Verilog-HDL

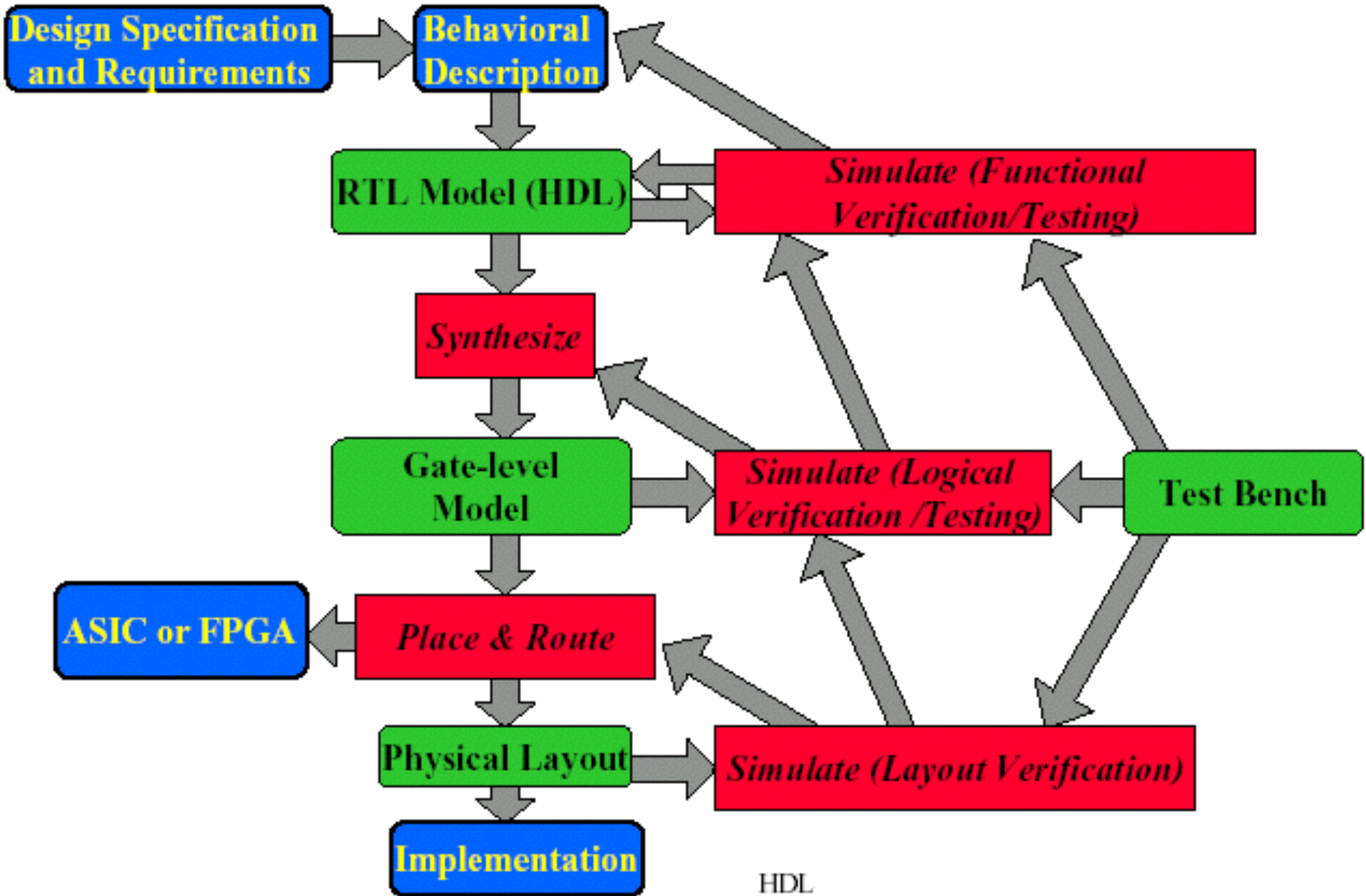
Reference: Verilog HDL: a guide to digital design and synthesis, Palnitkar, Samir

Some of slides in this lecture are supported by Prof. An-Yeu Wu, E.E., NTU.

OUTLINE

- Introduction
- Basics of the Verilog Language
- Gate-level modeling
- Data-flow modeling
- Behavioral modeling
- Task and function

Design Methodology



HDL

What is Verilog HDL ?

- Mixed level modeling
  - Behavioral
    - Algorithmic (like high level language)
    - Register transfer (Synthesizable)
  - Structural
    - Gate (AND, OR ……)
    - Switch (PMOS, NOMS, JFET ……)
- Single language for design and simulation
- Built-in primitives and logic functions
- User-defined primitives
- Built-in data types
- High-level programming constructs

Basic Conventions

- Verilog is case sensitive
  - Keywords are in lowercase
- Extra white space is ignored
  - But whitespace does separate tokens
- Comments
  - One liners are //
  - Multiple lines /* */
  - Comments may not be nested

Basic unit --*Module*

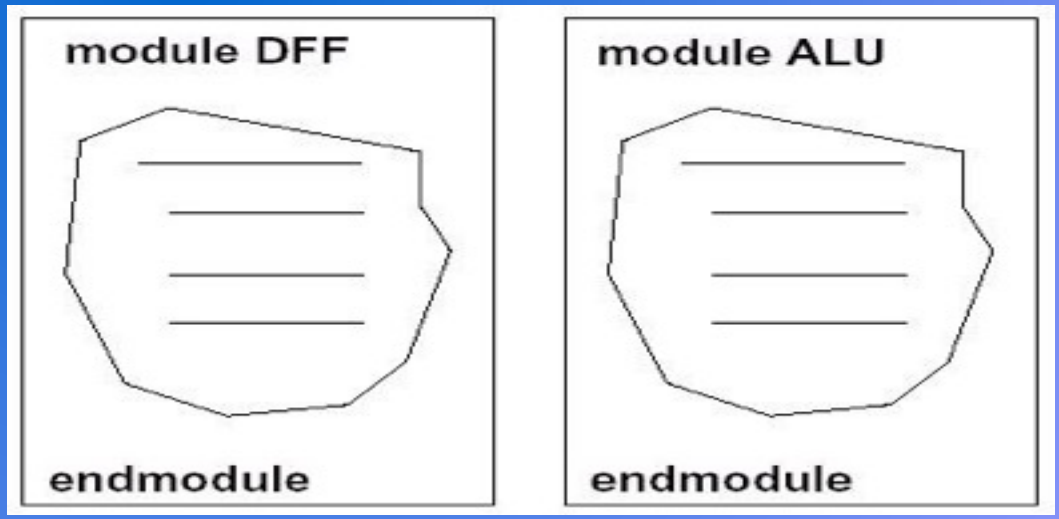
module module_name (port_name);

port declaration

data type declaration

module functionality or structure

Endmodule



D-FlipFlop

```
module D_FF(q,d,clk,reset);
output q;                 //port declaration
input d,clk,reset;  // data type declaration
reg q;
always @ (posedge reset or negedge clk)
if (reset)
   q=1'b0;
else
    q=d;
endmodule
```

Instance

- A module provides a template which you can create actual objects.
- When a module is invoked, Verilog creates a unique object from the template
- The process of creating a object from module template is called instantiation
- The object is called instance

Instances

```
module adder (in1,in2,cin,sum,cout);
.......
endmodule
```

```
module adder8(.....) ;
adder  add1(a,b,1'b0,s1,c1) ,
       add2(.in1(a2),.in2(b2),.cin(c1),.sum(s2)
            ,.cout(c2)) ;
.....
endmodule
```

Mapping port positions

Mapping names

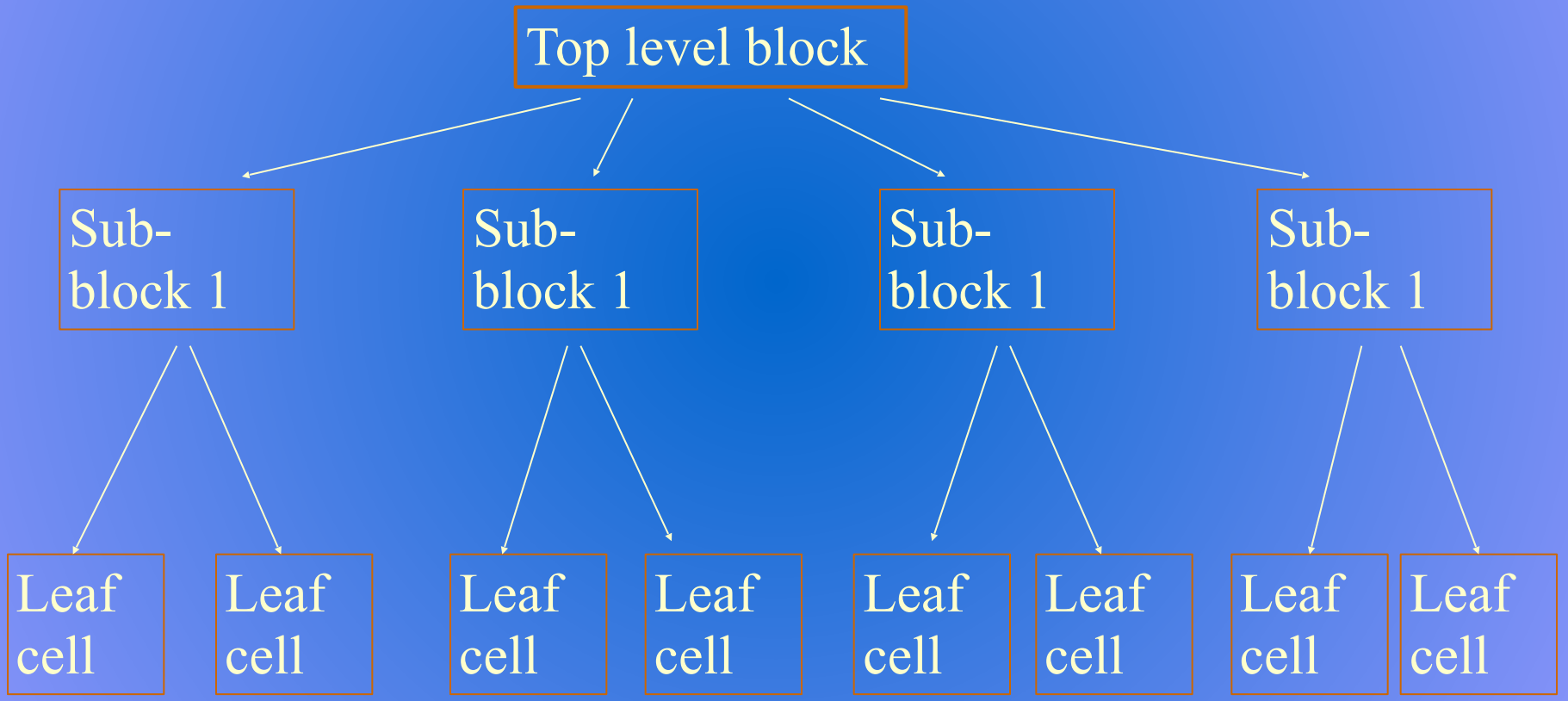
T-FlipFlop

```
module T_FF(q,clk,reset);
output q;
input clk,reset;
wire d;
D_FF dff0(q,d,clk,reset); // create an instance
not n1(d,q);
endmodule
```

Identifier & Keywords

- Identifier
  - User-provided names for Verilog objects in the descriptions
  - Legal characters are “a-z”, “A-Z”, “0-9”, “_”, and “$”
  - First character has to be a letter or an “_”
    - Example: Count, _R2D2, FIVE$
- Keywords
  - Predefined identifiers to define the language constructs
  - All keywords are defined in lower case
  - Cannot be used as identifiers
    - Example:initial, assign, module, always….

Hierarchical Modeling Concepts

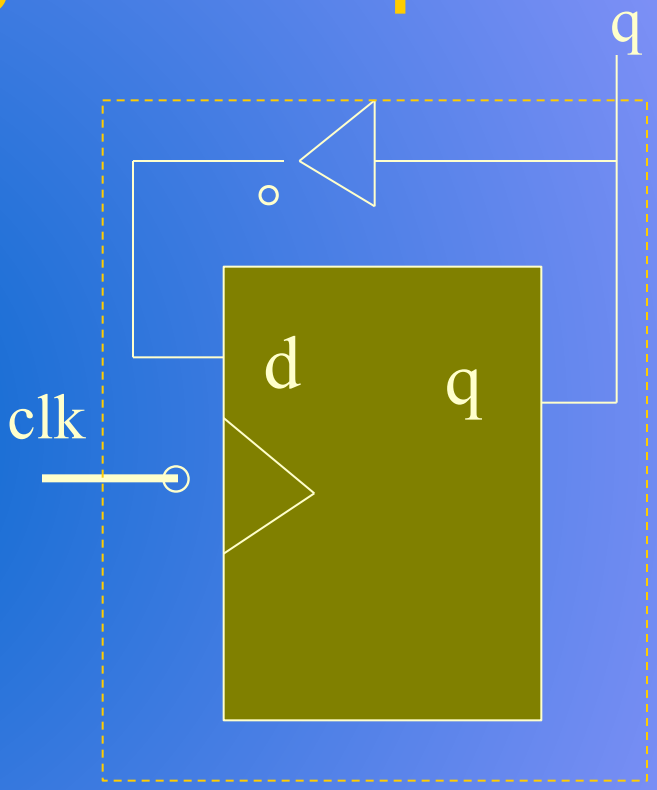


Hierarchical Modeling Concepts

- 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 tff0(q[2], q[1], reset);
- T_FF tff0(q[3], q[2], reset);
- endmpdule

Hierarchical Modeling Concepts

- module T_FF(q, clk, reset);
- output q;
- input clk, reset;
- wire d;
- D_FF dff0(q, d, clk, reset);
- not na(d, q);
- endmodule



Hierarchical Modeling Concepts

- module D_FF(q, d, clk, reset);
- output q;
- input d, clk, reset;
- reg q;
- always @(posedge reset or negedge clk)
- if (reset)
- q=1’b0;
- else
- q=d;
- endmodule

4-bits Ripple Carry Counter