HY330 –VLSI Digital systemss

Exercise set 4

8/1/2020

Βαγενάς Αναστάσης

Exercise 1

In this exercise we want to implement a decoder (initially 1 bit, then 3 in 8):

Part (α)

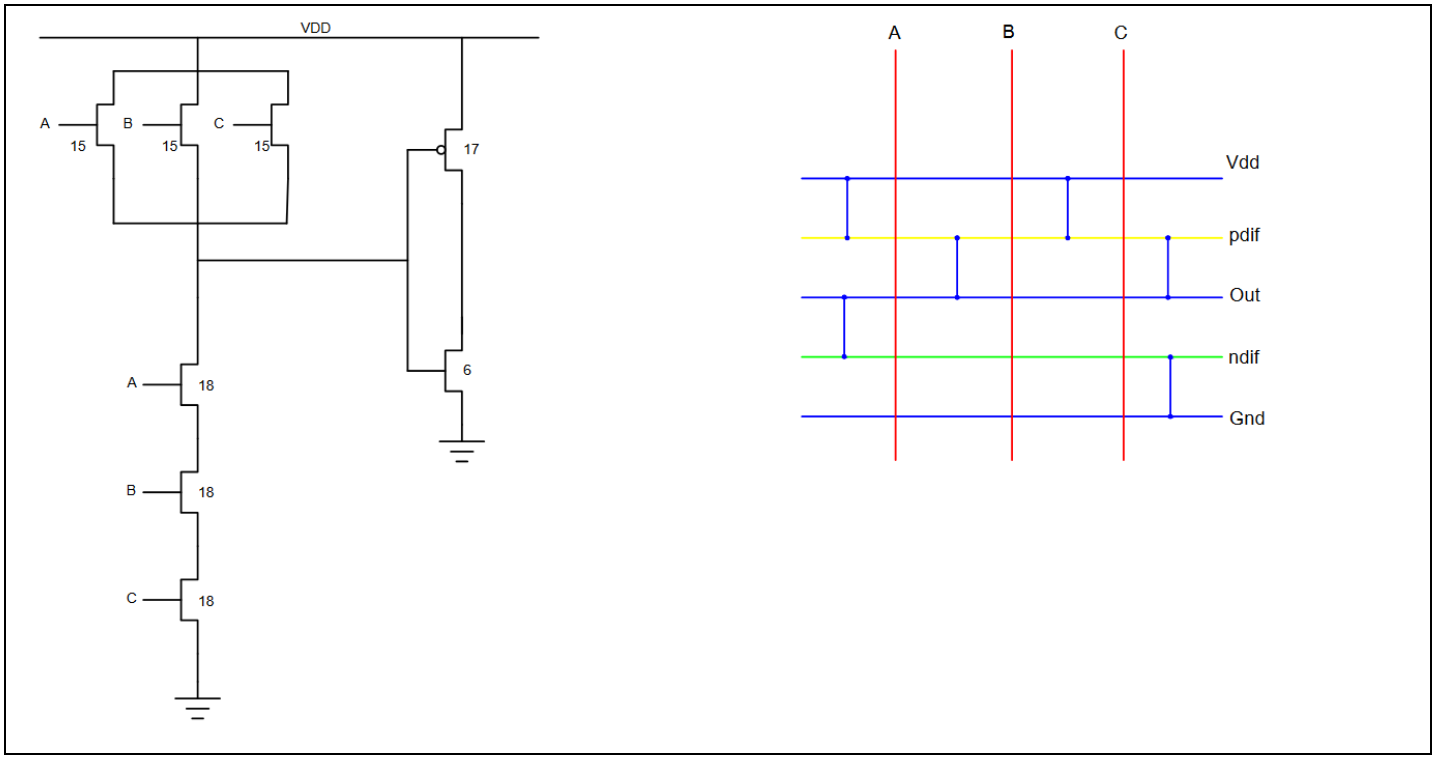
Truth table (Eg for the 1st bit)

A	B	C	out
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

The expression for the 1st bit is essentially out = A'B'C' .

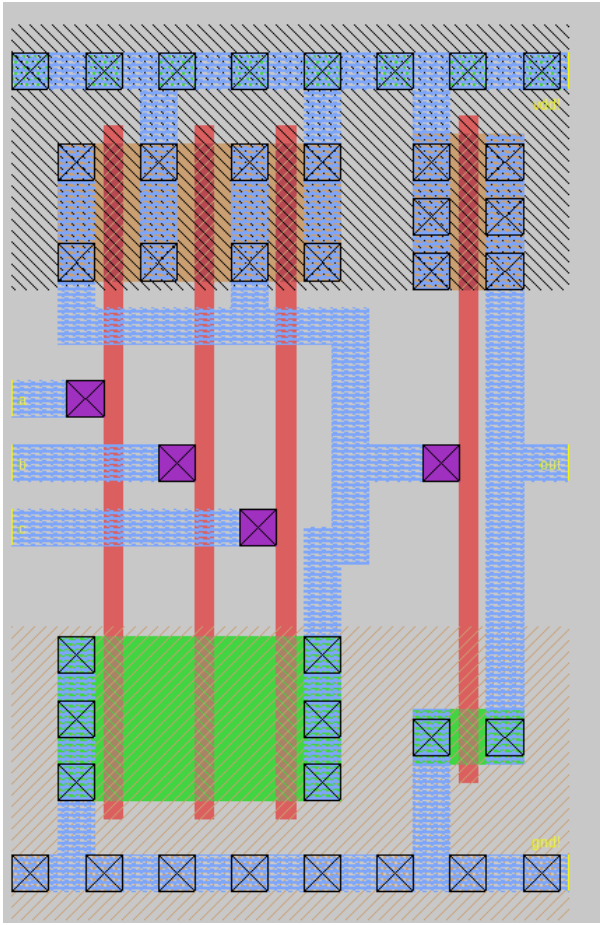
We have the same for the other cases where for a specific input vector, we have the output to be 1 while in all the other 0. So the gate to be designed is an AND gate.

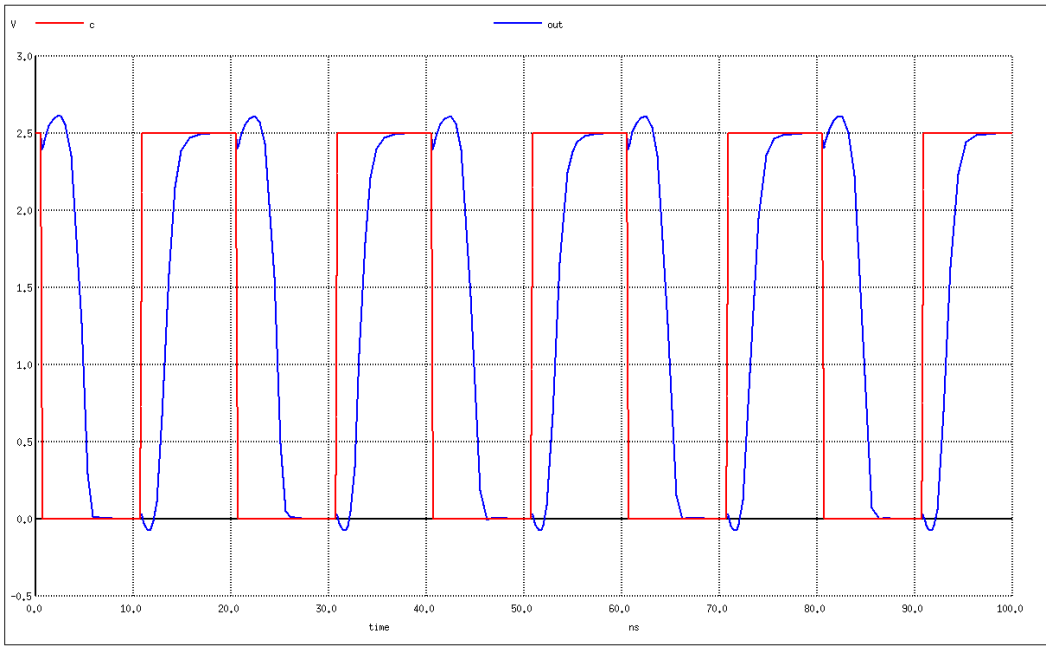
Below is the schematic:



The sizes are based on the 13kΩ and 31kΩ models in NMOS and PMOS respectively, with a target of 6.5kΩ in each network.

Design in MAGIC/simulation in NGSPICE of 1bit





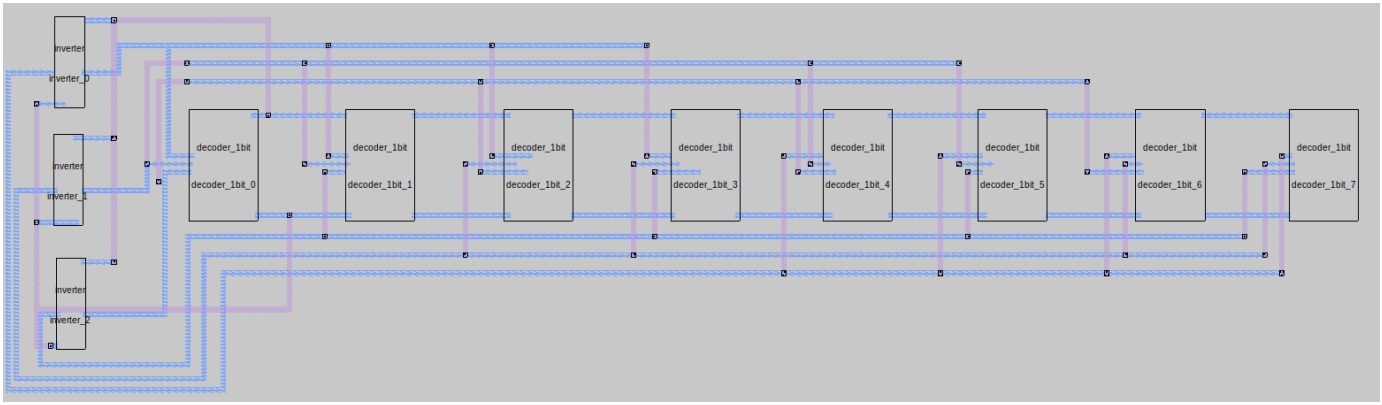
In the simulation we keep A and B at 2.5V constant and give a pulse to C (from 0 to 2.5V).

As can be seen the gate operates normally with the output becoming 1 when C goes to 2.5V.

The rise / fall times for the gate are 2.01ns and 1.78ns (these are the closest values that could reach each other).

Designing a 3 in 8 decoder

The design in MAGIC:

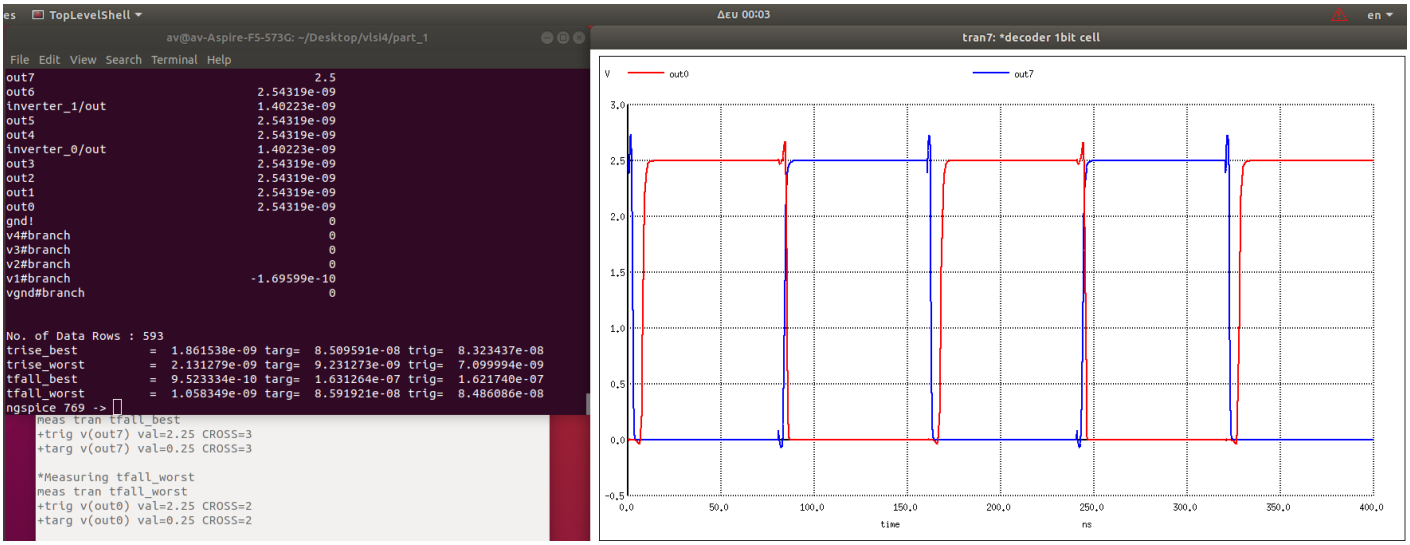


As shown above, first we have the inverters which give us the inverted input signals. These signals will be from the top of the decoders while the opposite ones from the bottom. In this way we do not have better organization or symmetry but we do not need to use beyond metal 2.

Finally, inverters are the subcircuit from exercise 1.

More can be seen, of course, in the .mag file.

Simulation of decoder in NGSPICE



For the simulation we give '000' and '111' as input vectors. We achieve this by having a pulse that changes each input at the same time (the A-BC) from 0-> 1 and 1-> 0. So we have these values for control.

These were chosen because they are also the extreme cases of the decoder. The '111' input is the best case since the signals to activate the 8th bit of the decoder come directly from the test environment, ie we only have the delay of the AND gateway. with the input '000' all the inputs must pass through the inverters so we have the delay of the inverters together with the AND gate.

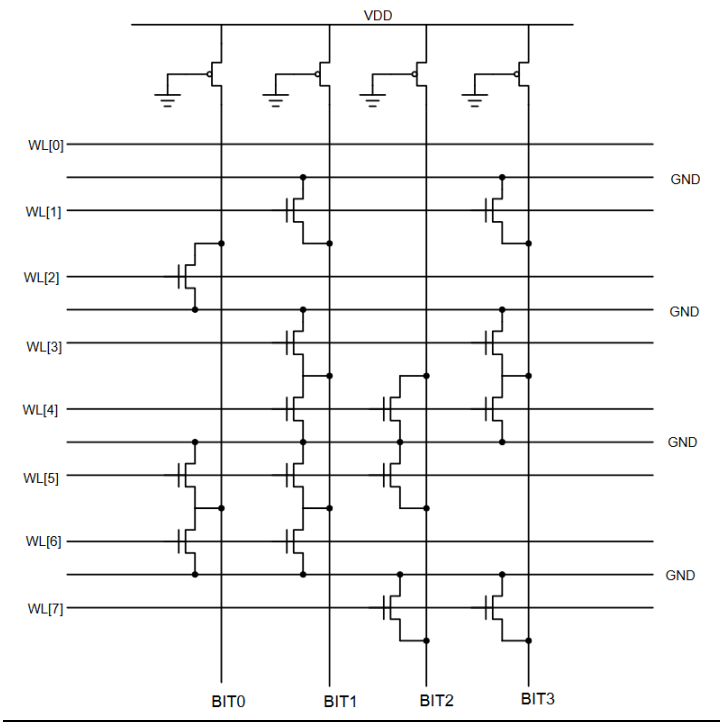
In the right image we see the switching of the outputs (all the others remain at 0). The out0 for input ‘000’ and out7 for input ‘111’.

On the left our hypothesis is confirmed. For input ‘000’ the rise/ fall time is 2.13ns / 1.05ns respectively. While for input ‘111’ the rise / fall time is 1.86ns / 0.95ns.

Άσκηση 2

In Exercise 2 we have to design a ROM memory, 8 slots, 4-bit and NOR type.

Schematic of ROM



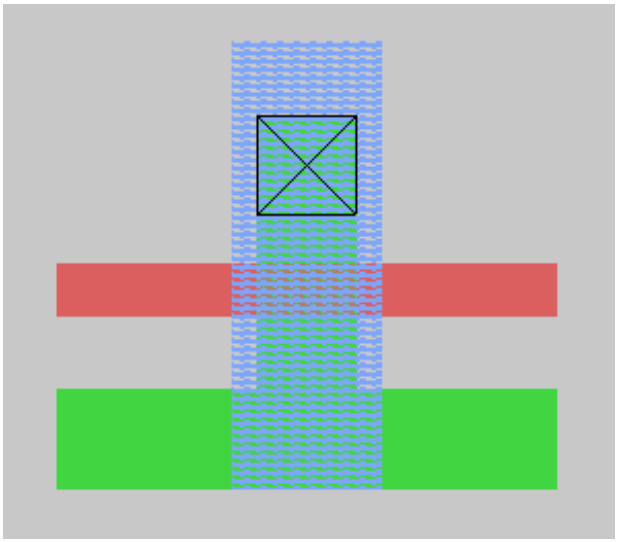
Apparently we have the memory layout requested. At the top we have the pull-up network from the PMOS transistors. Inside the memory at the points where we have a transistor we have stored ‘0’ while its absence is the logical ‘1’. In addition the MSB is the right bit and as we move to the left we have the LSBs of the stored word.

Designing a 1-bit cell

First we design the 1-bit memory and we design 2 types, the type that has stored ‘1’ (absence of transistor-diffusion) and the format that has stored ‘0’ (in the presence of transistor-diffusion).

This is because if we pass diffusion directly over the cell we will have an error, since the diffusion can not be above the polysilicon, the metal must be at the same level as the diffusion of the cell.

Below is the schematic of the 1-bit cell (of the type which stores ‘0’):

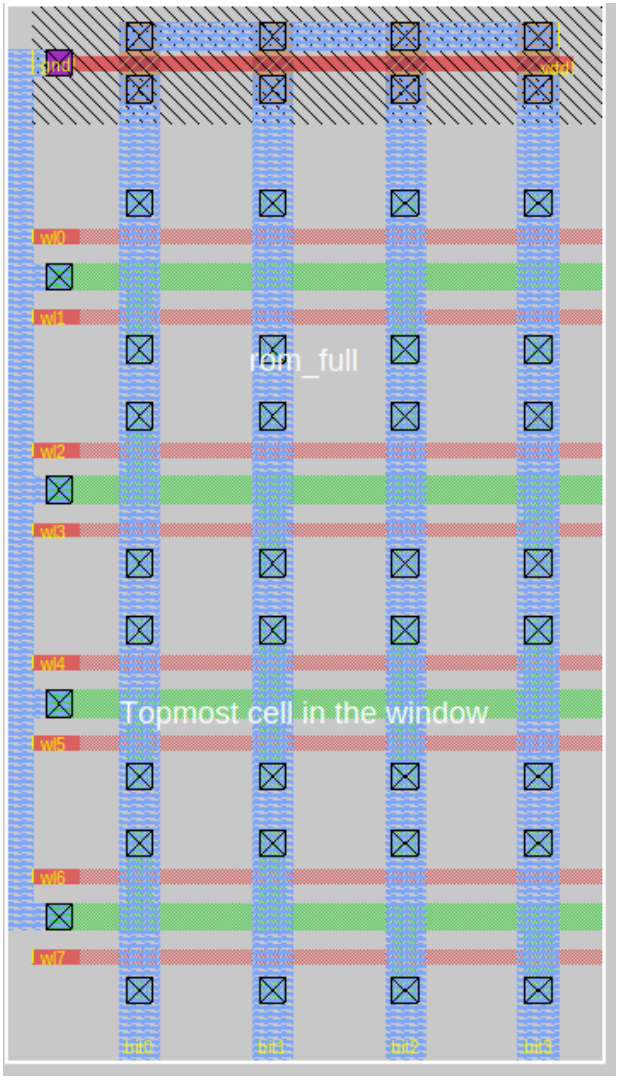


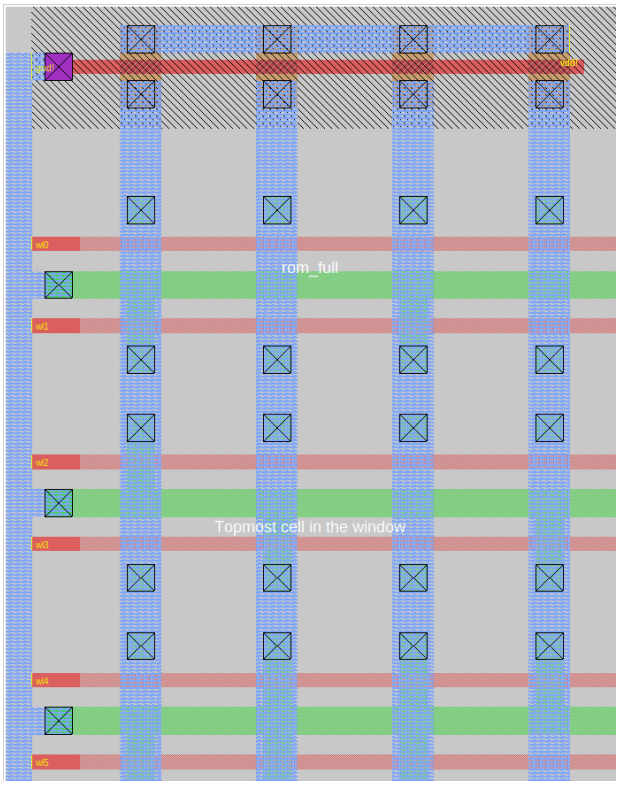
We do not place labels since (magic gives the cell names of the cell inA format, which the spice does not accept when we ask to plot).

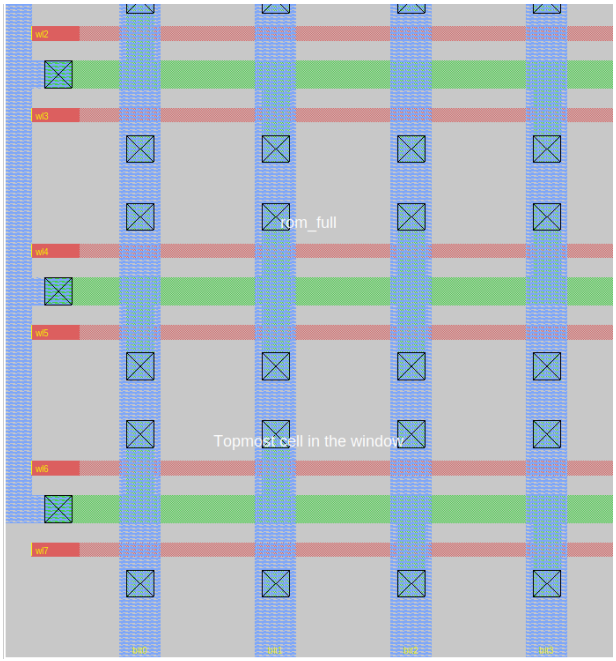
Respectively the other type (which has 1 saved) has only diffusion for the ground line.

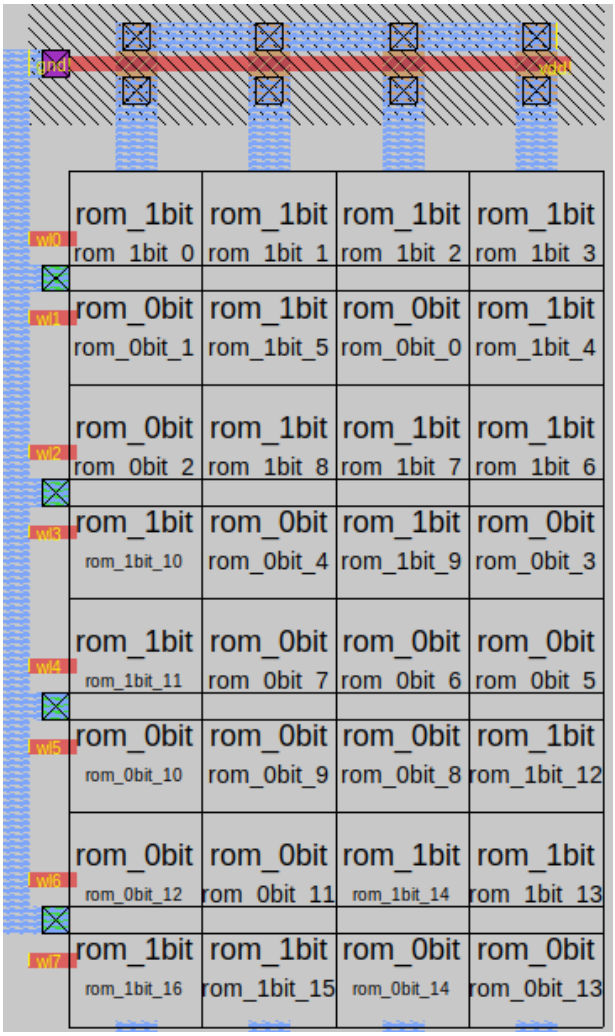
Designing the requester circuit

Below are images of the memory:




rom_full
Topmost cell in the window


rom_full
Topmost cell in the window



In the last image it is easy to see what we have stored in each bit (rom_0bit = 0, rom_1bit = 1), in addition the far right column of the memory is the MSB and as it moved to the left we go to the LSB.

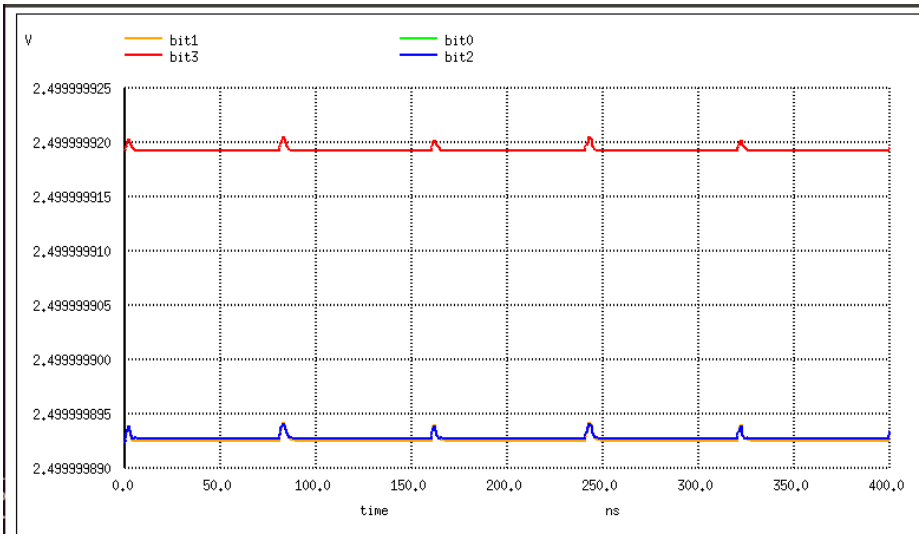
Above the memory we have a pull-up network from PMOS (continuously active) and on the left of the memory we have the connections for diffusion to the ground and the signals wl [0-7] to read from the memory.

It is also worth mentioning that the cells in the 2nd, 4th, 6th, 8th row are rotated 180 degrees so that they are correctly placed in the grid.

Προσομείωση στο NGSPICE

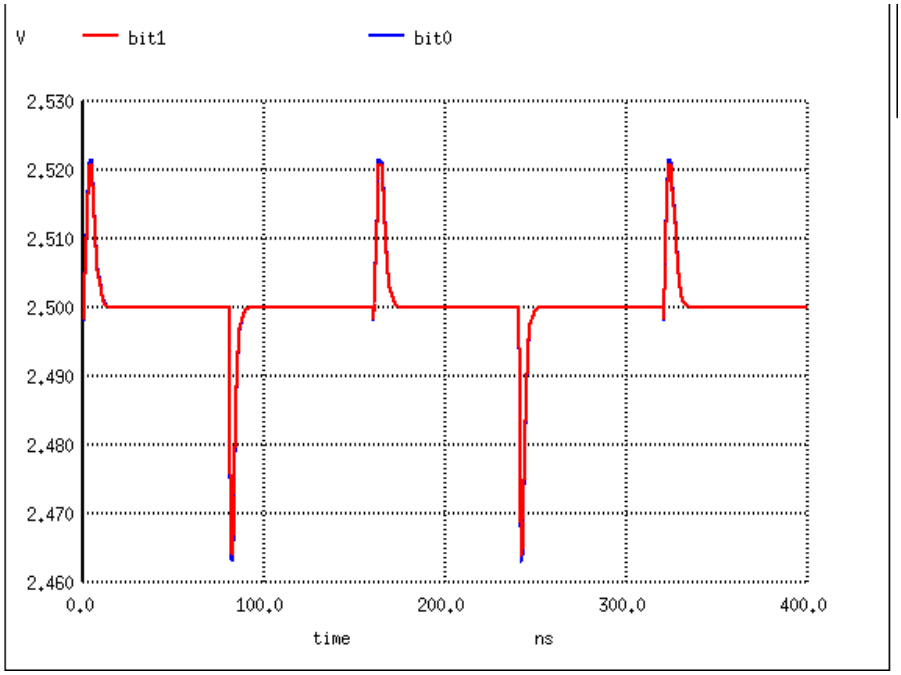
In NGSPICE we give 3 different inputs. Each input, the corresponding wordline, we will give it a pulse so that we go from active to inactive state. In addition we know that in inactive state the ROM has as output ‘1111’.

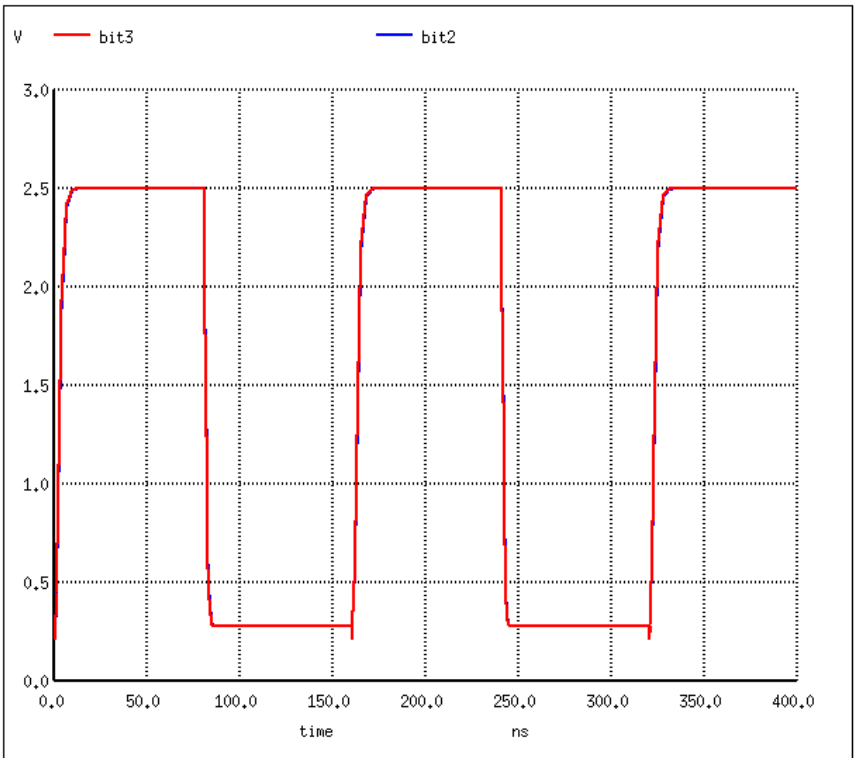
For address 000 (active wl0):



Output is at ‘1111’ which is correct.

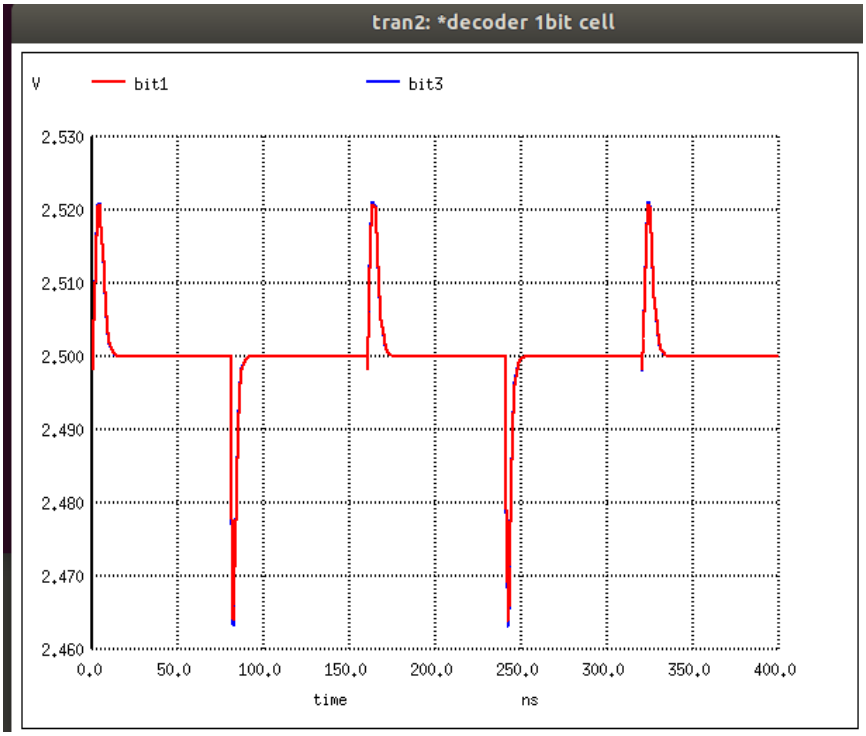
For address 111 (active wl7):

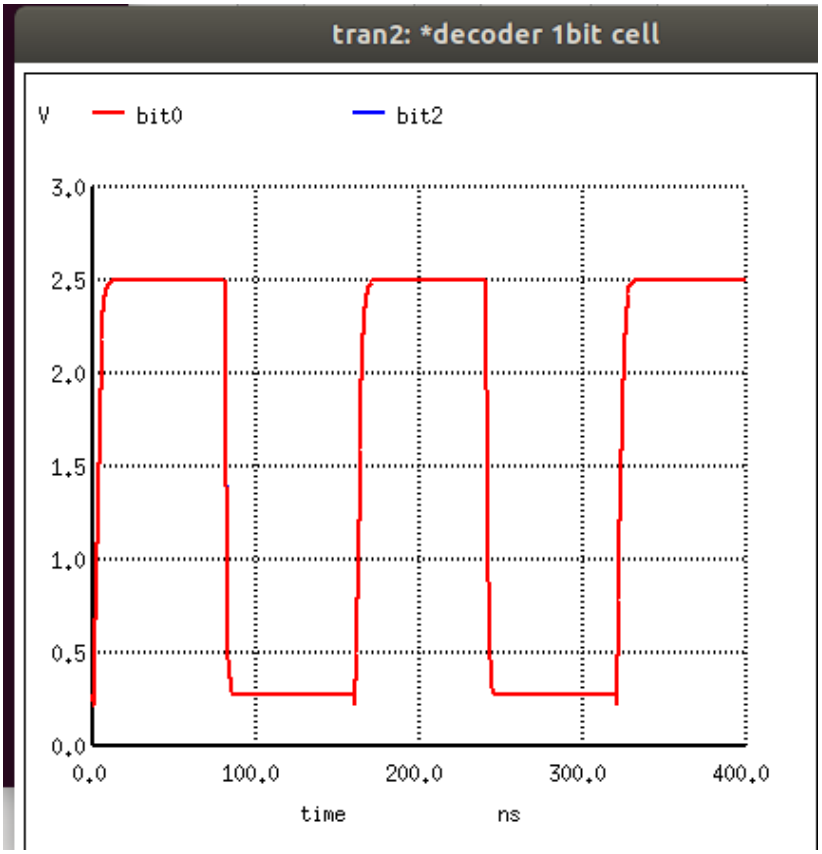




We can see the output set to ‘0011’. At points where wl7 is 1 then we have a drop of bit3, bit2 to 0.

For address 001(active wl1):





We can see the output set to ‘1010’. At the points where wl7 is 1 then we have a drop of bit2, bit0 to 0.

We can and do see in every simulation that our logic 0 is never at 0V (or even close to it).

On the contrary, it remains close to 0.27V in each case of reading with a different input. This voltage is also the V_T of NMOS.