hw08-solution

2020 Dan White

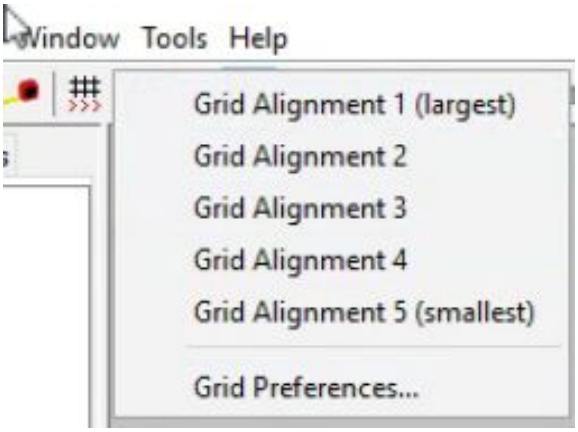
The JELIB for this is available by request.

See the hw07-solution for the corresponding schematic views.

1. Layout strategy

Notice across the layouts at every level that **all** metals (m1 and m2) are **always** 4 λ wide/tall and are **always** on a 4 λ grid spacing. The Weste and Harris book talks about this sort of layout strategy using keywords of wiring **track**s and **pitch** in sections:

- 1.5.5
- 1.10.2
- 1.10.3
- End of 3.3.3
- 6.1

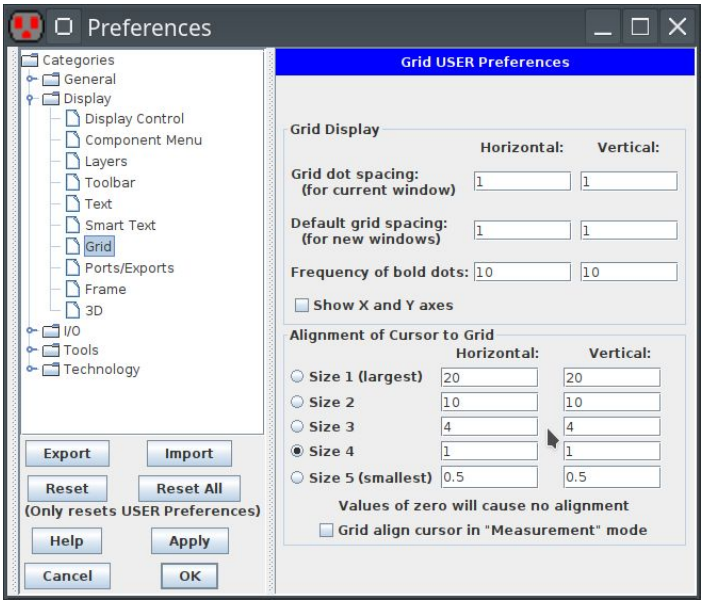


How can you easily get things on a 4 grid? Easy! Just change your grid settings:

Which pops up the Preferences window. My personal settings change **Size 3** to 4 units instead of the default value of 5.

You can also change the Grid Display options to put grid dots at other intervals.

Electric has a *ton* of options. Want to know how I know these things? I read the manual, I search in the manual, and I act with curiosity to discover if there are things that "would be nice."

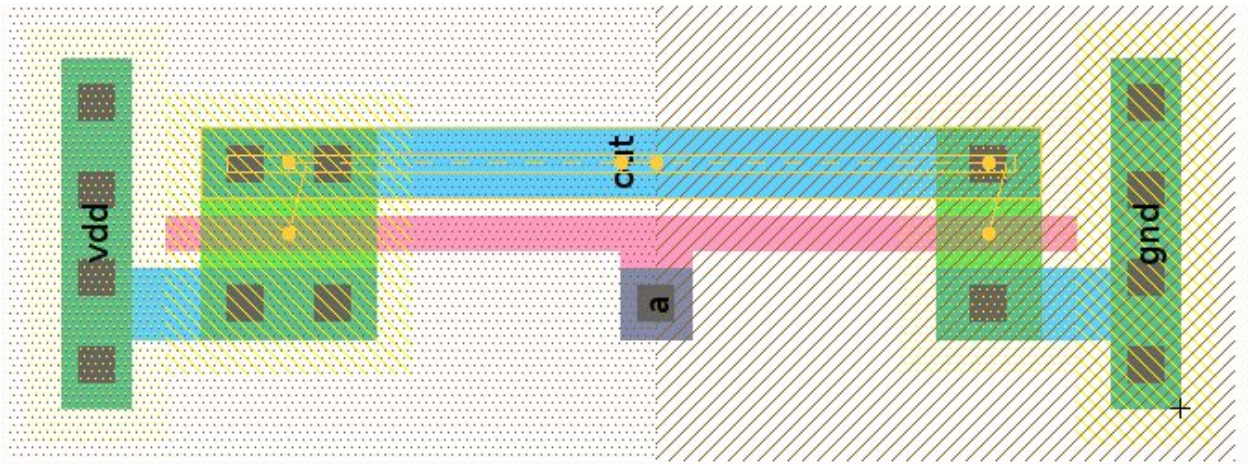


2. inv-x1

Notice the highlighted **out** pin and export? That pin is *exactly* the same size as the patch of metal that connects the two transistor drains. In Electric, this allows a horizontal arc (wire) that connects the output terminal of the inverter to somewhere else to the right (or left!) to slide vertically.

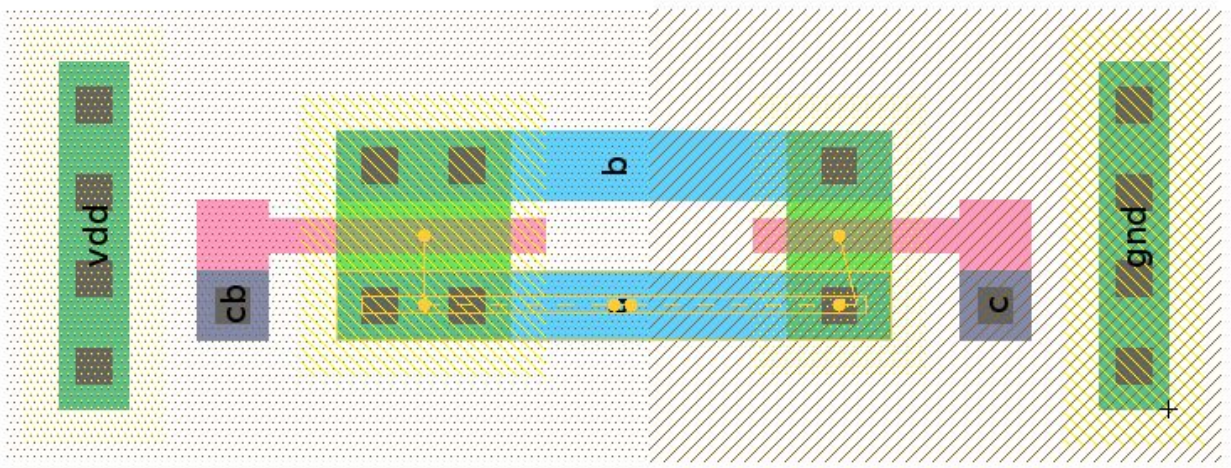
Create this huge pin with a ~~three~~ five-step process:

1. Place a pin component for the metal layer that you want the export on somewhere **to the side** of its intended location. These are on metal1.
2. Change the size of the pin to the same x and y dimensions as the patch of metal.
3. Select the pin and wire it to the metal to tell Electric they should be connected.
4. Move the pin so it is directly on top of the corresponding metal arc.
5. Create an export on the pin.



3. txgate-x1

The switch terminal **a** is highlighted to show the large pin+export technique that was done for both **a** and **b** exported terminals.

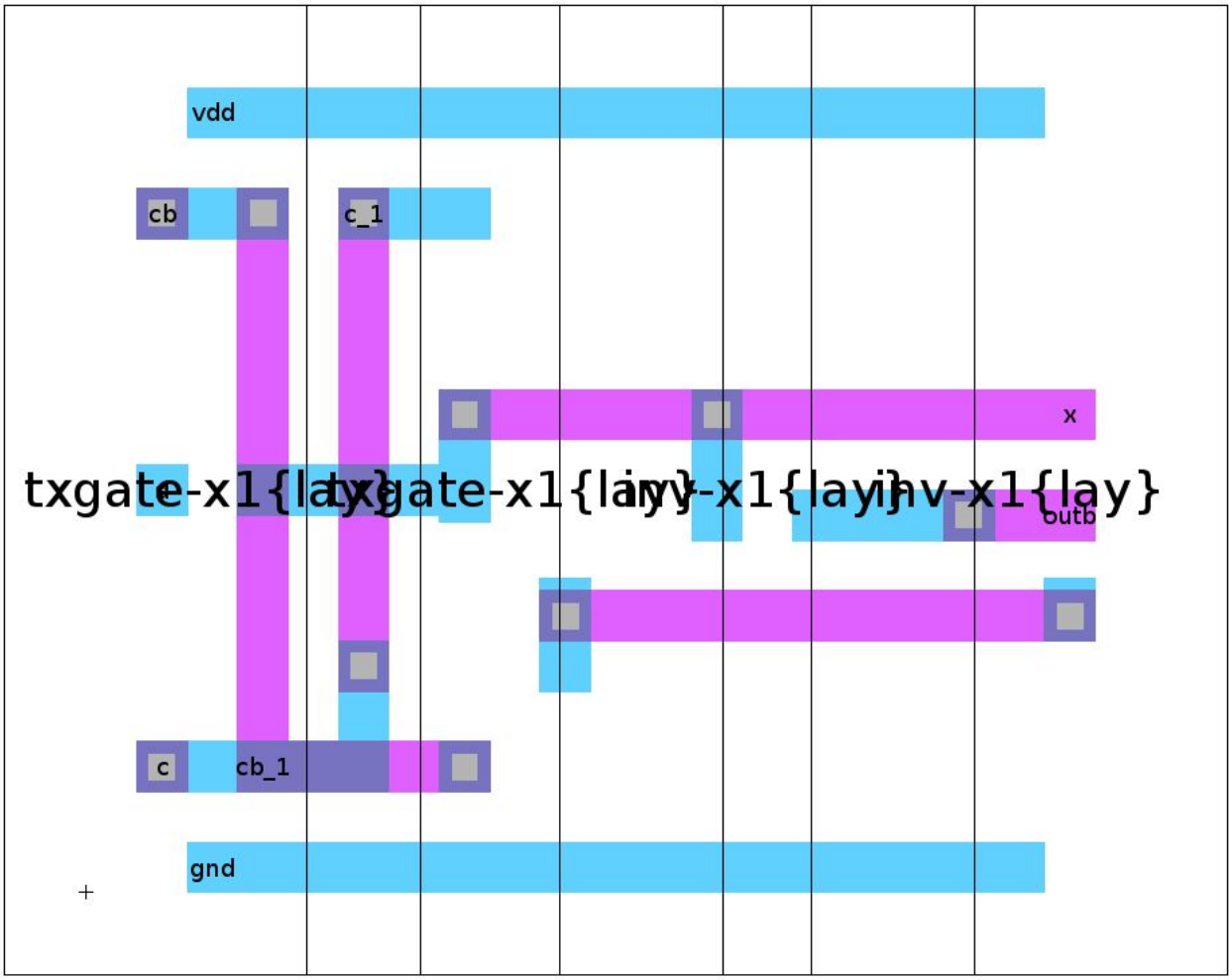


4. dlatch-x1

There are two exports attached to each of the clock inputs:

c and **c_1**, **cb** and **cb_1**

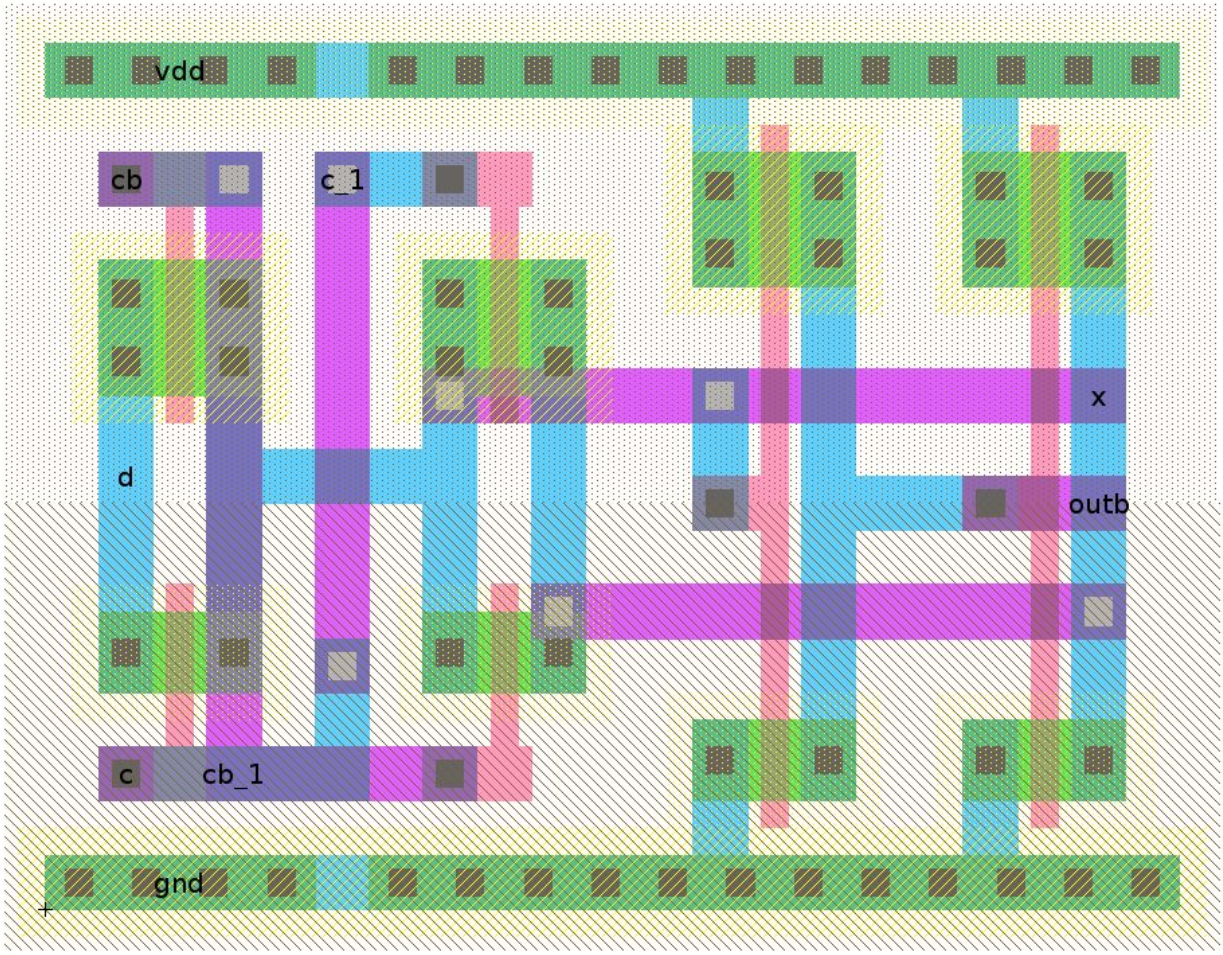
In Electric, *exports* are the ONLY place where a connection can be made from the current cell's layout to a subcell. This even though there are entire patches of metal that are electrically connected. Placing a few extra exports on these pins for the clock inputs allows a much easier connection at the DFF level when the clock nodes need to, again, cross wiring. The latch already has conductive patches for each of **c** and **cb** at on the upper and lower edges of the cell.



Notice how the **vdd** and **gnd** rails are abutted for the three rightmost cells? The cross-connection of the transmission gate clock signals required that the two txgate cells be spread apart by another 4 λ to make space for the vertical metal2 arcs.

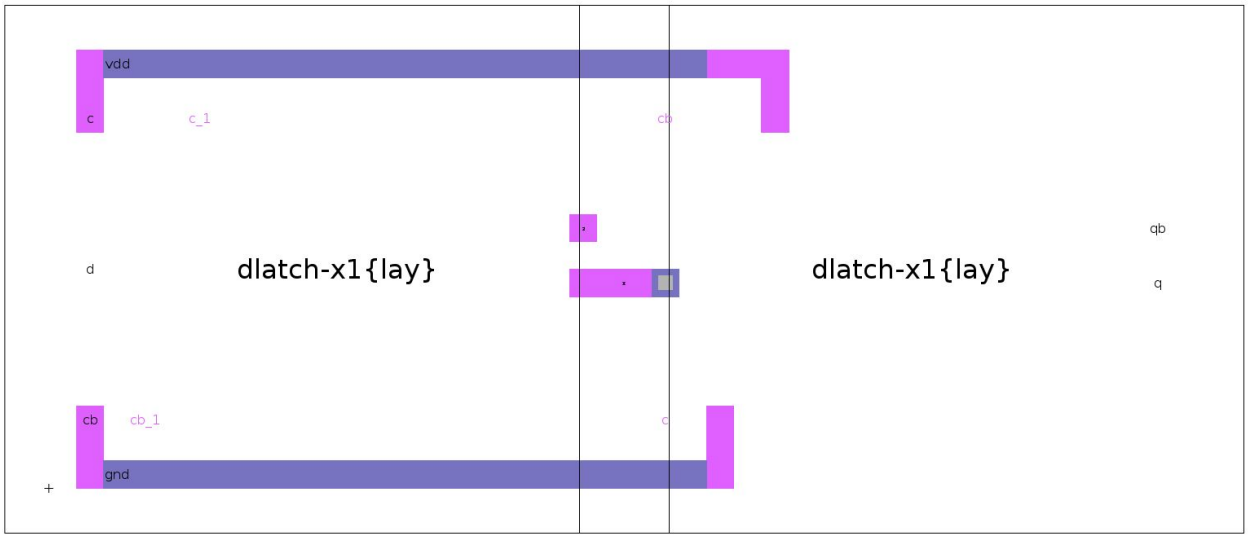
For the cells that abut, Electric has a command to detect and connect the **vdd** and **gnd** rails. This *explicit* connection is, remember, just the way Electric works → there must be drawn arcs connecting nodes together otherwise the DRC tool will flag a bunch of errors for things that are touching but not (apparently) supposed to be electrically connected.

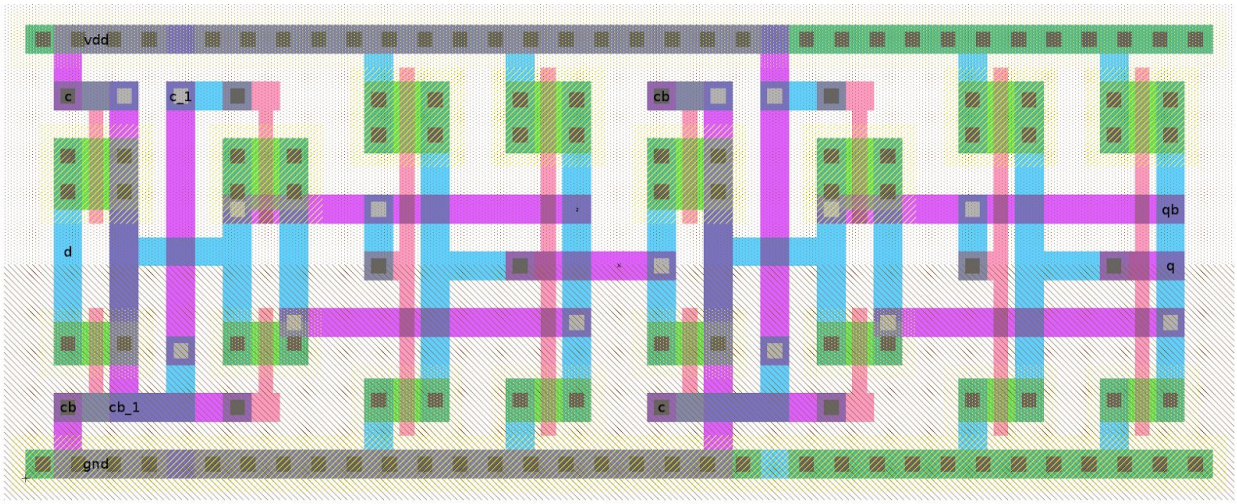
Tools → Routing → Auto-Stitch



5. dff-x1

By the time we are wiring up the DFF from the two latches you begin to see the wiring strategy payoff. We only route wires (arcs in Electric) on a 4 λ grid.

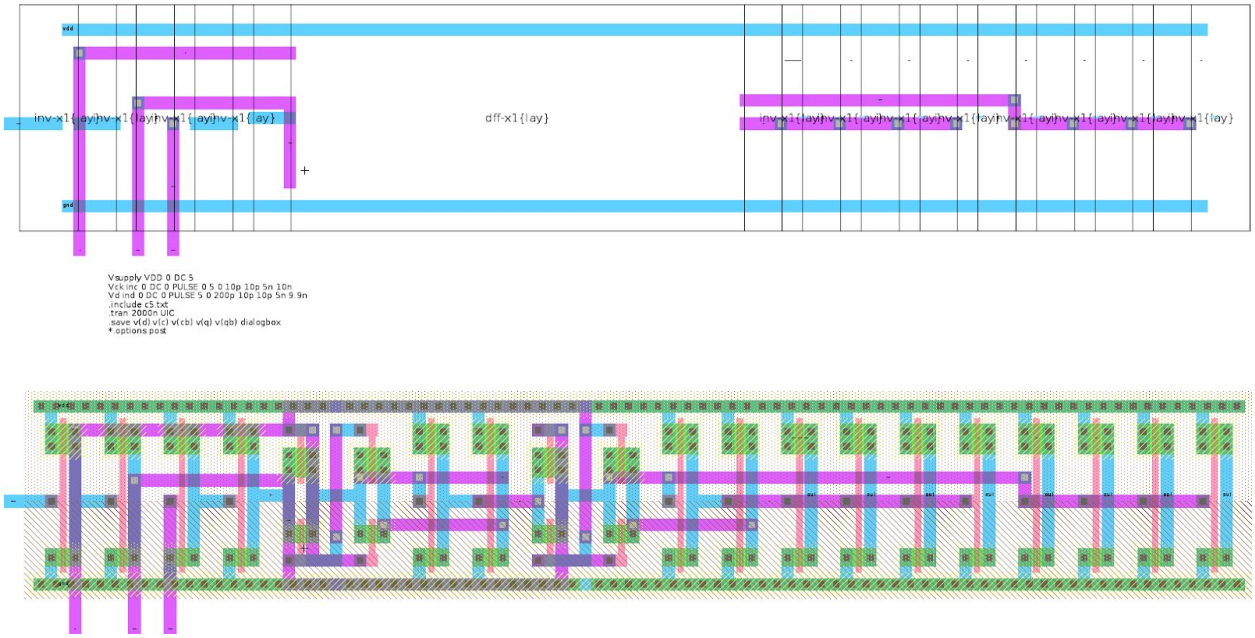




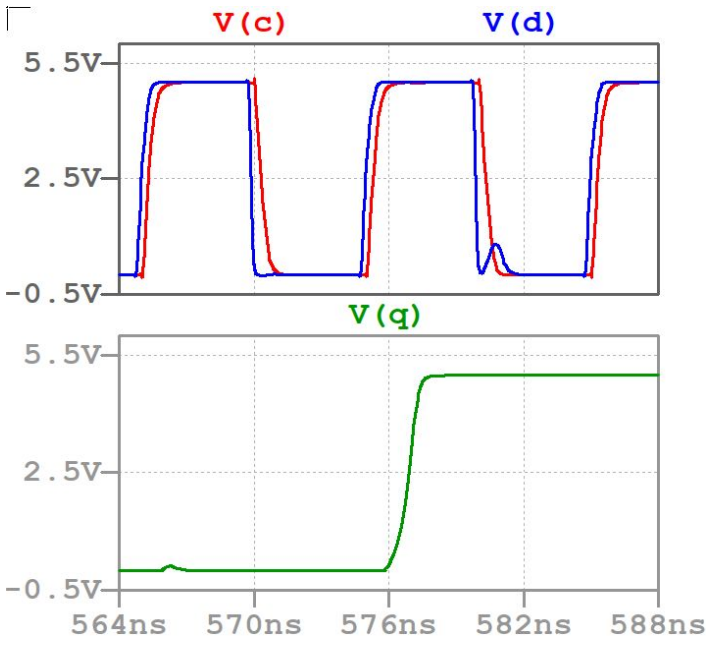
6. Testing

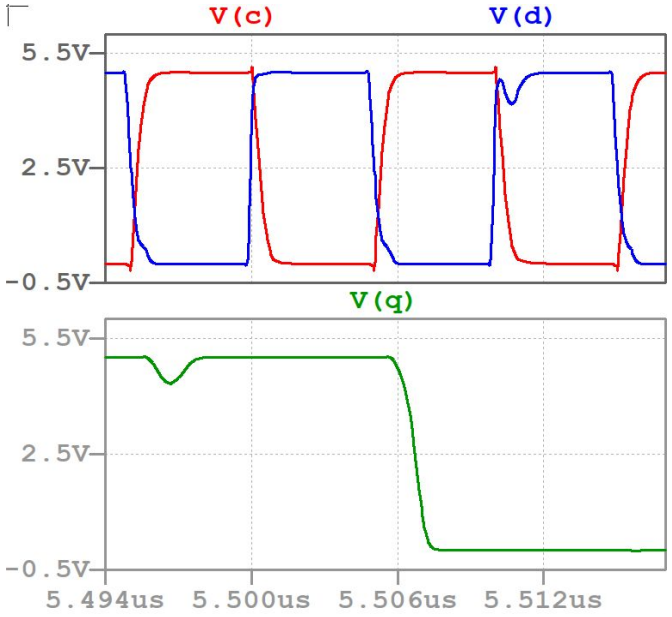
To test the layout version, it is *most appropriate* to also layout the inverters that buffer the input signals and the fanout of 4 loading inverters at each of the DFF's outputs. It is also possible to merely describe the testing inverters in the netlist and not draw and connect up their corresponding layouts. I personally don't trust that the numbers from the resulting simulations would be as realistic as this way. Since the ultimate goal of this is to make a functional chip design that gets fabricated and shipped back as a physical prototype, I want my "signoff" simulations to be as close to reality as I can manage.

The stubs on metal-2 on the lower-left are not necessary, but give a convenient arc to give names to which are attached to the SPICE voltage sources.



6.1. Setup time

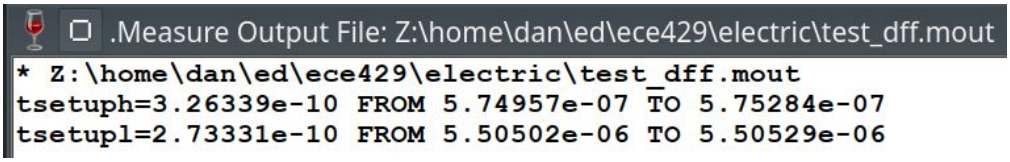




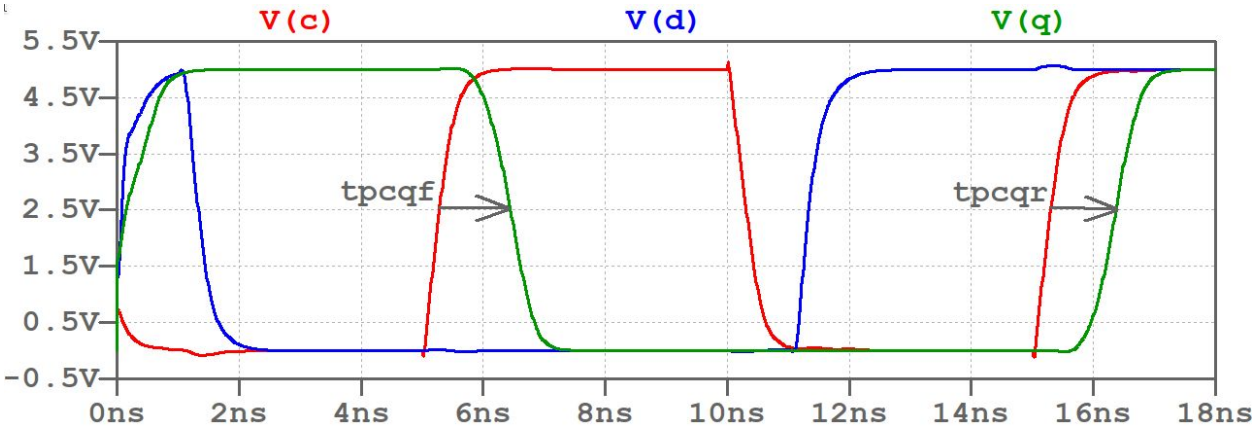
The measurements below were tested (and re-tested) **without** needing to re-run the loooong simulation. How, you ask? You *always* ask good questions, I'm impressed, good job!

- Create a plain text file and name it the same as your `.spi` file, but with a `.meas` extension.
- Write your .meas statements in there.
- File → Execute .MEAS Script
- This will use the **existing** simulation data (!) and pop up a window with the results, see the contents of my file and a screenshot of the results.

```
* \home\dan\ed\ece429\electric\test_dff.meas
.meas tran tsetuph trig v(d)=2.5 rise=59 targ v(c)=2.5 rise=58
.meas tran tsetupl trig v(d)=2.5 fall=552 targ v(c)=2.5 rise=551
```



6.2. Clock → Q delays




```
.meas results:

      tpcqr=1.06966e-09 FROM 1.52903e-08 TO 1.636e-08
      tpcqf=1.16548e-09 FROM 5.27493e-09 TO 6.44042e-09
```

6.3. Summary of measurements

The additional information that Electric adds to the `.spi` netlist include:

- source/drain *pn*-junction area for each transistor
- wiring resistance for each arc of {poly, metal-1, metal-2}
- total capacitance from each arc to each other arc and to the substrate
    - only includes **vertical** overlaps, not "fringing capacitance" from items horizontally adjacent to each other

We know that the transistor widths and lengths match exactly and that the schematic and layout have the same topology (devices and connections) thanks to the NCC or LVS (layout versus schematic) tool.

times (ps)	schematic	post-layout
tsetup 1	92	326
tsetup 0	85	273
tpcqr	487	1070
tpcqf	467	1165

Setup times are 3✕ and clock-Q delays are 2.5✕ longer than the schematic simulation, which, remember, did **not** include any other capacitances other than the transistor's gate-related capacitances.

Even these post-layout numbers are not reality. The wiring R-C circuits are lumped models of the true distributed-R-C reality. Computing a more accurate wire delay should use the Elmore approximation for RCRC... chains. Aaaaand this doesn't take into account the wiring's **inductance** and mutual inductance (magnetic interaction) between nearby wires. You want a design to go fast?[1] Then you should read Weste and Harris' book chapter 6, Interconnect, which is all about wiring.

(making decent looking tables in GDocs is a pain 😒)

[1] Nobody gets paid to do slow digital. --Prof. White