

# PLP 3 Quick Reference

R-Type Arithmetic Instructions	
addu \$rd, \$rs, \$rt	rd = rs + rt
subu \$rd, \$rs, \$rt	rd = rs - rt
and \$rd, \$rs, \$rt	rd = rs & rt
or \$rd, \$rs, \$rt	rd = rt   rt
nor \$rd, \$rs, \$rt	rd = ~(rs   rt)
slt \$rd, \$rs, \$rt	rd = (rs < rt) ? 1 : 0
sltu \$rd, \$rs, \$rt	rd = (rs < rt) ? 1 : 0
sll \$rd, \$rt, shamt	rd = rt << shamt
srl \$rd, \$rt, shamt	rd = rt >> shamt
mullo \$rd, \$rs, \$rt	rd = rs * rt
mulhi \$rd, \$rs, \$rt	rd = rs * rt (high bits)

R-Type Jump Instructions	
jr \$rs	PC = rs
jalr \$rd, \$rs	rd = PC + 4; PC = rs

J-Type Instructions	
j label	PC = label
jal label	ra = PC + 4; PC = label

I-Type Arithmetic Instructions	
addiu \$rt, \$rs, imm	rt = rs + SignExtend(imm)
andiu \$rt, \$rs, imm	rt = rs & ZeroExtend(imm)
ori \$rt, \$rs, imm	rt = rs   ZeroExtend(imm)
slti \$rt, \$rs, imm	rt = (rs < SignExtend(imm)) ? 1 : 0
sltiu \$rt, \$rs, imm	rt = (rs < SignExtend(imm)) ? 1 : 0
lui \$rt, imm	rt = imm << 16

Load and Store Instructions	
lw \$rt, imm(\$rs)	rt = SignExtend(imm) + rs
sw \$rt, imm(\$rs)	SignExtend(imm) + rs = rt

Pseudo Instructions	
nop	No operation
b label	unconditional branch
move \$rd, \$rs	register copy
li \$rd, imm	load immediate (32-bit)
li \$rd, label	load pointer (32-bit)
push \$rt	push rt onto stack and decrement
pop \$rt	pop from stack onto rt

Branch Instructions	
beq \$rt, \$rs, label	if (rt == rs) PC = PC + 4 + imm
bne \$rt, \$rs, label	if (rt != rs) PC = PC + 4 + imm

Memory Map	
0x00000000	boot ROM
0x10000000	RAM
0xf0000000	UART
0xf0100000	switches
0xf0200000	LEDs
0xf0300000	GPIO
0xf0400000	VGA
0xf0500000	PLPID
0xf0600000	timer
0xf0700000	interrupt controller
0xf0800000	performance counters
0xf0a00000	seven segment

Assembler directives	
.org <address>	Load following at address
label:	Label current location
.word <value>	Write 32-bit value
.ascii "string"	Place string at current address
.asciiz "string"	Place null-terminated string
.space <n>	Reserve n words