Lab 05: What it costs

In this lab you compare two layouts of the same two transistors, your Lab 03 inverter and the foundry's `inv_1`, to see what the drawing costs in area and in delay. You start by working out the area price from numbers you already have. Then you extract both cells with every parasitic capacitance kept, simulate them, and check your arithmetic against the measurement.

Package: `labs/lab-05-what-it-costs/`

Before you start

- Get Lab 03 to `PASS`.
- Read Area, parasitics and delay. This lab also uses Extraction.
- Have paper or a text file ready for the numbers you work out in Parts 1 and 5.

Time: about 1 hour, not counting the optional Part 6. Each full `make` takes about half a minute (27 to 39 s on seven timed clean runs, depending on what else the machine is doing). Nearly all of that time goes on the two ngspice runs, about 13 s for the fan-out deck and 15 s for the two ring oscillators, which between them have 62 extracted subcircuits to simulate for 20 ns of transient. Each run starts by reading the whole SKY130 model library.

Part 1: Predict the price

How much does the drawing cost in area, judging only from numbers you have already seen?

1. The Lab 03 cell is 3.00 × 6.00 µm, and Lab 01 measured `inv_1` at 1.38 × 2.72 µm. Work out how many times bigger the Lab 03 cell is, and write it down.
2. The extracted netlists give each transistor's drain junction area as `ad=`: 0.715 µm² for the Lab 03 NMOS and 1.1 µm² for its PMOS (Lab 03, Part 9), against 0.169 µm² and 0.26 µm² for `inv_1` (Lab 01, Part 5). Work out both ratios and write them down.
3. Guess how much slower the Lab 03 cell will be. Is it closer to the area ratio, the junction ratio, or much less than either? Write down a number, because Part 3 compares it with the measurement.

You should see three ratios and a guess in your notes. Part 2 checks the ratios against the table.

Part 2: Run the comparison

Does the measurement agree with your arithmetic?

1. Go into the lab package and run it:

```
cd labs/lab-05-what-it-costs
make
```

   By default it compares the **reference** Lab 03 inverter, in `mag/my_inv.mag`, with the foundry's. The reference uses the same transistor and wiring coordinates as Lab 03's steps, so if you followed those steps, the `yours` column describes the cell you drew. Part 6 (optional) swaps in your own file instead. The run prints:

```
  THE SAME TWO TRANSISTORS, DRAWN TWICE

                                      yours       foundry           ratio
  ------------------------------------------------------------------------
  cell width                              3          1.38 um       2.17 x
  cell height                             6          2.72 um       2.21 x
  cell area                              18         3.754 um2      4.80 x
```

```
  NMOS: sky130_fd_pr__nfet_01v8
     W                                0.65          0.65 um       1.00 x
     L                                0.15          0.15 um       1.00 x
     drain junction area             0.715         0.169 um2      4.23 x
     drain junction perim              3.5          1.82 um       1.92 x
  PMOS: sky130_fd_pr__pfet_01v8_hvt
     W                                   1             1 um       1.00 x
     L                                0.15          0.15 um       1.00 x
     drain junction area               1.1          0.26 um2      4.23 x
     drain junction perim              4.2          2.52 um       1.67 x

  extracted capacitance, total         1.86         1.424 fF       1.31 x
       ... of it on the output       0.4268        0.3519 fF       1.21 x

  DELAY, fan-out of 4
  t_pHL                              37.32         33.95 ps       1.10 x
  t_pLH                              88.43         81.51 ps       1.08 x
  average                            62.88         57.73 ps       1.09 x

  RING OSCILLATOR, 31 stages, nothing but the cell driving itself
  period                              2640          2034 ps       1.30 x
  frequency                          378.8         491.7 MHz      0.77 x
  one stage                          42.58          32.8 ps       1.30 x
  ...

PASS  Both cells extracted with parasitics, both simulated, and the
      comparison is above. There is no wrong answer here - it is a
      measurement, and yours is yours.
```

2. Find the four `1.00 x` rows for W and L. They're the control in this experiment: the transistors in the two cells are identical, down to the model name, so every other difference in the table comes from the way they were drawn.
3. Compare the `cell area` row and the two `drain junction area` rows with the ratios you worked out in Part 1.

You should see `1.00 x` on every W and L row, `4.80 x` for the cell area and `4.23 x` on both drain junction area rows, matching your Part 1 ratios to rounding, and `make` ending in `PASS`.

Part 3: Compare the two delay rows

Why does the same drawing cost more delay in one test than in the other?

1. Find the two delay ratios in your table: the fan-out 4 `average` row and the ring `one stage` row. On this page they are 1.09 x at fan-out 4 and 1.30 x in the ring, so the cell is 9 % slower in one test and 30 % slower in the other. Compare yours with the guess you wrote down in Part 1.
2. Work out why the two ratios differ before you read the `WHAT TO TAKE FROM THIS` block at the end of your `make` output, because that block gives the answer. Hint: a gate's delay is how long it takes to charge whatever hangs off its output, and that load has two parts: what you are driving, and what you brought with you.

   Answer: A1, in Answers at the end.

You should see that the self-loaded test (the ring) shows the bigger penalty.

Part 4: Look at the capacitors

Where does the extra capacitance live?

1. Print both parasitic netlists side by side:

```
make caps
```

2. Compare the two capacitance rows with the drain junction areas from Part 1:

```
  extracted capacitance, total          1.86         1.424 fF       1.31 x
       ... of it on the output        0.4268        0.3519 fF       1.21 x
```

   Your cell has more extracted capacitance than the foundry's, but only 1.2 to 1.3 times as much, even though its drain junctions are 4.23 times the size. As Area, parasitics and delay explains, the extracted `C` elements are the capacitance of the wiring and the well, and the drain junctions aren't among them.

3. Where does the junction capacitance of the source and drain diffusions live, and why isn't it in the `C` lines?

   Answer: A2, in Answers at the end.

You should see the `C` lines for both cells, with no junction capacitance among them.

Part 5: Read the ring oscillator

How long does one gate take when it drives nothing but a copy of itself?

A loop of an odd number of inverters can't settle, because whatever value a node holds comes back inverted one lap later. So the loop oscillates, with period $2Nt_p$ for N stages of delay t_p each. That makes it the cleanest way to measure a single gate delay, because there is no source and no external load, only the cell driving one copy of itself.

1. Open `src/mkring.py`, which writes both 31-stage rings. It is about fifty lines long, and `N = 31` near the top sets the number of stages. Near the end it starts each ring's nodes alternately low and high, so that a single edge runs round each ring.

2. From the `period` row of your table, work out the delay of one stage for each cell, $t_p = T/(2 \times 31)$, and write both down. Then compare them with the `one stage` row.

   Answer: A3, in Answers at the end.

The foundry's 33 picoseconds is the floor for this process at 1.8 V: it's the delay of one gate driving nothing but a copy of itself. The capstone's timing numbers are multiples of it, and What sets the clock frequency builds on it.

You should see your two stage delays agree with the `one stage` row, to rounding.

Part 6: Measure your own cell (optional)

1. Copy your Lab 03 cell in and run the comparison again:

```
make mine MAG=../lab-03-draw-an-inverter/mag/my_inv.mag
make
```

   A cell drawn by following Lab 03 lists its ports as `VGND VPWR A Y`, because the frame you started from already had `VGND` and `VPWR` labelled, while the reference lists them as `A Y VPWR VGND`. So `make` reads the order out of your extracted netlist, wires both decks around it, and prints it on its `wrote results/fo4.spice` line. If your cell is missing a port, `make` stops there instead and names the ports it found.

2. Check the W and L rows. If they don't read `1.00 x`, your transistors aren't the ones Lab 03 asked for, so the comparison no longer isolates the effect of the drawing.

To go back to the reference cell afterwards, run `git checkout mag/my_inv.mag` and then `make`.

You should see `1.00 x` on the W and L rows. If you used Lab 03's coordinates exactly, the whole table is the same as the reference's.

Finishing

You're done when the area and drain-area ratios you worked out in Part 1 match the table's `cell area` and `drain junction area` rows, and the stage delays you worked out in Part 5 match its `one stage` row.

Questions:

- Why does the ring show a bigger penalty than fan-out 4? Part 3 answers it.
- Where does junction capacitance live, and why is it not in the `C` lines? Part 4, step 3 answers it.

Submit: nothing for this lab.

Going further (optional)

`make table` prints the comparison again from the results already on disk, so it returns instantly. `make pex` runs only the extraction, which takes well under a second.

Next on the Course path: read Routing by hand and What sets the clock frequency, then do the Capstone: Place and route.

Answers

A1. Part 3: Compare the two delay rows

At fan-out 4 most of the load is the four downstream gates, and their input capacitance is the same whichever cell is driving them. In the ring each stage drives one gate and nothing else, so your 4.23 × junction area makes up a large share of the total load. Area, parasitics and delay has the full explanation.

A2. Part 4: Look at the capacitors

It's inside the device model, which computes it from the `ad`, `as`, `pd` and `ps` parameters on each device line, and your drain areas are 4.23 × theirs. Because it lives in the model, you can't read it off the parasitic netlist; the only way to see its effect is to simulate, which is why this lab does.

A3. Part 5: Read the ring oscillator

For the table on this page: 2640.2 ps / 62 = 42.58 ps for your cell, and 2033.6 ps / 62 = 32.80 ps for the foundry's, which the table prints as `32.8`. The table rounds the periods to 2640 and 2034, which is why dividing the rounded numbers gives 42.58 and 32.81.