OPCODES

Following are the opcode and function values chosen for the instruction set. The opcodes have been assigned according to the class of the instruction to provide flexibility for addition of more codes in any class.

CLASS	INSTRUCTION	TYPE	OPCODE	FUNCTION	RS	RT	SH	JTA
ARITHMETIC	add rs,rt	R	000000	000001	rs	rt	-	-
	comp rs,rt	R	000000	000010	rs	rt	-	-
	addi rs,imm	I	000001	-	rs	-	-	-
	compi rs,imm	I	000010	-	rs	-	-	-
LOGIC	and rs,rt	R	100000	000001	rs	rt	-	-
	xor rs,rt	R	100000	000010	rs	rt	-	-
SHIFT	shll rs,sh	R	110000	000000	rs	-	sh	-
	shrl rs,sh	R	110000	100000	rs	-	sh	-
	shllv rs,rt	R	110000	000001	rs	rt	-	-
	shrlv rs,rt	R	110000	100001	rs	rt	-	-
	shra rs,sh	R	110000	100010	rs	-	sh	-
	shrav rs, rt	R	110000	100011	rs	rt	-	-
MEMORY	lw rt,imm(rs)	I	110100	-	rs	rt	-	-
	sw rt,imm(rs)	I	110101	-	rs	rt	-	-
BRANCH	b L	J	010001	-	-	-	-	L
	br rs	R	010000	000001	rs	-	-	-
	bz L	J	010010	-	-	-	-	L
	bnz L	J	010011	-	-	-	-	L
	bcy L	J	010100	-	-	-	-	L
	bncy L	J	010101	-	-	-	-	L
	bs L	J	010110	-	-	-	-	L
	bns L	J	010111	-	-	-	-	L
	bv L	J	011000	-	-	-	-	L
	bnv L	J	011001	-	-	-	-	L
	Call L	J	011010	-	-	-	-	L
	Ret	R	010000	000010	11111	-	-	-

CONTROL SIGNALS

Following is the truth table for various control signals used as mentioned in the schematic diagrams. ‘X’ denotes don’t care.

OPCODE	**FUNCTION**	**Next Address**		**Register File**			**Shifter**	
		brType[2:0]	PCSel[1:0]	regDst[1:0]	writeEn	RegImm	AL	leftright
000000	000001	XXX	00	00	1	X	X	X
000000	000010	XXX	00	00	1	X	X	X
000001	-	XXX	00	00	1	X	X	X
000010	-	XXX	00	00	1	X	X	X
100000	000001	XXX	00	00	1	X	X	X
100000	000010	XXX	00	00	1	X	X	X
110000	000000	XXX	00	00	1	0	1	1
110000	100000	XXX	00	00	1	0	1	0
110000	000001	XXX	00	00	1	1	1	1
110000	100001	XXX	00	00	1	1	1	0
110000	100010	XXX	00	00	1	0	0	0
110000	100011	XXX	00	00	1	1	0	0
110100	-	XXX	00	01	1	X	X	X
110101	-	XXX	00	XX	0	X	X	X
010001	-	XXX	01	XX	0	X	X	X
010000	000001	XXX	11	XX	0	X	X	X
010010	-	100	10	XX	0	X	X	X
010011	-	000	10	XX	0	X	X	X
010100	-	101	10	XX	0	X	X	X
010101	-	001	10	XX	0	X	X	X
010110	-	110	10	XX	0	X	X	X
010111	-	010	10	XX	0	X	X	X
011000	-	111	10	XX	0	X	X	X
011001	-	011	10	XX	0	X	X	X
011010	-	XXX	01	11	1	X	X	X
010000	000010	XXX	11	XX	0	X	X	X

OPCODE	FUNCTION	Data Cache		ALU				
		Dread	Dwrite	ALUimm	ALUfn	logicfn	fnClass	regInData[1:0]
000000	000001	0	0	0	0	X	0	01
000000	000010	0	0	0	1	X	0	01
000001	-	0	0	1	0	X	0	01
000010	-	0	0	1	1	X	0	01
100000	000001	0	0	X	X	0	1	01
100000	000010	0	0	X	X	1	1	01
110000	000000	0	0	X	X	X	X	11
110000	100000	0	0	X	X	X	X	11
110000	000001	0	0	X	X	X	X	11
110000	100001	0	0	X	X	X	X	11
110000	100010	0	0	X	X	X	X	11
110000	100011	0	0	X	X	X	X	11
110100	-	1	0	1	0	X	0	00
110101	-	0	1	1	0	X	0	XX
010001	-	0	0	X	X	X	X	XX
010000	000001	0	0	X	X	X	X	XX
010010	-	0	0	X	X	X	X	XX
010011	-	0	0	X	X	X	X	XX
010100	-	0	0	X	X	X	X	XX
010101	-	0	0	X	X	X	X	XX
010110	-	0	0	X	X	X	X	XX
010111	-	0	0	X	X	X	X	XX
011000	-	0	0	X	X	X	X	XX
011001	-	0	0	X	X	X	X	XX
011010	-	0	0	X	X	X	X	10
010000	000010	0	0	X	X	X	X	XX

RTL Operations

ALU (Without shifter)

	0	1
ALUIMM	b<-(rt)	b<-imm
ALUFN	a<-(rs)	a<-32'b0
LOGICFN	L<-(rs)&(rt)	L<-(rs)^(rt)
FNCLASS	ALUOut<-L	ALUOut<-S

Next Address Architecture

	000	001	010	011	100	101	110	111
BRTYPE[2:0]	brTrue<-!zFlag	brTrue<-!carryFlag	brTrue<-!signFlag	brTrue<-!overflowFlag	brTrue<-zFlag	brTrue<-carryFlag	brTrue<-signFlag	brTrue<-overflowFlag
PCSEL[1:0] (LOWER2 BITS)	NextPC<-IncrPC	NextPC<-jta	NextPC<-branchAddr	NextPC<-register				

Register file

	00	01	10	11
REGDST[1:0]	writeAdd<-rs	writeAdd<-rt		writeAdd<-$31
WRITEEN (LOWER BIT)	don't write	(writeAdd)<-writeData		

Shifter

	0	1
REGIMM	shift by sh	shift by (rt)
AL	Arithmetic shift	Logical Shift
LEFTRIGHT	right shift	left shift