Solutions - Midterm Exam

(February 15th @ 7:30 pm)

Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (19 PTS)

- Compute the result of the following operations. The operands are signed fixed-point numbers. The result must be a signed fixed-point number. For the division, use $x = 5$ fractional bits.

1.0111 + 1.101010	1.010110 − 1000.0101	01.11111 + 0.10011
1.01101 × 10.101	0.111 × 1.0101	01.011 ÷ 1.011

```
c8=1 c7=1 c6=1 c5=1 c3=0 c2=0 c1=0 c0=0

   1.0 1 1 1 0 0 +
   1.1 0 1 0 1 0
   -------------
   1.0 0 0 1 1 0
```

```
1 1 1 1.0 1 0 1 1 0 -        c10=1 c9=1 c8=1 c7=1 c6=1 c5=1 c4=1 c3=1 c2=0 c1=0 c0=0
1 0 0 0.0 1 0 1 0 0     ➡     1 1 1 1.0 1 0 1 1 0 +
                              0 1 1 1.1 0 1 1 0 0
                              -------------------
                              0 1 1 1.0 0 0 0 1 0
```

```
c8=0 c7=0 c6=1 c5=1 c4=1 c3=1 c2=1 c1=1 c0=0

   0 0 1.1 1 1 1 1 +
   0 0 0.1 0 0 1 1
   ---------------
   0 1 0.1 0 0 1 0
```

```
 10.101 x  ➡  01.011 x  ➡    1 0 0 1 1 x
1.01101      0.10011            1 0 1 1
                             ---------
                             1 0 0 1 1
                           1 0 0 1 1
                         0 0 0 0 0
                       1 0 0 1 1
                     -----------------
                     0 1 1 0 1 0 0 0 1
                            ⬇
                     0.1 1 0 1 0 0 0 1
```

```
0.111 x  ➡  0.111 x   ➡       1 0 1 1 x
1.0101      0.1011               1 1 1
                              -------
                              1 0 1 1
                            1 0 1 1
                          1 0 1 1
                        ---------------
                        0 1 0 0 1 1 0 1
                               ⬇
                        0.1 0 0 1 1 0 1
                               ⬇
                        1.0 1 1 0 0 1 1
```

✓ $\frac{01.011}{1.011}$: To unsigned (denominator) and then alignment, $a = 3$: $\frac{01.011}{0.101} = \frac{01.011}{00.101} \equiv \frac{1011}{101}$

```
        001000110
101 ⟌  101100000
        101
        -----
          01000
            101
          -----
            110
            101
            ---
             10
```

Append $x = 5$ zeros: $\frac{1011\mathbf{00000}}{101}$

Integer Division:

$Q = 1000110, R = 10$

$\rightarrow Qf = 10.00110\ (x = 5)$

Final result (2C): $\frac{10.1010}{0.101} = 2C(010.0011) = 101.1101$

PROBLEM 2 (11 PTS)

- Represent these numbers in Fixed Point Arithmetic (signed numbers). Use the FX format [8 4]. Truncate (the LSB) and perform Saturation when required.

✓ -33.375

+33.375 = 0100001.011 ⇒ -33.375 = 1011110.101

-33.375 = 1011110.101 = 1000.0000 (saturation)

✓ 17.875

+17.875 = 010001.111

+17.875 = 0111.1111 (sat+truncation)

- Complete the table for the following fixed-point formats (signed numbers): (3 pts.)

Integer bits	Fractional Bits	FX Format	Range	Resolution
6	4	[10 4]	$[-2^5, 2^5 - 2^{-4}]$	2^{-4}

- Given the 32-bit floating-point number: **`ECE4710A`**. Complete the bits in the fields and the significand's FX format: (3 pts.)

sign	e+bias
1	110 1100 1

FX format of significand: [24 23]

significand

1. 110 0100 0111 0001 0000 1010

PROBLEM 3 (40 PTS)

- Perform the following 32-bit floating point operations. For fixed-point division, use 4 fractional bits. Truncate the result when required. Show your work: <u>how you got the significand and the biased exponents bits of the result.</u> Provide the 32-bit result.

✓ `4F480000 + D0A90000`	✓ `C1500000 - C36A0000`	✓ `FABC0000 × 80400000`	✓ `7BB80000 ÷ C9400000`

✓ $X =$ `4F480000 + D0A90000`:

`4F480000`: 0100 1111 0100 1000 0000 0000 0000 0000

$e + bias = 10011110 = 158 \rightarrow e = 158 - 127 = 31$ $\quad Significand = 1.1001$

`4F480000` $= 1.1001 \times 2^{31}$

`D0A90000`: 1101 0000 1010 1001 0000 0000 0000 0000

$e + bias = 10100001 = 161 \rightarrow e = 161 - 127 = 34$ $\quad Significand = 1.0101001$

`D0A90000` $= -1.0101001 \times 2^{34}$

$X = -1.0101001 \times 2^{34} + 1.1001 \times 2^{31} = -1.0101001 \times 2^{34} + \frac{1.1001}{2^3} \times 2^{34}$

$X = -(1.0101001 - 0.0011001) \times 2^{34}$ (unsigned subtraction)

$X = -1.001 \times 2^{34}$, $e + bias = 34 + 127 = 161 = 10100001$

X = 1101 0000 1001 0000 0000 0000 0000 0000 = `D0900000`

```
b8=0 b7=0 b6=0 b5=1 b4=0 b3=0 b2=0 b1=0 b0=0
  1.0 1 0 1 0 0 1 -
  0.0 0 1 1 0 0 1
  ---------------
  1.0 0 1 0 0 0 0
```

✓ $X =$ `C1500000 - C36A0000`:

`C1500000`: 1100 0001 0101 0000 1000 0000 0000 0000

$e + bias = 10000010 = 130 \rightarrow e = 130 - 127 = 3$ $\quad Significand = 1.101$

`C1500000` $= -1.101 \times 2^3$

`C36A0000`: 1100 0011 0110 1010 0000 0000 0000 0000

$e + bias = 10000110 = 134 \rightarrow e = 134 - 127 = 7$ $\quad Significand = 1.110101$

`C36A0000` $= -1.110101 \times 2^7$

$X = -1.101 \times 2^3 + 1.110101 \times 2^7 = -\frac{1.101}{2^4} \times 2^7 + 1.110101 \times 2^7$

$X = (-0.0001101 + 1.110101) \times 2^7$

```
c9=1 c8=1 c7=1 c6=1 c5=0 c4=0 c3=0 c2=1 c1=0 c0=0
  0 1.1 1 0 1 0 1 0 +
  1 1.1 1 1 0 0 1 1
  -----------------
  0 1.1 0 1 1 1 0 1
```

To subtract these unsigned numbers, we first convert to 2C:

$R = 01.110101 - 0.0001101 = 01.110101 + 1.1110011$

The result in 2C is: $R = 01.1011101$

For floating point, we need to convert to sign-and-magnitude:

$\Rightarrow R(SM) = +1.1011101$

* You can also do unsigned subtraction: $X = (1.110101 - 0.0001101) \times 2^7$

```
b8=0 b7=0 b6=0 b5=1 b4=1 b3=1 b2=0 b1=1 b0=0
  1.1 1 0 1 0 1 0 -
  0.0 0 0 1 1 0 1
  ---------------
  1.1 0 1 1 1 0 1
```

$X = 1.1011101 \times 2^7$, $e + bias = 7 + 127 = 134 = 10000110$

X = 0100 0011 0101 1101 0000 0000 0000 0000 = `435D0000`

✓ $X =$ `FABC0000 × 80400000`:

`FABC0000`: 1111 1010 1011 1100 0000 0000 0000 0000

$e + bias = 11110101 = 245 \rightarrow e = 245 - 127 = 118$ $\quad Significand = 1.01111$

`FABC0000` $= -1.01111 \times 2^{118}$

`80400000`: 1000 0000 0100 0000 0000 0000 0000 0000

$e + bias = 00000000 = 0 \rightarrow Denormal\ number \rightarrow e = -126$ $\quad Significand = 0.1$

`80400000` $= -0.1 \times 2^{-126}$

$X = (-0.1 \times 2^{-126}) \times (-1.01111 \times 2^{118}) = 0.101111 \times 2^{-8} = 1.01111 \times 2^{-9}$

$e + bias = -9 + 127 = 118 = 01110110$

X = 0011 1011 0011 1100 0000 0000 0000 0000 = `3B3C0000`

✓ $X =$ `7BB80000 ÷ C9400000`:

`7BB80000`: 0111 1011 1011 1000 0000 0000 0000 0000

$e + bias = 11110111 = 247 \rightarrow e = 247 - 127 = 120$ $Significand = 1.0111$

7BB80000 = 1.0111×2^{120}

C9400000: 1100 1001 0100 0000 0000 0000 0000 0000

$e + bias = 10010010 = 146 \rightarrow e = 146 - 127 = 19$ $Significand = 1.1$

C9400000 = -1.1×2^{19}

$$X = -\frac{1.0111 \times 2^{120}}{1.1 \times 2^{19}} = -\frac{1.0111}{1.1} \times 2^{101}$$

```
             000001111
11000  )    101110000
             11000
              101100
               11000
               101000
                11000
                100000
                 11000
                  1000
```

Alignment:
$$\frac{1.0111}{1.1} = \frac{1.0111}{1.1000} = \frac{10111}{11000}$$

Append $x = 4$ zeros: $\frac{10111\mathbf{0000}}{11000}$

Integer division
$Q = 1111 \rightarrow Qf = 0.1111$

Thus: $X = -0.1111 \times 2^{101} = -1.111 \times 2^{100}$
$e + bias = 100 + 127 = 227 = 11100011$

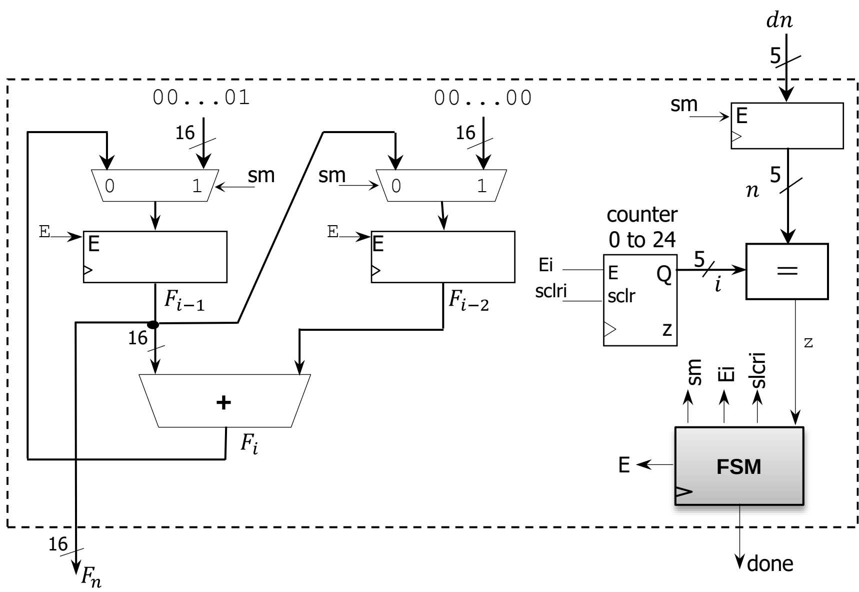
X = 1111 0001 1111 0000 0000 0000 0000 0000 = F1F00000

PROBLEM 4 (30 PTS)

- **Fibonacci numbers Computation**: This circuit reads an unsigned number ($n > 1$) and generates F_n:
$$F_n = F_{n-1} + F_{n-2},\ F_0 = 0,\ F_1 = 1$$

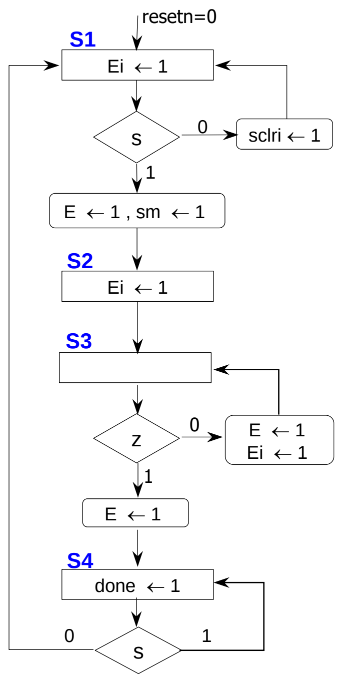
- The digital system is depicted below (FSM + Datapath).
  - ✓ Input Data dn ($1 < dn < 25$). Output Data: F_n (16-bit result; $n < 25$ as $F_{24} = 46368$)
  - ✓ $z = 1$ when $i = n$.
  - ✓ Sequential Algorithm:

```
n: unsigned integer
F0=0, F1=1
if n > 1
    for i = 2 to n
         Fi = Fi-1 + Fi-2
    end
end
return Fn
```



- Sketch the Finite State Machine diagram (in ASM form) given the sequential algorithm. (15 pts.)
  - ✓ The process begins when the s signal is asserted, at this moment we capture dn on register n. At the same time, the registers F_{i-1} and F_{i-2} are initialized with 1 and 0 respectively. Then, the process continues by updating F_{i-1} and F_{i-2}, and it is concluded when $i = n$. The signal done is asserted when the result is ready and appears on output F_n.
    - ▫ Note that the iteration index i index must start at $i = 2$.

Finite State Machine:



- Complete the timing diagram. n is provided as an unsigned decimal. You can provide F_{i-1}, F_{i-2}, and i as unsigned decimals. (15 pts.)