This implementation shows the data path design which is a set of functional units including ALU, Data Memory, File Register, etc.. , that carry out data processing operations.

Functions :

Input **output**

muxt(I0,I1,s,y1)

- 5-bit inputs and outputs, as shown in the figure.

muxtwo(i0,i1,s,y)

muxfour(i0,i1,i2,i3,s0,s1,y)

- These two functions have 32-bit input and output

functions.instmem(instruct,op,readrg1,readrg2,writerg,shift,func,datai,jump,pcup)

fileregister(readrg1,readrg2,writerg,writedata,regwrite,data1,data2)

ALU(a,b,ctrlalu,n,c,z,ov,res)

- This is called three times in main function. It has a 4-bit controller that initialized by *Func* and *aluop*

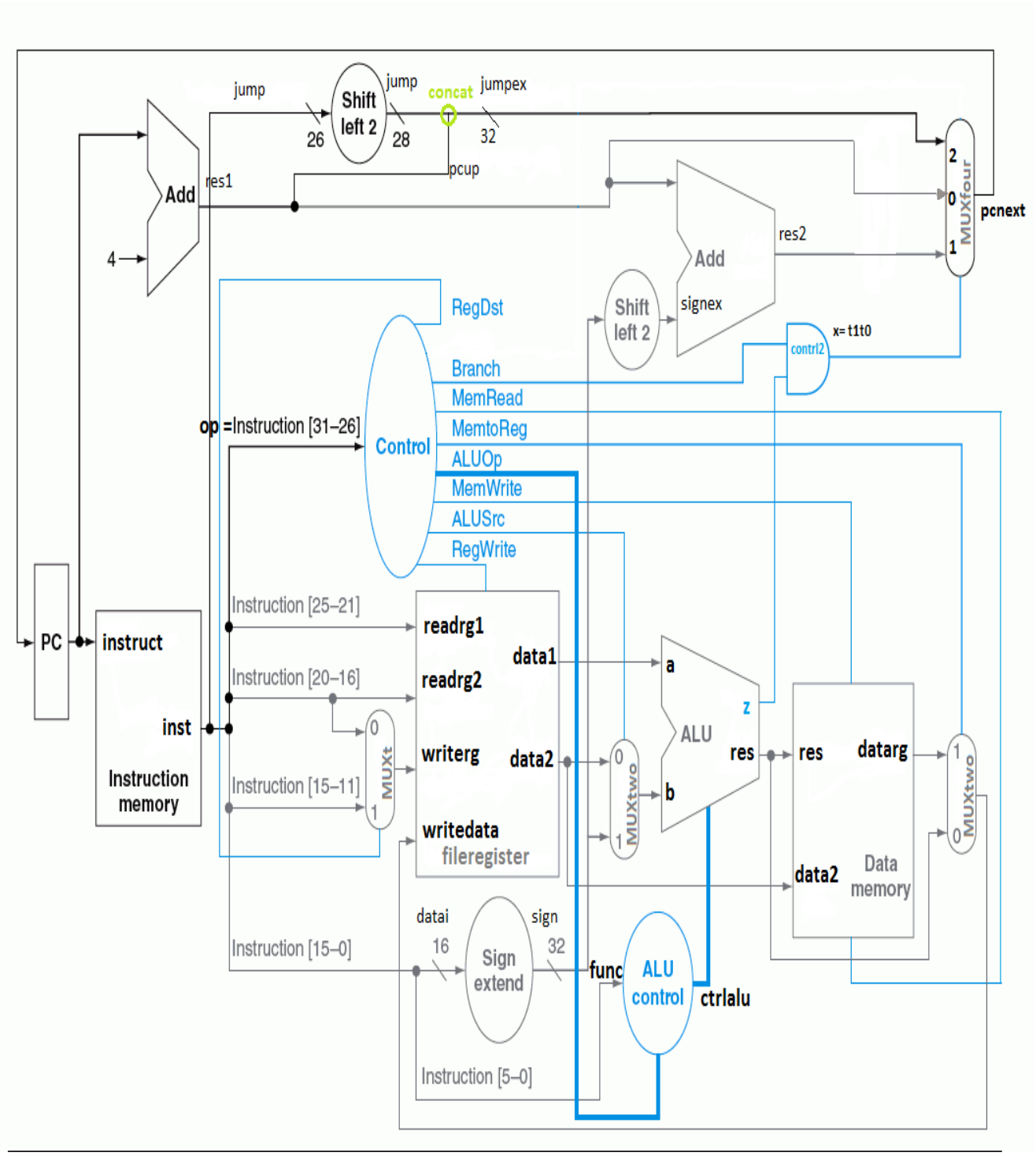
datamem(memread,memwrite,res,data2,datarg)

control(op,datai,func,jump,pcup,regdst,alusrc,memtoreg,regwrite,memread,mewrite,branch,aluop,ctrlalu,sign,signex,jumpex)

contrl2(branch,z,op,t0,t1)

- **Main Function:** (With nine inputs)

datapath(instruct,pcnext);


jump
Shift left 2
jump
concat
jumpex
26
28
32
pcup
Add
res1
4
2
0
1
MUXfour
pcnext
res2
Add
signex
Shift left 2
x= t1t0
contrl2
RegDst
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
op =Instruction [31–26]
Control
PC
instruct
inst
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Instruction [5–0]
readrg1
readrg2
writerg
writedata
fileregister
data1
data2
0
1
MUXt
0
1
MUXtwo
a
b
ALU
z
res
res
datarg
data2
Data memory
1
0
MUXtwo
datai
16
Sign extend
sign
32
func
ALU control
ctrlalu

Input or output	Signal name	addi	R-format	lw	sw	beq	jump
Inputs of control	Op5	0	0	1	1	0	1
	Op4	0	0	0	0	0	1
	Op3	0	0	0	1	0	1
	Op2	0	0	0	0	1	1
	Op1	0	0	1	1	0	1
	Op0	1	0	1	1	0	1
Outputs of control	RegDst	0	1	0	X	X	X
	ALUSrc	1	0	1	1	0	X
	MemtoReg	0	0	1	X	X	X
	RegWrite	1	1	1	0	0	X
	MemRead	0	0	1	0	0	X
	MemWrite	0	0	0	1	0	X
	Branch	0	0	0	0	1	0
	ALUOp1	1	1	0	0	0	X
	ALUOp0	0	0	0	0	1	X

ALUOp		Func						ctrlalu	OP	instruction
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0			
0	0	X	X	X	X	X	X	0010	100011	LW
0	1	X	X	X	X	X	X	0110	101011	SW
1	0	1	0	0	0	0	0	0010	000000	ADD
1	0	1	0	0	0	1	0	0110	000000	SUB
1	0	1	0	0	1	0	0	0000	000000	AND
1	0	1	0	0	1	0	1	0001	000000	OR
X	X	X	X	X	X	X	X	0010	000001	addi
X	X	X	X	X	X	X	X	XXXX	000100	beq
X	X	X	X	X	X	X	X	XXXX	111111	jump

	inst					
instruction	op	rs	rt	rd	shift	func
add / 32'd3	000000	00011	00010	00001	00000	100000
sub / 32'd4	000000	00011	00010	00001	00000	100010
and / 32'd5	000000	00011	00010	00001	00000	100100
or / 32'd6	000000	00011	00010	00001	00000	100101

	inst			
instruction	op	rs	rt	datai
LW / 32'd0	100011	00011	00010	0000100000000000
SW / 32'd1	101011	00011	00010	0000100000000000
beq / 32'd2	000100	00011	00011	0001100000000000
addi / 32'd7	000001	00011	00010	0000100000100000

	inst	
instruction	op	jump
jump / 32'd8	111111	00011000100000100000100000