



Convert 5.375 (base ten) to Floating Point Representation

Step 1 Convert Decimal to Binary 5.375

Convert Integer: 5

* successive division by 2 *

$$\begin{array}{r} \frac{2}{2} \\ \frac{-4}{1} \end{array} \rightarrow \begin{array}{r} \frac{1}{2} \\ \frac{-2}{0} \end{array} \rightarrow \begin{array}{r} \frac{1}{1} \\ \frac{-0}{1} \end{array}$$

↓ Reached zero

$$5_{(10)} = 101_{(2)}$$

End Conditions-

- 1) Reach Zero
- OR
- 2) Reach a Pattern

Together...

$$5.375 = 101.011$$

Note: To avoid confusion

#₍₁₀₎ → Number is of base ten
#₍₂₎ → Number is of base two

Convert Fractional: 0.375

* Successive multiplication of remaining products *

$$0.375 \times 2 = 0 + 0.75$$

$$0.75 \times 2 = 1 + 0.5$$

$$0.5 \times 2 = 1 + 0.0 \leftarrow \text{Reached zero}$$

$$0.375_{(10)} = 011_{(2)}$$

Step 2 Converting Binary to Scientific Notation

$$5.375_{(10)} = 101.011_{(2)}$$

Convert 101.011₍₂₎ to Scientific Notation

$$101.011_{(2)} \xrightarrow{\text{scientific Notation}} 1.01011_{(2)} \times 2^2$$

Recall
Convert: 631.65
Scientific Notation

$$631.65 \rightarrow 6.3165 \times 10^2$$

Step 3 Add Biased Exponent: 1.01011

Biased Offset = 7 * This will be provided in assignments and exams *

Biased Exponent = 1.01011 × 2⁷

$$2_{(10)} + 7_{(10)} = \boxed{9}_{(10)} \rightarrow 9_{(10)} = 1001_{(2)}$$

Convert
to Binary

Biased Offset Formula

$$2^{n-1} - 1$$

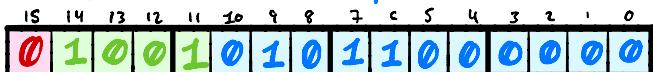
n = number of bits used to store Biased Exponent

Step 4 Fusion of Floating Point

Biased Component

$$1.01011 \times 2^9 = 1001_{(2)}$$

Signed Component Fractional Component



Signed Bit

Mantissa/Fraction

- * When signed component is positive set to 0(zero)
- * When signed component is negative set to 1

Step 5 Convert to Hex - Final Step

Number	0	1	2	3	4	5	6	7
Binary	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal	0	1	2	3	4	5	6	7

Number	8	9	10	11	12	13	14	15
Binary	1000	1001	1010	1011	1100	1101	1110	1111
Hexadecimal	8	9	A	B	C	D	E	F



Converted Hex Value: 0x4AC0

DeMorgan's Theorem

F1 $A \cdot B$

Inputs		Outputs
A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

F2 $\bar{A} + \bar{B}$

Inputs		Outputs
A	B	$\bar{A} + \bar{B}$
0	0	1
0	1	1
1	0	1
1	1	0



Karnaugh Maps (K-maps)

- Simplify Boolean Algebra
- Get a solution quickly by visual representation

Three Variable K-map

Ex 01 Minimize Boolean Expression Below

$$F = \bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + \bar{X}YZ + XY\bar{Z}$$

	$\bar{Y}\bar{Z}$	0	0	0	1	1	1	1	0
X	0	1			1	1	1		
	0	1			1	1	1		
1					1				

Ex 02 Minimize the logical expression below

$$f = \bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}YZ + X\bar{Y}Z$$

0 0 0 0 0 1 0 1 1 1 0 1

	$\bar{Y}\bar{Z}$	0	0	0	1	1	1	1	0
Z	0	1	1	1					
	0	1	1	1					
1									

* Group Impotence *

	$\bar{Y}\bar{Z}$	0	0	0	1	1	1	1	0
X	0	1	1	1	1	1	1	1	0
	0	1	1	1	1	1	1	1	0
1									

* Group Impotence *

	$\bar{Y}\bar{Z}$	0	0	0	1	1	1	1	0
X	0	1	1	1	1	1	1	1	0
	0	1	1	1	1	1	1	1	0
1									

$$F = Y\bar{Z} + \bar{X}\bar{Z} + \bar{X}Y$$

Combinational Circuit - Small Design Summary

1 | Truth table: Contains all the input logic for which an output is 0, as well as all the input conditions for which output bit or logic is a 1 or high.

2 | L.U.T : Entire truth table can be stored as LUT (Look Up Table).

- Advantage: Does not require further design steps.
Also, faster TTM (Time To Market).
- Disadvantage: Requires more H/W because you have to store all 0's and 1's.

3 | Minimal Logic Circuit Implementation

- Advantage: Requires less H/W in terms of gates with fewer inputs. Fewer wire connections.
- Disadvantage: Design time is much longer.

Logic Expressions

Implementation of SOP Expressions with NAND Gates

$$f = \bar{x} \cdot \bar{y} + \bar{x} \cdot \bar{y}$$