RISC_NET ISA

Instructions

1. Nop
2. Load
3. Load high
4. Load low
5. load pc
6. Store
7. Store high
8. Store low
9. Addition
10. Subtraction
11. Multiplication
12. And
13. Or
14. Not
15. Xor
16. Xnor
17. Left shift
18. Right shift
19. Rotate right
20. Rotate left
21. Set flags
22. Read flags
23. Compare
24. Jump
25. Jump carry
26. Jump equal
27. Jump less than
28. Jump less than or equal
29. Jump greater than
30. Jump greater than or equal

Registers

Register	Mnemonics	Encoding
Primary Accumulator	R1	0000
Secondary Accumulator	R2	0001
General purpose	R3	0010
General purpose	R4	0011
General purpose	R5	0100
General purpose	R6	0101
General purpose	R7	0110
General purpose	R8	0111

Flags

1. Zero flag
2. Carry flag
3. Less than flag

Instruction format

Opcode	Mode	Operand 1	Operand 2

Addressing modes

1. Register addressing mode

Opcode	Mode	Register	Register
6	2	4	4

2. **Memory addressing mode**

Opcode	Mode	Register	RAM addr
6	2	4	12

3. **Immediate addressing mode**

Opcode	Mode	Register	Value
6	2	4	12

Modes

Addressing mode	Mode bits
Register addressing mode	00
RAM addressing mode	01
Immediate addressing mode	10

Opcodes and Mnemonics

Opcode	Mnemonic	Example	Operation
000000	nop	nop	No operation
000001	ld	ld r1, #75	R1 <- 0x75
000010	ldl	ldl r2, #5a	R2L <- 0x5a
000011	ldh	ldh r2, #7a	R2H <- 0x7a
000100	ldpc	ldpc R1	R1 <- PC
000101	str	str r2, $24	Mem $24 <- R2L Mem $25 <- R2H

000110	strl	strl r2, $23	Mem $23 <- R2L
000111	strh	Strh r2, $24	Mem $24 <- R2H
001000	add	add r1, r2	R1 <- R1 + R2
001001	sub	sub r1, r2	R1 <- R1 - R2
001010	mul	mul r3, r4	R1 <- R3 * R4 H R2 <- R3 * R4 L
001011	and	and r4, r5	R4 <- R4 & R5
001100	or	or r4, r5	R4 <- R4 | R5
001101	not	not r4	R4 <- !R4
001110	xor	xor r5, r6	R5 <- R5^R6
001111	xnor	xnor r4, r5	R5 <- R5~^R6
010000	shr	shr r1, r2	R1 <- R1 << R2
010001	shl	shl r2, r3	R2 <- R2 >> R3
010010	ror	ror r3, r4	R3 <- rotate_right(R4, R3)
010011	rol	rol r3, r4	R3 <- rotate_left(R4, R3)
010100	stf	stf r1, r3	R1 <- flag id R3 <- flag value
010101	rdf	rdf r2	R2 <- flag values
010110	cmp	cmp r3, r5	Set zero flag if R3 == R4 Set less than flag If R3 < R4
010111	jmp	jmp #45	Jump to 45
011000	jc	jc #24	Jump to 24 If carry flag == 1
011001	je	je #67	Jump to 67 If zero flag == 1
011010	jl	jl #67	Jump to 67 If less flag == 1
011011	jle	jle #46	Jump to 67 If zero flag == 1

			or less flag == 1
011100	jg	jg #67	Jump to 67 If less flag == 0
011101	jge	jge #45	Jump to 67 If zero flag == 1 or less flag == 0