Solutions - Midterm Exam

(February 18th @ 7:30 pm)

Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (17 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 $x = 4$ fractional bits.

10.011 + 0.11111	1.010101 − 01.0101
1.10101 × 10.111	10.0110 ÷ 0.0111

```
c7=0 c6=0 c5=1 c4=1 c3=1 c2=0 c1=0 c0=0
  1 0.0 1 1 0 0 +
  0 0.1 1 1 1 1
  -------------
  1 1.0 1 0 1 1
```

```
                                 c8=1 c7=1 c6=1 c5=1 c4=1 c3=1 c2=0 c1=0 c0=0
1 1.0 1 0 1 0 1 -      ➡        1 1.0 1 0 1 0 1 +
0 1.0 1 0 1 0 0                 1 0.1 0 1 1 0 0
                                ---------------
                                1 0.0 0 0 0 0 1
```

```
1.10101 x   ➡   0.01011 x   ➡        1 0 1 1 x
10.111          01.001               1 0 0 1
                                     -------
                                     1 0 1 1
                                   0 0 0 0
                                 0 0 0 0
                               1 0 1 1
                             ---------------
                             0 1 1 0 0 0 1 1
                                   ⬇
                             0.0 1 1 0 0 0 1 1
```

✓ $\frac{10.0110}{0.0111}$: To unsigned and then alignment, $a = 4$: $\frac{01.1010}{0.0111} = \frac{01.1010}{0.0111} \equiv \frac{11010}{111}$

```
      000111011
111 ) 110100000
      111
      ----
       1100
        111
       -----
        1010
         111
        -----
         1100
          111
         -----
          1010
           111
          -----
            11
```

Append $x = 4$ zeros: $\frac{11010\mathbf{0000}}{111}$

Integer Division:

$Q = 111011, R = 11$
$\rightarrow Qf = 11.1011\ (x = 4)$

Final result (2C): $\frac{10.0110}{0.0111} = 2C(011.1011) = 100.0101$

PROBLEM 2 (8 PTS)

- Represent these numbers in Fixed Point Arithmetic (signed numbers). Use the FX format [12 4].
  - ✓ -17.125
  - ✓ 13.75

✓ -17.125:
17.125 = 010001.001 → -17.125 = 101110.111 = 0xEE.E

✓ 13.75:
13.75 = 01101.11 = 0x0D.C

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: how you got the significand and the biased exponent bits of the result. Provide the 32-bit result.

✓ C3FA8000 - C1E00000	✓ D0D80000 + D0FA0000	✓ 80C00000×FAD00000	✓ 7B380000 ÷ C8A00000

✓ X = C3FA8000 - C1E00000:

C3FA8000: 1100 0011 1111 1010 1000 0000 0000 0000

$e + bias = 10000111 = 135 \rightarrow e = 135 - 127 = 8$ $\quad Significand = 1.11110101$

C3FA8000 = -1.11110101×2^8

C1E00000: 1100 0001 1110 0000 0000 0000 0000 0000

$e + bias = 10000011 = 131 \rightarrow e = 131 - 127 = 4$ $\quad Significand = 1.11$

C1E00000 = -1.11×2^4

$$X = -1.11110101 \times 2^8 + 1.11 \times 2^4 = -1.11110101 \times 2^8 + \frac{1.11}{2^4} \times 2^8$$

$X = -(1.11110101 - 0.000111) \times 2^8$

We perform unsigned subtraction: $X = -1.11011001 \times 2^8$

```
b9=0 b8=0 b7=0 b6=0 b5=1 b4=1 b3=0 b2=0 b1=0 b0=0
 1.1 1 1 1 0 1 0 1 -
 0.0 0 0 1 1 1 0 0
 -----------------
 1.1 1 0 1 1 0 0 1
```

$X = -1.11011001 \times 2^8$, $e + bias = 8 + 127 = 135 = 10000111$

X = 1100 0011 1110 1100 1000 0000 0000 0000 = C3EC8000

✓ X = D0D80000 + D0FA0000:

D0D80000: 1101 0000 1101 1000 0000 0000 0000 0000

$e + bias = 10100001 = 161 \rightarrow e = 161 - 127 = 34$ $\quad Significand = 1.1011$

D0D80000 = -1.1011×2^{34}

D0FA0000: 1101 0000 1111 1010 0000 0000 0000 0000

$e + bias = 10100001 = 161 \rightarrow e = 161 - 127 = 34$ $\quad Significand = 1.1111010$

D0FA0000 = -1.1111010×2^{34}

$X = -1.1011 \times 2^{34} - 1.1111010 \times 2^{34}$ (unsigned addition)

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

$X = -11.101001 \times 2^{34} = -1.1101001 \times 2^{35}$

$e + bias = 35 + 127 = 162 = 10100010$

X = 1101 0001 0110 1001 0000 0000 0000 0000 = D1690000

✓ X = 80C00000 × FAD00000:

80C00000: 1000 0000 1100 0000 0000 0000 0000 0000

$e + bias = 00000001 = 1 \rightarrow e = 1 - 127 = -126$ $\quad Significand = 1.1$

80C00000 = $= -1.1 \times 2^{-126}$

FAD00000: 1111 1010 1101 0000 0000 0000 0000 0000

$e + bias = 11110101 = 245 \rightarrow e = 245 - 127 = 118$ $\quad Significand = 1.101$

FAD00000 = -1.101×2^{118}

```
    1.1 0 1 x
        1.1
    ---------
    1 1 0 1
  1 1 0 1
  -----------
 1 0.0 1 1 1
```

$X = (-1.1 \times 2^{-126}) \times (-1.101 \times 2^{118}) = 10.0111 \times 2^{-8} = 1.00111 \times 2^{-7}$

$e + bias = -7 + 127 = 120 = 01111000$

X = 0011 1100 0001 1100 0000 0000 0000 0000 = 3C1C0000

✓ X = 7B380000 ÷ C8A00000:

7B380000: 0111 1011 0011 1000 0000 0000 0000 0000

$e + bias = 11110110 = 246 \rightarrow e = 246 - 127 = 119$ $\quad Significand = 1.0111$

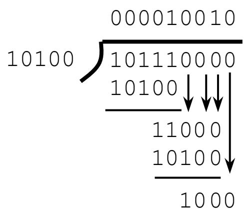
7B380000 = 1.0111×2^{119}

C8A00000: 1100 1000 1010 0000 0000 0000 0000 0000

$e + bias = 10010001 = 145 \rightarrow e = 145 - 127 = 18$ $\quad Significand = 1.01$

C8A00000 = -1.01×2^{18}

$$X = -\frac{1.0111 \times 2^{119}}{1.01 \times 2^{18}} = -\frac{1.0111}{1.01} \times 2^{101}$$



Alignment:
$\frac{1.0111}{1.01} = \frac{1.0111}{1.0100} = \frac{10111}{10100}$

Append $x = 4$ zeros: $\frac{10111\mathbf{0000}}{10100}$

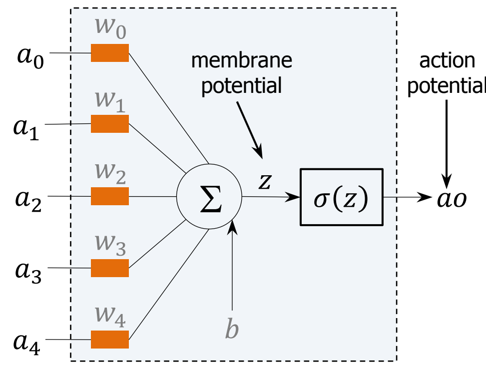
Integer division:
$Q = 10010 \rightarrow Qf = 1.001$

Thus: $X = -1.001 \times 2^{101}$, $e + bias = 101 + 127 = 228 = 11100100$
X = 1111 0010 0001 0000 0000 0000 0000 0000 = F2100000

PROBLEM 4 (35 PTS)

- **Artificial neuron model**. The membrane potential $\boldsymbol{z}$ is a sum of products (input activations a_i by weights w_i) to which a bias term b is added. The action potential $\boldsymbol{ao}$ is modeled as a scalar function of $\boldsymbol{z}$. The figure depicts a neuron with 5 inputs. The bias and the weights are constant values.

$$ao = \sigma\left(\sum_i a_i \times w_i + b\right)$$

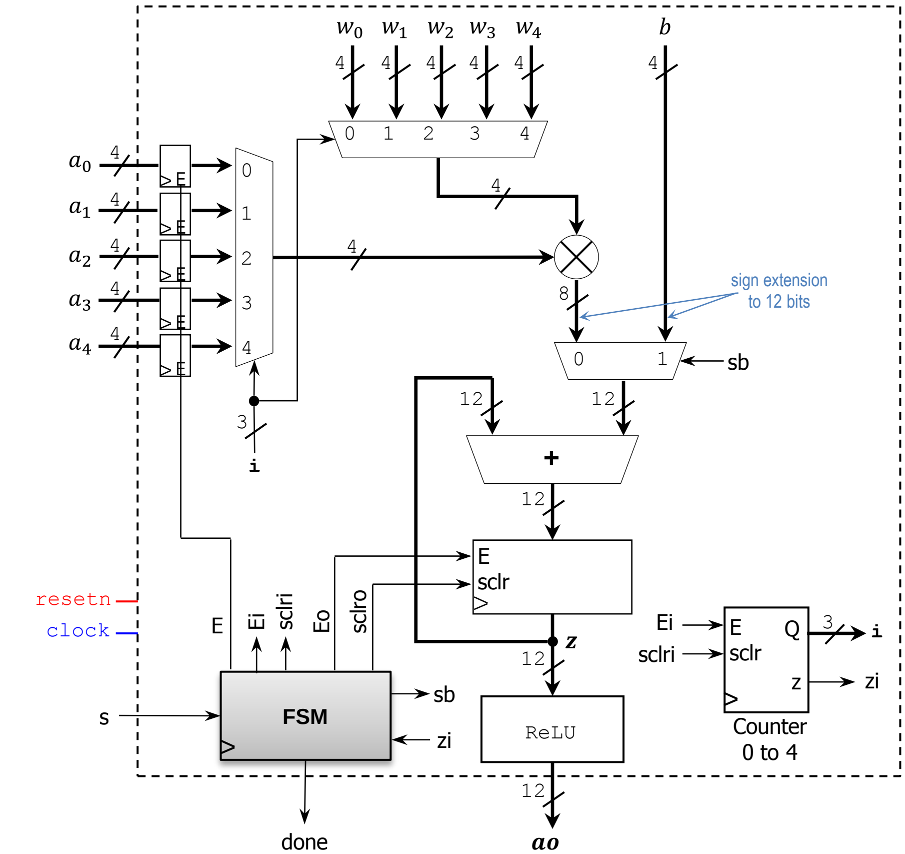


  - ✓ A popular and simple scalar function is the Rectified Linear Unit (ReLU):
    - ▫ $\sigma(z) = z$ if $z \geq 0$, otherwise $\sigma(z) = 0$.

- Digital System (FSM + Datapath): An iterative architecture for a 5-input neuron is depicted in next page. The circuit captures the input data $(a_0, a_1, a_2, a_3, a_4)$ and then computes $\boldsymbol{z}$ using a multiply-and-accumulate approach (see iterative algorithm). The output $\boldsymbol{ao}$ is computed by applying the ReLU function to $\boldsymbol{z}$.
  - ✓ All data is represented as <u>signed integers</u>:
    - ▫ Input activations $(a_0, a_1, a_2, a_3, a_4)$, weights $(w_0, w_1, w_2, w_3, w_4)$, bias (b): 4-bits wide.
    - ▫ Membrane potential $(\boldsymbol{z})$ and action potential $(\boldsymbol{ao})$: 12-bits wide (11 bits suffice, we select 12 for simplicity's sake).
  - ✓ Weights and biases: These are constant values (signed numbers represented as hexadecimals).
    - ▫ $w_0 = 0\times4$, $w_1 = 0\times1$, $w_2 = 0\times2$, $w_3 = 0\times8$, $w_4 = 0\times2$. $b = 0\times6$
  - ✓ Example:
    - ▫ If $a_0 = 0\times4$, $a_1 = 0\times E$, $a_2 = 0\times C$, $a_3 = 0\times5$, $a_4 = 0\times A$. These values appear in the timing diagram (next page).
    - ▫ Then $\boldsymbol{z} = 4\times4 + -2\times1 + -4\times2 + 5\times(-8) + -6\times2 = -40 = 0\times FD8$. Finally, $\boldsymbol{ao} = 0\times000$
  - ✓ Counter 0 to 4: If E=1, sclr=1, then Q ← 0. If E=1, sclr=0, then Q ← Q+1. Also: z=1 if Q = 4, else z=0.
  - ✓ Register: If E=1, sclr=1 → Clear. If E=1, sclr=0 → Load data.
  - ✓ 4×4 Signed Multiplier: This is a combinational circuit, whose result is 8-bits wide (it should be sign-extended to 12 bits).
  - ✓ ReLU: Combinational Block that implements the ReLU operation. For example, if $\boldsymbol{z} = 0\times FD8$, then $\boldsymbol{ao} = 0\times000$

- Sketch the Finite State Machine diagram (in ASM form) given the algorithm. (20 pts.)
  - ✓ The process begins when s is asserted, at this moment we capture a_0, a_1, a_2, a_3, a_4 on the input registers. Then $\boldsymbol{z}$ is updated until the counter reaches its maximum value (4). The signal done is asserted when the final result is computed.

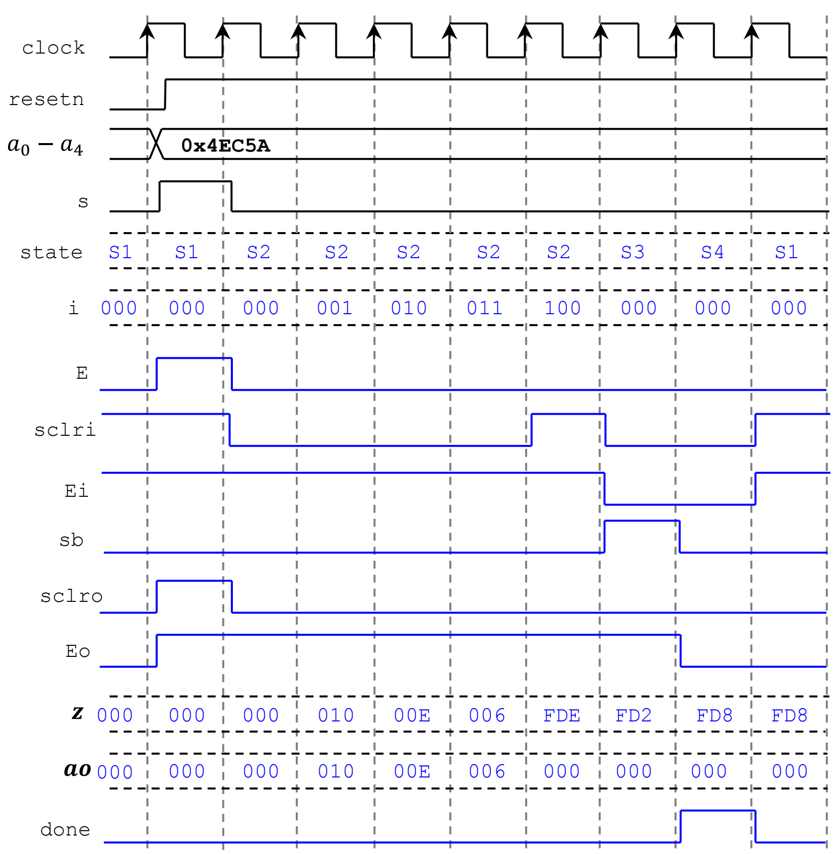
- Complete the timing diagram ($\boldsymbol{z}$ and $\boldsymbol{ao}$ are in hexadecimal format). (15 pts.)



$$ao = \sigma_{ReLU}\left(\sum_{i=0}^{4} a_i \times w_i + b\right)$$

ALGORITHM

$z \leftarrow 0$
$for\ i = 0\ to\ 4$
$\quad z \leftarrow z + a_i \times w_i$
end
$z \leftarrow z + b$
$ao \leftarrow z\ \ if\ z \geq 0, else\ 0$



FSM: