VHDL

VHSIC Hardware

Description

Language

VHSIC

Very High

Speed

Integrated

Circuits

©Copyright 1989 Alan R. Martello, Univ. of Pittsburgh

What is VHDL?

- a language for description
- a language for design, simulation and verification

©Copyright 1989 Alan R. Martello, Univ. of Pittsburgh

A VHDL design is a set of design entities, one of which is the "top level", which might invoke other design entities as components.

The entire design, with all nestings, is referred to as the "design hierarchy".

©Copyright 1989 Alan R. Martello, Univ. of Pittsburgh

Example:

```
1  PC System
   2  main CPU board
      3  external bus circuitry
         4  bus buffers
         4  bus control circuitry
      3  main processor
      3  local memory
   2  memory board
      3  bus interface
      3  memory control circuitry
   2  graphics adapter
      3  bus interface
      3  video control circuitry
      3  RGB generator
```

©Copyright 1989 Alan R. Martello, Univ. of Pittsburgh

entity – defines the inputs and outputs to a 'black box' which performs a specific function

Example:

```
entity ALU is
   port (arg1, arg2: in bitvec;
         add_or_sub: in bit;
         result: out bitvec);
end ALU;
```

©Copyright 1989 Alan R. Martello, Univ. of Pittsburgh

architecture – one possible realization (or implementation) of the "insides" of the "black box"

An architecture may contain:

- data declarations
- concurrent signal assigment
- process blocks
- component instantiations

©Copyright 1989 Alan R. Martello, Univ. of Pittsburgh

data declarations supported

- bit (e.g. A: bit;)
- bit_vector
  (e.g. B: bit_vector(31 downto 0))
- constants (e.g. "1", "110010")
- bitvec (e.g. B: bitvec;)
- integer (coerced to bitvec)

©Copyright 1989 Alan R. Martello, Univ. of Pittsburgh

concurrent signal assignment
(i.e. "standard gates")

```
NOT  (single input)
    a <= not(b);

AND  (multiple input)
    a <= b AND c;

OR   (multiple input)
    a <= c or d or e;

XOR  (multiple input)
    a <= (b xor c xor d) xor e;
```

©Copyright 1989 Alan R. Martello, Univ. of Pittsburgh

concurrent signal assignment (cont.)

```
NAND (two input)
    a <= (c NAND d) nand e;

NOR  (two input)
    a <= (c nor d) Nor e;

Bit-wise Concatenation ("&")
    a(6 downto 1) <=
      c & d(3 downto 0) & e;

Selected (multiplexer)
    a <= b when c else a;
```

©Copyright 1989 Alan R. Martello, Univ. of Pittsburgh

syntax notes

arbitrarily complex expressions supported with parentheses overriding any VHDL implicit precedence

e.g.

```
a <= ((b and c and f) or
        (t nor r)) nand p;
```

```
a(3 downto 0) <=
    b(4 downto 1) when (p and q)
    else (p & q & q & p);
```

comments supported via "—"

```
  e.g. -- this is a comment
```

all assignment statements must end with a semicolon (';')

©Copyright 1989 Alan R. Martello, Univ. of Pittsburgh

component instantiations

- conceptually similar to a function or procedure call, but actually results in the generation of hardware!
- "macro" capability
- instructs the system to "create" another occurrence of the specified "thing" (i.e. entity.architecture)

process blocks

- sequential programming language
- looping, conditional branching, local variables, etc.
- to VHDL, process blocks are arbitrarily large, user-defined, "big gates"

©Copyright 1989 Alan R. Martello, Univ. of Pittsburgh