

作业三答案与解析

T1

你的任务是构思新款LC-3。我们将向指令集添加16个操作码，寄存器的个数仍然是8。内存是字节可寻址的，总地址空间为32K字节。指令长度仍然是16位，尽管我们可能需要改变其中一些字段的位宽。

1. PC位宽至少需要多少位才足够表示内存中所有的位置？
2. 在这台新机器上，ADD指令中可以表示的最大立即数是多少？
3. 在这台新机器上，LD指令能够覆盖的地址范围是什么？

答案与解析：

1. 15位。 $32K = 2^{15}$.
2. 7。不同操作码共31个，需要5位表示；寄存器共8个，需要3位表示。因此留给立即数的位宽为4。
3. $[PC - 128, PC + 127]$.留给偏移量的位宽为8位。

T2

请参考第三版课本而不是第二版，可能会有点区别

LC-3指令周期包括六个阶段：取指令，译码，计算地址，取操作数，执行，存储结果。但不是所有指令都需要所有的六个阶段。

1. 简述指令周期中的各个阶段，以及各阶段分别执行什么操作。
 1. 取指令：从内存中获取下一条指令，载入指令寄存器。
 2. 译码：根据指令内容决定哪部分的微结构需要工作
 3. 计算地址：计算指令执行所需的**内存**地址
 4. 取操作数：获取指令执行所需的源操作数
 5. 执行：进行指令的执行
 6. 存储结果：将执行结果写入目的地
2. 填写下表。若某种指令执行时需要某阶段，则在对应位置打勾。

	FETCH	DECODE	EVALUATE ADDRESS	FETCH OPERANDS	EXECUTE	STORE RESULT
ADD	✓	✓		✓	✓	
AND	✓	✓		✓	✓	
ST	✓	✓	✓	✓		✓
STR	✓	✓	✓	✓		✓
LDI	✓	✓	✓	✓		✓
LEA	✓	✓	✓			✓
BR	✓	✓	✓		✓	
JMP	✓	✓	✓		✓	

解析： 首先，所有指令都需要取指和译码。

- ADD:
 1. 不需要evaluate address: 课本P131, the ADD and AND instructions in the LC-3 For those instructions, the EVALUATE ADDRESS phase is not needed.
 2. 需要fetch operands: 课本P132, In the ADD example, this phase consisted of obtaining the source operands from R2 and R6.
 3. 需要execute: 课本P132, In the ADD example, this phase consisted of the step of performing the addition in the ALU.
 4. ★ 不需要store result : 课本P132, the LC-3 ADD instruction does not require a separate EVALUATE ADDRESS phase or a separate STORE RESULT phase.
- AND: 和ADD相同
- ST: 和LD相同
- STR: 与ST相同
- LDI: 与LD相同
 1. 需要evaluate address: 课本P131, This calculation was performed during the EVALUATE ADDRESS phase.
 2. 需要fetch operands: 课本P131, In the LD example, this phase took two steps: loading MAR with the address calculated in the EVALUATE ADDRESS phase and reading memory that resulted in the source operand being placed in MDR.
 3. ★ 不需要execute : 课本P132, The LC-3 LD instruction does not require an EXECUTE phase.
 4. 需要store result
- LEA: (这条指令的执行过程, 课本中讲得不太清楚, 我也不太确定 🤔 以下是我自己的理解)
 1. 需要evaluate address: 将PC和PC offset相加, 计算要装入寄存器的地址
 2. 不需要fetch operands
 3. 不需要execute
 4. 需要store result ? : 把计算出的结果存入寄存器
- BR:

- 需要evaluate address: 课本P163, then the PC is loaded with the address obtained in the EVALUATE ADDRESS phase.
 - 不需要fetch operands: 没有操作数需要读取。条件码是在 execute 阶段检查的。
 - ★需要execute: 课本P162, the conditional branch's execute phase either does nothing or it loads the PC with the address of the instruction it wishes to execute next.
 - 不需要store results: 没有结果需要存储
 - JMP: 类似BR, 除了execute阶段不需要检查条件码, 直接向PC装载目标地址即可。
3. 假设取指令需要3个周期, 译码需要1个周期, 执行需要1个周期。如果需要读写内存, 则获取操作数或存储需要100个周期; 否则, 读写寄存器只需要1个周期。关于求值地址, PC相对模式和基地址+偏移模式需要1个周期, 而间接寻址模式则需要100个周期。你的任务是计算以下LC-3程序运行所需的周期数。

Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x30F6	1	1	1	0	0	0	1	1	1	1	1	1	1	0	1	
x30F7	0	0	0	1	0	1	0	0	0	1	1	0	1	1	1	0
x30F8	0	0	1	1	0	1	0	1	1	1	1	1	1	0	1	1
x30F9	0	1	0	1	0	1	0	0	1	0	1	0	0	0	0	0
x30FA	0	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1
x30FB	0	1	1	1	0	1	0	0	0	1	0	0	1	1	1	0
x30FC	1	0	1	0	0	1	1	1	1	1	1	1	0	1	1	1

答案与解析: 441个周期。

- LEA, 6个周期
 - ADD, 6个周期
 - ST, 106个周期
 - AND, 6个周期
 - ADD, 6个周期
 - STR, 106个周期
 - LDI, 205个周期
4. 考虑ADD, STR和BR三种指令, 完成下表。如果对应的组件在相应的阶段可能被读写, 则填入“R”或“W”。

	FETCH	DECODE	EVALUATE ADDRESS	FETCH OPERANDS	EXECUTE	STORE RESULT
PC	R, W		R BR		W BR	
IR	W	R	R STR BR			
MAR	R, W		W STR	R STR		
MDR	R			W STR		R STR
MEM	R					W STR
registers			R STR	R ADD STR	W ADD	
NZP					R BR, W ADD	

5. 题中描述的是流水线CPU。

T3

假设下面的LC-3程序被加载到了从x3000开始的内存中：

```
x3000: 0101 000 001 1 00000
x3001: 0001 000 000 1 00001
x3002: 0001 010 001 0 00 000
x3003: 0001 010 001 0 00 010
x3004: 1001 010 010 1111111
x3005: 0000 010 000000001
x3006: 1111 0000 00100101
x3007: .....
```

1. 假如x3007处的指令被执行了，则R1的初值是多少？
2. 我们打算在指令集中添加异或操作，并且用闲置的操作码1101来表示。其格式与ADD和AND指令相同。请重写题目中的程序段，至少需要使用一次异或指令。
3. 为了实现异或操作，我们需要对现有架构进行哪些改动？

答案与解析： 机器码翻译成汇编代码如下：

```
; 给R0赋初值为1
AND R0, R0, #0
ADD R0, R0, #1

; 计算R1+R1+1, 存入R2
ADD R2, R1, R0
ADD R2, R1, R2

; 若R2的值为0xffff则跳转
NOT R2, R2
BRz #1
HALT

; 以下开始为0x3007处的代码，略
```

1. 0xffff (-1)或0x7fff (32767)。这段代码的意义是检查R1与R1+1的和是否为0xffff，是才会执行0x3007处的代码。
2. 将 `ADD R2, R1, R2` 改为 `XOR R2, R1, R2` 即可。
3. ALU需要添加异或运算功能；control模块输出的ALU控制信号需要添加异或的情况。

T4

答案:

	R/W	MAR	MDR
Operation 1	W	x4000	1 1 1 1 0
Operation 2	R	x4003	1 0 1 1 0
Operation 3	W	x4001	1 0 1 1 0
Operation 4	R	x4002	0 1 1 0 1
Operation 5	W	x4003	0 1 1 0 1

	Memory before Access 1	Memory after Access 3	Memory after Access 5
x4000	0 1 1 0 1	1 1 1 1 0	1 1 1 1 0
x4001	1 1 0 1 0	1 0 1 1 0	1 0 1 1 0
x4002	0 1 1 0 1	0 1 1 0 1	0 1 1 0 1
x4003	1 0 1 1 0	1 0 1 1 0	0 1 1 0 1
x4004	1 1 1 1 0	1 1 1 1 0	1 1 1 1 0

课本第四章练习题4.17原题。

解析: 由题意可知, 只有R操作会改变MDR的值, W操作不会改变MDR的值。

Operation 1和Operation 3中MDR的值不同, 由此推断Operation 2只能是R操作, 因此改变了MDR的内容, 且Operation 2之后MDR的内容和3中相同, 开头两位都是 10. 观察可得只有 x4003 的内容符合该条件, 因此第二步操作的对象是 x4003, 完成读取操作后MDR是变成了 10110。

观察Memory before Access 1和after Access 3两张表可发现, x4000 和 x4001 中的值都发生了改变。由于前三次操作只包含两次W操作, 可以推断两次写操作的对象正是这两个地址。由Operation 3进行写入时MDR的值 10110 容易推断是写入了 x4001, 因此Operation 1只能是写入了 x4000。

由于前面的写操作均不涉及 x4003, 在Operation 3结束后其值仍然维持 10110, 但是在Access 5之后 x4003 中的值改变了, 可得四、五步操作中一定包含 x4003 的写操作。且观察到写入其中的新值与操作三结束时MDR的值不同, 由此推测出Operation 4是R操作, 以此将MDR的值修改为 01101, 之后的Operation 5则是关于 x4003 的写操作。观察Memory after Access 3表格可发现, 只有 x4002 处的值符合条件, 由此推出Operation 4读操作的对象是 x4002, 并且可以推出 x4002 处的值是 01101。

T5

答案:

X	Does the program halt?	Value stored in R0
000000010	Yes	2
000000001	Yes	3
000000000	Yes	6
111111111	No	--
111111110	Yes	x8004 ($-2^{15} + 4$)

解析： 题目代码的汇编码如下：

```
x3000: AND R0, R0, #0
x3001: ADD R0, R0, #2
x3002: BRzp X
x3003: ADD R0, R0, #3
x3004: ADD R0, R0, #1
x3005: HALT
```

X即为9位的PC offset。

1. $X = 2$ ，在 x3002 后直接跳转至 HALT。
2. $X = 1$ ，在 x3002 后跳至 x3004。
3. $X = 0$ ，相当于不跳转按序执行。
4. $X = -1$ ，则程序一直在 x3002 死循环。
5. $X = -2$ ，则R0会不断+2，直到溢出变成负数才会跳出循环。