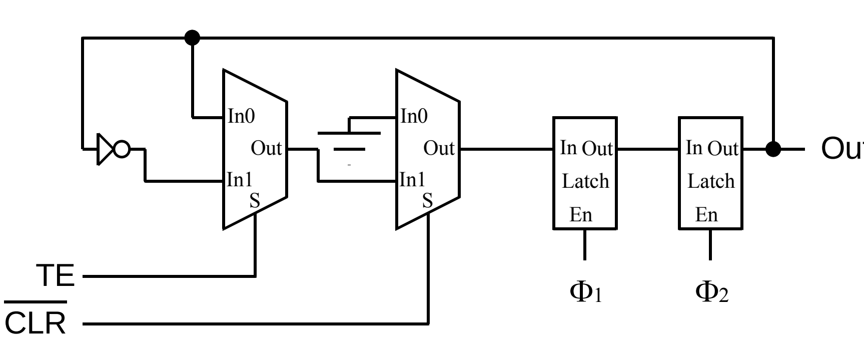
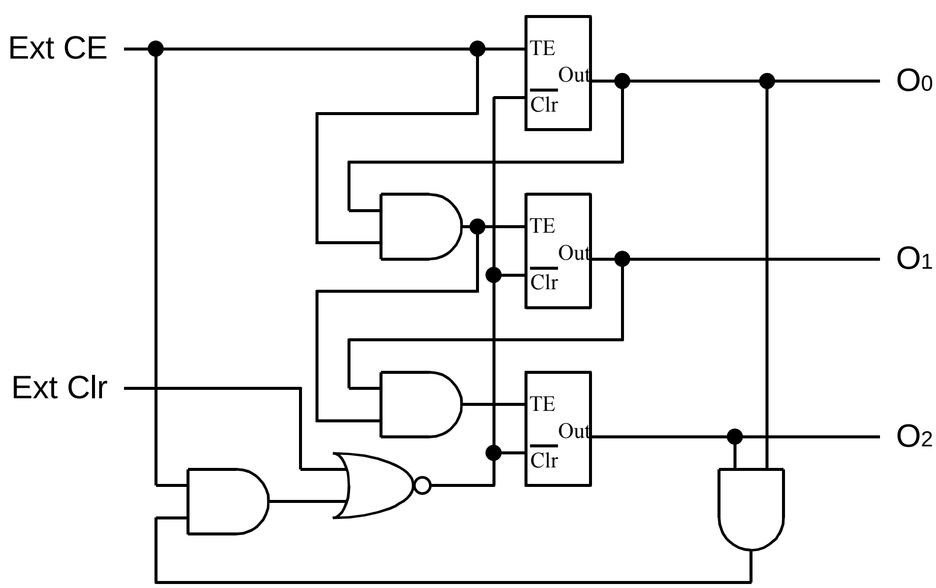
Problem 1 (2 parts, 22 points) Counters

Part A (10 points) Design a toggle cell using two transparent latches, two 2 to 1 muxes, and one inverter. Your toggle cell should have an active high toggle enable input **TE**, and an active low clear input **-Clear**, clock inputs Φ_1 and Φ_2, and an output **Out**. The **-Clear** signal has precedence over **TE**. Label all signals. Also complete the behavior table for the toggle cell.



TE	-Clear	CLK	Out
0	0	↑↓	**0**
1	0	↑↓	**0**
0	1	↑↓	**Q_o**
1	1	↑↓	**$-Q_o$**

Part B (12 points) Now combine these toggle cells to build a **divide by six** counter. Your counter should have an external clear, external count enable, and three count outputs O_2, O_1, O_0. Use any basic gates (AND, OR, NAND, NOR, & NOT) you require. Assume clock inputs to the toggle cells are already connected. Your design must support multi-digit systems.



Problem 2 (3 parts, 30 points) Memory Systems

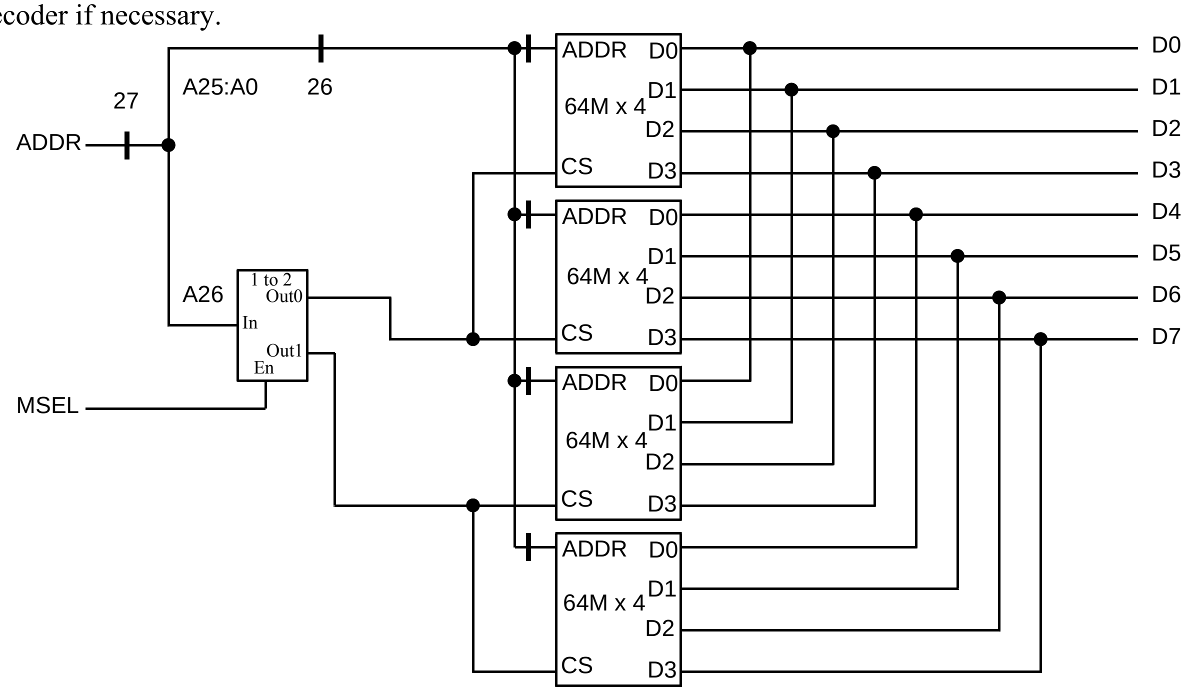
Part A (12 points) Consider a **256 Mbit** DRAM chip organized as **32 million addresses** of **eight bit words**. Assume both the DRAM cell and the DRAM chip are square. The column number and offset concatenate to form the memory address. Using the organization approach discussed in class, answer the following questions about the chip. *Express all answers in decimal (not powers of two).*

number of columns	sqrt(2^{28}) = 2^{14} = 16K
number of words per column	2^{14} / 2^3 = 2^{11} = 2K
column decoder required (*n* to *m*)	14 to 16K
type of mux required (*n* to *m*)	2K to 1
number of address lines in column number	14
number of address lines in column offset	11

Part B (10 points) Consider a **one gigabyte** memory system with **128 million addresses** of **8 byte words** using a **32 million** address by **8 bit word** memory DRAM chip.

word address lines for memory system	$\log_2$(128M) = 27
chips needed in one bank	8 bytes x 2^3 bits/byte / 8 bits = 8
banks for memory system	128M / 32M = 2^{27} / 2^{25} = 4
memory decoder required (*n* to *m*)	2 to 4
DRAM chips required	8 chips/bank x 4 banks = 32 chips

Part C (8 points) Design a 128 million address by 8 bit memory system with four 64M x 4 memory chips. *Label all busses and indicate bit width.* Assume R/W is connected and not shown here. Use a bank decoder if necessary.



Problem 3 (3 parts, 22 points) Datapath Elements and Instruction Formats

Part A (6 points) Suppose the following inputs (in `hexadecimal`) are applied to the 32-bit barrel shifter used in the datapath. Determine the output (in `hexadecimal`). Assume the shift amount (given in `hexadecimal`) is drawn from the 16-bit immediate value.

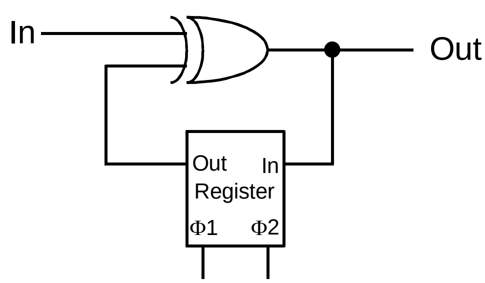
Shift Type	Shift Amount	Input Value	Output Value
logical	FFF0	12345678	56780000
arithmetic	0014	A1024562	FFFFFA10
rotate[1]	000C	F4A11BEE	BEEF4A11

Part B (8 points) For each bitwise logical function specification below, determine the LF code (in hexadecimal) to correctly program the logical unit.

X Y	Out
0 0	LF_0
1 0	LF_1
0 1	LF_2
1 1	LF_3

logical function	LF
$\overline{X \oplus Y}$	9
$X \cdot \overline{Y}$	2
$X + Y$	E
1	F

Part C (8 points) For the state machine below, determine the output given the input and current register output for that cycle. Assume the register output is 0 (low) during the first cycle.



cycle	1	2	3	4	5	6	7	8	9
In	0	1	1	1	0	1	0	1	1
Out	0	1	0	1	1	0	0	1	0

[1] Solution suggested by the National Turkey Association.

Problem 4 (2 parts, 26 points) — Microcode

Using the supplied datapath, write microcode fragments to accomplish the following procedures. Express all values in hexadecimal notation. Use 'X' when a value is don't cared. For maximum credit, complete the description field.

Part A (13 points) $R_4 = \frac{R_1 - (R_2 \times 5)}{2} - R_3$ **Use only R_1-R_4; modify only R_4.**

#	*X*	*Y*	*Z*	*rwe*	*im en*	*im va*	*au en*	*-a /s*	*lu en*	*lf*	*su en*	*st*	*description*
1	2	X	4	1	1	FFFE	0	X	0	X	1	1	R4 <- R2 x 4
2	4	2	4	1	0	X	1	0	0	X	0	X	R4 <- R4 + R2
3	1	4	4	1	0	X	1	1	0	X	0	X	R4 <- R1 – R4
4	4	X	4	1	1	0001	0	X	0	X	1	1	R4 <- R4 / 2
5	4	3	4	1	0	X	1	1	0	X	0	X	R4 <- R4 – R3
6													
7													

Part B (13 points) Write a microcode sequence that sums three packed ten bit values red, green, and blue of R_1 (format shown below). Assume the most significant two bits of the register are zero. Place the sum of the unpacked red, green, and blue values in R_2. **Use only R_1, R_2, and R_3; modify only R_2 and R_3.**

0	0	red	blue	green
31	30	29 … 20	19 … 10	9 … 0

#	*X*	*Y*	*Z*	*rwe*	*im en*	*im va*	*au en*	*-a /s*	*lu en*	*lf*	*su en*	*st*	*description*
1	1	X	2	1	1	03FF	0	X	1	8	0	X	R2 <- R1 & 3FF
2	1	X	3	1	1	000A	0	X	0	X	1	0	R3 <- R1 >> 10
3	3	X	3	1	1	03FF	0	X	1	8	0	X	R3 <- R3 & 3FF
4	2	3	2	1	0	X	1	0	0	X	0	X	R2 <- R2 + R3
5	1	X	3	1	1	0014	0	X	0	X	1	0	R3 <- R1 >> 20
6	2	3	2	1	0	X	1	0	0	X	0	X	R2 <- R3 + R2
7													
8													