

ISA

Instruction	Opcode (ALU op)	Type	Operation
add \$rd, \$rs, \$rt	00000 (00000)	R	\$rd = \$rs + \$rt \$rstatus = 1 if overflow
addi \$rd, \$rs, N	00101	I	\$rd = \$rs + N \$rstatus = 2 if overflow
sub \$rd, \$rs, \$rt	00000 (00001)	R	\$rd = \$rs - \$rt \$rstatus = 3 if overflow
and \$rd, \$rs, \$rt	00000 (00010)	R	\$rd = \$rs & \$rt
or \$rd, \$rs, \$rt	00000 (00011)	R	\$rd = \$rs \$rt
sll \$rd, \$rs, shamt	00000 (00100)	R	\$rd = \$rs << shamt
sra \$rd, \$rs, shamt	00000 (00101)	R	\$rd = \$rs >>> shamt
mul \$rd, \$rd, \$rt	00000 (00110)	R	\$rd = \$rs * \$rt \$rstatus = 4 if overflow
div \$rd, \$rs, \$rt	00000 (00111)	R	\$rd = \$rs / \$rt \$rstatus = 5 if exception
sw \$rd, N(\$rs)	00111	I	MEM[\$rs + N] = \$rd
lw \$rd, N(\$rs)	01000	I	\$rd = MEM[\$rs + N]
j T	00001	JJ	PC = T
bne \$rd, \$rs, N	00010	I	if (\$rd != \$rs) PC = PC + 1 + N
jal T	00011	JJ	\$r31 = PC + 1, PC = T
jr \$rd	00100	JII	PC = \$rd
blt \$rd, \$rs, N	00110	I	if (\$rd < \$rs) PC = PC + 1 + N
bex T	10110	JJ	IF (\$rstatus != 0) PC = T
setx T	10101	JJ	\$rstatus = T
custom_r \$rd, \$rs, \$rt	00000 (01000 - 11111)	R	\$rd = custom_r(\$rs, \$rt) (n.b. For use on Final Project)
custom ...	xxxxxx+	X	Whatever custom instructions you need for your Final Project

Instruction Machine Code Format

Instruction Type	Instruction Format						
R							
	Opcode [31:27]	\$rd [26:22]	\$rs [21:17]	\$rt [16:12]	shamt [11:7]	ALU op [6:2]	Zeroes [1:0]
I							
	Opcode [31:27]	\$rd [26:22]	\$rs [21:17]	Immediate (N) [16:0]			
JI							
	Opcode [31:27]	Target (T) [26:0]					
JII							
	Opcode [31:27]	\$rd [26:22]	Zeroes [21:0]				

ISA Clarifications

1. I-type immediate field [16:0] (N) is signed and is sign-extended to a signed 32-bit integer
2. Jll-type target field [26:0] (T) is unsigned. PC and STATUS registers' upper bits [31:27] are guaranteed to never be used