

一种有效的片上系统测试数据压缩算法

方建平, 郝跃

(西安电子科技大学 宽禁带半导体材料与器件教育部重点实验室 陕西 西安 710071)

摘要: 测试数据的规模和容量直接影响了片上系统的测试成本,故提出了一种测试数据编码的压缩算法——Min_Comp. 该方法采用不等间距的编码方式,根据测试数据中游程长度的统计分布情况来调整各组数据的大小,从而提高测试数据的压缩率,降低了测试成本. 为了使编码算法对应的解码电路的硬件开销最小化,该算法还引入了前后缀标识位的概念,这样可减小解码电路的规模和复杂度. 对 ISCAS89 benchmark 电路的实验结果表明,采用 Min_Comp 编码方式的压缩效率要比 Golomb 等编码方法好,而且实现方式简单.

关键词: 测试数据压缩; 哈夫曼编码; Golomb 编码; Min_Comp 编码

中图分类号: TP391.76 **文献标识码:** A **文章编号:** 1001-240X(2006)01-0001-04

Study of an efficient SOC test vector compression scheme

FANG Jian-ping, HAO Yue

(Ministry of Edu. Key Lab. of Wide Band-gap Semiconductor Materials and Devices,
Xidian Univ., Xi'an 710071, China)

Abstract: Test volume is an important factor affecting the cost of the SOC Testing. In order to reduce the cost of SOC Testing, the paper proposes a method for compression/decompression of test data, that is the Min_Comp. Based on the analysis of the different sizes of the run-length coding, the method determines the grouping of test data and it can improve the compression ratio and reduce the test cost. Furthermore, by introducing the prefix and tail flag, the decoder leads to lower hardware overhead compared with those in the references. Experiments were performed on the ISCAS 89 benchmark circuits, and the results show that the Min_Comp code has a better compression ratio than the Golomb code. Also, the decompression circuit has a low area overhead and is easy to design.

Key Words: test vector compression; Huffman code; Golomb code; Min_Comp code

由于现在电路规模越来越大,用于电路测试的测试数据规模不断增加,特别是基于 IP 核的片上系统(SOC)的测试问题^[1]. 不断攀升的测试数据增加了测试时间,提高了测试成本. 现在可行的缩短测试时间的方法包括:压缩测试数据和内建自测试法(BIST). BIST 方法的基本思想就是把测试向量的产生模块和测试响应的分析模块移入到待测芯片内部,从而减少测试电路对测试仪器的依赖性. 但目前为止,BIST 法只在电路结构比较规整的存储单元测试中被广泛应用.

压缩测试数据的方法可很好地解决上述难题. 这种方法的思路是:首先将自动测试向量生成程序(ATPG 工具)产生的测试向量,用一定的压缩算法将其压缩变小后放在测试设备的存储设备中. 当要进行芯片测试操作时,测试设备就将存放在自己存储设备内已经压缩过的测试向量送给待测芯片内相应的解码电路,由解码电路得到实际使用的测试向量,然后再将其加到测试芯片的输入引脚上完成测试. 实验表明,通过压缩算

法节省的数据容量非常可观.

笔者提出一种新数据压缩/解压缩方法.介绍了与压缩算法相对应的解压缩电路的硬件实现.用 ISCAS89 的 benchmark 电路进行实验,采用的测试向量是由 Duke 大学的 K. Chakrabarty 教授、Li Lei 博士生以及中科院计算所的韩银和博士提供的测试向量.

1 压缩算法

1.1 压缩算法的要求和分类

对于测试数据压缩算法一般有以下基本要求 (1)高压缩率 这是对任何压缩算法最基本的要求 (2)高压缩稳定性:一种测试数据压缩算法必须对各类测试数据在压缩率方面有比较好的一致性 (3)低复杂度的解码单元 对测试数据进行压缩的主要目的是为了降低成本 因此不允许其带来过于高昂的硬件代价 (4)解码单元的可扩展性:可测试性设备的可扩展性是确定其适用范围的重要因素之一,不同规模的电路对于压缩算法和译码单元要求是不同的,只有具备了良好的可扩展性才能降低设计成本 (5)算法对各类电路的普遍适用性 通常测试结构的设计与逻辑设计是相互独立的 因此压缩算法的实用化很大程度上取胜决于该算法是否有比较广的适用范围.

另外,测试数据压缩算法可从数学基础、压缩对象等方面分为不同的类型.在进行数据压缩时,首先需要对原始数据进行适当的划分,这样的划分可以是定长的,也可以是变长的.对于划分得到的字段,需要根据算法来分配相应的代码字,从而实现数据压缩.代码字也可分为定长和变长.因此根据编码方式的特征可将压缩算法分为4类,如表1.

表1 压缩算法分类

原始数据段	代码字	典型压缩算法	原始数据段	代码字	典型压缩算法
定长	定长	压缩效率最低	变长	定长	传统游程编码
定长	变长	哈夫曼编码	变长	变长	Golomb 编码

哈夫曼编码是一种基于统计的编码方式,该方式可提供最好的定长—变长的编码效果.可通过构造哈夫曼树来实现哈夫曼编码,但它是一种动态编码,这就增加了测试设备和解码电路之间实际传输的数据量^[2-4].Golomb 编码是在传统的游程编码基础上发展起来的一种改进算法^[5],其最大的改进之处在于改变了原来从变长到定长的编码方式,而选择了变长—变长的方案^[6].代码字分为前缀和后缀两部分,前缀代表了分组信息,而后缀则是位串长度相对信息,见表2.

1.2 Min_Comp 压缩算法

表2 $m=4$ 的 Golomb 编码

分组	游程	前缀	后缀	代码字
A_1	0	0	00	000
	1		01	001
	2		10	010
	3		11	011
A_2	4	10	00	1000
	5		01	1001
	6		10	1010
	7		11	1011
A_3	8	110	00	11000
	9		01	11001
	10		10	11010
	11		11	11011
...

表3 $p=2, t=1$ 的 Min_Comp 编码表

分组	游程	前缀	后缀	代码字
A_1	0	无	00	00
	1		01	10
	2	100	00	10000
A_2	3		01	10001
	4	101	00	10100
	5		01	10101
	6	110	00	11000
A_3	7		01	11001
	8	111	00	11100
	9		01	11101
	10	100100	00	10010000
	11		01	10010001
	12	100101	00	10010100
	13		01	10010101
...

Min_Comp 编码是一种变长—变长编码方式,在编码方式上和 Golomb 有相似之处,但与 Golomb 编码有明显的不同。(1)从压缩效率方面来考虑,Min_Comp 编码中分组的策略更加灵活,它是一种不等间距的编码方式,不等间距的编码方式是指在进行分组时每个组的大小根据其出现频率进行了适当的调整,表 3 给出了 Min_Comp 算法($p = \alpha$ 除去标识位,前缀位数) $l = l$ (除去标识位,后缀位数)的编码表,可看到从 A_1 到 A_3 ,每个分组的容量是依次递增的。(2)从解码电路的硬件开销来考虑,前缀和后缀分别有一个标识位,这个标识位从硬件实现上很方便。

可从概率理论来进行算法压缩效率的分析,假设没有压缩前的数据位为 T_D ,压缩以后测试数据的位数为 T_E ,则可用压缩倍数 D_M 来衡量压缩的效率 $D_M = T_D/T_E$ 。对于某个具体电路,其测试数据具有很强的随意性,所以,用概率的办法来统计一组测试数据中“1”和“0”出现的频率,首先做如下的假设。

假设 1 测试数据中某一位出现“0”的概率为 p ,则出现“1”的概率为 $(1 - p)$ ($0 \leq p \leq 1$)。

假设 2 测试数据中游程长度为 l 的位串出现在第 k 组中的概率为 $\mu(l, k)$ 。

根据假设 1 和游程长度的定义,可推算出测试向量在未压缩前的位数 T_D ,根据假设 2,以及各组代码字的长度定义可算出 T_E :

$$T_D = 1 + \sum_{l=1}^{\infty} l \times p^l(1 - p) = 1/(1 - p) \quad , \quad T_E = \sum_{k=1}^{\infty} \mu(l, k) \times L_k \quad , \quad (1)$$

其中 L_k 为 k 组序列中代码字的长度(同一组数据对应的代码字都是等长的),

$$L_k = (3k - 1) \quad ,$$

$$\mu(l, k) = \sum_{l=(2^{k-2})^{1/3}}^{(2^{k+1}-5)^{1/3}} p^l(1 - p) = \frac{[p^{(2^{k-2})^{1/3}}(1 - p)] [1 - p^{(2^{k-1})^{1/3}}]}{1 - p} = p^{(2^{k-2})^{1/3}}(1 - p^{(2^{k-1})^{1/3}}) \quad ,$$

$$D_M = \frac{T_D}{T_E} = \frac{1}{(1 - p) \times \sum_{k=1}^{\infty} (3k - 1) \times p^{(2^{k-2})^{1/3}}(1 - p^{(2^{k-1})^{1/3}})}$$

用同样的方法,可求出 Golomb 编码的压缩倍数为

$$D_G = 1 / [(1 - p) \log_2 m + 1/(1 - p^m)] \quad . \quad (2)$$

对两种算法的压缩率进行比较,得出相对的压缩比例系数 α 为

$$\alpha = \frac{D_M}{D_G} = \frac{\log_2 m + 1/(1 - p^m)}{\sum_{k=1}^{\infty} (3k - 1) \times p^{(2^{k-2})^{1/3}}(1 - p^{(2^{k-1})^{1/3}})} \quad . \quad (3)$$

从式(3)可看出,由于分母呈指数函数,所以随着 p 的增大, α 越来越小,也就是说 Min_Comp 压缩算法的压缩效果比 Golomb 编码算法要好。

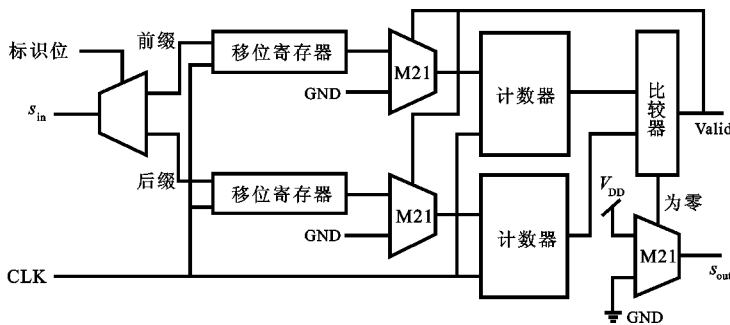


图 1 Min_Comp 解码电路结构示意图

2 解码电路

为了测试而引入的硬件开销是可测性设计方法学中的一个重要参数,所以在研究算法的时候应充分考

虑到这一点,从而使相应的硬件开销降到最小.笔者采用的办法是引入前后缀标识位,也就是在进行压缩编码的前后缀中专门选择其第一位为标识位,这样在解码电路硬件实现时可简化判断电路的规模,虽然这样做是以牺牲一定的压缩率为代价.具体的解码电路结构如图 1 所示.

解码电路的操作流程为 (1)自动测试设备(ATE)一旦接收到被测电路的 ready 信号,则 ATE 设备在下一个时钟周期开始向被测电路输出压缩的数据 T_E . (2)解码电路在接收到测试数据时,首先根据数据的标识位判断是前缀还是后缀. (3)如果是前缀,把相应的组信息放入前缀移位寄存器. (4)如果是后缀,把相应的偏移量信息放入后缀移位寄存器. (5)如果 Tail_phase 信号为低,则把前缀/后缀移位寄存器内的值并行放入相应的计数器. (6)前缀计数器开始减 1 操作,减到 0 时启动后缀计数器进行减 1 操作(此时只要两个计数器的值不为 0,则在 s_{out} 端输出 0,并置位 valid 信号). (7)如果两个计数器的值都为 0,则使比较器的输出为 1,这样就使 s_{out} 输出一个 1,并且使 valid 信号为 0. Min_Comp 算法解码电路工作时序参见图 2.

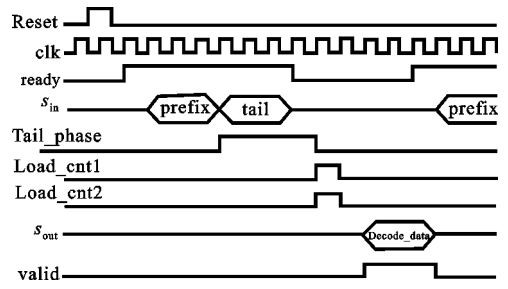


图 2 Min_Comp 算法解码电路工作时序图

3 实验结果

为了验证 Min_Comp 编码算法的有效性,针对 ISCAS89 中几个比较大的时序电路,采用美国 Duke 大学提供的 MinTest 测试向量集进行了实验,并与文献 [6] 中 Golomb 编码算法的实验数据进行比较,可看出,笔者提出的编码算法具有很好的压缩效率.具体实验结果如表 4 所示.

表 4 使用 MinTest 测试数据的实验结果

电路	原始数据位数	Golomb ^[6]		Min_Comp 编码
		m	编码位数	编码位数
S510f	23 754	4	14 085	11 206
S641f	39 273	4	22 250	19 433
S713f	165 200	16	41 658	30 261
S838f	76 986	4	40 717	28 423
S953f	28 208			24 012
S5378f	164 736	4	92 054	71 098
S9234	199 104	4	104 111	83 511

同样,如果对其他的 ATPG 工具产生的测试数据进行压缩,Min_Comp 编码的压缩率也好于 Golomb 编码的压缩率.表 5 是对 Atalanta 工具产生的测试向量进行压缩的实验结果.

表 5 使用 Atalanta 测试数据的实验结果

电路	原始数据位数	Golomb ^[6]		Min_Comp 编码
		m	编码位数	编码位数
S510f	1 500	4	1 038	826
S641f	4 806	4	2 074	1 743
S713f	4 752	4	2 056	1 699
S838f	12 596	4	5 054	4 012
S953f	5 445	4	2 389	1 975
S5378f	122 836	4	39 665	21 231
S9234	172 406	4	62 234	48 702

从表 4 和表 5 可看出,笔者提出的编码算法具有很好的压缩效率,并且具有很好的压缩稳定性,它适合于不同 ATPG 工具产生的测试向量.

