

1. (10 points) IEEE-754 Floating Point Representation

Represent 712.625 decimal in IEEE-754 format. Show all the steps for full credit.

Hint: Single precision: 32 bits:

Sign, excess 127 exponent, 1.x mantissa with hidden 1 *normalized*

$$(-631.125)_{10}$$



(FP IEEE -754 format)

2. (10 points) Field Programmable Logic Array (~~FPGA~~ *FPLA*)

Implement the following circuits using a NOT-AND-OR-XOR ~~FPGA~~ *FPLA*

$$F1(A,B,C,D) = (AB'C' + AB'D' + BCD)'$$

$$F2(A,B,C) = (ABC + A'B' + B'C)'$$

$$F3(B,C,D) = BCD + AB'C' + B'C' + A'B'$$

step 1 712 →

$$\begin{array}{r} 2 \overline{) 712} \\ \underline{2 \overline{) 356}} \quad - 0 \\ \underline{2 \overline{) 178}} \quad - 0 \\ \underline{2 \overline{) 89}} \quad - 0 \\ \underline{2 \overline{) 44}} \quad - 1 \\ \underline{2 \overline{) 22}} \quad - 0 \\ \underline{2 \overline{) 11}} \quad - 0 \\ \underline{2 \overline{) 5}} \quad - 1 \\ \underline{2 \overline{) 2}} \quad - 1 \\ \underline{1} \quad - 0 \end{array}$$

convert to binary

step 2:  $0.625 \times 2$

$$\begin{array}{r} 0.625 \times 2 \\ \hline 1.250 \times 2 \\ \hline 0.5 \times 2 \\ \hline 1.0 \end{array}$$

$0.101 \leftarrow (0.625)$

step 3:  $(712.625) \Rightarrow 1011001000.101 \times 2^0$

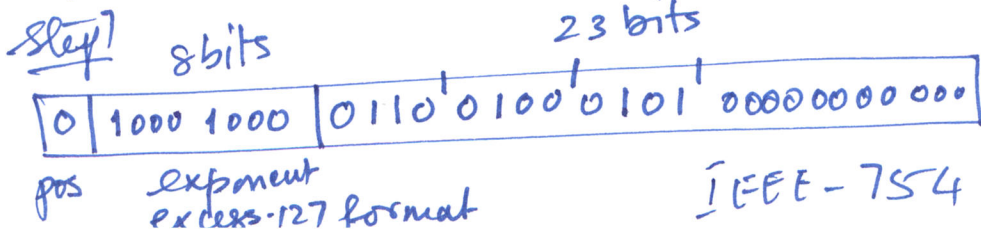
step 4: 1.x format and shift the exponent  
~~modify~~ the exponent

$$\begin{array}{r} \text{step 5: } 1011001000 \cdot 101 \times 2^0 \\ + 1.011001000101 \times 2^9 \\ \hline \end{array}$$

Sign      mantissa  
 hidden 1      exponent +127

step 6:  $9 + 127 = 136$

$$\begin{array}{r} 2 \overline{) 136} \\ \underline{2 \overline{) 68}} \quad - 0 \\ \underline{2 \overline{) 34}} \quad - 0 \\ \underline{2 \overline{) 17}} \quad - 0 \\ \underline{2 \overline{) 8}} \quad - 1 \\ \underline{2 \overline{) 4}} \quad - 0 \\ \underline{2 \overline{) 2}} \quad - 0 \\ \underline{1} \quad - 0 \end{array}$$



1. (10 points) IEEE-754 Floating Point Representation

Represent 712.625 decimal in IEEE-754 format. Show all the steps for full credit.

Hint: Single precision: 32 bits:

Sign, excess 127 exponent, 1.x mantissa with hidden 1

2. (10 points) Field Programmable Logic Array (FPGA) *FPLA* 7.6

Implement the following circuits using a NOT-AND-OR-XOR FPGA. *FPLA*

$$F1(A,B,C,D) = (\underline{A}B'C' + AB'D' + BCD)'$$

$$F2(A,B,C) = (ABC + \underline{A}'B' + B'C)'$$

$$F3(B,C,D) = \underline{BCD} + \underline{AB'C'} + \underline{B'C'} + \underline{A'B'}$$

