CET 486 – 586
Hardware Description Language: VHDL

Introduction to hardware description languages using VHDL. Techniques for modeling and simulating small digital systems using a VHDL simulator
Textbooks

- "The Designer’s Guide to VHDL" by Peter Ashenden, Morgan Kaufman. Reference text
- "VHDL for Designers" by Sjoholm and Lindh. Prentice Hall. Reference text
- "VHDL for Logic Synthesis. An introductory guide for achieving Design Requirements" by Andrew Rushton. McGraw Hill. Reference text
General Information

- Prerequisites
- Purposes
- Labs
- Exercises
- Quiz
- Tests
### Grading

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quizzes</td>
<td>10%</td>
</tr>
<tr>
<td>Mid Term Exam</td>
<td>20%</td>
</tr>
<tr>
<td>Labs</td>
<td>25%</td>
</tr>
<tr>
<td>Project</td>
<td>25%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>20%</td>
</tr>
</tbody>
</table>
Academic Integrity Policy

- AIP: http://www.asu.edu/studentlife/judicial/integrity.html
Contact Information

- Cristian Sisterna
- cristian.sisterna@asu.edu
- Website: www...
VHDL – Introduction

Chapter I
VHDL - Features

- Very High Speed IC Hardware Description Language
- The design is technologically independent
- Allow to design generic components
- Standard component already coded in VHDL
- Timing verification
- The designer concern is the functionality
- Portability: VHDL is an IEEE standard
- VHDL = Sequential Language + Concurrent Language + Net-List + Timing Constraints + Waveform Generation
VHDL – Features (cont’)

- Flexible design methodology: top-down, bottom-up
- It is not proprietary
- There is no limit for the design to be described in VHDL
- Allow description of delay time (minimum and maximum), hold time, setup time
- Very short development time
- Allow different levels of abstraction (Assembler, C, C++)
VHDL Flow Design

Specifications → VHDL Code → Compilation → Simulation & Verification

Synthesis & Optimization → Place & Route → Timing Verification

Front-end Tools

Back-end Tools

ECET – CET 486
VHDL – General View

Introduction to VHDL
A hardware abstraction of a Digital System is an entity.

Five VHDL design units describe an entity:
- Entity declaration
- Architecture body
- Configuration declaration
- Package declaration
- Package body
Entity declaration

- Describe the external view of an entity. The input and output signal names:

```
-- entity declaration syntax

entity <entity_name> is
  [generic]
  (list_of_generics_their_types_and_value);
  [port]
  (list_of_interface_port_names_mode_and_types);
  [entity_item_declaration]
  [begin]
    entity_statements
  [end]
end [entity] [entity_name];
```
Architecture body

- Contains the internal description of the entity

```
-- architecture body syntax
architecture <architecture_name> of <entity_name> is
begin
concurrent_statements;
end [architecture]
```
Example: Half-adder

C. Sisterna
Spring 2003

ECET – CET 486
--- Circuit: Half Adder
--- Objective: example
--- Ref: fig 2.3 "VHDL Primer"
---

entity half_adder is
  port (A: in bit;
        B: in bit;
        SUM: out bit;
        CARRY: out bit);
end half_adder;
STRUCTURAL

architecture HA_STRUCTURE of HALF_ADDER is
    component XOR2
        port (X, Y: in bit
                Z: out bit);
    end component;
    component AND2
        port (L, M: in bit;
              N: out bit);
    end component;
begin
    X1: XOR2 port map (A, B, SUM);
    A1: AND2 port map (A, B, CARRY);
end HA_STRUCTURE;
Half_adder: architecture body as a set of concurrent assignment statements

DATAFLOW

--
-- data flow style of modeling ======
--
architecture HA_DATA_FLOW of HALF_ADDER is
begin
    SUM <= A xor B;
    CARRY <= A and B;
end HA_CONCURRENT;
Half_adder: architecture body as a set of sequential assignment statements

BEHAVIORAL

architecture HA_BEHAVIORAL_2 of HALF_ADDER is
begin
process (A, B)
begin
if (A='0' and B='0') then
    SUM <= '0';
    CARRY <= '0';
elsif (A='1' and B='0' | A='0' and B='1') then
    SUM <= '1';
    CARRY <= '0';
else
    SUM <= '0';
    CARRY <= '1';
end if;
end process;
end HA_BEHAVIORAL_2;
Half_adder: architecture body as a mixed style

-- mixed style of modeling

architecture HA_BEHAVIORAL_2 of HALF_ADDER is

    signal SUM1: bit;

    component AND2
        port (L, M: in bit;
        N: out bit);

    end component;

begin

    X1: XOR2 port map (A, B, SUM1); -- structural

    process (A, B) -- behavior
    begin

        if (A='0' and B='0') then
            CARRY <= '0';
        elsif (A='1' and B='0') or (A='0' and B='1') then
            CARRY <= '0';
        else -- A='1' and B='1'
            CARRY <= '1';
        end if;

    end process;

    SUM <= SUM1; -- data flow

end HA_BEHAVIORAL_2;
Simulation

Stimulus

Device Under Test (DUT)

Results

Text

Waveform

Test Bench (VHDL Code)
Simulation

Waveform Stimulus

![Waveform Stimulus Graph](image-url)
Simulation

Test Bench

a <= '1', '0' after 10 ns,
    '1' after 39 ns, '0' after 110 ns,
    '1' after 145 ns, '0' after 155 ns;

b <= '0', '1' after 30 ns,
    '0' after 49 ns, '1' after 90 ns,
    '0' after 115 ns, '1' after 135 ns;

assert (a='1' and b='1' and sum='0')
    report "error when a=b='1' sum /= '0"
    severity note;