Comp E 475

Microprocessors

Lab 5 : instr Mem, Jmp

Student: Giorgi Tchkoidze

Red ID #: 822089301

Contents

Task Description

In this assignment, we have to modify existing decoder and add two output "mem_instr_type" and "jmp_instr_type" using following logic:

- **mem_instr_type**, 2 bits, values from 0 to 2
  - 1 if given Memory instruction type is of Immediate
  - 2 if it's "Register shifted by value" type
  - 0 if it's not identifiable
- **jmp_instr_type**, 2 bits , values from 0 to 2
  - 1 if given Jump instruction type is of Branch only
  - 2 if it's Branch and Link
  - 0 if it's not identifiable

Solution

Since we already have main part of the decoder from previous homework, we can use already defined logic and implement new in appropriate places.

For jmp_instr_type we can add following logic in the branch case (instr_type=3) :

```
10: begin
        instr_type=2'b11; //branch
        jmp_instr_type = instruction[25] && !instruction[24] ? 2'b01 :
                instruction[25] && instruction[24] ? 2'b10 : 2'b00;
end
```

For the mem_instr_type we can use already existing data_instr_type register which has same values in some cases.

```
mem_inst_type = data_instr_type == 1 ? 1 :
        data_instr_type == 2 ? 2 : 0;
```

Simulation & Verification

The code is synthesized without errors, but there are some warnings caused by unused bits.

decoder Project Status (12/08/2020 - 00:29:16)			
Project File:	Controler.xise	**Parser Errors:**	No Errors
Module Name:	decoder	**Implementation State:**	Placed and Routed
Target Device:	xc3s100e-5cp132	**•Errors:**	No Errors
Product Version:	ISE 14.7	**•Warnings:**	8 Warnings (0 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 Slice Latches	2	1,920	1%	
Number of 4 input LUTs	6	1,920	1%	
Number of occupied Slices	4	960	1%	
Number of Slices containing only related logic	4	4	100%	
Number of Slices containing unrelated logic	0	4	0%	
Total Number of 4 input LUTs	6	1,920	1%	
Number of bonded IOBs	11	83	13%	
IOB Latches	2			

Comparison

To achieve the purpose of this assignment can be used many approach. many of them can be implemented with same degree of efficiency, but this one is most readable and understandable. However approach used to find value for mem_instr_type register can be subject of discussion.

Conclusion

In this assignment I learn more about ARM instructions and get some practice in git and github.