CpE111
Introduction to Computer Engineering

Dr. Minsu Choi

CH 5: VHDL

UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.

Introduction to VHDL

- VHDL: VHSIC Hardware Description Language.
- VHSIC: Very High Speed Integrated Circuit; Dept of Defense first used it.
- VHDL is pronounced by just reading the letters. VHSIC, on the other hand, is pronounced as vis-hic.
- VHDL is a CAD (Computer Aided Design) tool for digital logic networks.

How to Describe a Module (or a Unit) in VHDL?

1. Define the block itself by giving it a name, and by specifying the input and output ports -> called "entity" statement.
2. Specify what the module actually does (i.e., how the outputs are related to the inputs) -> called "architecture" statement.

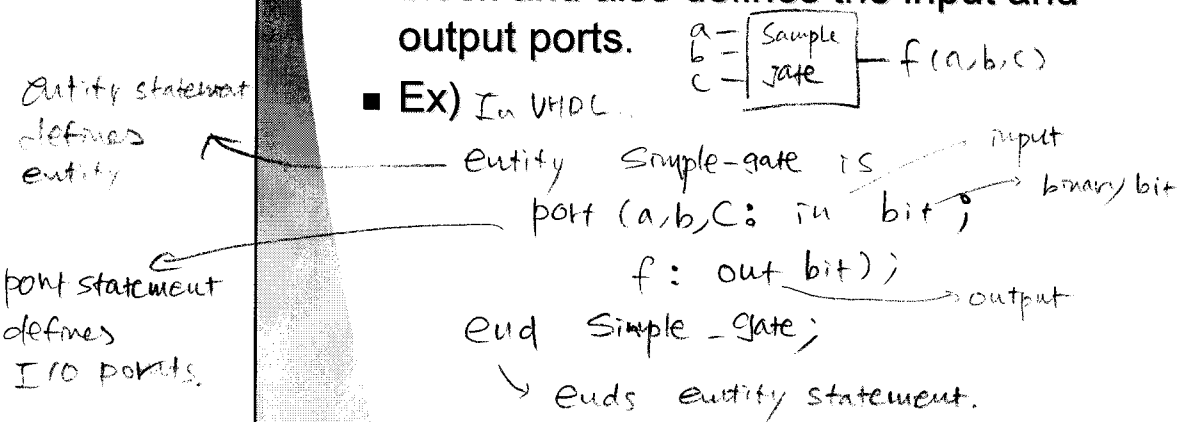
- Reserved words and defined symbols for VHDL are shown in next slide.

abs	file	of	sra		
access	for	on	srl		
after	function	open	subtype		
alias		or			
all	generate	others	then		
and	generic	out	to		
architecture	group		transport		
array	guarded	package	type		
assert		port			
attribute	if	postponed	unaffected		
	impure	procedure	units		
begin	in	process	until		
block	inertial	pure	use		
body	inout				
buffer	is	range	variable		
bus		record			
	label	register	wait		
case	library	reject	when		
component	linkage	rem	while		
configuration	literal	report	with		
constant	loop	return			
		rol	xor		
disconnect	map	ror	xnor		
downto	mod				
	nand	select			
else	new	severity			
elseif	next	shared			
end	nor	signal			
entity	not	sla			
exit	null	sll			

Symbol	Meaning
+	Addition, or positive number
–	Subtraction, or negative number
/	Division
=	Equality
<	Less than
>	Greater than
&	Concatenator
|	Vertical bar
;	Terminator
#	Enclosing based literals
(	Left parenthesis
)	Right parenthesis
.	Dot notation
:	Separates data object from type
"	Double quote
'	Single quote or tick mark
**	Exponentiation
=>	Arrow meaning "then"
=>	Arrow meaning "gets"
:=	Variable assignment
/=	Inequality
>=	Greater than or equal to
<=	Less than or equal to
<=	Signal assignment
<>	Box
--	Comment

Entity Statement

- Identifies the module as a distinct logic block and also defines the input and output ports.
- Ex) In VHDL



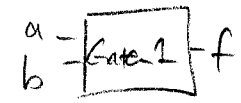
Architecture Declaration

- Describes exactly what the block does.
- Three major classifications for architectural declaration are:
  1. Behavioral descriptions where we provide an exact relationship between the inputs and the outputs.
  2. Structural listings that construct logic functions by combining more primitive elements, such as gates.
  3. Dataflow models that describe modules by defining the "flow" of the data signals. This can be viewed as a type of behavioral model.

General VHDL Architectural Description Format

architecture *type_of_description* **of** *gate_name* **is**
  Declaration_1;
  Declaration_2;
  ...
begin
  Statement_1;
  Statement_2;
  ...
end *type_of_description*;

Ex) OR2 Gate



Keyword	Example	Meaning
not	not *a*	$\overline{a}$
and	*c* and *d*	$c \cdot d$
or	*u* or *v*	$u+v$
xor	*x* xor *y*	$x \oplus y$
nand	*a* nand *b*	$\overline{a \cdot b}$
nor	*c* nor *d*	$\overline{c+d}$
xnor	*x* xnor *y*	$\overline{x \oplus y}$

predefined logic operators

```
-- First example
entity Gate_1 is
  port (a,b : in bit;
        f : out bit);
end Gate_1;
architecture Logic of Gate_1 is
begin
  f <= a or b;
end Logic;
```

name of declaration

gate name

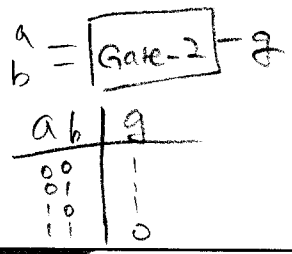
assignment operator

OR predefined logic operator

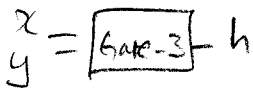
should be same

a	b	f
0	0	0
0	1	1
1	0	1
1	1	1

comment



a b	g
0 0	1
0 1	1
1 0	1
1 1	0



a b	h
0 0	1
0 1	0
1 0	0
1 1	0

More Examples

```
entity Gate_2 is
  port (a,b : in bit;
        g : out bit);
end Gate_2;
architecture Logic of Gate2 is
begin
    g<= a nand b;
end Logic;
```

same as Gate_2. except gate name (Gate_3) and assignment statement

(h<= x xnor y;)

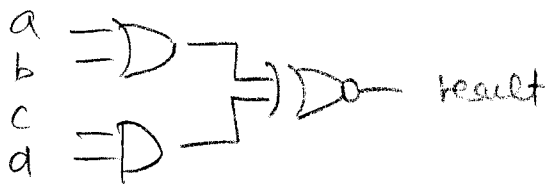
Concurrent Operations

- The keyword operators – not, and, or, xor, nand, nor, and xnor – are used to describe what are called "concurrent" operations -> operations are performed without regarding to any timing constraint.
- f <= x and y and z; <- the value of f is updated whenever there is a change in the status of the inputs x, y and z.

ex) $h = \overline{a + b \cdot \bar{c} + d}$

```
h<= not( a or (b and not() or d));
```

use parentheses to avoid miss interpretation.



Example

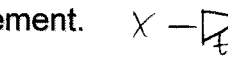
In VHDL

```
entity Gate-Ex is
    port (a, b, c : in bit)
          result : out bit);
end Gate-Ex;
architecture Concurrent of Gate_Ex is
begin
    result <= (a or b) xnor (c and d);
end concurrent;
```

VHDL Identifier Rules

1. Upper and lower case letters + digits + underscore symbol "_" can be used.
2. Single string of characters of any length.
3. Case sensitive: Or_gate_3 is not the same as OR_GATE_3.
4. The underscore "_" cannot be used as the first & last character and not two or more "_"s in a row.

How to Specify Propagation Delay & Transport Delay?

- No propagation delay is considered in concurrent operations.
- To specify propagation delay, use "after" keyword along with the value of the delay in the concurrent operation statement.  , if tp = 2ns
- Ex)

```
Result <= not x after 2ns;
```

- Transport delay: time delay associated with an interconnect wire.
- Ex) output <= transport(x) after 10ps;
- This describes a signal x that is moved to the output after a delay of 10 pico seconds (1ps = 10^-12 sec).

Structural Modeling

- Basic units can be combined to build more complex logic networks. (in structural modeling technique).
- Building blocks are called components.
- Ex) f = ab + cd -> two basic components can be used to build this function: AND2 and OR2.

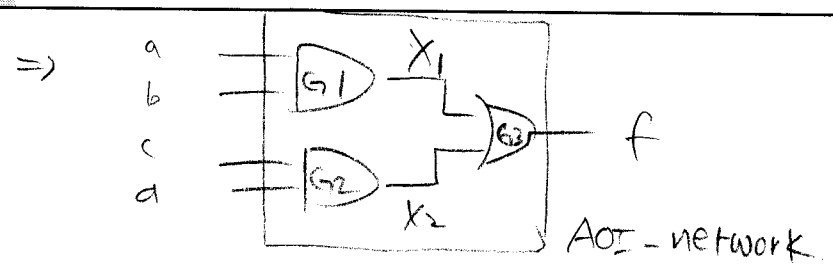
Construct two AND2 terms

$X_1 = a \cdot b$

$X_2 = c \cdot d$

then OR them together

$f = X_1 + X_2$



In VHDL...

```
-- Structural Description
entity AOI_Network is
        port(a, b, c, d : in bit;
                f : out bit);
end AOI_Network;
-- The listing below builds the network using components
architecture Structural of AOI_Network is
-- Define an AND gate as a component
component AND2
        port( x, y : in bit;
                z : out bit);
end component;
-- Define an OR gate as a component
component OR2
        port(x, y: in bit;
                z : out bit);
end component;
-- The next line declares the internal module signals
signal X1, X2 : bit;
-- The port maps specify the internal wiring
begin
        G1:AND2 port map (a, b, X1);
        G2:AND2 port map (c, d, X2);
        G3: OR2 port map (X1, X2, f);
end Structural
```

- In a design library which has AND2 and OR2 gates...

```
entity AND2 is
    port (u, v : in bit;
          g : out bit);
end AND2;
```

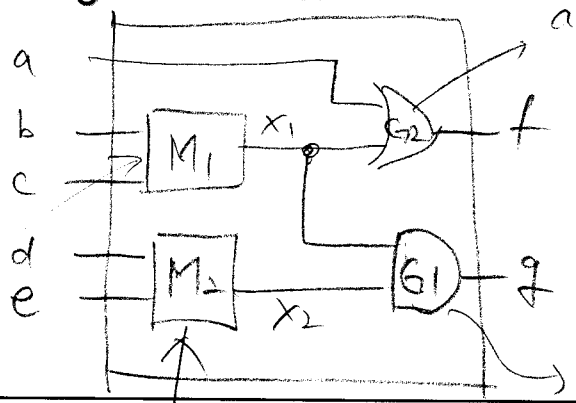
```
architecture Logic of AND2 is
begin
    g <= u and v;
end Logic;
```

OR2 similar.. "OR2"

g <= u or v;

Modular Design using Cell Library

- Choose cells (= unit gates) from a design library, then wire instances to build complex logic networks.
- Ex)



an instance of cell OR2

an instance of cell AND2

an instance of entity Gen_1

Network_Ex

an instance of entity Gen_2

VHDL Listing

```
entity Network_ex is
  port (a,b,c,d,e: in bit;
        f,g : out bit);
end Network_ex;
architecture structural of Network_ex is
  component Gen-1
    port (x,y: in bit;
          g : out bit);
  end component;
  component Gen-2
  :
  AND-2
  :
  OR-2
```

```
-- define internal signals
signal X1, X2: bit;
-- port maps
begin
  M1: Gen-1 port map (b,c,X1);
  M2: Gen-2 port map (d,e,X2);
  G2: OR-2 port map (a,X1,f);
  G1: AND-2 port map (X1,X2,g);
end structural;
```

Conditional Models

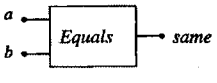
- VHLD conditional statement syntax: <statement1> when <condition> else <statement2>;
- Relational Operators:

=	equal to
/=	not equal to
>	gt
<	lt
>=	get
<=	let

ex)

```
entity Equals is
    port(a, b, : in bit;
         same : out bit);
end Equals;
architecture Dataflow of Equals is
begin
    same <= '1' when a = b else
            '0';
end Dataflow;
```

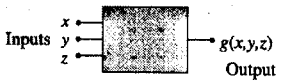
a	b	same
0	0	1
0	1	0
1	0	0
1	1	1



Example

```
entity Table_Example is
      port(x, y, z: in bit;
           g : out bit);
end Table_Example;
architecture Dataflow of Table_Example is
begin
      g <= '1' when (x = '0' and y = '0' and z = '1') else
           '1' when (x = '1' and y = '0' and z = '1') else
           '1' when (x = '1' and y = '1' and z = '0') else
           '1' when (x = '1' and y = '1' and z = '1') else
           '0';
end Table_Dataflow;
```

list these four conditions.



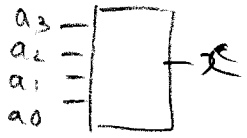
x	y	z	g
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

← we have 4 input bit patterns that result in g = 1.

Binary Words

- n-bit binary word represent 2^n distinct bit patterns.
- VHDL allows us to handle words using arrays of bits.
- Ex) 4-bit word In_a = $a_3a_2a_1a_0$

In_a

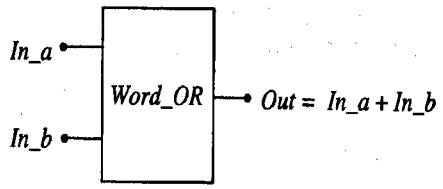


port (In_a : in bit_vector (3 downto 0);
x : out bit);

array of bits

interface from 3 to 0.

Ex) ORing Two 4-bit Words




```
entity Word_OR is
    port(In_a, In_b : in bit_vector (3 downto 0);
         Out : out bit_vector (3 downto 0));
end Word_OR;
architecture Listing of Word_OR is
begin
    Out(3) <= In_a(3) or In_b(3);
    Out(2) <= In_a(2) or In_b(2);
    Out(1) <= In_a(1) or In_b(1);
    Out(0) <= In_a(0) or In_b(0);
end Listing;
```

Libraries

- Library: collection of predefined quantities and procedures that are used by the VHDL compiler to interpret user's code.
- Packages: subgroups of a library.
- Ex) To introduce the ieee library and IEEE 1164 package, we initiate a VHDL description by;

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

☆ only basic concepts in VHDL covered in this chapter. More advanced topics covered in VLSI courses.

Program Completed

University of Missouri-Rolla

UMR
UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.