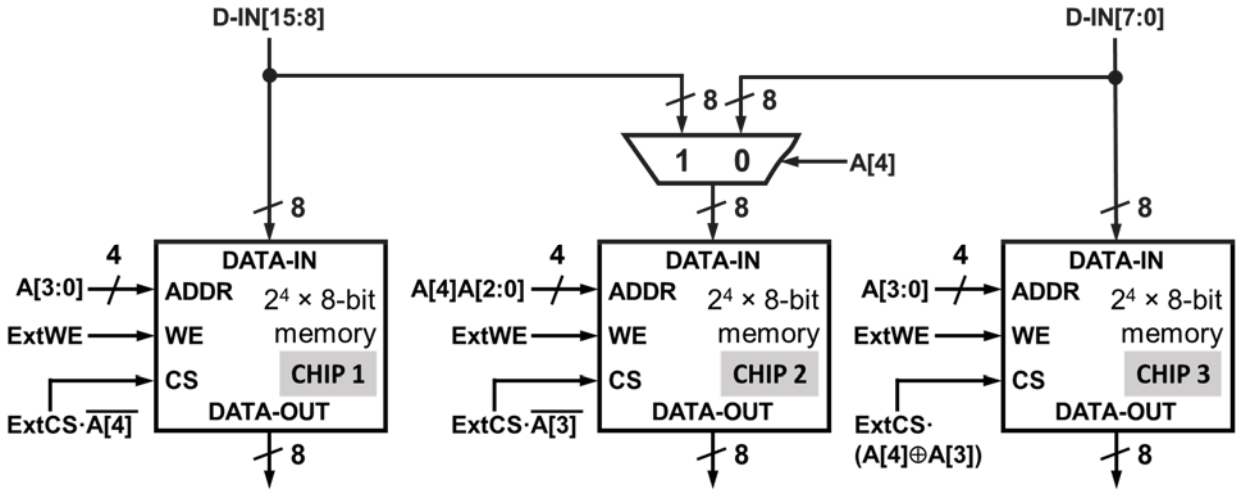
Problem 3 (20 points): Memory

Consider the memory below, which is constructed from three 2^4×8-bit memories. The chip input and output names match those used in the notes. In class, we had a slightly different way of writing the WE signal, which here stands for "write enable." In other words, WE=1 is a write, and WE=0 is a read.

For the system built from the three chips:

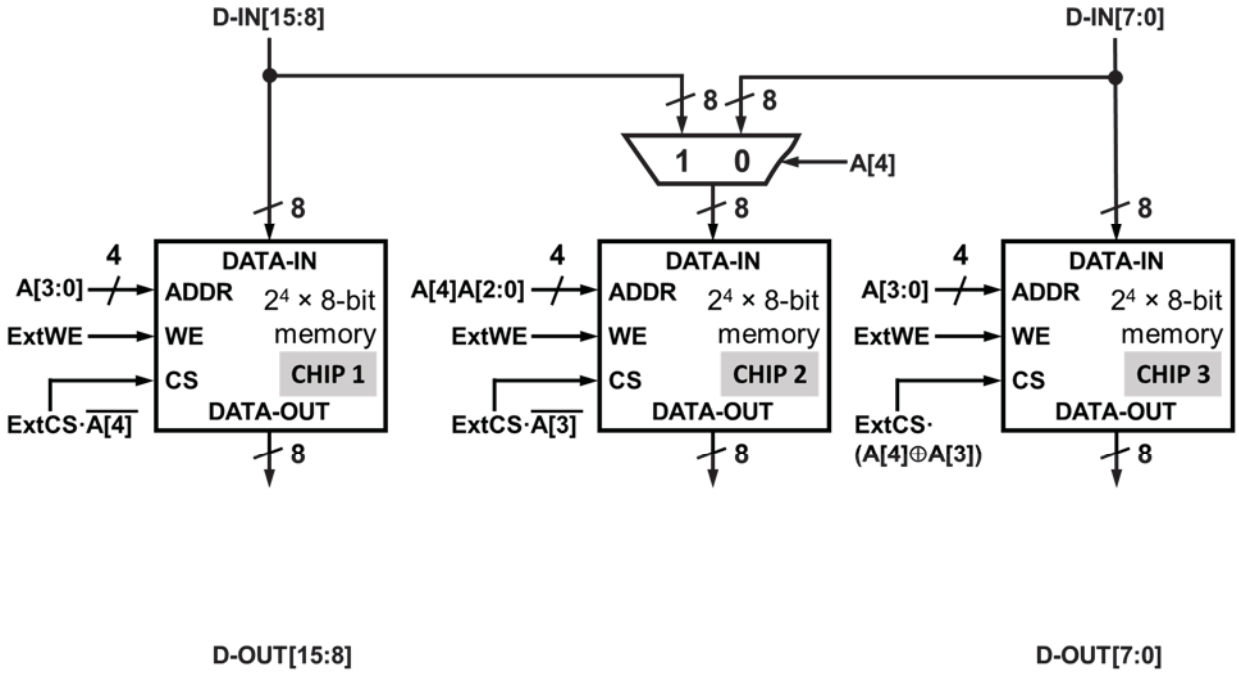
- The external D-IN[15:0] signal provides 16 bits of data input for writes.
- The external A[4:0] signal provides 5 bits of address.
- The external ExtWE signal is for write enable (1 means write).
- The external ExtCS signal is an chip select (1 means active).
- The external D-OUT[15:0] signal (**not shown on this page**; see next page) outputs 16 bits of data for reads.



(8 points) Fill in the following table to indicate in which chip (1, 2, or 3) the high (15:8) and low (7:0) halves of a write are stored, and at what address in the chip. If the write fails, write "N/A" in each box for that write (blank boxes earn no credit).

	D-IN[15:8] stored in…		D-IN[7:0] stored in…	
A[4:0]	**Chip #**	**at ADDR**	**Chip #**	**at ADDR**
01011				
00101				
10110				
00001				
11010				

Problem 3, continued: (diagram extended from previous page: D-OUT labels added)



(2 points) How many memory addresses are there in the memory shown above?

(6 points) Complete the diagram above by connecting the three chips' DATA-OUT signals to the external D-OUT signals. Remember that for any valid address A, a read operation on the address A should return the bits last written to address A.

Use any gates and components that suit your purpose (and were developed in class: adders, comparators, muxes, encoders, decoders, and so forth). However, **overly complex designs will earn little or no credit**.

You may not alter the implementation in any way other than to connect the chips to D-OUT.

(4 points) Circle EXACTLY ONE ANSWER:
Why is it hard to build a 2^5×12-bit memory from three 2^4×8-bit memories?

A) 12 is not a power of 2.

B) The number of bits stored in the three 2^4×8-bit memories does not match the number needed for a 2^5×12-bit memory.

C) The 2^4×8-bit memories do not allow writes to half of the bits at a specified address.

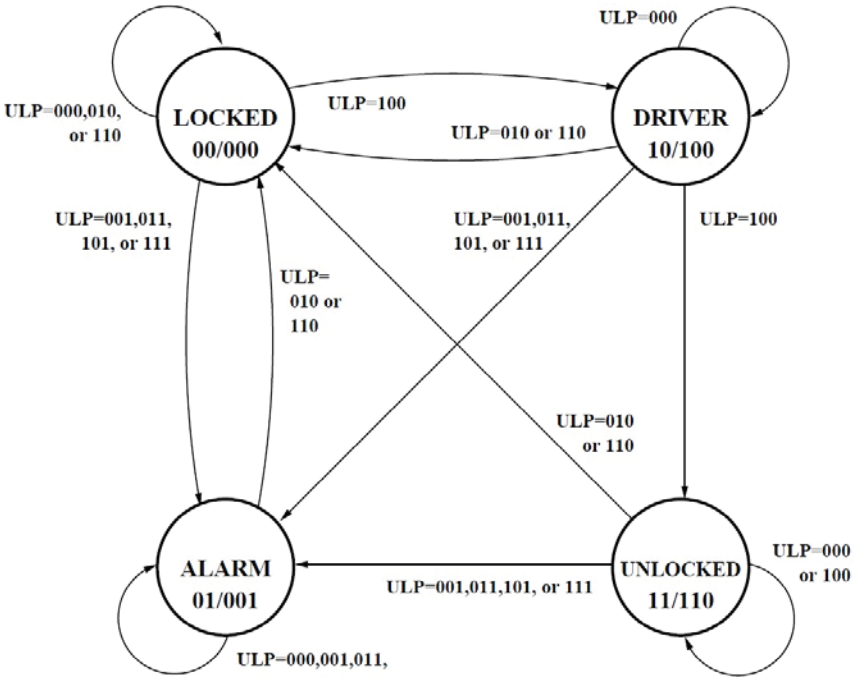
D) Using three memories dramatically decreases the speed of the system.

E) The logic necessary to read bits from the 2^4×8-bit memories is too complex to build.

Problem 4 (15 points): Keyless Entry Revisited

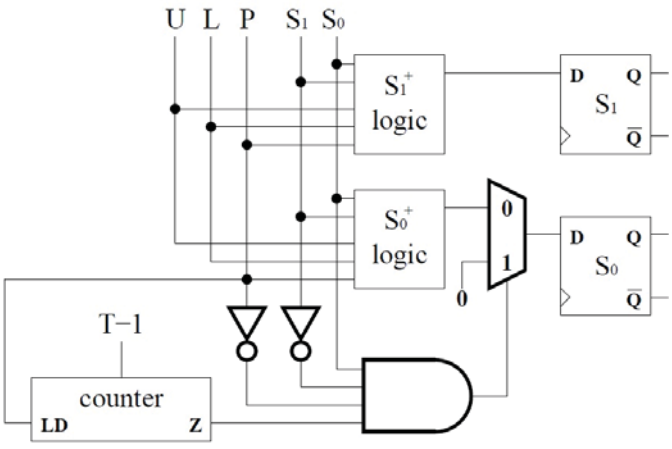
Prof. Lumetta has had another Good Idea! In this problem, you must make another extension to the keyless entry design. First, let's review the current design. The problem is on the next page.

The state diagram for the original keyless entry system appears to the right. As you may recall, states are marked with the state bits and the outputs, S_1S_0/DRA, where D controls the driver's door, R controls the other doors, and A controls the alarm. Transitions are marked with ULP, where U is the unlock button, L is the lock button, and P is the panic button.



In class and in the notes, you learned how to extend our original keyless entry system with a timeout to enable the alarm to turn off after T cycles.

The implementation of this extension appears to the right. The counter is set to T-1 whenever the panic button (P) is pressed. When the system is in the ALARM state, the timer has counted down to 0 (Z=1), and the panic button has not been pressed (P=0), the mux between the S_0^+ logic and the S_0 flip-flop forces the system to move from ALARM to LOCKED.



Problem 4, continued:

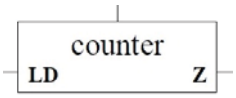
Now Prof. Lumetta wants you to **use the same counter** to lock the doors if no buttons have been pushed for Q cycles. In other words, if the system is in the DRIVER or the UNLOCKED state, and no buttons are pushed, the system should transition to the LOCKED state.

As with the extension made to turn off the alarm, your design should **reset the counter whenever the unlock button (U) is pushed** (whenever the system enters DRIVER or UNLOCKED because a button is pressed).

Draw circuits to complete the UNLOCKED/DRIVER timeout extension **while maintaining the original ALARM timeout extension**. The signals U, L, P, S_1, S_0, and Z (and all complemented versions) are available as inputs. The constants T and Q are also available.

Use any gates and components that suit your purpose (and were developed in class: adders, comparators, muxes, encoders, decoders, and so forth). However, **overly complex designs will earn little or no credit**.

(5 points) Begin by adding some logic to control the counter.



(5 points) Next, add logic between the original S_1^+ value and the S_1 flip-flop.

(5 points) Finally, add logic between the original S_0^+ value and the S_0 flip-flop.