EEM16 Discussion
Spring 2006

April 28, 2006
Heemin Park
hm park@ee.ucla.edu
Sample VHDL Code

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

-- entity
ENTITY And4 IS PORT (  
a, b, c, d: IN  std_logic;
z : OUT std_logic);
END And4;

-- architecture
ARCHITECTURE behavioral OF And4 IS
BEGIN
    PROCESS(a,b,c,d)
    BEGIN
        IF (a='1' and b='1' and c='1' and d='1') THEN z <= '1';
        ELSE z <= '0';
        END IF;
    END PROCESS;
END behavioral;
```
## 5-valued Logic

<table>
<thead>
<tr>
<th>Logic Value</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>Uninitialized</td>
</tr>
<tr>
<td>X</td>
<td>Unknown</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Z</td>
<td>High Impedance</td>
</tr>
</tbody>
</table>
Signals, Variables and Constants

Signal Assignment Statement

\[
x <= '1'; \quad \text{-- binary value 1 is assigned to } x
\]
\[
X <= u + y; \quad \text{-- value of (u+y) is assigned to } x
\]

Variable Assignment Statement

\[
a := x + y;
\]
Data Types

TYPE INTEGER IS RANGE -2,147,483,647 TO 2,147,483,647;
-- -(2^{31} - 1) to 2^{31} - 1

TYPE REAL IS RANGE -1.0E+38 TO 1.0E38;

TYPE BOOLEAN IS (FALSE, TRUE);

TYPE BIT IS ('0', '1');

TYPE CHARACTER IS (...,'A','B','C',...,'a','b','c',
...'0','1',...);

TYPE TIME IS RANGE 0 TO 2,147,483,647
UNITs ps; ns; us; sec; min; 
hr; END UNITs;
Data Types

TYPE STRING IS ARRAY(POSITIVE RANGE <>) OF CHARACTER;

TYPE BIT_VECTOR IS ARRAY(NATURAL RANGE <>) OF BIT;

-- User defined array types
TYPE BitVector4 IS ARRAY(3 DOWNTO 0) OF BIT;
TYPE BitVector8 IS ARRAY(7 DOWNTO 0) OF BIT;
TYPE Memory IS ARRAY(INTEGER RANGE <>) OF BitVector8;
Subtypes

SUBTYPE NATURAL IS INTEGER RANGE 0 TO 2,147,483,647;

SUBTYPE POSITIVE IS INTEGER RANGE 1 TO 2,147,483,647;

-- User defined subtypes

SUBTYPE ByteInt IS INTEGER RANGE 0 TO 255;

SUBTYPE LowerCase IS CHARACTER RANGE ‘a’ TO ‘z’;
Object Declarations

SIGNAL clk : BIT;
SIGNAL m, n : INTEGER;
SIGNAL x_in : BitVector4:=('0','1','0','0');
SIGNAL y_in : BitVector4:=('0101');
CONSTANT count_limit: INTEGER:=20;
VARIABLE count : INTEGER:=1;
VARIABLE z : BitVector8:=(1=>'1', 3=>'1', others=>'0');
Expressions

**SIGNAL**

\[ x, y, z : \text{INTEGER}; \]

**VARIABLE**

\[ a, b, c : \text{BitVector4}; \]

\[ x + y + z \quad -- \text{produces an INTEGER result} \]

\[ a \text{ OR } b \quad -- \text{produces a result of type BitVector4} \]

\[ \text{func}(x) \quad -- \text{produces a results of func}(x) \]

**VARIABLE**

\[ a : \text{BitVector4} := (\text{'0'},\text{'1'},\text{'1'},\text{'1'}); \]

**VARIABLE**

\[ b : \text{BitVector4} := (\text{'1'},\text{'1'},\text{'0'},\text{'0'}); \]

**VARIABLE**

\[ c : \text{BitVector4}; \]

\[ c := a \text{ AND } b; \quad -- \text{produces } c := (\text{'0'},\text{'1'},\text{'0'},\text{'0'}); \]
Control Flow Statements

- **IF Statement**
  
  ```
  IF (clk='1') AND (x='1') THEN
      z := a OR b;
  ELSIF (clk='0') THEN
      z := '0';
  END IF;
  ```

- **CASE Statement**
  
  ```
  CASE Data IS
      WHEN "000" => k:=1;
      WHEN "111" => k:=0;
      WHEN others => k:=k+1;
  END CASE;
  ```
Control Flow Statements

- **FOR LOOP**
  
  ```plaintext
  FOR i IN 1 TO 20 LOOP
      count := count + a(i);
  END LOOP;
  
  FOR i IN 20 DOWNTO 1 LOOP
      count := count + a(i);
  END LOOP;
  ```

- **WHILE LOOP**
  
  ```plaintext
  WHILE (a = b) LOOP
      i := i+1;
      a := x(i);
      b := y(i);
  END LOOP;
  ```
Sample VHDL Code

library IEEE;
use IEEE.std_logic_1164.all;

-- entity
ENTITY And4 IS PORT (
a, b, c, d: IN  std_logic;
z : OUT std_logic);
END And4;

-- architecture
ARCHITECTURE behavioral OF And4 IS
BEGIN
  PROCESS (a, b, c, d)
  BEGIN
    IF (a='1' and b='1' and c='1' and d='1') THEN z <= '1';
    ELSE z <= '0';
    END IF;
  END PROCESS;
END behavioral;
Example: Radix-4 Digit-comparator

- **Input**
  - \( x, y \in \{0, 1, 2, 3\} \)

- **Output**
  - \( z \in \{G, E, S\} \)

- **Function**
  - \( z = G \) if \( x > y \)
  - \( z = E \) if \( x = y \)
  - \( z = S \) if \( x < y \)
Packages

PACKAGE BitDefs_pkg IS
  SUBTYPE BitVector IS BIT_VECTOR;
  SUBTYPE BitVector2 IS BIT_VECTOR(1 DOWNTO 0);
  SUBTYPE BitVector3 IS BIT_VECTOR(2 DOWNTO 0);

  SUBTYPE Radix4 IS INTEGER RANGE 0 TO 3;
  TYPE MagnComp IS (G, E, S);
END BitDefs_pkg;

PACKAGE BODY BitDefs_pkg IS
END BitDefs_pkg
Example: Entity Body – High Level

USE WORK.BitDefs_pkg.ALL; -- package contains
-- definition of
-- Radix4 and MagnComp

ENTITY radix4_comp IS
  PORT (x,y : IN Radix4; -- inputs
       z : OUT MagnComp); -- outputs
END radix4_comp;
Example: Architecture Body – High Level

ARCHITECTURE behavioral OF radix4_comp IS
BEGIN
  PROCESS(x, y)
  BEGIN
    IF (x > y) THEN z <= G;
    ELSIF (x = y) THEN z <= E;
    ELSE z <= S;
  END IF;
  END PROCESS;
END behavioral;
Example: Entity Body – Binary

USE WORK.BitDefs_pkg.ALL; -- package contains
   -- definition of
   -- BitVector2 and
   -- BitVector3

ENTITY binradix4_comp IS
   PORT (x,y : IN BitVector2; -- inputs
         z : OUT BitVector3);  -- outputs
END binradix4_comp;
Example: Architecture Body – Binary

ARCHITECTURE behavioral OF binradix4_comp IS
BEGIN
    PROCESS(x, y)
    BEGIN
        IF (x > y) THEN z <= "100";
        ELSIF (x = y) THEN z <= "010";
        ELSE z <= "001";
        END IF;
    END PROCESS;
END behavioral;