Digital System Design

Mathias Balling & Mads Thede

Last updated: April 10, 2024

Contents

1 VHDL

Very High Speed Integrated Circuit Hardware Description Language. It is a programming language that has been designed and optimized for describing the behavior of digital systems.

1.1 VHDL Syntax

VHDL is not case sensitive, but it is a good practice to use upper case for keywords and lower case for everything else. VHDL describes hardware and so instructions are executed in a concurrent manner, meaning that all instructions are executed at once.

Comments are written with two hyphens (- -).

if, case and loop statements

Every if statement has a corresponding then component and each if statement is terminated with an end if; If an else if statement is needed then it can be written as elsif. Before each if statement a process statement must be written. The process statement includes the sensitivity list which is a list of signals that the process is sensitive to.

```
process (sel, I)
begin
  if (sel == "00") then
    a <= 'I';
    b <= '0';
    c <= '0';
    d <= '0';

  elsif (sel == "01") then
    a <= '0';
    b <= 'I';
    c <= '0';
    d <= '0';
  elsif (sel == "10") then
    a <= '0';
    b <= '0';
    c <= 'I';
    d <= '0';
  else
    a <= '0';
    b <= '0';
    c <= '0';
    d <= 'I';
  end if;
end process;
```

Each case statement is terminated with end case; Each loop statement has a corresponding end loop;

```
case (expression) is
  when choices =>
    <sequential statements>
  when choices =>
    <sequential statements>
  when others =>
    <sequential statements>
end case;
```

Signal and variable assignments

Among the most frequently used object types there is the signal object type, the variable object type, and the constant object type. The signal type is the software representation of a wire in the digital system. The variable type is used to store local information. The constant type is used to store constant information.

Signals are declared at the top of the architecture body, just before the keyword begin. Variables must be declared inside the process construct and are local.

Assigning a value to a signal is done using the signal assignment operator <=.

$$C <= not(A);$$

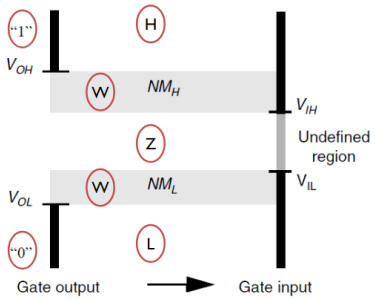
Assigning a value to a variable is done using the variable assignment operator :=.

$$B := A;$$

A variable changes its value soon after the variable assignment is executed. Instead a signal changes its value "some time" after the signal assignment expression is evaluated.

1.2 VHDL Design Units

STD LOGIC



Character	Value
'U'	uninitialized
'X'	strong drive, unknown logic value
'0'	strong drive, logic zero
'1'	strong drive, logic one
'Z'	high impedance
'W'	weak drive, unknown logic value
'L'	weak drive, logic zero
'H'	weak drive, logic one
'-'	don't care

Entity

The entity describes how the unit interfaces with the outside world. The entity lists the various inputs and outputs of the underlying system.

Architecture

Describes what the circuit actually does. It describes the internal implementation of the associated entity. An architecture can be written by means of three modeling techniques plus any combination of these three: dataflow, behavioral, and structura plus any combination of these three.

Karnaugh map (K-map)

A method of circuit optimization. Aims to reduce logic functions more quickly and easily compared to Boolean algebra.

Combinational circuits

It is defined as the time independent circuits which do not depend upon previous inputs to generate any output. e.g. Encoder, Decoder, Multiplexer, Demultiplexer etc.

Sequential circuits

Dependent on clock cycles and depends on present as well as past inputs to generate any output. e.g. Flip-flops, Counters, Registers etc.

Concurrent Statements CHDL has the ability to execute a virtually unlimited number of statements at the same time and in a concurrent manner. The key thing to remember is that we are designing hardware.

Conditional Signal Assignment when

Conditional signal assignment is a concurrent statement that assigns a value to a signal based on the value of a condition. The syntax is:

```
<target> <= <expression> when <condition> else
        <expression> when <condition> else
        <expression>;
```

Selected Signal Assignment with select

Selected signal assignments only have one assignment operator. Selected signal assignment differs from conditional assignment statements in that assignments are based upon the evaluation of one expression. The syntax is:

```
with <choose_expression> select
  <target> <= <expression> when <choices>,
              <expression> when <choices>
              <expression> when others;
```

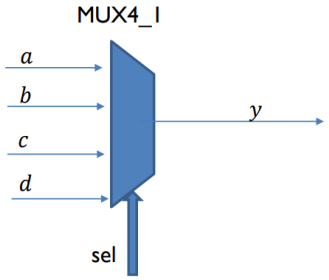
The general form of the selected signal assignment statement is similar to switch statements as seen in algorithmic programming languages such as C.

Process Statements

The process statement is a stement which contains a certain number of instructions that, when the process statement is executed, are executed sequentially. In other words the process statement is a tool that can be used for executing a certain number of instructions in a sequential manner.

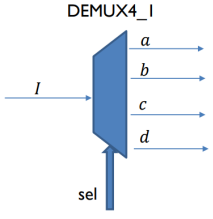
Multiplexer

A general multiplexer has n data inputs, one output, and m select inputs. The number of select inputs determines the number of data inputs. The number of select inputs determines the number of data inputs. The number of data inputs is 2 to the power of the number of select inputs.



Demultiplexer

It selects one ouput from the multiple output line and feteches the single input through the selection line.



Encoders

Encoders are used to encode data into a coded form and decoders are used to convert it back into its original form. An encoder that has 2^n input lines and n output lines is called an $n - to - 2^n$ encoder. It is assumed that only one input has a value of 1 at any given time. It is used to minimize the number of input lines.

Priority encders are a modified version of an encoder. It gives a priority to an input signal and provides an output based on that priority.

Example of a 4-to-2 priority encoder:

```
entity Priority_Encoder is
  port (D: in std_logic_vector(3 downto 0);
        Y: out std_logic_vector(1 downto 0));
end Priority_Encoder;

architecture Behavioral of Priority_Encoder is

begin
  process (D)
  begin
    if (D(3) = '1') then
      Y <= "11";
    elsif (D(2) = '1') then
      Y <= "10";
    elsif (D(1) = '1') then
      Y <= "01";
    elsif (D(0) = '1') then
      Y <= "00";
    else
      Y <= "---";
    end if;
  end process;
end Behavioral;
```

Decoders

Decoders are used to decode data that has been encoded using a binary encoder. An n-bit code can represent 2^n different values. An encoder can be used with enable inputs. The enable inputs are used to enable or disable the decoder.

Latches

Level sensitive memory cell that is transparent to signals passing from the D input to Q output when enabled. It holds the value of D on Q at the time when it becomes disabled.

Latches with clear input. The clear input means that when the clear signal goes to 1 the output is set to 0.

D-type Flip-flops

Is an edge triggered memory device that transfers a signal's value on its D input to its Q output, when an active edge transition occurs on its clock input. The output value is held until the next active edge transition occurs.

Rising edge is preferred. Using an implemented function called rising_edge(clk) is more robust to noise compared to using clk'event and clk = '1'.

If you want to use flip-flops with reset it is good practice to use asynchronous reset.

Registers

A parallel register is simply a bank of D-type flip-flops. Its implementation has the same structure as for one flip-flop but with extended inputs and outputs.

1.3 Standard models in VHDL Architecture

Data-flow style architecture

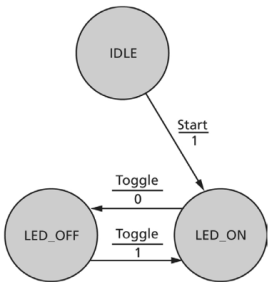
Circuits are described by showing the input and ouput relationships between the various built-in components of the VHDL language.

Behavioral style architecture

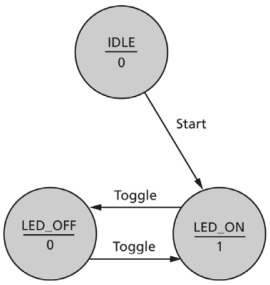
It provides no details as to how the design is implemented in actual hardware. The behavioral style models how the circuit outputs will react to the circuit inputs. Whereas in data-flow modeling you somewhat need to have a feel for the underlying logic in the circuit. In other words, data-flow modeling describes how the circuit should look in terms of logic gates whereas behavioral modeling describes how the circuit should behave.

2 Finite State Machines

FSM consists of states and transitions. State machiens are used for modeling systems that have a finite number of states and whose behavior is determined by the current state and the input. The states are drawn as circles and the transitions are drawn as arrows between the states. The transitions are labeled with the input that causes the transition. State machines can be made using two models: Mealy and Moore. In Mealy state machines, the output is a function of the current state and the input. When the input change, the outputs are updated without waiting for a clock edge. The mealy state machine is represented in the following figure:



In Moore state the output is a function of the current state only. The outputs are only written when the state changes (on clock edge). The moore state machine is shown on the following figure:



The hardware representation of a state machine can be visualized as: