The instruction and its 32-bit instruction format are shown below:
lw destinationReg, offset [sourceReg] (Sign extends data specified in instruction field (15:0) to 32-bits, add it with register specified by register number in rs field and store the data in rt from memory address [rs+offset]. Opcode for lw is 100011).

op	rs	rt	offset
6 bits (31-26)	5-bits (25-21)	5-bits (20-16)	16-bits (15-0)

sw sourceReg1, offset [sourceReg2] (Sign extends data specified in instruction field (15:0) to 32-bits, add it with register specified by register number in rs field. store the data from the reg rt to memory address [rs+offset]. Opcode for sw is 101011).

op	rs	rt	offset
6 bits (31-26)	5-bits (25-21)	5-bits (20-16)	16-bits (15-0)

or destinationReg, sourceReg1, sourceReg2 (Perform or operation on the registers specified by registers specified by register numbers in rs and rt fields and save the result in the register specified by register specified by register number in rd field. Opcode for or is 000000 and function is 100101).

op	rs	rt	rd	shamt	funct
6 bits (31-26)	5-bits (25-21)	5-bits (20-16)	5-bits (15-11)	5-bits (10-6)	6-bits (5-0)

andi destinationReg, sourceReg, immediate (Signextends data specified in instruction field (15:0) to 32-bits, and it with register specified by register number in rs field. And store the result in rt. Opcode for andi is 001100).

op	rs	rt	offset
6 bits (31-26)	5-bits (25-21)	5-bits (20-16)	16-bits (15-0)

j target (Shift left by 2 the data specified in offset field (25:0) to 28-bits, and append the first 4 bits of PC+4. Opcode for j is 000010).

op	offset
6 bits (31-26)	26-bits (25-0)

Sample instructions used:

lw R1, R2, #10
sw R1, R3, #5
or R2, R5, R3
or R1, R6, R7
andi R1, R3, #10

Clock by clock execution of sample instructions.

		CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	C10
I1	**lw**	IF	ID	RR	EX	M	WB				
I2	**sw**		IF	ID	RR	EX	M	WB			
I3	**or**			IF	ID	RR	EX	M	WB		
I4	**or**				IF	ID	RR	EX	M	WB	
I5	**andi**					IF	ID	RR	EX	M	WB