EECS 151/251A Homework 9

Due Wednesday, April 17$^{\text{th}}$, 2024

Introduction

This homework is meant to test your understanding of memories and energy/power. There are five total questions. Please check Ed first if you have any questions. Note for some questions you may need to consult with online resources, or other additional material beyond lectures.

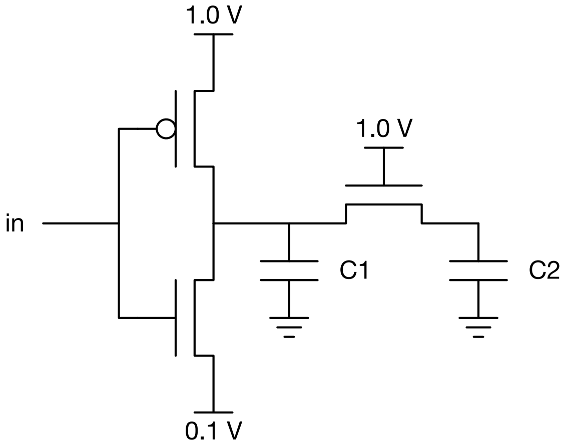
Problem 1: SRAM vs. DRAM

Fill in the table below for the different types of memory used in digital circuits (for “Compositon” describe how the memory is created i.e. latches, capacitors, etc and for “Noise Resilience” rate as *Low*, *Medium*, or *High*):

	Registers	SRAM	DRAM	Hard Disk	Flash
Approx. Access Time (ns or s)					
Density (Mb/area)					
Volatility					
Approx. Cost/Bit ($)					
Power Usage (W)					
Composition					
Example Usage					

Problem 2: Energy

Consider the circuit shown below. Every transistor has the same effective resistance, R. The first node after the inverter has capacitance $C1$ and the node after the pass-transistor has capacitance $C2$. These capacitance values represent *all* the capacitance associated with those nodes. Initially, the gate terminal of the pass-transistor is connected to V_{DD}. The input to the inverter is a square wave signal with frequency f.



$$C1 = 90fF$$
$$C2 = 53fF$$
$$V_{th,n} = |V_{th,p}| = 0.32V$$

(a) What is the dynamic power consumption of this circuit?

(b) Now suppose we set the gate terminal of the pass-transistor to 0V. What now is the dynamic power consumption.

(c) What can be done in principle to lower the power consumption of this circuit without decreasing f (reducing performance)?

Problem 3: Race to Halt

Background: *"Race to halt" is an effective energy efficiency strategy. In this scheme, a processor will run at it's maximum frequency to finish the required work as quickly as possible, then cut power to the circuit. Understandably, this strategy is effective when static power consumption is a dominant or significant component of total power consumption (although this specific strategy is implemented for CPUs, the general strategy can be applied to any circuit).*

Suppose you have a ML accelerator and you want to utilize it to run a parallelizable workload. The accelerator is composed of four matrix multiply sub-units. Functionally, all the sub-units are equivalent, capable of 9.5GFLOPs, however have different static power consumption. Sub-unit 0 is an efficiency unit which consumes 2W of static power while all the other sub-units consume 6W of static power. Regardless of the number of sub-units used, the accelerator consumes 2W of static power.

Additionally, there is a data partition unit which is by default configured to partition incoming data into blocks and send these blocks to individual sub-units (i.e. one block per sub-unit). The power consumed to partition is negligible. However, it costs 32J to reconfigure the data partition unit to not partition the data and send all incoming data to a single sub-unit. The interconnect which connects the data partition unit to the sub-units cost 0.25W per active sub-unit. Assume the interconnect is always on and consuming power regardless if data is being transmitted.

Your application requires an average of 200G floating point ops. Based upon estimates, it known that your application will consume 4W of dynamic power if ran on a single sub-unit.

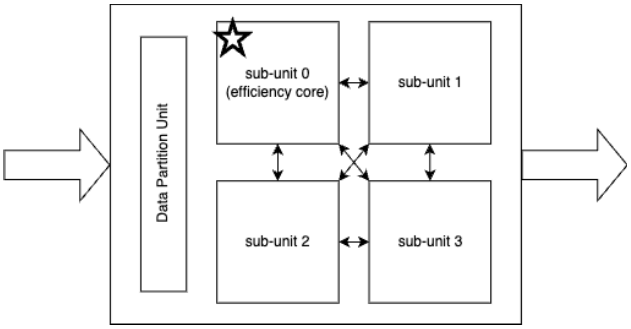


Figure 1: High-level block diagram of matrix multiplication accelerator

You would like to determine the most energy efficient way to run the application. You have the ability to control the supply voltage (V_{DD}), the clock frequency (f), and cut power to the accelerator when not in use. Assume that static power remains constant if voltage is scaled. Perform an energy analysis for three separate schemes (**assume voltage and frequency scale together**).

1. Voltage and clock frequency scaling running on single sub-unit
2. Voltage and clock frequency scaling by paralleling across multiple sub-units (**scaling factor = # of sub-units used*)
3. Implement a race-to-halt like energy savings scheme using a single sub-unit

Which approach is most energy efficient? Show your work and justify your answer.

Problem 4: DRAM in the Real World

DRAM has a high density which makes it useful, but also has a long access latency. Modern DRAM modules are organized and implemented to contain structures to minimize this latency. Reducing access latency is paramount to CPUs, but also any digital circuit which relies on high bandwidth, high density storage (GPUs, FPGAs, accelerators, network cards, etc). Another primary concern is power consumption. Over the years, power saving strategies have been implemented in DRAM. Provide answer to the following prompt to learn about DRAM organization and common performance techniques used to improve performance of DRAM (feel free to consult online resources):

(a) In your own words, describe what a memory bank is.

(b) In your own words, describe what a memory rank is.

(c) In your own words, describe what a memory module is.

(d) In your own words, describe what the power-down mode does in DRAM.

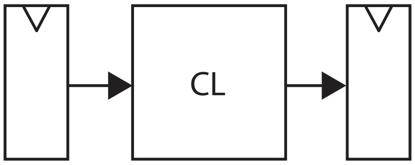
(e) From the answers above, it is understood that DRAM is architected as a hierarchy. Explain how this hierarchy can decrease latency and reduce power consumption.

(f) Provide the primary reason you cannot repeatedly access the same row, in the same bank as clock frequency increases.

(g) 1-T DRAM designs usually include a “row buffer”—a register on the periphery that is used to register an entire row. Explain how this register could be used and why it’s a good idea.

Problem 5: Energy Efficiency Improvements

The block diagram shown below represents the critical path for a circuit. This path has 0 slack. The timing specifications are as follows: $\tau_{CL} = 5ns$, and $\tau_{setup} = \tau_{clk-Q} = 1ns$.



On average, at some V_{DD} the energy for one data item passed through the combinational logic block is 3.6 J. The registers each consume 0.1 J on average for each new data word stored.

Assume you have an application where the latency for the output corresponding to the first input does not matter. In other words, your application is latency-insensitive. However, after the first output appears the application requires the circuit to produce results at a rate of 125MHz (one result per cycle).

It is possible to split the combinational logic evenly (in terms of both delay and energy) into multiple blocks.

Devise a scheme that would improve the switching energy efficiency while meeting the application requirements. Compare the switching energy per result of the original circuit and your new one.

Assume that voltage and clock frequency scale together.