

# 利用演化硬件技术提高控制系统可靠性的新方法<sup>\*</sup>

吴彩华<sup>1a</sup>, 彭世蕤<sup>1b</sup>, 李海鸿<sup>1a</sup>, 刘俊涛<sup>2</sup>

(1. 空军雷达学院 a. 5 系 信息作战指挥教研室; b. 5 系, 武汉 430019;

2. 军械工程学院 5 系 软件工程教研室, 石家庄 050003)

**摘要:**利用演化硬件技术改进了提高控制系统可靠性的传统方法,即在控制系统中增加演化装置,依靠演化算法实现系统的自维护和自修复,以此进一步提高系统的可靠性.并以改进 FPGA 实现的三模块冗余系统为例,详细说明了改进的方法,对改进后的系统进行了可靠性分析,结果表明,改进后的三模块冗余系统能够更持久的正常运行.

**关键词:**控制系统; 演化硬件; 演化算法

**中图分类号:** TP311

**文献标识码:** A

**文章编号:** 1006-0707(2009)09-0019-03

控制系统,例如星载控制系统,几乎完全依靠大规模集成电路来实现.如何提高芯片可靠性,成为提高控制系统可靠性的关键问题.然而,就芯片本身而言,其可靠性的提高不可避免地要受其物理结构和使用环境的限制,而且其可靠性还会随着系统使用时间的增加而急剧降低.尤其对于星载控制系统来说,它通常在异常恶劣的环境(例如,高温、核辐射、电磁辐射等)中运行,再加上其运行周期长,任务具有多样性、可变性、长期性和复杂性的特点,一旦发生故障,不可能进行人工修理.因此,传统意义上的提高可靠性方法已经难以适用于星载控制系统了.然而,演化硬件具有的自适应、自组织、自修复特性却可以满足这一要求.本文把演化硬件技术与提高系统可靠性的传统方法相结合,以此进一步提高系统的可靠性,为提高控制系统可靠性工作提供了新的思路.

## 1 演化硬件基本原理

演化硬件 (evolvable hardware, EHW), 实际上是一种特殊硬件,它可以像生物一样具有自适应、自组织、自修复的特性,从而可以根据使用环境的变化而改变自身结构以适应其生存环境.简单的说,演化硬件就是遗传算法和可编程器件的有机结合<sup>[1]</sup>.

演化硬件作为一种新的技术,它的实现就是要将硬件融入到遗传算法中去,在硬件上进行实时的硬件演化,即在遗传算法的基本框架中实现硬件的重构.将可编程器件中的能够形成的所有功能电路作为一个空间,对空间中的

个体,即硬件的一个设计,用编码来表示,利用遗传算法在这个空间中对问题进行搜索,找到满足预定功能的硬件结构,这就是演化硬件的基本原理<sup>[1]</sup>.

演化硬件技术除可按照功能或指标的要求在通用的 FPGA 芯片上形成不同的系统以外,还可在与外部环境相互作用之后,根据出现的问题调整部分参数,重新进行电路合成以形成新的电路结构,使其更加符合使用环境的具体情况.从而,使得此类硬件电路具有了自主的、动态的改变自身结构和行为的可能,即具有了自组织、自修复的功能.利用演化硬件的这个特性,可以使控制系统在没有人工干预的情况下进行维护工作,以提高其可靠性,这正是我们所需要的.

## 2 应用演化硬件技术提高系统可靠性的方法

### 2.1 提高可靠性的传统方法

可靠性是指产品在规定的条件下和规定的时间内完成规定功能的能力,可分为固有可靠性和使用可靠性.固有可靠性是系统在设计、制造过程中被赋予的.使用可靠性是系统在使用中呈现出来的可靠性<sup>[2]</sup>.据此,提高系统中集成电路芯片可靠性的方法可以分为 2 类:一是为了提高系统的固有可靠性,从加工过程内部的精选基础原材料、提高制作工艺水平做起;二是为了提高使用可靠性,尽量降低使用环境对芯片的干扰,例如在使用时在芯片外部增加各类密封和屏蔽装置等<sup>[3-5]</sup>.

另外,为了满足某些领域的对可靠性的较高要求,往

\* 收稿日期:2009-06-09

作者简介:吴彩华(1980—),女,河北衡水人,博士,主要从事演化硬件、软件测试、软件保障等方面的研究.

往对一些核心元件采用了并联冗余或等待冗余系统.其中,较为典型的的就是三模块冗余(TMR)技术<sup>[3]</sup>.

## 2.2 利用演化硬件技术提高控制系统可靠性

由于控制系统大都是由大规模集成电路构成的,所以要直接在芯片一级提高可靠性是最佳的.现有可重配置集成电路,如FPGA,CHLD等,技术已较为成熟,具有规模大、集成度高的特点,如果利用演化硬件技术,合理组织和动态调整芯片内部的电路结构,使其能够动态地“适应”任务以及运行环境,则必能提高系统可靠性.具体说来,是要在如下3种情况下保证系统正常运行:

1) 在正常情况下,系统内各单元均无故障.演化装置不工作,此时系统依靠器件内部的校验机制确保输出结果正确;

2) 在异常情况下,系统内故障单元数不超过半数,表决模块仍然有效.演化装置对发生故障的单元进行演化直到恢复初始状态.此时系统依靠器件内部的仲裁机制实现容错运行;

3) 在故障情况下,系统内故障单元数超过半数,表决模块失效.此时系统停止运行,演化装置对所有单元进行演化,直到恢复系统初始状态.这样提高了系统的可靠性,延长了系统的整体寿命,实现在各种环境下的运行.

根据上述思想,下面以传统的三模块冗余系统为例,详细介绍如何利用演化硬件技术对其进行改进.

## 3 应用演化硬件技术对三模块冗余系统的改进<sup>[6-10]</sup>

### 3.1 传统三模块冗余容错技术的弊端

在诸如星载控制系统等一类需要高可靠性的控制系统中,普通的硬件设备在强电磁辐射及高负荷运转等恶劣环境中极易丧失其功能,而无法完成其任务.例如当单个高能粒子轰击器件电路而发生单粒子翻转,使器件的逻辑状态翻转.因此采用容错运行的技术来确保设备功能的实现,从而实现设备长期可靠运行.通常采用冗余技术来实现容错运行.二模块冗余系统的故障覆盖率难以得到保证,切换时间过长,可靠性指标并不高.具有表决机构的多模块冗余系统,在很大程度上能消除输出结果的不确定性.表决、故障检测、屏蔽、隔离等容错机制由硬件或软硬件相结合(硬件占有相当的比重)的方式实现,所用时间短.所以比较适合要求高可靠性、高可用性、强实时性的场合.容错设计实际应用中最有代表性的容错机制是三模块冗余技术.

在图1中,F1、F2、F3为3个同样的功能模块,X为1个表决模块,FL为诊断输出,每个功能模块的输入为 $I = (i_1, i_2, \dots, i_k)$ ,表决模块输出为 $O = (o_1, o_2, \dots, o_n)$ ,表决模块使用三选二方法在其输入值中(3个功能模块的输出)选出合适的输出值.同时输出3个位值FL来判断表决器的可能错误输入(即与其它值不同的输入).判别方法可以使用真值表表示(如表1).为了简单表示,仅仅列出3个功能模块的输出为1位的情况.

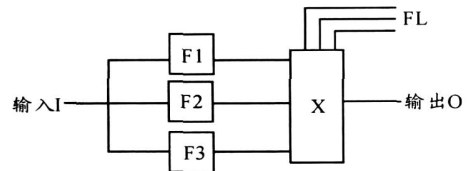


图1 三模块冗余容错电路

表1 输出为1位时FL的真值表

$O_1$	$O_2$	$O_3$	FL(0 1 2)
0	0	0	0 0 0
0	0	1	0 0 1
0	1	0	0 1 0
0	1	1	1 0 0
1	0	0	1 0 0
1	0	1	0 1 0
1	1	0	0 0 1
1	1	1	0 0 0

$O_1$ 、 $O_2$ 、 $O_3$ 为3个功能模块的输出,诊断输出FL的3个位值分别表示3个功能模块,FL中为1位值表示相应的功能模块出现了故障.比如,FL=010,说明第2个功能模块的输出与其它2个不同,可能是第2个功能模块出现了故障.如果FL=000,那么就说明3个功能模块均没有发生故障.当有模块发生故障时,通常的方法就是把它进行隔离,整个系统仍然运行.

虽然具有表决机构的三模块冗余系统,能够很大程度上消除输出结果的不确定性,甚至能够实现“容错运行”.但是在特别严重的情况下,可能有半数以上的模块出现故障而发生表决不一的情况,此时系统将无法正确运作.另外,依照传统的硬件设计技术,系统设计完成之后很难再行改变它的结构与功能.这就导致在传统的容错设计中,虽然能够保证某一模块一旦出现故障后系统还可正常工作,但是故障模块的功能恢复工作仍然需要人工干预,而人工干预在许多情况下是无法实现的.这就给传统的三模块冗余系统提出了挑战.

### 3.2 应用演化硬件技术改进TMR容错方法

为了进一步提高三模块冗余系统的可靠性,我们在系统中连接一个演化算法装置.正常情况下,加入演化装置的三模块冗余系统和传统的三模块冗余系统并没有太大区别,只要系统中故障单元不超过半数,这个系统就仍然类似于一个传统的三模块冗余系统.如果系统发生异常,其中1个单元失效,系统的表决模块将会定位这个失效单元,并启动演化装置,对失效的单元进行演化.此时,系统依靠未失效的单元和表决模块保证其正常运行,直到失效单元恢复到正常的状态.这样,系统最终能够恢复到正常状态.当系统中有超过半数的单元发生故障时,系统停止运行,表决模块认为所有的单元模块均出现故障,演化装置对所有的单元进行演化直到全部修复.之后,系统继续运行.理想的演化装置应该能够把失效的单元恢复到正常.

利用演化硬件技术对三模块冗余技术的改进,关键是增加了一个演化装置,在系统没有正常工作时,演化装置代替人对故障单元进行维修,以此实现自维护、自修复特性,进而提高控制系统的可靠性。

以 FPGA 实现的传统三模块冗余系统为例,在该系统中有 3 个完全一样的功能单元和 1 个表决模块,我们加入一个演化装置,该装置可以把演化后得到的配置数据通过配置总线写入功能单元中。当表决模块检测到某个功能单元发生故障时就终止其运行。演化装置根据设定的遗传算法参数,运行遗传算法演化出新的 FPGA 配置数据,将配置数据下载到相应的 FPGA 功能单元中。故障功能单元通过演化被恢复后系统继续正常运行。

整个过程中,通过遗传算法演化出新的电路是系统的关键问题。根据功能单元中电路的规模需要合理的选择遗传算法(例如,简单遗传算法、精英算法等)、以及编码方式(例如,二进制编码、实数编码等)。最后需要根据实际情况设定和调整遗传算法参数(交叉率、变异率、种群数等)。电路的演化过程如下:

1) 随机产生初始种群;

2) 计算种群中每个染色体的适应度:通常情况下,我们把适应度函数定义为出错功能模块的输出与期望输出的相似程度,即出错功能模块的输出与期望输出中相同的位数与输出位总数之比,用  $F$  表示( $0 \leq F \leq 1$ ,  $F=1$  表明出错模块恢复到正常状态)。然后根据适应度计算结果进行遗传操作(选择,交叉,变异)产生新的种群中的染色体;

3) 重复步骤 2) 直到有 1 个染色体的  $F=1$ ,将该染色体转换成 FPGA 的重配置数据并通过配置总线下载到相应的 FPGA 功能单元中,使出错的单元恢复其功能。

### 3.3 改进后三模块冗余系统的可靠性分析<sup>[11-13]</sup>

设三模块冗余系统中每个功能单元的可正常运行的概率为  $p$ , 设演化装置能够修复 1 个功能单元的概率是  $q$ , 需时  $t_0$ , 传统三模块冗余系统只要有 2 个模块不能正常工作整个系统就不能正常工作, 所以整个系统的可正常工作的概率是:  $P = 1 - 2p^3 + 3p^2$ , 过  $t_0$  时间后还能正常工作的概率是

$$P_{t_0} = p^3(p^3 + 3p^2(1-p)) + 3p^2(1-p)p^2 - 2p^6 + 3p^4 \quad (1)$$

如果加入演化装置, 过  $t_0$  时间后还能正常工作的概率是

$$P_{t_0} = p^3(p^3 + 3p^2(1-p)) + 3p^2(1-p)(p^2 + 2p(1-p)q) \quad (2)$$

$P_{t_0} > P_{t_0}$ , 可见改进后的三模块冗余系统能更持久的正常工作。

## 4 结束语

诸如星载控制系统一类的控制系统, 使用环境非常恶劣, 一旦出现故障后, 不可能进行人工修复, 如何提高其可靠性成为至关重要的问题。本研究在提高可靠性的传统方

法基础上, 应用演化硬件技术, 使系统可以“快恢复”。即系统一旦出现故障, 可以快速的被修复。而且, 系统还可以根据使用环境的变化, 快速调整电路结构使其更加适应环境, 进一步提高了系统的可靠性。通过对传统三模块冗余系统的改进及分析, 可以看到这种方法是可行的。

## 参考文献:

- [1] 赵曙光, 刘贵喜, 杨万海. 可进化硬件的基本原理与关键技术[J]. 系统工程与电子技术, 2002, 24(1): 70-73.
- [2] 周青龙, 贾希胜, 朱小冬, 等. 可靠性与维修工程[M]. 石家庄: 河北教育出版社, 1992.
- [3] 王国庆. 演化硬件在容错技术中的应用研究[D]. 石家庄: 军械工程学院, 2005.
- [4] 曲瀛. 硬件演化技术及其应用基础研究[D]. 南京: 南京航空航天大学, 2002.
- [5] 王小平, 曹立明. 遗传算法——理论、应用与软件实现[M]. 西安: 西安交通大学出版社, 2003.
- [6] Yao Xin, Higuchi T. Promises and Challenges of Evolvable Hardware[J]. IEEE Trans. on Systems Man and Cybernetics-Part C: Applications and Reviews, 1999, 29(1): 87-97.
- [7] Mizoguchi J. Production Genetic Algorithms for Automated Hardware Design Through an Evolutionary Process[C]// In Proc. first IEEE Conf. Evolutionary computation (ICEC '94), 1994: 661-664.
- [8] THOMPSON A. Analysis of unconventional evolved electronics[J]. Communication of the ACM, 2002, 42(4): 71-79.
- [9] Higuchi. Evolvable hardware at function level [C]// Proc 1997 IEEE Int Conf Evolutionary Computation, (ICEC '97), 1997: 187-192.
- [10] STOICA A. Evolution of analog circuits on field programmable transistor arrays [C]// Proc of the Second NASA/DOD Workshop on Evolvable Hardware (EH'00), 2000: 99-108.
- [11] Thompson A. Explorations in Design Space: Unconventional Electronics Design Through Artificial Evolution [J]. IEEE Trans. on Evolutionary Computation, 2000, 3(3): 167-196.
- [12] Stoica A. Evolution of Analog Circuits on Field Programmable Transistor Arrays [C]// In Proc. of the Second NASA/DOD Workshop on Evolvable Hardware (EH'00), 2000: 99-108.
- [13] Adrian Thompson. Temperature in Natural and Artificial Systems [C]// Proc. 4th Eur. Conf. on Artificial Life (ECAL97), Husbands, P. and Harvey, I., MIT Press, 1997: 388-397.