

Written examination in Integrated Circuit Design MCC091

Monday August 22, 2016, at 8.30-13.30 at Lecture halls, Hörsalsvägen

Staff on duty: Lena Peterson, D&IT, phone ext: 1822, or mobile 0706-268907. Will visit around 9.15 and 11.45.

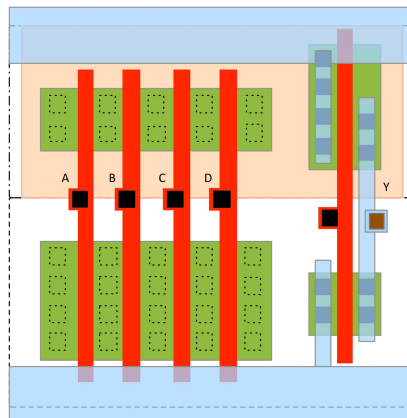
Administration: Send exams to Lena Peterson D&IT, and send lists to CSE student administration office.

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). **The results** from the examination will be sent to you via the Ladok system within three weeks. The grading reviews will take place Monday September 5 2015, 16.00-17.00 in room 4128 and Tuesday September 6, same time and place. The solution will be posted on the course web site August 23. Any student who does not have access to the 2015 course web site can contact Lena Peterson (via e-mail to lenap@chalmers.se) to obtain the solution.

The written examination contains six problems, each worth 10 points. You need 30 points to pass, 40 points for grade “4” and 50 points for grade “5”. Bonus points from the fall 2015 course instance will be added for the higher grades.

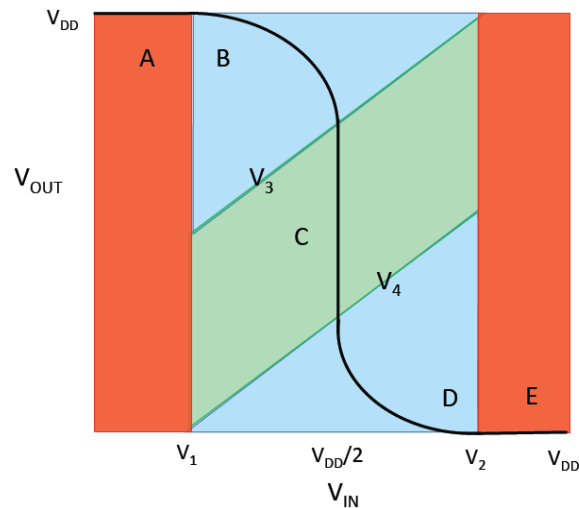
1) Logical effort, parasitic delay, layout

In this problem you are to layout and model a 4-input AND gate. You will use the cell template shown below; which already has the layout of an inverter to the right. Assume that this inverter has a parasitic delay, p_{INV} , equal to 1. The cell template is repeated at the end of the exam; draw your layout on that sheet and hand it in with your solution.



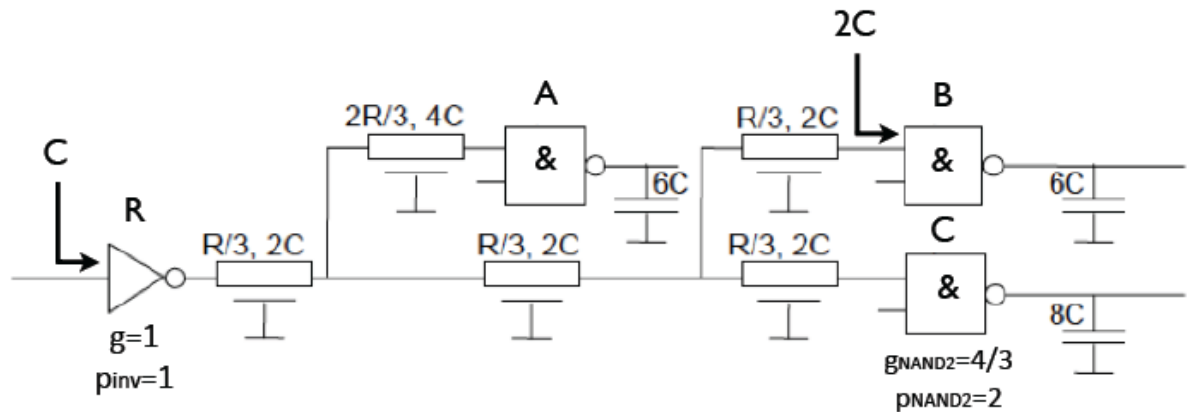
- Draw the circuit diagrams for both a 3-input and a 4-input NAND gate and calculate their respective logical efforts, g_{NAND3} and g_{NAND4} , and parasitic delays, p_{NAND3} and p_{NAND4} . (2 p)
- In the cell layout template above you see the layout of one inverter to the right. To the left of that inverter is a continuous-line-of-diffusion template. Draw the layout for a 4-input NAND gate there; also connect the NAND-gate output to the inverter input, thus forming a 4-input AND gate. Draw the layout such that you minimize the number of diffusion areas connected to the output of the NAND gate. (3p)
- The parasitic delay, p , depends on the capacitances due to the diffusion areas that are the drains of the transistors connected to the gate output. Find the value for p_{NAND4} for your own layout from task b). Assume that the capacitance of a diffusion area of a particular width is the same if it is shared between two transistors as if it is not shared. (2p)
- For the 4-input AND gate, formed by the inverter and the 4-input NAND gate, find the logical effort, g_{AND4} , and the parasitic delay, p_{AND4} , for the entire gate. Use the p_{NAND4} value resulting from task c) if you have solved that task, otherwise use the p_{NAND4} value from task a). (3p)

- 2) **Inverter static characteristics** In the diagram below is the ideal voltage transfer curve for a CMOS inverter, derived assuming quadratic current equations and $V_{TP} = -V_{TN}$, with the coloring scheme that we have used throughout the course.



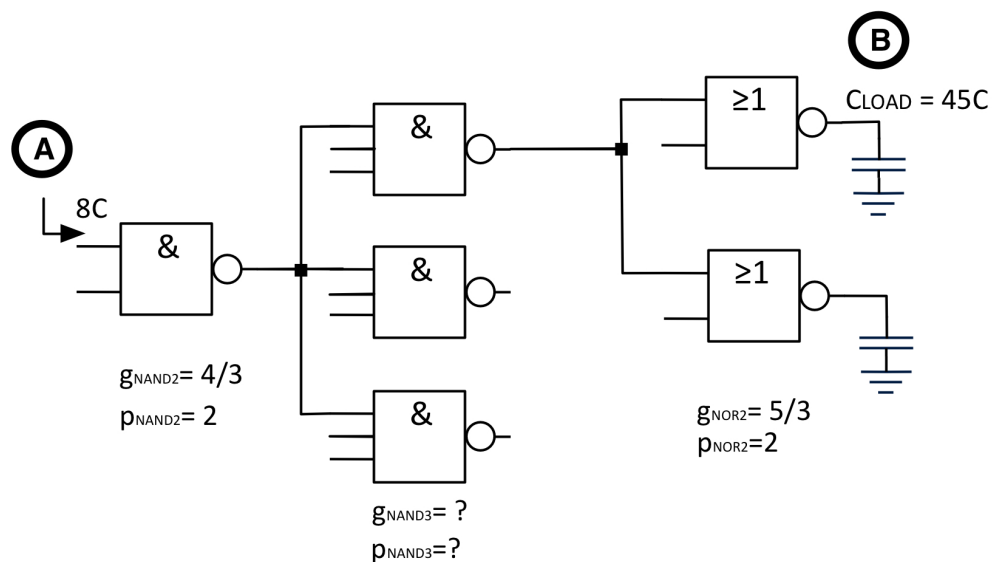
- For each of the five colored areas, A to E, indicate in which operating region the nMOS transistor and the pMOS transistor operate. Use the upper VTC on the tear-off sheet located at the end of the exam and write directly there. In that diagram, also add the expressions for the four voltage transitions: $V_1 - V_4$. (4 p)
- What if we made the nMOS transistor in the CMOS inverter **four** times wider than it is in the CMOS inverter of the VTC above; how would the transfer curve look then? Draw the new ideal VTC in the lower diagram on the tear-off sheet at the end of the exam. For this new VTC, indicate the input voltage at which the steep transition, the one in area C, occurs. (4 p)
- In the ideal VTC above, it looks as if the part of the VTC in area C is infinitely steep, but in reality that is not true. Explain why not. (2 p)

3) **Wire and gate delay, logical effort** The figure below shows part of a clock-distribution network that comprises an inverter that acts as the clock driver, some wiring, and three identical NAND gates that act as clock gaters.



- Calculate the clock skews at the **inputs** of all clock gates: A, B, and C. The clock driver has a driver resistance R and an input capacitance C . The identical NAND gates all have an input capacitance of $2C$. (6 p)
- Calculate the clock skews at the **outputs** of the three NAND gates: A, B and C. (4 p)

- 4) **Path logical effort, path delay, gate sizing** Below, you see a logical path from node A to node B. As indicated in the figure, the first gate has an input capacitance of $8C$ at node A and the output load at node B is $45C$. For the 3-input NAND gate use the g and p values calculated in task 1a).



- Calculate the path logical effort for the path from A to B. (3 p)
- Use the path logical effort resulting from task a) to find the optimal stage effort for minimum delay. Calculate the resulting gate sizes (that is, the gate input capacitances). (5 p)
- Calculate the resulting minimum delay for the entire path expressed in FO4 delays. (2 p)

5) **Power** As you know, the dynamic power consumption for a circuit block, or an entire chip for that matter, depends on the charged capacitance, C , the power supply voltage, V_{DD} , the clock frequency, f_{clk} , and the activity factor, α .

The capacitance that is charged in the circuit block or chip is fixed; but it is possible to adjust both the clock frequency and the supply voltage dynamically, that is, while the chip is operating. This procedure is called dynamic voltage scaling, DVS. In DVS, one decreases the clock frequency as much as possible while still achieving the required throughput and then, because there is more time to perform the required computation when the clock frequency is lower, one also decreases V_{DD} so that it is just high enough to fulfill the throughput requirement. Here is a figure from the Weste&Harris textbook that shows the principle setup of a DVS system:

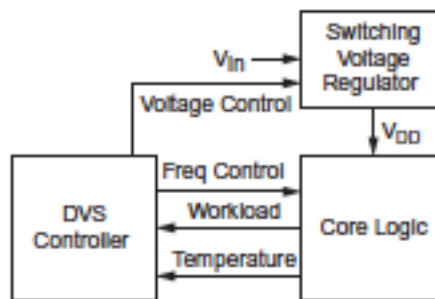


FIGURE 5.17 DVS system

In this problem, you are to estimate the power gain from a simplified version of DVS where just two levels of V_{DD} are used: a high one, the standard V_{DD} , and a lower one. Assume that you have a core logic block that has the highest workload, and thus the highest throughput requirement, only 25 % of the time. For that time the maximum clock frequency, f_{clkmax} , and the maximum supply voltage V_{DDmax} must be used. The rest of the time, 75 %, the workload is much lower. It is so low that a much lower clock frequency can be used: $f_{clkmin} = 1/4 * f_{clkmax}$. Assume that the activity factor is the same in the max- V_{DD} and min V_{DD} cases.

The chip is fabricated in a process with $V_{DDmax} = 1.2$ V and the threshold voltages are $V_{thn} = -V_{thp} = 0.12$ V = 0.1 V_{DDmax} . Assume that the quadratic current equations hold. For simplicity you can also assume that the voltage and clock frequency can be changed without any time overhead, but do assume that the total capacitance in the circuit has to be increased by 5% due to additional circuitry required for controlling the clock frequency and the supply voltage.

- Calculate V_{DDmin} , that is the lowest supply voltage that can be used while still maintaining the throughput when the lower clock frequency, f_{clkmin} , is used for clocking the core logic. (4 p)
- How much dynamic power is saved using this two-level DVS arrangement? (4 p)
- Are there any gains also for the **static** power consumption due to subthreshold leakage in using DVS? Motivate! (2 p)

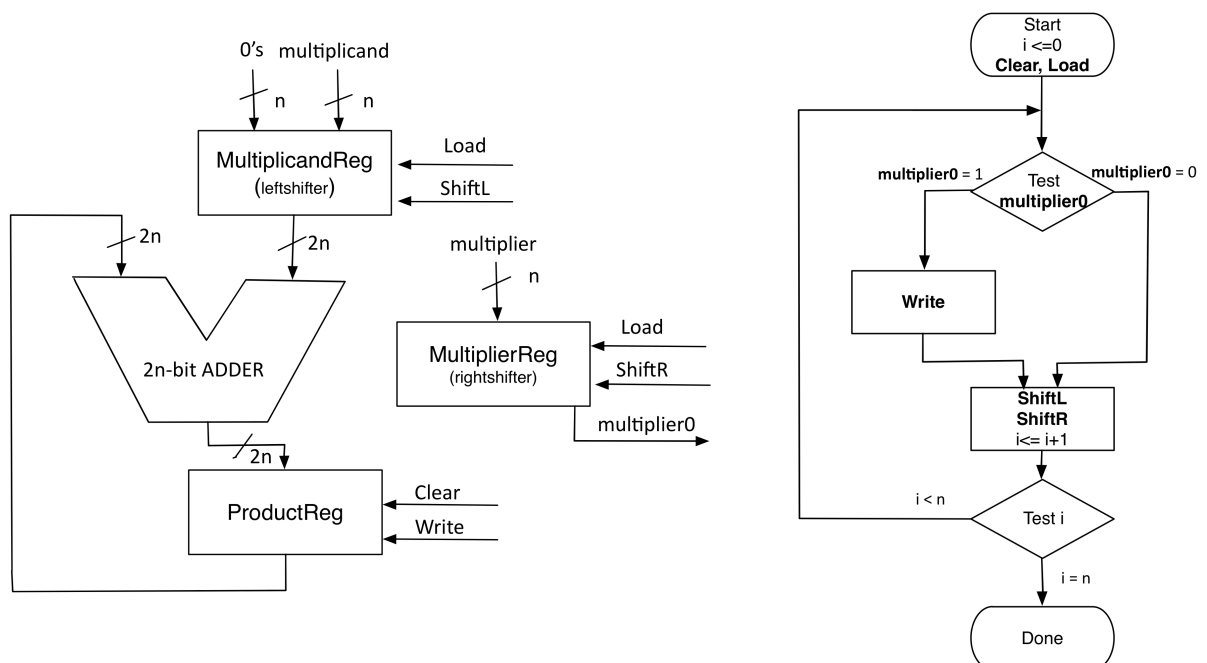
- 6) **Adders, critical path, iterative design** In this course we have designed many adders but no multipliers. In this problem you will investigate how to use adders to implement binary multiplication and the performance of such an approach.

Here is an example of a 6-bit binary multiplication from the Weste and Harris textbook:

$$\begin{array}{r}
 011001 : 25_{10} \\
 \times 100111 : 39_{10} \\
 \hline
 011001 \\
 011001 \\
 000000 \\
 000000 \\
 +011001 \\
 \hline
 001111001111 : 975_{10}
 \end{array}$$

multiplicand
 multiplier
 partial products
 product

From this example it is clear that the partial products are just left-shifted versions of the multiplicand. Binary multiplication can thus be performed by repeatedly shifting the multiplicand to the left and adding it to the product. Here is a figure that shows how a $2n$ -bit adder can be used to perform binary multiplication of two n -bit binary numbers. To the left you see the datapath with an adder, two shifters and a register, and to right the iterative control required to perform a multiplication:



In this problem, your task is to investigate the performance of this iterative multiplication for different types of adders and number of bits, n .

Here are worst-case delays for two types of adders, for 8- and 16-bit additions.

| Number of bits in adder, n | Ripple-carry adder Worst-case delay (ps) | Prefix adder (Sklansky) Worst-case delay (ps) |
|------------------------------|---|--|
| 8 | 130 | 200 |
| 16 | 250 | 250 |

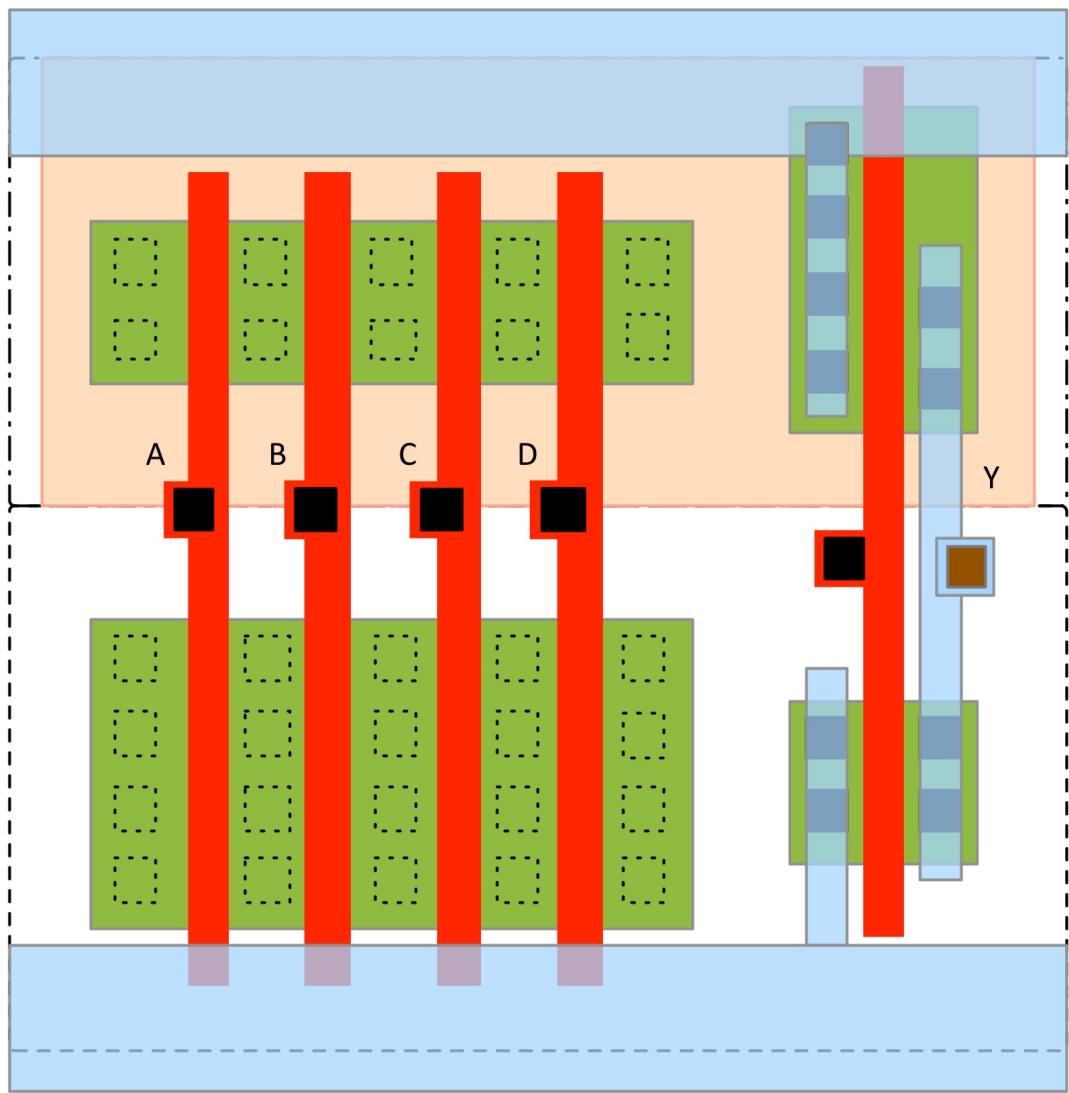
For the timing in the datapath assume that the adder output has to be stable for 20 ps before the **Write** signal is issued to the ProductReg, and that the worst-case delay, from when the **Write** signal is issued until when the output of ProductReg has changed, is 30 ps. Similarly, assume that the shifters have a 30-ps worst-case delay from the issue of the **ShiftR** or **ShiftL** signal until the shifted output is available at their outputs.

For the control logic assume that each step takes one clock cycle.

- Use the worst-case delay adder data in the table above to estimate the maximum clock frequency that can be used for the iterative 8-bit multiplier. With this clock frequency and assuming worst-case multiplier input data, how long would it take to complete one 8-bit multiplication? (4 p)
- What if we extend the iterative multiplier from task a) to multiply two 32-bit binary numbers? How will its worst-case delay change? Assume that you can generate wider versions of the two types of adders in the table above. Which type of adder would you select? Motivate! For the selected type of adder, estimate the maximum clock frequency with which one could clock the multiplier control logic and still ensure a correct result. How long would it then take to complete one 32-bit multiplication with the worst-case multiplier input data? (6 p)
- BONUS QUESTION The proposed multiplier is not that well designed. Suggest one substantial improvement that could be made to the **datapath** and estimate how much that improvement would increase the maximum clock frequency calculated in task b). (4 p)

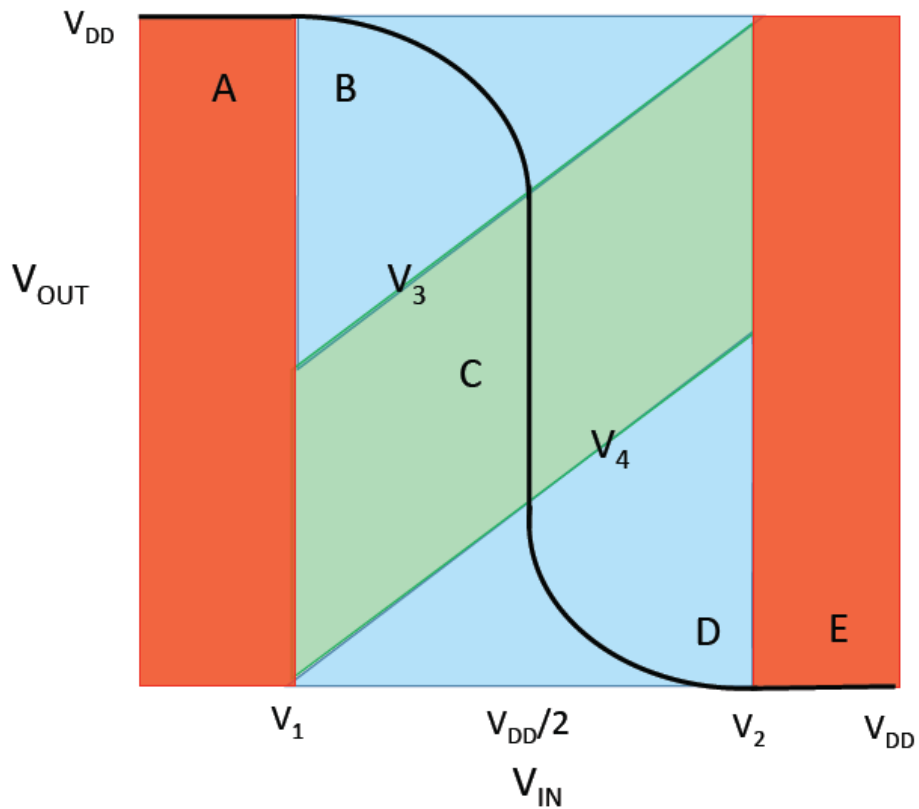
THE END!

Anonymous code: _____



Anonymous code: _____

2a) Indicate the transistor operating regions in areas A-E and the expressions for the voltages V_1 - V_4 here:



2b) Draw the VTC for the modified CMOS inverter below.

