

### 复习题参考答案

1. a) 试设计一个 8 位的掩码，当它与一个 8 位的数相与时，前 4 位保持不变而后 4 位被置 0；

b) 试设计一个 8 位的掩码，当它与一个 8 位的数相或时，前 4 位保持不变而后 4 位被置 1；

解答：

a) 11110000

b) 00001111

2. 指令构成及处理

A、一条指令有哪两个组成部分，它们的功能是什么？（4 分）

B、假设一个 32 位指令的格式如下：（6 分）

OPCODE	SR	DR	IMM
--------	----	----	-----

如果存在 70 种操作码和 40 个寄存器，试问立即数部分（IMM）可表达的范围是多少（IMM 编码方式为补码）？

解答：

指令由操作码与操作数两部分组成。操作码指示该指令的功能，操作数指示该指令的操作对象及其在计算机中的存储位置。

操作码 7bits, SR、DR 各 6bits, IMM 为 13bits。13bits 补码表示的范围为： $-2^{12} \sim 2^{12} - 1$ ，即 -4096 ~ 4095。

3. 分析图示状态图和状态表, 包含四个状态，输入变量 X，输出变量 Z。根据已有信息补充完整状态表和状态图中的遗漏部分



X3002	0	0	0	1	0	0	1	0	0	1	1	1	1	0	1
-------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

C. 给出程序结束后，R1 和 R2 的值。

R1=4

R2=20

5 、阅读下面这段 LC-3 汇编语言，写出程序的输出。

```

        .ORIG      X3000
L1      LEA        R1, L1
        AND        R2, R2, x0
        ADD        R2, R2, x3
        LD         R3, B1
L2      LDR        R0, R1, xF
        OUT
        ADD        R3, R3, #-1
        BRz        L3
        ADD        R1, R1, R2
        BR         L2
L3      HALT
B1      .FILL      x4
        .STRINGZ   "ABCDEFGHJKLMNOPQ"
        .END

```

解：输出\_\_ EHKN \_\_

6 下面给出一个程序的部分代码，该程序使用 STACK 来存储数据，STACK 在内存中实现，地址为 X5FF8(MAX)到 X5FFF(BASE)。当前执行的指令标号为 AGAIN，STACK 指针（R6）的初值为 x5FF8。代码中 POP 和 PUSH 为事先实现好的 STACK 操作（R0：传递数据，R6：堆栈指针，R5：返回状态信息，（0-成功, 1-溢出)）。表格中给出了相关的初始化值，请填写代码执行完后 STACK 的内容和 R6 的值。

```

AGAIN  AND R0,R0,#0
        JSR POP
        ADD R1,R0,#0
        AND R0,R0,#0
        JSR POP
        ADD R2,R0,#0
        AND R0,R0,#0
        JSR POP
        ADD R0,R0,R1
        ADD R0,R0,R2
        ADD R5,R5,#0
        BRP DONE
        JSP PUSH

```

```

        BRNZP AGAIN
DONE    JSP PUSH
        HALT

```

解答：

	代码执行前内存值	代码执行后
X5FF8	1	1
X5FF9	2	2
X5FFA	3	6
X5FFB	3	3
X5FFC	1	10
X5FFD	5	5
X5FFE	4	19
X5FFF	2	21
		R6=X5FFF

7 请设计一个汇编语言程序，该程序读入内存地址 x4000 的一个 16 位字，请判断该字的高 8 位与低 8 位是否相同，相同为 1，不同为 0，将判断结果存入 x4001 位置。

1) 请说明程序设计思路。

2) 设计一个完整的汇编语言程序完成该任务，代码加注释。

解答：

1) 思路：将要判断的数低 8 位左移 8 次到高八位→R2，将要判断的数低 8 位清零→R1

两者相减，判断是否相等。（思路正确得 5 分）

2) 代码 （代码正确 10 分）

```

        .ORIG x3000
        AND R5, R5, #0
        AND R3, R3, #0
        ADD R3, R3, #8
        LEA R0, BB
        LDR R1, R0, #1
        LDR R1, R1, #0
        ADD R2, R1, #0
AGAIN    ADD R2, R2, R2    ;将要判断的数低 8 位左移 8 次到高八位→R2
        ADD R3, R3, #-1
        BRp AGAIN
        LDR R4, R0, #0
        AND R1, R1, R4    ;将要判断的数低 8 位清零→R1
        NOT R1, R1
        ADD R1, R1, #1
        ADD R2, R2, R1    ;R2-R1
        BRnp NO
        ADD R5, R5, #1    ;r2-r1=0

```

```
LD R6, RES
STR R5, R6, 0
NO    TRAP x25
BB    .FILL xFF00
      .FILL x4000
RES   .FILL x4001
      .END
```