Midterm Exam

(February 17th @ 7:30 pm)

Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (20 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.

1.010001 + 1.011	1001.1101 - 1.011101	0.010101 + 01.11111
10.101 × 0.10011	1.011 × 1.0101	10.10010 ÷ 0.101

PROBLEM 2 (10 PTS)

- Represent these numbers in Fixed Point Arithmetic (signed numbers). Use the FX format [12 4].
  - ✓ -16.375
  - ✓ 32.3125

- Complete the table for the following fixed point formats (signed numbers): (6 pts.)

Integer bits	Fractional Bits	FX Format	Range	Resolution
6	3			
8	5			

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.

✓ C1500000 + 436A0000	✓ D0A90000 – CF480000	✓ 80400000 × 7AB80000	✓ FBB80000 ÷ 49400000

PROBLEM 4 (30 PTS)

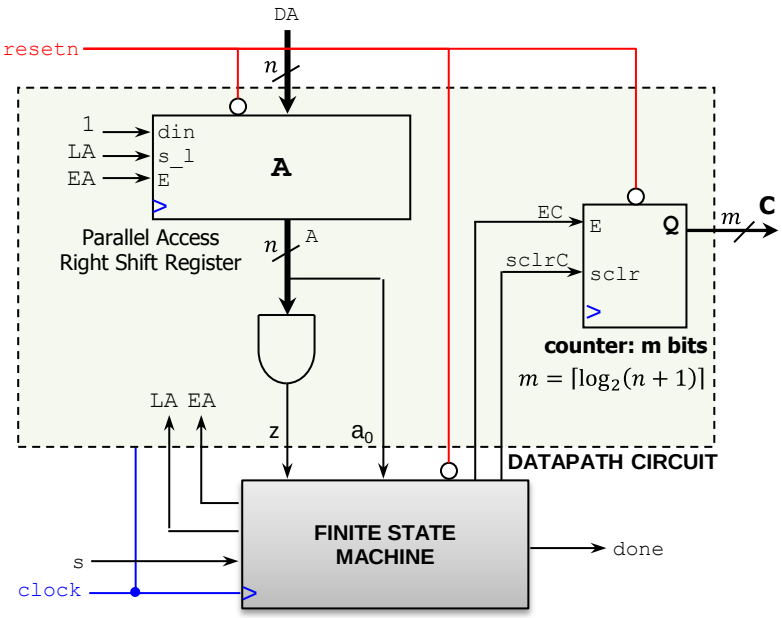
- "Counting 0's" Circuit: It counts the number of bits in register A with a '0' value. The digital system is depicted below.
  - ✓ Example: for $n = 8$: if $A = 00110010$, then $C = 0101$.
  - ✓ The behavior (on the clock tick) of the generic components is as follows:

m-bit counter (modulo-n+1): If E=0, the count stays.

```
if E = 1 then
   if sclr = 1 then
      Q ← 0
   else
      Q ← Q+1
  end if;
end if;
```

n-bit Parallel access shift register: If E=0, the output is kept.

```
if E = 1 then
   if s_l = '1' then
      Q ← D
   else
      Q ← shift in 'din' (to the right)
  end if;
end if;
```



ALGORITHM

```
C ← 0
while A ≠ 11...1 (2^n-1)
   if a0 = 0 then
      C ← C + 1
   end if
   right shift A
end while
```

- Sketch the Finite State Machine diagram (in ASM form) given the algorithm (for n=8, m=4). (18 pts.)
  - ✓ The process begins when s is asserted, at this moment we capture DA on register A. Then, we shift A one bit at a time. The process ends when $A = 2^n - 1$ (i.e., when z=1). The signal done is asserted when we finish counting.
  - ✓ As A is being shifted: we need to increase the count C every time $a_0 = 0$.

- Complete the timing diagram (n=8, m=4). A is represented in hexadecimal format, while C is in binary format (12 pts.)