TABLE II: SAYAC Instruction Set

[15:12]	[11:10]	[9]	[8]	[7:4]	[3:0]	Instruction	Notation
0000	Reserved						
0001	Reserved						
0010	00	0	0	rs1	rd	LDR	R_f(rd) <= MEM(R_f(rs1))
			1	rs1	rd	LIR	R_f(rd) <= IO(R_f(rs1))
		1	0			LDB	R_b(rd) <= MEM(R_f(rs1))
			1			LIB	R_f(rd) <= IO(R_b(5) + R_f(rs1))
	01	0	0	rs1	rd	STR	MEM(R_f(rd)) <= R_f(rs1)
			1	rs1	rd	SIR	IO(R_f(rd)) <= R_f(rs1)
		1	0			STB	MEM(R_b(3) + R_f(rd)) <= R_f(rs1)
			1			SIB	IO(R_b(5) + R_f(rd)) <= R_f(rs1)
	10	s	0	rs1	rd	JMR	PC <= PC + R_f(rs1) R_f(rd) <= PC + 1, if s=1
			1	rs1	rd	JMB	PC <= IO(R_b(rd) + rs1) R_f(15) <= IO(R_b(rd)), if s=1
	11	imm			rd	JMI	PC <= PC + SE”imm” R_f(rd) <= PC + 1
0011	rs2			rs1	rd	ANR	R_f(rd) <= R_f(rs1) AND R_f(rs2)
0100	imm				rd	ANI	R_f(rd) <= R_f(rd) AND USE”imm”
0101	imm				rd	MSI	R_f(rd) <= SE”imm”
0110	imm				rd	MHI	R_f(rd) ‘MSB <= “imm”
0111	rs2			rs1	rd	SIR	R_f(rd) <= R_f(rs1) LS± R_f(rs2)
1000	rs2			rs1	rd	SAR	R_f(rd) <= R_f(rs1) AS± R_f(rs2)
1001	rs2			rs1	rd	ADR	R_f(rd) <= R_f(rs1) + R_f(rs2)
1010	rs2			rs1	rd	SUR	R_f(rd) <= R_f(rs1) – R_f(rs2)
1011	imm				rd	ADI	R_f(rd) <= R_f(rd) + SE”imm”
1100	imm				rd	SUI	R_f(rd) <= R_f(rd) – SE”imm”
1101	rs2			rs1	rd	MUL	R_f(rd) <= R_f(rs1) × R_f(rs2) ‘LSB R_f(rd+1) <= R_f(rs1) × R_f(rs2) ‘MSB
1110	rs2			rs1	rd	DIV	R_f(rd) <= R_f(rs1) ÷ R_f(rs2) ‘Quo R_f(rd+1) <= R_f(rs1) mod R_f(rs2) ‘Rem
1111	00	0	0	0000	0000	MEC	PRV <= U-Mode EnvironmentCallException = 1
			1	rs1	rd	CMR	flags <= Cmp(R_f(rs1) , R_f(rd))
		1	imm		rd	CMI	flags <= Cmp(R_f(rd) , SE”imm”)
	01	0	flags interpretation bits		rd	BRC	PC <= R_f(rd) if flag
		1	flags interpretation bits		rd	BRR	PC <= PC + R_f(rd) if flag
	10	0	shim		rd	SHI	R_f(rd) <= R_f(rd) LS± “shim”
		1	shim		rd		R_f(rd) <= R_f(rd) AS± “shim”
	11	0	0	rs1	rd	NTR	R_f(rd) <= 1sComp(R_f(rs1))
			1		rd		R_f(rd) <= 2sComp(R_f(rs1))
		1	0	--	rd	NTD	R_f(rd) <= 1sComp(R_f(rd))
			1	--	rd		R_f(rd) <= 2sComp(R_f(rd))

Description	Transfers execution to the address contained in immediate value relative to the current instruction pointer and saves the address of the next instruction in register r_d.
Instruction Type	I
Instruction Fields	r_d= Index of destination register Imm = 6-bit signed immediate value

For branch instructions, PC is determined based on satisfying a condition. The condition is presented in 5 bits locating in Instruction [8:4] named as **F**lag **I**nterpretation **B**its, FIB[4:0]. Table 1 shows different conditions and FIB configuration.

	FIB[4:0]				
eq	X	X	0	0	0
lt	X	X	0	0	1
gt	X	X	0	1	0
gt/eq	X	X	0	1	1
lt/eq	X	X	1	0	0
neq	X	X	1	0	1

BRC

Opcode				Opcode-Extended			FIB					r_d			
1	1	1	1	0	1	0									

BRC	**Branch Conditional**
Instruction	Branch Registered with Condition
Operation	If (FIB) then PC← r_d
Assembler Syntax	BRC FIB r_d
Example	BRC 0x01 r_2
Description	If Flag Interpretation Bits (FIB) are true, then transfers program control to the instruction at the address specified by register r_d
Instruction Type	R
Instruction Fields	r_d= Index of destination register FIB = 5-bit compare flag interpretation bits

BRR

Opcode				Opcode-Extended			FIB					r_d			
1	1	1	1	0	1	1									

BRR	**Branch Conditional Relative**
Instruction	Branch Registered Relative with Condition
Operation	If (FIB) then PC←PC + r_d
Assembler Syntax	BRR FIB r_d
Example	BRR 0x01 r_2
Description	If Flag Interpretation Bits (FIB) are true, then transfers program control to the instruction to the address contained in register r_d relative to the current instruction pointer.
Instruction Type	R
Instruction Fields	r_d= Index of destination register FIB = 5-bit compare flag interpretation bits