analyzer

Add \_CLK2, \_DATA2, \_Q, and \_Qb to the analyzer window:

analyzer \_CLK2 \_DATA2 \_Q \_Qb

Next set \_DATA2 to high and \_CLK2 to low:

h \_DATA2
l \_CLK2

and advance time in the simulator by 10 nsec:

s 10n

You can now toggle \_DATA2 and advance time

l \_DATA2
s 10n

to see that the latch is transparent when the clock is low.

As an alternative to simulating in this way, you can define the clock as:

clock \_CLK2 0 1

and then use the cycle command to advance time by units of cycles:

c 1

Note that we are using IRSIM in a mode in which we are not trying to accurately model delay, just verify functionality. The reason for this is that timing analysis is best done with a tool known as a static timing analyzer. Commercial tools for static timing analysis include Pearl and Pathmill at the transistor level and Primetime at the gate level.
We will use switch-level simulation to verify the functionality of our transistor-level circuits; that is, verify that the circuit performs the logic function we intended. At the same time, we will want switch-level simulation to identify floating nodes and collision conditions that we did not intend.

In IRSIM, nodes are modelled as capacitors and transistors are modelled as a switch in series with a resistor, where the value of the resistor depends on the type of transistor and the quantized signal values (e.g., nMOS pulling low is stronger than nMOS pulling high). Nodes can have the (quantized) values '1', '0', or 'X'.

IRSIM requires two input files, a transistor parameter file (the .prm file) and the netlist file (the .sim file). The .sim netlist must be flat and is generated automatically from a flat HSPICE netlist by the spice2sim utility.

In the library tutorial3 is the circuit latch which we will use as our tutorial example for demonstrating how IRSIM works. First netlist the design by choosing Tools → Analog Artist. From the Analog Artist window, choose HSPICE as the simulator. Then from the Setup → Environment window, choose the netlist type of “flat.” Then netlist the design by choosing Simulation → Netlist → Create Final. We now need to convert the netlist (which is in the file ~simulation/latch/hspiceS/schematic/netlist/hspiceFinal) to a .sim file by executing the command:

```
spice2sim hspiceFinal amap/top_level_map.i.net > latch.sim
```

which creates the .sim file latch.sim. You will have to execute this from the ~simulation/latch/hspiceS/schematic/netlist directory.

Now, you can invoke IRSIM with the command:

```
irsim -s /usr/tech/tsmc025/irsim/tsmc025.prm latch.sim
```

Next set supply and ground:

```
h vdd!
l gnd!
```

Now bring up the analyzer by typing: