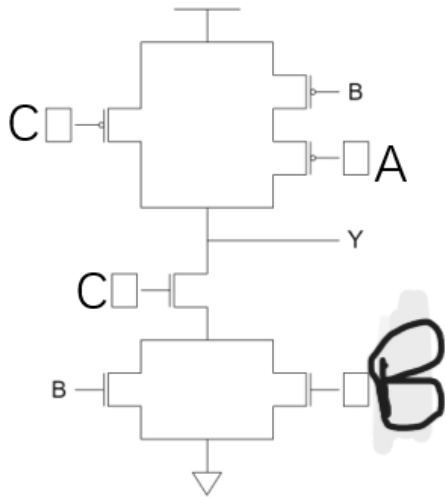


Homework 2

T1

1. 由真值表第二行，可知当 A、B 均为 低电平、C 带高电平时，Y 导通，而 Y 的上方是三个非门，故 A、B 必然在一起，C 单独控制 Y 到正极的一条路，所以在上方左边填入 C、右方填入 A，而真值表第七行验证了我们的想法；
2. 同时，为了不“放生几百度电”（笑），我们需要保证上方接通时下方不会接地。先考虑给出的真值表第七行，Y 下方的第一个位置应该填入 C；再考虑真值表第二行，最后一个位置应该填入 A 或 B，这里为了与左边对称填入 B。

从而本题的解如下：



A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

T2

NAND 门是逻辑完备的，因为我们可以用它来构造所有其他的逻辑门。

以下是使用 NAND 门来构造 NOT、AND、OR 和其他逻辑门的证明：

1. NOT 门：

如果我们将一个 NAND 门的两个输入都连接到同一个输入信号，当输入为 1 时，输出为 0；当输入为 0 时，输出为 1，这就构成了 NOT 门。即， $\text{NOT } A = A \text{ NAND } A$ 。

2. AND 门：

首先，将输入 A、B 接入一个 NAND 门，成为 \overline{AB} ，再通过一个由 NAND 构成的 NOT 门即可。

即， $A \text{ AND } B = \text{NOT } (A \text{ NAND } B) = (A \text{ NAND } B) \text{ NAND } (A \text{ NAND } B)$ 。

3. OR 门:
由摩根定律,

$$\overline{\overline{A} \text{ AND } \overline{B}} = A \text{ OR } B$$

首先, 我们用两个 NAND 门分别对输入 A 和 B 进行 NOT 操作, 然后将这两个 NAND 门的输出连接到第三个 NAND 门的输入端。
即, $A \text{ OR } B = \text{NOT } A \text{ NAND NOT } B = (A \text{ NAND } A) \text{ NAND } (B \text{ NAND } B)$ 。

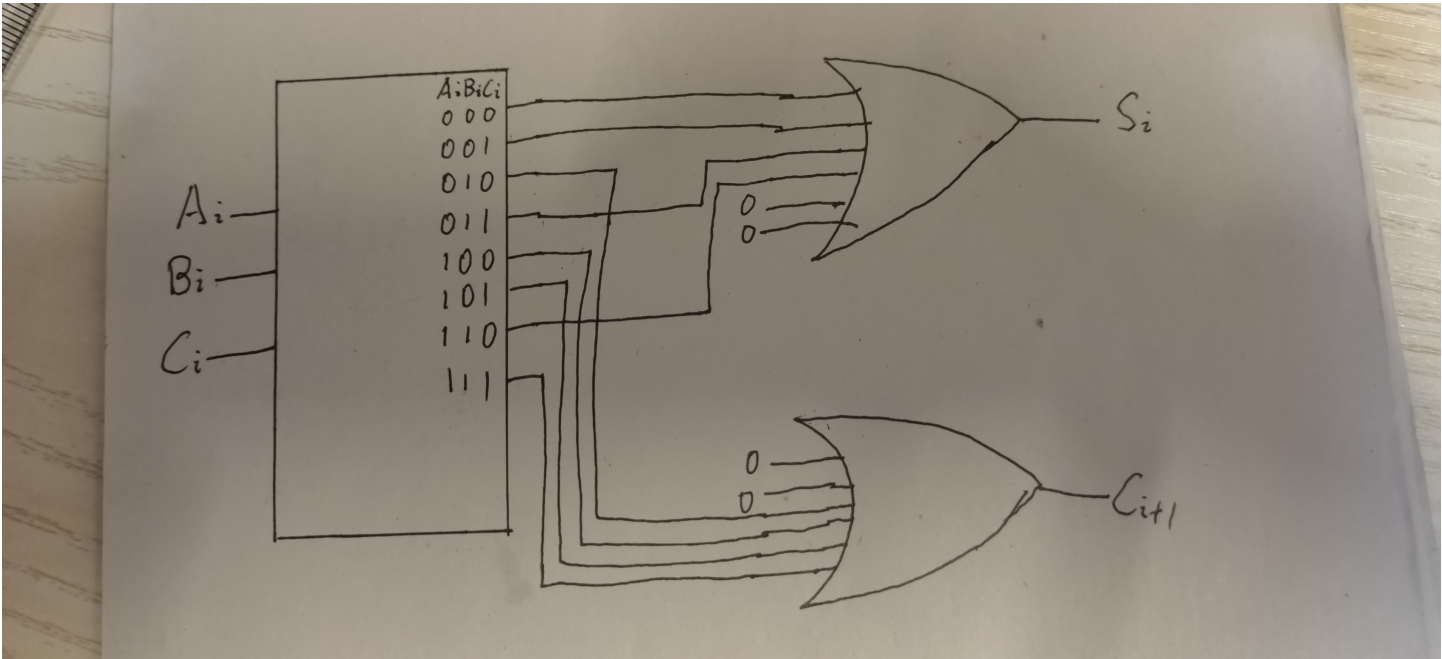
构造出了上述基本逻辑门, 我们就可以构造出其他的所有门, 下面举个例子:

- NOR 门:
将 NAND 构成的 OR 门与 NOT 门连接。首先, 我们用一个 NAND 门对输入 A 和 B 进行 NOT 操作, 然后将这个 NAND 门的输出连接到另一个 NAND 门的输入端。
即, $A \text{ NOR } B = \text{NOT } (A \text{ OR } B) = ((A \text{ NAND } A) \text{ NAND } (B \text{ NAND } B)) \text{ NAND } ((A \text{ NAND } A) \text{ NAND } (B \text{ NAND } B))$ 。

至此, 我们已经证明了只用 NAND 门就可以构造出所有其他的逻辑门, 所以 NAND 是逻辑完备的。

T3

对于求和位 S_i , 将译码器的输出 1、2、4、7 连接到或门;
对于进位位 C_{i+1} , 将译码器的输出 3、5、6、7 连接到另一个或门;
两个或门余下没连的输入全部接 0
由此我们可得电路图如下:



T4

设计一个乘法器, 接收两个 2 位输入 A[1:0]和 B[1:0], 并产生一个输出 Y, 即 A[1:0]和 B[1:0]的乘积。
为了更容易区分, 下面使用 verilog 的形式表示二进制数→(位数)'(进制标识)(值), 如: 2'b11 表示二位二进制数 11

1. 两位二进制输入 A[1:0] 的最大值是 2'b11, 即十进制里的 3
2. 同样, 两位二进制输入 B[1:0] 的最大值是 2'b11
3. 当 A 和 B 都为最大值时, Y 的最大可能值是 4'b1001, 即十进制中的 9。
4. 在回答 3 中我们实际已经给出了, 表示最大值 4'b1001 需要 4 位二进制数。
5. 真值表如下

A[1]	A[0]	B[1]	B[0]	Y[3]	Y[2]	Y[1]	Y[0]
0	0	0	0	0	0	0	0

A[1]	A[0]	B[1]	B[0]	Y[3]	Y[2]	Y[1]	Y[0]
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

6. 根据二进制乘法的定义，我们可以得到：

$$Y[2] = (A[1] \cdot B[1]) \oplus (A[0] \cdot B[1])(A[1] \cdot B[0])$$

- 观察真值表，首先，当 $(A[1] \cdot B[1])$ 为 0 时， $Y[2]$ 也为 0；
- 当 $(A[0] \cdot B[1])$ 与 $(A[1] \cdot B[0])$ 同时为 1 时，说明此时应该向 $Y[2]$ 进位；
- 当异或符号左右两边都为 0 时，此时 $Y[2]$ 显然为 0；
- 当异或符号两边一个为 1 一个为 0 时， $Y[2]$ 不需要向 $Y[3]$ 进位，为 1；
- 当两边都为 1 时， $Y[2]$ 需要向 $Y[3]$ 进位，为 0。

但是，本题中没有提供异或门，不过我们可以利用公式 $A \oplus B = A\overline{B} + \overline{A}B$ 来搭一个异或门。
所以有

$$Y[2] = (((A[1] \text{ AND } B[1])) \text{ AND } (\text{NOT}((A[0] \text{ AND } B[1])(A[1] \text{ AND } B[0])))) \text{ OR } ((\text{NOT}((A[1] \text{ AND } B[1]))) \text{ AND } (A[0] \text{ AND } B[1])(A[1] \text{ AND } B[0]))$$

T5

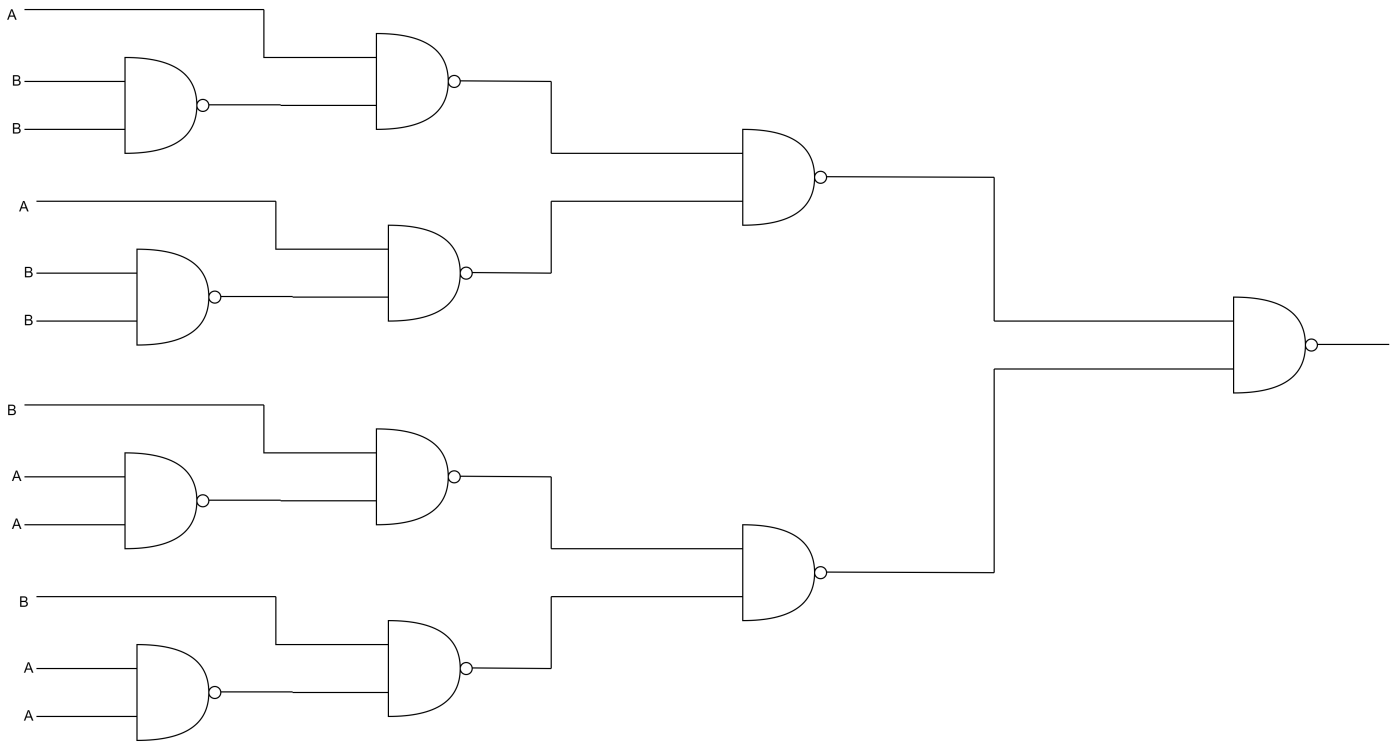
异或门的功能是：当输入的两个信号相同时，输出为 0；当输入的信号不同时，输出为 1。
因为有：

$$A \oplus B = A\overline{B} + \overline{A}B$$

从而结合 T2，我们可以得到：

$$\begin{aligned} A \text{ XOR } B = & (((A \text{ NAND } (B \text{ NAND } B)) \text{ NAND } (A \text{ NAND } (B \text{ NAND } B)))) \\ & \text{NAND} \\ & (((A \text{ NAND } A) \text{ NAND } B) \text{ NAND } ((A \text{ NAND } A) \text{ NAND } B)) \end{aligned}$$

电路图如下：



T6

D 为控制信号，当 A、B、C 符合条件且 D 为 0 时，Z 输出 1；否则，若 D 为 1，或 A、B、C 不符合条件，Z 一律输出 0
写出真值表如下：

A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

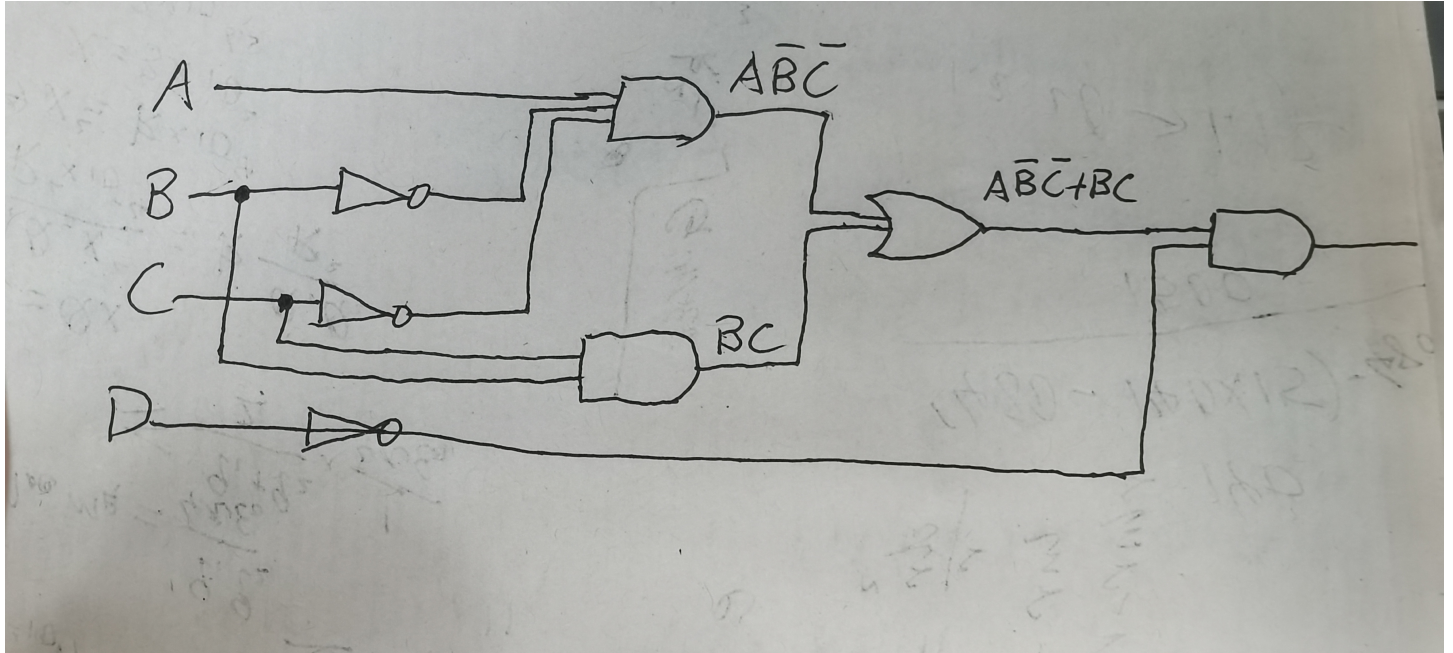
对于 A、B、C 我们有：

$$\begin{aligned}\sum m(3, 4, 7) &= A \cdot \bar{B} \cdot \bar{C} + \bar{A}BC + ABC \\ &= A \cdot \bar{B} \cdot \bar{C} + BC\end{aligned}$$

本题的公式为：

$$A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} + BC\bar{D}$$

逻辑门电路图如下：



T7

有限状态机这里暂时还看不太懂，有点云里雾里，难道是把尾端直接接到第二个状态？

T8

- 根据课本中提到的：内存中可独立识别的位置总数被称为“寻址空间”（address space）
本题中表示地址的 bit 数为 a ，故我们可以识别 2^a 个存储位置。
从而最大的寻址空间是 2^a
- 根据课本中给出的 $2^2 \times 3$ 内存小节的例子，
如果内存的寻址能力是 b 位，那么最多可以存储的位数将等于其最大地址空间乘以寻址能力，
即 $2^a \times b$ 位。

T9

- $A[1:0]$ 的值必须为 00，WE 的值必须为 1。
当 $A[1:0] = 00$ 时，会选择到第一个存储器位置，而 $WE = 1$ 则表示写入操作。
- 需要增加每个存储位置中的 bit 数目至 k 。
 - 解答本题过程中，因为感觉自己的概念有点模糊，上网搜索资料发现很多地方都说“ n 根地址线提供的寻址能力是 2^n ”，但书中有两根地址线，而寻址能力只有 3？？是 $2^n - 1$ 吗，示例有点少，学得不太明白
- 因为

$$2^8 = 256$$

而

$$2^9 = 512$$

所以，如果我们想要充分利用这个 PC 的地址空间，我们需要在图中的存储器中添加 7 条地址线（因为原始存储器只有 2 条地址线）。这样做之后，存储器的寻址能力将会变为

$2^9 = 512$

(菜鸡落泪)

T10

状态	可能的值
占领比例	0~99 (两队)
得分	0~3 (两队)
剩余时间	0:00 到 3:00 之间的任何时间, 格式为 m:s (分钟, 秒)
技能充能比例	0~100 (8 个玩家)

1. 最小位数计算:
- 占领比例: 需要 7 位

$2^7 = 128 > 99$
 $2^6 = 64 < 99$

- 得分: 需要 2 位

$2^2 = 4 > 3$

- 剩余时间: 分别以分钟与秒为单位, 分数最大为 3 分钟、秒数最大为 60 秒, 需要 $2+6 = 8$ 位

$2^2 = 4 > 3$
 $2^6 = 64 > 60$

- 技能充能比例: 需要 7 位

$2^7 = 128 > 100$
 $2^6 = 64 < 100$

- 所以, 总共需要的最小位数是:

$(7 \times 2) + (2 \times 2) + (2 + 6) + (7 \times 8) = 82$

2. 此时存储记分板状态所需的位数仍然是 82 位。
因为这种方法只改变了数据的组织方式, 没有改变数据本身。
3. 分开存储。
这些状态是比赛中的状态, 需要频繁单独访问或修改, 分开存储可以提高访问速度和灵活性。