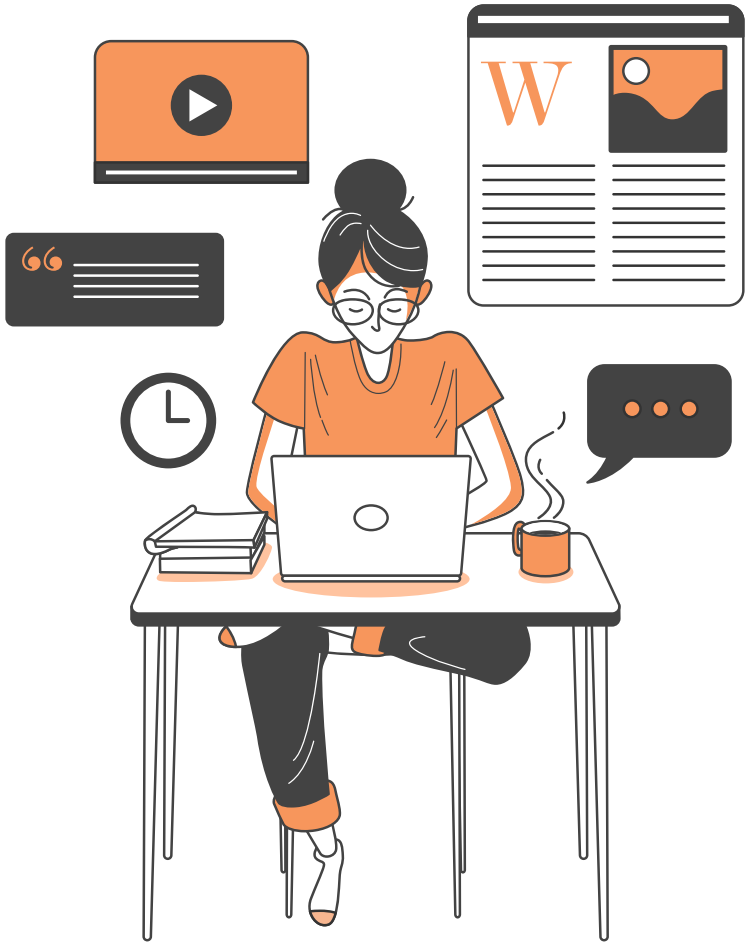
Nadar Saraswathi College of Engineering & Technology

NSCET E-LEARNING PRESENTATION

LISTEN … LEARN… LEAD…



DEPARTMENT OF ELECTRICAL AND ELECTRONICS

II YEAR / IIIrd SEMESTER

EE8351– DIGITAL LOGIC CIRCUITS

M ARIVALAGAN M.Tech., (Ph. D), M.I.S.T.E.,
Assistant Professor & HOD /EEE
Nadar Saraswathi College of & Technology,
Vadapudupatti, Annanji (po), Theni – 625531.

UNIT – V
VHDL

Introduction

- Hardware description languages (HDL)
  - Language to describe hardware
  - Two popular languages
    - VHDL: **V**ery High Speed Integrated Circuits **H**ardware **D**escription **L**anguage
      - Developed by DOD from 1983
      - IEEE Standard 1076-1987/1993/200x
      - Based on the ADA language
    - Verilog
      - IEEE Standard 1364-1995/2001/2005
      - Based on the C language

Applications of HDL

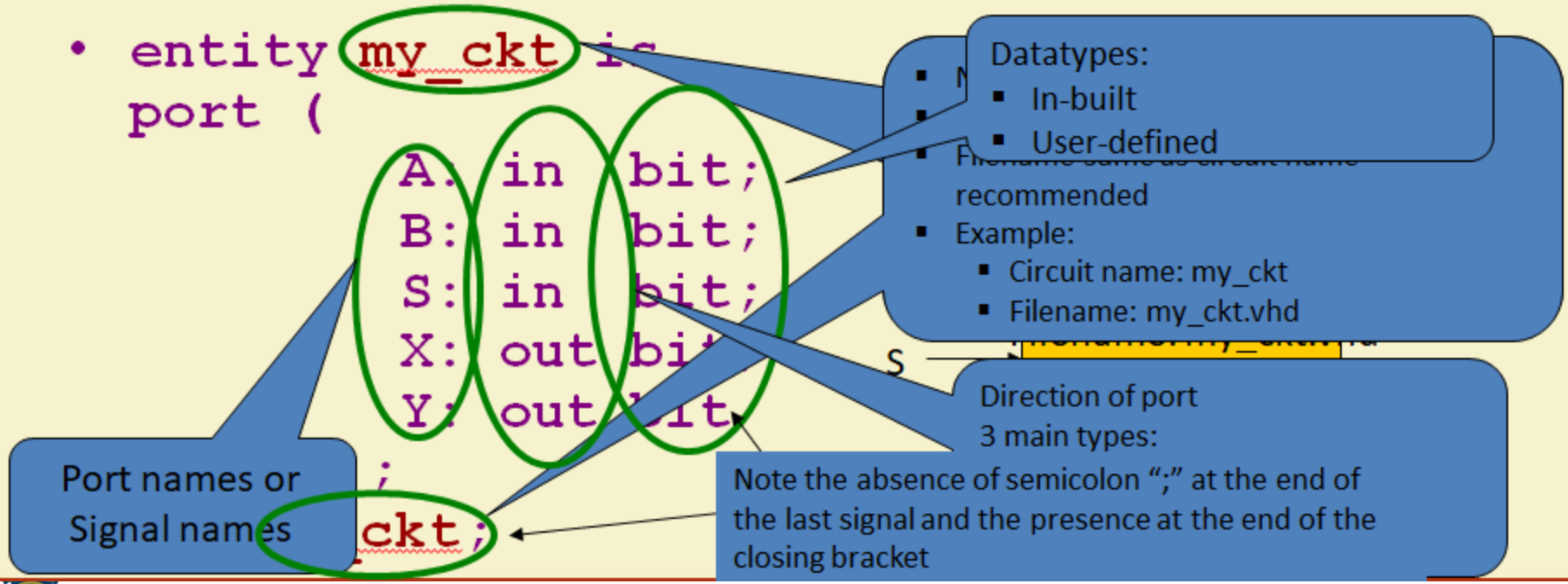
- Model and document digital systems
  - Different levels of abstraction
    - Behavioral, structural, etc.
- Verify design
- Synthesize circuits
  - Convert from higher abstraction levels to lower abstraction levels

Input-Output specification of circuit

- Example: my_ckt
  - Inputs: A, B, C
  - Outputs: X, Y
- VHDL description:

```
entity my_ckt is
port (
        A: in bit;
        B: in bit;
        S: in bit;
X: out bit;
        Y: out bit);
end my_ckt ;
```

VHDL entity


entity my_ckt is
port (
A: in bit;
B: in bit;
S: in bit;
X: out bit;
Y: out bit
);
end my_ckt;
Port names or Signal names
Datatypes:
In-built
User-defined
recommended
Example:
Circuit name: my_ckt
Filename: my_ckt.vhd
Direction of port
3 main types:
Note the absence of semicolon ";" at the end of the last signal and the presence at the end of the closing bracket

Built-in Datatypes

- Scalar (single valued) signal types:
  - `bit`
  - `boolean`
  - `integer`
  - Examples:
    - `A: in bit;`
    - `G: out boolean;`
    - `K: out integer range -2**4 to 2**4-1;`
- Aggregate (collection) signal types:
  - `bit_vector`: array of bits representing binary numbers
  - `signed`: array of bits representing signed binary numbers
  - Examples:
    - `D: in bit_vector(0 to 7);`
    - `E: in bit_vector(7 downto 0);`
    - `M: in signed (4 downto 0);`
      `--signed 5 bit vector binary number`

User-defined datatype

- Construct datatypes arbitrarily or using built-in datatypes
- Examples:
  - `type temperature is (high, medium, low);`
  - `type byte is array(0 to 7) of bit;`

Functional specification

- Example:
  - Behavior for output X:
    - When S = 0
      X <= A
    - When S = 1
      X <= B
  - Behavior for output Y:
    - When X = 0 and S =0
      Y <= '1'
    - Else
      Y <= '0'

VHDL Architecture

- VHDL description (sequential behavior):

```
architecture arch_name of my_ckt is
begin
 p1: process (A,B,S)
 begin
  if (S='0') then
   X <= A;
  else
   X <= B;
  end if;

  if ((X = '0') and (S = '0')) then
   Y <= '1';
  else
   Y <= '0';
  end if;

 end process p1;
end;
```

Error: Signals defined as output ports can only be driven and not read

VHDL Architecture

```
architecture behav_seq of my_ckt is

    signal Xtmp: bit;

    begin
     p1: process (A,B,S,Xtmp)
       begin
       if (S='0') then
        Xtmp <= A;
       else
        Xtmp <= B;
       end if;

       if ((Xtmp = '0') and (S = '0')) then
        Y <= '1';
       else
        Y <= '0';
       end if;

        X <= Xtmp;
      end process p1;
end;
```

Signals can only be defined in this place before the **begin** keyword

General rule: Include all signals in the sensitivity list of the process which either appear in relational comparisons or on the right side of the assignment operator inside the process construct.

In our example:

Xtmp and **S** occur in relational comparisons

A, **B** and **Xtmp** occur on the right side of the assignment operators

VHDL Architecture

- VHDL description (concurrent behavior):

```
architecture behav_conc of my_ckt is
   signal Xtmp: bit;
begin
   Xtmp <= A when (S='0') else
           B;
   Y <= '1' when ((Xtmp = '0') and (S = '0')) else
        '0';
   X <= Xtmp;
end ;
```

Signals vs Variables

- Signals
  - Signals follow the notion of 'event scheduling'
  - An event is characterized by a (time,value) pair
  - Signal assignment example:

    `x <= xtmp;` means

    Schedule the assignment of the value of signal `xtmp` to signal `x` at (Current time + delta)

    where delta: infinitesimal time unit used by simulator for processing the signals

Signals vs Variables

- Variables
  - Variables do not have notion of 'events'
  - Variables can be defined and used only inside the `process` block and some other special blocks.
  - Variable declaration and assignment example:

```
process (…)
variable K : bit;
 begin
   …
  -- Assign the value of signa                    ately
   K := L;
  …
end process;
```

Variables can only be defined and used inside the `process` construct and can be defined only in this place

Simulation

- Simulation is modeling the output response of a circuit to given input stimuli
- For our example circuit:
  - Given the values of A, B and S
  - Determine the values of X and Y
- Many types of simulators used
  - Event driven simulator is used popularly
  - Simulation tool we shall use: ModelSim

Simulation

```
architecture behav_seq of my_ckt is
     signal Xtmp: bit;

     begin
      p1: process (A,B,S,Xtmp)
       variable XtmpVar: bit;
      begin

       if (S='0') then
        Xtmp <= A;
       else
        Xtmp <= B;
       end if;

       if ((Xtmp = '0') and (S = '0')) then
        Y <= '1';
       else
        Y <= '0';
       end if;

       X <= Xtmp;
       XtmpVar := Xtmp;
      end process p1;
end;
```

Time 'T'	A	B	S	Xtmp	Y	XtmpVar	X
0-	U	U	U	'X'	'X'	'X'	'X'
0	0	1	0	'X'	'X'	'X'	'X'
0+d	0	1	0	0	0	0	'X'
0+2d	0	1	0	0	1	0	0
1	0	1	1	0	1	0	0
1+d	0	1	1	1	0	0	0
		1	1	1	0	0	1

Scheduled events

Scheduled events executed:
Xtmp = 0
Y = 1

Scheduled events list:
(empty)

Synthesis

- Synthesis:
  Conversion of behavioral level description to structural level netlist
  - Abstract behavioral description maps to concrete logic-level implementation
  - For ex. Integers at behavioral level mapped to bits at structural level
- Structural level netlist
  - Implementation of behavioral description
  - Describes interconnection of gates
- CAD tools are used to perform synthesis

Structural level netlist

- Behavior of our example circuit:
  - Behavior for output X:
    - When S = 0
      X <= A
    - When S = 1
      X <= B
  - Behavior for output Y:
    - When X = 0 and S =0
      Y <= 1
    - Else
      Y <= 0

- Logic functions
  - Sbar = ~ S
  - Xbar = ~ X
  - X = A*(Sbar) + B*S
  - Y = (Xbar)*(Sbar)

Structural level netlist

```
architecture behav_conc of my_ckt is
   -- component declarations

   signal Sbar, Xbar, W1, W2: bit;

begin

   G1: not port map(Sbar,S);
   G2: and port map(W1,A,Sbar);
   G3: and port map(W2,B,S);
   G4: or  port map(X,W1,W2);

   G5: not port map(Xbar,X);
   G6: and port map(Y,Xbar,Sbar);

end ;
```

- Gate level VHDL descriptions (`and`, `or`, etc) are described separately
- Design in which other design descriptions are included is called a “hierarchical design”
- A VHDL design is included in current design using port map statement

STRUCTURAL MODELING

1. Consider an example of a Dual 2 by 1 multiplexer using structural modeling.
2. Components required : 2- input AND gate
   2- input OR gate Inverter

```
ENTITY inv IS
        PORT(
                i1: IN BIT;
                o1: OUT BIT );
        END inv;
```

```
ARCHITECTURE single_delay OF inv IS  BEGIN

        o1 <= not (i1) after 5ns;  END single_dela

y;
```

```
ARCHITECTURE single_delay OF inv IS  BEGIN

        o1 <= not (i1) after 5ns;  END

single_delay;
```

DESIGN OF A SIMPLE 2 INPUT AND GATE WITH A DELAY

```
ENTITY and2 IS
        PORT(
                i1, i2 :IN BIT;  o1:
                        OUT BIT
                );
END and2;
ARCHITECTURE single_delay OF and2 IS  BEGIN

    END o1 <= (i1 AND i2) after 8ns

      single_delay;
```

DESIGN OF A SIMPLE 2 INPUT OR GATE WITH A DELAY

```
ENTITY or2 IS
    PORT(
            i1, i2 :IN BIT;
            o1:   OUT
            BIT);
END or2;
ARCHITECTURE single_delay OF
or2 IS BEGIN
ENDo1 <= (i1 or i2) after 8ns;
single_delay;
```

SINGLE AND DUAL 2-1 MULTIPLEXER ENTITY

```
ENTITY
    single_mux IS
    PORT(
            s: IN BIT ; --select input
            iA, iB : INT BIT; -- iA is selected
    if                                          -
    -  s='0' and iB is selected if s='1'
EN      singleoZnuOUT BIT –output of the mux
D       x;                  );
```

THANKS!

Department of EEE, NSCET, Theni

Nadar Saraswathi College of Engineering & Technology