CPEG 422/622 Spring 2020

Homework 5

Due April 10th at midnight (through Canvas)

1. Show the IEEE 754 single precision representation (32-bit) of -12.375_{10}. Show your steps to receive full credit.
   Answer:
   $(12.375)_{10}$ in binary is $(1100.011)_2$, thus $-12.375_{10} = (-1)^1 \times 1.100011 \times 2^3$
   Sign = 1 (negative)
   Fraction = 1000 1100 0000 0000 0000 000 – omit the hidden one, 23 bits in total
   Exponent = 3 + Bias (127) = 130 = $(1000\ 0010)_2$
   Signal precision: 11000001010001100 0…0 = 0xC1460000

   + 5 pts for the sign
   + 10 pts for fraction
   + 10 pts for the exponent
   + 5 pts for the final 32-bit binary presentation (or the hex form)

2. Follow the steps given in lecture, show the process of adding the following numbers in floating point representation: -12.375_{10} and 1.75.

Answer:

$-12.375_{10} = -1.100011_2 \times 2^3$

$1.75_{10} = 1.11_2 \times 2^0$

Step 0: Restore the hidden bits in the representation, already doen

Step 1: Shift the significand with the smaller exponent (1.11×2^0) right until its exponent matches the larger exponent, we have $1.11 \times 2^0 = 0.00111 \times 2^3$

Step 2: Add significands, we have -1.100011 + 0.00111 = - 1.010101

Step 3: Normalize the sum, it is already in normalized form

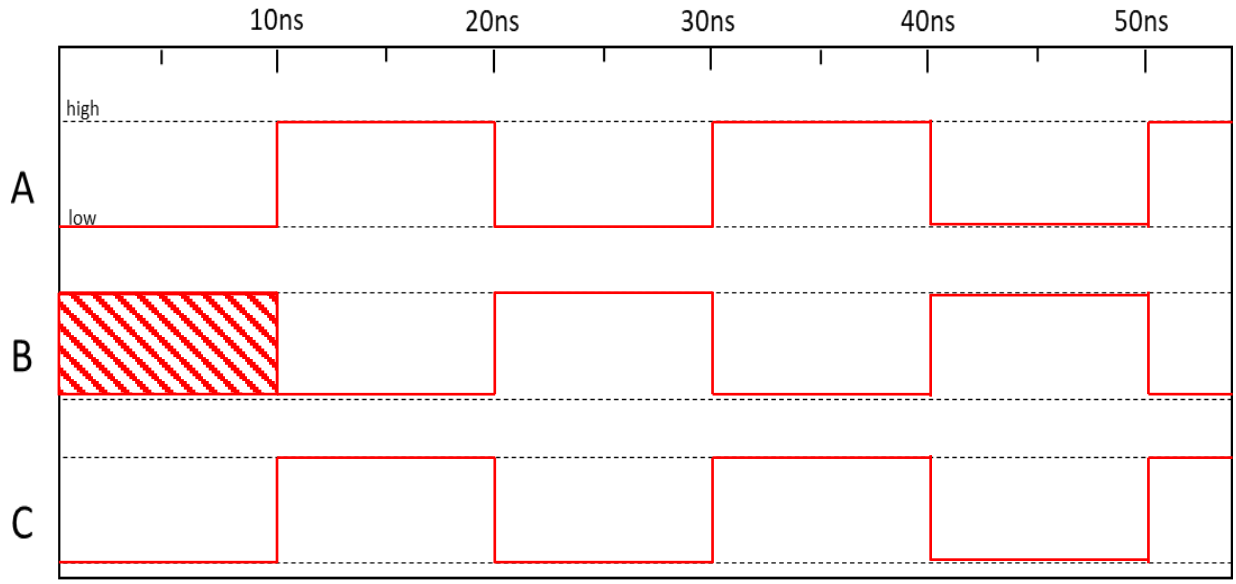
Step 4: round sum, it is already rounded

Step 5: rehide the leading bit before storing, we have S = 1, Exponent = 3 + Bias (127) = 130 = $(1000\ 0010)_2$, Fraction = 010101000…0, signal precision: 11000001010101000…0 = 0xC12A0000

+ 5 pts for converting -12.375 and 1.75 to binary
+ 10 pts for giving the sequence of steps
+ 5 pts for right shifting (step 1)
+ 5 pts for adding the significands (step 2)
+ 5 pts for rehidding the leading bit (step 5)

3. Assume a clock period of 10ns, Draw the waveforms (first 50ns) of signals A, B and C. Explain the difference between “B <= A” and “C <= A” in the code.

```
signal A: std_logic := 0;
signal B, C: std_logic;
gen: process(clock)
begin
    if(rising_edge(clock)) then
            A <= not A;
            B <= A;
    end if;
end process;
C <= A;
```



The two signal assignments “A <= not A” and “B <= A” take place right after exiting the process. So upon the first time existing the process, we will have A = 1, B = 0 (the old value of A), and C = 1 (the current value of A). The other cycles follow the same rule.

+10 pts for each waveform (30 in total)

+10 pts for the explanation