

第九章习题参考答案

包含题目：习题 9.1 – 9.5 以及 9.9 – 9.11

题目 9.1 对于 DLX 的 I-类型指令，试回答以下问题。

- (1) 立即数的范围是多少？
- (2) 如果重新定义 DLX 的 ISA，使得立即数表示无符号整数，那么，立即数的范围是多少？
- (3) 如果立即数表示无符号整数，那么执行位于 $x3000\ 0000 \sim x3000\ 0003$ 中的 LW 指令，该指令使用 R6 作为基址寄存器，且 R6 的值为 $x4000\ 0000$ ，那么，能够加载的数据的最大地址是多少？
- (4) 如果重新定义 DLX 的 ISA，将寄存器的数量从 32 个降低到 16 个，那么，在 I-类型的指令中能够表示的立即数的最大值是多少？假设立即数仍表示补码整数。

解答

- (1) $-2^{15} \sim 2^{15} - 1$
- (2) $0 \sim 2^{16} - 1$
- (3) $0x4000\ FFFF$
- (4) $2^{17} - 1$

题目 9.2 对于 DLX 的 R-类型指令，如果重新定义 DLX 的 ISA，将寄存器的数量从 32 个增加至 128 个，是否可行？

解答 在 R-类型指令中，有 5 个位没有使用。将寄存器的数量从 32 个增加到 128 个，所需位数加 2，总共需要位数加 6，而只有 5 个位可用，故不可行。

题目 9.3 假设某计算机的存储器包括 65 536 个单元，每个单元包含 16 位的内容，试回答以下问题。

- (1) 需要多少位表示地址？
- (2) 假设每条指令都由 16 位组成，其中一条指令与 DLX 无条件跳转指令（J 指令）工作机制类似。如果该指令位于单元 10 中，要跳转至地址 20，那么 PC 相对偏移量应为多少？

解答

- (1) 16 位
- (2) $20 - (10 + 1) = 9$

题目 9.4

- (1) 将 R8 乘以 8，并将结果存于 R9 中，使用一条 DLX 指令能够实现吗？

解答

- 可以, 相当于将 R8 左移 3 位后存入 R9 中, 相应的指令为 001101 01000 01001 0000000000000011
- 不可以, 左移 3 位可能导致高位溢出的问题

题目 9.5 试回答下列问题:

- (1) 使用一条 DLX 指令, 可以将 R1 中的值移至 R2 中吗?
- (2) 使用一条 DLX 指令, 可以将 R1 中的值按位取反吗?
- (3) 假设 R1 中存储的位组合的最右边两位有特殊的重要性, 根据这两位数值完成 4 个任务之一。使用一条 DLX 指令, 将这两位孤立出来。

解答

- (1) 可以, 使用如下指令中的任意一条均可满足: ORI R2, R1, #0; ANDI R2, R1, #-1; ADDI R2, R1, #0; ADD R2, R1, R0
- (2) 可以, 指令为 XORI r1, r1, #-1
- (3) ANDI R1, R1, #3

题目 9.9 如下表所示, 当一段起始于单元 x3000 0000 的程序执行结束后, R1~R6 的值分别是多少?

表 9.1 9.9 题存储器情况

地址	数据
x3000 0000	001100 00000 00001 0100 0000 0000 0000
x3000 0004	011100 00001 00010 0000 0000 0000 0100
x3000 0008	010110 00001 00011 0000 0000 0000 0100
x3000 000C	011101 00001 00010 0000 0000 0000 0000
x3000 0010	011100 00010 00100 0000 0000 0000 0100
x3000 0014	010110 00001 00101 0000 0000 0000 0001
x3000 0018	010111 00010 00100 0000 0000 0000 0011
x3000 001C	011100 00001 00110 0000 0000 0000 0000
x3000 0020	011100 00110 00110 0000 0000 0000 0000
x3000 0024	110000 00000 00000 0000 0000 0000 0000
⋮	⋮
x4000 0000	1000 0111 0110 0101 0100 0011 0010 0001
x4000 0004	0100 0011 0010 0001 0000 0000 0000 0000
⋮	⋮
x4321 0000	0000 0000 0000 0000 0100 0000 0000 0000
x4321 0004	0000 0000 0000 0000 1000 0000 0000 0000

解答 将机器代码翻译成汇编语言如下

地址	数据	解释
x3000 0000	001100 00000 00001 0100 0000 0000 0000	LHI R1, 0x4000
x3000 0004	011100 00001 00010 0000 0000 0000 0100	LW R2, 4(R1)
x3000 0008	010110 00001 00011 0000 0000 0000 0100	LBR3, 4(R1)
x3000 000C	011101 00001 00010 0000 0000 0000 0000	SW 0(R1), R2
x3000 0010	011100 00010 00100 0000 0000 0000 0100	LW R4, 4(R2)
x3000 0014	010110 00001 00101 0000 0000 0000 0001	LBR5, 1(R1)
x3000 0018	010111 00010 00100 0000 0000 0000 0011	SB 3(R2), R4
x3000 001C	011100 00001 00110 0000 0000 0000 0000	LW R6, 0(R1)
x3000 0020	011100 00110 00110 0000 0000 0000 0000	LW R6, 0(R6)
x3000 0024	110000 00000 00000 0000 0000 0000 0000	TRAP 0
-----	-----	-----
x4000 0000	1000 0111 0110 0101 0100 0011 0010 0001	0x87654321
x4000 0004	0100 0011 0010 0001 0000 0000 0000 0000	0x43210000
-----	-----	-----
x4321 0000	0000 0000 0000 0000 0100 0000 0000 0000	0x00004000
x4321 0004	0000 0000 0000 0000 1000 0000 0000 0000	0x00008000

R1 = 0x4000 0000, R2 = 0x4321 0000, R3 = 0x0000 0043, R4 = 0x0000 8000, R5 = 0x0000 0021, R6 = 0x0000 4000。

题目 9.10 下表显示了 DLX 存储器的一部分情况，如果条件分支将控制转移到 x3000 0000 单元，那么 R1 和 R2 有什么特点？

表 9.2 9.10 题存储器的一部分

地址	数据
x3000 0000	001011 00001 00001 1111 1111 1111 1111
x3000 0004	000000 00001 00010 00011 00000 000001
x3000 0008	001011 00011 00011 1111 1111 1111 1111
x3000 000C	101000 00011 00000 1111 1111 1111 0000

解答 将机器代码翻译成汇编语言如下

```

1 XORI r1, r1, #-1;
2 ADD r3, r1, r2;
3 XORI r3, r3, #-1;
4 BEQZ r3, 0xFFFF0;

```

由于条件分支将控制转移到 x3000 0000 单元，因此 $r3 = 0$ ，即 $r3 = \sim(\sim r1 + r2) = 0$ ，所以 $\sim r1 + r2 = -1$ ， $-r2 = \sim r1 + 1 = \sim r2 + 1$ ，因此有 $r1 = r2$ 。

题目 9.11 如果在下表所示的 DLX 指令序列执行结束时，R1 中存储的值为 7，由此可推知 R2 的什么信息？

表 9.3 9.11 题存储器情况

地址	数据
x3000 0000	000001 00000 00001 0000 0000 0000 0000
x3000 0004	000001 00000 00011 0000 0000 0001 0000
x3000 0008	000001 00000 00100 0000 0000 0000 0001
x3000 000C	000000 00010 00100 00101 00000 001001
x3000 0010	101000 00101 00000 0000 0000 0000 0100
x3000 0014	000001 00001 00001 0000 0000 0000 0001
x3000 0018	001101 00100 00100 0000 0000 0000 0001
x3000 001C	000011 00011 00011 0000 0000 0000 0001
x3000 0020	101001 00011 00000 1111 1111 1110 1000
x3000 0024	110000 00000 00000 0000 0000 0000 0000

解答 将机器代码翻译成汇编语言如下

地址	数据	解释
x3000 0000	000001 00000 00001 0000 0000 0000 0000	ADDI R1, R0, #0
x3000 0004	000001 00000 00011 0000 0000 0001 0000	ADDI R3, R0, 0x10
x3000 0008	000001 00000 00100 0000 0000 0000 0001	ADDI R4, R0, #1
x3000 000C	000000 00010 00100 00101 00000 001001	AND R5, R2, R4
x3000 0010	101000 00101 00000 0000 0000 0000 0100	BEQZ R5, 0x04
x3000 0014	000001 00001 00001 0000 0000 0000 0001	ADDI R1, R1, #1
x3000 0018	001101 00100 00100 0000 0000 0000 0001	SLLI R4, R4, #1
x3000 001C	000011 00011 00011 0000 0000 0000 0001	SUBI R3, R3, #1
x3000 0020	101001 00011 00000 1111 1111 1110 1000	BNEZ R3, 0xFFE8
x3000 0024	110000 00000 00000 0000 0000 0000 0000	TRAP 0

对应的 C 语言代码为

```

1  R1 = 0;
2  R3 = 16;
3  R4 = 1;
4  do{
5      R5 = R2 & R4;
6      if (R5 != 0){
7          R1 = R1 + 1;
8      }
9      R4 = R4 * 2;
10     R3 = R3 - 1;
11 }while (R3 != 0);

```

此程序检测 R2 的二进制形式中 1 的个数。由 R1 为 7 可以知道 R2 的二进制形式的后 16 位中有 7 个 1 和 $16 - 7 = 9$ 个 0。