EECS150: Homework 8, Power and Timing

UC Berkeley College of Engineering
Department of Electrical Engineering and Computer Science

1. You are given a static CMOS **6-input NAND** gate whose output is driving a capacitance $\mathbf{C_{load}}$. The gate is powered by a supply voltage $\mathbf{V_{DD}}$ and is part of a synchronous circuit running at some frequency **f**. Ignore any static (leakage) currents for this problem.

   (a) As a function of the given variables, how much energy is dissipated by the circuit each time the output makes a $0 \rightarrow 1$ transition? What about a $1 \rightarrow 0$ transition? Is any energy used if the output stays the same?

   $\frac{1}{2}C_{load}V^2$ is dissipated by the circuit when charging the load capacitor when the output makes a $0 \rightarrow 1$ transition. Likewise, $\frac{1}{2}C_{load}V_{DD}^2$ is dissipated by the circuit every time the load capacitor discharges on a $0 \rightarrow 1$ transition.

   Since we are ignoring any leakage currents, no energy is used if the output stays the same. This is because we are neither charging nor discharging the load capacitor, meaning that the current flowing through the NAND gate is 0 and no energy is dissipated.

   (b) As a function of the given variables, what is the average power dissipated by this circuit if, on any given cycle, the probability that the output will make either a $0 \rightarrow 1$ or $1 \rightarrow 0$ transition is α?

   The NAND gate disspates $\frac{1}{2}C_{load}V_{DD}^2$ for every transition, whether it be $0 \rightarrow 1$ or $1 \rightarrow 0$. Since it is part of a synchronous circuit running with some frequency **f**, its output could potentially change **f** times every second. The switching probability, α, tells us the chance that the output changes on a given cycle. So, the number of times the output changes every second is αf and the average amount of power (energy/second) that the NAND gate dissipates is:

$$\frac{1}{2}C_{load}V_{DD}^2\alpha f$$

   (c) Suppose each **input** has a 50% chance of switching (either a $0 \rightarrow 1$ or $1 \rightarrow 0$ transition) each cycle. Find a numerical value for α, the probability that the output of the gate switches on any given cycle.

   A NAND gate outputs a 0 only if all of its inputs are 1. Thus, the probability that the output of the NAND gate is 0, on any given cycle is just:

$$P(0) = P(I_0 = 1)P(I_1 = 1)P(I_2 = 1)P(I_3 = 1)P(I_4 = 1)P(I_5 = 1) = (0.5)^6 = \frac{1}{64}$$

   Likewise, the probability that the output of the NAND gate is 1 is:

$$P(1) = 1 - P(0) = \frac{63}{64}$$

   The probability that the NAND gate transitions on a given cycle is the chance that the NAND gate output makes a $0 \rightarrow 1$ transition plus the probability of a $1 \rightarrow 0$ transition. **Clarification: $P(X|Y)$ here means the probability that the output will be X on the NEXT cycle given that the output is Y THIS cycle**:

$$\alpha = P(0) \cdot P(1|0) + P(1) \cdot P(0|1)$$

   To find $P(1|0)$, we can first calculate $P(0|0)$. $P(0|0)$ is true only if **none** of the inputs switch. Thus:

$$P(1|0) = 1 - P(0|0) = 1 - (0.5)^6 = \frac{63}{64}$$

   To find $P(0|1)$ all six inputs have to switch in just the right way to give a 0:

$$P(0|1) = (0.5)^6 = \frac{1}{64}$$

Plugging in the numbers, we get:

$$\alpha = P(0) \cdot P(1|0) + P(1) \cdot P(0|1) = \frac{63}{2048} = .0307$$

Note that, in general, if the switching probability is 50% for each input, the whole analysis simplifies to just:

$$\alpha = P(0) \cdot P(1) + P(1) \cdot P(0)$$

From this, we can note that the power dissipation is also a function of the input pattern; if we used a different sequence of inputs (i.e. different switching probabilities), we would end up with a different power consumption!

(d) Let $\mathbf{C_{load}} = 100fF$, $\mathbf{V_{DD}} = 1.2V$, and $\mathbf{f} = 2GHz$. Using the α you found in the previous part, numerically calculate the average power dissipated by this gate.

Plug and chug:

$$P = \frac{1}{2} C_{load} V_{DD}^2 \alpha f$$

$$P = \frac{1}{2}(100 \cdot 10^{-15})(1.2)^2(0.0307)(2 \cdot 10^9)$$

$$P = 4.43\mu A$$

(e) Suppose instead that we were analyzing a **6-input XOR** gate being subjected to the exact same conditions. Would you expect its average power consumption be bigger, smaller, or equal to that of the **6-input NAND** gate? Why?

An XOR gate under the same conditions would be expected to draw **more** power than the NAND gate. This is because the output of an XOR gate has a considerably higher chance of transitioning on a given cycle ($\alpha = 0.5$ for an XOR under these same conditions).

2. Your boss at **chipsRus.com** asks you to determine the average node activity factor α of a particular chip. In the lab you find:

- A power supply for powering the chip at some particular voltage **V**.
- A clock generator to output some frequency **f**.
- A current meter for determining its average current consumption while running **I**.
- A pattern generator for applying typical input patterns to the chip while running.

Additionally you:

- Call up the designer of the chip and leave voice mail asking her the number of internal nodes in the chip **n**.
- and the average node capacitance **C**.

(a) Derive a formula you can use for finding α.

From the previous problem, we know that the average power used by each node in the chip is $\frac{1}{2} C_{load} V^2 \alpha f$. Thus if the chip is **n** nodes, the average power drawn by the entire chip is just $P = \frac{1}{2} C_{load} V^2 \alpha f \cdot n$

We also know that the average power drawn by the chip can be found by multiplying its supply voltage with its average current, namely $P = I \cdot V$. Thus we can equate these two equations and solve for α:

$$\alpha = \frac{2I \cdot V}{C_{load} V^2 f \cdot n}$$

(b) The designer calls back and says that the chip has 1 Million nodes, and each node has an average capacitance of **10fF**. You go to the lab and measure the average current consumption to be **0.25 A** at **2 V** and **100 MHz**. What is the value of α?

Plug and chug again:

$$\alpha = \frac{2I \cdot V}{C_{load} V^2 f \cdot n}$$

$$\alpha = \frac{2(0.25)(2)}{(10 \cdot 10^{-15})(2)^2 (100 \cdot 10^6)(1 \cdot 10^6)}$$

$$\alpha = .25$$

(c) Suppose you turn off your clock generator and find that the circuit is still drawing a hefty **0.10 A** at **2 V**. Explain what is happening and modify the activity factor α that you found earlier to account for this.

Turning off the clock source (same as setting $\mathbf{f} = 0$) essentially makes the average **dynamic power consumption**, the power consumed due to switching gate outputs, equal to 0. However, the fact that the circuit is still drawing current is a sign that there is a significant amount of **static power consumption**, which is independent of the clock frequency and typically due to transitor leakage currents and . The total power used by the circuit is just the sum of the dynamic and static terms, $P = P_{dynamic} + P_{static}$.

The dynamic power consumption is what we found in the previous problem:

$$P_{dynamic} = \frac{1}{2} C_{load} V^2 \alpha f \cdot n$$

To find P_{static}, we consider the case when the clock source is turned off. The total power drawn by the circuit is still $P = I \cdot V$. Thus, when the clock source is turned off, the power consumption is purely due to static/leakage currents:

$$P = P_{static} = I_{leak} \cdot V_{leak}$$

where I_{leak} and V_{leak} are the current and voltage delivered by the supply when the clock source is turned off.

Now that we have expressions for both the dynamic and static power terms, we can sum them together to get the total average power:

$$P = \frac{1}{2} C_{load} V^2 \alpha f \cdot n + I_{leak} \cdot V_{leak} = I \cdot V$$

Solving for α again:

$$\alpha = \frac{2I \cdot V - I_{leak} \cdot V_{leak}}{C_{load} V^2 f \cdot n}$$

Plugging in numbers:

$$\alpha = \frac{2(I \cdot V - I_{leak} \cdot V_{leak})}{C_{load} V^2 f \cdot n}$$

$$\alpha = \frac{2((0.25)(2) - (0.10)(2))}{(10 \cdot 10^{-15})(2)^2 (100 \cdot 10^6)(1 \cdot 10^6)}$$

$$\alpha = .15$$

3. Consider a CMOS AND gate implemented as a NAND gate followed by an inverter. Assume the inverter propagation delay is defined as follows (units in picoseconds):

$$\tau_p = 50 + 100 \cdot f$$

Where **f** is the fanout of the inverter, expressed in number of transistor gate inputs. For example, inverters contribute 2 to **f** and one input of a 2-input NOR gate contributes 2. **Note to people who have taken EE141: this is not the $f = \frac{C_{out}}{C_{in}}$ definition of fanout used in EE141**. Assume this inverter has the same propagation delay for both $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions.

The NAND gate propagation delay is expressed as (in ps):

$$\tau_{p0 \rightarrow 1} = 100 + 75 \cdot f$$

$$\tau_{p1 \rightarrow 0} = 100 + 125 \cdot f$$

For the $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions, respectively. Write the expressions for the $0 \rightarrow 1$ and $1 \rightarrow 0$ propagation delays of the AND gate.

The first-stage NAND gate is always driving the inverter, which adds a constant 2 to the fanout of the NAND gate. The delay through the first-stage NAND gate is just:

$$\tau_{p_{NAND}0\rightarrow1} = 100 + 75 \cdot 2 = 250$$

$$\tau_{p_{NAND}1\rightarrow0} = 100 + 125 \cdot 2 = 350$$

Delay of the inverter depends on whatever fanout the composite AND gate is driving, thus it will be a function of **f**:

$$\tau_{p_{INV}} = 50 + 100 \cdot f$$

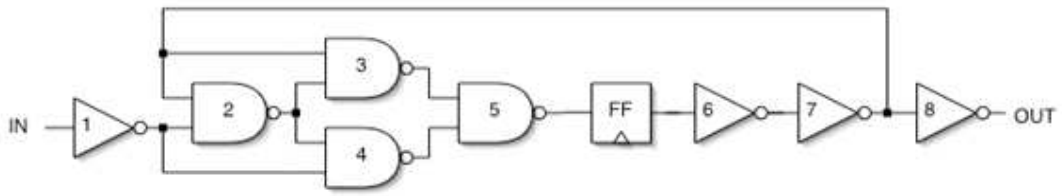
The delay of the composite AND gate is just the sum of the delay through the NAND gate and the delay through the inverter. Note, however, that the delay for a $0 \rightarrow 1$ transition will be different from that of a $1 \rightarrow 0$. Also, beware of the inverting properties of the inverter, which is the reason why $\tau_{pNAND0\rightarrow1}$ contributes to the delay of a $1 \rightarrow 0$ transition rather than the delay of a $0 \rightarrow 1$ (and vice versa).

$$\tau_{p0\rightarrow1} = \tau_{p_{NAND}1\rightarrow0} + \tau_{p_{INV}} = 400 + 100 \cdot f$$

$$\tau_{p1\rightarrow0} = \tau_{p_{NAND}0\rightarrow1} + \tau_{p_{INV}} = 300 + 100 \cdot f$$

4. Consider the circuit given in Figure 1.

Figure 1 Interesting Circuit



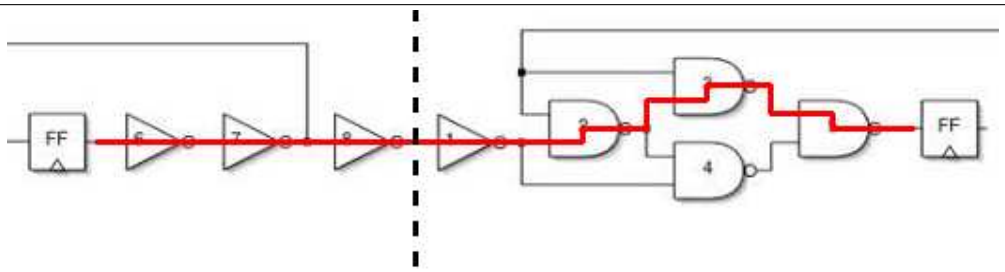
- The propagation delay (for both high-to-low and low-to-high transitions) of the inverters are $\tau_p = 50 + 100 \cdot f$ (in ps).
- The propagation delay (for both kinds of transitions) of the NAND gates are $\tau_p = 100 + 150 \cdot f$ (in ps).
- The flip-flop $t_{setup} = t_{clk-q} = 50ps$.
- There is no clock skew
- There are three instances of this circuit cascaded together, we are focusing our analysis on the middle one.

(a) Mark the critical path in the diagram
See Figure 2.

(b) List the gates of the critical path (by gate number) in the order of signal propagation and their associated delays. Remember to account for the fanout.
As per the newsgroup post, the flip-flop contributes a fanout of 2.
The order of the gates and their delays are:

i. Gate 6, $50 + 100 \cdot 2 = 250$ *ps*

Figure 2 Critical path of the circuit is shown in red. Note that the path is from the flip flop of the middle instance to the flip flop of the last instance. The dashed line represents where the critical path crosses boundary from the middle instance to the last instance



ii. Gate 7, $50 + 100 \cdot 6 = 650\ ps$
iii. Gate 8, $50 + 100 \cdot 2 = 250\ ps$
iv. Gate 1, $50 + 100 \cdot 4 = 450\ ps$
v. Gate 2, $100 + 150 \cdot 4 = 700\ ps$
vi. Gate 3, $100 + 150 \cdot 2 = 400\ ps$
vii. Gate 5, $100 + 150 \cdot 2 = 400\ ps$

(c) What is the minimum clock period **T** for correct operation of this circuit?
Remember that $T > t_{clk-q} + t_{CL} + t_{setup}$

t_{CL} is just the sum of the delays we found in the previous part.

$$t_{CL} = 250\ ps + 650\ ps + 250\ ps + 450\ ps + 700\ ps + 400\ ps + 400\ ps = 3100\ ps = 3.1\ ns$$

$$t_{clk-q} = t_{setup} = 50\ ps = 0.05\ ns.$$

$$T > 0.05 + 3.1 + 0.05 = 3.2\ ns.$$

The frequency is just $\frac{1}{T}$, thus the maximum clock frequency = 312.5 MHz.

5. Consider an **n-bit ripple-carry adder** implemented using the full-adder circuit given in DDCA **Figure 4.8**.

(a) Find the critical path through this adder and the input combination that triggers it.
The worst-case delay in a ripple-carry adder occurs when a carry-out is generated (or killed) in the lowest bit and its value propagated all the way to the highest bit. One such input combination is `111...111 + 000...001`. There are many other possibilities for this, such as `10101...0101 + 01010...1011` or `11001100...0011 + 00110011..1101`.

(b) Count the number of gate delays (i.e. if a signal passes through an XOR gate, add 1 to its gate delay) in the critical path as a function of **n**.
To do this, we simply calculate the the number of gate delays from each input to each output. For the full-adder in DDCA **Figure 4.8**:

$$\tau_{p_{A \to C_{out}}} = \tau_{p_{B \to C_{out}}} = 3$$

$$\tau_{p_{A \to S}} = \tau_{p_{B \to S}} = 2$$

$$\tau_{p_{C_{in} \to S}} = 1$$

$$\tau_{p_{C_{in} \to C_{out}}} = 2$$

In an n-bit ripple-carry adder, the critical path consists of a single $\tau_{p_{A \to C_{out}}}$ for the first carry generate/kill, $\tau_{p_{C_{in} \to C_{out}}}(n-1)$ as the carry propagates from the full adder at the least

significant bit to the full adder at the most significant bit, and a $\tau_{p_{C_{in} \to S}}$ for a carry arriving at the most significant bit to change the output S.

Thus the total delay is:

$$\tau_p = \tau_{p_{A \to C_{out}}} + (n-1) \cdot \tau_{p_{C_{in} \to C_{out}}} + \tau_{p_{C_{in} \to S}}$$

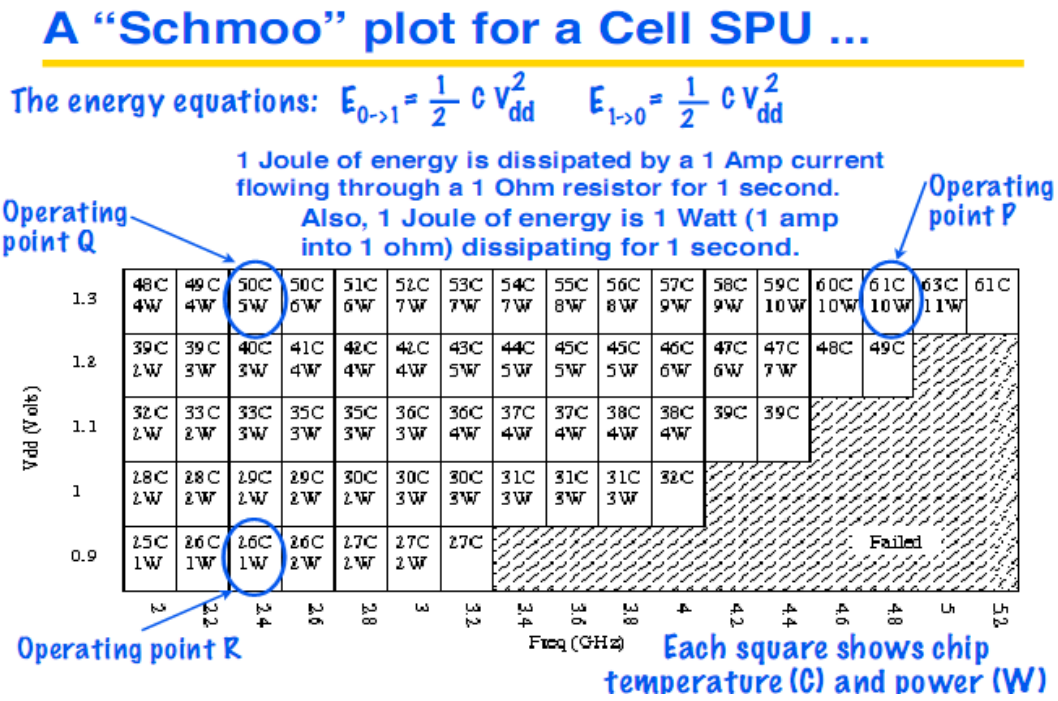
$$\tau_p = 3 + 2(n-1) + 1$$

$$\tau_p = 4 + 2(n-1)$$

Note that the delay grows linearly with the number of bits! Later on, we will be learning about different adder architectures, some of which can break break this linear dependence.

6. Figure 3 shows a Schmoo plot for a processor.

Figure 3 Schmoo plot of a processor



In this problem, the processor characterized by the Schmoo plot is used in a system along with support components that use K Watts of power. For example, in a laptop, the support chips might consume 2 Watts. For this system, K = 2.

When the processor is running, the support components must stay on. A CPU instruction may be used to turn the processor and the support components off. When off, the processor and the support components both use no power at all.

(a) Different systems may have different values of K. For example, the support chips for a laptop design may consume 2 Watts (K = 2), while support chips for a desktop design may consume 7 Watts (K = 7).

A program runs twice as fast at Operating Point P (shown in Figure 3) than at Operating Point Q. The last instruction of the program turns off power to the processor and its support chips.

i. For what range of values for K does Operating Point P use the lowest amount of energy to run the program?

ii. For what range of values for K does Operating Point Q use the lowest amount of energy to run the program?

iii. For what range of values for K do Operating Points P and Q use the same amount of energy?

We assume that operating point Q takes t seconds to run, and operating point P takes $\frac{t}{2}$ seconds to run. Given this definition, and the numbers in the Schmoo chart, we deduce:

$$\text{Total P Energy} = (\tfrac{t}{2})(K + 10)$$

$$\text{Total Q Energy} = (t)(K + 5)$$

For part (i) for this question, we solve the inequality:

$$\text{Total P energy} < \text{Total Q energy}$$

$$(\tfrac{t}{2})(K + 10) < (t)(K + 5)$$

$$(K + 10) < (2K + 10)$$

$$K < 2K$$

Since for all positive K this inequality holds, the answer to (i) is "all K greater than 0." Using the same technique, we discover the answer to (ii) is "never" (as K would need to be negative, which is impossible, unless we have an energy source), and the answer to (iii) is "if K is equal to 0."

(b) A program runs twice as fast at Operating Point P than at Operating Point R. The last instruction of the program turns off power to the processor and its support chips.

i. For what range of values of K does Operating Point P use the lowest amount of energy to run the program?

ii. For what range of values of K does Operating Point R use the lowest amount of energy to run the program?

iii. For what range of values for K do Operating Points P and R use the same amount of energy?

We assume that operating point R takes t seconds to run, and operating point P takes $\frac{t}{2}$ seconds to run. Given this definition, and the numbers in the Schmoo chart, we deduce:

$$\text{Total P Energy} = (\tfrac{t}{2})(K + 10)$$

$$\text{Total R Energy} = (t)(K + 1)$$

For part (i) for this question, we solve the inequality:

$$\text{Total P energy} < \text{Total R energy}$$

$$(\tfrac{t}{2})(K + 10) < (t)(K + 1)$$

$$(K + 10) < (2K + 2)$$

$$K > 8$$

Thus, the answer to (i) is "all K greater than 8." Using the same technique, we discover the answer to (ii) is "all K less than 8," and the answer to (iii) is "if K is equal to 8."