Table 1: Subset of MIPS Instruction Set

Class	Instruction	Usage	Meaning	op	fn
Arithmetic	Add	`add rd, rs, rt`	$rd \leftarrow (rs) + (rt)$; with overflow	0	32
	Add unsigned	`addu rd, rs, rt`	$rd \leftarrow (rs) + (rt)$; no overflow	0	33
	Subtract	`sub rd, rs, rt`	$rd \leftarrow (rs) - (rt)$; with overflow	0	34
	Subtract unsigned	`subu rd, rs, rt`	$rd \leftarrow (rs) - (rt)$; no overflow	0	35
	Set less than	`slt rd, rs, rt`	$rd \leftarrow$ if $(rs) < (rt)$ then 1 else 0	0	42
	Add immediate	`addi rt, rs, imm`	$rt \leftarrow (rs) +$ imm; with overflow	8	
	Add immediate unsigned	`addiu rt, rs, imm`	$rt \leftarrow (rs) + imm$; no overflow	9	
	Set less than immediate	`slti rt, rs, imm`	$rt \leftarrow$ if $(rs) < imm$ then1 else 0	10	
Shift	Shift left logical	`sll rd, rt, sh`	$rd \leftarrow (rt)$ left-shifted by sh	0	0
	Shift right logical	`srl rd, rt, sh`	$rd \leftarrow (rt)$ right-shifted by sh	0	2
	Shift left logical variable	`sllv rd, rt, rs`	$rd \leftarrow (rt)$ left-shifted by (rs)	0	4
	Shift right logical variable	`srlv rd, rt, rs`	$rd \leftarrow (rt)$ right-shifted by (rs)	0	6
Logic	AND	`and rd, rs, rt`	$rd \leftarrow (rs)$ AND (rt)	0	36
	OR	`or rd, rs, rt`	$rd \leftarrow (rs)$ OR (rt)	0	37
	XOR	`xor rd, rs, rt`	$rd \leftarrow (rs)$ XOR (rt)	0	38
	NOR	`nor rd, rs, rt`	$rd \leftarrow ((rs)$ OR $(rt))'$	0	39
	AND Immediate	`andi rt, rs, imm`	$rt \leftarrow (rs)$ AND (imm)	12	
	OR Immediate	`ori rt, rs, imm`	$rt \leftarrow (rs)$ OR (imm)	13	
	XOR Immediate	`xori rt, rs, imm`	$rt \leftarrow (rs)$ XOR (imm)	14	
Copy	Load upper immediate	`lui rt,imm`	$rt \leftarrow \{imm, 0x0000\}$	15	
Memory access	Load word	`lw rt, imm(rs)`	$rt \leftarrow mem[(rs) + imm]$	35	
	Load byte	`lb rt, imm(rs)`	Load byte-0, sign-extend	32	
	Load byte unsigned	`lbu rt, imm(rs)`	Load byte-0, zero-extend	36	
	Store word	`sw rt, imm(rs)`	$mem[(rs) + imm] \leftarrow rt$	43	
	Store byte	`sb rt, imm(rs)`	Store byte-0	40	
Control transfer	Jump	`j L`	goto L	2	
	Jump and link	`jal L`	goto L; \$31 $\leftarrow PC + 4$	3	
	Jump register	`jr rs`	goto (rs)	0	8
	Branch less than 0	`bltz rs, L`	if $(rs) < 0$ then goto L	1	
	Branch equal	`beq rs, rt, L`	if $(rs) = (rt)$ then goto L	4	
	Branch not equal	`bne rs, rt, L`	if $(rs) \neq (rt)$ then goto L	5	

Figure 1: MIPS Instruction Formats