

University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences

EECS151/251A
Spring 2021

J. Wawrzynek
5/14/21

Final Exam

Name: _____

Student ID number: _____

Class (EECS151 or EECS251A): _____

You have three hours to take the exam. This exam comprises a set of questions with 1 point per expected minute of completion with a total of 130 points. As with homework problems, submit your solutions using Gradescope. At the end of the exam time, you will have extra time to scan and submit your answers.

You are allowed to refer to your notes, the class lecture notes, and any other reference materials that you have available. You are not allowed to speak or communicate with anyone on any topic related to the exam during the exam period. After completing the exam, sign the following statement attesting that you did not discuss the exam problems with anyone else. You may either scan this page or copy the statement word-for-word.

I hereby declare that I have not spoken with nor otherwise communicated with anybody regarding the content of this exam while taking the exam (except for course staff):

(sign here): _____

For each problem if you find yourself taking excessive time to work out a solution consider skipping the problem or a fresh approach. Also, start by answering the easier questions and then move on to the more difficult ones.

Note: We will drop from your total exam score your lowest score from problems 6, 7, and 8. Therefore, you may choose to simply skip one of the three and we will drop it. Or you may try all three problems, and we will grade them then drop the lowest.

Neatness counts. We will deduct points if we need to work hard to understand your answer.

Before you turn in your exam, write your student ID number on all pages.

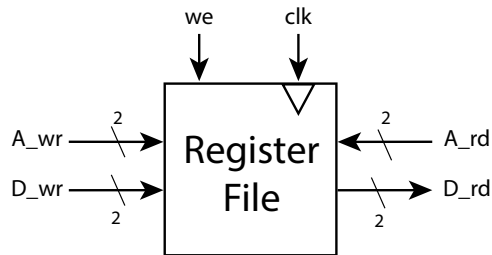
1. Register File [15 pts]

For this problem, you are given latches and must design a 4 word by 2-bit wide register file (8 bits total) using the *least number of latches possible*.

The latches are simple 1-bit wide level-sensitive latches with a clock input (clk), a data input (d), and an output (q).

The register file must have one read port and one write port, as shown below.

Read is asynchronous, but writing is synchronous and positive edge-triggered.



In addition to latches, you may use inverters, 2-input logic gates, and 2-input multiplexors as needed. Neatly draw your circuit diagram. Explain its operation.

2. Pop Count [10 pts]

A common function on some CPUs is the “population count” operation. The output is the binary encoded count of the number of 1’s in the input word.

Design a combinational logic circuit for population count with an 8-bit wide input.

For your design, you may use full-adder cells (FA), inverters, 2-input logic gates, and 2-input multiplexors. Keep your circuit as simple as possible.

Neatly draw your circuit and label all inputs as x_0, x_1, \dots, x_7 and outputs as y_0, y_1, \dots

3. Bit Reverser [5 pts]

Consider the design of a combinational circuit for conditional *bit reversal*. It takes an n -bit input x_{n-1}, \dots, x_1, x_0 and a control input, r , and outputs x_0, x_1, \dots, x_{n-1} , if $r = 1$, else it passes the input to the output.

Your friend tells you that in hardware the best attainable lower bound for the delay of this operation is $O(\log n)$. Are they correct?

To answer the question, first draw a circuit that can achieve conditional bit reversal. Then for your circuit, analyze and explain the lower bound on delay, considering logic delay and wire delay separately.

4. CPU Design [50 pts] *Note: This is a long problem. Allocate a significant portion of your time budget to this problem.*

Consider the design of a simple 16-bit CPU that has only two registers visible to the instruction set—“acc” is used for ALU and load/store operations, and “pc”, the program counter. The complete CPU instruction set is shown below.

name	action	opcode $\{op_3, op_2, op_1, op_0\}$	XX	operation
load	acc \leftarrow DMEM[immed]	11 11	00	ADD
store	DMEM[immed] \leftarrow acc	01 11	01	SUB
loadi	acc \leftarrow immed	10 11	10	NOR
alu	acc \leftarrow acc OP DMEM[immed]	11 XX	11	pass B
alui	acc \leftarrow acc OP immed	10 XX		
branch	if acc == 0 PC \leftarrow PC + immed	00 00		

Note, “XX” encodes the operation, A OP B, to be performed by the alu instructions. XX = 11 means to just pass the “B” input to the output. This “pass B” operation gives you a way to simplify the datapath and controller.

“immed” stands for the immediate value from the instruction. The CPU works with 16-bit values (for operands and memory addresses). Therefore the immediate must be sign-extended to 16 bits before it is used.

The CPU has the following instruction format:

opcode	immediate
4-bits	12-bits

For your design, you have only the following components, with delays given in ps:

module	timing (ps)	notes
ALU	$\tau_{ALU} = 350$	performs all the actions of XX above
adder	$\tau_{add} = 150$	16-bit addition
MUX-2	$\tau_{MUX} = 25$	2-input multiplexor
Equal-zero-compare	$\tau_{EQ} = 40$	takes 16-bit input, outputs 1 if equal to zero
2-input logic gates	$\tau = 10$	
inverter	$\tau = 5$	
16-bit register	$\tau_{clk-q} = 25,$ $\tau_{setup} = 25$	has a clock-enable input (ce)
DMEM	$\tau_{address-to-data} = 250,$ $\tau_{setup} = 25$	Asynch read/synch write, 16-bit address/data, word-addressable
IMEM	$\tau_{address-to-data} = 250$	Asynch read, 16-bit address/data, word-addressable

For both IMEM and DMEM, the 16-bit address points to a 16-bit word (they are not byte-addressable).

(a) Single Cycle Processor Design:

- i. Assuming one cycle per instruction, design the datapath:

Keep things as simple as possible. Draw your datapath circuit as neatly as possible (it might take a couple of tries to get a neat version). Use dots where wires are meant to connect. To the extent possible, avoid making connections without explicitly drawing a wire. Draw a circle around the name of each of your control signals where they appear in your datapath.

- ii. Design the controller. You don't need to show the detailed circuit, but write out the logic equations for each control signal.

(b) Analysis:

Based on your design, calculate the maximum clock frequency. You may assume perfect clock distribution and therefore ignore clock uncertainty. Show your work.

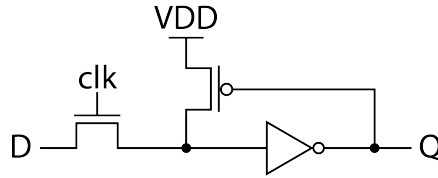
(c) Optimization:

Now we would like to pipeline your CPU. The target clock period is 600ps or less. Devise a pipelining scheme with as few stages as possible.

- i. Describe your pipeline structure, i.e., what operations occur in each pipeline stage.
- ii. Describe how your design would need to change to implement pipelining.
- iii. Describe what hazards are created by pipelining.
- iv. Describe what would need to change to mitigate hazards.

5. Dynamic Latch [5 pts]

Below is a circuit that has been used as a “dynamic latch”. It’s a level sensitive state element and found common use for pipelining and other places where the new state was loaded on each clock cycle.



- How does this latch hold state?
- What is the purpose of the single pFET in the middle of the circuit?
- How is a new state value written when we wish to change the held state value (explain both loading a 0 and loading a 1)?
- What are the constraints on the transistor sizing of this circuit? Assume the D input is driven with an ideal voltage source.

6. Configurable SRAM Design [15 pts]

We will include your best two of problems 6, 7, and 8.

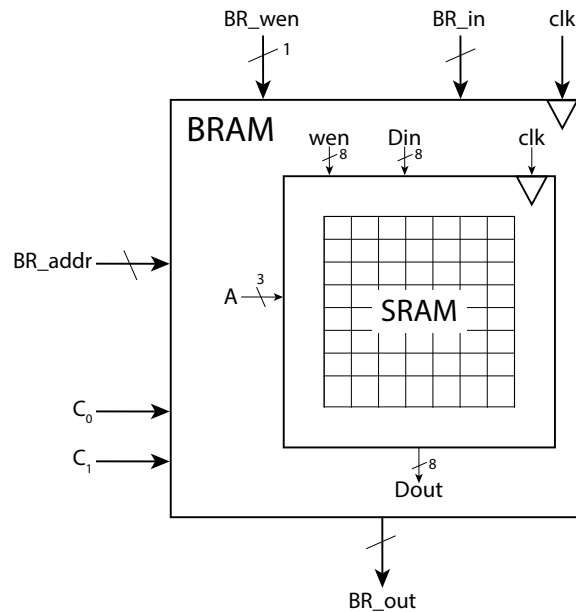
For this problem you are given a SRAM block that has 8 rows of 8-bits each. The SRAM has a single 3-bit address port, A, and separate 8-bit data in (Din) and data out (Dout) ports. Both read and write are synchronous.

The SRAM also has an 8-bit write enable input that controls which columns are driven on a write operation. The enable signals $wen[0], wen[1], \dots$ correspond to the input data bits, $Din[0], Din[1], \dots$. On the rising edge of the clock if $wen[i]=1$ then $Din[i]$ is written.

Your job is to add circuitry to implement a “configurable Block RAM” (BRAM), such as those in FPGAs. The BRAM can be configured to act as one of the following:

configuration	depth	width
3	64	1-bit
2	32	2-bit
1	16	4-bit
0	8	8-bit

As show below, for the BRAM, the configuration number comes in on two input ports, c_1, c_0 .



- Explain a scheme for the layout of BRAM words within the rows of the SRAM.
- Briefly explain your scheme for writing BRAM words to the SRAM.
- Using only inverters, simple logic gates (no more than 4-input), and 2-input multiplexors as needed, design the additional circuitry needed to implement the configurable BRAM *read* function. *You do not need to show the circuitry for writing.* Neatly draw your design.

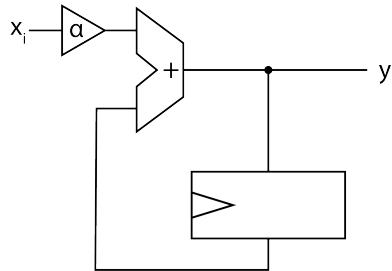
7. Dynamic Power [15 pts]

We will include your best two of problems 6, 7, and 8.

In this problem, you are faced with the task of improving dynamic power consumption in the following computation, while *maintaining the same throughput*.

$$y_i = \alpha x_i + y_{i-1}$$

The circuit you are given is shown below:



The circuit consumes one input and generates one output per 1 ns cycle.

Assume that $E_{mult} = 3E_{add}$ and $\tau_{mult} = 3\tau_{add}$ (in other words a multiply operation takes 3 times the energy of an addition and takes 3 times as long as an addition).

The basic multiply operator and adder are not divisible—they cannot be split into pipeline stages.

You may assume that there is no limit on the amount you can scale Vdd, and that circuit delay scales linearly with voltage. For this problem, ignore delay and energy in registers and multiplexors, and all static power consumption (leakage).

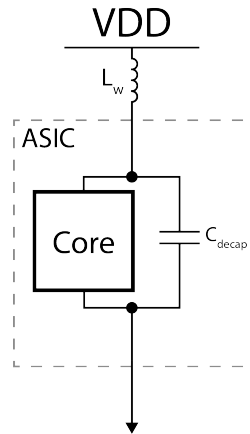
Your target is to reduce power consumption so that it is less than 1/2 of what it is in the given circuit. You may add circuit components, scale voltage to all or part of your circuit, and adjust clock frequency as necessary.

Explain your approach, and neatly draw your transformed circuit, and analyze the change in dynamic power.

8. Power Supply [15 pts]

We will include your best two of problems 6, 7, and 8.

You are designing an ASIC. In operation, your IC is expected to draw 1A of current at 1V over a period of 100 ps. You have allocated bump connections to the package with a total effective inductance of 25pH for supplying Vdd. The design specification says that the voltage variation on the internal power supply rail should never exceed 100mV (10% of the nominal value of 1V). All your IC connections are used up, so your only option is to add a decoupling capacitor on the IC that will connect between Vdd and GND, as shown below:



What is the minimum value for the capacitor that you should use? Assume that you size the power wires sufficiently, so that you can ignore IR voltage drops.

(Hint: remember the following equations for current through a capacitor and voltage across an inductor: $I_C = C \cdot dv/dt$, $V_L = L \cdot di/dt$)

Show your work.

Student ID number:

Saturday 15th May, 2021 01:53