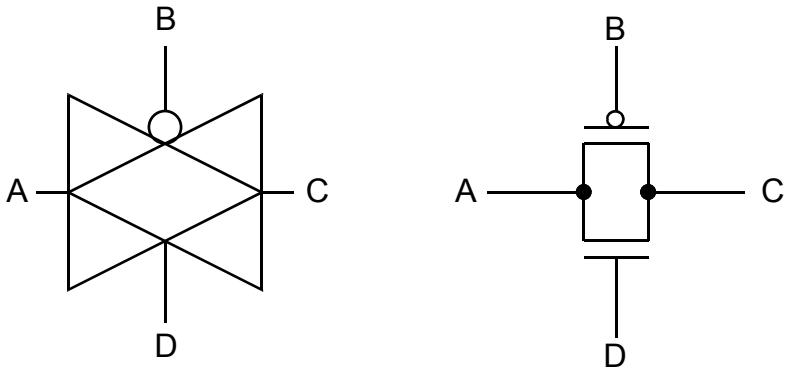
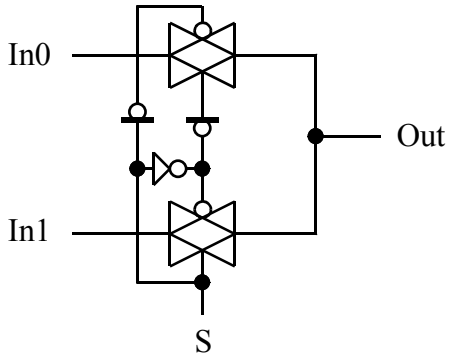
Problem 1 (3 parts, 22 points) Pass Gate Fun

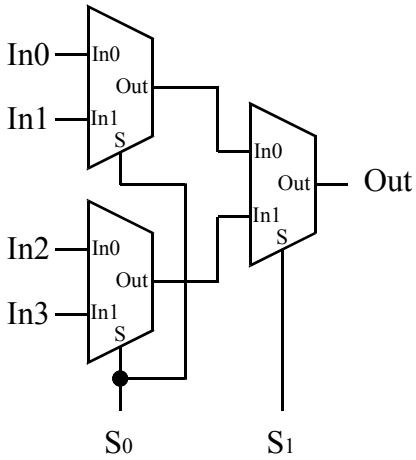
Part A (6 points) For the pass gate symbol shown on the left below, draw its implementation on the right. Be sure that the named terminals (A, B, C, and D) on the right correspond to the icon terminals on the left.



Part B (8 points) Implement a 2 to 1 multiplexer using only pass gates and inverters. Label all inputs (In_0, In_1, S) and output (Out).



Part C (8 points) Using the 2 to 1 mux implemented in part B (in icon form), implement a 4 to 1 mux. Label all inputs (In_0, In_1, In_2, In_3, S_0, S_1) and output (Out)



Problem 2 (3 parts, 28 points) Number Systems

Part A (10 points) Convert the following notations:

decimal notation	**binary notation**
25.125	11001.001
123.5	1111011.1
250.1875	11111010.0011
hexadecimal notation	**octal notation**
FDA.5	111 111 011 010.010 1 = 7732.24
1BE.1	110 111 110.000 1 = 676.04

Part B (12 points) For the 28 bit representations below, determine the most negative value, most positive value, and step size (difference between sequential values). **All answers should be expressed in decimal notation**. Fractions (e.g., 3/16ths) may be used. All signed representations are two's complement.

representation	most negative value	most positive value	step size
unsigned integer (28 bits) . (0 bits)	0	256M	1
signed fixed-point (20 bits) . (8 bits)	-512K	+512K	1/256
signed fixed-point (16 bits) . (12 bits)	-32K	+32K	1/4K
signed fixed-point (14 bits) . (14 bits)	-8K	+8K	1/16K

Part C (6 points) A 28 bit single precision floating point representation has a 21 bit mantissa field, a six bit exponent field, and one sign bit. Express answers in decimal, *not powers of two*.

What is the largest value that can be represented (closest to infinity)? 2B

What is the smallest value that can be represented (closest to zero)? 1/4G

How many decimal significant figures are supported? 6

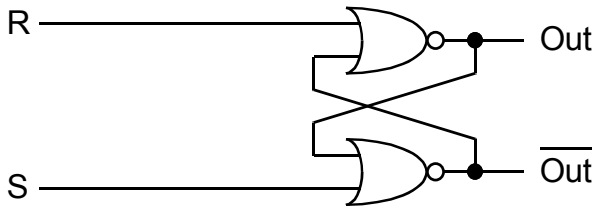
Problem 3 (1 part, 16 points) Arithmetic

Part A (16 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 **five bit unsigned fixed-point** and **five bit two's complement fixed-point** representations.

	1.011 + 1.011	11.111 + 0.001	0.000 - 0.001	10.001 - 10.010
result	10.110	0.000	11.111	11.111
unsigned error?	no	yes	yes	yes
signed error?	yes	no	no	no

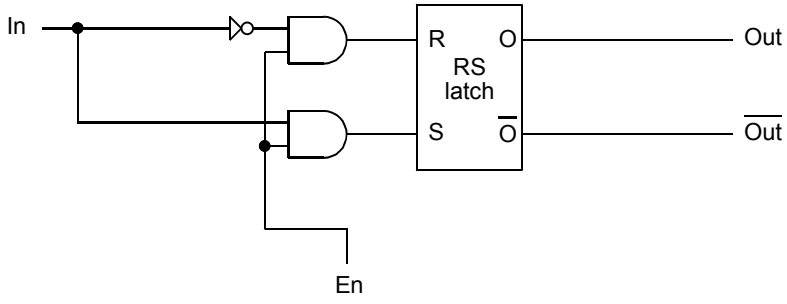
Problem 4 (4 parts, 34 points) State of the Union

Part A (7 points) Implement an RS latch with active high inputs, R and S. Use only basic gates (AND, OR, NAND, NOR, and NOT). Label the inputs and output. Also complete the behavior table. Note -Out means $\overline{Out}$.



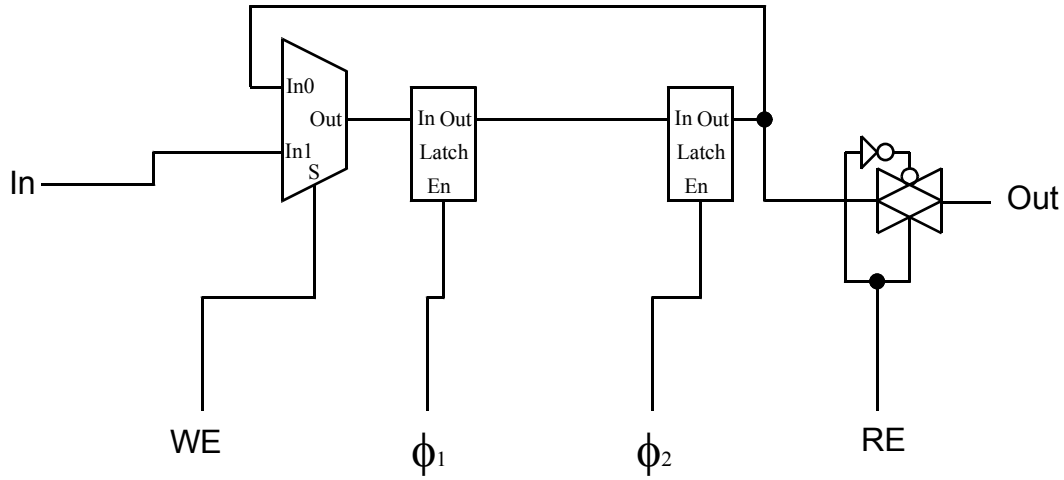
R	S	Out	-Out
0	0	Q_o	$-Q_0$
1	0	0	1
0	1	1	0
1	1	0	0

Part B (7 points) Expand the RS latch to a transparent latch and complete the truth table. Use only basic gates (AND, OR, NAND, NOR, and NOT). Label the inputs and output. Also complete the behavior table.



In	En	Out	-Out
A	0	Q_o	$-Q_o$
A	1	A	-A

Part C (12 points) Build a register using two transparent latches plus a 2to1 mux (draw the labeled icon), a pass gate, and an inverter. Again, complete the behavior table. Recall that the CLK signal indicates a full Φ_1 Φ_2 cycle; so the output should be the value at the end of a cycle (with the given inputs).



In	WE	RE	Clk	Out
A	0	0	↑↓	Z_o
A	1	0	↑↓	Z_o
A	0	1	↑↓	Q_o
A	1	1	↑↓	A

Part D (8 points) Assume the following signals are applied to your register. Draw the output signal **Out**. Draw a vertical line where **In** is sampled. *Draw crosshatch where **Out** is unknown.*