Sixth Semester B.E. Degree Examination, Dec.2013/Jan.2014

Microprocessors

Time : 3 hrs. Max. Marks: 100

Notes: *Answer any FIVE full questions, selecting at least TWO questions from each part.*

PART – A

1. a. Explain with block diagram the personal computer model showing address, data and control bus structure. (5M)

Ans: Please refer answer of Q.No. 1(a) of June–2012.

b. With a neat sketch, explain the execution unit and bus interface unit of the 8086 microprocessor. (10M)

Ans: Please refer answer of Q.No. 1(a) of December–2011.

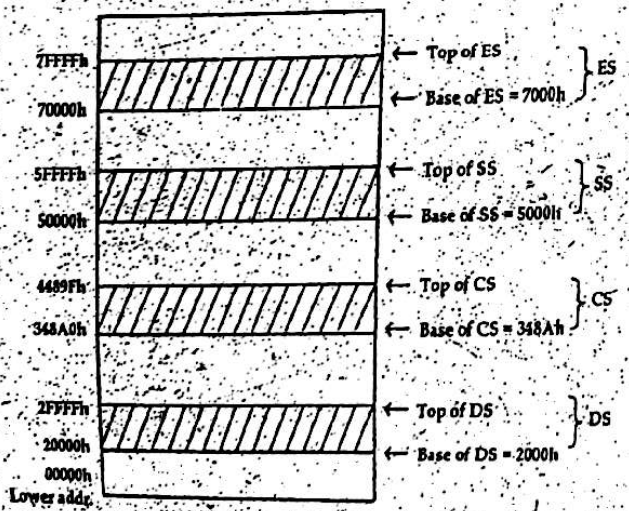
c. Explain segmentation in 8086 and advantages of using segment registers.(5M)

Ans: **Segmented Memory (Without overlapping)**

The total memory capacity of 8086 is 1 MB and is further divided into 16 segments each of 64K, to enable programmer to access 1 MB memory using any general purpose registers whose capacity is 16 bit.

Advantages of segmentation.

1. It allows to access the memory (of) upto 1 MB even though the address associated with each instruction is only 16 bit.
2. Facilitates use of separate memory areas for program (code), data and stack.



3. Permits the program or its data to be put in different areas of memory, each time the program is executed that is the program can be relocated which is very useful in multiprogramming.

Note: 1. The segment can start/begin at any memory location which is divisible by 16.

2. The 4 segments can overlap for small programs.

Segment can be overlapping as shown below. Suppose a segment starts at a particular address and its maximum size can be 64 K bytes. But if another segment starts before 64 K bytes the two segments are said to be overlapping.

The area of memory from start of 2nd segment to end of 1st segment is called overlapped segment area.

Default and alternate registers for various segments and method of finding physical address.

Type of Memory Reference	Default Segment	Alternate Segment	Offset (Logical Address)
Instruction fetch	CS	None	IP
Stack operation	SS	None	SP, BP
General data	DS	CS, ES, SS	Effective address
String source	DS	CS, ES, SS	SI
String destination	ES	None	DI
BX used as pointer	DS	CS, ES, SS	Effective Address
BP used as pointer	SS	CS, ES, DS	Effective Address

For the following examples we have assumed the contents as under
CS = 2000H, DS = 3000H,
SS = 4000H, ES = 5000H,
BP = 0010H
BX = 0020H, SP = 0030H,
SI = 0040H, DI = 0050H

2. a. Explain the different string instructions of the 8086. (8M)

Ans: Please refer answer of Q.No. 4(c) of June–2012.

b. What are assembler directives? Explain the following:
(i) total db 00h (ii) inc word ptr [si]
(iii) mov dx, offset msg (iv) assume (8M)

Ans: It is a direction to processor. It is also called pseudo code. No opcode will be generated.

(i) Total db 00h : declar total a variable as db(define byte) and initialised to zero

(ii) INC word ptr [SI]: Increment pointer SI by Two. Since it is pointing to a word.

(iii) mov dx, offset msg : Offset address of msg is moved to dx that is starting address of msg declared in data segment (ds) is initialised to dx for display.

(iv) Assume: Please refer answer of Q.No. 3(a) of July–2013.

c. Explain:
(i) MN/$\overline{MX}$ (ii) $AD_{15} - AD_0$ (iii) $\overline{RD}$ (iv) $\overline{WR}$ (4M)

Ans: (i) MN/$\overline{MX}$: Minimum/MAX mode if it is 1=MIN mode, 0-MAX mode.

(ii) AD_{15} – AD_0 : These pins act as address bus during first part of Maching cycle ie., T1 state and data bus during remaining part of M/C byte is ie., onwards. These are lower ordered address/data bus and are multiplexed.

(iii) $\overline{RD}$: It is low active o/p pin. It is low whenever microprocessor reads the data from memory or I/O devices. To read the data RD should be low. (T_2 & T_3 states).

(iv) : It is o/p low active signal. It indicates microprocessor is writing the data to memory or I/O device.

3. a. Write a display macro using for statement to display 'VTU' on the screen.(5M)

Ans: Please refer answer of Q.No. 3(a) of December-2011.

b. Write an assembly language program to arrange '10' bytes of data in descending order. (10M)

Ans: Please refer answer of Q.No. 2(a) of July-2013 and declare 10 numbers.

c. Differentiate between macros and procedures. (5M)

Ans: Please refer answer of Q.No. 4(c) of July-2013.

4. a. Draw the 8086 interrupt pointer table and explain the dedicated interrupt pointers, reserved interrupt pointers and available interrupt pointers. (10M)

Ans: Please refer answer of Q.No. 4(b) of July-2012.

b. Explain the priority of 8086 interrupts. (5M)

Ans: Please refer answer of Q.No. 4(b) and 4(c) (fo NMI, INTR) of December-2011.

c. Write a program to check if a given byte is bitwise palindrome. (5M)

Ans: BIT - WISE PALINDROME

```
DISP MACRO MSG
        MOV   AH, 09H
        LEA   DX, MSG
        INT   21H
ENDM
.MODEL SMALL
.DATA
        NUM   DW 8421H
        MSG1  DB 13, 10, 'THE NUMBER IS BITWISE PALIN' , '$'
        MSG2  DB 'THE NUMBER IS NOT BITWISE PALIN' , '$'
.CODE
        MOV   AX, @DATA
        MOV   DS, AX
        MOV   AX, NUM
        MOV   CL, 10H
```



```
        MOV   BX, 00H
        CLC
BACK :  ROL   AX, 1      ; Rotate AX left without carry
        RCR   BX, 1      ; Rotate right BX with carry
        DEC   CL
        JNZ   BACK
        CMP   AX, BX
        JZ    PALIN
        DISP  MSG2
        JMP   LAST
PALIN : DISP  MSG1
LAST :  MOV   AH, 4CH
        INT   21H
    END
```

PART - B

5. a. Explain the different key switches used on keyboards. (8M)

Ans: Please refer answer of Q.No. 5(c) of December-2011.

b. Explain the detection of matrix keyboard, key press, debouncing and encoding with a microcomputer using 4*4 keyboard. Also draw the flowchart for the same. (12M)

Ans: Please refer answer of Q.No. 5(b) of December-2012.

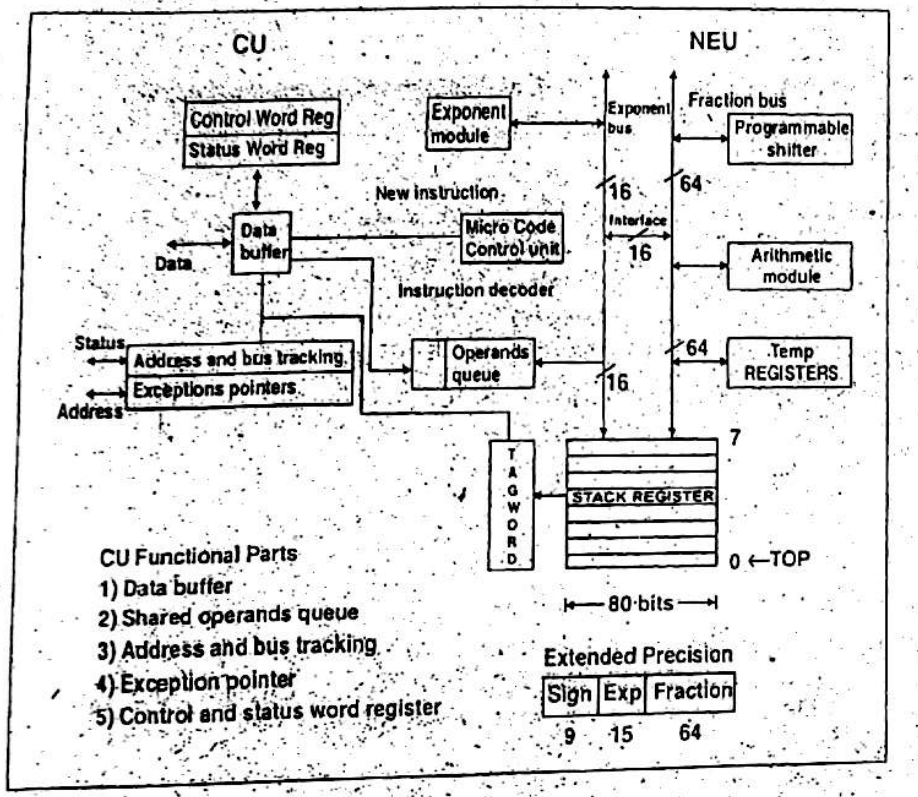
6. a. Explain the 8087 architecture. Also explain the bit pattern of status register and control register. (12M)

Ans: 80x87 ARCHITECTURE - BLOCK DIAGRAM

The architecture is divided as control unit (CU) and the numeric extension unit (NEU). The NEU executes all the numeric instructions while the CU receives, decodes, reads and writes memory operands and executes 8087 control instruction and also maintains parallel queue similar to give in main cpu. The main job of CU is to establish communication between the CPU and memory. The CPU while fetching the instructions monitors the databus to check for 8087 instructions. Also it monitors $\overline{BHE}$/S7 line to detect CPU type ie. 8086 or 8088 and adjust the queue length. The CPU identifies the coprocessor instructions using the ESCAPE code bits and triggers the execution of 8087.

If the instruction requires a memory operand to be fetched. Then the physical address of the operand is calculated and 8087 reads the operand and proceeds for executes. If the coprocessor instruction does not require any memory operand, then it is directly executed. If 8087 is ready with result, the CU gets the control of the bus from 8086 and executes a write cycle to write the result in the memory.

The NEU executes all the instructions like arithmetic, logical and date transfer instruction.



NEU Functional Parts

1) 8, Eighty bit register stack ST0 is top (LIFO to store operands,) SP holds number of register currently assessed. SP initialized to 00 on reset. It is a circular stack. It holds operands for arithmetic instructions and result.
2) Micro control unit
3) Exponent Module
4) Programmable shifter for bit shifting
5) Arithmetic module for arithmetic operations.
6) Temp register used for swaping
7) Shared operand queue to put instruction in FIFO queue.

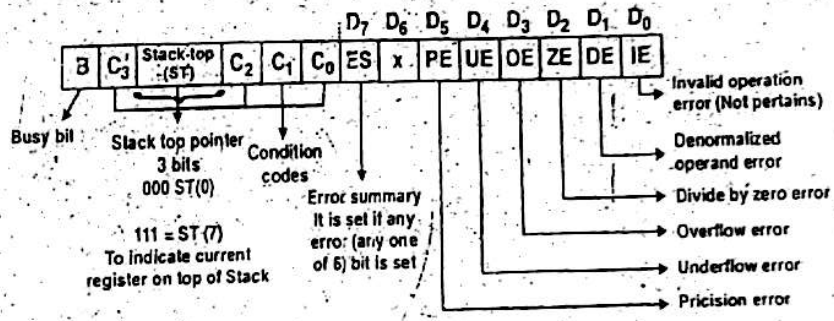
The data bus width is 80 bits (1 sign, 15 exponent and 64 fraction) while NEU begins execution it pulls up the BUSY signal and BUSY signal is connected to $\overline{TEST}$ input of 8086. This makes 8086 to wait till BUSY PIN of 8087 or $\overline{TEST}$ pin of 8086 goes low. It also contains programmable shiftes for shifting the operands during execution of instructions like FMUL and FDIV.

Stack register contains 8 registers of 80 bit wide. These registers always stores extended (80 bit) precision floating point number.

The data resides in memory in normal format. The co-processor converts it to extended precision from, when data is moved from memory to stack register of coprocessor.

Control and Status Word Register Format

1) Status Word Register : It is a 16 bit register. It stores status like errors if any (6 types) summary of errors (ES) condition of compare instruction ($C_3 - C_0$), address of current register on top of stack and is 8087 busy or not.



Condition codes to indicate condition of compare st with source operand

Funetion	C3	C2	C1	C0
[St] > source	0	0	0	0
[St] < source	0	0	0	1
[St] = source	1	0	0	0
St cannot be compared	1	1	1	1

IE - Invalid operation error like stack overflow, underflow, divide by zero etc.,

DE - One of the operand is denormalized

ZE - Zero error that is divisor (Nr) is zero while dividend (dr) is non-zero number.

OE - Over flow error that is result is too large to be represented with current precision selected.

UE - Under flow error that is a non-zero result that is too small to represent with current precision selected by control word.

PE - The result exceeds the selected precision (single, double, extended etc.,)

ES - Summary of error. It is set when any one error bit is set.

b. Explain:

(i) FLDZ (ii) FLD1 (iii) FLDPI (iv) FLDL2E (8M)

Ans: (i) FLDZ: Push zero to ST

This pushes 0.0 on to stack, after SP is decremented by 1

SP ← SP – 1.

(SP) ← 0.0

(ii) FLD1: Push one to St.

This pushes 1.0 on to stack, after SP is decremented by 1

SP ← SP – 1

(SP) ← 1.0

(iii) FLDPI: Push PI that is 22/7 on to stack after decrementing SP by one.

SP ← SP – 1

(SP) ← 22/7

(iv) FLDL2E: Push log 2^e on to stack after decrementing SP by one.

SP ← SP – 1

(SP) ← log 2e

7. a. Write a note on parallel printer interface (LPT). (10M)

Ans: Please refer answer of Q.No. 7(c) of July–2013.

b. Explain the write cycle timing diagram for minimum mode. (7M)

Ans: Please refer answer of Q.No. 7(b) of December–2011 and change $\overline{WR}$ TO $\overline{RD}$.

c. Explain the following:

(i) $M/\overline{IO}$ (ii) ALE (iii) $\overline{INTA}$ (3M)

Ans: (i) $M/\overline{IO}$ (PIN 28): It is used to indicate whether data transfer is to memory or I/O device. If it is high it is memory communication if it is low I/O device communication.

(ii) ALE (PIN 25): Address latch enable.

It is an O/P high active PIN.

It is to demultiplex the $AD_0 - AD_{15}$ PINS to $A_0 - A_{15}$ and $D_0 - D_{15}$ to carry address during first cycle and data in subsequent cycle.

(iii) $\overline{INTA}$: Interrupt acknowledge : It is low active interrupt acknowledge o/p pin. It indicates the recognision of an interrupt request by issuing the acknowledgement.

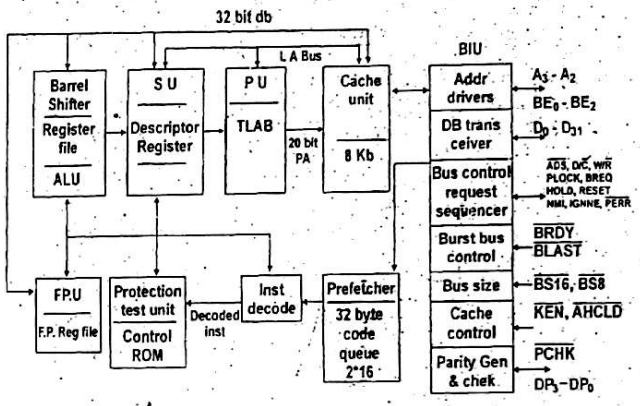
8. a. Draw the internal programming model of the 80486 and explain. (10M)

Ans: 80486 DX Architecture

It has 3 blocks BIU, EU and FPU. It has math coprocessor and 8K byte level-1 cache memory (internal). It also has 8 GPRS or 32 bit (EAX, EBX, ECX, EDX, EBP, EDI, ESI & ESP) can work as 8, 16 or 32 bit. It also has segment register CS, DS, ES, GS, FS each 16 bit wide. It also has global, local, and interrupt descriptor table and MMU as in 80386. It has EFLAG with one extra flag bit alignment check-AC (at D18 position) used to indicate microprocessor has accessed a misaligned address that is word at an odd address or dw at non-double word boundary. (Instead of LSB at even MSB at odd, it is LSB at odd, MSB at EVEN).

In real mode the interrupt structure is same as 8086 ie. 256 × 4 = 1K. In protected

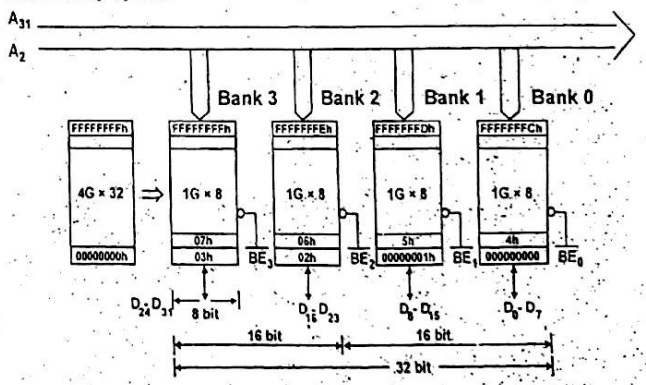
Architecture of 80486.



mode interrupts are handled by interrupt descriptor table which requires 256 × 8 = 2K bytes.

b. Explain the memory system of 80386. (5M)

Ans: Memory System:



80386 has 32 address/data lines (2^{32} = 4Gb, 2^{30} = 1GB)

The physical memory, 4GB of 80386 is used as 4 banks of 1GB. If virtual addressing is used 64Tb (2^{46}) are mapped to 4GB by Memory Management Unit (1GB contains 16Tb addresses.)