

基于可重构 MUXs 网络的低功耗测试数据压缩方法

刘 军^{1,2,3}, 吴 玺¹, 韩银和^{2,3}, 李晓维^{2,3}

(1. 合肥工业大学计算机与信息学院, 安徽合肥 230009; 2. 中国科学院计算技术研究所系统结构重点实验室, 北京 100190;
3. 中国科学院研究生院, 北京 100049)

摘 要: 测试数据和测试功耗是集成电路测试时关注的两个主要问题. 为缩减测试数据体积和降低测试功耗, 提出了一种基于可重构 MUXs 网络的低功耗测试数据压缩方法. 这种方法在保持压缩率不变的前提下, 充分利用 MUXs 网络中“空闲”的测试通道来降低测试功耗. 在降低测试功耗原则的指导下, 将一些“有用”的测试通道进行拆分, 即将这些“有用”通道驱动的一部分扫描链改由“空闲”的通道来驱动. 提出了怎样选择通道, 怎样将选择的通道进行拆分方法. 实验结果表明建议的方法有效降低了测试时的平均功耗和峰值功耗.

关键词: 低功耗; 测试数据压缩; 可重构; MUXs 网络

中图分类号: TP306+.2 **文献标识码:** A **文章编号:** 0372-2112 (2011) 05-1190-04

Low Power Test Data Compression Technique Based on Reconfigurable MUXs Network

LIU Jun^{1,2,3}, WU Xi¹, HAN Yin-he^{2,3}, LI Xiao-wei^{2,3}

(1. School of Computer and Information, Hefei University of Technology, Hefei, Anhui 230009, China;
2. Key Laboratory of Computer System and Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China;
3. Graduate University of Chinese Academy of Sciences, Beijing 100490, China)

Abstract: Test data and test power are the two major concerns during integrated circuits test. The proposed technique, low power test data compression technique for reconfigurable MUXs network, can effectively reduce test data and test power. The technique made full use of “idle” test channels to reduce test power without compression ratio loss. Under the guide of minimizing power consumption, some “useful” test channels are split, which means some scan chains driven by “useful” channels are adjusted to be driven by “idle” channels. The method, how to select channels and how to split the selected channels, was illustrated. Experimental results show the effectiveness of the proposed technique in reducing average power and peak power.

Key words: low power; test data compression; reconfigurable; MUXs network

1 引言

测试数据和测试功耗是集成电路测试时需要关注的两个主要问题. 为减少测试数据和降低测试功耗, 学者们提出了各种测试压缩方法^[1~7] 和低功耗测试技术^[8~10].

然而若同时考虑测试数据和测试功耗的协同优化, 就会发现一些好的测试压缩方法并不能有效的降低测试功耗, 而一些降低测试功耗的方法会造成测试数据压缩率很低. 针对这个问题, 学者们提出了能够同时缩减测试数据和降低测试功耗的方法^[11~14]. 如通过改变扫描链结构的“侵入式”方法^[11] 和在压缩的过程中考虑降低功耗的“非侵入式”方法^[12~14].

本文提出的基于可重构 MUXs 网络的低功耗测试压缩方法与文献[7]的方法保持相同的压缩率, 但能够

有效的降低测试功耗. 即在使用文献[7]的方法将测试数据进行压缩后, 再应用本文提出的方法可以使测试功耗大幅降低.

2 可重构的 MUXs 网络压缩方法

为了更好的理解本文的方法, 这部分首先介绍可重构的 MUXs 网络压缩方法^[7]. MUXs 网络压缩是将测试向量分段进行相容压缩的方法. 图 1 表示要移入电路的测试数据, 其中每条扫描链被分成 A、B 两部分. MUXs 网络压缩首先对 A 部分的数据(深颜色标记部分)进行相容合并, 这时需要四个外部测试通道, 即 $1 \rightarrow \{2, 3, 6\}$, $2 \rightarrow \{5, 8\}$, $3 \rightarrow \{7\}$, $4 \rightarrow \{1, 4\}$, 才能将测试数据移入扫描链. 而对 B 部分的数据仅需二个通道, 即 $1 \rightarrow \{3, 4, 6, 7\}$, $2 \rightarrow \{1, 2, 5, 8\}$. 其中, $A \rightarrow \{B_1, B_2, \dots, B_n\}$ 指的是第 A_n 测试通道驱动第 B_1, B_2, \dots, B_n 条扫描链.

解压时将 A 部分数据移入扫描链后,需重新配置 MUXs 网络将 B 部分数据移入.由于每种配置的 MUXs 网络持续 5 个时钟周期,因此称这个 MUXs 网络的测试周期为 5.

在图 1 例子中, A 部分的数据需要四个通道.但 B 部分数据仅使用 1,2 两个通道,本文称这两个通道为“有用”的通道,3,4 两个通道为“空闲”通道.虽然在移入 B 部分数据时,3,4 两个通道未使用,但并不代表这两条通道在这个测试周期内没有数据.实际上,这两条通道也是传递数据的,只是这些数据并没有被选中移入到扫描链,可以存储成任意值.

由于文献[7]的方案存在着大量的“空闲”通道,造成了很大的浪费.本文方案就是充分利用这些“空闲”的通道,将一部分扫描链重新分配到这些通道上,将这些“空闲”的通道变成“有用”的通道,从而达到降低功耗的目的并保持测试压缩率不变.

	A				B			
扫描链1	1	X	1	X	0	X	X	1
扫描链2	X	X	0	X	1	X	X	0
扫描链3	X	X	X	X	1	1	X	0
扫描链4	1	1	X	X	0	X	X	1
扫描链5	0	X	1	X	X	X	1	X
扫描链6	X	0	X	1	X	X	0	1
扫描链7	0	X	0	X	X	1	0	X
扫描链8	X	X	1	X	X	0	1	X

图1 MUXs网络压缩的测试向量

扫描链1	0	1	0	1	1	0	1	0	1	1
扫描链2	0	1	0	1	1	0	1	0	1	1
扫描链3	1	0	1	0	X	1	0	1	0	0
扫描链4	1	0	1	0	X	1	0	1	0	0
扫描链5	0	1	0	1	1	0	1	0	1	1
扫描链6	1	0	1	0	X	1	0	1	0	0
扫描链7	1	0	1	0	X	1	0	1	0	0
扫描链8	0	1	0	1	1	0	1	0	1	1

(a) 扫描链相容结果

(b) X位填充结果

跳变次数 72

图2 文献[7]相容和填充结果

扫描链1	0	1	X	1	1	0	1	1	1	1
扫描链2	X	1	0	X	1	1	1	0	0	1
扫描链3	1	X	1	0	X	1	1	1	0	0
扫描链4	1	X	1	0	X	1	1	1	0	0
扫描链5	X	1	0	X	1	1	1	0	0	1
扫描链6	1	0	1	X	X	1	0	1	1	1
扫描链7	1	0	1	X	X	1	0	1	1	1
扫描链8	0	1	X	1	1	0	1	1	1	1

(a) 建议方案扫描链相容结果

(b) X位填充结果

跳变次数 34

图3 建议方案相容和填充结果

从这个例子可看出,本文要解决的问题和思路可描述如下:对含有“空闲”通道的测试周期,将一个“有用”测试通道驱动的扫描链取出一部分改由“空闲”的测试通道来驱动,将“空闲”的测试通道变为“有用”的通道.上述过程重复进行,直到这个测试周期不存在“空闲”的测试通道为止.

因此本方案中需要解决两个关键问题:第一是选择哪一个通道进行拆分,第二是怎样对选中的通道进行拆分.下面具体介绍本文解决这两个问题的方法.

3.2 选择通道

本方案定义一个度量值 PW 来选择通道.通道的 PW 值就是对这个通道驱动的扫描链中的测试向量分别进行相容合并和 MT-filling 填充所产生跳变的差值.本方案选择具有最大 PW 值的通道进行拆分.

图 4 给出了 PW 值的计算例子.对图 1 中的 B 部分数据,按照文献[7]的方法需要两个通道来驱动,即 $1 \rightarrow \{3, 4, 6, 7\}$, $2 \rightarrow \{1, 2, 5, 8\}$.

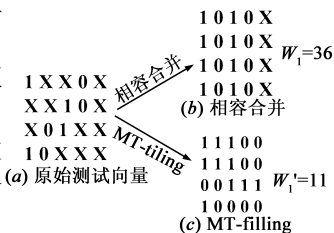


图4 PW 值的计算

3 建议方案

3.1 建议方案的思路

这部分首先通过图 1 中 B 部分数据来说明将“空闲”的测试通道变为“有用”的测试通道可以有效的降低测试功耗,并通过这个例子总结本方案解决问题的思路.

对图 1 中 B 部分数据,图 2(a)是按照文献[7]的方法在 $1 \rightarrow \{3, 4, 6, 7\}$, $2 \rightarrow \{1, 2, 5, 8\}$ 的配置下扫描链中的数据,图 2(b)是对图 2(a)中的 X 位进行 MT-filling^[9]结果.图 3(a)是建议方案在 $1 \rightarrow \{3, 4\}$, $2 \rightarrow \{1, 8\}$, $3 \rightarrow \{6, 7\}$, $4 \rightarrow \{2, 5\}$ 的 MUXs 网络配置下扫描链相容的结果,图 3(b)是对图 3(a)中的 X 位进行 MT-filling 的填充结果.

从图 2 和图 3 中可看出,采用文献[7]的方法,测试向量移入扫描链将会发生 72 次跳变.而采用本文建议的方案,仅发生 34 次跳变.

第一个通道驱动的扫描链最初的测试数据如图 4(a)所示.这些扫描链相容合并后的值如图 4(b)所示,将图 4(b)中的 X 位进行 MT-filling 后,可以得到这四条扫描链在移入时会发生 36 次跳变,记为 W_1 .若对每条扫描链中原始的测试向量分别进行 MT-filling,则结果如图 4(c)所示,这种情况下,仅发生 11 次跳变,记为 W_1' .本文就定义 $PW_1 = W_1 - W_1' = 25$.同理,可以计算出第二条通道的 $W_2 = 36$, $W_2' = 6$, $PW_2 = 30$.因为 $PW_2 > PW_1$,所以首先选择第二条通道进行拆分.

3.3 拆分通道

当选中通道后,下一步是将这个通道拆分为 2 个通道.假设这个通道驱动 m 条扫描链,其中原始的测试向量是 S_1, S_2, \dots, S_m .这些测试向量相容合并后可得到一个向量 U .

假设 U 的第 i 位和 $i+1$ 位存在一个 $0 \rightarrow 1$ 跳变,因为 U 是由 S_1, S_2, \dots, S_m 相容合并而成,其中 $S_j (1 \leq j \leq m)$ 的第 i 位和 $i+1$ 位必定是下面四种情况之一: A.01

B.X1 C.0X D.XX.本方案构建四个集合,即 Set_A , Set_B , Set_C , Set_D ,分别对应 A, B, C, D 这四种情况.这四个集合中的测试向量各有自己的特点. Set_A

集合中的测试向量第 i 位和第 $i + 1$ 位一定存在跳变. 而 Set_C 和 Set_D 中的测试向量第 i 位和第 $i + 1$ 位一定不存在跳变. 但 Set_B 中的测试向量第 i 位和第 $i + 1$ 位是否存在跳变, 是不能确定的. 例如测试向量 0XX1, 对它进行 MT-filling 的结果是 0001, 第 3 位和第 4 位存在跳变, 而对 1XX1 这个测试向量进行 MT-filling, 第 3 位和第 4 位就不存在跳变. 所以需要将 Set_B 中的测试向量再进一步细分成两个集合, Set_B_1 和 Set_B_2 . Set_B_1 表示第 i 位和第 $i + 1$ 位存在跳变, Set_B_2 表示第 i 位和第 $i + 1$ 位不存在跳变.

因此, 最终形成了五个集合, 问题是怎样将这五个集合重新组合, 最终形成两个集合, 使得这两个集合的跳变总数最小. 最终形成的两个集合就代表两个通道.

本文的思路: 是形成两个集合, 一个集合中的向量第 i 位和第 $i + 1$ 位存在跳变, 另一个集合中的元素第 i 位和第 $i + 1$ 位不存在跳变. 这样就可以缩减移位过程中的跳变数量. 本文建议方案分三步完成对这五个集合的重组. 假设重组后形成的两个集合是 $Chan_1$ 和 $Chan_2$, 最初这两个集合是空集. 本方案的三个步骤如下:

(1) 令 $Chan_1 = \{Set_B_2\}$, $Chan_2 = \{Set_C\}$. 之所以将 Set_B_2 和 Set_C 分在两个通道中, 是因为这两个集合中的元素第 i 位和第 $i + 1$ 位不存在跳变, 但若 Set_B_2 和 Set_C 放入一个通道中进行相容合并, 则会造成第 i 位和第 $i + 1$ 位的跳变, 因此将这两个集合分开, 有助于降低功耗. 设这步之后 $Chan_1$ 和 $Chan_2$ 中元素的数量分别为 num_1 和 num_2 .

(2) 确定 Set_A , Set_B_1 放入哪个通道. 因为这两个集合中的元素第 i 位和第 $i + 1$ 位存在跳变, 若将这两个集合分开放入两个通道中, 会造成两个通道的第 i 位和第 $i + 1$ 位都存在跳变. 因此这两个集合要放入同一个通道中. 本方案根据第一步得出 $Chan_1$ 和 $Chan_2$ 中元素的数量来确定这两个集合应该放入哪个通道. 如果 $num_1 > num_2$, 则 Set_A 和 Set_B_1 放入通道 $Chan_2$ 中, 否则将这两个集合放入 $Chan_1$ 中.

(3) 将 Set_D 放入不含 Set_A 和 Set_B_1 的那个通道中. 因为 Set_D 中的元素不存在跳变, 若放入含有 Set_A 和 Set_B_1 的通道中, 相容合并的结果会使 Set_D 中的元素也存在跳变而使功耗增大. 因此 Set_D 要放入不存在跳变的那个通道中. 但是若 Set_A 和 Set_B_1 都为空集, 则 $Chan_1$ 和 $Chan_2$ 都不存在跳变, 建议方案中将 Set_D 并入 $Chan_1$ 和 $Chan_2$ 中元素少的那个通道.

如图 4(a) 中的元素相容合并后的向量为 $U = 1010X$, 根据第 2 位和第 3 位之间的 0→1 跳变按照以上步骤进行拆分, 最终将形成的集合为:

$$\begin{aligned} Chan_1 &= \{(XX10X), (1XX0X)\} \\ Chan_2 &= \{(10XXX), (X01XX)\} \end{aligned}$$

上面是根据相容合并后向量 U 的第 i 位和第 $i + 1$ 位的跳变, 将一个测试通道拆成两个通道, 一个通道的第 i 位和第 $i + 1$ 位存在跳变, 而另一个不存在跳变. 若 U 中有多个数据位存在跳变, 本方案是针对每一个跳变都执行上述过程, 然后计算出根据哪一个跳变进行拆分产生的跳变最小, 就选择这种拆分方法.

以上的方法是消除第 i 位和第 $i + 1$ 位的 0→1 的跳变, 若第 i 位和第 $i + 1$ 位是 1→0 的跳变, 拆分的方法是一样的, 只是要根据第 i 位和第 $i + 1$ 位的下面四种情况将测试向量分类: A: 10 B: X0 C: 1X D: XX. 同样的也要将 B 中的元素再分成两个集合 Set_B_1 和 Set_B_2 . 下面的步骤与拆分 0→1 的跳变相同.

4 实验结果

本部份对 ISCAS'89 中最大的几个电路进行了实验, 表 1 中数据是建议方案与文献[7]相比降低的功耗和增加的硬件开销的百分比. 因为本方案的压缩率与文献[7]相同, 就没有给出电路的压缩率. 表中的功耗值是用测试向量的 WSA(weighted switching activity)值^[12]来替代的. 表 1 中第二列是扫描链的数量. 从表中可看出, 周期数为 4, 5 时平均功耗分别减少了 46.17% 和 49.57%, 峰值功耗分别降低了 33.48%, 34.95%. 在表 1 的电路中, s35932 的平均功耗缩减的幅度最小, 主要是因为 s35932 的测试向量数目少, 未能充分发挥本方案的优越性.

表 1 建议方案与文献[7]的功耗和硬件开销比较

电路	扫描链	周期数为 4			周期数为 5		
		平均功耗 耗缩减 (%)	峰值功 耗缩减 (%)	硬件 增加 (%)	平均功 耗缩减 (%)	峰值功 耗缩减 (%)	硬件 增加 (%)
s13207	16	43.67	33.48	33.3	46.47	24.66	38.5
	32	60.13	38.91	42.3	63.68	34.85	43.3
	64	76.27	44.48	43.9	79.33	52.74	48.5
s15850	16	57.82	33.64	33.3	61.69	25.57	30.8
	32	63.35	39.62	30.8	66.11	47.39	33.3
	64	63.94	34.84	31.3	69.66	33.80	36.1
s35932	16	16.36	46.99	7.14	17.61	42.20	5.88
	32	3.88	8.75	3.12	8.04	15.59	5.41
	64	24.44	41.20	7.55	28.22	60.38	2.99
s38584	16	53.94	14.42	55.6	59.60	22.96	50.0
	32	61.08	27.30	25.0	64.49	29.31	25.8
	64	62.70	35.19	18.4	67.13	39.98	29.2
s38417	16	41.22	27.33	20.0	45.93	31.60	18.2
	32	34.16	26.64	8.33	38.94	34.76	12.1
	64	29.54	49.46	12.0	26.63	28.47	14.2
均值		46.17	33.48	24.80	49.57	34.95	26.29

本方案充分利用了测试通道, 与文献[7]相比将会增加 MUX 的数量. 本文使用商业工具对建议方案和文献[7]的 MUXs 网络解压结构进行了综合, 其硬件开销的比较情况在表 1 中. 与文献[7]相比, 在周期数为 4 和 5 时, 硬件面积开销平均增加了 24.8% 和 26.29%.

本方案会增加 MUXs 网络的配置数目,从而有可能增加测试时间.实验表明,本方案仅造成了少部分电路的测试时间增加,测试时间的增加情况如表 2 所示.表 2 中给出了测试时间有变化的电路及其扫描链的数量,未列出的表明本方案与文献[7]有相同的测试时间.

表 2 建议方案与文献[7]的测试时间比较

周期数为 4			周期数为 5		
电路	扫描链	时间增加(%)	电路	扫描链	时间增加(%)
s13207	16	19.6	s13207	16	19.5
s13207	32	19.3	s15850	16	24.0
s15850	32	23.6	s15850	64	28.3

5 结论

本文发现在 MUXs 网络压缩中存在着大量的“空闲”通道,为充分利用这些“空闲”的测试通道,提出了一种基于可重构 MUXs 网络的低功耗测试数据压缩方法.这种方法将“有用”测试通道驱动的扫描链进行拆分,一部份扫描链仍由原来的测试通道驱动,其余的扫描链由“空闲”的测试通道来驱动,将“空闲”的测试通道变成“有用”的通道.提出了怎样选择拆分的通道以及怎样将选中的通道进行拆分,从而有效的降低了移位过程中的测试功耗,并保持测试压缩率不变.

参考文献

[1] N A Toub. Survey of test vector compression techniques[J]. IEEE Design & Test of Computers, 2006, 23(4): 294 – 303.

[2] Y Han, Y Hu, et al. Rapid and energy-efficient testing for embedded cores[A]. IEEE Asian Test Symposium[C]. Washington DC: IEEE Computer Society, 2004. 8 – 13.

[3] 方建平,郝跃,刘红侠.应用混合游程编码的 SOC 测试数据压缩方法[J].电子学报,2005,33(11):1973 – 1977.

Fang Jian-ping, Hao Yue, Liu Hong-xia. A hybrid run-length coding for SOC test data compression[J]. Acta Electronica Sinica, 33(11): 1973 – 1977. (in Chinese)

[4] 詹文法,梁华国,等.混合定变长码的测试数据压缩方案[J].计算机学报,2008,31(10):1826 – 1834.

Zhan Wen-fa, Liang Hua-guo, et al. A test data compression scheme Bbased on mixed fixed and variable length coding[J]. Chinese Journal of Computers, 2008, 31(10): 1826 – 1834. (in Chinese)

[5] 彭喜元,俞洋.基于变游程编码的测试数据压缩算法[J].电子学报,2007,35(2):197 – 201.

Peng Xi-yuan, Yu Yang. A test set compression algorithm based on variable-run-length Code[J]. Acta Electronica Sinica, 2007, 35(2): 197 – 201. (in Chinese)

[6] 方昊,姚博,宋晓笛,程旭.双游程编码的无关位填充算法[J].电子学报,2009,37(1):1 – 6.

Fang Hao, Yao Bo, Song Xiao-di, Cheng Xu. The algorithm of filling x bits in dual-run-length coding[J]. Acta Electronica

Sinica, 2009, 37(1): 1 – 6. (in Chinese)

[7] Y Han, Y Hu, et al. Embedded test decompressor to reduce the required channels and vector memory of tester for complex processor circuit[J]. IEEE Transactions on VLSI, 2007, 15(5): 531 – 540.

[8] Yu-Ze Wu, Mango C-T Chao. Scan chain reordering for minimizing scan-shift power based on non-specified test cubes[A]. IEEE VLSI Test Symposium[C]. Washington DC: IEEE Computer Society, 2008. 147 – 154.

[9] A Chandra, R Kapur. Bounded adjacent fill for low capture power scan testing[A]. IEEE VLSI Test Symposium[C]. Washington DC: IEEE Computer Society, 2008. 131 – 138.

[10] Yu Hu, Yinhe Han, et al. Compression/scan co-design for reducing test data volume, scan-in power dissipation, and test application time[J]. IEICE Transactions on Information and Systems, 2006, E89-D(10): 2616 – 2625.

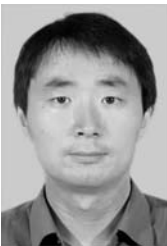
[11] D Xiang, D Hu, et al. Low-power scan testing for test data compression using a routing-driven scan architecture[J]. IEEE Transactions on CAD, 2009, 28(7): 1101 – 1105.

[12] A Chandra, K. Chakrabarty. Combining low power scan testing and test data compression for system-on-chip[A]. IEEE/ACM DAC[C]. Washington DC: IEEE Computer Society, 2001. 166 – 169.

[13] J Li, X Liu, et al. On capture power-aware test data compression for scan-based testing[A]. IEEE/ACM ICCAD[C]. Washington DC: IEEE Computer Society, 2008. 67 – 72.

[14] M Wu, J Huang, X Wen, et al. Power supply noise reduction for at-speed scan testing in linear decompression environment[J]. IEEE Transactions on CAD, 2009, 28(11): 1767 – 1776.

作者简介



刘 军 男,1978 年生于江苏新沂,合肥工业大学计算机与信息学院讲师,中科院计算所博士研究生.研究方向为内建自测试,测试数据压缩等.

E-mail: liujun@ict.ac.cn

吴 玺 女,1978 年生于安徽合肥,合肥工业大学计算机与信息学院讲师.研究方向为集成电路设计与测试,光纤通信.

E-mail: wuxi_xizi@126.com

韩银和 男,1980 年生于江苏阜宁,中科院计算所副研究员,硕士生导师,研究方向为集成电路互连、容错以及测试技术.

E-mail: yinhes@ict.ac.cn

李晓维 男,1964 年生于安徽合肥,中科院计算所研究员,博士生导师,主要研究方向为 VLSI/SoC 设计验证与测试、测试生成、易测试设计、低功耗设计、可信计算.

E-mail: lxw@ict.ac.cn