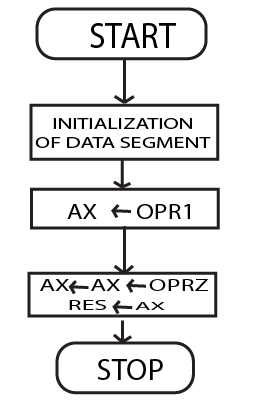
Experiment No: 1

Experiment Name: Perform the task to develop and execute an assembly language program for solving arithmetic problems using 8086/8088 uP trainer or MASM type tools software simulator.

Object: To implement assembly language program to perform basic arithmetic operation such as addition, subtraction, multiplication etc.

Apparatus: MASM software, PC.

Working Process: wb‡¤œ program flow chart cÖ`vb Kiv n‡jv, †hLv‡b `ywU msL¨v †hvM Kivi c×wZ †`Lv‡bv n‡q‡Q|



Program:

MODDEL SMALL

STACK 100H

DATA

A DW 02W

B DW 08H

CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV AX, A

MOV BX, B

ADD AX, BX

ADD AX, 30H

MOV AH, 02

MOV DX, AX

INT 21H

MAIN ENDP

END MAIN

Operation: c~‡e© ewY©Z MASM Gi gva¨‡g A¨v‡m¤^wj †cÖvMÖvg Gw·wKDkb c×wZ Abymv‡i|

Experiments result: `yB Ges AvU Gi †hvMdj wnmv‡e AvDUcyU `k †`Lv‡e|

Experiment No: 2

Experiment Name: Perform the task to develop and execute an assembly program for solving logical problems using 8086\8088 up trainer or MASM type tools or software simulator.

Object: To implement assembly language program to perform basic logical operation such as AND, OR , NOT etc.

Apparatus: MASM software, PC.

Working process: wb‡¤œ wZbwU msL¨vi gv‡S Kxfv‡e eo msL¨vwU †ei Ki‡Z nq Zvi GKwU †d¬v-Pv©U †`Lv‡bv n‡q‡Q|

TAKE THREE NUMBERS NUM1, NUM2, NUM3

COMPARE ACCUMULATOR WITH NUM2

START

MOVE NUM1 TO ACCUMULATOR

IS ACCUMULATOR

GREATER OR EQUAL?

MOV NUM2 TO ACCUMULATOR

NO

YES

PRINT LARGEST NUMBER

COMPARE ACCUMULATOR WITH NUM3

END

**Program:**

MODEL SMALL

STACK 100H

DATA

NUM1 DB 5

NUM2 DB 9

NUM3 DB 7

LRGT DB ?

CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV AL, NUM1

MOV LRGT, AL

CMP AL, NUM2

JGE SKIP1

MOV AL. NUM2

MOV LRGT, AL

SKIP1:

MOV AL, LRGT

CMP AL, NUM3

JGE SKIP2

MOV AL, NUM3

MOV LRGT, AL

SKIP2:

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

Operation: c~‡e© ewY©Z MASM Gi gva¨‡g A¨v‡m¤^wj †cÖvMÖvg Gw·wKDkb c×wZ Abymv‡i|

Experiments result: 5, 7, 9 Gi ga¨ eo msL¨vwU n‡”Q 9, GRb¨ AvDUcy‡U 9 †`Lv‡e|

Experiment No: 3

Experiment Name: Perform the task to develop and execute an assembly program to compute 1’s or 2’s complement of binary number using 8086\ 8088 trainer or MASM type tools or software simulator.

Object: To implement assembly language program to perform 2’s complement of a binary number.

Apparatus: MASM software, PC.

Working process: wb‡¤œ Kv‡Ri c×wZi GKwU †d¬v-PvU© cÖ`vb Kiv n‡jv|

START

**Program:**

MODEL SMALL

STACK 100H

TAKE BINARY NUMBER AS INPUT

DATA

NUM DB 000000010B

CODE

MAIN PROC

MOV AX, @DATA

LOAD BINARY NUMBER INTO ACCUMULATOR

MOV DS, AX

MOV ES, AX

MOV AH, 0000H

MOV AL, NUM

FIRST FIND 1’S COMPLEMENT

NOT AL

MOV BL, AL

ADC AL, 00000001B

MOV BL, AL

MOV AX, 4COOH

ADD 1 WITH RESULT

INT 21H

MAIN ENDP

END MAIN

END

Operation: c~‡e© ewY©Z MASM Gi gva¨‡g A¨v‡m¤^wj †cÖvMÖvg Gw·wKDkb c×wZ Abymv‡i|

Experiments result:

Before execution: Binary number = 00000010

After execution: Binary number = 11111110

Experiment No: 4

Experiment Name: Perform the task to transmit data from a microprocessor to an I\O using intel 8086/8088 based microprocessor trainer or MASM type tools or simulator software.

Object: Understand the way CPU sends data in order to transmit data by displaying I.O devices and to control them.

Apparatus: 8088 Microprocessor trainer, I\O board.

Working process: wb‡¤œ Kv‡Ri c×wZi GKwU †d¬v-PvU© cÖ`vb Kiv n‡jv|

START

SET: 8255

1. MODEL 0

2. PORT B : OUTPUT

MOVE DATA TO AL

WRITE AL TO PORT B

STOP

Program:

10000: 0400 B080 MOV AL, 80; LOAD AL WITH CB

20000: 0402 E631 OUT 13, AL

30000: 0404 BOFF MOV AL, FF; LOAD AL, WITH FF

40000: 0406 E611 OUT 11, AL; WRITE TO PORTB

50000: 0408 F4 HLT

Operation steps:

1. 8086 uP cÖwkÿ‡Ki mv‡\_ I\O †evW© mwVKfv‡e Kv‡b± Ki‡Z n‡e|

2. ‡cÖvMÖvg BbcyU Ki‡Z n‡e |

3. †cÖvMÖvg wbev©n Kiv Ges djvdj ch©‡eÿY Kiv|

4. if the setting value equals FF then the LED should light up

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AL | LED7 | LED6 | LED5 | LED4 | LED3 | LED 2 | LED 1 | LED 0 |
| 1H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2H | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4H | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 8H | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 10H | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 20H | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 40H | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 80H | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Experiment No: 5

Experiment Name: Perform the task to receive data from an I/O to the microprocessor using intel 8086/8088 based microprocessor trainer or MASM type tools or software simulator.

Objective: Understanding the CPU’s many ways of accepting data in order to use the CPU to process external styles of data to carry out effective applications.

Apparatus: 8088 uP trainer board. I/O board.

Work process:

START

SET: 8255

1. MOD 0, 2. PORT A: INPUT

READ DATA FROM PROT A

STOP

Program: 10000: 0400 B090 MOV AL, 90; LOAD AL WITH CB

20000: 0402 E613 OUT 13, AL

30000: 0404 E410 IN AL, 10; READ PORT A

40000: 0406 F4 HLT

Operation: i) Connecting correctly I/O board -01 with 8088uP AL WITH CB

ii) Inputting program. iii) Execute program and observe experiment results.

iv) When the program is executed wait until it has stopped and then simultaneously pressing F3 and R, then we see the registers AX, BX, CX, and DX. If the variable is preset to AL, then the experiment with be correct.

Experiments results:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SW7 | SW6 | SW5 | SW4 | SW3 | SW2 | SW1 | SW0 | AL |
| NO | NO | NO | NO | NO | NO | NO | OFF | 01H |
| NO | NO | NO | NO | NO | NO | OFF | NO | 02H |
| NO | NO | NO | NO | NO | OFF | NO | NO | 04H |
| NO | NO | NO | NO | OFF | NO | NO | NO | 08H |
| NO | NO | NO | OFF | NO | NO | NO | NO | 10H |
| NO | NO | OFF | NO | NO | NO | NO | NO | 20H |
| NO | OFF | NO | NO | NO | NO | NO | NO | 40H |
| OFF | NO | NO | NO | NO | NO | NO | NO | 80H |

Experiment No: 6

Experiment Name: Perform the task to develop and execute assembly program/subroutine to produce time delays of different duration using 8086/8088 uP trainer or MASM type tools or software simulator.

Objective: To implement assembly language subroutine to produce time delays of different duration.

Apparatus: MASM software, PC.

Working process: wb‡¤œ k ms delay on different durations Gi †d¬v-Pv©U cÖ`vb Kiv n‡jv|

(B,C K)

K1 ms delay

(D) N

(D) (D)-1

K3 ms delay

K2 ms delay

YES

NO

YES

NO

(B,C) (B, C)-1

(A) (B)

(A) (A)U(C)

IS, Z=1?

IS, Z=1?

Subroutine :

KDELAY : PUSH B

PUSH D

: PUSH PSW

LP2 : MOV D, 8CH

LP1 : DCR D

JNZ LP1

DCX B

MOV A, B

ORA C

JNZ LP2

POP PSW

POP D

POP B

RET

Operation steps: c~©‡e ewY©Z MASM Gi gva¨‡g A¨v‡m¤^wj †cÖvMÖvg Gw·wKDkb cØwZ Abymv‡i|

Experiment No: 7

Experiment Name: Perform the task to develop and execute assembly language programs that implement the branching and looping structures using 8086/8088uP trainer or MASM type tools or software simulator.

Objective: To implement assembly language program to perform branching and looping structures.

Apparatus: MASM software, PC.

Working process: wb‡¤œ k~Y¨ †\_‡K bq ch©šÍ msL¨v wcÖ›U Kivi ‡cÖvMÖv‡gi †d¬v-Pv©U †`Lv‡bv n‡q‡Q, †hLv‡b branching Ges looping ÷ªvKPvi implement n‡q‡Q |

START

INITIALIZE CX TO 10

SET DL=0

INCREMENT DL DECREMENT CX

STOP

Program:

MODEL SMALL

STACK 100H

DATA

CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV CX, 10

MOV AH, 2

MOV DL, 48

@LOOP:

INT 21H

INC DL

DEC CX

JNZ @LOOP

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

Operation: c~©‡e ewY©Z MASM-Gi gva¨‡g A¨v‡m¤^wj †cÖvMÖvg Gw·wKDkb cØwZ Abymv‡i|

Experiments results: k~Y¨ †\_‡K bq ch©šÍ msL¨v cÖ`wk©Z n‡e|