EE214: Introduction to VHDL

Madhav Desai

January 15, 2019

A two-input NAND gate

inputs		output
A	B	Y
0	0	1
0	1	1
1	1	0
1	0	1

$$Y = \overline{A.B}$$

A Vhdl Description

```
library ieee;
use ieee.std_logic_1164.all;
entity NAND2
  port (A, B: in std_logic; Y: out std_logic);
end entity NAND2;
architecture Equations of NAND2 is
begin
  -- concurrent assignment
  Y <= not (A and B);
end Equations;
```

Other two input gates, inverters

Write VHDL descriptions for the following gates:

- Not gate (INVERTER).
- Two input OR gate (OR2), two-input NOR gate (NOR2), two input XOR-gate (XOR2), etc.

A two-input half-adder

inputs		outputs	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

A two-input half-adder

$$S = A \oplus B$$

$$C = A.B$$

A two-input half-adder: VHDL description

```
library ieee;
use ieee.std_logic_1164.all;
entity Half_Adder
  port (A, B: in std_logic; S, C: out std_logic);
end entity Half_Adder;
architecture Equations of Half_Adder is
begin
  -- concurrent assignment
  S <=  (A xor B);
  -- concurrent assignment
  C <=  (A and B);
end Equations;
```

A two-input full-adder

inputs			outputs	
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

A two-input full-adder

$$
\begin{aligned}
S &= (A \oplus B) \oplus Cin \\
Cout &= (A.B) + (A + B).Cin \\
&= (A.B) + (A \oplus B).Cin
\end{aligned}
$$

A two-input full-adder: a VHDL description

```
library ieee;
use ieee.std_logic_1164.all;
entity Full_Adder
  port (A, B, Cin: in std_logic; S, Cout: out std_logic);
end entity Full_Adder;
architecture Equations of Full_Adder is
begin
  -- concurrent assignment
  S <=  (A xor B) xor Cin;
  -- concurrent assignment
  Cout <=  (A and B) or ((A or B) and Cin);
end Equations;
```

A two-input full-adder: another way to build it..

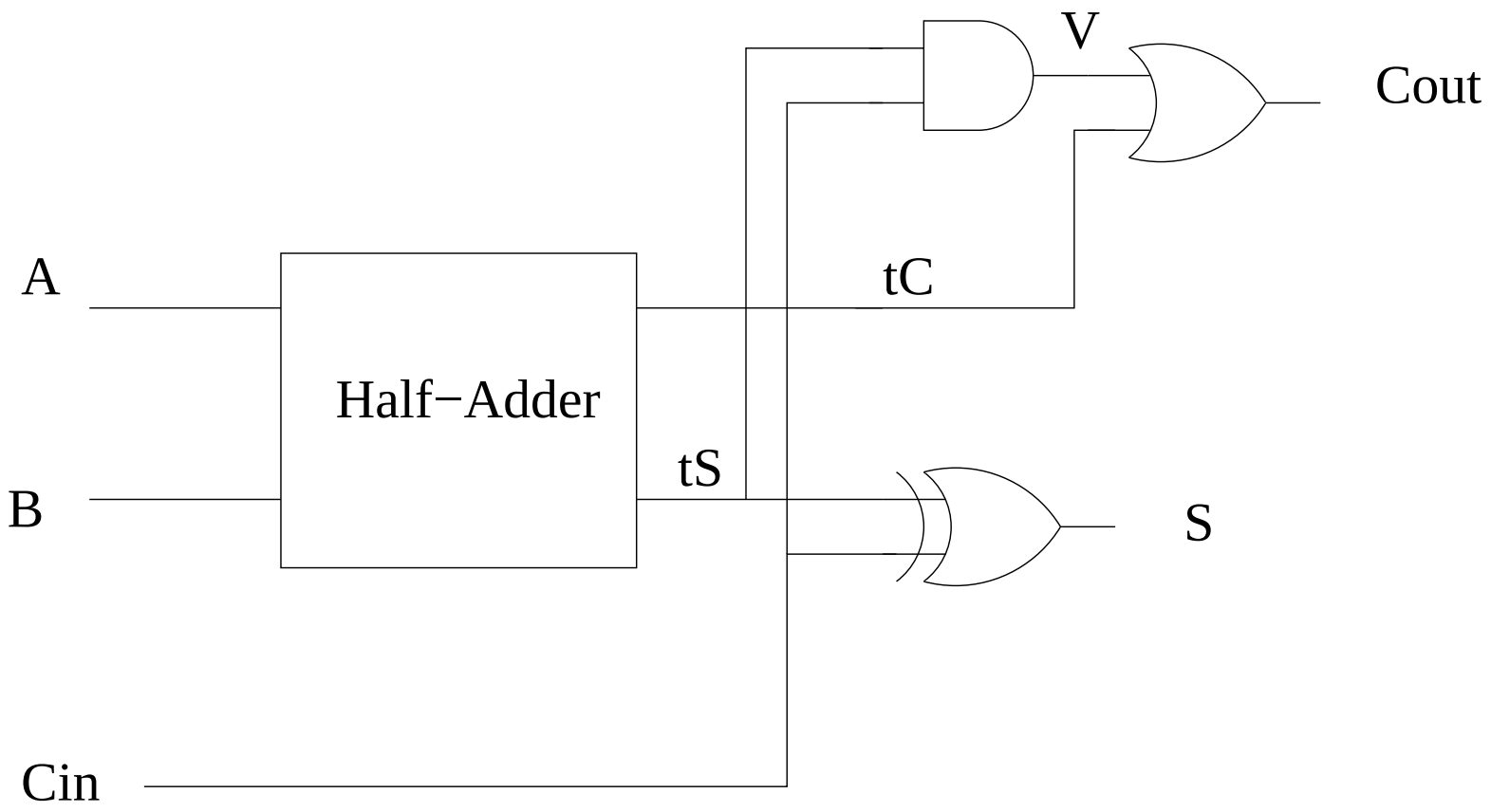

Figure: A full-adder using a half-adder and some more gates.

Components

```
library ieee;
use ieee.std_logic_1164.all;
package Gates is
  component HalfAdder is
   port (A, B: in std_logic; S, C: out std_logic);
  end component;
  component And2 is
   port (A, B: in std_logic; Y: out std_logic);
  end component;
... etc. ....
end package Gates;
```

A two-input full-adder: another VHDL description

```
library ieee;
use ieee.std_logic_1164.all;
library work;
use work.Gates.all;
entity Full_Adder  is
  port (A, B, Cin: in std_logic; S, Cout: out std_logic);
end entity Full_Adder;
architecture Struct of Full_Adder is
  signal tC, tS, U, V: std_logic;
begin
  -- component instances
  ha: HalfAdder
       port map (A => A, B => B, S => tS, C => tC);
  x1: Xor2 port map (A => tS, B => Cin, Y => S);
  a1: And2 port map (A => tS, B => Cin, Y => V);
  o1: Or2  port map (A => V, B => tS, Y => Cout);
end Struct;
```

Board, Socket, Chip analogy

- The architecture is analogous to a circuit board.
- The component is analogous to a socket.
- The entity/architecture corresponding to a component is analogous to a chip.
- When a component is instantiated, a socket is placed on the board.
- The VHDL compiler places a corresponding chip into the sockets (this process is called *binding*). Usually, the default binding is used.

Testbench

- We want a description of a unit which will apply a sequence of inputs to the device under test (DUT), observe the outputs of the DUT and confirm that it will work correctly.

A test trace for a two-input full-adder.

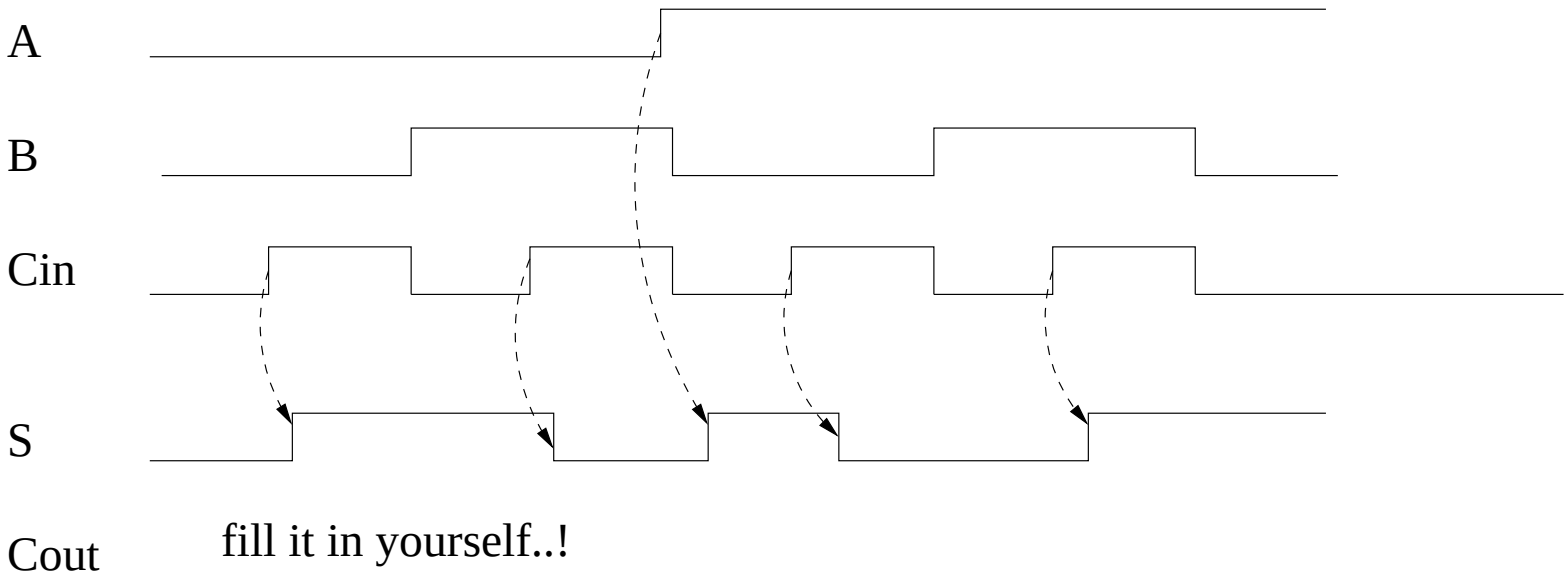

Figure: Test trace for full-adder

Generic Testbench

- We have provided you a generic test-bench which can test any of your designs.
- You will need to wrap your design in a DUT wrapper and specify the input/output trace in a trace-file.
- The test-bench file is then compiled together with your design files, the DUT wrapper file and other package files provided to you.
- The simulation of this compiled design then checks the correctness of your design (relative to the trace-file).
- Your main contribution is the trace-file!

Reference Example

We have provided a reference design example of a full adder for your use. Please study it and discuss it with your TA. Your design experiments can follow the pattern provided by this reference design.