COL215 Digital Logic and System Design

Introduction to VHDL

11 August 2017

Why HDLs (Hardware Description Language)

- Schematic diagrams are tedious and error prone for large circuits
- Schematic diagrams provide only structural abstraction, HDLs provide structural as well as behavioural abstraction
- Behavioural abstraction increases the scope of design automation

VHDL: Basic units

- Basic unit of description is an ENTITY - ARCHITECTURE pair
- ENTITY defines the exterior
- ARCHITECTURE defines the interior
  - structural
  - behavioral
- Multiple architectures possible for same entity

VHDL: information carriers

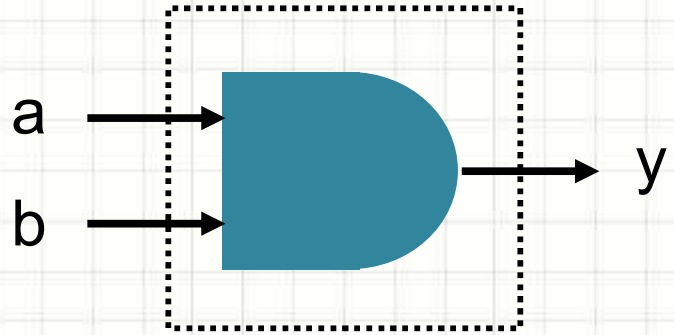
Physical

- PORTS - external
- SIGNALS - internal
  - Can be simply wires or have registers

Abstract

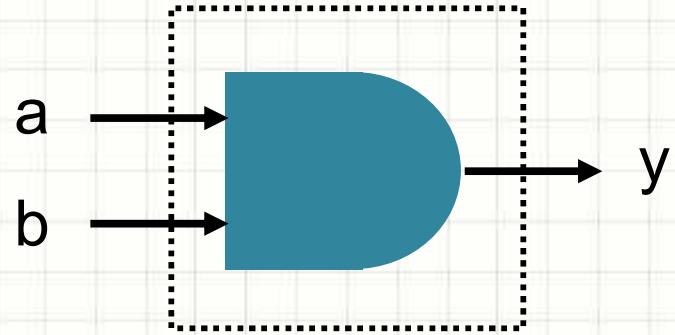
- Variables
  - may or may not correspond to physical objects

Interfaces: entities and ports




```
ENTITY and_gate IS
PORT (a: IN BIT;
      b: IN BIT;
      y: OUT BIT);
END and_gate;
```

Functionality: architectures




```
ARCHITECTURE data_flow OF and_gate IS
BEGIN
    y <= a AND b;
END data_flow;
```

Meaning of signal assignment

y <= a AND b;

Assignment: compute RHS, assign computed value to LHS

When is it executed? Notion of time?

Once initially, then every time there is a change in the value of a or b.

It is considered to be ***sensitive*** to a and b.

Combinational circuit example

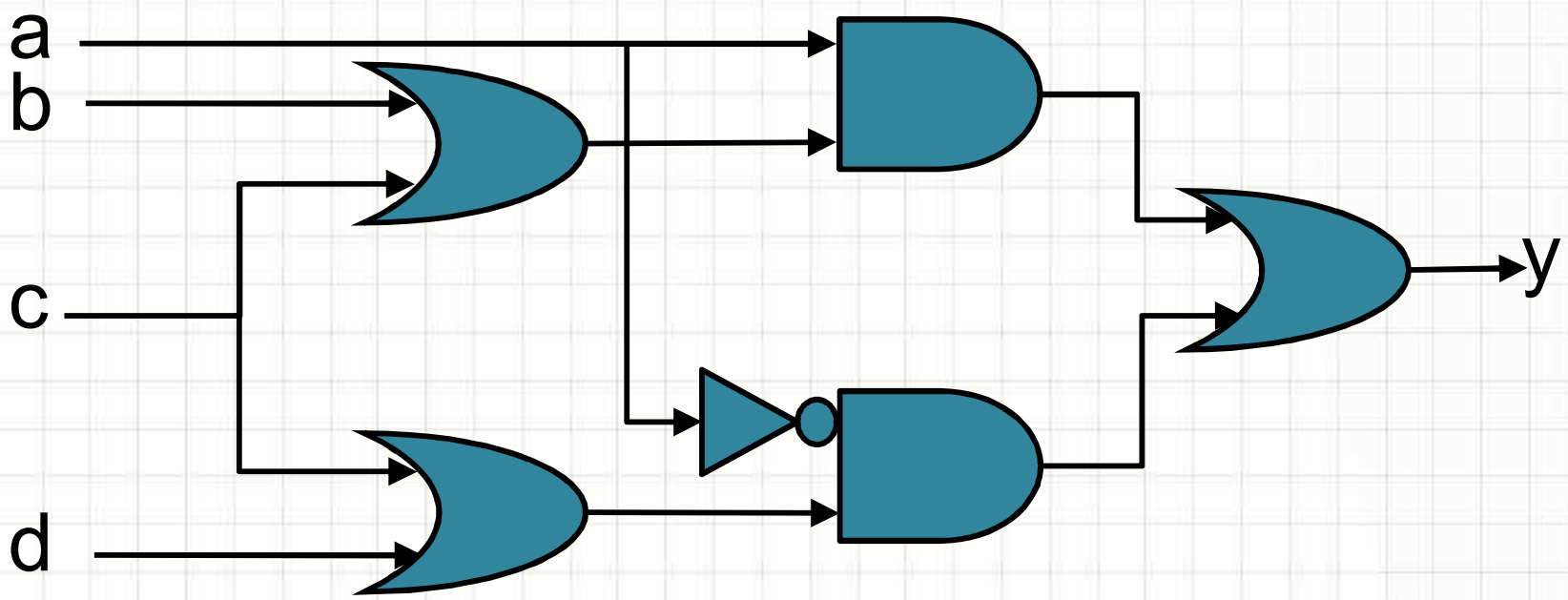
```
ENTITY cc IS
PORT (a, b, c, d: IN BIT;
          y: OUT BIT);
END cc;

ARCHITECTURE dd OF cc IS
BEGIN
    y <= a AND (b OR c) OR NOT a AND (c OR d);
END dd;
```

Signal Assignment

Does this expression imply a structure?

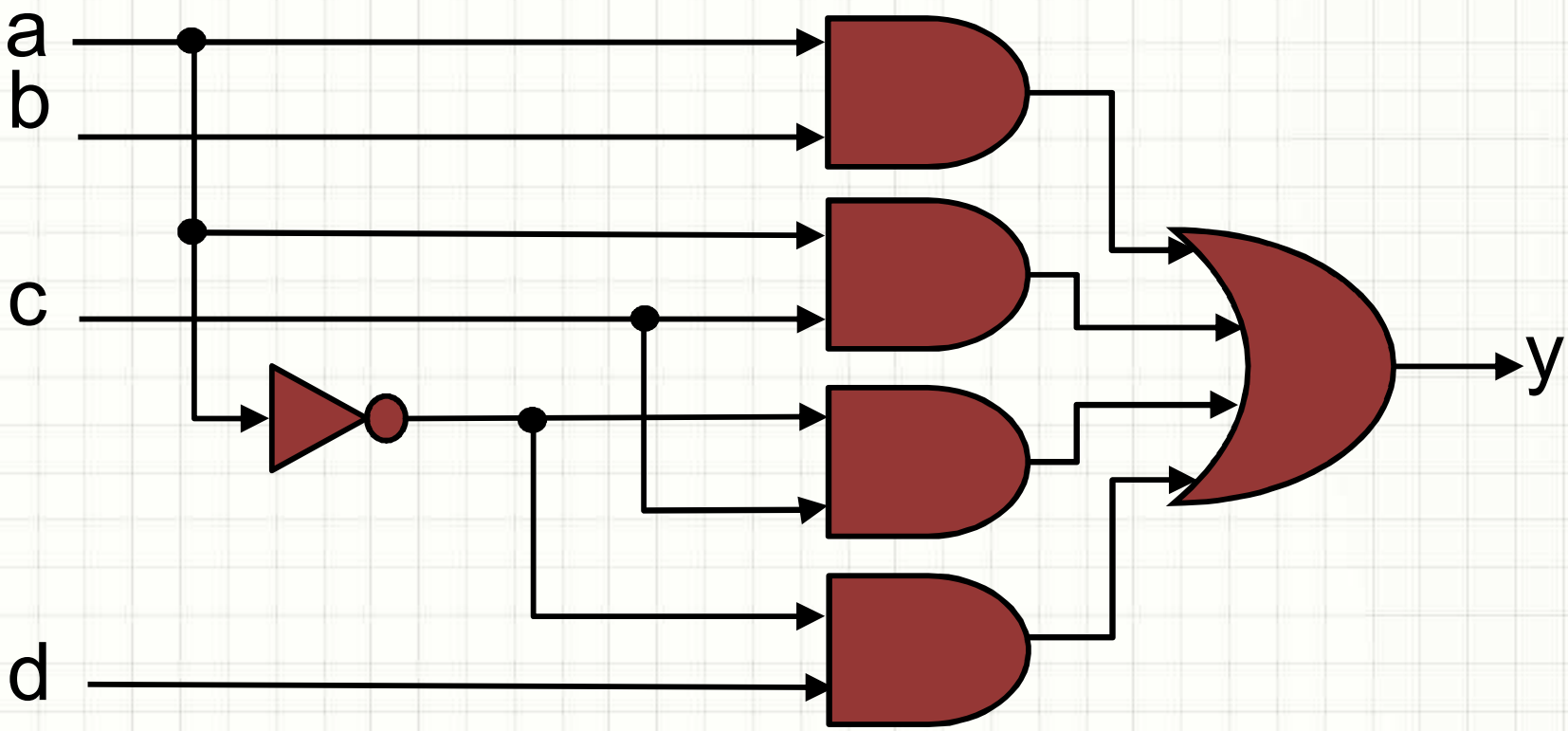
Implementation - 1

y <= a AND (b OR c) OR NOT a AND (c OR d);



Implementation - 2

y <= a AND (b OR c) OR NOT a AND (c OR d);

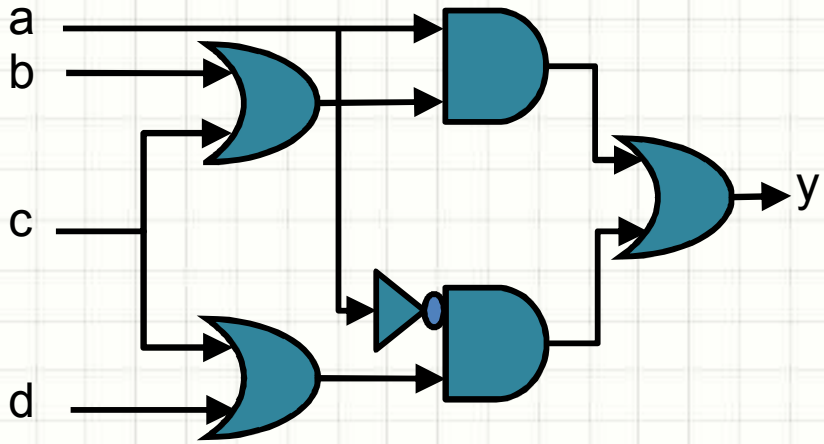


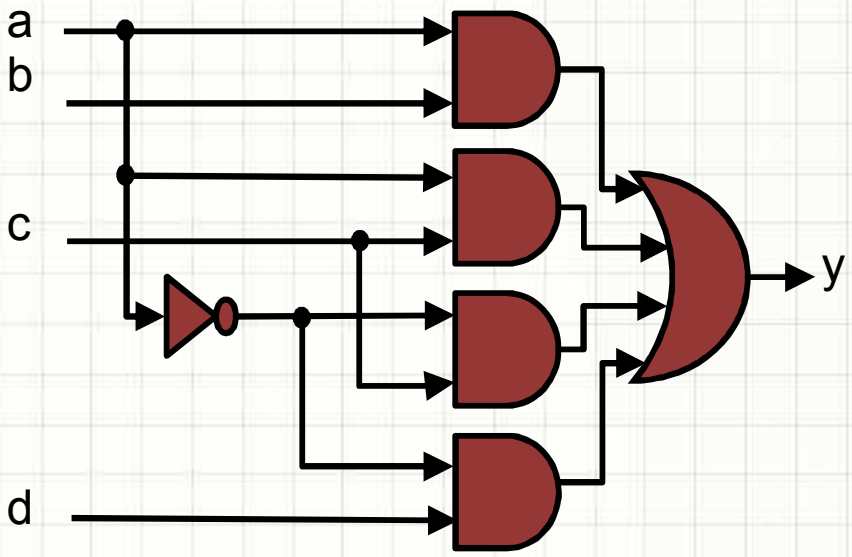
Combinational circuit example

```
ENTITY cc IS PORT (a, b, c, d: IN BIT; y: OUT BIT); END cc;

ARCHITECTURE dd OF cc IS
BEGIN
    y <= a AND (b OR c) OR NOT a AND (c OR d);
END dd;
```

Signal Assignment

Which structure is implied ?

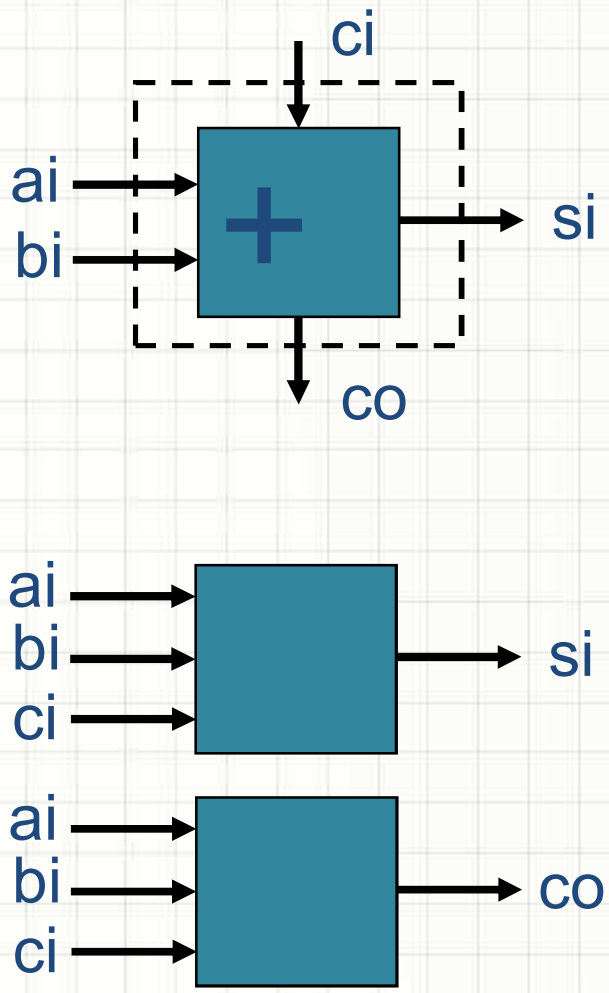




Circuit with multiple outputs

```
ARCHITECTURE data_flow
OF full_adder IS
BEGIN
  si <= ai XOR bi XOR ci;
  co <= (ai AND bi) OR (bi AND ci)
          OR (ai AND ci);
END data_flow;
```

Concurrent Signal Assignments



Is the order significant?

```
ARCHITECTURE data_flow1 OF full_adder IS
BEGIN
  si <= ai XOR bi XOR ci;
  co <= (ai AND bi) OR (bi AND ci) OR (ai AND ci);
END data_flow1;
```

```
ARCHITECTURE data_flow2 OF full_adder IS
BEGIN
  co <= (ai AND bi) OR (bi AND ci) OR (ai AND ci);
  si <= ai XOR bi XOR ci;
END data_flow2;
```

THANKS