KGP-miniRISC Processor

A 32-bit mini processor for AMD-Xilinx Artix-7 FPGA

Group: 80
Chirag Ghosh (20CS10020)
Sarita Singh (20CS10053)

Instruction Set Architecture (ISA)

Class	Instruction	Usage	Meaning
Arithmetic	Add	add rs,rt	$rs \leftarrow (rs) + (rt)$
	Comp	comp rs,rt	$rs \leftarrow$ 2's Complement (rs)
	Add immediate	addi rs,imm	$rs \leftarrow (rs) + imm$
	Complement Immediate	compi rs,imm	$rs \leftarrow$ 2's Complement (imm)
Logic	AND	and rs,rt	$rs \leftarrow (rs) \wedge (rt)$
	XOR	xor rs,rt	$rs \leftarrow (rs) \oplus (rt)$
Shift	Shift left logical	shll rs, sh	$rs \leftarrow (rs)$ left-shifted by sh
	Shift right logical	shrl rs, sh	$rs \leftarrow (rs)$ right-shifted by sh
	Shift left logical variable	shllv rs, rt	$rs \leftarrow (rs)$ left-shifted by (rt)
	Shift right logical	shrl rs, rt	$rs \leftarrow (rs)$ right-shifted by (rt)
	Shift right arithmetic	shra rs, sh	$rs \leftarrow (rs)$ arithmetic right-shifted by sh
	Shift right arithmetic variable	shrav rs, rt	$rs \leftarrow (rs)$ right-shifted by (rt)
Memory	Load Word	lw rt,imm(rs)	$rt \leftarrow mem[(rs) + imm]$
	Store Word	sw rt,imm,(rs)	$mem[(rs) + imm] \leftarrow (rt)$
Branch	Unconditional branch	b L	goto L
	Branch Register	br rs	goto (rs)
	Branch on less than 0	bltz rs,L	if$(rs) < 0$ then goto L
	Branch on flag zero	bz rs,L	if $(rs) = 0$ then goto L
	Branch on flag not zero	bnz rs,L	if$(rs) \neq 0$ then goto L
	Branch and link	bl L	goto L; 31 $\leftarrow$ (PC)+4
	Branch on Carry	bcy L	goto L if Carry = 1
	Branch on No Carry	bncy L	goto L if Carry = 0
Complex	Diff	diff rs, rt	$rs \leftarrow$ the LSB bit at which rs and rt differ

Instruction Format Encoding

R-type instruction format

- Arithmetic, Logic, Shift and Complex instructions
- Differentiation of instruction to be done using Opcode and func.

Opcode	destination reg.	source reg.	shamt	N/A	func
6 bit	5 bits	5 bits	5 bits	6 bits	5 bits

I-type instruction format

- Arithmetic Immediate and Memory instructions
- No func. Different Opcode for each instruction.
- Range of immediate: -32768 to 32767

Opcode	destination reg.	source reg.	immediate
6 bit	5 bits	5 bits	16 bits

Branching-type instruction format

- Branching instructions

Opcode	source reg.	address	func
6 bit	5 bits	16 bits	5 bits

Operation Encoding

Operation	OPCode	Func
Add	0	0
Comp	0	1
AND	1	0
XOR	1	1
Shift Left Logical	2	0
Shift Right Logical	2	1
Shift Left Logical Variable	2	2
Shift Right Logical Variable	2	3
Shift Right Arithmetic	2	4
Shift Right Arithmetic Variable	2	5
Diff	3	0
Unconditional Branch	4	0
Branch on Carry	4	1
Branch on No Carry	4	2
Branch Register	5	0
Branch < 0	5	1
Branch on Flag = 0	5	2
Branch on Flag != 0	5	3
Branch and Link	6	0
Add Immediate	60	-
Complement Immediate	61	-
Load Word	62	-
Store Word	63	-

Control Unit Encoding

OpCode (6)	RegDst (2)	RegWrite(1)	ALUop (3)	MemRead (1)	MemWrite (1)	Branch OP (2)	ALuSource (2)	PCMem Reg (2)
0	0	1	1	0	0	0	0	0
1	0	1	3	0	0	0	0	0
2	0	1	5	0	0	0	0	0
3	0	1	6	0	0	0	0	0
4	0	0	0	0	0	1	0	0
5	0	0	0	0	0	2	0	0
6	1	1	0	0	0	3	0	2
60	0	1	2	0	0	0	1	0
61	0	1	4	0	0	0	1	0
62	2	1	2	1	0	0	1	1
63	0	0	2	1	1	0	1	0

RegDst: Chooses whether to write to rs, rt or 31.

RegWrite: Chooses whether to write or not.

ALUop: 3-bit control output to determine the operation in ALU.

MemRead: Whether to read from memory.

MemWrite: Whether to write to memory.

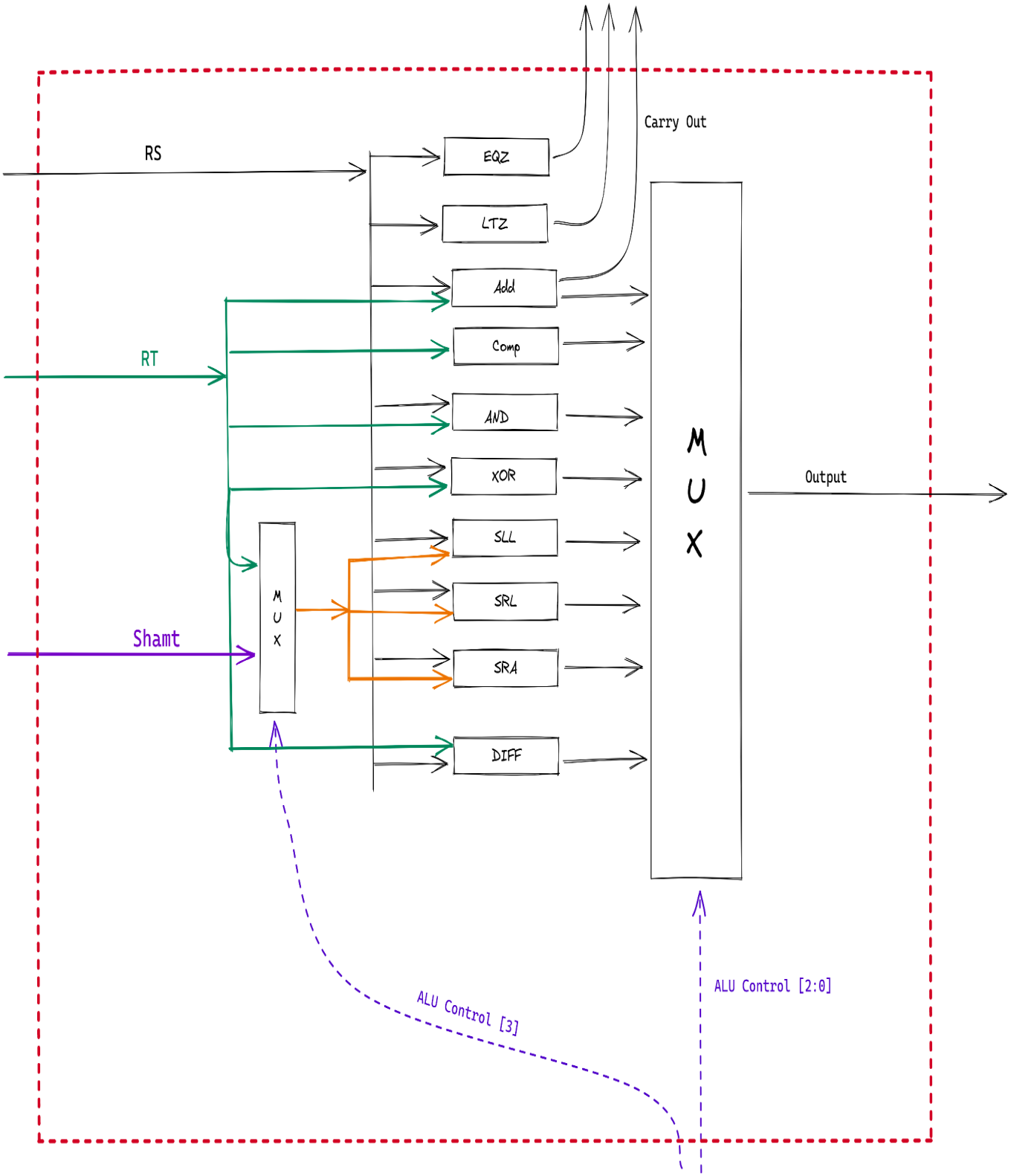
BranchOP: 2-bit control output to choose the branching logic according to various Opcodes.

ALUSource: Whether to take 2nd ALU input from register or immediate instruction.

PCMemReg: 2-bit control output to choose whether to write PC, Memory or Register data to register file.

ALU Control Encoding

ALUop	Function	ALU Control Output
0	-	-
1	0	0000
1	1	0001
2	-	1000
3	0	0010
3	1	0011
4	-	1001
5	0	1100
5	1	1101
5	2	0100
5	3	0101
5	4	1110
5	5	0110
6	0	0111

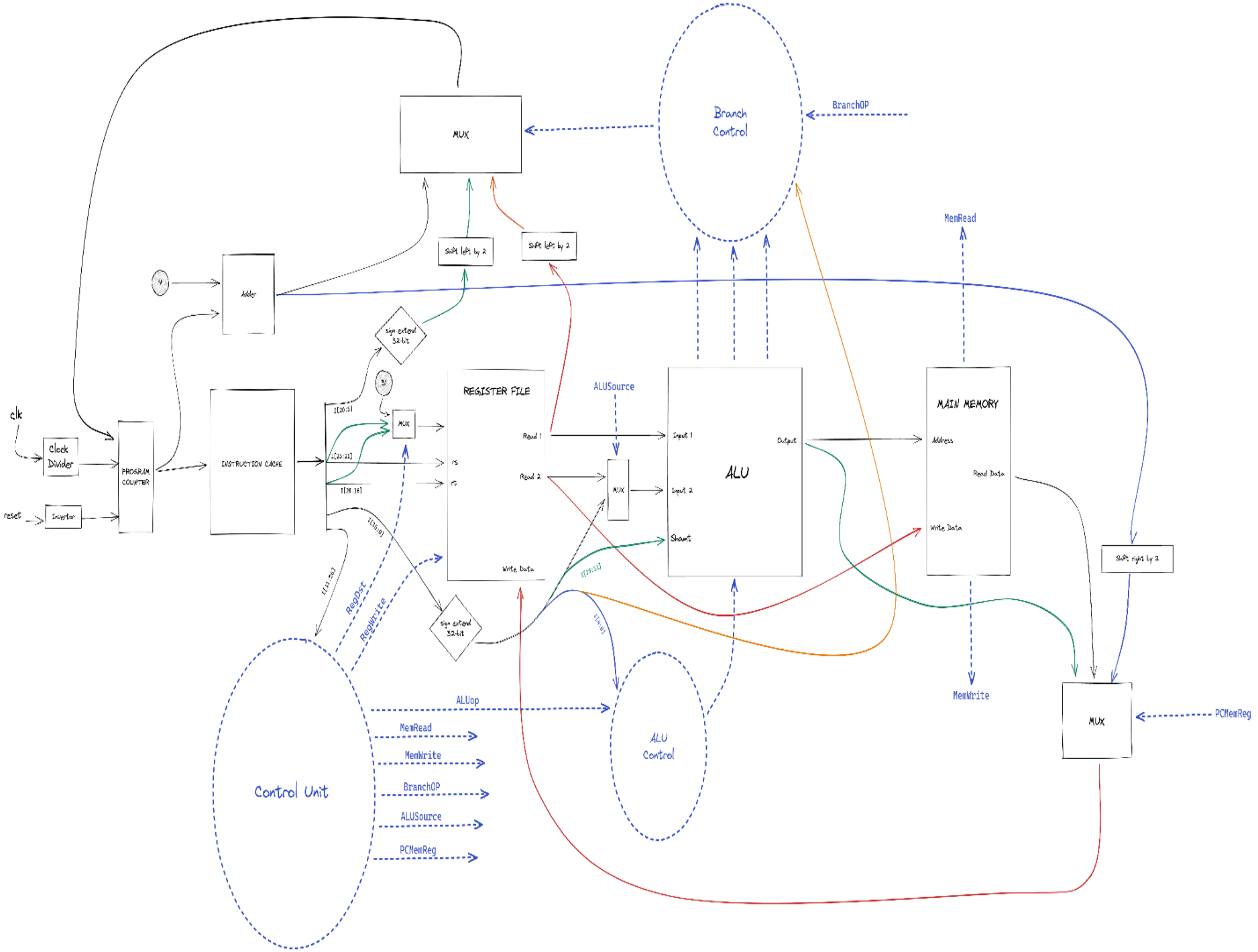

RS
RT
Shamt
EQZ
LTZ
Add
Comp
AND
XOR
SLL
SRL
SRA
DIFF
MUX
Carry Out
MUX
Output
ALU Control [3]
ALU Control [2:0]

Branch Operation Encoding

BranchOP	Function	Zero Flag	<0 Flag	Carry out	Output
0	-	-	-	-	00
1	0	-	-	-	01
1	1	-	-	0	00
1	1	-	-	1	01
1	2	-	-	0	01
1	2	-	-	1	00
2	0	-	-	-	10
2	1	-	0	-	00
2	1	-	1	-	01
2	2	0	-	-	00
2	2	1	-	-	01
2	3	0	-	-	01
2	3	1	-	-	00
3	0	-	-	-	01

Flags:
Zero Flag: Set to 1 when operation generates a zero output, otherwise 0
Sign bit (<0 Flag): Set to 1 when operation generates a negative output, otherwise 0
Carry out: Set to 1 when operation generates a carry, otherwise 0

Architecture


MUX
Branch Control
BranchOP
MemRead
Shift left by 2
Shift left by 2
4
Adder
sign extend 32-bits
clk
Clock Divider
reset
Inverter
PROGRAM COUNTER
INSTRUCTION CACHE
I[20:15]
I[20:16]
I[15:0]
I[31:26]
32
MUX
REGISTER FILE
rs
rt
Write Data
Read 1
Read 2
ALUSource
MUX
Input 1
Input 2
Shamt
ALU
Output
MAIN MEMORY
Address
Read Data
Write Data
Shift right by 2
sign extend 32-bit
I[5:0]
I[10:6]
RegDst
RegWrite
ALUop
MemRead
MemWrite
BranchOP
ALUSource
PCMemReg
Control Unit
ALU Control
MemWrite
MUX
PCMemReg