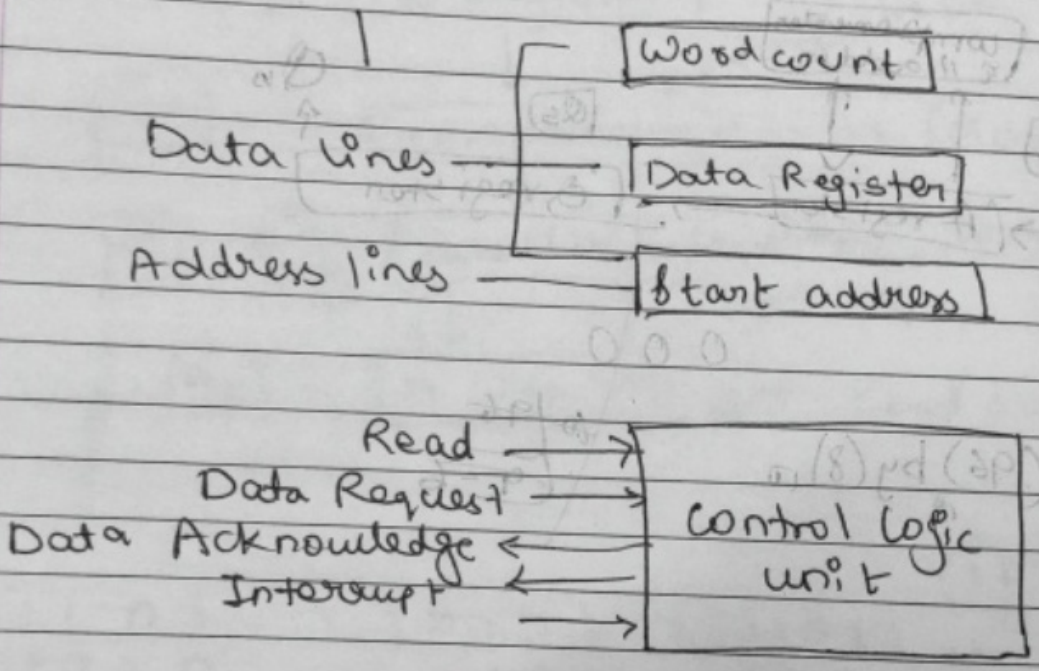
RANKA
DATE / /
PAGE

Assignment-4

1. Explain with the blak dia. the DMA transfer in a computer system.

Sol



* DMA block diagram

Word count - counts the no. of words in a bit.

Control register - specifies the transfer mode.

Address register - it contains the address to specify the desired location in memory.

Working :-

I/O device wants to transfer data to or from memory

↓

Sends DRQ (Data DMA request) to DMA controller

↓

accepts DRQ and send BR (Bus Request) or HLD (Hold request) to CPU to give up the control bus.

2. Explain different modes of data Transfer in detail.

Sol There are 3 possible modes :-

1. Programmed I/O - keeps CPU busy
   - several instructions need to be done in CPU to get the data transfer, it
   - doesn't have direct access to memory.
   - can be avoided by using interrupt facility

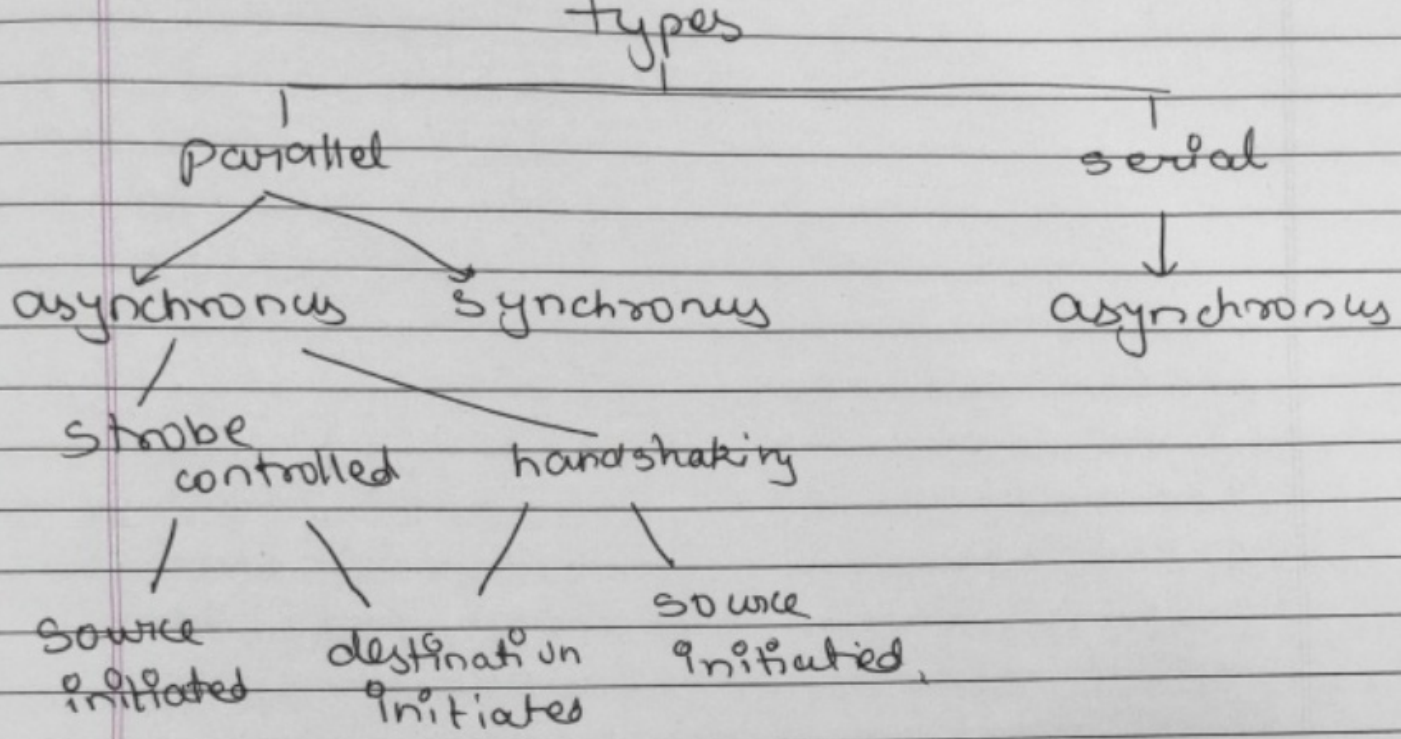
2. Interrupt initiated I/O.

- when interface is ready to accept for data transfer then it receives an 'interrupt signal' to the CPU.
- upon detection it will start working on it.
- if multiple interrupts occurs then the interrupt is done by on the basis of priority.

3. Direct Memory Access (DMA)

- DMA is a process of communication for data transfer b/w memory and I/O by device controlled by an external device controller known as DMA controller, without the intervention of CPU.
- reduces the load on CPU.
- takes over bus to transfer data.

3. Describe the types of data Transfer in detail.



Parallel d.t	Serial d.t
* each bit has its own path.	* each bit has same path.
* total message transmitted at the same time.	* each bit in message is transferred one at a time
* n wires for n-bits	* one wire for n-bits.
* expensive	* economic
* fast	* slow
* difficult to upgrade the system.	* easy to upgrade the system.
* suitable for short distance.	* suitable for long distance.
eg: CPU & memory bus	eg: CPU & I/O bus

⇒ Synchronous data transfer

* Both sender & receiver share the same clock.
* transmitter transmits block of character along with synchronization information.
* continous data transmission.
* accurate timing signals.
* faster than asynchronous d.t

Transmitter: ——— sends data + sync' info ———→ decode & : receiver

keeps clocks in sync, counts no of bits sent by transmitter

⇒ Asynchronous parallel

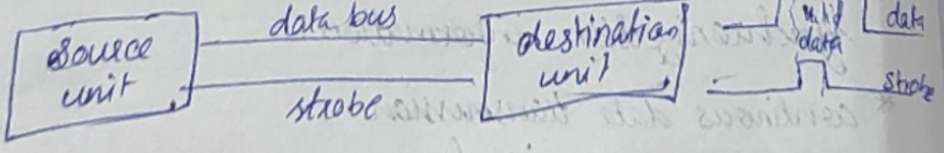
* control signals for synchronization: (2 ways):
  i) strobe control
  ii) Handshaking

i) Strobe control:
- one unit sends strobe pulse to another unit when transfer has to occur.

ii) Handshaking:
- A control signal which indicates the presence of data in the bus is sent along with the data to another unit.
- The receiving unit sends acknowledge receipt of the data.

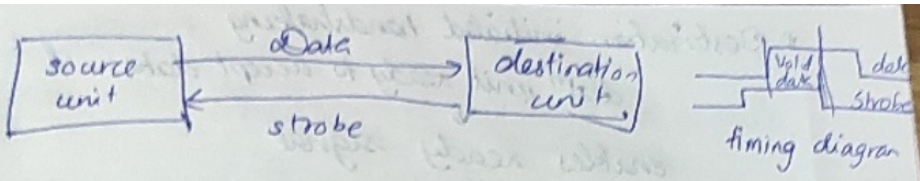
* Source initiated strobe controlled
- data transfer initiated by source.
- source places data in bus → generates strobe pulse once the data is steady → strobe pulse remain active for a period of time for dest^n to receive the data → ~~source removes data from bus~~ → source disables the strobe pulse → source removes data from bus.
- disabled strobe signal = no valid data

eg: CPU - memory
memory write



* Destination initiated strobe controlled
- data transfer initiated by destination.
- dest^n unit ready to receive data
  ↓
  generates strobe pulse
  ↓
  source unit may provide data (valid data for enough time)
  ↓
  dest^n unit receives data
  ↓
  dest^n unit disables strobe pulse
  ↓
  source removes data from data bus

eg: memory → CPU
memory read

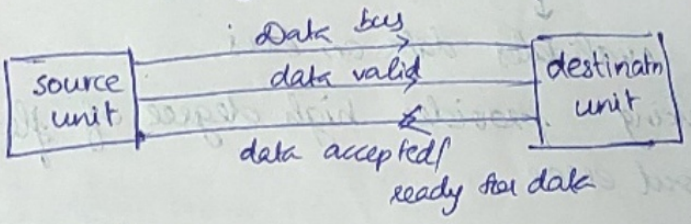


disadv * source unit won't know whether the dest^n got the data or not.

* dest^n can't know whether source placed data in the bus or not.

⇒ Handshaking

* 2 wire controlling method of data transfer.



1 control line : indicates presence of valid data
: source → dest^n

2 control line : dest^n → source
• to indicate whether data is accepted or not

* Source initiated handshaking

source places data on bus
↓
enables data valid signal
↓
dest^n accepts data
↓
dest^n activates data accepted sign
↓
source unit disables data valid sign
↓
dest^n unit disables data accepted sig
↓
system back to initial state.

* Destination initiated handshaking

dest^n unit ready to accept data

↓

enables ready signal

↓

source u places data on bus

↓

source unit enables valid data signal

↓

d.u accepts data from bus

↓

d.u disables ready for data signal

↓

s.u disables data valid signal

↓

invalidates data on bus.

* handshaking provides high degree of flexibility.
* time out errors.

⇒ Asynchronous serial

* different clocks for receiver & transmitter.
* low speed
* character by character data transfer
* framing technique is used

Framing : each character carries info^n of start bit & stop bit.

no transmission : line maintained in high voltage - MARK

start bit - 0 V space

start bit → parity bit → stop bit → 1 / more bits high state (1)

(optional) - if present either in low state (0) or high state (1)

Comm^n interface → ppt