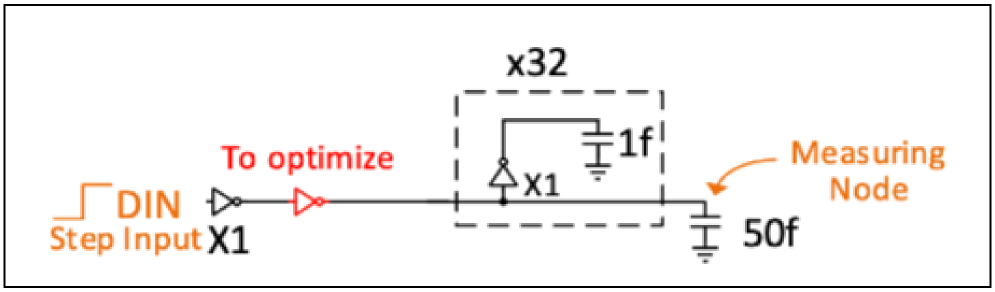
Lab 1.2: Script-based HSPICE Simulation

ELEC521, Felix Tse (Fall 2025)



Introduction

The experiment aims to optimize this circuit for the number of stages and the driver size of each stage. First, create an equivalent circuit "template" for simulation with a SPICE netlist. Assume that there are currently no stages in the template. Given the driver X1 and observing that each bitline is also driven by X1, it follows that the circuit, with 0 stages, is accurately modeled as such:

```
*driver
X_driver0 DIN node_measure vdd vss inv
*add stage inverter instantiations here

*32 load inverters and 1f cap
X_load1 node_measure out1 vdd vss inv
c1 out1 vss 1f

X_load2 node_measure out2 vdd vss inv
c2 out2 vss 1f
…

*50f load capacitance
c_wire node_measure vss 50f
```

To optimize, assume that the number of stages will not exceed 9 (run some experiments to see if this is the optimal max number of stages). The idea is to fix the number of stages, and sweep parameters in a multivariable optimization using gradient descent to find the best sizing for each stage.

Script General Flow

1. Create a copy of *ft33_netlist.sp*, called *ft33_nlistcopy.sp* to avoid modifying original .sp netlist
2. For N stages, add N base parameters and N instantiations of inverter to *ft33_nlistcopy.sp*. All modifications and simulations are now in reference to this netlist. Initial values for parameters are controlled by a "scale", e.g. scale = 2, then stage 1 = 2^1, stage 2 = 2^2, stage 3 = 2^3, or e.g. scale = 4, then stage 1 = 4^0, stage 2 = 4^1, stage 3 = 4^2
3. Run the simulation to get a base delay (from .mt0)
4. Add .alter parameters to sweep parameters, "wiggle" by a given step size
5. Run the simulation to get N altered delays (in .mt1, .mt2, .mt3 ….), parse with *ft33_parse.py* to *output{n}.txt*
6. Compare the altered delays to the base delay to get a gradient
7. Use gradient to find new sizes that reduce delay ($size_{new} = size_{old} - learningrate * gradient$)
8. Change base parameters with new sizes, rerun simulation to get new base delay (from .mt0)
9. Repeat until sizes no longer change, or reaching 20 iterations

Results

Below is a table summarizing the results from running simulations with 1 stage, 3 stage, 5 stage, 7 stage, and 9 stage. Step size was 0.1, learning rate was 0.01.

Stages	**Scale**	**Initial Sizes**	**Optimized Sizes (*n*)**	**Delay *(ps)***	**Energy (*pJ)***
1	2.5	2.5	5.86	181.09	0.55
3	2.5	2.5, 6.25, 15.62	2.68, 6.39, 15.85	160.41	0.68
5	2.5	2.5, 6.25, 15.63, 39.075, 97.69	2.67, 6.4, 15.83, 39.23, 97.76	207.00	1.64
7	1.5	1.5, 2.25, 3.38, 5.06, 7.59, 11.39, 17.09	1.71, 2.71, 3.94, 5.63, 8.19, 12.0, 17.84	224.42	0.88
9	1.5	1.5, 2.25, 3.38, 5.06, 7.59, 11.39, 17.09, 25.63, 38.44	1.74, 2.73, 3.98, 5.62, 8.2, 11.99, 17.71, 26.25, 39.03	258.19	1.35

From the table above, the delay improves from one stage to three stages by 21 *ps*, then continues to increase past three stages. This is likely due to the tradeoff of increased parasitic delay with the number of stages: therefore it can be concluded that three stages is the optimal number of stages, with a delay of 160.41 *ps*. The size of the drivers is a multiplier, applied to the base size of 900 *n* and 360 *n* for the PMOS and NMOS respectively. Sizing of the drivers was based on an initial scale, ideally between 2.4 and 6 (optimal stage effort), but due to sizing limits of the Skywater PDK, 2.5 was used for 1, 3, 5 stages while 1.5 was used for 7 and 9 stages. It can also be seen that energy slightly increases with the number of stages, but dramatically increases based on the size of the driver at each stage. These results are consistent with the theoretical model, $E = CV$, since energy is proportional to capacitance and capacitance is proportional to size.

Logical Effort & Comparison to Simulation (3 Stage)

From simulation, $C_{in} = cgtot_{nmos} + cgtot_{pmos} = .314\,fF + 1.224\,fF = 1.538\,fF$

$$F = GBH$$

$$G = 1 * 1 * 1 * 1$$

$$B = \frac{C_{on} + C_{off}}{C_{on}} = \frac{50\,fF + 32(1.538)\,fF}{50\,fF} = 1.984\,,$$

$$H = \frac{C_{out}}{C_{in}} = \frac{(50)\,fF}{1.538\,fF} = 32.50$$

$$F = 64.4904$$

$$F^{\frac{1}{N}},\ N = 4,\ F^{\frac{1}{4}} = 2.83$$

$$D_{handcalc} = N * F^{\frac{1}{N}} + P,\ P = 4,\ D = 15.33\tau$$

From simulation of X1 FO4 inverter, $\tau = 9.52\,ps$

$D_{handcalc} = 145.98\,ps$

Percentage difference: $\frac{(160.41 - 145.98)\,ps}{160.41\,ps} = \boldsymbol{8.99\%}$

Conclusion

The delay of the circuit was optimized at 3 stages with sizes [2.68, 6.39, 15.85], yielding 160.41 *ps* from HSPICE simulation and 145.98 *ps* from hand calculated logical effort. The results are in reasonable agreement, which validates the logical effort model for first-order approximation of the delay in digital circuits. The 8.99% difference can be attributed to the ideal assumptions of logical effort. Some possible reasons could be logical effort failing to account for wire capacitance, the assumption of an instantaneous step input, and idealized parasitic delay (ideally 1 unit for each inverter but in reality is non-linear and increases with size). As such, the simulation delay is expected to be a bit higher than the logical effort delay, which is consistent with the results.