

三进制编码实现硬件演化方法研究

张峻宾, 蔡金燕, 李丹阳, 潘 刚

(军械工程学院 光学与电子工程系, 河北 石家庄 050003)

摘 要: 针对硬件演化在演化算法中采用定长染色体编码结构位串, 其编码过长会造成演化速度慢、成本高等缺点, 提出基于三进制编码实现演化算法的方法. 利用 Matlab 矩阵运算的优势, 采用遗传算法实现演化算法, 找到最优电路编码矩阵, 得到了最佳电路函数表达式. 实例反映了三进制编码比定长染色体编码规模小, 遗传算法中采用多点定向变异具有收敛速度快等优点, 证明了三进制编码演化算法的有效性.

关键词: 硬件演化; 演化算法; 遗传算法; 故障诊断

中图分类号: TP301.6

文献标识码: A

文章编号: 1000-7180(2013)03-0001-04

Research of Ternary Code Implemented in Hardware Evolution

ZHANG Jun-bin, CAI Jin-yan, LI Dan-yang, PAN Gang

(Department of Electronic and Optical Engineering, Ordnance Engineering College, Shijiazhuang 050003, China)

Abstract: Evolvable arithmetic uses fixed length chromosome code which can decide configuration in traditional evolvable hardware, its long code will led to low evolvable rate and high cost. An evolvable arithmetic which based on ternary code was advanced, these disadvantages can be solved by it. Using the matrix operation predominance of Matlab and the traditional GA, which can implement evolvable arithmetic, and find optimal circuit encode matrix, and get the optimal circuit function expression. The instance reflects many merits, which including the ternary code has small dimensions than fixed length chromosome code, fast constrigent was led by multi-element directional variation, the validity of ternary code was proved.

Key words: hardware evolution; evolvable arithmetic; GA; fault diagnose

1 引言

随着信息科学技术的发展, 电子装备集成度越来越高, 功能更加完善. 伴随 FPGA 快速发展, 针对数字系统的故障诊断, 硬件演化技术逐步在故障诊断和自修复过程中有很好的表现^[1].

现阶段可编程逻辑器件已经得到了很大程度的发展, 硬件演化的研究的重点放在了演化算法方面^[2]. 有效进行编码是演化算法实现的基础, 也是节约成本的有效途径.

2 EHW 基本原理

硬件演化的原理可简单表示为:

$$EA + PLD = EHW$$

即: 演化算法 + 可编程逻辑器件 = 硬件演化

现场可编程门阵列(FPGA)其结构非常适合演化, 可编程逻辑器件(PLD)具有重构特点, 而二进制位串能够决定其结构, 这一点有利于实现硬件演化^[3-4]. 通过演化算法找出符合要求的染色体, 再对染色体译码得到与之相对应的硬件结构, 即为目标电路, 从而实现硬件演化.

当数字系统发生故障的时候, 需要对系统 LUT 进行自检, 发现是否存在 LUT 冗余单元. 在冗余单元个数多于故障单元个数的情况下, 只需要用冗余单元替换故障单元即可; 当在未发现 LUT 冗余单元或冗余单元个数少于故障单元个数的情况下, 才

收稿日期: 2012-05-12; 修回日期: 2012-07-16

基金项目: 河北省重点基础研究项目(10963529D)

有进行 EHW 的必要^[5].

文献[1,3]与多数文献一样,采用传统二进制制定长染色体编码对每个节点都实施编码.如图1所示可编程逻辑阵列(PLA)电路二进制编码为:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

其编码简单通用,但当原始输入端较多时,决定电路所需要的结构位串将很长.较长的结构位串编码需要占用更多的硬件资源,在一定程度上会增加成本.

上面的编码矩阵每一行代表一个 and 线编码,最后一列代表 or 线编码.其中 0 代表交叉点没有互连,1 代表交叉点互连.

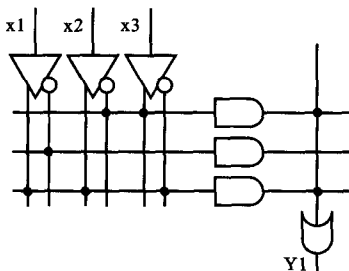


图1 PLA 结构示意图

采用这种电路编码的方式,忽略了只有少数开关决定硬件电路的事实,基因型中许多无用的基因位可以化简.因此提出了基于三进制染色体编码的方案,实现对基因位的有效编码.

3 三进制编码算法

3.1 编码操作

为了减少二进制定长染色体编码的长度,拓展有效进行硬件演化染色体编码途径,提出了基于三进制编码的方案,对 PLA 电路编码只包含“0”、“1”、“2”三个元素,其编码对应含义如表1所示.

表1 编码对应含义

编 码	含 义
0	X_i 输入端为取反输入
1	X_i 输入端为不取反输入
2	X_i 端没有引入输入

例如对图1的编码矩阵为:

$$\begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 2 \\ 1 & 1 & 0 \end{bmatrix}$$

每一列代表一根 and 线,三列代表有三根 and 线.基于此的编码方案,编码矩阵大小缩减为定长染

色体编码的一半,节约了内存资源.同时和 VGA(定长染色体算法)

相比,编码结果一目了然.从编码矩阵就能看出函数表达式中最小项的个数.

所提出的三进制编码每个样本的规模大小在定长染色体编码的基础上减半.对每个编码矩阵进行译码,并作出以下规定:

①编码矩阵的每一列代表一个 and 项,有多少列就可能有多个 and 项.

②如果编码矩阵某一列全为 2,则此 and 项自动消失;如果某两列完全相同,则可以将其中任一列(and 项)自动消除;如果某两行完全一样,则说明这两行对应的输入端等价.

③如果某一行全为 2,则代表其对应的原始输入端不存在于函数表达式中.

3.2 GA 算法操作

3.2.1 种群初始化

为了让算法具有最初的解空间,种群需要初始化,先采用均匀分布的随机数来产生 N 个初始种群^[6].

初始化开始后,随即产生一个编码矩阵,通过对矩阵译码,得出对应的电路逻辑函数,再通过真值表对此逻辑电路进行校验,如果完全满足真值表,代表校验通过,则将此对应的编码矩阵进行保存,这样得到一个样本.如此循环,直到产生 20 个编码矩阵作为初始样本.

3.2.2 适应度函数设计

在优化问题中,适应度函数多为目标函数经过转换得到的.适应度函数的数值称为适应度值,能够反映个体在进化中的适应能力,其值越大,适应能力越强,因而在进化中保留下来的可能性就越大.为了防止陷入局部最优、并避免收敛速度缓慢^[7],个体间的适应度值差别既不能太大也不能太小,所设计的适应度函数需要合理体现出个体间适应能力的差异.

演化电路需要实现的目标就是所需要的基本单元门(and-gate, or-gate)最少,从而节约硬件资源.基于此,算法中将演化需要的基本门个数的倒数作为适应度函数,从而达到对每个个体进行度量.

作为目标表达式 y ,其中包含的“+”代表 or_gate,“*”代表 and_gate.令需要的基本门总数为 gate_all,则目标函数可表示为:

$$\begin{aligned} \text{fitvalue} &= \frac{1}{\text{gate_all}} \\ &= \frac{1}{\sum \text{and_gate} + \sum \text{or_gate}} \end{aligned}$$

3.2.3 选择—交叉—变异操作

从初始种群中随即选择两个样本作为父代,进行交叉,交叉点随机选择,而样本变异能够提高种群的多样性. 本文根据实际需求,随机选择初始种群中的两个样本,算法中采用多点交叉和多点变异,能够提高搜索速度,降低搜索的时间.

计算交叉变异后个体的适应度值,如果比样本中的适应度好,则用交叉变异后的个体父代个体.

3.2.4 停止操作

GA 算法在两种情况下停止迭代:①当演化后的 fitvalue 没有变化且在规定演化次数内的时候,停止迭代,此时演化成功;②当演化的次数超过了规定演化次数的上限时候停止演化,此时演化失败.

当演化成功后,对最佳适应度值反求函数表达式,在对函数表达式进行分解,计算其基本门的总数.

4 实例分析

为了验证所提出的三进制编码方法的有效性,选取了文献[8]的例子作为验证,待演化电路函数表达式为:

$$F=\overline{A}C+ABC+\overline{A}CD+CD$$

其真值表如表 2 所示.

表 2 待演化电路 F 真值表

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

初始化样本数 $N=20$ 个,交叉率 $pc=0.95$,变异率 $pd=0.05$,交叉算子采用两点交叉,变异算子采用 4 点变异,加快了搜索速度,根据所研究对象和编码方案,变异时以较大概率变异为“2”,以较小的

概率变为“0”和“1”.
演化计算前,实现待演化函数表达式,需要使用 6 个 and_gate,3 个 or_gate,共需要 9 个基本单元门.

通过演化计算,允许最大迭代次数为 600 次,其仿真图如图 2 所示.

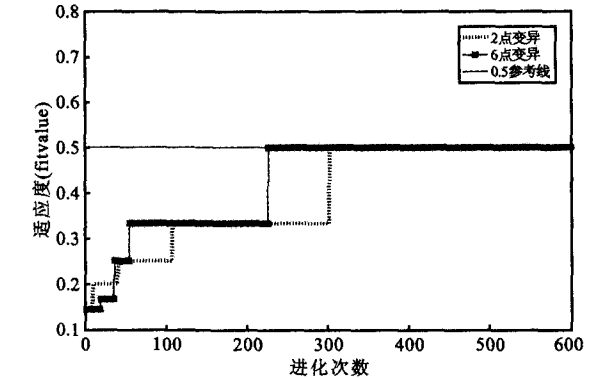


图 2 最佳适应度值 fitvalue 与迭代次数关系

从图 2 中可以看出:变异算子采用 6 点变异比 2 点变异先达到最佳适应度值,6 点变异效果更好. 采用 6 点变异时候,在第 226 次适应度值没有发生变化,其最佳适应度值 fitvalue=0.5,演化后最佳适应度所对应的编码矩阵为:

$$\begin{bmatrix} 1 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 1 & 2 & 2 \\ 2 & 1 & 2 & 2 \end{bmatrix}$$

值得注意的是最佳适应度所对应的编码矩阵不是唯一的,但是它们都有共同的列向量 $[1\ 2\ 2\ 2]^T$ 、 $[2\ 2\ 1\ 1]^T$ 、 $[2\ 2\ 2\ 2]^T$ 和 $[2\ 2\ 2\ 2]^T$,它们都决定了同一个函数表达式.

此矩阵有两列全为“2”,此两列最小项将消除;第 2 行全为“2”,根据定义 3 原始输入端 B 将消除;则其代表其函数表达式将只含有原始输入端 A、C 和 D. 对应的函数表达式为: $F1=A+CD$,与文献[8]中得到的最优函数表达式一样.

实现函数表达式 F1,需要 2 个基本单元门,其中 1 个 or_gate 和 1 个 and_gate. 和待演化的函数表达时相比消除了冗余输入 B,减少了基本门的使用,降低了硬件消耗.

5 结束语

本文对主流的演化算法编码存在的问题进行了分析,针对此提出了基于三进制编码演化算法方案. 利用 Matlab 矩阵运算的优势,相比在 FPGA 上编

程量较小,实现较为简单.与传统的定长编码算法相比,编码规模明显的减半.实例中演化算法采用6点变异,且以定向变异为元素“2”的概率最大,起到了加速演化收敛速度、缩减演化时间的作用.

文献[8]采用了定长染色体编码,编码矩阵规模比本文提出的编码矩阵规模扩大了1倍.三进制编码克服了传统定长编码硬件资源消耗量大的不足.证明了三进制编码实现演化算法的可行性,具有很大的工程应用价值.

参考文献:

- [1] 潘正君,康立山,陈毓屏.演化计算[M].北京:清华大学出版社,1997.
- [2] 康立山,何巍,陈毓屏.用函数型可编程器件实现硬件演化[J].计算机学报,1999,7(22):781-783.
- [3] 刘昌庆,张曦煌.硬件进化中演化算法的研究及应用[J].计算机工程与设计,2008,29(24):6390-6392.
- [4] 杨仕元.数字系统的故障诊断与可靠性设计[M].北京:清华大学出版社,1999.
- [5] 孟庆峰,涂航,李元香.一种外部进化的函数[J].计算机工程,2005,13(31):56-58.
- [6] 许珊娜,祁云嵩.基于遗传算法的基因微阵列数据聚类[J].微电子学与计算机,2012,29(4):123-126.
- [7] 林勇.基于进化型硬件的容错方法研究[D].合肥:中国科学技术大学,2007.
- [8] 张伟.基于进化算法是硬件演化基础研究[D].南京:南京理工大学,2008.

作者简介:



张峻宾 男,(1987—),硕士研究生.研究方向为电子系统性能检测与故障诊断.

蔡金燕 女,(1961—),教授,博士生导师.研究方向为检测理论、故障诊断、可靠性共性技术.