EECS 151/251A Homework 11

Due 11:59pm Friday, December 3rd, 2021

1 Memory Implementation

Suppose you want to design a 32-bit wide byte-addressable memory block with a capacity of 2K 32-bit words of storage (1K = 1024). Besides, we would like to make the memory block has a 1:1 aspect ratio (equal number of rows and columns).

a How many rows and columns of SRAM cells are there in your design?

[ ] (row) × [ ] (column)

b How many total address bits do you need for this memory? How many address bits are used by the row-decoder? How many address bits are used by the column-decoder?

It needs [ ] bits address

[ ] bits are used by row-decoder

[ ] bits are used by column-decoder

Solution:

The total number of cells is:

$$2 \times 1024 \times 32 = 65536$$

Since we want a square block, we take the square root of the memory size

$$\sqrt{65536} = 256$$

This means we have to design a 256 × 256 memory core. The total number of address bits required is

$$L = log_2(2 \times 1024 \times 4) = 13 \text{ bits}$$

Each row contains $256/32 = 32$ 8-bit bytes. So the column-decoder needs $K = log2(32) = 5$ bits The row-decoder then requires

$$L - K = 13 - 5 = 8 \text{ bits}$$

2 Building Bigger Blocks

As modern processors can handle more and more computations per second, the associated memories must also hold more and more data to keep up. However, one major problem with making huge SRAM blocks is that the bit lines will become extremely long as the memory increases in capacity, which causes issues with readability and write speed. One way to get around this problem is to make the overall memory block out of smaller sub-blocks.

a For this problem, you only have access to an SRAM block that is 32 words deep with a 16-bit word length. The basic SRAM block has a single read port and a single write port. Suppose you want to design a 32-bit wide memory block with a capacity of 2K 32-bit words of storage, and we will still want to stick to a 1:1 aspect ratio. How many rows and columns of this small SRAM blocks are there in your design?

_____ (row) × _____ (column)

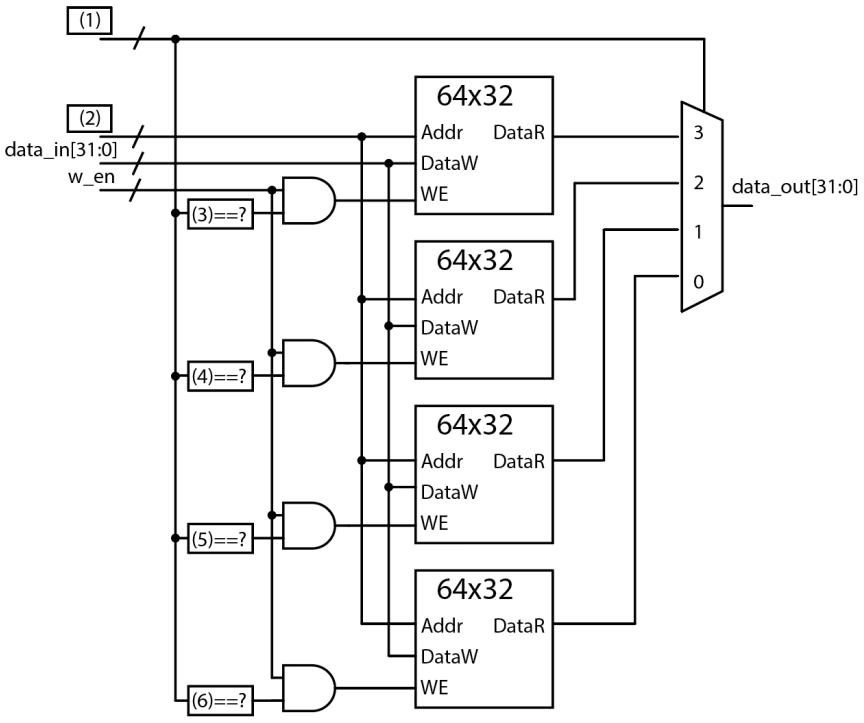
Solution:

Since we want a 1:1 aspect ratio he memory size

$$\sqrt{65536} = 256$$

-¿ 256×256 So we want $256/32 = 8$ rows and $256/16 = 16$ columns

b Now you are given a memory block that is 64×32 and supposes you want to use multiple instances to design a 256×32 memory. The diagram below is a possible implementation. Fill the address signal names in the blanks and use the little-endian convention. For the blanks at the input of AND gates, fill in the decoded result (==00, 01, 10 or 11). Assume it's word-addressable.



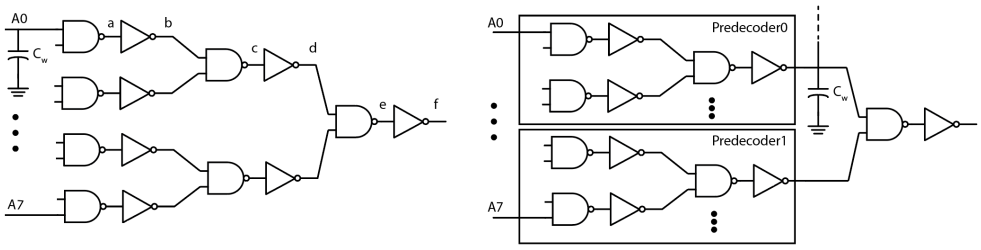
1.
2.
3.
4.
5.
6.

Solution:

1. addr[7:6]
2. addr[5:0]
3. 11
4. 10
5. 01
6. 00

3 Memory decoder

Now we want to design the row decoder for an SRAM that has 256 rows. We will compare the decoding with and without the pre-decoder technique presented in the lecture. We will only use 2-input gates and inverters. Each word line has 256 cells. And the capacitance C_{cell} for each cell includes the transistor and wire capacitance.



a We want to investigate the switching activity at each node. Now assume all the address inputs (A0-A7) have the same possibility equal 1 or 0 ($P(A_i = 1) = 1/2$). Determine the activity factor $\alpha_{0\to1}$ (the possibility of one node changes from 0 to 1) of each node in the 8-input AND gate decoder.

- $\alpha_a =$ ☐
- $\alpha_b =$ ☐
- $\alpha_c =$ ☐
- $\alpha_d =$ ☐
- $\alpha_e =$ ☐
- $\alpha_f =$ ☐

Solution:

- $\alpha_a = 3/4 * 1/4 = 3/16$
- $\alpha_b = 1/4 * 3/4 = 3/16$
- $\alpha_c = 1/16 * 15/16 = 15/256$
- $\alpha_d = 1/16 * 15/16 = 15/256$
- $\alpha_e = 1/256/ * 255/256 = 255/65536$
- $\alpha_f = 1/256/ * 255/256 = 255/65536$

b (251 only) Under the same assumption of question (a), consider the long address wire capacitance C_w in the circuit above. Assume the SRAM is consistently read by a clock with frequency f_{clk}. And the SRAM connects to supply V_{dd}. What is the power that it spends on one of those long wires? Provide your answer in terms of f_{clk}, V_{dd}, C_w

For the decoder without pre-decoding technique

For the decoder using pre-decoders

Solution:

For the decoder without pre-decoding technique, the switching activity at this node is 1/4. So Cw consumes

$$P = \alpha_{0\to1} C_w V_{dd}^2 f_{clk} = 1/4 C_w V_{dd}^2 f_{clk}$$

For the decoder using pre-decoders, it consume

$$P = \alpha_{0\to1} C_w V_{dd}^2 f_{clk} = \frac{15}{256} C_w V_{dd}^2 f_{clk}$$

c (Optional) What are the total fanout, branching effort, and path logical effort for each address wire? Here we assume the decoder input capacitance. In this question, we ignore the wire capacitance C_w in the diagram and we assume the input capacitance for each address is $C_{address} = C_{in}$ and also $C_{in} = 4C_{cell}$

H=

B=

G=

Solution:

Because each address line drives 128 AND8 gate, so B=128. And $G = \frac{3}{2}^3 = \frac{27}{8}$. So $H = 256 * (1/4) C_{in}/C_{in} * B * G = 64 * 128 * 27/8 = 27648$

d (Optional) Now we use the pre-decoder technique presented in the lecture slides. For the total 8-bit address, we use two 4-bit pre-decoder. And the final decoder is an AND2 gate. We still assume $C_{address} = C_{in}$ and $C_{in} = 4C_{cell}$. What are the total fanout, branching effort, and path logical effort for each address wire? In this question, we also ignore the wire capacitance C_w in the diagram.

H=

B= ☐

G= ☐

Solution:

Because each output of predecoder drives 16 AND2 gate and each address bit drives 8 predecoder. So branching effort B = 16*8 and H=64*16*8*27=27648.

e (Optional) What is the optimum number of stages for these two cases? Round up the result to an integer.

The optimum number of stage without predecoder is: ☐

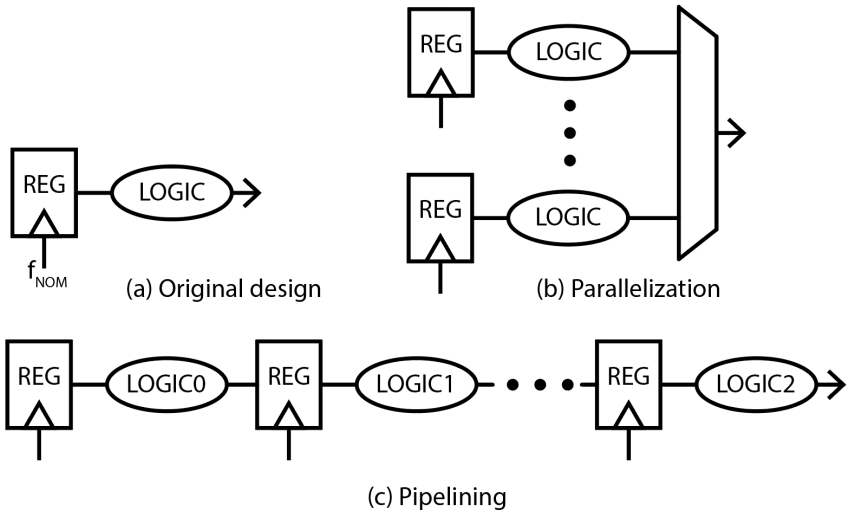
The optimum number of stage with predecoder is: ☐

Solution:

The optimum number of stage without predecoder is 8 and the optimum number of stage with predecoder is also 8.

4 Parallelism and Pipelining

In this problem, we will look into low power design tradeoffs. You are given the logic block is shown below, which, at nominal supply voltage $V_{DD,NOM}$ runs at frequency f_{NOM} . You may assume that the logic block has total average capacitance C_{logic} (average capacitance defined as the product of activity factor $\alpha_{0\to1}$ and total load capacitance C_L), the register block has average capacitance C_{reg}. The delay of a register (i.e., setup time, clk-to-Q, etc.) is negligible compared to the logic delay of the path. After analyzing your circuit you determined that the delay of the logic block can be simplified using a linear approximation: $t_d \propto \frac{1}{V_{DD}}$



a Derive an expression for the dynamic power (P_{NOM}) dissipated in the circuit at nominal operating conditions. Provide your anwser in terms of $V_{DD,NOM}, f_{NOM}, C_{logic}, C_{reg}$.

Solution:

$$P_{NOM} = C_{logic} * V_{DD,NOM}^2 * f_{NOM} + C_{reg} * V_{VDD,NOM}^2 f_{NOM}$$

b You decide to exploit N-times parallelization (N parallel paths) to decrease the power dissipation of your design while maintaining the same throughput. An N-input multiplexer with average capacitance $C_{MUX,data}$ at each data input, and average capacitance $C_{MUX,sel}$ at each select input has to be used in order to reserialize the parallel paths back to a single stream. What is the clock frequency of the registers and the data and control inputs of the MUX? What is the supply voltage you will use in terms of the nominal values and N? Note that you can assume that the delay of the MUX is negligible. Also, for simplicity, you can assume the MUX has M select signal inputs.

The frequency of the registers and the data inputs of the MUX is

The frequency of the MUX select inputs is

The supply voltage for the logic and register blocks is

Solution:

The clock frequency of the registers and the data inputs of the MUX is f_{NOM}/N, but the MUX control inputs still operate at f_{NOM}. This reduced clock frequency allows N time more time for the logic, and since $t_d \propto \frac{1}{V_{DD}}$, we can now reduce the supply by a factor of N. Therefore the new V_{DD} supply voltage is $V_{DD,NOM}/N$.

c Derive an expression for the total power dissipated in the parallelized design. You may assume that the multiplexer is the main contributor to overhead capacitance, and we can ignore other energy overhead (i.e., only consider the power consumed by the blocks in the figure above). Provide your anwser in terms of $V_{DD,NOM}, f_{NOM}, C_{logic}, C_{reg}, C_{mux,data}, C_{mux.sel}, N$ and M.

Solution:

$$P_{parallel} = N * (C_{reg} + C_{logic} + C_{MUX,data})(\frac{V_{DD,NOM}}{N})^2 \frac{f_{NOM}}{N} + MC_{MUX,sel}V_{DD,NOM}^2 f_{NOM}$$

d (251 Only) Another way to reduce power while maintaining the same throughput is to use pipelining. Let's assume we need M pipeline stages to maintain the same throughput as the original design, derive the equations to determine the total power for pipelining. You can assume the total capacitance in the logic circuit can be evenly assigned to each pipeline stage. Provide your anwser in terms of $V_{DD,NOM}, f_{NOM}, C_{logic}, C_{reg}, C_{mux,data}, C_{mux.sel}$ and M.

Solution:

$$P_{pipeline} = M * \frac{C_{logic}}{M}(\frac{V_{DD,NOM}}{M})^2 f_{NOM} + MC_{reg}(\frac{V_{DD,NOM}}{M})^2 f_{NOM}$$