SIA32

Instruction Formats

3 Register (3R) - 10

Immediate (8)	Rs1(5)	Rs2(5)	Function(4)	Rd (5)	Opcode(5)

2 Register (2R) - 11

Immediate (13)	Rs(5)	Function(4)	Rd (5)	Opcode(5)

Dest Only (1R) - 01

Immediate (18)	Function(4)	Rd (5)	Opcode(5)

No Register (0R) - 00

Immediate (27)	Opcode(5)

A SIA32 opcode is made up of two parts – the operation and the instruction format. The opcode is made by combining the instruction code and then the instruction format. For example – a 3R math operation is opcode 00010.

Rs1 and Rs2 are the two source registers; Rd is the destination register.

Registers

There are 32 general purpose registers (R0 – R31). R0 is hard-coded to 0; writing to it leaves it unchanged (is a NO OP).

There are 2 special purpose registers: Stack pointer (SP), Program Counter (PC). These are not directly readable or writable but are changed by instructions like branch, call, return, push and pop.

Instruction Definition Matrix

	3R (10)	2R (11)	Dest Only (01)	No R (00)
Math (000)	Rd ← Rs1 MOP Rs2	Rd ← Rd MOP Rs	COPY: Rd← imm	HALT
Branch (001)	pc ← Rs1 BOP Rs2 ? pc + imm : pc	pc ← Rs BOP Rd? pc + imm : pc	JUMP: pc ← pc + imm	JUMP: pc ← imm
Call (010)	pc ← Rs1 BOP Rs2 ? push pc; Rd + imm : pc	pc ← Rs BOP Rd? push pc; pc + imm : pc	push pc; pc ← Rd + imm	push pc; pc ← imm
Push (011)	mem[--sp] ← Rs1 MOP Rs2	mem[--sp] ← Rd MOP Rs	mem[--sp] ← Rd MOP imm	UNUSED
Load (100)	Rd ← mem [Rs1+ Rs2]	Rd ← mem[Rs + imm]	Rd ← mem [Rd + imm]	RETURN (pc ← pop)
Store (101)	mem[Rd + Rs1] ← Rs2	mem[Rd + imm] ← Rs	Mem[Rd] ← imm	UNUSED
Pop/interrupt (110)	PEEK: Rd ← mem [sp – (Rs1+ Rs2)]	PEEK: Rd ← mem[sp – (Rs + imm)]	POP: Rd ← mem[sp++]	UNUSED

	3R (10)	2R (11)	Dest Only (01)	No R (00)
Math (000)	MOP R# R# R#	MOP R# R#	COPY imm R#	HALT
Branch (001)	BRANCHBOP R# R# imm	BRANCHBOP R# R# imm	JUMPTO imm	JUMP imm
Call (010)	CALLBOP R# R# R# imm	CALLBOP R# R# imm	CALL R# imm	CALL imm
Push (011)	PUSHMOP R# R#	PUSHMOP R# R#	PUSHMOP R# imm	UNUSED
Load (100)	LOAD R# R# R#	LOAD R# imm R#	LOAD imm R#	RETURN
Store (101)	STORE R# R# R#	STORE R# R# imm	STORE imm R#	UNUSED
Pop/Peek (110)	PEEK R# R# R#	PEEK R# imm R#	POP R#	UNUSED

imm = immediate value
mem = main memory

MOP (math op)	
Bit Pattern	Meaning
1000	AND
1001	OR
1010	XOR
1011	NOT (negate op1; ignore op2)
1100	LSHIFT (“op1” is the value to shift, “op2” is the amount to shift; ignore all but the lowest 5 bits)
1101	RSHIFT (“op1” is the value to shift, “op2” is the amount to shift; ignore all but the lowest 5 bits)
1110	ADD
1111	SUB
0111	MULT
Other values are not valid	

BOP (boolean op)	
Bit Pattern	Meaning
0000	EQ (Equals)
0001	NEQ (Not Equal)
0010	LT (Less than)
0011	GE (Greater than or equal)
0100	GT (Greater than)
0101	LE (Less than or equal)
Other values are not valid	