ES 4: Introduction to VHDL

Steven Bell

26 September 2023

Tufts UNIVERSITY

By the end of class today, you should be able to:

- Using a cheatsheet, write a complete VHDL entity
- Use VHDL to implement a boolean equation or truth table


Introduction to Digital Logic
Computer Programming
Verilog
VHDL
Fields of Enlightenment
Memory
Sequential Logic
Process Block Perils
Swamps of Arcane Syntax
State Machines
Flip-flop Pass
Computer Architecture
YOU ARE HERE
Plains of Combinational Logic
Desert of "How things used to be done"

From the reading check

When will we start using VDHL?

Will we get practice working with VHDL in lab and in class?

How do i avoid confusing VHDL syntax and C syntax?

Is it still advantageous to draw circuit diagrams and truth tables before using VHDL, or can most of this be done with the language?

lecture will solve all my problems

VHDL vs SystemVerilog

It's kinda like MATLAB vs Python/NumPy

We're using VHDL because other courses at Tufts use VHDL


Welcome to the swamp of arcane syntax.
Adobe Firefly (Beta)

VHDL doesn't act like C/C++

- Not case sensitive (`INPUT_A` is the same as `input_a`)
- Funky assignments (`y <= a or b`)
- Lots of semicolons!

VHDL is picky about types

```
1      integer
'1'    bit
"1"    array (vector) of bits

"0010" = 2
```

"01" = y

array of bits | integer

error!

Software languages
(compiled imperative languages)

Code specifies a sequence of operations to perform.

Compiler optimizes this and creates a binary.

Computer executes the binary, doing things in sequence.

Hardware description languages (on FPGA)

Code specifies behavior of a digital circuit.

Synthesis optimizes the design.

Place & route figures out how to fit it into the hardware.

Bitstream is loaded into FPGA, and the circuit does "everything at once."

VHDL is designed for multiple things

- Design hardware you'll implement with an FPGA or ASIC process
- Model hardware that you'll build with "real" components
- Test designs or models for the above two scenarios

Not everything in VHDL makes hardware

Stuff that describes hardware (synthesizable)

```
unsigned   8d"5"
y <= a and b;
```

```
wait 10 ns;        assert CONDITION        report "A is ..."
```

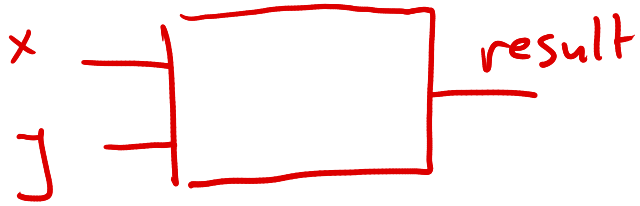
File I/O

Stuff you can use in simulation (i.e., all of VHDL)

VHDL designs are built with modules

Just like software programs are built with functions

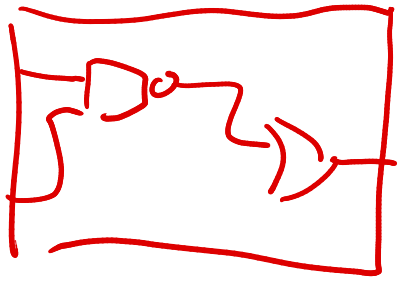
VHDL **entity**



C function **declaration**

```
double pow(double x, double y);
```

VHDL **architecture**



C function **definition**

Anatomy of a module

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

Required libraries

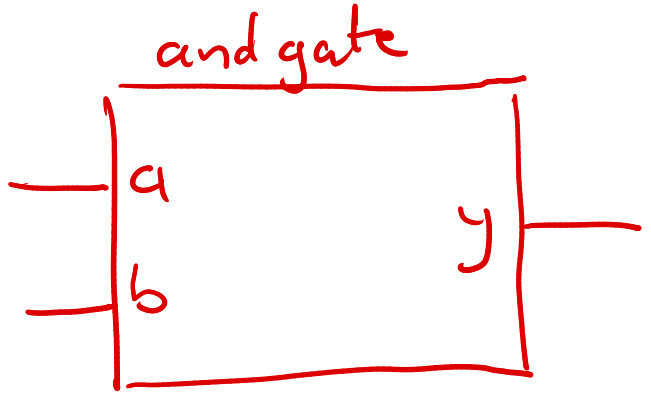
Anatomy of a module

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

entity andgate is
        port(
                a : in std_logic;
                b : in std_logic;
                y : out std_logic
        );
end;
```

Required libraries

Entity declaration



Anatomy of a module

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

Required libraries

```
entity andgate is
        port(
                a : in std_logic;
                b : in std_logic;
                y : out std_logic
        );
end;
```

Entity declaration

```
architecture synth of andgate is
begin
        y <= a and b;
end;    z <= a or b;
```

Implementation definition (architecture)

Show and tell time

How do I actually use VHDL to build something?

This is the focus of lab 4

Practice!

Go to vhdlweb.com

You should have received an email with your login info


Riva
Rivajunior

For Thursday

1. Read the book (**1.4**) and complete the pre-class quiz

2. Finish lab 3 (this week, report due next week)

3. Lab 4 has no prelab (yay!)