



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

J. M. Rabaey

511 Cory Hall

TuTh3:30-5pm

e141@eecs

## EECS 141: SPRING 97 — FINAL

**For all problems, you can assume the following transistor parameters:**

**NMOS:**

$$V_{Tn} = 0.75V, k'_n = 20 \mu A/V^2, \lambda = 0, \gamma = 0.5 V^{1/2}, 2\Phi_F = -0.6V, LD = 0.15 \mu m$$

**PMOS:**

$$V_{Tp} = -0.75V, k'_p = 7 \mu A/V^2, \lambda = 0, \gamma = 0.5 V^{1/2}, 2\Phi_F = -0.6V, LD = 0.15 \mu m$$

<b>NAME</b>	Last	First
-------------	------	-------

<b>GRAD/UNDERGRAD</b>	
-----------------------	--

**Problem 1:**

**Problem 2:**

**Problem 3:**

**Problem 4:**

**Problem 5:**

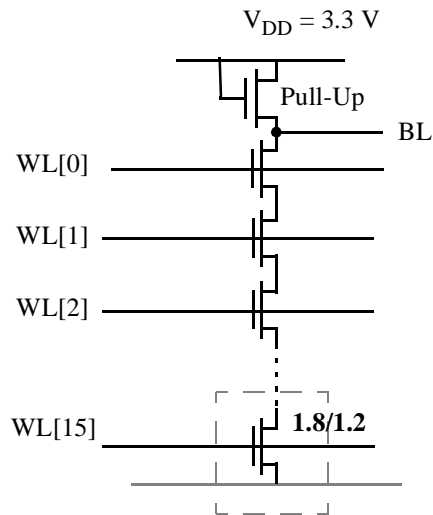
**Problem 6:**

<b>Total</b>	
--------------	--

**Have a great summer!**

### Problem 1: Memory Design

Designer Dilbert is asked to design a low power ROM. He decides that a NAND structure with 16 cells per column is probably the appropriate choice. He finally comes up with the structure shown below. The following properties of the cell are known: size:  $6\mu\text{m} \times 6\mu\text{m}$ ,  $(W/L) = (1.8/1.2)$ . Source and drain overlap capacitance of 3 fF, gate capacitance of 8 fF, source and drain diffusion capacitance = 10 fF, bitline/capacitance per cell = 1 fF.



a. Evaluate if Dilbert is correct in his assumption that this structure is indeed low power. Give the pro and con arguments.

b. Determine the size of the pull-up transistor such that the maximum voltage swing at BL equals 0.5 V. You may ignore body effect in this problem.

W/L=

c. Draw the equivalent model that you will use to determine the worst case delay for a read operation. Annotate the capacitance values on the elements of your model. You may assume that the load capacitance at BL equals 100 fF (this includes the capacitance of the pull-up).

d. Derive the value of the worst-case propagation delay.

$t_{p\text{worst}} =$

e. Determine the average power dissipation of a memory array composed of 16 of the above columns. Assume that the chance of having a 0 or 1 at a particular position in the array is equal. Determine both the static and dynamic components.

$P_{\text{static}} =$   
 $P_{\text{dynamic}} =$

f. Suddenly, Dilbert gets a stroke of genius. He figures out that by using a special encoding on the stored data he can save some more power (see Figure below). The scheme works as

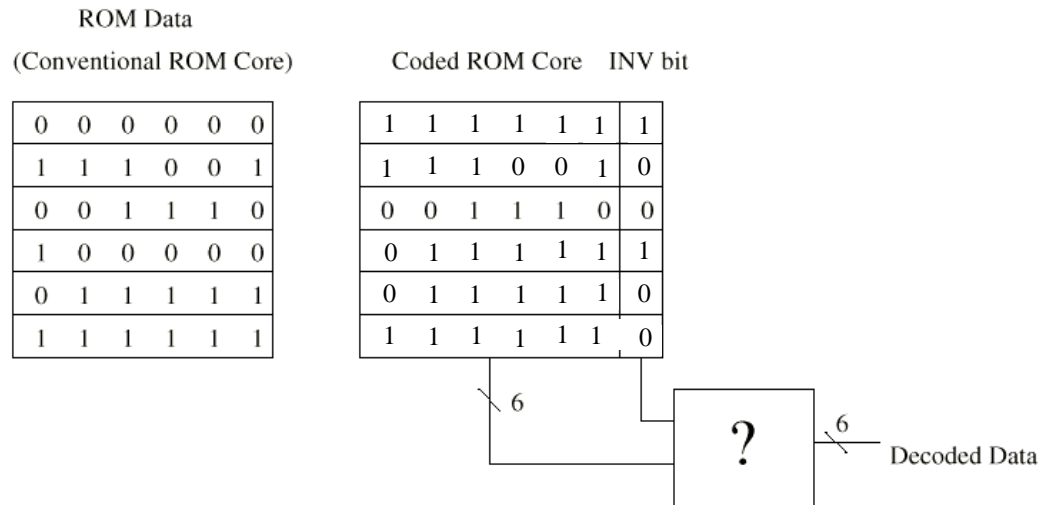


Figure: Example of encoding scheme (for 6 x 6 ROM)

follows: assume that for a given wordline  $WL[i]$  the number of stored 1's is larger than the number of 0's. In that case the data is left unchanged and  $INV[i]$  is set to 0. In case the number of stored 0's is larger than the number of 1's, all data in the memory is inversed and bit  $INV[i]$  is set to 1. The INV bits are stored as an extra column in the memory. Explain why this approach can help to save power.

g. Determine the logic function that should be performed by the box with the question mark in the Figure so that the correct data is retrieved from the memory.

## PROBLEM 2: Arithmetic

It is your task to implement an arithmetic circuit that performs the following task:  $x = (a + b) \geq c$ , where  $a$ ,  $b$ , and  $c$  are all  $N$ -bit two's complement numbers and  $x$  is a one bit result!

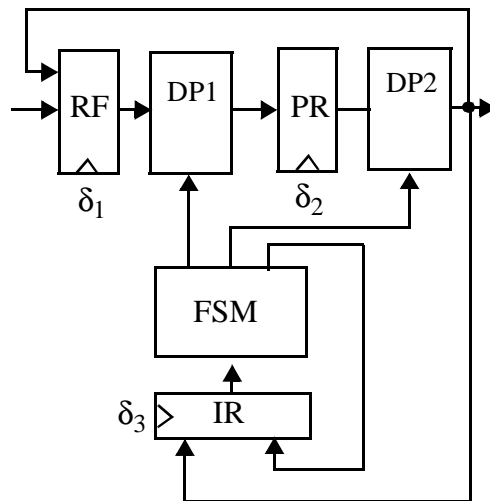
a. Draw the block diagram of your datapath using **RIPPLE-BASED LOGIC**. The block diagram should contain the basic cells you are using and their interconnections (in the style of Figure 7.31 in the text book). For each cell, describe the logic function.

b. Determine the delay of the datapath as a function of  $N$  and the delay parameters of the basic cells.

c. Describe an *architectural* technique to speed up the datapath and draw the revised block diagram. Architectural means at the overall block diagram or logic levels. For instance, picking another logic style does not count. Pipelining is not an allowed option for this problem.

### PROBLEM 3: Timing

Consider the following simple processor, consisting of a pipelined data path and a finite-state machine based controller. RF, PR, and IR denote edge-triggered flip-flops, while DP1, DP2 and FSM denote logic modules. Minimum and maximum delays of the modules are shown in the table next to the Figure. You may ignore the delay of the interconnect as well as the delays of the registers. The  $\delta$ 's at the clock inputs of the registers denote the absolute skew between the clock source and the register.



	min delay	max delay
DP1	3	10
DP2	2	8
FSM	1	5

Figure : Simple Processor

a. Write down the necessary constraints on the clock skews to avoid race conditions. Do not solve!.

b. Derive the constraints on the clock period in the presence of skew. Do not solve.

c. Assuming that you are free to set the values of the skews, determine the minimum possible clock period. .

$T_{\min}$
------------

d. Determine the values of the skews for which this minimum period is achieved.

e. Propose a revised **architecture** that would reduce the clock period (changing circuit style is not an option ...). Explain why. Discuss also the disadvantages of your approach.



#### PROBLEM 4: Interconnect

Designer I(mp) Edance is working hard to design a **fast** serial link between two chips placed together on a board. The wire between the two chips is 100 mm long and 5  $\mu\text{m}$  wide. It is terminated at the receiver end by a small inverter with an ignorable input capacitance. Imp has two drivers available in his library with an equivalent output resistance of 500  $\text{k}\Omega$  (driver 1) and 50  $\Omega$  (driver 2), respectively, in both up and down directions. The wire is best categorized by its inductance ( $l$ ) and capacitance ( $c$ ) per unit length, that equal 1.5 pH and 0.25 fF/ $\mu\text{m}$ , respectively.

a. Assume first that the wire is implemented in high resistance aluminum with a sheet resistance of 0.1  $\Omega/\text{square}$ . Compare both drivers by computing at which point the 90% point of the final voltage is reached. Determine which driver is the best solution for this case (**consider both performance and area**).

$t_{90\%}$  (Driver 1) =

$t_{90\%}$  (Driver 2) =

b. Assume next that the wire is implemented in copper, for which the resistance is that low that it may be ignored. Which driver is now the better choice? Compute for both cases at what point **90% of the final voltage** is reached. Explain your result.

$t_{90\%}$  (Driver 1) =

$t_{90\%}$  (Driver 2) =

c. Answer yes or no to the following questions related to interconnect:

- Electromigration problems can be resolved by increasing the wire width.
- When running into transmission line effects, it helps to replace the wire material by a wire material with lower sheet resistance.
- Crosstalk between wires can have an adverse effect on speed performance.
- Ldi/dt effects can be cured by increasing the sizes of the transistors in the drivers for the output pads.

### PROBLEM 5: Drivers

The following circuit is supposed to represent a better driver structure.

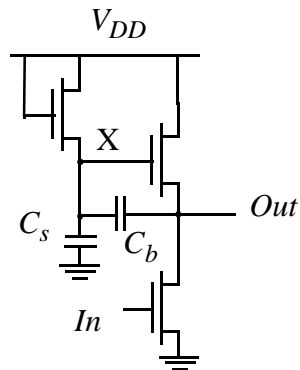
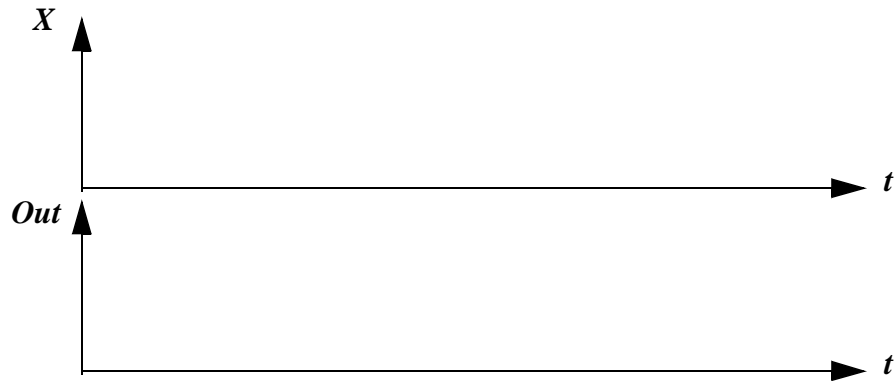


Figure: Driver

- a. Determine how the circuit operates by drawing the waveforms at nodes *X*, and *Out* during a high-to-low transition at the input. Assume that  $C_s \ll C_b$ .



- b. Derive an approximate expression for the voltage at node *X* at the end of the transition.

- c. Describe briefly why this leads to a better buffer.

d. The capacitor  $C_s$  is the result of the parasitic capacitances of the transistors and wires.  $C_b$  on the other hand has to be large and has to be explicitly added by the designer. Describe how you would implement such a capacitor in a traditional CMOS process .

### PROBLEM 6: Logic

The figure below shows an alternative implementation of Complementary Pass-transistor logic.

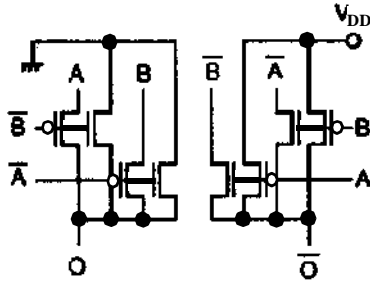


Figure: Logic

a. Determine the logic function of the gate.

$O =$
$\overline{O} =$

b. Discuss why this logic family would be preferable over traditional NMOS-only CPL and full CMOS transmission gate pass-transistor logic.

i) with respect to implementation complexity

ii) with respect to noise margins

iii) with respect to performance