

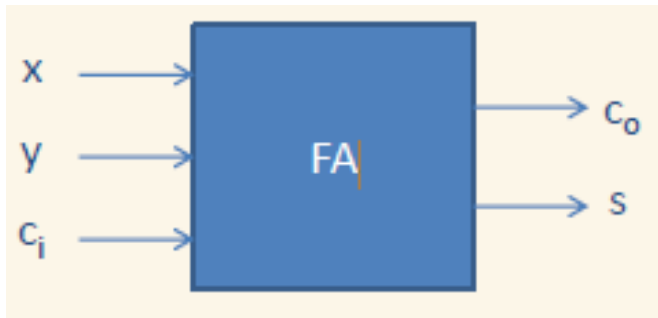
Module 4

VHDL Structure

Design

- A VHDL design includes the following elements
 - Entity Declaration
 - Describes the external interface to the outside world
 - Keywords used: **entity, is, port, end**
 - Architectural Body
 - Describes the internal behavior of the VHDL circuitry
 - Keywords used: **architecture, is, begin, end**

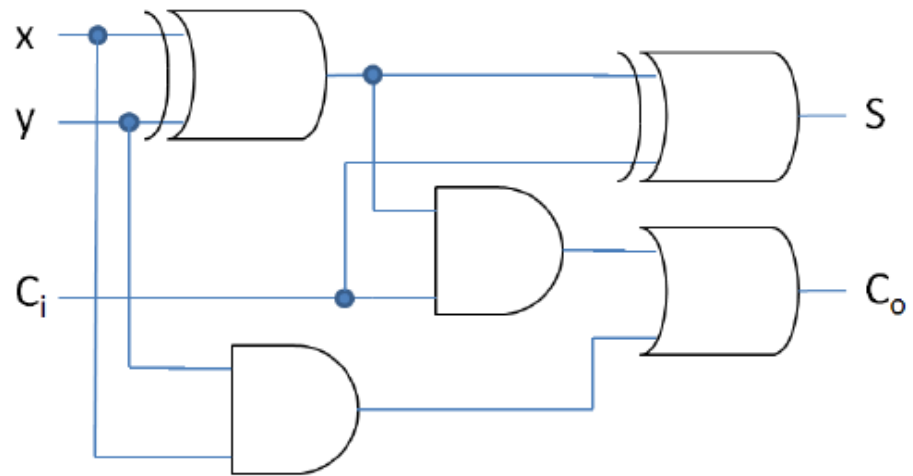
Full Adder Example



$$\begin{aligned} S &= x \oplus y \oplus C_i \\ C_o &= C_i(x + y) + xy \\ C_o &= C_i(x\bar{y} + \bar{x}y + xy) + xy \\ C_o &= C_i(x \oplus y) + xy \end{aligned}$$

x	y	C_i	C_o	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Full Adder Truth Table

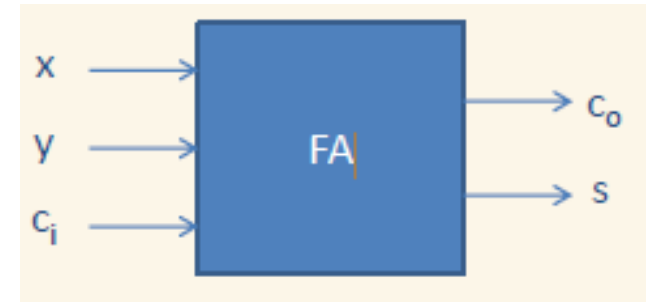


Full Adder Digital Logic Equivalent

Full Adder Example

```
1  -- Dataflow Full Adder
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  entity Full_Adder_1 is
6  port (  S:      out std_logic;
7         C_out:  out std_logic;
8         x:      in  std_logic;
9         y:      in  std_logic;
10        C_in:   in  std_logic);
11
12  end Full_Adder_1;
13
14  architecture dataflow of Full_Adder_1 is
15  begin
16
17      C_out <= (C_in and (x xor y)) or (x and y);
18      S <= x xor y xor C_in;
19
20  end dataflow;
```

Entity Declaration



Architectural Body

VHDL Entities

- Each port member has to have a specific mode and type

- Mode options are:

- **in**

- Input only

- **out**

- Output only

- **inout**

- Bi-directional

- **buffer**

- Output port that can also be read from inside architecture body

```
1  -- Dataflow Full Adder
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  entity Full_Adder_1 is
6  port (  S:      out std_logic;
7         C_out:  out std_logic;
8         x:      in  std_logic;
9         y:      in  std_logic;
10        C_in:   in  std_logic);
11
12  end Full_Adder_1;
```

Port Mode Options

In

Input to the design. Inputs cannot be assigned a value inside architectural body, they only can assign values to outputs and internal signals.

out

Outputs cannot drive other signals or outputs. Outputs are assigned values inside the architectural body.

inout

Bidirectional signal. They can be assigned values from inside the architectural body (acting as output) or they can drive signals inside the architectural body(acting as input)

buffer

Output signal that may also be read from inside the architectural body. Typically used in a feedback type of scenario.

VHDL Entities

- Entities can also contain generics

```
6  entity USR is
7  generic (
8      data_width  : integer := 8);
9  port (
10     A           : out std_logic_vector(data_width - 1 downto 0);
11     I           : in  std_logic_vector(data_width - 1 downto 0);
12     S           : in  std_logic_vector(2 downto 0);
13     reset       : in  std_logic;
14     clk         : in  std_logic);
15 end USR;
```

VHDL Architecture Body

- Architecture body consists of
 - Concurrent signal assignment statements
 - Process statements
 - Component instantiation statements

```
14 architecture dataflow of Full_Adder_1 is
15     begin
16         ...
17         C_out <= (C_in and (x xor y)) or (x and y);
18         S <= x xor y xor C_in;
19         ...
20     end dataflow;
```

Same name as the Entity

Declaration

Statement

Summary

Main elements of a VHDL design are
Entity Declaration
Architecture Body

Each VHDL design contains a single entity declaration and architectural body.

The Entity declaration is where inputs and outputs are specified.