**PROGRAM 1**

**AIM: ALP TO ADD TWO 16 BIT NUMBERS**

**APPARATUS:** 1. PC with MASM software

**PROGRAM:**

.MODEL SMALL

.CODE

MOV AX,2222H

MOV BX,1234H

ADD AX,BX

INT 03H

END

**RESULT**:

**F:\masm>DEBUG XADD.EXE**

**-G**

AX=3456 BX=1234 CX=0009 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000

DS=0B88 ES=0B88 SS=0B98 CS=0B98 IP=0008 NV UP EI PL NZ NA PE NC

0B98:0008 CC INT 3

**-U 0000,0008**

0B98:0000 B82222 MOV AX,2222

0B98:0003 BB3412 MOV BX,1234

0B98:0006 03C3 ADD AX,BX

0B98:0008 CC INT 3

**PROGRAM 2**

**AIM: ALP TO SUBTRACT TWO 16 BIT NUMBERS**

**APPARATUS:** 1. PC with MASM software

**PROGRAM:**

.MODEL SMALL

.CODE

MOV AX,2222H

MOV BX,1234H

SUB AX,BX

INT 03H

END

**RESULT**:

**F:\masm>DEBUG XADD.EXE**

-G

AX=0FEE BX=1234 CX=0009 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000

DS=0B88 ES=0B88 SS=0B98 CS=0B98 IP=0008 NV UP EI PL NZ AC PE NC

0B98:0008 CC INT 3

-U 0000,0008

0B98:0000 B82222 MOV AX,2222

0B98:0003 BB3412 MOV BX,1234

0B98:0006 2BC3 SUB AX,BX

0B98:0008 CC INT 3

**PROGRAM 3**

**AIM: ALP TO ADD TWO 16 BIT NUMBERS USING INDIRECT ADDRESSING MODE**

**APPARATUS:** 1. PC with MASM software

**PROGRAM:**

.MODEL SMALL

.CODE

MOV SI,3000H

MOV AX,[SI]

ADD SI,0002H

MOV BX,[SI]

ADD AX,BX

ADD SI,0002H

MOV [SI],AX

INT 03H

END

**RESULT**:

**F:\masm>DEBUG XIADD.EXE**

**-E DS:3000**

0B88:3000 04.22 8B.22 0F.33 FF.33

-Q

F:\masm>DEBUG XIADD.EXE

-G

AX=5555 BX=3333 CX=0012 DX=0000 SP=0000 BP=0000 SI=3004 DI=0000

DS=0B88 ES=0B88 SS=0B98 CS=0B98 IP=0011 NV UP EI PL NZ NA PO NC

0B98:0011 CC INT 3

-D DS:3000

0B88:3000 22 22 33 33 55 55

**PROGRAM 4**

**AIM: ALP TO ADD TWO 16 BIT NUMBERS USING DIRECT ADDRESSING MODE**

**APPARATUS:** 1. PC with MASM software

**PROGRAM:**

.MODEL SMALL

.CODE

MOV AX,DS:[4000H]

MOV BX,DS:[4002H]

ADD AX,BX

MOV DS:[4004H],AX

INT 03H

END

**RESULT**:

**F:\masm>DEBUG XIADD.EXE**

-E DS:4000

0B88:4000 65.33 72.33 66.44 6C.44

-Q

F:\masm>DEBUG XIADD.EXE

-G

AX=7777 BX=4444 CX=000D DX=0000 SP=0000 BP=0000 SI=0000 DI=0000

DS=0B88 ES=0B88 SS=0B98 CS=0B98 IP=000C NV UP EI PL NZ NA PE NC

0B98:000C CC INT 3

-D DS:4000,4006

0B88:4000 33 33 44 44 77 77 00 33DDww.

-U 0000,000C

0B98:0000 A10040 MOV AX,[4000]

0B98:0003 8B1E0240 MOV BX,[4002]

0B98:0007 03C3 ADD AX,BX

0B98:0009 A30440 MOV [4004],AX

0B98:000C CC INT 3

**PROGRAM 5**

**AIM: ALP TO MULTIPLY TWO 16 BIT NUMBERS USING DIRECT ADDRESSING MODE**

**APPARATUS:** 1. PC with MASM software

**PROGRAM:**

.MODEL SMALL

.CODE

MOV AX,DS:[4000H]

MOV BX,DS:[4002H]

MUL BX

MOV DS:[4004H],DX

MOV DS:[4006H],AX

INT 03H

END

**RESULT**:

**F:\masm>DEBUG XIMUL.EXE**

**-E DS:4000**

0B88:4000 65.01 72.01 66.02 6C.02

-Q

F:\masm>DEBUG XIMUL.EXE

-G

AX=0402 BX=0202 CX=0011 DX=0002 SP=0000 BP=0000 SI=0000 DI=0000

DS=0B88 ES=0B88 SS=0B98 CS=0B98 IP=0010 OV UP EI PL NZ NA PO CY

0B98:0010 CC INT 3

-D DS:4000,4008

0B88:4000 01 01 02 02 02 00 02 04-40 ........@

-**U 0000,0010**

0B98:0000 A10040 MOV AX,[4000]

0B98:0003 8B1E0240 MOV BX,[4002]

0B98:0007 F7E3 MUL BX

0B98:0009 89160440 MOV [4004],DX

0B98:000D A30640 MOV [4006],AX

0B98:0010 CC INT 3

**PROGRAM 6**

**AIM: ALP TO MOVE BLOCK OF DATA FROM ONE LOCATION TO OTHER LOCATION**

**APPARATUS:** 1. PC with MASM software

**WITHOUT OVERLAPPING**

**PROGRAM:**

.MODEL SMALL

.CODE

MOV CX,000AH

MOV SI,3000H

MOV DI,4000H

L1:MOV AL,[SI]

MOV [DI],AL

INC SI

INC DI

DEC CX

JNZ L1

INT 03H

END

**RESULT**:

**-E DS:3000**

0B88:3000 01.01 02.02 03. 04. 05. 06. 07. 08.

0B88:3008 09. 0A.

-Q

F:\masm>debug block.exe

-G

AX=000A BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=300A DI=400A

DS=0B88 ES=0B88 SS=0B98 CS=0B98 IP=0012 NV UP EI PL ZR NA PE NC

0B98:0012 CC INT 3

-D DS:4000,400A

0B88:4000 01 02 03 04 05 06 07 08-09 0A 23 ..........#

-U 0000,0012

0B98:0000 B90A00 MOV CX,000A

0B98:0003 BE0030 MOV SI,3000

0B98:0006 BF0040 MOV DI,4000

0B98:0009 8A04 MOV AL,[SI]

0B98:000B 8805 MOV [DI],AL

0B98:000D 46 INC SI

0B98:000E 47 INC DI

0B98:000F 49 DEC CX

0B98:0010 75F7 JNZ 0009

0B98:0012 CC INT 3

**WITH OVERLAPPING**

**PROGRAM:**

**.**MODEL SMALL

.CODE

MOV CX,0007

MOV SI,3006H

MOV DI,300AH

L1:MOV AL,[SI]

MOV [DI],AL

DEC SI

DEC DI

DEC CX

JNZ L1

INT 03H

END

**RESULT:**

F:\masm>DEBUG BLOCK.EXE

-E DS:3000

0B88:3000 04.01 8B.02 0F.03 FF.04 07.05 8B.06 D9.07 88.

-Q

F:\masm>DEBUG BLOCK.EXE

-G

AX=0001 BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=2FFF DI=3003

DS=0B88 ES=0B88 SS=0B98 CS=0B98 IP=0012 NV UP EI PL ZR NA PE NC

0B98:0012 CC INT 3

-D DS:3000,300A

0B88:3000 01 02 03 04 01 02 03 04-05 06 07 ...........

-U 0000,0012

0B98:0000 B90700 MOV CX,0007

0B98:0003 BE0630 MOV SI,3006

0B98:0006 BF0A30 MOV DI,300A

0B98:0009 8A04 MOV AL,[SI]

0B98:000B 8805 MOV [DI],AL

0B98:000D 4E DEC SI

0B98:000E 4F DEC DI

0B98:000F 49 DEC CX

0B98:0010 75F7 JNZ 0009

0B98:0012 CC INT 3

**PROGRAM 7**

**AIM: ALP TO ADD N 8 BIT NUMBERS**

**APPARATUS:** 1. PC with MASM software

**PROGRAM:**

.MODEL SMALL

.CODE

MOV CL,06H

MOV SI,3000H

MOV AH,00H

MOV AL,[SI]

L1:INC SI

ADD AL,[SI]

JNC L2

INC AH

L2:DEC CL

JNZ L1

INT 03H

END

**RESULT:**

**F:\masm>DEBUG ADDN.EXE**

**-E DS:3000**

0B88:3000 04.B8 8B.35 0F.36 FF.37 07.38 8B.39 D9.

-Q

F:\masm>DEBUG ADDN.EXE

-G

AX=02A4 BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=3006 DI=0000

DS=0B88 ES=0B88 SS=0B98 CS=0B98 IP=0014 NV UP EI PL ZR NA PE CY

0B98:0014 CC INT 3

-U 0000,0014

0B98:0000 B106 MOV CL,06

0B98:0002 BE0030 MOV SI,3000

0B98:0005 B400 MOV AH,00

0B98:0007 8A04 MOV AL,[SI]

0B98:0009 46 INC SI

0B98:000A 0204 ADD AL,[SI]

0B98:000C 7302 JNB 0010

0B98:000E FEC4 INC AH

0B98:0010 FEC9 DEC CL

0B98:0012 75F5 JNZ 0009

0B98:0014 CC INT 3

**PROGRAM 8**

**AIM: ALP TO SUBTRACT N 8 BIT NUMBERS**

**APPARATUS:** 1. PC with MASM software

**PROGRAM:**

.MODEL SMALL

.CODE

MOV CL,06H

MOV SI,3000H

MOV BL,00H

MOV AL,[SI]

L1:INC SI

SUB AL,[SI]

JNC L2

INC BL

L2:DEC CL

JNZ L1

INT 03H

END

**RESULT:**

**F:\masm>DEBUG SUBN.EXE**

**-E DS:3000**

0B88:3000 04.B9 8B.24 0F.35 FF.26 07.38 8B.45 D9.67

-Q

F:\masm>DEBUG SUBN.EXE

-G

AX=0056 BX=0001 CX=0000 DX=0000 SP=0000 BP=0000 SI=3006 DI=0000

DS=0B88 ES=0B88 SS=0B98 CS=0B98 IP=0014 NV UP EI PL ZR NA PE NC

0B98:0014 CC INT 3

-U 0000,0014

0B98:0000 B106 MOV CL,06

0B98:0002 BE0030 MOV SI,3000

0B98:0005 B300 MOV BL,00

0B98:0007 8A04 MOV AL,[SI]

0B98:0009 46 INC SI

0B98:000A 2A04 SUB AL,[SI]

0B98:000C 7302 JNB 0010

0B98:000E FEC3 INC BL

0B98:0010 FEC9 DEC CL

0B98:0012 75F5 JNZ 0009

0B98:0014 CC INT 3

**PROGRAM 9**

**AIM: ALP TO SORT THE GIVEN NUMBERS IN DESCENDING ORDER**

**APPARATUS:** 1. PC with MASM software

**PROGRAM:**

**DESCENDING ORDER:**

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

ARRAY DB 10H,11H,12H,13H,14H,15H,16H,17H

DATA ENDS

CODE SEGMENT

START:

MOV AX,DATA

MOV DS,AX

LABEL2: MOV CX,0008H

MOV AH,00H

MOV BX,OFFSET ARRAY

MOV DX,BX

LABEL3: MOV AL,[BX]

INC BX

CMP AL,[BX]

JNC LABEL1

JE LABEL1

MOV DL,[BX]

DEC BX

MOV [BX],DL

INC BX

MOV [BX],AL

MOV AH,01

LABEL1: LOOP LABEL3

CMP AH,00H

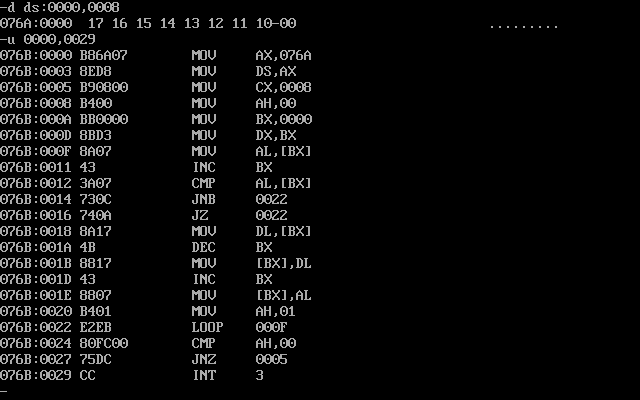
JNE LABEL2

INT 3H

CODE ENDS

END START

**RESULT:**



**PROGRAM 10**

**AIM: ALP TO SORT THE GIVEN NUMBERS IN ASCENDING ORDER**

**APPARATUS:** 1. PC with MASM software

**PROGRAM:**

**ASCENDING ORDER:**

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

ARRAY DB 17H,16H,15H,14H,13H,12H,11H,10H

DATA ENDS

CODE SEGMENT

START:

MOV AX,DATA

MOV DS,AX

LABEL2: MOV CX,0008H

MOV AH,00H

MOV BX,OFFSET ARRAY

MOV DX,BX

LABEL3: MOV AL,[BX]

INC BX

CMP AL,[BX]

JC LABEL1

JE LABEL1

MOV DL,[BX]

DEC BX

MOV [BX],DL

INC BX

MOV [BX],AL

MOV AH,01

LABEL1: LOOP LABEL3

CMP AH,00H

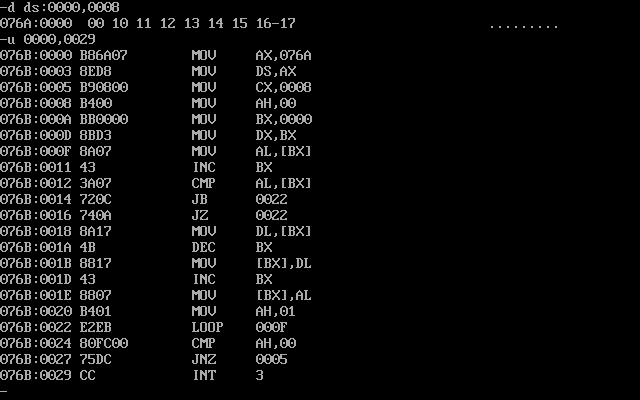
JNE LABEL2

INT 3H

CODE ENDS

END START

**RESULT:**



**PROGRAM 11**

**AIM: ALP TO GET 8 BIT DECIMAL UP AND DOWN COUNTER**

**APPARATUS:** 1. PC with MASM software

**UP COUNTER**

**PROGRAM:**

.MODEL SMALL

.CODE

MOV AL,00

MOV SI,3000H

L1:ADD AL,01H

DAA

MOV [SI],AL

INC SI

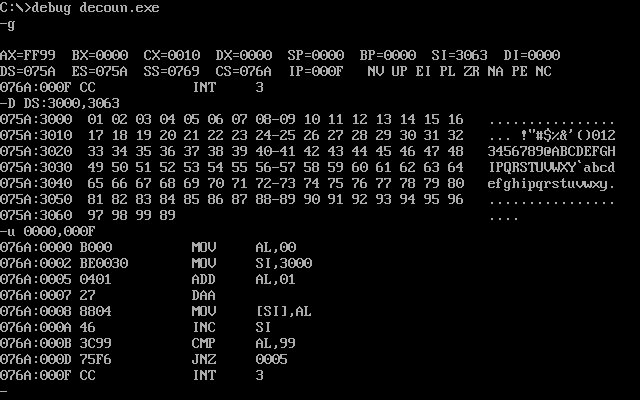
CMP AL,99H

JNE L1

INT 03H

END

**RESULT:**



**DOWN COUNTER**

**PROGRAM:**

.MODEL SMALL

.CODE

MOV AL,99H

MOV SI,3000H

L1:SUB AL,01H

DAS

MOV [SI],AL

INC SI

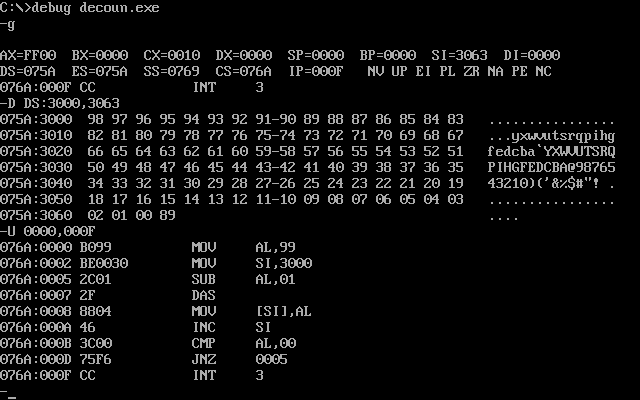
CMP AL,00H

JNE L1

INT 03H

END

**RESULT:**

****

**PROGRAM 12**

**AIM: ALP TO GET 8 BIT HEXADECIMAL UP AND DOWN COUNTER**

**APPARATUS:** 1. PC with MASM software

**UP COUNTER**

**PROGRAM:**

.MODEL SMALL

.CODE

MOV AL,00H

MOV SI,3000H

MOV [SI],AL

L1:INC SI

INC AL

MOV [SI],AL

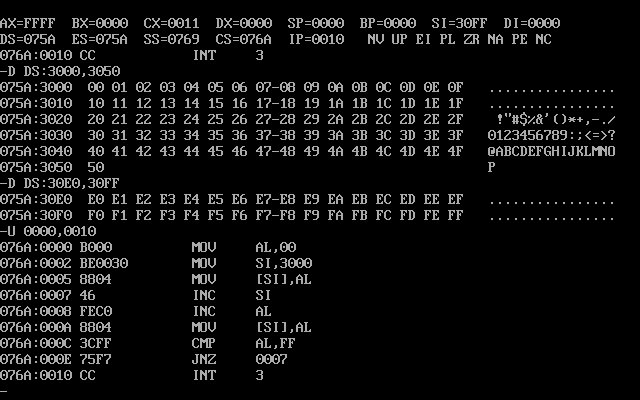
CMP AL,0FFH

JNE L1

INT 03H

END

**RESULT:**

****

**DOWN COUNTER**

**PROGRAM:**

.MODEL SMALL

.CODE

MOV AL,0FFH

MOV SI,3000H

MOV [SI],AL

L1:INC SI

DEC AL

MOV [SI],AL

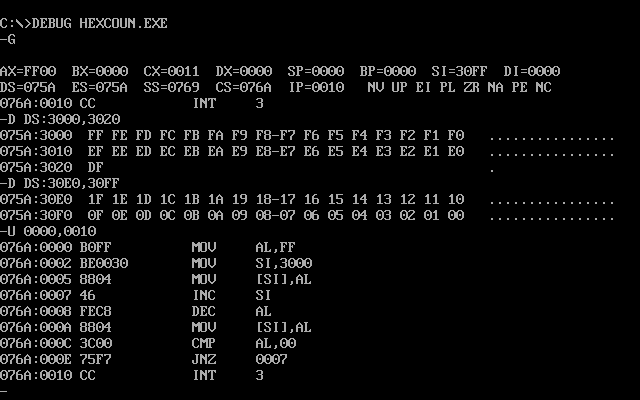
CMP AL,00H

JNE L1

INT 03H

END

**RESULT:**

****

**PROGRAM 13**

**AIM: ALP TO FIND HCF OF GIVEN NUMBERS**

**APPARATUS:** 1. PC with MASM software

**PROGRAM:**

ASSUME CS:CODE, DS:DATA

DATA SEGMENT

NUM1 EQU 75D

NUM2 EQU 37D

HCF DB 01H DUP(?)

DATA ENDS

CODE SEGMENT

START:MOV AH,00H

MOV AL,NUM1

MOV BL,NUM2

BACK:MOV DL,BL

DIV BL

CMP AH,00H

JE NEXT

MOV BL,AH

MOV AH,00H

MOV AL,DL

JMP BACK

NEXT:MOV DI,OFFSET HCF

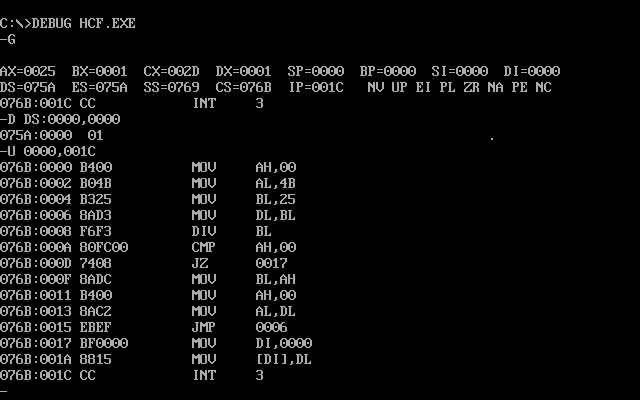
MOV [DI],DL

INT 03H

CODE ENDS

END START

**RESULT:**

****

**PROGRAM 14**

**AIM: ALP TO INTERFACE DIGITAL TO ANALOG CONVERTER**

**APPARATUS:** 1. PC with MASM software

2. 8086 Trainer Kit

3. DAC Interface Module

1. **GENERATION OF SAWTOOTH WAVEFORM**

**PROGRAM:**

.MODEL SMALL

.CODE

ORG 2000H

MOV AL,80H

MOV DX,0FFE6H

OUT DX,AL

BACK2:MOV AL,00H

MOV DX,0FFE2H

BACK1:OUT DX,AL

INC AL

CMP AL,0F2H

JNE BACK1

JMP BACK2

END

**RESULT:**

**F:\masm>DEBUG DAC.EXE**

**-G**

-U 2000,2012

0B4D:2000 B080 MOV AL,80

0B4D:2002 BAE6FF MOV DX,FFE6

0B4D:2005 EE OUT DX,AL

0B4D:2006 B000 MOV AL,00

0B4D:2008 BAE2FF MOV DX,FFE2

0B4D:200B EE OUT DX,AL

0B4D:200C FEC0 INC AL

0B4D:200E 3CF2 CMP AL,F2

0B4D:2010 75F9 JNZ 200B

0B4D:2012 EBF2 JMP 2006

1. **GENERATION OF TRIANGULAR WAVEFORM**

**PROGRAM:**

.MODEL SMALL

.CODE

ORG 2000H

MOV AL,80H

MOV DX,0FFE6H

OUT DX,AL

MOV AL,00H

MOV DX,0FFE2H

BACK1:OUT DX,AL

INC AL

CMP AL,0FFH

JNE BACK1

BACK2: OUT DX,AL

DEC AL

CMP AL,00H

JNE BACK2

JMP BACK1

END

**RESULT:**

**F:\masm>DEBUG DAC.EXE**

**-G**

-U 2000,2020

0B4D:2000 B080 MOV AL,80

0B4D:2002 BAE6FF MOV DX,FFE6

0B4D:2005 EE OUT DX,AL

0B4D:2006 B000 MOV AL,00

0B4D:2008 BAE2FF MOV DX,FFE2

0B4D:200B EE OUT DX,AL

0B4D:200C FEC0 INC AL

0B4D:200E 3CF2 CMP AL,F2

0B4D:2010 75F9 JNZ 200B

0B4D:2012 EE OUT DX,AL

0B4D:2013 FEC8 DEC AL

0B4D:2015 3C00 CMP AL,00

0B4D:2017 75F9 JNZ 2012

0B4D:2019 EBF0 JMP 200B

1. **GENERATION OF SQUARE WAVEFORM**

**PROGRAM:**

.MODEL SMALL

.CODE

ORG 2000H

MOV AL,80H

MOV DX,0FFE6H

OUT DX,AL

MOV DX,0FFE2H

BACK:MOV AL,00H

OUT DX,AL

CALL DELAY

MOV AL, 0F2H

OUT DX,AL

CALL DELAY

JMP BACK

DELAY:MOV CX,0FFH

L:LOOP L

RET

END

**RESULT:**

**F:\masm>DEBUG DAC.EXE**

**-G**

-U 2000,2020

0B5D:2000 B080 MOV AL,80

0B5D:2002 BAE6FF MOV DX,FFE6

0B5D:2005 EE OUT DX,AL

0B5D:2006 BAE2FF MOV DX,FFE2

0B5D:2009 B000 MOV AL,00

0B5D:200B EE OUT DX,AL

0B5D:200C E80800 CALL 2017

0B5D:200F B0FF MOV AL,FF

0B5D:2011 EE OUT DX,AL

0B5D:2012 E80200 CALL 2017

0B5D:2015 E8F2 JMP 2009

0B5D:2017 B9FF00 MOV CX,00FF

0B5D:201A E2FE LOOP 201A

0B5D:201C C3 RET

**D.GERERATION OF STAIRCASE WAVEFORM**

**PROGRAM:**

.MODEL SMALL

.CODE

ORG 2000H

MOV AL,80H

MOV DX,0FFE6H

OUT DX,AL

MOV DX,0FFE0H

MOV AL,00H

BACK: OUT DX,AL

CALL DELAY

ADD AL,064H

CMP AL,0FFH

JNZ BACK

DELAY:MOV CX,0FFH

L:LOOP L

RET

END

**RESULT:**

**F:\masm>DEBUG DAC.EXE**

**-G**

-U 2000,201A

0B5D:2000 B080 MOV AL,80

0B5D:2002 BAE6FF MOV DX,FFE6

0B5D:2005 EE OUT DX,AL

0B5D:2006 BAE0FF MOV DX,FFE2

0B5D:2009 B000 MOV AL,00

0B5D:200B EE OUT DX,AL

0B5D:200C E80815 CALL 2015

0B5D:200F B040 ADD AL,64D

0B5D:2011 3CFF CMP AL,0FFH

0B5D:2013 75F9 JNZ 200B

0B5D:2015 B9FF00 MOV CX,0FFH

0B5D:2018 E2FE LOOP 2018

0B5D:201A C3 RET

**PROGRAM 15**

**AIM: ALP to INTERFACE TRAFFIC LIGHTS**

.South can go North,West

.West can go North

.North can go south

.Pedestrians can cross on East

**APPARATUS:** 1. PC with MASM software

2. 8086 Trainer Kit

3. Traffic Light Interface Module

**PROGRAM:**

.MODEL SMALL

.CODE

ORG 2000H

MOV AL,80H

MOV DX,0FFE6H

OUT DX,AL

MOV AL,10H

MOV DX,0FFE0H

OUT DX,AL

MOV AL,81H

MOV DX,0FFE2H

OUT DX,AL

MOV AL,7AH

MOV DX,0FFE4H

OUT DX,AL

INT 03H

END

**RESULT:**

**F:\masm>DEBUG TRAF.EXE**

-U 2000,2018

0000:2000 BAE6FF MOV DX,FFE6

0000:2003 B680 MOV AL,80

0000:2005 EE OUT DX,AL

0000:2006 BAE0FF MOV DX,EEE0

0000:2009 B010 MOV AL,10

0000:200B EE OUT DX,AL

0000:200C BAE2FF MOV DX,FFE2

0000:200F B081 MOV AL,81

0000:2011 EE OUT DX,AL

0000:2012 BAE4FF MOV DX,FFE4

0000:2015 B07A MOV AL,7A

0000:2017 EE OUT DX,AL

0000:2018 CC INT 03

**16.ALP to INTERFACE STEPPER MOTOR**

**PROGRAM:**

.MODEL SMALL

.CODE

ORG 2100H

MOV AL,80H

MOV DX,0FFE6H

OUT DX,AL

MOV BX,00C8H

MOV AL,88H

MOV DX,0FFE0H

AGAIN:OUT DX,AL

CALL DELAY

ROR AL,01H

DEC BX

JNZ AGAIN

MOV BX,00C8H

MOV AL,88H

MOV DX,0FFE0H

AGAIN1:OUT DX,AL

CALL DELAY

ROL AL,01H

DEC BX

JNZ AGAIN1

INT 03H

DELAY:MOV CX,00H

L:LOOP L

RET

END

**RESULT:**

**F:\masm>DEBUG STEP.EXE**

**-U 2000,2028**

0B5A:2100 B080 MOV AL,80

0B5A:2102 BAE6FF MOV DX,FFE6

0B5A:2105 EE OUT DX,AL

0B5A:2106 BBC800 MOV BX,00C8

0B5A:2109 B088 MOV AL,88

0B5A:210B BAE0FF MOV DX,FFE0

0B5A:210E EE OUT DX,AL

0B5A:210F E81700 CALL 2129

0B5A:2112 D0CB ROR AL,1

0B5A:2114 4B DEC BX

0B5A:2115 75F7 JNZ 210E

0B5A:2117 BBC800 MOV BX,00C8

0B5A:211A B088 MOV AL,88

0B5A:211C BAE0FF MOV DX,FFE0

0B5A:211F EE OUT DX,AL

0B5A:2120 E80600 CALL 2129

0B5A:2123 D0C0 ROL AL,01

0B5A:2125 4B DEC BX

0B5A:2126 75E7 JNZ 211F

0B5A:2128 CC INT 03

0B5A:2129 B90000 MOV CX,0000

0B5A:212B E2FF LOOP 212B

0B5A:212D C3 RET

1. **ALP to INTERFACE LED**

**PROGRAM:**

.MODEL SMALL

.CODE

ORG 2000H

MOV AL,80H

MOV DX,0FFE6H

OUT DX,AL

MOV DX,0FFE0H

AGAIN:MOV AL,00H

BACK: OUT DX,AL

INC AL

CALL DELAY

CMP AL,0AH

JNE BACK

BACK1:DEC AL

OUT DX,AL

CALL DELAY

CMP AL,00H

JNE BACK1

JMP AGAIN

DELAY:MOV CX,00H

H:LOOP H

H1:LOOP H1

RET

END

**RESULT:**

**F:\masm>DEBUG LED.EXE**

**-U 2000,2028**

0B52:2000 B080 MOVAL,80

0B52:2002 BAE6FF MOV DX,FFE6

0B52:2005 FE OUT DX,AL

0B52:2006 BAE0FF MOV DX,FFE0

0B52:2009 B000 MOV AL,00

0B52:200B EE OUT DX,AL

0B52:200C E81200 CALL 2021

0B52:200F FFC0 INC AL

0B52:2011 3C0A CMP AL,0A

0B52:2013 75F6 JNZ 200B

0B52:2015 FEC8 DEC AL

0B52:2017 EE OUT DX,AL

0B52:2018 E80600 CALL 2021

0B52:201B 3C00 CMP AL,00

0B52:201D 75F6 JNZ 2015

0B52:201F EBE4 JMP 2006

0B52:2021 CC INT 03

0B52:2022 B90000 MOV CX,0000

0B52:2025 E2FF LOOP 2025

0B52:2027 E2FE LOOP 2027

0B52:2028 C3 RET

**MICROPROCESSORS LABORATORY**

**RECORD**



**Submitted by**

K. PRAVEEN KUMAR

Asst. Professor

Department of ECE

GITAM University

Hyderabad Campus