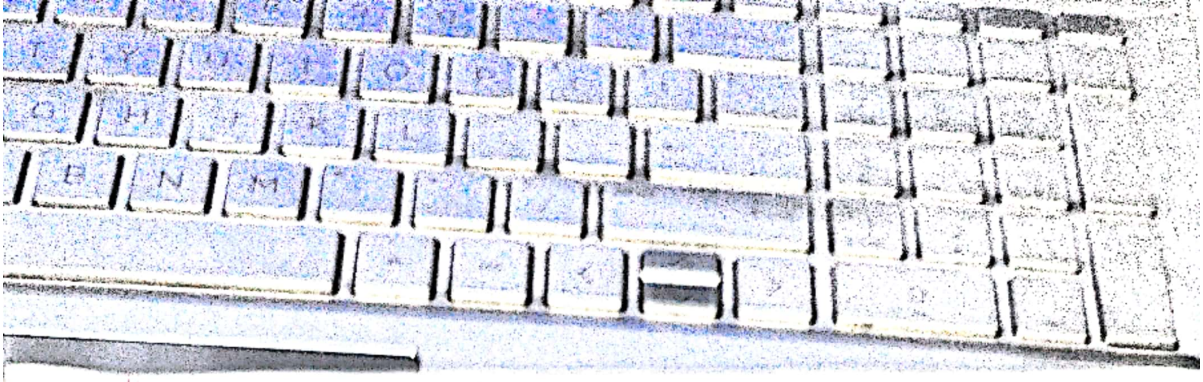


Input & Output

- Technique of data transfer between microcomputer and external device.

Three ways of transferring data.

a) Programmed I/O
b) Interrupt Driven I/O
c) Direct Memory Access (DMA)

#Standard I/O

Two separate address spaces.
one for memory location
other for I/O devices

M/$\overline{IO}$ pin High => mem. operation going on
M/$\overline{IO}$ pin Low => I/O operation going on

IN AL, PORTA
OUT PORTA, AL.

Memory Operation,

LDA 3000H (Copy the content of mem. Loc. 3000H)
STA 3000H (Store content into the mem. loc. 3000H)

#Advantage:

- 1 MB memory address available for 8086
- Special instruction for I/O operation.
- maximize I/O Performance.
- Used in system where complete memory capacity is required.

Disadvantage:

Data has to be transferred to the accumulator to perform logical or Arithmetic operation.

Memory - mapped I/O:

CPU address, I/O device like a memory location.

- Single address space is used by CPU

Msb=1 : I/O select.
Msb=0; Memory location selected.

MOV mem, reg ; Output.

MOV reg, mem; Input.

C. Direct memory access:

Data can be transfer between micro computer memory and external device without microprocessor involvement.

* DMA controller used (8237)
  - DMA address Register → contain mem address
    DMA count Register → no. of bytes of data (to be transfer)
    DMA control register → accepts commands from CPU.

3 Modes of Data Transfer:

- BURST mode
- CYCLE mode
- Transparent mode:

DMA Operation:

i) I/O device request DMA operation via the DMA request line of the controller chip.

ii) Controller chip activates the micro processor Hold pin requesting the microprocessor to release the bus.

iii) The microprocessor sends HLDA back to the DMA controller, indicating that bus is disable.

iv) The DMA controller places the memory address on the address bus. And send acknowledge to the peripheral device.

v) The DMA controller completes the DMA transfer and release the bus.

#BURST mode:

Helpful for loading program or data file into memory.

inactive mp cpu for long periods of time

T_1 (Bus Controll) + T_{cl} (Send) + T_1 (Return).

#Cycle Stealing Mode:

One byte per data.

$T_1 + T_{1MB} + T_1 + T_1 + T_{1mb} + T_1 + \dots + T_1 + T_{1mb} + T_1 = 1GB.$

Problem-1 (Soln)

$T_1 = 250\ ns$
$= 250 \times 10^{-9}\ s$.

75KB = 75 × 1024 Byte.

1B/s = 1.30×10^{-5} s

256B → $(256 \times 1.30 \times 10^{-5})$ s
$= 3.328 \times 10^{-3}$ s.

Total time:

$T_1 + T_w + T_1$

$= 250 \times 10^{-9} + 3.328 \times 10^{-3} + 250 \times 10^{-9}$

$= 3.328 \times 10^{-3}$ s

$= 3.328$ ms.

Problem-2 (Soln)

$\{(500 \times 10^{-9} + 1.30 \times 10^{-5}) \times 256\}$ s.

For 8086

Address bus = 20 bit.

Data bus = 16 bit. ← 16 bit Processor.

2^n ← add. bus size.

8bit = 1 byte.

#Size of Ram 1GB. Then Add. Bus = ?

$1GB = 2^{10} MB$

$= 2^{10} \times 2^{10} kB$

$= 2^{20} \times 2^{10} B$

$= 2^{30}$ Byte.

= 30 byte.

$2^n \times 8 = 1GB = 2^{30} \times 8$

$\therefore n = 30.$

#Maximum Size of Ram. for 8086.

Soln:

Address bus = 20 bit.

$\Rightarrow (2^{20} \times 2^3)$ bit.

$= \frac{2^{20}}{2^{10}} byte \cdot 2^{20}$ byte.

$= \frac{2^{20}}{2^{10}} kB$

$= 2^{10} kB$

$= \frac{2^{10}}{2^{10}} MB = 1MB.$

2. Interrupt Driven I/O

- an external device can force the microprocessor to stop executing the current program temporarily.

Two Types of interrupt:

i) External
ii) Internal.

i) External interrupt:

maskable, non-maskable.

- maskable interrupt can be enable or disable by instruction.
- non-maskable can not. (Power failure interrupt)

Logical address - 16 bit.
offset " = 16 bit.
Physical " = 20 bit.

PA = Segment x 10H + offset address

CS -> register of segment
IP -> " " offset.

Segment: 1234H.

Starting PA = 1234H x 10H + offset
= 12340 + 0000H.
= 12340H.

ending PA = 1234H x 10H + offset.
= 12340 + FFFFH
= 2233FH.

* Stack for temporally hold data.

For CS

1111H x 10H + 1232H

=> 11110H + 1232H

=> 12342H

For DS

3333H x 10H + 0020H

=> 33330 + 0020H.

=> 33350H

for SS

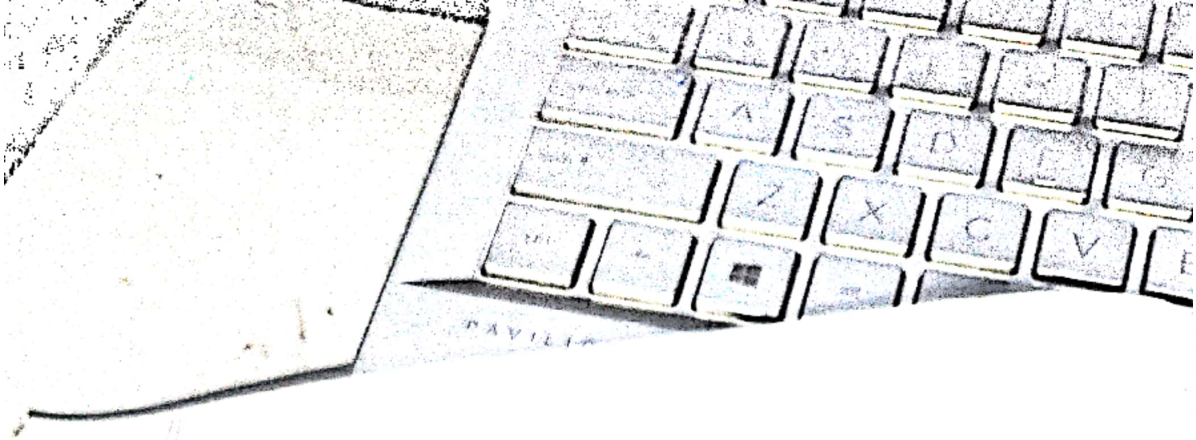
2526OH + 1100H

=> 26360H.

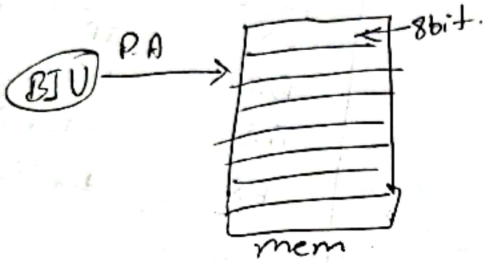
Segment	offset
CS	IP
DS	DI, SI, BX
SS	SP, BP
ES	DI, SI, BX.

EU -> Execution Unit.

BIU => Bus Interface Unit



#8086 Internal Architecture:



max=48 bit

Operating clock 5/8/10 MHz

$D_0 \rightarrow D_{15}$ ⇒ Data Bus.

$D_8 - D_{15}$ ⇒ High Data Bus. $D_0 - D_7$ = Low Data Bus

Sunday →,

Mem segmen

8086 Part 1,2,3,4

8086

→ 8085 => 8-bit micro-Processor.

8086 Series:

i) 8086 = 16-bit Processor.

ii) 80186 = 16-bit with Peripheral Device.

iii) 80286 - Used for multitasking.

iv) 80386 = 32-bit arch. Processor.

v) 80486 - With floating Point.

Internal Arch. of 8086

GND = PIN 20

Vcc = PIN = 40

• $AD_0 - AD_{15}$ PIN: (Address bus [Low Order])

→ Bidirectional.

$D_0 - D_7$ use for Low data bus.

$D_8 - D_{15}$ use for high data bus.

$A_{16} - A_{19}$

High order ~~Data~~ Address bus.

8086 Has two Units:

i) Bus Interface Unit (BIU)
ii) Execution Unit (EU)

BIU Function:

→ Sends out addresses.
→ fetches instruction from memory.
→ read data from Ports and memory
→ Write data to Ports and memory.
→ handles all transfer of data and addresses on the buses for the execution unit.

EU Function:

→ coordination of all the units.
→ Instruct BIU where to fetch instruction.
then decodes in the EU translates instruction fetched from memory into a series of action which the EU carries out.
→ ALU Performs arithmetic and logical operation over the data.

Description of EU

→ 16-bit arithmetic logic unit.

→ 4, 16-bit general purpose registers.
AX, BX, CX, DX. And furthere it can be divided into 2 8-bit registers.

AH + AL => AX [Accumulator Register]
BH + BL => BX [Base Register]
CH + CL => CX [counte Register]
DH + DL => DX [Data Register]

→ also 4 other register SI, DI, SP, BP.

→ 16-bit flag register contain 9 flags

→ rest of three are used to contedl a certain operations

Description of BIU:

-> BIU stores pre-fetched instruction byte in queue also called FIFO

-> Fetching the next instruction while executing the current instruction is ~~ref~~ referred as the instruction pipeline.

-> Had a adder to produce 20 bit physical address.

-> Bus controll logic generates all bus controll signal. read/write for memory/I/O ports.

-> 4 Segment Register. CS, DS, SS, ES

-> this hold the starting address of four memory segments.

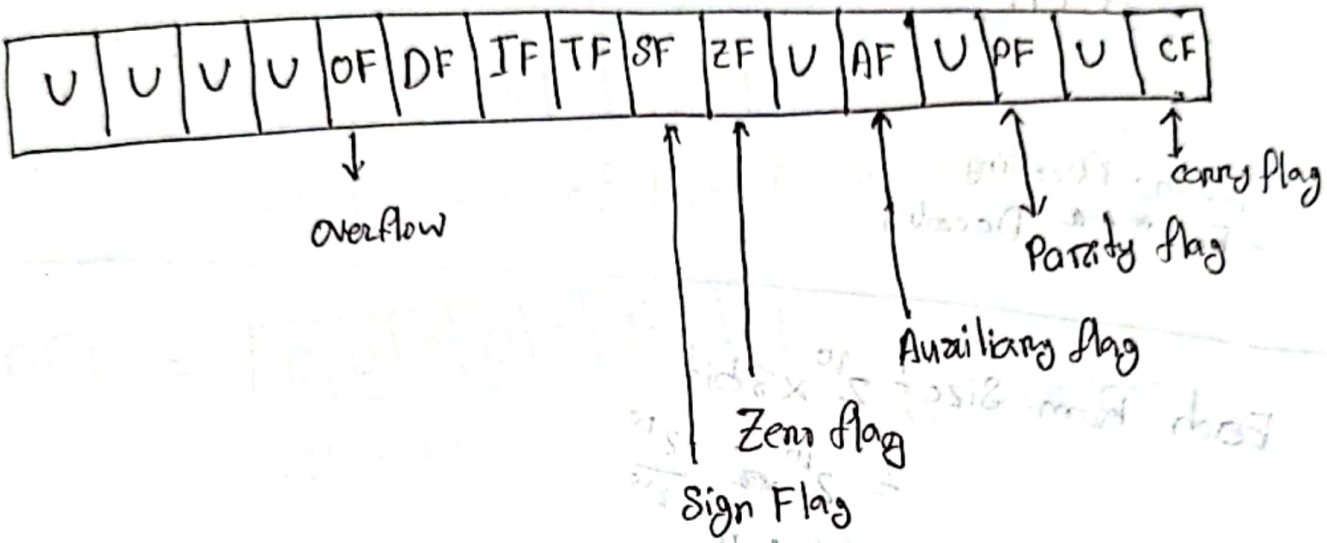
Conclusion:

BIU fetches instruction read/write from memory/IO

EU executes instructions that have already been fetched by BIU

- AX register:
  - Accumulator Register
  - use in arithmetic, logic and data transfer
  - generate shortest MLC.
  - must use in multiplication and division operations and I/O operation.

#Flag Register:




```
                    1
  0010   0101   1011   1111
  1001   1001   1011   0100
-------------------------------
  1011   1110   0111   0011
```

2^{-8} to $2^{8}-1$

-128 127.