Course Design of Computer Organization

Wang Zonghui

College of Computer Science & Technology,
Zhejiang University
March, 2010

topics

- 1. Verilog and Xilinx ISE
- 2. Basic logic component of datapath design
- 3. ALU and the ALU controller design
- 4. R-type instruction processor design
- 5. CPU controller design
- 6. Single clock-cycle CPU design
- 7. Multiple clock-cycle clock CPU design
- 8. Microprogrammed CPU controller unit design
- 9. Design of datapath of Microprogrammed CPU

Outline

- Experiment Purpose
- Experiment Task
- Basic Principle
- Operating Procedures

Experiment Purpose

- Understand the principles of Microprogrammed Multiple clock-cycle CPU
- Master design methods of Microprogram Controller
- Master design methods of Microprogrammed Datapath
- Master methods of program verification of Microprogrammed CPU

Experiment Task

- Design the Microprogrammed Datapath, and bring datapath and Microprogram Controller together to Implement CPU
- Verify the CPU with program and observe the execution of program

Basic Principle

- Microprogrammed Datapath
- Microprogram signals
- Sample Code
- Principle Diagram

Microprogrammed Datapath

- On the basis of a multiple clock-cycle CPU, replace the controller by microprogram controller
- Program and data share a common memory (Von Neumann architecture)
- When it is needed to add new instructions, you could modify the micro-program only.
  - Based on the last experiment, this experiment add “ADDI” instruction.
  - Only need to add a new micro-program instruction.

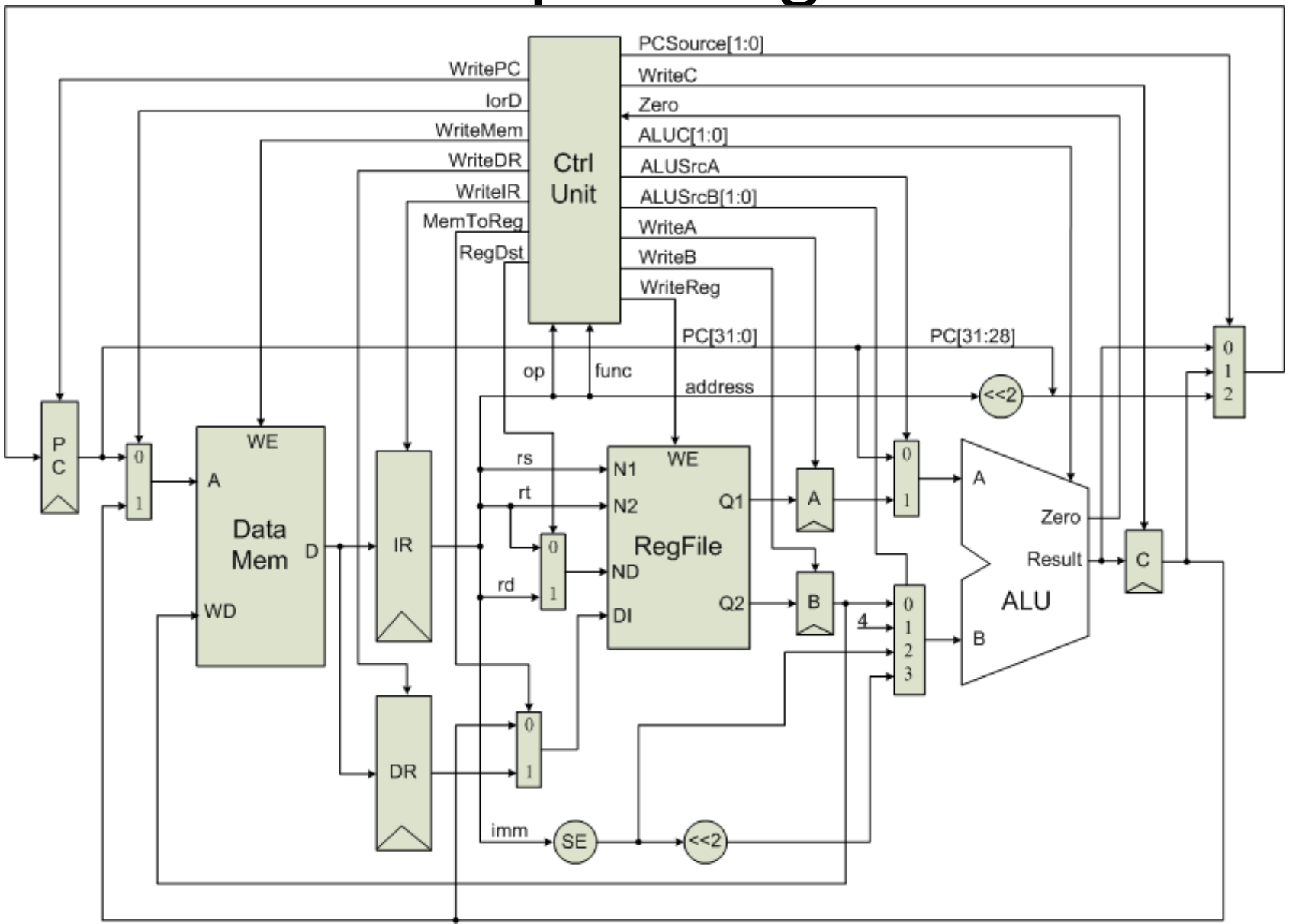
Microprogram

#	LABEL	ALU			Register	Memory	PC	Next	Control signals
		OP	A	B					PPABBRMGIRWTSSCPNN
0	Fetch	Add	PC	4		ReadPC	ALU	Seq	0000 1000 0101 0001 11
1		Add	PC	ExtSft	Read			Dispatch1	0001 1000 0000 0000 01
2	MEM1	Add	A	Ext				Dispatch2	0011 0000 0000 0000 10
3	LW2					ReadALU		Seq	0000 0000 1100 0000 11
4					WriteMDR			Fetch	0000 0011 0000 0000 00
5	SW2					WriteALU		Fetch	0000 0000 1010 0000 00
6	Rf1	Func	A	B				Seq	1010 0000 0000 0000 11
7					ALUout			Fetch	0000 0101 0000 0000 00
8	BEQ1	Sub	A	B			ALUCond	Fetch	0110 0000 0000 0110 00
9	J1						Jump		0000 0000 0000 1001 00
10	ADDi-2				WriteALUout			Fetch	0000 0001 0000 000 000

Sample Code

- PLS. refer to sample code in the Website.

Principle Diagram



Experiment Content

1. Write the Verilog module(refer to principle diagram and sample code).
2. Translat the test MIPS assembly code into machine code and write into the coe file, import the file when generate the IP core of memory .
3. Write the top module and i/o signals.
4. write the ucf.
5. Synthesize and impelment the project, generate code and download it to the FPGA experimental board.
6. Debug on the development board, observe and record the status of the various unit.

Test Program

address	MIPS compile	Machine code	description
0000	ADD $S0, $zero, $zero	00008020	$S0 number 16
0004	ADDI $S0, $zero, 48	20100030	$zero number 0
0008	LW $S1, $S0(0)	8E110000	$S1 number 17
000C	SW $S1, $S0(0)	AE110000	
0010	LW $S1, $S0(0)	8E110000	
0014	ADDI $S0, $S0, 4	22100004	
0018	LW $T0, $S0(0)	8E080000	$T0 number 8
001C	BEQ $zero, $T0, 28	10080002	
0020	ADD $S1, $S1, $T0	02288820	
0024	J 14	08000005	
0028	SW $S1, $0(0)	AE110000	
002C	END 4	FC000001	
0030		00000001	Starting data area

Input and Output

Light-emitting diodes								Seven-Segment Digital Tube			
7	6	5	4	3	2	1	0	3	2	1	0
display Mode			Display sources of digital tube data					Display 16-bit binary number			

Buttons				switched							
3	2	1	0	7	6	5	4	3	2	1	0
Enter	Left	Right	Reset	Set the memory address when Mode[1]=1							
							Set register address				

Mode Selection

Mode	value	description
[2]	0	When stroke Enter, then the execute one step
	1	When stroke Enter, then input data with the Mode [1:0]
[1]	0	Select memory address from: addr
	1	Select memory address from: switch[4:0]
[0]	0	The LED tube displays the data's low 16 bits specified by the Right.
	1	The LED tube displays the data's high 16 bits specified by the Right.
Note: The Left button changes Mode; when stroke Enter, should ensure that Mode [1] to be 0		

LED Tube Source and Content

hex	binary	description
5'h00	00000	Register PC's value
5'h01	00001	Memory value (Address is specified by the Switch [7:0] when Mode [1] = 1)
5'h02	00010	Instruction register IR's value
5'h03	00011	Register A's value
5'h04	00100	Register B's value
5'h05	00101	Register file's value (Address is specified by the Switch [4:0])
5'h06	00110	Register ALUout's value
5'h07	00111	Micro-controller's 18bit control signals
5'h08	01000	After Mode [1] choice, connect to the memory address
5'h09	01001	Register file's value(Address is specified by Instruction[25:21])
5'h0a	01010	Register file's value(Address is specified by Instruction[20:16])
5'h0b	01011	Others: {MemRead, Seq, carry, overflow, zero}
Other value		Input values by the switch Switch [7:0] when mode Mode [2] = 1
Note: LED tube's data source is changed by the Right button.		

Note

- Avoid to inconsistency between definition and use.
- Define a variable with type and width in obvious.
- Adopt the name association method in Module instance.
- Keep a minimum amount of warnings in ISE.
- Write the enough comment in code to facilitate the search for the error.

Thanks!