بسم الله الرحمن الرحيم

DEPARTMENT OF ELECTRICAL ENGINEERING

Batch (014)

UNIVERSITY OF GUJRAT

A WORLD CLASS UNIVERSITY

Presentation

Microprocessor

Instructor   Mam Yumna Bilal

Group Members

➤ Abdul Qayum

➤ Imran Khan

➤ Adnan Murtaza

➤ Usman Mahmood

➤ Ahmad Ali

➤ M.Shebaz Shreef

Binary and hex input/output

Binary input:

For binary Input, we assume a program reads In a binary number from the keyboard, followed by a carriage return.

The number actually Is a character string of O's and l's.

As each character Is entered, we need to convert it to a bit value, and collect the bits in a register.

Algorithm for Binary Input

Clear BX   /* BX will hold binary value */

Input a character   /* '0' or '1' */

WHILE character <> CR DO

Convert character to binary value

Left shift BX

Insert value into 1sb of BX

Input a character

END_WHILE

Demonstration·(for input 110)

Clear BX

BX | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Input character '1', convert to 1
AND AL,OFH
Left Shift BX

0   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

CF   BX

Insert value into lsb

BX | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

Input character '1' , convert to 1   AND AL,OFH
Left shift BX

CF | 0   BX | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

Insert value into lsb

BX | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1

Input character '0' convert to o   AND AL,OFH

 Left shift BX

        CF | 0 | BX | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

Insert value into lsb

BX = 0000 0000 0000 0110

BX contains 110b

The algorithm assumes (1) input characters are either "0", "1", or CR, and (2) at most 16 binary digits are input.AS a new digit Is Input, the previous bits in BX must be shifted to the left to make room; then an OR operation can be used to insert the new bit into BX.

```asm
XOR BX, BX              ; clear BX
MOV AH, 1               ;input char function
INT 21H                 ;read a character
WHILE_:
CMP AL, ODH              ;CR?
JE END_WHILE            ;yes, done
AND  AL, OFH             ;no convert to binary Value
SHL  BX, 1              ;make room for new value
OR   BL,AL              ;put value into BX
INT  21H                 ;read a character
JMP WHILE_               ;loop back
END_WHILE:
```

Binary Output

Algorithm for Binary Output:

FOR 16 times DO

Rotate left BX   /* BX holds output value,

put msb into CF */

IF CF = 1

THEN

output '1'

ELSE

output '0'

END_IF,

END_FOR

```asm
Mov CX,16
Top:
        ROL   BX,1
        JNC   Next
         Mov   AH,2
         Mov   DL,1
         JMP   Ali
        Next:
                Mov  DL,2
                Mov   DL,0
         Ali:
                Int   21H
                Int   Top
     loop Top
```

Hex Input

Hex input consists of digits ("0" to "9") and letters ("A" to "F") followed by a carriage return.

For simplicity, we assume that (1) only uppercase letters are used, and (2) the user inputs no more than four hex characters.

The process of converting characters to binary values Is more Involved than it was for binary input.

BX must be shifted four times to make room for a hex value.

Algorithm for Hex Input

Clear BX   /* BX will hold input value */

input hex character

WHILE character <> CR DO

convert character to binary value

left shift BX 4 times

insert value into lower 4 bits of BX

input a character

END_WHlLE

Demonstration (for input 6AB)

Clear BX

BX `0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0`

Input '6', convert to 0110

Left, shift BX 4 times

CF `0` BX `0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0`

Insert value into lower 4 bits of BX

BX `0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0`

Input 'A', convert to Ah = 1010

Left shift BX 4 times

CF `0` BX `0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0`

Insert value into lower 4 bits of BX

BX `0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0`

Input 'B', convert to 1011

Left shift BX 4 times

CF **0**   BX `0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0`

Insert value into lower 4 bits of BX

BX `0 0 0 0 0 1 1 0 1 0 1 0 1 0 1 1`

BX contains 06ABh.

Here is the code:

```
XOR BX,BX               ;clear BX
MOV CL,4                ;counter for 4 shifts
MOV AH,1                ; input character function
INT 21H                 ; input a character
WHILE_:
```

```asm
CMP AL,ODH              ;CR?
JE END WHILE            ;yes, exit
 ;convert character to binary value
AL, 39H                    ;a digit?
CMP AL, 39H                ;a digit?
JG   LETTER.                ; no, a letter
; input is a digit
AND  AL,OFH                ;convert digit to binary value
JMP  SHIFT                 ;go to insert in BX
LETTER:
SUB AL, 37H                ;convert letter to binary value
SHIFT:
SHL BX, CL                  ;make room for new value
```

```
;insert value into BX
CR   BL, AL                          ;put vi!lue into low 4 bits
                                     ;of BX

INT   21H                            ; input a character
JMP   WHILE_                         ; loop until CR
END_WHILE:
```

Hex Output

BX contains 16 bits, which equal four hex digit values.

To output the contents of BX, we start from the left and get hold of each digit, convert it to a hex character, and output it.

Algorithm for Hex Output:

```
 FOR 4 times DO
  Move  BH to DL              /* BX holds output value */
Shift DL  4 times to the right
IF DL < 10
THEN
convert to character in '0' .. '9.
ELSE
convert to character in 'A' .. 'F'
END_IF
output character
Rotate BX left 4 times
END FOR
```

Demonstration (BX Contains 4CA9h)

BX = 4CA9h = 0100 1100 1010 1001

Move BH to DL

DL = 0100 1100

Shift DL 4 times to the right

DL = 0000 0100

convert to character and output

DL = 0011 0100 = 34h = '4'

Rotate BX left. 4 times

BX = 1100 1010 1001 0100

Move BH to DL

 DL = 1100 1010

Shift DL 4 times t0 the right

DL = 0000 1100

Convert to character and output

DL = 0100 0011 = 43h ='C'

Rotate BX left 4 times

BX = 1010 1001 0100 1100

Move BH to DL

DL = 1010 1001

Shift DL 4 times to the right

DL = 0000 1010

Convert, to character and output

DL = 0100 0010 = 42h ='B'

Rotate BX left"4 times

BX = 1001 0100 1100 1010

Move BH to DL

DL = 1001 0100

Shift DL 4 times to the right

DL = 0000 1001

Convert to character and output

DL = 0011 1001 =39h = '9'

Rotate BX 4 times to the left

BX = 0100 1100 1010 1001 = original contents