Sharif University of Technology
Department of Computer Engineering

Digital System Design
Hierarchical modelling concepts

Siavash Bayat-Sarmadi

Outline

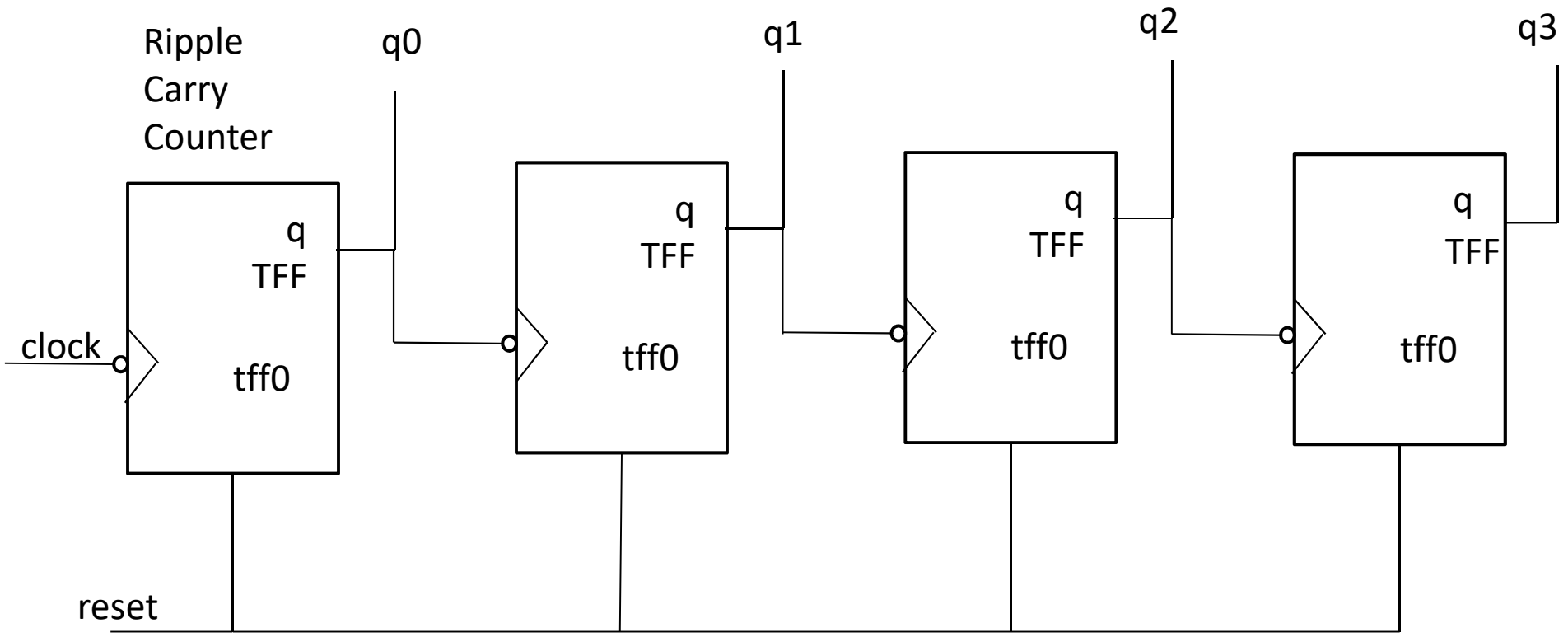
- Design methodologies
  - Top-down
  - Bottom-up
- Modules
- Module instances
- Components of a simulation

Design Methodology

- 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

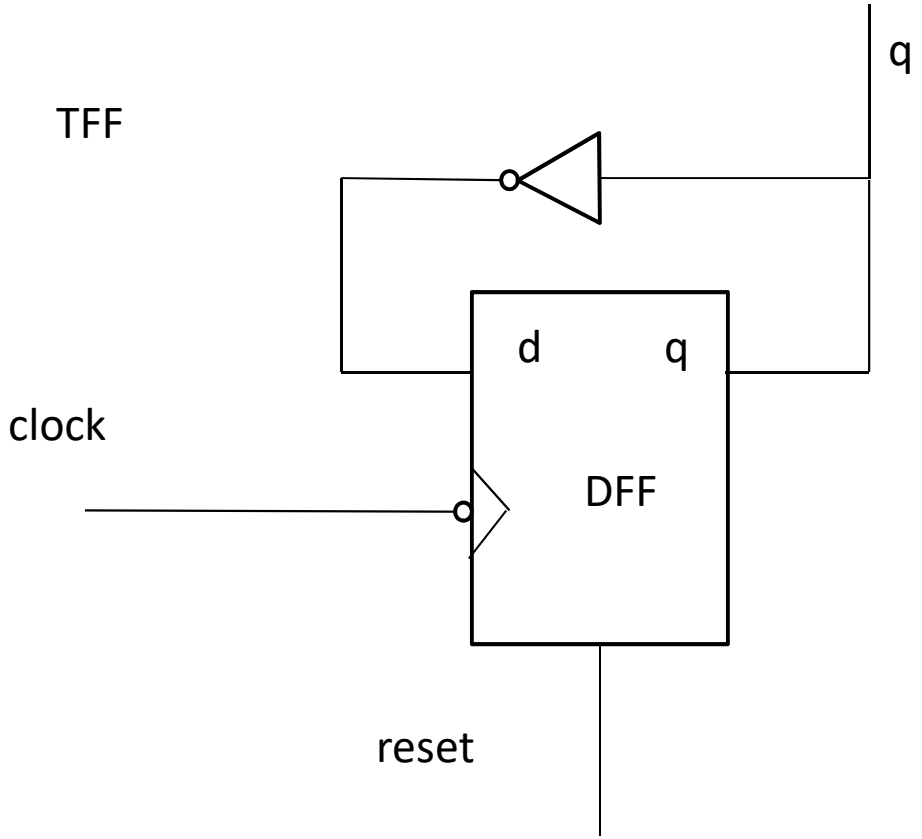
Design Methodology (Cont'd)

- 4-bit ripple carry counter



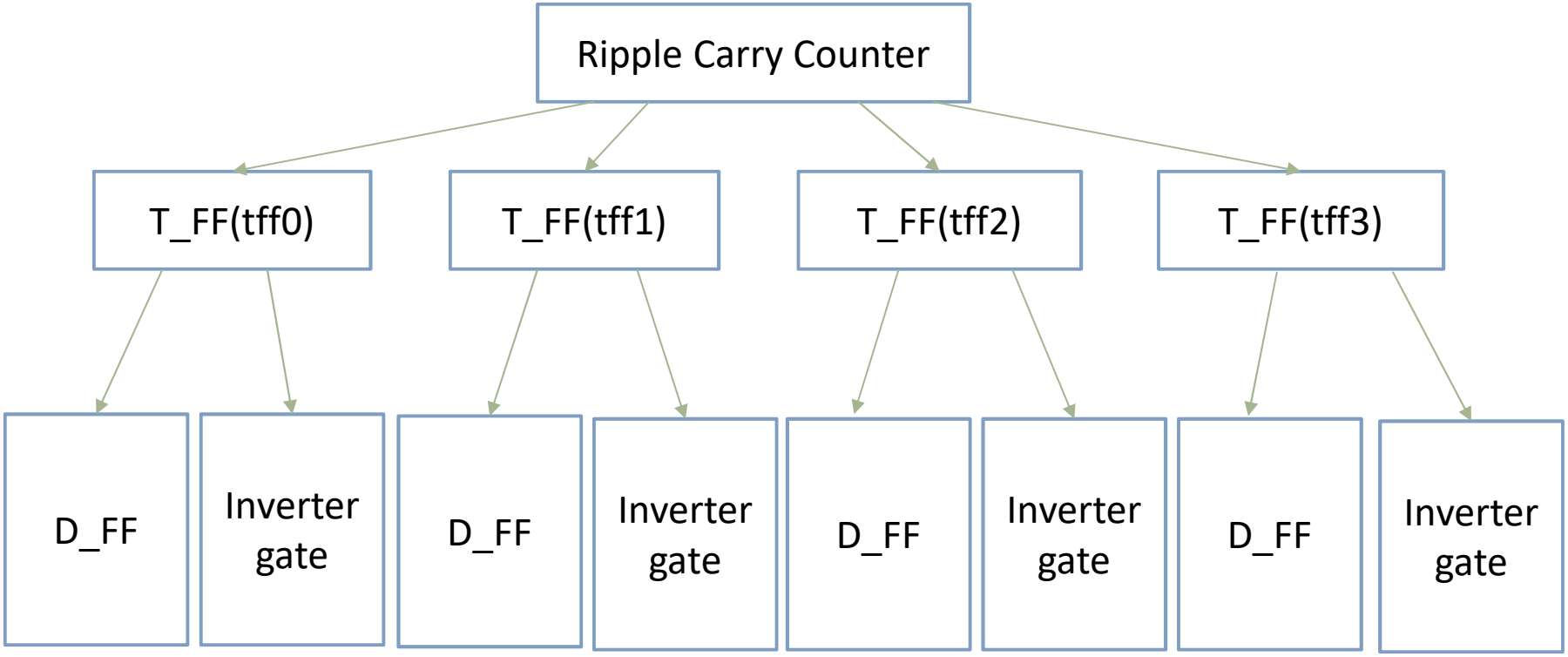
Design Methodology (Cont'd)

- Toggle flip-flop



Design Methodology (Cont'd)

- Design hierarchy



Module

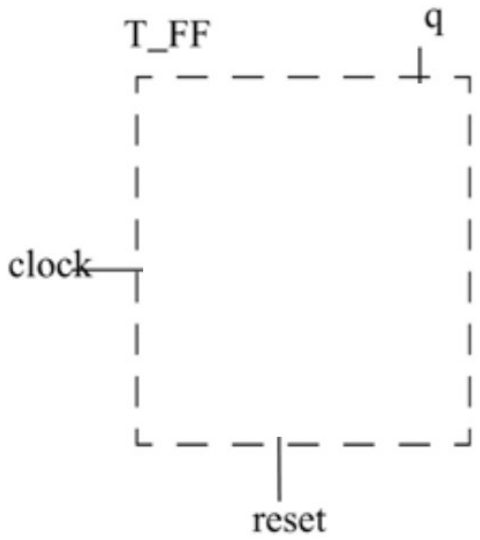
- Basic building block in Verilog
- Interacts through its port interface
- Internal implementation hidden to the rest of the design
- Declaration starts with the keyword “module”
- Ends with “endmodule”
- Module definition cannot be nested

Module (Cont'd)

Name

Port (terminal) list

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



Instance

- 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)

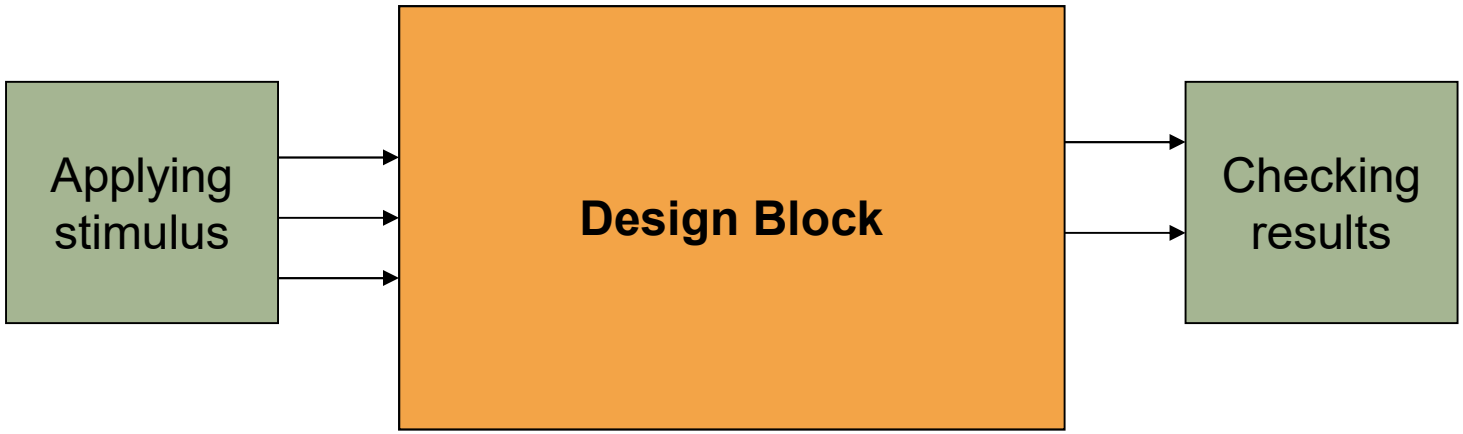

```
//top-level
module ripple_carry_counter(q, clk, reset);
output [3:0] q;
input clk, reset;

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
```

Module name

Instance name

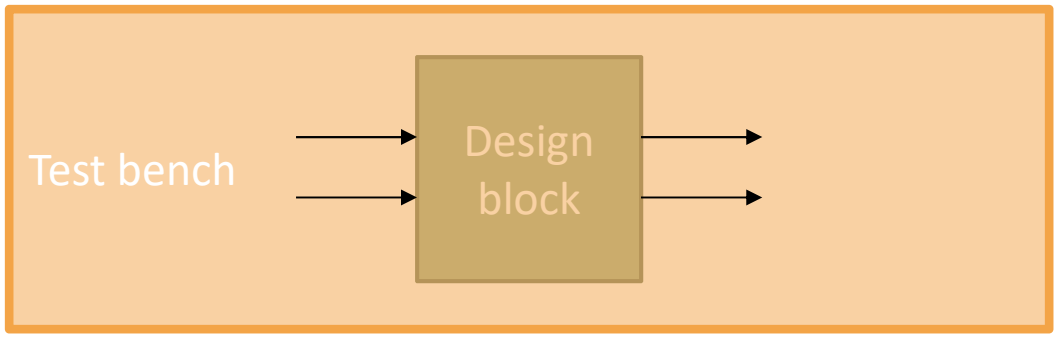
Components of a Simulation

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

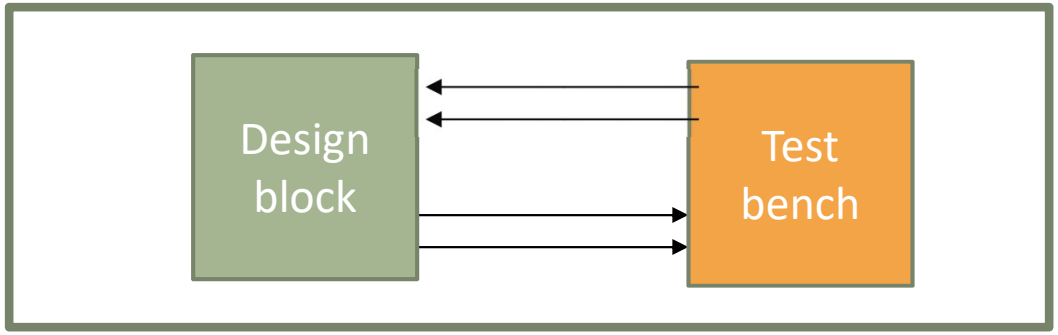


Components of a Simulation (Cont'd)

- Test bench instantiating the design block



- A dummy top-level module instantiating both



Components of a Simulation (Cont'd)


```
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)