Homework 5 Keys

Zhang Zheng

2022 年 7 月 19 日

(16') 5.15 State the contents of R1, R2, R3, and R4 after the program starting at location x3100 halts.

Address	Data
0011 0001 0000 0000	1110 001 000100000
0011 0001 0000 0001	0010 010 000100000
0011 0001 0000 0010	1010 011 000100000
0011 0001 0000 0011	0110 100 010 000001
0011 0001 0000 0100	1111 0000 0010 0101
:	:
:	:
0011 0001 0010 0010	0100 0101 0110 0110
0011 0001 0010 0011	0100 0101 0110 0111
:	:
:	:
0100 0101 0110 0111	1010 1011 1100 1101
0100 0101 0110 1000	1111 1110 1101 0011

细节：如果答案正确直接给分，否则如果汇编正确给 2 分，其他中间结果正确酌情给 1~2 分，但不超过 3 分。

Disassembly:

```
x3100 1110 001 000100000  LEA R1, x20    R1 = x3121 (4')
x3101 0010 010 000100000  LD  R2, x20    R2 = Mem[x3122] = x4566 (4')
x3102 1010 011 000100000  LDI R3, x20    R3 = Mem[Mem[x3123]] = xabcd (4')
x3103 0110 100 010 000001 LDR R4, R2, x1 R4 = Mem[R2 + x1] = xabcd (4')
x3104 1111 0000 0010 0101 HALT
...
x3122 0100 0101 0110 0110 x4566
x3123 0100 0101 0110 0111 x4567
...
x4567 1010 1011 1100 1101 xabcd
x4568 1111 1110 1101 0011 xfed3
```

(15’) 5.16 Which LC-3 addressing mode makes the most sense to use under the following conditions? (There may be more than one correct answer to each of these; therefore, justify your answers with some explanation.)

> 细节：根据题目要求，至少要给出一条解释，哪怕只解释了某种寻址可行或不可行的原因。若未给出解释或解释错误，扣 2 分；若寻址模式错误但解释合理，酌情给 1~3 分；否则不给分。

a. You want to load one value from an address that is less than $\pm 2^8$ locations away. (5’)

PC-relative mode (LD), since offset can fit into PCoffset9. LDI will take longer than LD, LDR will require >1 instructions.

b. You want to load one value from an address that is more than 2^8 locations away. (5’)

Indirect mode (LDI), store the address near PC, and use LDI to load. LD cannot load far address (or two LDs are equivalent to LDI), LDR will require >1 instructions.

c. You want to load an array of sequential addresses. (5’)

Base+offset mode (LDR), store the base address into a register, increment the base register (or specify different offsets if array length is constant and small and no loop used). LD and LDI cannot be used to access sequential addresses.

(14’) 5.37 Using the overall data path in Figure 5.18, identify the elements that implement the LDI instruction of Figure 5.8.

> 细节：只需标注（或写出）用到的部分，当然加上箭头也可以，加上 PCMUX 、自增、GatePC、取指令访存以及 FSM 也可以。其他多标注或少标注，1 个扣 1 分，扣完为止。

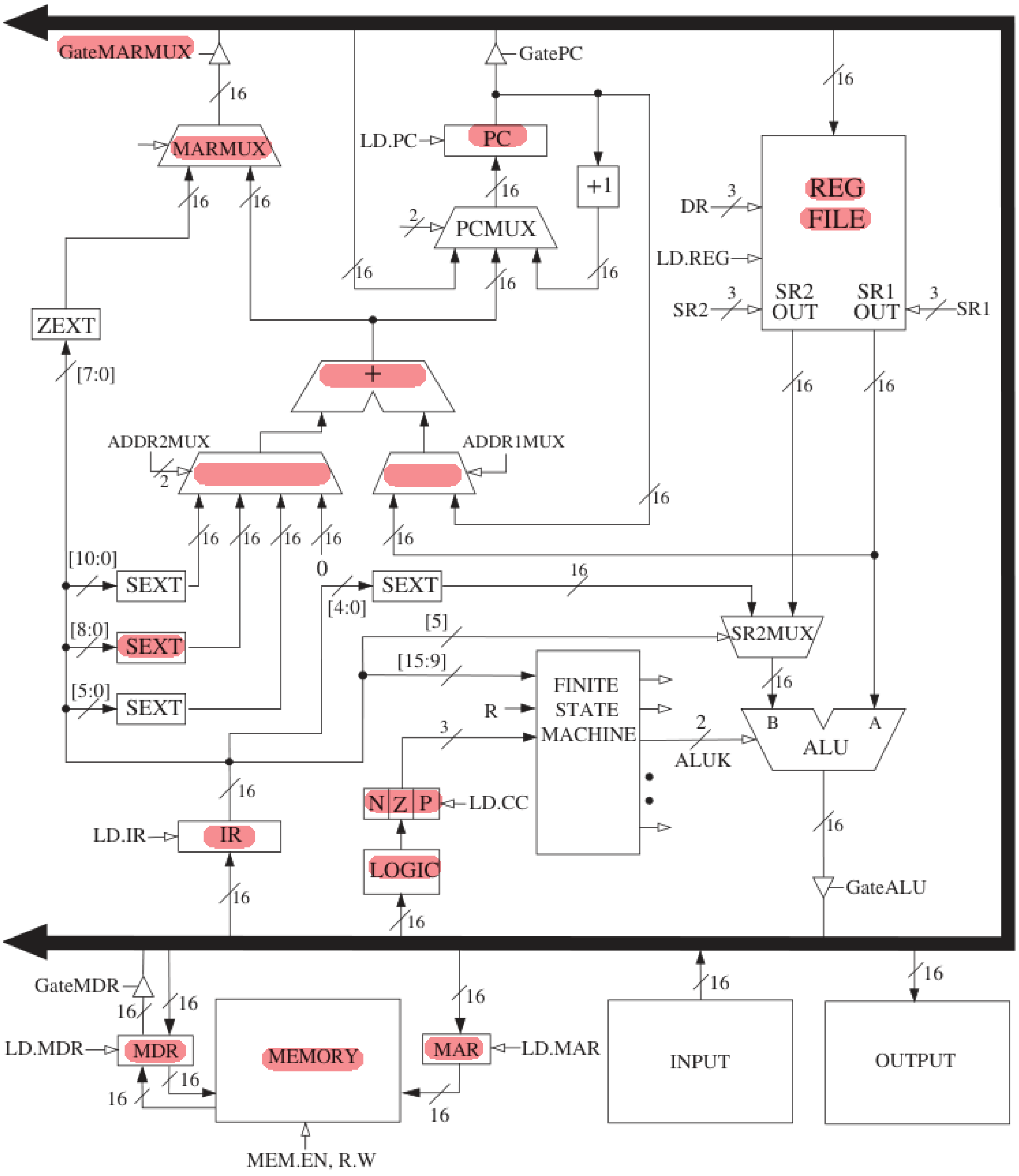


Figure 5.18 The data path of the LC-3.

(10') 5.39 Using the overall data path in Figure 5.18, identify the elements that implement the LEA instruction of Figure 5.6.

细节：只需标注（或写出）用到的部分，当然加上箭头也可以，加上 PCMUX 、自增、GatePC、取指令访存（但必须用箭头与数据访存明显区分）以及 FSM 也可以。根据第三版，LEA 不设置条件码，若设置了条件码扣 2 分（不含 LOGIC）。其他多标注或少标注，1 个扣 1 分，扣完为止。

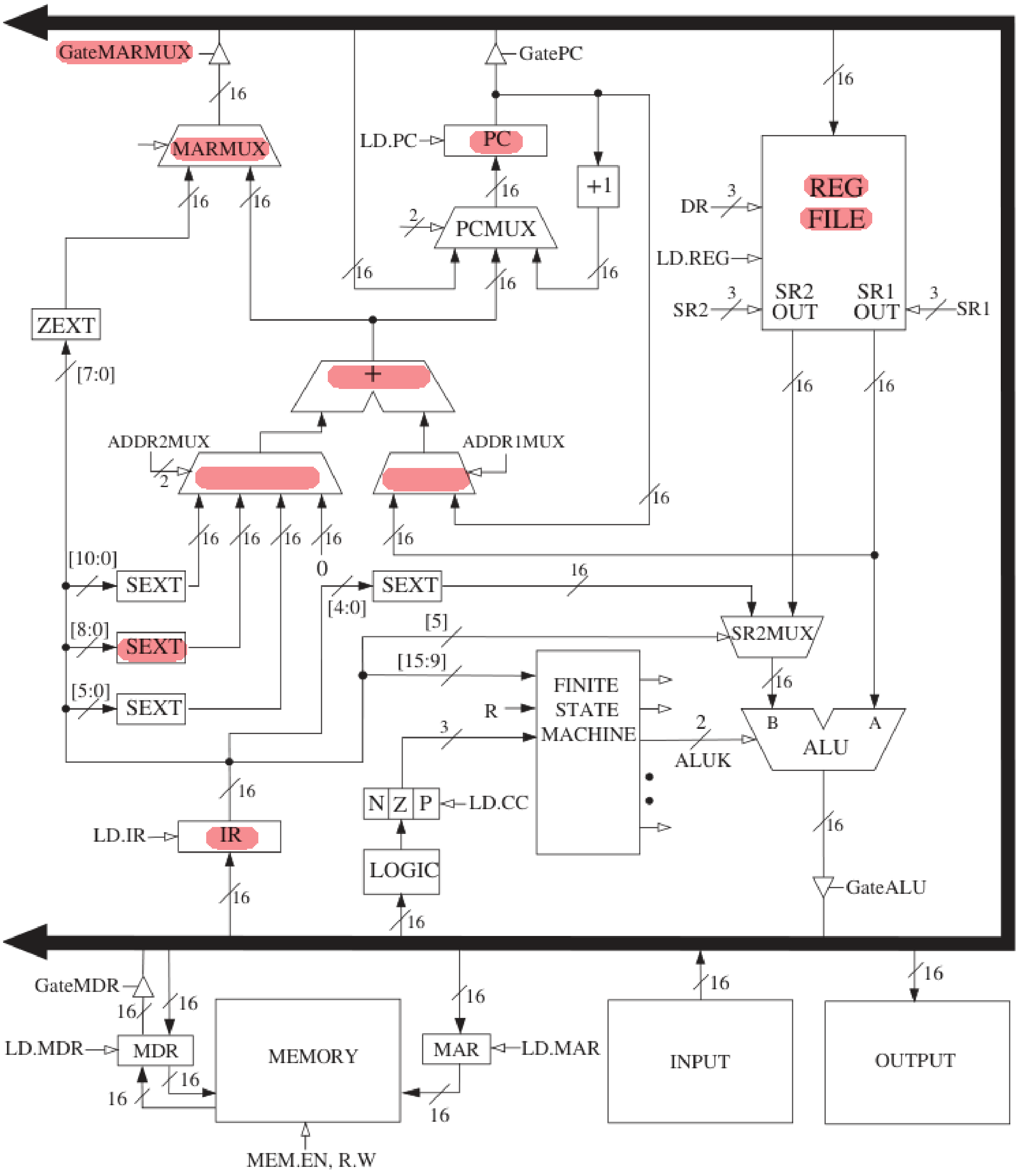
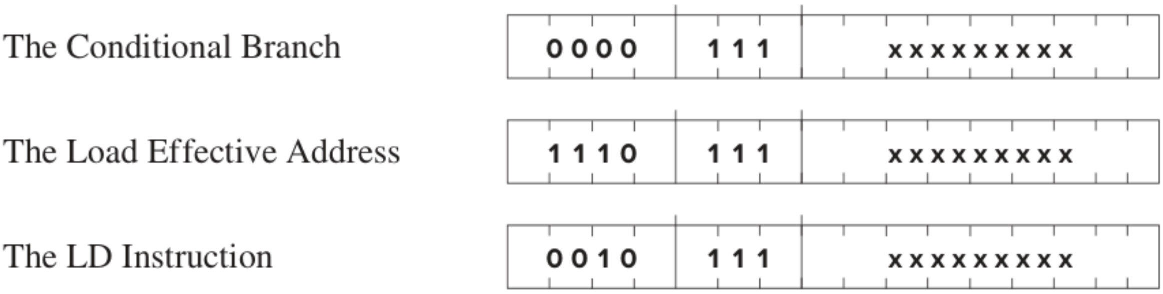


Figure 5.18 The data path of the LC-3.

(15’) 5.50 Three instructions all construct an address by sign-extending the low nine bits of the instruction and adding it to the incremented PC.



The xxxxxxxxx represents the nine-bit offset that is sign-extended.

Where does the LC-3 microarchitecture put the result of adding the nine-bit sign-extended offset to the incremented PC?

细节：若没有结合图中的指令，比如 BR 分情况讨论，LEA 没有指出 R7 等，1 个扣 2 分。若答案错误，但中间步骤正确，扣 2 分。

For BR, the result will be put into PC since it’s an unconditional (BRnzp) one. (5’)

For LEA, the result will be put into R7. (5’)

For LD, the result will be put into MAR (and use it to access memory). (5’)

(20’) 6.9 Using the iteration construct, write an LC-3 machine language routine that displays exactly 100 Zs on the screen.

细节：可以用 LC3Tools 运行一下确认正确性，如果没有 `.ORIG` 也没关系，手动加一下即可。为了计数方便，可以把 100 改小。若输出字符不是 Z，扣 5 分；若输出数量不是 100 个（或死循环），扣 5 分；若输出字符不相同，扣 5 分。若程序完全不对，可不给分。只给出了汇编但没有给出机器码，扣 10 分。一般的汇编错误（包括出现 0、1 外的其他数字，立即数超出范围等）1 个扣 1 分。

Reference answer:

```
0010 000 0 0000 0101          LD R0, Z
0010 001 0 0000 0101          LD R1, C
1111 0000 0010 0001      L    TRAP x21
0001 001 001 1 11111          ADD R1, R1, #-1
0000 001 1 1111 1101          BRp L
1111 0000 0010 0101           TRAP x25
0000 0000 0101 1010      Z    .FILL x5A
0000 0000 0110 0100      C    .FILL #100
```

(10’) 6.10 Using the conditional construct, write an LC-3 machine language routine that determines if a number stored in R2 is odd.

细节：如果实现了输出当然可以，第二个 BR 无条件也可以，两个 BR 顺序可交换。若修改了 R2，扣 3 分；若只判断了偶数或奇数中的一个，扣 5 分。若没有实现输出，且没有充分的注释说明，扣 1 分。只给出了汇编但没有给出机器码，扣 5 分。一般的汇编错误（包括出现 0、1 外的其他数字，立即数超出范围等）1 个扣 1 分。扣完为止。

Reference answer:

```
x3100 0101 000 010 1 00001 ( AND R0, R2, #1 )
x3101 0000 010 0 0000 1110 ( BRz Even )
x3102 0000 101 0 0001 1101 ( BRnp Odd )
```

This program tests whether R2 is even or odd. This program branches to x3110 if the number in R2 is even and branches to x3120 if the number in R2 is odd.