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

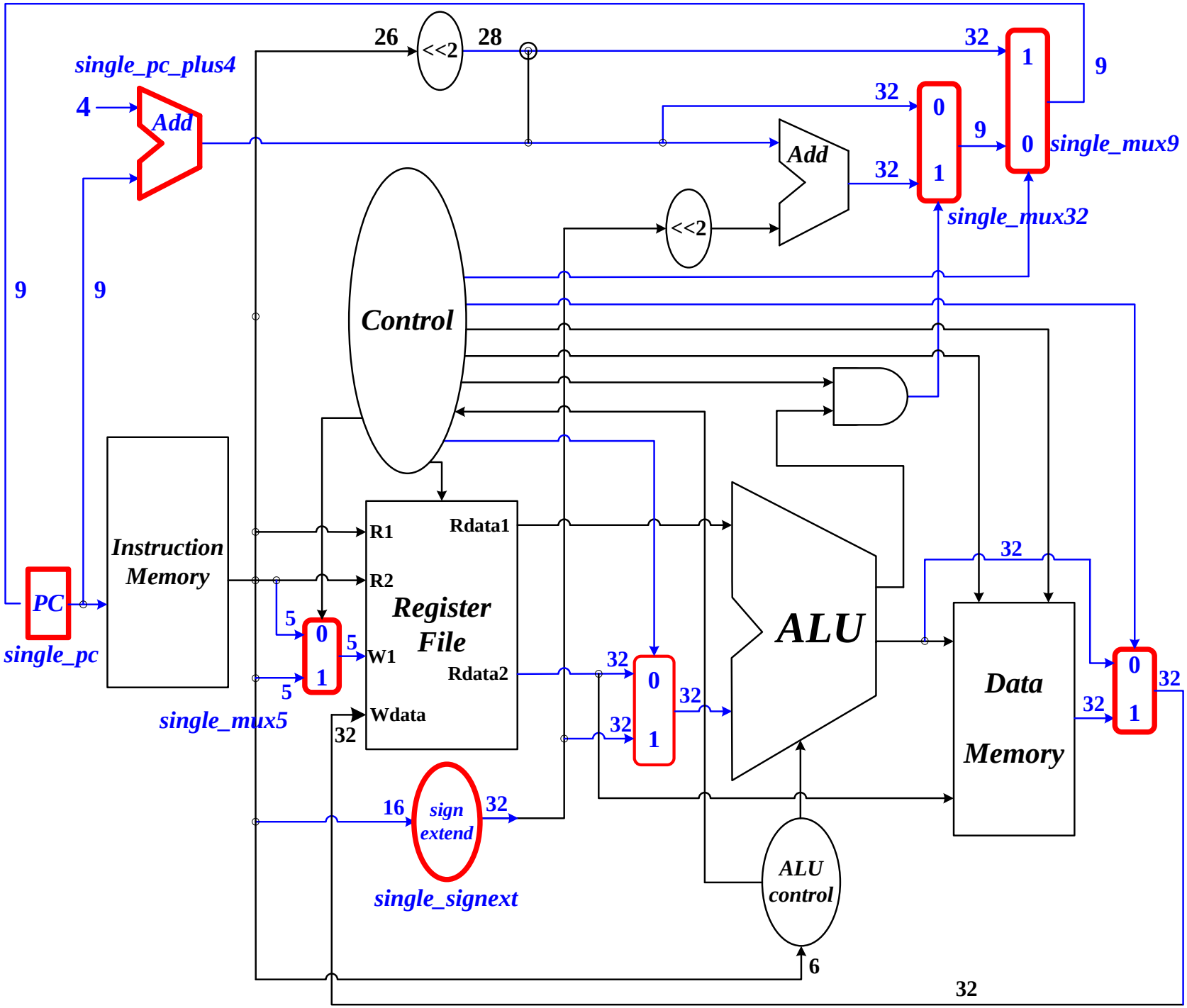
- Familiar with the design method of the basic logic components, and master the working principle and design method of the basic components.
- Master the design of the PC program counter and Implementation of PC + 4.
- Master the design of sign extension module.
- Master the design of a simple adder.

Experiment Task

- Implementation of Single PC Module (perform Waveform Simulation).
- Implementation of Single Mux Module (FPGA verification).
- Implementation of Sign Extension Module (perform Wavefrom Simulation).
- Implementation of Simple Adder Module (perform Wavefrom Simulation).

Basic Principle

- Program Counter
- Mux
- Sign extension
- Simple adder

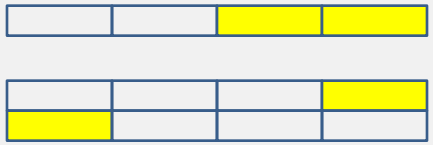

single_pc_plus4
4
Add
26
<<2
28
32
1
9
32
0
9
Add
32
1
0
single_mux9
single_mux32
<<2
9
9
Control
Instruction Memory
PC
single_pc
single_mux5
5
0
1
5
5
R1
R2
W1
Wdata
Rdata1
Register File
Rdata2
32
32
0
1
32
32
ALU
32
Data Memory
32
0
1
32
16
sign extend
32
single_signext
ALU control
6
32

Program Counter

- Program counter PC, used to control the machine instructions' implementation order , is essentially the instruction address in the memory.
- According to the PC, the current CPU reads instructions to execute, and change the value of PC for each reading.
- Change of the value of PC has deal with the internal structure of memory:
  - If the memory data width is 8 bits, and instruction length is 32-bit: Read an instruction, PC ← PC +4
  - If the memory data width is 32 bits, and instruction length is 32-bit: Read an instruction, PC ← PC +1

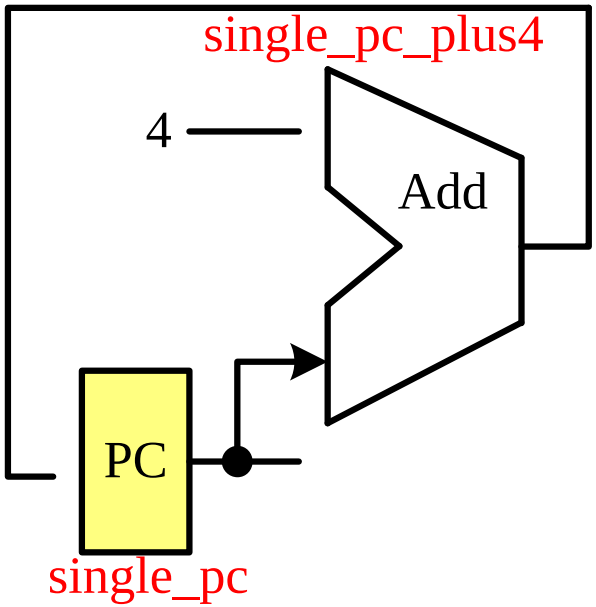
Address Alignment

- Instructions memory: 32bit×512, address width[$\log_2 512$] = 9bits.
- Read and write analysis of 32bit memory:

	Address feature	Instruction addressing		Data addressing		legend
		read	write	read	write	
word (alignment)	Last 2 bits are 0	Read once	N.A.	Read once	Write once	
Half word	Last 1 bit is 0	N.A.	N.A.	Read 1~2 times/ compute Additionally	Write 1~2times/ Compute Additionally	
byte	Last bit is unsure	N.A.	N.A.	Read 1 time/ compute Additionally	Write 1 time/ Compute Additionally	

PC Verilog Code

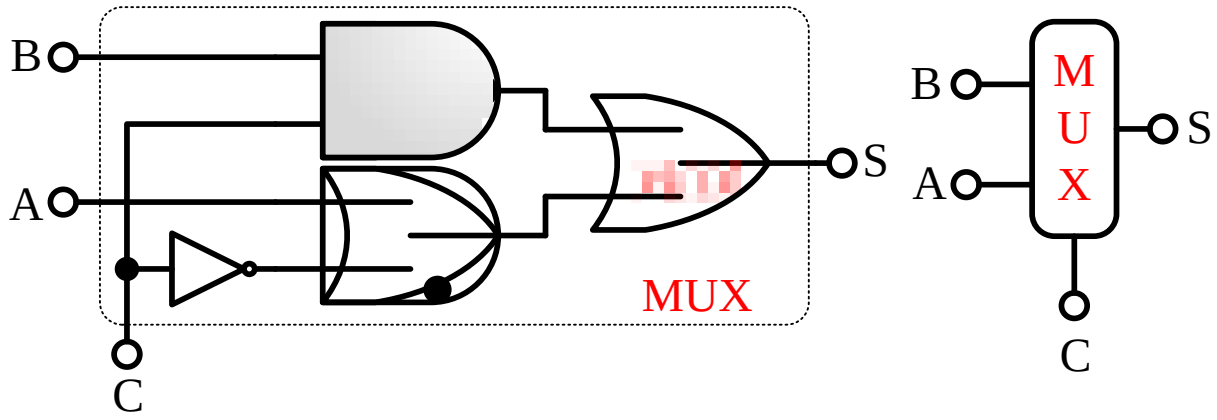
```
module single_pc(clk, rst, i_pc, o_pc)
    input wire clk, rst;
    input wire [8:0] i_pc;
    output reg [8:0] o_pc;
    always @(posedge clk)
        assign o_pc = rst ? 9{1'b1} : i_pc;
endmodule

module single_pc_plus4(i_pc, o_pc);
    input  wire [8:0] i_pc;
    output wire [8:0] o_pc;
    assige o_pc = i_pc[8:0] + 1;
endmodule
```



Mux

- Multiplexer output is the selected one from multiple input values by a control signal.
- CPU design uses these selectors:
  - 1-bit, 2-bit, 5-bit, 9-bit, 32-bit

C	S
0	A
1	B

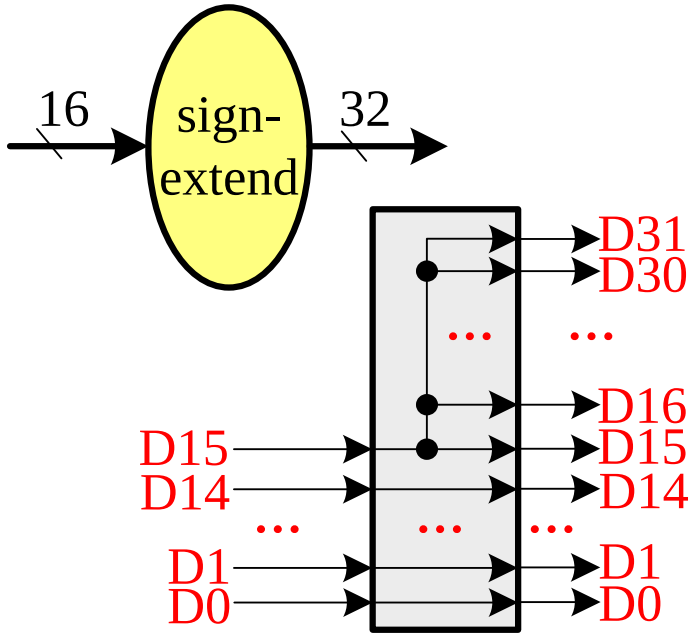


Mux Verilog Code

```
module single_mux (A, B, C, S);
   paramter N = 1;   // N: 1, 2, 5, 9, 32
   input  wire [N-1:0] A, B;
   input  wire         C;
   output wire [N-1:0] S;
   assign S = C ? B : A ;
endmodule
```

Sign Extension

- By complement rules, sign extension: If the 16th digit is 1, the extended bit is 1, otherwise 0.
- It is mainly used in the CPU design for extending the 16-bit input data to 32-bit data as output



Sign Extension Verilog Code

```
module single_signext(i_16, o_32);
  input wire [15:0] i_16;
  output reg [31:0] o_32;
  always @(i_16)
    o_32 <= {{16{i_16[15]}}, i_16[15:0]};
endmodule
```

Simple Adder Verilog Code

```
module single_add (i_op1, i_op2,  o_out);
   parameter N = 32;
   input  wire[N-1:0] i_op1, i_op2;
   output wire[N-1:0] o_out;
   assign o_out = i_op1 + i_op2;
endmodule
```

Experiment Content

- PC: Perform Waveform Simulation.
- Mux: FPGA Verification:

  8 switches divided into two groups of 4, with a button for selecting control: display the four digits selected with a digital tube and four LED.
- Sign Extension: Perform Waveform Simulation.
- Single Adder: Perform Waveform Simulation.

Precaution

- Simulation for PC module

```
module pc_top(clk, rst, i_pc, o_pc)
    …
    single_pc M1(clk, rst, i_pc, o_pc);
    single_pc_plus4 M2(o_pc, i_pc);
end module
perform waveform simulation for pc_top.
```

- Sign Extension inputs include 16th bit 0 and 1;