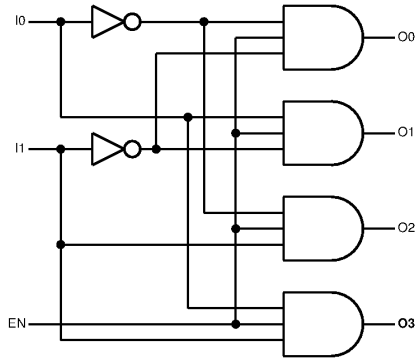
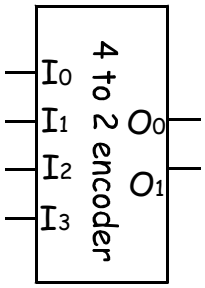
Problem 1 (4 parts, 22 points) Building Blocks

Part A (6 points) Implement a 2 to 4 decoder with basic gates.



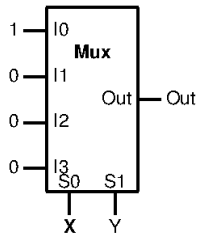

Part B (8 points) Suppose the circuit below has the following input priority: $I_0 > I_2 > I_3 > I_1$. Complete the truth table by filling in the input values that would produce the given outputs and derive a simplified expression for $O_1$.




| $I_3$ | $I_2$ | $I_1$ | $I_0$ | V | $O_1$ | $O_0$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | X | X |
| × | × | × | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| × | 1 | × | 0 | 1 | 1 | 0 |
| 1 | 0 | × | 0 | 1 | 1 | 1 |

$O_1 =$ $\overline{I_0} \cdot (I_2 + I_3)$

Part C (4 points) Which basic gate, having inputs X and Y, does the following circuit implement? Hint: **Complete the truth table at the right.**




| X | Y | Out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**This implements a NOR.**

Part D (4 points) If the mux in Part C is implemented with pass gates and inverters, does the implementation of the basic gate in Part C require fewer transistors than the gate's standard switch-level implementation?

Circle one: **No**

**Support your answer by giving the transistor count for each:**

**# transistors in MUX implementation: 16** **# transistors in switch-level implementation: 4**

Problem 2 (3 parts, 24 points) Number Systems

Part A (10 points) Convert the following notations:

| **binary notation** | **decimal notation** |
|---|---|
| 1100 0101. | **197** |
| **10101.101** | 21.625 |
| 1 1001.011 | **25.375** |
| **hexadecimal notation** | **octal notation** |
| **0x1EC.58** | 754.26 |
| D5.A | **325.5** |

Part B (8 points) For the 12 bit representations below, determine the most positive value and the step size (difference between sequential values). **Express all answers in decimal notation**. Fractions (e.g., 3/16ths) may be used. Signed representations are two's complement.

| representation | most positive value | step size |
|---|---|---|
| unsigned integer (12 bits) . (0 bits) | **4K-1=4095** | **1** |
| signed fixed-point (7 bits) . (5 bits) | **64-1/32** | **1/32** |
| signed integer (12 bits) . (0 bits) | **2K-1=2047** | **1** |
| signed fixed-point (9 bits) . (3 bits) | **256-1/8** | **1/8** |

Part C (6 points) A 16 bit floating point representation has a 10 bit mantissa field, a 5 bit exponent field, and one sign bit.

What is the largest value that can be represented (closest to infinity)? $2^{15}$

What is the smallest value that can be represented (closest to zero)? $2^{-16}$

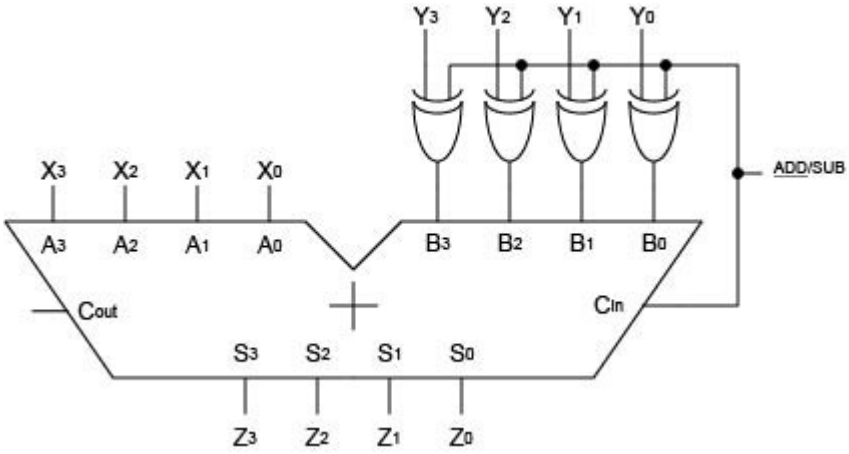
How many decimal significant figures are supported? **3**

Problem 3 (3 parts, 28 points) Adding & Subtracting

Part A (12 points) For each problem below, compute the operations using the rules of arithmetic, and indicate whether an overflow occurs assuming all numbers are expressed using a **six bit unsigned** and **six bit two's complement** representations.

| | 101010 + 11000 | 11011 + 1010 | 111000 - 110110 | 101000 - 1011 |
|---|---|---|---|---|
| result | **000010** | **100101** | **000010** | **011101** |
| unsigned error? | **yes** | **no** | **no** | **no** |
| signed error? | **no** | **yes** | **no** | **yes** |

Part B (6 points) The adder below adds two four bit numbers A and B and produces a four bit result S. Add extra digital logic to support subtraction as well as addition. Label inputs $X_3$, $X_2$, $X_1$, $X_0$, $Y_3$, $Y_2$, $Y_1$, $Y_0$, $\overline{ADD}/SUB$ and outputs $Z_3$, $Z_2$, $Z_1$, $Z_0$.




Part C (6 points) Write two Boolean expressions indicating *signed two's complement* addition and subtraction overflow using any of the signals labeled in part B that are necessary. These expressions should be true when overflow occurs.

addition overflow = $X_3 \cdot Y_3 \cdot \overline{Z_3} + \overline{X_3} \cdot \overline{Y_3} \cdot Z_3$ or $A_3 \cdot B_3 \cdot \overline{Z_3} + \overline{A_3} \cdot \overline{B_3} \cdot Z_3$

subtraction overflow = $X_3 \cdot \overline{Y_3} \cdot \overline{Z_3} + \overline{X_3} \cdot Y_3 \cdot Z_3$ or $A_3 \cdot B_3 \cdot \overline{Z_3} + \overline{A_3} \cdot \overline{B_3} \cdot Z_3$

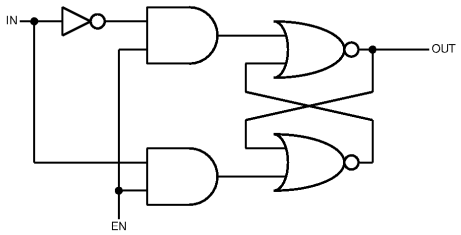
Part D (4 points) Write two Boolean expressions indicating *unsigned* addition and subtraction overflow using any of the signals labeled in part B that are necessary. These expressions should be true when overflow occurs.

addition overflow = $C_{out}$

subtraction overflow = $\overline{C_{out}}$

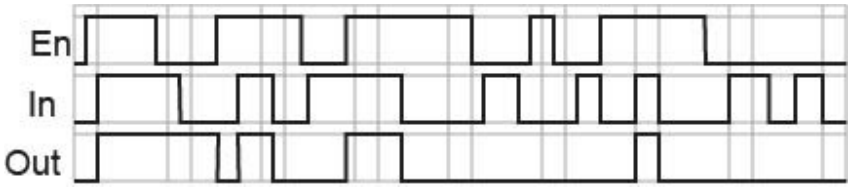
Problem 4 (4 parts, 26 points) Registers and Latches

Part A (8 points) Implement a transparent latch using only NOR gates, AND gates, and inverters. Label the inputs **In** and **En**, and output **Out**. Do not attempt to employ mixed logic notation. Also complete the truth table.

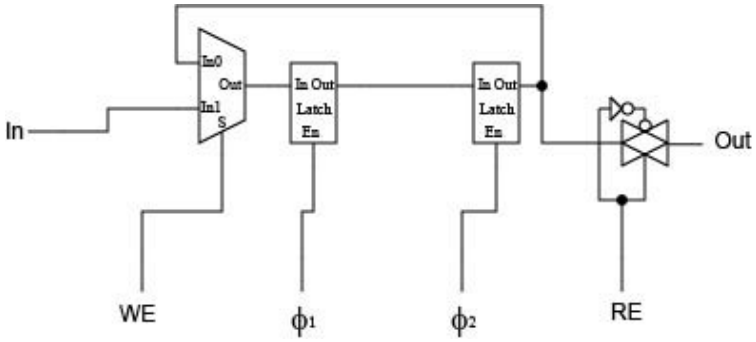



| In | En | Out |
|---|---|---|
| A | 0 | $Q_0$ |
| A | 1 | A |

Part B (7 points) Assume the following signals are applied to a transparent latch. Draw the output signal **Out**. *Assume **Out** is initially zero.*




Part C (4 points) Consider a register with a selectable *write enable* (WE) and *read enable* (RE), implemented as shown below. Describe its behavior by completing the output values. Recall that the CLK signal indicates a full $\Phi_1$ $\Phi_2$ cycle; so the output should be the value at the end of a cycle (with the given inputs). Also indicate when a write and/or a read is being performed.




| IN | WE | RE | CLK | OUT | W? | R? |
|---|---|---|---|---|---|---|
| A | 0 | 0 | ↑↓ | $Z_0$ | **N** | **N** |
| A | 1 | 0 | ↑↓ | $Z_0$ | **Y** | **N** |
| A | 0 | 1 | ↑↓ | $Q_0$ | **N** | **Y** |
| A | 1 | 1 | ↑↓ | **A** | **Y** | **Y** |

Part D (7 points) Assume the following signals are applied to a register with write enable. Draw the output signal **Out**. Draw a vertical line where **In** is sampled. *Assume **Out** is initially zero.*