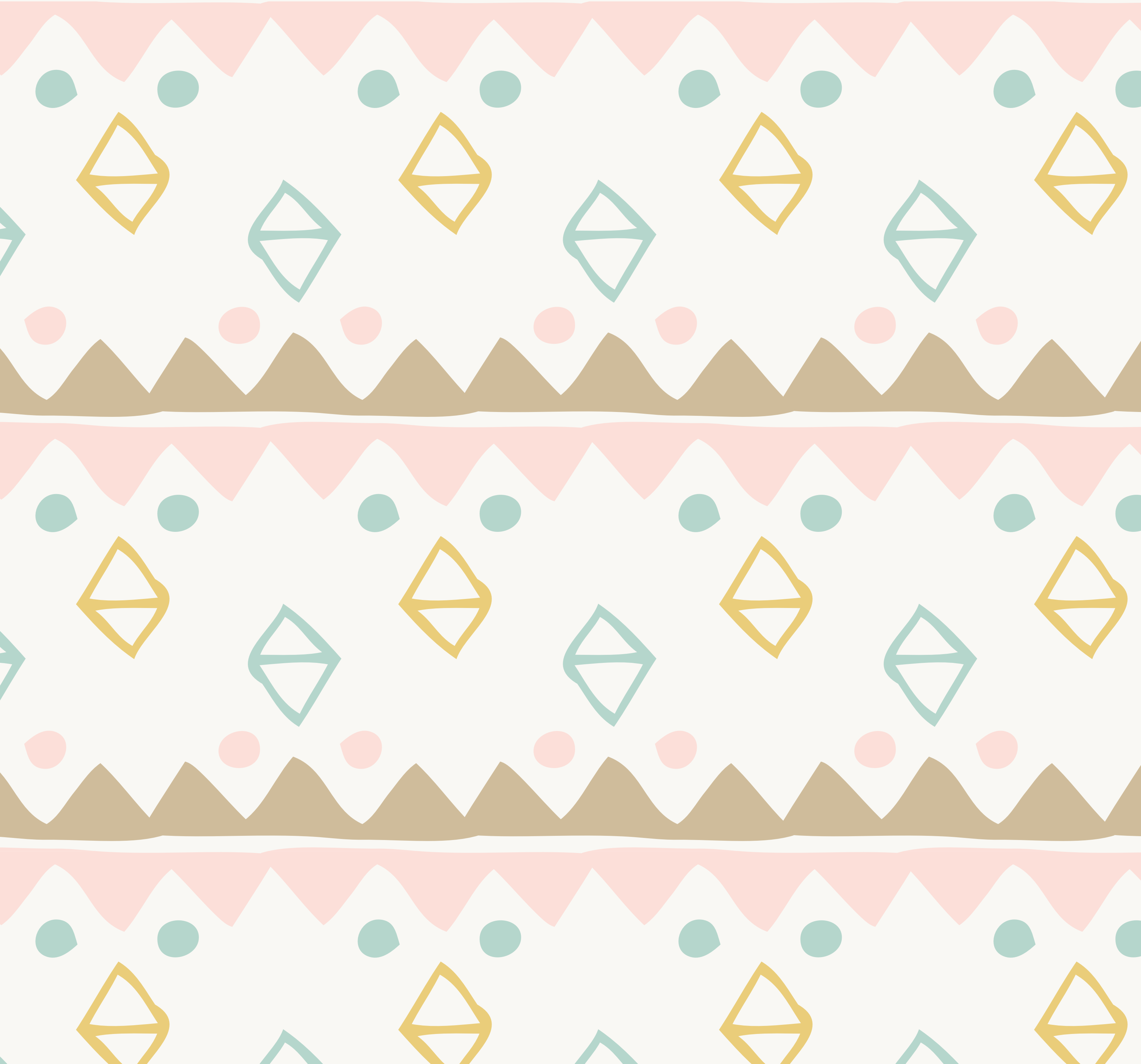

Convert 5.375 (base ten) to Floating Point Representation

Note: To avoid confusion
$\#_{(10)} \rightarrow$ Number is of base ten
$\#_{(2)} \rightarrow$ Number is of base two

Step 1 | Convert Decimal to Binary 5.375

Convert Integer: 5

* successive division by 2 *

$$2\overline{)5} \rightarrow 2\overline{)2} \rightarrow 2\overline{)1}$$

5 ÷ 2 = 2	2 ÷ 2 = 1	1 ÷ 2 = 0 ↙ Reached Zero
−4	−2	−0
1	0	1

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

End Conditions

1) Reach Zero
OR
2) Reach a Pattern

Together...

$$5.375 = 101.011$$

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

$$1\underbrace{0}_{2}\underbrace{1}_{1}.011_{(2)} \xrightarrow{\text{Scientific Notation}} 1.01011_{(2)} \times 2^{2}$$

Recall
Convert: 631.65

$$631.65 \xrightarrow{\text{Scientific Notation}} 6.3165 \times 10^{2}$$

Step 3 | Add Biased Exponent: 1.01011

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

Biased Exponent = 1.01011×2^{2}

$$2_{(10)} + 7_{(10)} = \boxed{9_{(10)}} \xrightarrow{\text{Convert to Binary}} 9_{(10)} = 1001_{(2)}$$

Biased Offset Formula

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

n = number of bits used to store Biased Exponent

Step 4 | Fusion of Floating Point

Biased Component

$$\underline{1}.\underline{01011} \times 2^{\boxed{9}} \quad \boxed{9} = 1001_{(2)}$$

Signed Component — Fractional Component

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	1	0	1	1	0	0	0	0	0	0

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

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

0x4 0xA 0xC 0x0

Converted Hex Value: 0x4AC0

DeMorgan's Theorem

F1 $\overline{A\cdot B}$

Inputs			Outputs
A	B	A+B	$\overline{A\cdot B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

$\Longleftrightarrow$

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

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

Karnaugh Maps (K-maps)

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

Three Variable K-map

Ex1 | Minimize Boolean Expression Below

$$F = \overline{X}\,\overline{Y}\,\overline{Z} + \overline{X}\,Y\,\overline{Z} + \overline{X}\,YZ + XYZ$$

X \ YZ	00	01	11	10
0	1		1	1
1			1	

* Group Impotence *

Imp #03

X \ YZ	00	01	11	10
0	1		1	1
1			1	

Imp#02 Imp#01

$$F = YZ + \overline{X}\,\overline{Z} + \overline{X}Y$$

Ex 02 | Minimize the logical expresson below

$$f = \overline{x}\,\overline{Y}\,\overline{z} + \overline{x}\,\overline{Y}\,z + \overline{x}\,YZ + x\overline{Y}Z$$

000 001 011 101

Z \ XY	00	01	11	10
0	1	1	1	
1		1		

* Group Impotence *

Imp#01 $(\overline{x}\cdot\overline{z})$ Imp#02 $(Y\overline{Z})$

X \ YZ	00	01	11	10
0	1	1	1	
1		1		

Imp#03 $(\overline{x}\,Y)$

Combinational Circut - 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 Circut 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

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