数字电路学习笔记（五）：逻辑设计基础

JoshCena

马上就要正式进入电路设计了，再来看最后一个知识点：逻辑设计吧。

之前我们花了两章，探讨了逻辑运算是什么，怎么算；但还有最后一个大问题，巧妇难为无米之炊，我们得先有一个逻辑式，才能对它化简，并基于结果做电路设计。所以，如何把实际生活中的问题转化为逻辑函数式呢？

先介绍两种比较标准的函数的形式：

- 最小项表达式，是若干单项式相加，可以类比成代数式展开后的样子，形式类似 $ABC+ABC'+AB'C+A'B'C'$，得名于其因为项中用乘法连接而使得每一项为 1 的概率都很小。
- 最大项表达式，是若干多项式相乘，可以类比成代数式因式分解后的样子，形式类似 $(A'+B'+C')(A+B+C')(A+B'+C)(A'+B'+C')$，得名于其因为项中加法连接而使得每一项为 1 的概率都很大。

其中实际中用得比较多的是最小项表达式。本文也将在此式的基础上讨论。

进一步地，我们还会再看逻辑函数与其他表现形式的转化关系，并正式介绍两个工具：逻辑图和卡诺图。

一、从对问题的描述得出函数式

我们开篇的问题中，就是利用了把问题转化为比较“标准”的逻辑命题，加以处理的。一般地说，通过自然语言表述得出方程，首先要把任务写成“只有... 且... 时，才...”（最大项表达式）或“只要... 或... 时，就...”（最小项表达式）的形式。它要求该问题不能包含太复杂的嵌套关系，变量也不能太多，所以局限比较大。但用这种方法得出的函数式往往可以省去化简的步骤。

二、从真值表得出函数式

假设我们有这样一个真值表：

A	B	C	D	X
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

如何方便地写出它的表达式?

先试试把它转述为自然语言后照着写：“只要 A，B，C，D 分别为 0000，或者 0101，或者 0110，或者...... 时，X 就为 1”。这样，的确可以写出方程。

那么，能不能直接通过真值表写出函数式呢？想一想，之前提到，所有的描述都可以转化为所谓“最小项表达式”，其中每一个项都是一个单项式，A、B、C、D 分别取原变量或反变量。比如先看第一行，当 $ABCD$ 取 0000 时，最终结果为 1。——这对应了 $A'B'C'D' = 1$。所以，从真值表写出最小项表达式的方法是先找出所有使得因变量为 1 的自变量值的组合，再把每一个组合对应的乘积项写出来，每个变量取值为 0 则写上反变量，1 则写上原变量，这样使得取这组值时该项为 1，最后把这些项用加法相连即可。

举个例子：我们至今没有推导过异或逻辑如何用与或非表达。现在让我们证明一遍。

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

有了真值表，就可以直接看使得 $X = 1$ 的 A，B 取值组合，发现有 01，10 两个；接着，分别写出它们对应的乘积项，即 $A'B$ 和 AB'；最后把它们连在一起：$AB' + A'B$，即可。

三、从函数式得出真值表

要画真值表，首先必须把所有自变量可能的取值都填在前几列中。建议使用二进制排列，即 0000，0001，0010⋯⋯等，不容易缺漏。

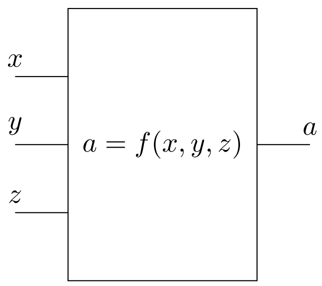
接下来，把函数式化成最小项表达式。比如，如果初始式为 $X = A(BC' + AC)$，则需要先展开成 $X = ABC' + AC$，再进一步写成每一项都含有三个变量的标准形式——具体地说，运用公式 $AB = AB(C + C') = ABC + ABC'$，得到 $X = ABC' + AB'C + ABC$。最后，在对应的真值表行中“X”一栏填上“1”，其他行则填 0——此处，三项分别对应 110，101，111。

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

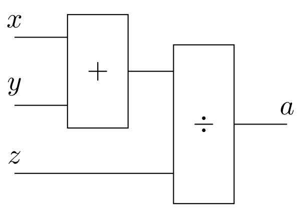
如果说这种方法是反向填表，那自然还有另一种正向方法——把每一行的 A，B，C 的值依次代入函数式，求出 X 的值。这种方法适用于变量较多，而且原式比较简单，如果完全展开很费时间的式子。

四、逻辑图与函数式的转化

什么是逻辑图？我们知道，任何一个函数，比如 $a = f(x, y, z)$，都可以表示为一个“黑箱”——

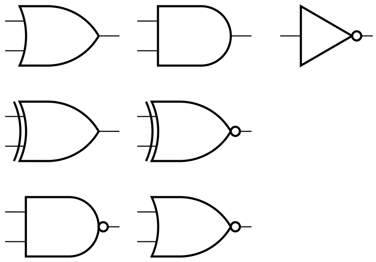


只要输入一个 x，一个 y，一个 z，这个黑箱就会返回一个对应的 a 的值。而如果我们想查看它的内部逻辑，我们可能会看到这样的：



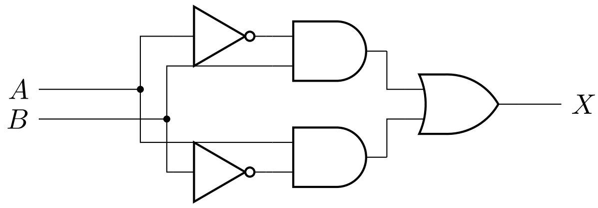
“+”把 x 和 y 连接起来，作 $m = x + y$ 运算；“÷”又连接了“+”的输出和 z，作 $a = m \div z$ 运算，并输出 a。所以，这个图就可以表示 $a = \dfrac{x+y}{z}$。

同样地，还记得七个基本逻辑的逻辑符号么？



从左至右，从上至下：与、或、非、异或、同或、与非、或非

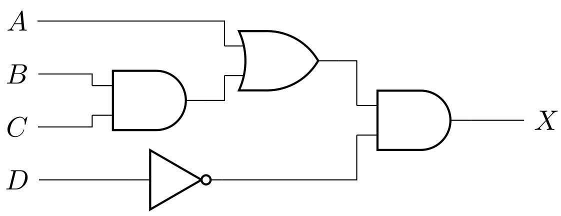
如果我们有式子 $X = A'B + AB'$，就可以用同样的思路，连出一个图：



上半部分，得到 $A'B$；下半部分，是 AB'。两个再用或连接，就有了 $A'B + AB'$。这样，便可以把一个函数式直观地表达出来。并且，在实际的电路制作中，这样的设计图也可以成为抽象的逻辑式与实际的电路板间的桥梁。

所以，要想绘出逻辑图，一般来说，只要先理清函数的运算顺序，再把对应的逻辑符号用线连接起来即可。再看一个例子：经典的档案室开门问题，已有函数式：$X = (A + BC) \cdot D'$

一步步看它的运算顺序：B 与 C 相乘；A 与乘积相加；D 的反变量与这个和相乘。最后输出 X。因此，可以画出对应的逻辑图。



反向地，如果有了这个逻辑图，就可以通过沿着逻辑图走向分析，最终得出函数式。

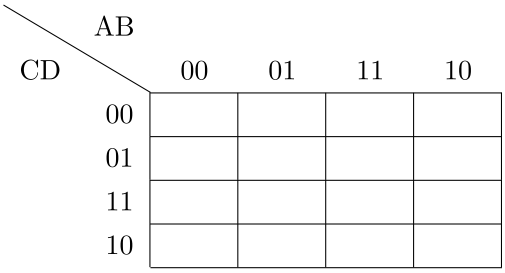
还可以发现，逻辑图和计算机中的运算树本质上是相同的。所以，我们可以用中序遍历的思路，写出函数式。

五、卡诺图

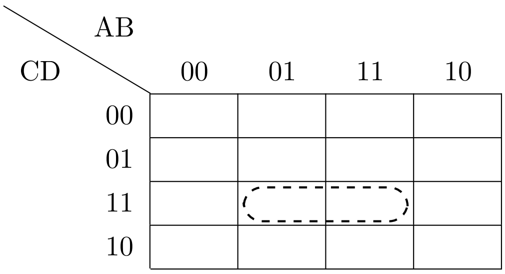
在最小项表达式的化简中，我们本质上在做什么呢？比如，$AB'C + ABC$——两项中只有一个变量不同，所以变成了 AC。

如果我们有 n 变量最小项表达式，那么它至多可以有 2^n 项——因为每个变量都会以原变量或反变量出现。而如果原变量表示 1，反变量表示 0，则每一项都可以对应一个唯一的 n 位二进制码，比如 $AB'C'D$ 就是 1010。

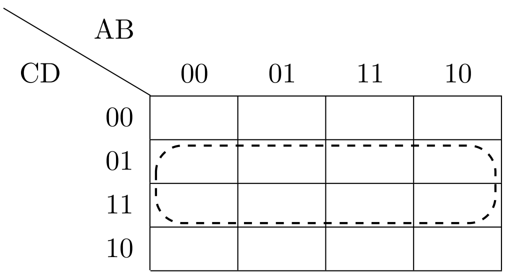
而 $AB'C, ABC$ 对应的二进制数为 101，111，也只差一个数位。那么，能不能用一个表格，可以容纳所有的可能项，并直观地发现这些能够合并的”相邻项“呢？于是，工程师莫里斯 · 卡诺便创造了卡诺图。



这是卡诺图的一般形式。先把所有相关变量分为数量大致相等的两部分，一部分（AB）沿行布置，一部分（CD）沿列布置。再把它们以格雷码编码——00，01，11，10 等；这样，每个格子就对应了最小项表达式中的唯一一个项，比如 $AB'C'D$ 就位于第四列，第二行位置，对应 AB=10，CD=01。并且注意到，如果两个项可以化简，如 $A'BCD + ABCD$：



则它们在卡诺图上必定相邻。它背后的原理是当 $BCD = 111$ 时，无论 A 为 0 还是 1，结果都为 1，所以 A 就成了无关项，可以消去。按照这个道理，只要是圈起的区域是**大小为 2^n 的矩形**，那么都可以化简成一个项：



圈起的区域覆盖了所有 A、B 的值，所以 A、B 都是无关项；同时还覆盖了 CD=01 与 11，所以 C 也是无关项，最终结果为 D。

卡诺图并不是二维的，而是循环的，可以从一个边界来到另一个边界，形成一个空间中闭合的形状。比如：

CD \ AB	00	01	11	10
00				
01				
11				
10				

也是可以合并的。

使用卡诺图化简时，先把所有最小项在表中对应的位置打勾，再用圈覆盖这些勾。具体有三个原则：

- 圈可以相互重叠交错，但必须覆盖所有勾，且不能覆盖空白格，否则会使函数发生改变。
- 每个圈必须尽量大，这样可以尽可能多地消去无关变量。
- 圈要尽量少，这样会使最终的项的数量最少，因为每个圈会对应最简式中的一个项。

比如，化简函数

$$X = A'B'C'D' + A'B'CD' + A'BC'D + A'BCD' + A'BCD + AB'CD' + ABC'D + ABCD$$

先把所有项填入卡诺图：

CD \ AB	00	01	11	10
00	✓			
01		✓	✓	
11		✓	✓	
10	✓	✓		✓

再把它们全部圈起来，尽量扩大圈的大小和减少圈的数量：

CD \ AB	00	01	11	10
00	✓			
01		✓	✓	
11		✓	✓	
10	✓	✓		✓

圈起来的区域有：(0000,0010), (0010,0110), (1010,0010), (0101,1101,0111,1111)。所以化简之后，剩下的项分别为：$A'B'D'$, $A'CD'$, $B'CD'$, BD。所以，我们最终的化简结果便是

$$X = \color{red}{A'B'D'} + \color{green}{A'CD'} + \color{blue}{B'CD'} + BD$$

卡诺图的使用需要一定技巧，所以不是非常常用。但是，卡诺图和真值表都可以作为函数的可视化表达。在“时序电路”一部分，我们将会看到它的作用。