Morse Code Decoder

ECEN406-FALL2021

Team Members:

- Ahmed Fouad	18101467
- Habiba Mahmoud	18100755
- Rawan Abdelkhaleq	18101703
- Seif Soliman	18102315
- Shorouk Alalem	18100433

Delivered to:

- Dr.Ahmed Soltan
- Eng.Mariam Salah

Contents

Objective

Morse code is one of the most recognized and used codes that was designed to secure (encrypt and decrypt) the communication from the sender to its receiver. The main target we aim to achieve is to implement a message decipherer using the 8086 processor through the aid of 3 buttons, each carrying out functioning roles. The first button represents a dot with length of one unit, second button represents a dash consisting of three units and finally a third button which handles the spaces between dots and dashes, to differentiate words from letters and different segments making up a letter. After processing on the 8086 the result will be outputted on a display.

Schematic

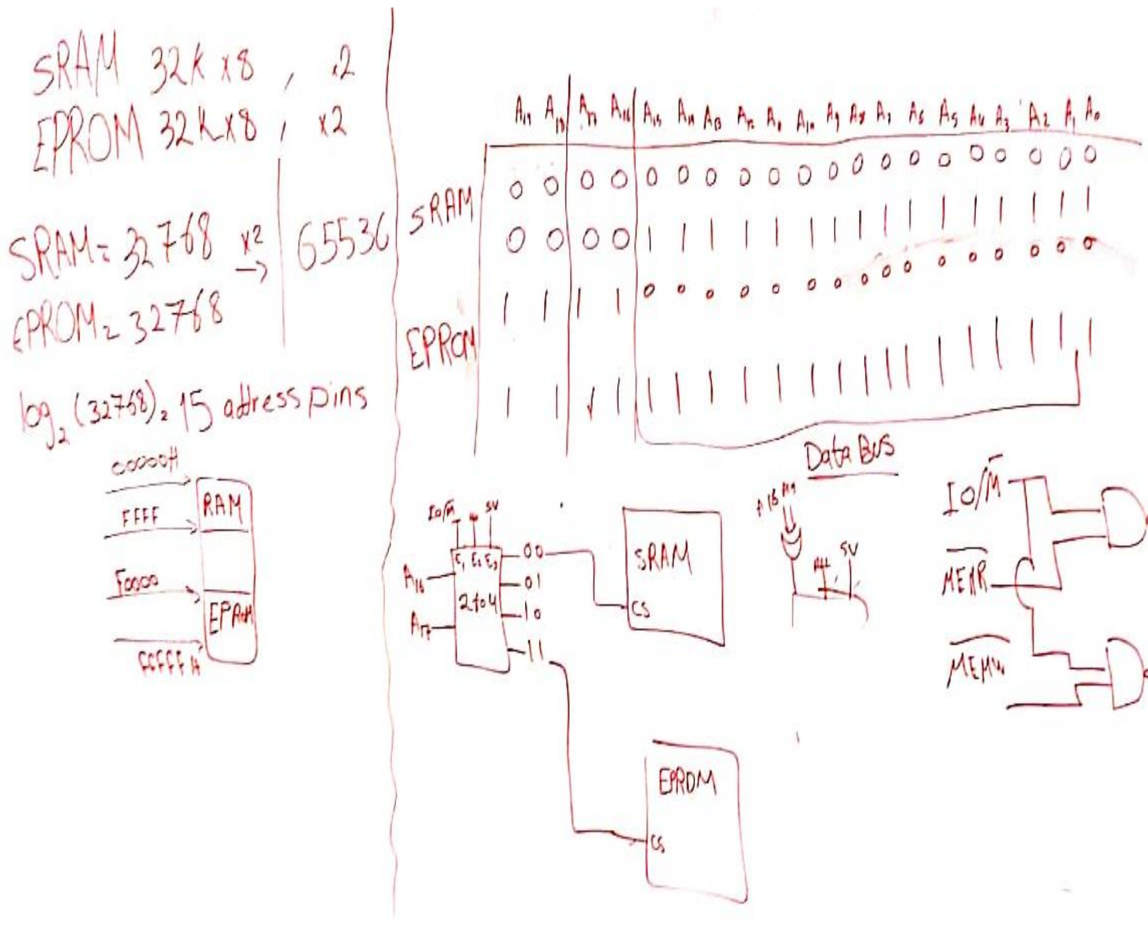


Figure 1:Memory Interface Schematic.

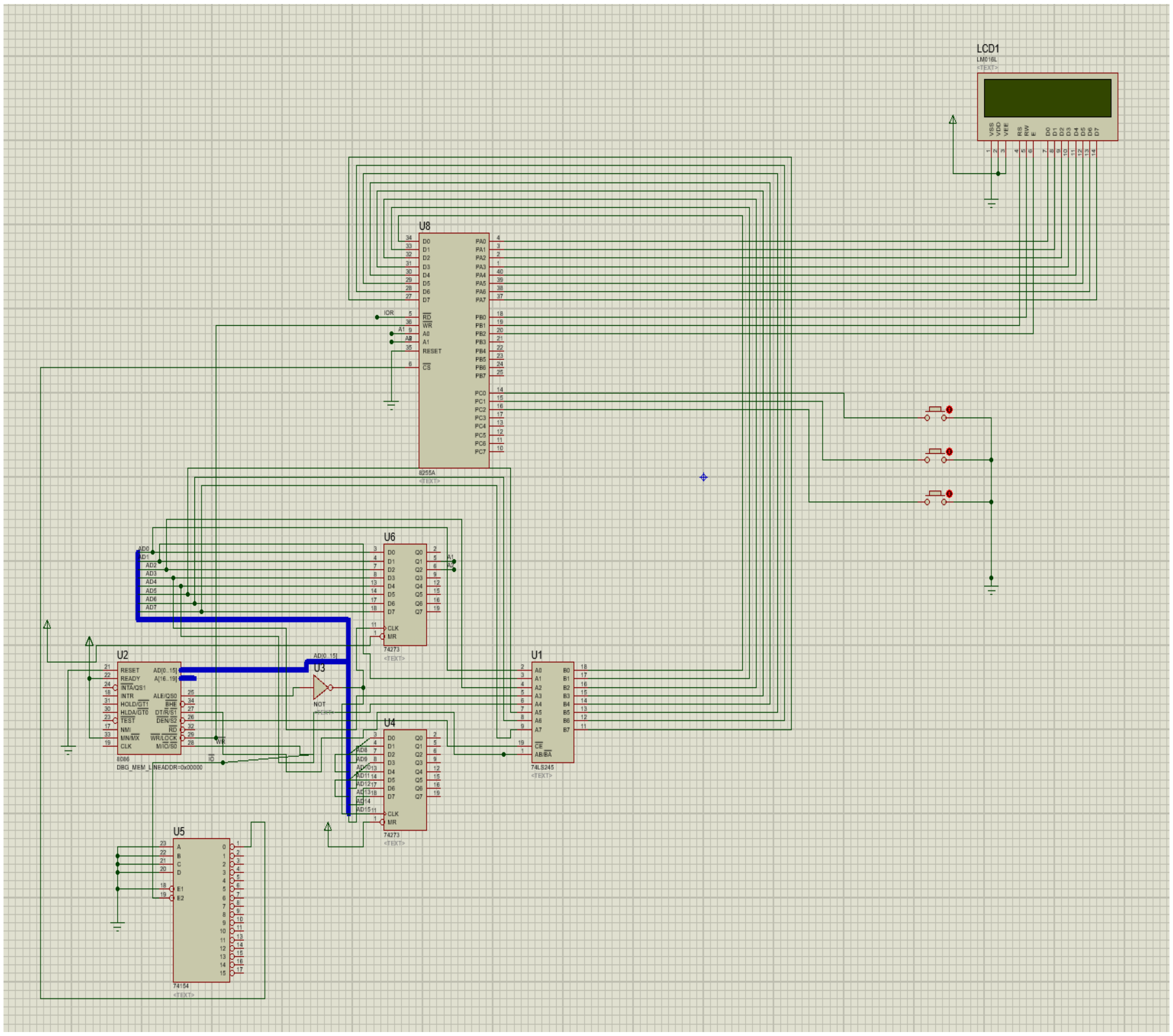


Figure 2: Proteus Simulation for Decoder.

Flowchart

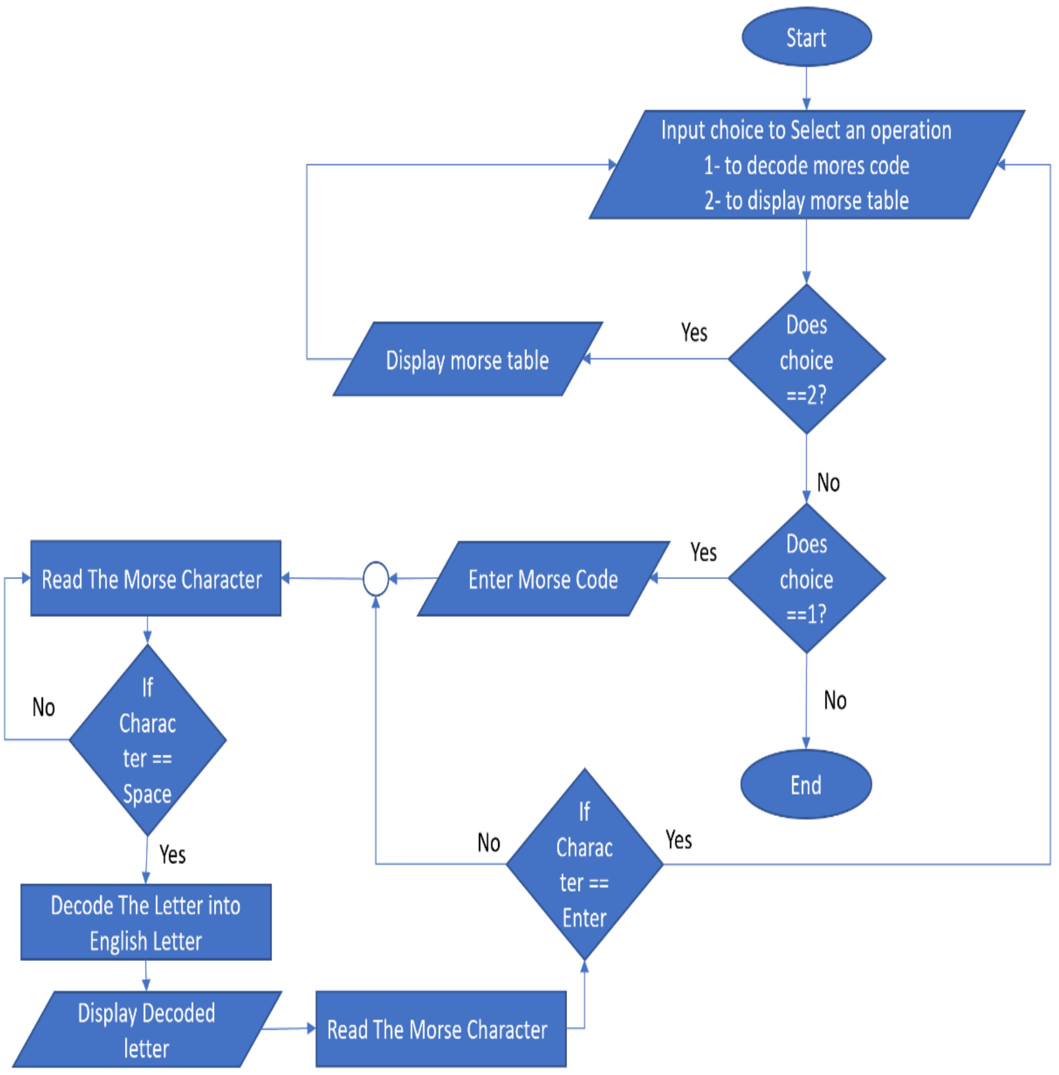


Figure 3:Flow chart of assembly program.

Code

```
.MODEL COMPACT


.DATA

    ;Decode
    INPUT_DECODE DB 50 DUP (?)
    INPUT_CHECKER DB 10 DUP (?)
    COUNT_DECODE_PERCHAR DB 00
    COUNT_DECODE DB 00
    INDEX_DECODE DB 00
    CODES DB 00
    LENGTHS_DECODE DB 00

    ;Morse Code for each character
    PRESS_ANY_KEY DB 13,10,13,10,"PRESS ANY KEY TO CONTINUE!$"

    STR_INPUT_DECODE DB 13," "
        DB 13,10,"DECODING..."
        DB 13,10," "
        DB 13,10,"Input the Morse Code to Decode"
        DB 13,10,10,"INPUT : ",13,10,"$"
    STR_OUTPUT DB 13,10,"OUTPUT: ",13,10,"$"
    STR_EMPTY DB "NO INPUT$"
    MORSE_LISTS DB 13," "
        DB 13,10,"MORSE CODE TABLE"
        DB 13,10," "
        DB 13,10,"A = .-"
        DB "    B = -..."
        DB "  C = -.-."
        DB "  D = -.."
        DB "   E = ."
        DB 13,10,"F = ..-."
        DB "  G = --."
        DB "   H = ...."
        DB "  I = .."
        DB "    J = .---"
        DB 13,10,"K = -.-"
        DB "   L = .-.."
```

```
           DB "  M = --"
           DB "    N = -."
           DB "    0 = ---"
           DB 13,10,"P = .--."
           DB "  Q = --.-"
           DB "  R = .-."
           DB "   S = ..."
           DB "   T = -"
           DB 13,10,"U = ..-"
           DB "   V = ...-"
           DB "  W = .--"
           DB "   X = -..-"
           DB "  Y = -.--"
           DB 13,10,"Z = --..$"
        HEADER DB
"*****************************************************"
             DB 13,10,    "****************                    ***************"
             DB 13,10,    "                MORSE CODE TRANSLATOR                "
             DB 13,10,    "****************                    ***************"
             DB 13,10,    "*****************************************************$"
        SELECT DB 13,10,10, "               DECODING MORSE CODE                 "
             DB 13,10,10, "                         "
             DB 13,10,    "MAIN MENU:               "
             DB 13,10,    "   1.Decode                 "
             DB 13,10,    "   2.Show Morse code lists    "
             DB 13,10,    "   3.Exit                   "
             DB 13,10,    "                        "
             DB 13,10,    "Your choice: $"

      .CODE
      .STARTUP
      ;Display Main Menu

        MAIN_MENU:

        CALL CLEAR_SCREEN

        MOV AH, 9H
        MOV DX, OFFSET HEADER
        INT 21H

        MOV AH, 9H
        MOV DX, OFFSET SELECT
```

```
    INT 21H

    MOV AH, 1H
    INT 21H

    CMP AL, 31H
    JNE NOT_DECODE
    JMP _DECODE
    NOT_DECODE:

    CMP AL, 32H
    JNE NOT_MORSECODE
    JMP _MORSECODE
    NOT_MORSECODE:

    CMP AL, 33H
    JNE NOT_REAL_EXIT
    JMP REAL_EXIT
    NOT_REAL_EXIT:
    JMP MAIN_MENU

     CLEAR_SCREEN:
        ;PUSHA
        MOV AH, 0H
        MOV AL, 3H
        INT 16
        ;POPA
       RET
;Decoding part

    _DECODE:
    MOV AX, 0000H
    MOV BX, 0000H
    MOV CX, 0000H
    MOV DX, 0000H
    MOV SI, 0000H
    MOV INPUT_DECODE, 0000H
    MOV INPUT_CHECKER, 0000H
    MOV COUNT_DECODE_PERCHAR, 0000H
    MOV COUNT_DECODE, 0000H
    MOV INDEX_DECODE, 0000H
    MOV CODES, 0000H
    MOV LENGTHS_DECODE, 0000H
```

```
CALL CLEAR_SCREEN
CALL GETINPUT_DECODE

MOV AX, 0000H
MOV AH, 9H
MOV DX, OFFSET STR_OUTPUT
INT 21H

CALL CHECK_INPUT_DECODE
GETINPUT_DECODE:
   MOV AH, 9H
   MOV DX, OFFSET STR_INPUT_DECODE
   INT 21H

   MOV SI, OFFSET INPUT_DECODE
   MOV LENGTHS_DECODE, 0

      LZ:
         MOV AH,1H
         INT 21H

         CMP AL ,13
         JE DONE_DECODE

         MOV [SI],AL
         INC SI
         INC LENGTHS_DECODE

         CMP AL, 8H
         JE BACK_DECODE


         JMP LZ
         BACK_DECODE:
            CMP LENGTHS_DECODE[0],0
            JE LZ
            DEC SI
            MOV AH, 2H
            MOV DL, 20H
            INT 21H
            MOV AH, 2H
            MOV DL, 8H
            INT 21H
            DEC SI
            DEC LENGTHS_DECODE
            DEC LENGTHS_DECODE
            JMP LZ

        DONE_DECODE:
          MOV AL, 20H
          MOV [SI],AL
          INC LENGTHS_DECODE
          INC SI
          MOV AL, 0DH
          MOV [SI],AL
          INC LENGTHS_DECODE
          RET
    CHECK_INPUT_DECODE:
      MOV AX, @DATA
      MOV DS, AX
      LEA SI, INPUT_DECODE
      MOV AX, 0H
      MOV AL, INDEX_DECODE[0]
      MOV SI, AX
      MOV DI, 0
      MOV COUNT_DECODE_PERCHAR, 0

      LD:
        MOV DL, INPUT_DECODE[SI]
        CMP DL, 20H
        JE ONE_CHAR_DONE
        MOV INPUT_CHECKER[DI], DL
        INC INDEX_DECODE
        INC COUNT_DECODE_PERCHAR
        INC COUNT_DECODE
        INC SI
        INC DI
        MOV BL, LENGTHS_DECODE[0]
        CMP COUNT_DECODE, BL
        JNE NOT_EXT
        JMP EXT
        NOT_EXT:
        JMP LD

        ONE_CHAR_DONE:
          INC INDEX_DECODE
          INC COUNT_DECODE
          CALL CHAR_CHECK


    CHAR_CHECK:
      MOV SI, OFFSET INPUT_CHECKER
      LX:
        MOV DL, [SI]
        INC SI
        CMP DL, 2EH
        JE DOT_DECODE
        CMP DL, 2DH
        JE STRIP_DECODE

        DOT_DECODE:
          MOV AL, COUNT_DECODE_PERCHAR[0]
          MOV DL, 1
          MUL DL
          MOV DL, AL
          MOV AL, COUNT_DECODE_PERCHAR[0]
          MUL DL
          ADD CODES, AL
          DEC COUNT_DECODE_PERCHAR
          CMP COUNT_DECODE_PERCHAR[0], 0
          JE DONE_PERCHAR
          JMP LX
        STRIP_DECODE:
          MOV AL, COUNT_DECODE_PERCHAR[0]
          MOV DL, 2
          MUL DL
          MOV DL, AL
          MOV AL, COUNT_DECODE_PERCHAR[0]
          MUL DL
          ADD CODES, AL
          DEC COUNT_DECODE_PERCHAR
          CMP COUNT_DECODE_PERCHAR[0], 0
          JE DONE_PERCHAR
          JMP LX
        DONE_PERCHAR:
          CMP CODES[0], 6
          JNE NOT_PRINT_A
          JMP PRINT_A
```

```
NOT_PRINT_A:
CMP CODES[0], 46
JNE NOT_PRINT_B
JMP PRINT_B
NOT_PRINT_B:
CMP CODES[0], 50
JNE NOT_PRINT_C
JMP PRINT_C
NOT_PRINT_C:
CMP CODES[0], 23
JNE NOT_PRINT_D
JMP PRINT_D
NOT_PRINT_D:
CMP CODES[0], 1
JNE NOT_PRINT_E
JMP PRINT_E
NOT_PRINT_E:
CMP CODES[0], 34
JNE NOT_PRINT_F
JMP PRINT_F
NOT_PRINT_F:
CMP CODES[0], 27
JNE NOT_PRINT_G
JMP PRINT_G
NOT_PRINT_G:
CMP CODES[0], 30
JNE NOT_PRINT_H
JMP PRINT_H
NOT_PRINT_H:
CMP CODES[0], 5
JNE NOT_PRINT_I
JMP PRINT_I
NOT_PRINT_I:
CMP CODES[0], 44
JNE NOT_PRINT_J
JMP PRINT_J
NOT_PRINT_J:
CMP CODES[0], 24
JNE NOT_PRINT_K
JMP PRINT_K
NOT_PRINT_K:
CMP CODES[0], 39
JNE NOT_PRINT_L
```

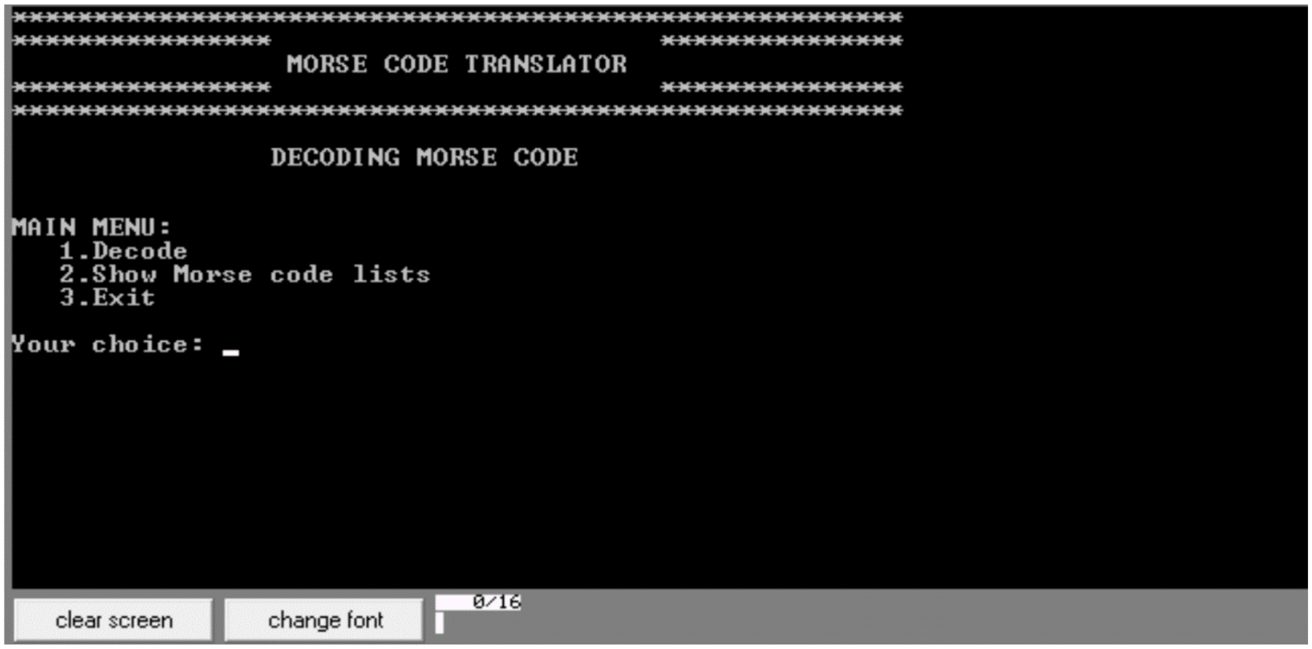
```
JMP PRINT_L
NOT_PRINT_L:
CMP CODES[0], 10
JNE NOT_PRINT_M
JMP PRINT_M
NOT_PRINT_M:
CMP CODES[0], 9
JNE NOT_PRINT_N
JMP PRINT_N
NOT_PRINT_N:
CMP CODES[0], 28
JNE NOT_PRINT_O
JMP PRINT_O
NOT_PRINT_O:
CMP CODES[0], 43
JNE NOT_PRINT_P
JMP PRINT_P
NOT_PRINT_P:
CMP CODES[0], 56
JNE NOT_PRINT_Q
JMP PRINT_Q
NOT_PRINT_Q:
CMP CODES[0], 18
JNE NOT_PRINT_R
JMP PRINT_R
NOT_PRINT_R:
CMP CODES[0], 14
JNE NOT_PRINT_S
JMP PRINT_S
NOT_PRINT_S:
CMP CODES[0], 2
JNE NOT_PRINT_T
JMP PRINT_T
NOT_PRINT_T:
CMP CODES[0], 15
JNE NOT_PRINT_U
JMP PRINT_U
NOT_PRINT_U:
CMP CODES[0], 31
JNE NOT_PRINT_V
JMP PRINT_V
NOT_PRINT_V:
CMP CODES[0], 19
```

```
            JNE NOT_PRINT_W
            JMP PRINT_W
            NOT_PRINT_W:
            CMP CODES[0], 47
            JNE NOT_PRINT_X
            JMP PRINT_X
            NOT_PRINT_X:
            CMP CODES[0], 51
            JNE NOT_PRINT_Y
            JMP PRINT_Y
            NOT_PRINT_Y:
            CMP CODES[0], 55
            JNE NOT_PRINT_Z
            JMP PRINT_Z
            NOT_PRINT_Z:
            CMP CODES[0], 60
            JNE NOT_PRINT_SP
            JMP PRINT_SP
            NOT_PRINT_SP:
            JMP PRINT_UNKNOWN

PRINT_A:
    MOV AH, 02H
    MOV DL, "A"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_B:
    MOV AH, 02H
    MOV DL, "B"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_C:
    MOV AH, 02H
    MOV DL, "C"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_D:
    MOV AH, 02H
    MOV DL, "D"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_E:
    MOV AH, 02H
    MOV DL, "E"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_F:
    MOV AH, 02H
    MOV DL, "F"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_G:
    MOV AH, 02H
    MOV DL, "G"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_H:
    MOV AH, 02H
    MOV DL, "H"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_I:
    MOV AH, 02H
    MOV DL, "I"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_J:
    MOV AH, 02H
    MOV DL, "J"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_K:
    MOV AH, 02H
    MOV DL, "K"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_L:
    MOV AH, 02H
    MOV DL, "L"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_M:
    MOV AH, 02H
    MOV DL, "M"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_N:
    MOV AH, 02H
    MOV DL, "N"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_O:
    MOV AH, 02H
    MOV DL, "O"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_P:
    MOV AH, 02H
    MOV DL, "P"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_Q:
    MOV AH, 02H
    MOV DL, "Q"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
PRINT_R:
    MOV AH, 02H
    MOV DL, "R"
    INT 21H
    MOV CODES[0], 0
    JMP CHECK_INPUT_DECODE
```

```
PRINT_S:
   MOV AH, 02H
   MOV DL, "S"
   INT 21H
   MOV CODES[0], 0
   JMP CHECK_INPUT_DECODE
PRINT_T:
   MOV AH, 02H
   MOV DL, "T"
   INT 21H
   MOV CODES[0], 0
   JMP CHECK_INPUT_DECODE
PRINT_U:
   MOV AH, 02H
   MOV DL, "U"
   INT 21H
   MOV CODES[0], 0
   JMP CHECK_INPUT_DECODE
PRINT_V:
   MOV AH, 02H
   MOV DL, "V"
   INT 21H
   MOV CODES[0], 0
   JMP CHECK_INPUT_DECODE
PRINT_W:
   MOV AH, 02H
   MOV DL, "W"
   INT 21H
   MOV CODES[0], 0
   JMP CHECK_INPUT_DECODE
PRINT_X:
   MOV AH, 02H
   MOV DL, "X"
   INT 21H
   MOV CODES[0], 0
   JMP CHECK_INPUT_DECODE
PRINT_Y:
   MOV AH, 02H
   MOV DL, "Y"
   INT 21H
   MOV CODES[0], 0
   JMP CHECK_INPUT_DECODE
PRINT_Z:
        MOV AH, 02H
        MOV DL, "Z"
        INT 21H
        MOV CODES[0], 0
        JMP CHECK_INPUT_DECODE
    PRINT_SP:
        MOV AH, 02H
        MOV DL, " "
        INT 21H
        MOV CODES[0], 0
        JMP CHECK_INPUT_DECODE
    PRINT_UNKNOWN:
        MOV AH, 02H
        MOV DL, "?"
        INT 21H
        MOV CODES[0], 0
        JMP CHECK_INPUT_DECODE
    EXT:
    MOV AH, 9H
    MOV DX, OFFSET PRESS_ANY_KEY
    INT 21H

    MOV AH, 1H
    INT 21H

    JMP MAIN_MENU
;Showing Morse Code Table
    _MORSECODE:
            CALL CLEAR_SCREEN
            MOV AH, 9H
            MOV DX, OFFSET MORSE_LISTS
            INT 21H
            JMP EXT

    REAL_EXIT:
    .EXIT
END
```

Code Results

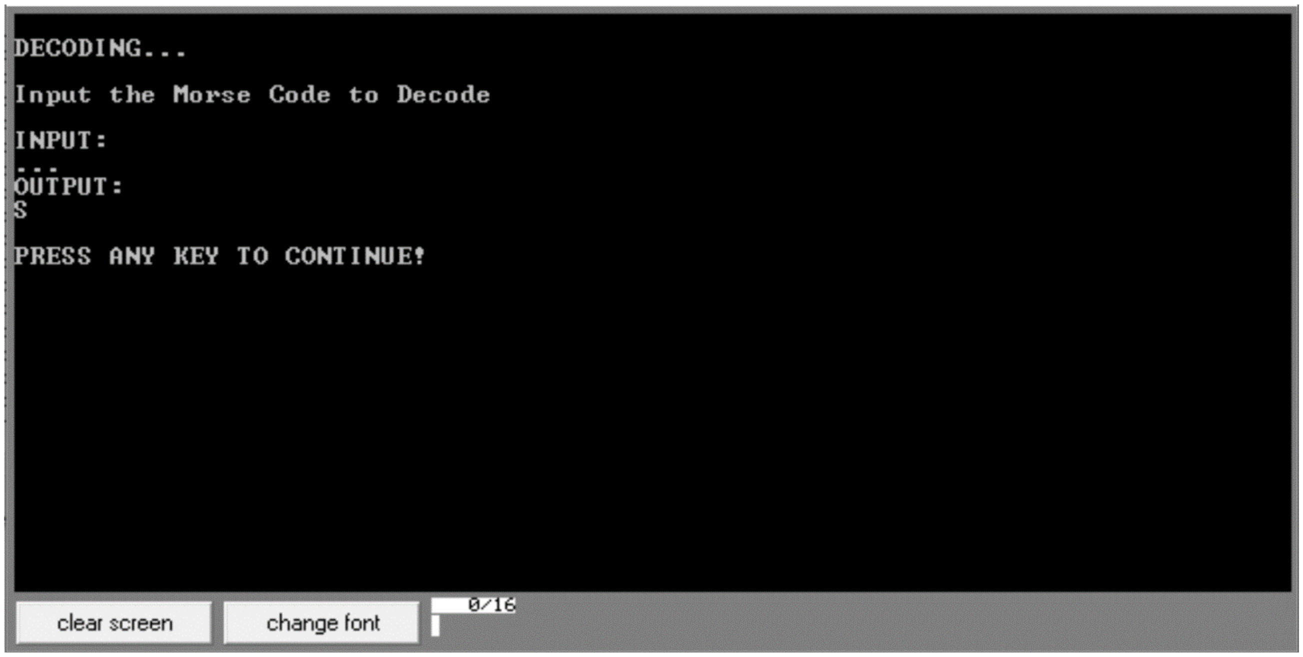
1. Main Menu



```
*******************************************************
****************                        ***************
                 MORSE CODE TRANSLATOR
****************                        ***************
*******************************************************

                DECODING MORSE CODE


MAIN MENU:
   1.Decode
   2.Show Morse code lists
   3.Exit

Your choice: _
```

Figure 4: The Mean Menu of the assembly Program

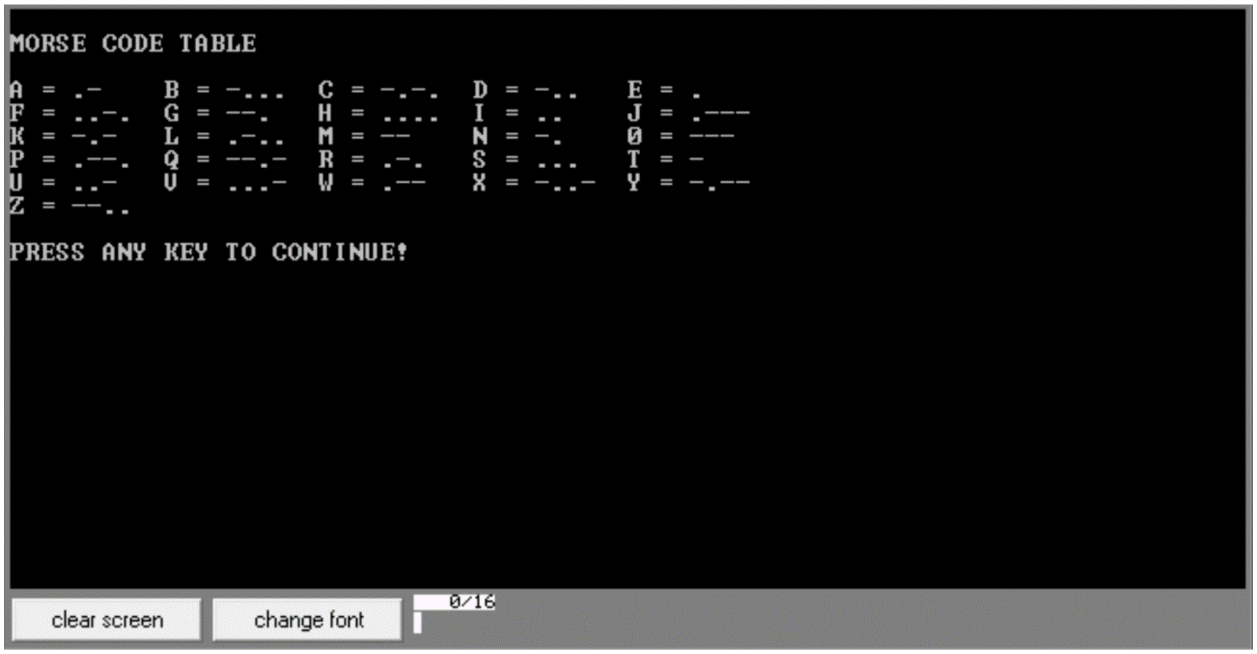
2. Decoding Part



```
DECODING...

Input the Morse Code to Decode

INPUT:
...
OUTPUT:
S

PRESS ANY KEY TO CONTINUE!
```

Figure 5: Decoding Part of the Assembly Program.

3. Morse Table



```
MORSE CODE TABLE

A = .-    B = -...  C = -.-.  D = -..   E = .
F = ..-.  G = --.   H = ....  I = ..    J = .---
K = -.-   L = .-..  M = --    N = -.    0 = ---
P = .--.  Q = --.-  R = .-.   S = ...   T = -
U = ..-   V = ...-  W = .--   X = -..-  Y = -.--
Z = --..

PRESS ANY KEY TO CONTINUE!
```

Figure 6:Displaying Morse Code Table