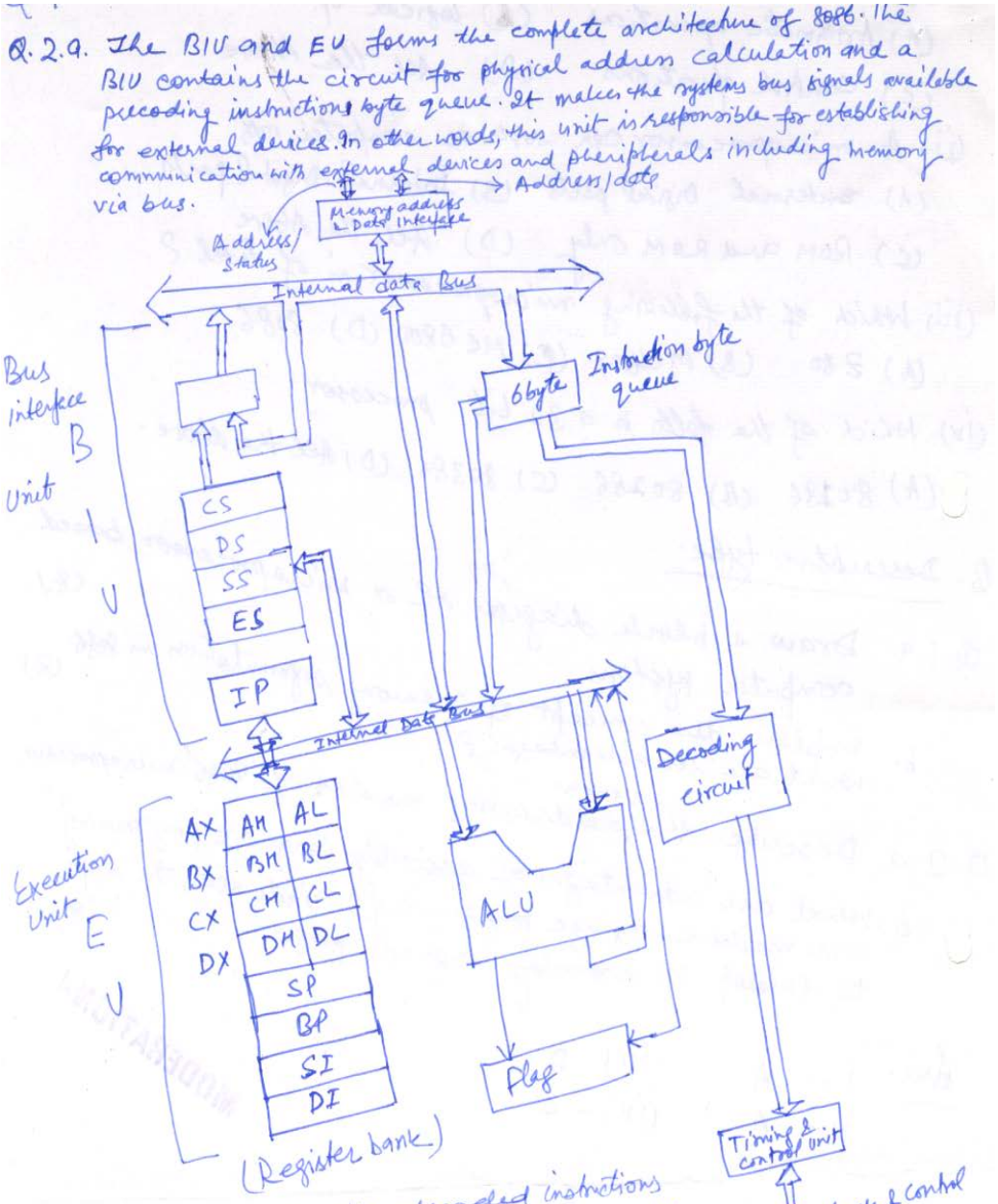
Q.2a. Explain BIU and EU of 8086 microprocessor.

Q.2.a. The BIU and EU forms the complete architecture of 8086. The BIU contains the circuit for physical address calculation and a decoding instructions byte queue. It makes the systems bus signals available for external devices. In other words, this unit is responsible for establishing communication with external devices and peripherals including memory via bus.



The EU executes the decoded instructions concurrently. The BIU along with EU forms a pipeline. The BIU manages the complete interface of EU with memory and I/O devices under the control of timing and control unit.

b.Describe the various segments register of 8086 microprocessor.

2b. There are four segment registers

1. Code segment register (CS)
2. Data segment register (DS)
3. Extra segment register (ES)
4. Stack segment register (SS)

The code segment register points at the segment containing the current program. Data segment points at the segment where variables are defined. ES is upto a coder to define its usage and SS points at the segment containing the stack.

c. Generate the machine code for MOV CH, BL using instruction templates.

2c. The 6 bit code for this instruction is 100010. D = 0 for BL is a source operand, W = 0 byte operation

In byte 2, since the second operand is a register MOD field is 11.

R/M field = 101 (CH)

REG field = 011 (BL)

∴ Machine code = 1000 1000 (Byte 1) 11 011 101 (Byte 2) = 88DDH.

Q.3a. Explain the following instructions of 8086 microprocessor:
(i) XLAT (ii) XCHG (iii) DAA (iv) STD (v) LOCK

Q.3 a. i) XLAT → Translate instruction is used for finding out the codes in case of code conversion problems using look up table technique.

(ii) XCHG → This instruction exchanges the contents of specified source and destination operands, which may be registers or one of them may be a memory location. However, exchange of contents of two memory location is not permitted.

(iii) DAA → Decimal Adjust Accumulator. It is used to convert the result of addition of two packed BCD numbers to a valid BCD number.

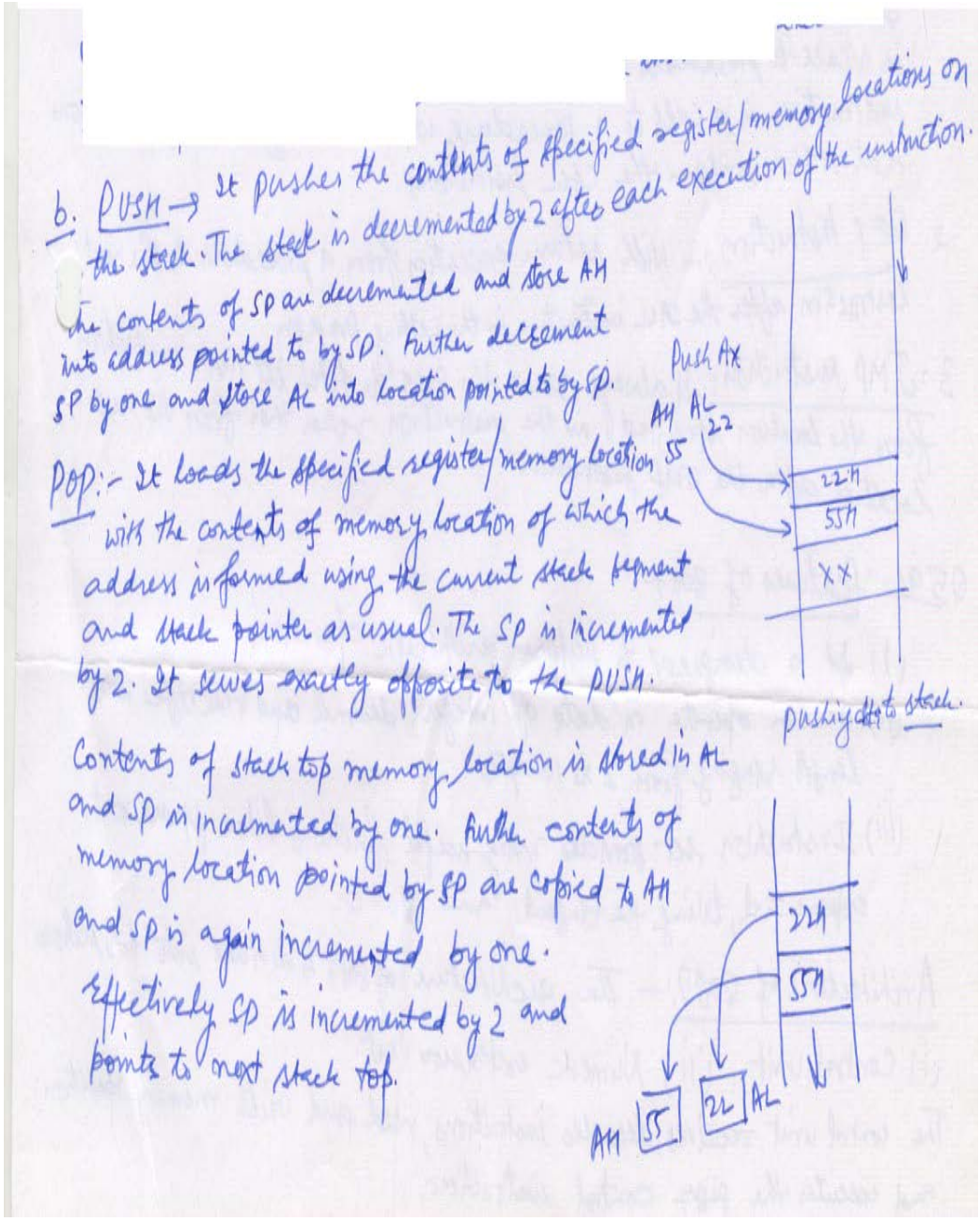
(iv) STD: → Set direction. This instruction modify direction flag.

v LOCK → Bus lock instruction prefix. When it is executed, the bus access is not allowed for another master till the lock prefixed instruction is executed.

b.Explain PUSH and POP instructions in detail.

b. PUSH → It pushes the contents of specified register/memory locations on the stack. The stack is decremented by 2 after each execution of the instruction. The contents of SP are decremented and store AH into address pointed to by SP. Further decrement SP by one and store AL into location pointed to by SP.

POP:- It loads the specified register/memory location with the contents of memory location of which the address is formed using the current stack segment and stack pointer as usual. The SP is incremented by 2. It serves exactly opposite to the PUSH. Contents of stack top memory location is stored in AL and SP is incremented by one. Further contents of memory location pointed by SP are copied to AH and SP is again incremented by one. Effectively SP is incremented by 2 and points to next stack top.



Q.4a. Describe various software interrupts of 8086 in brief. Give the sequence of interrupt execution.

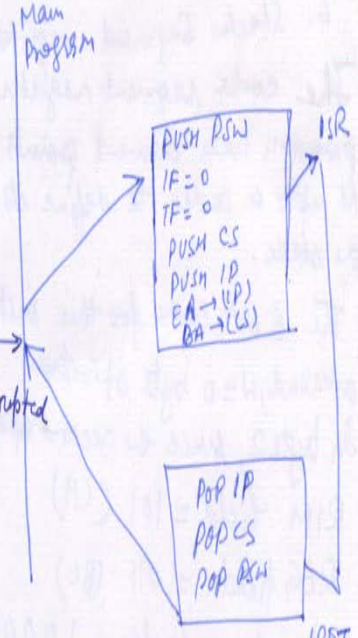
Q.4 a. There are 256 software interrupts i.e INT 0 to INT 255; out of which (i) INT 0 – INT 4 are reserved for dedicated operations (ii) INT 5 – INT 31 are reserved for particular operation and (iii) INT 32 – INT 255 are used by user.

Sequence of interrupt Execution in 8086:-

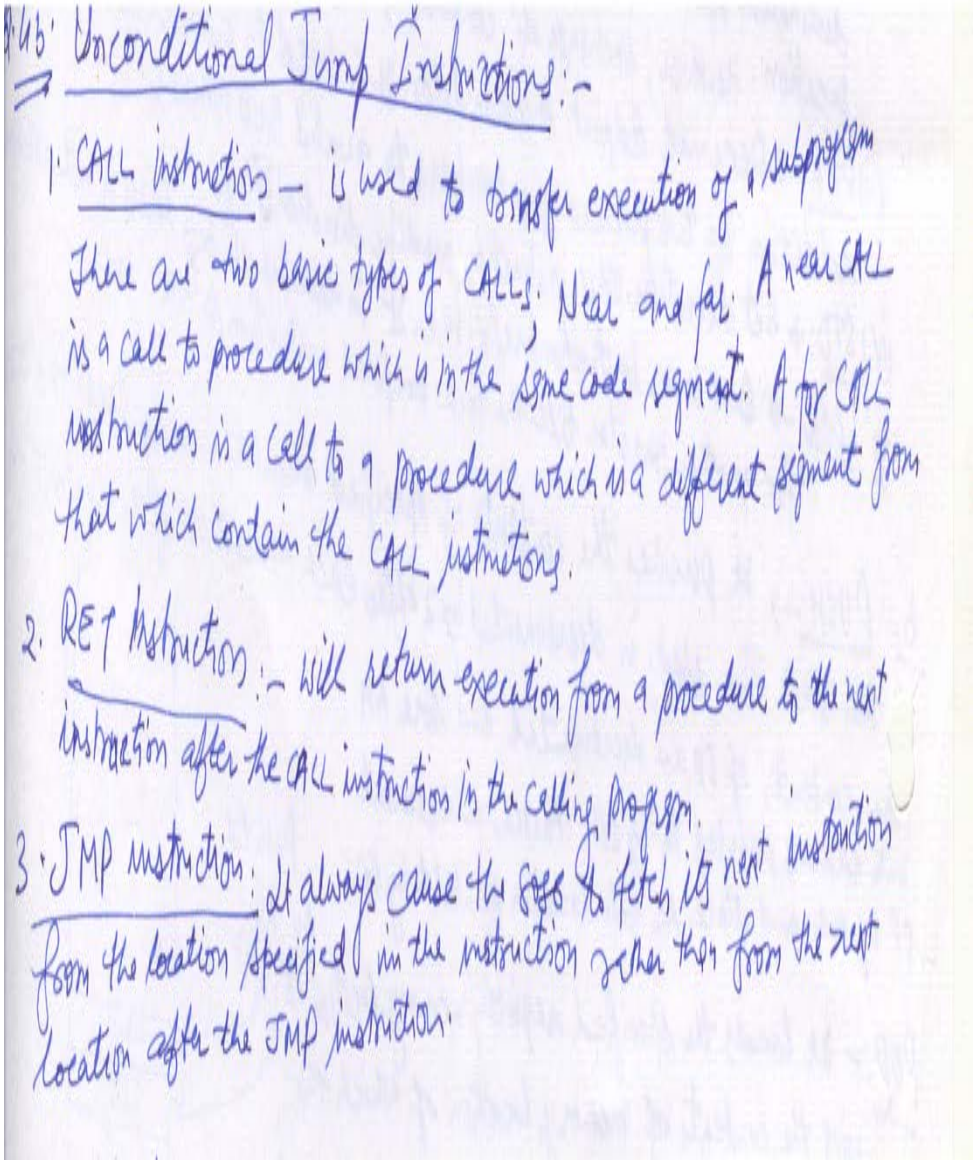
1. Push PSW — Flag Saved
2. IF = 0 — INTR disabled
3. TF = 0 — Microprocessor will not execute ISR by single stepping
4. PUSH CS — Address of next instruction of main program saved in stack.
5. Depending upon the value of n microprocessor will calculate the physical address of memory location. nxi
6. The last instruction of every ISR will be RET (Interrupt return). After executing IRET instruction microprocessing performs

   POP IP
   POP CS
   POP PSW

   The microprocessor returns back from ISR to main program instruction.
7. Status of flag remains unchanged.



b. Explain the unconditional jump instructions of 8086.



Ans: Unconditional Jump Instructions:-

1. CALL instruction - is used to transfer execution of a subprogram. There are two basic types of CALLs: Near and far. A near CALL is a call to procedure which is in the same code segment. A far CALL instruction is a call to a procedure which is a different segment from that which contain the CALL instructions.

2. RET instruction:- will return execution from a procedure to the next instruction after the CALL instruction in the calling program.

3. JMP instruction: It always cause the 8086 to fetch its next instruction from the location specified in the instruction rather than from the next location after the JMP instruction.

Q.5 a. Explain the features and architecture of 8087. (10)

Q5a: Features of 8087

(i) It is designed to perform arithmetic operation.

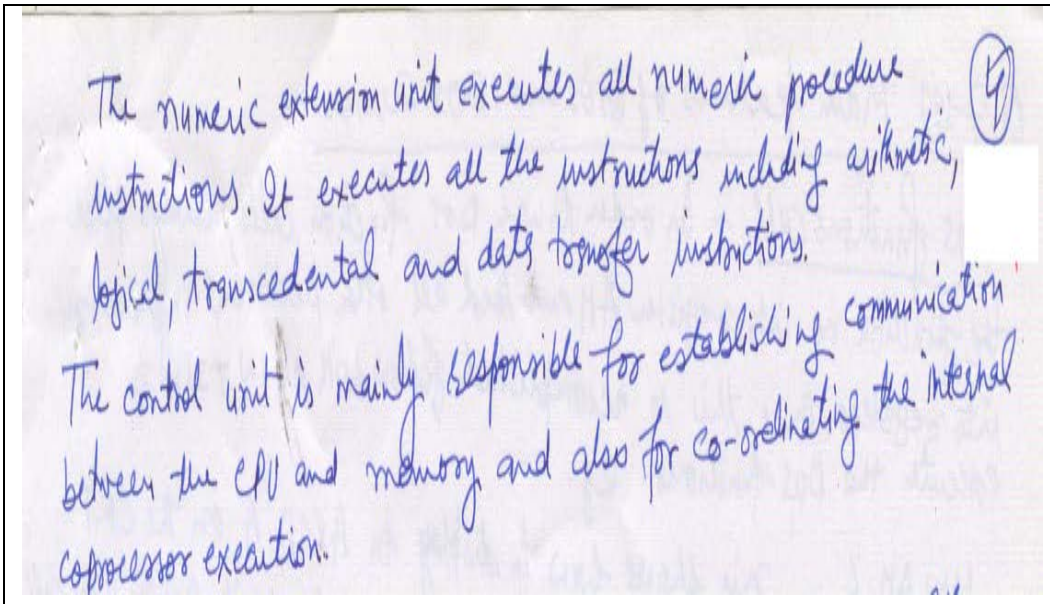
(ii) It can operate on data of integer, decimal and real types with length ranging from 2 to 10 bytes

(iii) Instruction set provides many useful functions like square root, exponential, taking the tangent and so-on.

Architecture of 8087:- The architecture of 8087 is divided into two sections

(i) Control unit (ii) Numeric extension unit.

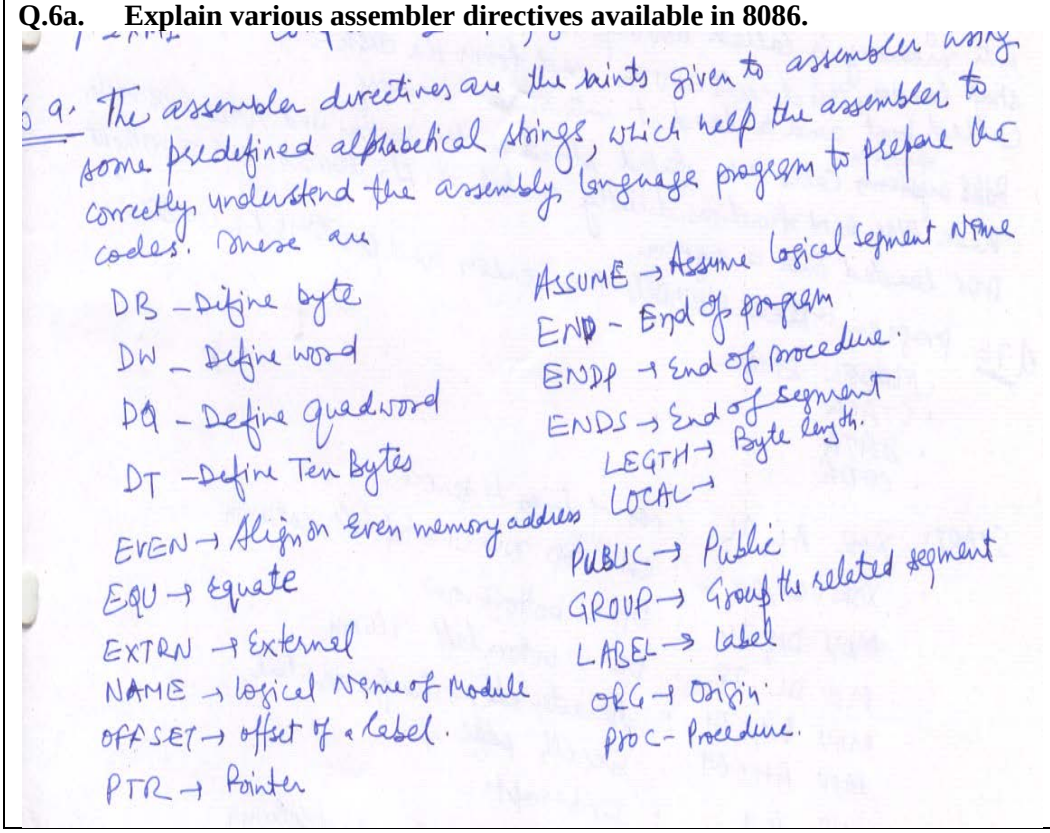
The control unit receives, decodes instructions, read and write memory operation. and executes the 8087 control instructions.


The numeric extension unit executes all numeric procedure instructions. It executes all the instructions including arithmetic, logical, transcedental and data transfer instructions.
The control unit is mainly responsible for establishing communication between the CPU and memory and also for co-ordinating the internal coprocessor execution.

b.Explain the various data transfer instructions of 8087.

Ans 5b. Text Book 1 – 13.2, Pg. No.-234-235

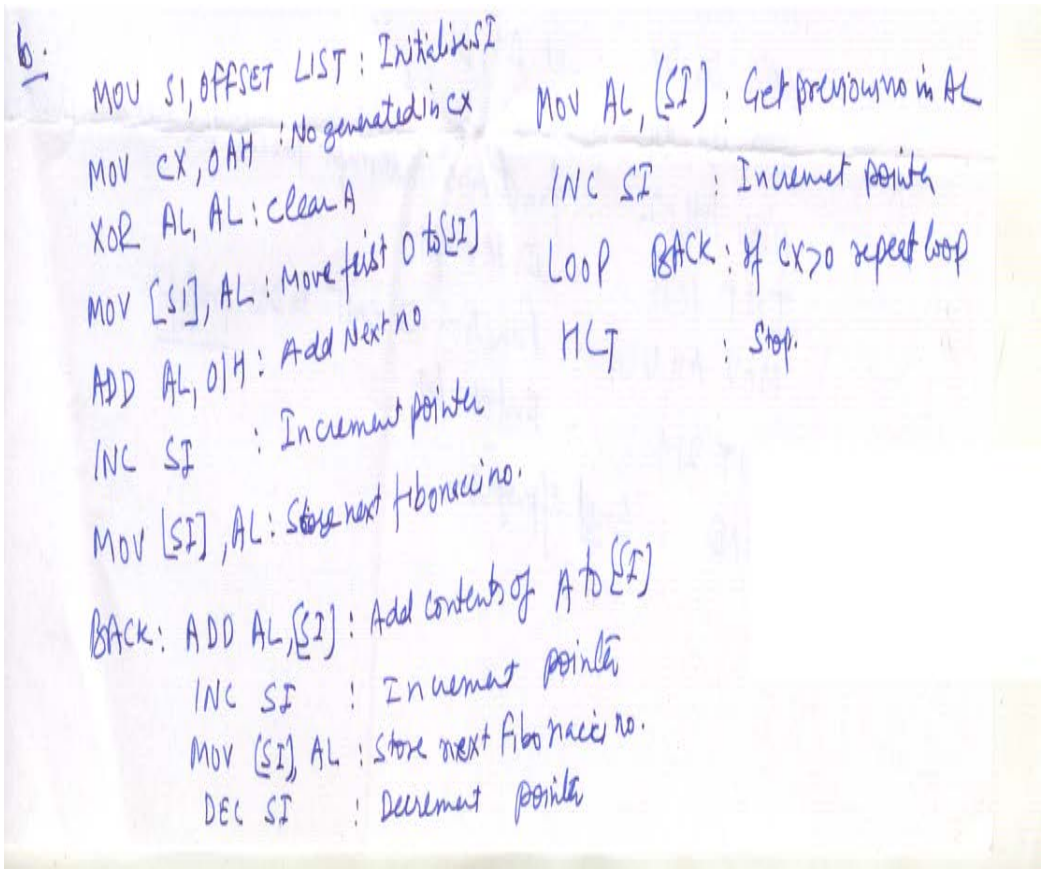
Q.6a. Explain various assembler directives available in 8086.


Q 9. The assembler directives are the hints given to assembler using some predefined alphabetical strings, which help the assembler to correctly understand the assembly language program to prepare the codes. These are

DB – Define byte
DW – Define word
DQ – Define quadword
DT – Define Ten Bytes
EVEN → Align on Even memory address
EQU → Equate
EXTRN → External
NAME → logical Name of module
OFFSET → offset of a label.
PTR → Pointer

ASSUME → Assume logical segment Name
END – End of program
ENDP → End of procedure.
ENDS → End of segment
LEGTH → Byte length.
LOCAL →
PUBLIC → Public
GROUP → Group the related segment
LABEL → label
ORG → Origin
PROC – Procedure.

b. Write an assembly language program to generate Fibonacci series and also write necessary comments.



Q.7 a. Explain the approach methodology to clear screen using BIOS Interrupt.

Ans. Pg no. 331 Text.1

b. Write an 8086 assembly language program to clear / scroll the screen and position of the cursor.

Q.7b Program clears/scrolls the screen and positions the cursor.

```
.MODEL SMALL
.STACK
.DATA
.CODE
START: XOR AL,AL   : No of lines to scroll
       XOR CX,CX   : CH = top row CL = top left column
       MOV DH,24   : DH = bottom row
       MOV DL,79   : DL = bottom left column
       MOV BH,07H  : Character attributes for new line
       MOV AH,06H  : Scroll page up
       INT 10H     : Interrupt
       XOR DX,DX   : Set DH, DL is Row and column
       XOR BH,BH   : cursor position
       MOV AH,02   : function 2 interrupt to sets the cursor Position
       INT 10H     : Interrupt
       MOV AH,4CH  : function to exit to DOS prompt
       INT 21H     : Interrupt.
       END         : End of program
```

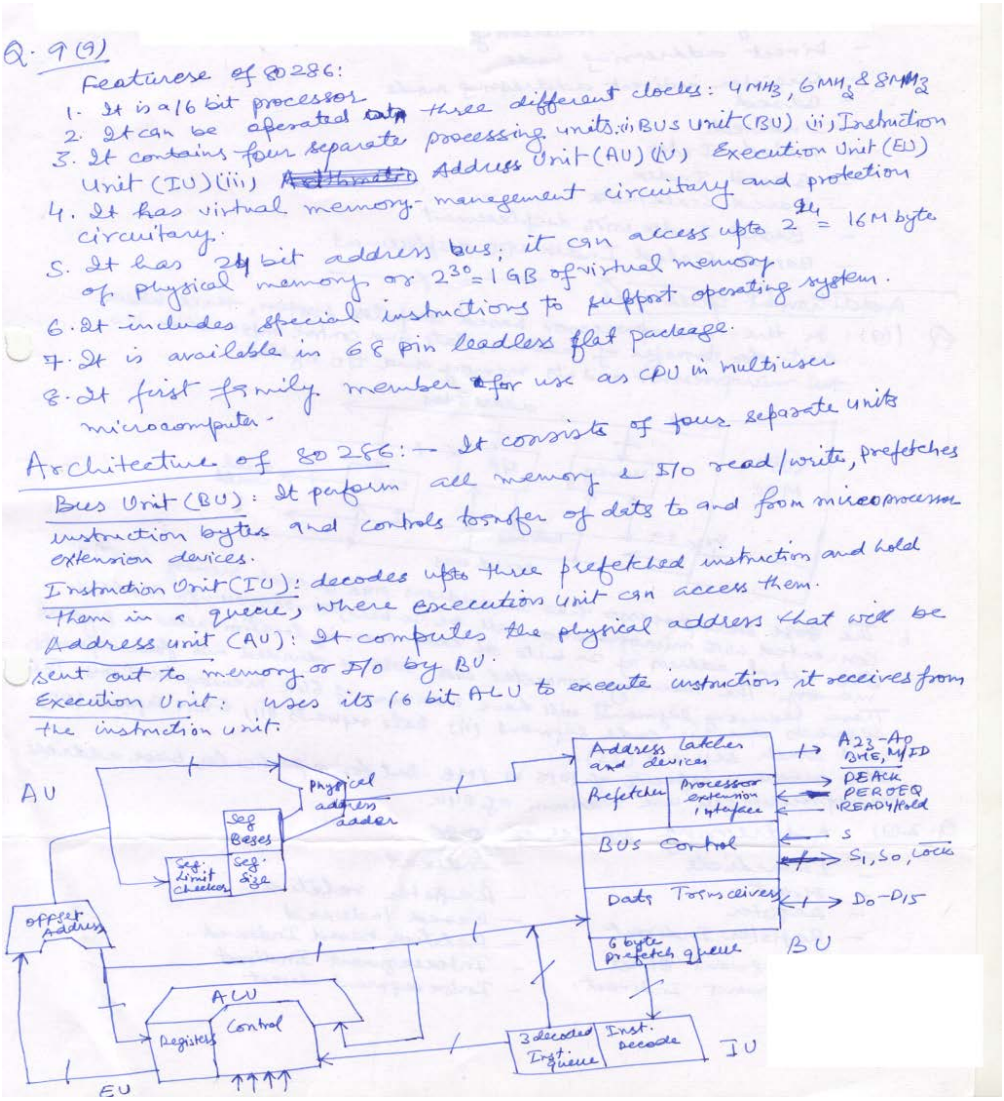
Q.8 a. Write the instructions for 8087 coprocessor and give an illustration to compute the square root.

Ans. Page No.356-363 of Textbook-I

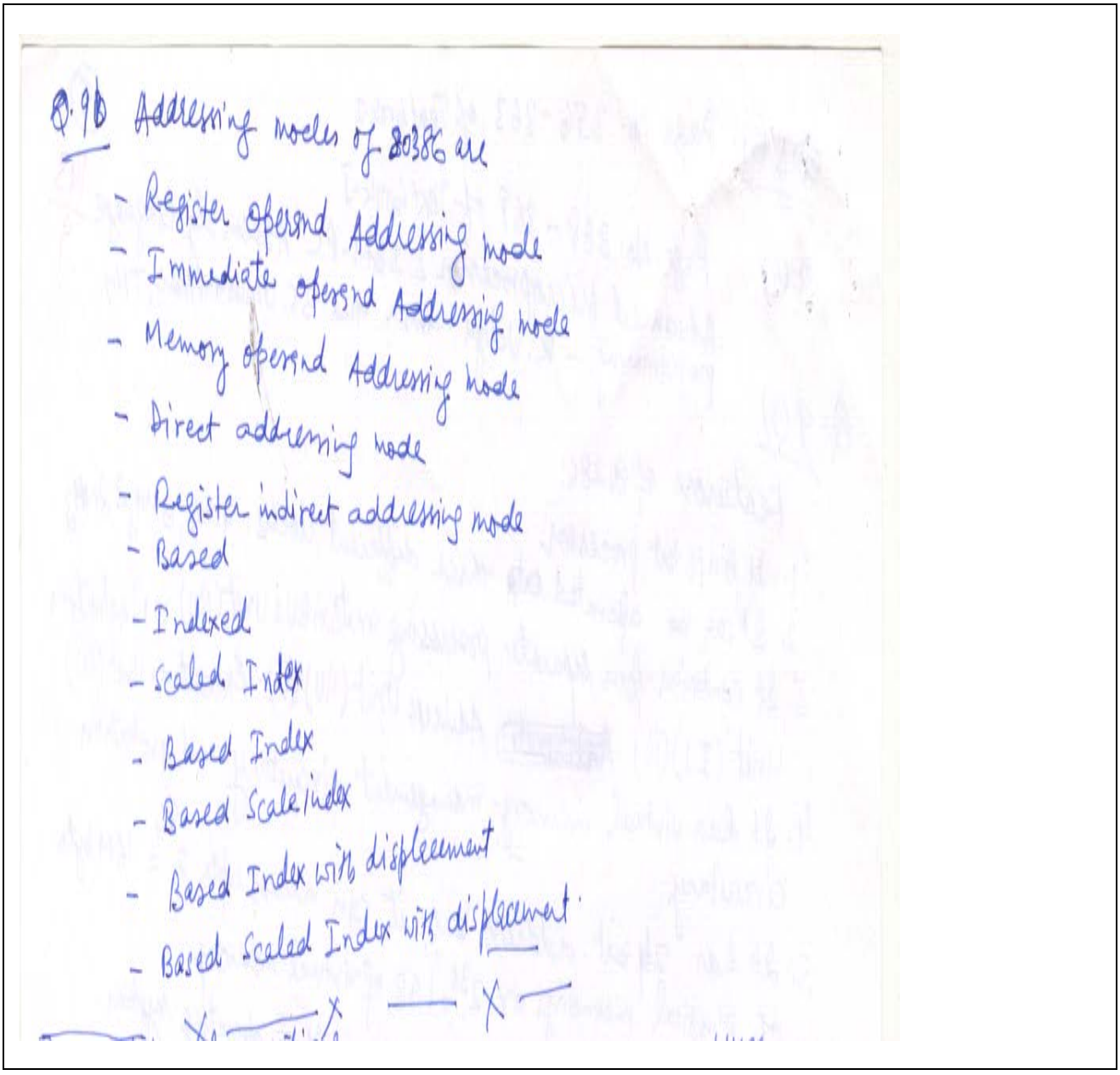
b. Write a C-program to create a subdirectory if it does not exist, using DOS interrupt. A suitable message should be displayed on CRT depending on the success or failure of the operation.

Ans. Page No. 365-366 of Textbook 1 21.2)

Q.9 a. What are the important features of 80286? Describe its internal architecture.



b. Mention the various addressing modes available in 80386 microprocessor.



Q.9b Addressing modes of 80386 are

- Register operand Addressing mode
- Immediate operand Addressing mode
- Memory operand Addressing mode
- Direct addressing mode
- Register indirect addressing mode
- Based
- Indexed
- Scaled Index
- Based Index
- Based Scale index
- Based Index with displacement
- Based Scaled Index with displacement.

Text book

I. Advanced Microprocessors & IBM-PC Assembly Language Programming, K. Udaya Kumar and B.S. Umashankar, TMH,

II. Advanced Microprocessors and Peripherals, A.K. Ray and K.M. Burchandi, TMH, 2000