University of British Columbia
Electrical and Computer Engineering
Digital Systems and Microcomputers CPEN312

L02: Number representation.

Dr. Jesús Calviño-Fraga. P.Eng.
Department of Electrical and Computer Engineering, UBC
Office: KAIS 3024
E-mail: jesusc@ece.ubc.ca
Phone: (604)-827-5387

January 04, 2019

Number Representation

- Number systems: Decimal, Binary, Octal, Hexadecimal.
- Conversion from/to binary to/from hex or octal.
- Conversion from/to binary to/from decimal

Number Representation

- All logic and arithmetic operations in a digital system are performed using numbers.
- All the results from a digital system are also numbers.
- The problem is that those numbers are usually just ones and zeroes!
- In order for a digital system to interact with the rest of the world, all physical quantities (temperature, voltage, pressure, sound, etc.) must be converted to numbers and all the output must be converted to physical quantities.

Positional Decimal Numbers

- Positional (or place-value notation) decimal numbers are a relatively recent invention. Before, people used other base systems, such as base 12 and base 60!
- The concept of zero is fundamental for place-value notation to work!
- It is believed that positional decimal numbers were invented between the 1^{st} and 4^{th} centuries by Indian mathematicians, picked-up the Arabs later on who spread it to the rest of the world. It also referred as the Hindu-Arabic numeral system.
- The system was brought to Europe by Fibonacci in the 13^{th} century.

Positional Decimal Numbers

European	0	1	2	3	4	5	6	7	8	9
Arabic-Indic	٠	١	٢	٣	٤	٥	٦	٧	٨	٩
Eastern Arabic-Indic (Persian and Urdu)	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹
Devanagari (Hindi)	०	१	२	३	४	५	६	७	८	९
Tamil		௧	௨	௩	௪	௫	௬	௭	௮	௯

http://en.wikipedia.org/wiki/Hindu-Arabic_numeral_system

Positional Decimal Numbers

- 2674=
- 1.234=
- -9.76=
- 3.14159265358979323846264338…=

Positional Decimal Numbers

- $2674 = (2 \times 10^3) + (6 \times 10^2) + (7 \times 10^1) + (4 \times 10^0)$
- $1.234= (1 \times 10^0) + (2 \times 10^{-1}) + (3 \times 10^{-2}) + (4 \times 10^{-3})$
- $-9.76=-1 \times ((9 \times 10^1) + (7 \times 10^{-1}) + (6 \times 10^{-2}))$
- 3.14159265358979323846264338…=π

We can only have a numerical approximation to π

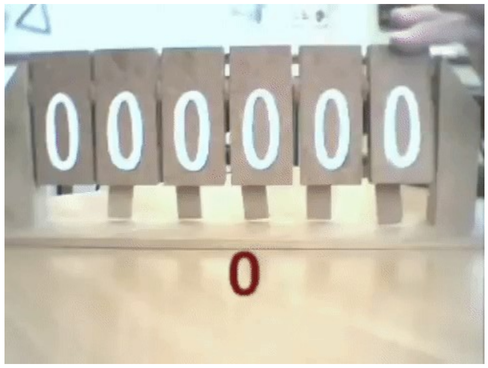
Counting in Decimal

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, and so on…

Now supposed we can only use two digits (0 and 1) to count:

0,1, 10, 11, 100, 101, 110, 111, 1000, 1001, and so on…

Counting in Binary



https://www.reddit.com/r/mechanical_gifs/comments/9cto4l/how_simple_pieces_of_wood_and_hinges_makes_a/

Positional Binary Numbers

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

Positional Binary Numbers

- $(1100\ )_2 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = 8 + 4 + 0 + 0 = 12$
- $(1.011)_2 = (1 \times 2^0) + (0 \times 2^{-1}) + (1 \times 2^{-2}) + (1 \times 2^{-3}) = 1 + 0 \times ½ + 1 \times ¼ + 1 \times 1/8 = 1.375$
- $-(10.01)_2 = -1 \times ((1 \times 2^1) + (0 \times 2^0) + (0 \times 2^{-1}) + (1 \times 2^{-2})) = -(2+0+0+1/4) = -2.25$
- $(100000000000)_2$=2048 (you do it!)

Other Bases (Radix)

- base 8 – called octal

  $(735)_8 = 7\text{x}8^2 + 3\text{x}8^1 + 5\text{x}8^0 = (448)_{10} + (24)_{10} + (5)_{10} = (477)_{10}$
- base 16 – called hexadecimal
  - what to do about running out of symbols?
  - use: A = 10, B=11, C=12, D=13, E=14, F=15

  $(BAD1)_{16} = (11)_{10}\text{x}16^3 + (10)_{10}\text{x}16^2 + (13)_{10}\text{x}16^1 + (1)_{10}\text{x}16^0 =$
  $(45056)_{10}+(2{,}560)_{10}+(208)_{10}+(1)_{10} = (47825)_{10}$

Binary, Octal, Decimal, Hex

Binary	Octal	Decimal	Hex
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1101	17	15	F

Bits, Nibble, Byte, Word

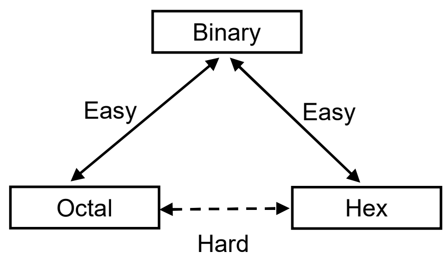
- Bit: Binary Digit, either zero or one.
- Nibble: Aggregation of 4-bits or one hex digit.
- Byte: Aggregation of 8-bits, 2-nibbles, or two hex digits.
- Word: Aggregation of 16-bits, 4-nibbles, 2-bytes, or four hex digits.

Word
Byte
Nibble
Bit

$(1001111000010010)_2$

Radix Conversion

It is very easy to convert a number between radixes if the input/output is binary and the output/input is either octal or hexadecimal:



Example: Convert These Numbers

- $(8C45)_{16}=(\qquad\qquad\qquad\qquad)_2$
- $(777)_8=(\qquad\qquad\qquad)_2$
- $(1001000110000110)_2=(\qquad)_{16}$
- $(111110101100)_2=(\qquad)_8$

Example: Convert These Numbers

- $(8C45)_{16}=(1000110001000101)_2$
- $(777)_8=(111111111)_2$
- $(1001000110000110)_2=(9186)_{16}$
- $(111110101100)_2=(7654)_8$

What about converting to/from decimal numbers? That is harder!

Converting Decimal to Binary

- Use the method of successive division with remainder. For example to convert 87 to binary:

Quotient	Divided by 2	Remainder
87	43	1
43	21	1
21	10	1
10	5	0
5	2	1
2	1	0
1	0	1

Answer: $(87)_{10}=(1010111)_2$

Convert Decimal to any Radix

- Same as above, but instead of dividing by 2, divide by the radix.
- Example: convert to Sexagesimal (radix 60 or if you prefer hours, minutes, seconds) $(15944)_{10}$ seconds. You can use a calculator for this one if you want, but calculators will NOT be allowed in the exams.

$(15944)_{10}=(\quad\quad)_{60}$

Quotient	Divide by 60	Remainder
15944	265	44
265	4	25
4	0	4

$(15944)_{10}=(4:25:44)_{60}$

Converting Binary to Decimal

- There are two easy ways to convert binary to decimal:
  - Multiply each bit by the corresponding power of two and add them. I call this one ‘add powers of two’. It requires you to remember the powers of two up to n.
  - Use the Double-Dabble algorithm. Doesn’t require memorizing, but it could be tedious. People can become really good at using this method!

Converting Binary to Decimal: Add Powers of Two

- This one requires a bit of memorizing for “fast” conversion: the powers of two. See table.
- Need help memorizing these numbers? Look for a game called “2048”.

n	2^n	n	2^n
0	1	8	256
1	2	9	512
2	4	10	1024
3	8	11	2048
4	16	12	4096
5	32	13	8192
6	64	14	16384
7	128	15	32768

Converting Binary to Decimal: Add Powers of Two

Convert $(10110110)_2$ to decimal:

1 x 128 + 0 x 64 + 1 x 32 + 1 x 16 + 0 x 8 + 1 x 4 + 1 x 2 + 0 x 1

$(10110110)_2$=182

Convert $(10101010)_2$ to decimal:

1 x 128 + 0 x 64 + 1 x 32 + 0 x 16 + 1 x 8 + 0 x 4 + 1 x 2 + 0 x 1

$(10101010)_2$=170

Converting Binary to Decimal: Double-dabble algorithm

1. BIN: $\underline{0}1011011$ Result=0
   Multiply Result by two and add the underlined bit:
   Result =((0 x 2)+0)=0
2. BIN: $0\underline{1}011011$ Result =0
   Multiply Result by two and add the underlined bit:
   BCD=((0 x 2)+1)=1
3. BIN: $01\underline{0}11011$ Result=1
   Multiply Result by two and add the underlined bit:
   BCD=((1 x 2)+0)=2
4. BIN: $010\underline{1}1011$ Result=2
   Multiply Result by two and add the underlined bit:
   BCD=((2x2)+1)=5

https://en.wikipedia.org/wiki/Double_dabble

Converting Binary to Decimal: Double-dabble algorithm

5. BIN: $0101\underline{1}011$ Result=5
   Multiply Result by two and add the underlined bit:
   Result=((5x2)+1)=11
6. BIN: $01011\underline{0}11$ Result=11
   Multiply Result by two and add the underlined bit:
   Result =((11x2)+0)=22
7. BIN: $010110\underline{1}1$ Result=22
   Multiply Result by two and add the underlined bit:
   Result =((22x2)+1)=45
8. BIN: $0101101\underline{1}$ Result=45
   Multiply Result by two and add the underlined bit:
   Result =((45x2)+1)=091 ← Final result is 91

Binary-Coded-Decimal System

- The binary-coded-decimal (BCD) system is used to represent each of the 10 decimal digits as a 4-bit binary code. Very useful to display the results of digital systems to humans!
- Example: Convert $(893)_{10}$ to BCD

$$(\overbrace{1000}^{8}\ \overbrace{1001}^{9}\ \overbrace{0011}^{3})_{BCD}$$

Exercises

- Convert your student number to binary, octal, and hex. (Frequently asked in midterm exam!)
- Convert your student number to BCD.
- Convert the following binary numbers to decimal using both the addition of powers of two and the double-dabble method:
  - $(11001)_2$,
  - $(01010101)_2$,
  - $(1100001110011110)_2$,