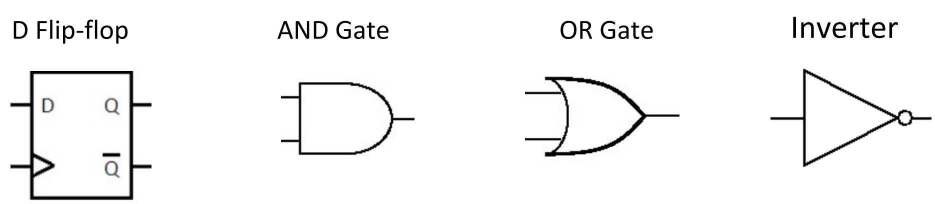
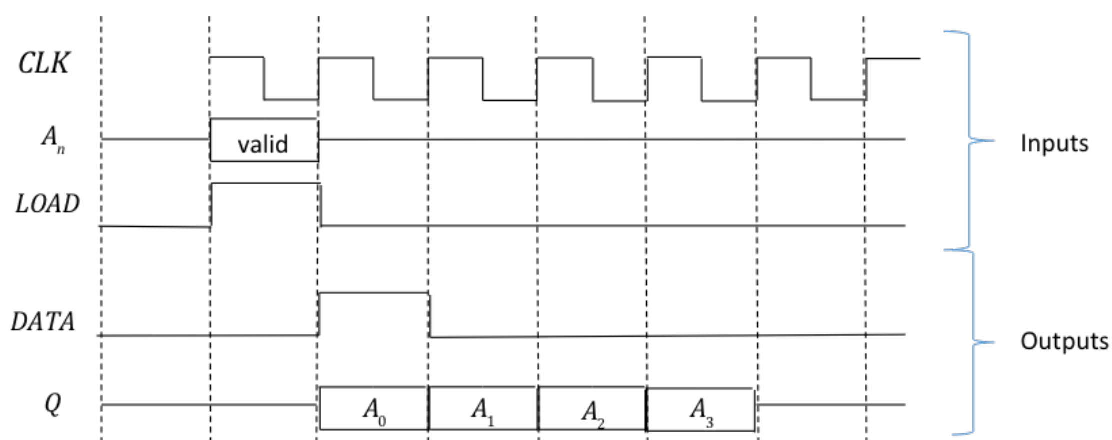
P116C Homework 1

Due 4/8/2022

1. Solve the following problems by converting the numbers to binary and solving with twos complemement arithmetic. For the subtractions, convert the second number to a 2s complement negative and add. Use 8-bit signed integers Convert back to decimal and verify that your answers are correct.
   (a) $42 + 33$
   (b) $37 - 5$
2. Perform the following calculations using 8-bit, 2s complement math
   (a) 10×5
   (b) -11×5
3. Using the following components:



design a 4-bit synchronous "parallel to serial" converter; that 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

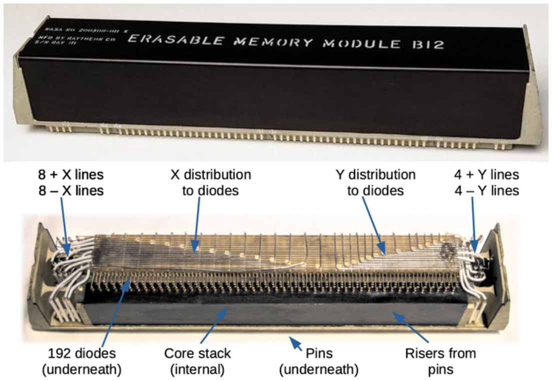


(traces between 0 and 1 indicate "don't care")

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. There's more than one way to do this. Try to design your circuit as efficiently as possible.

(Hint: start with a shift register, and then think about how you would modify it to load in the A_n bits on a LOAD cycle, rather then the next higher bit.)

4. 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.

5. Understanding data sheets.

   The UD61464 is a 64k x 4 bit dynamic ram. The datasheet for this chip can be found at the Canvas site, in the homework directory, labeled "UD61464-datasheet.pdf". 64k corresponds to a 16 bit address space, but locations are arranged in 256 rows of 256 columns each. Sequential memory locations start at row 0, column 0 and proceed by column, after column 255, they advance to column 1, and so on. In other words, for a 16 bit address, the most signifcant 8 bits are the column and the least significant 8 bits are the row; that is, in hex, the address is decodes at 0xRRCC. So for example, (decimal) address 13576 is (google it) 0x3508 or row = 0x35 (53 decimal), column 8 (8 decimal).

   Like most memory chips, the UD61464 uses the same 8 address lines for the row and column numbers, and what they're interpreted as depends on the $\overline{RAS}$ and $\overline{CAS}$ lines. In "Fast Page Mode Write", the key signals are act as follows:

   - $\overline{RAS}$ - When this signal goes from HI to LO, the chip will latch 8 address bits $A[7:0]$ in as the ROW number, and hold them for as long as the bit remains LO.
   - $\overline{CAS}$ - When this signal goes from HI to LO, the chip will load the 8 address bit $A[7:0]$ as the COLUMN number. If the $\overline{W}$ bit is also LO at this time, it will write the data on bits $D[3:0]$ into the memory location corresponding to the current ROW and COLUMN. If $\overline{W}$ is HI, it will read out the data stored at them memory location and assert it on the data lines.

   Based on this data sheet, create a time line to write 0xA to address 0xAA50 and 0x5 to address 0xAA51 using "Fast Page Mode Write", described on page 11 of the data sheet. Your time line should include the following signals:

   - $\overline{RAS}$
   - $\overline{CAS}$
   - $\overline{W}$
   - $A_0 \rightarrow A_7$ (use a separate line for each address bit)
   - $DQ_0 \rightarrow DQ_3$ (use a separate line for each data bit)

   You don't need to worry about the details of the timing, just make sure everything comes in the correct order.