CS / EE 320
Computer Organization and Assembly Language
Spring 2024
Lecture 16

Shahid Masud

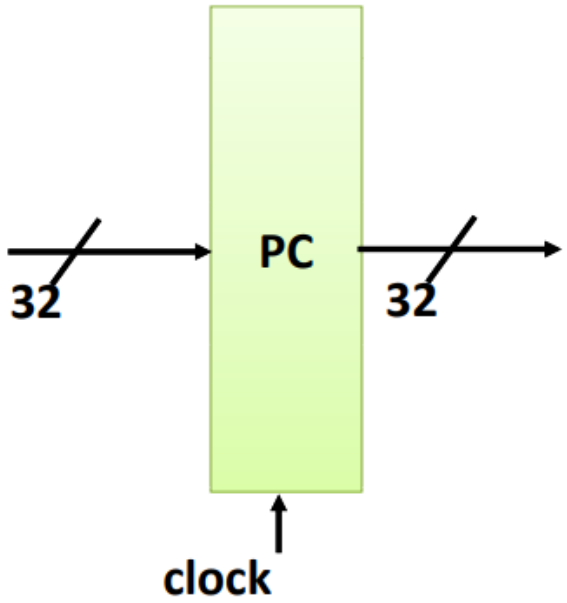
Topics: MIPS Single Cycle ALU Design, Adding Control Signals

Topics

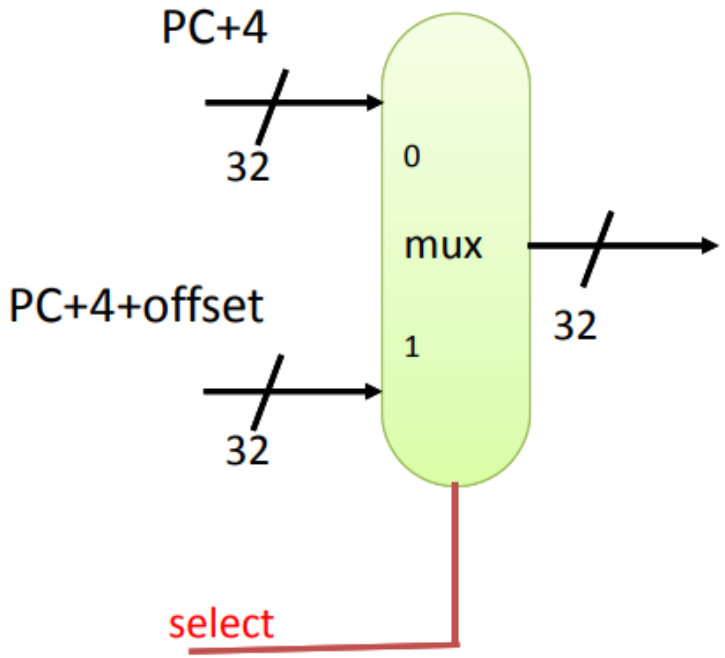
- Review Single Cycle MIPS Datapath design
- Different fields in R, I, J type instructions and how they are used in instruction decoding
- Design of Control Signals for single cycle Simple MIPS CPU
- Use of Sign-Extension and padding two LSB Bits in CPU architecture
- Incorporating multiplexers where multiple inputs / outputs are selected
- Inputs and Outputs of Control Unit
- Devising Control Words for all 6 instructions in simple MIPS CPU

MIPS Components - 1

MIPS Components - Register

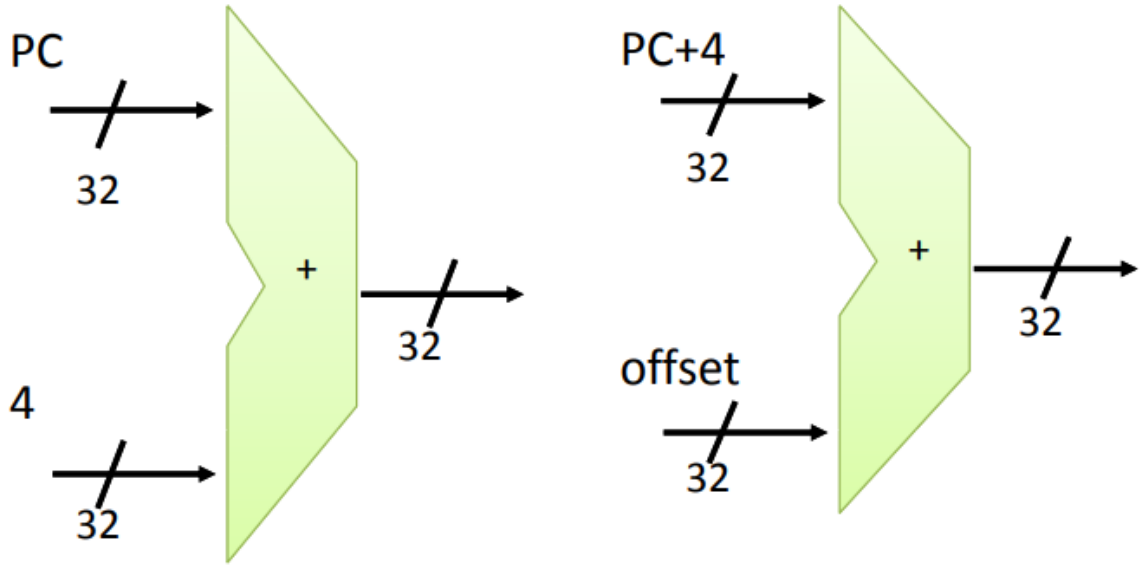


MIPS components - Multiplexers

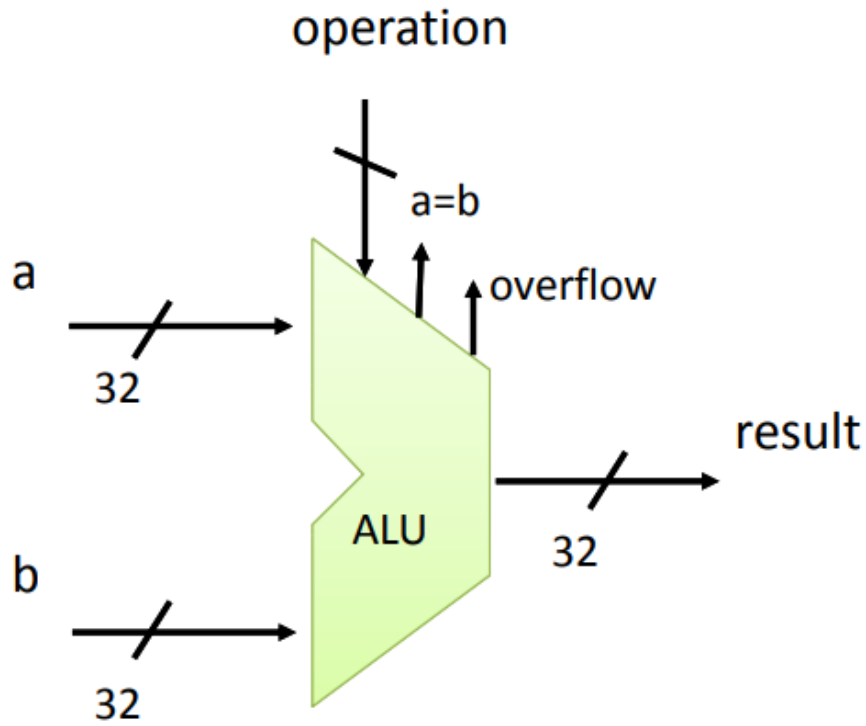


MIPS Components - 2

MIPS Components - Adder

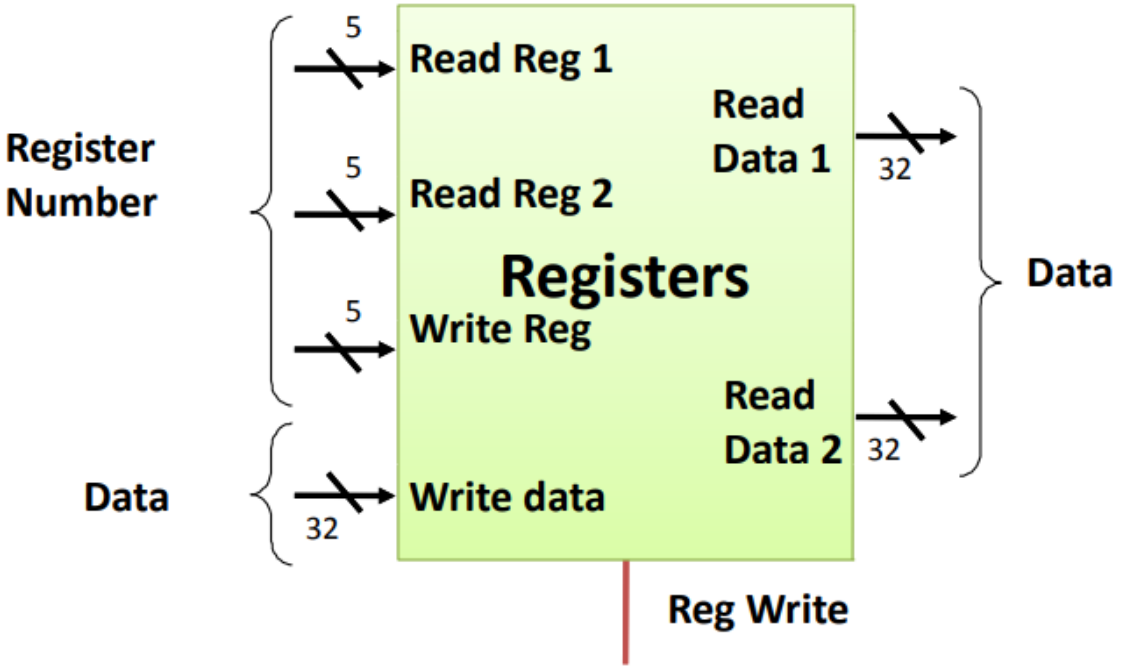


MIPS Components - ALU

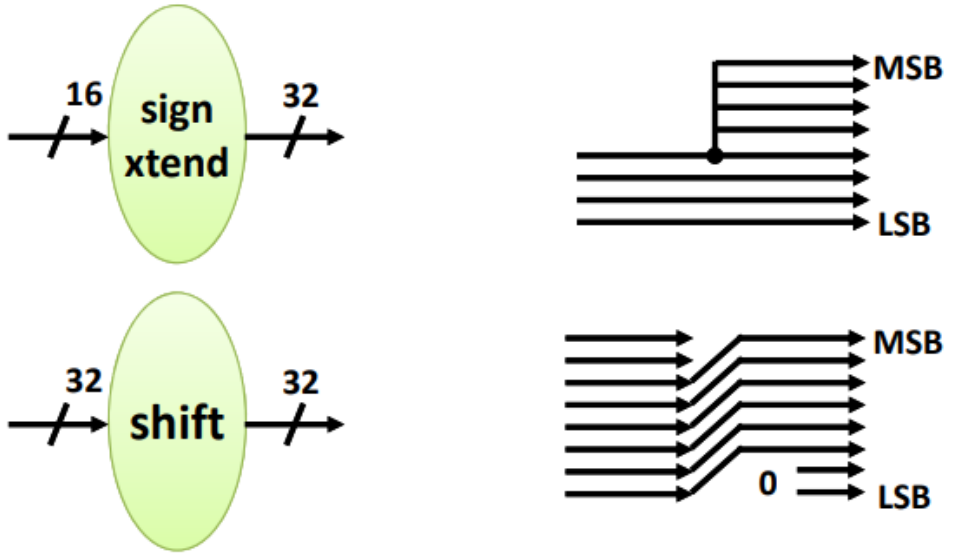


MIPS Components - 3

MIPS Components - register file

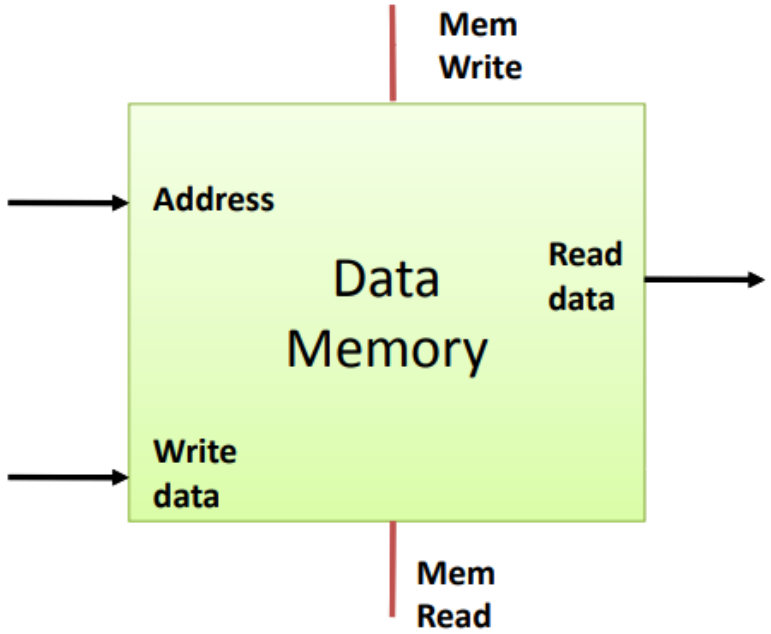


MIPS Components - Bit manipulation circuits

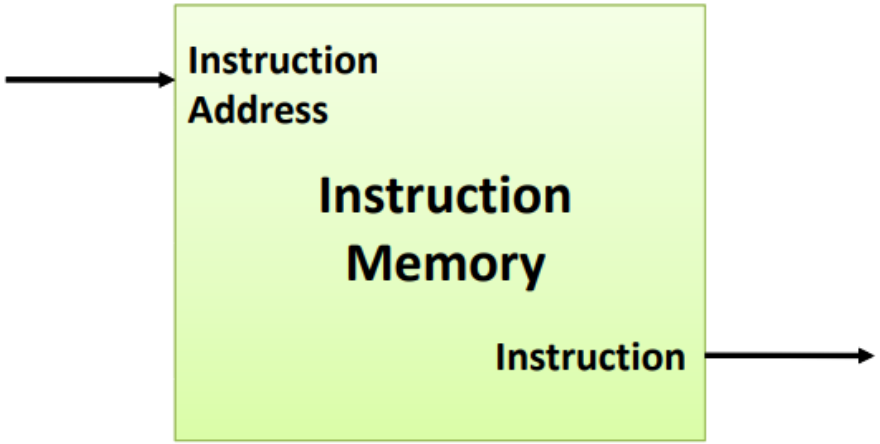


MIPS Components - 4

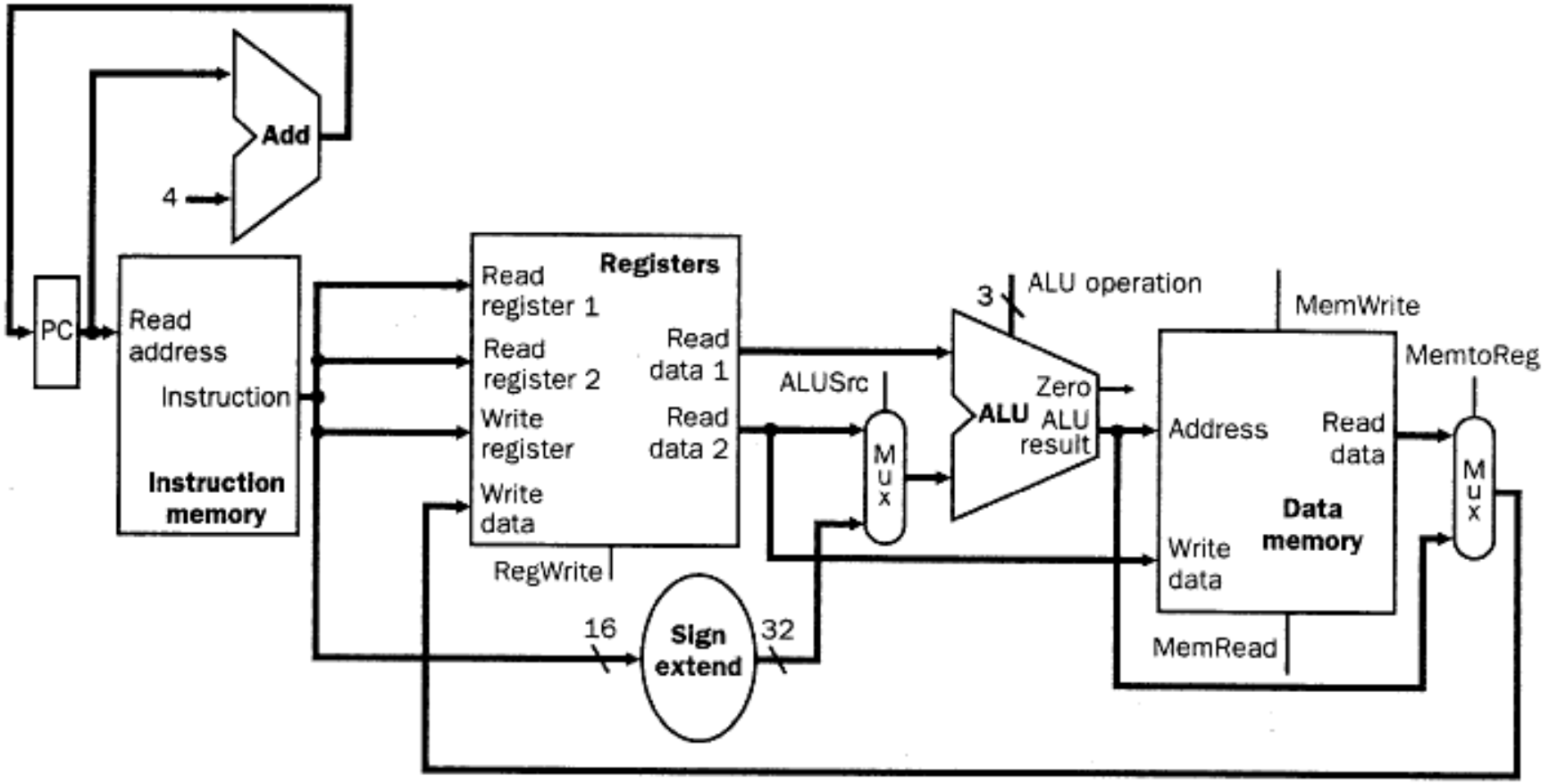
MIPS Components -Data memory



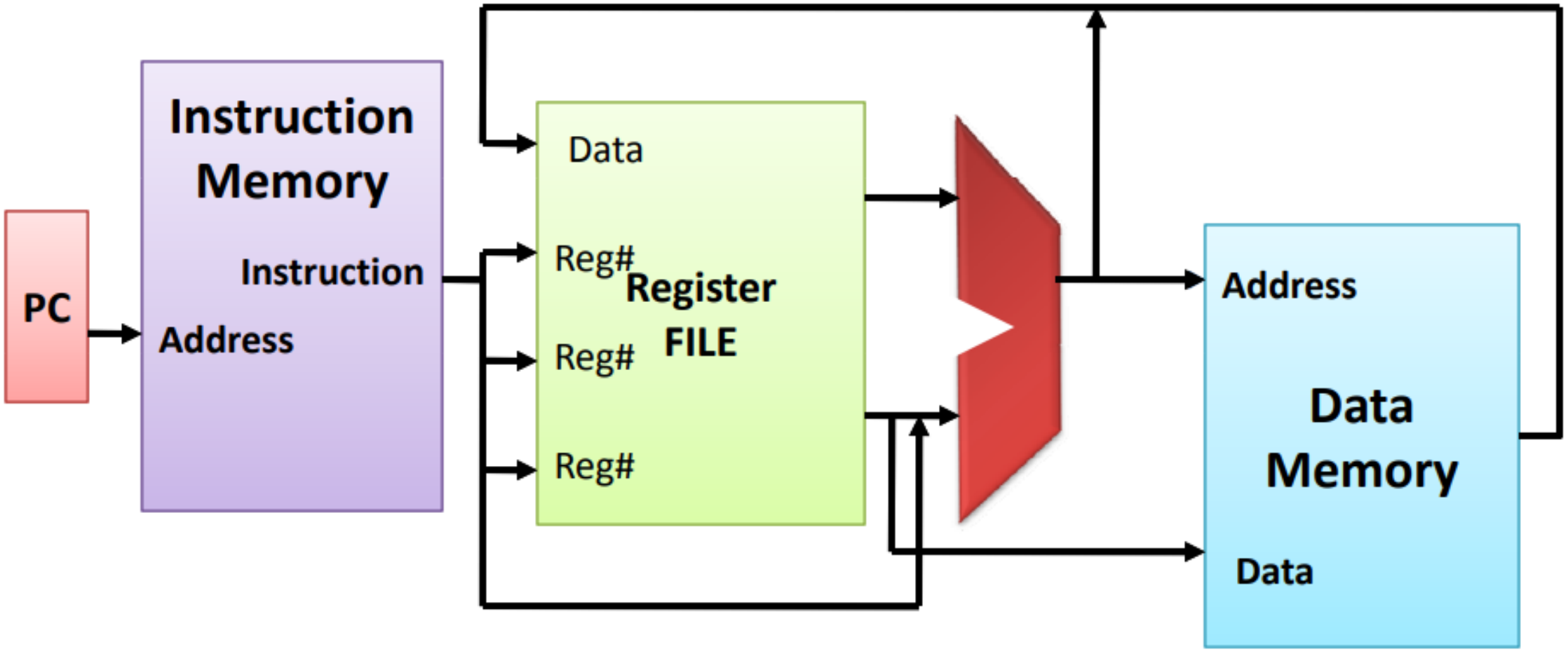
MIPS Components: Program memory



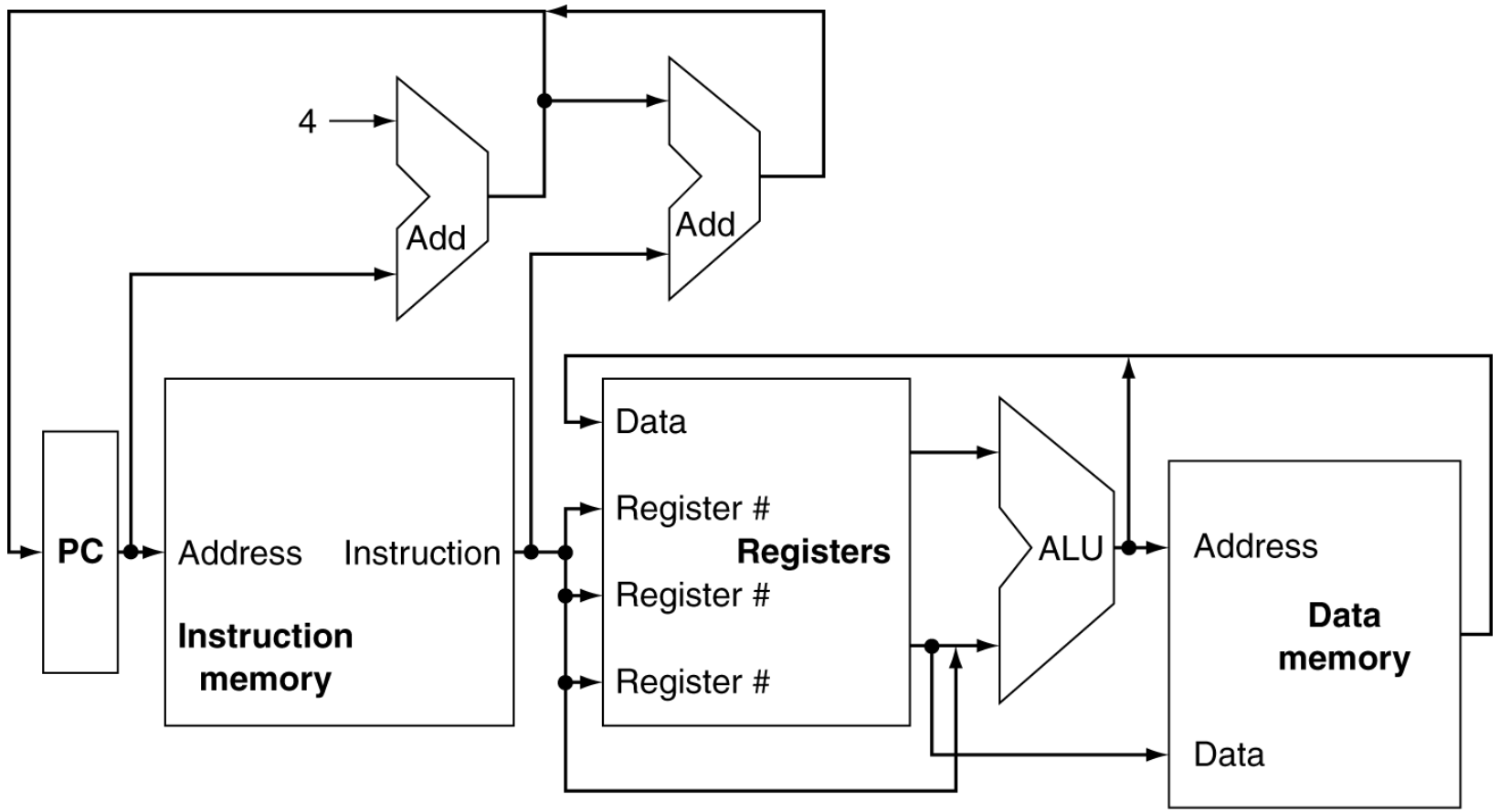
Data path without Jump and Branch



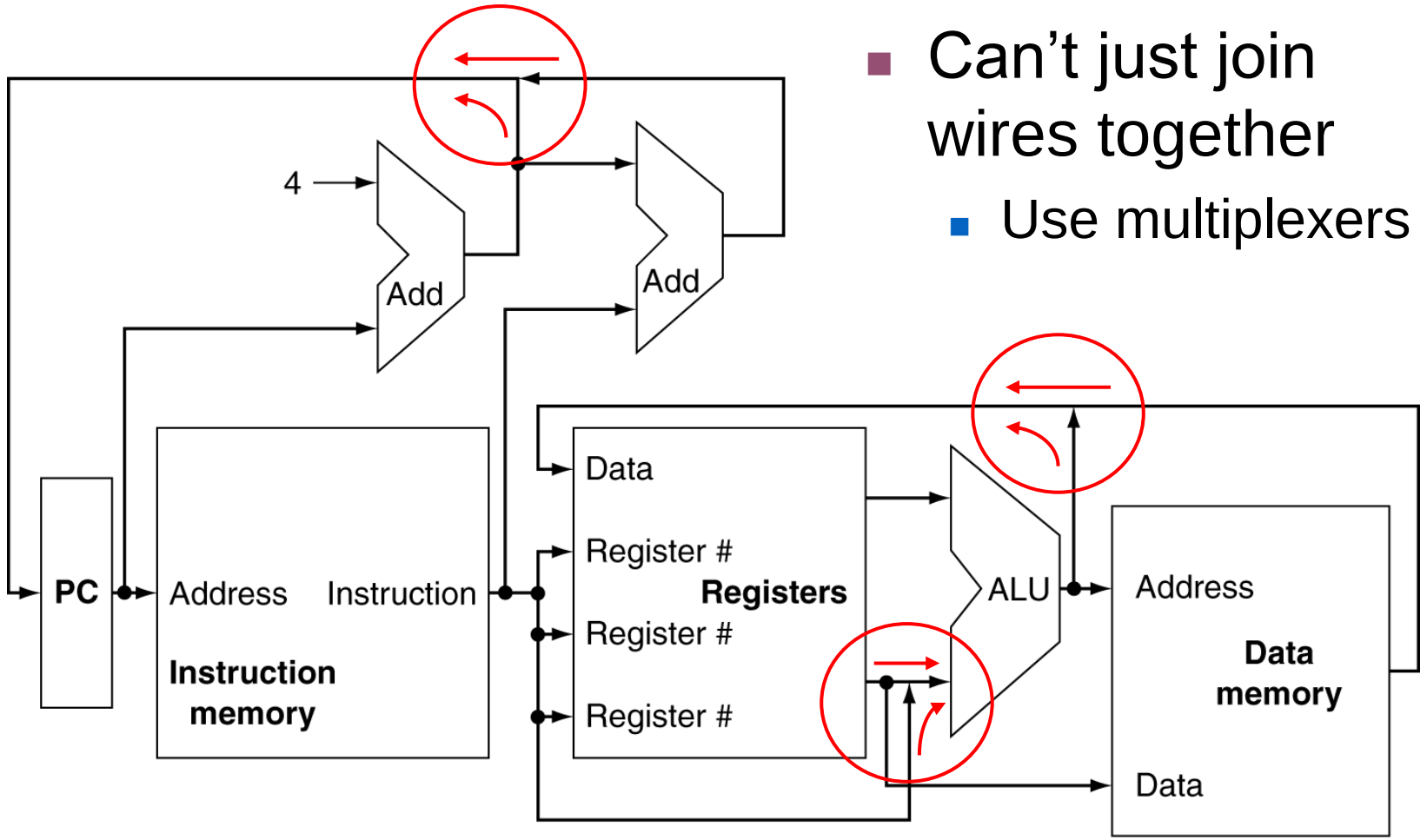
Overview of Simple CPU Design for R Format



CPU Overview

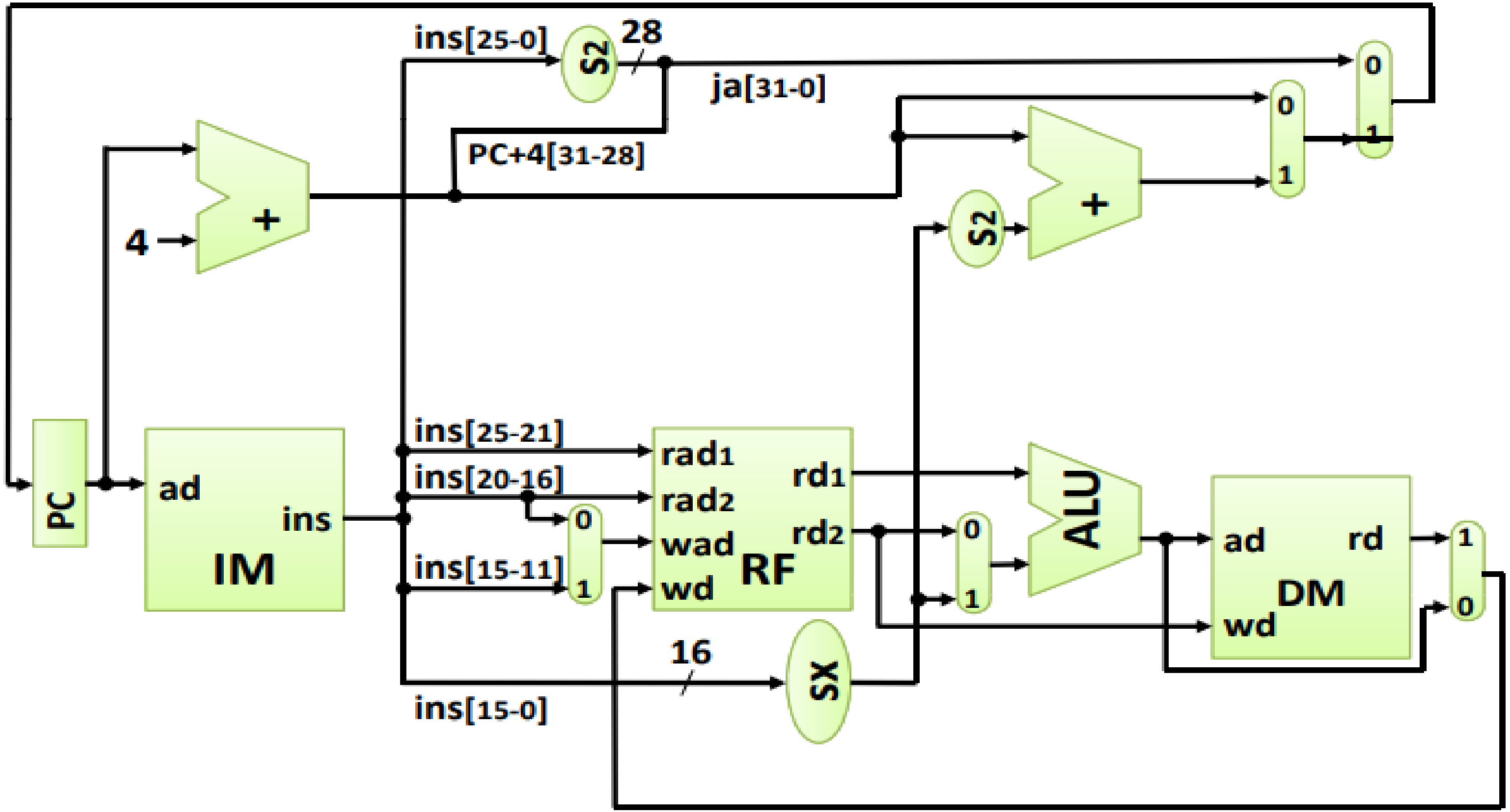


Multiplexers

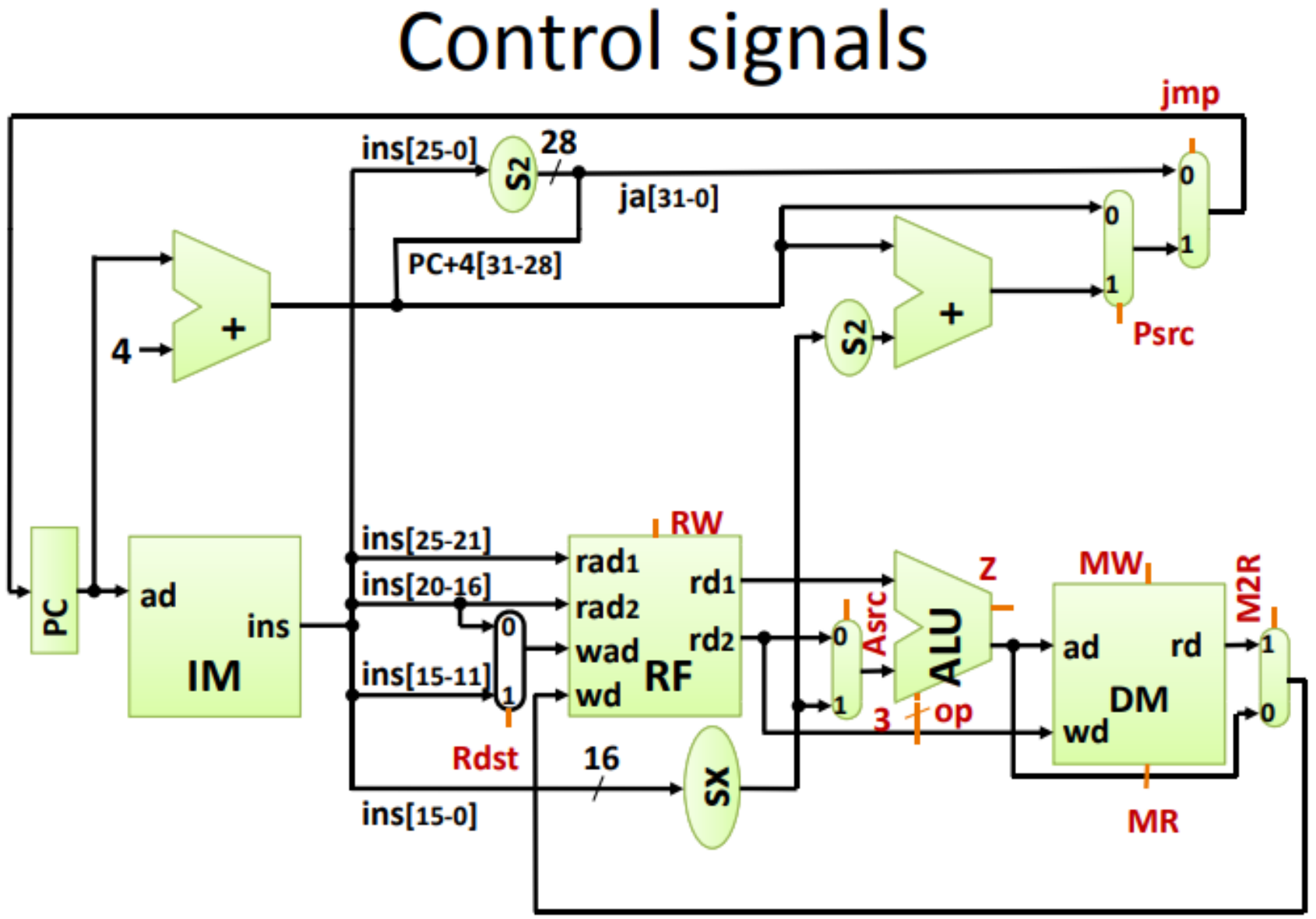


- Can't just join wires together
  - Use multiplexers

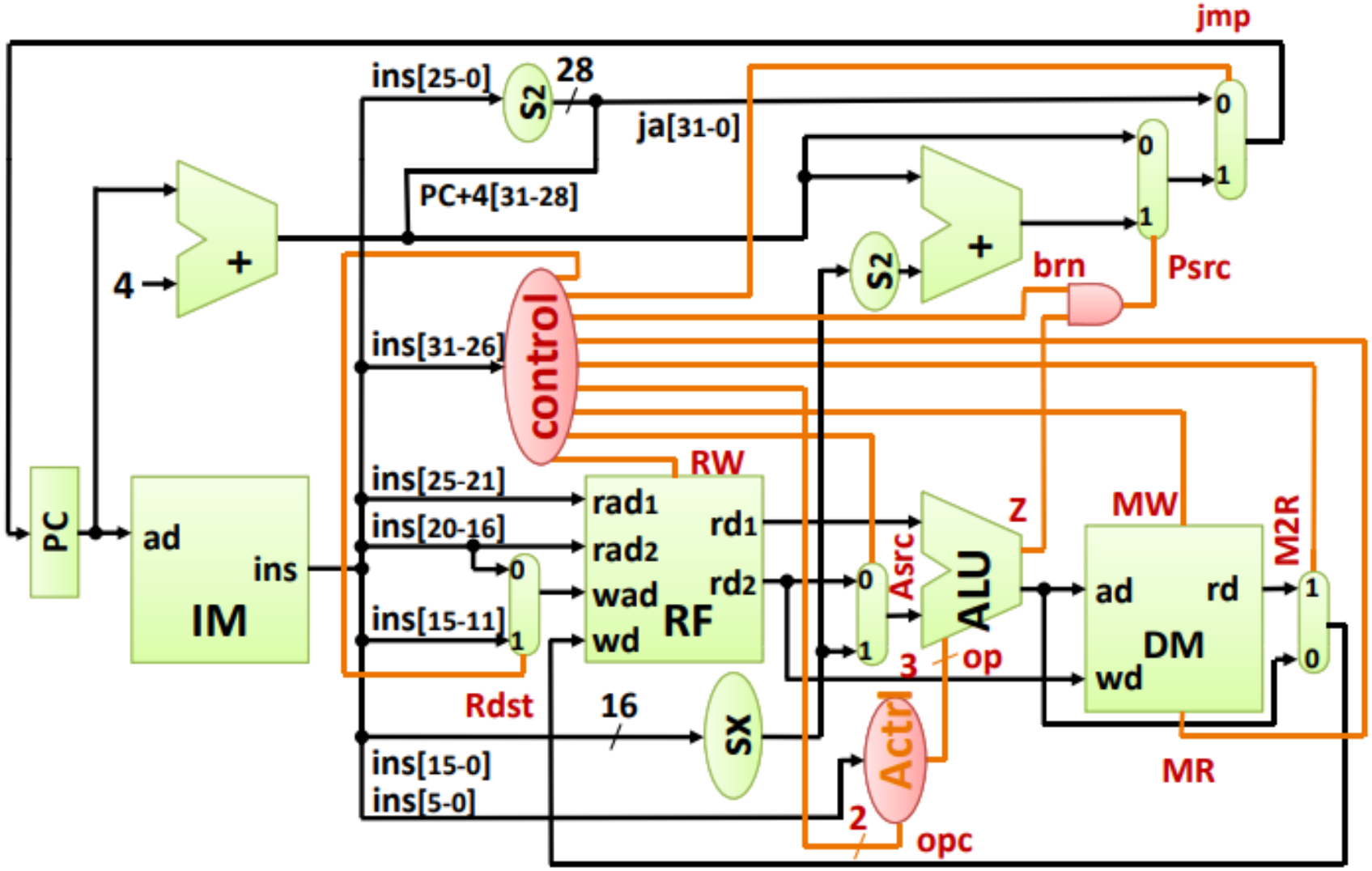
MIPS Data path, without Controls



Identify Control Signals for Simple CPU



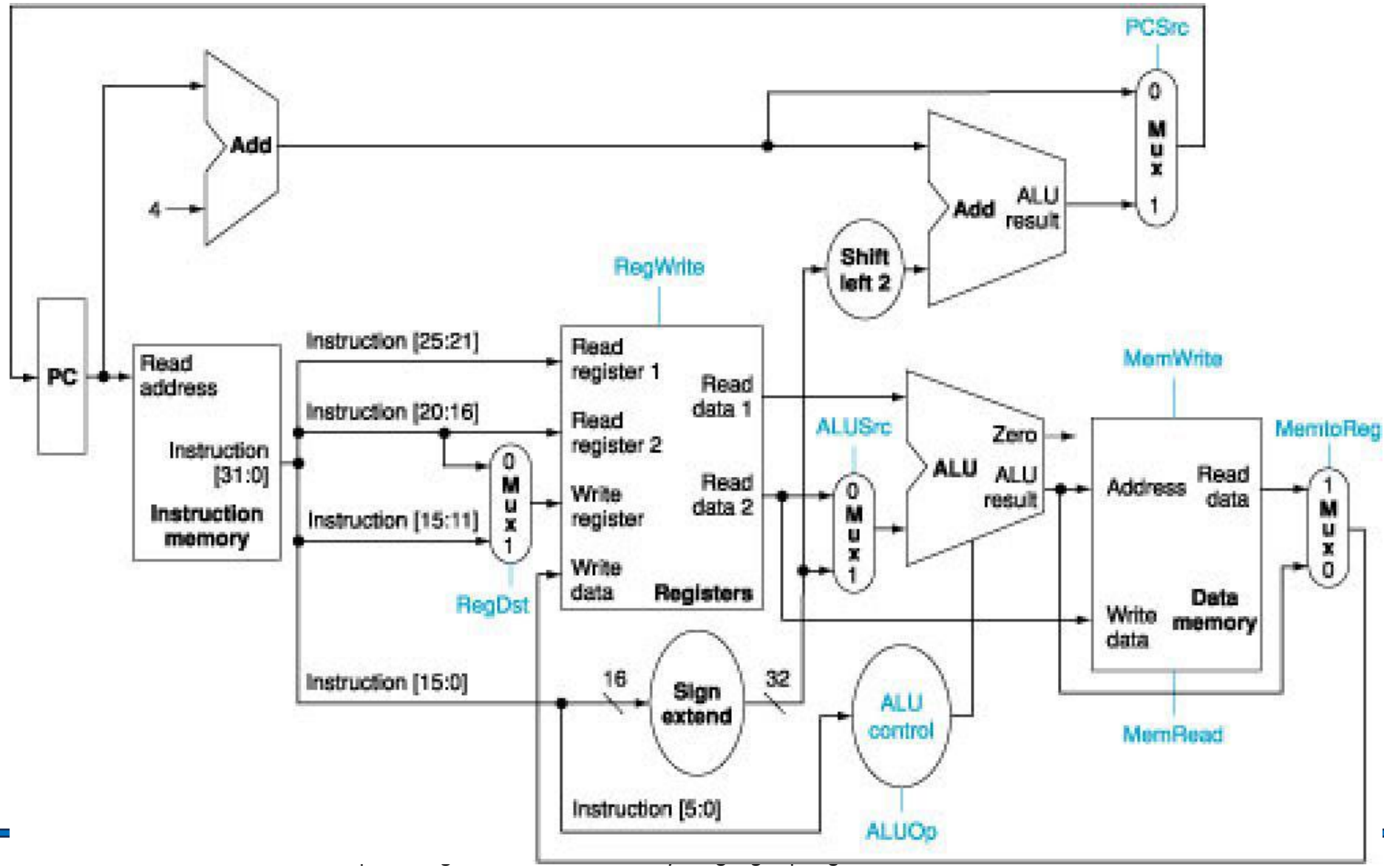
Combining Data path and Control path signals



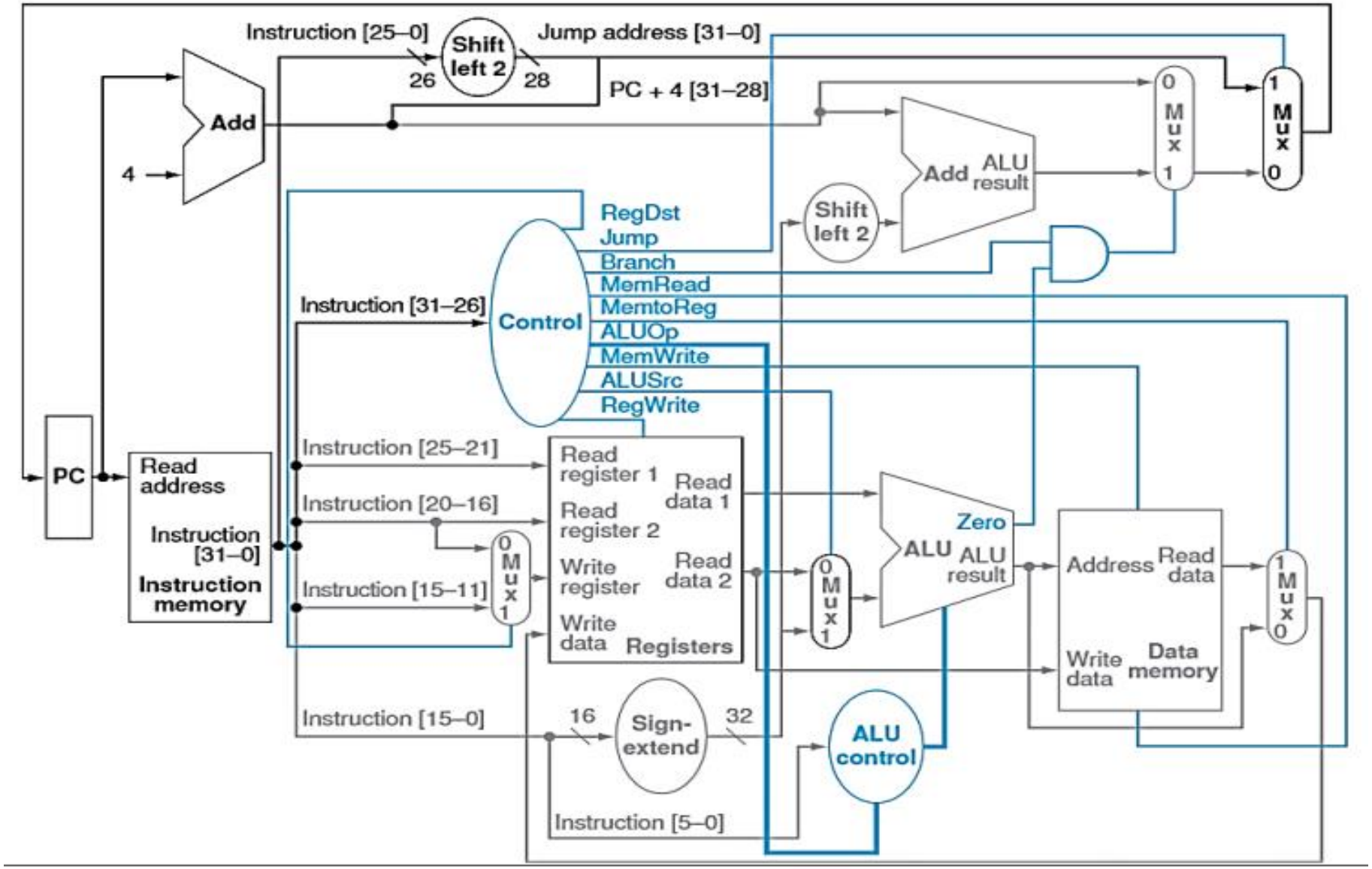
Control words in Simple MIPS CPU

Instruction	Opcode	Rdst	RW	Asrc	MW	MR	M2R	Brn	Jmp
Rtype	000000	1	1	0	0	0	0	0	0
Sw	101011	X	0	1	1	0	X	0	0
Lw	100011	0	1	1	0	1	1	0	0
Beq	000100	X	0	0	0	0	X	1	0
J	000010	X	0	X	0	0	X	X	1

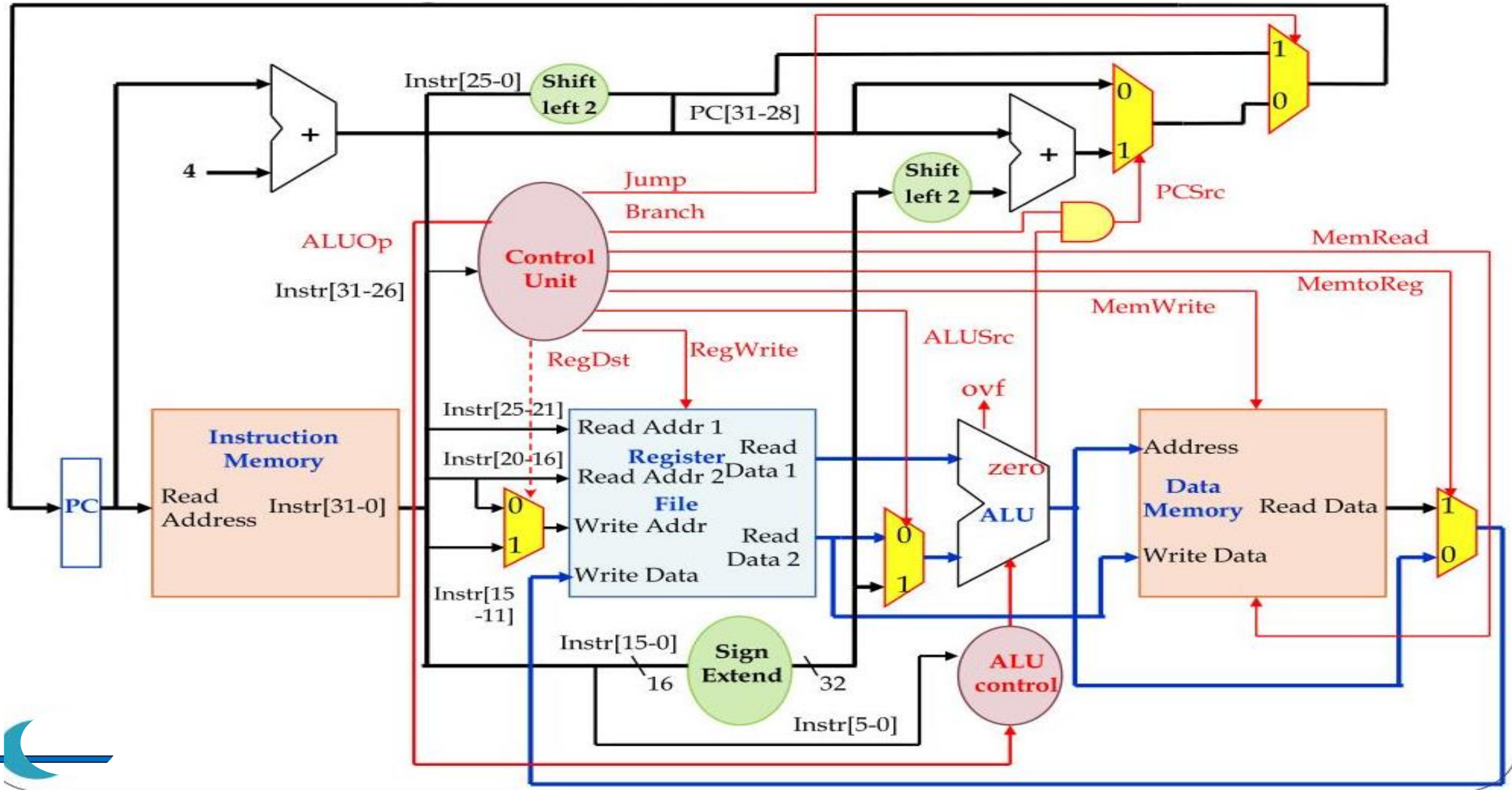
Control Signals needed in MIPS (P&H book style)



Data path with Control Signals in simple MIPS

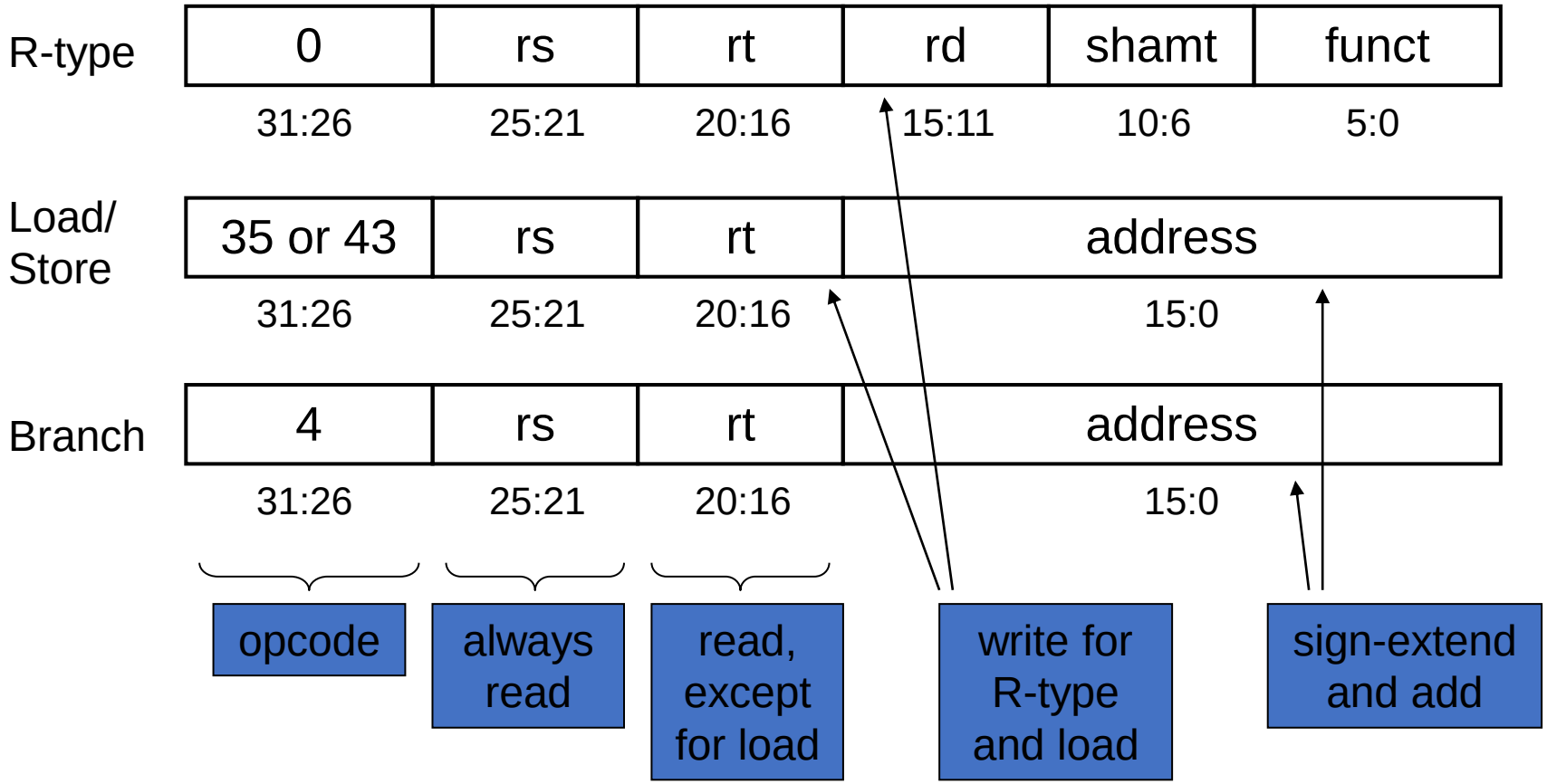


Another (clear) View of Control Signals



The Main Control Unit

- Control signals derived from instruction



Generating Control Signals

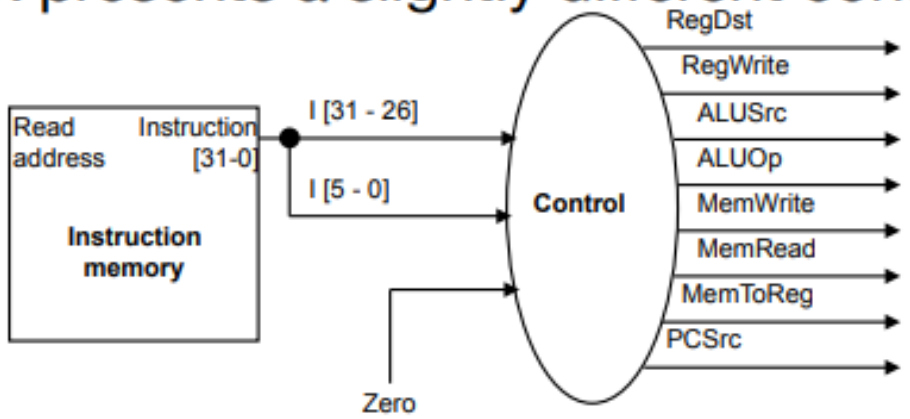
The control unit needs 13 bits of inputs

- Six bits make up the instruction's opcode
- Six bits come from the instruction's func field
- It also needs the Zero output of the ALU

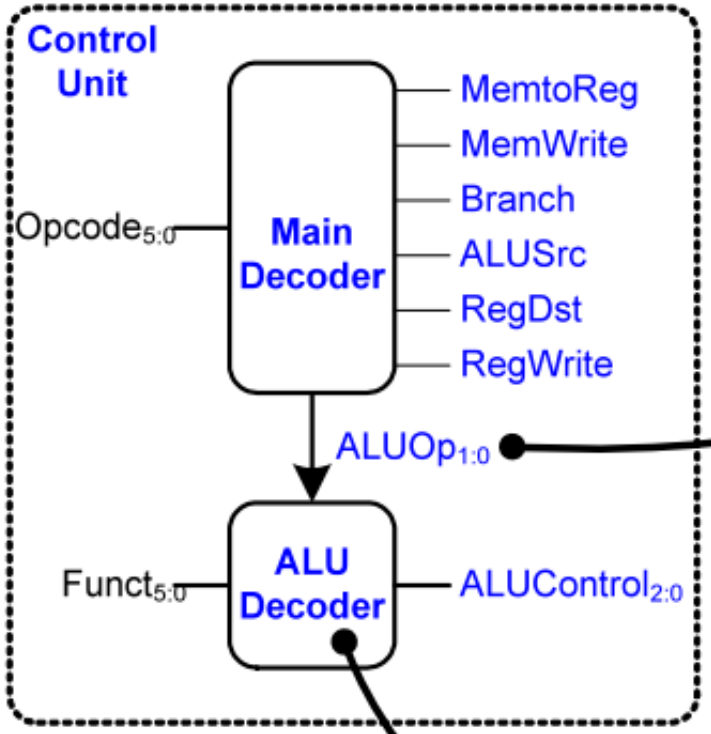
The control unit generates 10 bits of output, corresponding to the signals mentioned earlier

You can build the actual circuit by using big K-maps, big Boolean algebra, or big circuit design programs

The textbook presents a slightly different control unit



Control words with ALU Op codes



ALUOp1:0	Meaning
0 0	Add
0 1	Subtract
1 0	Look at Funct
1 1	*not used*

ALUOp1:0	Funct	ALUControl2:0
0 0	x	010 (Add)
x 1	x	110 (Subtract)
1 x	100000 (add)	010 (Add)
1 x	100010 (sub)	110 (Subtract)
1 x	100100 (and)	000 (And)
1 x	100101 (or)	001 (Or)
1 x	101010 (slt)	111 (SLT)

ALU Control

- Assume 2-bit ALUOp derived from opcode
  - Combinational logic derives ALU control

opcode	ALUOp	Operation	funct	ALU function	ALU control
lw	00	load word	XXXXXX	add	0010
sw	00	store word	XXXXXX	add	0010
beq	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
		subtract	100010	subtract	0110
		AND	100100	AND	0000
		OR	100101	OR	0001
		set-on-less-than	101010	set-on-less-than	0111

Control words for different Instructions

Control signal table

Operation	RegDst	RegWrite	ALUSrc	ALUOp	MemWrite	MemRead	MemToReg
add	1	1	0	010	0	0	0
sub	1	1	0	110	0	0	0
and	1	1	0	000	0	0	0
or	1	1	0	001	0	0	0
slt	1	1	0	111	0	0	0
lw	0	1	1	010	0	1	1
sw	X	0	1	010	1	0	X
beq	X	0	0	110	0	0	X

Delays in Data path and Timing Performance

Example values of Delays encountered in different components in MIPS Data path

Components	Delay	Example
Register	0	**0ns**
Adder	t_+	**4ns**
ALU	t_A	**5ns**
MUX	0	**0ns**
RF	t_R	**3ns**
Instruction Memory	t_I	**6ns**
Data Memory	t_m	**6ns**
Bit manipulation	0	**0ns**

INS	Delay
R	17ns
SW	20ns
LW	23ns
Beq	14ns
J	6ns