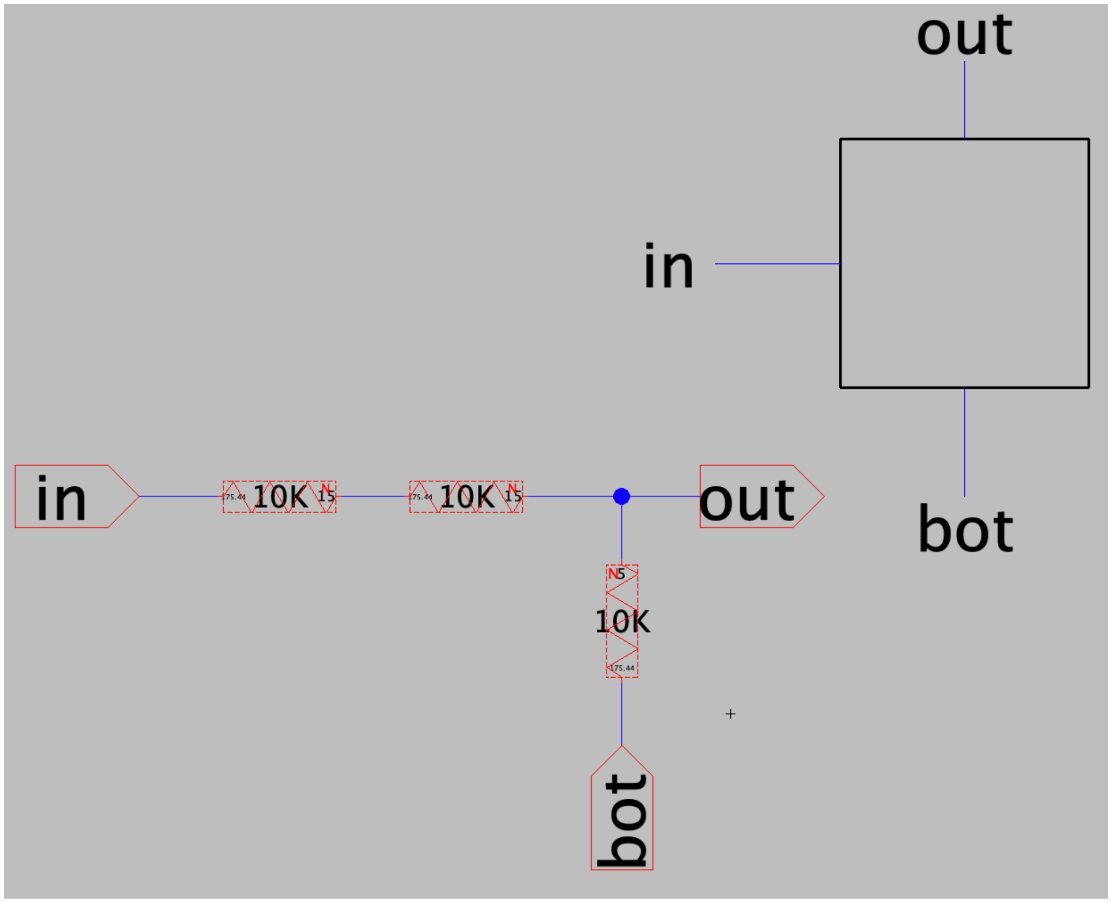
VLSI
September 22, 2025
Winston Har 873637451

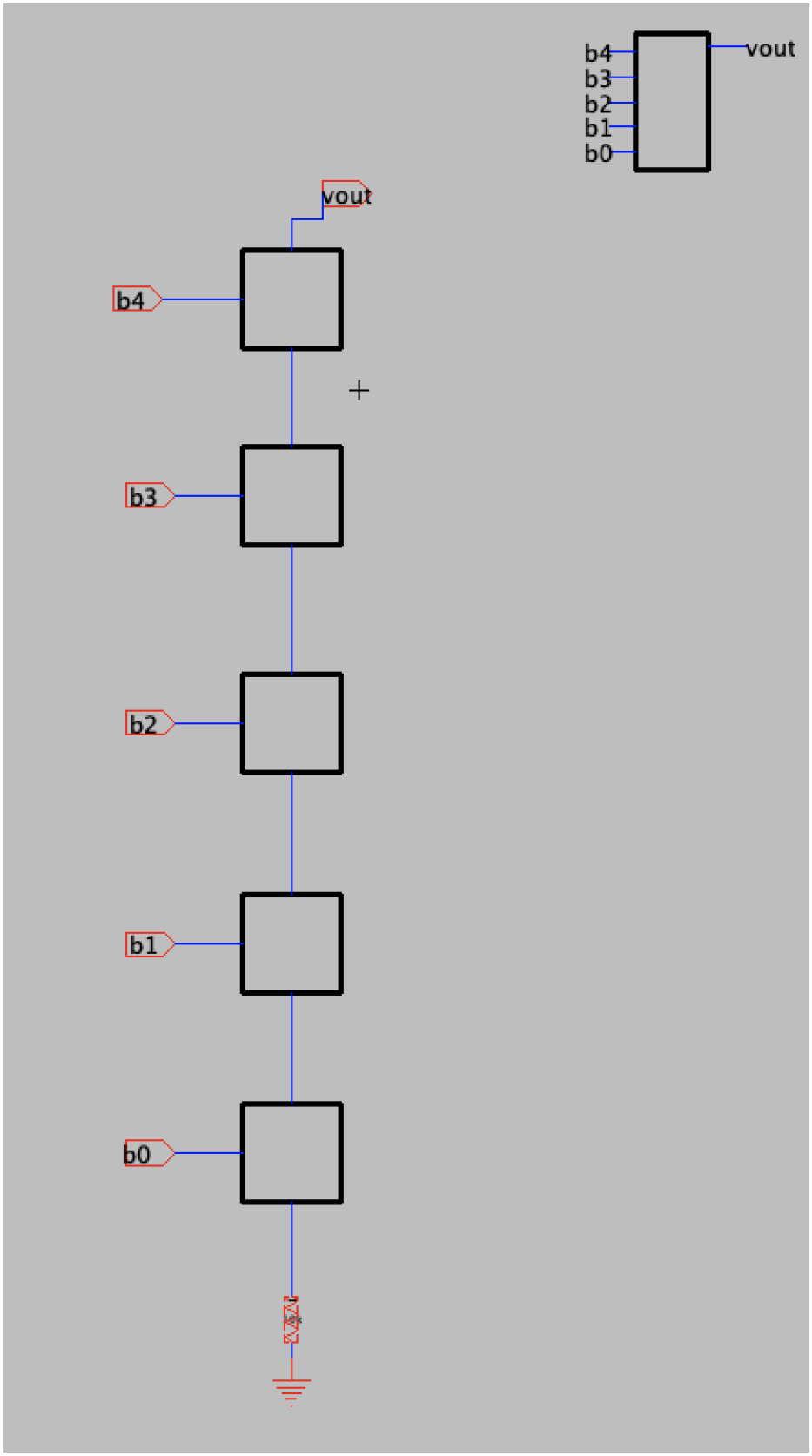
Lab 1: DAC

Introduction:
In this lab I was tasked to create a 5-bit DAC out of 10kΩ's in electric VLSI following the instruction photos posted on canvas. In the process of creating the DAC I had to calculate the necessary length and width of the resistors. Create icons and layouts for components of the DAC and then calculate then simulate various tests to confirm proper functionality. In the Lab work section below are the bullet points from the lab instructional document with corresponding screenshots and bolded text where I wrote in necessary excerpts for each section.

Lab work:

SCHEMATIC
- The design of a 5-bit DAC using an n-well R of 10k
o The 2R resistor should be implemented with two separate 10k resistors in series




b4
b3
b2
b1
b0
vout
vout
b4
b3
b2
b1
b0

- Find and explain how to determine the output resistance of the DAC

To analyze and predict the behaviour of a R02r DAC, you need to determine the output resistance. How? The output resistance can be calculated via the Thevenin equivalent resistance seen from the output flag Vout. Luckily a key characteristic of a R-2R ladder is that the output impedance is constant and equal to R, regardless of the digital input spice code being applied. So, to calculate Thevenin resistance, all independent voltage sources are set to zero. So set VDD supply and b0 through b1 all to ground, making the circuit a passive network of resistors. Then precede to calculate the resistance via simplifying the resistor networking starting from the furthest point from the output (LSB side) and moving towards vout.

At the b0 node: the 2R resistor connected to the b0 input is in parallel with the 2R terminating resistor to ground. The equivalent resistance is: $R_{eq1} = (2R * 2R) / (2R + 2R) = 4R^2 / 4R = R$

At the b1 node: the equiv resistance R_{eq1} is in series with the R resistor between the b0 and b1 node: $R_{eq2} = R + R_{eq1} = R + R = 2R$, Now this 2R resistance is in parallel with the 2R resistor connected to the b1 input: $R_{eq3} = (2R * 2R) / (2R + 2R) = R$

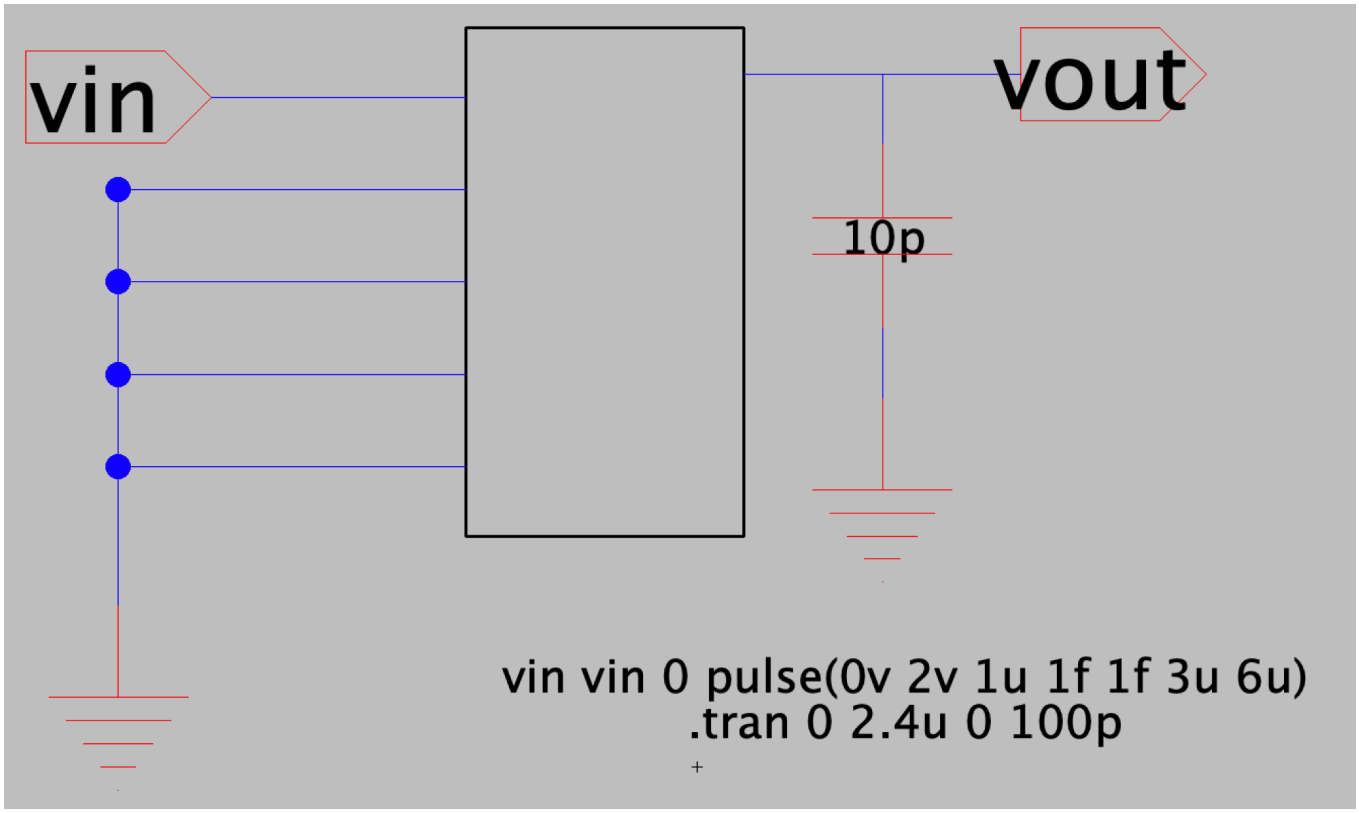
This pattern of series and parallel combinations continues for each bit. Looking from any node b_n towards the LSB side, the equivalent resistance is always R. For example, moving to the b2 node:
The equivalent resistance from the b1 node ($R_{eq3} = R$) is in series with the next R resistor: R + R = 2R. This is in parallel with the 2R resistor at the b2 input, resulting in an equivalent resistance of R.

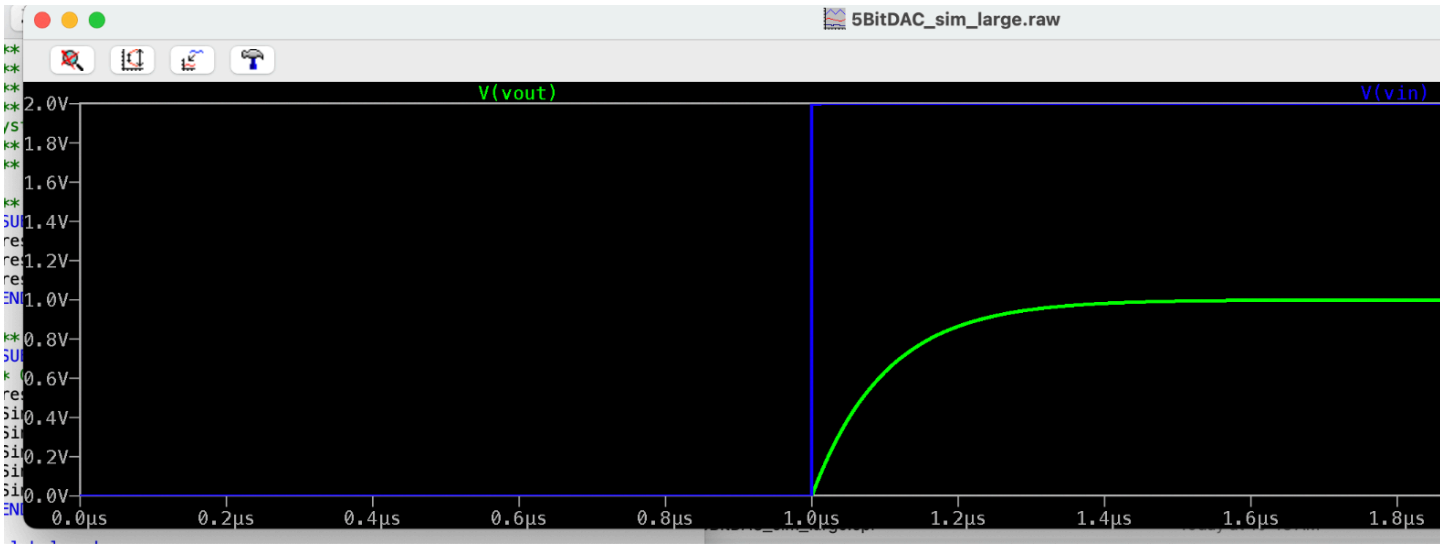
At the Output Node (Vout): This process is repeated up to the most significant bit (MSB), b4. After simplifying the ladder up to the b4 node, you are left with an equivalent resistance of R looking from the b4 node towards the LSBs. This equivalent resistance is in series with the final R resistor connected to Vout. This gives a total of 2R. Finally, this is in parallel with the 2R resistor connected to the b4 input. Therefore, the final Thévenin equivalent resistance seen from the Vout terminal is:
$R_{out} = R_{th} = (2R * 2R) / (2R + 2R) = R$

The output resistance of the 5-bit R-2R ladder DAC is R. Given that the resistors are set at 10kΩ for this DAC, the output resistance of the DAC is 10kΩ.

- Delay, driving a load
o Ground all DAC inputs except B4. Connect B4 to a pulse source (0 to VDD) and show, and predict using 0.7RC, the delay the DAC has

driving a 10pF load





o Verify the simulation results match your hand calculations

Above are the screenshots from my simulation running the DAC with a 10pF capacitive load with a pulse on Vin from 0 to 2v, with other pins grounded. The delay for an RC circuit, which models the DAC output driving a capacitor, can be estimated with the t_delay formula: t_delay /equiv 0.7 * R_out * C_L, with R_out being output resistance, and C_L being capacitive load.
In this case R_out = 10kΩ and C_L = 10 pF.

Substituting these values into formula:
t_delay ≈ 0.7 * (10 * 10^3 Ω) * (10 * 10^-12 F)
t_delay ≈ 0.7 * 10^-7 s
t_delay ≈ 70 ns
So via formula, we predict delay for output to reach 50% of its final voltage to be 70ns.

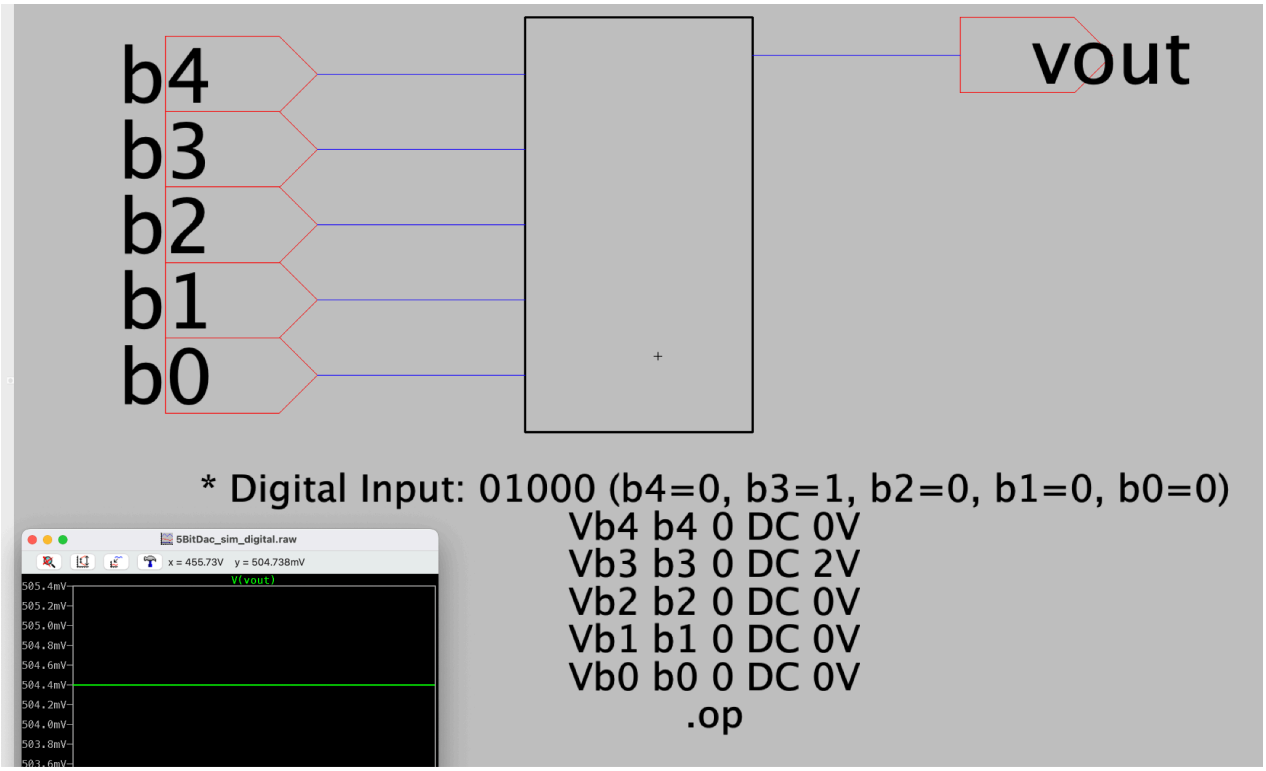
So in the simulated output, as seen in the screenshots above, was ran with vdd at 2v. So the expected final output voltage (V_final) is: V_final = (16 / 2^5)* VDD = .5 * 2V = 1V
The simulation waveform confirms that the output correctly settles at 1V. To delay you just need to find th etime it takes for the output to reach 50% of the final value, which is 0.5v.
The input pulse begins at t = 1.0 micro seconds. Observing the simulation plot, the output voltage V(vout) crosses the 0.5v mark at around 1.07 micro seconds.
So the simulated difference between the two times, t_simulated = 1.07 us - 1.0 us = 0.07 us = 70ns.
This matches the calculated predicted delay.

- Simulations to verify your design functions correctly
o Apply different values to the input of the DAC and check if the output has the correct voltage

To confirm that the 5-bit DAC operates correctly (and to meet the lab requirements), the output voltage was simulated for several digital input codes. The theoretical output voltage of an R-2R DAC is given by the formula: V_out = (Decimal Value of Input / 2^N) * VDD
With N being the number of bits (5) and VDD is the supply voltage (2V).

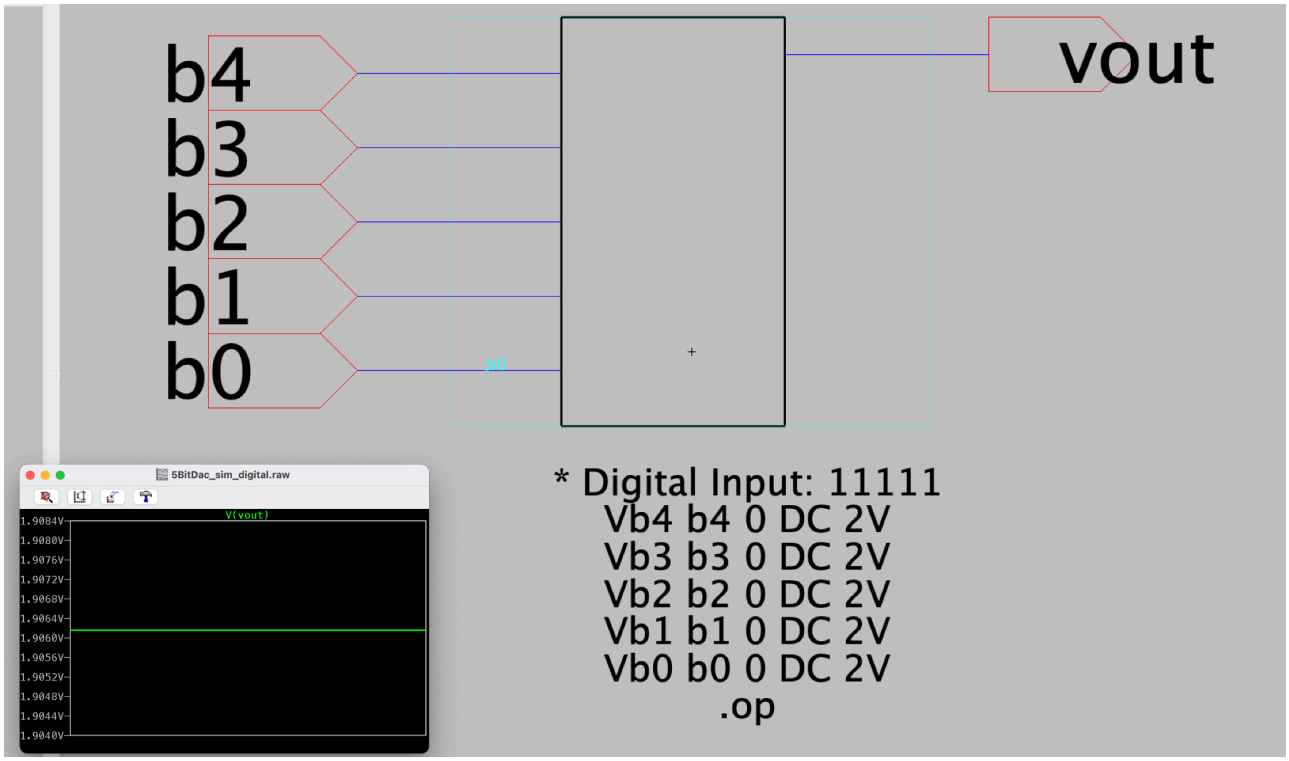
So for a simulation for the digital input 01000 the decimal value is 16, and the theoretical output is then: (8 / 32) * 2 = 0.5. Below is the image of the simulated voltage for 01000 504.4 mV which is within a good margin of error of the theoretical output.

Simulation for 01000



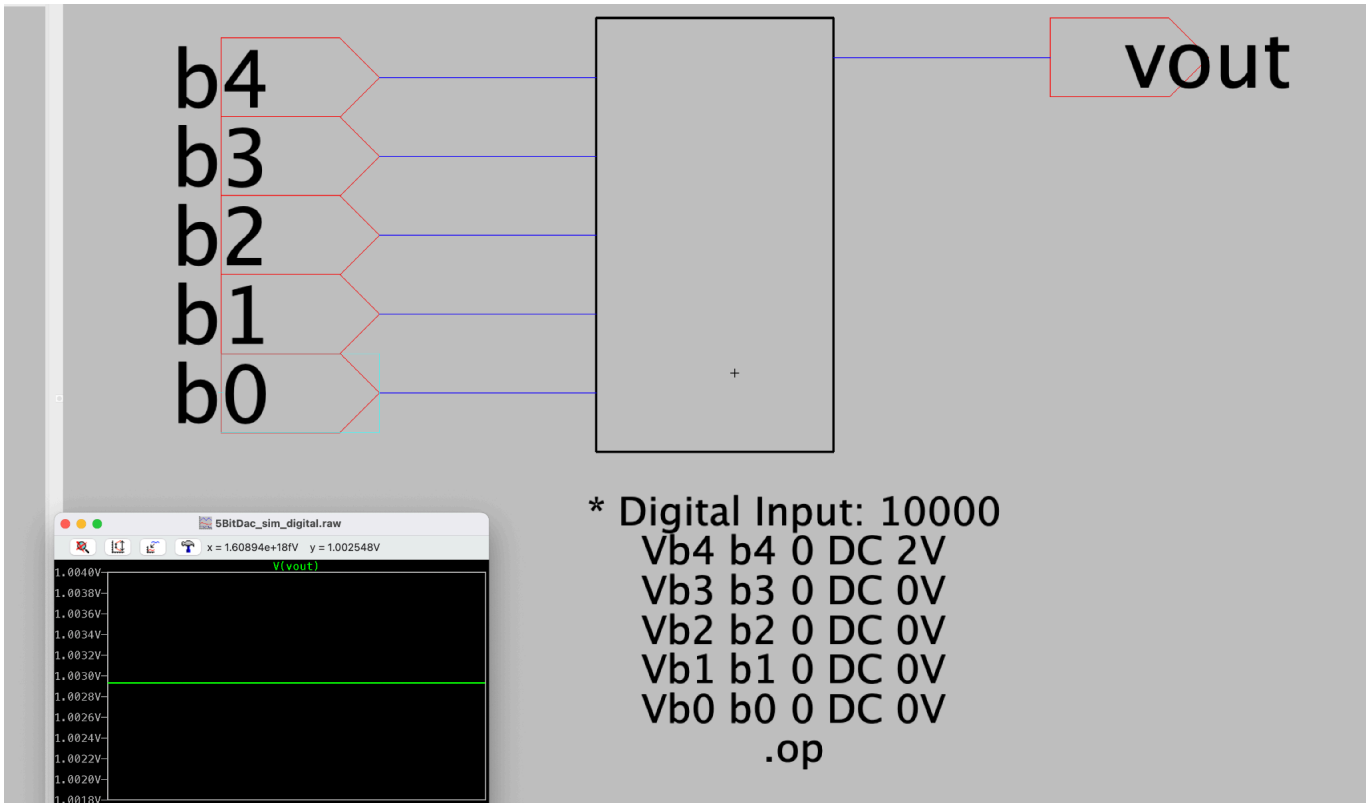
So for a simulation for the digital input 11111 the decimal value is 31, and the theoretical output is then: (31 / 32) * 2 = 1.9375. Below is the image of the simulated voltage for 11111 between 1.9060 V and 1.9064 V which is within a good margin of error of the theoretical output.

Simulation for 11111



So for a simulation for the digital input 10000 the decimal value is 16, and the theoretical output is then: (16 / 32) * 2 = 1.000V. Below is the image of the simulated voltage for 10000 between 1.0028 V and 1.0030 V which is within a good margin of error of the theoretical output.

Simulation for 10000



o Explain what happens if the DAC drives a 10k load

When the DAC drives a resistive load, its performance is affected by its own internal out;ut resistance, which in this case is 10kΩ. If you connect a 10kΩ load resistor (R_L) from vout to gnd, the DAC's output resistance and the load resistor form a voltage divider.

The output of the DAC can be modeled as an ideal voltage source (v_unloaded) in series with its output resistance (r_out). The voltage across the load resistor (v_loaded) is then given by the voltage divider formula: v_loaded = v_unloaded * (R_L/(R_out + R_L))
In this case, R_out = 10kΩ and R_L = 10kΩ.
V_loaded = V_unloaded * (10kΩ / (10kΩ + 10kΩ))
V_loaded = v_unloaded * (10kΩ / 20kΩ)
V_loaded = V_unloaded * 0.5

This means that driving a 10kΩ load will cut the DAC's output voltage exactly in half.

- Use the n-well to layout a 10k resistor
o Discuss, in your lab report, how to select the width and length of the Resistor

The equation you use to determine the width/length for your n-well resistor is this: R = Rs * L/W
We are given R, 10k, the goal resistance, the sheet resistance is given at 855 ohms/square. So, then I decided to start with the default value for width 12, and calculate the length from there:
10k = 855 * L / 12 -> L = 140.35. Upon placing this in Electric VLSI I got an error in my layout N-RWell resistor: "Resolution error: cell 'SingleCell{lay}' node N-Well-Resistor['resnwell@4'] resolution of 0.005(X=-73.675) less than 0.01 on layer N-Active [rule 'Resolution']"
From what I understand this means the resistor I created is too small (or something) so I decided to step up the width to 15 and pray that the larger width would magically fix all my problems.
So, 10k = 855 * L / 15 -> L = 175.44. So now upon placing that on my layout the errors disappear! So that's how I ended up with the L = 175.44 and W = 15.

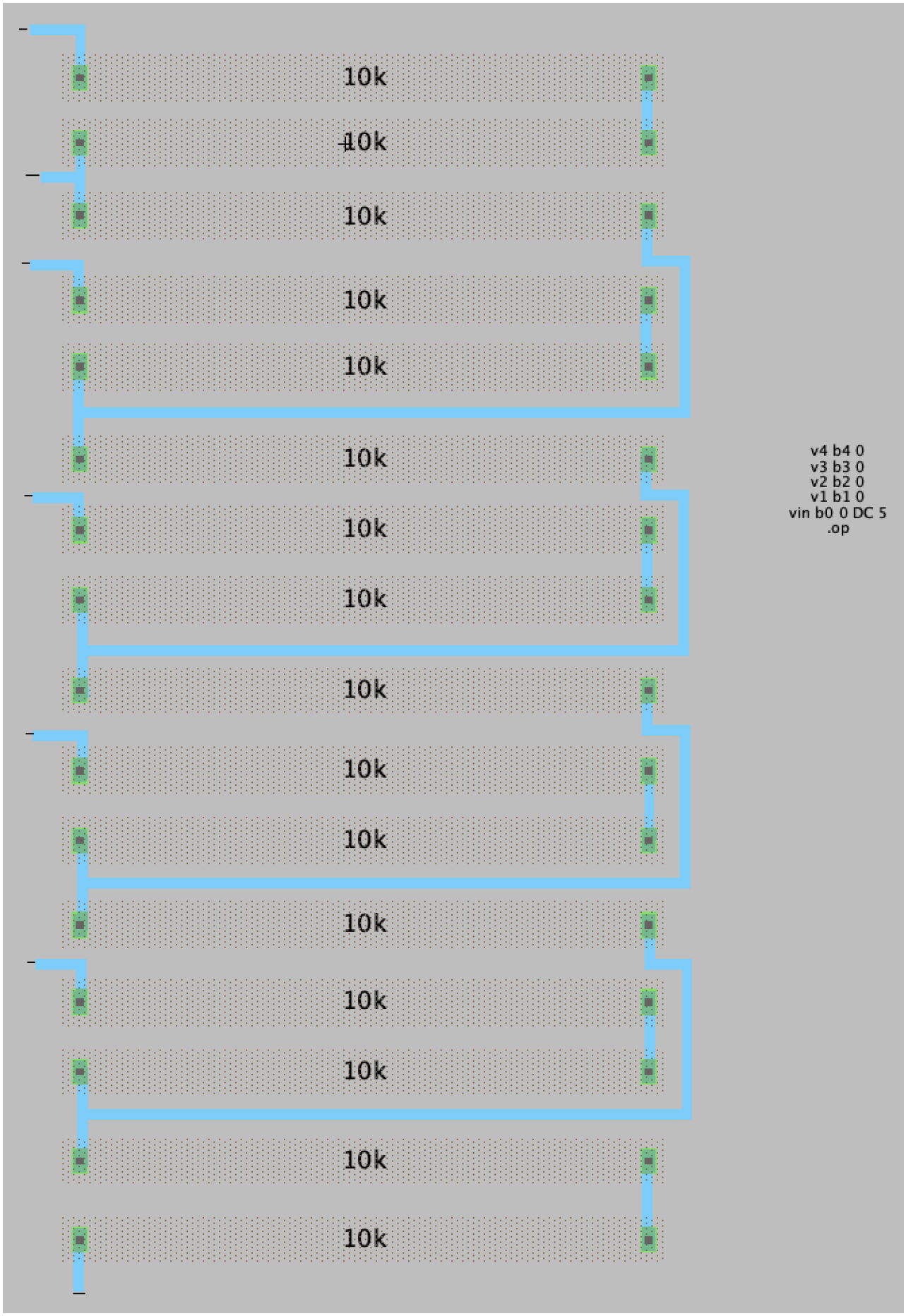
- Use this n-well resistor in the layout of your DAC
- Ensure that each resistor in the DAC is laid out in parallel having the same x-position but varying y-positions (the resistors are stacked – serpentine layout)
- All inputs and outputs Pins should be on metal 1
- DRC and LVS your design (show the results in the lab report)

(**Layout DRC):**
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.0 secs)
Found 18 networks
Checking cell '5BitDAC{lay}'
No errors/warnings found
0 errors and 0 warnings found (took 0.011 secs)

(Layout LVS):
Hierarchical NCC every cell in the design: cell '5BitDAC{sch}' cell '5BitDAC{lay}'
Comparing: Lab_1_DAC:5BitDAC{sch} with: Lab_1_DAC:5BitDAC{lay}
exports match, topologies match, sizes not checked in 0.068 seconds.
Summary for all cells: exports match, topologies match, sizes not checked
NCC command completed in: 0.083 seconds.

(Schematic DRC):
Checking schematic cell 'SingleCell{sch}'

```
  No errors found
Checking schematic cell '5BitDAC{sch}'
  No errors found
Checking icon cell 'SingleCell;1{ic}'
  No errors found
Checking icon cell '5BitDAC{ic}'
  No errors found
0 errors and 0 warnings found (took 0.001 secs)
```


10k
10k
10k
10k
10k
10k
10k
10k
10k
10k
10k
10k
10k
10k
10k
v4 b4 0
v3 b3 0
v2 b2 0
v1 b1 0
vin b0 0 DC 5
.op

- Zip your final design and report and submit it on canvas

Conclusion:
Overall I thought that this lab was pretty interesting and challenging. When making the layouts the issue with the sizing for the resistors was very befuddling until I got to the source of the error. Placing the N-RWell's in all the layouts proved to be a little fiddly as well but with some patience weren't too bad to do either. I feel like I learned a lot in the process of making all the different components in Electric VLSI. This definitely helped me get a better grasp of n-wells, schematics and layouts for VLSI, in comparison to where I started at the beginning of this lab.