## 半导体

通过电压来控制电路开关，而不是用机械开关。

好处：响应快(~10M HZ)

晶体二极管：单向导通

晶体三极管：外部电压控制电路是否连通

## 逻辑门电路

|  |  |  |  |
| --- | --- | --- | --- |
| 门 | 符号 | 逻辑 | 电路 |
| 与(AND) |  | Y = A & B |  |
| 或(OR) |  | Y = A | B |  |
| 非(NOT) | A picture containing wire  Description automatically generated | Y = !A |  |

Note: A, B is input, Y is output.

|  |  |  |
| --- | --- | --- |
| 门 | 逻辑 | 电路 |
| 异或(XOR) | Y = |  |
| 与非(NAND) | Y = !(A & B) |  |
| 或非(NOR) | Y= !(A | B) |  |

Note: Vcc 接电源正极， GND接电源负极。

## 组合逻辑电路

特点: 在任意时刻的输出仅仅取决于该时刻的输入，与电路以前的状态无关。可以用来完成逻辑运算，但是没有记忆功能, stateless

对模拟信号进行编码 =>

将需求转换为真值表 =>

根据真值表，使用门电路设计组合电路

## 时序逻辑电路

特点：可以记忆电路的状态，作为下一次的输入.

触发器(flip-flop)：记忆1 bit 二进制信号.

内存数据<==(LOAD/STORE)==>寄存区的本质是什么？

内存和寄存器不过相当于一堆开关(比如内存数据是01010000，它本质上就是8个开关的开闭状态)，它们通过导线连接。那么数据加载到寄存器，使得寄存器的开关状态也变成了01010000, 那么是怎么实现的呢？

寄存器和内存的基本单元是锁存器(更高级的是触发器)，可以让输入的数据保持住，只需一次输入，即使后面输入全部撤销或发生改变， 输出也不会改变，只需要保持通电状态即可(通常元器件需要专门的引脚接电源)。

Diagram, schematic

Description automatically generated

那么寄存器其实也是多组锁存器的组合(每个bit一个锁存器)， 只要一次输入（比如01001000），就会被保存，以后的输出总是和输入相同（01001000）， 从而实现保存数据。

那么可以再加一个时钟引脚，当时钟信号为1时，允许重新输入，当时钟信号为0时，锁存器不响应输入的变化，保持之前状态。

A picture containing text, whiteboard

Description automatically generated

### CPU指令执行

Liang: 在冯诺依曼体系提出之前，计算机像一个黑盒，有一个或多个输入(假设有8个开关表示八个bit)，然后有一个输出。

首先，程序员需要：

1. 调整输入开关来表示输入的二进制数(每个开关表示一个bit);

2. 调整控制加法器到底是加法还是减法的开关;

3. 控制电源通电，得到一个输出.

4. 重复上述步骤执行下一条指令

Diagram

Description automatically generated

* CPU的Core相当于黑盒，输入/输出/控制位(二进制数: 也就是开关的开闭状态) 保存在寄存器中。

*注：CPU一直处于通电状态，此处省略接电脚针*

* Clock：由于电路中电流传输需要一定的时间，所以需要等待一段时间让整个电路达到一个稳定的状态。一条指令需要用到多个总线上的设备, 所以需要一个clock来协调各个设备的协同；

*注: CPU内部集成了时钟电路，每次在时钟信号上沿执行指令并输出。*

汇编语言可以简单地理解为二进制机器语言的一种符号表示，跟机器语言是一对一的关系。使用汇编语言执行一条指令的过程大致就是：

* LOAD立即数/内存地址/其他寄存器 into register.
* 执行指令: ADD reg\_1, reg\_2
* STORE register into memory