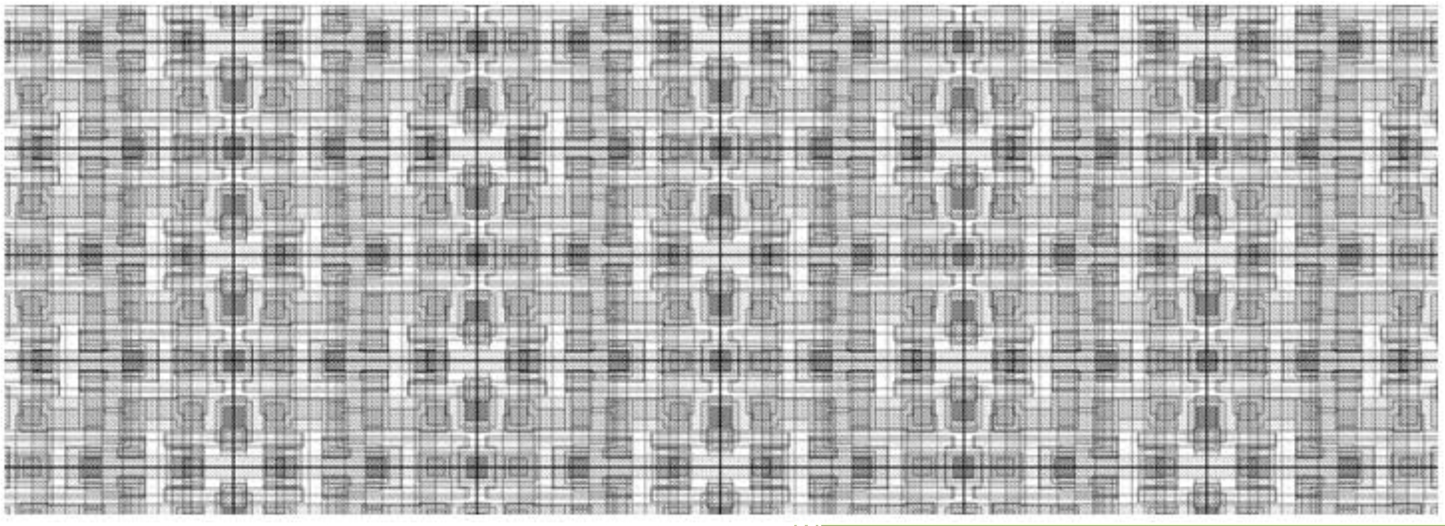
ELEN 4321 VLSI

Project – 4 kilobyte SRAM array



Shrivathsa Bhargav

Jaime Peretzman

ELEN 4321 VLSI circuits, Fall 2007

Designing an SRAM array at the 90nm CMOS tech node

Shrivathsa Bhargav and Jaime Peretzman
ELEN 4321 – Digital VLSI circuits
Columbia University, Fall 2007

Table of Contents

Goals 3
Introduction 3
Functional blocks 4
    SRAM single cell 4
    SRAM Array 8
    Precharge 9
    Sense Amplifier 10
    Decoder 11
Putting it together 13
Conclusion 13
DRC and LVS 16

Goals

The main goal of this project was to design a 64-bit x 64-bit (4 kilobyte) SRAM array. A single SRAM was provided, and all surrounding circuitry, including the read, precharge, and address decoding mechanisms had to be designed and implemented. The final design had to be tested in its schematic form (in Cadence Schematic and Analog Artist), as well as in layout (in Cadence Virtuoso).

Introduction

The SRAM is heralded as the most optimized digital circuit in the semiconductor industry. The MOSIS design rules (that ensure proper fabrication) have long been jettisoned in order to conserve valuable space and boost packing density.

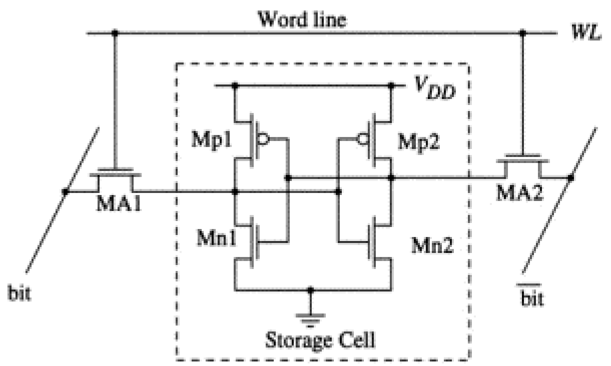


Figure 1 - 6-transistor SRAM cell

The detailed functional theory behind the SRAM cell is not discussed here, but can be summarized as follows [W&H]:

At the heart of the SRAM cell is a dual-inverter latch. Each end of the latch holds a value that's the complement of the other side. The values held at each end of the cell are tapped into by means of a bitline (and its complement, ~bitline). These will be henceforth referred to as BL and ~BL respectively. The transfer of the values of the cell to the bitline are controlled by means of pass transistors at either end, which are in turn turned on by the wordline (WL).

There are different BL and ~BL conditions that represent the process of reading and writing into the cell. These will be discussed in a little more detail in the testing/simulation section of this report.

Functional blocks

SRAM single cell

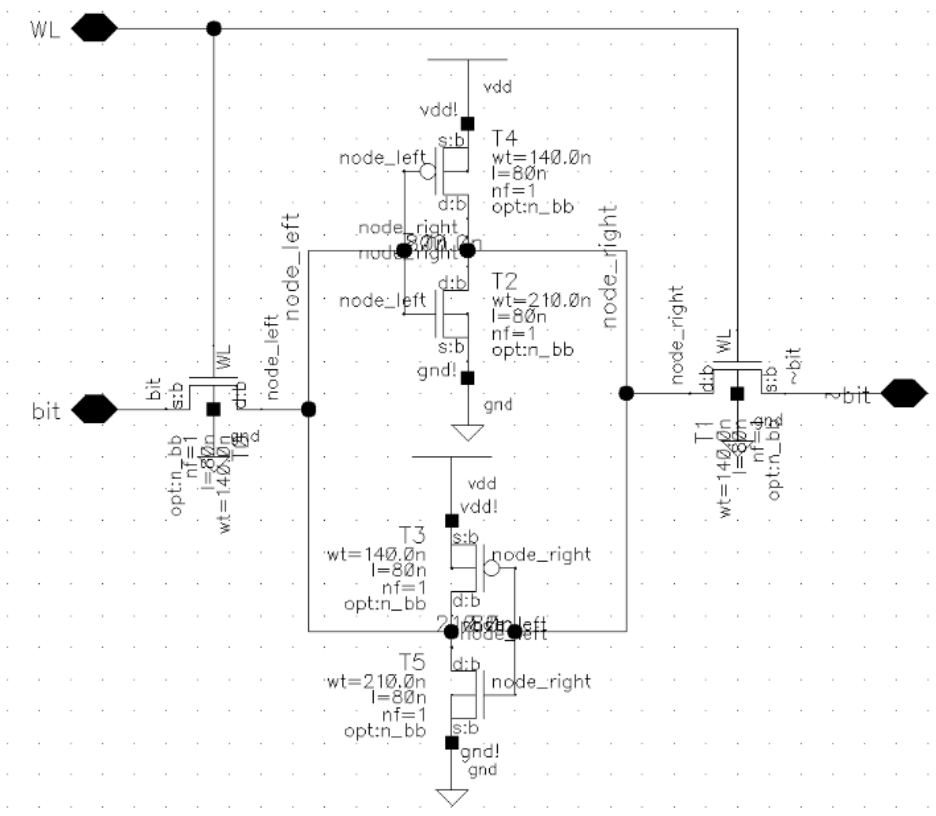


Figure 3 - SRAM single cell schematic

The schematic for the SRAM is shown on the left, and the symbol is shown on the right.

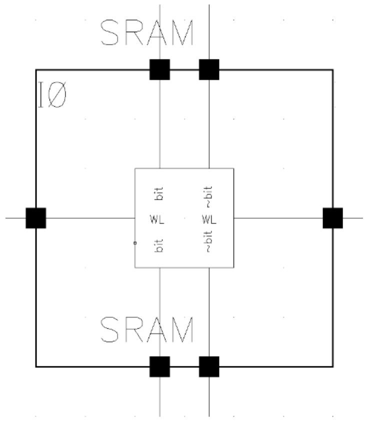


Figure 2 - SRAM single cell symbol

The pre-made layout of the SRAM cell is shown below. It is obvious that the design has undergone several iterations and been optimized to the extreme.

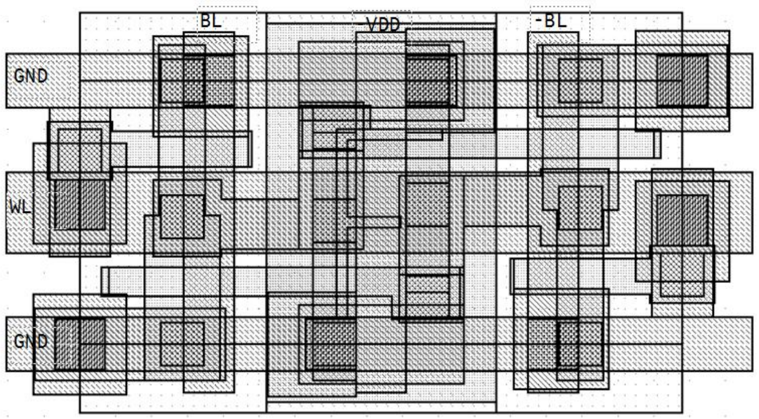


Figure 4 - SRAM single cell layout

First, the SRAM cell was characterized to determine its performance and requirements. The ideal SRAM voltage transfer characteristic (VTC, or the Butterfly curve) is [Uyemura] shown below.

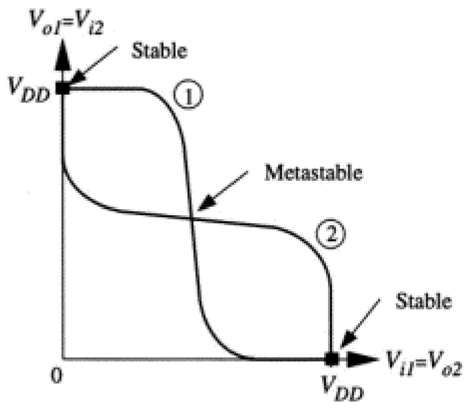


Figure 5 - Ideal SRAM curve [Uyemura]

A similar curve was obtained in the test circuit, as well.

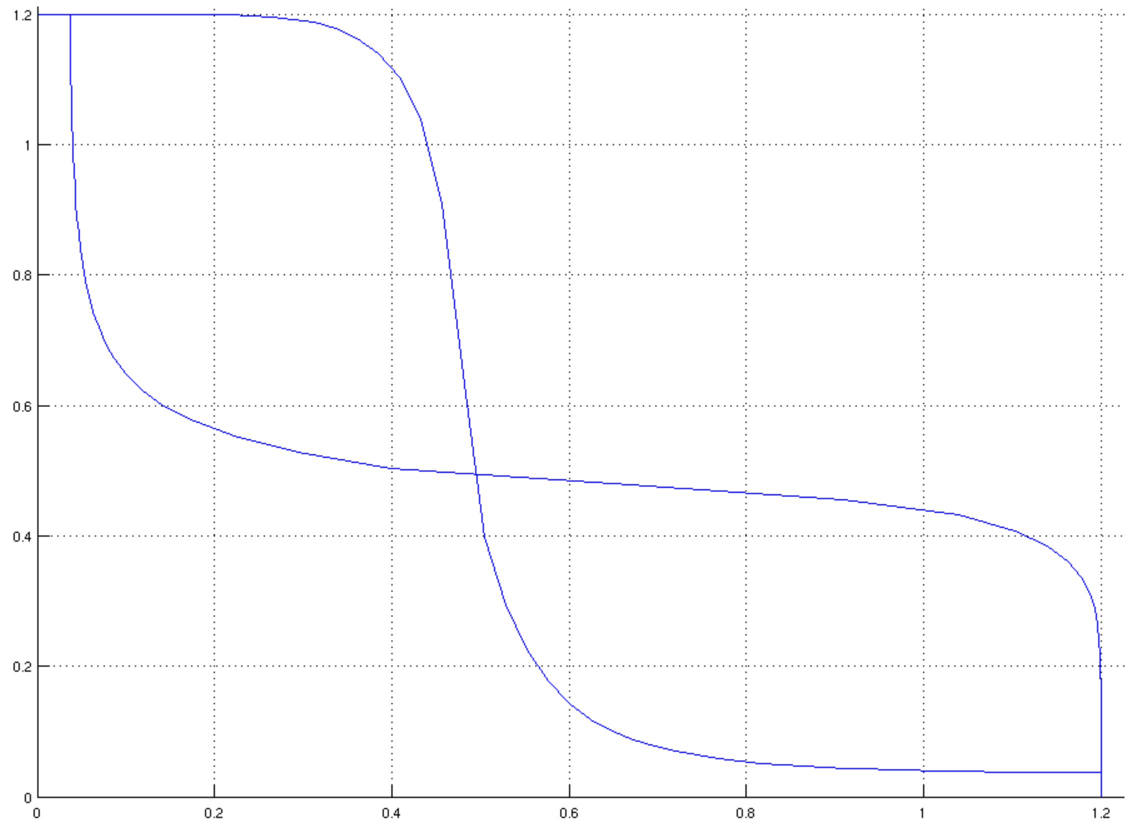
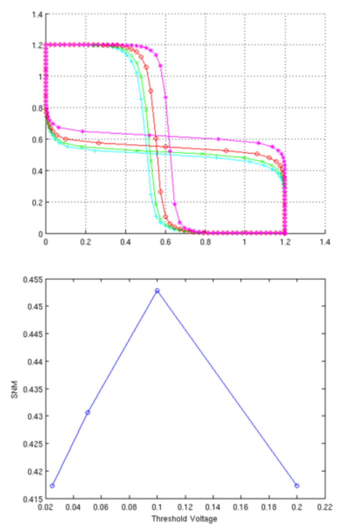


Figure 6 - SRAM VTC

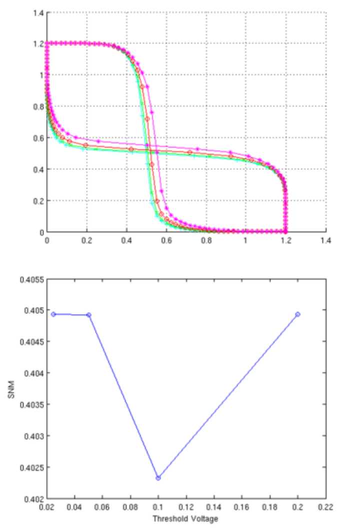
The Static Noise Margin was measured to be around 0.44 V.

In addition, the threshold voltage (VT) of several different transistors was changed (by varying the gate voltages), and their effect on the SNM was studied:

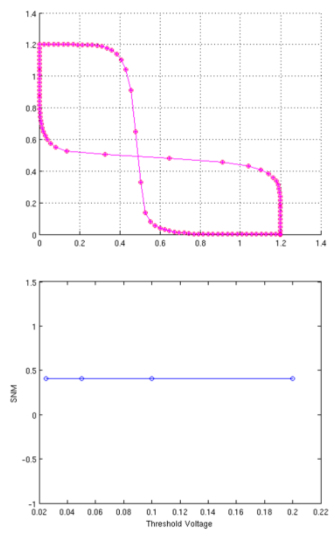
BL and ~BL at ground, testing pull-down



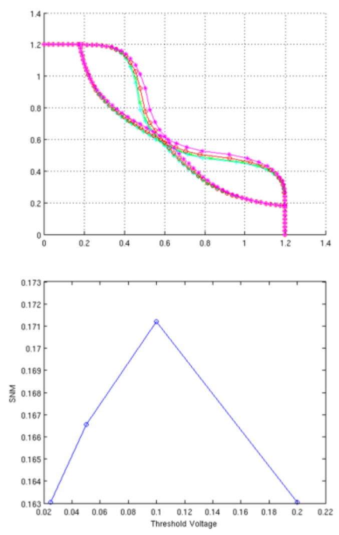
BL and ~BL at ground, testing pull-up



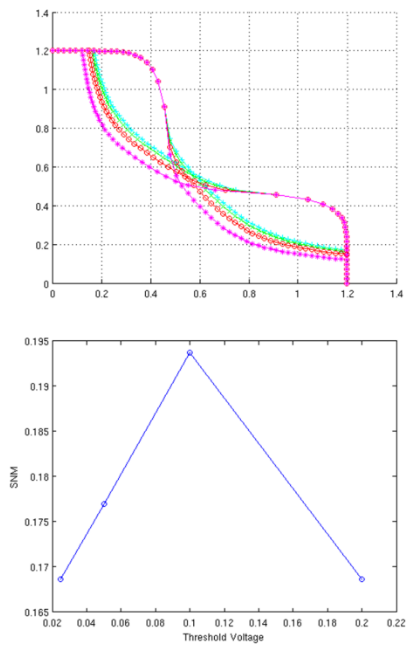
BL and ~BL at VDD, testing pull-down



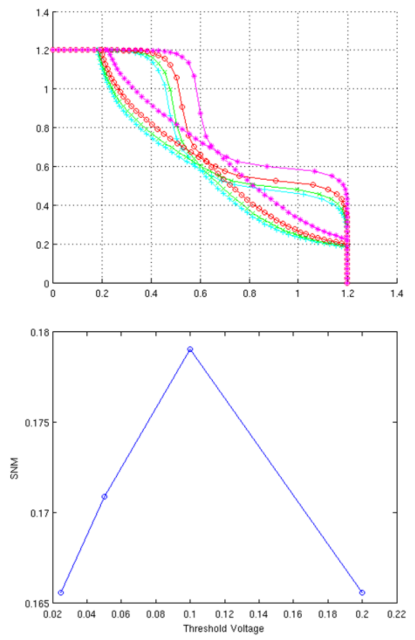
BL and ~BL at VDD, testing pull-up



BL and ~BL at ground, testing pass FET



BL and ~BL at VDD, testing pass FET



The single SRAM cell was then written to and read from. The waveform of the output is shown below:

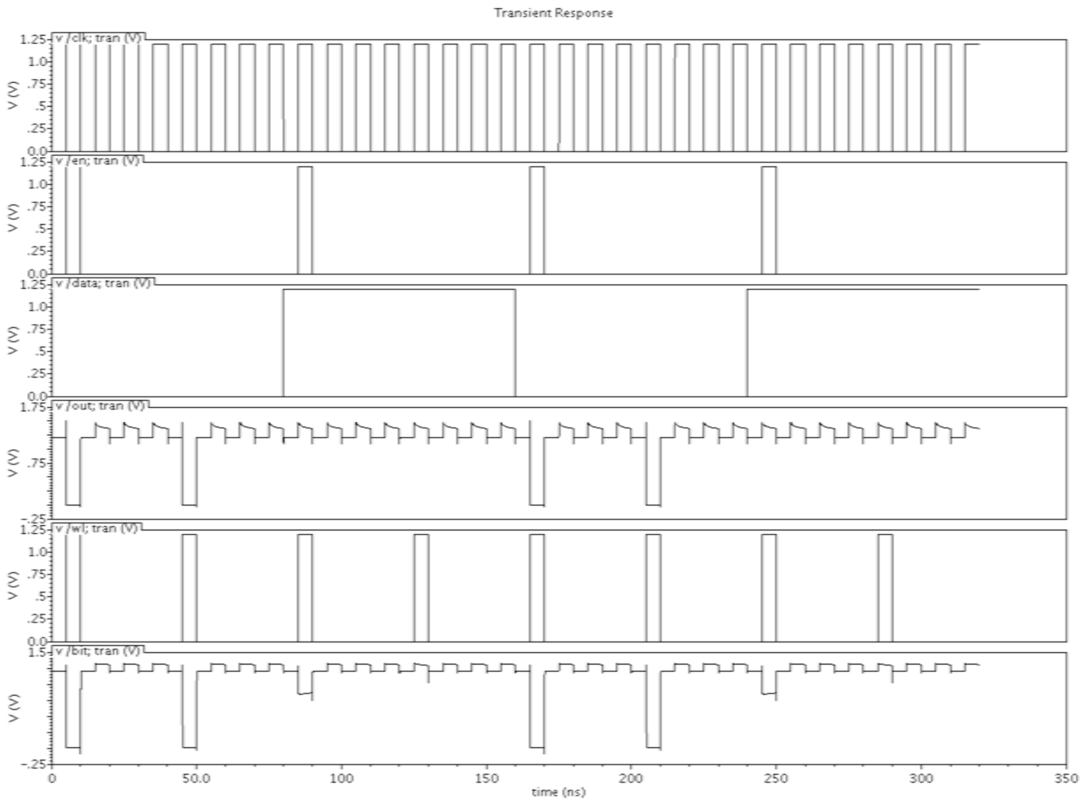


Figure 7 - SRAM read and write waveform

SRAM Array

Optimizing the packing density of the SRAM requires some overlap between cells. In order to accomplish the overlap accurately, a small pack of four SRAM cells was made (4cell). The 4cell has an overlap between ground lines of adjacent cells, WL, BL, and ~BL.

The 4cell of SRAM cells is shown below:

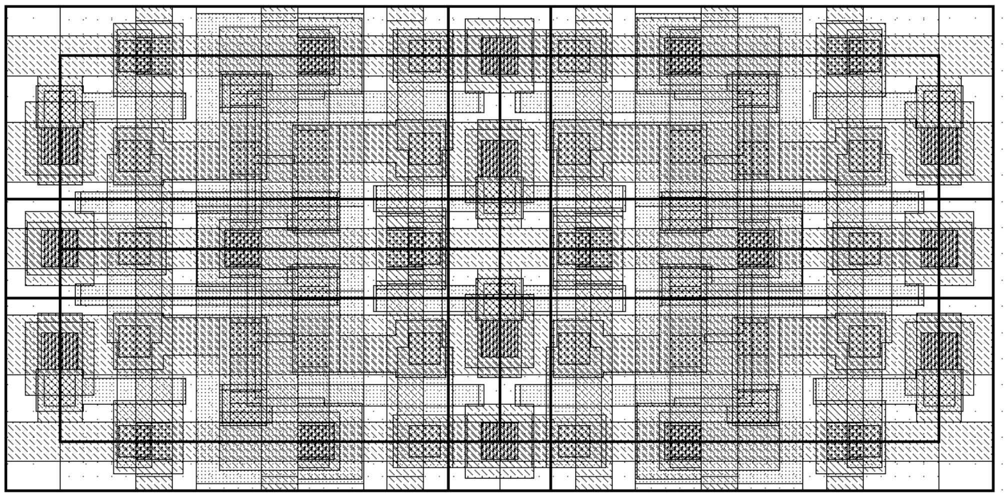


Figure 8 - SRAM 4cell showing high packing density

Once a 4cell is made, replicating it in layout is not particularly hard, once the exact overlap distances are figured out.

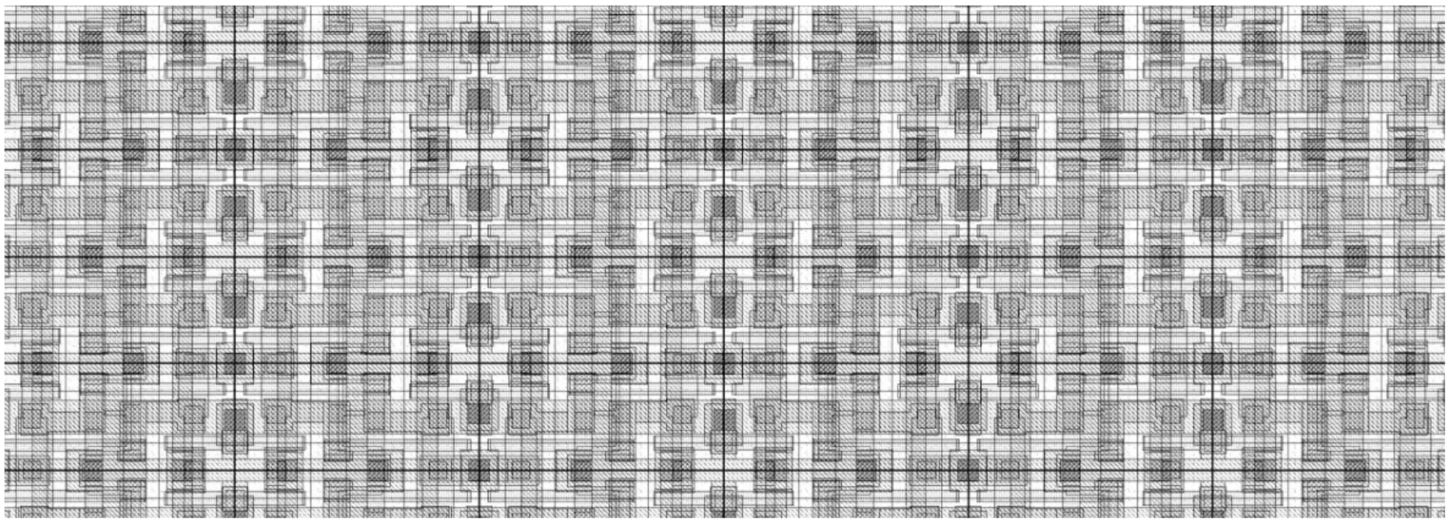


Figure 9 - An array of 16 SRAM cells illustrating overlap

This method of packing ensures that no wires need be used to connect the inputs and outputs of each individual SRAM cell. Of course, wires will be needed at the periphery of the array to connect to the precharge, sense amplifier, and the wordline decoder.

Precharge

The surrounding circuitry has more leeway in terms of spatial constraints. The first of this surrounding circuitry is the precharge circuitry. The precharge circuit serves to refresh or precharge the value held in the SRAM cell every clock cycle. This is achieved by raising both BL and ~BL to VDD during the low edge of every clock cycle. The PFETs on either side are the main functioning units of the precharge circuit, while the pFET on the bottom serves as an equalizer PFET by compensating for non-idealities between the other PFETs [Weste & Harris].

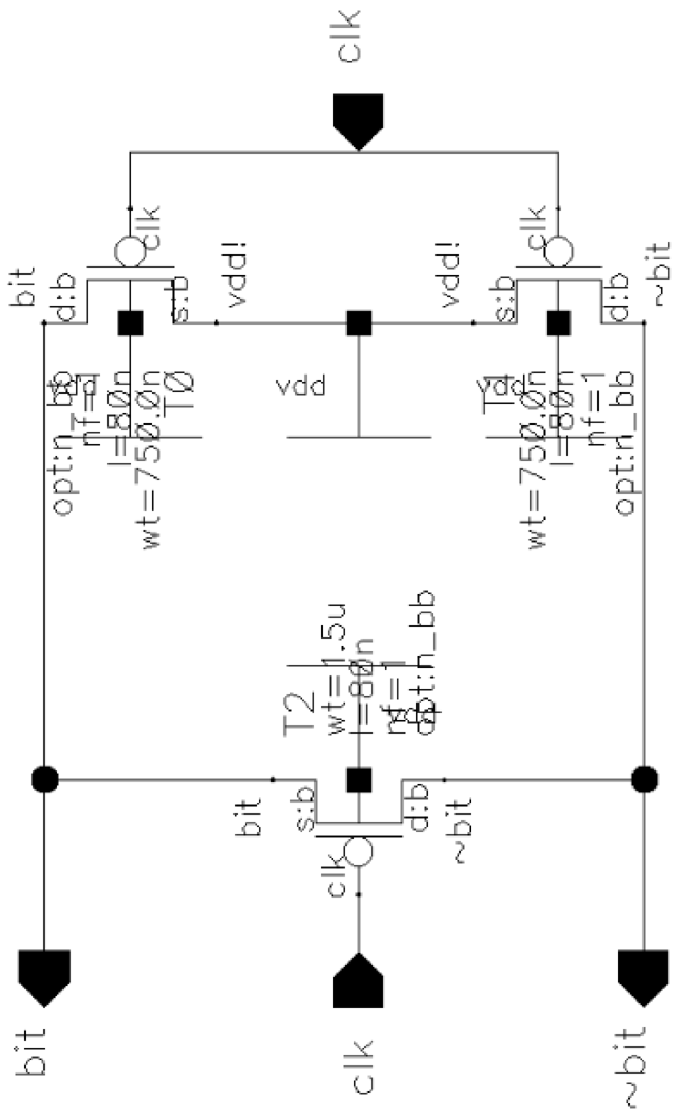


Figure 10 - Precharge schematic

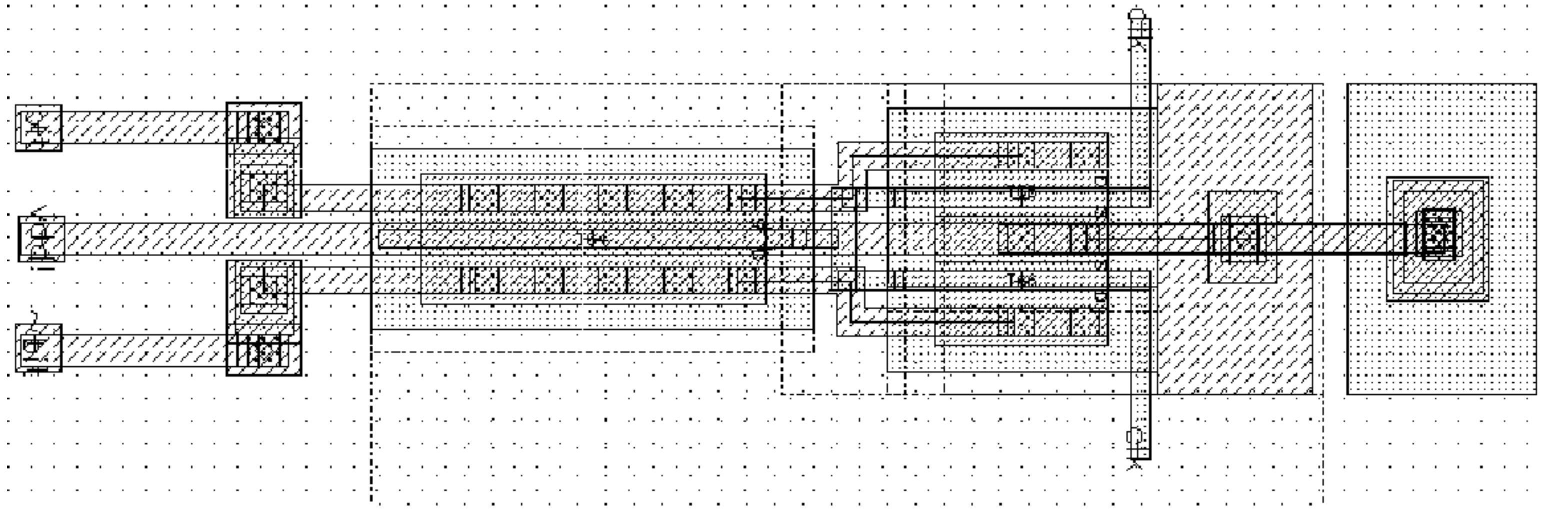


Figure 11 - Precharge layout

The array of precharge cells on the top of the array is shown below:

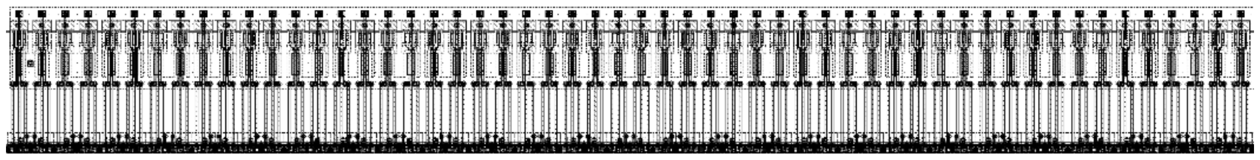


Figure 12 - Part of an array of precharge cells

Sense Amplifier

The large signal sense amplifier was chosen over the differential sense amplifier simply because it has been optimized to a form that uses up the least number of transistors while still being reliable at high speeds. It is also much simpler than the small-signal (differential sensing) circuit.

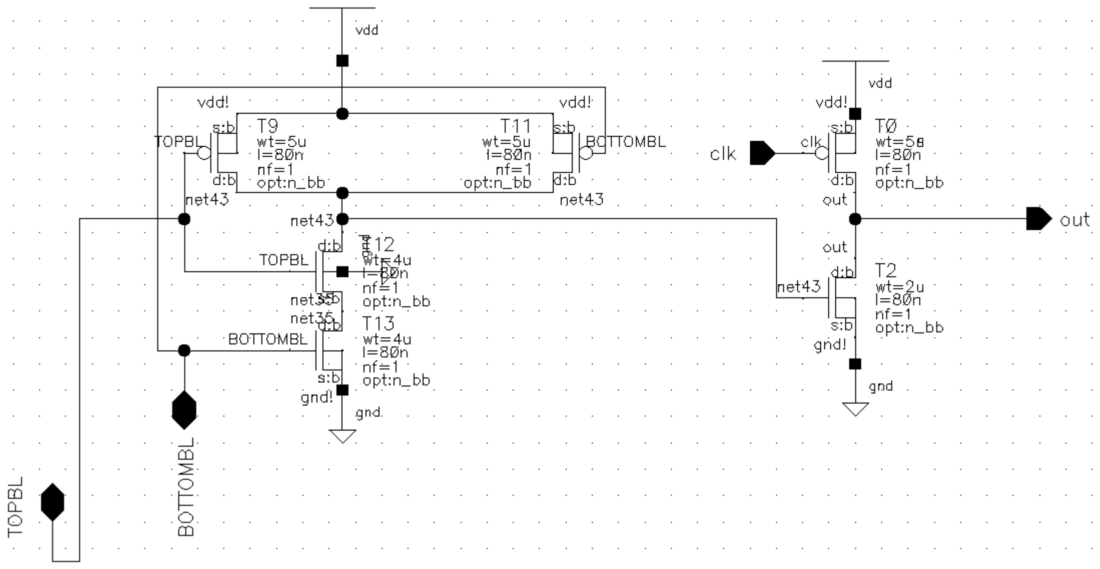


Figure 13 - Sense amplifier schematic

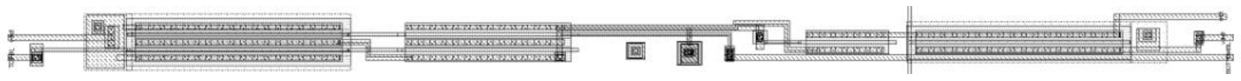


Figure 14 - Sense amplifier layout

Since the SRAM cell was flipped inside the 4cell, the sense amplifier also had to be flipped. This is shown as a pair of sense amplifiers, below.

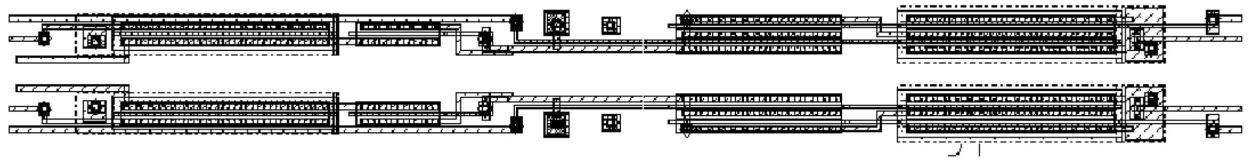


Figure 15 - A pair of mirrored sense amplifiers in layout

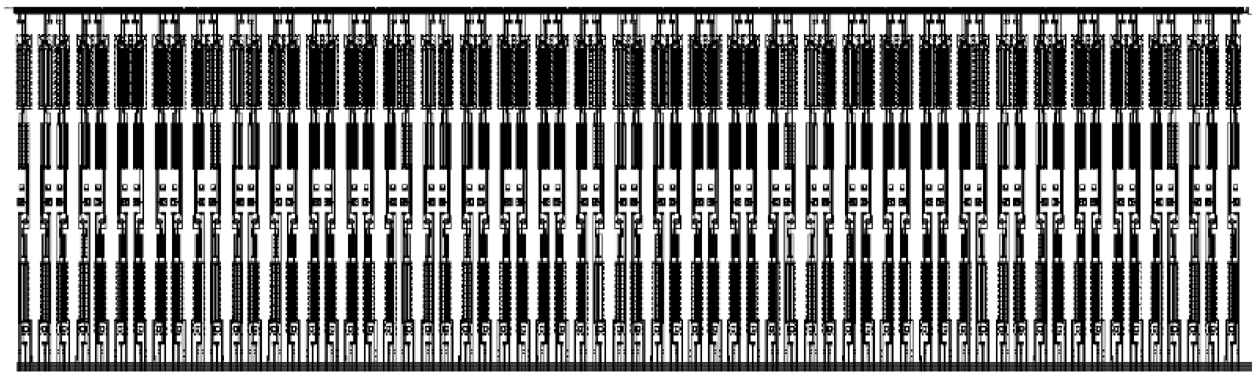


Figure 16 - Sense amplifier array layout

Decoder

This is the only part of the circuit where there truly were chances to make design decisions. Dynamic logic was picked over static logic, since this would make the circuit faster. More specifically, domino logic was used. Keeper PFETs were considered, but the added complexity did not translate well into layout, and so the idea was abandoned. From the input of the decoder to the output, there were a total of four stages (3NAND + INV connected to 2NAND + INV). They were sized according to static sizing rules (since there are no specific sizing rules for dynamic/domino logic).

However, since we know that in dynamic logic, the discharge from the FET capacitances cause current to add up as you go lower in the pull-down network, lower NFETs have to be made larger than ones closer to the output. Through simulations, the ratio was found to be 1.3. Thus, if the NFET closest to the output is sized at X micron, the NFET immediately below it is sized at 1.3*X micron.

An analysis was done on the expected capacitance on one single WL from the layout. This was estimated to be 37 fF. Logic Effort was then used to calculate the sizes of each individual stage in the critical path in the decoder. A 3-to-8 decoder was designed in lieu of multiple 2-to-4 decoders for two reasons:

1. This eliminates the possibility of having an asymmetric set of critical paths

2. This reduces the number of stages, therefore easing implementation

The 3AND and 2AND gates are shown below:

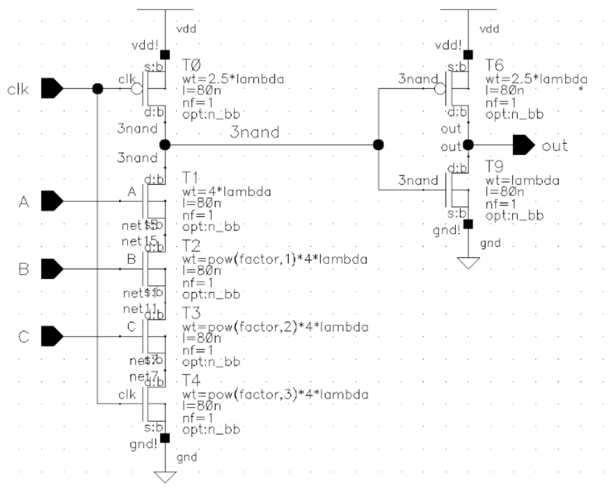


Figure 17 - 3AND schematic

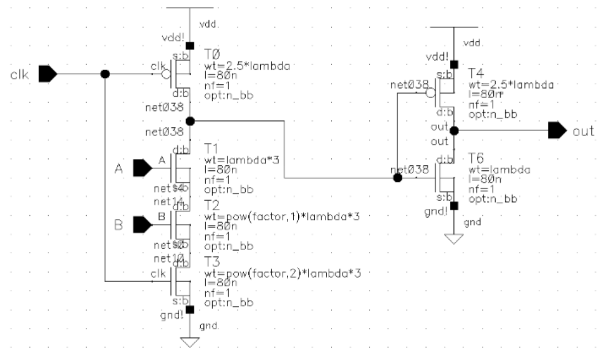


Figure 18 - 2AND schematic

The delay of the decoder (measured as the time difference between 50% of the clock edge and 50% of the output) is around 70 ps. This means that the decoder's theoretical maximum frequency is 14 GHz. Of course, there may be other bottlenecks in the circuit (including the SRAM itself).

To match exactly with the SRAM array (and minimize wasted space in the array), the last stage of the decoder (the 2AND) has to be "pitch-matched" with the SRAM cell. This means that the 2AND has to fit exactly within the dimensions of an overlapped SRAM cell. This particular task was quite difficult. The final result of the 2AND pitch-match is shown below. The SRAM array is seen as the black mass on the right-edge of the figure.

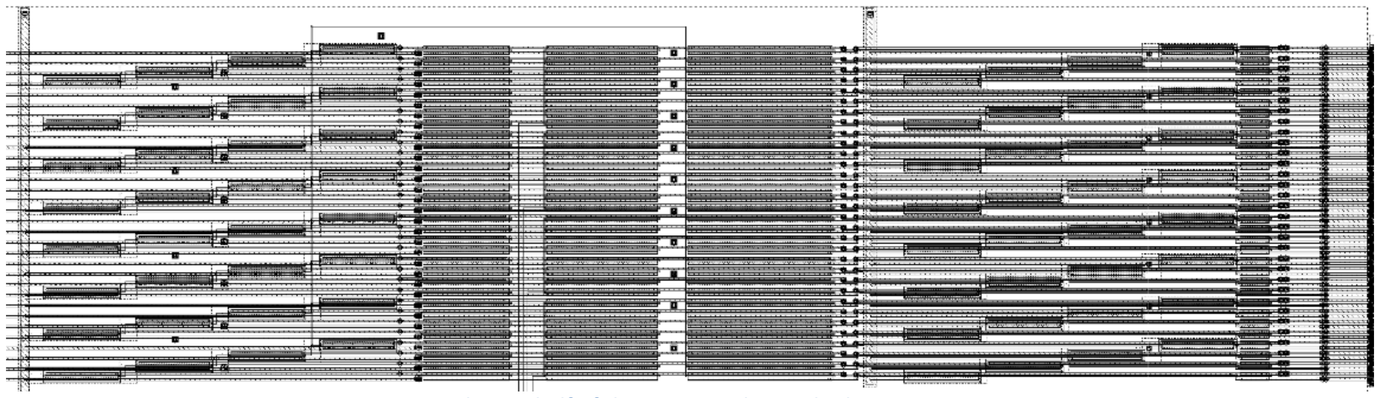


Figure 19 - The top half of the 2AND pitch-matched array

Putting it together

The complete layout is shown below. Prof. Bhavnagarwala suggested that we split the SRAM array into two 32x64 sections to reduce the delay and capacitances between the cells and the read circuitry, and also because a large signal sensing circuit could be used.

The 3-to-8 decoders are on the bottom. They could have been packed tighter and placed in-between the pitch-matched last stages of the decoder, but this was not done due to time-constraints, and because it would have made the gap between the SRAM arrays larger (there’s only so much optimization that can be done on the 3-to-8 decoder).

The complete schematic, illustrating the SRAM array is shown on the following page.

Conclusion

Designing an SRAM array is not an easy task. The SRAM cell itself is so optimized that it is a tragedy to place circuits around it that aren’t as optimized. Unfortunately, optimization (spatial and performace) is a time-consuming task, and is a process that demonstrates diminishing returns. It was definitely a learning experience, especially with Cadence, which is the Microsoft of CAD tools.

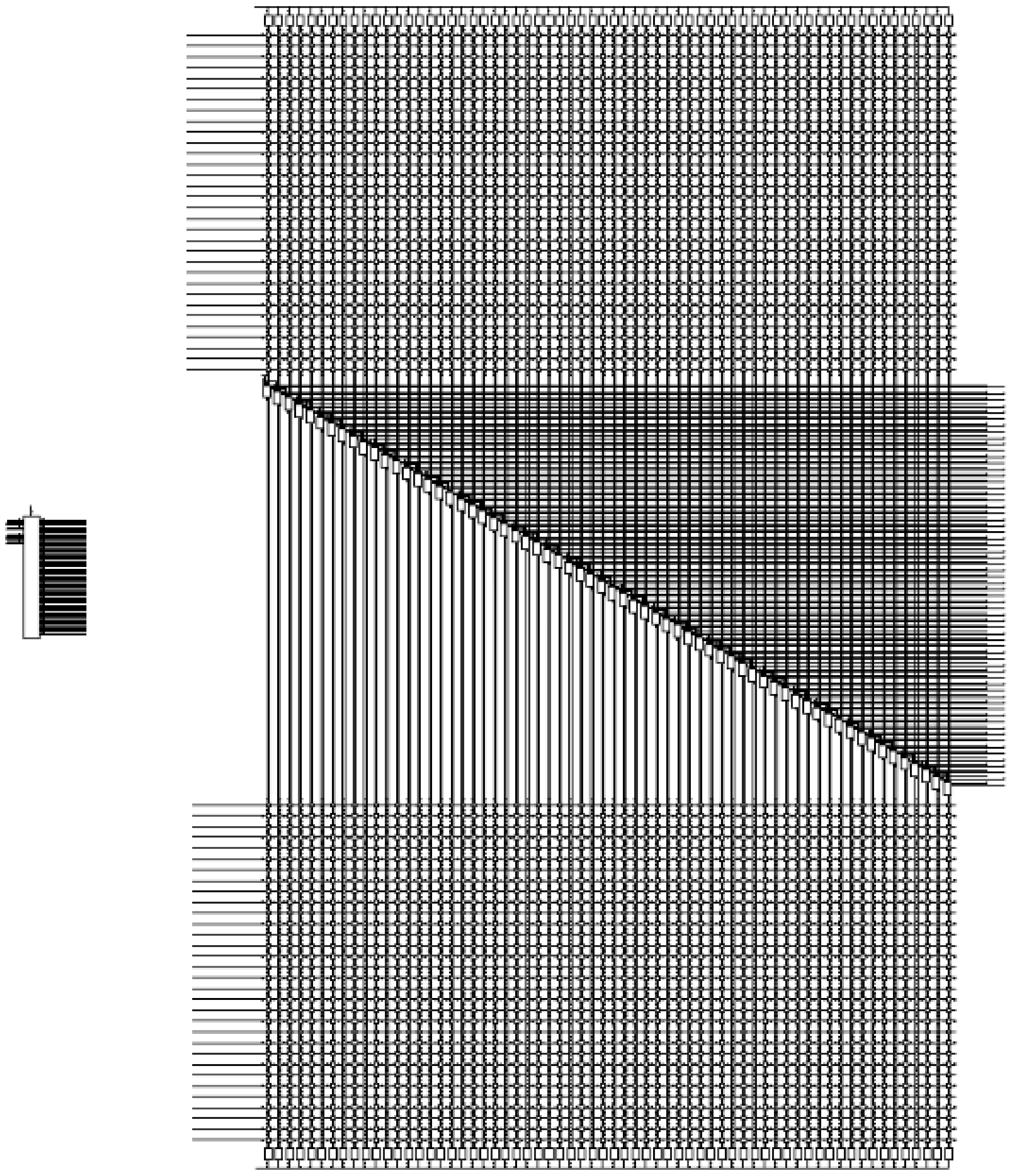


Figure 20 - Complete schematic

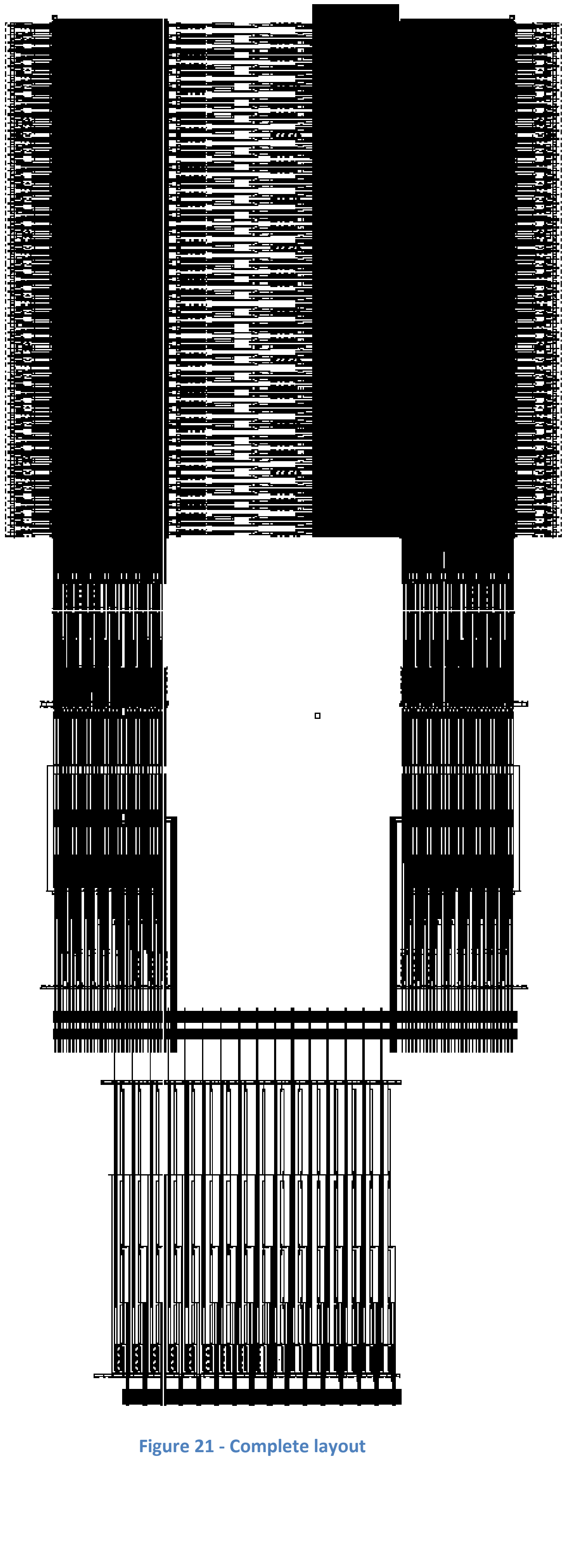


Figure 21 - Complete layout

DRC and LVS

Extraction could not be done, because Calibre has not been set up to be compatible with the 90nm XRC files.

The DRC and LVS results are shown below:

Precharge circuit:

```
--------------------------------------------------------------------------------

--- RULECHECK RESULTS STATISTICS (BY CELL)

---

CELL PreCharge .................. TOTAL Result Count = 1 (1)

    RULECHECK GRESD01_warning ... TOTAL Result Count = 1 (1)

--------------------------------------------------------------------------------

                  ##################################################

                  ##                                              ##

                  ##           C A L I B R E    S Y S T E M       ##

                  ##                                              ##

                  ##               L V S   R E P O R T            ##

                  ##                                              ##

                  ##################################################




REPORT FILE NAME:          PreCharge.lvs.report

LAYOUT NAME:               PreCharge.lay.net ('PreCharge')

SOURCE NAME:               PreCharge.src.net ('PreCharge')

RULE FILE:                 /tmp/tmpb5/_cmos9sf.lvs.cal_

HCELL FILE:                (-automatch)

CREATION TIME:             Tue Dec 11 19:05:49 2007

CURRENT DIRECTORY:         /tmp/tmpb5

USER NAME:                 sb2784

CALIBRE VERSION:           v2004.3_9.21    Thu Sep 30 11:25:17 PDT 2004
```

```
                    OVERALL COMPARISON RESULTS




          #       ###################      _   _
         #        #                 #       *   *
      #   #       #     CORRECT     #         |
       # #        #                 #       \___/
        #         ###################
```

```
Sense amplifier:

--------------------------------------------------------------------------------
--- RULECHECK RESULTS STATISTICS (BY CELL)
---
CELL Sense ......................... TOTAL Result Count = 1 (1)
    RULECHECK GRESD01_warning ....... TOTAL Result Count = 1 (1)
--------------------------------------------------------------------------------
                  ##################################################
                  ##                                              ##
                  ##          C A L I B R E    S Y S T E M        ##
                  ##                                              ##
                  ##              L V S   R E P O R T             ##
                  ##                                              ##
                  ##################################################



REPORT FILE NAME:         Sense.lvs.report
LAYOUT NAME:              Sense.lay.net ('Sense')
SOURCE NAME:              Sense.src.net ('Sense')
RULE FILE:                /tmp/tmpb5/_cmos9sf.lvs.cal_
```

```
HCELL FILE:                (-automatch)

CREATION TIME:             Tue Dec 11 19:08:31 2007

CURRENT DIRECTORY:         /tmp/tmpb5

USER NAME:                 sb2784

CALIBRE VERSION:           v2004.3_9.21    Thu Sep 30 11:25:17 PDT 2004




                                OVERALL COMPARISON RESULTS




                         #       ###################      _   _

                        #        #                 #      *   *

                   #   #         #     CORRECT     #        |

                    # #          #                 #      \___/

                     #           ###################
```

```
Decoder:

----------------------------------------------------------------------------------

--- RULECHECK RESULTS STATISTICS (BY CELL)

---

CELL SRAM_pitch_matched_decoder ... TOTAL Result Count = 1 (1)

    RULECHECK GRESD01_warning ..... TOTAL Result Count = 1 (1)

----------------------------------------------------------------------------------

                  ##################################################

                  ##                                              ##

                  ##          C A L I B R E    S Y S T E M        ##

                  ##                                              ##

                  ##              L V S   R E P O R T             ##

                  ##                                              ##

                  ##################################################
```

```
REPORT FILE NAME:          SRAM_pitch_matched_decoder.lvs.report

LAYOUT NAME:               SRAM_pitch_matched_decoder.lay.net ('SRAM_pitch_matched_decoder')

SOURCE NAME:               SRAM_pitch_matched_decoder.src.net ('SRAM_pitch_matched_decoder')

RULE FILE:                 /tmp/tmpb5/_cmos9sf.lvs.cal_

HCELL FILE:                (-automatch)

CREATION TIME:             Tue Dec 11 19:12:31 2007

CURRENT DIRECTORY:         /tmp/tmpb5

USER NAME:                 sb2784

CALIBRE VERSION:           v2004.3_9.21    Thu Sep 30 11:25:17 PDT 2004



                              OVERALL COMPARISON RESULTS



                         #       ###################       _   _
                        #        #                 #       *   *
                   #   #         #     CORRECT     #         |
                    # #          #                 #       \___/
                     #           ###################
```

The SRAM array, obviously, won’t pass DRC.

The paths to the files are:

/home/user5/fall07/sb2784/Desktop/4321/project/4321projectLib

/home/user5/fall07/sb2784/Desktop/4321/project/4321projectLib_jp