INPUT - OUTPUT Organisation

Peripheral devices

The i/p - o/p subsytem of a computer known as I/O provides an efficient mode of communication between the central system and the outside environment.

→ Input & output devices attached to the computer are called peripherals

→ Computer peripherals can be divided into 2 categories -

1) The devices that perform input & output operations such as keyboard, mouse, monitor etc.

2) The devices for secondary storage of data such as magnetic disc, flopy etc.

12/11/18

Input - Output interface.

→ Peripherals connected to a computer need special communication links for interfacing them. with the CPU. The purpose of the communication link is to resolve the differences that exist between the central computer and each peripheral.

1/1

The major differences are -

1) Peripherals are electromechanical & electro-magnetic devices & there manner of operation is different from the operation of CPU. which are electronic devices so, conversion of signal value is required.

Date: 12/11/18
Page:

2) Data transfer rate of peripherals are slower than of CPU so, synchronization mechanism is needed.

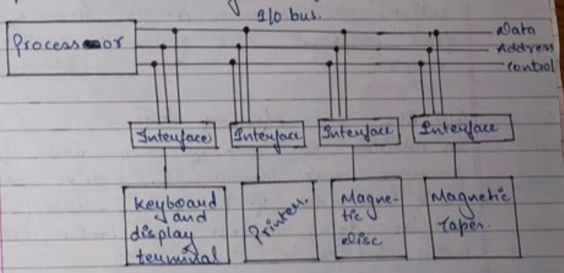
3) differ of data format and codes from word format in CPU and memory.

4) differ of operating mode.

To resolve these differences computer system include special hardware component b/w CPU & peripherals to supervise & synchronise all input & output transfer. These components are called interface units.

write this also in I/O interface

→ I/O BUS and interface moduler.



At the same time address is made available in the address lines, the processor provides a function code in the control lines. The interface selected responds to function code & proceeds to execute it. The function code is known as I/O command.

There are 4 types of command that an interface may receive -

1) **Control command** : The control command is issued to activate the peripheral & to inform it what to do.

2) **Status command** : It is used to test various status condition in the interface and peripheral.

3) **Data output command** : It causes the interface to respond by transferring data from the bus into one of its registers.

4) **Data input command.** : It is opposite of data output. The interface receives item of data from peripheral & place it in its buffer register.

I/O Bus and Memory Bus

In addition to communicating with I/O the processor must communicate with memory unit like the I/O Bus, the memory bus contains data, address and ~~contro~~ read/write control lines.

There are 3 ways the computer buses can be communicated with memory & I/O -

1) Use 2 separate buses one for memory & other for I/O

2) Use 1 common bus for both memory & I/O but separate control lines for each (isolated I/O)

3) Use 1 common bus for memory and I/O with common control lines (memory mapped I/O)

Isolated Vs Memory Mapped I/O

Isolated :

- Separate I/O read/write control lines in addition to memory read/write control lines
- separate (isolated) memory and I/O address spaces
- distinct input and output instructions.

Memory-mapped. :

- A single set of read/write control lines (no distinction b/w memory & I/O transfer)
- Memory and I/O addresses share the common address space
  → reduces memory address range available.
- No specific input or output instruction.
  → The same memory reference instr. can be used for I/O transfers.
- Considerable flexibility in handling I/O operations.

13/11/18

DATA TRANSFER OPERATIONS.

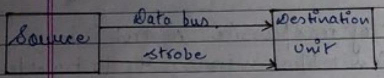
Two units such as CPU & I/O interface are designed independently of each other. If the registers in interface share a common clock with CPU registers the transfer b/w 2 units is said to be Synchronous. In the most cases, the internal timing in each unit is independent from other. In that case each unit uses its own private clock for internal registers. Then the 2 units are said to be asynchronous to each other.

strobe ↙ ↘ handshaking.

i) Strobe control method.

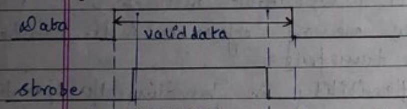
- Employs a single control line to time each trans.
- The strobe may be activated by either the source or the destination unit.

Soure-initiated strobe for data transfer.



(block diagram)

Timing Diagram-

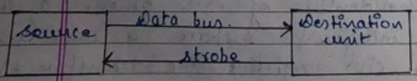


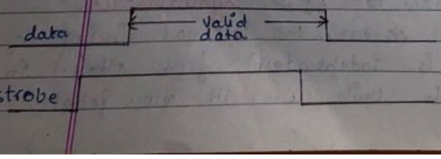
when data reaches steady state then enable strobe.

strobe disabled, but interface first transfer to internal register then disable.

Destination-initiated strobe for data transfer.

1)





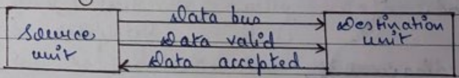
ii) Handshaking method.

The disadvantage of strobe method is that the source unit that initiates the transfer has no way of knowing whether the destination unit has actually received the data item that was placed in the bus. Similarly, a destination unit that initiates the transfer has no way of knowing whether the source unit has actually placed the data on the bus.

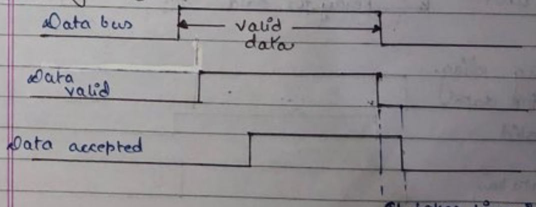
Handshake method solves this problem by introducing a second control signal that provides a reply to the unit that initiates the tranfer.

Source-initiated transfer using handshaking method.

Block diagram

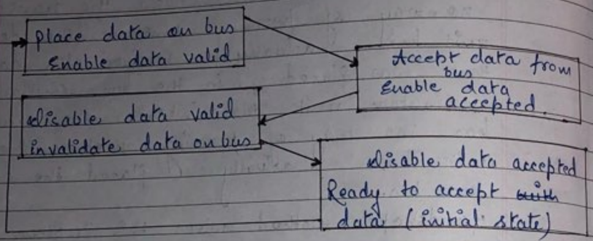


Timing Diagram



It takes time in receiving.

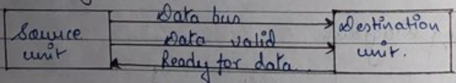
Sequence of events.



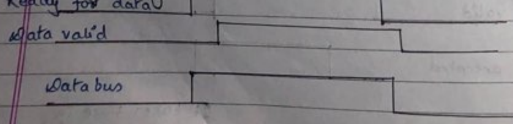
- Allows arbitrary delays from one state to the next
- Permits each unit to respond at its own data transfer rate.

Destination - initiated. transfer using handshaking method.

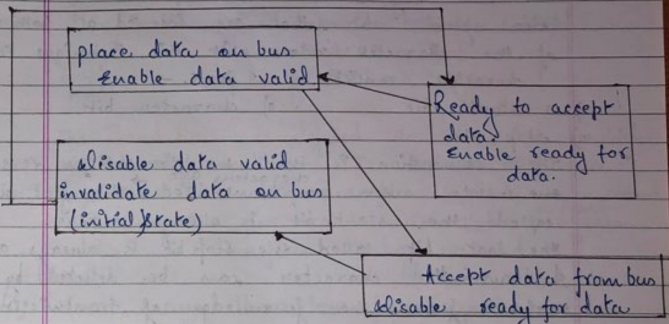
Block diagram



Timing diag.



Sequence of events.



- Handshaking provides high degree of flexibility & reliability because the successful completion of a data transfer relies on active participation by both units.
- If one unit is faulty, data transfer will not be completed.
  → can be detected by means of a timeout mechanism.

Asynchronous Serial Transfer.

A serial asynchronous data transmission technique use special bits that are inserted at both ends of the character code. With this technique each character consist of 3 parts. –

i) a start bit
ii) character bits
iii) stop bits.

The convention is that the transmitter rests at ~~one~~ 1-state when no characters are transmitted. The first bit called the start bit is always a zero. The last bit called the stop bit is always a 1.

→ A transmitted character can be detected by the receiver from the knowledge of transmission rules.

Rules → 1) when the character is not being sent the line is kept in the 1-state.

2) The initiation of a character transmission is detected from start bit which is always zero.

3) The characters bit always follow start bit.

4) After the last bit of the character is transmitted a stop bit is detected when line returns to 1-state for atleast 1 bit time.

start bit $\rightarrow$ $0\,\underbrace{1011010}_{\text{message}}\,1$ $\leftarrow$ stop bit

Receiver should also know the no. of bits in message else there will occur confusion in identifying the stop bit.

Consider the serial transmission of a terminal whose transfer rate is 10 characters per second. Each transmitted character consists of a start bit, 8 information bits & 2 stop bits.

$1 \text{ bit} = \frac{0.1}{11} = 9.09 \times 10^{-3}$ sec.

10 character/sec means each characters take 0.1sec for transfer.

bit time (time taken to transfer 1-bit)

Baud rate

$= 9.09$ msec.

$= 9.09 \times 10^{-3}$ sec.

The Baud Rate is defined as the rate at which serial information is transmitted and is equivalent to data transfer the bits per second. 10 character per second with 11-bit format ~~per seco~~ has a transfer-rate of 110 baud.

Ques. How many characters per second can be transmitted over a 1200 baud lines. in each of the following mode. –

i) synchronous serial transmission
ii) asyn. serial transmission with 2 stop bits
iii) asyn. " " with 1 stop bit.
(assume a character code of 8 bits.)

ii) $1 + 8 + 2 = 11$ ~~char~~ bits.

$\text{no. of char.} = \frac{1200}{11} = 109$ cps

iii) $1 + 8 + 1 = 10$ bits

$\text{no. of char.} = \frac{1200}{10} = 120$ cps

i) 8 bits (no start bit, no stop bit)

$\text{no. of char} = \frac{1200}{8} = 150$ character per sec or 150 cps

Example how interrupt works with inte i.e. without in how sy

INTERRUPTS

→ It is a mechanism by which modules like I/O or Memory may interrupt the normal processing of CPU.

→ Interrupt is required for improving the processing efficiency of processor.

Types of interrupt -

1) Program interrupt
It occurs when some instruction within the program creates a condition that leads to an interrupt.
Ex : Divide by 0, arithmetic overflow, attempt to access an illegal memory location.

2) Timer interrupt.
- Generated by the timer present within the processor.
- Operating system sets the timer to perform certain operation on regular basis.
- ~~I/O interrupt~~

3) I/O interrupt.
- Generated by I/O devices
- signal successful task completion or error conditions.
-

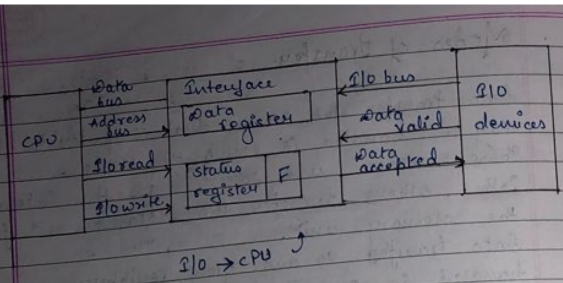
4) Hardware interrupt.
failure related to hardware.

Modes of Transfer

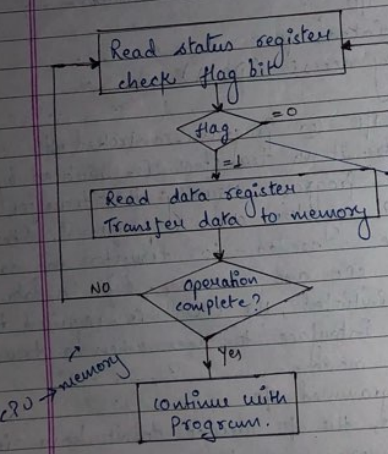
Data transfer b/w the central computer & I/O devices may be handled in varity of modes.
Some modes use CPU as the intermediate path, others transfer the data to and from the memory unit.
Data transfer to and from peripheral may be handled in one of 3 possible modes -

1) Programmed I/O.
2) interrupt initiated I/O
3) direct memory access (DMA)

1) Programmed I/O (or) Program-controlled I/O
Each data ~~tro~~ item transfer is initiated by an instruction in program. Usually the transfer is to and from CPU register and peripheral. Other instruction are needed to transfer the data to and from CPU and memory. Once a data transfer is initiated, the CPU is required to monitor the interface to see when a transfer can again be made.
Example of Programmed I/O
In this method, the I/O device doesn't have direct access to memory a transfer from an I/O device to memory requires the execution of several instr. by CPU including an o/p instruction to transfer the data from the device to the CPU and the stored instruction to transfer the data from CPU to memory.



I/O → CPU



Polling or status checking.

- continuous CPU involvement
- CPU slowed down to I/O speed
- simple
- ~~latest~~ least hardware

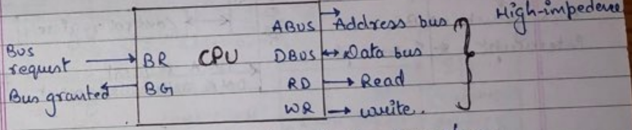
2) Interrupt- initiated I/O

An alternative to CPU constantly monitoring the flag is to let the interface inform the computer when it is ready to transfer data. This mode of transfer uses the interrupt facility.

V.Imp

3) DMA (Direct Memory access)

Removing the CPU from the path and the peripheral device manage the memory buses directly will improve the speed of transfer. This transfer technique is called DMA. (Direct memory access). During DMA transfer the CPU is idle and has no control of the memory buses. A DMA controller takes over the buses to manage the transfer directly between the I/O device and memory.



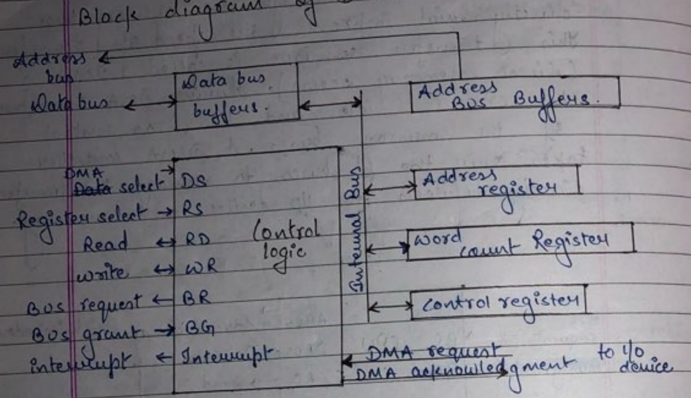
CPU bus signals for DMA transfer.

when DMA takes control of bus system it communicates directly with memory, the transfer can be made in several ways.-

→ In DMA burst transfer, a block sequence consisting of a number of memory words is transferred in a continuous burst while the DMA controller is the master of the memory buses.

→ An alternative technique called cycle-stealing allows the DMA controller to transfer one data word at a time after which it must return control of the buses to the CPU. The CPU delays its operation for 1 memory cycle to allow the direct memory I/O transfer to steal 1 memory cycle.

Block diagram of DMA controller.



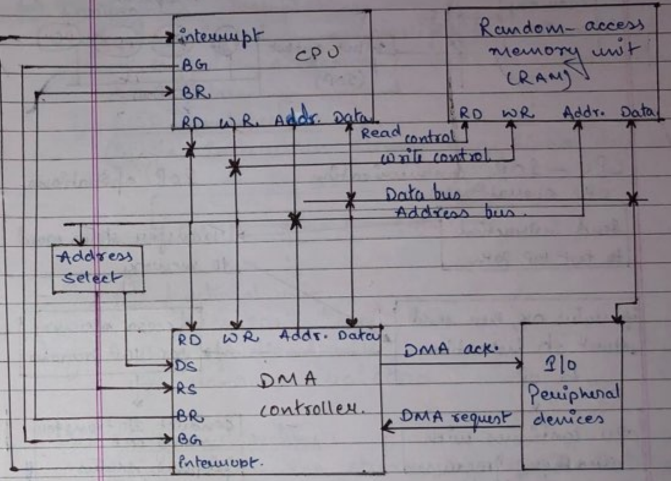
~~BG=0~~ , ~~communication : CPU can read/write DMA reg.~~
~~BG≠0~~ , ~~Transfer b/w~~

The registers in DMA are selected by CPU through the address bus by enabling the DS and RS inputs. The read & write i/p are bidirectional when BG i/p is zero, the read-& write are input lines, allow the CPU to communicate with internal DMA register.

when BG=1, the RD and WR are o/p lines from DMA controller to random access memory to specify the read or write operation for the data.

The DMA controller has 3 register, a address register, a word counte register, a control register.
& (no. of memory word decrement in value) Refer book to explain

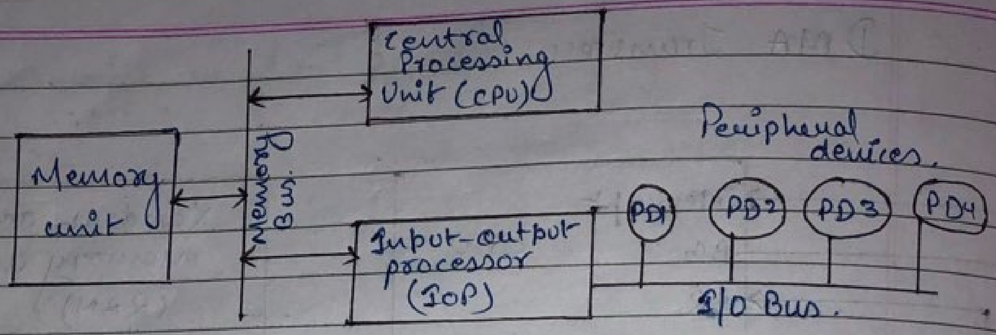
DMA Transfer.



21/11/18

Input-Output Processor.

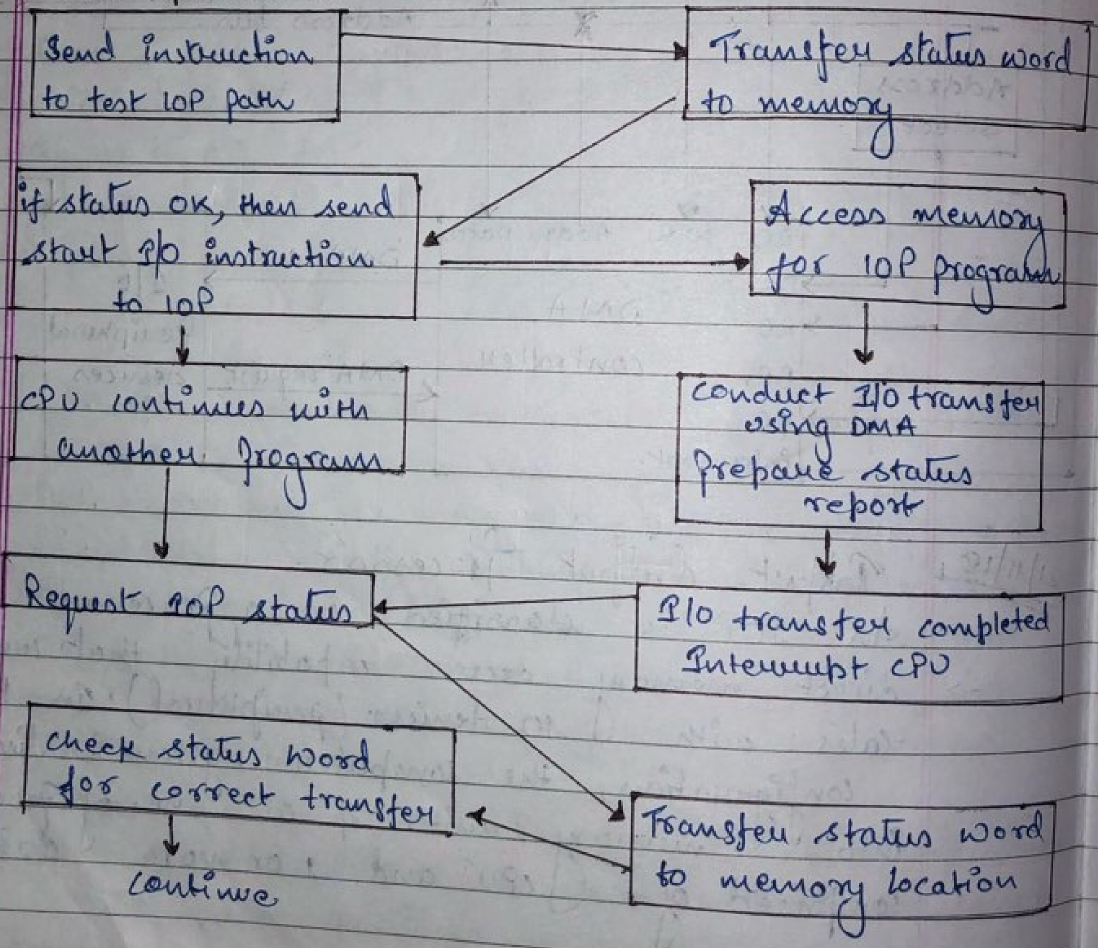
An IOP is classified as a processor with direct memory access capability that communicates with IO devices (peripheral). In this configuration, the computer system is divided into memory unit & a no. of processor comprised of CPU and 1 or more IOP.



CPU – IOP communication

CPU operations.

IOP operations.


Send instruction to test IOP path
Transfer status word to memory
if status OK, then send start I/O instruction to IOP
Access memory for IOP program
CPU continues with another program
conduct I/O transfer using DMA
Prepare status report
Request IOP status
I/O transfer completed
Interrupt CPU
check status word for correct transfer
Transfer status word to memory location
continue