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 design
- 5. CPU controller design
- 6. Design of datapath of single-cycle clock CPU
- 7. Multiple-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

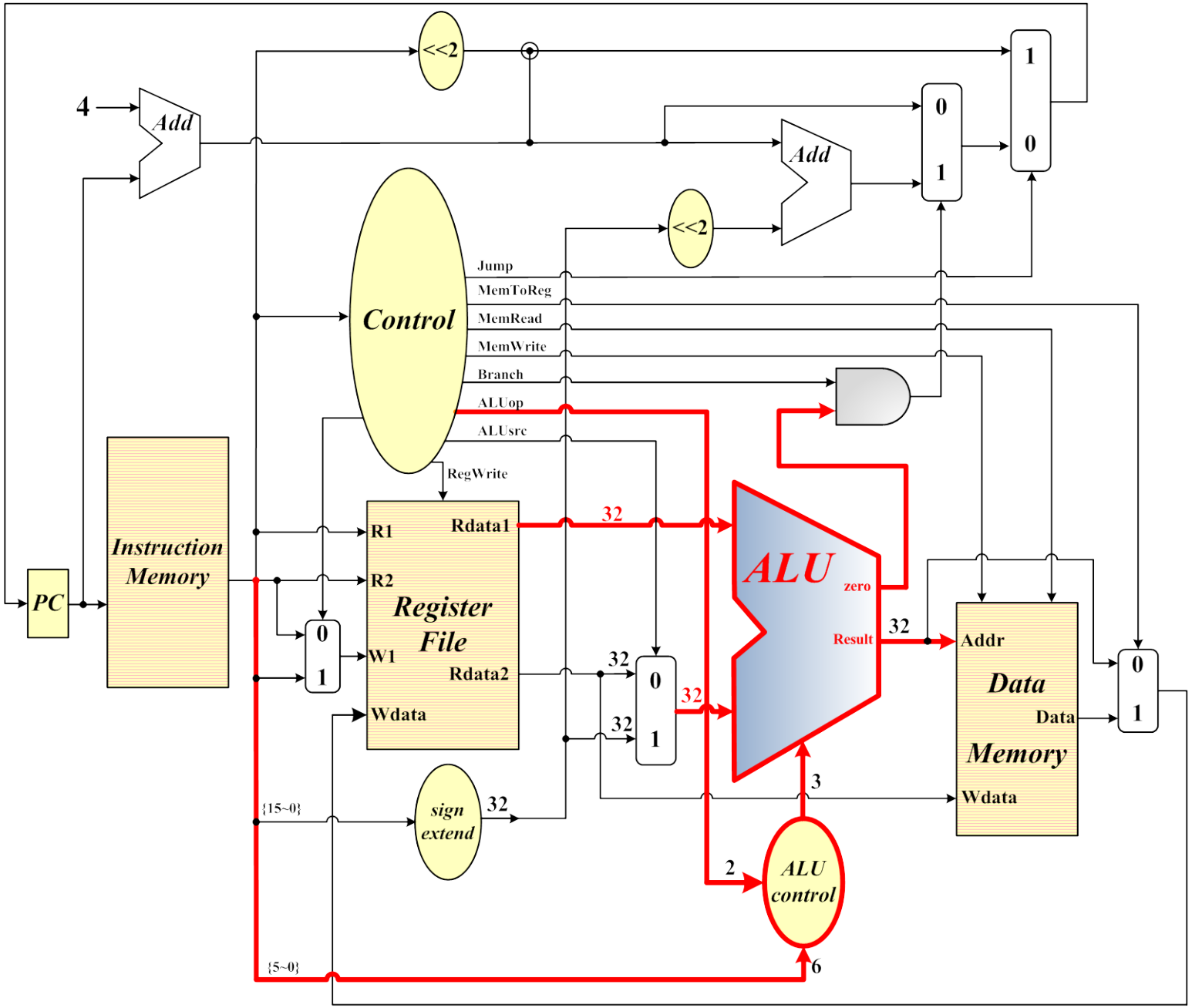
- Master the principle and design method of ALU.
- Master the principle and design method of ALU Controller.

Experiment Task

- Implementation of 32-bit ALU.
- Implementation of ALU Controller.

Basic Principle

- ALU
- ALUC


4
Add
<<2
Add
<<2
0
1
1
0
Control
Jump
MemToReg
MemRead
MemWrite
Branch
ALUop
ALUsrc
RegWrite
PC
Instruction Memory
R1
R2
W1
Wdata
Rdata1
Rdata2
Register File
32
32
32
32
ALU
zero
Result
32
Addr
Data
Memory
Data
Wdata
3
0
1
0
1
0
1
{15~0}
sign extend
32
2
ALU control
{5~0}
6

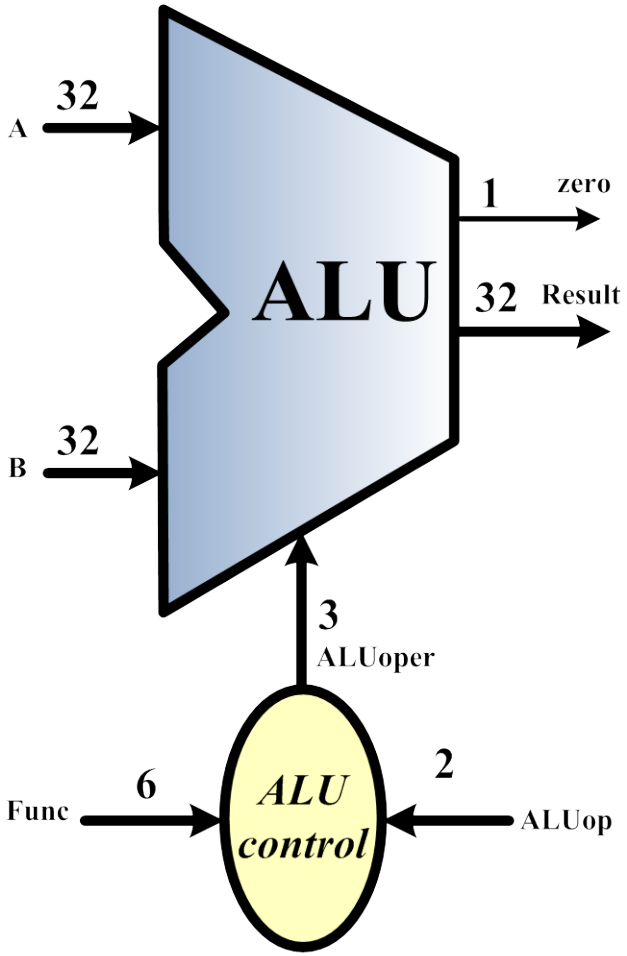
R-type Instr. Format

op	rs	rt	rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

field	signification
op	Instruction opcode
rs	The first source operand register
rt	The second source operand register
rd	Dest. operand register, store the result
shamt	Shift amount
funct	Function field, to select an operation from the op field

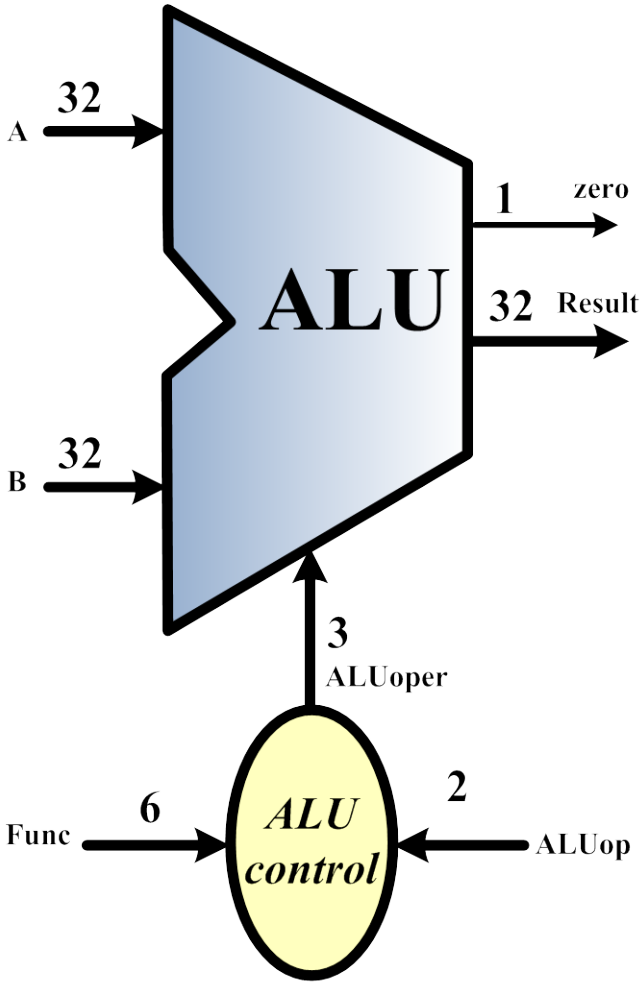
ALU

Control	function	operation
000	and	Result = A & B
001	or	Result = A | B
010	add	Result = A + B
110	sub	Result = A – B
111	slt	Result = A < B ? 1 : 0



ALU Controller

input		output
ALUop	Function	Operation
10	and	000
10	or	001
00	add	010
01	sub	110
10	slt	111



ALU Operation Code Table

ALUop		Func（from R instruction）						operation
ALUop1	**ALUop2**	**F5**	**F4**	**F3**	**F2**	**F1**	**F0**	
0	**0**	×	×	×	×	×	×	**010:ADD**
0	**1**	×	×	×	×	×	×	**110:SUB**
1	×	×	×	**0**	**0**	**0**	**0**	**010:ADD**
1	×	×	×	**0**	**0**	**1**	**0**	**110:SUB**
1	×	×	×	**0**	**1**	**0**	**0**	**000:AND**
1	×	×	×	**0**	**1**	**0**	**1**	**001: OR**
1	×	×	×	**1**	**0**	**1**	**0**	**111:SLT**

Experiment Content

- ALU: FPGA Verification.
- ALU Controller: FPGA Verification.

ALU I/O

- Input
  - 3 slide switches for ALU Controller code
  - 2 slide switches for Output control.
- Output
  - 4 LED tubes

2 switches	output
00	A = 1122
01	B = 3344
1×	Result: A op B

3 switches	Control input	Operation result
000	and	1100
001	or	3366
010	add	4466
110	sub	ddde
111	slt	0001

ALU Controller I/O

- Input
  - 2 buttons for ALUop
  - 4 slide switches for function code
  - 2 slide switches for Output control.
- Output
  - 4 LED tubes

Precaution(1)

- ALU top

```
module alu_top(clk, switch, o_seg, o_sel)
    …
    alu M1(i_r, i_s, swtich[4:2], o_zf, disp_code);
    display M3(clk, disp_num, o_seg, o_sel);
    assign disp_num = switch[0]?disp_code: (switch[1]?i_r:i_s);
end module
```

Precaution(2)

- ALU Controller top

```
module aluc_top(clk, btn, swtich, o_seg, o_sel)
    …
    alu M1(i_r, i_s, aluc, o_zf, disp_code);
    aluc M2 (btn, swtich[5:2], aluc);
    display M3(clk, disp_num, o_seg, o_sel);
    assign disp_num = switch[0]?disp_code: (switch[1]?i_r:i_s);
end module
```