I/O Organization & Peripherals

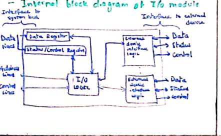
I/O System

- I/O module is interface to CPU & memory with one or more peripherals

Functions of I/O module

- Issue of control & timing signal
- Communication with CPU
- Communication with peripherals
- Detection of errors

Internal block diagram of I/O module



Parallel	Serial
- parallel lines of 8/16/32 bits, hence 8/16/32 are transmitted	- only one bit is transmitted at a time
- data transfer rate is faster	- data rate transfer is slower
- cross talk possible	- crosstalk not possible
- cannot be used for distant communication	- can be used for distant communication
- more parallel hardware is required	- less parallel h/w required
- costlier	- cheaper

Mega batch

Types of communication Systems

① Simplex
- One way transmission
- Connection exists such that data transfer takes place only in one direction
- No possibility of data transfer in other direction
- System A is transmitter & System B is receiver only

② Duplex
- Two way transmission

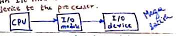
→ Half Duplex
- Connection between 2 terminals such that data may travel in both directions but transmission activates in one direction at a time
- This indicates that line has to turn around after communication is complete in one direction

→ Full Duplex
- Connection between 2 terminals such that data may travel in both directions simultaneously.
- So it can contain one way transmission or two way transmission at a time

I/O Module

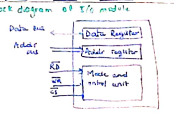
Need of I/O module

- Each output device operates at a different speed
- It has different data format & different protocol
- Most of the I/O devices are slower than the speed of processor
- An I/O module is used to interface the I/O device to the processor.



Mega batch

Block diagram of I/O module



- Data register is used to store the data given by the i/p device to be forwarded to the processor OR given by the processor to be forwarded to an o/p device
- Address register is used to provide address of the IO device to be accessed
- Mode and Control unit indicates the mode of operation for the I/O module as well as controls the transfer of data between IO module and IO device & IO Module and CPU

8089 IO Processor

It is an IO Processor
- It was available for use with the 8086/8088 Central processor.
- It uses same programming technique as 8087 for I/O operations. Such as transfer of data from memory to a peripheral device

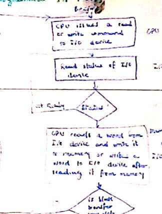
Features
- 8089 has very high speed DMA capability
- It has 1 MB address capability
- It is compatible with iAPX 86, 88
- It supports local mode & remote mode IO processing
- 8089 allows mixed interface of 8 & 16 bit peripherals to 8 & 16 bit processor buses
- It supports 2 IO channels
- Multibus compatible system interface
- Memory based communications with CPU

Types of data transfer techniques

Polling
- It is a mechanism wherein the processor checks each & every device for it needs a service or not

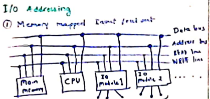
① Programmed I/O
- CPU has direct control over I/O
- Sequence of operations to be carried out
  1. CPU requests for I/O operation
  2. I/O module performs the said operation
  3. I/O module updates the status bits
  4. CPU checks these status bits periodically. Neither the IO module cannot inform CPU directly nor can IO module interrupt CPU directly
  5. CPU may wait for the opn to complete or may continue the opn later

Transferring a block of data using programmed I/P



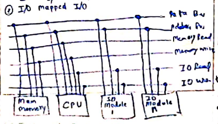
I/O Addressing

① Memory mapped Input/output

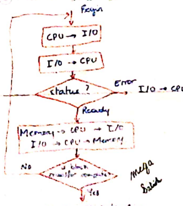


- Read command is for i/p of data
- Write command is for o/p of data
- I/O module is used for interfacing of a device with system bus.

② I/O mapped I/O



③ Interrupt Driven IO



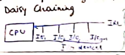
Mega batch

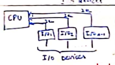
Interrupt Processing

CPU takes following steps in response to an interrupt
- CPU identifies source of interrupt
- CPU find address of interrupt handling pgm
- PC and status word PSW is saved on stack
- PC loaded with interrupt handler, this will transfer control to interrupt handler pgm
- execution of interrupt handler proceeds until returned instruction is encountered, which transfers control back to interrupt program

Interrupt Selection

- Each device may have an independent interrupt request line going upto CPU
- Multiple devices may share the single IRL
- CPU may be using vectored interrupt using Daisy Chaining





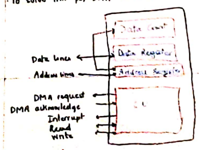
Independent request line
Software Pole
Vectored interrupt using Daisy Chaining
Bus Arbitration (Vectored) using 8259
Subroutine and interrupt service routine
Subroutine call

Types of interrupts

- Program interrupts
- Timer interrupts
- I/O interrupts
- Hardware failure

DMA

- Direct Memory Access
- I/O module can directly access (read or write) the memory using DMA
- Interrupt Driven and Programmed I/O requires active operation of CPU. Hence transfer rate is limited & CPU is also busy doing the transfer operation.
- To solve this pb, DMA is used

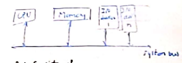


DMA Transfer modes
- Single transfer mode
- Block transfer mode
- Demand transfer mode
- Hidden transfer mode

Bus

- group of wires, pins, signals, connection it having common function is called as Bus
- the Bus or connection that do the function of carrying data is called as data bus
- Bus or connection that do the function of carrying address is called as address bus
- Bus or connection that do the function of carrying control signal is called control bus
- The 3 buses together are called as system bus

Single Bus Structure



System Bus Consists of
- data bus
- address bus
- control bus

Operation of the bus
- obtain control of bus
- transfer data via bus
} if one module sends data

- obtain control of bus
- makes a request to other module over the appropriate control lines & address lines
- it must then wait for requested module to send data
} if one module requests data

Multiple Bus Hierarchies

- if a greater no of devices are connected to the BUS, performance will suffer due to
  → in general more devices attached to Bus, the greater will be the propagation delay
  → the bus may contain a bottleneck as aggregate data transfer demand approaches the capacity of Bus
  → pb can be solved to some extent by increasing data rate that Bus can carry and by using wider buses

Bus Arbitration

- When many bus-master are connected to a single one, there is bus congestion
- To avoid the pb, we use bus arbitration
- The arbitration procedure comes into picture whenever there are more than one processor requesting the services of bus
- The process of selecting processor is called as arbitration
- Selection mechanism must be based on fairness or priority basis

3 representative arbitration schemes
- Daisy chaining
- Polling
- Independent requesting

Daisy Chaining

Adv
- simple arbitration scheme
- requires very few control signals
- additional devices can easily be added

Disadv
- priority is wired and cannot be changed
- could be a problem of starvation
- if master 1 is generating bus request at a high rate, then rest of masters may not get bus sometimes

Polling

Adv.
- priority can be changed by changing sequence of generation of addresses on poll count line
- Failure one of master cannot effect other masters

Disadv:
- polling requires more control line
- Max no. of master to be connected to Bus is restricted by poll count lines with n poll count lines, we can have max 2^n masters

Element of Bus Design

Bus Type → dedicated
→ multiplexed

Method of Arbitration
→ centralized
→ distributed

Timing
- Synchronous
- Asynchronous

Bus Width
→ Address Bus width
→ Data Bus width

Data Transfer type
- Read
- write
- Read modified write
- Read after write
- Block

Interrupt Types

Interrupt

- mechanism by which an I/O device or instruction can suspend the normal execution of processor and get itself serviced

ISR

- Interrupt service routine
- small pgm or routine that when executed services the corresponding interrupt source is called an ISR.

Vectored/Non Vectored Interrupt

- If ISR address of interrupt is to be taken from interrupting source itself, then it's called as non vectored interrupt else vectored interrupt

Maskable / Non maskable Interrupt

Interrupt that can masked (disabled) or unmasked (enable) by programmer is called as maskable interrupt else non maskable interrupt

Sources of interrupt
- H/W interrupt
- S/W interrupt
- Error conditions

H/W Interrupt	S/W Interrupt
- pin is given on processor	- an instruction is given in the instruction set of processor
- maskable / non maskable	- mostly non maskable & may have lower priority
- vectored or non vectored	- Vectored
- of 8086 are NMI & INTR	- of 8086 are INTn where n is any value from 00H to FFH

Interrupt handling

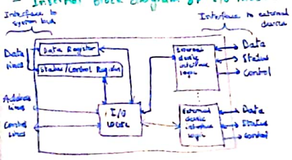
- at the end of each instruction cycle the 8086 checks to see if any interrupt has been requested.
- Therefore, whenever interrupt occurs, it won't immediately checked by microprocessor
- Microprocessor first complete execution of current instruction & then checks for an interrupt

I/O System

- I/O module is interface to CPU & memory with one or more peripherals

Functions of I/O module
- Issue of control & timing Signal
- Communication with CPU
- Communication with peripherals
- Detection of errors

- Internal block diagram of I/O module



Parallel	Serial
- parallel lines of 8/16/32 bits, hence 8/16/32 are transmitted	- only one bit is transmitted at a time
- data transfer rate is faster	- data rate transfer is slower
- cross talk possible	- crosstalk not possible
- cannot be used for distant communication	can be used for distant communication
more parallel hardware is required	- less parallel h/w required
- costlier	- cheaper

Types of communication Systems

① Simplex
- One way transmission
- Connection exists such that data transfer takes place only in one direction
- No possibility of data transfer in other direction
- System A is transmitter & System B is receiver only

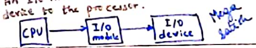
② Duplex
- Two way transmission

→ Half Duplex
- Connection between 2 terminals such that data may travel in both directions but transmission activates in one direction at a time
- This indicates that line has to turn around after communication is complete in one direction

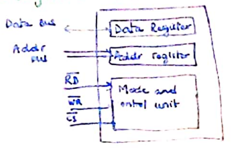
→ Full Duplex
- Connection between 2 terminals such that data may travel in both directions simultaneously.
- So it will contain one way [illegible] or two way transmission at a time

I/O Module

- Need of I/O module
- Each output device operates at a different speed
- It has different data format & different protocol
- Most of the I/O devices are slower than the speed of processor
- An I/O module is used to interface the I/O device to the processor.



- Block diagram of I/O module



- Data register is used to store the data given by the i/p device to be forwarded to the processor OR given by the processor to be forwarded to an o/p device
- Address register is used to provide address of the IO device to be accessed
- Mode and Control unit indicates the mode of operation for the I/O module as well as controls the transfer of data between IO module and IO device & IO Module and CPU

8089 IO Processor

It is an IO Processor
- It was available for use with the 8086/8088 central processor.
- It uses same programming technique as 8087 for I/O operations such as transfer of data from memory to a peripheral device

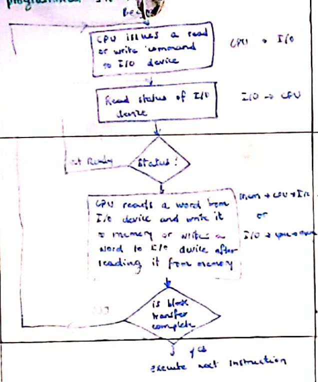
Features
- 8089 has very high speed DMA capability
- It has 1 MB address capability
- It is compatible with iAPX 86, 88
- It supports local mode & Remote mode IO Processing
- 8089 allows mixed interface of 8 & 16 bit peripherals to 8 & 16 bit processor buses
- It supports 2 IO channels
- Multibus compatible system interface
- Memory based communications with CPU

Types of data transfer techniques

Polling
- It is a mechanism wherein the processor checks each & every device for it needs a service or not

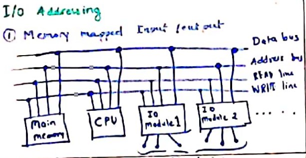
① Programmed I/O
- CPU has direct control over I/O
- Sequence of operations to be carried out
  ① CPU requests for I/O operation
  ② I/O module performs the said operation
  ③ I/O module updates the status bits
  ④ CPU checks these status bits periodically. Neither the I/O module cannot inform CPU directly nor can I/O module interrupt CPU
  ⑤ CPU may wait for the op to complete or may continue the op later

Transferring a block of data using programmed I/O



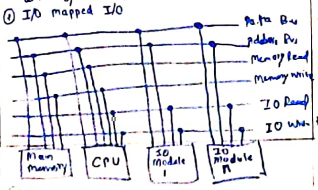
I/O Addressing

① Memory mapped Input/output

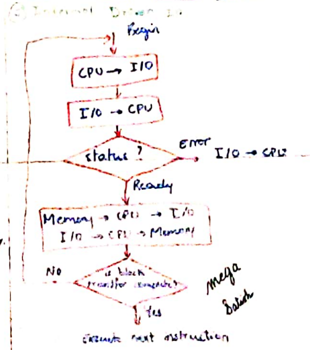


- Read command is for i/p of data
- Write command is for o/p of data
- I/O module is used for interfacing of a device with system bus.

② I/O mapped I/O



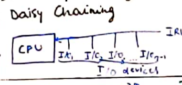
② Interrupt Driven IO

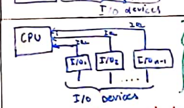


Interrupt Processing

CPU takes following steps to response to an interrupt
- CPU identifies source of interrupt
- CPU find address of interrupt handling pgm
- PC and status word PSW is saved on stack
- PC loaded with interrupt handler, this will transfer control to interrupt handler pgm
- execution of interrupt handler proceeds until returned instruction is encountered, which transfers control back to interrupt program

Interrupt Selection
- Each device may have an independent interrupt request line going upto CPU
- Multiple devices may share the single IRL
- CPU may be using vectored interrupt using Daisy Chaining



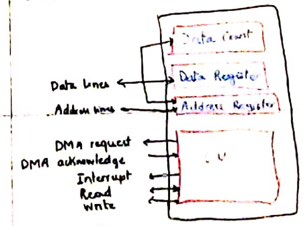


Independent request line
Software Pole
Vectored interrupt using Daisy Chaining
Bus Arbitration (Vectored) using 8259
Subroutine and interrupt service routine
Subroutine call

Types of interrupts
- Program interrupts
- Timer interrupts
- I/O interrupts
- Hardware Failure

DMA
- Direct Memory Access
- I/O module can directly access (read or write) the memory using DMA

- Interrupt Driven and Programmed I/O requires active operation of CPU. Hence transfer rate is limited & CPU is also busy doing the transfer operation.
- To solve this pb, DMA is used.

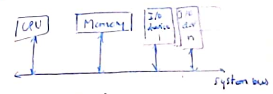


DMA Transfer modes
- Single transfer mode
- Block transfer mode
- Demand transfer mode
- Hidden transfer mode

bus

Group of wires, pins, signals, connection etc. having common function is called as Bus
- The Bus or connection that do the function of carrying data is called as data bus
- Bus or connection that do the function of carrying address is called as address bus
- Bus or connection that do the function of carrying control signal is called control bus
- The 3 buses together are called as system bus

Single Bus Structure



System Bus Consists of
- data bus
- address bus
- control bus

Operation of the bus
- obtain control of bus
- transfer data via bus
} if one module sends data

- obtain control of bus
- makes a request to other module over the appropriate control lines & address lines
- it must then wait for requested module to send data
} if one module requests data

Multiple Bus Hierarchies
- if a greater no of devices are connected to the BUS, performance will suffer due to
  → in general more devices attached to BUS, the greater will be the propagation delay
  → the bus may contain a bottleneck as aggregate data transfer demand approaches the capacity of Bus
  → pb can be solved to some extent by increasing data rate that Bus can carry and by using wider buses

Bus Arbitration
- When many bus-master are connected to a single one, there is bus congestion
- To avoid this pb, we use bus arbitration
- The arbitration procedures comes into picture whenever there are more than one processor requesting the services of bus
- The process of selecting processor is called as arbitration
- Selection mechanism must be based on fairness or priority basis

3 representative arbitration schemes
- Daisy chaining
- Polling
- Independent Requesting

Daisy Chaining

Adv:
- simple arbitration scheme
- requires very few control signals
- additional devices can easily be added

Disadv
- priority is wired and cannot be changed
- could be a problem of starvation
- if master 1 is generating bus request at a high rate, then rest of masters may not get bus sometimes

Polling

Adv:
- priority can be changed
- by changing sequence of generation of addresses on poll count line
- Failure one of master cannot affect other masters

Disadv:
- polling requires more control line
- Max no. of master to be connected to Bus is restricted by poll count lines with n poll count lines, we can have max 2^n masters

Element of Bus Design

Bus Type → dedicated
→ multiplexed

Method of Arbitration
→ centalized
→ distributed

Timing
→ Synchronous
→ Asynchronous

Bus Width
→ Address Bus Width
→ Data Bus Width

Data Transfer type
→ Read
→ Write
→ Read modified write
→ Read after write
→ Block

Interrupt

Interrupt
- mechanism by which an I/O device or instruction can suspend the normal execution of processor and get itself serviced

ISR
- interrupt service routine
- Small prgm or routine that when executed services, the corresponding interrupt source is called an ISR.

Vectored/Non Vectored Interrupt
- If ISR address of interrupt is to be taken from interrupting source itself, then it's called as non vectored interrupt else vectored interrupt

Maskable / Non maskable Interrupt
- Interrupt that can masked (disabled) or unmasked (enable) by programmer is called as maskable interrupt else non maskable interrupt

Sources of interrupt

H/W Interrupt	S/W interrupt
- pin is given on processor	- an instruction is given in the instruction set of processor
- maskable / non maskable	- mostly non maskable & may have lower priority

- vectored or non vectored
- of 8086 are NMI & INTR

- Vectored
- of 8086 are INTn where n is any values from 00H to FFH

Interrupt handling

- at the end of each instruction cycle the 8086 checks to see if any interrupt have been requested.
- Therefore, whenever interrupt occurs, it won't immediately checked by microprocessor
- Microprocessor first complete execution of current instruction & then checks for an interrupt