



Sharif University of Technology  
Department of Computer Engineering

# Digital System Design

## Modules and ports

Siavash Bayat-Sarmadi

# Outline

2

- Components of a Verilog module, i.e. name, port list, etc.
- Port connection rules
- Connecting port by ordered list and name
- Hierarchical name

# Components of a module

3

- Previously discussed how to
  - ▣ Define
  - ▣ Instantiate
- Currently
  - ▣ Module internals

# Components of a module

4

Module name,

Port List, port declaration (if ports present)

Parameters (optional)

Declaration of **wires** , **regs** and other variable

data flow statement (**assign**)

Instantiate of lower level modules



**Always** and **initial** blocks: all behavioral statements go in these blocks.

**Task** and **function**


**Endmodule statement**

# Components of a module

5

**Name**  **module**  **Port (terminal) list** adder (sum, cout, a, b);

**output** [1:0] sum;  
**output** cout;  
**input** [1:0] a, b;

 **Port declaration**

**parameter** delay=1;

**Parameters**

...

**endmodule**

# Components of a module (cont'd)

6

```
module adder (sum, cout, a, b);
```

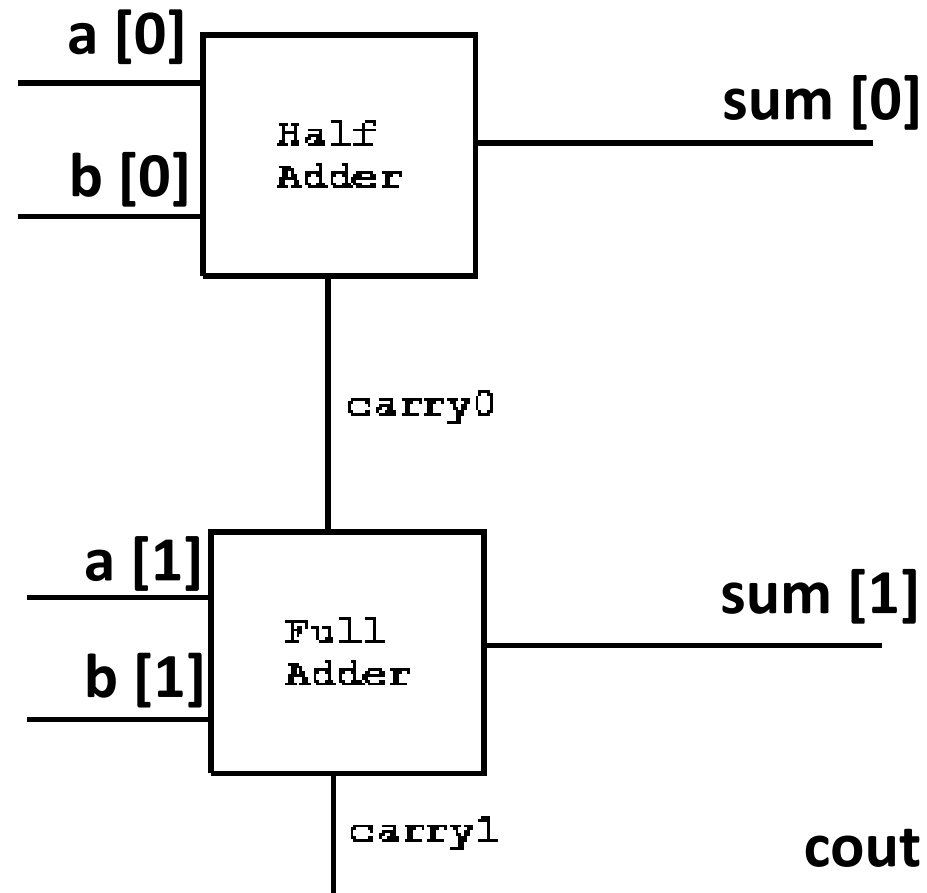
...

Declaration of wires  
and regs and other variables

```
wire carry0, carry1;
```

...

```
endmodule
```



# Components of a module (cont'd)

7

- Default type for port
  - ▣ wire
  - ▣ If not specified
- Port list and declaration can be mixed
  - ▣ **output wire** [1:0] sum;
  - ▣ **module** adder (**output** [1:0] sum, ...
  - ▣ **module** adder (**output wire** [1:0] sum, ...

# Components of a module (cont'd)

8

```
module adder (sum, cout, a, b);
```

```
...
```

```
...
```

Instantiation of lower  
modules

```
halfAdder ha (sum[0], carry0, a[0], b[0]);
```

```
fullAdder fa (sum[1], cout, a[1], b[1], carry0);
```

```
endmodule
```



# Components of a module (cont'd)

9

- Dataflow statements
  - ▣ **assign**
- Behavioral blocks
  - ▣ **always**
  - ▣ **initial**
- **tasks** and **functions**

# Ports

10

- Module inputs and outputs
- Optional for modules
  - ▣ Top-level module doesn't have ports
- Port types
  - ▣ **input**
  - ▣ **output**
  - ▣ **inout**
    - Be extra-careful with this type

# Ports (cont'd)

11

- Width matching
  - ▣ Internal and external connections can have different sizes
  - ▣ A warning will be issued by synthesizer
- Unconnected ports
  - ▣ `fulladd4 fa0(SUM, , A, B, C_IN);`

# Ports (cont'd)

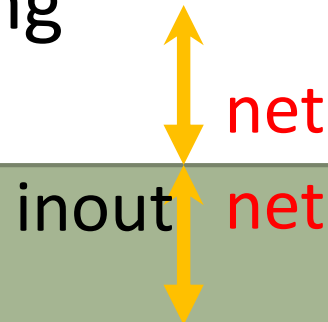
12

- Connecting to external signals (cannot be mixed):
  - ▣ By ordered list
  - ▣ by name
- Example: **module** adder (sum, cout, a, b);
  - ▣ adder adderInstance1 (s, c, a, b);
  - ▣ adder adderInstance2 ( .sum (s), .cout (c), .a (a), ...

# Port Connection Rules

13

Note when instantiating  
a module



Note when declaring  
a module

input

output



# Hierarchical Names

14

- Every identifier a unique name
- \$display ("%m")

