WISC-S14 ISA

and nor sill srl aaa sra lw sw lhb lib neq eq gt lt gte lte ovfl uncond	aaa dddd ssss iiii aaa dddd ssss iiii aaa tttt ssss oooo aaaa dddd iiii iiii	add R3, R2, R1 addz R6, R5, R4 sub R9, R8, R7 and R12, R11, R10 nor R15, R14, R13 sll R1, R0, 14 srl R3, R2, 1 sra R5, R4, 3 lw R7, R6, 5 sw R15, R14, 13 lhb R13, 12 llb R12, 11 b neq, label b eq, label b gt, label b lt, label b gte, label b gte, label	0x0321 == 0000 0011 0010 0001 0x1654 == 0001 0110 0101 0100 0x2987 == 0010 1001 1000 0111 0x3CBA == 0011 1100 1011 1010 0x4FED == 0100 1111 1110 1101 0x510E == 0101 0001 0000 1110 0x6321 == 0110 0011 0010 0001 0x7543 == 0111 0101 0100 0011 0x8765 == 1000 0111 0110 0101 0x9FED == 1001 1111 1110 1101 0xAD0C == 1010 1101 0000 1100 0xBC0B == 1011 1101 0000 1000 0xBC0R == 1010 000? ???? ???? 0xC??? == 1100 001? ???? ???? 0xC??? == 1100 011? ???? ????	R3 <= R2 + R1 R6 <= R5 + R4 only if Z=1 R9 <= R8 - R7 R12 <= R11 & R10 R15 <= ~(R14 R13) R1 <= R0 << 14 R3 <= R2 >> 1 R5 <= R4 >>> 3 R7 <= mem[R6 + 5] mem[R14 + 13] <= R15 R13 <= {12, R13[7:0]} R12 <= sign-extend{11} Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	Saturating arithmetic. Updates the Z, V and N flag registers. Updates the Z flag register. Unsigned 4-bit immediate in range [0, 15] Updates the Z flag register. Signed 4-bit offset in two's complement Signed 8-bit immediate in two's complement Signed 9-bit offset in two's complement (? ???? ????
addz sub and nor sill srl aaa sw lw sw aaa lbb neq eq gt lt gte lte ovfl uncond	aaa dddd ssss iiii aaa dddd ssss iiii aaa tttt ssss oooo aaaa dddd iiii iiii	addz R6, R5, R4 sub R9, R8, R7 and R12, R11, R10 nor R15, R14, R13 sll R1, R0, 14 srl R3, R2, 1 sra R5, R4, 3 lw R7, R6, 5 sw R15, R14, 13 lhb R13, 12 llb R12, 11 b neq, label b eq, label b gt, label b lt, label b lt, label	0x1654 == 0001 0110 0101 0100 0x2987 == 0010 1001 1000 0111 0x3CBA == 0011 1100 1011 1010 0x4FED == 0100 1111 1110 1101 0x4FED == 0100 0111 1110 1101 0x510E == 0101 0001 0000 1110 0x6321 == 0110 0011 0010 0001 0x7543 == 0111 0101 0100 0011 0x8765 == 1000 0111 0110 0101 0x9FED == 1001 1111 1110 1101 0xAD0C == 1010 1101 0000 1100 0xBC0B == 1011 1100 0000 1011 0xC??? == 1100 000? ???? ???? 0xC??? == 1100 01? ???? ????	R6 <= R5 + R4 only if Z=1 R9 <= R8 - R7 R12 <= R11 & R10 R15 <= ~(R14 R13) R1 <= R0 << 14 R3 <= R2 >> 1 R5 <= R4 >>> 3 R7 <= mem[R6 + 5] mem[R14 + 13] <= R15 R13 <= {12, R13[7:0]} R12 <= sign-extend{11} Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	Updates the Z, V and N flag registers. Updates the Z flag register. Unsigned 4-bit immediate in range [0, 15] Updates the Z flag register. Signed 4-bit offset in two's complement Signed 8-bit immediate in two's complement Signed 9-bit offset in two's complement (? ???? ????
sub and and nor sill srl aaa sra lw sw aaa lhb in neq eq gt lt gte lte ovfl uncond	aaa dddd ssss iiii aaa dddd ssss iiii aaa tttt ssss oooo aaaa dddd iiii iiii	sub R9, R8, R7 and R12, R11, R10 nor R15, R14, R13 sll R1, R0, 14 srl R3, R2, 1 sra R5, R4, 3 lw R7, R6, 5 sw R15, R14, 13 lhb R13, 12 llb R12, 11 b neq, label b eq, label b gt, label b lt, label b lt, label	0x2987 == 0010 1001 1000 0111 0x3CBA == 0011 1100 1011 1010 0x4FED == 0100 1111 1110 1101 0x510E == 0101 0001 0000 1110 0x6321 == 0110 0011 0010 0001 0x7543 == 0111 0101 0100 0011 0x8765 == 1000 0111 0110 0101 0x9FED == 1001 1111 1110 1101 0xAD0C == 1010 1101 0000 1100 0xBC0B == 1011 1100 0000 1011 0xC??? == 1100 000? ???? ???? 0xC??? == 1100 01? ???? ????	R9 <= R8 - R7 R12 <= R11 & R10 R15 <= ~(R14 R13) R1 <= R0 << 14 R3 <= R2 >> 1 R5 <= R4 >>> 3 R7 <= mem[R6 + 5] mem[R14 + 13] <= R15 R13 <= {12, R13[7:0]} R12 <= sign-extend{11} Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	Updates the Z flag register. Unsigned 4-bit immediate in range [0, 15] Updates the Z flag register. Signed 4-bit offset in two's complement Signed 8-bit immediate in two's complement Signed 9-bit offset in two's complement (? ???? ????
and nor sill srl aaa sra lw sw aaa lib b neq eq gt lt gte lte ovfl uncond	aaa dddd ssss iiii aaa tttt ssss oooo aaaa dddd iiii iiii	and R12, R11, R10 nor R15, R14, R13 sll R1, R0, 14 srl R3, R2, 1 sra R5, R4, 3 lw R7, R6, 5 sw R15, R14, 13 lhb R13, 12 llb R12, 11 b neq, label b eq, label b gt, label b lt, label b lt, label	0x3CBA == 0011 1100 1011 1010 0x4FED == 0100 1111 1110 1101 0x510E == 0101 0001 0000 1110 0x6321 == 0110 0011 0010 0001 0x7543 == 0111 0101 0100 0011 0x8765 == 1000 0111 0110 0101 0x9FED == 1001 1111 1110 1101 0xAD0C == 1010 1101 0000 1100 0xBCOB == 1011 1100 0000 1011 0xC??? == 1100 000? ???? ???? 0xC??? == 1100 010? ???? ????	R12 <= R11 & R10 R15 <= ~(R14 R13) R1 <= R0 << 14 R3 <= R2 >> 1 R5 <= R4 >>> 3 R7 <= mem[R6 + 5] mem[R14 + 13] <= R15 R13 <= {12, R13[7:0]} R12 <= sign-extend{11} Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	Updates the Z flag register. Unsigned 4-bit immediate in range [0, 15] Updates the Z flag register. Signed 4-bit offset in two's complement Signed 8-bit immediate in two's complement Signed 9-bit offset in two's complement (? ???? ????
nor sill sirl aaa w sw aaa lib b neq eq gt It gte lte ovfl uncond	aaa dddd ssss iiii aaa tttt ssss oooo aaaa dddd iiii iiii	nor R15, R14, R13 sll R1, R0, 14 srl R3, R2, 1 sra R5, R4, 3 lw R7, R6, 5 sw R15, R14, 13 lhb R13, 12 llb R12, 11 b neq, label b eq, label b gt, label b lt, label b lt, label	0x4FED == 0100 1111 1110 1101 0x510E == 0101 0001 0000 1110 0x6321 == 0110 0011 0010 0001 0x7543 == 0111 0101 0100 0011 0x8765 == 1000 0111 0110 0101 0x9FED == 1001 1111 1110 1101 0xAD0C == 1010 1101 0000 1100 0xBC0B == 1011 1100 0000 1011 0xC??? == 1100 000? ???? ???? 0xC??? == 1100 010? ???? ????	R15 <= ~(R14 R13) R1 <= R0 << 14 R3 <= R2 >> 1 R5 <= R4 >>> 3 R7 <= mem[R6 + 5] mem[R14 + 13] <= R15 R13 <= {12, R13[7:0]} R12 <= sign-extend{11} Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	Unsigned 4-bit immediate in range [0, 15] Updates the Z flag register. Signed 4-bit offset in two's complement Signed 8-bit immediate in two's complement Signed 9-bit offset in two's complement (? ???? ????
sra aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	aaa dddd ssss iiii aaa tttt ssss oooo aaaa dddd iiii iiii aa ccco oooo oooo	srl R3, R2, 1 sra R5, R4, 3 lw R7, R6, 5 sw R15, R14, 13 lhb R13, 12 llb R12, 11 b neq, label b eq, label b gt, label b lt, label	0x6321 == 0110 0011 0010 0001 0x7543 == 0111 0101 0100 0011 0x8765 == 1000 0111 0110 0101 0x9FED == 1001 1111 1110 1101 0xAD0C == 1010 1101 0000 1100 0xBC0B == 1011 1100 0000 1011 0xC??? == 1100 000? ???? ???? 0xC??? == 1100 010? ???? ????	R3 <= R2 >> 1 R5 <= R4 >>> 3 R7 <= mem[R6 + 5] mem[R14 + 13] <= R15 R13 <= {12, R13[7:0]} R12 <= sign-extend{11} Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	Updates the Z flag register. Signed 4-bit offset in two's complement Signed 8-bit immediate in two's complement Signed 9-bit offset in two's complement (? ???? ????
sra aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	aaa dddd ssss iiii aaa tttt ssss oooo aaaa dddd iiii iiii aa ccco oooo oooo	srl R3, R2, 1 sra R5, R4, 3 lw R7, R6, 5 sw R15, R14, 13 lhb R13, 12 llb R12, 11 b neq, label b eq, label b gt, label b lt, label	0x6321 == 0110 0011 0010 0001 0x7543 == 0111 0101 0100 0011 0x8765 == 1000 0111 0110 0101 0x9FED == 1001 1111 1110 1101 0xAD0C == 1010 1101 0000 1100 0xBC0B == 1011 1100 0000 1011 0xC??? == 1100 000? ???? ???? 0xC??? == 1100 010? ???? ????	R3 <= R2 >> 1 R5 <= R4 >>> 3 R7 <= mem[R6 + 5] mem[R14 + 13] <= R15 R13 <= {12, R13[7:0]} R12 <= sign-extend{11} Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	Updates the Z flag register. Signed 4-bit offset in two's complement Signed 8-bit immediate in two's complement Signed 9-bit offset in two's complement (? ???? ????
sra lw sw aaa lhb llb aa b neq eq gt lt gte lte ovfl uncond	aaa tttt ssss oooo aaaa dddd iiii iiii aa ccco oooo oooo	sra R5, R4, 3 lw R7, R6, 5 sw R15, R14, 13 lhb R13, 12 llb R12, 11 b neq, label b eq, label b gt, label b lt, label	0x7543 == 0111 0101 0100 0011 0x8765 == 1000 0111 0110 0101 0x9FED == 1001 1111 1110 1101 0xAD0C == 1010 1101 0000 1100 0xBC0B == 1011 1100 0000 1011 0xC??? == 1100 000? ???? ???? 0xC??? == 1100 010? ???? ????	R5 <= R4 >>> 3 R7 <= mem[R6 + 5] mem[R14 + 13] <= R15 R13 <= {12, R13[7:0]} R12 <= sign-extend{11} Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	Signed 4-bit offset in two's complement Signed 8-bit immediate in two's complement Signed 9-bit offset in two's complement (? ???? ????
lib aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	aaa tttt ssss oooo aaaa dddd iiii iiii aa ccco oooo oooo	lw R7, R6, 5 sw R15, R14, 13 lhb R13, 12 llb R12, 11 b neq, label b eq, label b gt, label b lt, label	0x8765 == 1000 0111 0110 0101 0x9FED == 1001 1111 1110 1101 0xAD0C == 1010 1101 0000 1100 0xBC0B == 1011 1100 0000 1011 0xC??? == 1100 000? ???? ???? 0xC??? == 1100 001? ???? ???? 0xC??? == 1100 010? ???? ????	R7 <= mem[R6 + 5] mem[R14 + 13] <= R15 R13 <= {12, R13[7:0]} R12 <= sign-extend{11} Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	Signed 4-bit offset in two's complement Signed 8-bit immediate in two's complement Signed 9-bit offset in two's complement (? ???? ????
sw aaa lhb aa b neq eq gt It gte Ite ovfl uncond	aaaa tttt ssss oooo aaaa dddd iiii iiii aa ccco oooo oooo	sw R15, R14, 13 Ihb R13, 12 Ilb R12, 11 b neq, label b eq, label b gt, label b It, label	0x9FED == 1001 1111 1110 1101 0xAD0C == 1010 1101 0000 1100 0xBC0B == 1011 1100 0000 1011 0xC??? == 1100 000? ???? ???? 0xC??? == 1100 001? ???? ???? 0xC??? == 1100 010? ???? ????	mem[R14 + 13] <= R15 R13 <= {12, R13[7:0]} R12 <= sign-extend{11} Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	Signed 8-bit immediate in two's complement Signed 9-bit offset in two's complement (? ???? ????
hb aa neq eq gt It gte lte ovfl uncond	aaaa tttt ssss oooo aaaa dddd iiii iiii aaaa ccco oooo oooo	sw R15, R14, 13 Ihb R13, 12 Ilb R12, 11 b neq, label b eq, label b gt, label b It, label	0x9FED == 1001 1111 1110 1101 0xAD0C == 1010 1101 0000 1100 0xBC0B == 1011 1100 0000 1011 0xC??? == 1100 000? ???? ???? 0xC??? == 1100 001? ???? ???? 0xC??? == 1100 010? ???? ????	mem[R14 + 13] <= R15 R13 <= {12, R13[7:0]} R12 <= sign-extend{11} Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	Signed 8-bit immediate in two's complement Signed 9-bit offset in two's complement (? ???? ????
hb aaaaa gt lte ovfl uncond	aaaa dddd iiii iiii	Ihb R13, 12 Ilb R12, 11 b neq, label b eq, label b gt, label b It, label	0xAD0C == 1010 1101 0000 1100 0xBC0B == 1011 1100 0000 1011 0xC??? == 1100 000? ???? ???? 0xC??? == 1100 001? ???? ???? 0xC??? == 1100 010? ???? ????	R13 <= {12, R13[7:0]} R12 <= sign-extend{11} Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	Signed 9-bit offset in two's complement (? ???? ????
b neq eq gt lt gte lte ovfl uncond	aaaa dddd iiii iiii	b neq, label b eq, label b gt, label b It, label	0xBC0B == 1011 1100 0000 1011 0xC??? == 1100 000? ???? ???? 0xC??? == 1100 001? ???? ???? 0xC??? == 1100 010? ???? ????	R12 <= sign-extend{11} Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	Signed 9-bit offset in two's complement (? ???? ????
b neq eq gt lt gte lte ovfl uncond	aaaa dddd iiii iiii	b neq, label b eq, label b gt, label b It, label	0xBC0B == 1011 1100 0000 1011 0xC??? == 1100 000? ???? ???? 0xC??? == 1100 001? ???? ???? 0xC??? == 1100 010? ???? ????	R12 <= sign-extend{11} Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	Signed 9-bit offset in two's complement (? ???? ????
neq eq gt It gte lte ovfl uncond	aa ccco oooo oooo	b neq, label b eq, label b gt, label b lt, label	0xC??? == 1100 000? ???? ???? 0xC??? == 1100 001? ???? ???? 0xC??? == 1100 010? ???? ????	Branch if Z=0 Branch if Z=1 Branch if {Z,N}==2'b00	-
neq eq gt It gte lte ovfl uncond	aa ccco oooo oooo	b eq, label b gt, label b lt, label	0xC??? == 1100 001? ???? ???? 0xC??? == 1100 010? ???? ????	Branch if Z=1 Branch if {Z,N}==2'b00	- ` ` ` `
eq gt It aaaa gte Ite ovfl uncond	aa ccco oooo oooo	b eq, label b gt, label b lt, label	0xC??? == 1100 001? ???? ???? 0xC??? == 1100 010? ???? ????	Branch if Z=1 Branch if {Z,N}==2'b00	-
gt It aaaa gte Ite ovfl uncond	aa ccco oooo oooo	b gt, label b lt, label	0xC??? == 1100 010? ???? ????	Branch if {Z,N}==2'b00	- ` ` ` `
It aaaa gte Ite ovfl uncond	aa ccco oooo oooo	b It, label		` '	Branch target address =
gte Ite ovfl uncond			0xC??? == 1100 011? ???? ????	Danasa ala if NIII	
Ite ovfl uncond		h ata Jahal		Branch if N=1	(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.
ovfl uncond	f	b gle, label	0xC??? == 1100 100? ???? ????	Branch if N=0	
uncond		b Ite, label	0xC??? == 1100 101? ???? ????	Branch if N=1 or Z=1	
		b ovfl, label	0xC??? == 1100 110? ???? ????	Branch if V=1	
al aaaa		b uncond, label	0xC??? == 1100 111? ???? ????	Branch unconditionally	
al aaaa					
	aa oooo oooo oooo	jal label		R15 <=	Signed 12-bit offset in two's complement (???? ???? ????) Jump target address = (Address of jump instruction + 2) + (2 * offset)
			0xD??? == 1101 ???? ???? ????	(Address of jal instruction + 2)	
				Jump to target address	
				bump to target address	
					(Fitalises enjamp mediastion 2) (2 enest)
ir aaaa 0000 tttt 0000	aaa 0000 tttt 0000	ir R15	0×E0F0 == 1110 0000 1111 0000	Jump to target address given	Can be used to return from function calls (jal)
		j	V.20.0 2222 0000	by contents of R15	
nlt aaaa	22 0000 0000 0000	hlt	0,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	Halt the processor	
iii dddd	aaaa 0000 0000 0000	TIIL	0×F000 == 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					
		N Co			
1. Flag registers are	re Z-zero, V-overflow	, in-negative/sign.	ivo undorflow		
2. THE OVERTION HAS		verflow as well as negatical cannot be written to.	live undernow.		
5. Register R0 is rial 4. jal instruction alwa	ard wired to 0v0000	Calliol DE Willeli IO.			