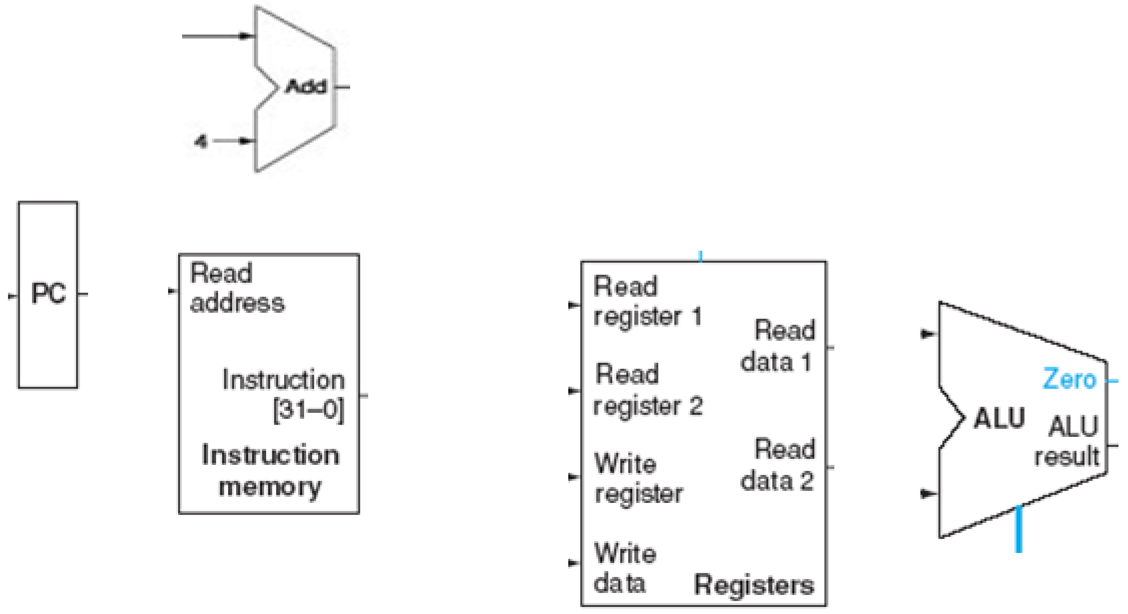
MIPS processor continued

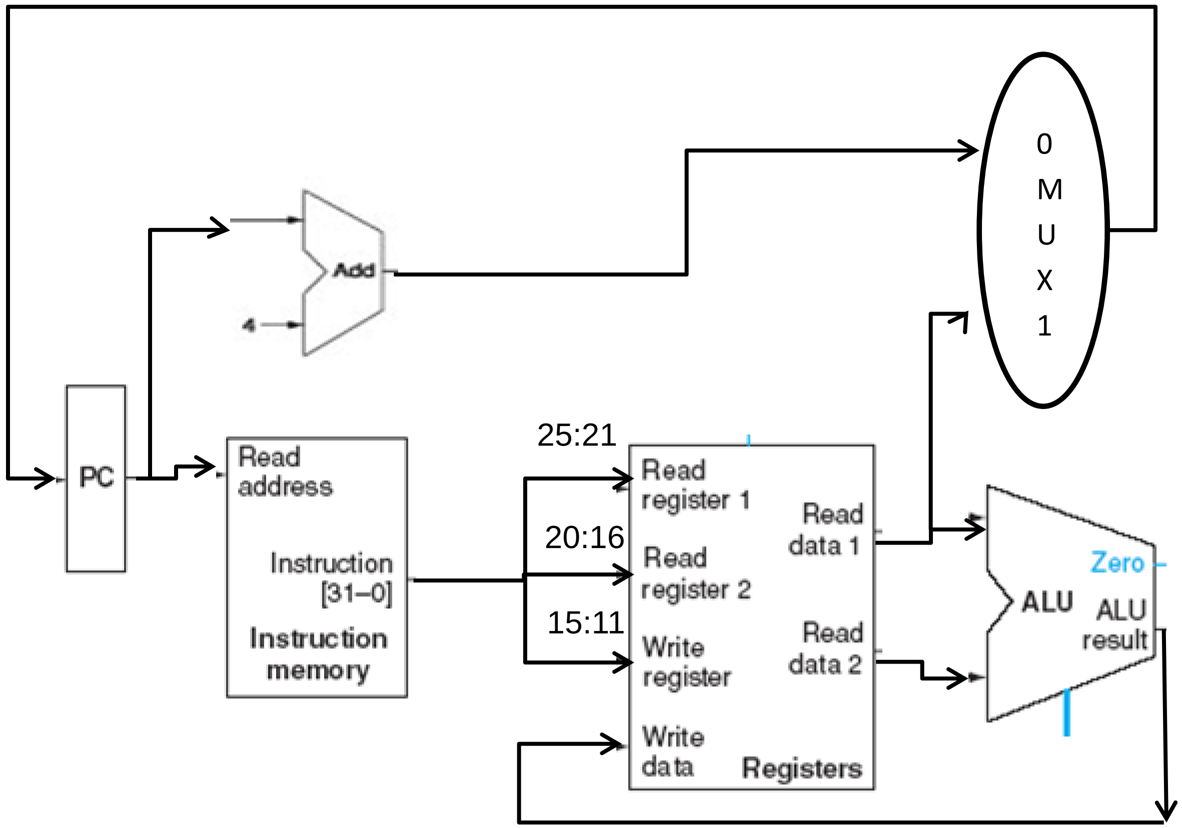
In Class Exercise Question

- Show the datapath of a processor that supports only R-type and jr reg instructions



In Class Exercise Answer

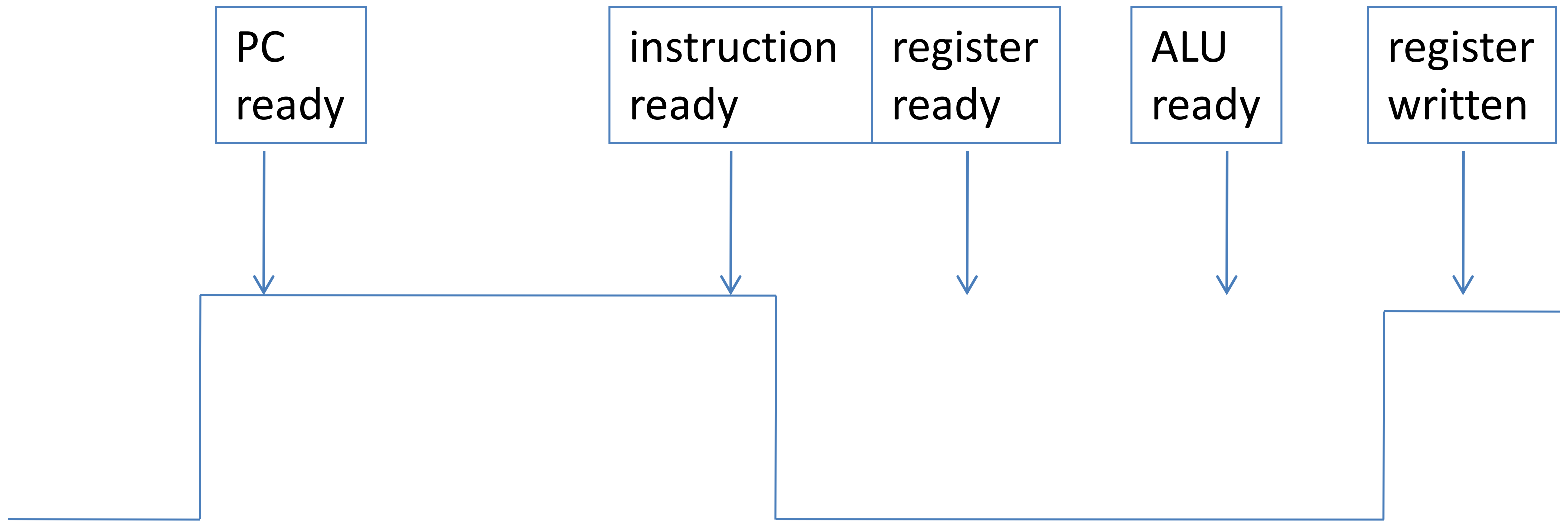
- Show the datapath of a processor that supports only R-type and jr reg instructions



Performance

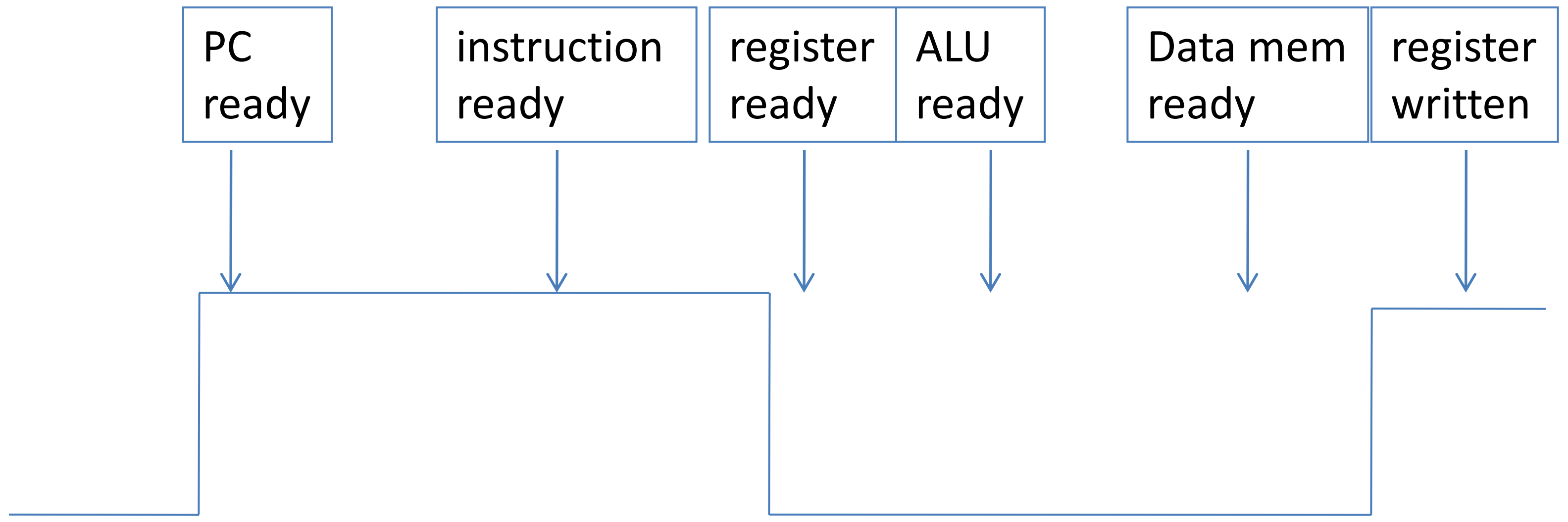
- Assume that
  - Memory access: 200ps
  - ALU and adders: 100 ps
  - Register file read: 50ps
  - Register file write: 10ps (the clk-to-q delay)
  - PC update: 10ps (the clk-to-q delay)
  - The setup time of DFFs: 10ps
  - Other parts do not have delay
- How fast is
  - An R-type instruction?
  - A lw instruction?
  - A sw instruction?
  - A beq instruction?
- Need to find the critical path – the longest path

R-type



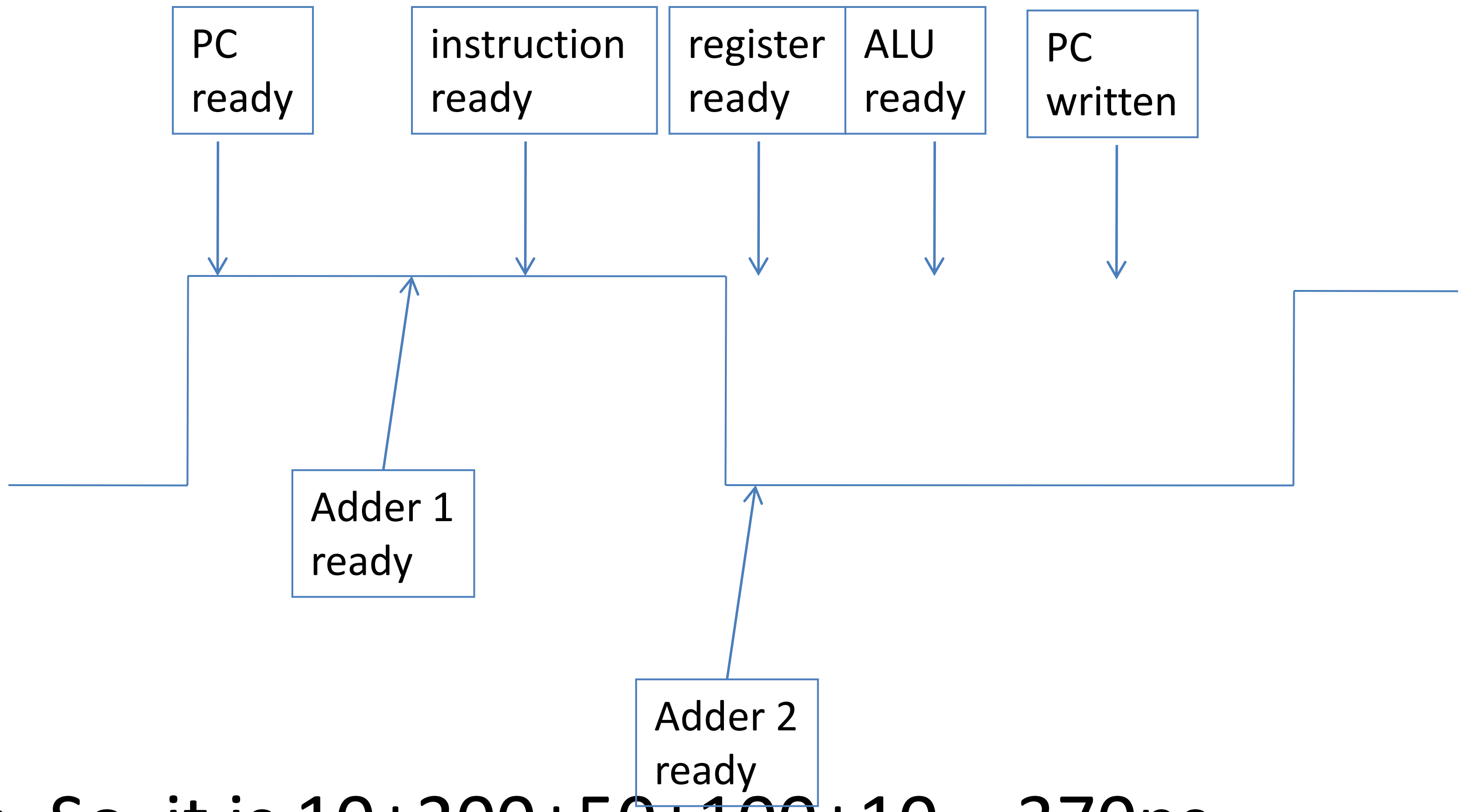
- So, the clock needs to be at least 10+200+50+100+10 = 370ps
- Will there be a problem if the next instruction is also an R-type instruction, considering that the register is written and stable only after the next rising edge of the clock?
- Figure not to the exact scale

lw



- So, the clock needs to be at least 10+200+50+100+200+10 = 570ps
- Figure not to the exact scale

beq



- So, it is 10+200+50+100+10 = 370ps
- Figure not to the exact scale

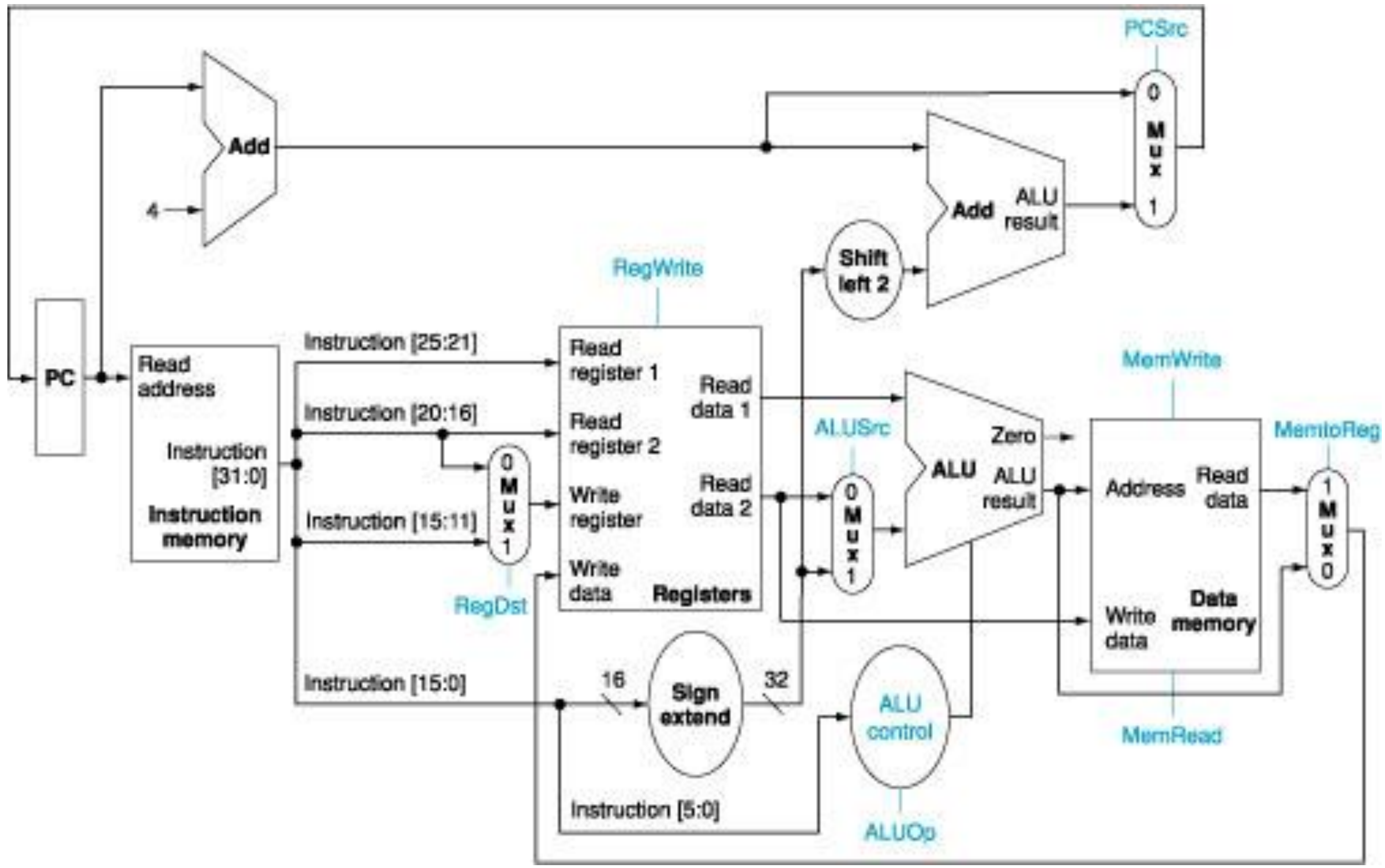
Clock cycle

- So, how long should the clock cycle be?
- Is it efficient?

Control Signals

- Control signals include ALUCtrl and the signals to control the 2-1 selectors
- They are generated according to the current instruction, using the opcode [31-27] and the funct [5-0] field in the instruction.

Datapath for Memory, R-type and Branch Instructions, plus the control signals



The Effect of Control Signals

Signal name	Effect when deasserted	Effect when asserted
RegDst	The register destination number for the Write register comes the rt field (20:16)	The register destination number for the Write register comes the rd field (15:11)
RegWrite	None.	The register on the Write register input is written with the value on the Write data input.
ALUSrc	The second ALU operand comes from the second register file output	The second ALU operand is the sign-extended, lower 16 bits of the instruction
PCSrc	The PC is replaced by the output of the adder that computes the value of PC + 4	The PC is replaced by the output of the adder that computes the branch target
MemRead	None.	Data memory contents designated by the address input are out on the Read data output.
MemWrite	None.	Data memory contents designated by the address input are replaced by the value on the Write data input.
MemtoReg	The value fed to the register Write data input comes from the ALU	The value fed to the register Write data input comes from the data memory

Table for Control Line Setting

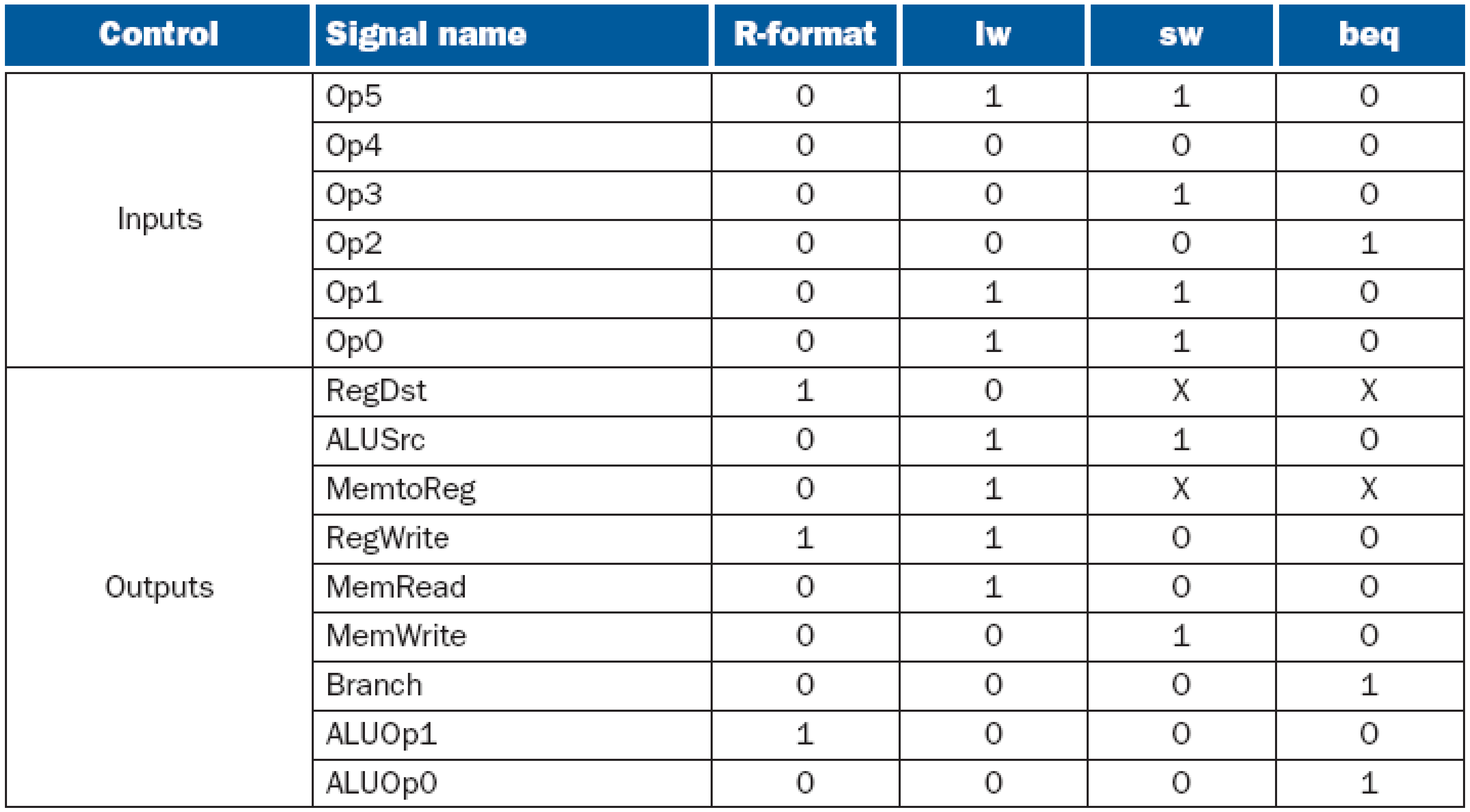
Note: Branch is anded with ALU zero output to produce PCSrc

Instruction	RegDst	ALUSrc	Memto-Reg	Reg Write	Mem Read	Mem Write	Branch	ALUOp1	ALUOp0
R-format									
Lw									
Sw									
beq									

Table for Control Line Setting

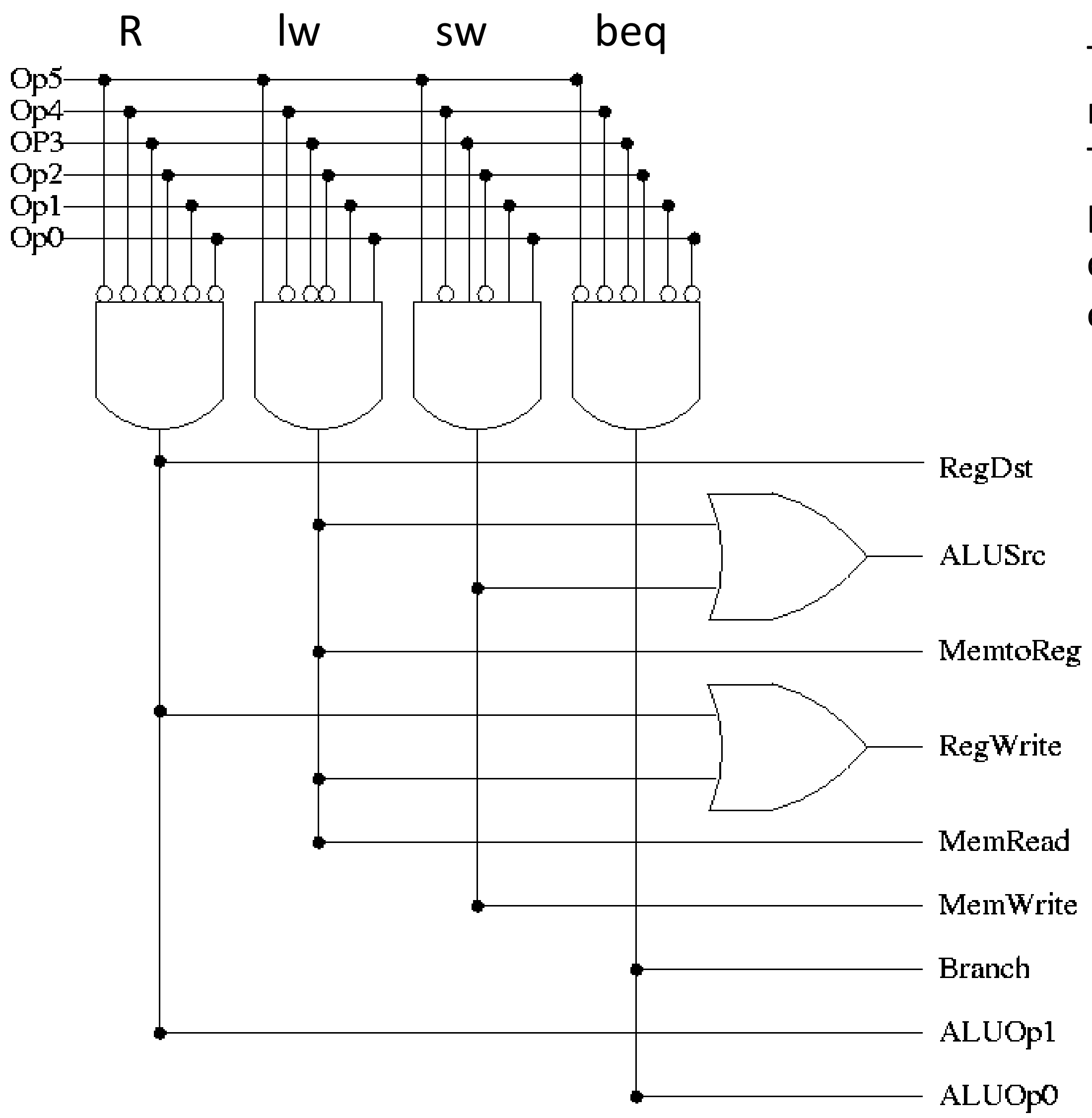
Instruction	RegDst	ALUSrc	Memto-Reg	Reg Write	Mem Read	Mem Write	Branch	ALUOp1	ALUOp0
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0
beq	X	0	X	0	0	0	1	0	1

Truth Table for Control Function



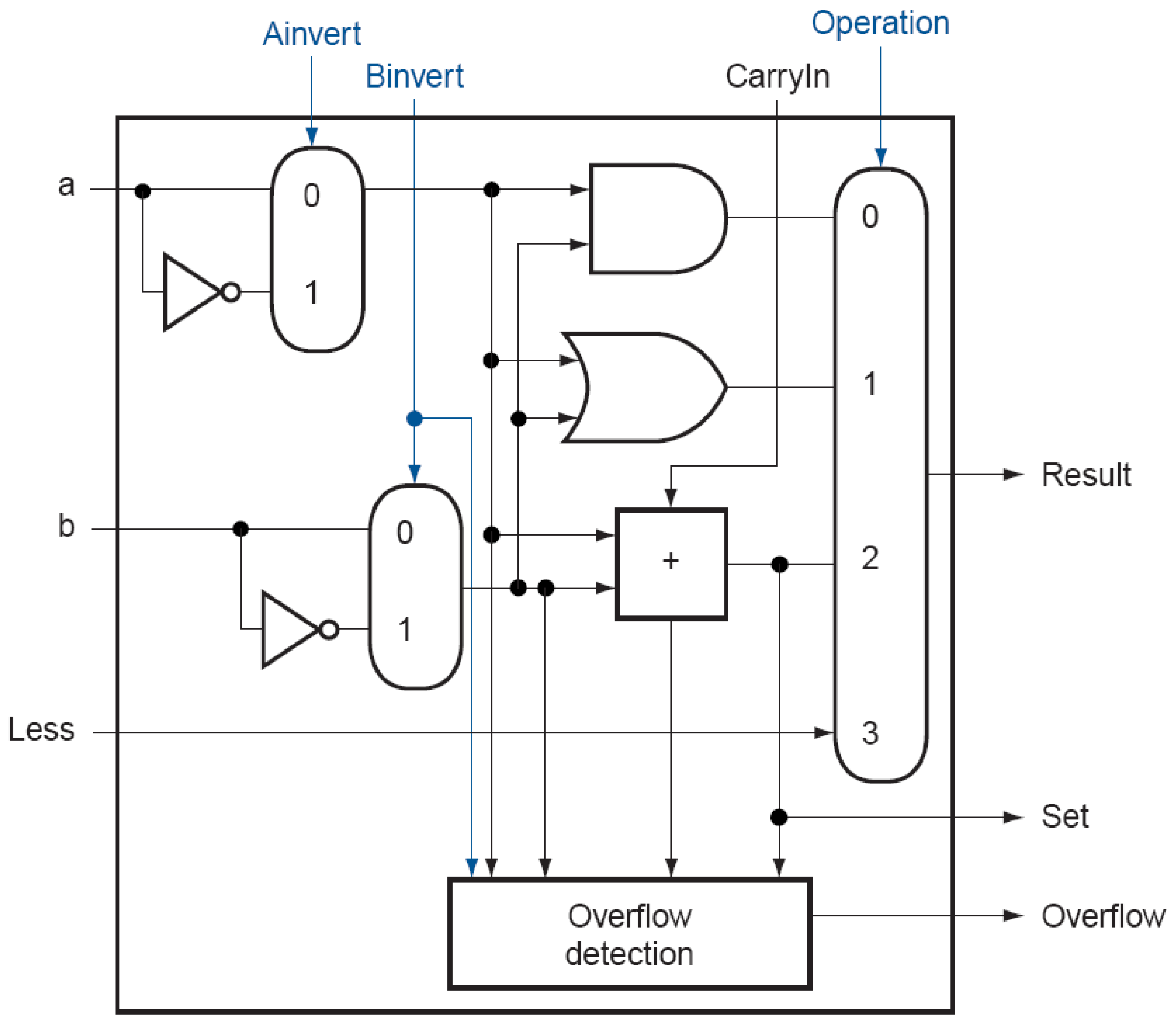
Control	Signal name	R-format	lw	sw	beq
Inputs	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
Outputs	RegDst	1	0	X	X
	ALUSrc	0	1	1	0
	MemtoReg	0	1	X	X
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1

Implementation Using PLA



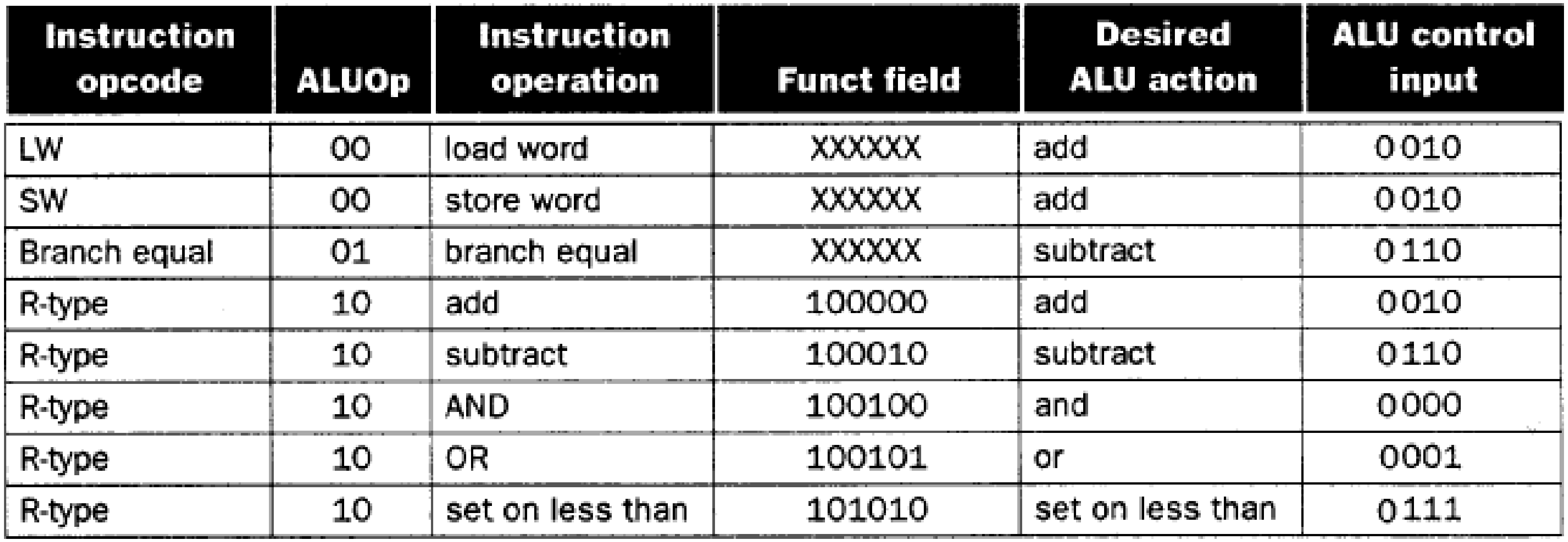
The way to read this -- There are only 4 possible combination of inputs

MIPS ALU unit



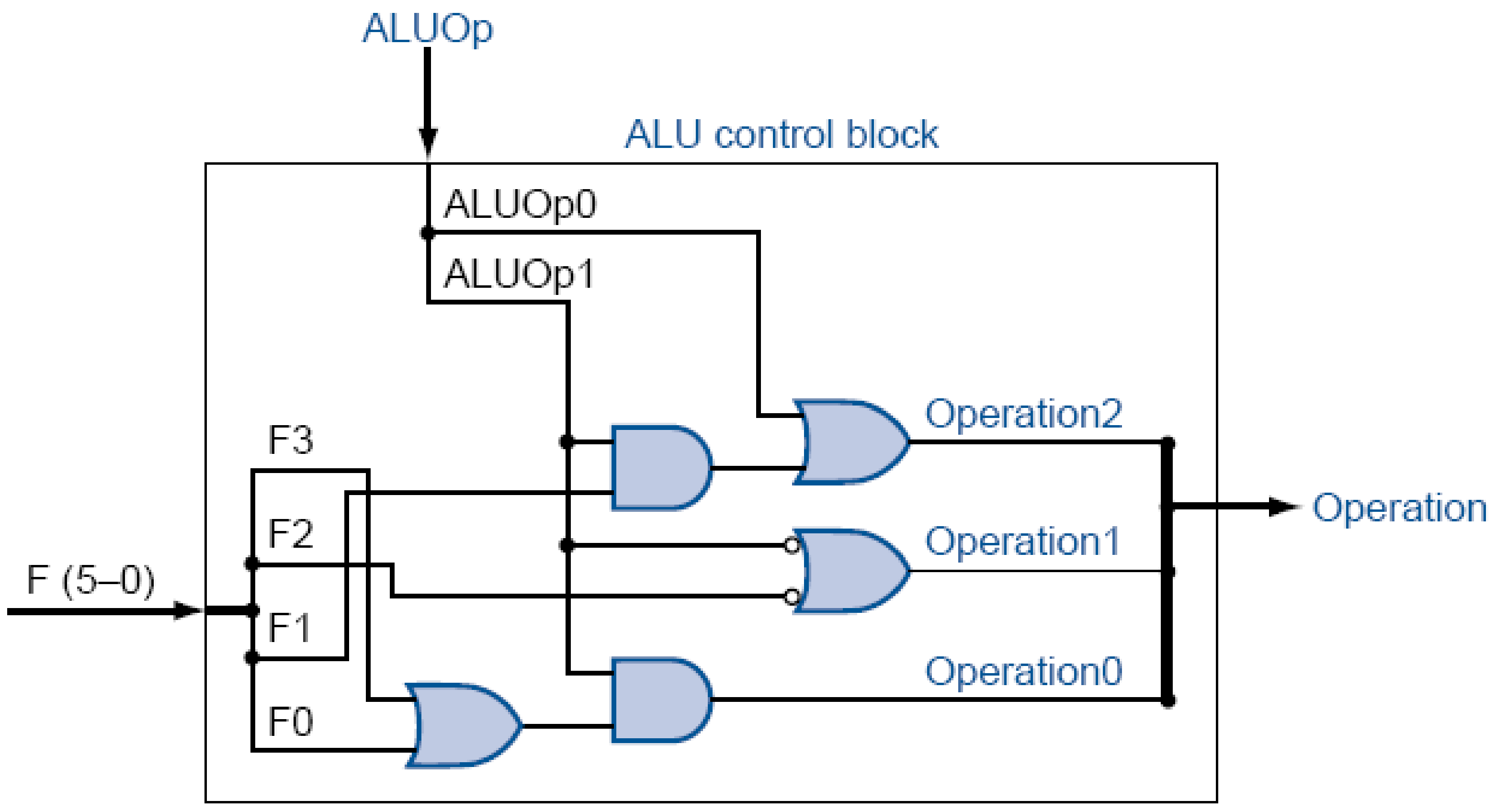
ALU Control

- Use Opcode to get ALUOp, then combine ALUOp with Funct
- Two levels of decoding, more efficient
- Assume ALUOp has been determined as such for each instruction



Instruction opcode	ALUOp	Instruction operation	Funct field	Desired ALU action	ALU control input
LW	00	load word	XXXXXX	add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
R-type	10	subtract	100010	subtract	0110
R-type	10	AND	100100	and	0000
R-type	10	OR	100101	or	0001
R-type	10	set on less than	101010	set on less than	0111

One Implementation



ALU control bit 3 is always 0 for this set of instructions

Can verify that the output is correct for lw, sw, beq

For R-type, op2=F1, op1= ~F2, op0 = F3 | F0