Student Honor Pledge:
All work submitted is completed by me directly without the use of any unauthorized resources or assistance
Initials: _____

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

PROBLEM 2 (8 PTS)

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

PROBLEM 3 (40 PTS)

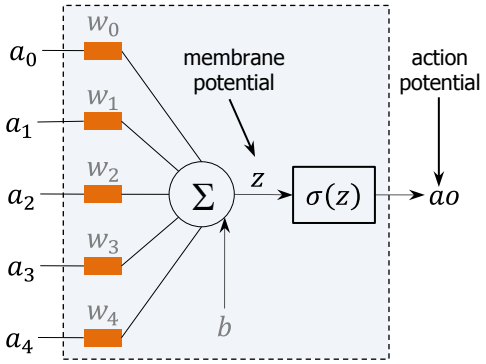
- Perform the following 32-bit floating point operations. For fixed-point division, use 4 fractional bits. Truncate the result when required. Show your work: how you got the significand and the biased exponent bits of the result. Provide the 32-bit result.

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

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.

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

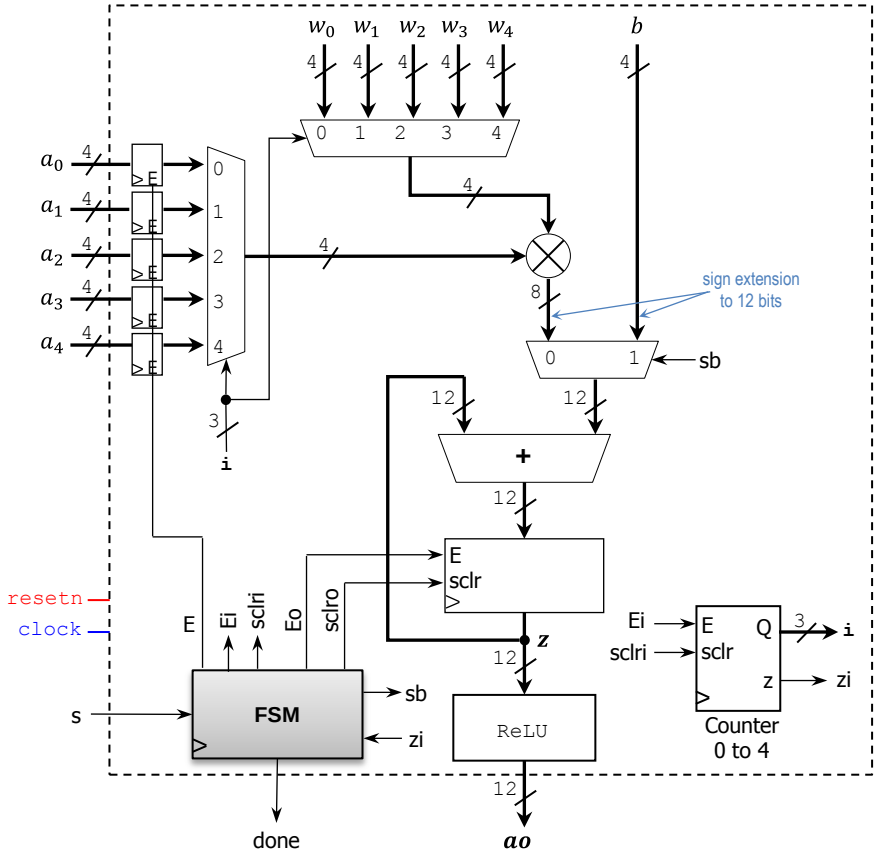


  - ✓ A popular and simple scalar function is the Rectified Linear Unit (ReLU):
    - $\sigma(z) = z$ if $z \geq 0$, otherwise $\sigma(z) = 0$.

- 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 signed integers:
    - 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).
    - $w_0 = 0{\times}4$, $w_1 = 0{\times}1$, $w_2 = 0{\times}2$, $w_3 = 0{\times}8$, $w_4 = 0{\times}2$. $b = 0{\times}6$
  - ✓ Example:
    - If $a_0 = 0{\times}4$, $a_1 = 0{\times}E$, $a_2 = 0{\times}C$, $a_3 = 0{\times}5$, $a_4 = 0{\times}A$. These values appear in the timing diagram (next page).
    - Then $\boldsymbol{z} = 4{\times}4 + -2{\times}1 + -4{\times}2 + 5{\times}(-8) + -6{\times}2 = -40 = 0{\times}FD8$. Finally, $\boldsymbol{ao} = 0{\times}000$
  - ✓ 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 $\boldsymbol{ao} = 0{\times}000$

- 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
$z \leftarrow z + b$
$ao \leftarrow z\ \ if\ z \geq 0, else\ 0$