

第十章 专用集成电路的测试

随着集成电路的规模越来越大，测试问题也日益尖锐地摆在了芯片设计师们的面前。流片结束之后的芯片测试与电路设计阶段的电路模拟不同，电路模拟验证电路的设计是否正确，而芯片测试是检验在正确的电路设计之下，生产线上流出的芯片是否成功。这是两者的第一点不同。

第二，对于设计阶段的电路模拟(包括逻辑模拟)，测试者尽可以在充足的模拟时间内，用完备的测试矢量集(即输入数据)对电路进行测试。当发现问题时，计算机测试软件还可以将测试点深入到电路内部任意一个结点，以确定故障所在。但对于流片之后的芯片测试，问题就不是那么简单，它要考虑如下一些问题：

1. 根据正确的电路设计而生产的芯片，由于加工过程引入的离散型缺陷，比如某个引线孔未刻蚀干净造成电路开路，或某两条金属线之间发生粘连造成电路短路等，使芯片不能正常工作。这就是所谓的“成品率”问题。现时的生产工艺尚无法保证 100%的成品率，这就意味着对每一只生产出来的芯片都必须进行功能测试，以分检出成品与次品。我们不防称这样的测试为“功能测试”。功能测试检测产品的好坏。

2. 如果是由于某个共同的原因(或故障)，比如掩膜设计或制作时出了问题没有被发现，而这一问题又被复制到整块掩模板上的所有电路，造成整批电路的投片失败。或者经功能测试发现成品率太低等等。总之，在所有这些故障已确定存在的情况下，我们都希望能有一种测试方法，通过它能发现问题的原因也就是故障所在，从而找出相应的解决办法。这类测试可称之为是“故障测试”。故障测试确定故障所在。

下面就来分述这两类测试问题。

§ 10-1 功能测试

如前所述，功能测试(包括性能测试)的主要目的是区分好产品而淘汰坏产品。测试速度非常重要。

对于一个大规模的集成电路，其输入/输出管脚可能很多。理论上讲，只有穷尽了输入信号的所有组合，并验证了输出信号相应无误后，才能断定此电路是合格产品。比如对于一个二输入端的“与非”门电路，只有测试了 00, 01, 10, 11 四种输入状态下的输出分别是 1, 1, 1, 0, 才能断定该“与非”门是合格产品。这就如同在电路设计阶段，电路的逻辑模拟(逻辑仿真)所做的工作一样。

然而在实际应用中，这样的穷尽测试法是不太现实的。下一节将要介绍的提取测试矢量集的方法能在一定程度上缓解穷尽法对测试量的压力，也是当前集成电路测试的常规方法，但需要价格昂贵的专用集成电路测试设备。对于小批量 ASIC 产品的研制人员和客户来说，如果能够找到适合他们产品特点的简单易行的测试方法，降低测试成本，也是非常有意义的。

下面就是一个简化测试的典型例子。

图 10-1 是一个乘法/累加器专用集成电路的电路图。该电路的输入端为被乘数 X (8bits)、乘数 Y (7bits)和加数 Z (15bits)，输出端为 C (15bits)。为验证该芯片功能的正确性，用穷尽法将要做 $2^8 \times 2^7 \times 2^{15}$ 次测量，测试量是非常巨大的。但如果采用基于穷尽法原理设计出的“准穷尽法”测试，将会发现后者是这一电路比较具有实用价值的测试方法。

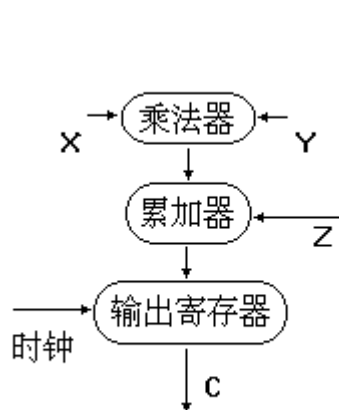


图 10-1 乘法/累加器专用集成电路框图

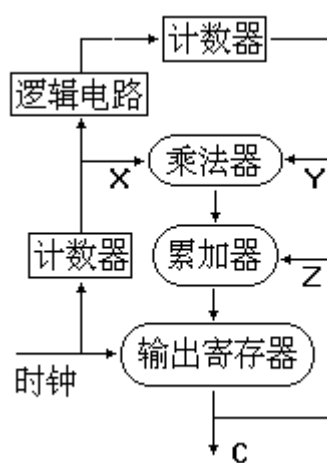


图 10-2 乘法/累加器的准穷尽测量法

图 10-2 是该乘法/累加器电路的准穷尽测试法示意图。将芯片的输出端 C 与累加输入端 Z 相连。将整个电路清零后，固定乘数 $Y=1$ ，让被乘数 X 通过对时钟脉冲的计数，从 1 到 255 不断变化，进行 X 与 Y 的乘法操作，并将乘/加结果 C 通过输出寄存器(亦在时钟的控制下)返回给累加器，与下一拍的乘积进行不断的累加。当 X 从 1 变到 255 之后，再控制变量 Y 加 1，即让 $Y=2$ ，再进行第二轮 $X \times Y + Z$ 的乘法累加运算(X 仍从 1 变到 255)。如此反复，直到 Y 增加到最大值 127 为止。这样就完成了 X 和 Y 的穷尽法相乘，以及与 Z 的准穷尽法相加，并且只有一个最终累加结果。将这一最终累加结果从 C 中取出，看它是否与它应该是的真值 C_0 相等，如果相等就可以认为电路工作正常，也就是功能正常。

这一电路的真值 C_0 可以这样计算获得：

$$\begin{aligned}
 C_0 &= \sum_{i=1}^{127} \sum_{j=1}^{255} i \times j \\
 &= 1 \times 1 + 1 \times 2 + \dots + 1 \times 255 +
 \end{aligned}$$

$$\begin{aligned}
& 2 \times 1 + 2 \times 2 + \dots + 2 \times 255 + \\
& \dots + \\
& 127 \times 1 + 127 \times 2 + \dots + 127 \times 255 \\
= & 1 \times (1 + 2 + \dots + 255) + \\
& 2 \times (1 + 2 + \dots + 255) + \\
& \dots + \\
& 127 \times (1 + 2 + \dots + 255) \\
= & (1 + 2 + \dots + 127) \times (1 + 2 + \dots + 255) \\
= & [(1 + 127) \times 127 / 2] \times [(1 + 255) \times 255 / 2] \\
= & 8128 \times 32640 \\
= & 265297920
\end{aligned}$$

将这个十进制数转化为二进制数，得：

32 24 16 8

0 0 0 0 1 1 1 1	1 1 0 1 0 0 0 0	0 0 1 0 0 0 0 0	0 0 0 0 0 0 0 0
-----------------	-----------------	-----------------	-----------------

由于实际电路的输出端只有低 15 位，故其真值应为：

$C_0 = 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0$

这一方法还可以用来测试乘法/累加电路的最高工作频率(亦即性能测试)：当时钟脉冲的频率较低时，芯片输出 $C = C_0$ ，芯片工作正常。将时钟脉冲的频率不断提高，每提高一次，进行一次测试，直至测试结果 $C \neq C_0$ ，说明电路的工作状态已发生紊乱，工作频率上限即可确定。

我们称这种方法为“准穷尽法”，是因为累加器输入端 Z 的累加信号并不是按照穷尽法的方法来提供测试码的。另外，因为测试只保留了一个最终数据 C，因此并不排除有几个中间错误互相抵销，却使最终结果 $C = C_0$ 。这样一种可能性，虽然这种可能性极小。

总之，这种具体问题具体分析，将复杂的测试问题化简的方法，是值得我们在实际工作中借鉴的。

§ 10-2 故障测试

故障测试与功能测试不同，它的主要目的在于解决失效电路的故障定位问题。帮助设计人员查出原因，改正错误，重新获得正确的结果。因此，它要求电路本身要有较强的可测性(可测性设计是 ASIC 设计师在电路设计初期就应考虑的问题)，提取出的测试矢量集要有较高的故障覆盖率，并有尽可能完备的“故障字典”等。

§ 10-2-1 故障字典

电路故障多种多样，人们建立了各种故障模型来研究这些故障。目前常用的、较有普遍意义且在理论上比较容易分析和处理的故障模型有“对电源短路模型”S-a-1(Stuck at 1) 和“对地短路模型”S-a-0(Stuck at 0)。由这两个模型发展起来的一套故障测试理论可以较好地诊断出由对电源短路或对地短路引起的电路失效问题。

图 10-3 是一个由一个“与非”门和一个“或非”门组成的简单电路，它有 5 个测点：A, B, C, P, D, 共有 10 种故障可能：A1(表示 A 点的 S-a-1 故障)，A0(表示 A 点的 S-a-0 故障，下同)，B1, B0, C1, C0, P1, P0, D1, D0。

不难分析，当图 10-3 电路无故障时，输入与输出之间应有真值表如表 4-1 第 1 - 4 列所示。当有各种 S-a-1 和 S-a-0 故障时，D 端的测试值将如表的后 10 列所示。该表即为图 10-3 电路的一个故障字典。

有了这张故障字典表，根据测量到的 D 值，即可分析电路中 S-a-1 故障和 S-a-0 故障所处的位置。

表 4-1 图 10-3 电路的故障字典

A	B	C	D	D(A0)	D(A1)	D(B0)	D(B1)	D(C0)	D(C1)	D(P0)	D(P1)	D(D0)	D(D1)
0	0	0	0	0	0	0	0	0	0	1	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	1	0	0	0	0	1	0	0	1
0	1	1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1	0	0	1	0	0	1
1	0	1	0	0	0	0	0	0	0	0	0	0	1
1	1	0	1	0	1	0	1	1	0	1	0	0	1
1	1	1	0	0	0	0	0	1	0	0	0	0	1

例如，当输入端 A B C 取值(0 0 0)时，即可测得 P0 故障或者 D1 故障；当取(0 1 0)时，可测得 A1 故障或者是 P0 故障或者是 D1 故障 …。(0 0 0)，(0 0 1)，(0 1 1)，(1 0 1)等称为测试向量。

有些故障是不可区分的，例如 A0, B0, C1, P1, D0 故障，无论用哪一组测试向量测试都无法将它们区分开来，这种故障称为“等价故障”，因为它们本身在电路中就是逻辑等价的。

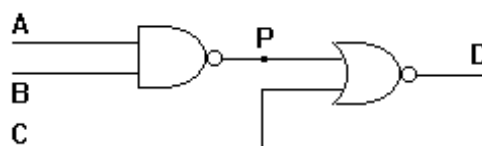


图 10-3 简单的组合电路

§ 10-2-2 测试向量集及其提取

进一步分析还可发现，要诊断图 10-3 电路中所有的 S-a-1 故障和 S-a-0 故障，并不需要对全部的输入向量(8 个)进行测试，而只需其中的 6 个就够了。它们是(0 0 0)，可测 P0, D1；(0 0 1)，可测 D1；(0 1 0)，可测 A1, P0, D1；(1 0 0)，可测 B1, P0, D1；(1

1 1) , 可测 C0, D1; (1 1 0) , 可测 A0, B0, C1, P1, D0。而 (0 1 1) 和 (1 0 1) 对测试故障没有特别贡献。因此对这一电路, 只要将 T(000, 001, 010, 100, 110, 111) 提取为测试向量集就可以了。

§ 10-2-3 故障覆盖率

故障覆盖率定义为: 用测试向量集可以测出的故障与电路中所有可能存在的故障之比。在图 10-3 电路中, 共有 10 个可能故障, 而上节给出的测试向量集 T(000, 001, 010, 100, 110, 111) 能够检测出这所有的 10 个故障, 因此 T 的故障覆盖率为 100%。而测试向量集 $T_1(011, 101)$ 仅能测出 D1 这一个故障, 故其故障覆盖率为 10%。 $T_2(110)$ 的故障覆盖率为 50%, 虽然它不能给等价故障 A0, B0, C1, P1, D0 精确定位。

§ 10-2-4 电路的可测性分析

电路测试的难易程度可由“测试因子”定量地描述。测试因子共有 6 个, 它们是:

CC0(X) -- 表示组合电路中 X 结点 S-a-0 故障的可控制性。

CC1(X) -- 表示组合电路中 X 结点 S-a-1 故障的可控制性。

C0(X) -- 表示组合电路中 X 结点故障的可观察性。

SC0(X) -- 表示时序电路中 X 结点 S-a-0 故障的可控制性。

SC1(X) -- 表示时序电路中 X 结点 S-a-1 故障的可控制性。

S0(X) -- 表示时序电路中 X 结点故障的可观察性。

所谓某结点某故障的可控制性, 是指人为地将该结点设置为该故障的难易程度。对于原始输入端 I 和原始输出端 O, 一般有如下约定:

$$CC0(I) = CC1(I) = 1$$

$$SC0(I) = SC1(I) = 0$$

$$C0(O) = S0(O) = 0$$

值越小表示越易观察和控制。

对于“与”、“或”、“非”等不同的电路, 求各点测试因子的公式是不相同的。例

如对于“与非”门 $Y = \overline{X_1 X_2}$ (图 10-4), 有如下一套公式:

$$CC1(Y) = \min[CC0(X_1), CC0(X_2)] + 1$$

$$CC0(Y) = CC1(X_1) + CC1(X_2) + 1$$

$$C0(X_1) = CC1(X_2) + C0(Y) + 1$$

$$C0(X_2) = CC1(X_1) + C0(Y) + 1$$

$$SC1(Y) = \min[SC0(X_1), SC0(X_2)]$$

$$SC0(Y) = SC1(X_1) + SC1(X_2)$$

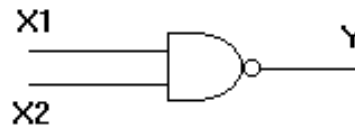


图 10-4 “与非”门 $Y = \overline{X_1 X_2}$

$$S0(X1) = SC1(X2) + S0(Y)$$

$$S0(X2) = SC1(X1) + S0(Y)$$

对于“或非”门 $Y = \overline{X1 + X2}$ (图 10-5), 公式如下:

$$CC1(Y) = CC0(X1) + CC0(X2) + 1$$

$$CC0(Y) = \min[CC1(X1), CC1(X2)] + 1$$

$$C0(X1) = CC0(X2) + C0(Y) + 1$$

$$C0(X2) = CC0(X1) + C0(Y) + 1$$

$$SC0(Y) = \min[SC1(X1), SC1(X2)]$$

$$SC1(Y) = SC0(X1) + SC0(X2)$$

$$S0(X1) = SC0(X2) + S0(Y)$$

$$S0(X2) = SC0(X1) + S0(Y)$$

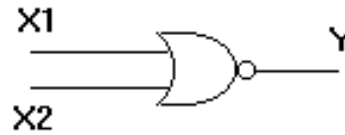


图 10-5 “或非”门 $Y = \overline{X1 + X2}$

因此, 图 10-3 所示电路, 其测试因子可按如

下方式计算, 并得结果示于表 10-2 中 (因为是组合电路, 故其与时序有关的测试因子值均为 0)。

$$CC0(A) = CC0(B) = CC0(C) = 1$$

$$CC1(A) = CC1(B) = CC1(C) = 1$$

$$C0(D) = 0$$

$$CC1(P) = \min[CC0(A), CC0(B)] + 1 = 1 + 1 = 2$$

$$CC0(P) = CC1(A) + CC1(B) + 1 = 1 + 1 + 1 = 3$$

$$CC1(D) = CC0(P) + CC0(C) + 1 = 3 + 1 + 1 = 5$$

$$CC0(D) = \min[CC1(P), CC1(C)] + 1 = 1 + 1 = 2$$

$$C0(P) = CC0(C) + C0(D) + 1 = 1 + 0 + 1 = 2$$

$$C0(C) = CC0(P) + C0(D) + 1 = 3 + 0 + 1 = 4$$

$$C0(B) = CC0(A) + C0(P) + 1 = 1 + 2 + 1 = 4$$

$$C0(A) = CC0(B) + C0(P) + 1 = 1 + 2 + 1 = 4$$

$$SC0(A) = SC0(B) = SC0(C) = SC0(P) = SC0(D) = 0$$

$$SC1(A) = SC1(B) = SC1(C) = SC1(P) = SC1(D) = 0$$

$$S0(A) = S0(B) = S0(C) = S0(P) = S0(D) = 0$$

表 10-2 图 10-3 电路的测试因子

结点	CC0	CC1	C0	SC0	SC1	S0
A	1	1	4	0	0	0
B	1	1	4	0	0	0
C	1	1	4	0	0	0
P	3	2	2	0	0	0
D	2	5	0	0	0	0

当电路某一结点的测试因子过大时, 可通过增加一个 I/O 端口的的方法使之降低。例如,

当某一结点 X 的可观察因子 $CO(X)$, $SO(X)$ 过大时, 可通过增加一输出端口的办法, 将该点直接引出; 当某点的可控制因子过大时, 可通过下述几种增加输入端的方法加以改善:

当 $CC0(X)$ 或 $SC0(X)$ 较大时, 可按图 10-6 的方法插入一个“与门”使之下降。这时:

$$CC0(X') = \min [CC0(X), CC0(I)] + 1 = 1 + 1 = 2$$

$$SC0(X') = \min [SC0(X), SC0(I)] = 0$$

当 $CC1(X)$ 或 $SC1(X)$ 较大时, 可按图 10-7 的方法插入一个“或门”使之下降。这时:

$$CC1(X') = \min [CC1(X), CC1(I)] + 1 = 1 + 1 = 2$$

$$SC1(X') = \min [SC1(X), SC1(I)] = 0$$

在某些特殊情况下, 还可直接将该 X 点与增加的输入端 I 相连。如图 10-8 所示。

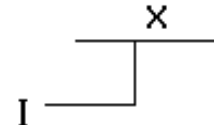
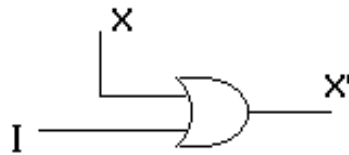
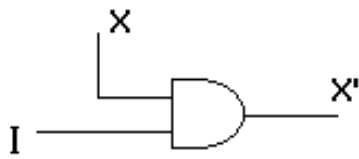


图 10-6 插入“与门”的方法

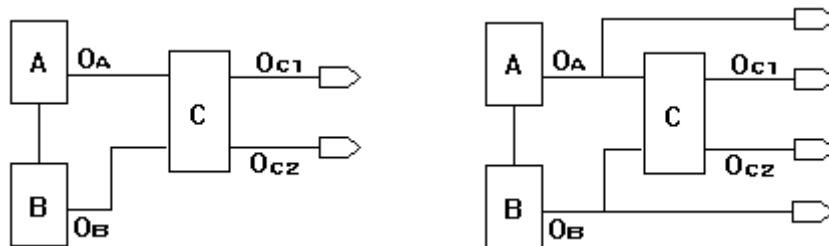
图 10-7 插入“或门”的方法

图 10-8 直接相连的方法

§ 10-3 电路的可测性设计

专用集成电路的测试虽然是在电路制造出来以后进行的, 但专用集成电路的可测性设计却是要在早期的电路设计阶段就给以考虑的。下面的两个例子, 就给出了专用电路可测性设计方面的一些初步的考虑方法。

1. 设图 10-9(a) 为某一专用集成电路的功能框图。从图中可见, 该电路是由 A、B、C 三个子电路组成的。电路的原始输出端为 O_{c1} 和 O_{c2} 。若芯片制作完成后, 测试 O_{c1} 和 O_{c2} , 发现它们的测试值与原设计的真值不符, 则至少说明有下面三种情况中的一种发生了: (1)子电路 A 出错; (2)子电路 B 出错; (3)子电路 C 出错。



(a)

(b)

图 10-9 子电路框图及测试管脚

排查的方法，自然是希望测试 O_A 与 O_B 的值，以确定错误究竟发生在什么地方。图 10-9(b) 即是根据这一思想而额外增加了两个测试管脚 O_A 与 O_B 的示意图。

增加测试管脚并不是实用的好方法。测试管脚对电路的正常工作毫无用处，其压焊块的存在增大了版芯面积。尤其是对那些 I/O 管脚数已经很多或子电路个数很多的电路，这样的方法更是不能接受。

为寻求对这一问题的妥善解决，基本思路应该是寻求一种具有某种特殊功能的简单电路，借助这种电路，既可在不增加额外 I/O 管脚的情况下，通过外部测试，确定芯片内部的故障所在，又由于该电路形式简单，不致使原电路的规模或复杂性受到显著影响。

双路选择器(二选一电路)即是可以满足这种需要的理想电路, 见图 10-10。

图 10-10 中, 当二选一电路的控制端 T 接高电平时, O_1 , O_2 给出的是内部测点 O_A , O_B 的值; 当控制端 T 接低电平时, O_1 , O_2 即为本来的原始输出值 O_{C1} , O_{C2} 。这样, 无论芯片内部需要增加多少测点, 只要不超过电路的原始输出管脚数, 都可以用这样的方法在外部得到内部子电路的测试值, 而整个电路外部只需增加一个统一的测试控制管脚 T。当电

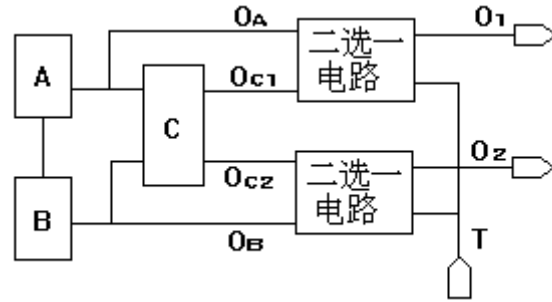


图 10-10 用二选一电路对芯片进行测试

路最终通过测试，交付正常使用时，只需将该脚 T 接地(低电平)即可。

根据前面章节的介绍，已知二选一电路的电路形式非常简单，只有四只 MOS 管，因此增加这样的电路，对整个芯片的规模或面积的影响确实是微乎其微的。

2. 不但将多路选择器用在芯片电路的输出端，而且将它们用在芯片的内部，利用它们把内部的子电路分割开来并彼此绝缘，单独测试，是大规模、超大规模集成电路可测性设计的更常用的方法。见图 10-11。

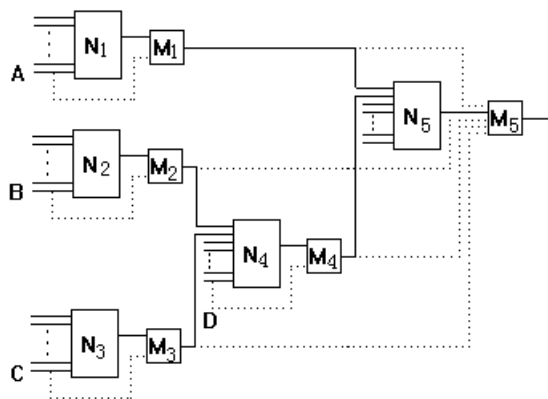


图 10-11 用多路选择器划分电路

图 10-11 中, $N_1 - N_5$ 为 5 个子电路, $M_1 - M_5$ 为 5 个多路选择器。每一个子电路的后继分别插入一个多路选择器, 将前级与后级子电路隔离开来。为了测试一个多路选择器的后继子电路, 将该多路选择器直接与前驱子电路的一个原始输入端相连(即图中的 M_1, M_2, M_3, M_4 分别与 A、B、C、D 相连)。通过这些原始输入端, 直接向多路选择器的后继子电路设置测试码。每一个多路选择器的输出, 与最后一个多路选择器(即原始输出端的 M_5) 的输入相连。根据这样的结构, 当要对子电路 N_i 进行测试时, 可让 M_1 和 M_5 分别选择 N_i 和 M_i ; 当要对子电路 N_5 进行测试时, 可让 M_1, M_4 和 M_5 分别选择 A、D 和 N_5 。5 次顺序测试, 即可测完图 10-11 中的所有子电路。