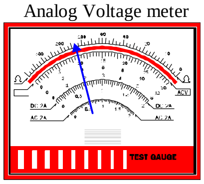
Digital Devices

- Integrated Circuits that operate on Digital Data are in 95% of every electrical powered device in the U.S.
- The theory of operation of these devices form a basis for many other courses in the EE/CS/COEN curriculum
- The job market for engineers and computer scientists with Digital Design skills is at an all time high and will continue growing.

BR 8/99

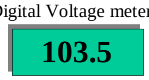
Analog versus Digital

Analog Voltage meter



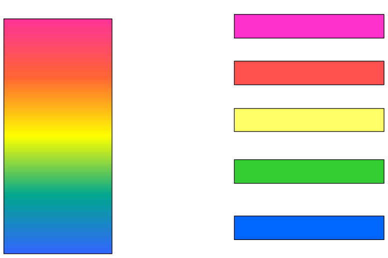
About 100

Digital Voltage meter



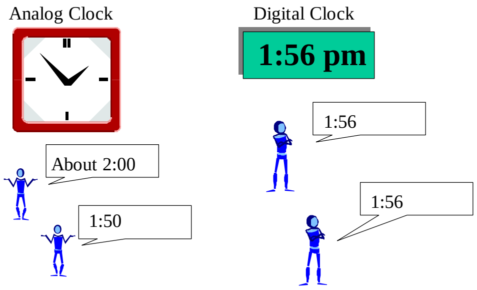
BR 8/99

Analog offers Continuous Spectrum Digital offer distinct Steps



BR 8/99

Analog has Ambiguity
Digital has only one interpretation



BR 8/99

Another Advantage of Digital Data

- Digital data can have additional data added to it to allow for detection and correction of errors
  - Scratch a CDROM - will still play fine
  - Scratch, stretch an analog tape - throw it away
- Digital data can be transmitted over a medium that introduces errors that are corrected at receiving end
  - Satellite transmission of DirectTV - each ‘screen’ image is digitally encoded; errors corrected when it reaches your digital Set Top receiver, shows up as a ‘Perfect’ Picture.

BR 8/99

Binary Representation

- The basis of all digital data is binary representation.
- Binary - means ‘two’
  - 1, 0
  - True, False
  - Hot, Cold
  - On, Off
- We must be able to handle more than just values for real world problems
  - 1, 0, 56
  - True, False, Maybe
  - Hot, Cold, LukeWarm, Cool
  - On, Off, Leaky

BR 8/99

Number Systems

- To talk about binary data, we must first talk about number systems
- The decimal number system (base 10) you should be familiar with!
  - A digit in base 10 ranges from 0 to 9.
  - A digit in base 2 ranges from 0 to 1 (binary number system). A digit in base 2 is also called a 'bit'.
  - A digit in base R can range from 0 to R-1
  - A digit in Base 16 can range from 0 to 16-1 (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F). Use letters A-F to represent values 10 to 15. Base 16 is also called Hexadecimal or just 'Hex'.

BR 8/99

Positional Notation

Value of number is determined by multiplying each digit by a weight and then summing. The weight of each digit is a POWER of the BASE and is determined by position.

953.78 = $9 \times 10^2 + 5 \times 10^1 + 3 \times 10^0 + 7 \times 10^{-1} + 8 \times 10^{-2}$
= 900 + 50 + 3 + .7 + .08 = 953.78

% 1011.11 = $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$
= 8 + 0 + 2 + 1 + 0.5 + 0.25
= 11.75

$ A2F = $10 \times 16^2 + 2 \times 16^1 + 15 \times 16^0$
= 10 x 256 + 2 x 16 + 15 x 1
= 2560 + 32 + 15 = 2607

BR 8/99

Base 10, Base 2, Base 16

The textbook uses subscripts to represent different bases (ie. $A2F_{16}$, 953.78_{10}, 1011.11_{2})

I will use special symbols to represent the different bases. The default base will be decimal, no special symbol for base 10.

The '$' will be used for base 16 ($A2F)

The '%' will be used for base 2 (%10101111)

If ALL numbers on a page are the same base (ie, all in base 16 or base 2 or whatever) then no symbols will be used and a statement will be present that will state the base (ie, all numbers on this page are in base 16).

BR 8/99

Common Powers

$2^{-3} = 0.125$
$2^{-2} = 0.25$
$2^{-1} = 0.5$
$2^{0} = 1$
$2^{1} = 2$
$2^{2} = 4$
$2^{3} = 8$
$2^{4} = 16$
$2^{5} = 32$
$2^{6} = 64$
$2^{7} = 128$
$2^{8} = 256$
$2^{9} = 512$
$2^{10} = 1024$
$2^{11} = 2048$
$2^{12} = 4096$

$16^{0} = 1 = 2^{0}$
$16^{1} = 16 = 2^{4}$
$16^{2} = 256 = 2^{8}$
$16^{3} = 4096 = 2^{12}$

2^{10} = 1024 = 1 K
2^{20} = 1048576 = 1 M (1 Megabits) = 1024 K = $2^{10} \times 2^{10}$
2^{30} = 1073741824 = 1 G (1 Gigabits)

BR 8/99

Conversion of Any Base to Decimal

Converting from ANY base to decimal is done by multiplying each digit by its weight and summing.

Binary to Decimal

% 1011.11 = $1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 + 1x2^{-1} + 1x2^{-2}$
= 8 + 0 + 2 + 1 + 0.5 + 0.25
= 11.75

Hex to Decimal

$ A2F = $10x16^2 + 2x16^1 + 15x16^0$
= 10 x 256 + 2 x 16 + 15 x 1
= 2560 + 32 + 15 = 2607

BR 8/99

Conversion of Decimal Integer To ANY Base

Divide Number N by base R until quotient is 0. Remainder at EACH step is a digit in base R, from Least Significant digit to Most significant digit.

Convert 53 to binary

53/2 = 26, rem = 1 ← Least Significant Digit
26/2 = 13, rem = 0
13/2 = 6 , rem = 1
6 /2 = 3, rem = 0
3/2 = 1, rem = 1
1/2 = 0, rem = 1 ← Most Significant Digit

53 = % 110101
= $1x2^5 + 1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0$
= 32 + 16 + 0 + 4 + 0 + 1 = 53

BR 8/99

Least Significant Digit
Most Significant Digit

53 = % 110101

Most Significant Digit (has weight of 2^5 or 32). For base 2, also called Most Significant Bit (MSB). Always LEFTMOST digit.

Least Significant Digit (has weight of 2^0 or 1). For base 2, also called Least Significant Bit (LSB). Always RIGHTMOST digit.

BR 8/99

More Conversions

Convert 53 to Hex

53/16 = 3, rem = 5
3 /16 = 0 , rem = 3
53 = $ 35
= $3 \times 16^1 + 5 \times 16^0$
= 48 + 5 = 53

BR 8/99

Hex (base 16) to Binary Conversion

Each Hex digit represents 4 bits. To convert a Hex number to Binary, simply convert each Hex digit to its four bit value.

Hex Digits to binary:
$ 0 = % 0000
$ 1 = % 0001
$2 = % 0010
$3 = % 0011
$4 = % 0100
$5 = % 0101
$6 = % 0110
$7 = % 0111
$8 = % 1000

Hex Digits to binary (cont):
$ 9 = % 1001
$ A = % 1010
$ B = % 1011
$ C = % 1100
$ D = % 1101
$ E = % 1110
$ F = % 1111

BR 8/99

Hex to Binary, Binary to Hex

$ A2F = % 1010 0010 1111

$ 345 = % 0011 0100 0101

Binary to Hex is just the opposite, create groups of 4 bits starting with least significant bits. If last group does not have 4 bits, then pad with zeros for unsigned numbers.

% 1010001 = % 0101 0001 = $ 51

Padded with a zero

BR 8/99

A Trick!

If faced with a large binary number that has to be converted to decimal, I first convert the binary number to HEX, then convert the HEX to decimal. Less work!

% 110111110011 = % 1101 1111 0011
= $ D F 3
= $13 \times 16^2 + 15 \times 16^1 + 3 \times 16^0$
= 13 x 256 + 15 x 16 + 3 x 1
= 3328 + 240 + 3
= 3571

Of course, you can also use the binary, hex conversion feature on your calculator. Too bad calculators won't be allowed on the first test, though…...

BR 8/99

Bah! I thought we were talking about Binary DATA!!!

Yah, we were!

Key Question we must answer is ...

How many binary DIGITS does it take to represent our data??

BR 8/99

Binary Codes

One Binary Digit (one bit) can take on values 0, 1. We can represent TWO values:

(0 = hot, 1 = cold), (1 = True, 0 = False), (1 = on, 0 = off).

Two Binary digits (two bits) can take on values of 00, 01, 10, 11. We can represent FOUR values:

(00 = hot, 01 = warm, 10 = cool, 11 = cold).

Three Binary digits (three bits) can take on values of 000, 001, 010, 011, 100, 101, 110, 111. We can represent 8 values
000 = Black, 001 = Red, 010 = Pink, 011 = Yellow, 100 = Brown, 101 = Blue, 110 = Green , 111 = White.

BR 8/99

Binary Codes (cont.)

N bits (or N binary Digits) can represent 2^N different values.

(for example, 4 bits can represent 2^4 or 16 different values)

N bits can take on unsigned decimal values from 0 to 2^N-1.

Codes usually given in tabular form.

000	black
001	red
010	pink
011	yellow
100	brown
101	blue
110	green
111	white

BR 8/99

Binary Data (again!)

The computer screen on your Win 98 PC can be configured for different resolutions. One resolution is 600 x 800 x 8, which means that you have 600 dots vertically x 800 dots horizontally, with each dot using 8 bits to take on 256 different colors. (actually, a dot is called a pixel).

Need 8 bits to represent 256 colors (2^8 = 256). Total number of bits needed to represent the screen is then:

600 x 800 x 8 = 3,840,000 bits (or just under 4 Mbits)

Your video card must have at least this much memory on it.

1 Mbits = 1024 x 1024 = 2^{10} x 2^{10} = 2^{20} .
1 Kbits = 1024 = 2^{10}.

BR 8/99

Codes for Characters

Also need to represent Characters as digital data.
The ASCII code (American Standard Code for Information Interchange) is a 7-bit code for Character data. Typically 8 bits are actually used with the 8th bit being zero or used for error detection (parity checking).
8 bits = 1 Byte.

'A' = % 01000001 = $41
'&' = % 00100110 = $26

7 bits can only represent 2^7 different values (128). This enough to represent the Latin alphabet (A-Z, a-z, 0-9, punctuation marks, some symbols like $), but what about other symbols or other languages?

BR 8/99

UNICODE

UNICODE is a 16-bit code for representing alphanumeric data.
With 16 bits, can represent 2^{16} or 65536 different symbols.
16 bits = 2 Bytes per character.

$0041-005A A-Z
$0061-4007A a-z

Some other alphabet/symbol ranges

$3400-3d2d Korean Hangul Symbols
$3040-318F Hiranga, Katakana, Bopomofo, Hangul
$4E00-9FFF Han (Chinese, Japenese, Korean)

UNICODE used by Web browsers, Java, most software these days.

BR 8/99

Codes for Decimal Digits

There are even codes for representing decimal digits. These codes use 4-bits for EACH decimal digits; it is NOT the same as converting from decimal to binary.

BCD Code
0 = % 0000
1 = % 0001
2 = % 0010
3 = % 0011
4 = % 0100
5 = % 0101
6 = % 0110
7 = % 0111
8 = % 1000
9 = % 1001

In BCD code, each decimal digit simply represented by its binary equivalent.
96 = % 1001 0110 = $ 96 (BCD code)

Advantage: easy to convert
Disadvantage: takes more bits to store a number:

255 = % 1111 1111 = $ FF (binary code)
255 = % 0010 0101 0101 = $ 255 (BCD code)

takes only 8 bits in binary, takes 12 bits in BCD.

BR 8/99

Gray Code for decimal Digits

Gray Code
0 = % 0000
1 = % 0001
2 = % 0011
3 = % 0010
4 = % 0110
5 = % 1110
6 = % 1010
7 = % 1011
8 = % 1001
9 = % 1000

A Gray code changes by only 1 bit for adjacent values. This is also called a 'thumbwheel' code because a thumbwheel for choosing a decimal digit can only change to an adjacent value (4 to 5 to 6, etc) with each click of the thumbwheel. This allows the binary output of the thumbwheel to only change one bit at a time; this can help reduce circuit complexity and also reduce signal noise.

BR 8/99

What do you need to Know?

- Convert hex, binary integers to Decimal
- Convert decimal integers to hex, binary
- Convert hex to binary, binary to Hex
- N binary digits can represent 2^N values, unsigned integers 0 to 2^N-1.
- ASCII, UNICODE are binary codes for character data
- BCD code is alternate code for representing decimal digits
- Gray codes can also represent decimal digits; adjacent values in Gray codes change only by one bit.

BR 8/99