

组合逻辑作业

16061160 陈麒先

***郑重声明：关于诚实守信公约：以下题目标号前注明*号表示本题参考了互联网资料，题目标号为红色表示本题是与同学研究后的结论，其余未注明题目均为翻阅课件、课堂笔记和教材后独立思考的结果。特此声明。**

一、填空题：

1. PN 结的单向导电性即反向偏置时 截止，正向偏置时 导通。
2. TTL 与非门的两个状态通常称为关态和开态，当输入全为高电平时对应的是 开 态，此时输出为 低电平；当输入有一为低电平时，对应的是 关 态，此时输出为 高电平。
3. TTL 与非门的额定输出逻辑低电平 $V_{OL} = \underline{0.3}$ 伏，额定输出逻辑高电平 $V_{OH} = \underline{3.6}$ 伏。（设电源电压 $V_{CC} = +5V$ ）
4. *对于 ECL、TTL、CMOS 集成电路，按静态功耗低和高的顺序依次为 CMOS < TTL < ECL，按工作速度快慢的顺序依次为 ECL > TTL > CMOS，按抗干扰能力强弱的顺序依次为 CMOS > TTL > ECL。
5. 逻辑变量和函数只有 0, 1 两种取值，而且它们只是表示两种不同的逻辑状态。
6. 逻辑函数 $F = \overline{AC} + \overline{BD}$ ，其反函数 $\overline{F} = \underline{(A + \overline{C}) \cdot (B + \overline{D})}$ ，其对偶式 $F^* = \underline{(\overline{A} + C) \cdot (\overline{B} + D)}$ 。（*注：以下用“~”表示“非”）
7. 函数 $F = AB + AC + \overline{CD} + ACDE$ 的最简与或式是 $AC + AB + \overline{CD}$ 。
8. 从结构看，组合逻辑电路由门电路构成，不含 环路，也不含 存储电路，信号从输入开始单向传输到输出。对于组合逻辑电路，任何时刻电路的输出仅由当时的 输入 决定。
9. 将加在电路若干输入端中的某一个输入端的信号变换成相应的一组二进制代码输出的过程叫做 编码。
10. 将二进制代码所表示的信息翻译成对应输出的高低电平信号的过程称为 译码；n 位二进制译码器有 n 个输入，有 2^n 个输出，

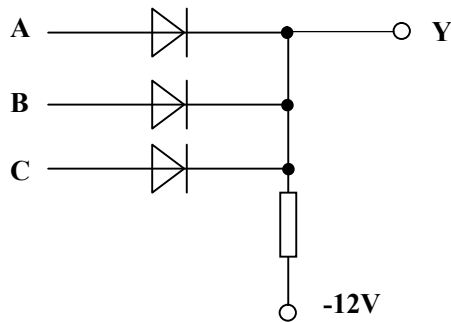
工作时译码器只允许有一个输出有效。

11. 输出低电平有效的二-十进制译码器的输入 8421BCD 码 $A_3 \sim A_0$ 为 0111 时, 其输出 $\bar{Y}_0 \sim \bar{Y}_9 =$ 11_1111_1011。

二、选择题:

1. 二极管门电路如下图所示, 假定输入变量的值可在 0V 和 3V 两种电平下变化, 输出和输入之间的正逻辑关系为 (②)。

① 与逻辑 ② 或逻辑 ③ 与非 ④ 或非



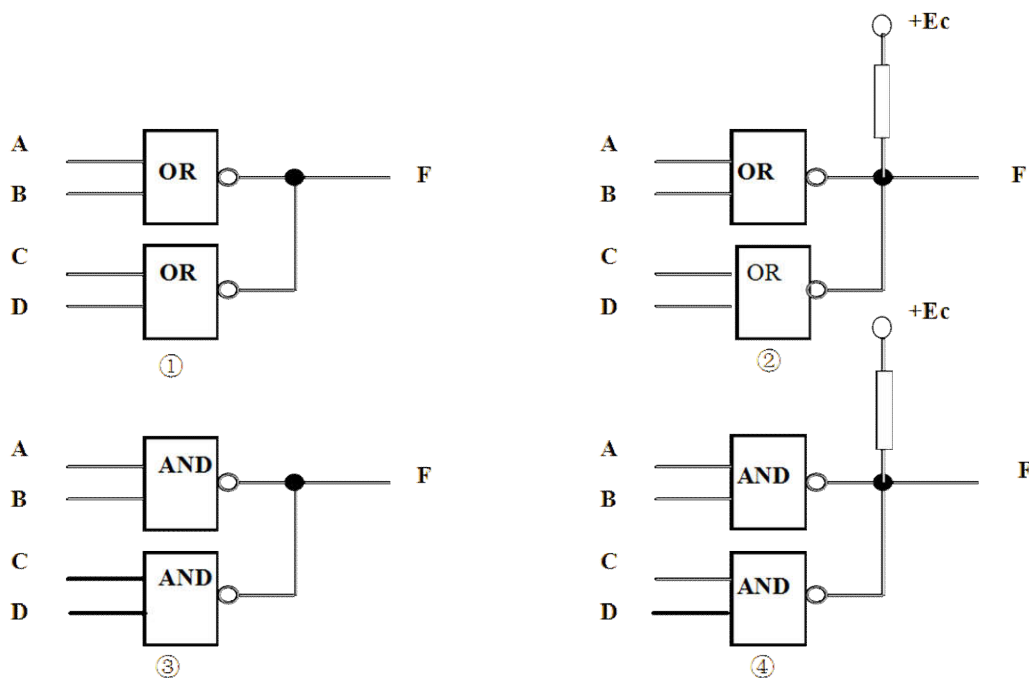
2. 逻辑函数 $F = \overline{A + BC(A + B)}$, 当 ABC 的取值为 (②) 时, $F = 1$ 。

① 000 ② 011 ③ 101 ④ 111

3. 硅二极管导通和截止的条件是 (③)。

① $U_D \geq 0.7V, U_D < 0.5V$ ② $U_D \geq 0.5V, U_D < 0.7V$
③ $U_D \geq 0.7V, U_D < 0.7V$ ④ $U_D \geq 0.5V, U_D < 0.5V$

4. 为实现 $F = \overline{AB} \cdot \overline{CD}$, 下列电路接法正确的是 (④)。



*5. 组合逻辑电路的竞争-冒险是由于 (③) 引起的。

- ① 电路不是最简
- ② 电路有多个输出
- ③ 构成电路的逻辑元件存在传输延迟
- ④ 电路使用不同的门电路

*6. 用电位关系描述晶体三极管工作时的三种状态, 正确的可靠工作条件是 (④)。

- ① 截止区: $V_{BE} < 0V$, 放大区: $V_{BC} > 0V$, 饱和区: $V_{BC} > 0V$ 且 $V_{BC} > 0V$
 ② 截止区: $V_{BE} < V_T$, 放大区: $V_{BE} > V_T$ 且 $V_{BC} < 0V$; 饱和区: $V_{BC} <$

0V

- ④ 截止区: $V_{BE} \leq 0V$; $V_{BC} < 0V$; 放大区: $V_{BE} > 0$, 且 $V_{BC} < 0V$; 饱和区: $V_{BE} \geq 0.7V$, 且 $V_{BC} > 0V$

7. 如需要判断两个二进制数的大小或相等, 可以使用 (④) 电路。

- ① 译码器 ② 编码器 ③ 数据选择器 ④ 数值比较器

8. 函数 $F = AB + \overline{AC} + \overline{BC} + \overline{CD} + \overline{D}$ 的最简与或式为 (①)。

- ① 1 ② 0 ③ AB ④ 以上均不是

9. 只考虑本位数而不考虑低位来的进位的加法称为 (②)。

- ① 全加 ② 半加 ③ 全减 ④ 半减

10. 能实现从多个输入端中选出一路作为输出的电路称为 (③)。

- ① 触发器 ② 计数器 ③ 数据选择器 ④ 译码器

11. 函数 $A \oplus B$ 与 $\overline{A \oplus B}$ (③)。

- ① 互为反函数 ② 互为对偶式 ③ 相等 ④ 答案都不正确。

三、问答与计算题：

说明：分析与计算题要求写出分析推导过程，给出必要的公式。

1. 将逻辑函数 $F = ABC + \overline{A}BD + ABCD$ 写成标准与或表达式。

答：

$$F = ABCD + ABC\overline{D} + \overline{A}BCD + \overline{A}BC\overline{D}$$

2. 推导出函数 $F = \overline{AB + BC} + \overline{AC} + \overline{AB}$ 的最简与或式。

解：

$$\begin{aligned} \text{原式} &= (\overline{A + B})(\overline{B + C}) + \overline{AC} + \overline{AB} \\ &= \overline{A}\overline{B} + \overline{A}\overline{B} + \overline{B} + \overline{A}\overline{C} + \overline{AC} + \overline{BC} \\ &= \overline{B} + \overline{C} + \overline{BC} \\ &= \overline{B} + \overline{C} \end{aligned}$$

3. 列出下述问题的真值表，利用最小项推导法写出其逻辑函数表达式，利用公式简化法进行简化并给出逻辑电路图。最后，写出完整的 Verilog HDL 程序。

设计一个投票表决器，三个投票人分别为 A、B、C，同意为“1”，不同意为“0”。按规定只要二人以上同意才能通过，输出为“1”表示通过，为“0”表示不通过。

解：

(1) 真值表

	A	B	C	Result
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1

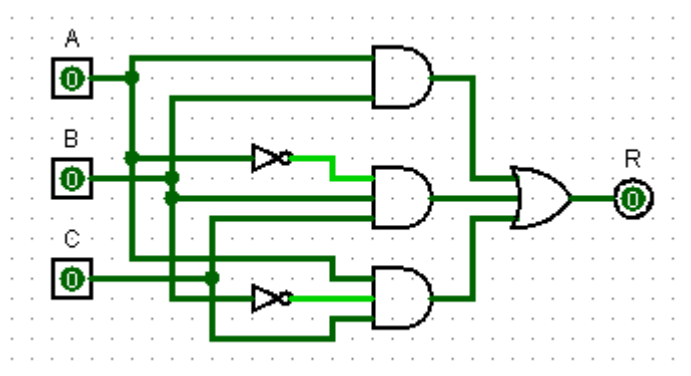
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

(2) 表达式

$$R = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

$$= \overline{A}BC + A\overline{B}C + AB$$

(3) 电路图



(4) Verilog HDL 程序

```

21 module voter(
22     input A,
23     input B,
24     input C,
25     output Result
26 );
27     assign Result = (~A && B && C) || (A && ~B && C) || (A && B);
28
29 endmodule
30

```

4. 用公式法证明下列等式:

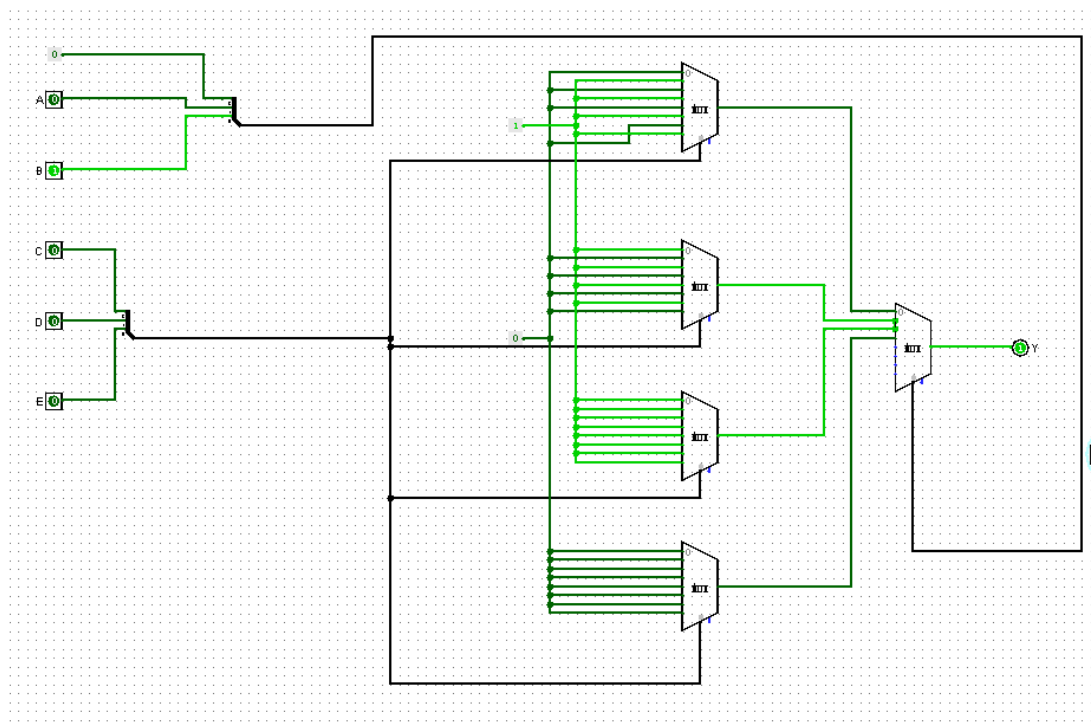
$$\overline{B}C\overline{D} + B\overline{C}D + ACD + \overline{A}B\overline{C}D + \overline{A}B\overline{C}D + \overline{B}C\overline{D} + B\overline{C}D = B + C$$

证明:

$$\begin{aligned}
 \text{左端} &= \overline{B}C\overline{D} + B\overline{C}D + (A + \overline{A}B)\overline{C}D + \overline{A}B\overline{C}D + B(\overline{C}D + CD) \\
 &= \overline{B}C\overline{D} + B\overline{C}D + CD + \overline{A}B\overline{C}D + B \\
 &= B(\overline{C}D + \overline{A}C\overline{D} + 1) + C(\overline{B}D + D) \\
 &= B + C
 \end{aligned}$$

5. 用 8 选 1 数据选择器 CT74151 实现下列函数(给出连接方式以及地址信号、数据输入端的对应关系图, 如果用都其它器件请标出其它器件类型):

$$F(A,B,C,D,E)=\sum m(1, 3, 5, 7, 8, 10, 12, 14, 16, 17, 18, 19, 20, 21, 22, 23)$$



分析：

首先，变量数量为 5，因此共需要有 $2^5=32$ 个输入端口，因此在选择部分，应该有 4 个 CT74151 多路选择器。

其次，由于一个 CT74151 多路选择器只能选择 3 位地址，因此，应将 5 位输入分成高 2 位和低 3 位处理，在最终得到结果的时候，也应该使用一个 CT74151 多路选择器。

最后，根据题意，对多路选择器的 32 个输入端口初始化，赋值规则如题目要求即可。

6. 在四进制数系统中，存在四个数字：0, 1, 2, 3。表 1 定义了一个四进制数的半加器（简要叙述思路，并给出主要的 Verilog HDL 程序段）。

（1）设计一个实现此半加器的电路。要求用二进制编码表示四进制数，例如每个四进制数用 2 位表示。令 $A = a_1a_0$ ， $B = b_1b_0$ ， $Sum = s_1s_0$ ，进位信号 Carry 是二进制信号。编码方案为：00 = (0)₄，01 = (1)₄，10 = (2)₄，11 = (3)₄。要求电路的成本最低。

（2）使用上述描述的方法，设计一个四进制全加器电路。

表 1 四进制半加器

A	B	Carry	Sum
---	---	-------	-----

0	0	0	0
0	1	0	1
0	2	0	2
0	3	0	3
1	0	0	1
1	1	0	2
1	2	0	3
1	3	1	0
2	0	0	2
2	1	0	3
2	2	1	0
2	3	1	1
3	0	0	3
3	1	1	0
3	2	1	1
3	3	1	2

解：

(1) 分析：首先应根据待实现的功能，列出真值表，编码依照编码规则

	a1	a0	b1	b0	carry	s1	s0
1	0	0	0	0	0	0	0
2	0	0	0	1	0	0	1
3	0	0	1	0	0	1	0
4	0	0	1	1	0	1	1
5	0	1	0	0	0	0	1
6	0	1	0	1	0	1	0
7	0	1	1	0	0	1	1
8	0	1	1	1	1	0	0
9	1	0	0	0	0	1	0
10	1	0	0	1	0	1	1
11	1	0	1	0	1	0	0
12	1	0	1	1	1	0	1
13	1	1	0	0	0	1	1
14	1	1	0	1	1	0	0
15	1	1	1	0	1	0	1

16	1	1	1	1	1	1	0
----	---	---	---	---	---	---	---

随后，依照真值表，写出 **carry, s1, s0** 的逻辑表达式。

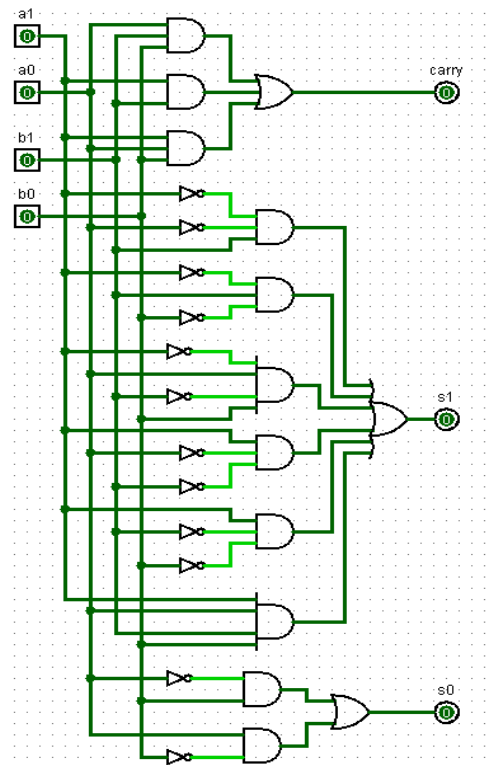
$$\text{carry} = a_0 b_1 b_0 + a_1 b_1 + a_1 a_0 b_0$$

$$s_1 = \sim a_1 \sim a_0 b_1 + \sim a_1 b_1 \sim b_0 + \sim a_1 a_0 \sim b_1 b_0 + a_1 \sim a_0 \sim b_1 + a_1 \sim b_1 \sim b_0 + a_1 a_0 b_1 b_0$$

$$s_2 = \sim a_0 b_0 + a_0 \sim b_0$$

(*注：以上用 “~” 代替 “逻辑非”)

最后，画出电路图有：



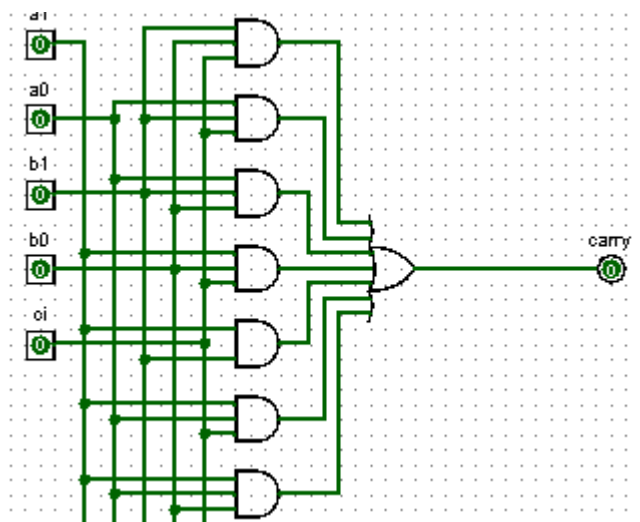
同时，给出相应的 Verilog HDL 语言有：

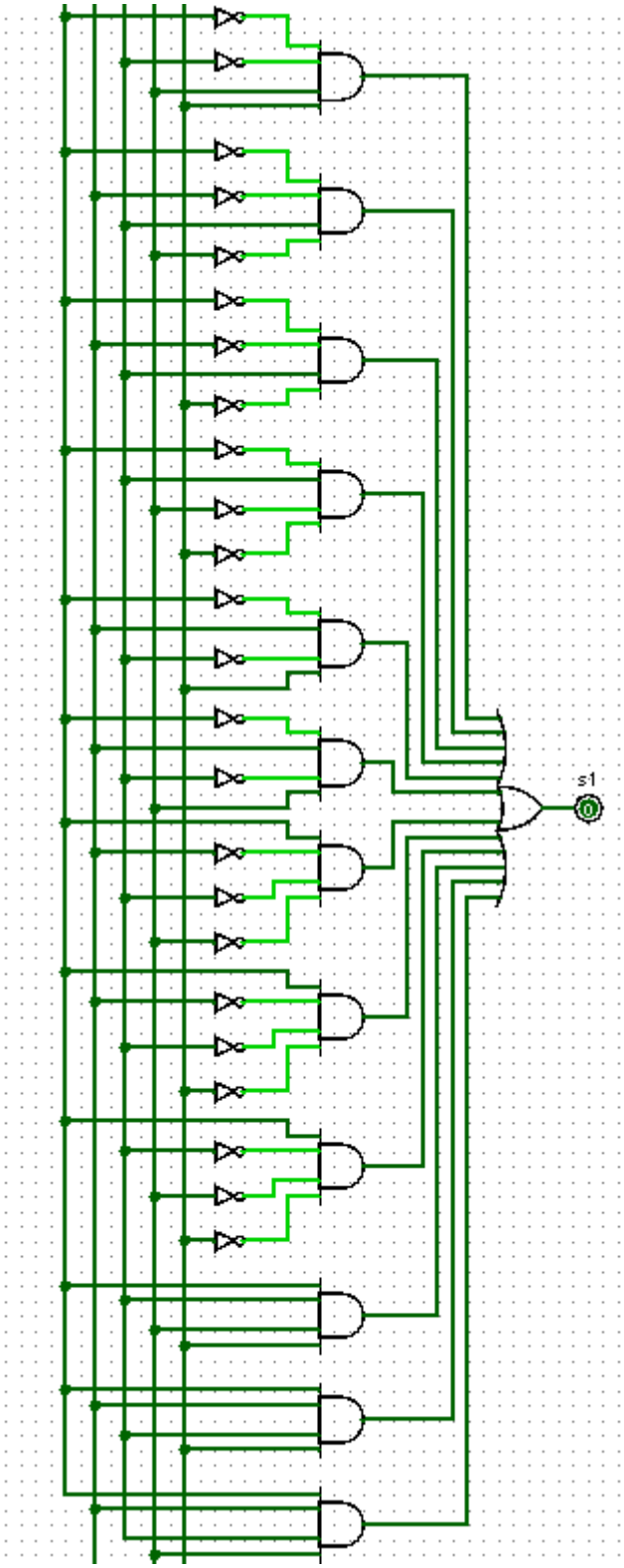

```

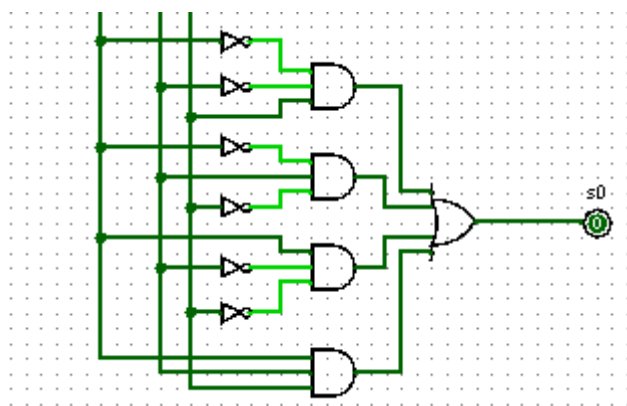
21 module bit4_hadder(
22     input a1,
23     input a0,
24     input b1,
25     input b0,
26     output carry,
27     output s1,
28     output s0
29 );
30
31     wire w [0:30];
32
33     and (w[0],a0,b1,b0);
34     and (w[1],a1,b1);
35     and (w[2],a1,a0,b0);
36     or  (carry,w[0],w[1],w[2]);
37     and (w[3],!a1,!a0,b1);
38     and (w[4],!a1,b1,!b0);
39     and (w[5],!a1,a0,!b1,b0);
40     and (w[6],a1,!a0,!b1);
41     and (w[7],a1,!b1,!b0);
42     and (w[8],a1,a0,b1,b0);
43     or  (s1,w[3],w[4],w[5],w[6],w[7],w[8]);
44     and (w[9],!a0,b0);
45     and (w[10],a0,!b0);
46     or  (s2,w[9],w[10]);
47
48 endmodule

```

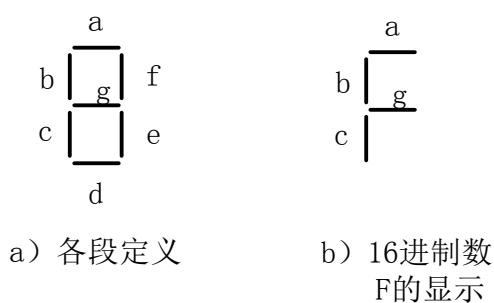
(2) 由于 4 进制全加器和半加器的设计原理相似，只是在半加器的基础上，增加了低位的进位 C_i (进位是 1 位二进制数，取值为 0/1)。因此不重复全加器的手工设计过程，只呈现最终电路图如下。







7. 7 段数码管是由 7 个独立的发光管构成的，每个发光管有一个驱动控制信号。当驱动控制信号为高电平(逻辑 1)时，则信号对应的发光管发光。现需设计 7 段数码管的控制电路，使之能够根据 4 位输入 $x[3:0]$ 显示 0~9, A~F 共 16 个图案。7 段数码管控制电路输出信号为各数码管的驱动控制信号，即 a, b, c, d, e, f, g。数码管各段的定义和 16 进制数“F”（对应 abcdefg 的二进制输出为 1110001）的显示如下图所示。



- (1) 请给出 7 段数码管控制电路的输入输出信号真值表。
- (2) 根据真值表写出各输出信号的逻辑表达式，并化简。
- (3) 采用结构描述法，用 Verilog 语言实现上述的数码管控制器。

解：

- (1) 列出输入输出信号真值表如下：

x3	x2	x1	x0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	0	0	0	1	1	0
0	0	1	0	1	0	1	1	0	1	1
0	0	1	1	1	0	0	1	1	1	1
0	1	0	0	0	1	0	0	1	1	1
0	1	0	1	1	1	0	1	1	0	1
0	1	1	0	1	1	1	1	1	0	1
0	1	1	1	1	0	0	0	1	1	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1
1	0	1	0	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	0	0	0
1	1	0	1	1	1	1	1	1	1	0
1	1	1	0	1	1	1	1	0	0	1
1	1	1	1	1	1	1	0	0	0	1

(2) 根据真值表，经最小项表达式化简（过程已经略去），可以得到如下逻辑表达式：

$$a = \overline{x_2x_0} + x_1 + x_2x_0 + x_3$$

$$b = \overline{x_1x_0} + \overline{x_2x_1} + x_2\overline{x_0} + x_3$$

$$c = \overline{x_2x_0} + \overline{x_1x_0} + x_2x_1 + x_3x_2$$

$$d = \overline{x_3x_2x_0} + \overline{x_2x_1x_0} + \overline{x_2x_1x_0} + \overline{x_2x_1x_0} + x_3\overline{x_1}$$

$$e = \overline{x_3x_1} + \overline{x_3x_0} + \overline{x_1x_0} + \overline{x_3x_2} + x_3\overline{x_2}$$

$$f = \overline{x_3x_1x_0} + \overline{x_2} + \overline{x_3x_1x_0} + x_3\overline{x_1x_0}$$

$$g = \overline{x_2x_1} + \overline{x_1x_0} + \overline{x_3x_2x_1} + x_3\overline{x_2} + x_3x_1$$

(3) 根据逻辑表达式，写出结构级描述的 Verilog HDL 表达式，以下仅给出实现该制定功能的核心代码

```

21 module digital_pipe(
22     input x3,
23     input x2,
24     input x1,
25     input x0,
26     output a,
27     output b,
28     output c,
29     output d,
30     output e,
31     output f,
32     output g
33 );
34 wire gate [30:0];
35 and (gate[0], !x2, !x0);
36 and (gate[1], x2, x0);
37 or (a, gate[0], gate[1], x1, x3);
38 and (gate[2], !x1, !x0);
39 and (gate[3], !x2, !x1);
40 and (gate[4], x2, !x0);
41 or (b, gate[2], gate[3], gate[4], x3);
42 and (gate[5], x1, !x0);
43 and (gate[6], x2, x1);
44 and (gate[7], x3, x2);
45 or (c, gate[0], gate[5], gate[6], gate[7]);
46 and (gate[8], !x3, !x2, !x0);
47 and (gate[9], !x2, x1, x0);
48 and (gate[10], x2, !x1, x0);
49 and (gate[11], x2, x1, !x0);
50 and (gate[12], x3, !x1);
51 or (d, gate[8], gate[9], gate[10], gate[11], gate[12]);
52 and (gate[13], !x3, !x1);
53 and (gate[14], !x3, x0);
54 and (gate[15], !x1, x0);

54 and (gate[15], !x1, x0);
55 and (gate[16], !x3, x2);
56 and (gate[17], x3, !x2);
57 or (e, gate[13], gate[14], gate[15], gate[16], gate[17]);
58 and (gate[18], !x3, !x1, !x0);
59 and (gate[19], !x3, x1, x0);
60 and (gate[20], x3, !x1, x0);
61 or (f, gate[18], !x2, gate[19], gate[20]);
62 and (gate[21], !x2, x1);
63 and (gate[22], x1, !x0);
64 and (gate[23], !x3, x2, !x1);
65 and (gate[24], x3, !x2);
66 and (gate[25], x3, x1);
67 or (g, gate[21], gate[22], gate[23], gate[24], gate[25]);
68
69 endmodule
70

```