

# Short note on VHDL Programming

Andersen Ang

First created: 2013. Last update: 2017-Feb-1

## 1 Sample Program

The VHDL program consist of 2 part

1. Input output port
2. Behav

### 1.1 Modulization in VHDL

### 1.2 Block

```
<Optional_Block_Name> Block
BEGIN
:
END Block <Optional_Block_Name> ;
```

#### 1.2.1 Block Properties

Concurrency : The statements inside Block is concurrent statement, they occur at the same time. They are not carried out sequentially !

```
ENTITY example IS
PORT ( a : IN Integer ;
      b : OUT Integer );
END example
Architecture action of example IS
BEGIN
Block
  Begin
    a <= a + 1 ;
    b <= a ;
  END Block;
END action;
```

#### Guarded Block

Because of the concurrency of block, sometimes it will be useful for block being executed after some condition is being fulfilled, this is guarded block

```

ENTITY example IS
PORT ( a, control : IN Integer ;
      b : OUT Integer );
END example
Architecture action of example IS
BEGIN
Block ( control = 1 )
  Begin
    a <= a + 1 ;
    b <= a ;
  END Block;
END action;

```

### 1.3 Process

```

<Optional_Process_Name> Process ( signal parameter list )
Begin
:
END Process;

```

- Process is sequential
- Variable can only defined inside process
- Process will be sensitive to the signal parameter list
- When there is change in the signal parameter, the process will stop, and start again from the first statement
- i.e. Process will run again if parameter changed
- When there are multiple process, they will run concurrently

### 1.4 Subprogram

Finally, the concept of “reuse” is also in VHDL

#### 1.4.1 Function

```

Function <Function_Name> ( parameter list )
  Return Data_Type IS
[ Statements ];
BEGIN

```

## 2 Data Type

### 2.1 Integer

- Range :  $-(2^{31} - 1)$  to  $(2^{31} - 1)$

## 2.2 Real Number

- Range :  $- 1.0 \cdot 10^{38}$  to  $+1.0 \cdot 10^{38}$
- Note :  $1 \neq 1.0$  ( Different type ! )

## 2.3 Bit

- '1' or '0'
- Single quotation mark is necessary for a bit , but no need for integer
- When only ONE bit, use single quotation mark '1' , '0'

## 2.4 Bit Vector

- A group of bit
- When multiple bit (bit vector), use double quotation mark "1001" "110"

## 2.5 Boolean

- TRUE or FALSE
- Boolean  $\neq$  Bit : No numerical value, can not do numerical computation
- Boolean is usually used in conditional statement
- By default, all boolean data are defined to be FALSE if not initialed.

## 2.6 Character

- Single character use single quotation
- Range : a to z , 0 to 9 , ! to ) , all the ASCII code
- VHDL is not case sensitive , but remember 'A'  $\neq$  'a' , '1<sub>Char</sub>'  $\neq$  '1<sub>bit</sub>'  $\neq$  '1<sub>Int</sub>'  $\neq$  '1.0<sub>Real</sub>'

## 2.7 String

- Character string encapsulated by double quotation mark
- Example : "This is a character string 01234"
- String usually is used in giving instruction

## 2.8 Time

- A complete time information include 2 parts : Integer value + time unit
- Example : 500 ms
- The unit : fs, ps, ns, us, ms, sec, min, hr
- Time information is usually used in *signal* to simulate real time processing.

## 2.9 Severity Level

- NOTE , WARNING, ERROR, FAILURE
- Used to indicate the state of the system.

## 2.10 Vector Definition

- Downto : In downto, it means most significant bit is the first bit
- Example : BIT\_VECTOR(3 downto 0)
- To : In to, it means the first bit is least significant bit

## 3 User Defined Type

- Syntax : TYPE <TYPE\_NAME> Data\_Type;
- Data\_Type can be : Enumerated , Integer, Real, Array, File, Record, Time

### 3.1 Enumerated

- Sometimes it is more useful to use word to enumerate some information
- Syntax : TYPE <TYPE\_Name> IS ( element1, element2 , ... );
- Example , compare the following
- TYPE week IS( mon, tue, wed, thur, fri, sat, sun );
- TYPE week IS( 001, 010, 011, 100, 101, 110, 111);
- Obviously the first one using word is more readable.
- Note, when using IEEE library, the STD\_LOGIC is actually an enumerated type
- TYPE STD\_LOGIC IS ('U' , 'X' , '0' , '1' , 'Z' , 'W' , 'L' , 'H' , '-');

### 3.2 Integer, Real

- Syntax : TYPE <Type\_Name> IS Data\_Type Range ;
- TYPE digitCount IS INTEGER RANGE 0 to 9;

### 3.3 Array

- Syntax : TYPE <Type\_Name> IS ARRAY Range OF Data\_Type;
- Example
- TYPE Byte IS ARRAY (7 downto 0 ) OF bit;

### 3.4 Record ( Composite Type )

- TYPE <TYPE\_NAME> IS Record
- <sub\_type\_name> : Data\_Type ;
- <sub\_type\_name> : Data\_Type ;
- ...
- END Record;

–END–