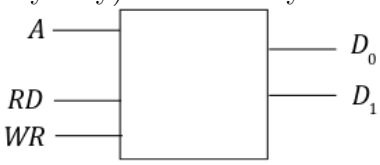
P116B Homework 4

Due 3/1/2019

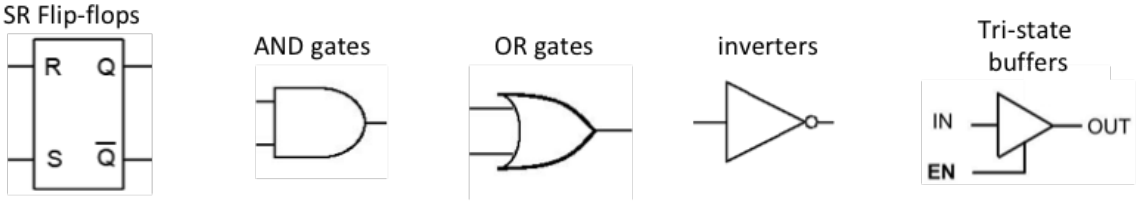
1. Design a simple (some would say silly) 2x2 memory



to store data into two 2-bit registers. It should behave as follows:

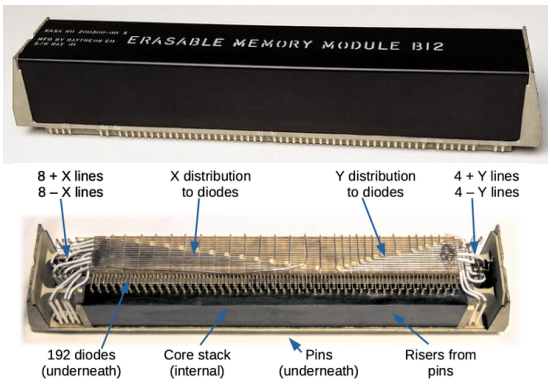
- When $WR = 1$, the two D pins will act as inputs
  - $A = 0 \rightarrow$ store the values of D_0 and D_1 in to data register 0.
  - $A = 1 \rightarrow$ store the values of D_0 and D_1 in to data register 1.
- When $RD = 0$, the two D pins will be high-Z.
- When $RD = 1$, the two D pins will act as outputs
  - $A = 0 \rightarrow$ assert the values in register 0 to D_0 and D_1.
  - $A = 1 \rightarrow$ assert the values in register 1 to D_0 and D_1.
- Operation is asynchronous (i.e., there's no clock signal).
- WR and RD are never both equal to 1, so you don't need to handle conflicts.

You may use the following parts in your design.



To simplify the design, your AND and OR gates may have *any number of inputs you need*. You may not need all parts. Hint: your "registers" will be made from the SR flip-flops. You will need (2 registers)x(2 bits) = 4 of them.

2. Just for fun: advances in memory. Below are two RAM devices:



The device shown on the left is the Erasable Memory Unity (EMU) from the Apollo Guidance Computer (AGC), which took men to the moon and back. It had 2048 16-bit words (4kByte), stored in RAM based on magnetic cores. On the right is a 512 GByte MicroSD memory-based "disk drive".

Answer the following questions:

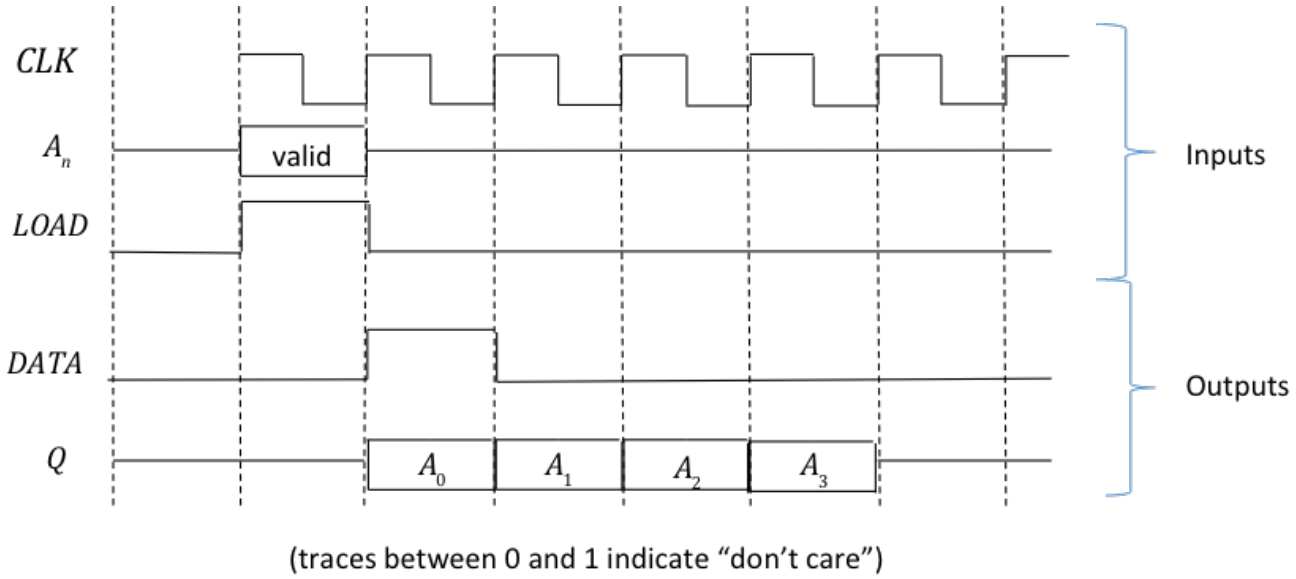
(a) How many of the EMUs would it take to equal the memory of the MicroSD card shown?

(b) If the dimensions of each EMU are 5cm x 5cm x 30cm, and the EMUs in part (a) were stacked in a cube, how long would the sides be?

(c) The dimensions of the MicroSD chip are 11mm x 15mm. If the active memory area is about 1cm^2, what is the average feature size of 1 bit of memory? Please express your answer in nm^2.

3. Repeat the first problem from the last homework using VHDL code rather than discrete parts to design a 4-bit synchronous "parallel to serial" converter.

   As a reminder, this is a circuit that will load a four bit word (A_3, A_2, A_1, A_0) in response to a "LOAD" signal, and then clock the bits out serially to Q over the next four clock cycles. The circuit should also issue a DATA output concurrent with the first serial data bit, to signal the start of serial data, as illustrated in the timeline below



(traces between 0 and 1 indicate "don't care")

   As before, you may assume that the A_n bits are valid for at least one cycle, and the LOAD bit is synchronously asserted for exactly one clock cycle.

   Make your code as complete as possible, including the ENTITY, ARCHITECTURE, and PROCESS parts. The signals defined above should be the inputs and outputs, but you do not need to assign them to pins.

   I understand it's hard to write code without a compiler to check it, so you'll get full credit if the design is logically correct, even if it still has some syntax errors.