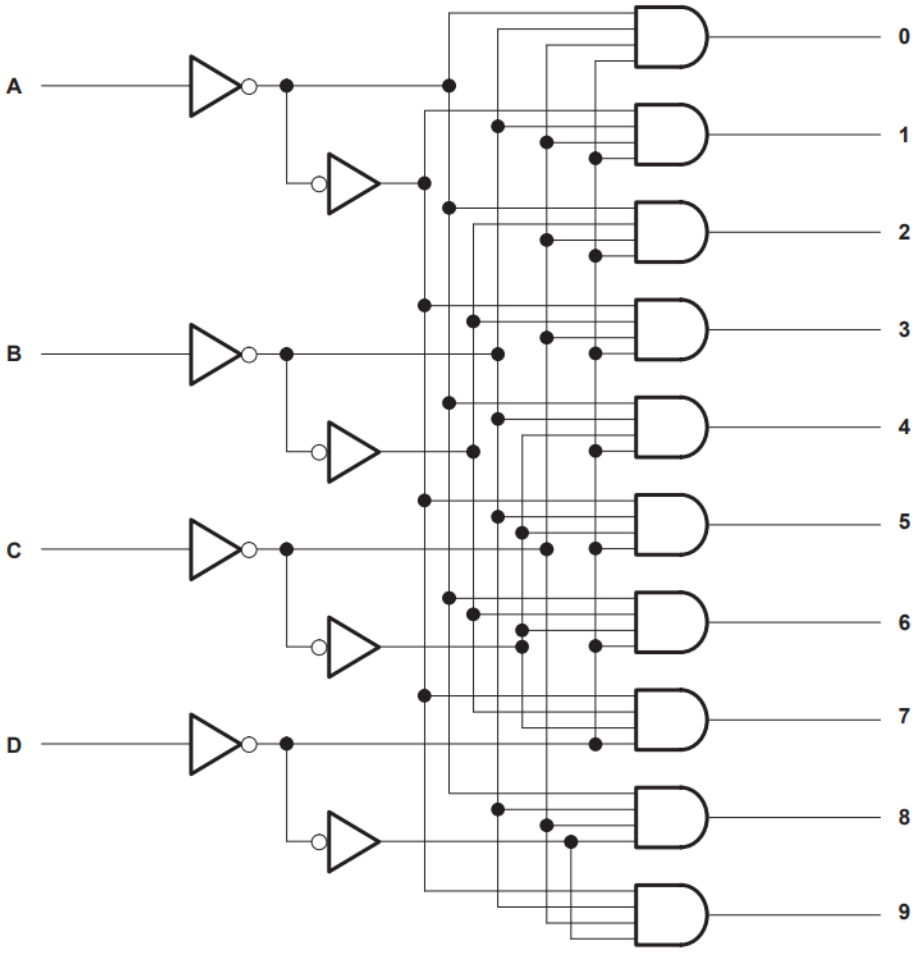
Problem 1 (32 points)

Below is a (slightly modified) excerpt from a 74HC42 chip – a 4-to-10 line decoder. Although 4-to-16 is more traditional, 10 is a common number, and this fits nicely into a standard 16-pin chip (4 inputs + 10 outputs + power + ground).



a) If the inputs (A, B, C, and D) are interpreted as a binary number, which one would be considered the least-significant (as related to the output numbering)?

A

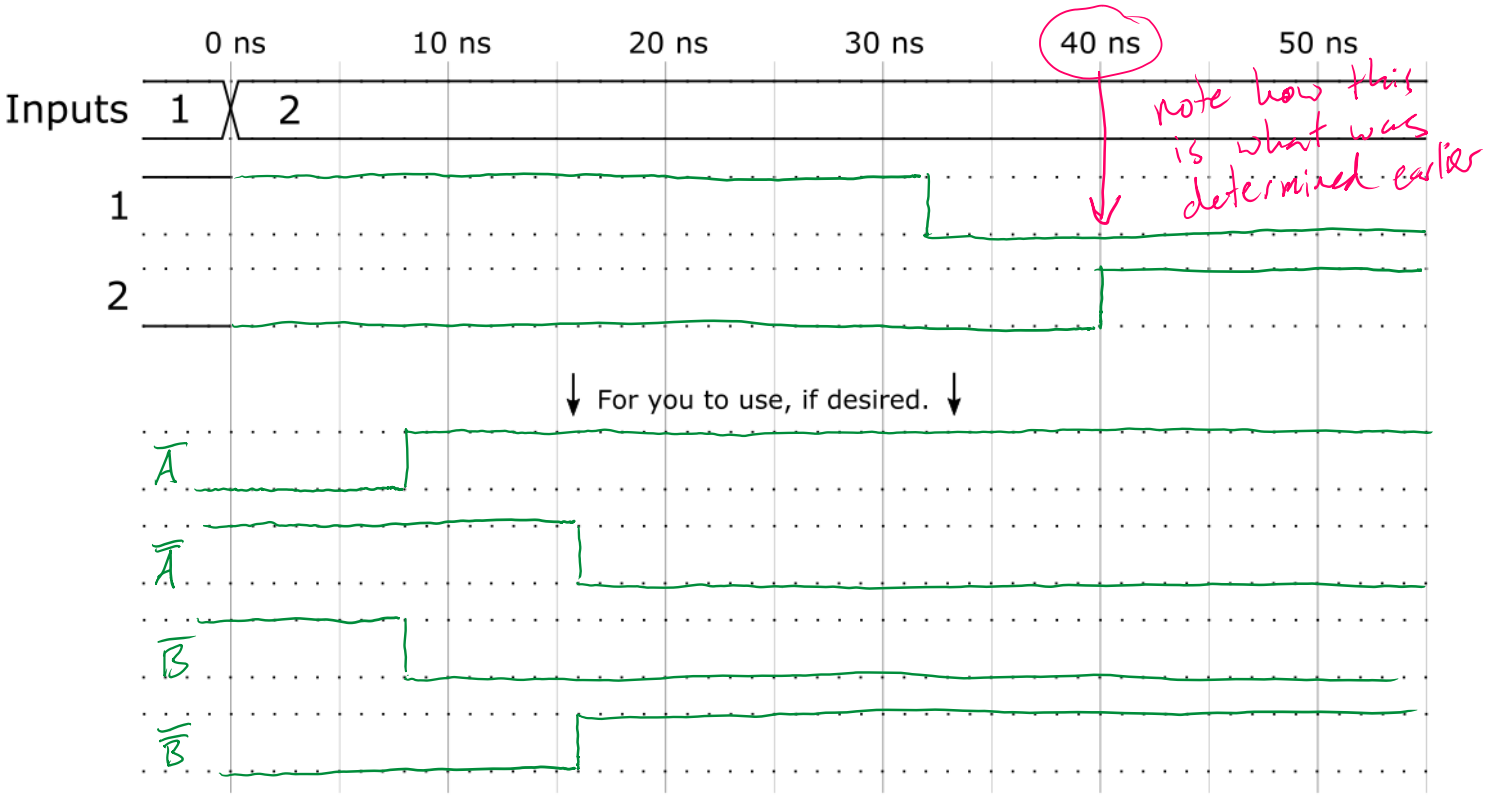
b) Which output(s) will be high if all of the inputs are high?

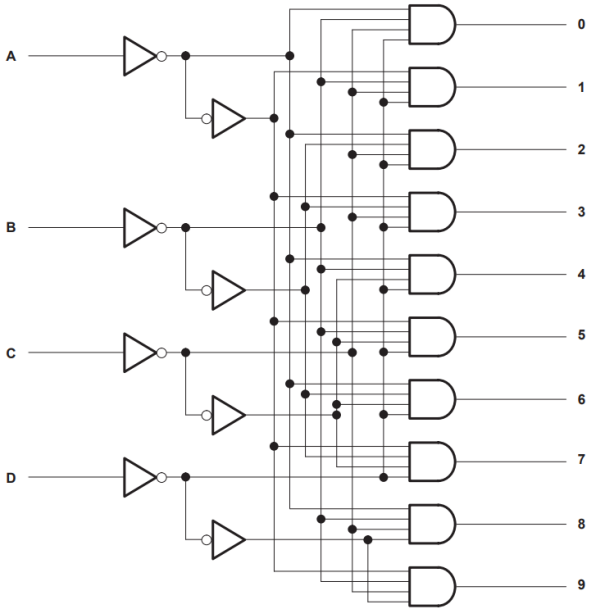
none

c) If the propagation delay through an inverter is 8 ns and the propagation delay through a 4-input AND gate is 24 ns, what is the worst-case delay from any input to any output? In other words, if you change all of the inputs, how long do you have to wait before all of the outputs are correct?

8 + 8 + 24 = 40 ns

d) Complete the following timing diagram for outputs 1 and 2 after the inputs switch from “1” to “2”; in other words, the inputs change from “0001” to “0010” at t = 0 ns. Output 1 is currently high, and eventually output 2 should be high. Extra space is provided at the bottom in case you want to sketch additional signals, and an extra copy of the schematic is below for your reference. Use the 8 ns and 24 ns delays from above.





Problem 2 (25 points)

Provide the decimal equivalent if these binary numbers are interpreted as unsigned, sign-magnitude, 2's complement, and fixed-point numbers in the specified format.

	unsigned	sign-magnitude	2's complement	unsigned fixed-point (NN.NN)
0100	4	4	4	1
1011	11	-3	-5	2.75
1100	12	-4	-4	3

Perform the following conversions to hexadecimal.

$1010_2 =$ A $_{16}$

$11010_2 =$ 1A $_{16}$

$0000111101011010_2 =$ 0F5A $_{16}$

Perform the following unsigned operations. Restrict the result to 5 bits, and indicate if overflow occurred.

```
  01101
 +10001
 -----
  11110
```

overflow? y / n

```
  11
  01101
 +10100
 -----
  00001
```

overflow? y / n

Perform the following 2's complement operations. Restrict the result to 5 bits, and indicate if overflow occurred.

```
  10001
 +11101
 -----
  01110
```

overflow? y / n

```
  01101
 -10100
 -----
  11001
```

negate → 10100 → 01011 → +1 → 01100

```
  11
  01101
 +01100
 -----
  11001
```

overflow? y / n

Perform the following sign-magnitude operation. Restrict the result to 5 bits, and indicate if overflow occurred.

```
  01101
 +10001
 -----
  01100
```

overflow? y / n

You cannot add sign-magnitude numbers bit by bit!

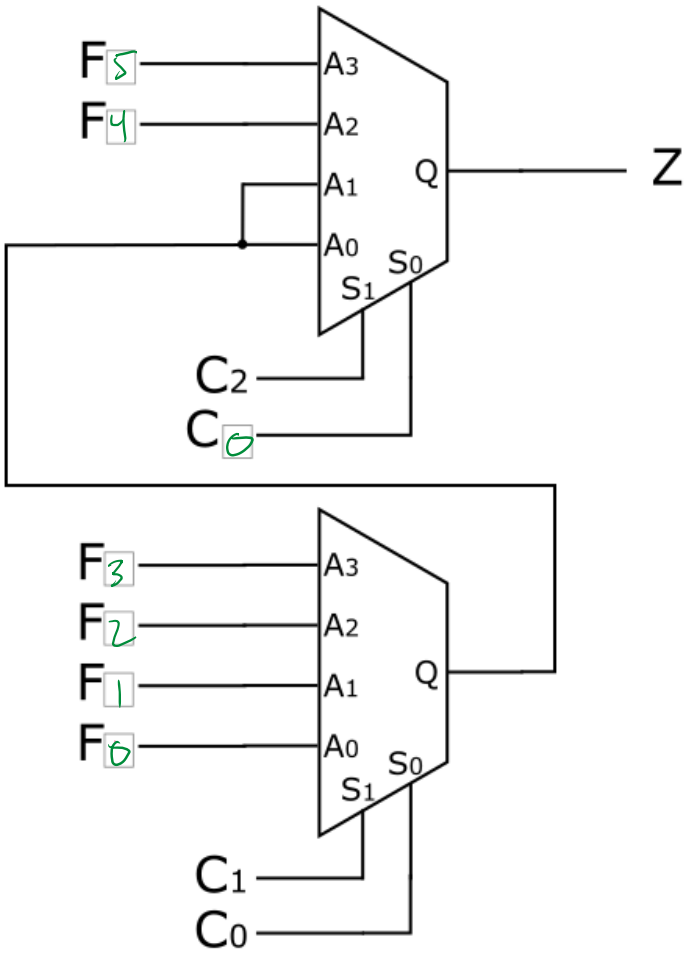
Problem 3 (24 points)

a) Consider two 8-to-3 priority encoders, each with inputs $A_{7\text{-}0}$ and outputs $E_{2\text{-}0}$, but with opposite priority: in one, A_7 has the highest priority and A_0 has the lowest, and in the other, A_0 is highest and A_7 is lowest.
Provide a set of input values (1s and 0s) that would result in the *same* output from both encoders, and one that would result in *different* outputs from each encoder. There are many correct answers here; you only need to provide one for each case.

These inputs will result in the same output from both encoders:	
A_7 =	
A_6 =	
A_5 =	Anything with a single 1
A_4 =	
A_3 =	
A_2 =	
A_1 =	
A_0 =	

These inputs will result in different outputs from the encoders:	
A_7 =	
A_6 =	
A_5 =	Anything with multiple 1s
A_4 =	
A_3 =	
A_2 =	
A_1 =	
A_0 =	

b) The goal of the circuit below is to multiplex six signals (F_5-F_0) onto signal Z based on the three control signals C_2-C_0. Complete the design by filling in the grey boxes. You need to choose the one remaining control signal, and you need to number the F signals appropriately (so that they make sense based on the control signals).



Problem 4 (18 points)

Fill in the result of the specified shifts.

	1 0 1 1 0 1 0 1
Logical shift right by 2	0 0 1 0 1 1 0 1

	1 0 1 1 0 1 0 1
Logical shift left by 2	1 1 0 1 0 1 0 0

	1 0 1 1 0 1 0 1
Arithmetic shift right by 2	1 1 1 0 1 1 0 1

	1 0 1 1 0 1 0 1
Arithmetic shift left by 2	1 1 0 1 0 1 0 0

	1 0 1 1 0 1 0 1
Barrel shift left by 2	1 1 0 1 0 1 1 0

	1 0 1 1 0 1 0 1
Barrel shift right by 2	0 1 1 0 1 1 0 1