KGP RISC PROCESSOR ARCHITECTURE

Computer Organization LABORATORY (CS39001)

Group no. : 30

Date: 08-Nov-2023

- First 6 bit of instruction of any type consists the opcode and first 2 bits of opcode decides the Format type of the opcode.

OPCODE [31:30]	Format type	Classes of Instruction	Instructions
00	R-type	Arithmetic/Logic/Shift	ADD, SUB, AND, OR, XOR, NOT, SLA, SRA, SRL
01/00	I-type	Immediate	ADDI, SUBI, ANDI, ORI, XORI, NOTI, SLAI, SRAI, SRLI, BR, BMI, BPL, BZ,LD, ST, LDSP, STSP
10	I2-type	Stack	PUSH, POP, CALL, RET
11	J-type	Load and store	HALT,NOP

- **R-type (opcode[31:30] = 00)**

  Op = opcode
  SR1 = Source Register 1
  SR2 = Source Register 2
  DR = Destination Register
  SHAMT = Shift amount
  FUNC = function code

OPCODE	SR1	SR2	DR	SHAMT	FUNC

6 bit	5 bit	5 bit	5 bit	5 bit	6 bit

OPCODE [29:26] Encoding

OPCODE	Function	INSTRUCTION	DEFINITION
0000	0000	ADD	DR ← SR1 + SR2
0000	0001	SUB	DR ← SR1 + (-SR2)
0000	0010	AND	DR ← SR1 ^ SR2
0000	0011	OR	DR ← SR1 v SR2
0000	0100	XOR	DR ← SR1 v SR2
0000	0101	NOT	DR ← ~SR1
0000	0110	SLA	DR ← SR1 << SR2
0000	0111	SRA	DR ← SR1 >> SR2(arithmetic)
0000	1000	SRL	DR ← SR1 >> SR2

Where,
(-SR2) = 2's compliment of SR2
v = XOR

- **I-type (opcode[31:30] = 01/00)**
  Op = opcode
  SR = Source Register
  DR = Destination Register
  IMM = immediate field

OPCODE	SR	DR	IMM
6 bit	5 bit	5 bit	16 bit

OPCODE [31:26] Encoding

OPCODE	INSTRUCTION	DEFINITION
010000	ADDI	DR ← SR1 + #
010001	SUBI	DR ← SR1 + #
010010	ANDI	DR ← SR1 ^ #
010011	ORI	DR ← SR1 v #
010100	XORI	DR ← SR1 v̲ #
010101	NOTI	DR ← ~#
010110	SLAI	DR ← SR1 << #
010111	SRAI	DR ← SR1 >> #(arithmetic)
011000	SRLI	DR ← SR1 >> #
011001	BR	PC ← PC + #
011010	BMI	PC ← PC - #, if(Ri<0)
011011	BPL	PC ← PC + #, if(Ri>0)
011100	BZ	PC ← PC - #, if(Ri=0)
000001	LD	Ri←Mem[Rj+#]
000010	ST	Mem[Ri+#]←Rj
000011	LDSP	SP←Mem[Ri+#]
000100	STSP	Mem[Ri+#]←SP

Where, # represents an immediate value

- **I2-type (opcode[31:30] = 10)**
  Op = opcode
  SR = Source Register
  DR = Destination Register

IMM = immediate field

OPCODE	SR	DR	IMM
6 bit	5 bit	5 bit	16 bit

OPCODE [29:26] Encoding

OPCODE	INSTRUCTION	DEFINITION
0000	PUSH SR	Pushing R6 into stack
0001	POP DR	Pop from stack and store in DR
0010	CALL #	SP ← SP – 4; Mem[SP] ← PC + 4; PC ← PC + #;
0011	RET	PC ← Mem[PC]; SP ← SP + 4

- **J-type (opcode[31:30] = 11)**

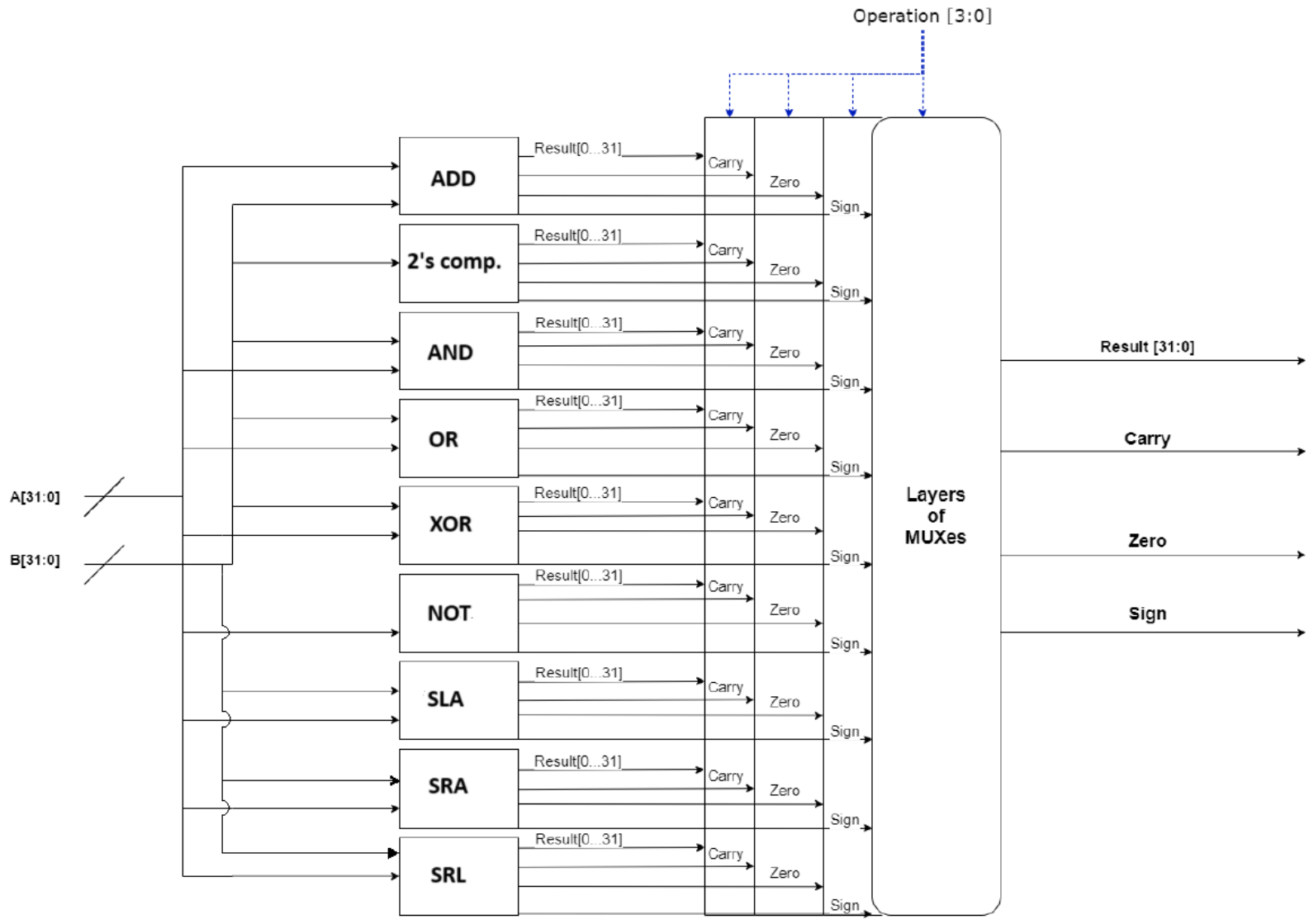
Op = opcode
SR = Source Register
DR = Destination Register
IMM = immediate field

OPCODE	IMM
6 bit	26 bit

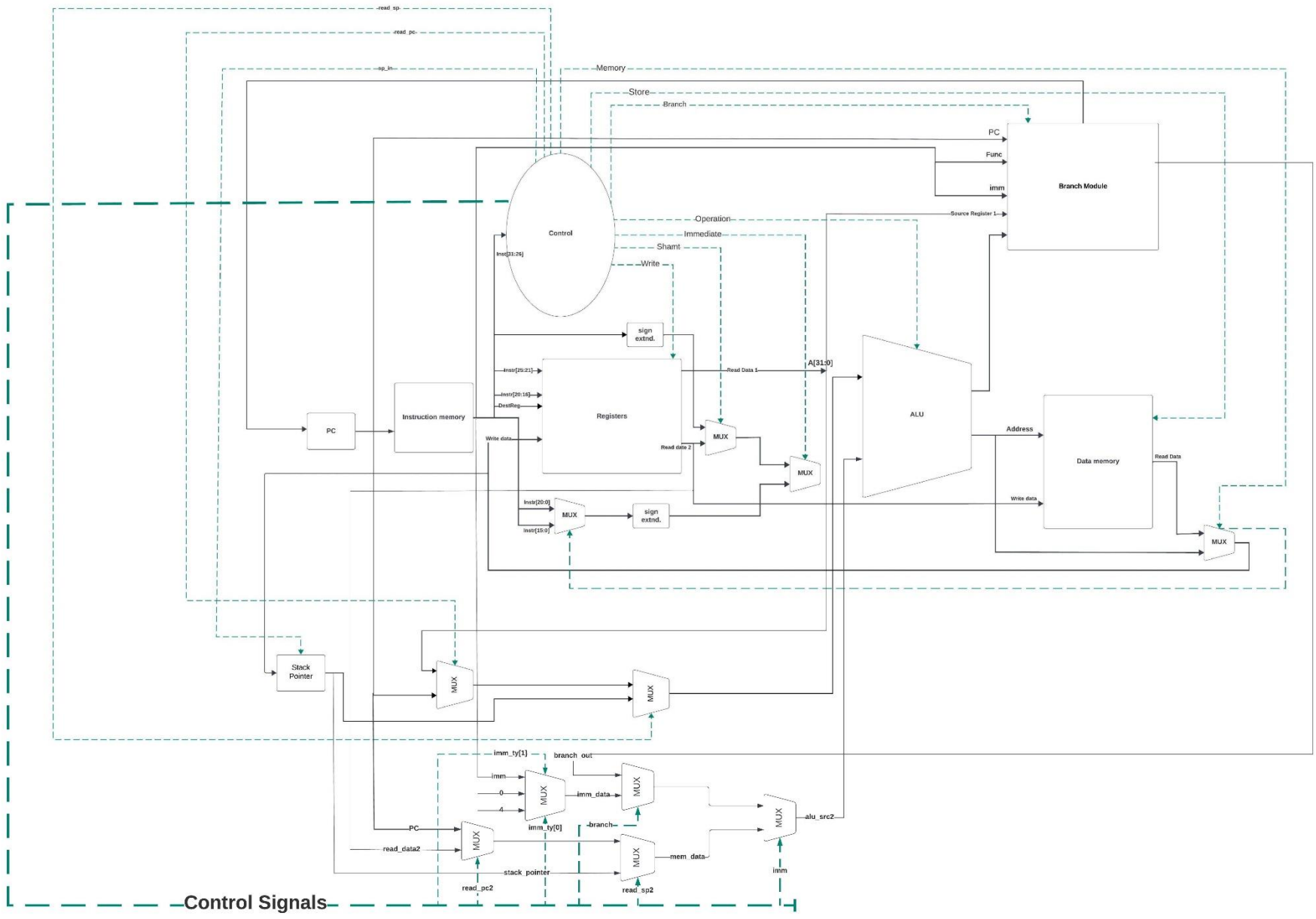
OPCODE [29:26] Encoding

OPCODE	INSTRUCTION	DEFINITION
0000	HALT	
0001	NOP	PC←PC+4

ALU ARCHITECHTURE



SINGLE CYCLE EXECUTION UNIT



TRUTH TABLE FOR CONTROL SIGNALS

Instruction	imm_type	immediate	sp_in	pc_in	alu_code	read_sp1	read_pc1	memory	store	branch	branch_op	write	read_sp2	read_pc2
ADD	-	0	Depends on destreg number	Depends on dest reg number	0	Depends on src reg number	Depends on src reg number	0	0	0	-	If gpr is used	Depends on src reg number	Depends on src reg number
	1	1	0	1	0	0	1	0	0	0	-	0	0	0
ADDI	2	1	Depends on destreg number	Depends on destreg number	0	Depends on src reg number	Depends on src reg number	0	0	0	-	If gpr is used	0	0
	1	1	0	1	0	0	1	0	0	0	-	0	0	0
LD	2	1	Depends on destreg number	Depends on destreg number	0	Depends on src number	0	1	0	0	-	If gpr is used	0	0
											-			
ST	2	1	0	0	0	Depends on src number	0	0	1	0	-	0	0	0
											-			
BMI	2	1	0	1	0	Depends on src number	Depends on src number	0	0	1	1	0	0	0
	1	1	0	1	0	0	1	0	0	0	-	0	0	0
CALL	1	1	0	1	0	0	1	0	0	0	-	0	0	0
	1	1	1	0	1	1	0	0	0	0	-	0	0	0
	0	1	0	0	0	1	0	0	1	0	-	0	0	0
	2	1	0	1	0	0	1	0	0	0	-	0	0	0
	1	1	0	1	0	0	1	0	0	0	-	0	0	0
RET	0	1	0	1	0	0	1	1	0	0	-	0	0	0
	1	1	1	0	0	1	0	0	0	0	-	0	0	0
	1	1	0	1	0	0	1	0	0	0	-	0	0	0
MOVE	0	1	Depends on destreg number	Depends on destreg number	0	Depends on src number	Depends on src number	0	0	0	-	If gpr is used	0	0
	1	1	0	1	0	0	1	0	0	0	-	0	0	0