# Digital Electronics Exam 2023

By Jesper Bertelsen, AU-ID: au689481

Indholdsfortegnelse

[Digital Electronics Exam 2023 1](#_Toc157064858)

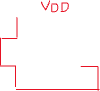
## Question 1.

### A (10%) Sketch a static CMOS gate to implement the following equation. You may assume you have both true and complementary versions of the inputs available. If it is possible, simplify the equation to reduce the number of transistors.

Firstly I will try to reduce the code:



Which can be simplified down to:



Which is a simple function.

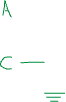


What to do:

Things to know.



After that is done I will invert the signal



### Design the size of transistors in a way that tpdr=tpdf. Explain the different steps that you took to reach the final answer.

Wmin=200nm

L=60nm

μn = 2μp

VDD=1.0 V

Let me look at the worst case sceneries.

There is some sceneries here.

A = C = 0 -> Two times the pmos signals and sends it to the pull down.

A = 0, C = 1 -> One pmos signal into one nmos.

A = 1, C = 0 -> One pmos signal into two nmos in which one of them doesn’t go all the way to ground.

A = C = 1 -> Rise case.

Okay so which case is the worst.

The 1. Case uses 3 resistors, but are in parallel, so the pmos part just quickens the signal.

The 2. Case uses one pmos and one nmos.

The 3. Case uses one pmos and one nmos but another nmos is activated but not sending to ground.

The 4. Case is the rise case.

3. and 4 case are equally bad in my opinion, both sending signals through 3 mosfets, the one however not using the signal it send through a mosfet, but that doesn’t matter on the delay.

Et billede, der indeholder diagram, tekst, håndskrift, Font/skrifttype

Automatisk genereret beskrivelseI will do it for the rise case.



Let me try to find the equivalent resistance.



being the factor times the minimum width, and *m* being a mobility constant, used to differentiate the difference between nmos and pmos.

Pmos has lower mobility, so to compensate for that, I set

So the alpha for the rise   pmos is

As the other pmosses doesn’t contribute, as

The other pmosses then only add to the fall cases.

And for the nmos of this combination:

Taking a lot towards the paths are similar, when , so the worst case here will be, when only one is on.

So for the worst case, pmosses should have a factor of two on it.

And for the fall of   into the nmos.

### First, convert the circuit to its RC model. Then, estimate tpd of the designed circuit using the Elmore delay model. Suppose that a unit transistor has R=10k and C=0.1fF and your circuit is deriving a 10fF capacitance (a 10fF capacitor is connected to the output).

Let me just convert every transistor with a



drain capacitance going to ground,



a resistor, a switch, and a drain capacitance.



Here it is:



Simplifying for AC=1



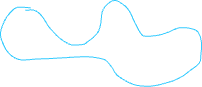
### Et billede, der indeholder tekst, diagram, linje/række, Plan Automatisk genereret beskrivelse



So the circuit can be simplified to.



Now using elmores delay formular



With *n* being the amounts of nodes with capacitors.



Now substituting:



Making the delay time:

=======

=======

### Simulate the designed circuit in LTspice and check if tpdr=tpdf. In the case tpdr≠tpdf, try to make them equal by changing the size of the transistors and explain why the sizes of these particular transistors should be increased/decreased.

Testing all cases:

Et billede, der indeholder skærmbillede, Kurve

Automatisk genereret beskrivelseMy setup: Resulting in the simulation:

Et billede, der indeholder tekst, diagram, Plan, skærmbillede

Automatisk genereret beskrivelseWhere the delay is very small.

Using measuring to find the exact delays gives me:

Measuring it

Et billede, der indeholder tekst, Font/skrifttype, hvid, algebra

Automatisk genereret beskrivelse

Measuring both the rise delay and fall delay to be:

=============================

=============================

One third of a change between them is actually a lot. As the rise time takes the longest, the AC pmos should probably be bigger, let me scale it by 2.

This doesn’t make an impact, so it might have to do with the A & B nmos then, as the pmos might already be on, when the ground signal comes to .

Looking at my measurement commands the and has not been helpful.

.meas tran t1 find time when v(A)=0.5 and v(C)=0.5 rise=1

In LTSpice actually takes the or value of the two, and I thought that it would take the time when both criteria’s met.

But why I did above would have been how I calculated it all.

### First, find the input change (from ABCD=”XXXX” to ABCD=”YYYY”) in which the dynamic power consumption will be maximized. Then, evaluate the dynamic power consumptions for the worst case in LTspice. Måske kom tilbage hertil

Commands to check for the 1. Average static power consumption of a whole period. Using

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, hvid

Automatisk genereret beskrivelse

Resulting in.

Et billede, der indeholder tekst, Font/skrifttype, hvid, algebra

Automatisk genereret beskrivelse

Total power being the static power

And the dynamic power failing to measure.

I could then try to find the dynamic power if I had the periods of rise and fall correct.

And maybe see if that would be equal to

But I don’t have the time measurements correct, so I can’t right now.

## Question 2. Compare the two implementations of a full adder in terms of energy consumption, speed, size, ability to remove noise.

Et billede, der indeholder diagram, skitse, Teknisk tegning, Plan

Automatisk genereret beskrivelse

## Question 3. Compare the dynamic and static logic in terms of energy consumption, speed, size, and ability to remove noise.

Assuming question 3 continues from Question 2.

## Question 4. VHDL

### Write a VHDL code for implementing the following truth table

Et billede, der indeholder skærmbillede, tekst, Font/skrifttype, nummer/tal

Automatisk genereret beskrivelse



I see the outputs A0, A1, but I don’t see the input I0-3, these must be the B0, B1, B2, B3 instead.

What I immediately notice is, that this is a 4 to 2 encoder. When enable is on.

-----------------------------------------------------------------------------------------------------------------------------------

;

;

-----------------------------------------------------------------------------------------------------------------------------------

a being the output vector with 2 bits. b being the input vector with 4 bits.

Then making a process in which b is being encoded into 2 bit output a, checking for all cases, b isn’t 4 bit decoded, I set the value of the output *a* to a default 00, which I think indicates the most, that somethings might be odd, but this can be configured to something else also.

### Then, write a test bench for it to verify the operation of your VHDL code.

-----------------------------------------------------------------------------------------------------------------------------------

# Making a process for the entity I want to test:

# Making a process for the combination.

# Testing for all combinations. 32 combinations in total. I will just summaries, what will be put in.

# Testing the True cases.

# These 4 test to see the wanted results.

# Now the default cases.

# Default case

# Default case

… And so on for the next which would make the default cases.

# Now checking for when the circuit is not enabled.

# When the circuit is outputted, it should maybe hold the last value, or in my case I just make the output 00.

# We don’t care about b in this case.

-----------------------------------------------------------------------------------------------------------------------------------