1. **(10 points) IEEE-754 Floating Point Representation**
   Represent 712.625 decimal in IEEE-754 formal. Show all the steps for full credit.

   Hint: Single precision: 32 bits:
   Sign, excess 127 exponent, 1.x mantissa with hidden 1 normalized

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'

$(-631.125)_{10}$ ⇓ (FP IEEE -754 format)

step 1 712 → convert to binary

2 | 712
2 | 356 − 0
2 | 178 − 0
2 | 89 − 0
2 | 44 − 1
2 | 22 − 0
2 | 11 − 0
2 | 5 − 1
2 | 2 − 1
1 − 0

step 2: 0.625 × 2
1.250 × 2
0.5 × 2
1.0

0.101 ← (0.625)

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

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

$1011001000.101 \times 2^0$

step 5: $+1.011001000101 \times 2^9$

↑ sign — mantissa ↓ hidden 1 — exponent +127

step 6: 9 + 127 = 136

2 | 136
2 | 68 − 0
2 | 34 − 0
2 | 17 − 0
2 | 8 − 1
2 | 4 − 0
2 | 2 − 0
1 − 0

step 7

0	1000 1000	0110 0100 0101 00000000000
pos	exponent excess-127 format	23 bits

8 bits — 23 bits — IEEE-754

1. **(10 points) IEEE-754 Floating Point Representation**
   Represent 712.625 decimal in IEEE-754 formal. 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) = (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'