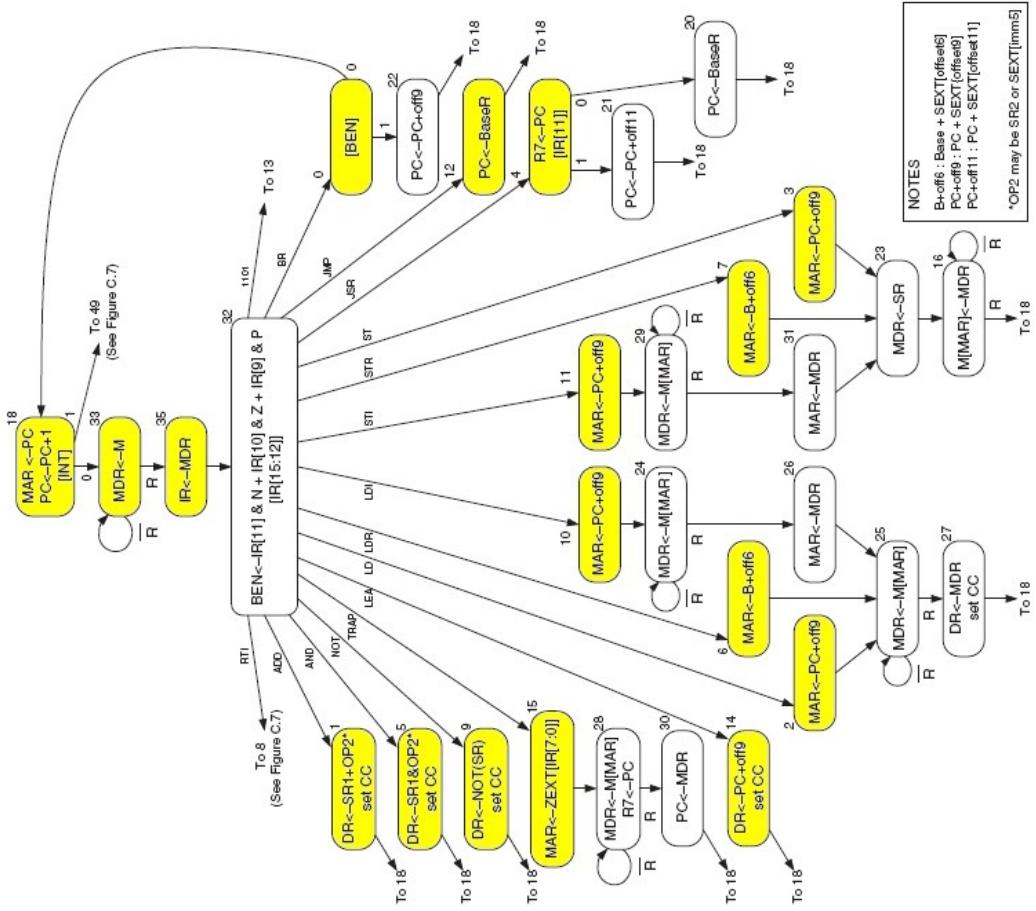


## LC-3 Instructions

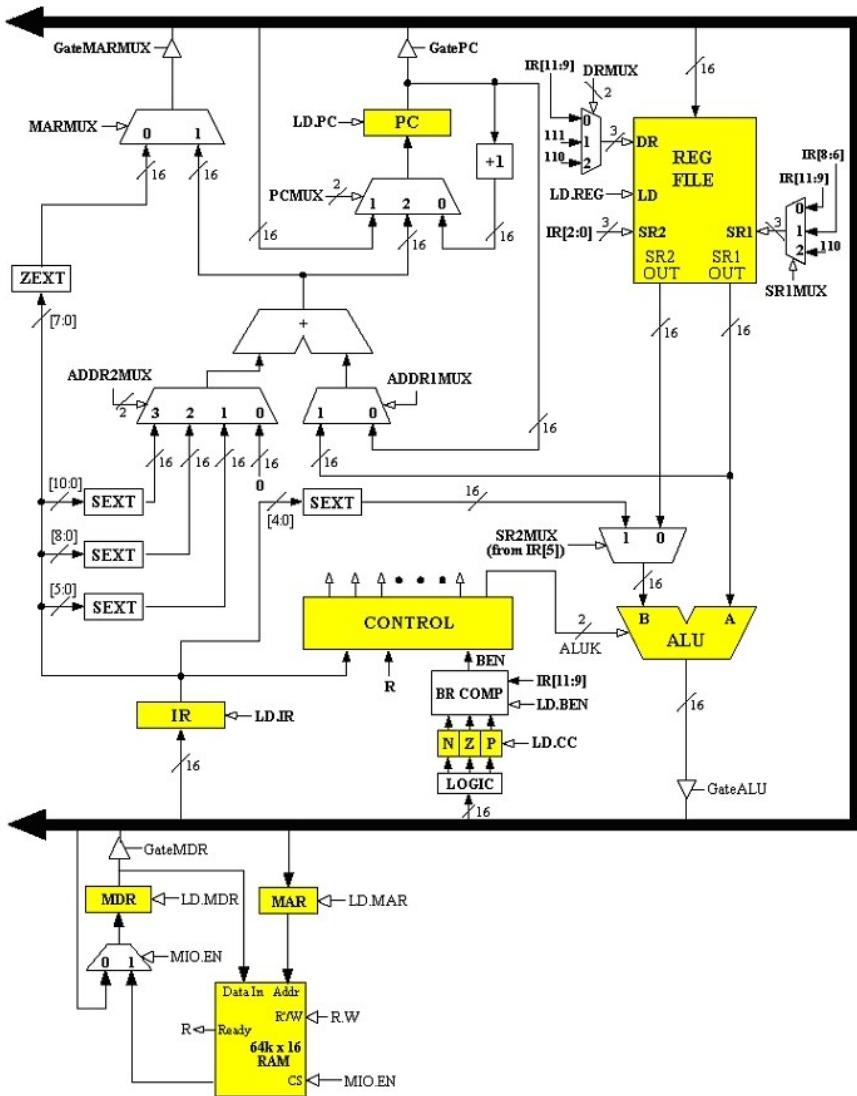
## LC-3 FSM



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

ADD	<table border="1"><tr><td>0001</td><td>DR</td><td>SR1</td><td>0</td><td>00</td><td>SR2</td></tr></table>	0001	DR	SR1	0	00	SR2	ADD DR, SR1, SR2	LD	<table border="1"><tr><td>0010</td><td>DR</td><td colspan="3">PCoffset9</td></tr></table>	0010	DR	PCoffset9			LD DR, PCoffset9
0001	DR	SR1	0	00	SR2											
0010	DR	PCoffset9														
	DR $\leftarrow$ SR1 + SR2, Setcc			DR $\leftarrow$ M[PC + SEXT(PCoffset9)], Setcc												
ADD	<table border="1"><tr><td>0001</td><td>DR</td><td>SR1</td><td>1</td><td colspan="2">imm5</td></tr></table>	0001	DR	SR1	1	imm5		ADD DR, SR1, imm5	LDI	<table border="1"><tr><td>1010</td><td>DR</td><td colspan="3">PCoffset9</td></tr></table>	1010	DR	PCoffset9			LDI DR, PCoffset9
0001	DR	SR1	1	imm5												
1010	DR	PCoffset9														
	DR $\leftarrow$ SR1 + SEXT(imm5), Setcc			DR $\leftarrow$ M[M[PC + SEXT(PCoffset9)]], Setcc												
AND	<table border="1"><tr><td>0101</td><td>DR</td><td>SR1</td><td>0</td><td>00</td><td>SR2</td></tr></table>	0101	DR	SR1	0	00	SR2	AND DR, SR1, SR2	LDR	<table border="1"><tr><td>0110</td><td>DR</td><td>BaseR</td><td colspan="2">offset6</td></tr></table>	0110	DR	BaseR	offset6		LDR DR, BaseR, offset6
0101	DR	SR1	0	00	SR2											
0110	DR	BaseR	offset6													
	DR $\leftarrow$ SR1 AND SR2, Setcc			DR $\leftarrow$ M[BaseR + SEXT(offset6)], Setcc												
AND	<table border="1"><tr><td>0101</td><td>DR</td><td>SR1</td><td>1</td><td colspan="2">imm5</td></tr></table>	0101	DR	SR1	1	imm5		AND DR, SR1, imm5	LEA	<table border="1"><tr><td>1110</td><td>DR</td><td colspan="3">PCoffset9</td></tr></table>	1110	DR	PCoffset9			LEA DR, PCoffset9
0101	DR	SR1	1	imm5												
1110	DR	PCoffset9														
	DR $\leftarrow$ SR1 AND SEXT(imm5), Setcc			DR $\leftarrow$ PC + SEXT(PCoffset9), Setcc												
BR	<table border="1"><tr><td>0000</td><td>n</td><td>z</td><td>p</td><td colspan="2">PCoffset9</td></tr></table>	0000	n	z	p	PCoffset9		BR(nzp) PCoffset9 ((n AND N) OR (z AND Z) OR (p AND P)): PC $\leftarrow$ PC + SEXT(PCoffset9)	NOT	<table border="1"><tr><td>1001</td><td>DR</td><td>SR</td><td colspan="2">111111</td></tr></table>	1001	DR	SR	111111		NOT DR, SR
0000	n	z	p	PCoffset9												
1001	DR	SR	111111													
				DR $\leftarrow$ NOT SR, Setcc												
JMP	<table border="1"><tr><td>1100</td><td>000</td><td>BaseR</td><td colspan="3">000000</td></tr></table>	1100	000	BaseR	000000			JMP BaseR	ST	<table border="1"><tr><td>0011</td><td>SR</td><td colspan="3">PCoffset9</td></tr></table>	0011	SR	PCoffset9			ST SR, PCoffset9
1100	000	BaseR	000000													
0011	SR	PCoffset9														
	PC $\leftarrow$ BaseR			M[PC + SEXT(PCoffset9)] $\leftarrow$ SR												
JSR	<table border="1"><tr><td>0100</td><td>1</td><td colspan="3">PCoffset11</td></tr></table>	0100	1	PCoffset11			JSR PCoffset11	STI	<table border="1"><tr><td>1011</td><td>SR</td><td colspan="3">PCoffset9</td></tr></table>	1011	SR	PCoffset9			STI SR, PCoffset9	
0100	1	PCoffset11														
1011	SR	PCoffset9														
	R7 $\leftarrow$ PC, PC $\leftarrow$ PC + SEXT(PCoffset11)			M[M[PC + SEXT(PCoffset9)]] $\leftarrow$ SR												
TRAP	<table border="1"><tr><td>1111</td><td>0000</td><td colspan="3">trapvect8</td></tr></table>	1111	0000	trapvect8			TRAP trapvect8	STR	<table border="1"><tr><td>0111</td><td>SR</td><td>BaseR</td><td colspan="2">offset6</td></tr></table>	0111	SR	BaseR	offset6		STR SR, BaseR, offset6	
1111	0000	trapvect8														
0111	SR	BaseR	offset6													
	R7 $\leftarrow$ PC, PC $\leftarrow$ M[ZEXT(trapvect8)]			M[BaseR + SEXT(offset6)] $\leftarrow$ SR												

## LC-3 Datapath



## LC-3 Datapath Control Signals

Signal	Description	Signal	Description
LD.MAR	LD.MAR = 1, MAR is loaded	LD.CC	LD.CC = 1, updates status bits from system bus
LD.MDR	LD.MDR = 1, MDR is loaded	GateMARMUX	GateMARMUX = 1, MARMUX output is put onto system bus
LD.IR	LD.IR = 1, IR is loaded	GateMDR	GateMDR = 1, MDR contents are put onto system bus
LD.PC	LD.PC = 1, PC is loaded	GateALU	GateALU = 1, ALU output is put onto system bus
LD.REG	LD.REG = 1, register file is loaded	GatePC	GatePC = 1, PC contents are put onto system bus
LD.BEN	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	$\begin{cases} = 1, \text{M[MAR]<-MDR when MIO.EN = 1} \\ = 0, \text{MDR}<\text{-M[MAR] when MIO.EN = 1} \end{cases}$
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	$\begin{cases} = 00, \text{chooses "0...00"} \\ = 01, \text{chooses SEXT[IR[5:0]} \\ = 10, \text{chooses SEXT[IR[8:0]} \\ = 11, \text{chooses SEXT[IR[10:0]} \end{cases}$	PCMUX	$\begin{cases} = 00, \text{chooses PC + 1} \\ = 01, \text{chooses system bus} \\ = 10, \text{chooses address adder output} \end{cases}$
SR1MUX	$\begin{cases} = 00, \text{chooses IR[11:9]} \\ = 01, \text{chooses IR[8:6]} \\ = 10, \text{chooses "110"} \end{cases}$	DRMUX	$\begin{cases} = 00, \text{chooses IR[11:9]} \\ = 01, \text{chooses "111"} \\ = 10, \text{chooses "110"} \end{cases}$