mips-harvard: Summary of supported CPU Instructions

Assembly	Name	Action	Opcode					
		Arithmetic Operations						
ADD rd,rs,rt	Add	rd=rs+rt	000000	rs	rt	rd	00000	100000
ADDI rd,rs,imm	Add Immediate	rt=rs+SE(imm)	001000	rs	rt	imm		
SUB rd,rs,rt	Subtract	rd=rs-rt	000000	rs	rt	rd	00000	100010
LUI rt,imm	Load Upper Immediate	rt=imm<<16	001111	rs	rt	imm		
		Bitwise Operations						
AND rd,rs,rt	And	rd=rs & rt	000000	rs	rt	rd	00000	100100
NOR rd,rs,rt	Not-Or	Rd=~(rs | rt)	000000	rs	rt	rd	00000	100111
OR rd,rs,rt	Or	rd=rs | rt	000000	rs	rt	rd	00000	100101
ORI rd,rs,imm	Or Immediate	rt=rs | ZE(imm)	001101	rs	rt	imm		
		Bit shift						
SLL rd,rt,sha	Shift Left Logical	rd=rt << sha	000000	rs	rt	rd	sha	000000
SRA rd,rt,sha	Shift Right Arithmetic	rd=rt >> sha	000000	rs	rt	rd	sha	000011
SRL rd,rt,sha	Shift Right Logical	rd=rt >>> sha	000000	rs	rt	rd	sha	000010
		Comparison						
SLT rd,rs,rt	Set if Less Than	rd=1, if rs < rt, else rd=0	000000	rs	rt	rd	00000	101010
		Branching and Jumping						
BEQ rs,rt,offset	Branch if EQual	if(rs==rt), pc=pc+SE(offset)*4	000100	rs	rt	offset		
BNE rs,rt,offset	Branch if Not Equal	if(rs ≠ rt), pc=pc+SE(offset)*4	000101	rs	rt	offset		
J target	Jump to target	pc=pc[31:28] | target*4	000010	target				
JAL target	Jump And Link	r31=pc, then pc=pc[31:28] |target*4	000011	target				
JR rs	Jump Register	pc=rs	000000	rs	000000	000000	00000	001000
		Memory access						
LW rt,offset(rs)	Load Word	load memory word from the address (offset+rs) into rt	000100	rs	rt	offset		
SW rt,offset(rs)	Store Word	store content of rt into memory word at the address (offset+rs)	000101	rs	rt	offset		

Notes:

1. ZE(imm) means “Zero-extended immediate”, SE(imm) means “Sign-extended immediate”
2. Some arithmetic commands have “unsigned” variants (ADDU, SUBU, ADDIU, SLTU). The only difference between signed and unsigned versions is that unsigned versions do not trigger arithmetic overflow exception.