

逻辑门及其应用

Ming Hsiung

mingshone@163.com

School of Politics and Administration
South China Normal University

主要内容

- 1 逻辑门
- 2 一个应用
- 3 线路图设计
- 4 加法器

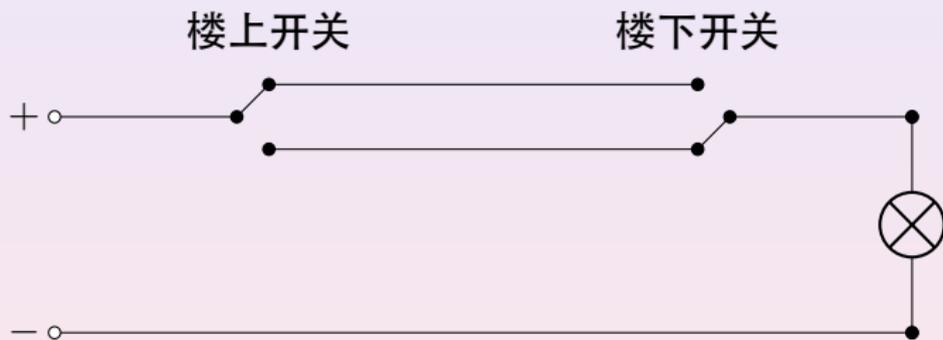
主要内容

- 1 逻辑门
- 2 一个应用
- 3 线路图设计
- 4 加法器

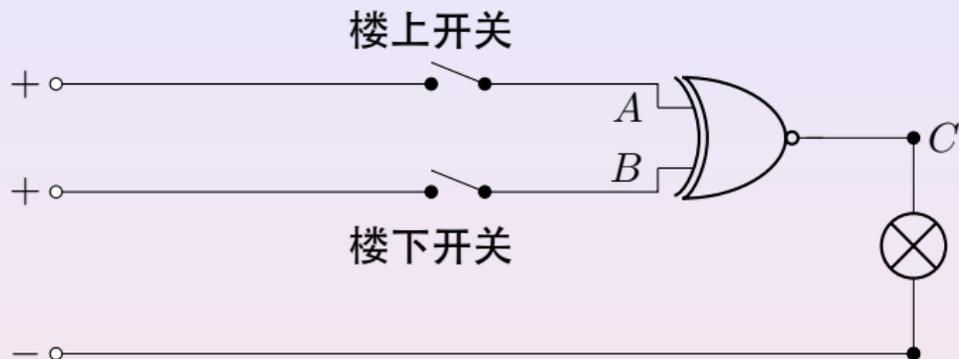
一个开关设计问题

如何用两个开关各自独立地控制一盏灯（即改变任何一个开关的状态，都会改变灯的状态），画出线路图。

使用单刀双掷开关



使用异或门



- A、B 端都断开电流时，C 端无电流
- A 端接通电流，而 B 端断开电流时，C 端有电流
- A 端断开电流，而 B 端接通电流时，C 端有电流
- A、B 端都接通电流时，C 端无电流

异或门及其功能



异或门

| A | B | C |
|-----|-----|-----|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

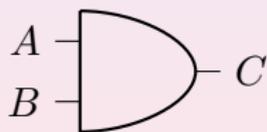
回忆

| A | B | $\neg A$ | $A \vee B$ | $A \wedge B$ | $A \leftrightarrow B$ |
|-----|-----|----------|------------|--------------|-----------------------|
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | | 0 | 0 | 0 |

说明：1 = T, 0 = F

基本的逻辑门

| A | B | C |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |



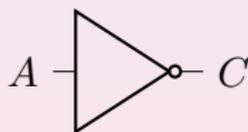
与门

| A | B | C |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |



或门

| A | C |
|---|---|
| 1 | 0 |
| 0 | 1 |



非门

异或门及其功能

| A | B | C |
|-----|-----|-----|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

$$C = (A \wedge \neg B) \vee (\neg A \wedge B)$$

异或门及其功能

| A | B | C |
|-----|-----|-----|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

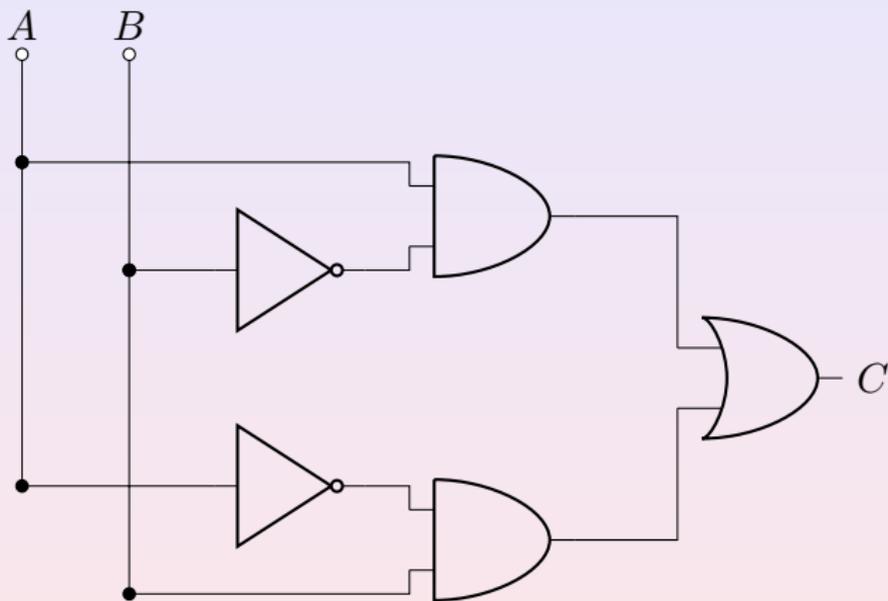
$$C = (A \wedge \neg B) \vee (\neg A \wedge B)$$

异或门及其功能

| A | B | C |
|-----|-----|-----|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

$$C = (A \wedge \neg B) \vee (\neg A \wedge B)$$

异或门的等效组合



$$C = (A \wedge \neg B) \vee (\neg A \wedge B)$$

主要内容

- 1 逻辑门
- 2 一个应用
- 3 线路图设计
- 4 加法器

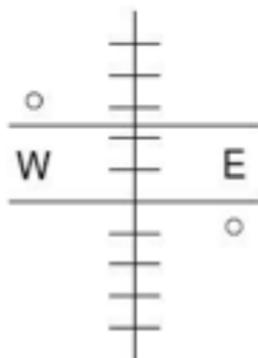
如何控制红绿灯



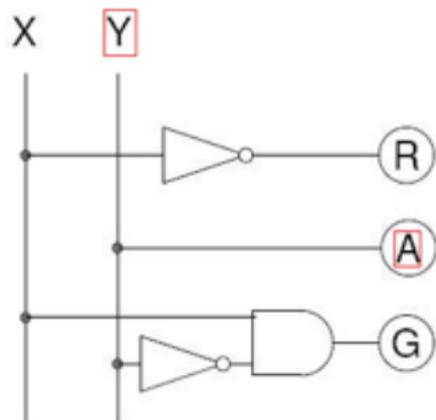
- 红: Red, 黄: Amber, 绿: Green
- 闪烁顺序: 红, 红/黄, 绿, 黄, 红,

Level Crossing

| Inputs | Lights |
|--------|--------|
| X Y | R A G |
| 0 0 | 1 0 0 |
| 0 1 | 1 1 0 |
| 1 0 | 0 0 1 |
| 1 1 | 0 1 0 |



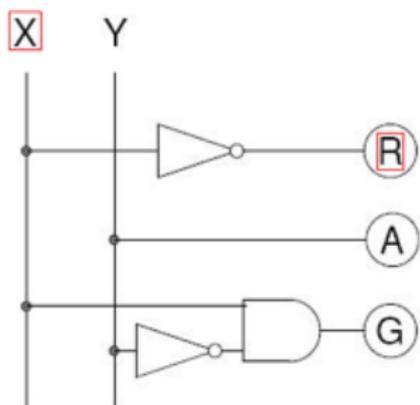
$$A = Y$$



Level Crossing

| Inputs | | Lights | | |
|--------|---|--------|---|---|
| X | Y | R | A | G |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

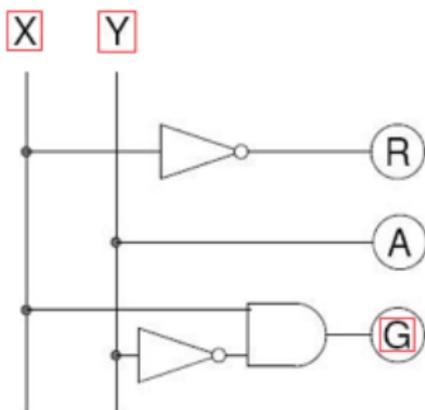
$$R = \bar{X}$$



Level Crossing

| Inputs | | Lights | |
|--------|---|--------|----|
| X | Y | R | G |
| 0 | 0 | 1 | 00 |
| 0 | 1 | 1 | 10 |
| 1 | 0 | 0 | 01 |
| 1 | 1 | 0 | 10 |

$$G = X \text{ AND } \bar{Y}$$



Level Crossing

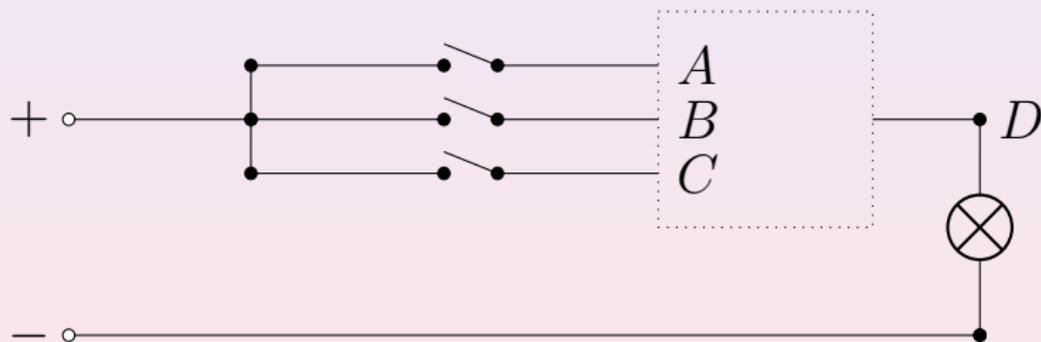
| Inputs | | Lights | |
|--------|---|--------|---|
| X | Y | R | G |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

主要内容

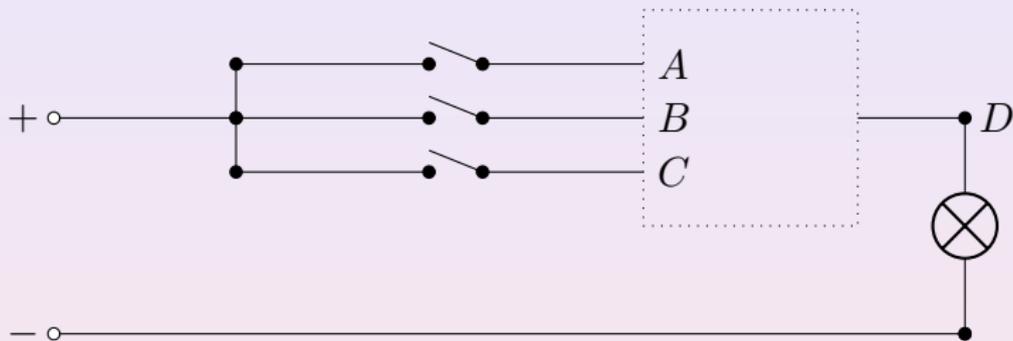
- 1 逻辑门
- 2 一个应用
- 3 线路图设计**
- 4 加法器

问题

用逻辑门来设计线路，使得三个开关可独立地控制一盏灯



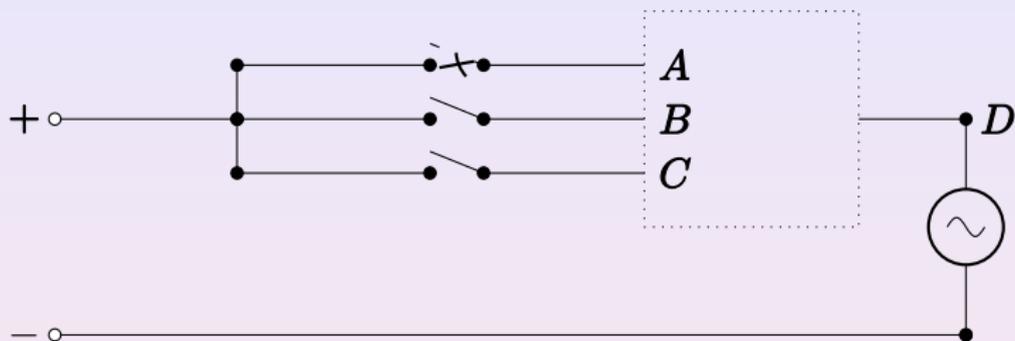
解答



在正常情况下，在三个单刀开关都是断开状态时，灯不亮。

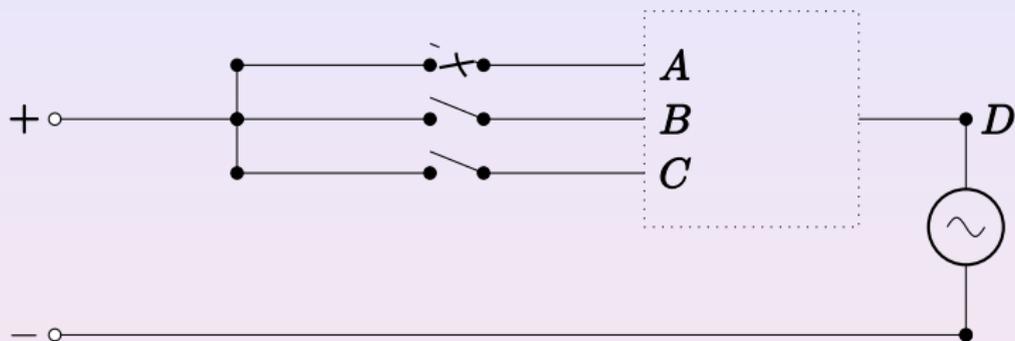
所以，

- 当输入 A 、 B 、 C 都为 0 时，输出 D 应为 0。



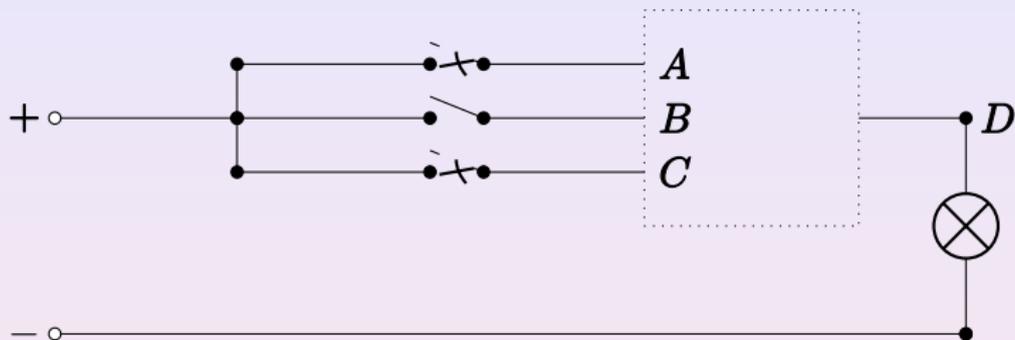
改变任何一个单刀开关的状态（即使得三个单刀开关之一接通）都会使灯亮，所以，

- 当输入 A 、 B 、 C 有且仅有一个为 1 时，输出 D 应为 1。



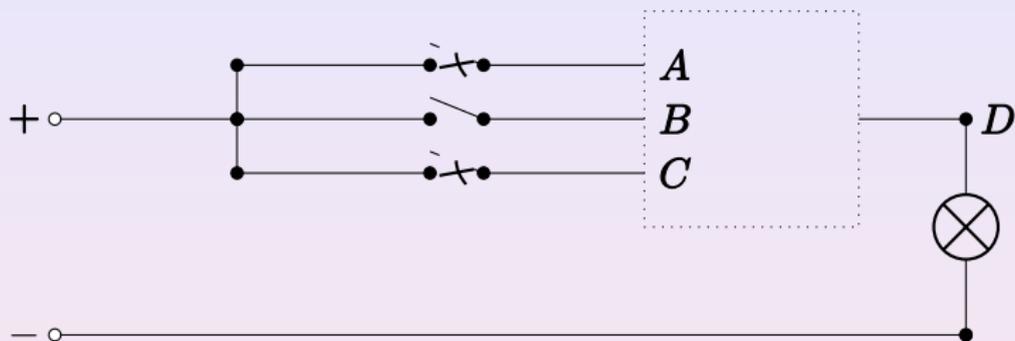
改变任何一个单刀开关的状态（即使得三个单刀开关之一接通）都会使灯亮，所以，

- 当输入 A 、 B 、 C 有且仅有一个为 1 时，输出 D 应为 1。



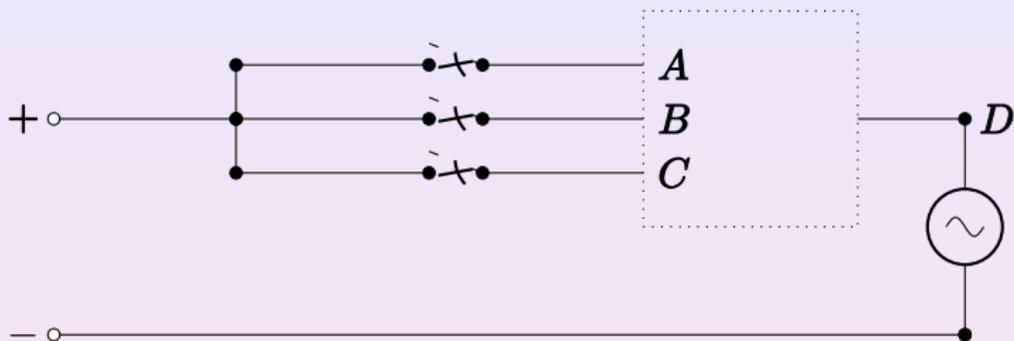
如果再接通三个单刀开关中先前没有的接通的一个，灯的状态又会发生改变，变为熄灭状态，即

- 当输入 A 、 B 、 C 有且仅有两个为 1 时，输出 D 应为 0。



如果再接通三个单刀开关中先前没有的接通的一个，灯的状态又会发生改变，变为熄灭状态，即

- 当输入 A 、 B 、 C 有且仅有两个为 1 时，输出 D 应为 0。



最后，当把三个单刀开关都接通，灯再一次被点亮，亦即

- 当输入 A 、 B 、 C 都为 1 时，输出 D 应为 1。

上面的结果画在下面的表格中：

| A | B | C | D |
|-----|-----|-----|-----|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 |

由此，可以得到，

$$D = (A \wedge B \wedge C) \vee (A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge C)。$$

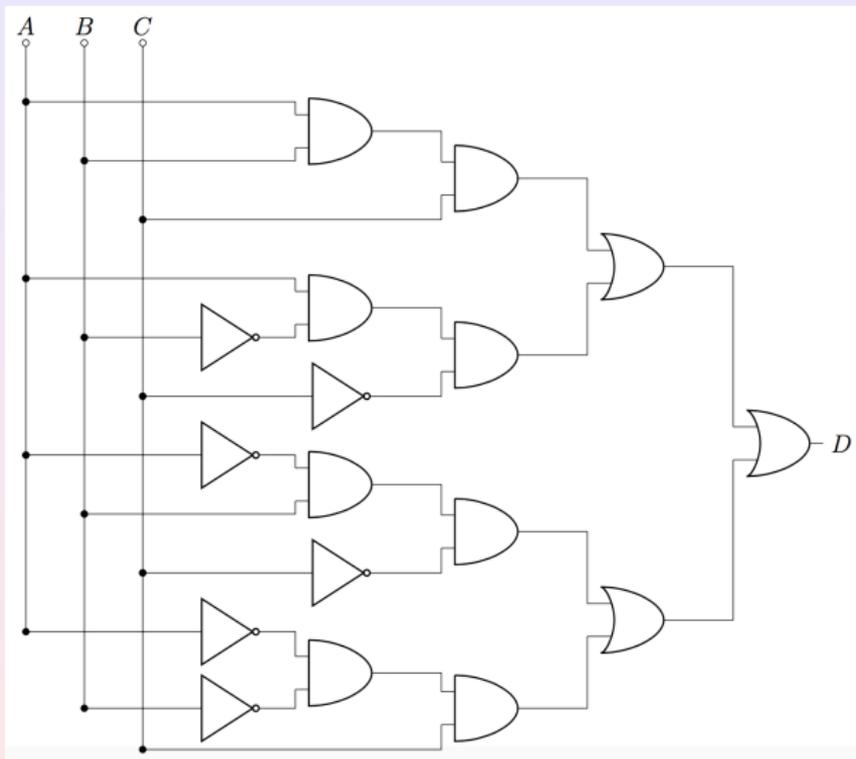
上面的结果画在下面的表格中：

| A | B | C | D |
|-----|-----|-----|-----|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 |

由此，可以得到，

$$D = (A \wedge B \wedge C) \vee (A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge C)。$$

由此，不难画出所需的逻辑门组合，如下图所示：



问题

某公司要求设计如下的保险柜：该保险柜有三把钥匙，分别由经理、会计和出纳三人掌握，他们中任何一人都无法用自己的钥匙打开保险柜，必须至少两个人的钥匙才能打开保险柜。请使用逻辑门设计保险柜的线路。

解答

用 A 、 B 、 C 分别表示经理、会计和出纳的钥匙状态 (1 表示钥匙入锁, 0 表示钥匙不入锁), 则根据保险柜开锁的要求, 可确定 A 、 B 、 C 的输入与输出 D 关系如下:

- 当输入 A 、 B 、 C 都为 0 时, 输出 D 应为 0。
- 当输入 A 、 B 、 C 有且仅有一个为 1 时, 输出 D 应为 0。
- 当输入 A 、 B 、 C 有且仅有两个为 1 时, 输出 D 应为 1。
- 当输入 A 、 B 、 C 都为 1 时, 输出 D 应为 1。

解答

用 A 、 B 、 C 分别表示经理、会计和出纳的钥匙状态 (1 表示钥匙入锁, 0 表示钥匙不入锁), 则根据保险柜开锁的要求, 可确定 A 、 B 、 C 的输入与输出 D 关系如下:

- 当输入 A 、 B 、 C 都为 0 时, 输出 D 应为 0。
- 当输入 A 、 B 、 C 有且仅有一个为 1 时, 输出 D 应为 0。
- 当输入 A 、 B 、 C 有且仅有两个为 1 时, 输出 D 应为 1。
- 当输入 A 、 B 、 C 都为 1 时, 输出 D 应为 1。

解答

用 A 、 B 、 C 分别表示经理、会计和出纳的钥匙状态 (1 表示钥匙入锁, 0 表示钥匙不入锁), 则根据保险柜开锁的要求, 可确定 A 、 B 、 C 的输入与输出 D 关系如下:

- 当输入 A 、 B 、 C 都为 0 时, 输出 D 应为 0。
- 当输入 A 、 B 、 C 有且仅有一个为 1 时, 输出 D 应为 0。
- 当输入 A 、 B 、 C 有且仅有两个为 1 时, 输出 D 应为 1。
- 当输入 A 、 B 、 C 都为 1 时, 输出 D 应为 1。

解答

用 A 、 B 、 C 分别表示经理、会计和出纳的钥匙状态 (1 表示钥匙入锁, 0 表示钥匙不入锁), 则根据保险柜开锁的要求, 可确定 A 、 B 、 C 的输入与输出 D 关系如下:

- 当输入 A 、 B 、 C 都为 0 时, 输出 D 应为 0。
- 当输入 A 、 B 、 C 有且仅有一个为 1 时, 输出 D 应为 0。
- 当输入 A 、 B 、 C 有且仅有两个为 1 时, 输出 D 应为 1。
- 当输入 A 、 B 、 C 都为 1 时, 输出 D 应为 1。

解答

用 A 、 B 、 C 分别表示经理、会计和出纳的钥匙状态 (1 表示钥匙入锁, 0 表示钥匙不入锁), 则根据保险柜开锁的要求, 可确定 A 、 B 、 C 的输入与输出 D 关系如下:

- 当输入 A 、 B 、 C 都为 0 时, 输出 D 应为 0。
- 当输入 A 、 B 、 C 有且仅有一个为 1 时, 输出 D 应为 0。
- 当输入 A 、 B 、 C 有且仅有两个为 1 时, 输出 D 应为 1。
- 当输入 A 、 B 、 C 都为 1 时, 输出 D 应为 1。

解答

用 A 、 B 、 C 分别表示经理、会计和出纳的钥匙状态 (1 表示钥匙入锁, 0 表示钥匙不入锁), 则根据保险柜开锁的要求, 可确定 A 、 B 、 C 的输入与输出 D 关系如下:

- 当输入 A 、 B 、 C 都为 0 时, 输出 D 应为 0。
- 当输入 A 、 B 、 C 有且仅有一个为 1 时, 输出 D 应为 0。
- 当输入 A 、 B 、 C 有且仅有两个为 1 时, 输出 D 应为 1。
- 当输入 A 、 B 、 C 都为 1 时, 输出 D 应为 1。

解答

用 A 、 B 、 C 分别表示经理、会计和出纳的钥匙状态 (1 表示钥匙入锁, 0 表示钥匙不入锁), 则根据保险柜开锁的要求, 可确定 A 、 B 、 C 的输入与输出 D 关系如下:

- 当输入 A 、 B 、 C 都为 0 时, 输出 D 应为 0。
- 当输入 A 、 B 、 C 有且仅有一个为 1 时, 输出 D 应为 0。
- 当输入 A 、 B 、 C 有且仅有两个为 1 时, 输出 D 应为 1。
- 当输入 A 、 B 、 C 都为 1 时, 输出 D 应为 1。

上面的结果画在下面的表格中：

| A | B | C | D |
|-----|-----|-----|-----|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |

可以计算出，

$$D = (A \wedge B \wedge C) \vee (A \wedge B \wedge \neg C) \vee (A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge C)。$$

上面的结果画在下面的表格中：

| A | B | C | D |
|-----|-----|-----|-----|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |

可以计算出，

$$D = (A \wedge B \wedge C) \vee (A \wedge B \wedge \neg C) \vee (A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge C)。$$

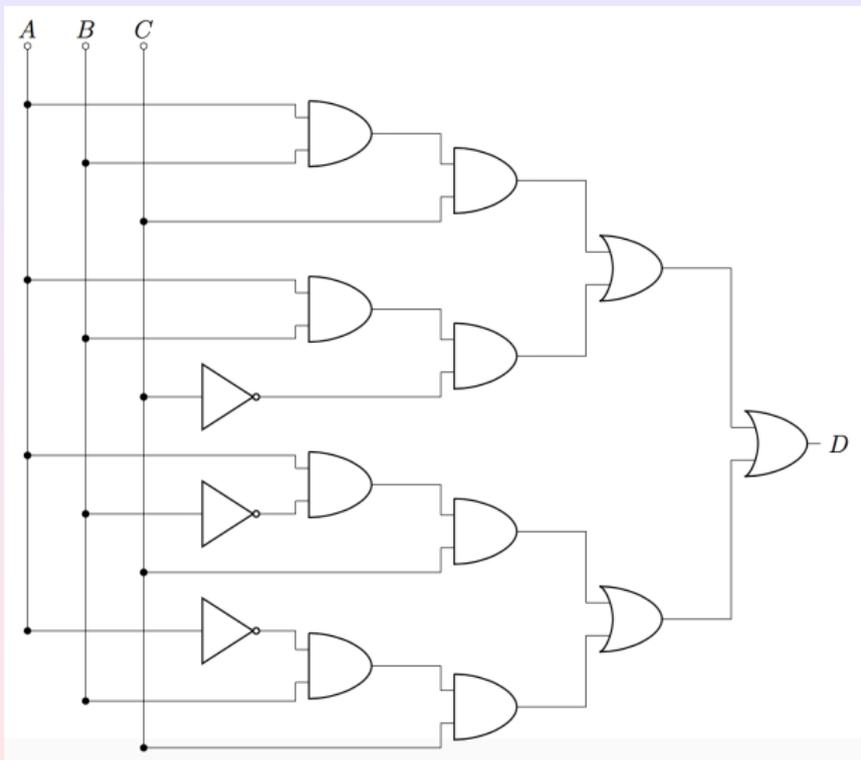
上面的结果画在下面的表格中：

| A | B | C | D |
|-----|-----|-----|-----|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |

可以计算出，

$$D = (A \wedge B \wedge C) \vee (A \wedge B \wedge \neg C) \vee (A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge C)。$$

由此，不难画出所需的逻辑门组合，如下图所示：



主要内容

- 1 逻辑门
- 2 一个应用
- 3 线路图设计
- 4 加法器**

二进制加法简介

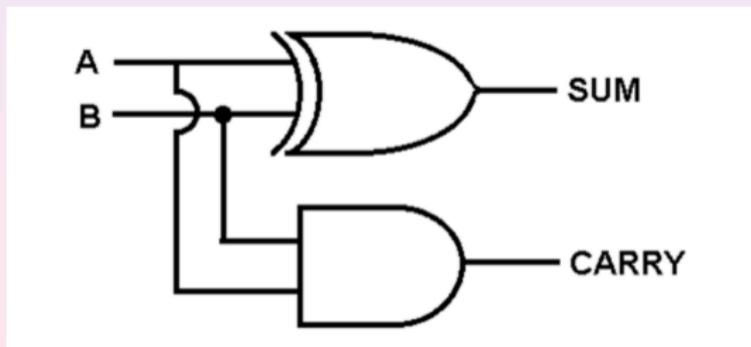
- 如同十进制有 0, 1, 2,, 9 十个数字, 二进制有两个数字: 0, 1。
- 一位二进制数的加法基本规则: $0+0 = 0$, $0+1 = 1+0 = 1$, $1+1 = 10$ (1 为进位)。
- 多位二进制数的加法演示:

| | | | | | |
|--|---|---|---|---|---------|
| | 1 | 1 | 1 | 1 | ← carry |
| | 1 | 1 | 1 | 0 | 1 |
| (+) | 1 | 1 | 0 | 1 | 1 |
| <hr style="border: 1px solid black;"/> | | | | | |
| | 1 | 1 | 1 | 0 | 0 |
| | 1 | 1 | 1 | 0 | 0 |
| <hr style="border: 1px solid black;"/> | | | | | |

Circuit Globe

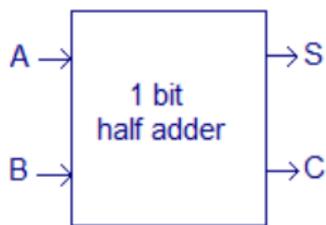
半加法器 (half adder)

- 一位二进制数的加法基本规则可用半加法器来“封装”。
- 半加法器是如下的逻辑门组合：

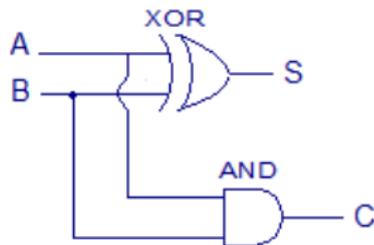


| Inputs | | Outputs | |
|--------|---|---------|---|
| A | B | S | C |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Truth table



Schematic

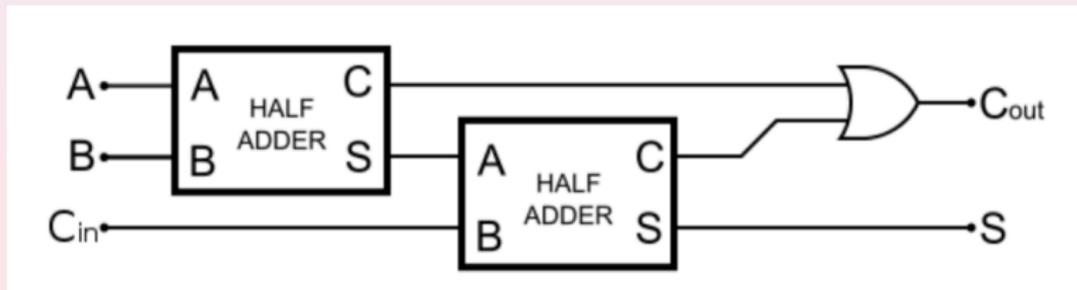


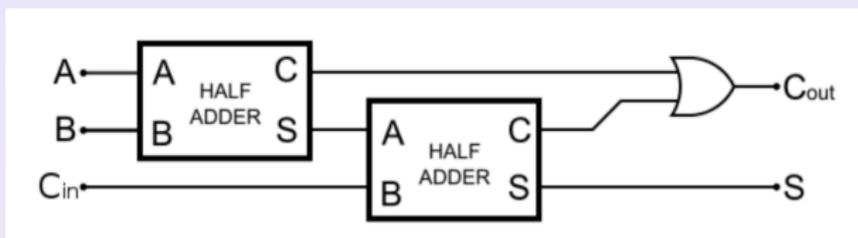
Realization

半加法器演示： 点击 [Half Adder in Wiki](#)

全加法器 (full adder)

- 全加法器将两个一位二进制数相加，并根据接收到的低位进位信号，输出和、进位输出。全加器的三个输入信号为两个加数 A 、 B 和低位进位 C_{in} 。全加器的输出和半加器类似，包括向高位的进位信号 C_{out} 和本位的和信号 S 。
- 全加法器也是逻辑门组合，但可以用已经封装好的组合来进行再组合即可：

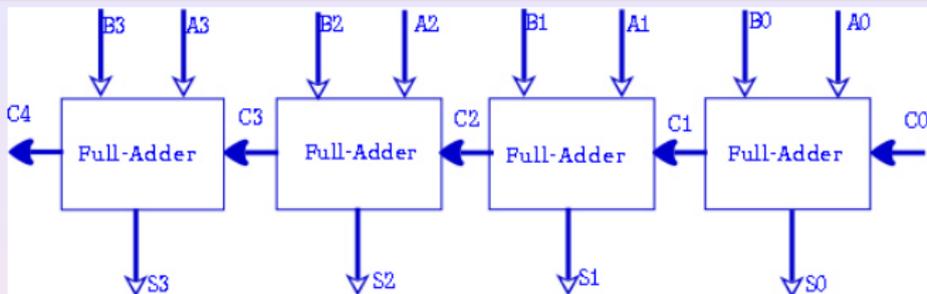




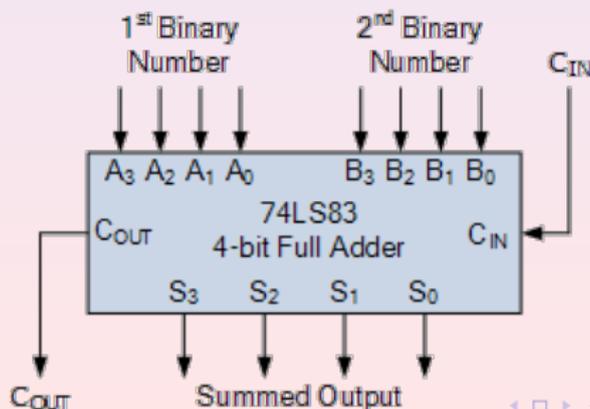
| A | B | Carry In | Sum | Carry out |
|---|---|----------|-----|-----------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

全加法器演示：点击 [Full Adder in Wiki](#)

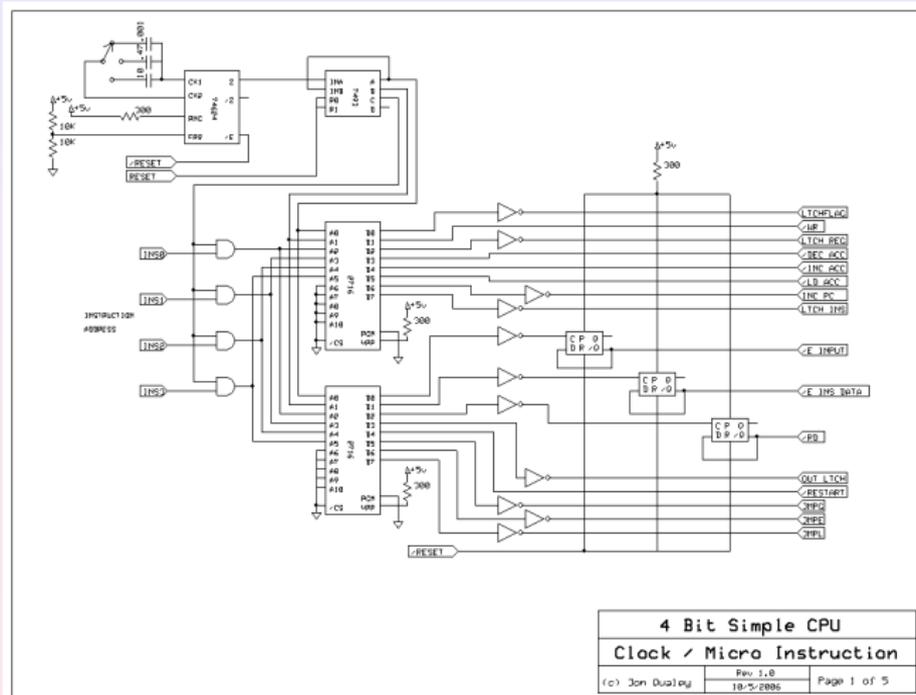
4 位二进制数的加法器



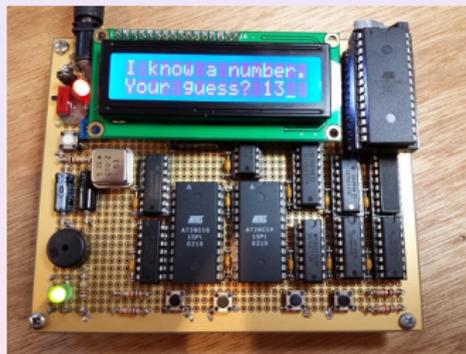
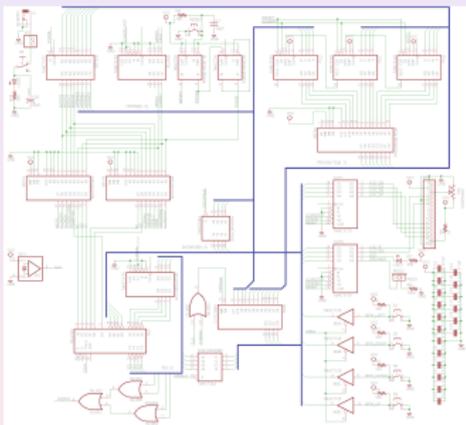
进一步封装



4 位二进制数的一个 CPU



Nibbler 4 Bit CPU



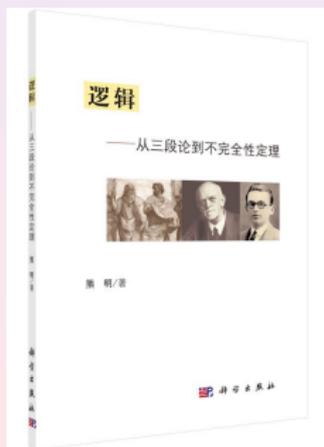
Intel CPU



参考资料

更多的细节，请参考：

- 熊明，《逻辑：从三段论到不完全性定理》（第9章），科学出版社，2016年



视频资料

- 点击：[超形象分析计算机晶体管的工作原理](#)
- 点击：[计算机晶体管巧妙实现全加器](#)

Thanks for your attention!

Q & A