Panel 1

Prior to Le06

Number Systems

Mainly Binary

ME430 Mechatronics

1

Panel 2

Binary and bases other than ten.

It is said that there are 10 kinds of people in the world...
those who understand binary...and those that don't.
-- author unknown.

2

Panel 3

Decimal, Base 10

The 10 symbols --> 0 1 2 3 4 5 6 7 8 9

Multiplying factor based on location:

Digit	Nth	…	4th	3rd	2nd	1st digit
Multiplying factor	10^{N-1}	…	10^3=1000	10^2=100	10^1=10	10^0=1

Example: 3105 is expressed as

$$3*10^3 + 1*10^2 + 0*10^1 + 5*10^0 = 3105$$

Decimal numbers may also sometime be given using the notation $(3105)_{10}$ but usually base 10 numbers are written are the default when you see any number

3

Panel 4

Binary system: system of counting based on 2 symbols or digits (Base 2)
0 and 1

0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011

4

Panel 5

Binary, Base 2

The 2 symbols --> 0 1

Multiplying factor based on location:

Digit	Nth	…	4th	3rd	2nd	1st digit
Multiplying factor	2^{N-1}	…	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$

Converting from Binary to Decimal:
Example: 0b01100111 is expressed as

$$0*2^7+1*2^6+1*2^5+0*2^4+0*2^3+1*2^2+1*2^1+1*2^0 = (103)_{10}$$

Binary numbers may also sometime be given using the notation $(01100111)_2$ but usually I try to use the 0b01100111 style notation since that is the form used in PICs

5

Panel 6

Converting from Decimal to Binary:

Method 1:

Subtract decreasing powers of 2 from the original value:

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

Step 1: Obtain decimal value N.

Example: Given decimal value: 345 Find the binary equivalent.

$345 - 2^8 = 345 - 256 = 89$ → 256 → 1 0000 0000

$89 - 2^6 = 89 - 64 = 25$ → 64 → 0 0100 0000

$25 - 2^4 = 25 - 16 = 9$ → 16 → 0 0001 0000

$9 - 2^3 = 9 - 8 = 1$ → 8 → 0 0000 1000

$1 - 2^0 = 1 - 1 = 0$ → 1 → 0 0000 0001

adding up the total

$1\ 0101\ 1001_2 = 345_{10}$

6

Panel 7

Method 2:
While subtraction of powers of 2 is easy to apply for relatively small numbers, a more efficient algorithm to find the binary equivalent is to continuously divide a number by 2. At each division, if there is a remainder, a 1 is written; if there is no remainder, a 0 is written. This continues until a value of 1 is reached.

Let's see an example converting 345 to binary:

Example:	345÷2 =	172 r 1	Lowest bit		
	172÷2 =	86 r 0			
	86÷2 =	43 r 0			
	43÷2 =	21 r 1			
	21÷2 =	10 r 1			
	10÷2 =	5 r 0			
	5÷2 =	2 r 1		Write number from highest to lowest bit	
	2÷2 =	1 r 0			
	1÷2 =	0 r 1	Highest bit	⑦	$1\,0101\,1001_2$

7

Panel 8

What is 93 in binary?

101

8

Panel 9

Best methods for converting:

#1. Memorize some of the low numbers 0000-1111 is plenty

#2. Figure out how it works on you calculator

#3. Figure out how to use Excel

#4. Figure out how to use Maple, Matlab, or other math programs

#5. Find a good internet link

9

Panel 10

http://acc6.its.brooklyn.cuny.edu/~gurwitz/core5/r

10

Panel 11

Hexadecimal system:

More commonly used for programming, hexadecimal is based on a base-16 system. It groups four bits of a binary number together to represent 1 of 16 possible symbols.

0, 1, 2, 3, 4, 5, 6, 7 ,8, 9, A, B, C, D, E, and F

binary	hexadecimal		binary	hexadecimal
0000	0		1000	8
0001	1		1001	9
0010	2		1010	A
0011	3		1011	B
0100	4		1100	C
0101	5		1101	D
0110	6		1110	E
0111	7		1111	F

11

Panel 12

Hexadecimal, Base 16

The 16 symbols --> 0 1 2 3 4 5 6 7 8 9 A B C D E F

Multiplying factor based on location:

Digit	Nth	…	3rd	2nd	1^{st} digit
Multiplying factor	16^{N-1}	…	$16^2 = 256$	$16^1 = 16$	$16^0 = 1$

Example: 0x3EB $3*16^2+E*16^1+B*16^0$

(which may make more sense as) $3*16^2+14*16^1+11*16^0 = 1003_{10}$

Hexadecimal numbers may also sometime be given using the notation $(3EB)_{16}$ but usually I try to use the 0x3EB style notation since that is the form used in PICs

12

Panel 13

How would the binary number 0100 0111 0011_2 be written in hexadecimal?

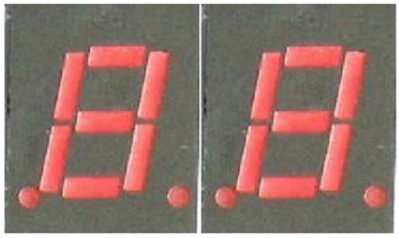
13

Panel 14

Binary Coded Decimal

binary	decimal	BCD
0000	0	0000
0001	1	0001
0010	2	0010
0011	3	0011
0100	4	0100
0101	5	0101
0110	6	0110
0111	7	0111
1000	8	1000
1001	9	1001
1010	10	0001 0000
1011	11	0001 0001
1100	12	0001 0010

Common 7 Segment display:



14

Panel 15

How would the binary number 010001110011_2 be written in BCD?

15

Panel 16

A Focus on binary and binary operations

16

Panel 17

Binary Addition:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 plus a carry out to the next significant bit

Example: Find the sum of these two binary numbers using binary and decimal:
$0010\ 1101_2 + 0100\ 1011_2 = \quad 45_{10} + 75_{10}$

Panel 18

Negative Binary Numbers

1s complement form.
The ones complement is formed by flipping each bit of the corresponding positive number. By doing this the most significant bit (bit 7) becomes a 1 and indicates a negative value. However the bits that represent the magnitude are quite different from the true magnitude form.

Example: -12_{10} ⊘ bit flip($0000\ 1100_2$) ⊘ $1111\ 0011_2$ (1s complement form)

2s complement form. The 2s complement form also indicates negative when the most significant bit (bit 7) equals 1. To find the 2s complement of a number, take the corresponding positive value of it and form the 1s complement and then add 1.

Example -12_{10} ⊘ bit flip($0000\ 1100_2$) + $0000\ 0001_2$ ⊘ $1111\ 0011_2 + 0000\ 0001_2$

⊘ $1111\ 0100_2$ (2s complement form)

18

Panel 19

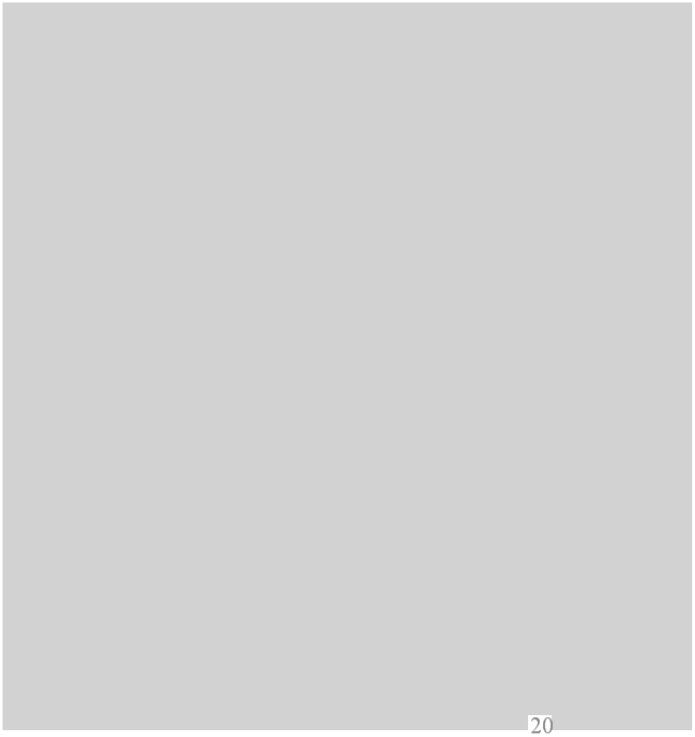
2's complement

8	1000
7	111
6	110
5	101
4	100
3	11
2	10
1	1
0	0
-1	
-2	
-3	
-4	
-5	
-6	
-7	
-8	

19

Panel 20

Example: To express -67 as a 2s-complement signed integer



20

Add 68 to see if it equals 1

Panel 21

Short cut for converting to 2's complement

Start at the right of the binary and write number
when you get to the first 1 write it then invert the rest

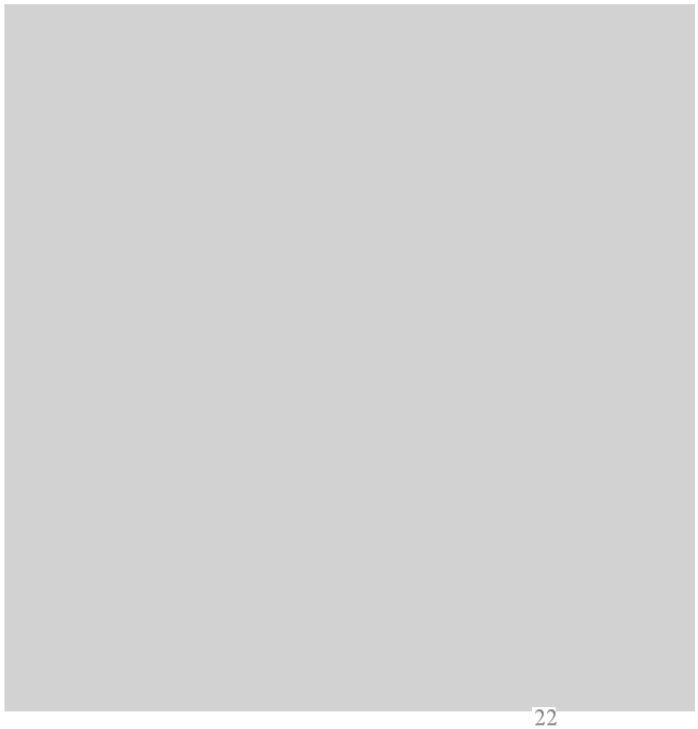
-6 = - 0000 0110

2's = 1111 1010

21

Panel 22

Show how a computer calculates -3_{10} - 4_{10} using 8 bit signed integers.



Try to use the shortcut approach

22

Panel 23

Overflow Example:

Use 2s complement to complete the following calculation:

$102_{10} + 36_{10} = 0110\ 0110_2 + 0010\ 0100_2$

$102 + 36 = 0b01100110 + 0b00100100$

```
  1 1   1
  0110 0110
+ 0010 0100
  1000 1010 2
```

If interpreted as an unsigned integer result the number represents: 138_{10}
If interpreted as a signed 2s compliment number this represents -118_{10}

(This would cause an error or flag called Overflow to be set)

Overflow occurs when either adding together two positive numbers or two negative numbers that cause the allowed number range to be exceeded.

Notice that Overflow is different than a Carry. It represents an Out of Range calculation.

23

Panel 24

Using 2s complement complete the following calculation:

$-45_{10} - 26_{10} = -(0010\ 1101_2) - (0001\ 1010_2)$

24

Panel 25

Carry: Occurs when a 1 is carried out of bit 7.

Did a Carry out occur in the last example?

Overflow: occurs if both operands were positive and the result is negative
or if both operands were negative and the result is positive.

Did an Overflow occur in the last example?

25