

Input/Output of Numbers

- A common task is to input or output numbers in ASCII format
- Output tasks:
 - Output an 8-bit value as a ASCII string in binary format (see 'pbin.asm' example on WWW page)
 - Output an 8-bit value as ASCII string in HEX format
 - Output an 8-bit value as ASCII string in DECIMAL format
- Input tasks:
 - Input a string representing an 8-bit number in Hex format
 - Input a string representing an two digit decimal number (unsigned)

BR 6/00

1

Output an 8 bit number in Hex Format

- Each Hex character represents 4-bits in a number.
 - 0000 = '0' (ASCII code = 30h)
 - 0001 = '1' (ASCII code = 31h)
 - 1001 = '9' (ASCII code = 39h)
 - 1010 = 'A' (ASCII code = 41h)
 - 1011 = 'B' (ASCII code = 42h)
 - 1111 = 'F' (ASCII code = 46h).
- If 4-bits is between 0-9, then ASCII = 30h + 4bits
- If 4-bits is between A-F, then ASCII = 37h + 4bits

BR 6/00

2

Output an 8 bit number in Hex Format

Approach: Write a subroutine called 'OutHex'. This will output the lower 4 bits of register 'AL' as an Hex digit to the screen.

To output an 8-bit value, the main routine will call 'OutHex' twice – once for the most significant HEX digit, and one for the least significant Hex digit.

```
out2hex  proc
;; output value in 'al' as 8 bit hex character
push ax          ; save al
shr  ax, 4        ; put upper 4 bits into lower
and  ax, 0x0F     ; upper 4 bits = 0
call OutHex      ; print most sig. hex digit
pop  ax          ; get back original al
and  ax, 0x0F     ; upper 4 bits = 0
call OutHex      ; print least sig. hex digit
out2hex  endp
```

BR 6/00

3

OutHex

Pseudo code for OutHex:

```
    if ( AL > 09H) jump to SKIP
    AL = AL + 30H
    Use Int21H to print character
    return

skip:  AL = AL + 37H
    Use Int21H to print character
    return
```

BR 6/00

4

Output a 16 bit hex number? 32 bits?

- How would you print out a 16 bit value?
 - Call this subroutine Out4Hex (out 4 hex digits). Output the value passed in AX
 - Save AX on stack, move AH to AL and use Out2Hex to print out 8-bit value in AL
 - Restore AX, and use Out2Hex to print out 8-bit value in AL
- How would you printout a 32-bit value?
 - Call 'Out4Hex' twice – once for each 16 bits of the 32-bit value.

BR 6/00

5

Decimal Output

- How would you output a number in Decimal format?
- Assume that AL contains a value between 0 and 99 and you want to print this out as a decimal value
- The value of the first digit is 'AL divided by 10' (quotient value of AL/10).
- The value of the 2nd digit is REMAINDER of AL divided by 10!!

BR 6/00

6

Input an 8-bit number in HEX format

- An 8-bit hex number will require two ASCII characters to represent it
- Need to get 4-bit value of digit from ASCII character code
- If ASCII is between 30H and 39H ('0' and '9'), then four-bit value is ASCII value – 30H.
- If ASCII is between 41H and 46H ('A' and 'F'), then four-bit value is ASCII value – 37H

BR 6/00

7

Input an 8-bit number in HEX format

Assume AX has the two ASCII digits that represent a HEX number

Example: AX = 4335 h, AH = 43h = 'C', AL=35h = '5'.

Want to convert this to AL = C5h.

```
in2hex  proc
        push ax          ;; save AX
        mov al,ah        ;; get most sig. char into AL
        call inhex       ;; convert ASCII hex code in AL to 4 bit value
        mov bl, al       ;; save in BL
        pop ax           ;; get AX back
        call inhex       ;; convert ASCII hex code in AL to 4-bit value
        shl bl,4         ;; shift bl to left to move lower 4bit to upper
        or al, bl        ;; combine upper and lower bits, AL has value!
        ret
in2hex  endp
```

BR 6/00

8

inhex Subroutine

Want to convert the ASCII code in AL that is a HEX digit to its 4-bit value

```
Pseudo code:  if (AL > 39h) jump to skip
               AL = AL - 30h
               return
skip:         AL = AL - 37H
               return
```

BR 6/00

9

Input an 8-bit number in Decimal format

Assume AX has the two ASCII digits that represent a DECIMAL number

Example: AX = 3731 h, AH = 38h = '7', AL=31h = '1'.

Want to convert this to AL = 71 (decimal) = 47h !!

Approach:

- Convert the most significant ASCII digit to its four bit value.
- Multiply this by 10 and save.
- Convert the least significant ASCII digit to its four bit value and ADD it to the value produced in 'b'!!

$$71 = 7 * 10 + 1 = 71 = 47h.$$

BR 6/00

10

Using a Link Library

- Your book has a library of subroutines that can be very useful, especially for your end of semester project
- The library is called 'irvine.lib'.
- To link your .obj code to this library, do:
 - Copy the 'irvine.lib' to your current directory
 - When 'link' prompts you for a library name, enter 'irvine.lib'

BR 6/00

11

irvine.lib Subroutines

- Section 4.7 in the the Irvine textbook lists some of the available routines
 - Most of them require parameters to be passed in AX or EAX
 - Subroutines are available for printing/reading numbers, time of day, generating random numbers, etc
- The Irvine subroutines that write strings or input strings expect or return a NULL-terminated string
 - Null-terminated means that the last byte is a 00h. This is the commonly accepted manner of terminating strings (used by most high-level programming languages).
- You MUST use **“.model small”** instead of **“.model medium”** when using *irvine.lib*
 - “.model small” means that all calls are to procedures in same segment.

BR 6/00

12

Example: Generate Random numbers

- See 'libexam.asm' on WWW page.
- When using external subroutines, you must declare them as "*extern*"

```
extrn Writestring:proc, Crlf:proc, Clrscr:proc, Writeint:proc  
extrn Random_range:proc, Randomize:proc, Readint:proc
```
