Comp E 475

Microprocessors

Assignment 4 : instr Data

Student: Giorgi Tchkoidze

Red ID #: 822089301

Contents

Task Description

For this assignment we have to make ARM instruction decoder in verilog using following logic:

- you have inputs**:**
  - **instruction**, 32 bits
  - **clk**
- outputs:
  - **instr_type**, 2 bits, values from 0 to 3
    - 1 if given 32 bit instruction is Data Processing instruction
    - 2 if it's memory type instruction
    - 3 if it's branch instruction
    - 0 if not identifiable
  - **data_instr_type**, 3 bits, values from 0 to 4
    - 1 if given Data Processing instruction is "Immediate" type
    - 2 if it's "Register shifted by value" type
    - 3 if it's "Register shifted by register" type
    - 4 if it's "Multiplication" type
    - 0 if not identifiable

Solution

For this assignment I used combination of cases and if statemets.

To find the data_insts_type I used if statements:

```
data_instr_type = instruction[25] ? 3'b001 :
        !instruction[25] && !instruction[4] ? 3'b010 :
        instruction[25] && instruction[7] && instruction [4] ? 3'b011 :
        !instruction[25] && !instruction[24] && !instruction[7] &&
        instruction[6] && instruction[5] && !instruction[4] ? 3'b100 : 3'b000;
```

To find instr_type I used case statement:

```
always @(*) begin
    case(instruction[27:26])
            00: begin
                    instr_type = 2'b01; //Data Processing
                    data_instr_type = instruction[25] ? 3'b001 :
                            !instruction[25] && !instruction[4] ? 3'b010 :
                            instruction[25] && instruction[7] && instruction [4] ? 3'b011 :
                            !instruction[25] && !instruction[24] && !instruction[7] &&
                            instruction[6] && instruction[5] && !instruction[4] ? 3'b100 : 3'b000;
            end

            01: instr_type=2'b10; //Memory type

            10: instr_type=2'b11; //branch

            default: instr_type=2'b00; // not indentifiable
    endcase
```

Simulation & Verification

decoder Project Status			
Project File:	Controler.xise	**Parser Errors:**	No Errors
Module Name:	decoder	**Implementation State:**	Programming File Generated
Target Device:	xc3s100e-5cp132	• **Errors:**	No Errors
Product Version:	ISE 14.7	• **Warnings:**	9 Warnings (9 new)
Design Goal:	Balanced	• **Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• **Timing Constraints:**	All Constraints Met
Environment:	System Settings	• **Final Timing Score:**	0 (Timing Report)

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	4	1,920	1%	
Number of occupied Slices	3	960	1%	
Number of Slices containing only related logic	3	3	100%	
Number of Slices containing unrelated logic	0	3	0%	
Total Number of 4 input LUTs	4	1,920	1%	
Number of bonded IOBs	9	83	10%	
IOB Latches	2			
Average Fanout of Non-Clock Nets	1.50			

There are no errors but 9 warning which can not avoided since we are not using many defined bits from inputs.

Comparison

To complete this assignment we can use many approach. But I think, combination of cases and if statements is optimal.

Conclusion

In this assignment I learned how to use git and github. Also I got some knowledge how ARM instruction decoder works, and can be implemented using verily.