Written examination for

MCC092 Introduction to Integrated Circuit Design

Thursday October 26, 2017, at 8.30-13.30 at Hörsalsvägen lecture halls

Staff on duty: Lena Peterson, D&IT, phone ext: 1822, or mobile 0706-268907. Lena will visit around 9.30 and 12.00.

Administration: Send exams to Lena Peterson CSE dept. Send lists to CSE student administration office.

Allowed technical aids for students: This is a closed-book exam. Allowed aids: A Chalmers-allowed calculator (non graph-drawing) plus pencil, eraser, ruler, and dictionary (these are always allowed).

Results: The results from the examination will be sent to you via the Ladok system within three weeks. The grading review will take place Thursday November 16, 12.00-13.00 in room 4128 at the CSE department.

Solutions: Solutions will be posted on the course web site in PingPong no later than Friday October 27. Any student who does not have access to the 2017 course web site can contact Lena Peterson (via e-mail to lenap@chalmers.se) to obtain the solutions.

Instructions:

- Write legibly.
- State any assumptions you make.
- Explain your reasoning and calculations (except when the problem says otherwise). Partial credits can be awarded, but if we do not understand you that is not possible.
- Number all pages and write your code on each page.
- Put only one problem per page.
- Do not write in red since that is the color used for the grading.

Good luck!

Grades:

The written examination contains six problems, each worth 10 points. You need 30 points to pass (grade "3"), at least 40 points for grade "4" and at least 50 points for grade "5". Bonus points from the fall 2016 course instance will be added before the higher grades are assigned.

Problem 1: Sequencing and metastability. **NOTE! If you are taking this exam for the old course, MCC091, you can select to solve problem 7 instead of this problem.**

a) What is metastability? Why is it a problem in clocked digital systems? Describe at least two negative consequences. (4 p)

b) In a sequential system with flip-flops what is a setup violation? Describe under what conditions for the combinational logic between consecutive flip-flips, a setup violation occurs. No calculations required - we want a description in your own words! (3 p)

c) In a sequential system with flip-flops what is a hold violation? Describe under what conditions for the combinational logic between consecutive flip-flips, a hold violation occurs. No calculations required - we want a description in your own words! (3 p)

Problem 2: Tapered buffers and clock drivers

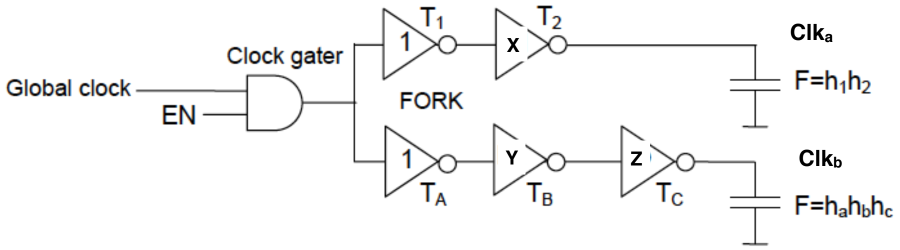


Figure 1: A clock fork for gating a clock signal, with two clock phases: Clk_a, and its inverse, Clk_b.

Consider the clock gater circuit with a fork shown in Figure 1. Shoji[1] has shown that if the delays of the second inverter in each path are equal ($T_2 = T_B$), but not necessarily equal to the other delays, the two paths will have equal delay in all four process corners (that is, for all worst-case fabrication variations).

Your task is to determine the optimum tapering factors h_a, h_b, and h_1, so that you both minimize the path delays and satisfy the constraints of equal delay in both paths, given that $h_2 = h_b$. The related inverter sizes are X for inverter 2, Y for inverter B and Z for inverter C, as shown in Figure 1. Assume $p_{\text{inv}} = 1$. (10 p)

Problem 3: Logical function. logical effort and layout

Figure 2 shows the circuit schematic for a cell that can be used in a valency-3 PG (that is, dot-operator) cell for a prefix adder.

a) What is the logical function for the cell output Y? (2 p)

b) For the compound gate calculate the logical effort, for the input G1, g_{G1}, and the parasitic delay, p. Do not include the inverter, just the compound cell! You can assume that $p_{\text{inv}} = 1$. (2 p)

c) Is it possible to lay out the compound gate as drawn in the schematic in Figure 2 using continuous-line-of-diffusion for both the n-net and the p-net?Motivate. (2p)

d) Complete the layout of this cell shown in the template in 3 where the n-net has already been completed. Note that for simplicity all transistors have the same widths in the template, although that may not be optimal. Identify the five inputs and mark them clearly in the template. Then draw the required connections for the p-net. Merge as many p-diffusion areas as possible to simplify your layout. The template is repeated twice on tear-off sheets at the end of the exam. (4 p)

[1] M. Shoji, "Elimination of process-dependent clock skew in CMOS VLSI," in IEEE Journal of Solid-State Circuits, vol. 21, no. 5, pp. 875-880, Oct 1986.

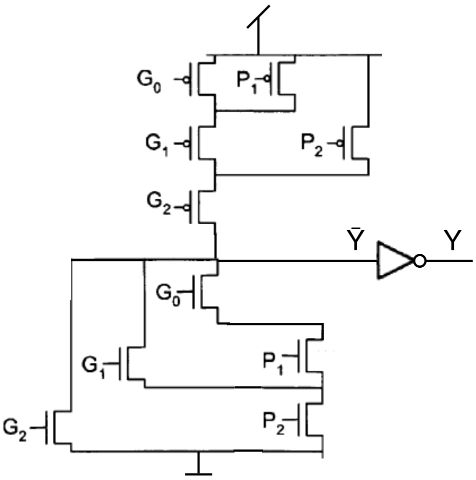


Figure 2: The schematic for a five-input cell that can be used in a three-input PG (dot-operator) cell.

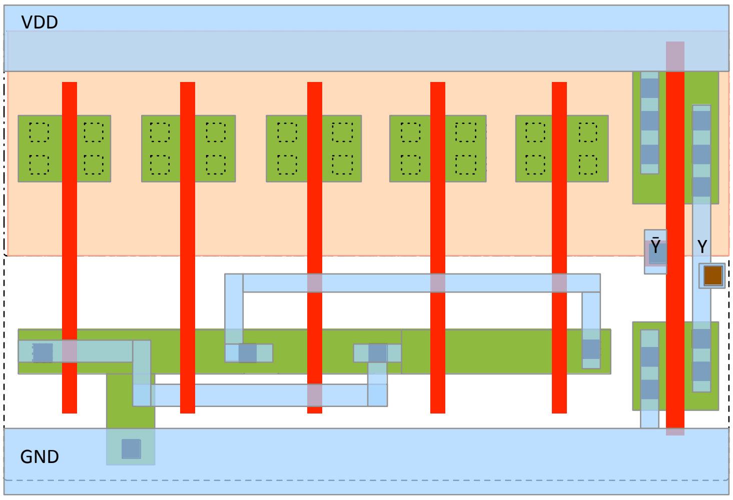


Figure 3: The template for the layout of the cell in Figure 2. The template is repeated twice at the end of the exam for your convenience.

Problem 4: Static and dynamic power consumption

An efficient way of saving **switching** power is to decrease V_{DD}. One way of achieving this is by decreasing V_{DD} for existing hardware, while also decreasing the clock frequency, which results in performance loss, that is less computations performed per time unit. The loss of performance can (at least ideally) then be made up by using more hardware (for example as is done in multicore processors). Another option is to use similar hardware that has been fabricated in a scaled down CMOS process, where V_{DD} is scaled down. In this problem we assume so called Dennard scaling, where all voltages and physical dimensions are scaled down by the same scale factor.

a) Which of the two solutions outlined above for saving switching power is preferable when it comes to the **static** power consumption? For this comparison assume that in both cases the supply voltage has been decreased by 50 %, from 1.5 V to 0.75 V, and that for the first case, with existing hardware, four times as much hardware has to be used to make up for the lower clock frequency. Also assume that the threshold voltages are 25 % of V_{DD}. (3 p)

b) One way of limiting the static power consumption is to execute as fast as possible when required and then turn the supply voltage off when execution is not needed. For the scaled down option outlined in task a) we could double the clock frequency and then turn the power off half of the time (so called power gating). A

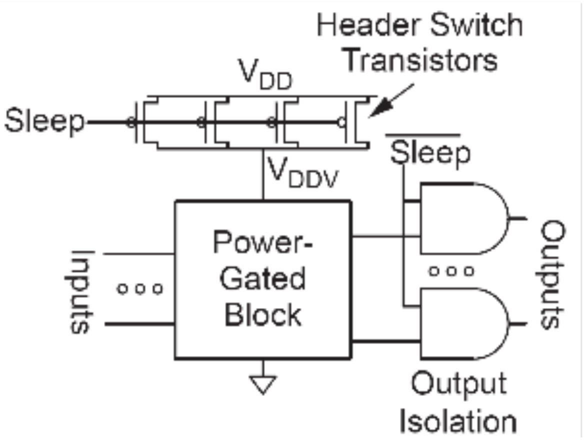


Figure 4: The concept of power gating shown in a figure from Weste & Harris.

conceptual view of power gating is shown in Figure 4 from Weste & Harris.

Here your task is to design the pMOS header switch transistor for hardware in the scaled-down process from task a). Assume that a voltage drop of 5 % of V_{DD} is the maximum we can allow. Also assume that the average power consumption of the scaled down hardware is 2 W and that in the process used the pMOS transistors have an ON resistance of R = 2 kΩµm. How wide does the pMOS header switch have to be? (In practice it may be implemented as many transistors in parallel - here we want the total width.) (4 p)

c) If you have an application that needs to be executed for 50 µs every 100 µs, what will be the power consumption of the power gating? Estimate this power assuming that the tapered buffer that drives the *Sleep* signal in total has half the transistor width of the header switch itself. Assume a transistor gate capacitance of 1 fF/µm. (3 p)

Problem 5: Wire delay, clock distribution

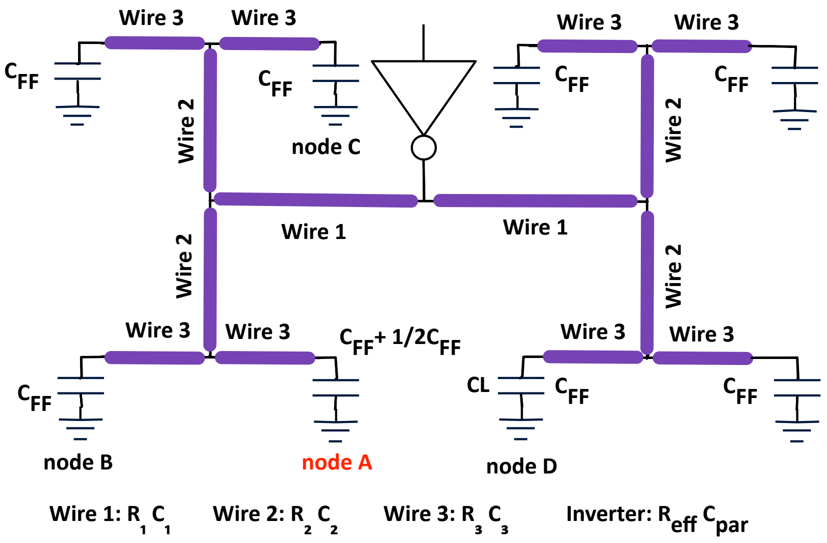


Figure 5: The schematic for a three-level H-tree for the Neil processor clock tree.

You are the designer of the clock H-tree for the new processor in the ArmStrong family called Neil. The Neil processor is quite minimalistic and has no repeaters in the clock tree. Therefore you have worked hard with the engineering team to get the clock tree completely balanced and you have succeeded. See Figure 5 for a schematic picture of the clock tree.

Now, your colleague Kim tells you that there has been some last minute changes in the design of the pipelining, with quite a few additional flipflops added, which will increase the capacitive load at node A of the clock tree with 50 percent, as indicated in Figure 5. The Neil processor must soon go into fabrication so you need to quickly assess the impact of this change. Fortunately, the driver of the clock tree is designed such that this increase in the total capacitance will not require a redesign. But can the processor still be clocked as fast as was specified, or do you now risk malfunction?

a) Find **expressions** for the clock skews between node A and each of its adjacent nodes: B, C and D in the Neil processor clock tree. To which of the threes adjacent nodes does node A have the largest clock skew? (6 p)

b) With the new pipelining and the current clocking requirements you have a clock skew budget of 15 ps. Will the maximum clock skew be larger than the budget with these data for the wires, the inverter and the flip-flop capacitances: $C_{FF} = 500\,\text{fF}$, Wire 1: $R_1 = 50\,\Omega$, $C_1 = 2\,\text{pF}$, Wire 2: $R_2 = 25\,\Omega$, $C_2 = 1\,\text{pF}$ Wire 3: $R_3 = 12.5\,\Omega$, $C_3 = 0.5\,\text{pF}$, and for the inverter: $R_{eff} = 2\,\Omega, C_{par} = 3.6\,\text{pF}$? (2 p)

c) What can you do to alleviate your clock-skew problem? Suggest one thing you can do that will help at least locally (that is, it will decrease the clock skew to neighboring h-tree nodes). Complete redesigns are not possible at this late stage in the design process. (2 p)

Problem 6: Faster adders

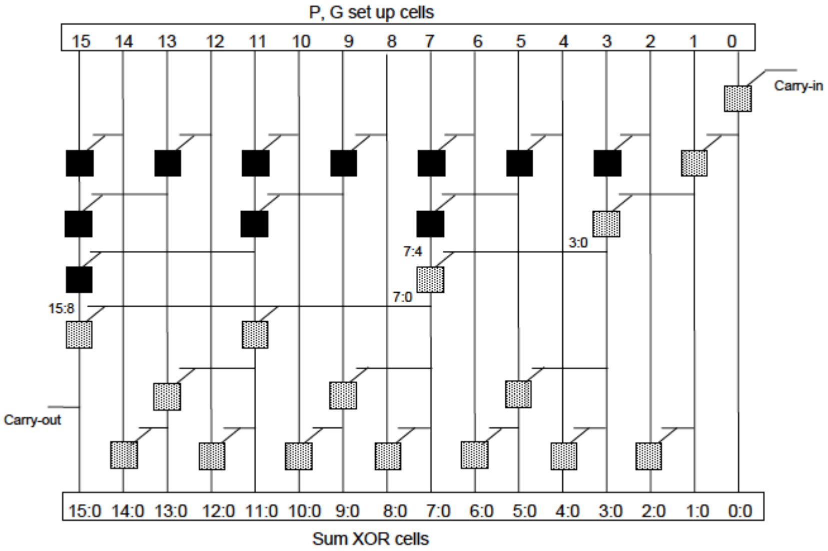


Figure 6: A 16-bit Brent-Kung prefix adder cell with carry-in signal.

Assume that you have a 16-bit Brent-Kung adder available in your cell library; see Figure 6. The 16-bit Brent-Kung adder module is characterized by its set-up delay, t_{pg}, its summation delay, t_{XOR}, and the delay of the PG cells (also called dot-operator cells) in the internal prefix adder tree, t_{AO}. You now intend to construct a 64-bit adder by cascading four such Brent-Kung adders as shown in Figure 7.

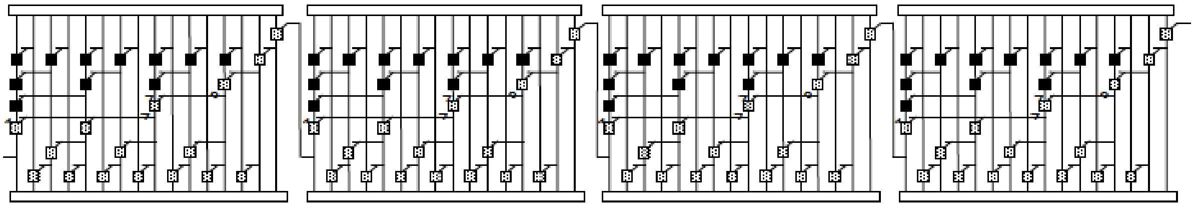


Figure 7: A 64-bit adder constructed from four cascaded 16-bit Brent-Kung adders.

a) In the 64-bit adder the output bits are numbered from 0 (least significant bit) to 63 (most significant bit). Which sum bit will be the last one to complete? What would then the propagation delay of the 64-bit adder be, expressed in the delays given above? (4 p)

b) What if you wanted to also calculate the block propagate signal for the Brent-Kung 16-bit adder. What would you need to modify in the 16-bit adder cell shown in Figure 6? The figure is repeated on a tear-off sheet at the end of the exam - indicate your proposed changes on that sheet and hand it in with you solutions. (2 p)

c) How could you use this added block-propagate signal of the 16-bit Brent-Kung adder to decrease the propagation delay of a 64-bit adder made up of four of the 16-bit Brent-Kung adders? Assume that you have additional PG cells available that you could use for this purpose. (4 p).

BONUS QUESTION

d) How large is the improvement in propagation delay of your proposed solution in task c), compared to the delay calculated in task a)? (3 p)

Problem 7: Inverters and analog aspects **This problem is only for students who are taking this exam for the old course MCC091**

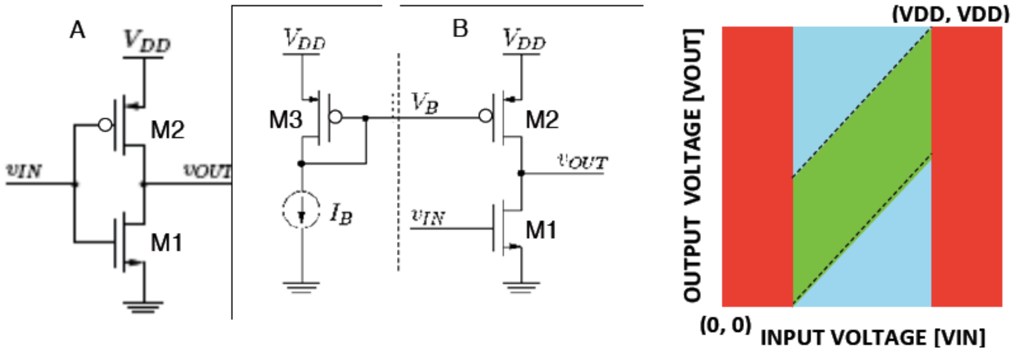


Figure 8: Two analog amplifiers. A is a regular CMOS inverter B is a pseudo-NMOS inverter intended as an amplifier.

In Figure 8 you find the circuit diagram for two CMOS inverters, (A) and (B). The inverter to the left, (A), is a regular CMOS inverter that switches at $V_{\mathrm{DD}}/2$. The inverter to the right, (B), is a pseudo- NMOS inverter intended as an amplifier. Its load p-channel MOSFET, M2, is biased at an unknown gate voltage, VB, determined by a current mirror. A current mirror takes current I_B from a constant-current source and mirrors it to the inverter. Otherwise, the two CMOS inverters are identical except for the biasing. Transistors M1 and M2, respectively, are the same MOSFETs in both inverters. The rightmost diagram in the figure above shows the CMOS inverter regions of operation.

a) Relate the two current gain factors k1 and k2 of MOSFETs M1 and M2 to each other considering that inverter (A) switches at $V_{\mathrm{DD}}/2$ assuming symmetrical threshold voltages, $V_{TN} = -V_{TP} = V_{\mathrm{DD}}/5$? (2 p)

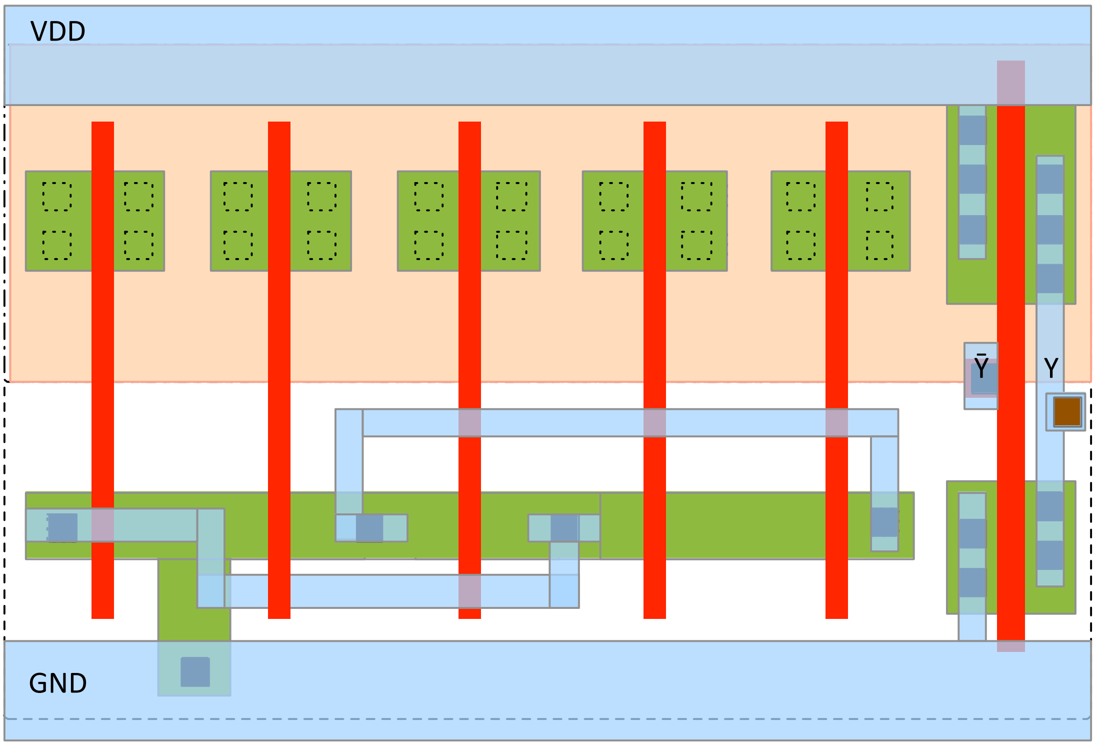
b) Assuming $V_B = 0.6V_{\mathrm{DD}}$, what is the switching voltage of the pseudo-NMOS inverter? (2 p)

c) Calculate current I_B if $V_{\mathrm{DD}} = 1.2\,\mathrm{V}$, and $k = 600\,\mu\mathrm{A/V}^2$. (2 p)

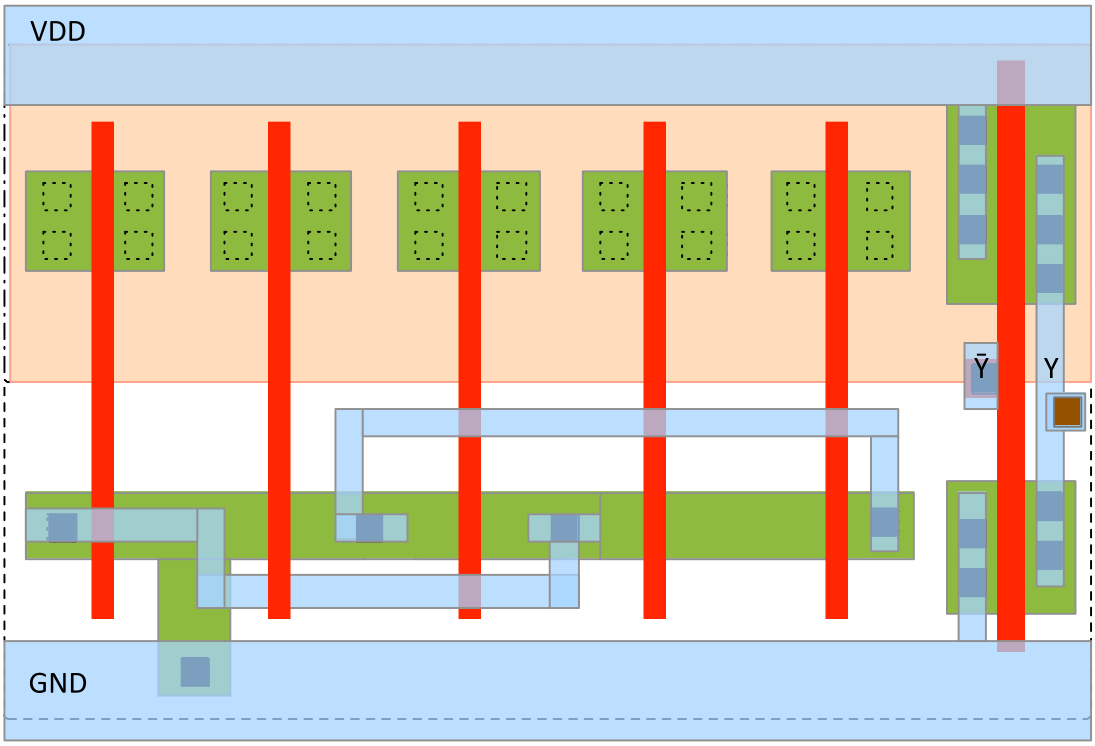
d) For what output voltage range are both MOSFET devices saturated in the CMOS inverter (A)? Refer to the right-hand diagram showing the CMOS regions of MOSFET operation. (2 p)

e) For what output voltage range are both devices saturated in the pseudo-NMOS inverter? Refer to the right-hand diagram showing the CMOS regions of MOSFET operation. (2p)

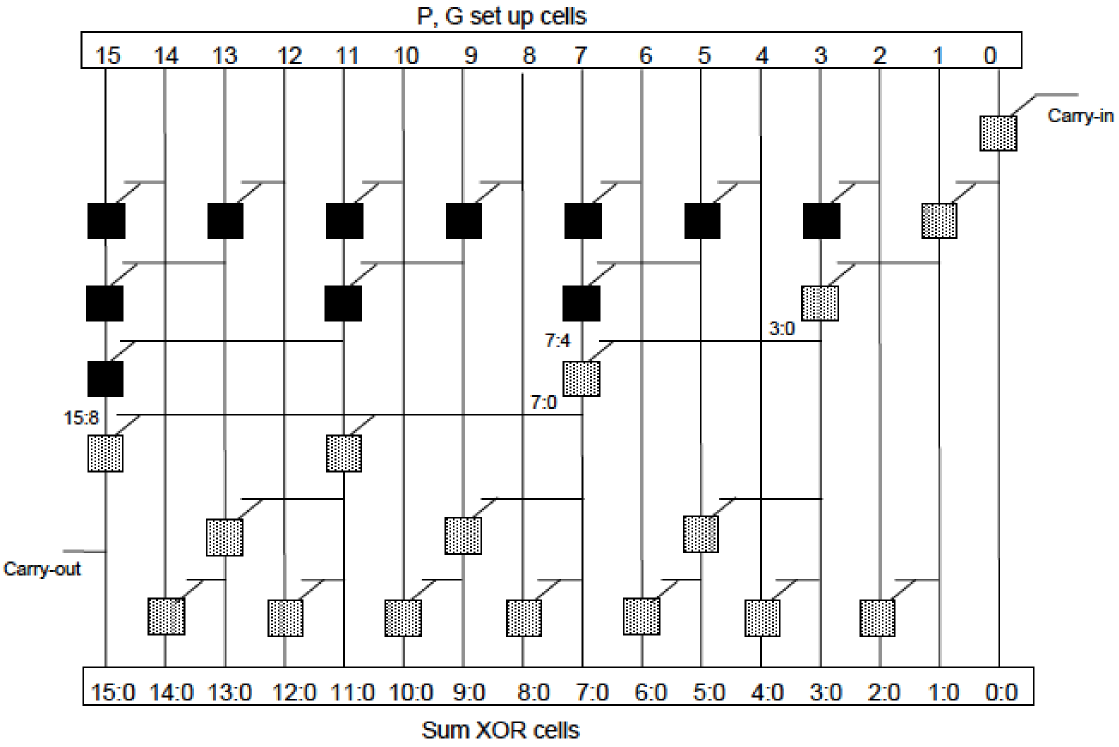
MCC092 2017-10-26 Tear-off page for task 3 d). Write your anonymous code here:



MCC092 2017-10-26 Tear-off page for task 3 d). Write your anonymous code here:



MCC092 2017-10-26 Tear-off page for task 6 b). Write your anonymous code here:



Solution for written examination for

MCC092 Introduction to Integrated Circuit Design, October 26, 2017

Solution to problem 6 d) updated 2018-10-29

The solution for problem 7 is still missing.

Solution 1: Sequencing and metastability

a) Metastability is when a system is temporarily stuck in a meta-stable state that is not globally stable. For example a system with two cross-coupled inverters, such as the ones we have in flip-flops, can be metastable in a point where both inverters have an input around VDD/2. Eventually, the system will move to a state where one voltage is 0 and the other one is VDD. The main problem with metastability is that **it can take infinitely long time for the system to leave the metastable state and go to the stable state**. In the meantime erroneous values can be read.

   Two problem: As hinted above a system can read a non-valid digital value when for example a flip-flop is in a metastable state. This can most easily happen when we try to synchronize an incoming signal that is not synchronized with the internal system clock. Another problem is that we alleviate the problem with multiple flip-flops at the input the uncertainty in value becomes an uncertainty in time. Thus, the internal system has to be designed to allow for longer times before reading the incoming values.

b) A **setup violation** occurs when the result from the combinational logic between two consecutive flip-flops is not available in time for consecutive flip-flop to securely lock the result. (One result may be that the flip-flop enters metastable state!). In terms of characteristics of the combinational logic the **propagation delay is too long**.

c) A **hold violation** occurs when the result from the combinational logic between two consecutive flip-flops disappears (is overrun by the next computation) before the consecutive flip-flop has locked the result. In terms of characteristics of the combinational logic the **contamination delay is too short**.

Solution 2: Tapered buffers and clock drivers

The delays in the two paths after the fork should be the same. That means that our goal is to make:

$$T_1 + T_2 = T_a + T_b + T_c$$

Expressed with normalized delay we can write the same relationship as:

$$h_1 + h_2 + 2p_{inv} = h_a + h_b + h_c + 3p_{inv}$$

where h_i is the electrical effort for inverter i. Since we know that we should also have $h_2 = h_b$ we can simplify the constraint to:

$$h_1 = h_a + h_c + p_{inv} \tag{1}$$

For minimum delay we also need to optimize the paths so that:

$$F = h_a \times h_b \times h_c = h_1 \times h_2$$

Since we have to have $h_2 = h_b$, this relationship can be expressed as:

$$\frac{F}{h_2} = h_a \times h_c = h_1 \tag{2}$$

For minimum delay we should select the stage efforts equal, that is have $h_a = h_c$. We now call this electrical effort h_{opt}. We can eliminate h_1 using equations (1) and (2) resulting in this quadratic equation:

$${h_{\text{opt}}}^2 - 2h_{\text{opt}} - p_{inv} = 0$$

The solution for this equation is

$$h_{\text{opt}} = 1 + \sqrt{1 + p_{inv}}.$$

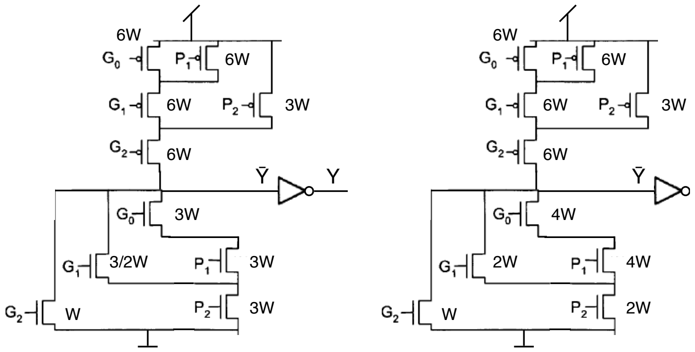
Thus, with $p_{inv} = 1$ we arrive at the solutions:

$$\begin{aligned} h_a = h_c &= 2.4 \\ h_1 &= 5.8 \\ h_2 = h_b &= \frac{F}{5.8} \end{aligned}$$

Obviously, it does not work to use these tapering factors if the load capacitance, F, is very large because then h_2 becomes too large. With F in the range 20 to 30 it should work reasonably well.

The problem did not ask for the inverter sizes, but for completeness we calculate them too. We find $X = h_1 = 5.8$, $Y = h_a = 2.4$ and $Z = h_b \times Y = h_b \times h_a = F\frac{2.4}{5.8} = \frac{F}{2.4}$.

Solution 3: Logical function, logical effort and layout



(a) $g_{G1} = 5/2$, $p = 23/6\ p_{inv}$

(b) $g_{G1} = 8/3$, $p = 13/3\ p_{inv}$

Figure 9: Two possible scalings for equal worst-case resistance and the corresponding values for g_{G1} and p.

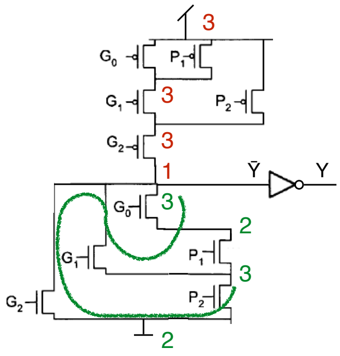


Figure 10: Schematic with the number of transistors marked for each net, with red numbers for the p-net and green for the n-net. The Euler path used for the n-net in the layout template is indicated in green.

a) The logical function is: $Y = G_2 + P_2(G_1 + P_1 G_0)$ which, since the output is inverted by the inverter in the cell, can be found be inspecting the n-net of the compound gate.

b) There are some different possibilities for how to scale the transistors to get the same worst-case resistance. The two most likely scalings of the schematic are shown in Figure 9 with the corresponding values for g_{G1} and p. Solution (a) is probably the one most students have done. Here is that derivation of the logical effort in detail:

$$g_{G1} = \frac{\frac{3}{2}W + 6W}{3W} = \frac{\frac{3+12}{2}}{3} = \frac{5}{2}.$$

And for the parasitic delay we have this derivation:

$$p = \frac{W + \frac{3}{2}W + 3W + 6W}{3W} = \frac{\frac{2+3+6+12}{2}}{3} = \frac{23}{6}.$$

c) It is not possible to lay out the p-net with continuous-line-of-diffusion. We can easily show this by counting the number of transistors that are connected to the circuit nodes in the p-net. See Figure 10. The requirement for an Euler path to exist is that there can be either zero or two circuit nodes with an odd number of transistors connected to them. In the p-net for the compound gate all four nodes have an odd number of connected transistors. The n-net is possible since there are two nodes with an odd number of transistors.

d) The Euler path for the supplied layout template is shown in Figure 10. One example of the complete layout drawn in the supplied template is shown in Figure 11. Once you know where the gap in the p-net is placed you can compact the layout. The result of such a compaction is shown in 12. To compact the layout was not required in the exam, of course.

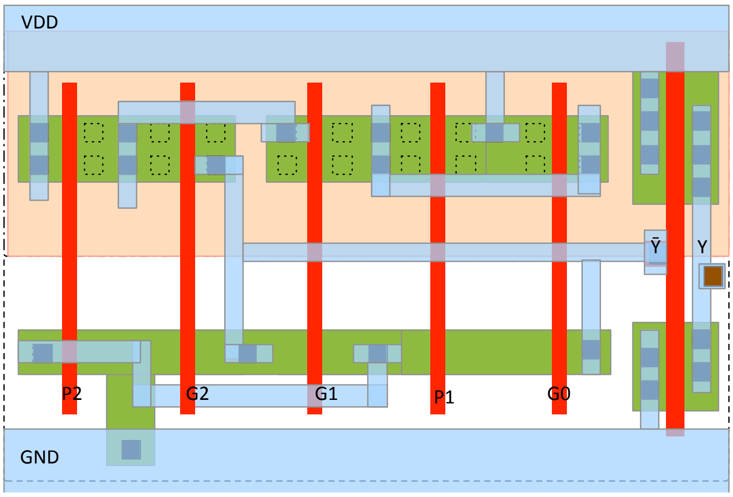


Figure 11: One solution for task 3 d. The cell can, of course be compacted once we know how where the gaps in the diffusion are for the p-net.

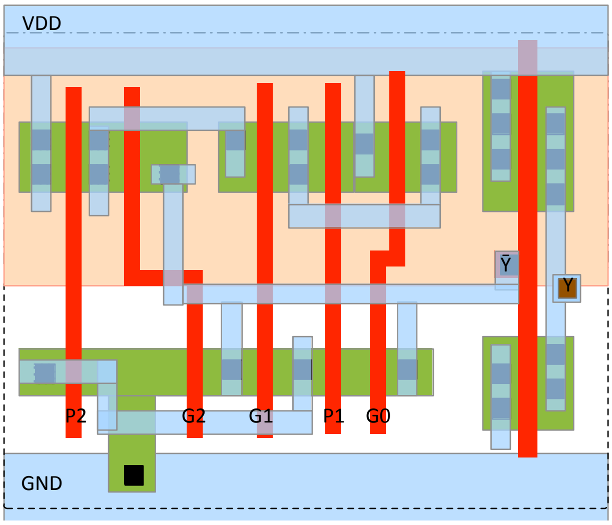


Figure 12: The compacted version of the layout for task 3 d shown in Figure 11.

Solution 4: Static and dynamic power

a) We start by investigating the power dissipation due to transistor sub-threshold leakage; that is the current that flows when a transistor is nominally off. This power is expressed as $P_{\text{leakage}} = I_{\text{leakage}} \times V_{\text{DD}}$. In both cases we have $V_{\text{DD}} = 0.75\,\text{V}$, so any difference in leakage power has to be due to difference in the leakage current.

   In the first case we achieved this lower V_{DD} by decreasing V_{DD} for a processor fabricated in a process with nominal $V_{\text{DD}} = 1.5\,\text{V}$. The threshold voltages in this process are around 0.375 V. In the second case, the processor is fabricated in a process where the threshold voltages are also scaled down with V_{DD} to around 0.1875 V. This decrease in threshold voltages with almost 200 mv causes more than a ten-fold increase in the leakage currents (100 mV gives around a factor ten). Even though in the first case we have four times as many transistors that will leak, it is still not enough to trump the much higher leakage current in the scaled down process. So the worse case for sub-threshold leakage is the scaled-down processor.

   So what about gate leakage? It could also play a part. (We do not require you to consider gate leakage in your solutions, but for completeness we do so here). Gate leakage current is increased when the gate oxide thickness is scaled down. The current is due to tunneling so the scaling is not linear - once the thickness is small enough the tunneling increases drastically. Here we should expect the current to at least double for the scaled down process. So also here we expect the scaled down processor to do similar or worse than four of the not-scaled down processor, even though the difference is probably not as large as for the leakage current. But the gate leakage current is usually smaller than the subthreshold current, so its impact ought to be less significant.

   So, all in all case one is better in terms of leakage power.

b) We need to design one pMOS transistor for the header switch. The voltage drop across is can be max $V_{\text{max}} = 0.05 \times 0.75\,\text{V} = 375\,\text{mV}$. The average power is 2 W which at $V_{\text{DD}} = 0.75\,\text{V}$ corresponds to a current of $\frac{2\,\text{W}}{0.75\,\text{V}} = 2.67\,\text{A}$. Consequently, the maximum resistance allowed for the switch transistor is: $R_{\text{max}} = \frac{0.375\,\text{V}}{2.67\,\text{A}} = 0.14\,\Omega$. The minimum required width of the pMOS transistor is then $W_{\text{min}} = \frac{2\,\text{k}\Omega}{0.14\,\Omega} = 14\,286\,\mu\text{m}$.

c) The expression for the total capacitance is:

$$C_{tot} = C_{\text{gate}}(1 + p_{inv}) + C_{\text{tapered}}(1 + p_{inv})$$

   In this case we know that $C_{\text{tapered}} = C_{\text{gate}}/2$ and we assume that $p_{inv} = 1$ since it is not stated in the problem. Thus, we have

$$C_{tot} = 3 \times C_{\text{gate}}$$

   The pMOS header transistor of width 14 286 µm has a gate capacitance of 14 286 fF. So the total capacitance is $C_{tot} = 43\,\text{pF}$. For each switching event the power is thus:

$$E_{sw} = C_{tot}V_{\text{DD}}^2 = 43\,\text{pF} \times 0.75^2\text{V}^2 = 24\,\text{pJ}$$

   The event of switching the power supply off and then on again occcurs once every 100 µs, that is with a frequency of 10 kHz. So the power is

$$P_{sw} = 10\,\text{kHz} \times 24\,\text{pJ} = 0.24\,\text{mW}$$

   Note that we do not have to worry about any activity factor when we know that the switching occurs every cycle. The power for switching the power on and off is quite small compared to the power we save which we can assume is roughly 1 W out of the 2 W average power since the power supply will be off half of the time.

Solution 5: Wire delay and clock distribution

a) In this solution we add the factor 0.7 at the end so as to not complicate things too much on the way.

   The Elmore delay for a capacitance on the main path is the resistance up to capacitance times the capacitance. For node A the additional Elmore delay due to the increased capacitance is thus

$$T_{\text{added}A} = (R_{\text{inv}} + R_1 + R_2 + R_3)\frac{C_{FF}}{2}$$

   The impact on the nodes B, C, and D, is that of a capacitance being added on a branch, that is not on the main path to these nodes. The additional branch delay is the resistance of the main path up to where the branch starts times the capacitance. So in this case we find for the three nodes:

$$\begin{aligned}
T_{\text{added}B} &= (R_{\text{inv}} + R_1 + R_2)\frac{C_{FF}}{2} \\
T_{\text{added}C} &= (R_{\text{inv}} + R_1)\frac{C_{FF}}{2} \\
T_{\text{added}D} &= (R_{\text{inv}})\frac{C_{FF}}{2}
\end{aligned}$$

To find the resulting clock skews we have to take the differences between the delay increase to A and the delay increases to the other three nodes. Thus, we have:

$$\begin{aligned}
T_{\text{skew}AB} &= (R_3)\frac{C_{FF}}{2} \\
T_{\text{skew}AC} &= (R_2 + R_3)\frac{C_{FF}}{2} \\
T_{\text{skew}AD} &= (R_1 + R_2 + R_3)\frac{C_{FF}}{2}
\end{aligned}$$

Thus, we find that the maximum clock skew is to node D and that the actual difference in delay is:

$$t_{\text{skew}} = 0.7(R_1 + R_2 + R_3)\frac{C_{FF}}{2}.$$

b) The maximum clock skew with values is:

$$t_{\text{skew}} = 0.7(50\,\Omega + 25\,\Omega + 12.5\,\Omega) \times 250\,\text{fF} = 15.3\,\text{ps}.$$

So the clock skew is just above the budget.

c) If the clock-skew limit is due to the setup condition of the flip-flops, one can maybe lower the clock frequency slightly. That would not help if the limit is due to the hold condition though. Another possibility is to do a new floorplan of the chip and move clock tree A away from node D, if the problem is only due to neighboring clock trees. Another solution could be to use slightly different flip-flop, but that would require some redesign.

Solution 6: Faster adders

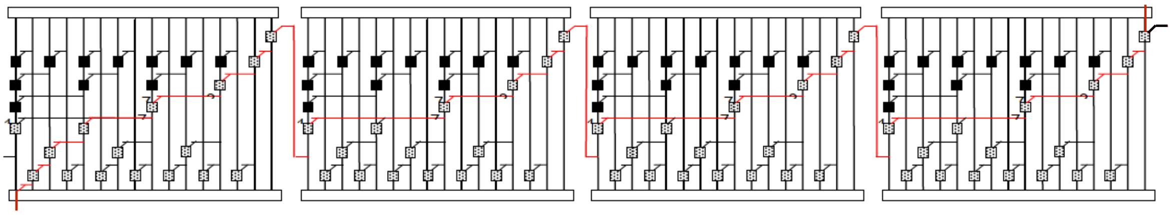
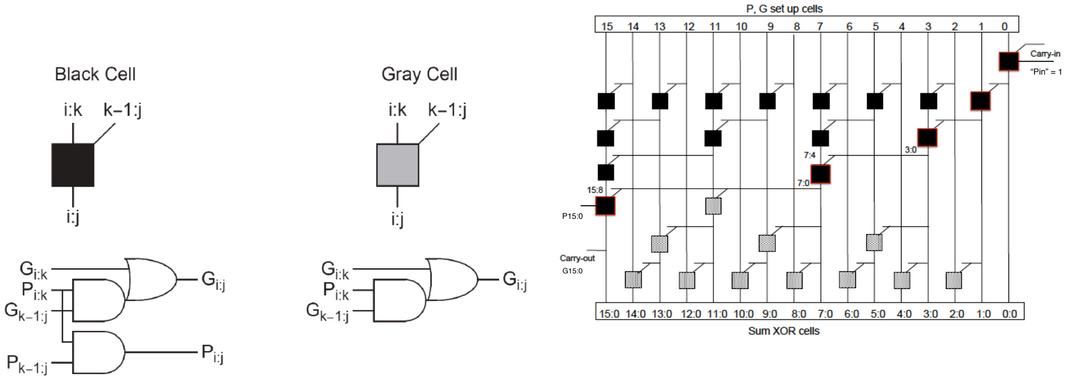


Figure 13: The 64-bit adder with its critical path drawn in red.

a) The last sum to be done is sum 63, which requires G62:0 as an input to its XOR gate. The delay for the critical path is $t_{pg} + (5 + 5 + 5 + 7)t_{AO} + t_{XOR} = t_{pg} + 22t_{AO} + t_{XOR}$. See Figure 13.

b) The grey PG cells do not contain the circuitry (that is, AND gates) for computing the propagate output (see Figure 14 (a)). So we need to switch some grey cells to black cells to create the block propagate signal for the span 15:0. The carry-in also has to have an accompanying propagate signal, which is always 1, as an input to the first added black cell. See Figure 14 (b).

c) We could use a carry-skip solution to skip the second and third Brent-Kung adders. For a detailed drawing of how this is can be done, look at the solution of problem 6 from the exam of 2017-08-21. In that solution it was four 8-bit ripple-carry adders that were used, but the same solution can be used for other types of sizes of adders. (I will add a nice drawing here later.) Maybe there are some other clever ideas. Those would also give points, of course.



(a)

(b)

Figure 14: (a) The contents of black and grey PG cells. Excerpt from Figure 11.16 in Weste & Harris, (b) The modified 16-bit Brent-Kung adder. The black PG cells with a red border are the ones that have been modified.

d) Instead of $5t_{AO}$ for each of the two middle Brent-Kung adders we would only have a delay of $1t_{AO}$ for each of them. We will also need an additional AO gate after the least significant Brent-Kung adder which adds an additional delay $1t_{AO}$. Thus, we would save $10t_{AO} - 3t_{AO} = 7t_{AO}$ in delay.

Solution 7: Inverters and analog aspects

You can look at the solution for problem 1 in the exam from 2013-08-26. When I have time I will add this solution; since noone selected this problem it is not my highest priority