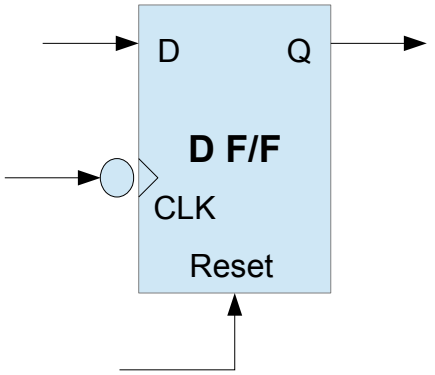
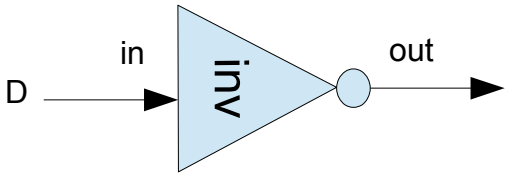
CS147 - Lab 02

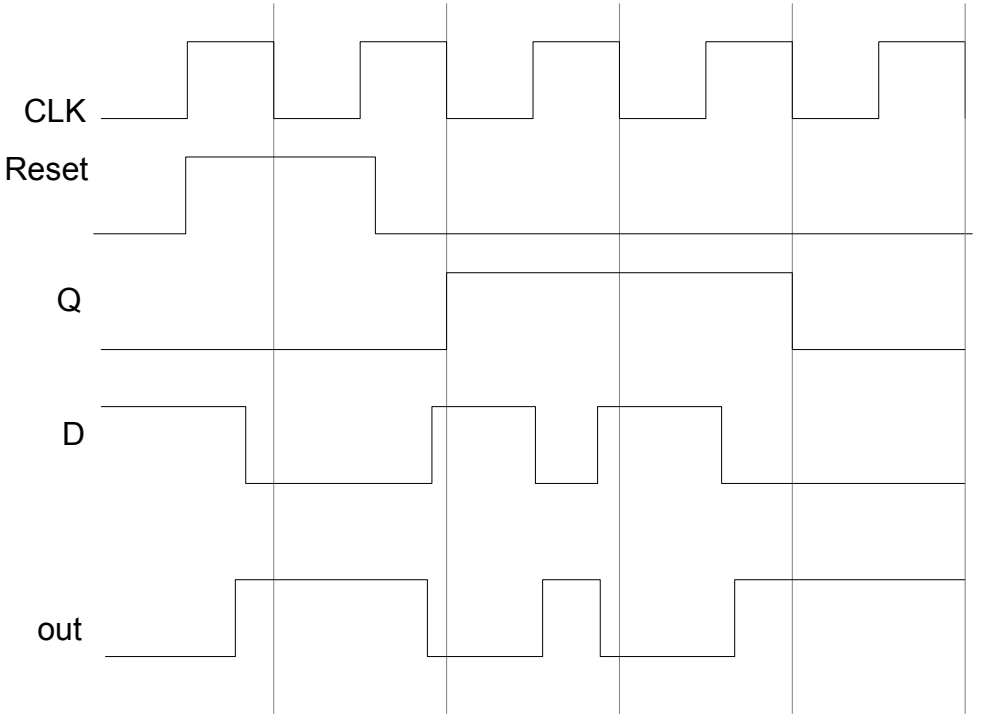
Kaushik Patra
(kaushik.patra@sjsu.edu)

Hierarchical Modeling

- Let's define a hardware entity (gate) D-flip-flop which stores the 1-bit input value on negative edge of the clock signal. It has a reset signal which reset the stored value to 0 on positive edge of the reset signal.
- Let's also defines another hardware entity inverter which outputs 0 if input is 1 and vice-verse. There is no clock involve.

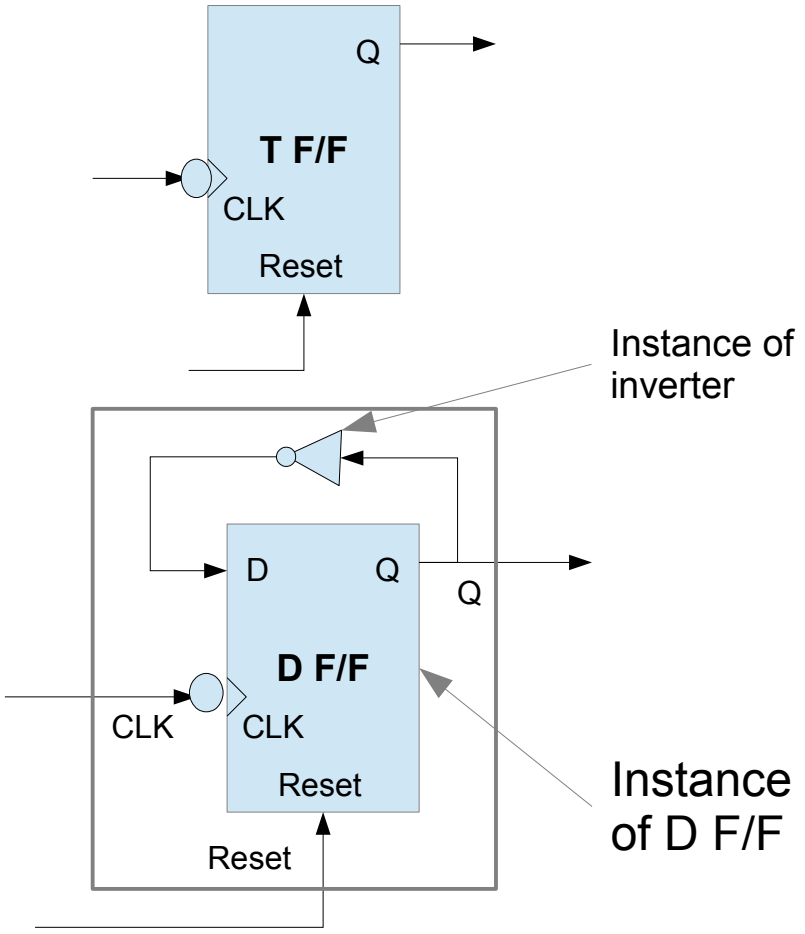


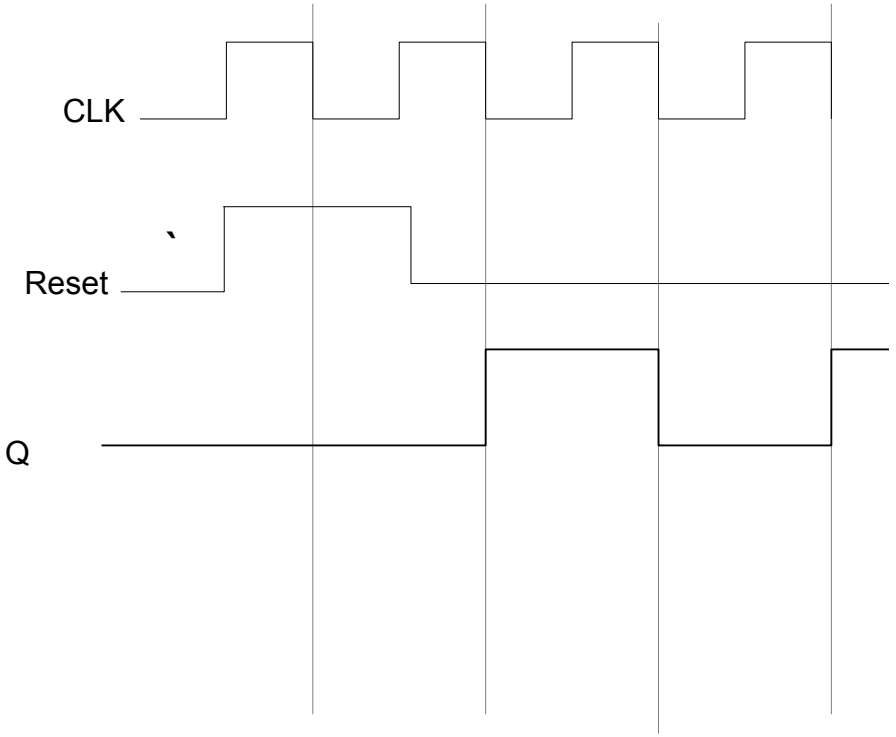




Hierarchical Modeling

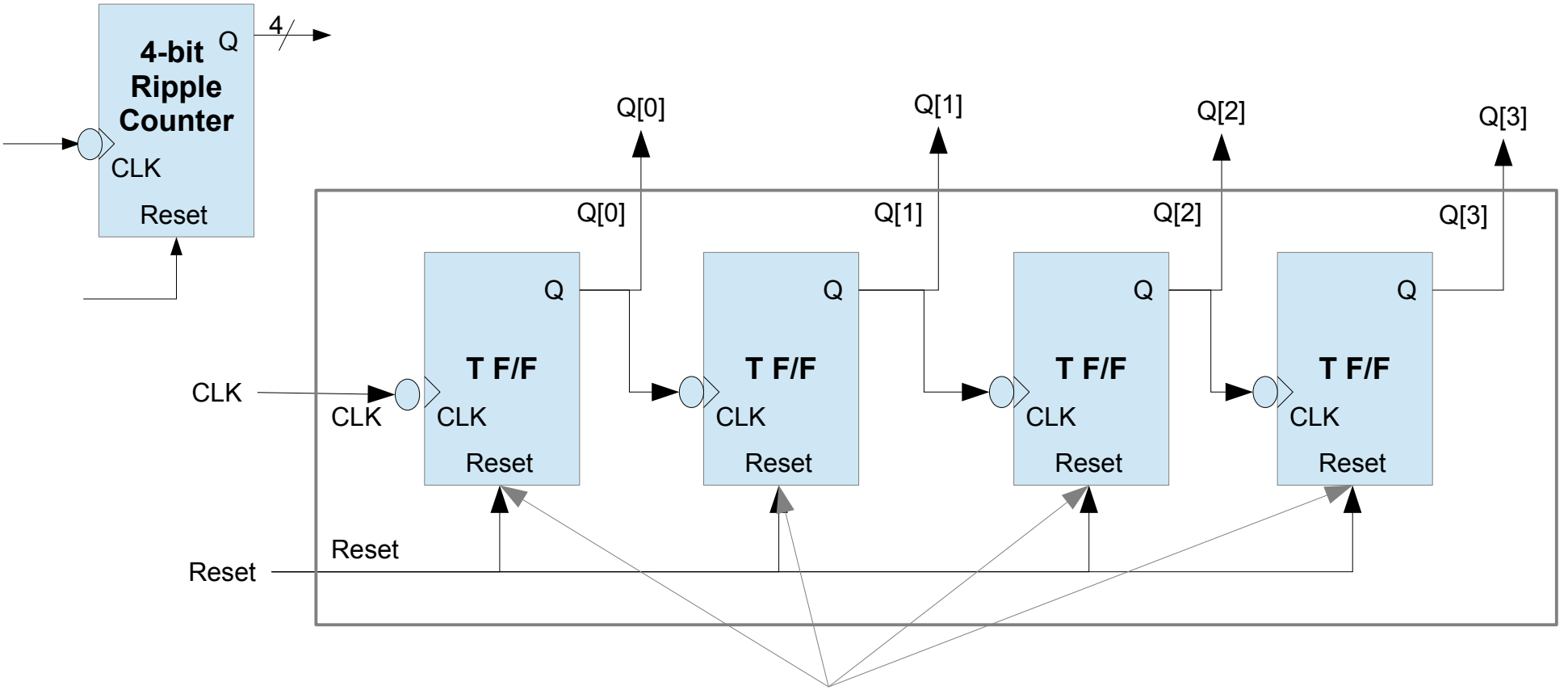
- Let's define a hardware entity (gate) T-flip-flop whose output always toggles on negative edge of the clock. It has a reset signal which reset the stored value to 0, again at the positive edge of the reset signal. One way to implement T-flip-flop is to use a D-flip-flop and an inverter.



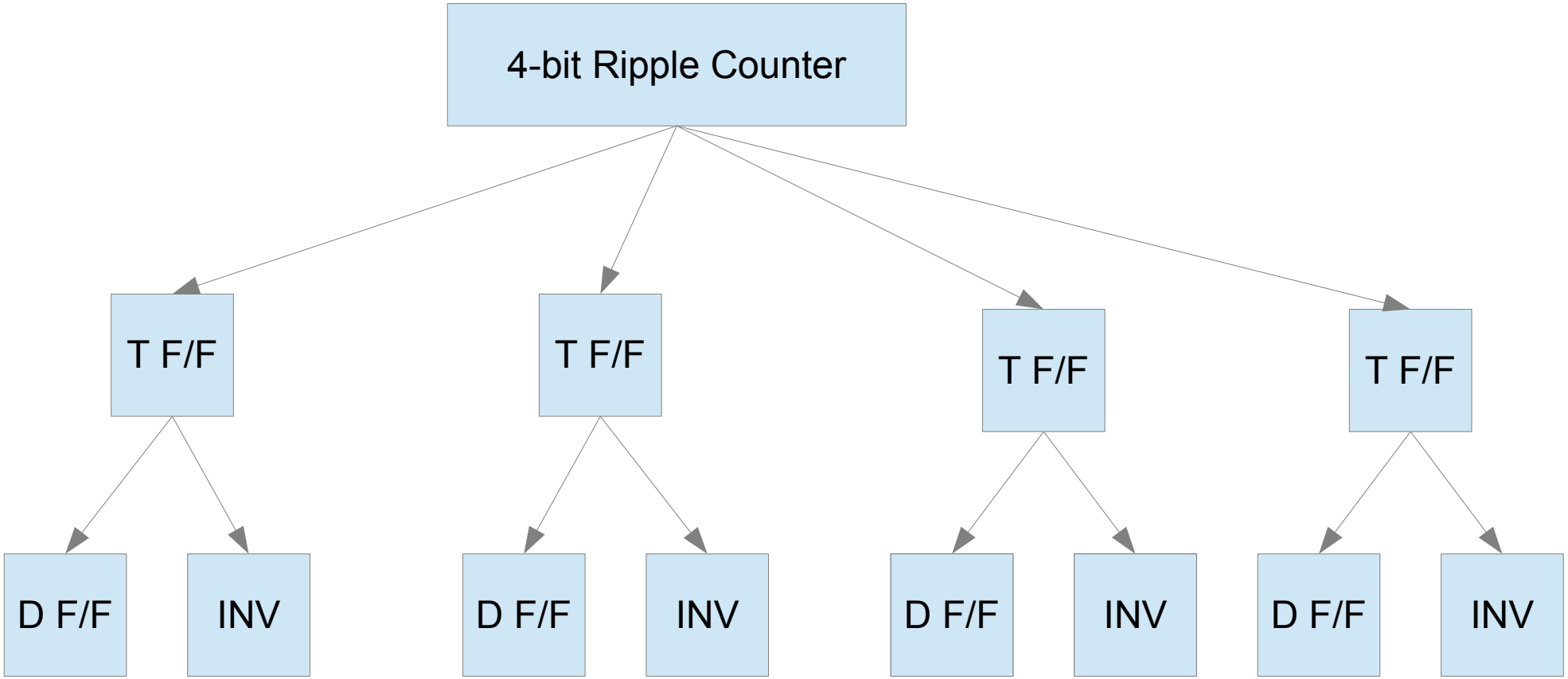


Hierarchical Modeling

- Let's now implement a 4-bit ripple carry counter using T-flip-flop. This hardware entity keeps on counting from 0-15 and back to 0 on each negative edge of the input clock.



Hierarchical Modeling



Hierarchical Modeling

- We define 'module' for each hardware entity, including the port list. Then we instantiate module into higher blocks for re-using functionality.

```
module D_FF (Q, D, CLK, RST);
input D, CLK, RST;
output Q;

… Implementation of the D-F/F …

endmodule
```

```
module INVERTER (OUT, IN);
input  IN;
output OUT;


… Implementation of the inverter …

endmodule
```

```
module T_FF (Q, CLK, RST);
input CLK, RST;
output Q;
wire d_in;

D_FF dff_inst(.Q(Q), .D(d_in) ,
              .CLK(CLK), .RST(RST));
INVERTER inv_inst(.OUT(d_in), .IN(Q));

endmodule
```

4

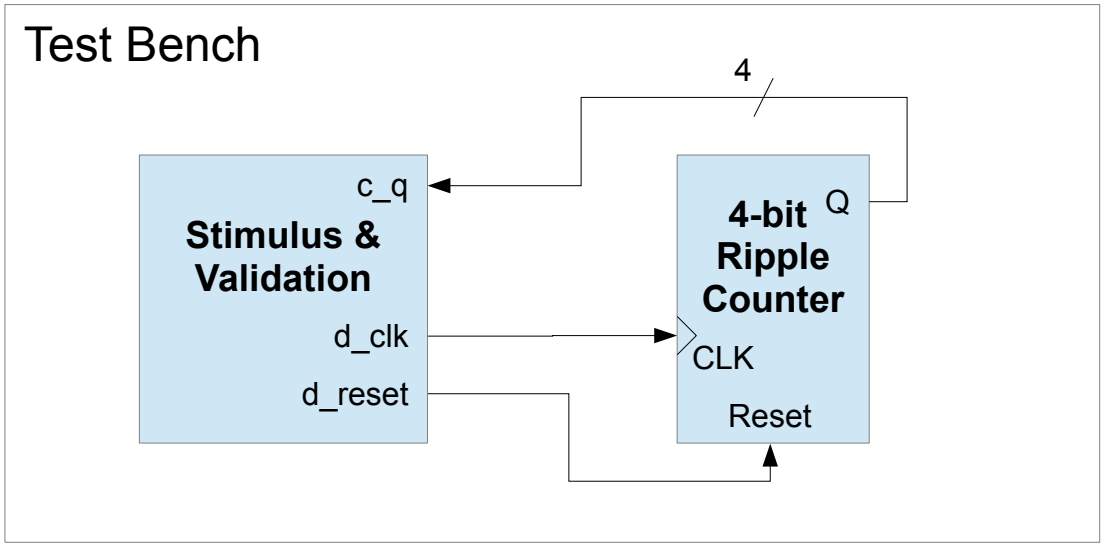
```
module Ripple_Counter (Q, CLK, RST);
input CLK, RST;
output [3:0] Q;

T_FF    tff0 (.Q(q[0], .CLK(CLK), .RST(RST));
T_FF    tff1 (.Q(q[1], .CLK(q[0], .RST(RST));
T_FF    tff2 (.Q(q[2], .CLK(q[1], .RST(RST));
T_FF    tff3 (.Q(q[3], .CLK(q[2], .RST(RST));

endmodule
```

Hierarchical Modeling

- Now that we have 4-bit ripple counter implemented, how to test it?
- The counter must be instantiated within a top level block.
- Some block must be driving an instance of this ripple counter by generating clock and reset signal. This is the stimulus block.
- There should be some mechanism for reading back the output from the counter and test it. This is the validation code.
- The top level module is called a test bench. Test bench does not have any input or output.



CS147 - Lab 02

Kaushik Patra
(kaushik.patra@sjsu.edu)