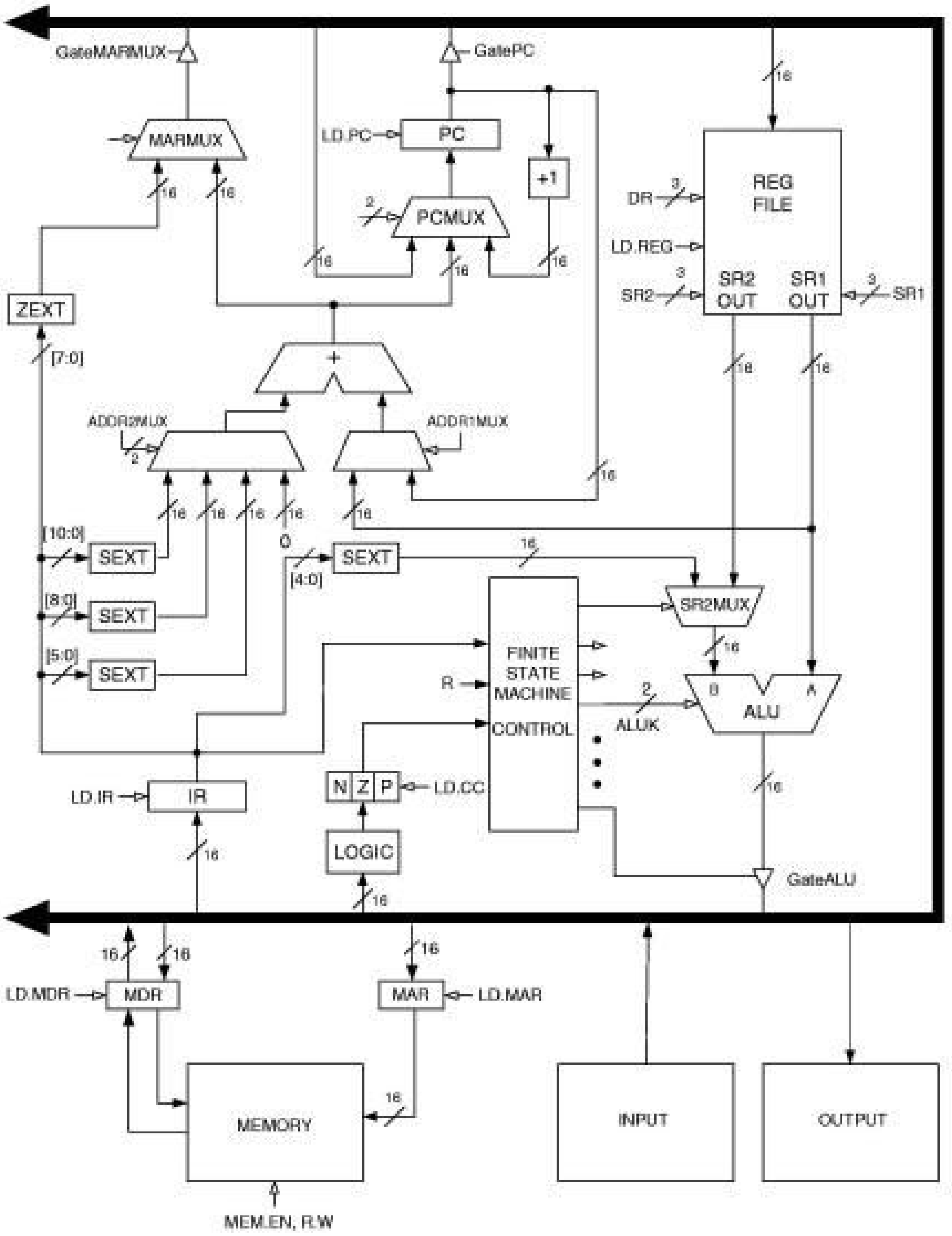
LC-3 instruction datapaths & control signals

Compiled by Daniel Stinson-Diess and Aaroh Mankad, Alex Chen

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD$^+$	0001				DR			SR1			0	00		SR2		
ADD$^+$	0001				DR			SR1			1	imm5				
AND$^+$	0101				DR			SR1			0	00		SR2		
AND$^+$	0101				DR			SR1			1	imm5				
BR	0000				n	z	p	PCoffset9								
JMP	1100				000			BaseR			000000					
JSR	0100				1	PCoffset11										
JSRR	0100				0	00		BaseR			000000					
LD$^+$	0010				DR			PCoffset9								
LDI$^+$	1010				DR			PCoffset9								
LDR$^+$	0110				DR			BaseR			offset6					
LEA$^+$	1110				DR			PCoffset9								
NOT$^+$	1001				DR			SR			111111					
RET	1100				000			111			000000					
RTI	1000				000000000000											
ST	0011				SR			PCoffset9								
STI	1011				SR			PCoffset9								
STR	0111				SR			BaseR			offset6					
TRAP	1111				0000				trapvect8							
reserved	1101															



Not explicitly shown on this simplified schematic:

- MDRMUX: selects either BUS or MEMORY as input to the MDR
- MDR Gate: allows the MDR to write to the bus
- DR MUX: selects either IR[11:9] *(default)* or [111] as input to the DR decoder
- SR1MUX: selects either IR[8:6] *(default)* or IR[11:9] as input to the SR1 decoder

List of all control signals:

GATE: Allows data from a register or other source to be written to the BUS

Gate.ALU BUS <- ALU

Gate.MARMUX BUS <- MARMUX

Gate.MDR BUS <- MDR

Gate.PC BUS <- PC

LD: Write-enables a register

LD.IR IR <- BUS

LD.MAR MAR <- BUS

LD.MDR MDR <- BUS

LD.REG REG[DR] <- BUS

LD.PC PC <- PCMUX

MEM.EN: Activates the Memory

MEM.EN/R MDR <- MEM[MAR] (Read from memory)

MEM.EN/W MEM[MAR]<- (MDR) (Write to memory)

R (from Mem to FSM): Signals that Memory operation is complete

MUX: Selects data from one of multiple sources

only matters if the destination register is write enabled or Gate is open

ADDR1MUX ADDR ADD<- (PC) / SR1bus (choose "starting" value for address calc)

ADDR2MUX ADDR ADD<- SEXT(IR[5:0]) (for STR, LDR)
/ SEXT(IR[8:0]) (PCOffset9 for BR, LD, LDI, LEA, ST, STI)
/ SEXT(IR[10:0]) (PCOffset11 for JSR)
/ 0 (for JMP, JSRR)

DRMUX DR <- IR[11:9] (for most instructions) / [111] (R7 <- (PC))

MARMUX MAR via Gate.Marmux/bus <- ADDR ADD / ZEXT(IR[7:0]) (for TRAP)

MDRMUX LD.MDR <- BUS (for writing) / MEM (for reading)

PCMUX PC <- ADDR ADD / BUS / PC + 1

SR1MUX SR1 decoder <- IR[8:6] (most instructions) / [11:9] (STORE instructions)

SR2MUX ALU.B <- REG[SR2] / SEXT(IR[4:0]) (= imm5)

Other

ALUK ALU <- A + B / A & B / ~A / A

LD.CC NZP cc's write enabled for instructions that write to GPR bank

Instruction Fetch sequence

Register Transfer Notation

IR <= Mem[(PC)]

Control Signals

1. Gate.PC
2. LD.MAR
3. MEM.EN/R
    a. Wait for R
4. MDRMUX selects Memory *(the MDRMUX is not shown in the simplified datapath)*
5. LD.MDR
6. Gate.MDR *(the MDR Gate is not shown in the simplified datapath)*
7. LD.IR

ADD (register mode)

Opcode				DR			SR1			Mode	Unused		SR2		
0	0	0	1	x	x	x	x	x	x	0	0	0	x	x	x

Register Transfer Notation

DR ⇐ (SR1) + (SR2)

Control Signals

1. SR1MUX selects IR[8:6] *(not shown in the simplified datapath)*
   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
2. SR2MUX selects SR2OUT
3. ALUK selects ADD
4. GateALU
5. DRMUX selects IR[11:9] *(not shown in the simplified datapath)*
   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
6. LD.REG

ADD (immediate mode)

Opcode				DR			SR1			Mode	IMM5				
0	0	0	1	x	x	x	x	x	x	1	x	x	x	x	x

Register Transfer Notation

DR ⇐ (SR1) + SEXT(IR[4:0])

Control Signals

1. SR1MUX selects IR[8:6] *(not shown in the simplified datapath)*
   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
2. SR2MUX selects SEXT(IR[4:0])
3. ALUK selects ADD
4. GateALU
5. DRMUX selects IR[11:9] *(not shown in the simplified datapath)*
   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
6. LD.REG

AND (register mode)

Opcode				**DR**			**SR1**			**Mode**	**Unused**		**SR2**		
0	1	0	1	x	x	x	x	x	x	0	0	0	x	x	x

Register Transfer Notation

DR ⇐ (SR1) bitwise and (SR2)

Control Signals

1. SR1MUX selects IR[8:6] *(not shown in the simplified datapath)*
   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
2. SR2MUX selects SR2OUT
3. ALUK selects AND
4. Gate.ALU
5. DRMUX selects IR[11:9] *(not shown in the simplified datapath)*
   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
6. LD.REG

AND (immediate mode)

Opcode				**DR**			**SR1**			**Mode**	**IMM5**				
0	1	0	1	x	x	x	x	x	x	1	x	x	x	x	x

Register Transfer Notation

DR ⇐ (SR1) bitwise and SEXT(IR[4:0])

Control Signals

1. SR1MUX selects IR[8:6] *(not shown in the simplified datapath)*
   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
2. SR2MUX selects IR[4:0]
3. ALUK selects AND
4. Gate.ALU
5. DRMUX selects IR[11:9] *(not shown in the simplified datapath)*
   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
6. LD.REG

NOT

Opcode				DR			SR			Unused					
1	0	0	1	x	x	x	x	x	x	1	1	1	1	1	1

Register Transfer Notation

DR ⇐ bitwise not (SR1)

Control Signals

1. SR1MUX selects IR[8:6] *(not shown in the simplified datapath)*

   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
2. ALUK selects NOT
3. Gate.ALU
4. DRMUX selects IR[11:9] *(not shown in the simplified datapath)*

   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
5. LD.REG

LEA

Opcode				DR			PCOffset9								
1	1	1	0	x	x	x	x	x	x	x	x	x	x	x	x

Register Transfer Notation

DR ⇐ (PC) + SEXT(IR[8:0])

Control Signals

1. ADDR1MUX selects PC
2. ADDR2MUX selects IR[8:0]
3. MARMUX selects ADDR ADD
4. Gate.MARMUX
5. DRMUX selects IR[11:9] *(not shown in the simplified datapath)*

   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
6. LD.REG

LD

Opcode				**DR**			**PCOffset9**								
0	0	1	0	x	x	x	x	x	x	x	x	x	x	x	x

Register Transfer Notation

DR ⇐ Mem[(PC) + SEXT(IR[8:0])]

Control Signals

1. ADDR1MUX selects PC
2. ADDR2MUX selects SEXT(IR[8:0])
3. MARMUX selects ADDR ADD
4. Gate.MARMUX
5. LD.MAR
6. MEM.EN/R
    a. Wait for R
7. MDRMUX selects Memory *(the MDRMUX is not shown in the simplified datapath)*
8. LD.MDR
9. Gate.MDR *(the MDR Gate is not shown in the simplified datapath)*
10. DRMUX selects IR[11:9] *(not shown in the simplified datapath)*
    This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
11. LD.REG

LDI

Opcode				DR			PCOffset9								
1	0	1	0	x	x	x	x	x	x	x	x	x	x	x	x

Register Transfer Notation

DR ⇐ Mem[Mem[(PC) + SEXT(IR[8:0]]]

Control Signals

1. ADDR1MUX selects PC
2. ADDR2MUX selects SEXT(IR[8:0])
3. MARMUX selects ADDR ADD
4. Gate.MARMUX
5. LD.MAR
6. MEM.EN/R
    a. Wait for R
7. MDRMUX selects Memory
8. LD.MDR
9. Gate.MDR
10. LD.MAR
11. MEM.EN/R
    a. Wait for R
12. MDRMUX selects Memory *(the MDRMUX is not shown in the simplified datapath)*
13. LD.MDR
14. Gate.MDR *(the MDR Gate is not shown in the simplified datapath)*
15. DRMUX selects IR[11:9] *(not shown in the simplified datapath)*
    This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
16. LD.REG

LDR

Opcode				DR			BaseR			offset6					
0	1	1	0	x	x	x	x	x	x	x	x	x	x	x	x

Register Transfer Notation

DR ⇐ Mem[(IR[8:6]) + SEXT(IR[5:0]]]

Control Signals

1. SR1MUX selects IR[8:6]

   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
2. ADDR1MUX selects SR1OUT
3. ADDR2MUX selects SEXT(IR[5:0])
4. MARMUX selects ADDR ADD
5. Gate.MARMUX
6. LD.MAR
7. MEM.EN/R
   a. Wait for R

2. MDRMUX selects Memory *(the MDRMUX is not shown in the simplified datapath)*
3. LD.MDR
4. Gate.MDR *(the MDR Gate is not shown in the simplified datapath)*
5. DRMUX selects IR[11:9] *(not shown in the simplified datapath)*

   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
6. LD.REG

ST

Opcode				SR			PCOffset9								
0	0	1	1	x	x	x	x	x	x	x	x	x	x	x	x

Register Transfer Notation

Mem[(PC) + SEXT(IR[8:0]] ⇐ SR

Control Signals

1. ADDR1MUX selects PC
2. ADDR2MUX select SEXT(IR[8:0])
3. MARMUX selects ADDR ADD
4. Gate.MARMUX
5. LD.MAR
6. SR1MUX selects IR[11:9] *(not shown in the simplified datapath)*
7. ALUK selects Pass Through
8. Gate.ALU
9. MDRMUX selects bus *(the MDRMUX is not shown in the simplified datapath)*
10. LD.MDR
11. MEM.EN W
    (Note: no need to wait for R, as there is no subsequent step in this instruction)

STI

Opcode				**SR**			**PCOffset9**								
1	0	1	1	x	x	x	x	x	x	x	x	x	x	x	x

Register Transfer Notation

Mem[Mem[(PC) + SEXT(IR[8:0]]] ⇐ SR

Control Signals

1. ADDR1MUX selects PC
2. ADDR2MUX selects SEXT(IR[8:0])
3. MARMUX selects ADDR ADD
4. Gate.MARMUX
5. LD.MAR
6. MEM.EN/R
    a. Wait for R
7. MDRMUX selects Memory *(the MDRMUX is not shown in the simplified datapath)*
8. LD.MDR
9. Gate.MDR
10. LD.MAR
11. SR1MUX selects IR[11:9] *(not shown in the simplified datapath)*
12. ALUK selects Pass Through
13. Gate.ALUK
14. MDRMUX selects Bus *(the MDRMUX is not shown in the simplified datapath)*
15. LD.MDR
16. MEM.EN/W
    (Note: no need to wait for R, as there is no subsequent step in this instruction)

STR

Opcode				SR			BaseR			offset6					
0	1	1	1	x	x	x	x	x	x	x	x	x	x	x	x

Register Transfer Notation

Mem[(IR[8:6]) + SEXT(IR[5:0]] ⇐ SR

Control Signals

1. SR1MUX selects IR[8:6]
2. ADDR1MUX selects SR1OUT
3. ADDR2MUX selects SEXT(IR[5:0])
4. MARMUX selects ADDR ADD
5. Gate.MARMUX
6. LD.MAR
7. SR1MUX selects IR[11:9] *(not shown in the simplified datapath)*
8. ALUK selects Pass Through
9. Gate.ALU
10. MDRMUX selects Bus *(the MDRMUX is not shown in the simplified datapath)*
11. LD.MDR
12. MEM.EN/W
    (Note: no need to wait for R, as there is no subsequent step in this instruction)

JMP

Opcode				Unused			BaseR			Unused					
1	1	0	0	0	0	0	x	x	x	0	0	0	0	0	0

Register Transfer Notation

PC ⇐ (BaseR)

Control Signals

1. SR1MUX selects IR[8:6] *(not shown in the simplified datapath)*

   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
2. ADDR1MUX selects SR1OUT
3. ADDR2MUX selects 0
4. PCMUX selects ADDR ADD
5. LD.PC

BR

Opcode				n	z	p	PCOffset9								
0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x

Register Transfer Notation

PC ⇐ (PC) + SEXT(IR[8:0]) **iff** (N.n + Z.z + P.p)

Control Signals

1. ADDR1MUX selects PC
2. ADDR2MUX selects SEXT(IR[8:0])
3. PCMUX selects ADDR ADD
4. LD.PC iff (N.n + Z.z + P.p)

JSR

Opcode				Mode	PCOffset11										
0	1	0	0	1	x	x	x	x	x	x	x	x	x	x	x

Register Transfer Notation

R7 ⇐ (PC)
PC ⇐ (PC) + SEXT(IR[10:0])

Control Signals

1. GatePC
2. DRMUX selects [111] *(not shown in the simplified datapath)*
3. LD.REG
4. ADDR1MUX selects PC
5. ADDR2MUX selects SEXT(IR[10:0])
6. PCMUX selects ADDR ADD
7. LD.PC

JSRR

Opcode				Mode	Unused		BaseR			Unused					
0	1	0	0	0	0	0	x	x	x	0	0	0	0	0	0

Register Transfer Notation

R7 ⇐ (PC)
PC ⇐ (BaseR)

Control Signals

1. GatePC
2. DRMUX selects [111] *(not shown in the simplified datapath)*
3. LD.REG
4. SR1MUX selects IR[8:6] *(not shown in the simplified datapath)*
   This does not really need to be mentioned, as it is the default, i.e. the select bit stays 0
5. ADDR1MUX selects SR1OUT
6. ADDR2MUX selects 0
7. PCMUX selects ADDR ADD
8. LD.PC

TRAP

Opcode				Unused				trapvec8							
1	1	1	1	0	0	0	0	x	x	x	x	x	x	x	x

Register Transfer Notation

R7 ⇐ (PC)
PC ⇐ Mem[ZEXT(IR[7:0])]

Control Signals

1. GatePC
2. DRMUX selects [111] *(not shown in the simplified datapath)*
3. LD.REG
4. MARMUX selects ZEXT(IR[7:0)
5. Gate.MARMUX
6. LD.MAR
7. MEM.EN/R
    a. Wait for R
8. MDRMUX selects MEM. *(the MDRMUX is not shown in the simplified datapath)*
9. LD.MDR
10. Gate.MDR *(the MDR Gate is not shown in the simplified datapath)*
11. PCMUX selects bus.
12. LD.PC