

Instruction	Mnemonic	Action
Load word	LD ws, offset(rs1)	ws := mem16[rs1 + offset]
Store word	ST rs2, offset(rs1)	mem16[rs1 + offset] := rs2
Add	ADD ws, rs1, rs2	ws := rs1 + rs2
Subtract	SUB ws, rs1, rs2	ws := rs1 - rs2
Invert (1s complement)	INV ws, rs1	ws := !rs1
Logical Shift Left	LSL ws, rs1, rs2	ws := rs1 << rs2
Logical Shift Right	LSR ws, rs1, rs2	ws := rs1 >> rs2
Bitwise AND	AND ws, rs1, rs2	ws := rs1 & rs2
Bitwise OR	OR ws, rs1, rs2	ws := rs1 rs2
Set on Less Than	SLT ws, rs1, rs2	ws := 1 if rs1 < rs2 ws := 0 if rs1 >= rs2
Branch on Equal	BEQ rs1, rs2, offset	PC := PC + 2 + offset << 1 if rs1 != rs2
Branch on Not Equal	BNE rs1, rs2, offset	PC := PC + 2 + offset << 1 if rs1 == rs2
Jump	JMP offset	PC := {PC[15:13], (offset << 1)}

OP	Instruction
0000	Load word
0001	Store word
0002	Add
0003	Subtract
0004	Invert (1s complement)
0005	Logical Shift Left
0006	Logical Shift Right
0007	Bitwise AND
0008	Bitwise OR
0009	Set on Less Than
0011	Branch on Equal
0012	Branch on Not Equal
0013	Jump

Load word

Op	rs1	ws	offset
4	3	3	6

Store word

Op	rs1	rs2	offset
4	3	3	6

Data processing

Op	rs1	rs2	ws	
4	3	3	3	3

Branch

Op	rs1	rs2	offset
4	3	3	6

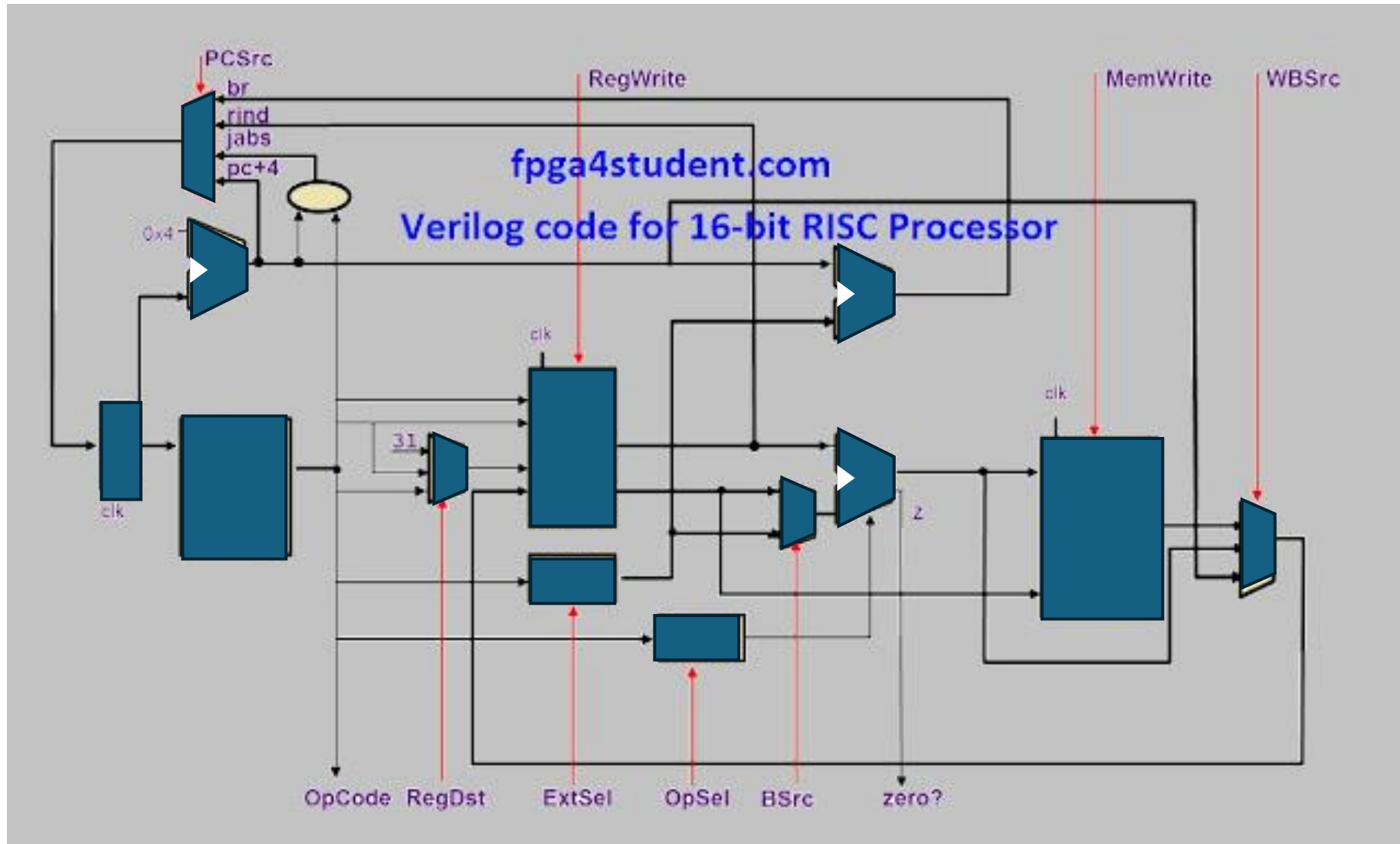
Jump

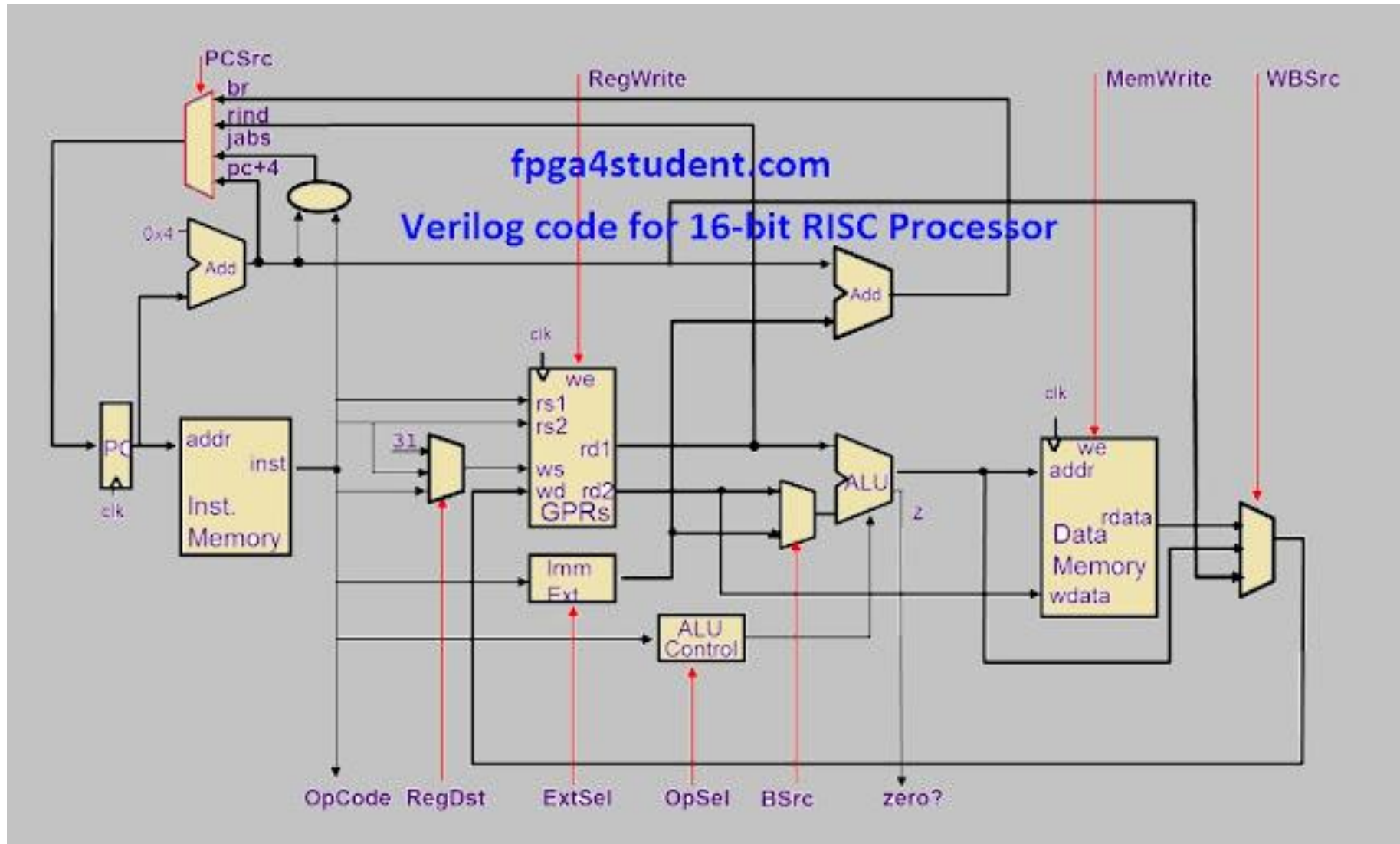
Op	offset
4	12

Control signals									
Instruction	Reg Dst	ALUSrc	Memto Reg	Reg Write	Mem Read	Mem Write	Branch	ALUOp	Jump
Data-processing	1	0	0	1	0	0	0	see below	0
LW	0	1	1	1	1	0	0	000	0
SW	0	1	0	0	0	1	0	000	0
BEQ,BNE	0	0	0	0	0	0	1	001	0
J	0	0	0	0	0	0	0	000	1

ALU Op			
Opcode	Instruction	ALU Operation	ALU Op
02	D-type: ADD	ADD	000
03	D-type: SUB	SUB	001
04	D-type: INVERT	INVERT	010
05	D-type: LSL	LSL	011
06	D-type: LSR	LSR	100
07	D-type: AND	AND	101
08	D-type: OR	OR	110
09	D-type: SLT	SLT	111

ALU Op			
Opcode	Instruction	ALU Operation	ALUop
00	LW	ADD	000
01	SW	SUB	001
02	D-type: ADD	ADD	000
03	D-type: SUB	SUB	001
04	D-type: INVERT	INVERT	010
05	D-type: LSL	LSL	011
06	D-type: LSR	LSR	100
07	D-type: AND	AND	101
08	D-type: OR	OR	110
09	D-type: SLT	SLT	111
11	BEQ	SUB	001
12	BNE	SUB	001
13	JMP	ADD	000





ALU Control				
ALUOp	Opcode (hex)	ALUcnt	ALU Operation	Instruction
10	xxxx	000	ADD	LW,SW
01	xxxx	001	SUB	BEQ,BNE
00	0002	000	ADD	D-type: ADD
00	0003	001	SUB	D-type: SUB
00	0004	010	INVERT	D-type: INVERT
00	0005	011	LSL	D-type: LSL
00	0006	100	LSR	D-type: LSR
00	0007	101	AND	D-type: AND
00	0008	110	OR	D-type: OR
00	0009	111	SLT	D-type: SLT