

Language, HDL)来表达的。HDL 是一种文本表示,看上去和编程语言类似,但是它是用来描述硬件结构而不是程序行为的。最常用的语言是 Verilog, 它的语法类似于 C; 另一种是 VHDL, 它的语法类似于编程语言 Ada。这些语言本来都是用来表示数字电路的模拟模型的。20 世纪 80 年代中期,研究者开发出了逻辑合成(logic synthesis)程序,它可以根据 HDL 的描述生成有效的电路设计。现在有许多商用的合成程序,已经成为产生数字电路的主要技术。从手工设计电路到合成生成的转变就好像从写汇编程序到写高级语言程序,再用编译器来产生机器代码的转变一样。

我们的 HCL 语言只表达硬件设计的控制部分,只有有限的操作集合,也没有模块化。不过,正如我们会看到的那样,控制逻辑是设计微处理器中最难的部分。我们已经开发出了将 HCL 直接翻译成 Verilog 的工具,将这个代码与基本硬件单元的 Verilog 代码结合起来,就能产生 HDL 描述,根据这个 HDL 描述就可以合成实际能够工作的微处理器。通过小心地分离、设计和测试控制逻辑,再加上适当的努力,我们就能创建一个可以工作的微处理器。网络旁注 ARCH:VLOG 描述了如何能产生 Y86-64 处理器的 Verilog 版本。

4.2.1 逻辑门

逻辑门是数字电路的基本计算单元。它们产生的输出,等于它们输入位值的某个布尔函数。图 4-9 是布尔函数 AND、OR 和 NOT 的标准符号, C 语言中运算符(2.1.8 节)的逻辑门下面是对应的 HCL 表达式: AND 用 `&&` 表示, OR 用 `||` 表示,而 NOT 用 `!` 表示。我们用这些符号而不用 C 语言中的位运算符 `&`、`|` 和 `~`, 这是因为逻辑门只对单个位的数进行操作,而不是整个字。虽然图中只说明了 AND 和 OR 门的两个输入的版本,但是常见的是它们作为 n 路操作, $n > 2$ 。不过,在 HCL 中我们还是把它们写作二元运算符,所以,三个输入的 AND 门,输入为 a 、 b 和 c ,用 HCL 表示就是 `a&&b&&c`。

逻辑门总是活动的(active)。一旦一个门的输入变化了,在很短的时间内,输出就会相应地变化。

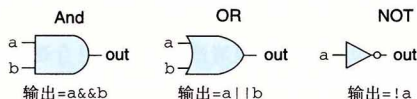


图 4-9 逻辑门类型。每个门产生的输出等于它输入的某个布尔函数

4.2.2 组合电路和 HCL 布尔表达式

将很多的逻辑门组合成一个网,就能构建计算块(computational block),称为组合电路(combinational circuits)。如何构建这些网有几个限制:

- 每个逻辑门的输入必须连接到下述选项之一: 1) 一个系统输入(称为主输入), 2) 某个存储器单元的输出, 3) 某个逻辑门的输出。
- 两个或多个逻辑门的输出不能连接在一起。否则它们可能会使线上的信号矛盾,可能会导致一个不合法的电压或电路故障。
- 这个网必须是无环的。也就是在网中不能有路径经过一系列的门而形成回路,这样的回路会导致该网络计算的函数有歧义。

图 4-10 是一个我们觉得非常有用的简单组合电路的例子。它有两个输入 a 和 b , 有唯一的输出 eq , 当 a 和 b 都是 1(从上面的 AND 门可以看出)或都是 0(从下面的 AND 门可以看出)时,输出为 1。用 HCL 来写这个网的函数就是:

```
bool eq = (a && b) || (!a && !b);
```