



Sharif University of Technology  
Department of Computer Engineering

# Digital System Design

## Hierarchical modelling concepts

Siavash Bayat-Sarmadi

# Outline

2

- Design methodologies
  - ▣ Top-down
  - ▣ Bottom-up
- Modules
- Module instances
- Components of a simulation

# Design Methodology

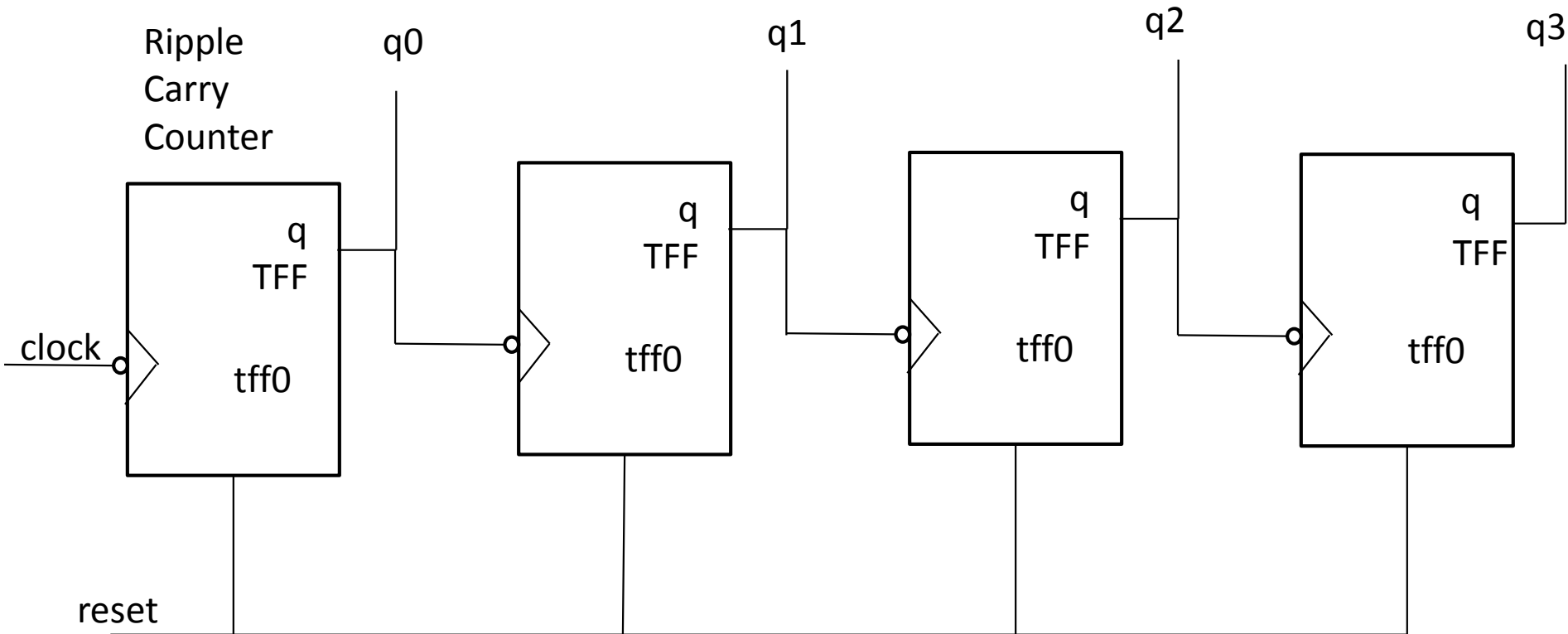
3

- Top-down
  - ▣ Define the top level
  - ▣ Identify the sub-blocks till reaching leaf cells
- Bottom-up
  - ▣ Identify the available building blocks
  - ▣ Build bigger till reaching the top-level

# Design Methodology (Cont'd)

4

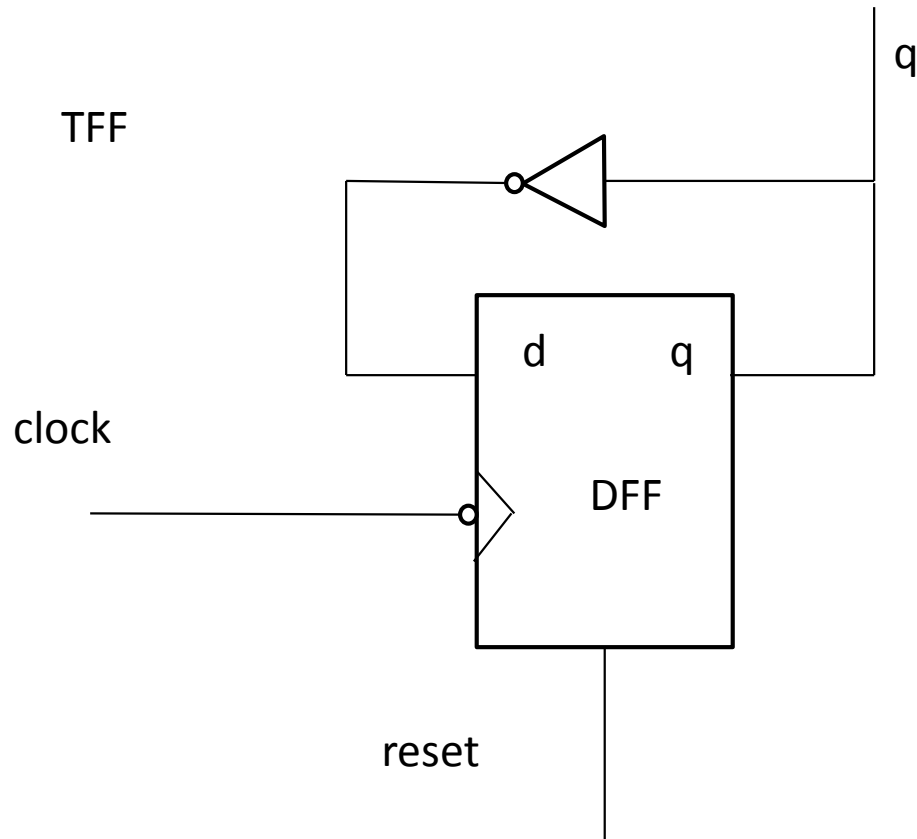
## □ 4-bit ripple carry counter



# Design Methodology (Cont'd)

5

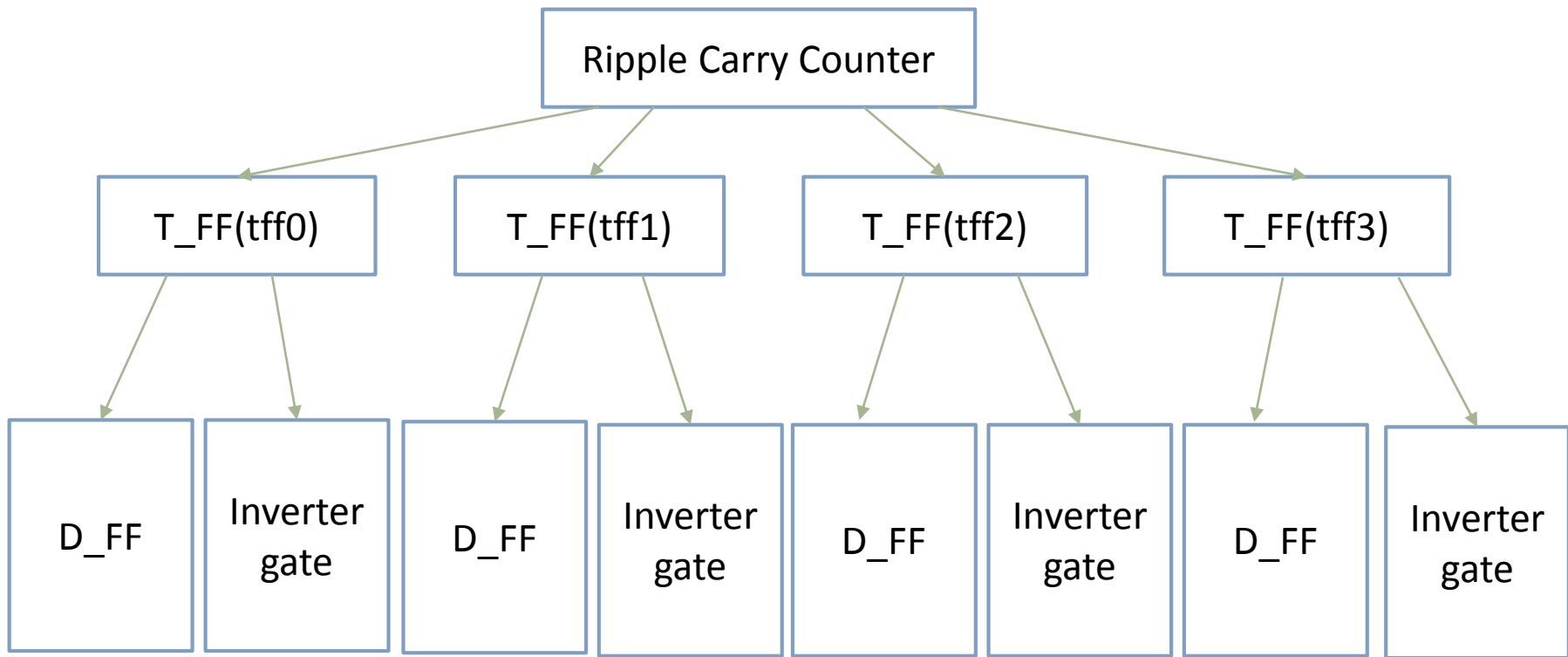
## □ Toggle flip-flop



# Design Methodology (Cont'd)

6

## □ Design hierarchy



# Module

7

- Basic building block in Verilog
- Interacts through its port interface
- Internal implementation hidden to the rest of the design
- Declaration starts with the keyword “module”
- Ends with “endmodule”
- Module definition cannot be nested

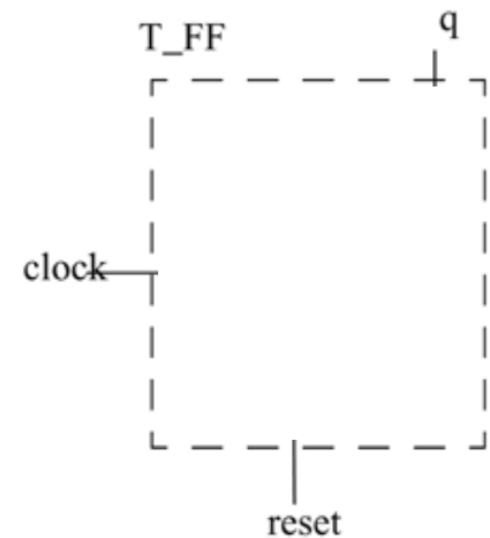
# Module (Cont'd)

8

Name

Port (terminal) list

```
module T_FF (q, clock, reset);  
    ..  
    <functionality of T flip-flop>  
    ..  
endmodule
```





# Instance

9

- Modules must be instantiated to be used in the design
- Comparing with C++/Java
  - ▣ Module ~ Class
  - ▣ Instances ~ Objects
- Each instance has its own
  - ▣ Name, variables, parameters, I/O interface, etc.

# Instances (Cont'd)

10

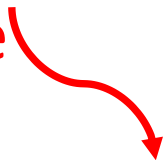
//top-level

module ripple\_carry\_counter(q, clk, reset);

output [3:0] q;

input clk, reset;

Module  
name



Instance name



T\_FF tff0 (q[0], clk, reset);

T\_FF tff1 (q[1], q[0], reset);

T\_FF tff2 (q[2], q[1], reset);

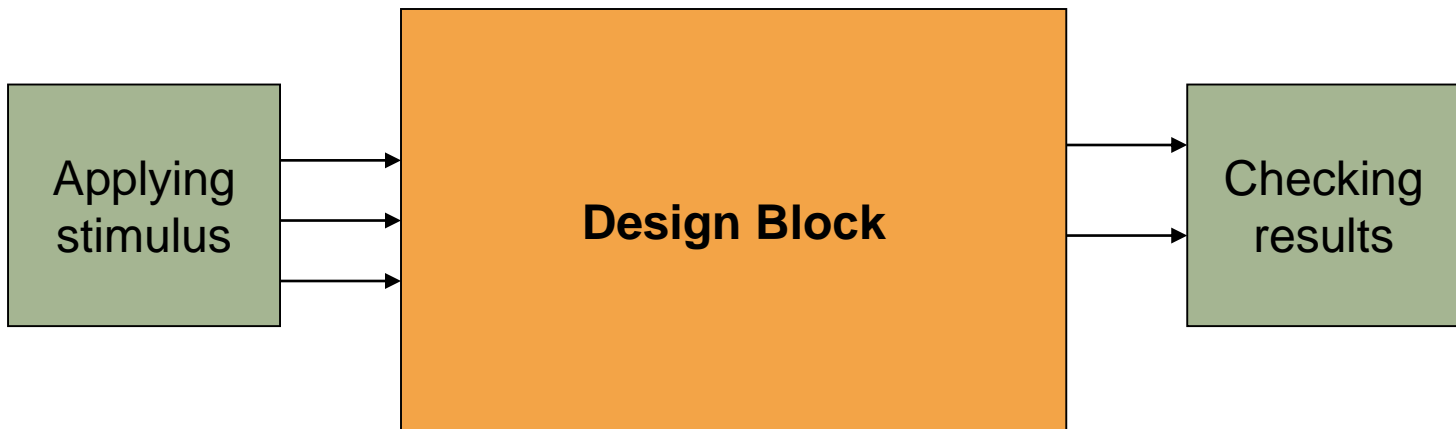
T\_FF tff3 (q[3], q[2], reset);

endmodule

# Components of a Simulation

11

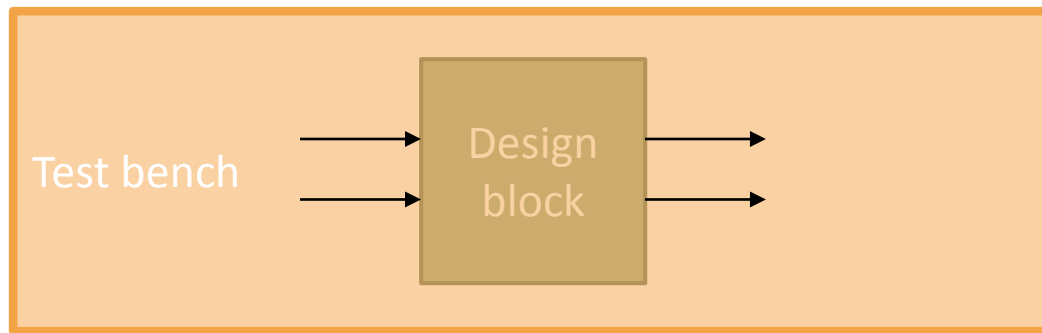
- Test bench
  - ▣ Testing functionality of the design
    - By applying stimulus
    - And checking the results



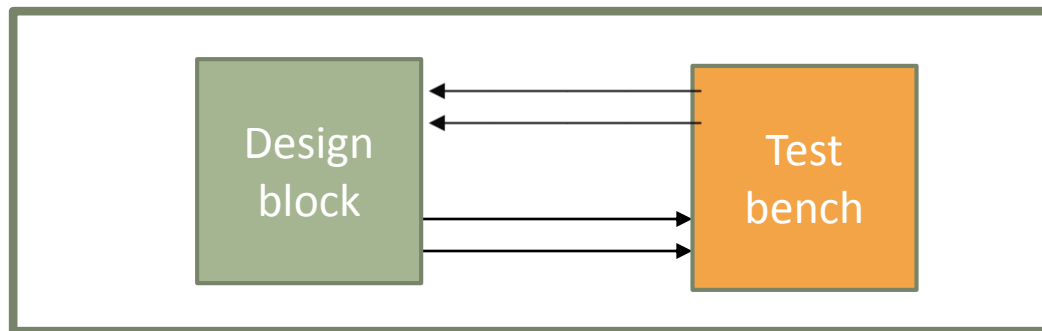
# Components of a Simulation (Cont'd)

12

- Test bench instantiating the design block



- A dummy top-level module instantiating both



# Components of a Simulation (Cont'd)

13

```
module stimulus;  
reg clk;  
reg reset  
wire [3:0] q;  
  
ripple_carry_counter  
r1 (q, clk , reset);  
  
initial  
    clk = 1'b0;
```

```
always  
    #5 clk = ~clk;  
initial  
begin  
    reset = 1;  
    #15 reset = 0;  
    #180 reset = 1;  
    #10 reset = 0;  
end
```

# Components of a Simulation (Cont'd)

14

