**EE313 Project Part II – Progress Report**

Please turn in your Progress Report in the format specified below. Please try to keep the main part of the report to be 6 or fewer pages. Avoid lengthy descriptions. If you need any extra pages to explain your analysis or to show some schematics, please attach them at the back of the report. In order for us to know the distributions of modifications/improvements/performance being planned/executed, we may ask you to fill out a simple on-line survey form, so please stay tuned.

Start with:

**Group Members:**

**Andrew Dupree**

**Victoria Harvey**

**Reed Nightingale**

**Overall Design Objectives (low power, high speed, robustness, high throughput):**

The main objective of the design is low power which will be implemented by blocking the wordline, reducing the bitline capacitance, reducing the supply voltage, and changing the topology and sizing of the decoder.

Since blocking the wordline and reducing the bitline capacitance will decrease the overall capacitance within the system, these improvements will help increase speed and throughput.

These refer to the following suggested improvements:

* Category 1: 1.b
* Category 1:2
* Category 2:3
* Category 5:1
* Category 5:2

1. Discuss the modifications that you are planning to implement in the memory system. Information that you will need to provide may include **any** of the following:

a. What are you planning to do to improve the decoder design? How do you think your design change will help your memory in its overall performance?

To improve the decoder design we are planning on blocking the wordline, reducing the bitline capacitance, reducing the supply voltage, and changing the topology and sizing of the decoder. Blocking the wordline will reduce the overall capacitance seen by the wordline when it flips so this will help in both power and delay in the memory design. Reducing the bitline capacitance will improve both the delay the power. Reducing the supply voltage reduce the overall power of the system. Changing the topology and size of the decoder will improve both power and speed.

b. How do plan to modify/redesign your sense amplifier? Are you going to use a static sense amplifier? Why or why not?

We are not modifying the sense amplifier.

c. What are the issues you will need to deal with in terms of timing? How do you plan to deal with these issues?

The main issue we expect we will run in to is collisions between the precharge circuits and the read/sense circuits. We will handle them by creating delays and offsets such that all signals are well timed.

d. What are the disadvantages/advantages of using divided bitlines or divided wordlines? Do you plan to implement either of these designs? Why or why not?

Dividing the lines means you can reduce your capacitance per read/write and thus reduce power, but at the cost of additional logic. We plan on implementing divided wordlines.

e. How much of an impact on the power/performance of your design do you expect from using a lower supply voltage? How will your performance and robustness be

affected with lower supplies?

We are planning on decreasing our supply voltage as a last optimization step rather than a primary optimization. Therefore, we won’t know how much lower of a supply we can use, and thus cannot say exactly how much power savings we will get.

f. What read/write assist circuits do you plan to implement? How will they affect

the performance/power in normal Vdd operation?

We plan on implementing dual-voltage read and write for robustness. They should tax the power, but should not affect performance much, if at all.

g. What other improvements in the overall system do you plan to make (e.g.,

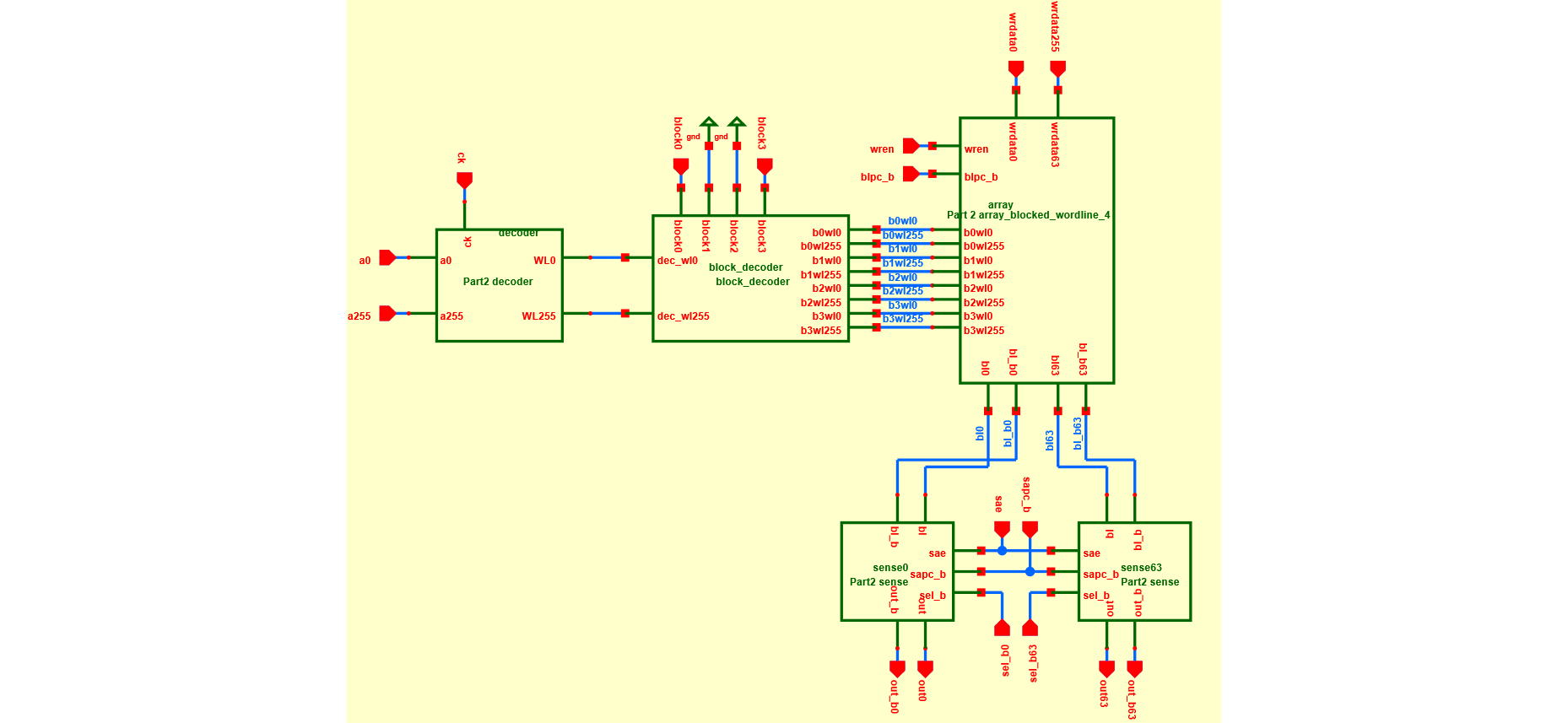
limiting bitline swing, pipelined SRAM, low power writes etc.)?

We considering inputting the clock at the decoder stage rather than predecoder. While this should significantly reduce the pre-decoder power

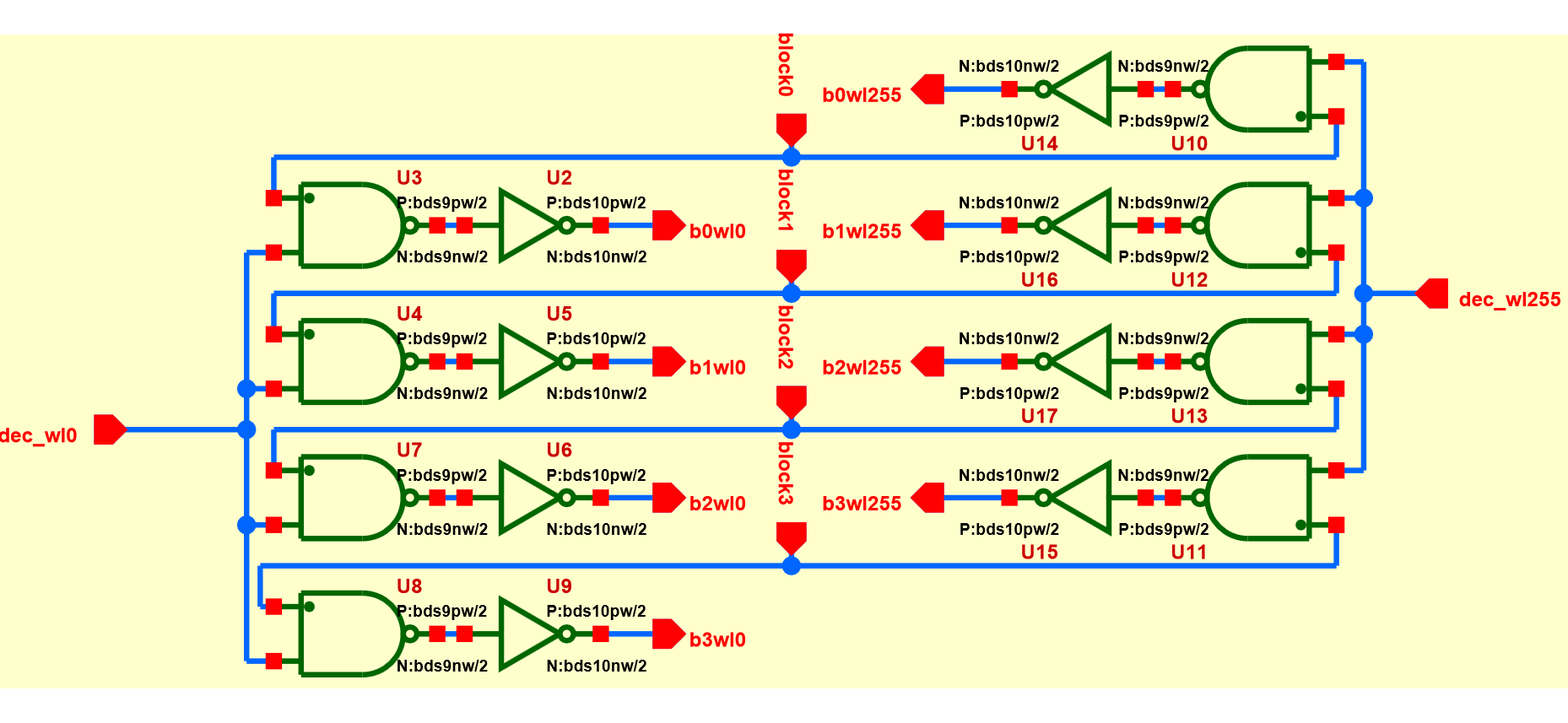
2. If you are going to use a clocked sense amplifier, give a complete description of how you have designed the sense amplifier enable timing generation circuitry. What have you done to ensure the memory will work even with timing variations in this circuit? Please include the following:

We currently have the sense amplifier running off an ideal pulse, and are checking two versions of the spice deck, one with a 30% timing offset, the other with the nominal timing, to verify proper operation.

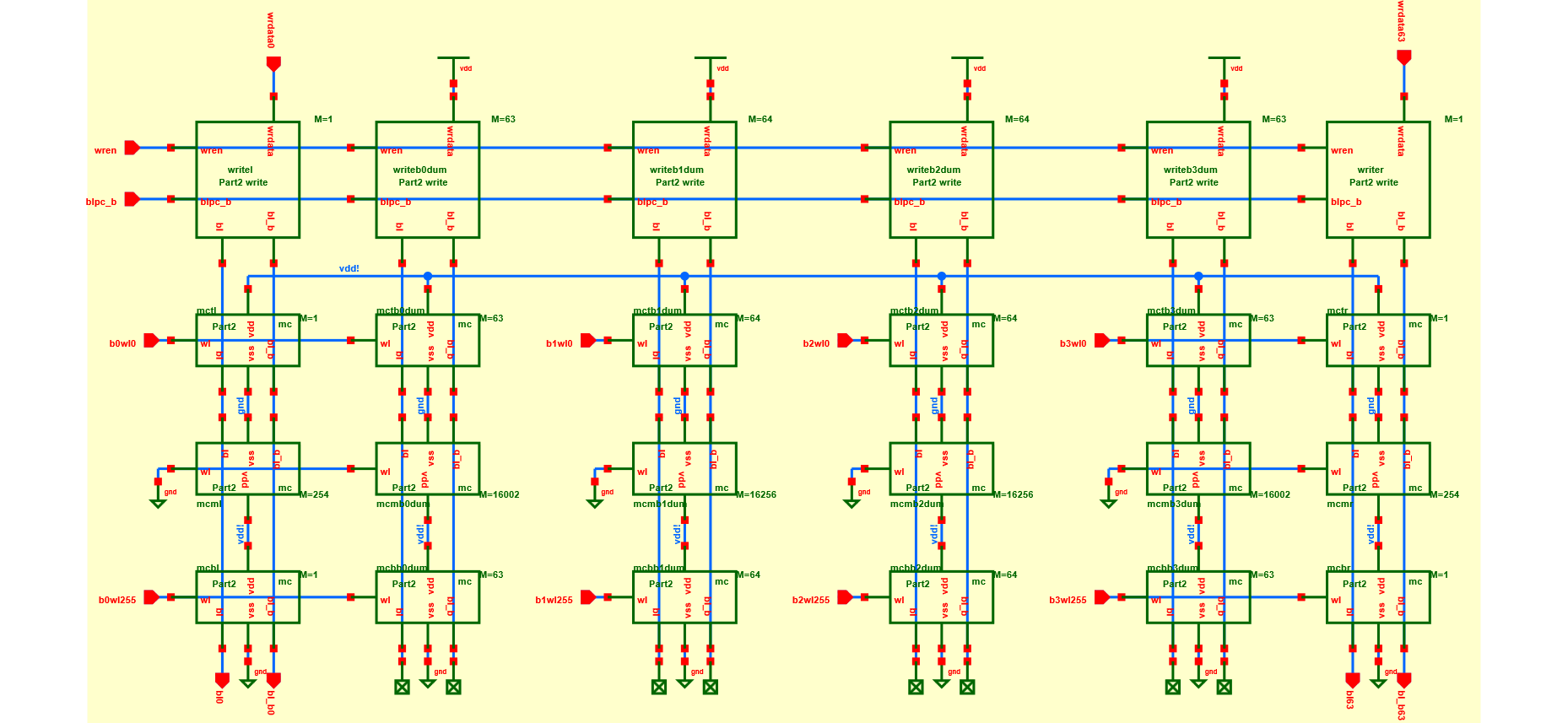
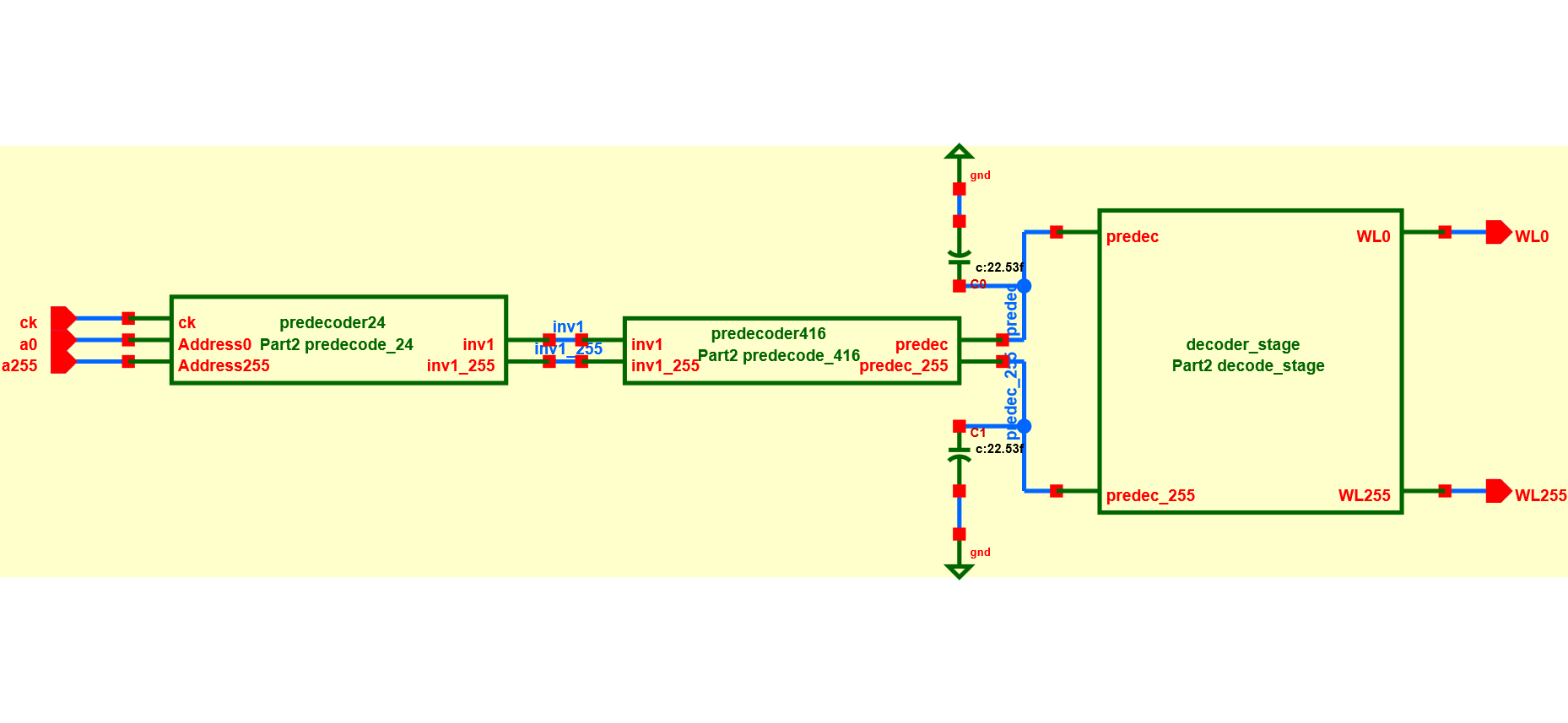
a. Top level, decoder and dummy path schematics.



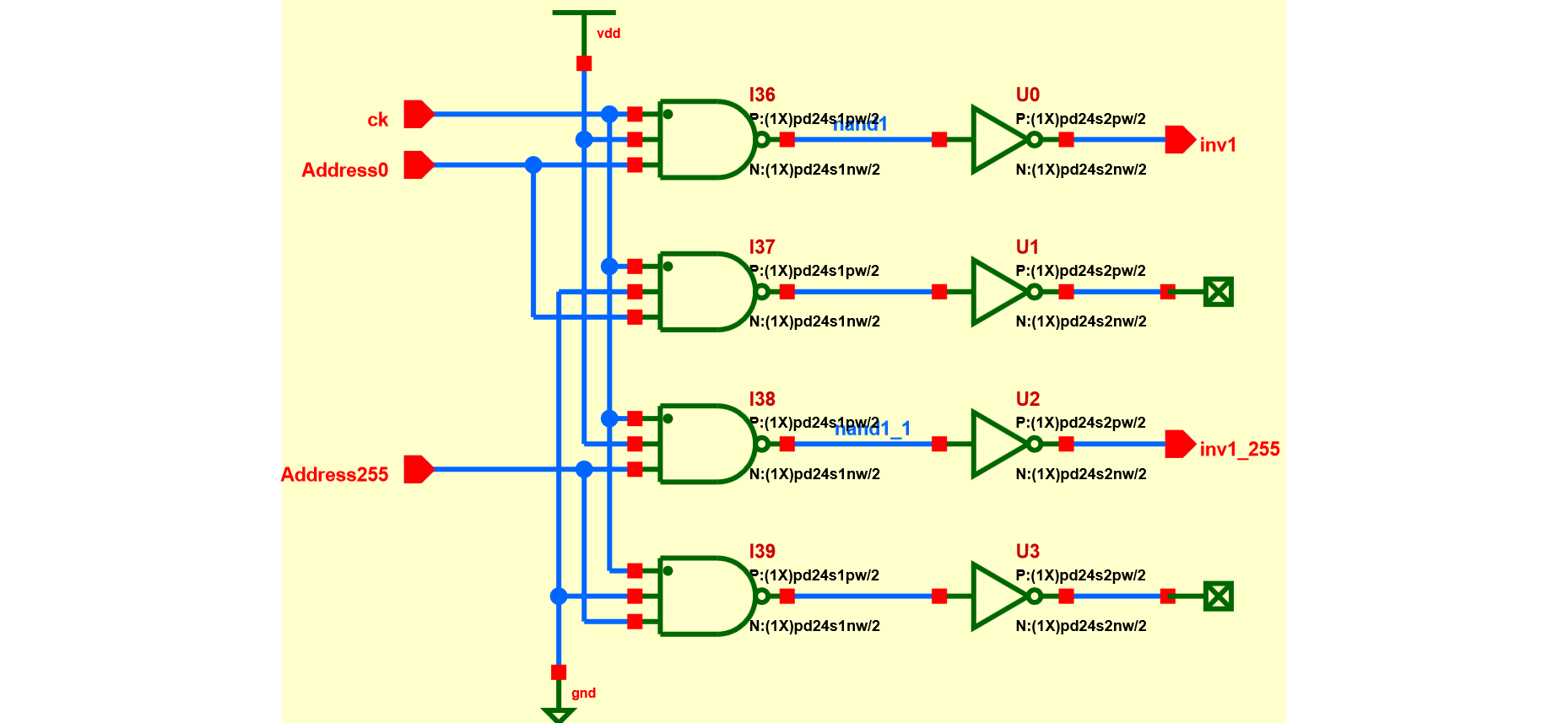
Top Level



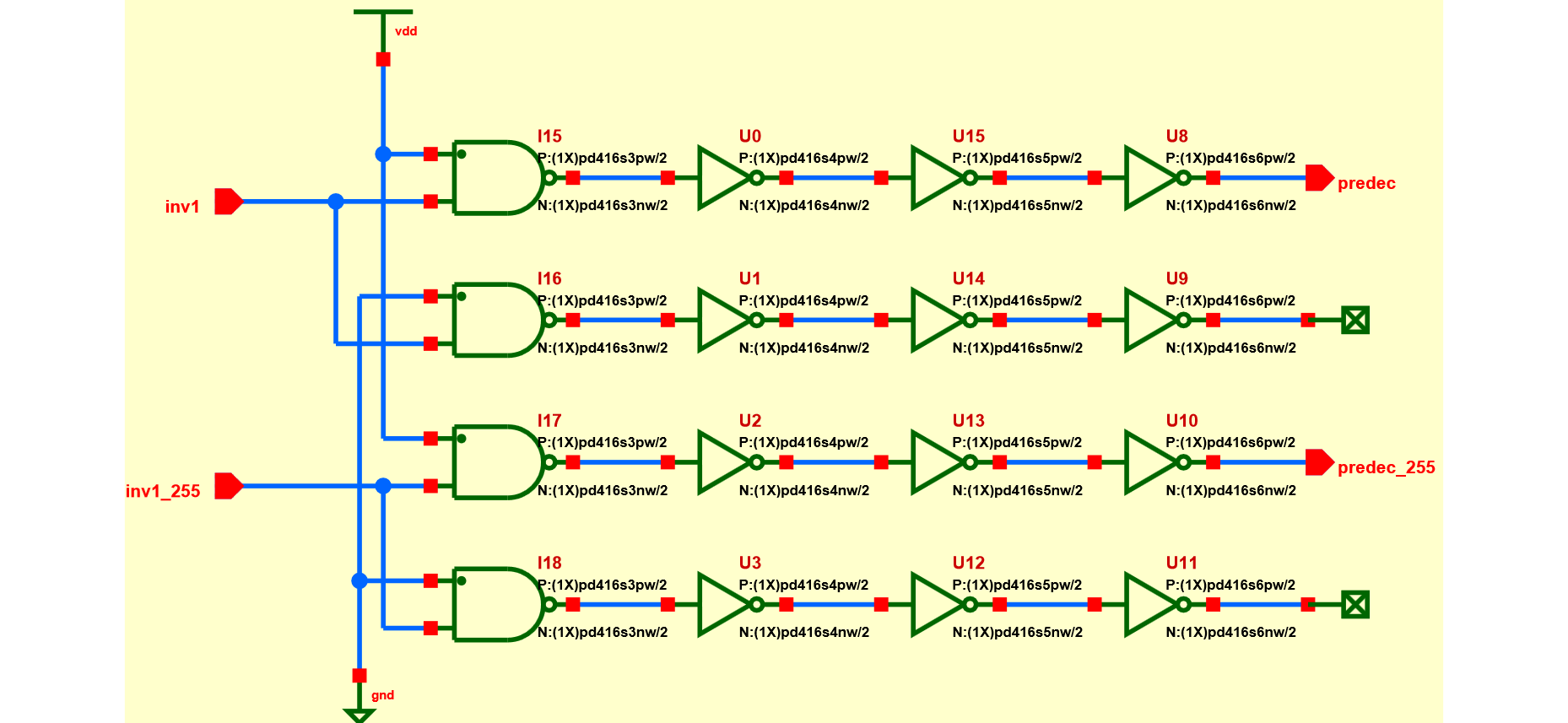
Block Decoder

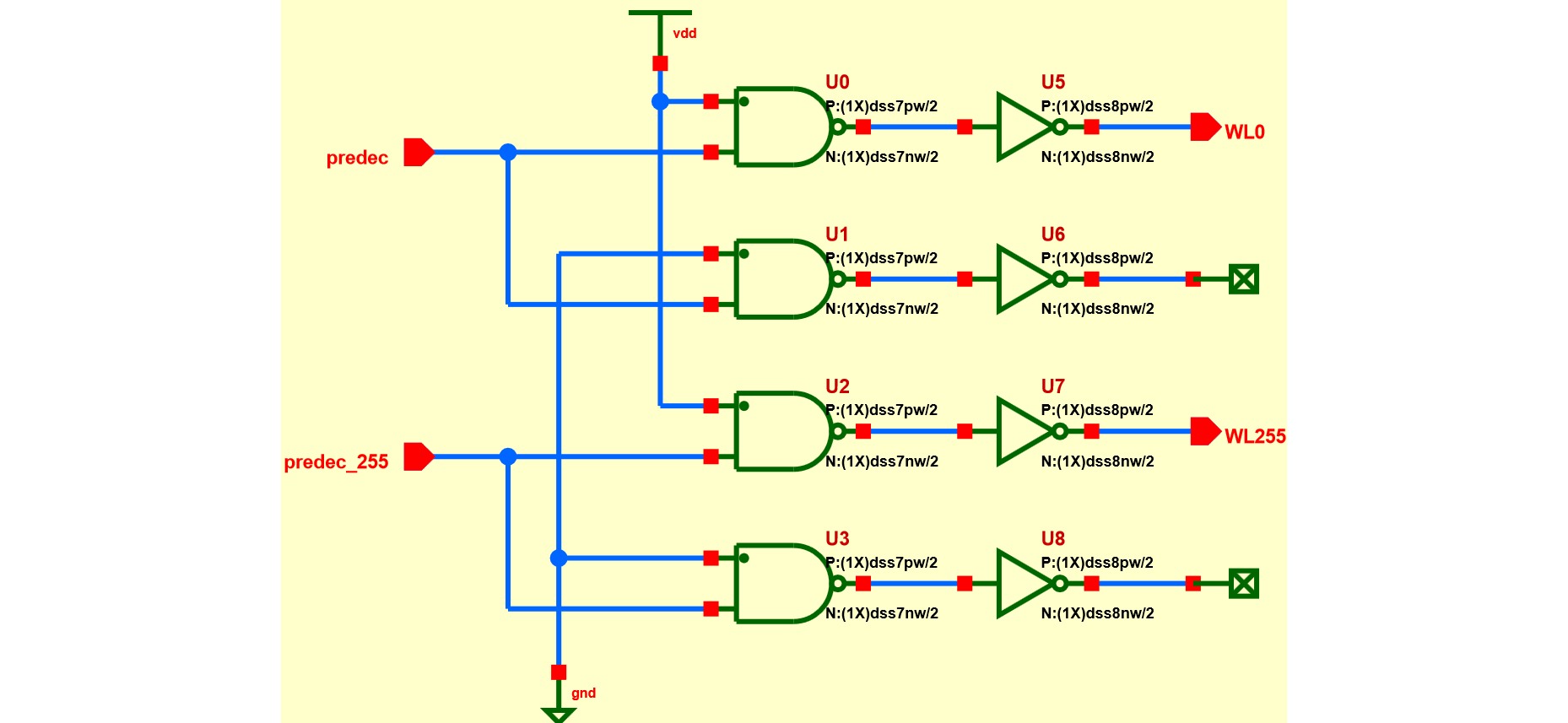
Memory Array Row Decoder



Pre-Decoder 2-4

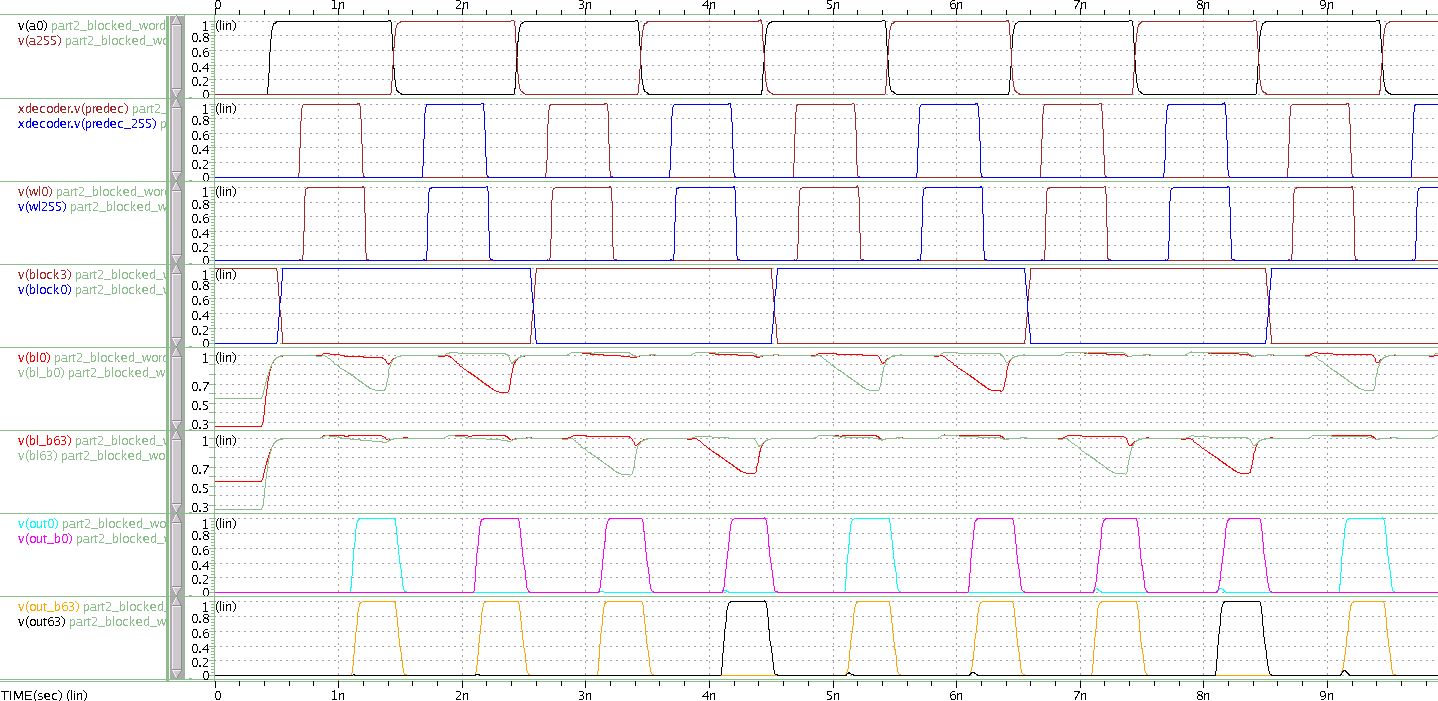


Pre-Decoder 4-16



Decoder Stage

b. Simulated waveforms for the predecoder, decoder, bitlines, sense amp and dummy bitlines. Measure the delays for the wordline to bitline, wordline to dummy bitline, clk to bitline and clk to SAE.



You may use ideal pulses for sapc\_b and blpc\_b for this progress report, but remember you must generate them from clk for the Final Project Report.

If you plan on changing your timing generation path later, but did not have enough time to complete it for this report, please include a discussion about what you plan to change and why.

We will be generating the various signals for precharging and sensing using inverter chain delays from the ck signal since they are currently ideal pulses. The inverter chains will be sized to match the ideal pulse delays we find give the best margins.

~~For those of you implementing a static sense amplifier, if you are still going to use a clock (to equalize, or save power, or pulse the wordlines), please discuss how you have constructed this clock, and indicate how you have margined it. In addition if you want feedback on your static sense amp design, please discuss how you have designed the static sense amplifier, including as quantitative a discussion of the various tradeoffs that you have taken into account in your design.~~

3. Descriptive analysis of one of the improvements you plan to implement. This must include hand analysis as well as simulations to backup your analysis. You should be as complete as possible.

For wordline blocking, we effectively add two additional logic elements to the decoder chain. We will optimize the gate sizes for the decoder delay optimization part, so for now we consider the non-optimal sizings below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **7 (nand2)** | **8 (inv)** | **9 (nand2)** | **10 (inv)** | **Total** |
| **Cload** | 0 | 0 | 0 | 804.5 | 804.5 |
| **Cg** | 907 | 20 | 32 | 0 | 959 |
| **Cj** | 404.64 | 777.299 | 28.8 | 27.424 | 1238.163 |
| **Total** | 1311.64 | 797.299 | 60.8 | 831.924 | **3001.663λ** |
|  |  |  |  |  | **92.4512204μW** |
| **Power total** | **92.45122** | **μW** |  |  |  |

From Spice simulation, we have a leakage power 2.1861μW and 133.5672μW per pre-decoder toggle.

Total Dynamic Power = (Total Power – Leakage Power) + xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

= (133.5672μW - 2.1861μW) + 8∙2.1861μW

= 148.87μW