Lecture 3 CS1110/CS5019 20/1/20

Counting In Different Bases (Positional Number System)

Decimal		Hex	Octal		Binary (4-bit)			
0	0	0	0	0	0	0	0	0
0	1	1	0	1	0	0	0	1
0	2	2	0	2	0	0	1	0
0	3	3	0	3	0	0	1	1
0	4	4	0	4	0	1	0	0
0	5	5	0	5	0	1	0	1
0	6	6	0	6	0	1	1	0
0	7	7	0	7	0	1	1	1
0	8	8	1	0	1	0	0	0
0	9	9	1	1	1	0	0	1
1	0	A	1	2	1	0	1	0
1	1	B	1	3	1	0	1	1
1	2	C	1	4	1	1	0	0
1	3	D	1	5	1	1	0	1
1	4	E	1	6	1	1	1	0
1	5	F	1	7	1	1	1	1
10^1	10^0	16^0	8^1	8^0	2^3	2^2	2^1	2^0

Note: We can always compare numbers in counting order!

When counting in any base, B, the least significant digit changes every time (i.e., every B^0 times).

The next order of magnitude changes every B^1 times and the order of magnitude after that changes every B^2 times, etc.

We can use this rule as a trick to remember how to count in any base.

In binary, for example,

the LSB changes every time, the next column up changes every 2 times, the next every 4 times, etc.

Note

Having a limited number of bits, limits the number of unique combinations of those bits.

Thus, if we have only 4 bits we are limited to 16 (2^4) combinations of these bits

In general, n-bits give 2^n unique combinations.

A physical machine will always be limited in the number of bits that it can hold and "directly" manipulate. Thus, there will be a limit to the number of unique things that can be represented by a given number of bits.

Computer Engineering recognizes these limits and develops trade-offs and strategies to mitigate their effects.

However, these limitations cannot be eliminated altogether.

If our machine is limited to manipulating 32-bits directly, the maximum amount of addressible memory will be 2^{32} (= 4 Giga locations), assuming a 32-bit address.

An Alternative method for Converting from Decimal to Binary.

Break the decimal number Into a Sum of powers of 2, each multiplied by 0 or by 1.

the Binary number equivalent will be the Sequence of 1s and 0s of the Corresponding Powers

For Example: Convert $25_{10} \rightarrow$ Binary

Powers of 2	....	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	....	64	32	16	8	4	2	1
multiplier	....	0	0	1	1	0	0	1

$$16 + 8 + 0 + 0 + 1 = 25$$

Therefore $25_{10} = 11001_2$

Convert back, as before.

In fact this method works for any base, when the appropriate modifications are used:

"Break the decimal number Into a Sum of Powers of ~~2s~~ base each multiplied by ~~0 or by 1.~~ 0 to base −1

the ~~Binary~~ base number equivalent will be the Sequence of ~~1s and 0s~~ of the Corresponding Powers."

0.. base−1 numbers

For Example: Convert $106_{10} \rightarrow$ Base 3

Powers of 3	3^5	3^4	3^3	3^2	3^1	3^0
	243	81	27	9	3	1.
multiplier	0	1	0	2	2	1

$1 \times 81 + 0 \times 27 + 2 \times 9 + 2 \times 3 + 1 \times 1 = 106_{10}$

Answer: 10221_3 — This method is a little harder for bases > 2.

So Continuous division method recommended.

Going between bases that are not a Perfect Power of 2, requires some work, as seen Previously.

When a base is a perfect Power of 2:

$$2,\ 4,\ 8,\ 16,\ 32,\ \ldots$$

each digit in the base will have a Corresponding unique bit Pattern using the number of bits in the associated Power of 2, with no bit patterns left over.

Hex	Binary (4-bit)			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

Octal	Binary (3-bit)		
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Hence, Converting between binary and these bases is a direct substitution.

Example: Hex → Binary

Convert FE10A → Binary

Each Hex digit Corresponds to a unique 4-bit Value

$\therefore$

F	E	1	0	A	
↓	↓	↓	↓	↓	
1111	1110	0001	0000	1010	→ answer

Example Octal → Binary

Convert 1247_8 → Binary

Each octal digit Corresponds to a unique 3-bit Value

1	2	4	7	
↓	↓	↓	↓	
001	010	100	111	→ answer

Converting In the opposite direction uses the same equivalences

Example Binary to octal

Convert 1101101011 to octal

Break the Sequence Into groups of 3, starting from the right-hand side, and prepend 0s to the left-hand side, if needed, to make a group of 3:

001	101	101	011	
1	5	5	3	→ Answer

Example Binary to Hex

Convert 110 111 001 to hex

Start on the Right-hand side, Break into groups of 4, Prepend 0s as needed.

0001	1011	1001	
1	B	9	→ Answer.

To Convert from Hex ⇔ octal go through binary.

Example Convert FED to octal

Hex	F	E	D		
Groups of 4:	1111	1110	1101		
Group into 3s:	111 111	101	101		
octal	7 7	5	5	→	Answer.

Converting from Binary to octal/Hex represents a Second level of Abstraction In our Journey from the machine towards the higher levels of Information representation