Danowitz | Fall 2020, CPE523
Handout 2 | Due 9/29/2020

# Lab 1: Exploring SRAM Layout

## Introduction

An SRAM cell is a basic memory element that consists of two cross-coupled inverters. Due to its high speed and small footprint, it is used as the core of most logic memory structures, including processor caches. The gate level schematic of this cell is given in Figure 1.

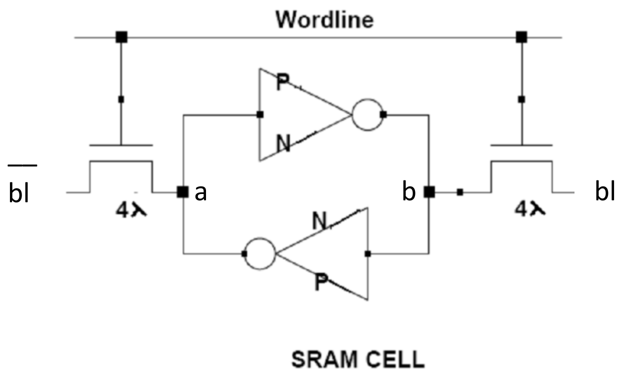


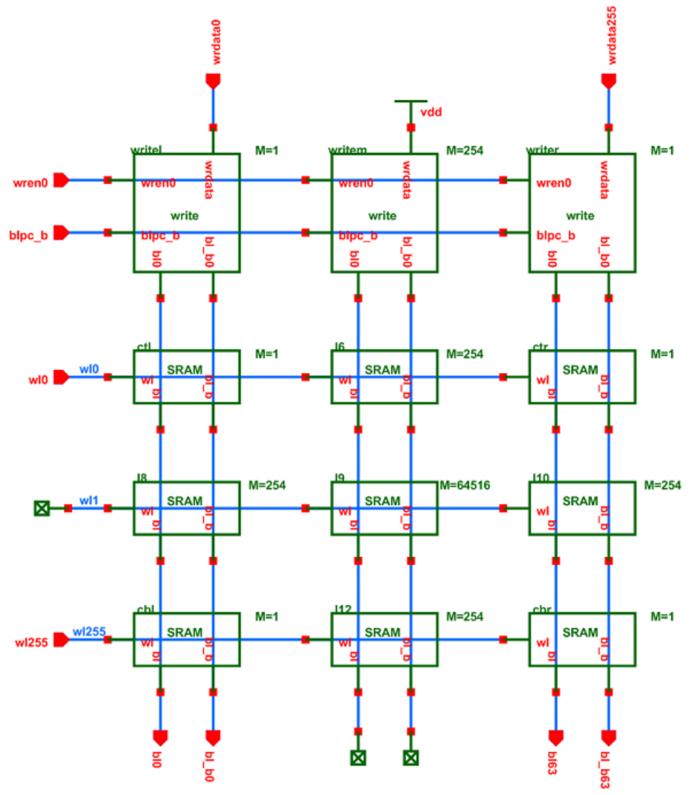
**Figure 1. Gate level schematic of an SRAM Cell**

The operation of an SRAM is a little tricky, since it only has one control line (the wordline) that is used for both reading and writing the cell. The trick is that a zero (low voltage) on a bitline will overpower a high voltage in the cell, but a one (high voltage) on the bitline will not over power a low voltage in the cell. So to write a zero, the bl line is pulled to Gnd, to write a one, the $\overline{bl}$ line is pulled to Gnd, and to read both $bl$ and $\overline{bl}$ are pulled initially high. Let's look at writing a one in a little more detail. The Wordline is pulled high and $\overline{bl}$ is driven to Gnd and *bl* is driven to Vdd. The drive strength of $\overline{bl}$ is sufficient to overpower the charge stored on node "a," and the pMOS load. It drives node a low, which then flips the other inverter, which then drives a low. The cell is now storing a one, and the wordline can fall without having the cell forget its value. Once the value has been flipped, the wordline is dropped low, leaving a stable 0 on node a, and a stable 1 on node b. To read a value from the SRAM cell, $bl$ and $\overline{bl}$ are charged to 1 and left floating. The wordline is pulled high, and the inverters either pull $bl$ or $\overline{bl}$ low depending on the values stored on "a" and "b."

In a large memory, SRAM cells are arrayed into a rectangle, with several cells sharing every wordline, and bitline. A sample 3x3 SRAM memory array is shown in Figure 2. Since most memories have a large number of cells, memory designers spend tons of time trying to make the memory cell small. They usually have a different set of design rules for memory cells to allow them to cheat to make the cells smaller. In this problem, you will complete a stick diagram of an SRAM cell and of the 3x3 array.

**Figure 2. A sample 3x3 SRAM array**

You can flip/rotate every other cell/row to make the arrayed area smaller. Designers often do this to allow two cells to share contacts / put nMOS devices in one cell close to the nMOS devices in another cell. Your memory cell can use two levels of metal in the cell. I suggest you either use M1 for the wordline, and M2 for the bitlines or vice versa. Remember that Vdd and Gnd need to be routed in metal too, although they can be routed either in parallel with the wordline or the bitlines. It is better if Gnd is routed with the bitlines, but you can do it either way.



## Tasks/Deliverables

a. First draw a transistor level diagram of the SRAM Cell.

b. Complete a stick diagram of the SRAM cell. Use colored pencils or some other method to clearly indicate the different layout elements (e.g. pink for polysilicon, blue for M1, etc). Make sure you have routed Vdd and Gnd. Also if you are not putting in well and substrate contacts in your cell, you need to create a cell with these contacts in them, and show where they will be placed in the array.

c. Remember that a stick-level diagram is an immediate precursor to a custom layout. Since full layout is beyond the scope of this class (Take Prof. Smilkstein's 431 if interested!), I'd like you to go on Google Scholar or elsewhere online and find a picture of a layout of an actual SRAM 6T transistor cell. Include this picture in your write-up and compare their layout to your layout. Why might they differ?