MicroMIPS Single-Cycle Data Path

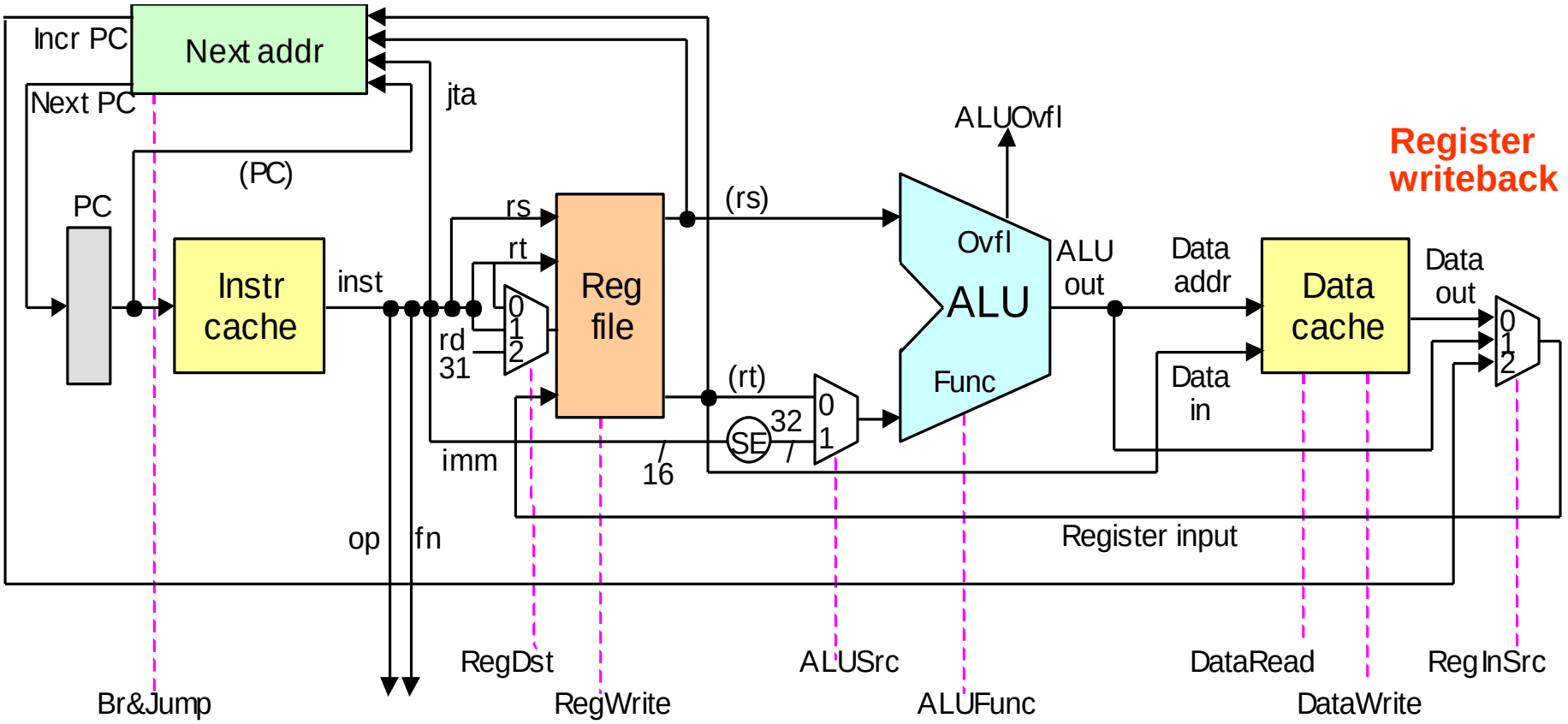

Fig. 13.3 Key elements of the single-cycle MicroMIPS data path.

Control Signals

Table 13.2 Control signals for the single-cycle MicroMIPS implementation.

	Control signal	0	1	2	3
Reg file	RegWrite	Don't write	Write		
	$RegDst_1$, $RegDst_0$	rt	rd	$31	
	$RegInSrc_1$, $RegInSrc_0$	Data out	ALU out	IncrPC	
ALU	ALUSrc	(rt)	imm		
	Add′Sub	Add	Subtract		
	$LogicFn_1$, $LogicFn_0$	AND	OR	XOR	NOR
	$FnClass_1$, $FnClass_0$	lui	Set less	Arithmetic	Logic
Data cache	DataRead	Don't read	Read		
	DataWrite	Don't write	Write		
Next addr	$BrType_1$, $BrType_0$	No branch	beq	bne	bltz
	$PCSrc_1$, $PCSrc_0$	IncrPC	jta	(rs)	SysCallAddr

MIPS Instruction Formats

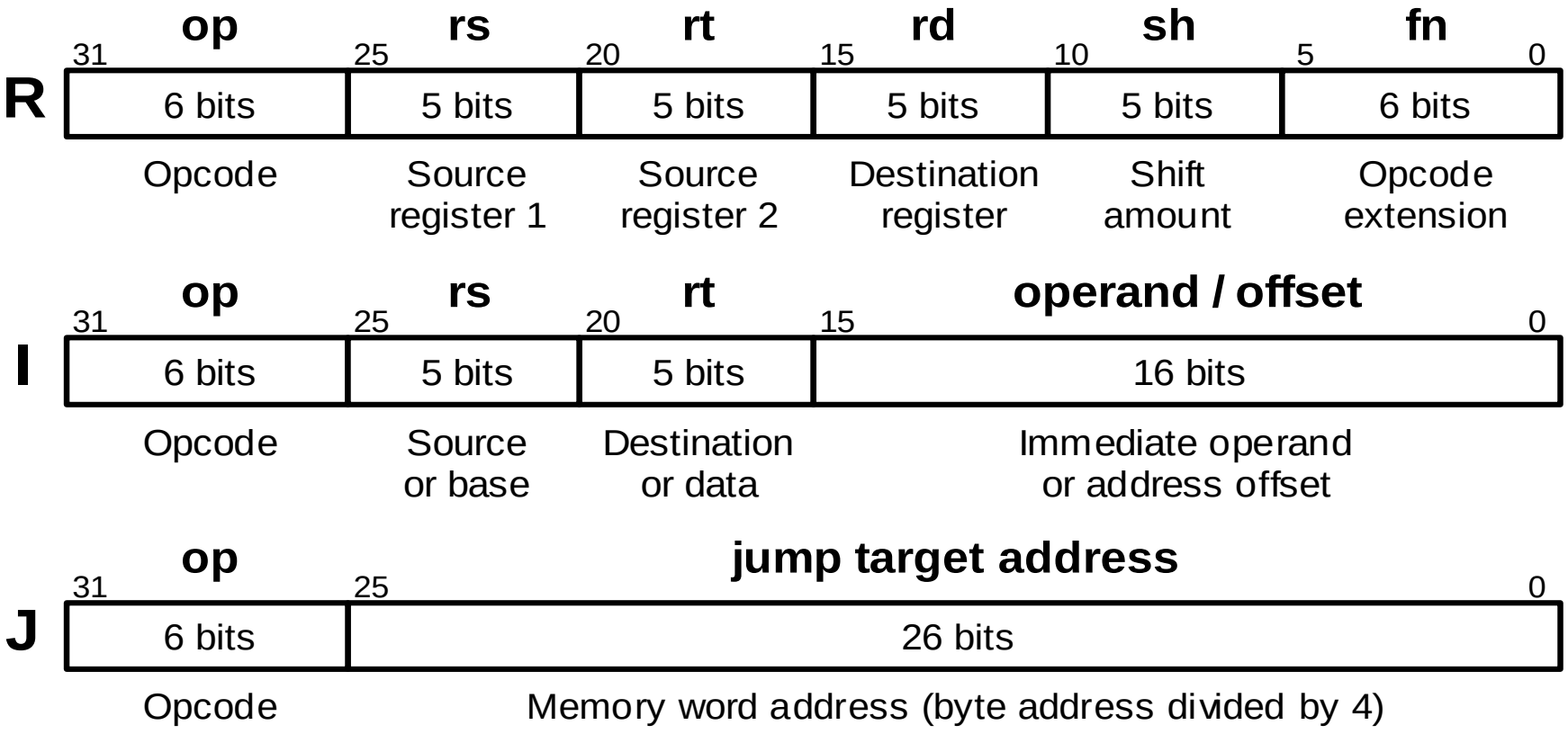

Figure 5.4 MicroMIPS instructions come in only three formats: register (R), immediate (I), and jump (J).

The MicroMIPS Instruction Set

	Instruction	Usage	op	fn
Copy	Load upper immediate	`lui rt,imm`	15	
Arithmetic	Add	`add rd,rs,rt`	0	32
	Subtract	`sub rd,rs,rt`	0	34
	Set less than	`slt rd,rs,rt`	0	42
	Add immediate	`addi rt,rs,imm`	8	
	Set less than immediate	`slti rt,rs,imm`	10	
Logic	AND	`and rd,rs,rt`	0	36
	OR	`or rd,rs,rt`	0	37
	XOR	`xor rd,rs,rt`	0	38
	NOR	`nor rd,rs,rt`	0	39
	AND immediate	`andi rt,rs,imm`	12	
	OR immediate	`ori rt,rs,imm`	13	
	XOR immediate	`xori rt,rs,imm`	14	
Memory access	Load word	`lw rt,imm(rs)`	35	
	Store word	`sw rt,imm(rs)`	43	
Control transfer	Jump	`j L`	2	
	Jump register	`jr rs`	0	8
	Branch less than 0	`bltz rs,L`	1	
	Branch equal	`beq rs,rt,L`	4	
	Branch not equal	`bne rs,rt,L`	5	
	Jump and link	`jal L`	3	
	System call	`syscall`	0	12

Table 13.1