Lab Report for Group 31

KGP-miniRISC Processor

Srishty Gandhi(20CS30052) and Yatindra Indoria (20CS30060)

COMPUTER ORGANIZATION LABORATORY
Department of Computer Science and Engineering

November 13, 2022

Instruction Format Encoding

Number of supported registers → 64 → 2^6

Formats

Arithmetic, Logical and shift operators				
OPCODE	Reg 1	Reg 2	Shamt	Funct
4	6	6	5	11

- add rs, rt
- comp rs, rt
- and rs, rt
- xor rs, rt
- sll rs, sh
- srl rs, sh
- sra rs, sh
- sllv rs, rt
- srlv rs, rt
- srav rs, rt
- diff rs, rt

Memory Operations			
OPCODE	Reg 1	Reg 2	Immediate
4	6	6	16

- lw rt, rs, imm
- sw rt, rs, imm

Immediate and Branch Operations		
OPCODE	Reg 1	Immediate
4	6	22

- addi rs, imm
- compi rs, imm
- br rs
- bltz rs, L
- bz rs, L
- bnz rs, L

Branch Operations	
OPCODE	Immediate
4	28

- b L
- bl L
- bcy L
- bncy L

This ISA prioritises performance and maximum utilization of resources hence we have given a provision of 64 registers, and we have taken a rather complex decoding architecture to allow for larger branching distances and bigger immediates constants. Hence the requirement to break down the same into multiple instructions will not come in majority of the scenario, even in large and complex code basis.

OPCODE and FUNCTION CODE

Operation	OpCODE	Fuct
And	1	0
Compliment	1	1
Addi	2	NA
Compliment Immediate	3	NA
AND	1	2
XOR	1	3
SLL	1	4
SRL	1	5
SLL variable	1	6
SRL variable	1	7
SRA	1	8
SRA variable	1	9
Load word	4	NA
Store word	5	NA
Unconditional Branch	6	NA
Branch Register	7	NA
Branch on less than zero	8	NA
Branch on flag zero	9	NA
Branch on flag not zero	10	NA
Branch and link	11	NA
Branch on carry	12	NA
Branch on no carry	13	NA
Diff	14	NA

The Control Unit Table

OpCode	ALUSource	ImmSel	ALUOp	WriteReg	MemWrite	MemRead	MemRegPC	Branch_ctl
0001	1	0	100	10	0	0	10	000
0010	0	1	110	10	0	0	10	000
0011	0	1	101	10	0	0	10	000
0100	0	0	001	11	0	1	11	000
0101	0	0	001	00	1	0	00	000
0110	0	0	000	00	0	0	00	100
0111	0	0	000	00	0	0	00	111
1000	0	0	000	00	0	0	00	101
1001	0	0	000	00	0	0	00	101
1010	0	0	000	00	0	0	00	101
1011	0	0	000	00	0	0	00	100
1100	0	0	000	00	0	0	00	100
1101	0	0	000	00	0	0	00	100
1110	1	0	111	10	0	0	10	000

- branch_ctl–
    - Branch_ctl [2] – whether or not branching occurs
    - Branch_ctl[1] – decides between branch register and branching to immediate constant
    - Branch_ctl[0] – decides between 22 bit and 28 bit immediate constant
- MemRegPC –
    - MemRegPC[0] – decides between memory and ALU output
    - MemRegPC[1] – decides between the above and Program counter (in case of branch and link)
- Mem read – 1 only for lw (when we have to read from memory)
- MemWrite – i only for sw (when we have to write onto memory)
- Write reg – similar encoding as memregPC
- ImmSel – select between immediate and select

ALU (op codes 1,2,3)

Input → 32 bit reg1, 32bit reg2 , 5bit shift amount, Control signal
Output → 32 bit register , flag [3] → zero negative carry
First bit will be one or zero depending upon what is the input source
1 – reg2
0 – 5 bits of shamt

Next 3 bits denote
000 – ADD
001 – Compliment
010 – AND
011 – XOR
100 – SLL
101 – SRL
110 – SRA

Special Case
1111 → reset
0111 → DIFF

Function	OpCode	ALUOp	FuncCode	ALUSource	Shamt or Reg2	Control signal
add	1	100	0	1	1	1000 -8
compliment	1	100	1	1	1	1001 - 9
AND	1	100	2	1	1	1010 - 10
XOR	1	100	3	1	1	1011 - 11
SLL	1	100	4	1	0	0100 - 4
SRL	1	100	5	1	0	0101 - 5
SLLv	1	100	6	1	1	1100 - 12
SRLv	1	100	7	1	1	1101 - 13
SRA	1	100	8	1	0	0110 - 14
SRAv	1	100	9	1	1	1110 - 6
addi	2	110	NA	0	1	1000 - 8
compi	3	101	NA	0	1	1001 - 9
Diff	14	111	x	1	1	1111 - 7
lw	4	001	NA	0	1	1000 - 8
sw	5	001	NA	0	1	1000 - 8

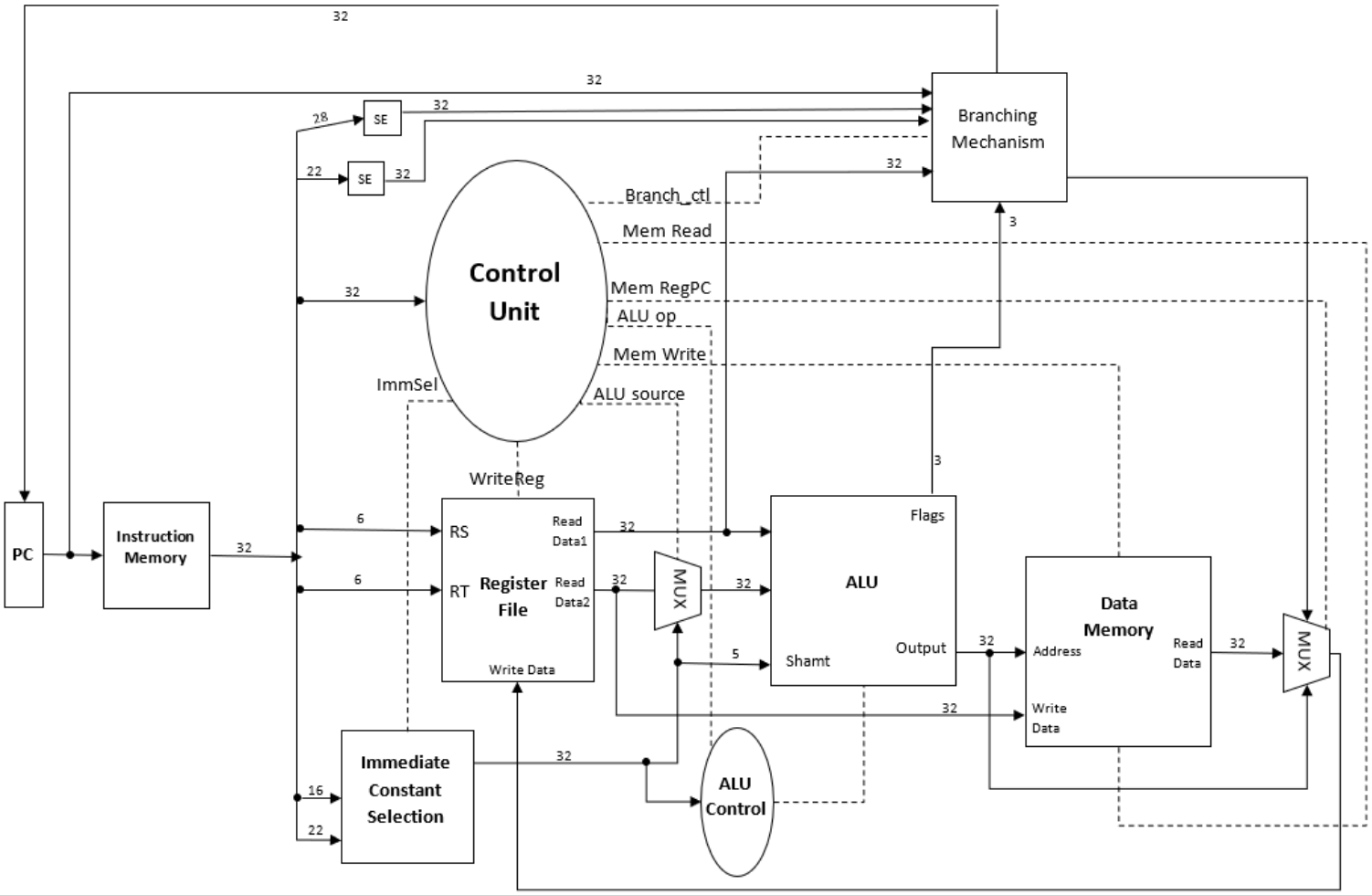

Figure 1 - Datapath diagram

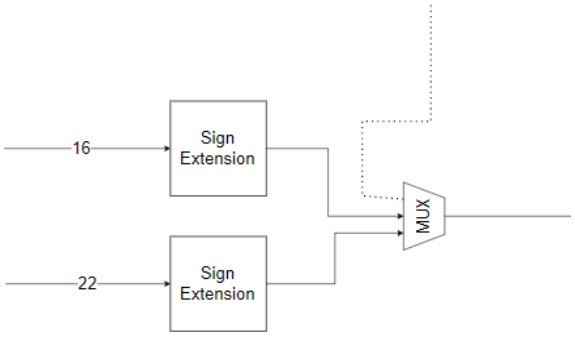

Figure 2 - Immediate Constant Selection unit

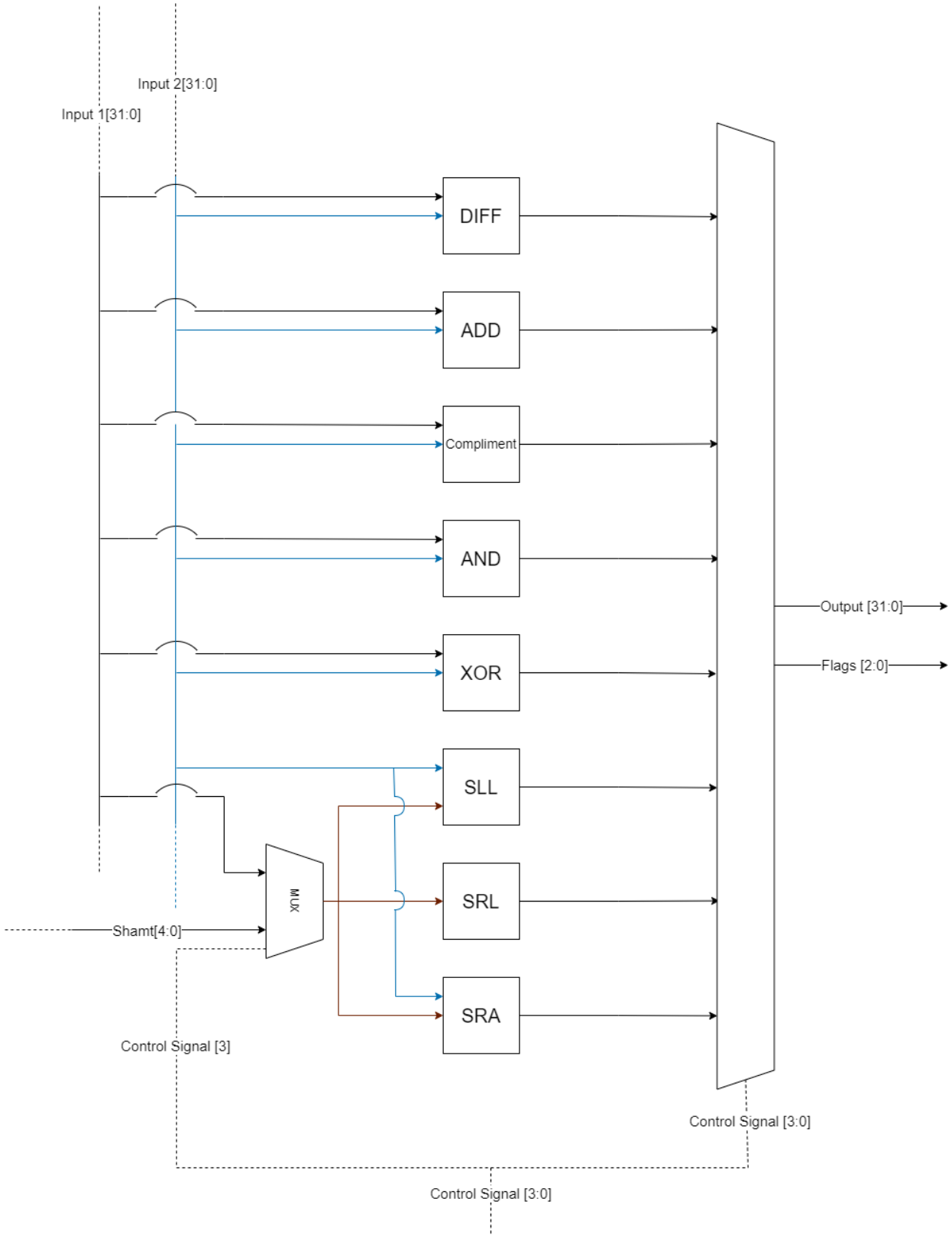
Input 2[31:0]
Input 1[31:0]
DIFF
ADD
Compliment
AND
XOR
SLL
SRL
SRA
MUX
Shamt[4:0]
Control Signal [3]
Output [31:0]
Flags [2:0]
Control Signal [3:0]
Control Signal [3:0]

Figure 3 – ALU