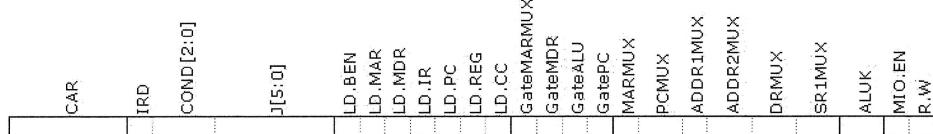


Fundamental Laws of Boolean Algebra

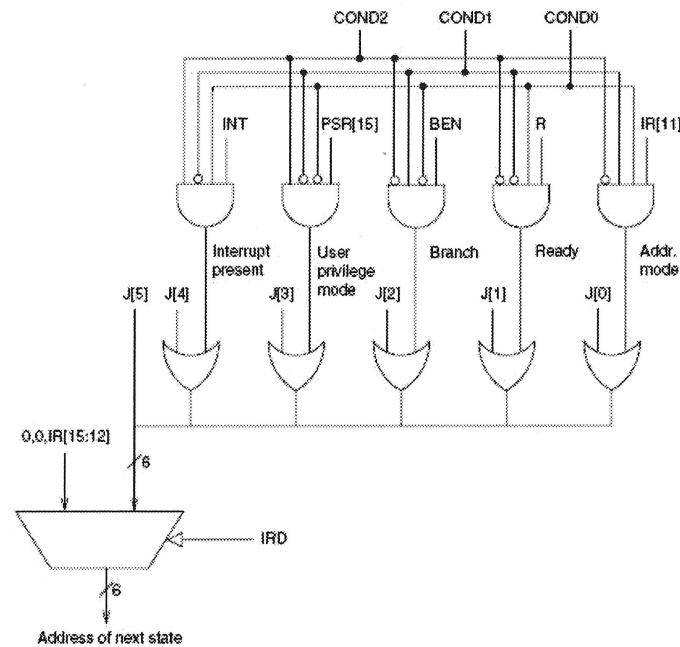
Commutativity	$x \cdot y = y \cdot x$	$x + y = y + x$
Associativity	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	$(x + y) + z = x + (y + z)$
Distributivity	$x \cdot (y + z) = xy + xz$	$x + yz = (x + y)(x + z)$
Idempotence	$x \cdot x = x$	$x + x = x$
Identity	$x \cdot 1 = x$	$x + 0 = x$
Null	$x \cdot 0 = 0$	$x + 1 = 1$
Complementarity	$x \cdot x' = 0$	$x + x' = 1$
Involution		$(x')' = x$
DeMorgan's	$(x \cdot y)' = x' + y'$	$(x + y)' = x' \cdot y'$
Absorption	$x \cdot (x + y) = x$	$x + x \cdot y = x$
No-Name	$x \cdot (x' + y) = x \cdot y$	$x + x' \cdot y = x + y$
Consensus	$(x+y) \cdot (y+z) \cdot (x'+z) = (x+y) \cdot (x'+z)$	$x \cdot y + y \cdot z + x' \cdot z = x \cdot y + x' \cdot z$

Table of ASCII Characters

Char	Dec	Hex		Char	Dec	Hex		Char	Dec	Hex		Char	Dec	Hex
(nul)	0	00		(sp)	32	20		@	64	40		`	96	60
(soh)	1	01		!	33	21		A	65	41		a	97	61
(stx)	2	02		"	34	22		B	66	42		b	98	62
(etx)	3	03		#	35	23		C	67	43		c	99	63
(eot)	4	04		\$	36	24		D	68	44		d	100	64
(enq)	5	05		%	37	25		E	69	45		e	101	65
(ack)	6	06		&	38	26		F	70	46		f	102	66
(bel)	7	07		'	39	27		G	71	47		g	103	67
(bs)	8	08		(40	28		H	72	48		h	104	68
(ht)	9	09)	41	29		I	73	49		i	105	69
(nl)	10	0a		*	42	2a		J	74	4a		j	106	6a
(vt)	11	0b		+	43	2b		K	75	4b		k	107	6b
(np)	12	0c		,	44	2c		L	76	4c		l	108	6c
(cr)	13	0d		-	45	2d		M	77	4d		m	109	6d
(so)	14	0e		.	46	2e		N	78	4e		n	110	6e
(si)	15	0f		/	47	2f		O	79	4f		o	111	6f
(dle)	16	10		0	48	30		P	80	50		p	112	70
(dc1)	17	11		1	49	31		Q	81	51		q	113	71
(dc2)	18	12		2	50	32		R	82	52		r	114	72
(dc3)	19	13		3	51	33		S	83	53		s	115	73
(dc4)	20	14		4	52	34		T	84	54		t	116	74
(nak)	21	15		5	53	35		U	85	55		u	117	75
(syn)	22	16		6	54	36		V	86	56		v	118	76
(etb)	23	17		7	55	37		W	87	57		w	119	77
(can)	24	18		8	56	38		X	88	58		x	120	78
(em)	25	19		9	57	39		Y	89	59		y	121	79
(sub)	26	1a		:	58	3a		Z	90	5a		z	122	7a
(esc)	27	1b		;	59	3b		[91	5b		{	123	7b
(fs)	28	1c		<	60	3c		\	92	5c		}	124	7c
(gs)	29	1d		=	61	3d]	93	5d		~	125	7d
(rs)	30	1e		>	62	3e		^	94	5e		(del)	126	7e
(us)	31	1f		?	63	3f		_	95	5f			127	7f

LC-3 Control Word Fields**LC-3 Microsequencer Control**

Signal	Description
IRD	$\begin{cases} = 1, \text{CAR} \leftarrow 00 \text{ opcode (opcode = IR[15:12])}, \text{only during decode} \\ = 0, \text{CAR} \leftarrow J \text{ (plus 1,2,4,8,16 depending on COND bits)} \end{cases}$
COND	$\begin{cases} = 000, \text{CAR} \leftarrow J \\ = 001, \text{IF } (R=1 \text{ and } J[1]=0) \text{ THEN } (\text{CAR} \leftarrow J \text{ plus 2}) \text{ ELSE } (\text{CAR} \leftarrow J) \\ = 010, \text{IF } (\text{BEN}=1 \text{ and } J[2]=0) \text{ THEN } (\text{CAR} \leftarrow J \text{ plus 4}) \text{ ELSE } (\text{CAR} \leftarrow J) \\ = 011, \text{IF } (\text{IR}[11]=1 \text{ and } J[0]=0) \text{ THEN } (\text{CAR} \leftarrow J \text{ plus 1}) \text{ ELSE } (\text{CAR} \leftarrow J) \end{cases}$
J	6-bit next value for CAR (plus modifications depending on COND bits)

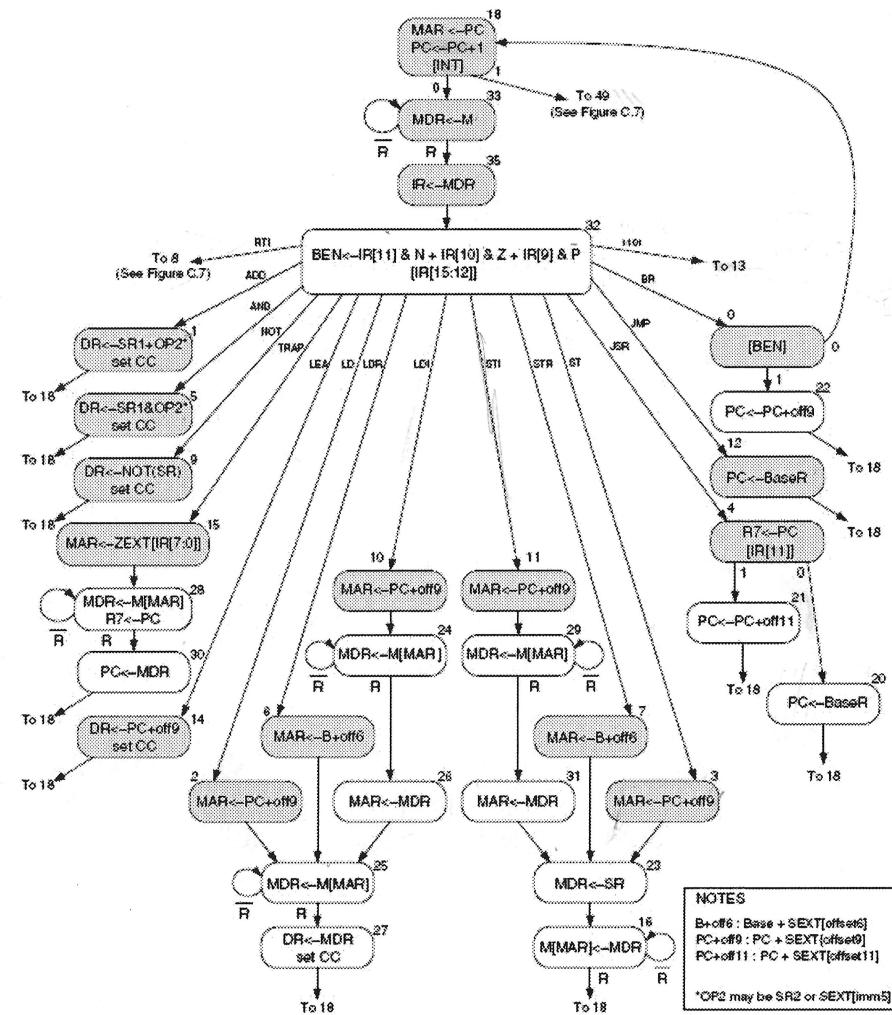


LC-3 Instructions

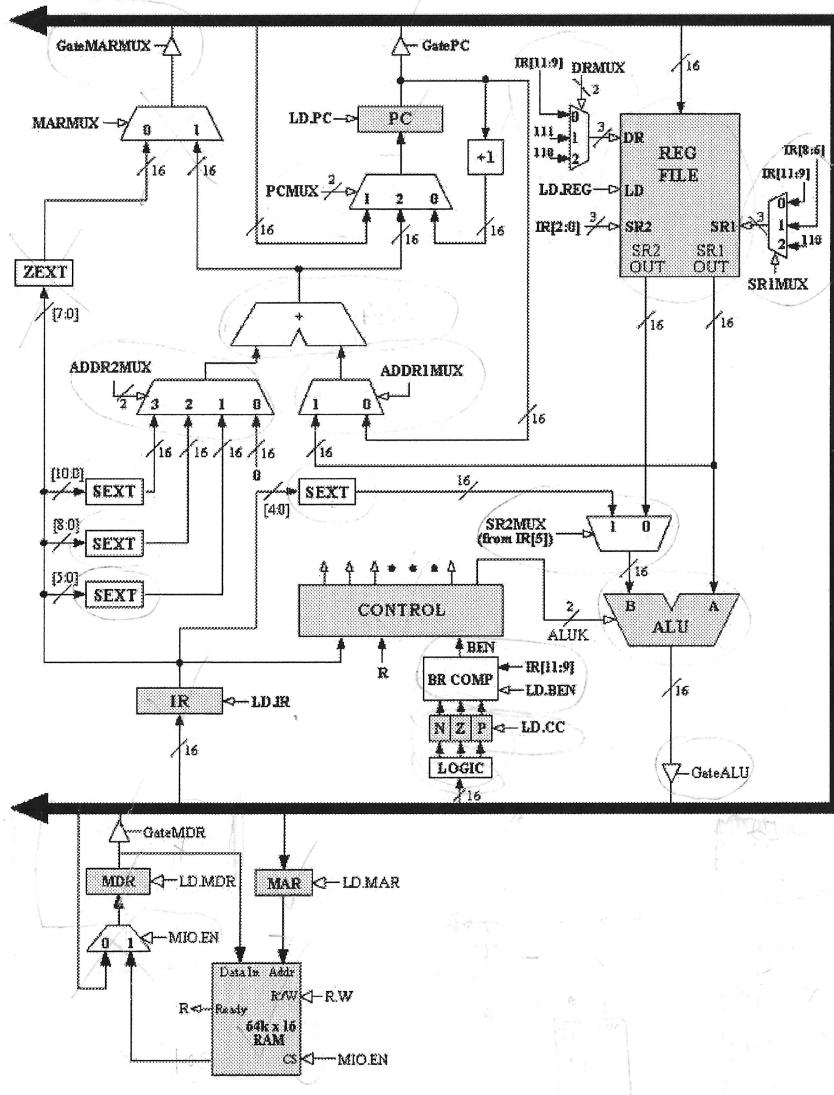
ADD	0001	DR	SR1	0	00	SR2	ADD DR, SR1, SR2	LD	0010	DR	PCoffset9	LD DR, PCoffset9
							DR ← SR1 + SR2, Setcc					DR ← MPC + SEXT[PCoffset9], Setcc
ADD	0001	DR	SR1	1		imm5	ADD DR, SR1, imm5	LDI	1010	DR	PCoffset9	LDI DR, PCoffset9
							DR ← SR1 + SEXT[imm5], Setcc					DR ← MM[MP] + SEXT[PCoffset9], Setcc
AND	0101	DR	SR1	0	00	SR2	AND DR, SR1, SR2	LDR	0110	DR	BaseR	LDR DR, BaseR, offset6
							DR ← SR1 AND SR2, Setcc					DR ← MB[BaseR] + SEXT[offset6], Setcc
AND	0101	DR	SR1	1		imm5	AND DR, SR1, imm5	LEA	1110	DR	PCoffset9	LEA DR, PCoffset9
							DR ← SR1 AND SEXT[imm5], Setcc					DR ← PC + SEXT[PCoffset9], Setcc
BR	0000	n	z	p		PCoffset9	BR{nzp} PCoffset9	NOT	1001	DR	SR	NOT DR, SR
							((n AND N) OR (z AND Z) OR (p AND P)): PC ← PC + SEXT[PCoffset9]					DR ← NOT SR, Setcc
JMP	1100	000	BaseR	000000			JMP BaseR	ST	0011	SR	PCoffset9	ST SR, PCoffset9
							PC ← BaseR					M[PC + SEXT[PCoffset9]] ← SR
JSR	0100	1			PCoffset11		JSR PCoffset11	STI	1011	SR	PCoffset9	STI SR, PCoffset9
							R7 ← PC, PC ← PC + SEXT[PCoffset11]					M[MP + SEXT[PCoffset9]] ← SR
TRAP	1111	0000			trapvect8		TRAP trapvect8					STR SR, BaseR, offset6
							R7 ← PC, PC ← M[ZEXT[trapvect8]]					

NOTES: RTL corresponds to execution (after fetch()); JSRR not shown

LC-3 FSM



LC-3 Datapath



LC-3 Datapath Control Signals

Signal	Description	Signal	Description
LD.MAR	= 1, MAR is loaded	LD.CC	= 1, updates status bits from system bus
LD.MDR	= 1, MDR is loaded	GateMARMUX	= 1, MARMUX output is put onto system bus
LD.IR	= 1, IR is loaded	GateMDR	= 1, MDR contents are put onto system bus
LD.PC	= 1, PC is loaded	GateALU	= 1, ALU output is put onto system bus
LD.REG	= 1, register file is loaded	GatePC	= 1, PC contents are put onto system bus
LD.BEN	= 1, updates Branch Enable (BEN) bit	MIO.EN	$\begin{cases} = 1, \text{ Enables memory,} \\ \quad \text{chooses memory output for MDR input} \\ = 0, \text{ Disables memory,} \\ \quad \text{chooses system bus for MDR input} \end{cases}$
MARMUX	$\begin{cases} = 0, \text{ chooses ZEXT IR[7:0]} \\ = 1, \text{ chooses address adder output} \end{cases}$	R.W	$\begin{cases} = 1, \text{ M[MAR] < MDR when MIO.EN = 1} \\ = 0, \text{ MDR < M[MAR] when MIO.EN = 1} \end{cases}$
ADDR1MUX	$\begin{cases} = 0, \text{ chooses PC} \\ = 1, \text{ chooses reg file SR1 OUT} \end{cases}$	ALUK	$\begin{cases} = 00, \text{ ADD} \\ = 01, \text{ AND} \\ = 10, \text{ NOT A} \\ = 11, \text{ PASS A} \end{cases}$
ADDR2MUX	$\begin{cases} = 00, \text{ chooses "0..00"} \\ = 01, \text{ chooses SEXT IR[5:0]} \\ = 10, \text{ chooses SEXT IR[8:0]} \\ = 11, \text{ chooses SEXT IR[10:0]} \end{cases}$	DRMUX	$\begin{cases} = 00, \text{ chooses IR[11:9]} \\ = 01, \text{ chooses "111"} \\ = 10, \text{ chooses "110"} \end{cases}$
PCMUX	$\begin{cases} = 00, \text{ chooses PC + 1} \\ = 01, \text{ chooses system bus} \\ = 10, \text{ chooses address adder output} \end{cases}$	SR1MUX	$\begin{cases} = 00, \text{ chooses IR[11:9]} \\ = 01, \text{ chooses IR[8:6]} \\ = 10, \text{ chooses "110"} \end{cases}$