Problem 1 (3 parts, 30 points) Memory Systems

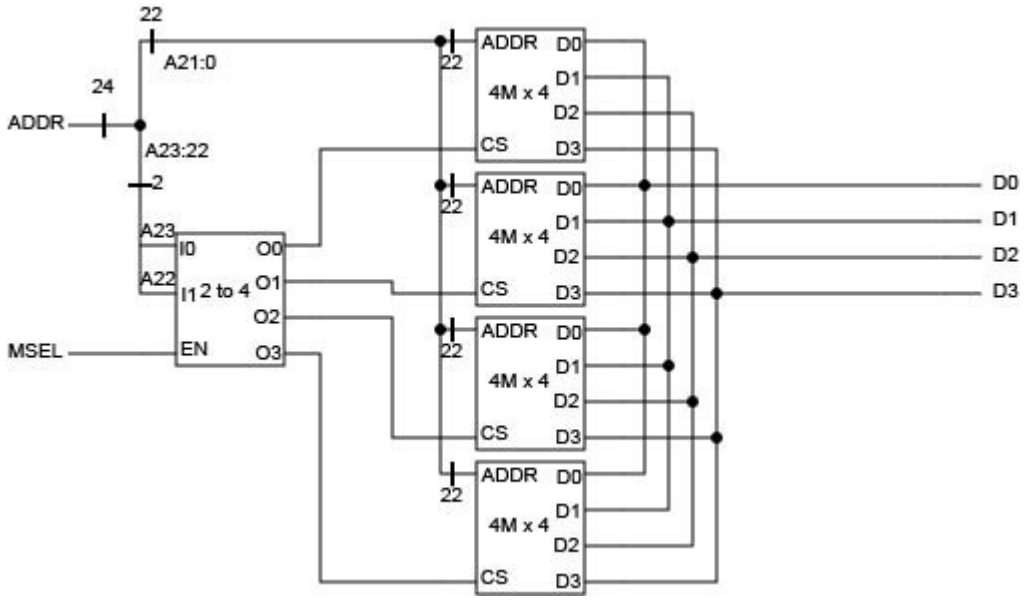
Part A (12 points) Consider a **1 Gbit** DRAM chip organized as **32 million addresses** of **32-bit words**. Assume both the DRAM cell and the DRAM chip are square. The column number and offset concatenate to form the memory address. Using the organization approach discussed in class, answer the following questions about the chip. *Express all answers in decimal (not powers of two)*.

total number of bits in address	**log(32M) = log(2^{25}) = 25**
number of columns	$\sqrt{2^{30}} = 2^{15} = 32\text{K}$
column decoder required (*n* to *m*)	**15 to 32K**
number of words per column	**$2^{15}/2^5 = 2^{10}$ = 1K**
type of mux required (*n* to *m*)	**1K to 1**
number of address lines in column offset	**10**

Part B (10 points) Consider an **8 Gbyte** memory system with **1 billion addresses** of **64-bit words** using a **32 million** address by **32-bit word** memory DRAM chip.

word address lines for memory system	**log(1B) = log(2^{30}) = 30**
chips needed in one bank	**2**
banks for memory system	**$2^{30}/2^{25} = 2^5 = 32$**
memory decoder required (*n* to *m*)	**5 to 32**
DRAM chips required	**64**

Part C (8 points) Design a **16 million address by 4 bit** memory system with **4M 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. Be sure to include the address bus, data bus, and MSEL.



Problem 2 (3 parts, 26 points) Datapath Elements

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 is drawn from the 16-bit immediate value.

Shift Type	Shift Amount	Input Value	Output Value
arithmetic	0x0008	ABCD12EF	**0xFFABCD12**
rotate	0xFFF4	DEAF7892	**0xF7892DEA**
arithmetic	0xFFF0	FACE2537	**0x25370000**

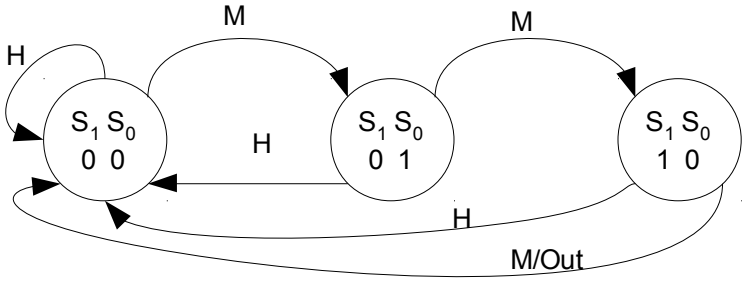
Part B (8 points) For each bitwise logical function specification below, determine the LF code (in hexadecimal) to correctly program the logical unit.

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

logical function	LF
$Y + X$	**E**
X	**A**
$X \oplus Y$	**6**
$X \cdot Y$	**8**

Part C (12 points) Given the following state table, draw the corresponding state diagram below.

S_1	S_0	H/$\overline{M}$	NS_1	NS_0	Out	S_1	S_0	H/$\overline{M}$	NS_1	NS_0	Out
0	0	0	**0**	**1**	**0**	1	0	0	**0**	**0**	**1**
0	0	1	**0**	**0**	**0**	1	0	1	**0**	**0**	**0**
0	1	0	**1**	**0**	**0**	1	1	0	**x**	**x**	**x**
0	1	1	**0**	**0**	**0**	1	1	1	**x**	**x**	**x**



Give the simplified Boolean expression for computing **Out** in terms of the current state and the input.

Out = **S_0M**.

Problem 3 (3 parts, 26 points) Microcode

Using the supplied datapath, write microcode fragments to accomplish the following procedures. Express all values, except memory addresses, in hexadecimal notation. Use 'X' when a value is don't cared. For maximum credit, complete the description field. **In each part, modify only registers 7 & 8.**

Part A (5 points) $R_7 = (R_8 - 21)/128$

#	X	Y	Z	rwe	im en	im va	au en	s/a	lu en	lf	su en	st	ld en	st en	r/w	msel	description
1	8	x	7	1	1	C	1	1	0	x	0	x	0	0	x	0	R7 ← R8-12
2	7	x	7	1	1	7	0	x	0	x	1	1	0	0	x	0	R7 ← R7>>7
3																	

Part B (15 points) Compute the logical OR of mem[4000] and R_3 and store the result in mem[4004] (that is, mem[4004] gets mem[4000]+R_3, where "+" is "logical or").

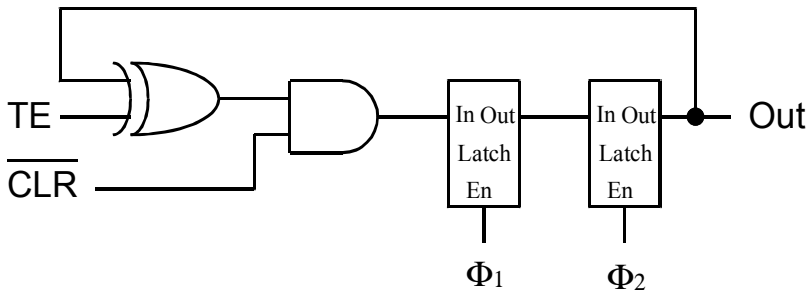
#	X	Y	Z	rwe	im en	im va	au en	s/a	lu en	lf	su en	st	ld en	st en	r/w	msel	description
1	x	x	7	1	1	4000	0	x	1	C	0	x	0	0	x	0	R7 ← 4000
2	7	x	8	1	0	x	0	x	0	x	0	x	1	0	1	1	R8 ← mem[4000]
3	8	3	8	1	0	x	0	x	1	E	0	x	0	0	x	0	R8 ← R8+R3
4	7	x	7	1	1	4	1	0	0	x	0	x	0	0	x	0	R7 ← R7+4
5	7	8	x	0	0	x	0	x	0	x	0	x	0	1	0	1	mem[4004] ← R8
6																	

Part C (6 points) $R_7 = 15 \cdot R_8$ (multiply R_8 by 15)

#	X	Y	Z	rwe	im en	im va	au en	s/a	lu en	lf	su en	st	ld en	st en	r/w	msel	description
1	8	x	7	1	1	FFFC	0	x	0	x	1	1	0	0	x	0	R7 ← R8<<4
2	7	8	7	1	0	x	1	1	0	x	0	x	0	0	x	0	R7 ← R7 - R8
3																	
4																	

Problem 4 (2 parts, 18 points) Counters

Part A (8 points) Design a toggle cell using ***only transparent latches and basic gates (XOR, AND, OR, NAND, NOR, NOT).*** Use an icon for the transparent latches. Your toggle cell should have an active high toggle enable input **TE**, and an active low clear input $\overline{CLR}$, clock inputs Φ_1 and Φ_2, and an output **Out**. The $\overline{CLR}$ signal has precedence over **TE**. Label all signals. Also complete the behavior table for the toggle cell.



TE	$\overline{CLR}$	CLK	Out
0	0	↑↓	**0**
1	0	↑↓	**0**
0	1	↑↓	$\mathbf{Q_0}$
1	1	↑↓	$\mathbf{\overline{Q}_0}$

Part B (10 points) Now combine these toggle cells to build a **divide by thirteen** counter. Your counter should have an external clear, external count enable, and four count outputs O_3, O_2, O_1, O_0. Use any basic gates (AND, OR, NAND, NOR, XOR & NOT) you require. Assume clock inputs to the toggle cells are already connected. Your design must support multi-digit systems.