

软件与 MBIST 协同的片内 SRAM 测试方法研究

廖寅龙,田 泽,赵 强,刘敏侠

(中航工业西安航空计算技术研究所,陕西 西安 710068)

摘 要:随着 SoC 电路功能的日益复杂,SoC 电路中大量应用了内嵌随机静态存储器,传统的通过 MBIST(存储器内建自测试)方式对 SoC 电路中 SRAM 的测试,将给 SoC 电路带来功耗及管芯面积显著增大的问题;同时在传统 MBIST 方式下,SoC 内嵌 SRAM 的测试严重依赖先进的 ATE 测试设备,需要付出昂贵的测试成本。文中提出一种软件与 MBIST 相协同的 SRAM 测试方法,利用 SoC 内嵌处理器运行特定算法软件的方式,实现 SoC 电路中大部分 SRAM 的测试,同时结合传统 MBIT 测试方式对其余内嵌处理器难以访问的 SRAM 进行测试,既实现了复杂 SoC 中内嵌 SRAM 测试的完备性,也很好解决了测试完备性与测试成本的矛盾。

关键词:SoC;存储器内建自测试;测试成本

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2015)06-0155-03

doi:10.3969/j.issn.1673-629X.2015.06.034

Research on SRAM Testing Integrated in SoC of Software and MBIST

LIAO Yin-long, TIAN Ze, ZHAO Qiang, LIU Min-xia

(ACTRI, Xi'an 710068, China)

Abstract: With the development of IC, system-on-chip may have many SRAM blocks, the traditional testing method according to MBIST could not only increase the huge numbers of test logic, chip power consumption, but also tie the test to advanced ATE test machine tightly which could also cost much. A kind of new SRAM test method by combining the software achievement and MBIST together is proposed to achieve the testing of SRAM integrated in SoC. The SRAM which could hardly be accessed by any other integrated processing units would also be tested by combing the traditional MBIT and the methods mentioned in this paper, not only achieving the test completeness of SRAM integrated in SoC, but also handling the conflict between full verification and cost.

Key words: SoC; MBIST; test cost

0 引言

随着微电子工艺技术的不断发展,SoC 芯片设计的复杂度、电路规模日益提升,当前 SoC 芯片中大量应用内嵌随机静态存储器(SRAM)。据统计,目前 SoC 芯片中 SRAM 所占比例普遍超过了 50%。由于 SRAM 与普通逻辑电路相比具有晶体管密集度更高、布线密度更大、时序更严格,在制造过程中更易发生物理故障,或者失效,因此必须采取有效的措施进行测试,才能保证其功能和性能正确^[1-2]。

完备的集成电路测试显得尤为重要,同时由测试引入的成本也在不断增加。然而残酷的市场竞争对成本提出了更加苛刻的要求,在满足测试的条件下如何降低由测试引起的成本是每一个研发人员必须面对的

问题。为了降低 SoC 芯片的测试成本和难度,提高芯片的质量和成品率,需要对芯片进行可测性设计,尤其是 SoC 内部的片内存储器^[3]。文中提出一种通过软件自检与 MBIST 相结合的方法,既实现了复杂 SoC 中内嵌存储器生产测试,同时也很好地解决了测试完备性与测试成本的矛盾^[4-5]。

1 存储器内建自测试

1.1 故障分类

由于工艺的不断进步,存储器的集成度不断提高,使得每个存储器的面积和价格都按指数下降,同时存储器在制造和使用过程中都易受到干扰,故障也越来越复杂。根据故障的成因以及所影响的大小,可将存

收稿日期:2014-07-17

修回日期:2014-10-23

网络出版时间:2015-05-20

基金项目:“十二五”微电子预研(51308010603,51308010710);总装预研基金(9140A08010712HK6101)

作者简介:廖寅龙(1982-),男,工程师,研究方向为 SoC 设计与验证;田 泽,博士,研究员,研究方向为 SoC 设计、嵌入式系统设计、VLSI 设计。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20150520.1513.010.html>

储器的故障分成以下几种:AddressFaults(AF)、Address Decoder Open Faults(ADOF)、Coupling Faults(CF)、Data Retention Faults(DRF)、Stuck-at Faults(SAF)、Stuck Open Faults(SOF)和 Transition Faults(TF)。为了合理地测试这些故障,主要的测试算法大致有:March C、CheckerBoard 和 Rom^[6]。

1.2 MBIST 原理和测试步骤

MBIST 是目前业界用来测试存储器的一种常见方法,将 BIST 控制电路嵌入芯片内部,通过给相应嵌入式存储器的外围加一层测试控制电路,作为存储器与芯片其他逻辑电路的接口,负责相应的测试及控制功能,最终实现片上自动测试存储器^[7]。

MBIST 采用 March C-(march1)算法,该方法大致的测试步骤如下:

- 地址(0)→地址(最高位):Write(55)
- 地址(0)→地址(最高位):Read(55)-Write(AA)
- 地址(0)→地址(最高位):Read(AA)-Write(55)
- 地址(最高位)→地址(0):Read(55)-Write(AA)
- 地址(最高位)→地址(0):Read(AA)-Write(55)
- 地址(最高位)→地址(0):Read(55)

其原理是通过多次反复读写 SRAM 来确定其是否存在制造缺陷,其中选择 55 和 AA 的原因是因为它们的二进制代码是互补的,而且 01 交互,有利于全面覆盖故障。EDA 工具可针对每片 SRAM 额外生成 MBIST 控制逻辑,用于实现测试模式下对 SRAM 的读写测试,并通过控制逻辑判断内嵌 SRAM 是否失效^[8-9]。

1.3 MBIST 优缺点

MBIST 是目前针对嵌入式随机存储器测试的一种有效途径,具有故障覆盖率高和测试效率高的优点,但是采用这种方法必须依赖昂贵的自动测试设备(ATE),且当 SRAM 的存储容量大或 SRAM 块数增加时,采用这种方法会生成很多块 MBIST 控制逻辑,从而增大芯片面积,增加芯片功耗,甚至加长芯片测试时间^[10]。

2 SoC 中 SRAM 的软件测试方法

基于 SoC 内嵌的功能强大的处理器,通过内嵌处理器运行特定算法的软件,以一定顺序反复读写 SRAM,从而达到 SRAM 测试的目的。SRAM 失效模型有多种情况,针对特定的故障模式,设计对应的软件,控制处理器访问 SRAM 的有序集,完成对 SRAM 故障模式的覆盖测试。下面分别介绍了几种故障的测试方法^[11-12]。

2.1 地址译码故障

地址译码决定了 SRAM 内部存储单元的寻址,如

果地址译码出现故障,其典型表现为 SRAM 内部某个存储单元被屏蔽,不能被处理器访问到。因此对此故障模式内嵌处理器可以运行以下程序进行检查。

```
Void ADD_CHECK(void) /* Address decode check */
{
    Unsigned int ADD_Begin; /* SRAM begin address */
    Unsigned int ADD_End; /* SRAM end address */
    Unsignedint ADD_OP /* SRAM operated address */
    Unsigned short READ_Date;
    Unsigned inti, ;
    i = 1;
    for ( ADD_OP=ADD_Begin; ADD_OP<ADD_End; ADD_OP=
ADD_OP +2)
    { * (unsigned short *) ADD_OP = i;
    i ++; }
    i = 1;
    for ( ADD_OP=ADD_Begin; ADD_OP<ADD_End; ADD_OP=
ADD_OP +2)
    { READ_Date = * (unsigned short *) ADD_OP;
    if ( READ_Date! = i )
    { TEST_FAIL(); /* Report test fail */ }
    i ++;
    }
    TEST_SUCCESS(); /* Report test success */ }
```

2.2 数据存储单元相关故障测试

存储器的存储单元可能由于制造而引入一系列故障,比如某一存储位恒为 0,或者恒为 1,CPU 对其写操作不生效。也有可能内部存储单元存在相关性,对某一存储单元的写操作,而引起相邻存储单元的存储数值改变。因此对此故障模式内嵌处理器可以运行以下程序进行检查。

```
Void DATA_CHECK(void) /* Address decode check */
{
    Unsigned int ADD_Begin; /* SRAM begin address */
    Unsigned int ADD_End; /* SRAM end address */
    Unsigned int ADD_OP; /* SRAM operated address */
    Unsigned int ADD_tmp;
    Unsigned short READ_Date;
    for ( ADD_OP=ADD_Begin; ADD_OP<ADD_End; ADD_OP=
ADD_OP +2)
    {
        * (unsigned short *) ADD_OP =0x5555;
    }
    for ( ADD_OP=ADD_Begin; ADD_OP<ADD_End; ADD_OP=
ADD_OP +2)
    {
        READ_Date = * (unsigned short *) ADD_OP;
        if ( READ_Date! =0x5555)
        {
            TEST_FAIL(); /* Report test fail */ }
```

```

}
}
for ( ADD_OP=ADD_Begin;ADD_OP<ADD_End;ADD_OP =
ADD_OP +2)
{
    * (unsigned short * ) ADD_OP =0xaaaa;
    for ( ADD_tmp = ADD_Begin; ADD_tmp <= ADD_OP; ADD_
tmp = ADD_tmp +2)
    {
        READ_Date= * (unsigned short * ) ADD_tmp;
        if ( READ_Date! =0xaaaa)
        {
            TEST_FAIL();/* Report test fail */
        }
    }
    for ( ADD_tmp=ADD_OP+2;ADD_tmp <ADD_End;ADD_tmp
= ADD_tmp +2)
    {
        READ_Date= * (unsigned short * ) ADD_tmp;
        if ( READ_Date! =0x5555)
        {TEST_FAIL();/* Report test fail */}
    }
    for ( ADD_OP= ADD_Begin;ADD_OP<ADD_End;ADD_OP =
ADD_OP +2)
    {
        * (unsigned short * ) ADD_OP =0x5555;
        for ( ADD_tmp = ADD_Begin; ADD_tmp <= ADD_OP; ADD_
tmp = ADD_tmp +2)
        {
            READ_Date= * (unsigned short * ) ADD_tmp;
            if ( READ_Date! =0x5555)
            {TEST_FAIL();/* Report test fail */ }
        }
        for ( ADD_tmp=ADD_OP+2;ADD_tmp <ADD_End;ADD_tmp
= ADD_tmp +2)
        {
            READ_Date= * (unsigned short * ) ADD_tmp;
            if ( READ_Date! =0xaaaa)
            {
                TEST_FAIL();/* Report test fail */
            }
        }
        TEST_SUCCESS();/* Report test success */
    }
}
```

2.3 基于软件存储器测试方法的优缺点

该方法充分利用了 SoC 中的内嵌处理器,对 SRAM 测试不需要在设计中添加额外的测试电路,也不依赖于昂贵的 ATE 测试设备,通过外部软件程序可

以灵活配置。另外,如果芯片在应用过程中存在偶发性 SRAM 故障也可以很方便地通过这种方法进行在线测试。

该方法的不足之处是当 SoC 的 SRAM 存储单元无法用内嵌处理器访问时,不能使用这种方法进行测试,此时必须依赖 MBIST 方式对 SRAM 进行测试^[13-14]。

3 SRAM 的软硬件协同测试方法

随着 SoC 功能的日益复杂,各 IP 电路中都嵌入了不同容量的多块 SRAM,用于支持其数据缓冲或者功能实现。这导致复杂 SoC 芯片内嵌的 SRAM 存贮块不但数目多,而且 SRAM 的容量大小也不一样。如果为每一块 SRAM 建立 MBIST 测试电路,这将显著增加芯片面积。

为解决这一问题,文中提出了一种软件与 MBIST 相协同的高效 SRAM 测试方法。其核心思想是根据 SRAM 软件测试和 MBIST 测试的优缺点对 SoC 中内嵌 SRAM 进行如下归类:

- (1)大容量的 SRAM,一般大于 4 k 字节的 SRAM 存贮快将归于此类;
- (2)SoC 内嵌处理器不可访问的 SRAM;
- (3)小容量而且 SoC 内嵌处理器可访问的 SRAM。

在 SRAM 的测试中,把(1)、(2)两类 SRAM 使用传统 MBIST 的方式进行测试;把(3)类 SRAM 按照文中软件测试 SRAM 的方法,利用 SoC 中内嵌处理器以软件测试的方式进行测试。通过这种软件与 MBIST 相结合的 SRAM 测试手段,既保证了 SRAM 的测试覆盖率,也显著减小了 MBIST 测试逻辑的规模。

4 结束语

文中针对 SoC 芯片内部集成了大量的 SRAM,分别介绍了 SRAM 的存储器内建自测试和软件测试的方法、原理以及各自的优缺点,在此基础上提出了一种通过软件自检与 MBIST 相结合的存储器测试方法,既实现了复杂 SoC 中片内存储器的有效测试,保证了测试的覆盖率和测试效率,同时也很好地解决了测试完备性与测试成本的矛盾。该方法已成功应用于多款 SoC 芯片测试中,取得了很好的测试效果和经济效益。

参考文献:

[1] Quasem M S,Jiang Z G,Gupta S K. Benefits of a SoC-specific test methodology[J]. IEEE Design and Test of Computers, 2003,20(3):68-77.

[2] 李华伟,李晓维,尹志刚,等. 可测试性设计技术在通用 CPU 设计中的应用[C]//全国测试学术会议. 出版地不

estOrderView. cpp) 中进行 WM_COMM_RXCHAR 消息映射:

```
BEGIN _ MESSAGE _ MAP ( CTramSwicherTestOrderView, CFormView)
//{ AFX_MSG_MAP( CTramSwicherTestOrderView)
ON_MESSAGE( WM_COMM_RXCHAR, OnCommunication)
//} AFX_MSG_MAP
END_MESSAGE_MAP()
```

接着,在 CTramSwicherTestOrderView. cpp 中加入函数 OnCommunication() 的实现代码^[14]。

3 软件测试及其结果分析

软件程序测试过程中,首先在本机上利用虚拟串口和串口调试助手进行演示,检查串口能否正常工作,测试软件能否按要求发送控制命令帧数据以及接收状态命令帧数据。接着连接硬件,利用开发的测试软件对道岔驱动板和信号机驱动板进行通信测试。

在此,以测试信号机驱动板 1 上的第一架信号机点红灯为例进行说明。首先运行程序,配置串口参数并打开串口,单击信号机 1 测试栏中的红灯按钮,信号机驱动板 1 上的第一架信号机红灯指示灯(1H)点亮,同时测试软件上的数据接收框中显示信号机驱动板 1 上传的状态命令帧数据,并且界面上的第一架信号机同步亮红灯。

4 结束语

文中利用 VC++ 编程工具和多线程串口编程工具 CSerialPort 类开发了一种通信测试软件,对 CSerialPort 类在 VC++ 中实现串口通信的方法进行了探讨,主要实现了对有轨电车道岔和信号机驱动板通信功能的测试。通过实例表明,利用 CSerialPort 类实现串口通信简单、方便、灵活;另外,所开发的测试软件还可以作为

串口调试助手使用。测试结果表明,该软件简单易用,数据通信可靠稳定,具有一定的实用价值。

参考文献:

- [1] 汪 滢,孙华军. 基于 VC++ 串口通信的设计与实现[J]. 现代电子技术,2011,34(14):19-20.
- [2] 龚建伟,熊光明. Visual C++/Turbo C 串口通信编程实践[M]. 北京:电子工业出版社,2004.
- [3] 李景峰,潘 恒,杨丽娜. Visual C++/Turbo C 串口通信技术详解[M]. 北京:机械工业出版社,2013.
- [4] 王 娇,姚竹亭. 基于 VC++6.0 的串口通信设计[J]. 电子测试,2009(11):47-50.
- [5] 王 坤,高 贇. 基于 VC++ 实现串口通信的方法[J]. 信息化研究,2010,36(10):52-54.
- [6] 赵田雨,王崇宇. 单文档窗口分割视图和普通视图的切换[J]. 科技信息,2007(25):79-81.
- [7] 陈艾春. 基于 VC++ 串口通信[J]. 黑龙江科技信息,2010(1):82-82.
- [8] 丁云霞,曾连荪. 基于 VC++6.0 的 RS-422 串口通信设计[J]. 信息技术,2010(9):73-75.
- [9] Kruglinski D J, Wingo S, Shepherd G. Programming Visual C++ 技术内幕[M]. 第 5 版. 北京:北京希望电子出版社,2002:254-271.
- [10] Lenecker R C. Visual C++ Bible[M]. 北京:电子工业出版社,1999.
- [11] Denver A. Serial communications in Win32[M]. USA: Microsoft,1995.
- [12] 赵晓辉,陈艳萍,张科英,等. 基于 VC++ 环境下串口通信程序的研究与实现[J]. 科技信息,2010(1):39-40.
- [13] 王 臻,陈永泰. VC++ 实现串口通信的应用程序设计[EB/OL]. 2007-06-06. <http://www.paper.edu.cn/>.
- [14] Wang Qi, Zhu Hao, Gao Yingke. Accelerating processor verification based on ESL model[M]. Beilin: Springer-Verlag, 2012:745-751.

+++++ (上接第 157 页)

- 详:出版者不详,2002:191-195.
- [3] 王新安,蒋安平,宋春殚. 数字系统测试[M]. 北京:电子工业出版社,2007.
 - [4] 王新安,吉利久. SOC 测试中的 BIST 的若干思考[J]. 微电子学与计算机,2003,20(10):41-44.
 - [5] 桂江华,钱黎明,申柏泉,等. SOC 中的 MBIST 设计[J]. 电子与封装,2011,11(1):26-28.
 - [6] Rajsuman R. Design and test of large embedded memories:an overview[J]. IEEE Design and Test of Computers,2001,18(3):16-27.
 - [7] Rajsuman R. System-on-a-chip: design and test[M]. [s. l.]:Artech House Inc,2000.
 - [8] 余晓文. SRAM 时序单元设计及验证[D]. 合肥:中国科学技术大学,2004.
 - [9] 姚 俊. 基于 BIST 的嵌入式存储器可测性设计算法研究[D]. 哈尔滨:哈尔滨工程大学,2007.
 - [10] 方祥圣,曹先霞. 系统芯片 SOC 的 BIST 测试研究[J]. 安徽建筑工业学院学报:自然科学版,2006,14(3):59-61.
 - [11] 徐 磊. 基于 SOC 架构的可测性设计方法学研究[D]. 北京:清华大学,2002.
 - [12] 陈 凯. 基于嵌入式核测试复用的数字 SOC 测试技术研究[D]. 桂林:桂林电子科技大学,2006.
 - [13] 刘建都. CMOS 存储单元的开路故障可测性设计[J]. 现代电子技术,1994,22(2):10-14.
 - [14] 刘建都,张安堂. CMOS 存储器中地址译码器的开路故障及测试[J]. 空军工程大学学报:自然科学版,2000,1(2):78-81.

软件与MBIST协同的片内SRAM测试方法研究

作者：[廖寅龙](#)，[田泽](#)，[赵强](#)，[刘敏侠](#)，[LIAO Yin-long](#)，[TIAN Ze](#)，[ZHAO Qiang](#)，[LIU Min-xia](#)

作者单位：[中航工业西安航空计算技术研究所, 陕西 西安, 710068](#)

刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2015(6)

引用本文格式：[廖寅龙](#). [田泽](#). [赵强](#). [刘敏侠](#). [LIAO Yin-long](#). [TIAN Ze](#). [ZHAO Qiang](#). [LIU Min-xia](#) [软件与MBIST协同的片内SRAM测试方法研究](#)[期刊论文]-[计算机技术与发展](#) 2015(6)