

数字演化硬件的函数级在线进化技术研究^①平建军^② 王友仁^③ 高桂军 孔德明 姚 睿 张 岩

(南京航空航天大学自动化学院 南京 210016)

摘 要 采用基于现场可编程门阵列(FPGA)模型的数字电路函数级在线自适应进化方法,提出了一种新的有效的进化策略:根据目标电路的功能组成将目标电路分解为多个子模块单元,然后对多个子模块单元分别进行进化设计,再利用子模块单元进行预期目标数字电路的在线进化,以降低每一级进化的染色体长度和提高收敛速度。以 4×2 乘法器电路为例,用这种方法成功实现了数字电路在线进化,结果表明,与逻辑门级进化方法相比,该方法改善了在线进化时存在染色体过长、收敛速度缓慢、测试评估困难的状况,收敛速度提高了5倍左右。

关键词 演化硬件, 数字电路, 在线进化, 函数级进化

0 引 言

演化硬件^[1-4](evolvable hardware, EHW)是基于生物进化理论设计的电子电路,具有自组织、自修复、自适应等特性。与传统硬件电路相比,EHW能够根据环境的变化自适应改变自身的结构,可用于实现复杂电路自组织设计、电子系统芯片级在线故障修复,在工业控制、航空航天、深海作业、核工业等领域有广阔的应用前景^[5-7]。

演化硬件进化方法是演化硬件的关键技术与难点问题。目前演化硬件主要还是基于逻辑门级的进化,如 Jim Torresen 等人^[8]利用逻辑门级进化设计了机电假手控制器,Adrian Burian 等人^[9]基于逻辑门级进化设计了用于图像还原的门阵列电路。然而,逻辑门级进化方法用于大规模电路时,电路编码染色体太长,导致进化速度缓慢,甚至难以收敛。针对这种问题,一些专家和学者提出了函数级进化方法,如 Masahiro Murakawa^[10]提出了使用高级函数单元取代逻辑门作为基本可重构单元,通过增大基本单元的粒度来缩短染色体长度,从而减小了进化算法的搜索空间,提高了进化速度,乔双^[11]提出了一种用于函数级进化的可编程浮点处理单元,Weixin Liu^[12]利用函数级进化实现了异步传输模式(asynchronous transfer mode, ATM)单元调度。这些现有的函数级进化方法虽然一定程度上提高了数字电路进化速度与

成功率,但还存在明显不足之处:一是数字电路进化一般是基于离线仿真进化的,而不是在线进化;二是基本进化逻辑单元大多是通过预先设计的,只能提供有限的几种可编程功能。且所进化电路规模也不大,进化成功率虽有所提高,但进化速度还是不能满足在线进化要求。因此,如何实现复杂数字电路的快速、有效的在线进化仍是目前迫切需要研究解决的问题。

本文研究了基于现场可编程门阵列(field-programmable gate array, FPGA)模型的数字演化硬件函数级在线进化方法提出了一种加速进化的策略,即根据电路功能组成把电路分解为多个子模块,进化设计各个子模块,再基于子模块进行函数级在线进化,并利用优良基因锁定^[13]技术来提高演化硬件的函数级进化的收敛速度。针对该方法引起的“早熟”缺陷,通过引入二次初始化技术,提高了电路的进化成功率。

1 函数级在线进化及实现

1.1 函数级进化原理

数字电路函数级进化原理如图1所示。利用粗粒度函数功能单元(function unit, FU)来在线进化实现目标电路,而函数功能单元必须事先进行分解进化。由于分解所得的函数功能单元逻辑电路规模较小,可基于逻辑门来进化设计,而 Xilinx 公司的 FP-

① 国家自然科学基金(60374008、90505013)和航空科学基金(2006ZD52044、04152068)资助项目。
② 男,1982年生,硕士;研究方向:数字演化硬件应用技术研究;E-mail: pingjianjun@yahoo.com.cn
③ 通讯作者, E-mail: wangyrc@nuaa.edu.cn
(收稿日期:2008-03-21)

GA 的逻辑门是以查找表(look-up table, LUT)为最小逻辑单元。这种分层次函数级进化方法可以有效缩

短染色体长度,简化了电路功能的在线测试,能保证快速寻找到满意解。

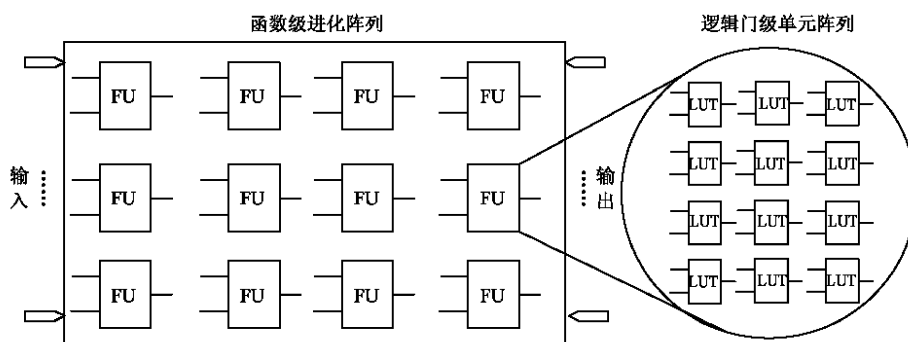


图1 数字电路函数级进化原理

1.2 在线进化系统构建

Xilinx 公司的 Virtex 系列 FPGA 具有快速可局部重构的特性,选择这一系列的 FPGA 为硬件实现平

台,利用基于 JAVA 语言的 JBits2.8 API 软件包为软件平台。在线进化实现的系统组成如图 2 所示。

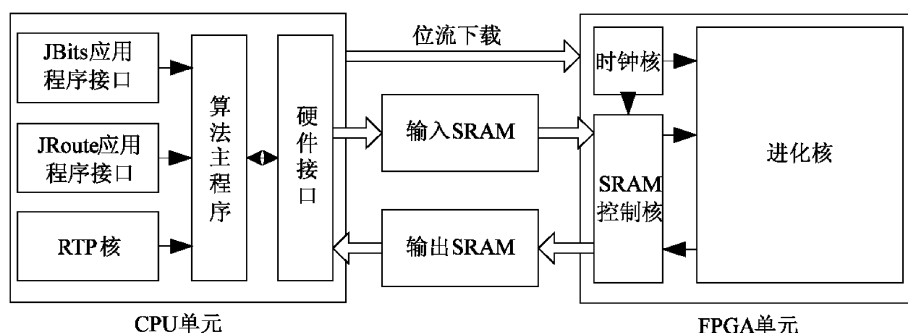


图2 数字电路在线进化系统组成

图2中前半区域为CPU执行部分,其中JBits提供了对FPGA进行实时重构和可参数化设计,RTP(run-time parameterized)核为实时可参数化核,其提供JBits配置命令的抽象方法,JRoute是实时布线器,提供对FPGA的抽象层布线工具,在CPU中通过算法进行电路的进化,然后通过提供的硬件应用程序接口使用PCI等方式下载到FPGA中实时评估,测试数据由输入输出静态随机存储(static random access memory, SRAM)来进行传送。图2的后半区域为FPGA中的硬件实现结构示意图。

1.3 自适应策略

通常为了提高进化效率,进化算法中进化算子(变异率和交叉率)随进化过程而自适应变化。为此,采用了自适应的变异因子,变异方式为:

$$M_Rate = ((\maxScore - \maxCScore) / \maxScore) \times \maxMPercent \times length \quad (1)$$

其中, M_Rate 表示当前电路染色体当前变异的位

数, $\maxScore$ 表示评价电路的最大适应度, $\maxCScore$ 表示当前评价电路的适应度, $\maxMPercent$ 表示最大变异百分比, $length$ 表示染色体的长度。

根据公式(1)可知,随着进化过程中电路适应度的增大,染色体的变异位数将会自适应减少,从而既避免了优良基因被破坏,又可加快收敛速度。

1.4 生长进化策略

进行数字电路在线进化设计需要预先确定待进化设计电路的功能,在利用进化算法寻找满意电路解的过程中,需把每代、每个染色体所对应的配置位流下载到现场可编程逻辑器件中进行验证测试评估,如果电路的性能满足设计要求或满足终止条件时,即结束进化设计过程。

然而,通过分析电路进化设计过程发现,某些染色体的一部分基因进化到优良(即子模块电路功能达到进化要求),但没有满足整个系统功能要求,在后续进化过程中,优良的染色体基因又被破坏,需要

重新进化,从而会增加进化时间。因此,我们提出了这样一种思想:当某个子模块功能满足要求时,不再变动该子模块而只进化其他模块功能以及模块和模块之间的连接关系,这样可提高进化效率。具体实现方法为:以单染色体变异进化为例,在进化过程中,当某一个或多个输出引脚进化到满足要求时,为

了避免这些“好”引脚被破坏后又重复进化,将与这些输出引脚有关的功能单元及连线进行标示并将之“锁定”,并将这些锁定单元相应的基因用于种群学习,然后再进行进化(如图3)。这种方法可称为“生长型”进化策略。

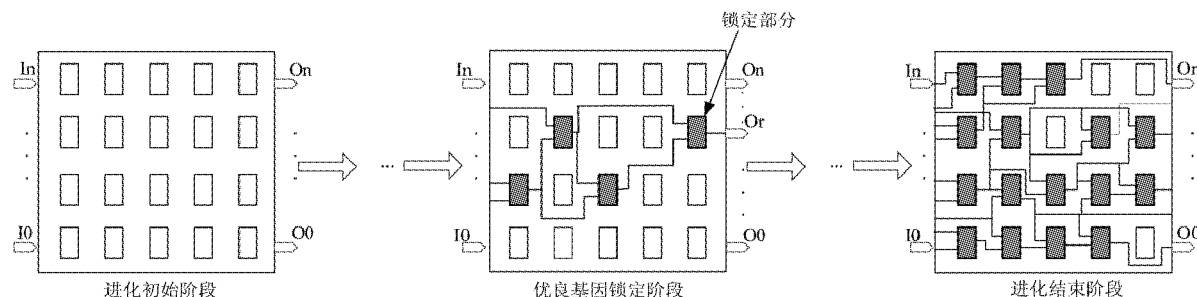


图3 生长进化策略加速收敛进化过程

2 设计实例与结果分析

以 4×2 乘法器为例进行在线进化设计验证,采用函数级两级进化方法,先进化函数模块,再通过子模块进化电路。

2.1 电路分解与编码方案

首先将 4×2 的乘法器分解为 2×2 的乘法器与 $2 + 2$ 的加法器进行进化,再基于这两种函数功能单元来进化 4×2 乘法器。为了便于二级进化过程中

评估染色体功能和布线,在一级进化时,设置 2×2 乘法器与 $2 + 2$ 加法器有相同的进化规模,例如,在一级进化时,选择 4 行 $\times$ 6 列的演化区域;在二级进化时,选取 2 行 $\times$ 3 列的进化规模。一级编码中功能单元使用 4 种逻辑门: $F1 \& F2$ 、 $F1 \& (\sim F2)$ 、 $F1 \vee F2$ 、 $F1 \vee \sim F2$,而二级编码中功能单元则使用一级进化出的乘法器与加法器。分级编码方法如图4所示,在一级编码中染色体编码代表逻辑门级,而在二级编码中染色体编码代表函数级。

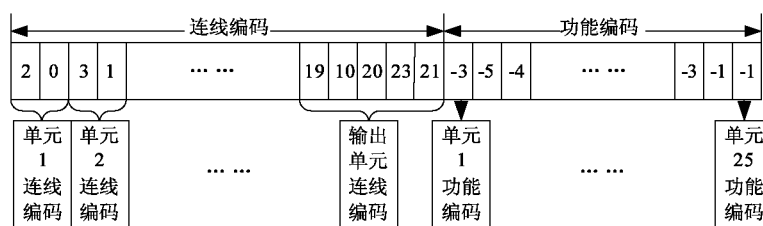


图4 染色体编码方案

2.2 实验结果分析

“生长型”函数级进化与非“生长型”函数级进化的结果统计如表1所示。

表1 “生长型”函数级进化与非“生长型”函数级进化的实验结果分析

进化策略	收敛代数			进化时间/(min)			成功率
	最大	平均	最小	最大	平均	最小	
“生长”进化1	112	62	18	12	8.1	4	65%
“生长”进化2	454	152	12	41	17	2	93%
非“生长”进化	1339	662	110	178	89	14	90%

由表1可知,由于优良基因位保护的原因,使得“生长”进化1的“早熟”变得更加明显,因此,“生长”进化1的成功率相比非“生长型”的成功率要低,但是“生长”进化的收敛代数与收敛时间明显比非“生长”进化要快。本文利用在进化过程中引入二次初始化技术来提高“生长”函数级进化的成功率,其结果如表1中的“生长”进化2所示,从表中可以看出,通过改进后的成功率明显提高,而收敛速度仍然比非“生长”进化的快5倍多。

三种方法的收敛曲线如图5所示。从图5中可

以看出,“生长”进化的收敛速度明显快于非“生长”进化,图 5 中“生长”进化 2 的曲线取的是有二次初始化的一种结果,由于二次初始化是针对“生长”进化 1 中出现“早熟”现象的改进,因此未出现“早熟”的曲线与“生长”进化 1 相同,此处没有示出。

采用本文方法进化出的电路结构如图 6 所示,其 4×2 的乘法器由 2 个 2 位乘法器与 3 个 2 位的加法器构成,子模块 2 位乘法器电路结构如图 7 所示,2 位加法器电路结构如图 8 所示。由于加速进化仅仅是提高进化的速度,并未优化电路,因此此处只给出了一种电路结构。

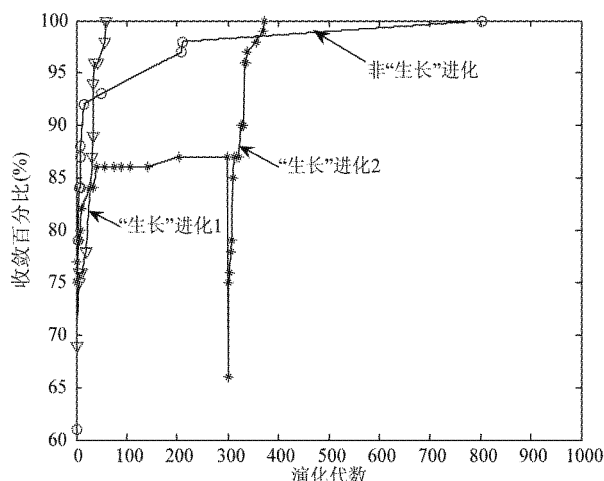


图 5 两种函数级进化策略得到收敛速度比较

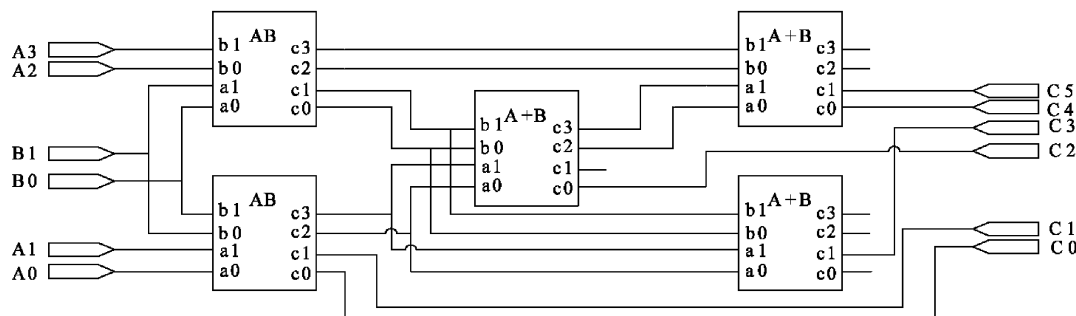


图 6 4×2 乘法器结构图

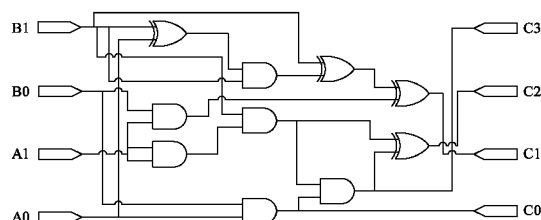


图 7 2 位乘法器电路结构

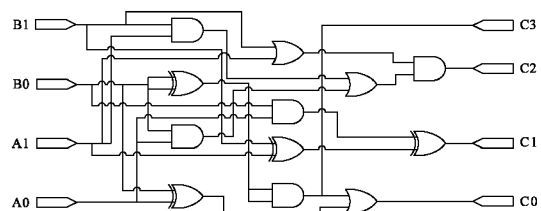


图 8 2 位加法器电路结构

3 结论

本文研究了数字演化硬件在线进化技术,采用了分级函数级在线进化方法,提出了一种加速进化策略。相比非“生长”进化,新方法不仅使得较大规模电路的进化速度提高了 5 倍多,而且使成功率也

略有提高。然而,进化成功率仍然是一个可以改善的方面。另外,进化得到的电路结构可能不是速度、面积等最优的,因此可以进一步研究函数级电路在线进化的多目标优化问题。

参考文献

- [1] Zhang Y, Smith S L, Tyrrell A M. Digital Circuit design using intrinsic evolvable hardware. In: Proceedings of the 6th NASA/DoD Evolvable Hardware Workshop, Seattle, USA, 2004.55-62
- [2] 王友仁,姚睿,朱开阳等. 仿生硬件理论与技术的研究现状与发展趋势分析. 中国科学基金, 2004, 5: 273-277
- [3] Yao X, Higuchi T. Promises and challenges of evolvable hardware. *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, 1999, 29(1): 87-97
- [4] Lee J, Sitte J. Gate-level morphogenetic evolvable hardware for scalability and adaptation on FPGAs. In: Proceedings of the 1st NASA/ESA Conference on Adaptive Hardware and Systems, Istanbul, Turkey, 2006.145-152
- [5] Leitao P. A bio-inspired solution for manufacturing control systems. In: Proceedings of IFIP International Federation for

- Information Technology for Balanced Automation Systems. US, Boston: Springer, 2008. 303-314
- [6] Gajda P, Krawiec K. Evolving a vision-driven robot controller for real-world indoor navigation. In: Proceedings of Lecture Notes in Computer Science, Applications of Evolutionary Computing, Heidelberg, Germany. Berlin: Springer, 2008. 184-193
- [7] Zykov V, Mytilinaios E, Desnoyer M, et al. Evolved and Designed Self-Reproducing Modular Robotics. *IEEE Transactions on Robotics and Automation*, 2007, 23(2):308-319
- [8] Torresen J. A dynamic fitness function applied to improve the generalisation when evolving a signal processing hardware architecture. In: Proceedings of the Applications of Evolutionary Computing on EvoWorkshops, London, UK, 2002. 267-279
- [9] Burian A, Takala J. Evolved gate arrays for image restoration. In: Proceedings of Congress on Evolutionary Computation, New Jersey, US, 2004. 1185-1192
- [10] Murakawa M, Yoshizawa S, Kajitanni I, et al. Hardware evolution at function level. In: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, London, UK, 1996. 62-71
- [11] 乔双. 函数级硬件进化. 小型微型计算机系统, 2001, 22(11):1406-1408
- [12] Liu W X, Murakawa M, Higuchi T. ATM cell scheduling by function level evolvable hardware. In: Proceedings of the 1st International Conference on Evolvable Systems, London, UK, 1996. 180-192
- [13] Gao G J, Wang Y R, Cui J, et al. Research on multi-objective on-line evolution technology of digital circuit based on FPGA model. In: Proceedings of the 7th International Conference on Evolvable Systems, Wuhan, China, 2007. 67-76

Research on the technology for on-line evolution of digital hardware at function level

Ping Jianjun, Wang Youren, Gao Guijun, Kong Deming, Yao Rui, Zhang Zhai

(College of Automation and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016)

Abstract

The paper presents a new strategy for effective evolution of digital hardware by using the on-line adaptive evolution of digital circuits at function level based on the field programmable gate array (FPGA) model. The strategy firstly divides the target digital circuit into several separate sub-units according to its functions, then makes on-line evolutionary design of every sub-unit, and lastly, on-line evolution of the desired circuit was implemented based on the designed sub-units. Thus, the length of chromosome can be reduced and the evolutionary convergence speed can be raised, compared with the circumstance when the gate-level evolutionary method is applied to design of large-scale digital circuits. The on-line evolutionary design of a 4×2 multiplier was performed as an example, and the experimental results showed that the digital target circuit was successfully on-line evolved and the convergence speed was increased by about 5 times.

Key words: evolvable hardware, digital circuit, on-line evolution, function level evolution