MIPS Instruction Implementation Types

Instruction Type	Example		Instruction Coding
	ALU Usage		
Non-Jump R-Type	add rd, rs, rt	R	31 op 26 | 25 rs 21 | 20 rt 16 | 15 rd 11 | 10 sa 6 | 5 fn 0
	The ALU performs the operation indicated by the mnemonic, which is coded into the fn field.		
Immediate	addi rt, rs, imm	I	31 op 26 | 25 rs 21 | 20 rt 16 | 15 imm 0
	The ALU performs the operation indicated by the mnemonic, which is coded into the op field.		
Branch	beq $rs, $rt, imm	I	31 op 26 | 25 rs 21 | 20 rt 16 | 15 imm 0
	The ALU subtracts rt from rs for comparison.		
Load	lw rt, imm(rs)	I	31 op 26 | 25 rs 21 | 20 rt 16 | 15 imm 0
	The ALU adds rs and imm to get the address.		
Store	sw rt, imm(rs)	I	31 op 26 | 25 rs 21 | 20 rt 16 | 15 imm 0
	The ALU adds rs and imm to get the address.		
Non-Register Jump	jal target	J	31 op 26 | 25 target 0
	The ALU is not used.		
Jump Register	jalr rd, rs	R	31 op 26 | 25 rs 21 | 20 rt 16 | 15 rd 11 | 10 sa 6 | 5 fn 0
	The ALU is not used.		

MIPS Single-Cycle Control Signals

Activity	Signal	Purpose
PC Update	Branch	Combined with a condition test boolean to enable loading the branch target address into the PC.
	Jump	Enables loading the jump target address into the PC (only appears in Figure 4.24 in Patterson and Hennessey).
Source Operand Fetch	ALUSrc	Selects the second source operand for the ALU (rt or sign-extended immediate field in Patterson and Hennessey).
ALU Operation	ALUOp	Either specifies the ALU operation to be performed or specifies that the operation should be determined from the function bits.
Memory Access	MemRead	Enables a memory read for load instructions.
	MemWrite	Enables a memory write for store instructions.
Register Write	RegWrite	Enables a write to one of the registers.
	RegDst	Determines how the destination register is specified (rt or rd in Patterson and Hennessey).
	MemtoReg	Determines where the value to be written comes from (ALU result or memory in Patterson and Hennessey).