

# 基于位流在线生成的高效演化系统设计方法

易思静

(南京航空航天大学 自动化学院, 江苏 南京 210016)

**摘要:**为了解决演化硬件实现过程中演化粒度较大造成资源浪费以及演化耗时长的问题,提出对 FPGA 资源-配置位流关系进行建模,在线生成位流的设计方法,并给出资源位流信息建模方法与位流在线生成算法。通过演化 2 位乘法器验证设计方法的有效性。结果表明:所提出的方法具有更低的资源代价和更高的演化速度。

**关键词:**FPGA; 演化硬件; 位流在线生成; 位流解析

**中图分类号:**TP391.9   **文献标志码:**B   **文章编号:**1671-5276(2022)05-0107-03

## Efficient Evolutionary System Design Method Based on Bitstream Online Generation

YI Sijing

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

**Abstract:** In order to solve the problem of resource waste and long evolution time caused by large evolution granularity in the process of evolution hardware implementation, a method of modeling the relationship between FPGA resource and bitstream was proposed to generate bitstream online. The relationship modeling of resource-bitstream and the online generation algorithm of bitstream are given. The validity of the design method is verified by the evolution of a 2-bit multiplier. The results show that the proposed method has lower resource cost and higher evolution speed.

**Keywords:** FPGA; evolution hardware; bitstream online generation; bitstream parsing

## 0 引言

演化硬件能够在演化算法的控制下改变自身结构演化得到目标电路。基于 FPGA 的片上可编程系统(system programmable on chip, SPOC)设计可以对硬件资源的配置进行反复修改,在保证电路功能不变的情况下改变自身结构,实现演化。基于动态部分重构(dynamic partial reconfiguration, DPR)的演化方法作为主流演化方法的一种,通过位流文件对模块功能进行配置,实现系统的粗粒度演化<sup>[1]</sup>。由于 DPR 方法演化粒度大多着眼于模块级别,有相当一部分的资源冗余,且进行系统容错时,往往会造成一整块资源被弃用,导致资源浪费<sup>[2]</sup>。因此,通过位流操作进行细粒度的演化方式已逐渐成为研究热点。早期实现位流操作的工具主要有 JBits<sup>[3]</sup>和 GeneticFPGA<sup>[4]</sup>等,这些工具可以根据需求对 FPGA 生成的位流文件直接进行修改。然而,这些工具仅支持部分老旧型号 FPGA,且需要借助于外部 PC 机运行,不利于嵌入式实现。另外,GLETTE K 提出以 LUT 作为演化对象,来提高资源的利用率<sup>[5]</sup>,但其无法对布线进行演化操作。细粒度演化的研究中,大多都只针对逻辑资源或者布线资源中的一环进行,不能做到资源的任意利用;同时,也未摆脱需要借助外部 EDA 工具进行设计所带来的时间消耗问题。

为此,本文提出了对 FPGA 资源-配置位流关系进行建模,并利用所建模型在线生成位流的方法,开发可嵌入式实现的位流在线生成算法。通过操作位流的方式对逻辑功能和布线资源进行细粒度编程,实现电路结构的修改。本文使用该方法设计演化系统,并以演化 2 位乘法器电路进行验证。

辑功能和布线资源进行细粒度编程,实现电路结构的修改。本文使用该方法设计演化系统,并以演化 2 位乘法器电路进行验证。

## 1 FPGA 资源-位流信息建模

### 1.1 FPGA 资源与配置数据帧位置映射建模

配置数据帧囊括了 FPGA 设计过程中所使用到的资源信息,其大小与划分的重构区域大小相关联。FPGA 的最小重构区域包含一行、一列的 Tile 资源,最小重构区域内资源与配置数据帧关系模型如图 1 所示。其对应部分位流包括  $N$  个配置数据帧,前  $R$  帧为布线资源配置数据。逻辑功能的配置信息共计  $2L$  帧+ $M$  帧,其中左右两侧 SLICE 中的逻辑资源分别对应  $L$  帧配置信息。

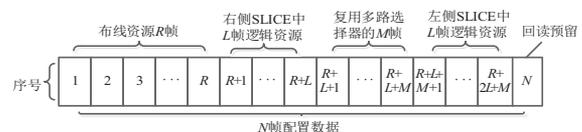


图 1 最小重构区域资源与配置数据帧关系模型

### 1.2 FPGA 资源与配置信息位置关系建模

#### 1) 逻辑资源与配置信息位置关系建模

作者简介:易思静(1996—),女,贵州都匀人,硕士研究生,研究方向为智能仪器及自动化技术。

逻辑功能主要通过 LUT 的真值表变换来实现<sup>[6-7]</sup>, 每个 LUT 的配置信息共有 4 处, 分别位于连续 4 帧内, 且相对帧起始地址的偏移量相同。定义 LUTSTART 表示 LUT 配置信息的基地址, 其值为  $R+1$ ; Col\_offset 为 LUT 所属 CLB 在重构区域的列偏移量; slice 为该 LUT 所属 SLICE 类型(0 代表右侧 SLICE, 1 代表左侧 SLICE); lut 表示该 LUT 的位置(0 代表 A、B LUT; 1 代表 C、D LUT); introffset 表示帧内偏移量。则 LUT 对应配置帧地址为

$$\text{Col\_offset} \times (N-1) + \text{LUTSTART} + \text{slice} \times (L+M)。$$

设每个最小可重构区域包含 CLBsPerRR 个 CLB, 以底部 CLB 坐标为零点, 当垂直方向向上使用第 Y 个 CLB 时, 所在帧内偏移量 introffset 为

$$\text{introffset} = 2 \times Y + \text{lut} \quad (Y < 10),$$

$$\text{introffset} = 2 \times Y + 1 + \text{lut} \quad (10 \leq Y \leq 19)。$$

设一帧内有 Num 个字, 故该 LUT 的第一个配置字的起始地址为

$$\text{Staddr} = (\text{Col\_offset} \times (N-1) + \text{LUTSTART} + \text{slice} \times (L+M)) \times \text{Num} + \text{introffset}$$

## 2) 布线资源与配置信息关系建模

与逻辑资源的配置信息位置建模类似, 定义变量 frame 和 frame\_offset 分别表示布线资源配置信息的起始帧地址和两帧配置数据间的帧偏移量, introffset 代表帧内偏移量。定义变量 index 表示需修改的配置信息起始字, index1、index2 分别为第一处及第二处配置信息字位置。对单段连线进行修改时, 其配置字地址为

$$\text{index} = \text{frame} \times \text{Num}, \text{index1} = \text{index} + \text{introffset},$$

$$\text{index2} = \text{index1} + \text{frame\_offset} \times \text{Num}。$$

由于布线资源的数目太庞大, 无法像逻辑资源那样用统一的函数来描述其配置信息内容<sup>[8]</sup>。因此, 本文通过建立布线资源库来描述其配置信息内容。布线资源库中每一条布线路径均可通过三段转折的形式实现, 即初始段 L1、中间转折段 L2 以及终止段 L3。经由布线约束对电路布线路径进行规划, 使 L2 段与初始段 L1 位于同一列资源。布线资源库包含的信息有: {起始帧位置; 两帧之间的偏移量; 第一帧内的位流数据(32 bit); 第二帧内的位流数据(32 bit)}。

## 2 基于位流在线生成的高效演化系统设计

### 2.1 演化系统的总体结构

演化系统总体结构如图 2 所示。其中, 演化区域作为功能区 IP 核, 可以根据需要配置为不同的功能; MicroBlaze 作为系统控制器, 用于运行演化算法和实现位流在线生成算法; SystemACE 用于控制位流数据传输, HWICAP 用于将其配置到系统功能区; UART 用于实现 FPGA 和 PC 机的通信。

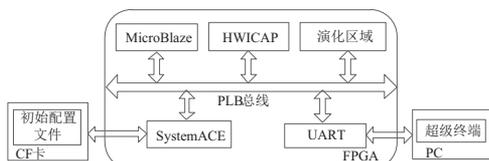


图 2 演化系统总体结构图

### 2.2 演化区域规划与染色体编码方式

为方便演化过程中对资源进行编、解码和实现高效演化, 采用图 3 所示模型描述演化区域。将逻辑资源抽象为 M 行 N 列的 K 输入 1 输出网格节点阵列。整体演化区域由 3 个部分组成, 分别是输入列、演化列以及输出列。其中, 输入列与输出列功能固定, 作为缓冲实现与外部的通信; 中间部分的资源用于生成电路拓扑。演化列内的每一个演化节点都可配置为不同功能。

某一候选电路拓扑所实现的功能由各节点功能和节点之间的连接共同确定, 采用染色体编码的形式进行描述。如图 4 所示, 本文采用多参数级联的 CGP 编码方式, 将整条染色体分成两段。图 4(a) 中 StringA 段的作用是确定不同节点之间的连线关系, 称之为连线编码; 图 4(b) 中 StringB 是用来确定节点所表达的功能, 称之为节点功能编码; 编码示意情况如图 4(c) 所示。

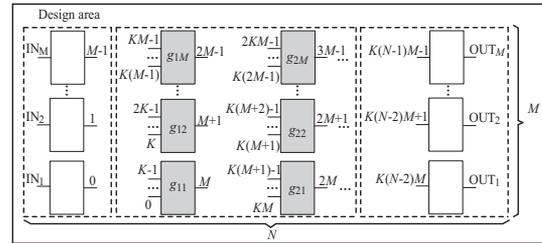


图 3 演化区域结构图



图 4 多参数级联分段编码方法

图 4 中节点功能的编码用负数表示, 节点连线的编码用非负数表示。对连线进行编码时, 先对其输入和输出引脚进行独立编号(图 3), 输入引脚的标号作为染色体的基因位, 将连接到该输入引脚的某功能节点输出引脚标号作为其基因值。

### 2.3 位流在线生成算法设计

使用代表候选电路拓扑的染色体配置演化区域时, 首先要根据图 4 所示编码规则对染色体进行解码, 确定各节点所对应的逻辑资源位置与功能以及节点之间的布线资源连接路径, 然后采用逻辑资源与布线资源位流在线生成算法生成配置位流。

#### 1) 逻辑资源位流在线生成算法

通过对 LUT 功能及其对应配置信息的研究, 本文提出了一种基于最小项表达式的 LUT 功能编程算法。首先将某 LUT 需要实现的功能规范化为最小项表达式, 其位流配置信息可根据式(1)和式(2)进行计算。 $m$  为定义的布尔型数组中的某一位, 变量  $f_1$  为字选择标志,  $i_1$  为位选择标志。

$$f_1 = m \% 8 \quad (1)$$

$$i_1 = m / 4 \quad (2)$$

定义 4 个字符型数据 cf0-cf3 来描述 LUT 功能的位

流数据,其内容确定方法如图5所示。 $k$ 用于确定配置帧。通过计算 $f_i$ 与 $i_i$ 的值,按照不同的位置将修改后的位流插入到配置信息中。

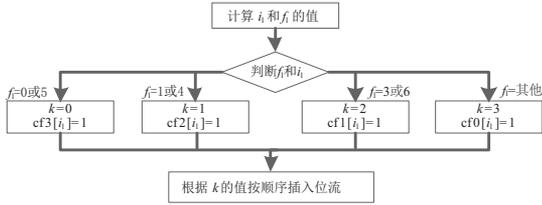


图5 基于最小项的LUT功能编程算法

逻辑资源通过按位插入的方法在线生成位流,逻辑资源位流在线生成算法主要步骤如下。

步骤1:经1.2节可计算出需要修改的初始位流字地址 startaddr。

步骤2:经式(1)、式(2)计算 $f_i$ 、 $i_i$ 的值,并判断 $k$ 的值,定位需将配置信息具体写入哪一处位置中。

步骤3:读取配置数据,保存在以字数为第一维度,字长为第二维度的二维数组 Bitstr[m][n]中。

步骤4:根据变量 $k$ 的不同取值, $k=0$ 时将数组 cf3[ $i_i$ ]内的数值插入LUT配置信息对应4处修改的最后一处,即 Bitstr[Staddr + 3 × Num][ $i_i$ ]处; $k=1$ 时将数组 cf2[ $i_i$ ]内的数值插入倒数第二处,即 Bitstr[startaddr + 2 × Num][ $i_i$ ]处,以此类推,实现配置信息的按位插值。

步骤5:按照最小项布尔型数组的个数进行循环操作,当数组取值为真时进行插值操作,取值为假则继续进行循环。修改完毕后再将配置信息写回原位流文件。

### 2) 布线资源位流在线生成算法

布线资源位流在线生成算法主要步骤如下。

步骤1:计算每条布线路径的列偏移量 Col\_offset 和 introffset 值。定义 L1 段的起始 CLB 坐标为 (sourceCLBrow, sourceCLBcol), L3 段目的 CLB 坐标为 (sinkCLBrow, sinkCLBcol), 则布线路径中 3 条线段的 Col\_offset 计算为 Col\_offset = sinkCLBrow (L3 段), Col\_offset = sourceCLBcol (L1/L2 段)。L1 段及 L2/L3 段的帧内偏移量 introffset 可计算为 introffset = 2 × sourceCLBrow + 1 +

sourceCLBrow / (A/2), introffset = 2 × sinkCLBrow + 1 + sinkCLBrow / (A/2)。

步骤2:读取位流文件并存储到二维数组中。

步骤3:对待修改字进行数值判断并覆盖。首先对待修改第一帧中 index1 处的内容进行判断,自第一位开始循环,若该位为“0”,则使用资源库中对应的帧中第1位内容与之替换;若该位为“1”,则不进行替换,循环次数为该字的长度。对需修改的第二帧中的 index2 处位流重复上述操作。

步骤4:最后把修改后的配置数据帧数组 Bitstr[m][n] 写回原位流文件中。

## 3 演化系统板级验证及结果分析

下面以演化2位乘法器电路为例对基于位流在线生成的高效演化系统进行验证。实验在 Xilinx Virtex-5 ML507 FPGA 上进行。

### 3.1 硬件框架设计

在对2位乘法器演化区域进行设计时,如图6所示,把中间用于电路结构设计LUT之间的连线以及功能去除,仅留下第一列和最后一列作为框架支撑,中间部分的电路通过电路编码来进行自由组合设计。将该结构生成初始位流进行演化操作。

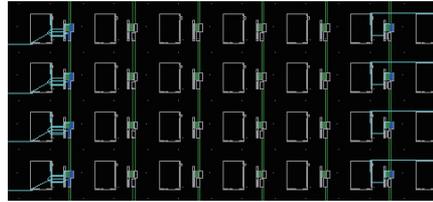


图6 底层电路框架

### 3.2 电路验证

系统演化成功后,电路连接情况以及功能节点的内容如图7所示。按图7中所示连接方式以及节点逻辑功能配置情况进行电路连接,该电路整体即能实现2bit乘法器功能,电路验证成功。

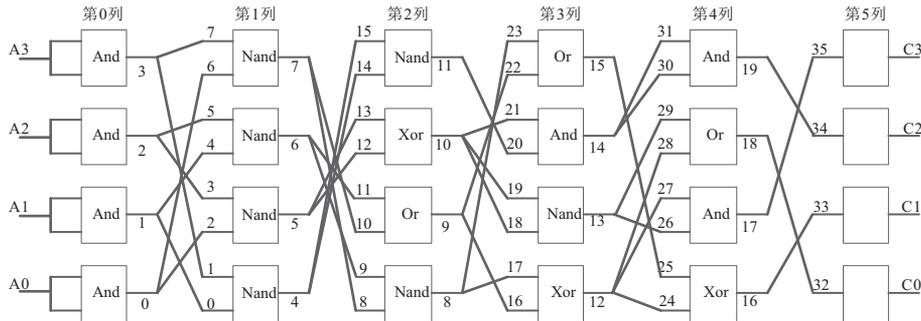


图7 2位乘法器实际电路连接结构

### 3.3 与其他演化方法对比

采用预先生成位流的DPR演化方法实现2位乘法器,需要划分16个重构区域来表示演化列中的功能节点,

至少需要320个CLB来组成重构区域;而本文方法只需24个LUT,其资源消耗比DPR方法减少了99.06%。

(下转第113页)

- [2] 陆凤霞,王涛,朱如鹏. 基于高斯粗糙表面的角接触球轴承微弹流润滑研究[J]. 机械设计与制造,2019,48(6):118-122.
- [3] PLOURABOUÉ F,BOEHM M. Multi-scale roughness transfer in cold metal rolling[J]. Tribology International,1999,32(1):45-57.
- [4] MA B,TIEU A K,LU C, et al. An experimental investigation of steel surface characteristic transfer by cold rolling[J]. Journal of Materials Processing Technology,2002,125/126:657-663.
- [5] ZOU M Q,YU B M,FENG Y J, et al. A Monte Carlo method for simulating fractal surfaces[J]. Physica A:Statistical Mechanics and Its Applications,2007,386(1):176-186.
- [6] WHITEHOUSE D J. The generation of two dimensional random surfaces having a specified function[J]. CIRP Annals,1983,32(1):495-498.
- [7] ZAHOUANI H,VARGIOLU R, LOUBET J L. Fractal models of surface topography and contact mechanics[J]. Mathematical and Computer Modelling,1998,28(4/5/6/7/8):517-534.
- [8] PATRIKAR R M. Modeling and simulation of surface roughness[J]. Applied Surface Science,2004,228(1/2/3/4):213-220.
- [9] WU J J. Simulation of non-Gaussian surfaces with FFT[J]. Tribology International,2004,37(4):339-346.
- [10] MANDELBROT B B. Fractals in physics: squig clusters, diffusions, fractal measures, and the unicuity of fractal dimensionality [J]. Journal of Statistical Physics, 1984, 34(5/6):895-930.
- [11] 原园,成雨,张静. 基于分形的三维粗糙表面弹塑性接触力学模型与试验验证[J]. 工程力学,2018,35(6):209-221.
- [12] 史建成,刘检华,丁晓宇,等. 基于确定性模型金属表面多尺度接触行为研究[J]. 机械工程学报,2017,53(3):111-120.
- [13] YAN W,KOMVOPOULOS K. Contact analysis of elastic-plastic fractal surfaces[J]. Journal of Applied Physics,1998,84(7):3617-3624.
- [14] LIOU J L, LIN J F. A modified fractal microcontact model developed for asperity heights with variable morphology parameters[J]. Wear,2010,268(1/2):133-144.
- [15] 索双富,葛世荣. 磨削加工表面形貌的分形研究[J]. 中国机械工程,1996,7(1):41-42.
- [16] GB/T 131—2006 产品几何技术规范(GPS) 技术产品文件中表面结构的表示法[S]. 北京:中国标准出版社,2007.
- [17] 王润琼,朱立达,朱春霞. 基于域扩展因子和微凸体相互作用的结合面接触刚度模型研究[J]. 机械工程学报,2018,54(19):88-95.

收稿日期:2021-05-27

(上接第109页)

采用将 EDA 工具并入演化环的方法设计 2 位乘法器,产生每条染色体时均需调用 EDA 工具从顶层电路开始进行重新设计。反复的调用操作将会使演化时间大大

拉长。采用该方法与本文方法设计 2 位乘法器时,时间消耗对比如表 1 所示。由表 1 可见,使用 EDA 设计方法产生一个电路个体的配置位流需要 17 s,而本文方法仅需 0.1 s,比 EDA 工具法速度提高了将近 170 倍。

表 1 EDA 设计方法与本文设计方法消耗时间对比

单位:s

方法	单个电路配置时间			演化一代所需时间	整体操作所需时间
	代码综合时间	布局布线时间	位流生成时间		
EDA 设计方法	11	0.53	17	5 726	5 726
本文设计方法		0.1		20.0	383.0

## 4 结语

本文采用基于位流在线生成的 DPR 方法设计了演化系统,给出了演化嵌入式系统的总体结构,通过演化 2 位乘法器验证本方法的有效性。本文所提出方法在演化粒度上达到了 FPGA 板级可操作资源的最小级别 LUT 级,通过操作位流的方式实现对逻辑功能和布线资源的在线编程。实验结果表明:本文方法具有更低的资源代价和更高的演化速度。

### 参考文献:

- [1] 朱萍. 基于快速混合重构的自适应数字系统设计与验证[D]. 南京:南京航空航天大学,2018.
- [2] PEZZAROSSA L,SCHOEBERL M, SPARSØ J. A controller for dynamic partial reconfiguration in FPGA-based real-time systems[C]//2017 IEEE 20th International Symposium on Real-Time Distributed Computing (ISORC). Toronto: IEEE, 2017:92-100.
- [3] SINGH S,JAMES-ROXBY P. Lava and JBits: from HDL to bitstream in seconds[C]//The 9th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'01). California:IEEE,2001:91-100.
- [4] LEVI D,GUCCIONE S A. GeneticFPGA: evolving stable circuits on mainstream FPGA devices [C]//Proceedings of the First NASA/DoD Workshop on Evolvable Hardware. California:IEEE, 1999:12-17.
- [5] GLETTE K,KAUFMANN P. Lookup table partial reconfiguration for an evolvable hardware classifier system [C]//2014 IEEE Congress on Evolutionary Computation (CEC). Beijing: IEEE, 2014:1706-1713.
- [6] ZHANG K F,LU H Z,HU W D, et al. A LUT manipulation based intrinsic evolvable system[J]. IEICE Electronics Express,2014,11(4):1-7.
- [7] DANG PHAM K,HORTA E,KOCH D. BITMAN:a tool and API for FPGA bitstream manipulations [C]//Design, Automation & Test in Europe Conference & Exhibition (DATE). Lausanne: IEEE,2017:894-897.
- [8] BOZZOLI L, STERPONE L. Self rerouting of dynamically reconfigurable SRAM-based FPGAs [C]//2017 NASA/ESA Conference on Adaptive Hardware and Systems (AHS). California:IEEE,2017:77-84.

收稿日期:2021-04-26