

Circuit Representation: Hardware Description Languages (HDL)

Virendra Singh

Professor

Computer Architecture and Dependable Systems Lab

Department of Computer Science & Engineering

Indian Institute of Technology Bombay

<http://www.cse.iitb.ac.in/~viren/>

E-mail: viren@cse.iitb.ac.in

CS-232 Digital Logic Design & Computer Architecture Lab.



Lecture-VHDL: 12 January 2022 CADSL

Modeling Digital Systems

- VHDL is for writing models of a digital system

- Semi-formal representation

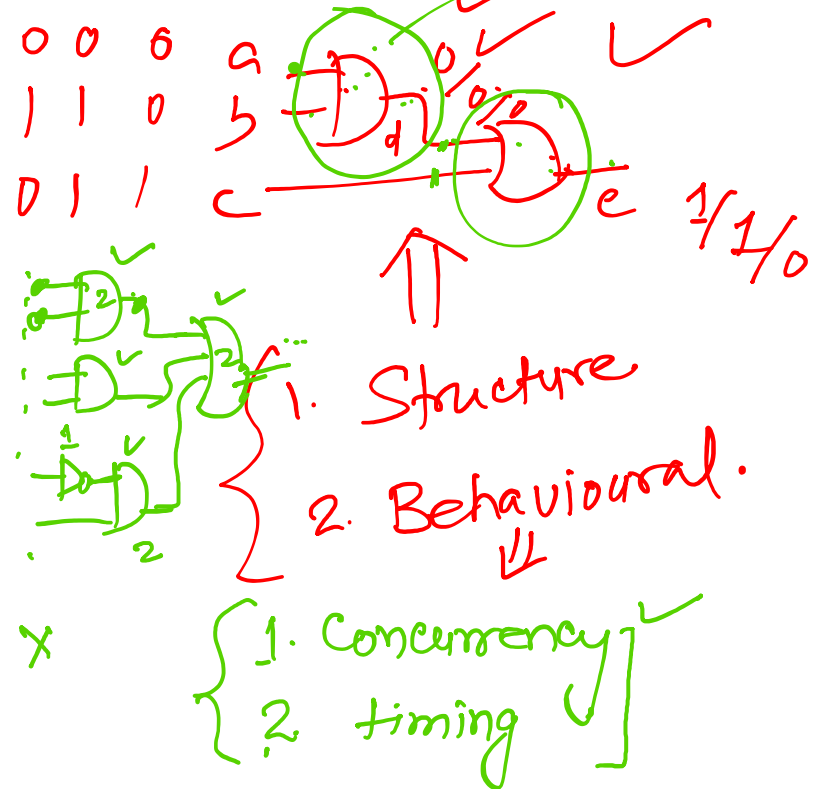
- Reasons for modeling

- requirements specification
- documentation
- testing using simulation
- formal verification

- synthesis

- Goal

- most reliable design process, with minimum cost and time
- avoid design errors!



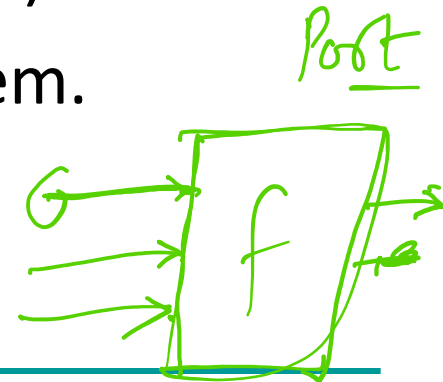
What is VHDL?

- **V**ery High Speed Integrated Circuit **H**ardware **D**escription **L**anguage
- Used to describe a desired logic circuit
- Compiled, Synthesized and burned onto a working chip
- Simplifies hardware for large projects ✓



VHDL

- VHDL is a **programming language** that allows one to model and develop complex digital systems in a dynamic environment.
- Object Oriented methodology -- modules can be used and reused.
- Allows you to designate in/out ports (bits) and specify behavior or response of the system.



VHDL

- But VHDL is NOT C ...

There are some similarities, as with any programming language, but syntax and logic are quite different; so get over it !!

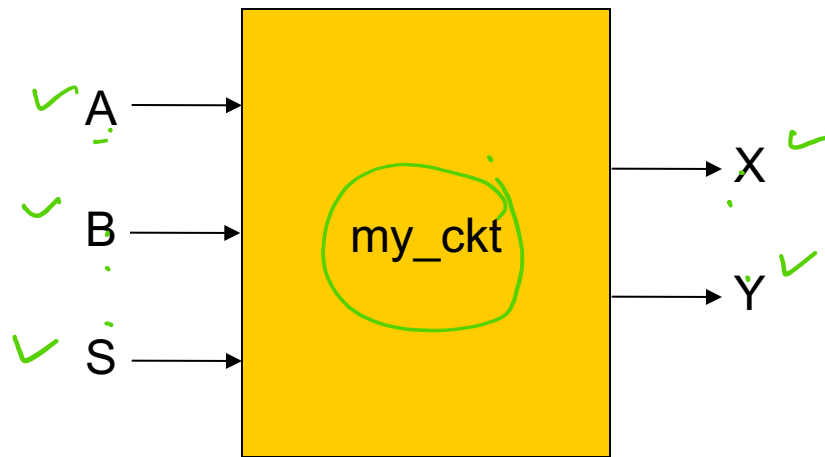


HDL Requirements

- Abstraction ✓
- Modularity
- Concurrency
- Hierarchy



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 ;
```

{0,1}

External Interface: 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

name recommended

▪ Example:

- Circuit name: my_ckt
- Filename: my_ckt.vhd

Direction of port
3 main types:

- in: Input

Note the absence of semicolon “;” at
the end of the last signal and the
presence at the end of the closing
bracket

nal



Built-in Datatypes

- Scalar (single valued) signal types:
 - bit
 - boolean
 - integer
 - Examples:
 - A: in bit;
 - G: out boolean;
- 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); D(0)
 - E: in bit_vector(7 downto 0); D(7)
 - M: in signed (4 downto 0);
--signed 5 bit_vector binary number



Modeling the Behavior Way

- Architecture body ✓ Entity
 - describes an implementation of an entity
 - may be several per entity
- *Behavioral architecture*
 - describes the algorithm performed by the module
 - contains
 - signal assignment statements



Syntax of the Architecture

architecture <architecture_name> **of** <entity_identifier> **is**
[<architecture_declarative_part>]

begin

<architecture_statement_part> -- The body of the arch.

end [*architecture*] [<architecture_name>];

- The word “architecture” in the last line is not supported before the VHDL-93 standard



Entity

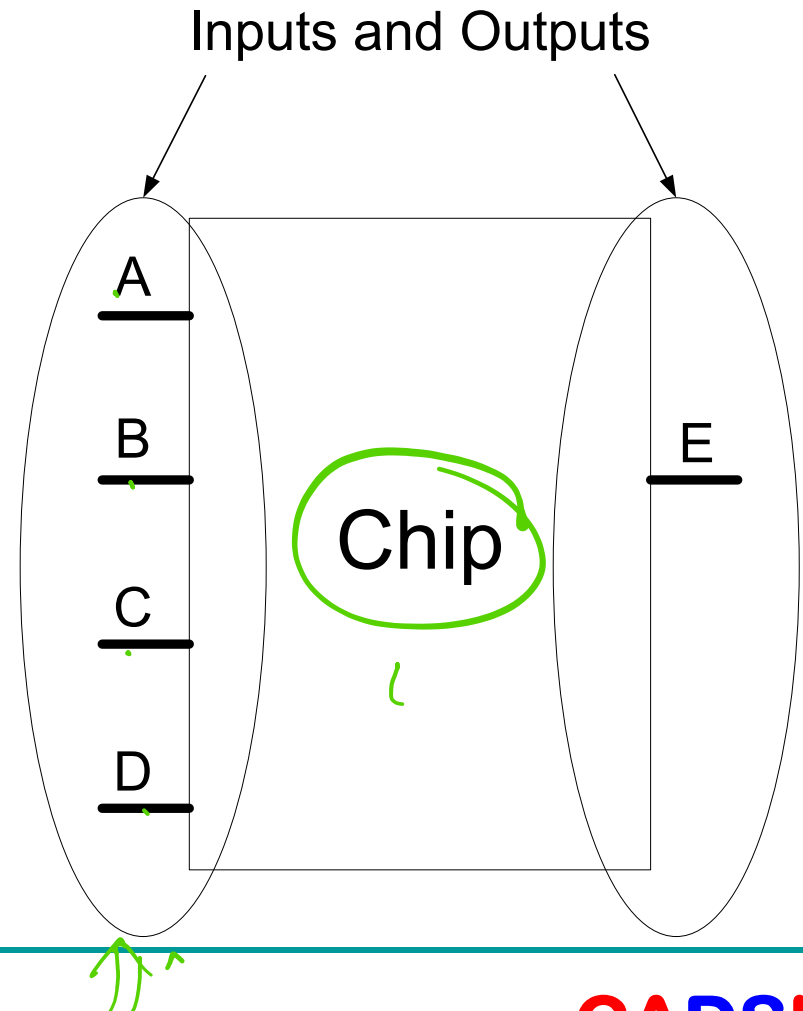
- Define inputs and outputs
- Example:

A: in bit;
B: in bit;
:

Entity test is

Port(A,B,C,D: in bit;
E: out bit);

End test;



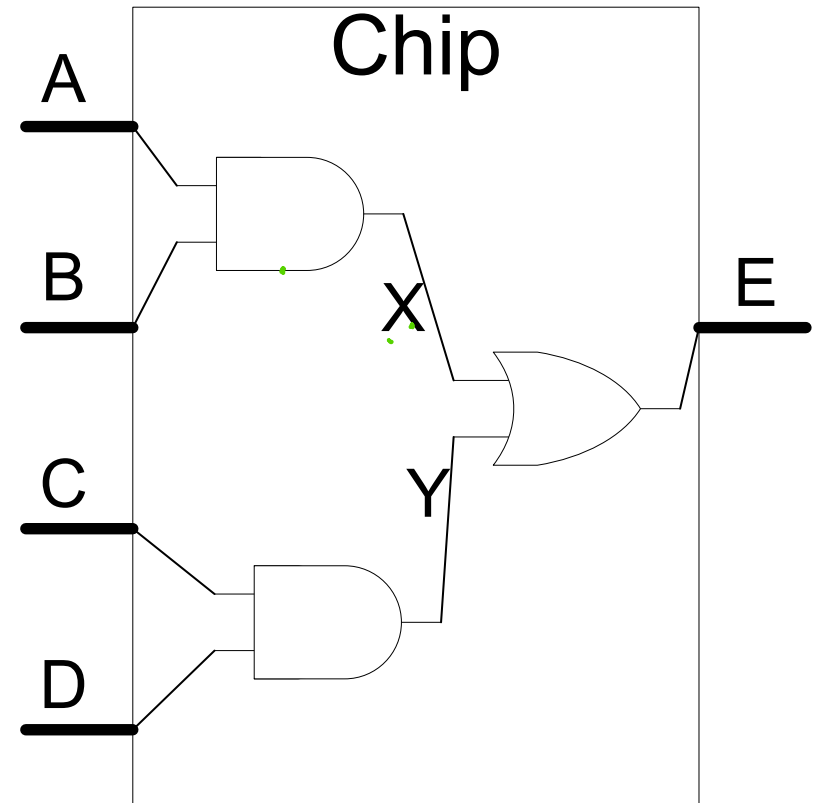


Architecture

- Define functionality of the chip



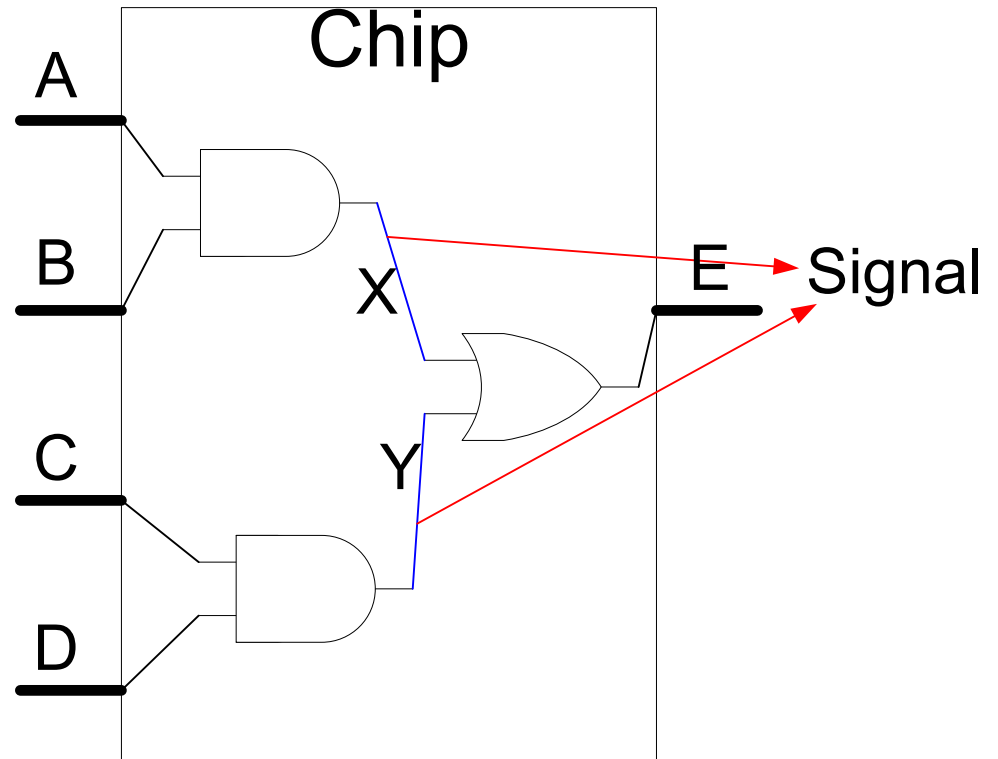
- $X \leq A \text{ and } B;$
- $Y \leq C \text{ and } D;$
- $E \leq X \text{ or } Y;$



Signal

- All internal variables

signal X,Y : bit;



Architecture

architecture behaviour of test is
signal X,Y : bit;

begin

X <= A and B;
Y <= C and D;
E <= X or Y;

end behaviour;



Final code

```
library IEEE;
use IEEE.std_logic_1164.all;
entity TEST is
port (A,B,C,D : in bit;
      E       : out bit);
end TEST;
architecture BEHAVIOR of TEST is
signal X,Y : bit;
begin
    X <= A and B;
    Y <= C and D;
    E <= X or Y;
end BEHAVIOR;
```

Behav



VHDL Features

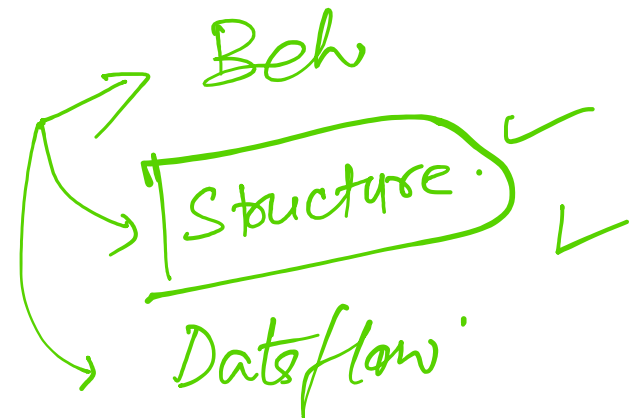
- Case insensitive
 - inputa, INPUTA and InputA are refer to same variable
- Comments
 - '--' until end of line
 - If you want to comment multiple lines, '--' need to be put at the beginning of every single line
- Statements are terminated by ';
- Signal assignment:
 - '<='
- User defined names:
 - letters, numbers, underscores ('_')
 - start with a letter



VHDL Structure

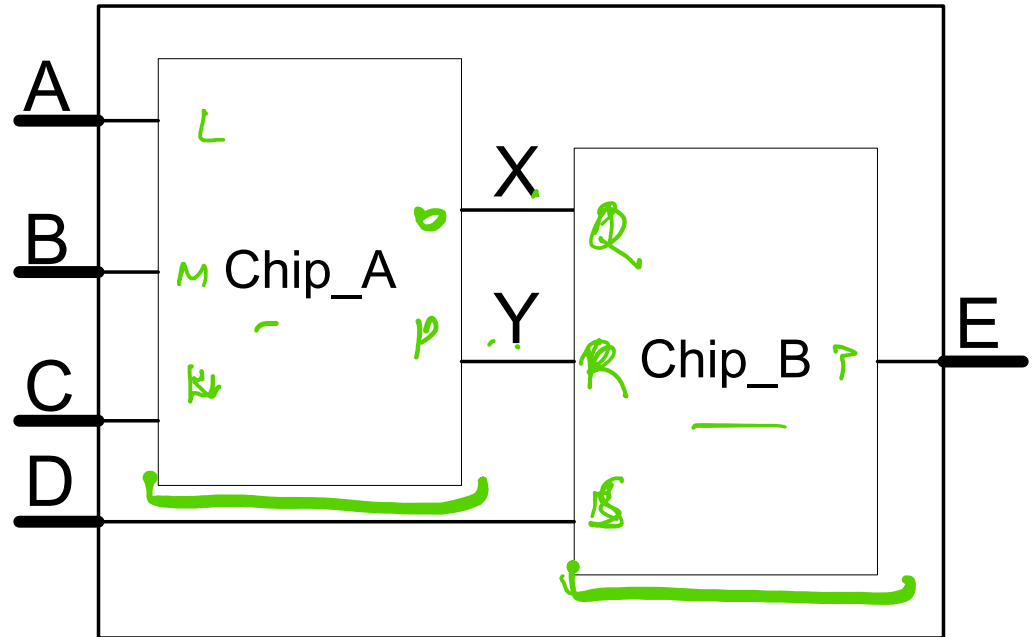
- ✓ • Library
 - Definitions, constants
- { • Entity
 - Interface
- { • Architecture
 - Implementation, function

$c \leftarrow A \text{ and } B;$



```
Chip1 : Chip_A ✓  
port map (A,B,C,X,Y);
```

```
Chip2 : Chip_B ✓  
port map (X,Y,D,E);
```



Final code

```
library IEEE;
use ieee.std_logic_1164.all;

entity TEST is
port (A,B,C,D : in bit;
      E       : out bit);
end TEST;

architecture BEHAVIOR of TEST is
  signal X,Y : bit;
  component Chip_A
  port (L,M,N : in bit;
        O,P   : out bit);
  End component;
```

```
component Chip_B
port (Q,R,S : in bit;
      T     : out bit);
end component;

begin

  Chip1 : Chip_A
  port map (A,B,C,X,Y);
  Chip2 : Chip_B
  port map (X,Y,D,E);

end BEHAVIOR;
```

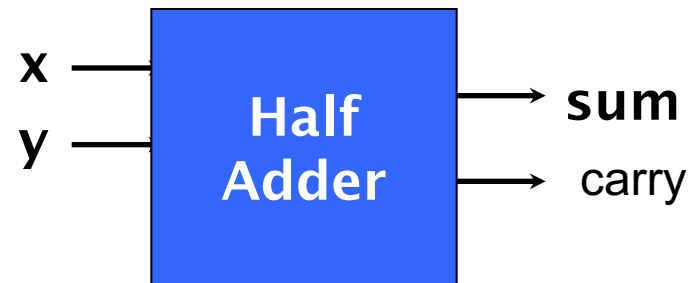


VHDL Design Example

Entity Declaration

- As a first step, the entity declaration describes the interface of the component
 - input and output *ports* are declared

```
entity half_adder is  
  port( x, y: IN BIT;  
        carry, sum: OUT BIT);  
end half_adder;
```



Syntax of the Architecture

architecture <architecture_name> **of** <entity_identifier> **is**
[<architecture_declarative_part>]

begin

<architecture_statement_part> -- The body of the arch.

end [***architecture***] [<architecture_name>];

- The word “architecture” in the last line is not supported before the VHDL-93 standard



VHDL Design Example

Behavioral Specification

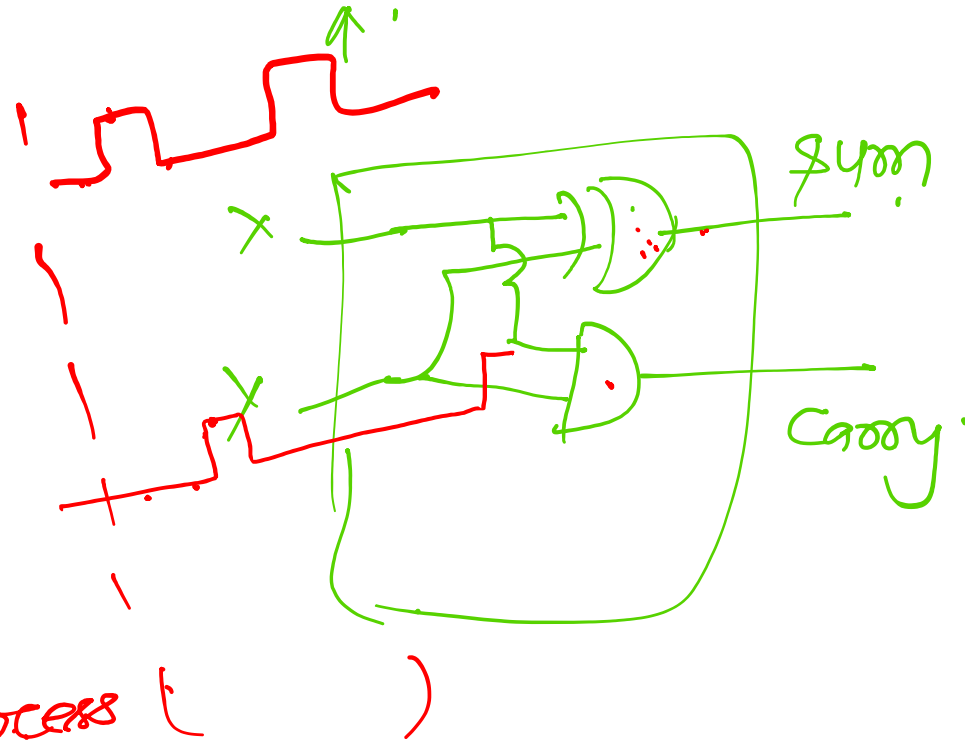
```
architecture half_adder_a of half_adder is  
begin
```

```
    result <= x xor y;  
    carry <= x and y;
```

```
end half_adder_a;
```

result <= x xor y after 2ns;

*process t
begin
end*



Concurrency in VHDL

- Achieved through processes
- Concurrent assignments are also process by itself
- These are non-terminating
- Communicating through signals
- Variables are allowed inside the processes
- Multiple processes are active at the same time



Thank You

