Digital Design Principles

Counters

Name: Rajkaran Singh Grewal

Student No: 8882386

Email: rgrewal2386@conestogac.on.ca

Objective:

- Design some simple counters.

- Learn to use Asynchronous Reset in VHDL

VHDL Introduction:

To make the program in VHDL we will need to initialize the library and entity. Within the entity we will have the following inputs, we will have a standard logic with clock, a timer as a standard logic, a Switch to switch for the Johnson ring counter using a standard logic, a switch for to use the clock or timer using a standard logic. For the output we will use 4 standard logic vector of total length of seven units which will be used for the Seven Segment Display.

Within the Architecture of the program we will define a signal to store the clock load Value which is a standard logic vector of 4 units. We will also have a timer load value which is a standard logic vector which stores 4 units, a load value which is a standard logic vector which stores 4 units, we will also initialize counter which also a standard logic vector which stores 27 units. We will also have standard logic variable to store clock out. Then we will initialize a function to change the state of the ring. We will also initialize a function to get the seen segment value according to the value given in the function. We than will initialize the first process which is checking on the clock which is the push button. We will write an if statement where the rising edge of the clock will be checked if its true and within the block. We will check the if statement where we will check if the clock load value has the value 0000 and we the switch for Johnson is not turned on, then we will change the clock load value to 1000. Than after this block we will have another if statement where we will check if the switch for timer is not turned on. And than we will assign the clock load value to the return value of function where we assign the pattern according to the pattern sent. And we are done that process.

In the next process we will check for the changes in value of timer, we will check if the timer is in rising edge, and then we will increment the counter. We will check if the counter will equal to the decimal value of 49,999,999 and if that is true we will reset the counter back to zero, and assign clock out to zero. Else if we will check if the counter is equal to the decimal value of 49,999,998 and if true we will assign clock out to one. Then we will check if the switch for timer is turned on and if true we will check if the clock out is one if that is true we will check if timer load value is equal to 0000 and the switch for Johnson is not turned on we will set the value of timer load value to 1000. After wards we will set the value of timer load value to the value returned from the function where the pattern is set by supplying the value of the current load Value. And we end that process.

With the final function we will check for the changes in the value of clock load value or timer load value and set the value of load value to either clock load value or timer load value if the switch for timer is turned off or on respectively. Then we will set the value of the seven segment display from the return value of the function which is written to set the seven segment display.

VHDL program screenshot:Text

Description automatically generatedGraphical user interface, text

Description automatically generatedText

Description automatically generated

Text

Description automatically generated Text

Description automatically generated

VHDL compilation report: Graphical user interface, text, application, email

Description automatically generated

Pin Planner screenshot:

Table

Description automatically generated

Table

Description automatically generated

Verilog introduction:

For the program in Verilog, we will need to initialize the module with the parameters having the following inputs, first we will have a clock, timer, switch for Johnson, switch for timer. For the output we will have seven element arrays for the seven-segment display. Within the module we will have the array of four element clock load value, we will have the array of four element for the timer load value, and we will have an array of four element that stores the load value. We will have array of 26 element of counter. We will also initialize a variable clock out.

Then we will initialize always where the positive edge of the clock is checked. And we will check if the clock load value is equal to 0000 and the switch for johnson is not turned on. And if that is true than we will set the clock load value to 1000. Afterwards we will assign the value of clock load value to the return value of the function which sets the next value of the current value of clock load value.

In the next process where we will check of the positive edge of the timer. We will increment the counter by one. And if the counter value equals the decimal value of 49,999,999 we will set the value of counter to zero and clock out to the value of zero. Else if we will check if the counter will equal to 49,999,998 we will set the clock out value to 1. Than we will check of switch for timer is turned on we will than check if the vale of clock out is true , we will finally check if the value of timer load value is equal to 0000, and if the switch for johnson is turned off. And if its true we will assign the value of timer load value to equal 1000. And afterwards we will assign the timer load value to the return of the function which is returned to the next value.

In the final process we will check for the changes in either value clock load value or timer load value and we will set the load value to either timer load value or clock load value if the switch for the timer is turned on. And than we will set the seven segment display to the value of the return value of function of the display seven segment which sets the pattern for the character provided. Than we will write the function afterwards.

Verilog program screenshot: Text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedBackground pattern

Description automatically generated

Verilog compilation report: Graphical user interface, text, application, email

Description automatically generated

Verilog pin planner screenshot: Table

Description automatically generatedTable

Description automatically generated

Conclusion:

Thus we can conclude that by both the programming languages we can change the pattern of the ring counter or the johnson ring counter either using the inbuilt clock or a push button by a switch case that we inbuilt into a function to get the next pattern symbol. The program can be consider a state machine where by the input of the clock or the push button can change the state to the next state.