Problem 1 (3 parts, 30 points) Memory Chips/Systems

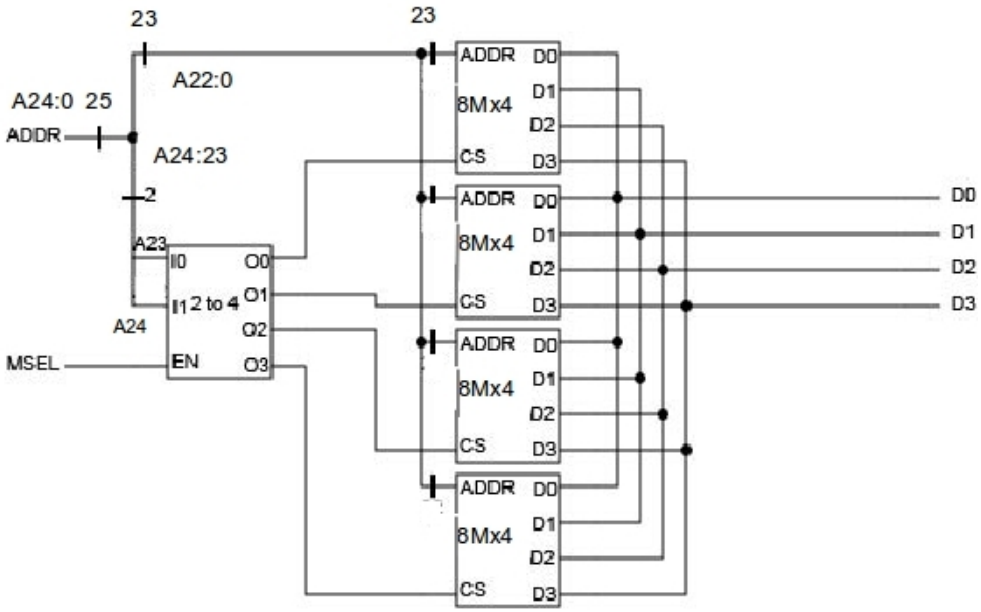
Part A (12 points) Consider a **256 Mbit** DRAM chip organized as **16 million addresses** of **16-bit words**. Assume both the DRAM cell and the DRAM chip are square. The column number and offset concatenate to form the memory address. Using the organization approach discussed in class, answer the following questions about the chip. *Express all answers in decimal (not powers of two)*.

total number of bits in address	**24**
number of columns	$\sqrt{2^{28}}=2^{14}=16\text{K}$
column decoder required (*n* to *m*)	**14-to-16K**
number of words per column	$2^{14}/2^{4}=2^{10}=1\text{K}$
type of mux required (*n* to *m*)	**1K-to-1**
number of address lines in column offset	**10**

Part B (10 points) Consider an **2 GByte** memory system with **256 million addresses** of **64-bit words** using a **16 million** address by **16-bit word** memory DRAM chip.

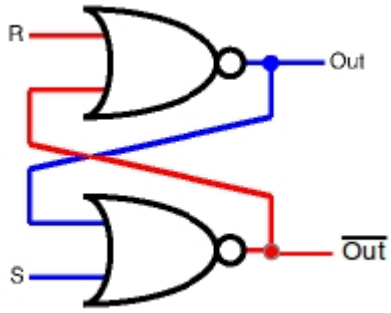
word address lines for memory system	**28**
chips needed in one bank	**64/16 = 4**
banks for memory system	$2^{28}/2^{24}=2^{4}=16$
memory decoder required (*n* to *m*)	**4-to-16**
DRAM chips required	**4*16 = 64**

Part C (8 points) Design a **32 million address by 4 bit** memory system with **8 million x 4-bit** 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.

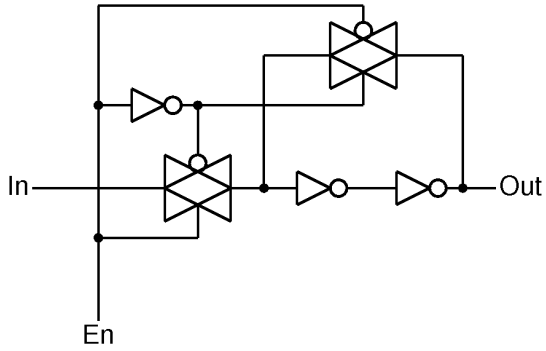


Problem 2 (7 parts, 28 points) Latches, Registers, and Microcode

Part A (4 points) Implement an RS latch using NOR gates. Label the inputs **R** and **S** and the outputs **Out** and **Out**.

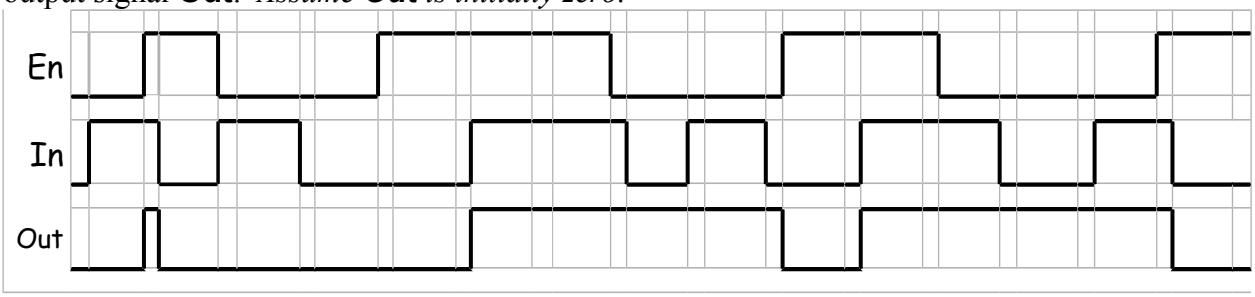


Part B (2 points) Fill in the behavior table for this 10T transparent latch.

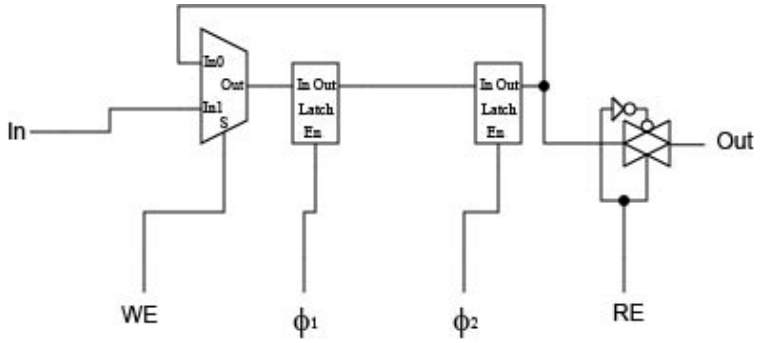


In	En	Out
X	0	Q_0
A	1	A

Part C (6 points) Assume the following signals are applied to a transparent latch. Draw the output signal **Out**. *Assume **Out** is initially zero.*

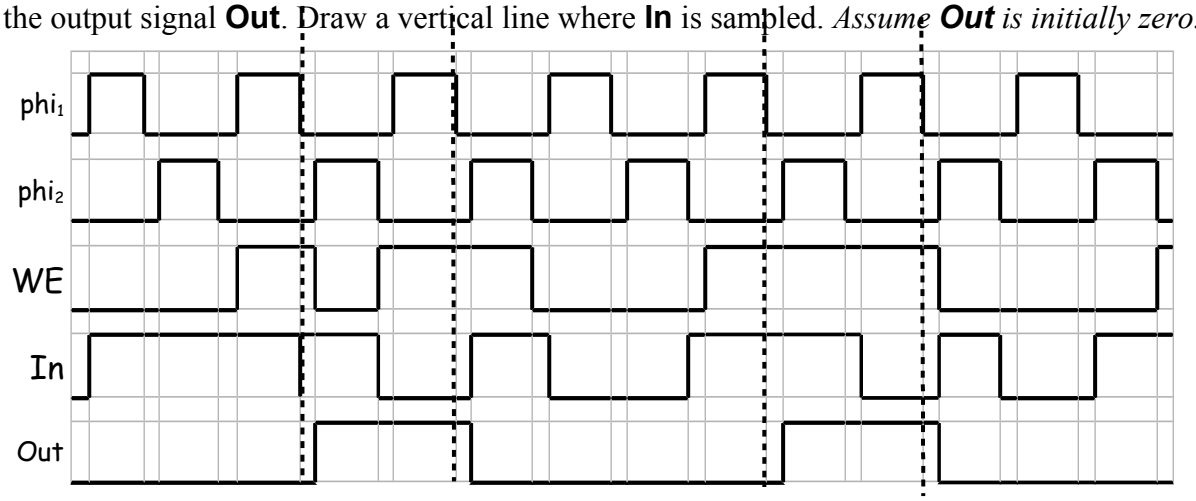


Part D (4 points) Consider a register with a selectable *write enable* (WE) and *read enable* (RE), implemented as shown below. Describe its behavior by completing the output values. Recall that the CLK signal indicates a full Φ_1 Φ_2 cycle; so the output should be the value at the end of a cycle (with 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 E (6 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.*



Part F (3 points) Write a microcode fragment to compute $R_2 = (R_5 - R_7) + R_1$. **Modify only register 2.**

cycle #	*X*	*Y*	*Z*	*rwe*	*au en*	*s/$\bar{a}$*	*description*
1	**5**	**7**	**2**	**1**	**1**	**1**	**R2 = R5-R7**
2	**2**	**1**	**2**	**1**	**1**	**0**	**R2 = R2+R1**

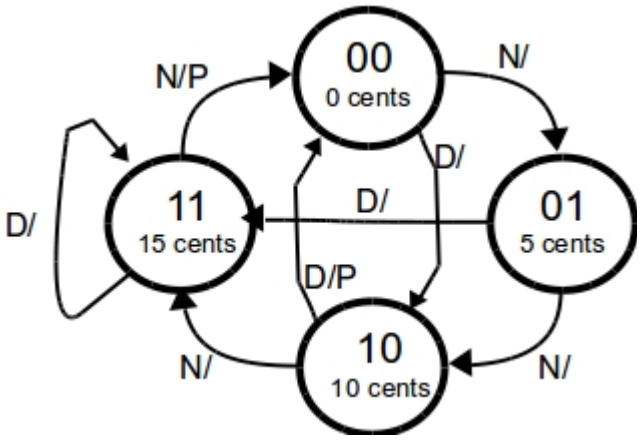
Part G (3 points) Write a microcode fragment to compute $R_2 = 3 \cdot R_5$. **Modify only register 2.**

cycle #	*X*	*Y*	*Z*	*rwe*	*au en*	*s/$\bar{a}$*	*description*
1	**5**	**5**	**2**	**1**	**1**	**0**	**R2 = R5+R5**
2	**2**	**5**	**2**	**1**	**1**	**0**	**R2 = R2+R5**

Problem 3 (3 parts, 22 points) "Keep the Change" State Machines

A state diagram for the controller of the StickyFinger snack machine is shown below. The machine accepts nickels and dimes, but does not give change. It sometimes dispenses a bag of pretzels when coins are inserted. Slugs are not detected. The input to the controller is $\overline{N}/D$, which indicates whether a nickel ($\overline{N}/D=0$) or dime ($\overline{N}/D=1$) is inserted.

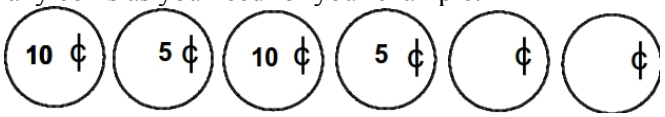
When the P output signal is asserted, a snack bag of pretzels is dispensed. The states keep track of how much money the machine "believes" you have inserted so far. They are number 00 for zero cents, 01 for five cents, 10 for ten cents, 11 for fifteen cents. Unfortunately, as you can see from the state diagram, the machine is not always accurate in keeping track of the money you have inserted.



Part A (4 points) Determine the minimum cost of a bag of pretzels.

Pretzels cost a minimum of _____20_____cents.

Part B (6 points) The machine occasionally gives you pretzels but keeps the change. Give an example of a sequence of coins where you would end up in the 0 cents state (00) with a bag of pretzels, but were overcharged. Fill in as many coins as you need for your example:



How much change did StickyFingers keep in your example? _____**10**_____**cents.**

Part C (12 points) Fill in the state table corresponding to this state machine.

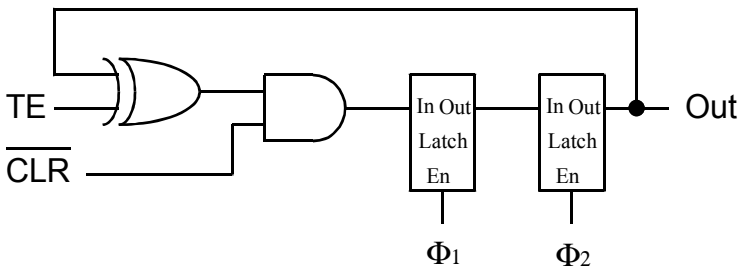
S_1	S_0	$\overline{N}$/D	NS_1	NS_0	P	S_1	S_0	$\overline{N}$/D	NS_1	NS_0	P
0	0	0	**0**	**1**	**0**	1	0	0	**1**	**1**	**0**
0	0	1	**1**	**0**	**0**	1	0	1	**0**	**0**	**1**
0	1	0	**1**	**0**	**0**	1	1	0	**0**	**0**	**1**
0	1	1	**1**	**1**	**0**	1	1	1	**1**	**1**	**0**

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

P = $S_1(S_0 N+\overline{S_0} D)=S_1(S_0 \, xor \, D)$

Problem 4 (2 parts, 20 points) Counters

Part A (10 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	↑↓	**Q_0**
1	1	↑↓	$\overline{Q_0}$

Part B (10 points) Design a **divide by five** counter using toggle cells and the *minimum* number of basic gates. 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, XOR & NOT) you require. Assume clock inputs to the toggle cells are already connected. Your design must support multi-digit systems.