# Reexam

Indholdsfortegnelse

[Reexam 1](#_Toc167962610)

[Question 1. CMOS network 1](#_Toc167962611)

[a. Sketch the network 1](#_Toc167962612)

[b. Design the size of the transistors such that 4](#_Toc167962613)

[c. Convert the circuit to its RC model. Then estimate using elmore delay model. Kom tilbage hertil 6](#_Toc167962614)

[d. Simulating in LTSpice. 7](#_Toc167962615)

[e. Evaluate static and dynamic power consumptions for your circuit in LTspice. For dynamic power change the input from to 8](#_Toc167962616)

[Question 2. Sketch the implementation of a 2 input NAND using dynamic and static logic and compare them in terms of speed and size. 9](#_Toc167962617)

[Question 3. Sketch the circuit of a pulse generator and describe how it can be used to make a circuit sensitive to both rise edge and fall edge of the clock. (Dual edge triggered). 11](#_Toc167962618)

[11](#_Toc167962619)

[Question 4 VHDL code 12](#_Toc167962620)

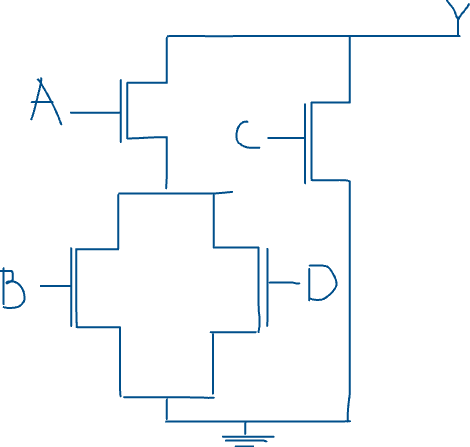
[a. Write a VHDL code for implementing the following truth table 12](#_Toc167962621)

[12](#_Toc167962622)

## Question 1. CMOS network

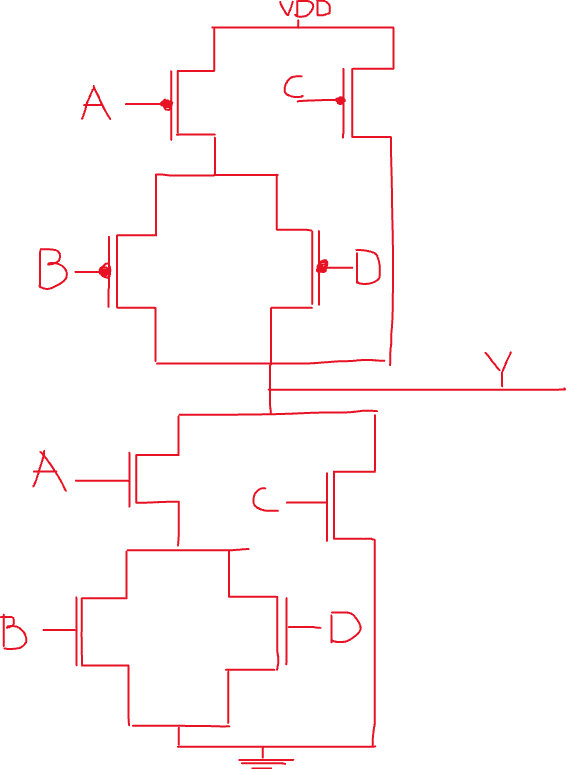
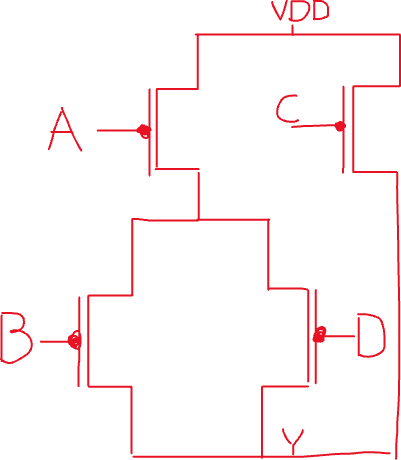
### Sketch the network

Let me first start with the nmos network



For the pull down network I need the function but knowing, that the inputs are gonna get inverted.

The cmos network:



### Design the size of the transistors such that

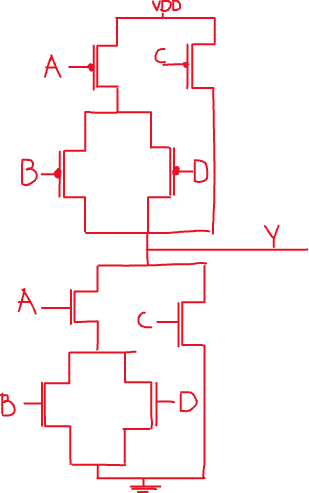
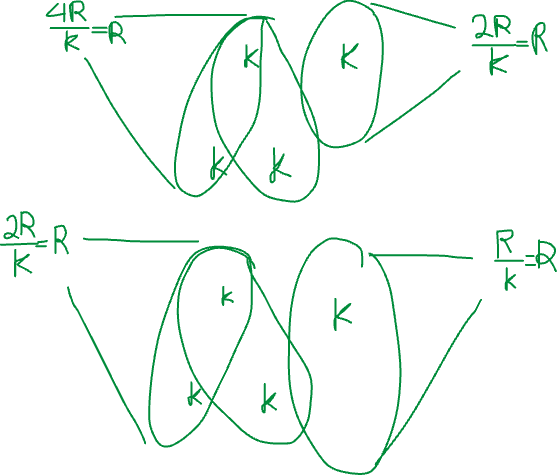
As the mobility can be described as resistance in the paths the voltage can go through the network with, I know that:

And I know that the current through a transistor can be described as

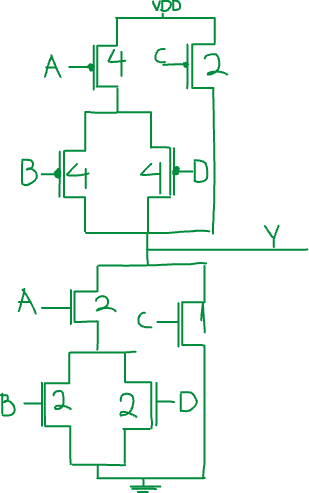
And as I know, that the current has a linear factor of I know that the resistance in a pmos is greater than the resistance in a nmos by a factor of two.

So I say that the resistance for a unit width for

When looking at the paths I want to make the path a unit resistance. I also know that the current has a linear factor of Width two. With these things in mind I find the width for every transistor.



My findings:



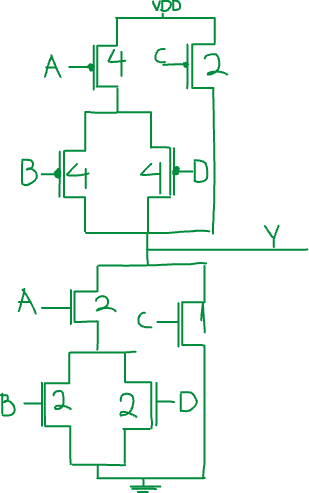
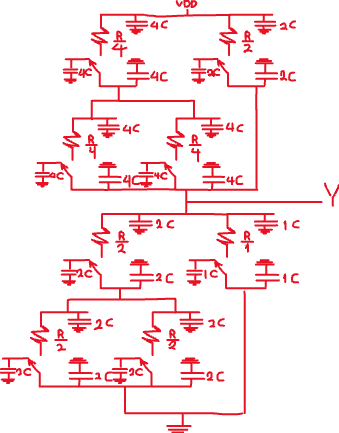
================================================================================

### Convert the circuit to its RC model. Then estimate using elmore delay model.

The mosfets RC equivalents:

Et billede, der indeholder diagram, tekst, skærmbillede, linje/række

Automatisk genereret beskrivelse



Estimating for rise and fall should equal the same, as that’s what I designed the transistors for.

I look at the worst fall time and estimate the from that. The worst case is when either B og D is 1 but not both and .

I look at when A has been open for a long time, and B switching from 0 -> 1

I realised I have made a flaw in my CMOS network. The VDD can go straight through

If I hadn’t made a mistake in my network I would have:

1. Simplified the circuit.
2. Solved for the time constant tau by:

With the sum of resistances being the ones to go through to get to the capacitor.

### Simulating in LTSpice.

Simulating the circuit doesn’t make sense at this point as I know it will fail.

But I would have done as I’ve previously done in this circuit:

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

Automatisk genereret beskrivelse

And I would have calculated the by looking at a time when a change in inputs from 0 -> 0,5\*VDD would’ve resulted in a fall, and then look at, when the output got to 0,5\*VDD. Then subtracting the first time from the second.

Same procedure for the .

would then have been:

My measurements would have looked a little like this.

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

Automatisk genereret beskrivelse

If I were to change the widths of the transistors to compensate if the rise wouldn’t equal the fall propagation delay, I would have changed the width of the transistors corresponding to the change in inputs.

### Evaluate static and dynamic power consumptions for your circuit in LTspice. For dynamic power change the input from to

For the static power I would’ve looked at the average voltage over a period of the inputs changing (ABCD = 0000 -> 1111). I would’ve then timed it by the average current running from the VDD, from the same current (Real components have leakage currents), and then made a parameter for that.

The measurement would’ve been something like:

With my circuit only running one period.

For the dynamic power consumption I would then look at the graph of the leakage current and:

1. Find the ∆t in which the leakage current spiked the most.
2. Multiplying them together with a target over that period.

It would’ve looked something like this:

Assuming that the would be time symmetric around 0 -> 0,5, 0,5 -> 1.

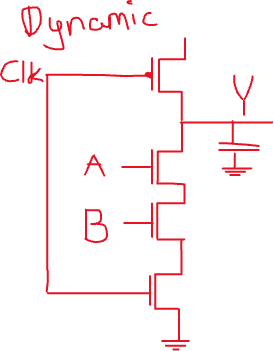
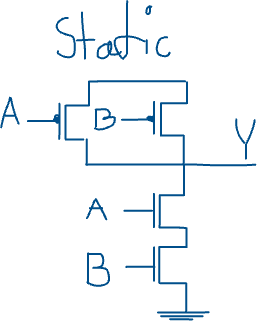
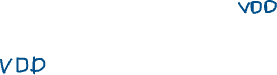
## Question 2. Sketch the implementation of a 2 input NAND using dynamic and static logic and compare them in terms of speed and size.

My static implementation:

For the pull down network:

For the pull up network:

With their inputs being inverted.



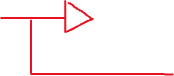
The comparison.

Static over dynamic:

* The static experiences lower delay when starting up:
  + The static doesn’t wait one clk cycle to be active. The dynamic does
* When at the same time period, the static will have a fast speed:
  + Two paths to VDD decreases the delay. The dynamic both have to wait for the clock, and has only 1 path two VDD.
* The size of the static is lower than the size of the static.
  + The static uses 2 pmoses, the dynamic uses 1. Designing for then:
  + The static would have worth of transistors
  + The dynamic would have worth of transistors
* Dynamic circuits require to run a setup phase before the output can be valid.

## Question 3. Sketch the circuit of a pulse generator and describe how it can be used to make a circuit sensitive to both rise edge and fall edge of the clock. (Dual edge triggered).

## 



Assuming all wires are 0 at t0, this is how the output would go.

- a = 0, b = 1, out = 0 -> 1.

-

-

This is due to the time in which the signals are switching.

As signal a experiences some propagation delay, there will be a time period, in which both , this is where the signal will be 0.

When the signals has changed to their supposed values, the output will then again be 1.

It would look something like this:



Denote my use of “Something” as not the actual signal.

## Question 4 VHDL code

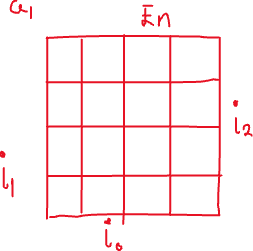
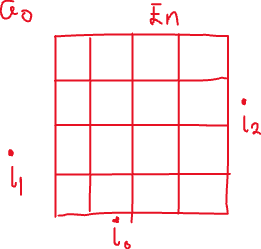
### Write a VHDL code for implementing the following truth table

### Et billede, der indeholder tekst, skærmbillede, Font/skrifttype Automatisk genereret beskrivelse



There is an error in the truth table, therefore row number 3 will be ignored.

Let me write KMAPs for the two bits.



*How I do my implementation in VHDL is by the behavioral logic.*

So by simplification the function can be written as conditional assigments:

My declaration and architecture:   
Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, lilla/violet

Automatisk genereret beskrivelse

For the test bench I use a test bench generator just to get the template of the code that’s required.

It just ads my logic as a component, assign the signals and starts a process for me.

The template:

library IEEE;

use IEEE.Std\_logic\_1164.all;

use IEEE.Numeric\_Std.all;

entity logik\_tb is

end;

architecture bench of logik\_tb is

component logik

Port ( i : in STD\_LOGIC\_VECTOR (2 downto 0);

a : out STD\_LOGIC\_VECTOR (1 downto 0);

en : in STD\_LOGIC);

end component;

signal i: STD\_LOGIC\_VECTOR (2 downto 0);

signal a: STD\_LOGIC\_VECTOR (1 downto 0);

signal en: STD\_LOGIC;

begin

uut: logik port map ( i => i,

a => a,

en => en );

stimulus: process

begin

wait;

end process;

end;

I then just assign the signals through 0000 -> 1111 and look at the output signal for every one

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

Automatisk genereret beskrivelse



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

Automatisk genereret beskrivelse

Steps then:

1. “Run synthesis”, making sure that the circuit compiles.
2. “Run simulation”
3. Look at the output of and making sure, that it’s the result of the right inputs.

Note:

I’m using an Mac with an ARM system running the M chips. I’m interpretting linux to get Vivado to run. This comes with the downside being, that I cannot use testbench. Doing lab however I just used the ZYBO board for finding the true values, and that worked fine.