Penn State Abington
CMPEN 271
Lecture Set #3

Topics:

- Review Digital System Concepts — Video part 1 of 4 ←

- Base 2, 8 (octal), 16 (hex) conversions to Base 10
- Base 10 to Base 2, 8, 16 — Video part 2 of 4

- BCD
- ASCII Code,Unicode — Video part 3 of 4
- Parity Bits
- Serial and Parallel Communication

- Exercises and Review Questions — Video part 4 of 4

Homework #1 Update

- HW #1 -- check schedule for due date (digitize signal; answering machine). **Upload solutions to Canvas (PPT, Word, or PDF)**.
- Any problems or questions about HW#1?
- Please ask questions.
- There is no penalty of any kind for asking questions.
- **HW format**: include name, date, course, number and title of HW or problem; brief description or objective (a few sentences). Always include problem statement in every homework (what is the problem are you trying to solve?)
- **Schedule time** each week to review lecture and complete HW

MultiSim Software Update

- MultiSIM software is a requirement for CMPEN 271 (and CMPEN 275)
- MultiSIM will also be required for EE 210 in spring (EEs and CMPEN majors)
- Check with instructor before purchasing (check for free version first)
- Check syllabus for links to Multisim software
- What do you do if you have Apple Mac machine?
  - Use Multisim on campus labs
  - Window emulator for Mac
  - Borrow Windows machine and install Multisim
  - Other digital circuit simulators (apps or browser-based)
    - Example: LogicWorks

Review Digital System Concepts

- What is a **digital system**?
- What are the **advantages** of a digital system?
- What are the **disadvantages** of a digital system?
- What is **bit**?
- How are bits **physically** represented? Examples?
- Is digital always **better**?
- What is your **response to the person** who says analog sounds better than digital?
- **Ask questions**. Check Youtube videos for "digital counter"

Penn State Abington
CMPEN 271
Lecture Set #3

(Number systems, BCD, ASCII)

Topics:

- Review Digital System Concepts — Video part 1 of 4

- Base 2, 8 (octal), 16 (hex) conversions to Base 10
- Base 10 to Base 2, 8, 16 — Video part 2 of 4 ←

- BCD
- ASCII Code,Unicode — Video part 3 of 4
- Parity Bits
- Serial and Parallel Communication

- Exercises and Review Questions — Video part 4 of 4

Problem Statement

Digital **hardware and software systems** often require the **conversion** of numeric values from one base to another (for example: convert a base 10 (decimal) number to a base 2 (binary) number).

As a **Penn State co-op student**, you have been asked to develop an **algorithm (set of steps) in C** (or any other programming language) to **convert** a base 10 number to a base 2 number and also a base 2 number to a base 10 number for use in a new calculator for possibly a handheld computer. **Part 1** of your task is to evaluate existing conversion calculators, and **Part 2** is to create your own program for conversion using appropriate algorithms and lessons learned from existing software. The software should be **well organized and user friendly**.

Base 2 (binary) to Base 10 (decimal) - 1

10^2 10^1 10^0 10^{-1} 10^{-2}

Base 10: **3 7 8 . 4 5** $= 3 * 10^2 + 7 * 10^1 + 8 * 10^0 +$

$+ 4 * 10^{-1} + 5 * 10^{-2}$

2^2 2^1 2^0 2^{-1} 2^{-2}

Base 2: **1 0 1 . 1 1** $= 1 * 2^2 + 0 * 2^1 + 1 * 2^0 +$

$+ 1 * 2^{-1} + 1 * 2^{-2}$

= 5 . 75 (in base 10)

Base 2 (binary) to Base 10 (decimal) - 2

…	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	…
	8	**4**	**2**	**1**		**0.5**	**0.25**	**0.125**	
	1	1	0	1	.	1	0	1	

Radix ("decimal") point

$= (\ \ ?\ \ )_{10}$

- Only 2 allowed symbols in base 2 ----- "0" and "1"
- Any number in base 10 can be represented in base 2 (1's and 0's)
- Any number in base 2 can be represented in base 10 (or any base)
- "Radix" is another word for "base". Example, base 10 is radix 10
- Recall: $2^3 = 2*2*2$ and $2^{-3} = 1 / 2^3$

Base 2 (binary) to Base 10 (decimal) - 3

Example 1

$(\ 100101.011\ )_2 = (\ ?\ )_{10}$

$= 1*32 + 0 + 0 + 1*4 + 0 + 1 + 0 + 1*0.25 + 1*0.125 =$ $37.375\ _{10}$

Example 2

$(\ 11100.1001\ )_2 = (\ ?\ )_{10}$

$= 1*16 + 1*8 + 1*4 + 0 + 0 + 1*0.5 + 0 + 0 + 1*0.0625 =$ 28.5625_{10}

Base 8 (octal) to Base 10 (decimal) - 1

Example: (237.4)$_8$ = (?)$_{10}$

...	8^3	8^2	8^1	8^0		8^{-1}	8^{-2}	...
	512	64	8	1		0.125	0.015625	
	0	2	3	7	.	4	0	

$$2 * 64 + 3 * 8 + 7 * 1 + 4*(1/8) = 159.5_{10}$$

- Only 8 allowed symbols in base 8 ----- “0” through “7”
- Any number in base 10 can be represented in base 8
- Any number in base 8 can be represented in base 10 (or any base)

Base 8 (octal) to Base 10 (decimal) - 2

Example 1

$(106.5)_8 = (?)_{10}$

$= 1*64 + 0 + 6*1 + 5*0.125 = 70.625_{10}$

Example 2

$(77.7)_8 = (?)_{10}$

$= 7*8 + 7*1 + 7*0.125 = 63.875_{10}$

Base 16 (hex) to Base 10 (decimal) - 1

Example: (8A.B)$_{16}$ = (?)$_{10}$

… 16^3	16^2	16^1	16^0		16^{-1}	16^{-2} …
4096	**256**	**16**	**1**		**0.0625**	**0.0039**
		8	A	•	B	

$$8 * 16 + 10 * 1 + 11 * (1/16) = 138.6875_{10}$$

- Only 16 allowed symbols in base 16 ----- “0” through “9”, A (=10), B (=11), C (=12), D (=13), E (=14), F (=15)
- Any number in base 10 can be represented in base 16
- Any number in base 16 can be represented in base 10 (or any base)

Base 16 (hex) to Base 10 (decimal) - 2

<u>Example 1</u>

$(1F.8)_{16} = (?)_{10}$

$= 1*16 + 15*1 + 8*(1/16) = 31.5_{10}$

<u>Example 2</u>

$(ABC.D)_{16} = (?)_{10}$

$= 10*256 + 11*16 + 12*1 + 13*(1/16) = 2748.8125_{10}$

Powers of 2

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 = 16$

$2^5 = 32$

$2^6 = 64$

$2^7 = 128$

$2^8 = 256$

$2^9 = 512$

$2^{10} = 1024 =$ **1 K**

$2^{11} = 2048 = 2^1 * 2^{10} = 2K$

$2^{12} = 4096 = 2^2 * 2^{10} = 4K$

$2^{13} = 8192 = 2^3 * 2^{10} = 8K$

$2^{14} = 16384 = 2^4 * 2^{10} = 16K$

$2^{15} = 32748 = 2^5 * 2^{10} = 32K$

$2^{16} = 65536 = 2^6 * 2^{10} = 64K$

$2^{20} = 1048576 = 2^{10} * 2^{10} =$ **1 M**

$2^{30} = 2^{10} * 2^{20} =$ **1 G**

Counting in Base 2, 8, 16

Base 10	Base 2	Base 8	Base 16
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11

Range of values

- **Given 3 bits**, what is the **range of positive values** from lowest to highest? (provide answer in both binary and decimal)
- **Given 3 bits**, how many **unique codes** can be generated?
- **Given n bits**, what is 1) number of unique codes, 2) range of values?
- How many bits needed to **represent cards in a standard playing deck**?
- How many **phone numbers** are theoretically possible (without area code)? xxx-xxxx format

LSB and MSB

LSB = Least Significant Bit
MSB = Most Significant Bit

msb lsb

1 0 0 1 0 . 0 1 1

msb lsb

1 1 1 0

Converting Base 10 to Base 2

Example: $\mathbf{(\ 37.6\ )_{10} = (\ ?\ )_2}$

Technique: Split decimal number into integer and fractional parts and convert each part separately. Combine when done.

Step 1 - Convert integer part by repeated division by 2 and save remainder.

37 / 2 = 18	+ Rem of	1	
18 / 2 = 9	+ Rem of	0	
9 / 2 = 4	+ Rem of	1	
4 / 2 = 2	+ Rem of	0	
2 / 2 = 1	+ Rem of	0	
1 / 2 = 0	+ Rem of	1	msb

Step 2 - Convert fractional part by repeated multiplication by 2 and save integer result.

0.6 * 2 =	1	+ 0.2	
0.2 * 2 =	0	+ 0.4	
0.4 * 2 =	0	+ 0.8	
0.8 * 2 =	1	+ 0.6	
0.6 * 2 =	1	+ 0.2	(repeats forever)

Final answer = 100101 . 10011 $_2$ (verify)

Converting Base 10 to Base 8

Example: $\mathbf{(\ 37.6\ )_{10} = (\ ?\ )_8}$

<u>Technique</u>: Split decimal number into integer and fractional parts and convert each part separately. Combine when done.

<u>Step 1</u> - Convert integer part by repeated division by 8 and save remainder

$37 / 8 = 4 + \text{Rem of } 5$

$4 / 8 = 0 + \text{Rem of } 4$ msd

<u>Step 2</u> - Convert fractional part by repeated multiplication by 8 and save integer result.

$0.6 * 8 = 4 + 0.8$

$0.8 * 8 = 6 + 0.4$

$0.4 * 8 = 3 + 0.2$

$0.2 * 8 = 1 + 0.6$

$0.6 * 8 = 4 + 0.8$ (repeats forever)

Answer = 45 . 46314 $_8$ (verify)

Converting Base 10 to Base 16

Example: $\mathbf{(37.6)_{10} = (?)_{16}}$

Technique: Split decimal number into integer and fractional parts and convert each part separately. Combine when done.

Step 1 - Convert integer part by repeated division by 16 and save remainder

$37 / 16 = 2 \ + \text{Rem of } \boxed{5}$

$2 / 16 = 0 \ + \text{Rem of } \boxed{2}$ msd

Step 2 - Convert fractional part by repeated multiplication by 16 and save integer result.

$0.6 * 16 = \boxed{9} + 0.6$

$0.6 * 16 = \boxed{9} + 0.6$

$0.6 * 16 = \boxed{9} + 0.6$

$0.6 * 16 = \boxed{9} + 0.6$ (repeats forever; stop after 4 places)

Answer = 25 . 9999 $_{16}$ (verify)

Useful Trick: Base 2 → 8

Split into groups of 3 bits (start at radix point)

<u>Example</u>: $\mathbf{1100111001.10111_2 = (\ ?\ )_8}$

001 100 111 001 . 101 110

add zero(s) if needed

add zero(s) if needed

$\mathbf{1471.56_8}$

(Can also go in reverse -- base 8 to base 2)

Useful Trick: Base 2 → 16

Split into groups of 4 bits (start at radix point)

<u>Example</u>: $\mathbf{1100111001.10111_2 = (\ ?\ )_{16}}$

0011 0011 1001 . 1011 1000

add zeros

add zeros

$\mathbf{339\,.\,B8_{16}}$

(Can also go in reverse -- base 16 to base 2)

Useful Trick: Base 8 → 16

Convert to groups of 3 bits (start at radix point), to convert to base 2, then group in sets if 4

Example: $\mathbf{1471\,.\,56_{8} = (\ ?\ )_{16}}$

$\mathbf{001\ \ 100\ \ 111\ \ 001\ .\ 101\ \ 110_{2}}$

(regroup by 4) $\mathbf{0011\ \ 0011\ \ 1001\ .\ 1011\ \ 1000_{2}}$

$\mathbf{339\,.\,B8_{16}}$

(Can also go in reverse -- base 16 to base 8)

Penn State Abington
CMPEN 271
Lecture Set #3

(Number systems, BCD, ASCII)

<u>Topics:</u>

- Review Digital System Concepts — Video part 1 of 4

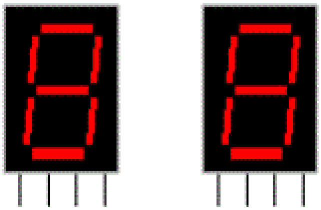
- Base 2, 8 (octal), 16 (hex) conversions to Base 10
- Base 10 to Base 2, 8, 16 — Video part 2 of 4

- BCD
- ASCII Code,Unicode — Video part 3 of 4 ←
- Parity Bits
- Serial and Parallel Communication

- Exercises and Review Questions — Video part 4 of 4

BCD Code

BCD = **B**inary **C**oded **D**ecimal



Convert Base 10 (decimal) to BCD
Generate 4 bits for each decimal digit

Example: $\mathbf{91803_{10} = (\ ?\ )_{BCD}}$

$\mathbf{= 1001\ 0001\ 1000\ 0000\ 0011_{BCD}}$

- BCD used in some calculators to facilitate input and output. Explain.
- BCD codes 1010, 1011, 1100, 1101, 1110, and 1111 never used
- Is BCD binary? Yes. Is BCD the same as base 2? No.

ASCII Code-1

- ASCII = American Standard Code for Information Interchange
- ASCII is a 7-bit code -- how many unique values?

Partial listing..

char	binary	char	binary
A	1000001	blank	0100000
B	1000010	.	0100001
C	1000011	+	0101011
…		=	0111101
0	0110000	return	0001101
1	0110001	linefeed	0001010
2	0110010	...	
...			

(Note: convert above codes to decimal, octal, hex)

ASCII Code-2

- ASCII codes are stored as 1 byte (8 bits); extra bit can be zero (or parity bit)
- What is difference between ASCII file and a text file?
- How do you create a text file in Windows OS?
- Is MS Word doc file a text or ASCII file? Explain.
- What is difference between a text file and an executable file?
- What is a binary file?
- Is a C++/Java source program a text/ASCII file?
- Each keystroke on keyboard produces a ASCII character?
- How many characters can fit on CD-ROM? DVD? 8GB Flash drive?
- Create a file in NotePad with the following contents:

  CMPEN 271
- How many bytes is above file? Demonstrate on PC.
- What is Unicode? How many bits? How many codes?

- **<u>Computer Lab Exercise</u>**: Create an ASCII file in NotePad

Parity Bits

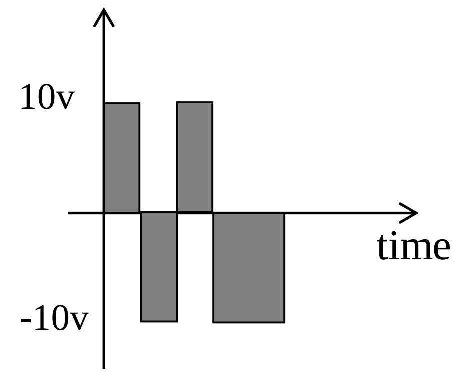
- Parity bit is an extra bit included in a binary message to help detect **transmission errors**
- A parity bit can be "even" or "odd"
- For **even parity**, the extra bit is set so that the total number of 1's in the binary message (including the parity bit) is an even number
- For **odd parity**, the extra bit is set so that the total number of 1's in the binary message (including the parity bit) is an odd number
- Odd or even parity are **equally valid**. Both the transmitter and receiver must agree on either even or odd parity.
- We will always insert the parity bit in the **msb** of the binary data
- **Receiver will check parity** (count 1's) to see if there was an error. If error is detected, the receiver will request transmitter to resend data.
- **Example**: Send an ASCII character for "A" with an even parity bit. Place the parity bit at msb location.

A = 1000001 (without even parity bit)

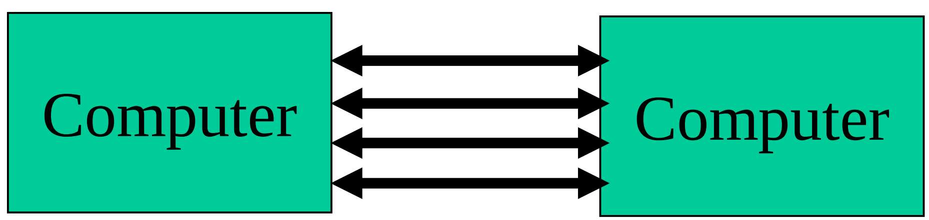
A = 01000001 (with even parity bit)

Serial and Parallel Communication

1. Serial (e.g. RS-232/USB)



2. Parallel



- Advantages and disadvantages? Compare serial versus parallel
- Is phone line serial or parallel?

Penn State Abington
CMPEN 271
Lecture Set #3

(Number systems, BCD, ASCII)

Topics:

- Review Digital System Concepts — Video part 1 of 4

- Base 2, 8 (octal), 16 (hex) conversions to Base 10
- Base 10 to Base 2, 8, 16 — Video part 2 of 4

- BCD
- ASCII Code,Unicode — Video part 3 of 4
- Parity Bits
- Serial and Parallel Communication

- Exercises and Review Questions — Video part 4 of 4 ←

Review Questions

- If you are offered a **$60K** starting salary for a job, what is the meaning of “K” in this context?
- A resistor has a value of **10K ohms** -- what is “K”?
- You have just purchased **2G** of internal memory for your desktop PC -- what is the meaning of “G” here?
- Consider the bit pattern “**01010010**” found in the memory of a computer. Name 3 possible interpretations.
- Write your **name in ASCII** with an **<u>even</u>** parity bit. (Parity bit goes in msb position)

Exercises

Perform the following conversions by hand (verify with calculator):

#1. $(10101.01)_2 = (?)_{10}$

#2. $(64.45)_{10} = (?)_2$

#3. $(100101.011)_2 = (?)_8$

#4. $(37.6)_{10} = (?)_8 = (?)_{16}$

#5. $(AB)_{16} = (?)_{10}$

#6. $(3773.61)_8 = (?)_{16}$

#7. $(6190)_{10} = (?)_{BCD}$

#8. Express string "X = A + 20" in ASCII with odd parity.

Exercises

#1. $(1\ 0\ 1\ 0\ 1\ .\ 0\ 1)_2 = (?)_{10}$

Exercises

#2. $(64.45)_{10} = (\ ? \ )_2$

Exercises

#3. $(1\ 0\ 0\ 1\ 0\ 1\ .\ 0\ 1\ 1)_2 = (\ \ ?\ \ )_8$

Exercises

#4. $(\ 37.6\ )_{10} = (\ \ ?\ \ )_8 = (\ \ ?\ \ )_{16}$

Exercises

Perform the following conversions by hand (verify with calculator):

#5. $(\ A\ \ B\ )_{16} = (\qquad\qquad)_{10}$

Exercises

#6. $(3\ 7\ 7\ 3\ .\ 6\ 1)_8 = (\ ?\ )_{16}$

Exercises

Perform the following conversions by hand (verify with calculator):

#7. $(6\,1\,9\,0)_{10} = (\; ? \;)_{BCD}$

Exercises

#8. Express string "X = A + 20" in ASCII with odd parity. (include any blank spaces)

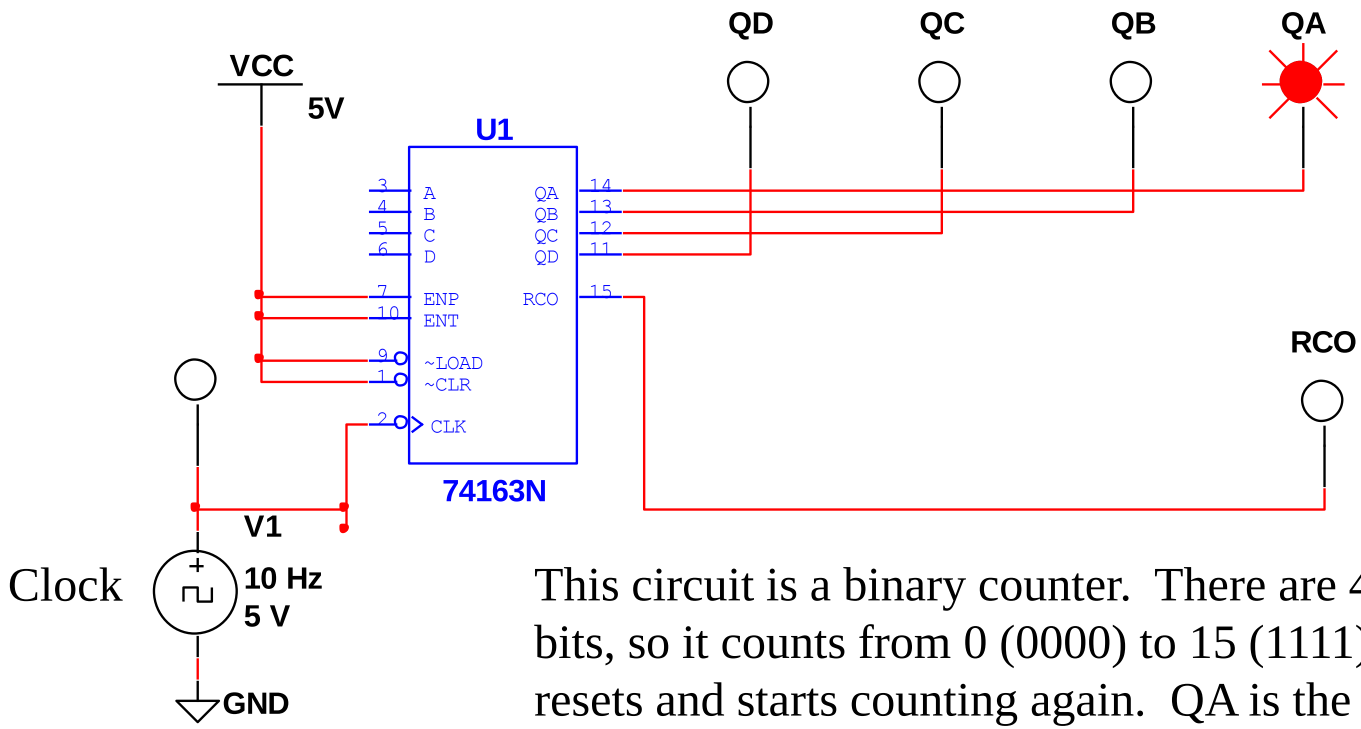
How many characters? How many bytes? How many bits?

From ASCII table (see web):

char	dec	hex	octal	binary (7-bit ASCII)	binary with odd parity (8 bits)	hex with odd parity
X	88	58	130			
sp	32	20	040			
=	61	3D	075			
sp	32	20	040			
A	65	41	101			
sp	32	20	040			
+	43	2B	053			
sp	32	20	040			
2	50	32	062			
0	48	30	060			

MultiSim Counter Circuit

74163 MSI Counter



This circuit is a binary counter. There are 4 output bits, so it counts from 0 (0000) to 15 (1111), then resets and starts counting again. QA is the LSB, and QD is the MSB. You can edit the frequency of the clock to make it slower or faster. (RCO output is used to connect to other circuits if needed.)

Download 74163 Counter demo file and run in Multisim

What you should know...

- Be able to convert by hand any number (with fraction) from base 2, 8,10, or 16 to the equivalent number in another base 2, 8, 10, or 16.
- Know shortcut to convert from among bases 2, 8, 16 (group bits by 3 or 4))
- Understand the meaning of and be able to identify lsb and msb.
- Know how to convert an arbitrary number between base 10 and BCD and BCD to base 10.
- Know how to count from 1 to 20_{10} in base 2, base 8, and base 16.
- Know powers of 2 from 0 to 20.
- Know definition of K, M, G in base 2.
- Using an ASCII chart be able to convert any character to its equivalent ASCII code represented in decimal, hex, and binary.
- Be able to generate an odd or even parity bit for a given binary number.
- Be able to explain the advantage of Unicode over ASCII.
- Given n bits, be able to determine the total number of possible codes.
- Given n bits, be able to determine the range of possible positive values.
- Describe the pros and cons of serial versus parallel communication.

Summary

- The binary or base 2 number system uses only two symbols → 1 and 0.
- Any number in base 10 can be converted to a base 2 value (or base 8 or base 16).
- A bit is a binary digit.
- Fractional numerical values can also be converted from base 10 to base 2 (and back).
- By grouping bits in a base 2 number, you can easily convert from base 2 to base 8 or to base 16.
- Base 8 is also known as "octal".
- Base 16 is also known as "hex"
- There are 8 allowable symbols for each digit in a base 8 number: 0 → 7
- There are 16 allowable symbols for each digit in a base 16 number: 0 → 9, A, B, C, D, E, F
- For a binary number with n bits, there are 2^n unique values or codes possible.
- The numeric range of an n-bit number is 0 to $2^n – 1$
- The least significant bit (LSB) of a binary number is the bit with the least positional weight, which is the right-most bit.
- The most significant bit (MSB) of a binary number is the bit with the highest positional weight (or the left-most bit).
- A BCD code is generated by converting each digit of a base 10 number to its corresponding value in 4 bits. A BCD value is not the same as a base 2 value.
- An ASCII code is the numeric representation of a single character. Text is stored and transmitted digitally in ASCII code. ASCII codes are generally 8 bits or 1 byte each. UNICODE is a 16-bit code and is an extension of ASCII . ASCII files can be created with MS NotePad.
- A parity bit is a bit added to a data message to performs error detection when the message is transmitted.
- Investigate other binary codes, such as Gray code. What is the advantage of a Gray code? (only one bit changes as you increment the values). Example: 000, 001, 011, 010, 110, 111, 101, 100.

Further Reading

- Mano, Kime, Logic and Computer Design Fundamentals, Prentice Hall,, Chapter 2.

- Tocci R., Digital Systems, Prentice Hall, see Chapter 2.

- Review www.howstuffworks.com 1) How Bits and Bytes Work", 2) "How Boolean Logic Works"
  see other tutorials on digital logic, Boolean algebra, electronics, computer technology)

- www.play-hookey.com/digital (tutorials on digital logic topics)

- www.whatis.com (general info. on computer topics)

- www.wikipedia.org

- Video tutorial (Engr Techn orientation)
  http://www.youtube.com/user/billkleitz

http://www.youtube.com/watch?v=s0pi-VoaZyU&list=PLE7E5C88AA6AEA0A2&index=3&feature=plpp_video