Sigma16 Instruction Formats

RRR instructions

An RRR instruction is represented in one 16-bit word, which has four 4-bit fields: op, d, a, b. The op field contains the operation code, which determines the specific instruction. The d field is the destination register, while the a and b fields are the source registers (source a and source b). An RRR instruction performs an operation on data in registers. It is one word, organised into four 4-bit fields:

- The *op field* contains the operation code, specifying which instruction is to be executed.
- The *d field* specifies the destination register, where the result of the operation will be loaded.
- The *a field* specifies the register that contains the first operand
- The *b field* specifies the register that contains the second operand

Arithmetic instructions have RRR format.

The following table shows the RRR instructions. For example, add R4,R8,R1 has an op field of 0 (which means this is an add instruction), a d field of 4, an a field of 8, and a b field is 1.

RRR format instructions.

op	format	mnemonic	operands	action
0	RRR	add	R1,R2,R3	R1 := R2+R3
1	RRR	sub	R1,R2,R3	R1 := R2-R3
2	RRR	mul	R1,R2,R3	R1 := R2*R3
3	RRR	div	R1,R2,R3	R1 := Int(R2/R3), R15 := R2 mod R3
4	RRR	cmplt	R1,R2,R3	R1 := (R2<R3)
5	RRR	cmpeq	R1,R2,R3	R1 := (R2=R3)
6	RRR	cmpgt	R1,R2,R3	R1 := (R2>R3)
7	RRR	inv	R1,R2,R3	R1 := NOT R2
8	RRR	and	R1,R2,R3	R1:=R2 AND R3
9	RRR	or	R1,R2,R3	R1:=R2 OR R3
a	RRR	xor	R1,R2,R3	R1:=R2 XOR R3
b	RRR	shiftl	R1,R2,R3	R1 := R1 (R3 left shift) R2
c	RRR	shiftr	R1,R2,R3	R1 := R1 (R3 right shift) R2
d	RRR	trap	R1,R2,R3	pc := interrupt handler
e	XX	(expand to	Xformat)	
f	RX	(expand to	RX format)	

RX and X format

An RX instruction is represented in two 16-bit words, which must appear in consecutive words in memory. The first word has four 4-bit fields: op, d, a, b. The second word is a single 16-bit field called the displacement. The op field is 15 for all RX instructions, and the b field determines the actual instruction. Thus an op of 15 means that the real opcode has been "escaped" to the b field. This technique is called an "expanding opcode". The RX format is used for instructions that have two operands, one in a register (the R operand) and the other specified by an indexed address (the X operand). Examples are load, store, jumpf.

The X format is a special case of the RX format, where there is only one operand, which is an indexed address. In the machine language, an X format instruction has a "don't care" value in the d field. The jump instruction has X format.

RX (and X) format instructions

op	b	format	mnemonic	operands	action
f	0	RX	lea	Rd,x[Ra]	Rd := x+Ra
f	1	RX	load	Rd,x[Ra]	Rd := mem[x+Ra]
f	2	RX	store	Rd,x[Ra]	mem[x+Ra] := Rd
e	3	X	jump	x[Ra]	pc := x+Ra
f	4	RX	jumpf	Rd,x[Ra]	if Rd==0 then pc := x+Ra
f	5	RX	jumpt	Rd,x[Ra]	if Rd<>0 then pc := x+Ra
f	6	RX	jal	Rd,x[Ra]	Rd := pc, pc := x+Ra