



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

E. Alon

Friday, Apr 7th, 2017

5:00-6:30pm

EECS 151/251A: SRPING 2017—MIDTERM 2

NAME	Last	First
-------------	------	-------

GRAD/UNDERGRAD	
-----------------------	--

Problem 1: ____ / 14

Problem 2: ____ / 16

Problem 3: ____ / 20

Problem 4: ____ / 20

Total: ____ / 70

PROBLEM 1. (14 pts) CMOS Power and Energy

- a) **(6 pts)** Assuming that our design consists entirely of LUTs, in this problem we will examine how many such LUTs we can include in our design while staying within a given power budget. For each LUT, you can assume that the total capacitance associated with that LUT is 5fF, the activity factor associated with that capacitance ($\alpha_{0 \rightarrow 1}$) is 0.1, and the average leakage current is 100nA. If the design operates at a supply voltage of $V_{dd} = 0.8V$ and the clock frequency is 1GHz, how many LUTs can the design include if we want to keep the total power under 1W?

- b) **(8 pts)** Now assuming that the leakage current of each LUT is set by $I_0 e^{\left(\frac{-V_T}{1.5 \times 25mV}\right)}$, where V_T is the threshold voltage, and that increasing V_T by 100mV forces you to increase V_{dd} by 100mV (i.e., to $V_{dd} = 0.9V$) to maintain the same clock frequency, how many LUTs can you now include on the chip?

PROBLEM 2. (16 pts) SRAMs and Decoders

- a) **(8 pts)** Using no more than 9 transistors in total, sketch a design for an SRAM cell that would support 2 independent read ports and 1 independent write port. Qualitatively explain how the transistors in your cell should be sized to ensure sufficient stability under the worst-case conditions for reading to/writing from your cell.

- b) **(8 pts)** Now let's look at the delay of a decoder for reading from one of the ports from an array made out of the multi-ported SRAM cell from part (a). Assuming that the array is 128×64 (i.e., each wordline drives 64 cells, and each bitline has 128 cells on it), that the total capacitance on one of the read wordlines from each cell is 0.25fF , that the total capacitance on each address input (you are given both address and address_b as inputs) must be less than 1fF , and that the decoder is built using only NAND2s and inverters, what is the minimum delay of the decoder for this array? You should provide your answer in terms of t_{inv} ; note that you can assume that the parasitic delay of the NAND2 gates is γt_{inv} (i.e., the same as an inverter), and you can ignore the fact that you can't actually build a fractional number of stages.

PROBLEM 3. (20 pts) Architectural Optimization

In this problem we'll consider implementing an Infinite Impulse Response (IIR) filter, whose output y at a given time index n ($y[n]$) is given by:

$$y[n] = x[n] + k*y[n-1]$$

where k is some fixed constant and $x[n]$ is the input at time n . Throughout this problem, you are allowed to use only adders, multipliers, and registers, whose delays (or clk-q/setup for the registers) are t_{add} , t_{mult} , $t_{\text{clk-q}}$, and t_{setup} . Note also that you should assume that the input x to this block is always produced directly by a register.

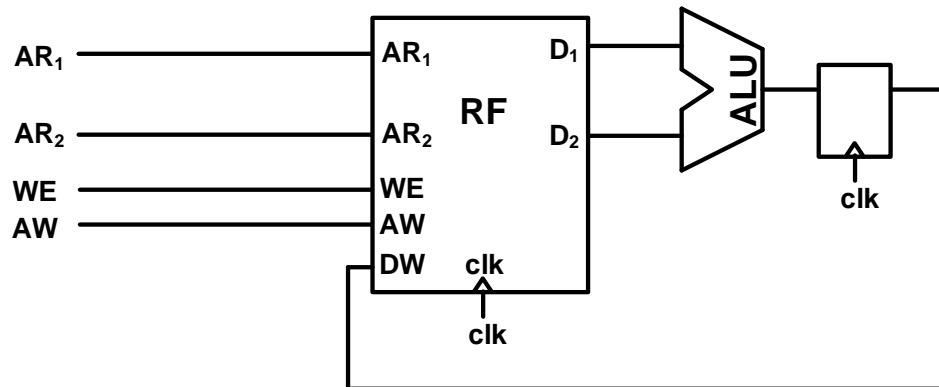
- a) **(6 pts)** Assuming that we want the output of our block to be $y[n-1]$, sketch a simple hardware implementation of this IIR filter and provide an expression for the maximum frequency of this design.

- b) **(8 pts)** Now sketch a new design that makes use of unrolling to compute both $y[n]$ and $y[n-1]$ (with the outputs being $y[n-1]$ and $y[n-2]$) in each cycle. Note that you can assume that both $x[n]$ and $x[n-1]$ are available to your block as inputs. What is the maximum clock frequency for this new design? Is the maximum achievable throughput of the unrolled design higher or lower than that of the original design from part (a)?

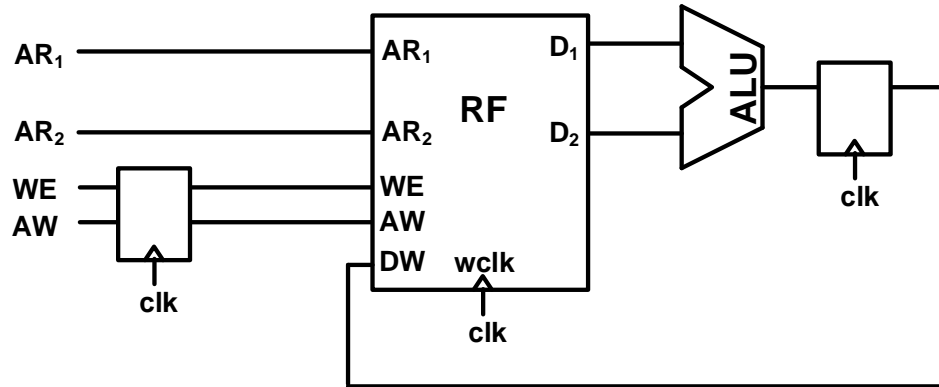
- c) **(6 pts)** Modify your design from part (b) to include an additional stage of pipelining. Sketch the new design and compute the maximum clock frequency of this new (pipelined and unrolled) design.

PROBLEM 4. (20 points) Processors

- a) (6 pts) Shown below is a portion of a prototype design for a pipelined CPU's datapath that uses a register file with **synchronous reads and writes**. However, even ignoring any potential data hazards, this design does not function correctly – in particular, register type instructions (where the ALU operates on data from two registers – specified by AR_1/AR_2 – and writes back to a third – specified by AW) do not seem to execute correctly. Explain what the error is caused by and add any extra components necessary to correct the design.



- b) (8 pts) Now let's examine a slightly different design shown below, where the register file has synchronous writes but **asynchronous reads**. Add appropriate forwarding hardware to eliminate all data hazards from this pipeline; you should show both the forwarding path(s) as well as the hardware you would need to detect the data hazard(s). You are only allowed to use MUXes, adders, comparators (i.e., a block that checks whether two inputs are equal to each other), and any basic logic gates (AND, OR, etc.) in the forwarding hardware you add.



- c) **(6 pts)** In order to accelerate certain applications, your company has decided to augment its processor with specialized instructions and hence hardware units within the processor. After adding the new hardware unit to the processor (the unit operates on the same inputs as the ALU, and when the special instruction is called, the unit's output would replace the ALU's output) you find out that the processor's clock frequency is now substantially reduced, and is totally set by the delay of that unit. In order to try and bring the frequency back up, you decide to add two pipeline registers inside of the new unit.

Assuming that the new instruction is called `MLinst`, and that `MLinst(R1, R2, R3)` implies using the data stored in registers `R1/R2` (whose addresses are specified in `AR1/AR2`) for `D1/D2` and storing the result in `R3` (whose address is specified in `AW`), consider the following snippet of code:

```
MLinst(r1, r2, r3)
MLinst(r3, r4, r5)
```

Is it possible to execute this code snippet without introducing any stalls in to the pipeline? If so, describe and/or sketch what the additional forwarding hardware would need to be. If not, explain why not.

EXTRA PAGE FOR ADDITIONAL/SCRATCH WORK