GONZAGA UNIVERSITY

VHDL: Data Types

GONZAGA UNIVERSITY

Predefined Data Types

- package **standard** of library **std**
  - BIT (and BIT_VECTOR) ← **AVOID**
  - BOOLEAN (true, false) ← **NOT RECOMMENDED in SYNTHESIZABLE CODE**
  - INTEGER (32-bit integers: from $-2,147,483,647$ to $+2,147,483,647=2^{31}-1$)
  - NATURAL (non negative integer: from 0 to $+2^{31}-1$)
  - REAL (floating point number ranging from $-1.0x10^{38}$ to $+1.0x10^{38}$)
  - TIME (fs, ps, ns, us, ms, sec, min, hr)
  - CHARACTER (example: `'j'`)
  - STRING (example: `"john"`)

NOT SYNTHESIZABLE

GONZAGA UNIVERSITY

Predefined Data Types

- package **std_logic_1164** of library **ieee**
  - std_logic and std_logic_vector
  - std_ulogic and std_ulogic_vector

('X', '0', '1', 'Z', 'W', 'L', 'H', '–')

('U', 'X', '0', '1', 'Z', 'W', 'L', 'H', '–').

Examples:

```
signal a: std_logic;
a <= '1';
signal b: std_logic_vector(7 downto 0);
signal c: std_logic_vector(0 to 7);
signal d: std_logic_vector(3 downto 0);
b <= "10000010";
c <= "01000001";
d <= (others => '0'); -- d <= "0000"
d <= b(4 to 7);       -- d <= "0001"
d(1) <= c(0);         -- d <= "0011"
d(3 downto 2) <= b(7 downto 6); -- d <= "1011"
a <= b(6); -- a <= '0'
b(6) <= '1'; -- b(6) <= "1" is wrong !
```

GONZAGA UNIVERSITY

Predefined Data Types

- package **std_logic_arith** of library **ieee**
  - unsigned
  - signed

```
Example:
signal a: unsigned(7 downto 0);
a <= “11100011”;
```

GONZAGA UNIVERSITY

Predefined Data Types

- package **numeric_std** of library **ieee**
  - unsigned
  - signed

GONZAGA UNIVERSITY

User Defined Data Types

```
type integer is range -2147483647 to +2147483647;
type state_t is (zero, pedge, nedge, one);
type logic_t is ('0','1','Z'); -- upper case Z
type bit_vector is array(natural range<>) of bit;
```

The range is unconstrained except it must fall in the NATURAL range

GONZAGA UNIVERSITY

Subtypes

- operations between different types are not allowed (but … they are allowed between subtypes)
  - `subtype logic_t is std_logic range '0' to 'Z';`
  - `subtype small_integer_t is integer range -32 to 32;`

```
signal a: bit;
signal b: std_logic;
signal c: logic_t;

b <= a; -- illegal
b <= c; -- legal
b <= 'X'; -- legal: is std_logic
c <= b; -- illegal
```

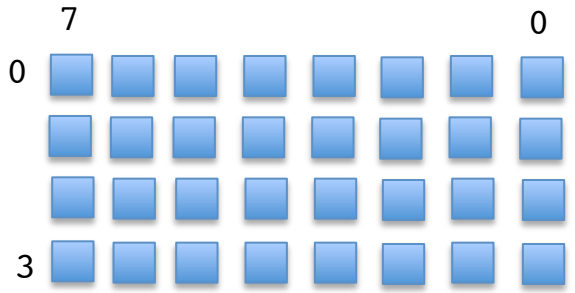
Array

```
type byte_t is array (7 downto 0) of std_logic; -- 1D array
type matrix1D is array (0 to 3) of byte_t;      -- 1Dx1D array
type matrix2D is array (0 to 3, 7 downto 0) of std_logic; -- 2D array
type mem_t is array (0 to 3) of std_logic_vector (7 downto 0) -- 1Dx1D array

signal x: byte_t;
signal y: std_logic_vector(7 downto 0); -- same as byte_t
signal w1: matrix1D;
signal w2: matrix2D;
signal w3: mem_t; -- same as matrix1D

x <= "1111110";
x <= ('1','1','1','1','1','1','0');
Y <= "1111110";
y <= ('1','1','1','1','1','1','0');
y <= (7 => '0', 1 => '0', others => '1'); -- "01111101"
x(2) <= w1(2,1) -- one element
x <= y;         -- one row
w1(1)(1) = w2(3,1);
w3 <= w1;
w2(1,1) <= x(7);
w2 <= ((OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'),(OTHERS=>'0'));
w3(2)    <= y;
w1(1)(1) <= w2(3,1);
-- data independent initialization
FOR i IN 0 TO 3 LOOP
   FOR j IN 7 DOWNTO 0 LOOP
       x(j)     <= '0';
       y(j)     <= '0';
       w1(i)(j) <= '0';
       w2(i,j)  <= '0';
       w3(i)(j) <= '0';
    END LOOP;
END LOOP;
```

1D array



1Dx1D array

2D array

GONZAGA UNIVERSITY

Records

```
TYPE birthday IS RECORD
    day: INTEGER RANGE 1 TO 31;
    month: month_name;
END RECORD;
```