Sixth Semester B.E. Degree Examination, June/July.2014

Microprocessor

1. a. Determine the appropriate register/memory locations that are used to compute the 5 digit hex address when the processor needs to address the contents of

i) Data segment memory.

ii) Program segment memory.

iii) Stack segment memory.

iv) Extra segment memory. **(08 Marks)**

Ans:- The processor computes 5-digit physical address using the convention

PA = (Segment register)*10h+Pointer/Memory Location. or PA = Segment register * 10h offset .

Segment registers are CS(code segment), Stack segment(SS), Data segment(DS) & Extra segment(ES). The index registers are BX, Instruction pointer(IP), Stack pointer(SP), Base pointer(BP), Source index(SI), & Destination index(DI). These registers are 16 bits wide and effective address has only 16bits . But the address put on the address bus called the physical address must contain 20 bits. The extra 4 bits are obtained by adding the effective address to the contents of one of the segment register and Four '0' bits are appended to the right of the number in the segment register before addition is made.

(i) Data segment memory

PA=[DS]*10H+offset memory [offset→BX,DI/SI]

(ii) program segment memory

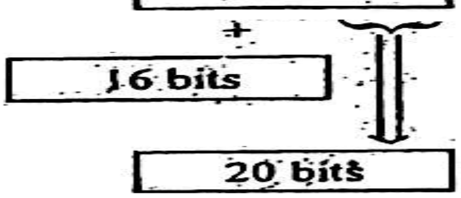
PA=[CS]*10H+(IP)

(iii) Stack segment memory

PA=[SS]*10H+[BP] [offset→SP,BP]

(iv)Extra segment memory

PA=[ES]*10H+offset from memory[offset → DI]



Ex : [CS] = 123A, [IP] = 341B.

341 B ——→ Offset (Ip)

123 A 0 ——→ [Segment address * 10H]

157 B B ——→ Physical address

b. **Explain the flag register of the processor in accordance with the respective bit positions.** **(05 Marks)**

c. **Write an 8086 assembly code to copy the contents of flag register into accumulator register following any arithmetic or logical operation.** **(07 Marks)**

Ans. Program to copy the contents of flag register into accumulation register following any arithmetic or logical operation.

```
. Model small
. Data
  d1 db 0A0H
  d2 db 0B0H
  memloc db  ?
. Code
Mov ax ,@data
Mov ds ,ax
Mov AL , d1
ADD AL ,d2          ; performing addition
mov memloc, al      ; store the result in some memory location
PUSHF               ; Push flag to top of stack
POP AX              ; transfer stack top to accumulator
MOV AH,4CH
INT 21H
```

2. a. **Explain the meaning of the following independent bits of 8086 assembly instruction templates: i) W-bit; ii) d-bit; iii) v-bit; iv) s-bit; v) z-bit.** (10 Marks)

Ans. The independent bit position of assembly instruction template has the following meaning,

(i) W bits :- when w=0, the operand in instruction is 8 bit long; when w=1, the operands are 16 bit long.

(ii) D bit :- when d=0, the register specified by REG filed is source, else it is destination.

(iii) V bit :-This is used in shift & rotate instruction to determine the number of shifts

If v=0, the shift count is 1.

if v=1 then the shift count is stored in CX register.

(iv) S bit :- This is called sign extension bit. The S bit appears with the W bit in the immediate to register/memory add, Subtract & compare instruction.

- 8 bit operation with 8 bit immediate operand is indicated by S=0 & w=0
- 16 bit operation with 16 bit immediate operand is indicated by S=0 & w=1
- 16 bit operation with sign extended immediate data is indicated by S=1 & w=1

(v) Z bit :- This is used by the REP instruction to control instructions. If z = 1, then instruction with REP. loop prefix is executed until the zero flag matches the Z bit.

b. **Write an optimum number of assembly instructions for the following objectives. Also indicate the type of addressing mode used in each case.**

i) Shift the contents of accumulator register 4 bits left.

ii) Rotate the contents of base register right by 2 bits.

iii) Divide the contents of accumulator register by 2.

iv) Multiply the contents of base register by 4.

v) If AL register contains a two digit BCD number, display the same on monitor using necessary DOS interrupts. (10 Marks)

v) If AL register contains a two digit BCD number, display the same on monitor using using necessary DOS interrupts. (10 Marks) arks)

. Ans. (i) Shift the contents of accumulator registers 4 bit lit

MOV CL 04H→immediate addr mode

SHL AX, CL →register addr mode

(ii) Rotate the contents of base register right by 2 bit

MOV CL,02H→immediate AM

ROR BX, CL→register AM

(iii) Divide the content of accumulator register by 2

MOV BL, 02H→immediate data AM

DIV BL →Register AM OR Shr al,01h

(iv) multiply the content of base register by 4

MOV AL,04H→immediate AM OR Shl al,02h

MUL BL →register AM

(v) If AL register contain a2 digit BCD number display the same on monitor using necessary DOS interrupt

Assume AL has 2 digit BCD number, say AL = 43 or AX = 0043h

```
XOR AH, AH        | ; Clearing AH AX = 0043H
MOV CL, 04H       | ; CL = 04 H
ROL AX,CL         | ; AX = 0430H
ROR AL, CL        | ; AX = 0403H
ADD AX, 3030H     | ; AX = 3433H
MOV DL, AH        | ; DL = 34H
MOV AH, 02H       | ; Display 4
INT 21H           |
MOV DL, AL        | ; DL = 33H
MOV AH, 02H       | Display 3
INT 21H           |
```

3. a. Consider that a symbolic memory address 'DISPTBL' contains a BCD to seven segment code starting from 4000 H to 400A H. Design an assembly code to meet the following objectives:

i) Send a message to screen 'PRESS ANY KEY 0 to 9

ii) Read the key pressed from the key board.

iii) If invalid kev is found. the program to loop back to step (i) with a suitable warning message.

iv) On correct key press, compute BCD to 7 segment code and store into memory location "DISPCODE'.

v) Use XLAT assembly instruction to achieve your objective

vi) Design a suitable flow diagram to show your approach. (10 Marks)

```
Ans.  . model small
      . data
      MSG 1 db 'PRESS ANY KEY 0 to9 $'
      MSG 2 db 'INVALID KEY $'
      DISPTBL db 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 6FH


  DISPCODE db?
  .Code
     Mov AX, @ data
     Mov DS, AX
 LEA BX, DISPTBL
 LEA DX, MSG 1
 Up : MOV AH, 09H
     INT 21H
    MOV AH, 01H
    int 21h
    CMP AL, 30h
   JNC L1
    LEA DX, MSG 2
   JMP UP
 I : SUB AL, 30H
   CMP AL, 0AH
   JNC UP
   XLAT
  MOV DISPCODE AL
  MOV AH, 4CH
  INT 21H
  END
```

b. i) Differentiate between the usage of assembler directives MACRO and PROCEDURE.
ii) Develop a suitable MASM code to display minimum of 3 different line text message by using MACRO directive and PRINTF as macro name. (10 Marks)

Ans. i)

MACRO	PROCEDURE
(1)Accessed using macro name during assembly by the assembler	(1)Accessed using CALL&RET mechanism during execution.
(2)Macros do not use the concept of stack	(2)They use stack mechanism
(3) Takes less time to execute since there is no transfer of control	(3)Takes more time to execute since the control has to be transferred to/ from procedure
(4) machine code is generated for instructions each time macro is called	(4)machine code is generated & placed in memory only once
(5)The size of the execution file is more	(5) The size of the file is less.
(6) Open subroutine because they expand at the point of call	(6) Closed subroutine because control goes to sub routine when call is made.
(7)parameters are passed as a part of the statement	(7)parameters can be passed using register, memory location

ii)

```
PRINTF MACRO MSG
MOV AH, 09H
```

```
LEA DX, MSG
INT 21H
ENDM
. model small
 data
  MSG1 DB 'Message 1 $', 10, 13
  MSG2 DB 'Message 2 $', 10, 13
  MSG3 DB 'Message 3 $', 10, 13
 CODE
 MOV AX, @ Data
 MOV DS, AX
 PRINTF MSG1
 PRINTF MSG2
 PRINTF MSG3
MOV AH, 4CH
INT 21H
END
```

4. a. **With reference to the internal architecture of 8086 processor, explair .**

i) The different eiternal souices external sources of hardware interrupts.

ii) How the processor checks to see an interrupt have been occurred.

iii) List of major actions performed to process an interrupt. (10 Marks)

Ans. **(i)An 8086 interrupt can come from any of the 3 source**

(a) An external signal applied to the NMI input pin. OR to INTR input pin.

INTR can be masked (disabled) so that it cannot cause an interrupt but NMI(non maskable interrupts) cannot be masked. If interrupt flag is cleared, then INTR input is disabled, when the 8086 is reset the interrupt flag is automatically cleared. Before the 8086 can respond to an interrupt signal on its INTR input, you have to set IF with an STI instruction.

(b) Execution of interrupt instruction INT, referred to as software interrupt.

The 8086 INT instruction can be used to cause the 8086 to do any one of the 256 possible interrupt type. eg int 21h is an interupt(s.W) to operating system DOS.

Ex : INT 32, will cause the 8086 to do a type 32 interrupt response.

(c) Error conditions produced by the execution of instructions such as divide by zero. If you attempt to divide an operand by zero, the 8086 will automatically interrupt the currently executing program.

(ii) How to processor checks to see an interrupt have been occurred.

At the end of each instruction cycle, the 8086 checks to see if any interrupts have been requested. If an interrupt has been requested, the 8086 responds to the interrupt by stepping through series of major actions.

(iii) List of major actions performed to process an interrupt

- It decrements the stack pointer by 2 & pushes the flag register on the stack
- It disables the 8086 INTR interrupts input by clearing the Interrupt flag (IF) in the flag register.
- It resets the trap flag (TF) in the flag register

- It decrements the stack pointer by 2 and pushes the current code segment register contents on the stack
- It decrements the stack pointer again by 2 and pushes the current instruction pointer contents on the stack.
- It does indirect far jump to the start of the procedure you wrote to respond to the interrupt.

b. **Explain the following internal interrupts generated within the processor while executing the program:**

i) TYPE - 0 divide by zero interrupt.

ii) TYPE -1 singlt step interrupt. **(10 Marks)**

Ans. i) TYPE - 0 divide by zero interrupt.

When an attempt is made to divide a 32 bit number or 16-bit number by zero, the result is infinity (undefined), which is too large to fit in AX and AL.

Whenever the quotient from a DIV or IDIV operation is too large to fit in the result registe:, the 8086 will automatically do a type 0 interrupt. Then 8086 stores flag register, clears IF and TF. It saves the return address on the stack,pushes CS value of the return address on the stack and then pushes IP. The 8086 then gets the starting address of the interrupt service procedure from the 0 location in the interruptvector table and loaded to CS, IP. 8086 Micro processor executes the procedure present in the new CS and IP address. At the end of interrupt service procedure, an IRET is used to return execution to the interrupted program INT 0 is a non maskable interrupt.

ii) TYPE -1 single step interrupt.

When you tell a system to single step, it will execute one instruction and stop. You can examine he contents of registers and memory locations. If they are correct you can tell the system tc go on and execute the next instruction.

If the 8086trap flag is set, the 8086 will automatically do a type 1 interrupt. It pushes the flag register on the stack, resets TF and IF and pushes CS and IP values of the next instruction on the stack. It then gets the CS value for the start of the type 1 interrupt service procedure 8086 has no instructions to directly set or reset the trap flag. The following code is used to achieve TF = 1

```
PUSH F                                Push flags on stack
MOV BP, SP                            copy SP to BP for use as index.
OR WORD PTR [BP + 0], 0100H; set TF bit
POP F                                 ; Restore flag register
```

The trap flag is reset when the 8086 does a type 1 interrupt, so the single step mode will be disabled during the interrupt service procedure.

PART - B

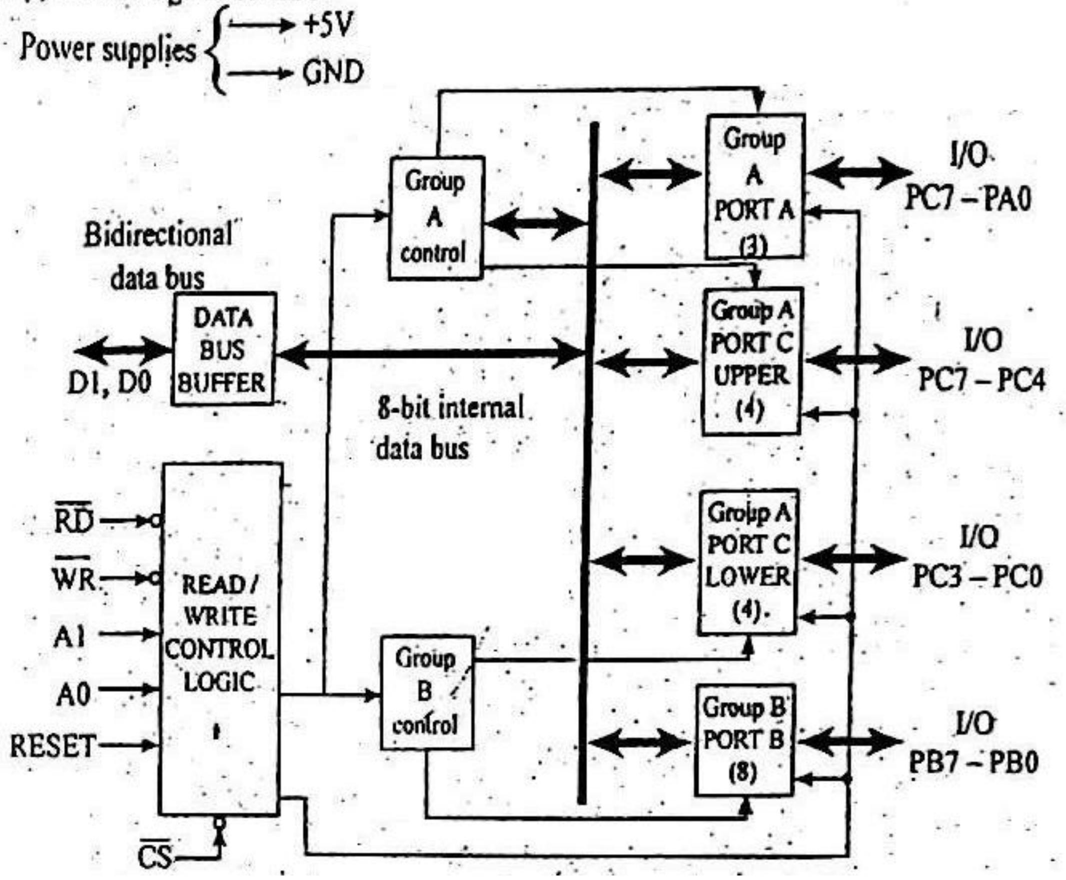
5. a. **With respect to programmable peripheral interface (PPI) 8255A:**

i) Draw a neat block schematic showing its functional description.

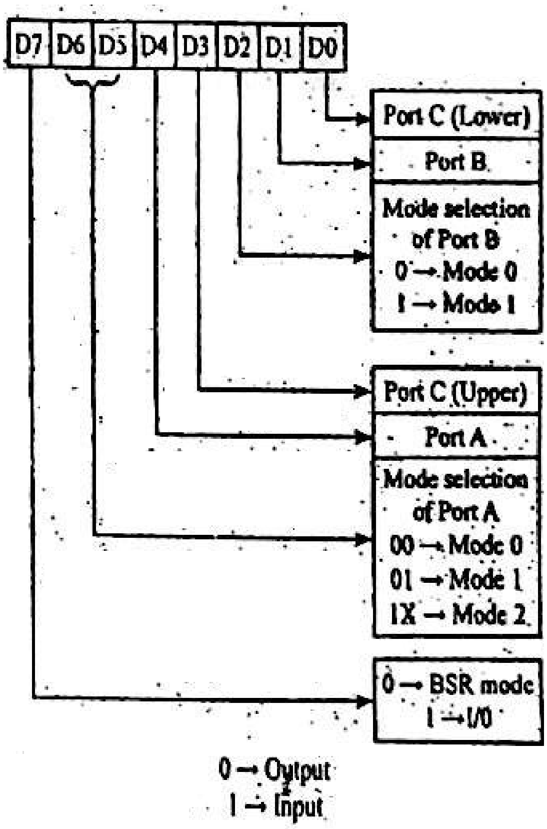
ii) Draw mode definition format the control word.

iii) Explain various possible modes of operation. **(10 Marks)**

Ans. (i) Block diagram of 8255



(ii) Control word



(iii) Various possible modes of operation.

Mode 0 : Basic I/0 :- Ports A and B are used as 2 simple 8-bit I/0 ports. Port C are 2 4-bit ports. The 2 halves of port C are independent, i.e., one half can be initialized as input and other half can be used as output.

Mode 1 ; Strobed I/0 :- In this mode, input or output data transfer is controlled by handshaking signals. Handshaking signals are used to transfer data between devices whose data transfer speeds are not same.

Some of the pins of port C function as handshakes lines.

- PC0, PC1 and PC2 function as handshakes lines for Port B
- PC3, PC4 and PC5 function as handshakes lines for Port A
- PC6, PC7 are available for use as input lines or output lines.

Mode 2 : Bi-directional bus

In mode 2, only port A can be used for bi-directional handshake data transfer.

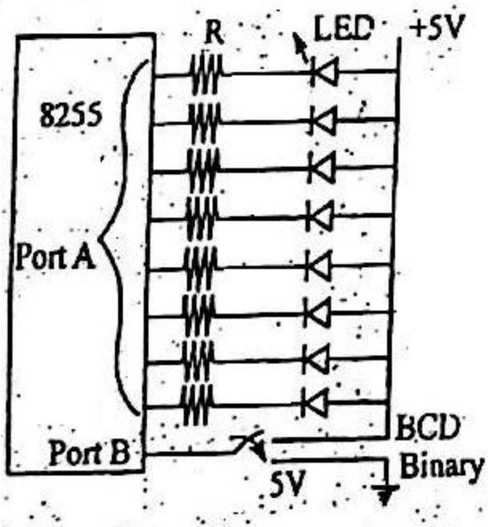
i.e., data can be output or input on the same eight lines.

PC3 to PC7 are used as handshake line for Port A. PC0 to PC2 can be used for I/0 if Port B is in mode 0.

PC0 to PC2 can be used for port B handshake lines if Port B is initialized in mode 1.

b. **Design an 8255 based event counting system. Port A is connected to 8 LEDs and Port B is connected to a toggling switch having 2 positions for binary and BCD. Draw the interfacing diagram and a program for binary or BCD count as selected by switch. Given that the control port address is 50B3, assume safe current to glow each LED is 25mA. A suitable delay between counts is considered. (10 Marks)**

Ans. Circuit Diagram



Given $I = 25\text{mA}$

$$\therefore \frac{V}{I} = \frac{5V}{25\text{mA}} = 200\Omega.$$

$V = IR, R = \frac{V}{I}$

Program

```
. model small
. data
  PA equ 50B0H
  PB equ 50B1H
  CR equ 50B3H
  Code
  MOV AX, @ data
  MOV DS, AX
  MOV AL, 82H                 ; PA – o/p
  MOV DX, CR                  ; PB – i/p, Hence control word = 82H in AL
  OUT DX, AL
  MOV CL, 00
  MOV DX, PB
L1 : IN AL, DX
  OR AL, AL
   JZ L1
  MOV AL, CL
  MOV DX, PA
  OUT DX, AL
  CALL Delay
  ADD AL, 01
  DAA
  MOV CL, AL
  CMP AL, FFH
  JNZ L1
  MOV CL, 00H
  JMP L1
 delay proc near
  Push BX
  Push CX
  MOV BX, 0FFFFH
L3 : MOV CX, 0FFFFH
L2 : Loop L2
  Dec BX
  JNZ L3
  RET
delay ENDP
MOV AH, 4CH
INT 21h
```

6. a. What is meant by numeric data processor 8087 (NDP)? What are the benefits of interfacing the same with the host processor? (04 Marks)

Ans. 8087 processor is an extension of 86/88 processor architecture.

- It can operate on data of type integer, decimal and real types, with lengths ranging from 2 to 10 bytes.
- Numeric co processor is designed to execute various forms of addition and subtraction,

square root, exponential, tangent etc.

- It follows IEEE floating point standard (Real numbers)

When we interface 8087 with the host processor, it dramatically increases the processing speed of application which utilize mathematical operations such as CAM, numeric controller, CAD, Graphics etc. It is also called co-processor since it helps the main processor for doing specific operations.

b. Explain briefly the role played by the following pins of 8087 during interaction.

i) Bus high enable ($\overline{BHE}$ / S7).

ii) Status pins $\left(\overline{s2},\overline{s1},\overline{s0}\right)$.

iii) Request / Grant $\left(\overline{RQ}/\overline{GT}\right)$. (06 Marks)

Ans. **(i) $\overline{BHE}$/S7** :- During T1, $\overline{BHE}$ is used to enable data on to the most significant half of the data bus, pins D15 – D8. If $\overline{BHE}$ = 0 in T1 AD_0 - AD_{15} carry address on A0-A15 lines.

(ii) Status pins ($\overline{S2}$, $\overline{S1}$, $\overline{S0}$)

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$ → Status pins
0	X	X → unused
1	0	0 → unused
1	0	1 → Read memory
1	1	0 → Write memory
1	1	1 → Passive

(iii) Request / Grant ($\overline{RQ}$/$\overline{GT}$)

It is used by 8087 to gain control of the local bus from the CPU for operand transfer or on behalf of another bus master.

It is used for inputting bus requests and outputting bus grants.

c. Consider the given decimal number 178.625 convert it into

i) Short – real format (Single precision representation)

ii) Long – real format (double precision representation) **(04 Marks)**

Ans. $(178.625)_{10}$ → Short – real format (Single precision representation)

Step 1

Convert d cimal number to binary

$$\begin{array}{r|c|l} 16 & 178 & 2 \\ \hline & 11 & 11 \text{ (remainders)} \end{array}$$

Take Remainders

$$\begin{array}{ccc} = & 11 & 2 \\ & \downarrow & \downarrow \end{array}$$

⇒ 1011 0010

$0.625 \times 2 = 1.25 \Rightarrow 1$

$0.25 \times 2 = 0.5 \Rightarrow 0$

$0.5 \times 2 = 1.0 \Rightarrow 1$

⇒ 0.101

Step 2

Binary number = 10110010.101

Normalize the binary number by moving the binary point until it is between the first and second bits.

$\therefore 1.0110010101 \times 2^7$

Bias for short real format is 127

Here S = 0 Since given number is positive

E = 127 + 7

= 134 = 10000110

E = 0110010101

$\therefore$ In short real format(1) (8) (23)

S	E	F
0	10000110	01100101010000000000000

4 3 3 2 A 0 0 0

ii) Long real format (Double precission 64 bit)

Bias for long real format is 1023

After converting decimal number to binary we get

10110010.101

After normalizing

1.0110010101×2^7

S = 0

E = 1023 + 7 = 1030

= 10000000110

F = 0110010101

$\therefore$ In long real format

S	E	F
0	10000000110	0110010101 00.......................0

63 62 52 51

4 0 6 6 5 4 0000000000

Answer is 4066540000000000H

d. Write a program to calculate the volume of a sphere having radius of the sphere is specified. The result is to be stored in the memory location VOLUME. Volume of a sphere is given by (4/3) * (PI) * (r**3). **(06 Marks)**

Ans. Program for volume calculation

```
. Data
Radius dd 5.0233
Const Equ 1.33
Volume dd 01 dup(?)
MOV AX, @ data
MOV DS, AX
FINIT
FLD Radius
FST ST(4)
FMUL. ST(4)


FMUL ST(4)
FLD const
FMUL
FLD PI
FMUL
FST volume
RET P
```

7. a. Draw a schematic diagram when 8086 processor is operating in maximum mode configuration. (06 Marks)

b. Explain the function performed by pins exclusive for minimum mode configuration. i) HOLD and HLDA; ii) M/$\overline{IO}$; iii) $\overline{RD}$; iv) $\overline{WR}$; v) MN/$\overline{MX}$ (08 Marks)

Ans. **i) HOLD and HLDA :** HOLD indicate that another master is requesting a local bus. The 8086 will not gain control of the bus until this signal is dropped HLDA is Hold Acknowledge. This pin outputs a bus grant to a requesting master.

ii) M/$\overline{IO}$: It is used to distinguish between memory access and I/0 access. For a memory access, it is 1.

iii) $\overline{RD}$: This indicates that the processor is performing memory or I/0 read cycle depending on the state of $\overline{S2}$.

iv) $\overline{WR}$: when 0, it indicate a write operation is being performed. This operation depends on state of M / $\overline{IO}$ signal

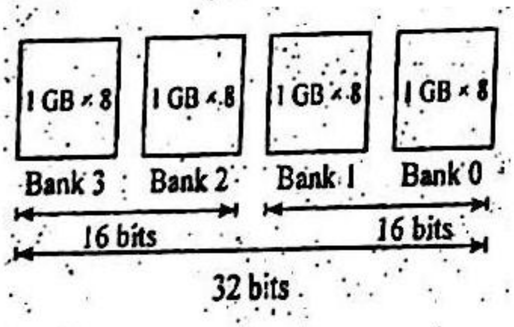
v) MN/$\overline{MX}$: Indicates the mode of the processor

If MN / $\overline{MX}$ = 1, Minimum mode

MN / $\overline{MX}$ = 0, maximum mode

8. a. Explain the memory bank system architecture for the 80386DX microprocessor with a block schematic. Explain how interleaved memory system is used for speed improvement. (10 Marks)

Ans. The memory bank system for 80386 DX up.



- The physical memory system of the 80386 DX is 4GB is size. The memory is divided into Four 8-bit wide memory banks each containing upto 1GB of memory.

- This 32–bit wide memory organization allows bytes, words or double words of memory data to accessed directly. The 80386DX transfers upto a 32-bit wide numbers in a single memory cycle with memory bytes numbered from location 00000000H –FFFFFFFFFFH
- In 80386 DX, the memory banks are accessed via four bank enable signal $\overline{BE}_3 - \overline{BE}_0$
- This arrangement allows a single byte to be accessed when one bank enable signal is activated by the microprocessor. It also allows a word to be addressed when 2 bank enable signals are activated. A word is addressed/ accessed in bank 0 and 1 or in bank 2 and 3.

interleaving :- An interleaved memory system is a method of improving the speed of a system. An interleaved memory system requires 2 or more complete sets of address buses and a controller that provides addresses for each bus. While the up accesses locations 000000H – 000001H, the interleave control logic generates the address store signal for locations 000002H – 000003H. This selects and accesses the word at the locations 000002H –000003H. This process alternates memory sections. thus increasing the berformance of the memory system.

b. Draw the block schematic of the control register of 80836 microprocessor and explain the following special control bits of operation i) PG; ii) ET; iii) TS; iv) EM; v) MP; vi) PE. (10 Marks)

Ans. Control register of 80386 Microprocessor

PG	000000000000000	000000000000	ET	TS	EM	MP	PE	CR0
Not used								CR1
Page fault linear address								CR2
Page directory base			000000000000					CR3

- Control register O(CRO) is identical to the MSW (Machine status word) in 80286 up.
- CR1 Is not used in 80386, but is reserved for future products.
- CR2 holds the linear page address of the last page accused before a page fault interrupt.
- CR3 holds the base address of the page directory.

CR0 contains a number of special control bits

i) PG – Selects page table translation of linear addresses into physical addresses when PG = 1. Page table translation allows any linear address to be assigned any physical memory location.

ii) ET – When ET = 0 → Selects the 80827 coprocessor

When ET = 1 → 80387 coprocessor is selected.

iii) TS – Indicates flat the 80386 has switched tasks. If TS = 1, a numeric coprocessor instruction causes a type 7 interrupt.

iv) EM – Is set to cause a type 7 interrupt for each ESC instruction. We use this interrupt to emulate, with software, the function of the up.

v) MP – Is set to indicate that the arithmetic co-processor *is present in the system.*

vi) PE – Is set to select the protected mode of operation *for the 80286 and cleared to re*-enter the real mode. In 80286, this bit can only be set in the. *80286*. The 80286 could not return to real mode without a hardware reset.