

华中科技大学

二〇〇二年招收硕士研究生入学考试试题

考试科目: 综合考试一

适用专业: 计算机应用技术

(除画图题外, 所有答案都必须写在答题纸上, 写在试题上及草稿纸上无效, 考完后试题随答题纸交回)

数字电路与逻辑设计试题 (50 分)

一. 多项选择题 (从下列各题的四个备选答案中选出两个或两个以上正确答案, 并将其代号填写在题中括号内. 答案选错或未选全者, 该题不得分. 每小题 1.5 分, 共 12 分)

1. 下列信号中, () 是数字信号。

A. 交流电压; B. 交通灯状态; C. 开关状态; D. 温度变化。

2. 下列电路中, () 是非数字电路。

A. 差动放大电路; B. 集成运放电路; C. 逻辑运算电路; D. RC 振荡电路。

3. 在下列逻辑门中, () 可用来实现任意逻辑函数。

A. 与非门; B. 与门; C. 或非门; D. 异或门。

4. 若 $PB_4B_3B_2B_1$ 是采用偶检验的奇偶检验码, 则逻辑表达式 () 成立。

A. $P = B_4 \oplus B_3 \oplus B_2 \oplus B_1$; B. $P = B_4 \oplus \bar{B}_3 \oplus B_2 \oplus \bar{B}_1$;

C. $P = B_4 \odot B_3 \odot B_2 \odot B_1$; D. $P = B_4 + B_3 + B_2 + B_1$ 。

5. 半导体存储器 () 是可编程只读存储器。

A. MROM; B. PROM; C. EPROM; D. E^2 PROM。

6. 下列器件中, () 是组合电路。

A. 四位并行加法器; B. 四位计数器; C. 四位数据选择器; D. 3-8 译码器。

7. 图 1 所示电路的输出函数 F 可表示成 ()。

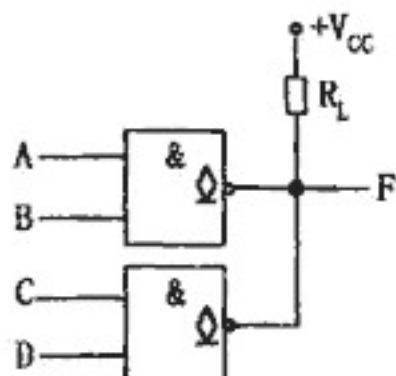


图 1

A. $F = \overline{AB} + \overline{CD}$;

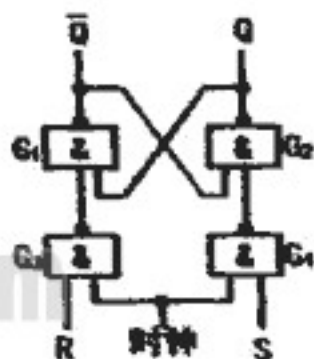
B. $F = (\overline{A} + \overline{B})(\overline{C} + \overline{D})$;

C. $F = \overline{AB + CD}$;

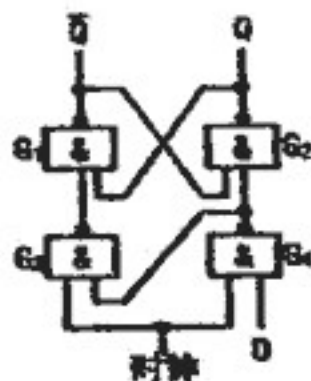
D. $F = \overline{\overline{AB} + \overline{CD}}$.

8. 下列触发器中, () 存在空翻问题。

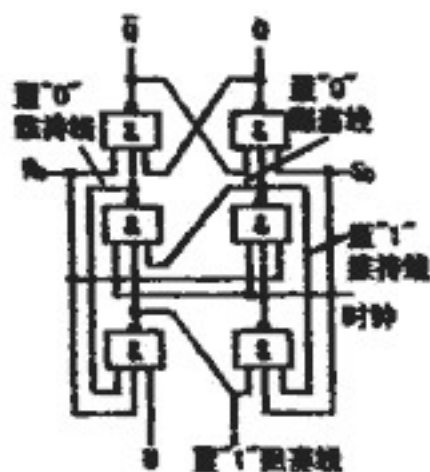
A.



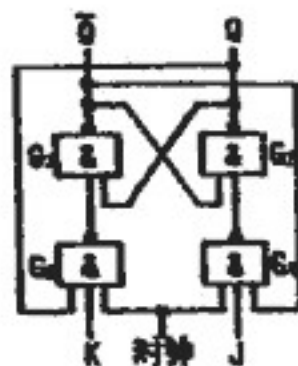
B.



C.



D.



二、填空(每小题4分,共8分)

1. 某工作小组由 A、B、C、D 四人组成，其工作关系如下：

- ① A 工作时，必须 B、C 两人中至少有一人工作；
- ② B 只能和 A 一起工作；
- ③ C 能独立工作；
- ④ D 工作时，A 必须停止工作。

假定 F 为不停止工作条件，由上述关系可求出 F 的最简“与-或”表达式为_____；最简“或-与”表达式为_____。

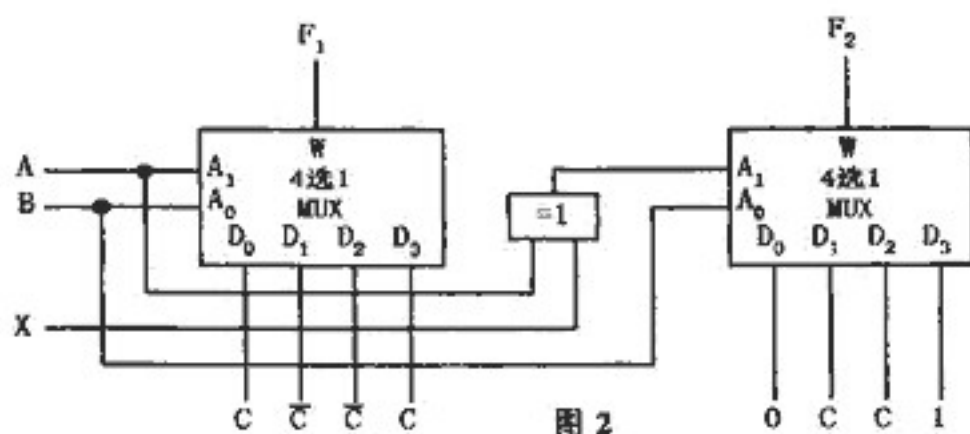
2. 有两个相同型号的与非门 A 和 B, 对它们进行参数测试的结果为: A 的开门电平为 1.4V, 关门电平为 0.8V; B 的开门电平为 1.7V, 关门电平为 1.0V。

根据所测参数可知，输入相同高电平时，_____的抗干扰能力强；输入相同低电平时，_____的抗干扰能力强。

三、分析与设计 (每小题 10 分, 共 30 分)

1. 图 2 所示电路中, 变量 A、B、C 分别表示一位二进制数据, x 为控制变量。

- ② 选择合适的逻辑门, 设计出实现该电路功能的最简电路。



2. 已知某电平异步时序逻辑电路的结构框图如图 3 所示, 表 1 是描述该电路功能的不完整流程表。测试表明, 该电路的反馈回路之间既存在非临界竞争, 又存在临界竞争。

① 请问表 1 中激励状态 A 和 B 各为何值?

② 为了使电路既能实现预定功能, 又能消除反馈回路间的竞争, 激励状态 A、B 应取何值?

③ 求出消除竞争后的激励函数和输出函数最简表达式。

表 1

二次状态 $y_2 y_1$	激励状态 $Y_2 Y_1$ / 输出 Z			
	$x_2 x_1 = 00$	$x_2 x_1 = 01$	$x_2 x_1 = 11$	$x_2 x_1 = 10$
0 0	01 / 0	11 / 0	A / 0	ⓧ / 0
0 1	ⓧ / 0	11 / 0	11 / 0	00 / 0
1 1	B / 0	ⓧ / 0	ⓧ / 1	10 / 1
1 0	ⓧ / 0	ⓧ / 0	11 / 1	ⓧ / 1

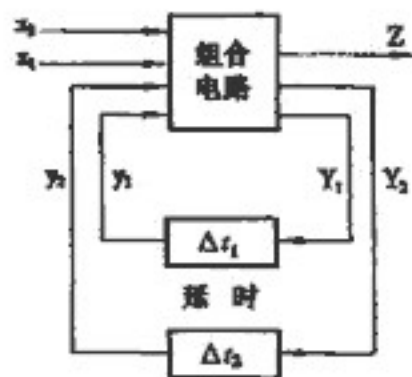


图 3

3. 已知某同步时序电路如图 4 所示。图中, x 为随机输入信号, Z 为输出信号。

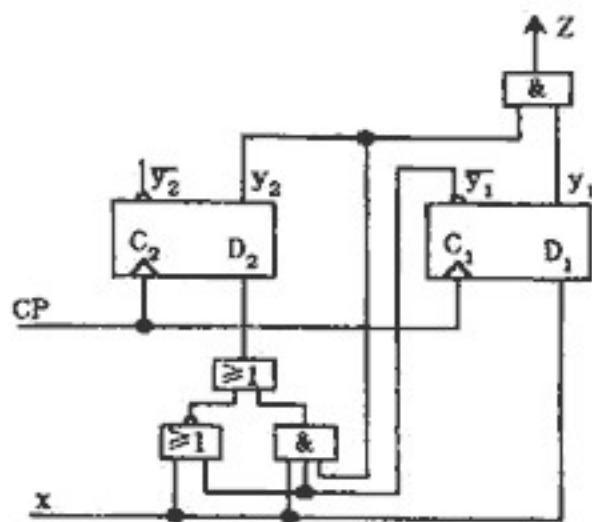


图 4

① 分析该电路, 写出输出函数和激励函数表达式, 并作出状态表和状态图;

② 设电路初始状态为“00”, 说明该电路功能;

③ 改用 JK 触发器作为存储元件, 求出实现该电路功能的激励函数和输出函数最简表达式;

④ 若将电路中的输出函数修改为 $Z = xy_2y_1$, 请问电路功能有何改变?

数据结构 (50 分)

一 选择填空 (每空 1 分, 共 10 分)

- 1 假设 8 枚金币中有一枚是伪造的, 真假金币的区别仅是重量不同, 现以天平作工具, 要求用最少的比较次数找出这枚伪造的金币, 则最少需 _____ 次比较。
(A) 2 (B) 3 (C) 7 (D) 4
- 2 若线性表最常用的操作是存取第 i 个元素及其前趋的值, 则采用 _____ 存储方式节省时间。
(A) 单链表 (B) 双向链表 (C) 单循环链表 (D) 顺序表
- 3 已知数据表 A 中每个元素距其最终位置不远, 则采用 _____ 排序算法最省时间。
(A) 堆排序 (B) 插入排序 (C) 直接选择排序 (D) 快速排序
- 4 n 个顶点的无向图的 邻接表中结点总数最多有 _____ 个。
(A) $2n$ (B) n (C) $n/2$ (D) $n(n-1)$
- 5 设连通图 G 的顶点数为 n , 则 G 的生成树的边数为 _____。
(A) $n-1$ (B) n (C) $2n$ (D) $2n-1$
- 6 在建立邻接表或逆邻接表时, 若输入的顶点信息即为顶点的编号, 则建立图的算法的时间复杂度为 _____。
(A) $O(n+e)$ (B) $O(n^2)$ (C) $O(n \times e)$ (D) $O(n^3)$
- 7 快速排序算法在最好情况下的时间复杂度为 _____。
(A) $O(n)$ (B) $O(n^2)$ (C) $O(n \log_2 n)$ (D) $O(\log_2 n)$
- 8 判断一个有向图是否存在回路, 除了可以利用拓扑排序方法外, 还可以利用 _____。
(A) 求关键路径的方法 (B) 求最短路径的方法
(C) 深度优先扫描算法 (D) 宽度优先扫描算法
- 9 线索二叉树中指向某结点的指针为 D , 该结点没有左孩子的条件是 _____。
(A) $D \rightarrow Lchild = \text{NULL}$ (B) $D \rightarrow Rchild = \text{NULL}$
(C) $D \rightarrow Ltag = 0$ (D) $D \rightarrow Ltag = 1$
- 10 下列排序算法中, 某一趟 (轮) 结束后未必能选出一个元素放在其最终位置上的是 _____。
(A) 堆排序 (B) 冒泡排序 (C) 直接插入排序 (D) 快速排序


```

(2) void a2 ( stack &s, int e ) {
    stack t;
    int d;
    initstack (t);
    while (! stackempty (s))
    {
        pop (s, d);
        if ( d != e ) push ( t,d );
    }
    while (! stackempty (t))
    {
        pop (t, d);
        push ( s,d );
    }
}

```

三、算法设计（每题 10 分，共 20 分）

- 1 若输入 12000 个不同的整数, 其值介于 0 和 19999 之间, 采用哈希方式存储这些数, 哈希函数为 $h(k)=k/2$, 用链地址法克服冲突, 请你设计一个实现算法。提示: 利用两个数组来进行。一维数组 $r[6000]$ 存放发生冲突时的关键字, 结构数组 HT 有二个成员, 一个存放哈希函数的关键字, 另一个指示发生冲突时存于 r 中的关键字的地址。
- 2 试设计一个非递归算法把二叉树的叶结点按从左到右的顺序连成一个单链表, 二叉树用二叉链表存储, 链接时叶子结点的 $rchild$ 域用做单链表的指针。

算法所涉及的结构如下:

```

typedef struct bnode{
    elementype info;
    struct bnode *lchild, * rchild;
} bnode, bitree; // 结点类型和指向二叉树的指针的类型

```

```

typedef struct seqstack {
    bitree data [maxsize]; // 栈的最大容量为 maxsize
    int top;
} seqstack; // 定义顺序栈类型

```