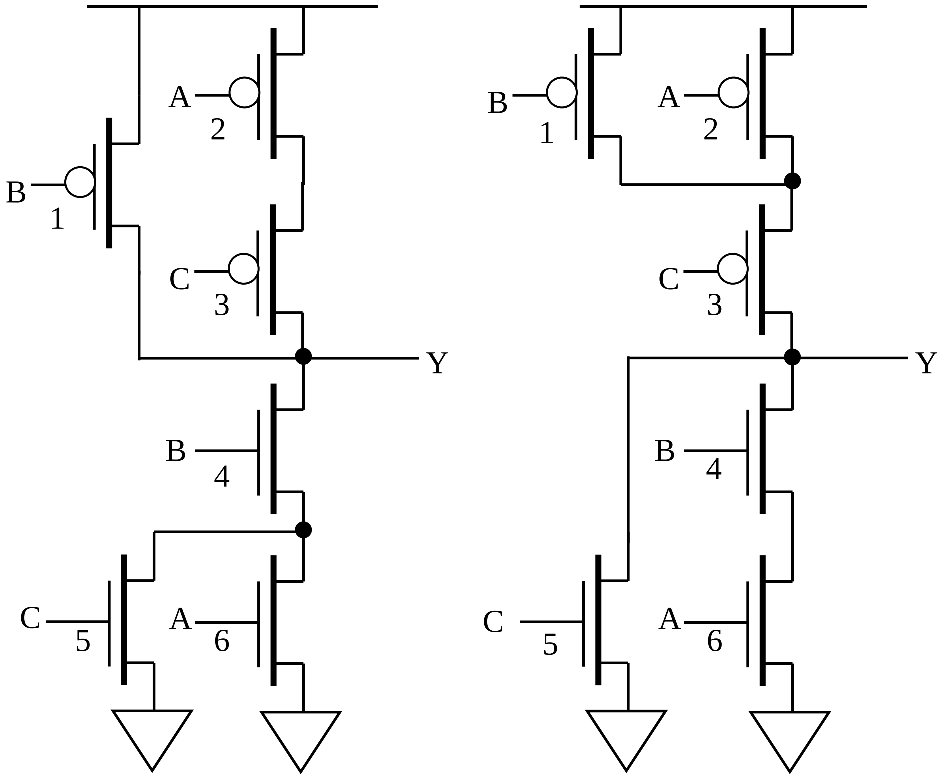
DEPARTMENT OF ELECTRONIC AND ELECTRICAL ENGINEERING

Spring Semester 2009-2010 (2 hours)

Answers to Introduction to VLSI Design 3/6, Questions 1…4

1. a. *The three-input CMOS logic circuit shown in* ***Figure 1*** *is incomplete:*

i) *Draw both of the possible complete circuits that could be formed by adding transistors (you cannot cut any of the existing wires);*



Circuit 1 **Circuit 2**

(8)

ii) *Write down their logical functions;*

Circuit 1: $Y = \overline{(A+C)B}$

Circuit 2: $Y = \overline{C + AB}$ **(6)**

iii) *Size the transistors for each of the circuits (as multiples of the width of a minimum-size N-type FET) assuming a minimum sized gates (using the normal assumptions).*

Circuit 1: 1=2; 2,3=4; 4,5,6=2

Circuit 2: 1,2,3=4; 4=2; 5=1; 6=2 **(6)**

2. **a.** *Distinguish between static and dynamic power dissipation in circuits.*

Static power consumption arises because of the intrinsic behaviour of the devices whilst dynamic power consumption arises due to the activity of the circuit i.e. changes of voltage on circuit nodes. **(4)**

b. *One of the major sources of dynamic power dissipation in CMOS circuits is due to switched capacitance.*

i) *Develop an expression to show what the power dissipated in a CMOS circuit due to switched capacitance should be (ensure that you define all terms and state assumptions).*

Consider charging a capacitor, C to V_{DD}. The charge on the capacitor will be CV_{DD}. As the capacitor is discharged to 0V, this charge will flow down to earth. The net charge moved through V_{DD} is, therefore, CV_{DD}. If this operation is being done f times a second then the charge moved per second is fCV_{DD}. and this is current, axiomatically. This current flows across V_{DD} and so the power dissipated by this switching activity, P_{sw}, is fCV_{DD}^2. To put this in terms of a circuit, $C=C_{in}+C_{wire}$ (the sum of the gate's input capacitance and the capacitance of the wire driving the gate input), and f is the frequency at which the input is being driven.

If we are to extend this expression from a single gate to an entire circuit we must perform a summation across all of the gates and interconnect in the circuit. So if we assume that there are n wires in the design and the total capacitance associated with wire$_i$ and the load that it is driving is C_i then the total switching power dissipated by the circuit should be:

$$P_{sw} = \sum_{i=1}^{n} f_i C_i V_{DD}^2$$

This expression assumes that each wire is switching at its own frequency, f_i. However, in practice, the switching of all the wires will be controlled by a single frequency f_{clk} and each wire will change state *on one* clock edge with a defined probability α_i. In this case, the expression becomes:

$$P_{sw} = f_{clk} V_{DD}^2 \sum_{i=1}^{n} \alpha_i C_i$$

In many cases, it is possible to simplify the expression further by assuming a value for α that is representative for the whole circuit rather than an individual wire.

$$P_{sw} = \alpha f_{clk} V_{DD}^2 \sum_{i=1}^{n} C_i$$

(4)

ii) *Why has reducing the power supply voltage traditionally been the best way to reduce power consumption in CMOS circuits and why is this not likely to be the case in the future?*

As technology scales down V_{DD} has traditionally scaled down at a similar rate (constant field conditions) and the quadratic term has a significant effect on power dissipation. Clearly, C changes as well but not as strongly. However, the need to keep static power dissipation down has required that V_T, the threshold voltage, is kept reasonably high to minimise sub-threshold conduction and, consequently, V_{DD} cannot be scaled as strongly so that the term $1/(V_{DD}-V_T)$, which controls current drive is not compromised too far because this impacts on circuit speed. **(2)**

d. *A 2-input standard-CMOS, minimum-sized NAND gate is part of a clocked circuit and has the following attributes:*

- *Gate capacitance of minimum-sized n-FET = 2fF;*
- *The inputs are essentially independent of each other and each changes state with a probability of 0.25 at each rising edge of the clock;*
- *The system clock is 1GHz;*
- *The power supply voltage is 1.2V;*
- *The output of the gate drives a 10fF load.*

Estimate the power dissipation associated with the NAND gate.

If a signal changes state at a rising edge of clock with a probability of 0.25 then on average it changes state every 4 cycles and there is one cycle every 8 clock cycles. Hence, α=0.125.

Each input sees an *n*- and *p*- FET (all will be width 2) and, hence, the capacitance on each input will be $4C_g$=8fF.

The capacitance on the output is 10fF with an additional $(2C_g+2C_g+2C_g)/2$=6fF due to capacitance seen at the drains of the FETs connected to the output. Consequently, the total capacitance at the output is 16fF.

However, given that the inputs are changing state with a low probability – how often will the output change state?

3 times out of 4, a 0 stays as a 0 and a 1 stays a 1. Looking at the pairs of inputs, and the probability that they will change given the current state:

Current Next	00	01	10	11
00	9/16	3/16	3/16	1/16
01	3/16	9/16	1/16	3/16
10	3/16	1/16	9/16	3/16
11	1/16	3/16	3/16	9/16

Consequently, the conditional probabilities ($P_{new|old}$) for the output states are:

$P_{0|0}$=9/16

$P_{1|0}$=7/16

The probability that the inputs are in each of the 4 states is unchanged because the inputs are still, essentially, random. Consequently P_{O1}, the probability that the output is 1, is 0.75. Hence, $P_{O1}.P_{0|1}$ = 0.25*7/16=0.109 and this is the probability of an output changing from 1 to 0 and hence from 0 to 1. Using the same terms as before, the probability of a transition will be 2*0.109 and, hence, α=0.109

Consequently, the power dissipation is:
Input: 2*0.125*8e-15*1e9*1.2^2=2.88nW
Output: 0.109*16e-15*1e9*1.2^2=2.51nW
Total: 5.39nW **(10)**

3. a. i) *As ICs are scaled down in size what is the particular problem that interconnect presents?*

As ICs scale down the performance of the interconnect (relative to gate delays) does not scale down as quickly. Consequently, the wiring contributes a bigger and bigger delay. Very local, short lines do tend to scale reasonably well but depends on compromises that have to be made with width.v.height to stop resistance rising too quickly. However, longer and global lines that span the entire chip tend to scale very badly because the ICs are not reducing in size. This is particularly the case because the delay of a signal along a wire is proportional to rcL^2 where r is resistance per unit length and c is capacitance per unit length. As the technology shrinks then r will go up as the square of the scaling factor unless the aspect ratio of the wires is changed and c may not scale at all especially if the wires' aspect ratio is changed. **(4)**

ii) *How is the problem due to interconnect normally managed?*

Because the delay along long lines is proportional to L^2, splitting the wire up into sections and adding repeaters reduces the delay.

Tiered interconnect where the multiple levels of metallization are split into 3 groups: the bottom group are minimum-sized and for local, short wires; the middle group are larger and more widely spaced (lower r and c) and intended for intermediate length wires; the top group are larger still and more widely spaced (further lower r and c) and intended for global wires.

Using Cu for interconnect (lower r) and using low k dielectrics between the metallization (lower c). **(4)**

b. *Why might a logic gate within an IC be required to drive a large capacitance?*

Driving a long wire, driving an internal capacitive structure (e.g. part of a sensor), driving an external capacitance via a pad. **(2)**

c. *How is driving a large capacitance in an IC normally accomplished?*

By using a tapered set of buffers. Consider a sequence of N inverters each one k times wider than the previous one. What are the values of N and k that minimise the overall delay from input to output.

Assuming that the first inverter is of minimum size, the i^{th} inverter in the series drives an input capacitance of k^iC_{in} in the following gate and is of width $k^{i-1}W_{min}$. Consequently, delay of this stage is:

$$\frac{4k^iC_{in}}{k^{i-1}\beta(V_{DD}-V_T)} = \frac{4kC_{in}}{\beta(V_{DD}-V_T)}$$ where we model delay as R_oC_{next}

That is, independent of the position in the sequence: the delay of each stage is equal. Thus, the overall delay through N stages is:

$$\frac{4kC_{in}(N-1)}{\beta(V_{DD}-V_T)} + \frac{4C_{load}}{k^{N-1}\beta(V_{DD}-V_T)}$$

where the second term represents the final stage that drives the load capacitance. We can impose the boundary condition that $C_{load} = k^NC_{in}$ and this simplifies the expression for delay to:

$$\frac{4kC_{in}N}{\beta(V_{DD}-V_T)}$$

From the expression for C_{load} we can see that: $N = \dfrac{\ln\left(\dfrac{C_{load}}{C_{in}}\right)}{\ln(k)}$

and the expression for delay becomes: $\dfrac{4kC_{in}}{\beta(V_{DD}-V_T)\ln(k)}\cdot\ln\left(\dfrac{C_{load}}{C_{in}}\right)$

differentiating this expression *w.r.t. k* and setting equal to 0 yields:

$$\frac{4C_{in}}{\beta(V_{DD}-V_T)}\cdot\ln\left(\frac{C_{load}}{C_{in}}\right)\cdot\frac{1}{\ln(k)}-\frac{4C_{in}}{\beta(V_{DD}-V_T)}\cdot\ln\left(\frac{C_{load}}{C_{in}}\right)\cdot\frac{k}{\ln(k)^2}\cdot\frac{1}{k}=\frac{4C_{in}}{\beta(V_{DD}-V_T)}\cdot\ln\left(\frac{C_{load}}{C_{in}}\right)\cdot\left(\frac{1}{\ln(k)}-\frac{1}{\ln(k)^2}\right)=0$$

and from this we can find that:

$$\ln(k)=1$$

$$k=e^1=2.71828$$

Therefore: $k=e^1=2.71828$ and $N=\ln\left(\dfrac{C_{load}}{C_{in}}\right)$

(6)

d. *A signal, inside an IC, is to be used to optimally drive a load of 18pF (i.e. with minimum delay). You know that a minimum-sized inverting buffer has an input capacitance of 6fF and an area 0.2μm x 1.5μm. Design a set of inverting buffers to drive this load. You need to identify the width of the intermediate buffers as a multiple of the width of the minimum-sized inverting buffer and the area occupied by the set of buffers.*

N = ln(18e-12/6e-15) = 8.006

This is convenient because 8 is an even number and so the number of stages needed does not result in an overall inversion. The buffers are as follows:

Buffer	Width	Area (pm)
1	1	0.3
2	2.7	0.8
3	7.4	2.2
4	20.1	6
5	54.6	16.4
6	148.4	44.5
7	403.4	121.1
8	1096.6	329

Total area = 520pm = 22.8μm x 22.8μm. **(4)**

4. *The circuit shown in* ***Figure 4*** *is not a standard CMOS circuit but is from a logic family called Domino Logic.*

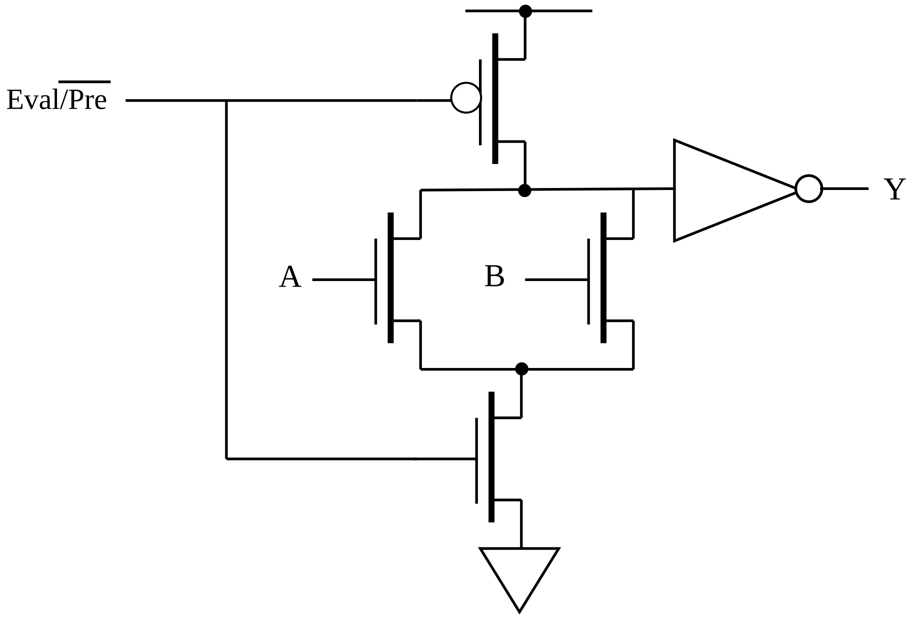


Figure 4: Domino Logic Circuit

The logical output is only valid (the evaluated value) when the $\text{Eval}/\overline{\text{Pre}}$ *input is 1 (Evaluate) and the* $\text{Eval}/\overline{\text{Pre}}$ *input must go low (Precharge) between evaluations. You can assume that the A and B inputs are driven from the outputs of other Domino Logic circuits (which will behave similarly) – all of which share a common* $\text{Eval}/\overline{\text{Pre}}$ *signal.*

a. **i)** *How can you tell that this is not a standard CMOS circuit?*

The pull-up network in a CMOS circuit should be complementary to the pull-down network which it is not. **(2)**

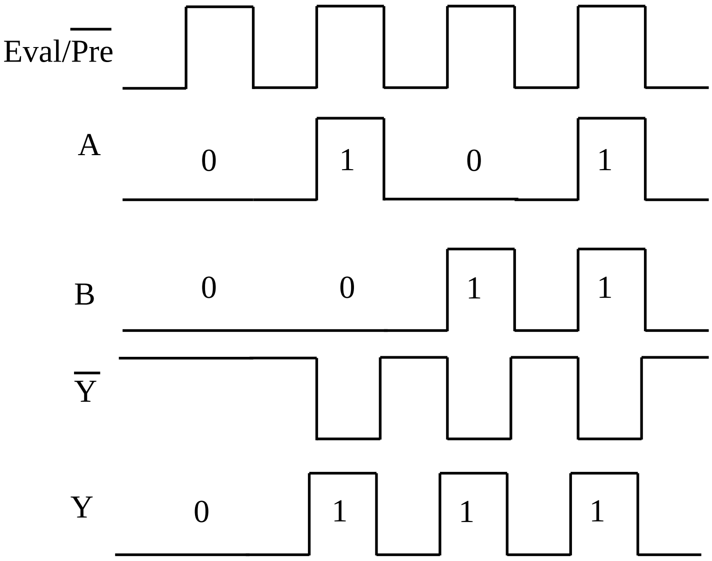
ii) *What happens to Y during the Precharge period?*

During precharge, $\text{Eval}/\overline{\text{Pre}}$ goes to 0 and this means that the pull-down network must be high-impedance and the pull-up network must be low impedance. Consequently, the intermediate node goes high and Y, at the output of the inverter must be low. In turn this must mean that the output of every logic circuit must be low during the precharge period. **(2)**

iii) *What does this, in turn, mean about the behaviour of circuit node at the input of the inverter?*

This means that there are combinations of A, B, and $\text{Eval}/\overline{\text{Pre}}$ that will cause the intermediate node not to be driven i.e. its value depends on the voltage held by the capacitance at the node and this can only be relied on for a short time – i.e. the logic must be *dynamic*. **(4)**

b. *The* Eval/$\overline{\text{Pre}}$ *input is driven by a regular, clock-like signal, draw waveforms showing the behaviour of the circuit for various evaluated values of A and B. Hence determine the function of the circuit in terms of the evaluated values of A and B.*



The truth table relating the evaluated values of A and B to the evaluated value of Y is an OR function – not inverting. **(8)**

c. *What happens when a number of these circuits (with a common* Eval/$\overline{\text{Pre}}$ *signal) are cascaded together (Hint: why is the logic family called Domino Logic).*

During precharge all the output nodes after the inverter go to 0 (corresponding to the intermediate node at the input of the inverters going to 1). When the Eval/$\overline{\text{Pre}}$ signal goes to 1 then if an evaluated value is supposed to be 1 the pull down network will switch on and the node at the output of the circuit will go high. Notice that even after Eval/$\overline{\text{Pre}}$ goes to 1 because all of the output nodes (and hence input nodes to which they are connected) are 0, initially, all of the output nodes remain at 0 (implying that the intermediate node is being held dynamically. As the output of a logic circuit at the beginning of a cascade of circuits goes high, this in turn can cause the output of the next circuit to go high and the effect ripples through a cascade of circuits like a line of dominos falling over. **(4)**

NLS