				Rsrc			Rdest					ROM2								ROM1								ROM0								ROM2	ROM1	ROM0		
				rdOE	rs2OE	rs3OE	rs3WE	rs2WE	rdWE	rs1WE	rs0WE	reserved	reserved	reserved	brWE	br2	br1	br0	sf	reserved	reserved	immOE	memWE	memOE	outEN	imem	imm	aluOP5	aluOP4	aluOP3	aluOP2	aluOP1	aluOP0	hlt	no-op					
				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
dec addr	assy mne	# Cycles	description																																				dec addr	hex addr
00	nop	1	nop	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00	0x00	0x00	00	00
01	hlt	1	hlt stop clock	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0x00	0x00	0x02	01	01
02	brk	1	brk stop clock	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0x00	0x00	0x02	02	02
03	mov	1	mov rdest, rsrc	x	x	x	x	x	0	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00	0x00	0x00	03	03
04	movi	2	movi rdest, imm	0	0	0	x	x	0	x	x	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0x00	0x21	0x00	04	04
05	mrd	1	mrd rdest, mem[iem]	0	0	0	x	x	0	x	x	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0x00	0x12	0x00	05	05
06	mwt	1	mwt mem[iem], rsrc	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0x00	0x0A	0x00	06	06
07	mwri	2	mwri mem[iem], imm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0x00	0x29	0x00	07	07
08	bcar	2	pc = br(bem) if carry	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x13	0x00	0x00	08	08
09	bzer	2	pc = br(bem) if rs1=0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x14	0x00	0x00	09	09
10	j	2	pc = jmp(bem)	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x18	0x00	0x00	10	0A
11	add	1	rd = rs1 + rs2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0x00	0x00	0x84	11	0B
12	sub	1	rd = rs1 - rs2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0x00	0x00	0x88	12	0C
13	addi	2	rd = rs1 + imm	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	1	1	0	0	0x00	0x21	0x8C	13	0D
14	subi	2	rd = rs1 - imm	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	0	0	0	0	0x00	0x21	0x90	14	0E
15	and	1	rd = rs1 AND rs2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0x00	0x00	0x94	15	0F
16	or	1	rd = rs1 OR rs2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0x00	0x00	0x98	16	10
17	xor	1	rd = rs1 XOR rs2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0x00	0x00	0x9C	17	11
18	andi	2	rd = rs1 AND imm	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0x00	0x21	0xA0	18	12
19	ori	2	rd = rs1 OR imm	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	1	0	0	0x00	0x21	0xA4	19	13
20	sll	1	rd = sll(rs1)	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0x00	0x00	0x68	20	14
21	srl	1	rd = srl(rs1)	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0x00	0x00	0x6C	21	15
22	rol	1	rd = rol(rs1)	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0x00	0x00	0x70	22	16
23	ror	1	rd = ror(rs1)	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0x00	0x00	0x74	23	17
24	sld	1	sld = rs0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0x00	0x00	0x78	24	18
25	out	1	disp = Rout	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0x00	0x04	0x00	25	19
26	reserved			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00	0x00	0x00	26	1A
27	reserved			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00	0x00	0x00	27	1B
28	reserved			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00	0x00	0x00	28	1C
29	reserved			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00	0x00	0x00	29	1D
30	reserved			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00	0x00	0x00	30	1E
31	reserved			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00	0x00	0x00	31	1F

Note: 2-cycle commands have a imm or bem 16-bit value in the next PM address
Note: iem is an 8-bit memory address
Note: D is the data bus

Control Signal List

Branch	CTRL0	br0
	CTRL1	br1
	CTRL2	br2
Branch Re	brWE	
PC	none	
Prog Mem	none	
rs3	rs3WE	
	rs3OE	
rs2	rs2WE	

	rs2OE
Decoder	none
Imm Reg	imm
	imem
	immOE
RAM	memWE
	memOE
Out Reg	outEN
rs1	rs1WE
rs0	rs0WE
ALU	aluop0
	aluop1
	aluop2
	aluop3
	aluop4
	aluop5
Flag Reg	sf
rd	rdWE
	rdOE