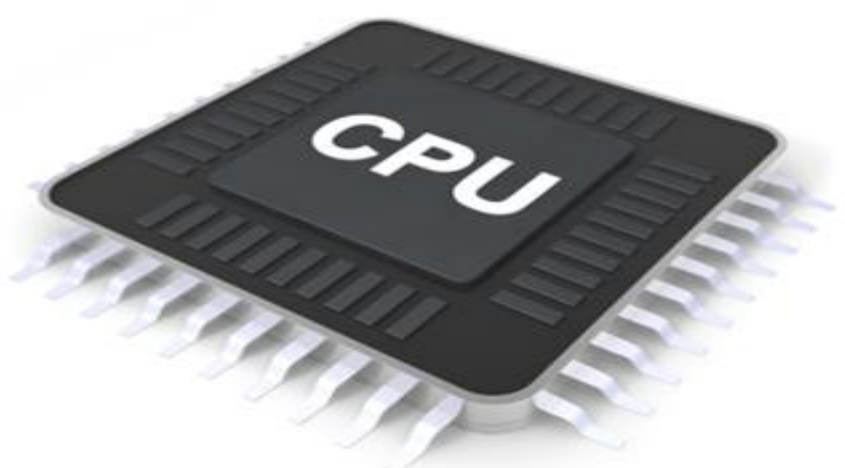

Syllabus

ENEE 3587

Microp Interfacing

Binary

- MSB – most significant bit and LSB – least significant bit

MSB															LSB
1	0	1	1	0	0	1	0	1	0	0	1	1	1	0	0
15															0

- n bits can represent 2^n different values.
- Binary can be in the following formats:
  - Unsigned Integer binary
    - BCD (Binary Coded Decimal)
  - Signed Integer
  - Floating Point or Real: not applicable to this course
- Use prefix to denote base:
  - Dec: no prefix
  - Binary: %, e.g.: %1011 = 11; No C notation
  - Hex: $, e.g.: $89AB = 35243; C notation: 0x, e.g.: 0x89AB
  - Octal: @, e.g.: @567 = 375; C notation: 0, e.g.: 0567

Unsigned Binary

- Given any unsigned number in base x:

$$a_{n-1}a_{n-2}a_{n-3}.......a_2a_1a_0$$

$a_i = 0, 1, …., x-1.$

Binary: $x = 2$; $a_i = 0$ or 1.

Hex: $x =$; $a_i = 0,1, …$

Octal: $x =$; $a_i =$

- To convert to base 10:

$$a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + a_{n-3}x^{n-3} +.......a_2x^2 + a_1x^1 + a_0x^0$$

- Example: %10111011

n :	8	7	6	5	4	3	2	1
a_i :	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0
	1	0	1	1	1	0	1	1
x^{n-1} :	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

2^n	Decimal	2^n	Decimal
2^0	1	2^8	256
2^1	2	2^9	512
2^2	4	2^{10}	1024 (1K)
2^3	8	$2^{11} = 2 \times 2^{10}$	2048 (2K)
2^4	16	$2^{12} = 2^2 \times 2^{10}$	4096 (4K)
2^5	32	$2^{13} = 2^3 \times 2^{10}$	8192 (8K)
2^6	64	$2^{20} = 2^{10} \times 2^{10}$	1048576 (1M)
2^7	128	2^{30}	1073741824 (1G)

Example:

%1011 1011 = $1*2^7+0*2^6+1*2^5+1*2^4+1*2^3+0*2^2+1*2^1+1*2^0=$
$128+0+32+16+8+0+2+1 = 187$

Shortcut: %1111 1111 – %0100 0100 = (256 – 1) – 64 – 4 =

%1011 0010 1001 1100 = ?

Conversion to Unsigned Binary

To convert a number, n, from decimal to base x:

1. Divide by base x: n/x = quotient + remainder
2. Record remainder
3. Divide quotient by base x
4. Repeat steps 2 and 3 until quotient is zero
5. Reorder remainders from last to first.

Works for any base.

Example: Convert 38 to binary (base 2)

$$\begin{array}{r} 19 \\ 2\overline{)38} \end{array} \quad r0$$

$$\begin{array}{r} 9 \\ 2\overline{)19} \end{array} \quad r1$$

$$\begin{array}{r} 4 \\ 2\overline{)9} \end{array} \quad r1$$

$$\begin{array}{r} 2 \\ 2\overline{)4} \end{array} \quad r0$$

$$\begin{array}{r} 1 \\ 2\overline{)2} \end{array} \quad r0$$

$$\begin{array}{r} 0 \\ 2\overline{)1} \end{array} \quad r1$$

$$0$$

Remainder

38= 100110 b

BCD

- Stored in 4-bit binary packets.
- 2 basic BCD formats:
  - Packed BCD - a string of decimal digits are stored in a sequence of 4-bit groups.

    example: 9502 would be stored as:

    9 5 0 2

    % 1001 0101 0000 0010
  - Unpacked BCD - digits are stored in the low-order half of an 8-bit group (what is in the high half is undefined - usually zero).

    example: 9502 would be stored as:

    9 5 0 2

    %uuuu1001 uuuu0101 uuuu0000 uuuu0010

Signed Binary: 2's Complement

- The numbers we will work with will be mostly 8 and 16 bit
- 2's complement: to create signed number consider MSB negative.
- Examples:

  %0110 0000 = $2^6 + 2^5 = 64 + 32 = 96$

  %1110 0000 = $-2^7 + 2^6 + 2^5 = -128$
- Hint: in signed binary whenever the first bit is 1 then the no is negative. When it 0 the number is poistive.
- Steps to convert a decimal no. into sb (2's complement):
  - Drop the sign and convert the decimal no. into binary.
  - If the sign of the given no. is positive do not change the binary number.
  - If sign is negative then flip 1's and 0's then add 1.

- Convert the 12 into 8 bit signed binary:
  - Drop the sign (doesn't matter)
  - 12 = %0000 1100
  - Sign is positive. 12= %0000 1100
- Convert the -12 into 8 bit signed binary :
  - Drop the sign.
  - 12 = %0000 1100
  - Sign is negative.
  - Flip 1's and 0's: %1111 0011
  - Add 1: %1111 0011 + 1 = %1111 0100
  - -12 = %1111 0100
- Convert 21d into 8 bit signed binary :
  - 21 = %0001 0101
- Convert -21d into 8 bit signed binary :
  - -21 = %1110 1011

Binary Number Range

- Expand a signed bin. by duplicating MSB.
  - E.g. -8 = %1111 1000. In 16 bits: -8 = %1111 1111 1111 1000
  - E.g. 8 = %0000 1000. In 16 bits: 8 = %0000 0000 0000 1000
- Difference between signed -8 and unsigned 8
- No of combinations of n bit no is 2^n
- Range of n-bit unsign bin. no. is: 0 to (2^n-1)
- Range of n-bit sign. bin. no. is: $-(2^{n-1})$ to $(2^{n-1}-1)$

  Example: show signed and unsigned range for n = 3
- What is the largest/smallest number that can be represented in
  - 8 bit binary?
  - 16 bit binary?
- What is the largest/smallest number that can be represented in
  - 8 bit signed bin?
  - 16 bit signed bin?

Addition and Subtraction

- Binary Addition: 0+0=0, 0+1=1, 1+0=1, 1+1=0 carry 1

					1				
	0	0	0	0	0	1	0	0	4d
+	0	0	0	0	0	1	1	1	7d
	0	0	0	0	1	0	1	1	11d
bit position:	7	6	5	4	3	2	1	0	

- Binary Subtraction: 0-0=0, 1-0=1, 1-1=0, 10-1=1
- Binary Subtraction by signed addition: A – B = A + (-B)

	0 0 0 0 1 1 0 0	→		0 0 0 0 1 1 0 0
–	0 0 0 0 0 0 1 1		+	________________
	b			sb

Base Conversions

- Octal is base 8, i.e. 2^3, and Hex is base 16, i.e. 2^4.
- Quick method: take 3 bits to represent 1 octal digit and 4 bits to represent one hex digit and vice-versa. Examples

  Binary to Hex:

  %0110 1011 0111

  $ 6 B 7

  Hexadecimal-to-binary:

  $ A 1 9

  %1010 0001 1001

- Bin and hex conversions shortcuts

  16 = 2^4 = (4)(4) = $10

  1K = 2^{10} = $(2^4)(2^4)(2^2)$ = ($10)($10)(4) = $400

  1M = 2^{20} = $(2^{10})(2^{10})$ = ($400)($400) = (4)($100)(4)($100) = ($10)($10000) = $100000

Signed Conversions

- Popular sizes:
  - 8 bits = ? hex digits,
  - 16 bits = ? hex digits,

- 8 bits: unsigned
  - 00 h = %0 = 0
  - 80 h = %1000 0000 =
  - 7F h = %0111 1111 =
  - FF h = %1111 1111 =

- 16 bits: unsigned
  - 0000 h = %0 = 0
  - 8000 h = %1000 0000 0000 0000 =
  - 7FFF h = %0111 1111 1111 1111 =
  - FFFF h = %1111 1111 1111 1111 =

- 8 bits: signed
  - 00 h = %0 = 0
  - 80 h = %1000 0000 =
  - 7F h = %0111 1111 =
  - FF h = %1111 1111 =

- 16 bits: signed
  - 0000 h = %0 = 0
  - 8000 h = %1000 0000 0000 0000 =
  - 7FFF h = %0111 1111 1111 1111 =
  - FFFF h = %1111 1111 1111 1111 =