

随机测试程序发生器的设计与实现

于 伽, 黑 勇, 陈黎明

(中国科学院微电子研究所, 北京 100029)

摘 要: 随机测试程序生成技术是当前处理器功能验证中一项重要的支撑技术. 本设计面向一种专用指令集处理器 FlexEngine, 在指令集模型建立时, 按功能分类, 实现对处理器关键单元的选择性测试; 引入 ISS 对指令执行的动态数据分析, 增加了寄存器数据范围监控、死循环预警等指令约束. 实验结果表明, 本设计的选择性测试功能, 能够在 3000 条程序的测试长度下, 对关键模块达到超过 90% 的覆盖率, 有效提高了测试效率.

关键词: 指令分类; 随机指令测试; 指令约束; 覆盖率

中图分类号: TP302.1

文献标识码: A

文章编号: 1000-7180(2012)07-0103-04

Design and Implementation of Random Test Program Generator

YU Jia, HEI Yong, CHEN Li-ming

(Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China)

Abstract: Random test program generation is an important supporting technology for the function verification of processors. We design an efficient random test program generator for FlexEngine, an ASIP. We create the instruction set model by classifying instructions according to the function, which enable it test the key modules of FlexEngine selectively. Additionally, the traditional static constraints are strengthened by ISS, which analyze data dynamically. So, we add some accurate constraints, such as monitoring the range of register data, warning the endless loop. Experimental results show that the ATPG can test FlexEngine efficiently, which achieves high coverage beyond 90%, when the length of test program is only 3000.

Key words: instruction classification; random instruction test; instruction constraints; coverage

1 引言

随着集成电路设计复杂度的提高, 处理器的功能验证已经成为设计的瓶颈^[1]. 仿真验证是主要的验证手段^[2], 通过让仿真器和处理器运行同样的测试程序, 比较执行结果, 来验证处理器功能的正确性. 因此测试程序的生成是仿真验证的重点.

当前针对随机测试程序发生器的方法主要有四种: 基于指令树方法的测试程序生成; 基于约束满足问题(CSP)的测试程序生成; 基于遗传算法的测试程序生成和基于巴科斯-瑙尔范式(BNF)的测试程序生成. 其中基于指令树方法的测试程序生成方法

是 IBM 在测试 PowerPC 时而开发的, 它尽可能的降低了随机测试生成器的复杂性, 使得测试程序的生成简单而高效.

本文面向 FlexEngine 处理器, 设计了随机指令发生器, 整体框图如图 1 所示.

2 指令集分类模型

设计随机指令发生器的第一步, 是对指令集进行分类, 建立指令模型. 指令集的分类方法很多, 近年针对这方面的研究也非常多^[3]. 本文结合 FlexEngine 处理器的结构特点, 没有采用传统分类方式^[4], 提出了一种基于处理器功能的分类方法.

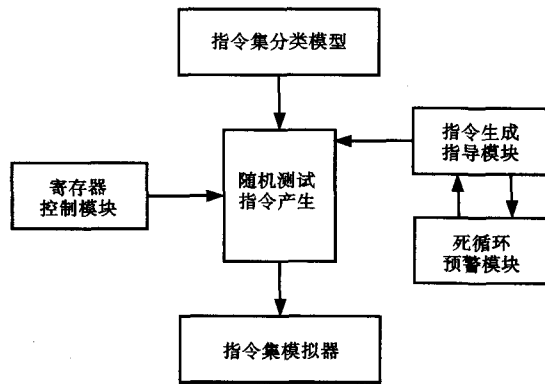


图1 随机测试程序生成器结构框图

FlexEngine 是本课题组设计的面向极低功耗语音处理类应用的专用处理器,指令集包括数据传输类、算术操作类(分为算术操作、逻辑操作和移位操作三小类)、长算术操作类、流程控制类以及定制加速指令类这 7 大类,并且支持丰富的寻址模式。基于功能分类的方法如图 2 所示。基于这种分类方式建立的指令集模型,在产生随机测试指令时,测试对象是可调的,可以对整个处理器测试,也可以针对 ALU、MAC 等处理器关键模块进行单独或任意组合测试,增强了测试程序的针对性,提高了测试效率。

分类步骤:

Step 1: 一级分类。根据指令功能,将指令集分成 7 类,即前述 7 类处理器指令,例如 ADD 指令属于 short_arith 类。

Step 2: 二级分类。对每类指令,根据指令语义进行二级分类,例如一级分类 Ld_st 类指令包含: memory 存取指令、寄存器间数据传递指令、从 memory 中同时读或取两组数据的指令等等。

Step 3: 三级分类。在二级分类的基础上,再根据操作数的个数以及是否条件执行来做出区分。

图 3 以指令 ANDN. eq r1, r2, r3 为例,介绍了指令集的分类标准。算法步骤如下:

.....

```
switch(tree_top)
{
case mem_ld_st:
{mem_ld_st_1.display();...} break;
case short_arithmetic:
{short_arith_1.display();...} break;
case short_shift:
{short_shift_1.display();...} break;
case short_logic:

```

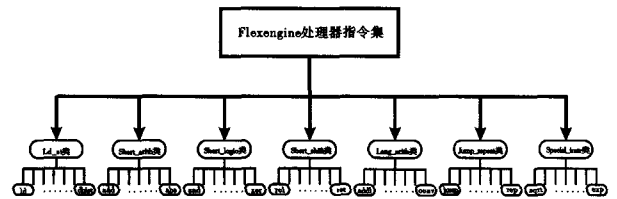


图2 指令集分类结构图

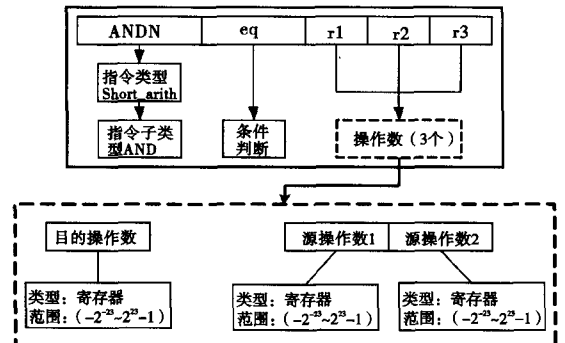


图3 指令 ANDN. eq r1, r2, r3

```
{short_logic_1.display();...} break;
case long_arithmetic:
{long_arithmetic_1.display();...} break;
case flow_control:
{flow_control_1.display();...} break;
case special_instr:
{special_instr_1.display();...} break;
}
.....
```

这样分类的好处在于分类明确,既可以生成所有指令的随机组合,又可以通过人为选择,来生成满足特定要求的指令序列,比如:只生成 ALU 类指令,或者只生成加速指令,或者可以随机组合指令类型。这样,对于处理器测试而言,就能够针对特定待测模块进行测试程序生成,大大降低了仿真时间,提高了测试覆盖率。

3 指令约束

随机测试程序生成器的核心是保证生成的指令序列合法有效,为此需要在测试程序生成时施加一定的约束。本设计在传统的基于语法规则的静态约束基础上,结合了 ISS 对指令执行的动态数据分析,进一步增强了寄存器数据范围监控、死循环预警等指令约束,大大提高了测试程序的有效性。

3.1 寄存器控制单元

在 FlexEngine 处理器中,数据位宽 24bit,地址位宽 16bit,当通用寄存器用作地址产生时,可能超

出寻址范围,但传统的静态约束基于指令规则分析,无法动态监控寄存器值的变化,因此本设计引入ISS,同时增加寄存器控制单元,用以对指令执行进行动态数据分析。

寄存器控制单元检测寄存器的内容是否超过了范围,如果超过,重新在寄存器堆里寻找,直到找到一个符合要求的寄存器。如果遍历所有32个寄存器后发现所有寄存器(特殊寄存器)中的值都超过了地址范围,那么启动寄存器更新模块,里面包含32条指令,将所有的寄存器(特殊寄存器)中的值更新,所有的值都符合寻址范围。执行完这32条指令后,再继续随机生成指令用于检测处理器,当前这条指令作废,具体的指令流程如图4所示。通过这部分的控制,随机指令发生器在生成指令的时候,在内部进行了判断,送入到指令模拟器的指令都是有效的指令,相对于通常情况下随机生成很多指令,寄存器控制模块的添加大大地提高了仿真效率。

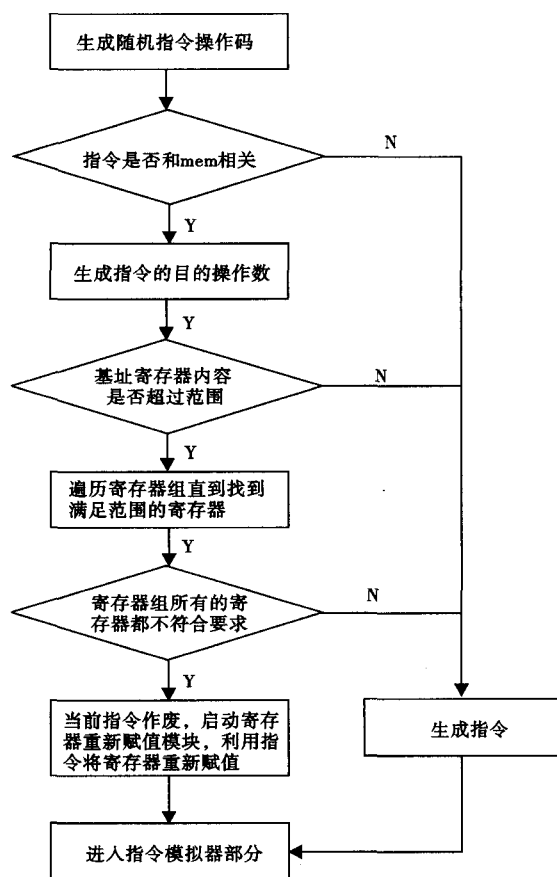


图4 寄存器控制模块流程图

3.2 死循环预警单元

随机测试程序中的结构控制是一个复杂的问题,主要集中在如何解决跳转指令的前向和后向跳转。许多的随机生成算法只处理前向跳转,而回避了

可能引起死循环的后向跳转^[5]。本设计增加了死循环预警单元,专门用来解决后向跳转可能出现的死循环问题。

当生成的指令是后向跳转时,首先找到该指令的跳转条件,借助ISS判断执行跳转。若跳转条件满足,则启动死循环预警模块,并且在死循环预警模块中记录当前的判断条件。根据指令指导模块要求,此时RTPG暂停产生指令,并且指令集模拟器ISS也暂停,直到确定该后向跳转指令不是死循环指令后,再将指令送入模拟器。

将条件满足的后向跳转指令的地址送入死循环预警模块,该模块从程序存储器中读取指令预执行,在第一遍执行的时候,死循环预警模块要找到离生成的后向跳转指令最近的一条并且影响该后向跳转指令条件的ALU/MAC指令,记录该指令的源操作数,如果:

(1) 第一遍执行完后,该后向跳转指令条件还满足,并且第二遍执行的过程中,检测出影响后向跳转条件的ALU/MAC指令的源操作数没有一个发生变化;

(2) 从要跳转地址到当前跳转指令之间所有的指令都不会改变跳转条件,也就是说当前跳转指令的执行条件是恒满足的。

在这两种情况下,确定当前构成后向跳转指令是个死循环指令,那么死循环预警模块发信号给指令指导模块,指令指导模块收到后,用NOP指令代替产生的后向跳转指令送入ISS,并且可以产生下一条指令。

3.3 指令生成指导模块

FlexEngine处理器的跳转指令多样,结构比较复杂,比如条件执行、多延迟槽、延迟槽内不支持跳转指令、循环体内不支持跳转指令等。因此,本设计增加了指令生成指导模块。

指令生成指导模块主要是根据当前指令的内容来确定下一条指令的生成范围。该模块也是整个随机指令生成器的一个核心部分。首先是判断当前指令是否跳转指令,如果是跳转指令,则判断指令是否能正确执行,若指令是条件执行指令,则判断条件是否允许,否则,默认条件允许;其次判断该跳转类指令延迟槽中的指令条数是否为0(ds0),若延迟槽指令条数为0,判断当前跳转指令是前向跳转还是后向跳转。若是前向跳转的指令,在当前地址未到跳转地址前,允许RTPG生成符合指令要求的任意指令;若是后向跳转,则RTPG暂停生成指令,只有ISS

模块执行,等到完成前向跳转后再根据上述的死循环预警模块做相应的判断.如果当前指令不是跳转类指令,那么也要根据指令序列的要求,判断当前的指令是否跳转指令延迟槽中的指令,来指导下一条指令的生成.通过指令指导模块的限定,RTPG在生成指令的时候,可以生成包含跳转指令的指令序列,极大地提高了整体代码的覆盖率.指令生成指导模块流程如图5所示.

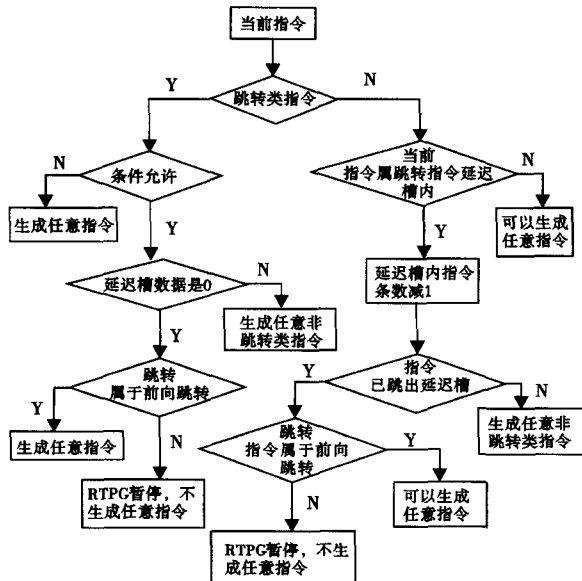
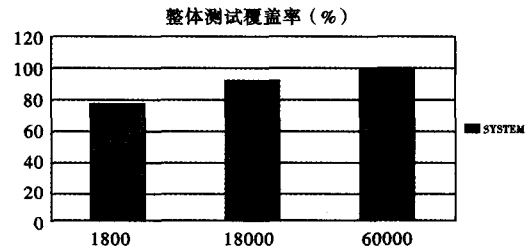


图5 指令生成指导模块流程图

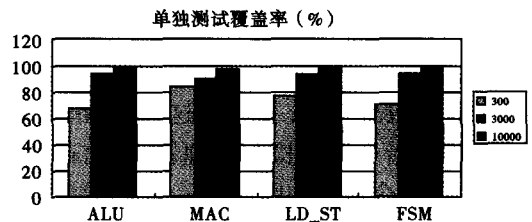
4 实验结果

应用本文设计的随机指令发生器,对项目组自行设计的FlexEngine处理器进行测试,仿真环境采用synopsis的VCS 2010.06版本,代码覆盖率统计可以利用VCS的code coverage功能方便得到.图6(a)所示为系统整体测试数据,结果显示测试长度增加到18000时,测试覆盖率才达到91.30%.对于系统的整体测试,要得到较高的覆盖率,需要较长的测试程序.

本设计通过对随机指令发生器进行参数配置,能够实现对处理器关键模块的单独测试.图6(b)给出了本设计ALU、MAC、LD_ST、FSM等处理器关键模块的单独测试结果.结果表明,通过对关键模块的单独测试,300条汇编指令的测试程序就能达到70%以上的覆盖率;当测试程序长度达到3000时,测试覆盖率就达到93%左右.测试结果表明,本设计采用的按功能划分的指令模型,有效的提高了测试程序的针对性,提高了测试效率.



(a)



(b)

图6 FlexEngine测试覆盖率

5 结束语

在处理器的功能验证中,生成满足有效性并且覆盖率高的指令测试序列是整个验证过程中的重中之重.由实验结果可知,本文设计的随机测试程序发生器取得了很好的效果,当测试数据达到一定数量时,测试系统具有很高的代码覆盖率,并且覆盖率都在95%以上.根据功能进行指令分类,可以分块测出处理器不同模块的覆盖率,动态的指令约束使得处理器的覆盖率大幅度提高.后期的研究目标是将本文设计的随机测试代码发生器,以及RTL仿真模型和参考模型相结合,形成一套输入随机验证代码,输出验证结果的处理器验证系统,为当前的项目提供完整的验证平台支持.

参考文献:

- [1] Bergeron J Ed. Writing testbenches: functional verification of HDL models[M]. London: Kluwer Academic Publishers, 2000.
- [2] 张山刚,高德远,樊晓桢,等.基于虚拟机的兼容微处理器功能验证平台[J].微电子学与计算机, 2205, 22(1):140.
- [3] Aharon Aharon, Dave Goodman. Test program generation for functional verification of powerPC processors in IBM[C]//proceedings of the 32nd ACM IEEE Design Automation Conference NY, USA: ACM, 1995: 279-285.
- [4] 张振军,毛志刚.基于伪随机的微处理器验证方法及改进[J].信息技术, 2008(10): 59-61.

(下转第111页)

link 工具对其进行系统建模和仿真,在输入信号频率为 1 MHz、过采样率为 64 时,理想系统的峰值信噪比可达到 105 dB. 进一步分析了调制器系统的非理想因素并建模,当运放的直流增益大于 40 dB,单位增益带宽大于 2 倍采样频率时,可获得 90 dB 以上的信噪比. 由于连续时间型 $\Sigma\Delta$ 调制器不需要额外的抗混叠滤波器,相比于离散时间型 $\Sigma\Delta$ 调制器,功耗更小且适合宽频领域的应用.

参考文献:

- [1] Jiang Y, Bin Z. Continuous-time sigma-delta modulator design for low power communication applications [C] // The 7th IEEE International Conference on ASIC. Guilin, China, 2007: 715-720.
- [2] Schreier R, Zhang B. Delta-sigma modulators employing continuous-time circuitry [J]. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 1996, 43(4): 324-332.
- [3] Malcovati P, Brigati S, Francesconi F, et al. Behavioral modeling of switched-capacitor sigma-delta modulators [J]. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 2003, 50(3): 352-364.
- [4] Boser B E, Wooley B A. The design of sigma-delta modulation analog-to-digital converters [J]. IEEE Journal of Solid-State Circuits, 1988, 23(6): 1298-1308.
- [5] 褚子乔,王东辉,侯朝焕. 高性能连续时间 Sigma-Delta 调制器系统级设计[J]. 微电子学与计算机, 2008, 25(8): 45-47.
- [6] Shamsi H, Shoaie O. A novel structure for the design of 2-1-1 cascaded continuous time delta sigma modulators [C] // IEEE International Symposium on Circuits and Systems. New Orleans, USA, 2006: 4751-4754.
- [7] Ortmanns M, Gerfers F, Manoli Y. A case study on a 2-1-1 cascaded continuous-time sigma-delta modulator [J]. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 2005, 52(8): 1515-1525.
- [8] Yan C, Kong-Pang P. A 0.5V 2-1 cascaded continuous-time delta-sigma modulator synthesized with a new method [C] // IEEE International Midwest Symposium on Circuits and Systems. Seattle, USA, 2010: 1165-1168.
- [9] Tongyu S, Zhiheng C, Shouli Y. A 2.7-mW 2-MHz continuous-time sigma-delta modulator with a hybrid active-passive loop filter [J]. IEEE Journal of Solid-State Circuits, 2008, 43(2): 330-341.
- [10] Paton S, Di Giandomenico A, Hernandez L, et al. A 70 mW 300 MHz CMOS continuous-time sigma-delta ADC with 15 MHz bandwidth and 11 Bits of resolution [J]. IEEE Journal of Solid-State Circuits, 2004, 39(7): 1056-1063.
- [11] Gerfers F, Ortmanns M, Manoli Y. A 1.5 V 12 bit power-efficient continuous-time third-order $\Sigma\Delta$ modulator [J]. IEEE Journal of Solid-State Circuits, 2003, 38(8): 1343-1352.

作者简介:

沈琪 男,(1987—),硕士研究生.研究方向为专用集成电路与系统设计.

王伟印 男,(1988—),硕士研究生.研究方向为专用集成电路与系统设计.

顾晓峰 男,(1971—),博士,教授,博士生导师.研究方向为半导体器件物理、集成电路设计.

赵琳娜 女,(1979—),讲师.研究方向为集成电路设计.

于宗光 男,(1964—),博士,高级工程师,教授,博士生导师.研究方向为集成电路设计.

(上接第 106 页)

- [5] 王惊雷,汪东升. 微处理器随机测试程序生成器[J]. 计算机工程与设计, 2004(25): 1144-1146.

作者简介:

于伽 女,(1986—),硕士研究生.研究方向为处理器验证

平台设计.

黑勇 男,(1974—),研究员.研究方向为可重构 DSP 及低功耗数字基带芯片.

陈黎明 男,(1982—),助理研究员.研究方向为计算机体系结构与 VLSI 设计.