**论文编号：**

**XXXXXXXXXXXX**

**摘要**: 在线测试处理器时，控制模块的结构复杂，现有的方法难以对此达到理想的故障覆盖率。

**关键词:** 软件自测试

**A Gate-Level XXXX**

**Abstract**: As control unit becomes increasingly complex, online testing hardly leads to satisfying fault coverage on that unit.

**Key Words**: SBST; Model checking; Abstraction

—————————————

**1 引 言**

由于处理器主频的不断升级，而外部测试仪的速度提升缓慢而且造价昂贵，因此直接使用外部测试仪对处理器进行实速测试，变得越来越困难。然而，处理器的很多故障只有在实速条件下才能被激活。另一方面，针对处理器这种核心部件，用户有强烈的现场测试需求，以确保整个计算机系统的正确性。因此，为确保处理器的可靠运行，迫切需要一种在线测试方法能够实速测试处理器。

**2 相关工作介绍**

**2.1 基于软件的自测试方法**

基于软件的自测试方法（SBST）通过运行程序直接检测处理器中功能与结构故障。这种方法不需要引入任何硬件开销，就能够在线地测试处理器，并且达到理想的故障覆盖率，已经成为了一种非常理想的测试方法。

**2.2 有界模型检测**

有界模型检测[5][6]是一种很重要的自动验证技术。它不仅可以自动地验证有穷状态系统中命题的正确性，而且一旦命题有错，它能够提供不满足命题的违例。因此，有界模型检测能够在控制器的门级电路上，直接获得触发难测故障property的测试序列。有界模型检测由于穷尽地搜索了所有的测试序列，因此它能够控制器中故障难以触发的问题，进而在控制器上达到良好的测试效果。。

**3 采用有界模型检测的门级软件自测试方法**

本文采用有界模型检测技术，提出了基于BMC的门级软件自测试方法（Gate-Level Software-Based Self-Testing using Bounded Model Checking，GB-SBST）。这种方法能够针对控制模块中的难测故障产生测试程序，有效地测试时序的控制模块，进一步提升现有ATIG测试方法的测试效果。

**3.1 模型的获得与简化**

目前，有界模型检测工具只能处理采用smv语言[11]编写的设计，被测电路的门级电路需要采用smv语言建模描述。由于门级电路中的组合门对应smv语言中的线变量，而且时序的寄存器对应smv语言中的状态变量，因此我们编写了一个转化程序，将被测电路的门级电路文件直接转化成smv语言描述的文件。

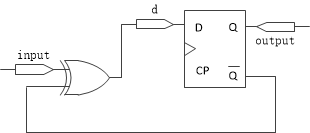


图4　一个时序电路

具体而言，为了缩减被测模块的规模，整个系统按功能被分成若干功能模块。由于系统中模块之间并不是相对独立的，而是相互协作的，这样模块的上游模块的输出取值范围就是该模块的输入约束，而模块的下游模块能够接受输入的取值范围就是该模块的输出约束。一当获得这些约束，它们将被转化成为约束模块，如图3所示。约束模块的作用就是控制待测模块的输入与输出，它能够替代其它模块的功能，因此其它模块就抽象成为这些约束模块。这样，抽象模块与待测模块共同组成了一个功能完备的系统，它等价于原系统。

**3.2 抽取激活难测故障的违例**

在获得约束模块后，我们需要针对产生有界模型检测的属性，激活被测电路中的难测故障。如图2流程图中2\_2所示，我们首先获得被测电路的难测故障，判断这个故障集合是否为空。如果集合不空，则逐个从故障集合中取出难测的故障，编写激活该属性的基于线性时序逻辑（LTL）的属性。然后，我们运行BMC工具验证这条属性。如果在给定的时间片内，BMC工具未能搜索到任何违例，那么属性恒成立，难测故障在给定的时间片内是无法被激活的。；而如果属性不成立，那么BMC工具将给出一个违例，违反这条属性。

**4 GB-SBST对Parwan处理器的控制模块的测试**

对于每一个状态位的约束，也可以由下面四组布尔等式(2)-(5)表示。根据每个状态位的约束，我们设计约束电路，表示在处理器正常工作模式下，状态寄存器SR对控制模块施加的影响。我们将等式(2)-(5)转化成为约束电路的门级网表文件,进一步转化成为smv文件，施加到控制模块smv文件的四位输入状态位上。

表2 SR状态位的约束关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | V | C | Z | N |
| 约束 1 | x | C=V | 1 | 0 |
| 约束 2 | V= C xor N | X | 0 | x |
| 约束 3 | 0 | 0 | 0 | 0 |

**4.1 得到难测故障对应的违例**

使用同样的方法，我们可以测试控制模块的所有难测故障。表3展示了针对难测故障运行BMC的结果，其中控制模块有40个难测故障，BMC成功地产生了32个故障的违例，但是有8个故障不存在激活它们的违例。这是因为在控制模块的状态机中，这些信号始终保持固定值，它们是不可测试的故障。

表3. 针对难测故障运行BMC的结果

|  |  |  |
| --- | --- | --- |
| 难测故障数 | 获得违例的故障数 | 未获得违例的故障数 |
| 40 | 32 | 8 |

**5 实验建立与结果分析**

本实验使用商业ATPG工具进行故障模拟，获得故障覆盖率。图8显示了实验的主要步骤。首先，触发指令序列与观测指令序列组成自测试程序，并且与测试数据一起编译成为内存文件。然后，实验将内存文件加载到Parwan处理器上，在modelsim工具进行模拟，并且保存模拟过程中处理器输入输出端口值的变化。最后，实验将值变化文件和Parwan处理器加载到TetraMAX进行故障模拟，并且得到测试的故障覆盖率。

表4 不同测试方法的故障覆盖率

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 故障覆盖率 | Full-scan | RSBST | ATIG | ATIG+GB-SBST |
| 控制模块 | 95.1 | 79.0 | 84.4 | 88.9 |
| 处理器 | 97.9 | 92.1 | 94.8 | 95.5 |

**6 结 论**

处理器中核心的控制模块由于结构复杂，并且时序深度较大，现有方法难以有效地测试它。

**参考文献**

1. Juin-Ming Lu; Cheng-Wen Wu, “Cost and benefit models for logic and memory BIST”, Proceeding of Design, Automation and Test in Europe Conference and Exhibition, 2000, pp. 710-714.
2. J. M. Miranda, "A BIST and boundary-scan economics framework", IEEE Design & Test of Computers, 1997, pp. 17-23.