Instruction	Encoding	Sample Instruction	Sample Encoding	Sample Explanation	Other Comments
add	aaaa dddd ssss tttt	add R3, R2, R1	0x0321 == 0000 0011 0010 0001	R3 <= R2 + R1	Saturating arithmetic.
addz		addz R6, R5, R4	0x1654 == 0001 0110 0101 0100	R6 <= R5 + R4 only if Z=1	
sub		sub R9, R8, R7	0x2987 == 0010 1001 1000 0111	R9 <= R8 – R7	Updates the Z, V and N flag registers.
and		and R12, R11, R10	0x3CBA == 0011 1100 1011 1010	R12 <= R11 & R10	Updates the Z flag register.
nor		nor R15, R14, R13	0x4FED == 0100 1111 1110 1101	R15 <= ~(R14 | R13)	
sll	aaaa dddd ssss iiii	sll R1, R0, 14	0x510E == 0101 0001 0000 1110	R1 <= R0 << 14	Unsigned 4-bit immediate in range [0, 15]
srl		srl R3, R2, 1	0x6321 == 0110 0011 0010 0001	R3 <= R2 >> 1	
sra		sra R5, R4, 3	0x7543 == 0111 0101 0100 0011	R5 <= R4 >>> 3	Updates the Z flag register.
lw	aaaa tttt ssss oooo	lw R7, R6, 5	0x8765 == 1000 0111 0110 0101	R7 <= mem[R6 + 5]	Signed 4-bit offset in two's complement
sw		sw R15, R14, 13	0x9FED == 1001 1111 1110 1101	mem[R14 + 13] <= R15	
lhb	aaaa dddd iiii iiii	lhb R13, 12	0xAD0C == 1010 1101 0000 1100	R13 <= {12, R13[7:0]}	Signed 8-bit immediate in two's complement
llb		llb R12, 11	0xBC0B == 1011 1100 0000 1011	R12 <= sign-extend{11}	
b					
neq	aaaa ccco oooo oooo	b neq, label	0xC??? == 1100 000? ???? ????	Branch if Z=0	Signed 9-bit offset in two's complement (? ???? ????) Branch target address = (Address of branch instruction + 2) + (2 * offset) PC holds byte addresses, each instruction is 2 bytes wide, offset is specified as the number of instructions with respect to the instruction following the branch instruction.
eq		b eq, label	0xC??? == 1100 001? ???? ????	Branch if Z=1	
gt		b gt, label	0xC??? == 1100 010? ???? ????	Branch if {Z,N}==2'b00	
lt		b lt, label	0xC??? == 1100 011? ???? ????	Branch if N=1	
gte		b gte, label	0xC??? == 1100 100? ???? ????	Branch if N=0	
lte		b lte, label	0xC??? == 1100 101? ???? ????	Branch if N=1 or Z=1	
ovfl		b ovfl, label	0xC??? == 1100 110? ???? ????	Branch if V=1	
uncond		b uncond, label	0xC??? == 1100 111? ???? ????	Branch unconditionally	
jal	aaaa oooo oooo oooo	jal label	0xD??? == 1101 ???? ???? ????	R15 <= (Address of jal instruction + 2) Jump to target address	Signed 12-bit offset in two's complement (???? ???? ????) Jump target address = (Address of jump instruction + 2) + (2 * offset)
jr	aaaa 0000 tttt 0000	jr R15	0xE0F0 == 1110 0000 1111 0000	Jump to target address given by contents of R15	Can be used to return from function calls (jal)
hlt	aaaa 0000 0000 0000	hlt	0xF000 == 1111 0000 0000 0000	Halt the processor	Completes execution of all instructions ahead of the halt instruction, freezes the PC at the address of the instruction following the halt instruction and does not execute any instruction(s) following the halt instruction.
Other Notes					
1. Flag registers are Z-zero, V-overflow, N-negative/sign. 2. The overflow flag denotes positive overflow as well as negative underflow. 3. Register R0 is hard-wired to 0x0000, cannot be written to. 4. jal instruction always stores the return address in register R15. Do not write R15 inside function calls if you wish to return.					