Lab 5

Verilog Basics

Continuous Assignment

Learning objectives of this lab,

* Continuous Assignment
* Pre-Lab
  1. Code Structure(Continuous Assignment)
     1. Overview

Continuous assignments provide a way of modeling combinational logic at a higher level of abstraction than Gate-Level logic. It allows the use of Boolean logic rather than gate connections. The left-hand side of an assignment is a variable to which the right-side value is to be assigned and must be a net. The right-hand side of an assignment, separated from the left-hand side by the equal (=) character, can be a net, a reg or any expression that evaluates a value including function calls. Continuous assignments are always active and assignments occur whenever the right-hand side operands changes. It drives both vector and scalar.

* + 1. Operators

|  |  |
| --- | --- |
| **Function** | **Operator** |
| NOT | ~ |
| AND | & |
| OR | | |
| XOR | ^ |
| XNOR | ~^ or ^~ |

Table 6.1 (Bitwise operators)

The code example explains these.

|  |  |
| --- | --- |
| **Function** | **Operator** |
| Left Shift | << |
| Right Shift | >> |

Table 6.2 (Bit Shifting)

a>>2 will right shift ‘a’ by 2 bits.

|  |  |
| --- | --- |
| **Function** | **Operator** |
| Concatenation | { } |

Table 6.3 (Concatenation)

Concatenation is used to merge multiple variables. For example if ‘a’ is 8-bit variable ‘b’ is 16-bit variable {a , b[3:0], 4’b1001’} results in 16-bit value.

|  |  |
| --- | --- |
| **Function** | **Operator** |
| Conditional operation | **? :** |

Table 6.4 (Conditional Execution)

Conditional operator is used to implement a circuit using if-else like structure

* + 1. Example

The code for Fig 6.1 using continuous assignment is given below.

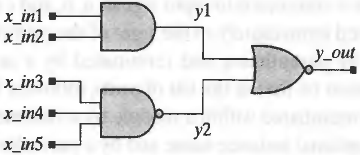


Fig 6.1(five-input and-or-invert (AOI) circuit)

This circuit can be implemented using a single assign statement

module AOI5\_CA0 (y\_out, x\_in);

input [4:0] x\_in;

output y\_out;

assign y\_out = ~((x\_in[0] & x\_in[1])|(x\_in[2]& x\_in[3]& x\_in[4]));

endmodule

~ means inversion (Not gate), | is Or, and & is And gate.

The module AOI5\_CA0 can also be written using multiple continuous assignments

module AOI5\_CA3 (y\_out, x\_in1, x\_in2, xin\_3, xin\_4, xin\_5);

input x\_in1, x\_in2, x\_in3, x\_in4,x\_in5;

output y\_out;

wire y1 = x\_in[0] & x\_in[1] & x\_in[4]; // Bitwise and operation

wire y2 = xin\_[2] & xin\_[3];

assign y\_out = ~(y1 | y2); // Complement the result of bitwise OR //operation

endmodule

* + 1. Conditional operator

Conditional operator can be used to implement enables

The five-input AOI circuit can be modified to have an additional input, enable, and to have a three-state output, as described by AOC5\_CAI below.

module AOC5\_CA1 (y\_out, x\_in1, x\_in2, x\_in3, x\_in4, x\_in5, enable);

input x\_in1, x\_in2, x\_in3, x\_in4, x\_in5, enable;

output y\_out;

assign y\_out = enable? ~((x\_in1 & x\_in2)|(x\_in3 & x\_in4 & x\_in5)): 1'bz;

endmodule

The conditional operator **(? :**) acts like a software if-then-else switch that selects between two expressions. In the above example, the conditional execution stement means,

if enable == 1

y\_out=~((x\_in1 & x\_in2)|(x\_in3 & x\_in4 & x\_in5))

else

y\_out= 1’bz;

Example code given below implements a 32-bit 4-1 mux using conditional operator.

Input [31:0] input0, input1, input2, input3;

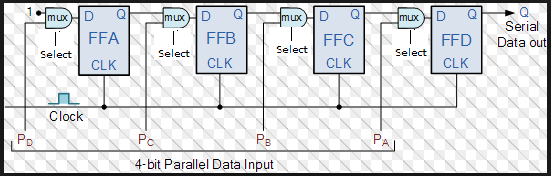
output [31:0] muxout;  
input [1:0] select;

assign muxout = (select == 2'b00) ? input0:  
  ( (select == 2'b01) ? input1 :  
  ( (select == 2'b10) ? input2 :  
  ( (select == 2'b11) ? input3 : 32’bz )))

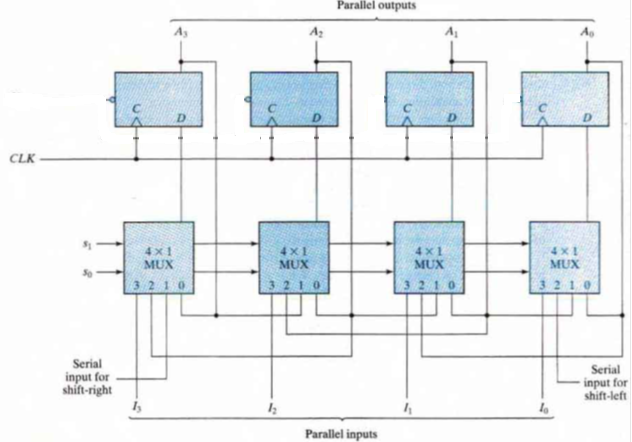
* In-Lab

**Task (Use Vectors for declaring variables)**

* Write Verilog HDL description of previous lab tasks using continuous assignment.
* Write the Verilog HDL description of following shift register. Write a module 2-1 mux using continuous assignment, then instantiate D Flip-Flop and 2-1 mux multiple times in a module to describe the given circuit



* Post-Lab
* Write Verilog HDL description of Universal shift register. Waveform should display all inputs and contents of register for different select options

****

* **Your lab report, a .doc file, should contain properly commented Post-Lab task code, with Screenshots(of print preview) of Schematic and waveforms, and Critical Analysis.**
* **The report must have a title page in the pescribed format.**
* **Name the .doc file RegNo.docx; eg SP14-BCE-99.docx**
* **Send it to** [**khiyamiftikhar@gmail.com**](mailto:khiyamiftikhar@gmail.com)**. With subject Lab No.**