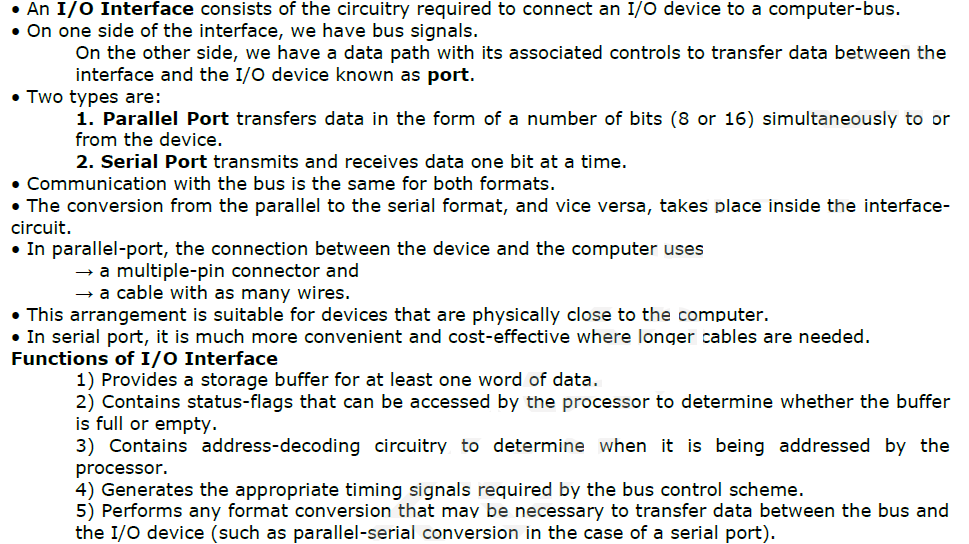


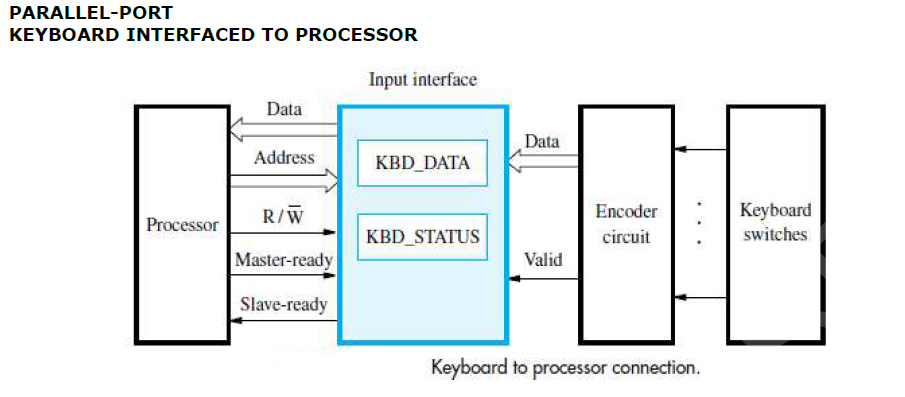


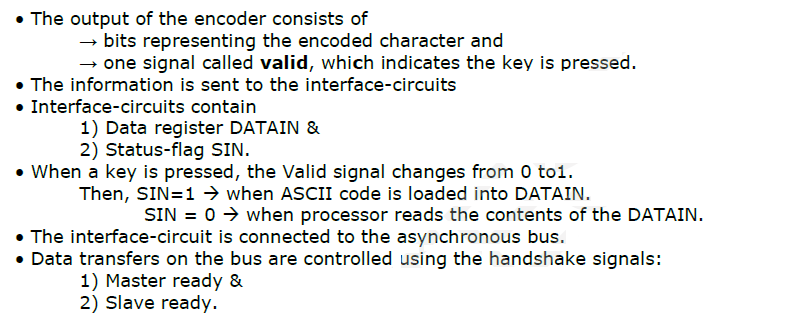




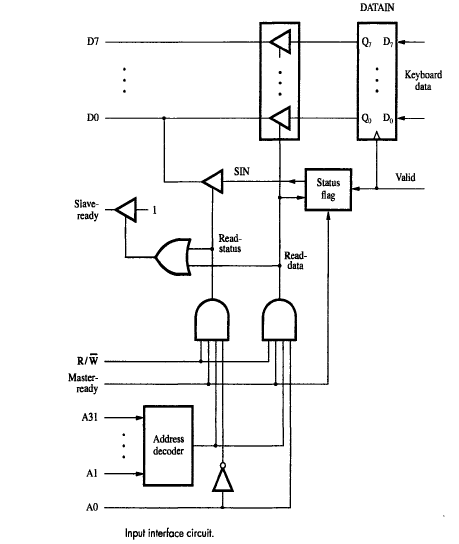
**INTERFACE- CIRCUITS:**

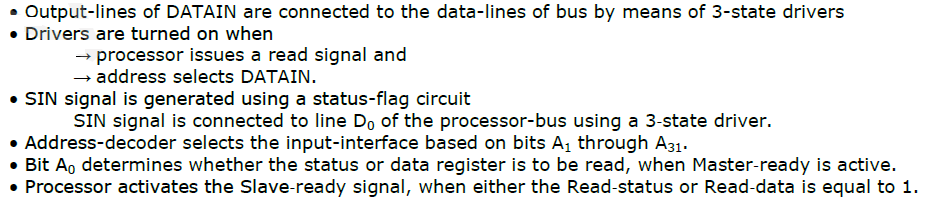
****

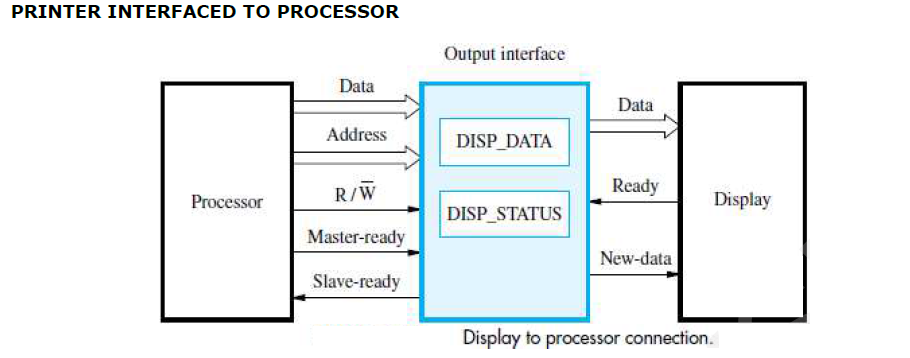
****

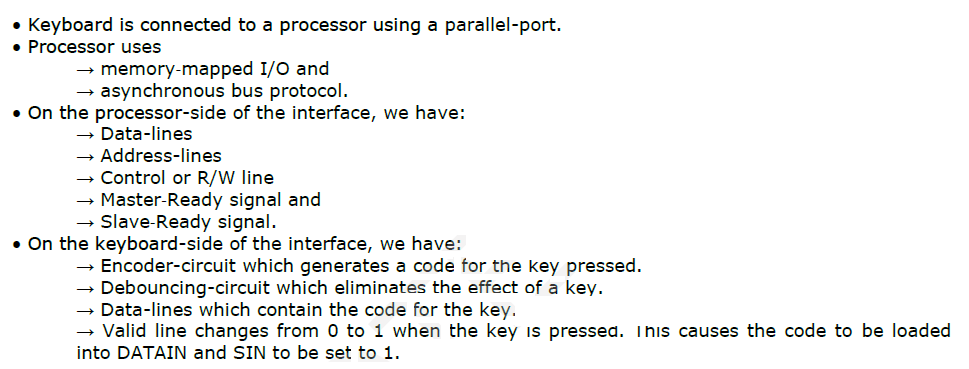
****

****

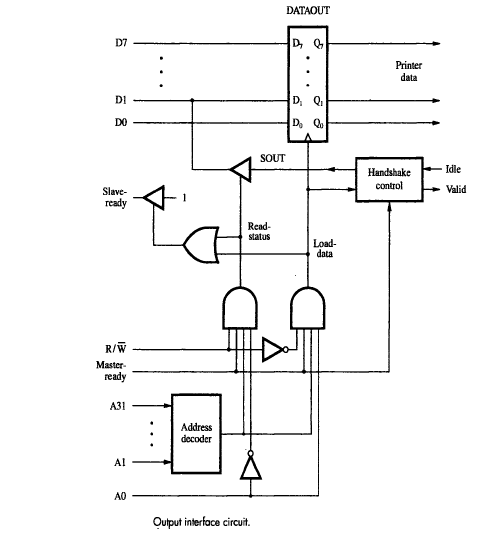






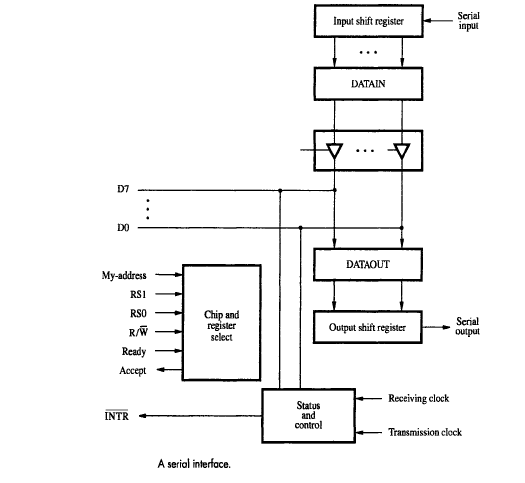


**OUTPUT INTERFACE CIRCUIT**

****

**SERIAL PORT:-**

* A Serial port is used to connect the processor to I/O devices that require transmission of data one bit at a time.
* The key feature of an interface circuit for a serial port is that it is capable of communicating in a bit-serial fashion on the device side and in a bit-parallel fashion on the bus side.
* The transformation between the parallel and serial formats is achieved with shift registers that have parallel access capability.
* A block diagram of a typical serial interface. It includes the familiar DATAIN and DATAOUT registers.
* The input shift register accepts bit-serial input from the I/O device. When all 8 bits of data have been received, the contents of this shift register are loaded in parallel into the DATAIN register.
* Similarly, output data in the DATAOUT register are loaded into the output register, from which the bits are shifted out and sent to the I/O device.



* A simpler interface could be implemented by turning DATAIN and DATAOUT into shift registers and eliminating the shift registers.
* However, this would impose awkward restrictions on the operation of the I/O device; after receiving one character from the serial line, the device cannot start receiving the next character until the processor reads the contents of DATAIN.
* Thus, a pause would be needed between two characters to allow the processor to read the input data.
* With the double buffer, the transfer of the second character can begin as soon as the first character is loaded from the shift register into the DATAIN register.
* Thus, provided the processor reads the contents of DATAIN before the serial transfer of the second character is completed, the interface can receive a continuous stream of serial data.

**STANDARD I/O INTERFACE:**

