Slicudis RISC Machine reference Base instruction set

FMT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DI	opcode						rd					imm [20:0]																				
I	opcode						imm [25:0]																									
DSS	opcode						rd					rs1					fn4				rs2					fn7						
DSI	opcode						rd					rs1					fn4				imm [11:0]											
SSI	opcode						imm [11:7]					rs1					fn4				rs2					imm [6:0]						

Arithmetic

Instruction	Description	FMT	Opcode	FN4	Note
add rd, rs1, rs2	rd = rs1 + rs2	DSS	0x0	0x0	-
sub rd, rs1, rs2	rd = rs1 - rs2	DSS	0x0	0x1	-
and rd, rs1, rs2	rd = rs1 & rs2	DSS	0x0	0x2	-
or rd, rs1, rs2	rd = rs1 | rs2	DSS	0x0	0x3	-
xor rd, rs1, rs2	rd = rs1 ^ rs2	DSS	0x0	0x4	-
shr rd, rs1, rs2	rd = rs1 >> rs2	DSS	0x0	0x5	Zero-extends
asr rd, rs1, rs2	rd = rs1 >>> rs2	DSS	0x0	0x6	Sign-extends
shl rd, rs1, rs2	rd = rs1 << rs2	DSS	0x0	0x7	-
cch rd, rs1, rs2	rd = (rs1 + rs2)[32]	DSS	0x0	0x8	-
bch rd, rs1, rs2	rd = (rs1 - rs2)[32]	DSS	0x0	0x9	-
addi rd, rs1, imm12	rd = rs1 + imm12	DSI	0x1	0x0	-
andi rd, rs1, imm12	rd = rs1 & imm12	DSI	0x1	0x2	-
ori rd, rs1, imm12	rd = rs1 | imm12	DSI	0x1	0x3	-
xori rd, rs1, imm12	rd = rs1 ^ imm12	DSI	0x1	0x4	-
shri rd, rs1, imm12	rd = rs1 >> imm12	DSI	0x1	0x5	Zero-extends
asri rd, rs1, imm12	rd = rs1 >>> imm12	DSI	0x1	0x6	Sign-extends
shli rd, rs1, imm12	rd = rs1 << imm12	DSI	0x1	0x7	-
cchi rd, rs1, imm12	rd = cout of rs1 + imm12	DSI	0x1	0x8	-
bchi rd, rs1, imm12	rd = borrow of rs1 - imm12	DSI	0x1	0x9	-

Memory and data

Instruction	Description	FMT	Opcode	FN4	Note
lui rd, imm21	rd = imm21 << 11	DI	0x2	-	-
ldb rd, [rs1, imm12]	rd = m[rs1 + imm12][7:0]	DSI	0x4	0x0	Zero-extends
ldw rd, [rs1, imm12]	rd = m[rs1 + imm12][15:0]	DSI	0x4	0x1	Zero-extends
ldd rd, [rs1, imm12]	rd = m[rs1 + imm12][31:0]	DSI	0x4	0x2	Zero-extends
ldsb rd, [rs1, imm12]	rd = m[rs1 + imm12][7:0]	DSI	0x4	0x4	Sign-extends
ldsw rd, [rs1, imm12]	rd = m[rs1 + imm12][15:0]	DSI	0x4	0x5	Sign-extends
stb rs2, [rs1, imm12]	m[rs1 + imm12] = rs2[7:0]	SSI	0x3	0x0	-
stw rs2, [rs1, imm12]	m[rs1 + imm12] = rs2[15:0]	SSI	0x3	0x1	-
std rs2, [rs1, imm12]	m[rs1 + imm12] = rs2[31:0]	SSU	0x3	0x2	-

Jumping

Instruction	Description	FMT	Opcode	FN4	Note
jal rd, label	rd = ip+4; ip = ($-label)	DI	0x5	-	-
jalr rd, rs1, imm12	rd = ip+4; ip = rs1 + imm12	DSI	0x6	0x0	-
jeq rs1, rs2, label	if (rs1 == rs2): ip = ip = ($-label)	SSI	0x7	0x0	-
jlt rs1, rs2, label	if (rs1 < rs2): ip = ip = ($-label)	SSI	0x7	0x1	-
jslt rs1, rs2, label	if (rs1 < rs2): ip = ip = ($-label)	SSI	0x7	0x2	Signed args.
jne rs1, rs2, label	if (rs1 != rs2): ip = ip = ($-label)	SSI	0x7	0x4	-
jge rs1, rs2, label	if (rs1 >= rs2): ip = ip = ($-label)	SSI	0x7	0x5	-
jsge rs1, rs2, label	if (rs1 >= rs2): ip = ip = ($-label)	SSI	0x7	0x6	Signed args.

System

Instruction	Description	FMT	Opcode	FN4	Note
syscall	s = k; k = 1; ir = ip+4; ip = s_int	DSS	0x8	0x0	-
sysret	k = s; ip = ir	DSS	0x9	0x0	-

64-bit extension

Instruction	Description	FMT	Opcode	FN4	Note
stq rs2, [rs1, imm12]	m[rs1 + imm12][63:0] = rs2	SSI	0x3	0x3	-
ldsd rd, [rs1, imm12]	rd = m[rs1 + imm12][31:0]	DSI	0x4	0x6	Sign-extends
ldq rd, [rs1, imm12]	rd = m[rs1 + imm12][63:0]	DSI	0x4	0x3	Zero-extends

Multiplication/Division extension

Instruction	Description	FMT	Opcode	FN4	Note
mul rd, rs1, rs2	rd = (rs1 * rs2)[31:0]	DSS	0xA	0x0	-
mulh rd, rs1, rs2	rd = (rs1 * rs2)[63:32]	DSS	0xA	0x2	-
smulh rd, rs1, rs2	rd = (rs1 * rs2)[63:32]	DSS	0xA	0x3	Signed args.
div rd, rs1, rs2	rd = rs1 / rs2	DSS	0xA	0x4	-
sdiv rd, rs1, rs2	rd = rs1 / rs2	DSS	0xA	0x5	Signed args.
mod rd, rs1, rs2	rd = rs1 % rs2	DSS	0xA	0x6	-
smod rd, rs1, rs2	rd = rs1 % rs2	DSS	0xA	0x7	Signed args.

Register file

Register	Name	Function	Saver
r0	zr	Constant 0	-
r1	sr	Control status register	-
r2	ir	Interrupt return pointer	-
r3	at	Assembler temporary	Caller
r4	rp	Return pointer	Callee
r5	sp	Stack pointer	Callee
r6	fp	Frame pointer	Callee
r7	gp	Global pointer	-
r8-10	a0-2	Function arguments/return values	Caller
r11-12	s0-1	Saved registers	Callee
r13-20	t0-7	Temporaries	Caller
r21-26	s2-7	Saved registers	Callee
r27-31	a3-7	Function arguments/return values	Caller