

引用格式:肖贞杰,彭斐,徐栋,等. QCA 电路仿真软件的可扩展性框架设计[J]. 微电子学与计算机,2019,36(12):6-10. [XIAO Z J,PENG F,XU D,et al. Design of an extensible framework for QCA circuit simulation software [J]. Microelectronics & Computer,2019,36(12):6-10.]

# QCA 电路仿真软件的可扩展性框架设计

肖贞杰, 彭 斐, 徐 栋, 邝 锐, 解光军  
(合肥工业大学 电子科学与应用物理学院, 安徽 合肥 230009)

**摘 要:** 为了克服 QCA Designer 在设计上的缺陷,提出了一个 QCA 电路仿真软件框架,并在此框架之上对“双稳态”和“相干矢量”算法进行了重新实现. 为了验证该框架的可扩展性,利用该框架实现了一个 QCA 电路功耗分析算法.  
**关键词:** 量子元胞自动机;可扩展性框架;双稳态;相干矢量;功耗分析  
**中图分类号:** TP391.9      **文献标识码:** A      **文章编号:** 1000-7180(2019)12-0006-05

## Design of an extensible framework for QCA circuit simulation software

XIAO Zhen-jie, PENG Fei, XU Dong, KUANG Rui, XIE Guang-jun

(School of Electronic Science and Applied Physics, Hefei University of Technology, Hefei 230009, China)

**Abstract:** To overcome the shortcoming, a QCA circuit simulation software framework has been implemented, upon which bistable and coherence vector algorithms are reimplemented. Besides, a QCA power analysis algorithm is implemented using this framework, demonstrating its well extensibility.  
**Key words:** quantum-dot cellular automata(QCA); extensible framework; bistable; coherence vector; poweranalysis

### 1 引言

由于 CMOS 技术已经接近其物理极限,可靠性和功耗问题正日益凸显. 为了克服这些问题进而实现集成度的持续增加,近年来涌现出了大量新型器件<sup>[1]</sup>,由 Lent 和 Tougaw<sup>[2]</sup>提出的量子元胞自动机 QCA(Quantum-dot Cellular Automata)正是其中一种非常有前景的纳米器件. QCA 元胞具有双稳态特性,可以实现信息的二进制编码;元胞之间通过库仑力的相互作用进行信息的处理和传输;元胞的不同排布方式可以完成各种逻辑操作. 因此其具有实现完整的数字系统的能力. 由于 QCA 元胞具有纳米尺寸、超低功耗和超高时钟频率等诸多优良特性,QCA 器件已经成为一个很有

潜力的研究领域<sup>[3-5]</sup>.

目前被学界广泛接受的标准仿真工具是 QCADesigner,其“双稳态(Bistable)”和“相干矢量(Coherence vector)”<sup>[6-10]</sup>仿真引擎是许多其他仿真算法的基础.

由于有着坚实的量子物理学基础,且在模型精度和计算效率之间取得了良好的平衡,这两个算法成为 QCA 仿真中最重要的两个算法. 假设电路中的元胞数量为  $N$ ,则“双稳态算法”和一阶“相干矢量算法”的计算复杂度仅为  $O(N)$ . 如果考虑两点相干性,则相干矢量算法的算法复杂度为  $O(N^2)$ <sup>[11]</sup>. 无论哪种情况都远远低于全量子力学模拟的  $O(2^N)$  复杂度,这就是为什么许多其他仿真算法都是建立在它们算法之上的原因.

虽然 QCA Deisnger 是一个开源工具,但在编程实现方面并没有考虑软件的架构问题,这使得对这些算法进行扩展变得非常困难. 本文提出了一个 QCA 电路仿真软件框架,并在此框架上实现了这两种算法.

2 QCA 电路仿真软件框架

本文提出了一个 QCA 电路仿真软件框架,并以泛型编程技术进行实现. 电路仿真过程被抽象成一个基类,并为不同的仿真算法提供了子类实现.

2.1 目录结构约定

为了规范算法的开发过程,我们对软件框架进行了诸多约定.

在该框架中可以把整个问题分为两大类:第一类与数据结构有关,另一类与算法本身有关. 数据结构问题涉及到三个文件,即“Modle. hpp”、“Grammar. hpp”和“IO. hpp”. 算法方面涉及两个文件,即“Simulation. hpp”和“Application. hpp”.

2.2 类型注册系统

为了增强代码的可复用性,我们利用 C++ 的类型萃取(type traits)技术对框架中不同部分之间的依赖关系进行解耦. 每个仿真算法都与一个选项(option)类相关联,如“class BistableOption”是双稳态算法的选项,这种关联关系正是在萃取类中体现的. 在源程序中,不直接访问“BistableOption”类,而是通过下面示例代码中的方式来访问:

```
struct bistable_engine;
using Option = typename engine_option<bistable_engine>::type;
```

这里的 engine\_option 就是一个类型萃取类.

为了使类型萃取机制能够正确工作,需要提前对 engine\_option 这个模板类进行特化,从而完成对这个关联关系的注册. 这里提供了一组宏定义来实现此功能. 下面代码显示了用于算法选项注册的宏的定义:

```
# define REGISTER_ENGINE_OPTION ( Engine-
Name, OptionClass)\
Template<>\
struct engine_option<EngineName>{\
using type = OptionClass;\
};
```

有了这个宏定义,类型之间关联关系的注册过程就可以用一种更加语义化的方式进行表达. 如下所示:

```
REGISTER_ENGINE_OPTION(bistable_engine, BistableOption)
```

我们不直接使用双稳态类“BistableEngine”,而是使用以下形式来访问某个具体的仿真算法类:

```
using Engine = typename engine<bistable_engine>::type;
```

这种基于类型萃取和宏定义的注册系统大大降低了程序的耦合性.

2.3 通用的遍历策略

在仿真算法中,遍历电路中的所有元胞(Cell)是一种常见的操作,而且不同的场景需要不同的遍历策略. 由于 QCA 电路由不同的层(Layers)组成,所以电路设计(Design)类通常实现为一个嵌套的数据结构,即 Design 是 Layer 的向量,而 Layer 是 Cell 的向量. 如果只是顺序遍历电路中的元胞,那么可以简单地使用 std::vector 本身的接口,但该接口并不直接支持随机化的遍历策略. 为了解决这个问题,需要一个通用的遍历策略. 在 C++ 语言环境中,我们需要定义一个特殊的迭代器,它可以直接访问 Design 类中的元胞,我们将此迭代器类命名为“DesignTraversalIterator”.

“DesignTraveralIterator”类是使用 Boost iterator 库实现的,定义为“iterator\_facede”类的子类. “iterator\_facede”类所需满足的接口很简单,唯一的困难在于从一维的索引号  $i$  (即 Design 中的第  $i$  个元胞)映射到二维索引号  $(j, k)$  (即第  $j$  层中的第  $k$  个元胞),实现方法如下:

```
Input: An one dimensional index i //一维索引号 i
Output: Pair of two dimensional index (j, k) //二维索引号(j, k)
j ← 0;
k ← 0;
foreach layer of design do
  if i < layer.size then
    k ← i;
    break;
  else
    i ← i - layer.size;
    j ← j + 1;
  end
end
```

除了迭代器类之外,还提供了两个能够实现不同遍历策略的模板函数,分别实现了顺序访问和随机访问两种不同的遍历方式,这两个辅助函数以及迭代器类构成了 QCA 电路的通用遍历策略.

## 2.4 QCA 时钟方案

如前所述,时钟用于控制 QCA 元胞的隧穿能量,所以隧穿能量的波形与时钟的波形是同步的.在仿真框架中,时钟方案由生成器类“GammarGenerator”生成,该类提供了“operator()”操作符的重载.生成器类的作用在于将程序所需要的输入数据和产生这些输入数据的方式进行解耦合.仿真算法需要的只是输入数据本身,而并不关心这些数据的产生方式.而在不同的时钟方案下,时钟数据的生成方式必然会有所区别.通过引入生成器类,也就提供了一个对不同的时钟方案进行扩展的解决方案.下面显示了标准 4 相位时钟方案的实现代码:输入参数  $i$  表示第  $i$  个时钟区域,范围从 0 到 3;输入参数  $j$  表示第  $j$  个采样点;输出值表示第  $i$  个时钟区与第  $j$  个采样点的隧穿能量.“GammarGenerator”类的生成波形是一个带有  $i \times \pi/2$  相移的  $\cos$  函数.

```
template<class Engine> class GammarGenerator {
public:
    GammarGenerator(engine_option_t<Engine> const
&opt)
        : engine_opt(opt){}
    double operator(std::size_t i, std::size_t j) const {
        return engine_opt.amplitude *
            std::cos(2 * Pi * cycle_n/sample_n * j - Pi/2 * i)
            + engine_opt.shift;
    }
private:
    engine_option_t<Engine> &engine_opt;
};
```

当需要其他的时钟方案时,只需要提供一个定义了“operator()”并遵循相同调用形式的新的生成器类即可.

## 2.5 输入生成器

输入生成器的思想与时钟生成器是相同的,但有一个小的区别,这是由于仿真过程可以在两种不同的模式下运行.一种是固定模式,在该模式下,所有可能的输入组合都将被仿真;另一个为选择模式,在这种模式下,用户必须提供一组测试向量用于仿真.仿真模式本身是一个算法选项.

## 2.6 基类框架

虽然每种算法都有其独特的底层数学模型,但从电路仿真过程中可以抽象出一个通用的模式,这个模式也就是通用仿真框架的主体.我们用“SimulationEngine”类对其进行了封装,其实现技术使用

了 CRTP (Curiously Recurring Template Pattern)<sup>[12]</sup> 设计模式,这种 C++ 特有的设计模式是构建软件框架的理想选择.由于它是基于静态类型转换和静态函数转发的,因此其性能通常要优于基于虚函数的面向对象方案.作为抽象仿真框架,它只实现了对仿真过程本身的抽象,具体的仿真行为需要通过子类来提供.类似于面向对象,这也需要我们对某些特定函数进行定制,从而实现行为的变化.

图 1 显示了“SimulationEngine”类中“run\_simulation”算法的流程.仿真开始时,对 QCA 电路的数据结构、输入数据等都进行了初始化;然后是一个可选的 QCA 电路稳态(steady state)计算步骤;接下来,利用 for 循环计算每个时钟采样点;当循环结束时,仿真结果将以适当的文件格式输出.

在所有的仿真算法中,“电路初始化”计算阶段是常用的操作,并直接在 CRTP 基类中实现;“计算电路稳态”及“计算当前样点”计算阶段,它在每种仿真算法中都是不同的,所以它们的实现在相应的派生类中定义.

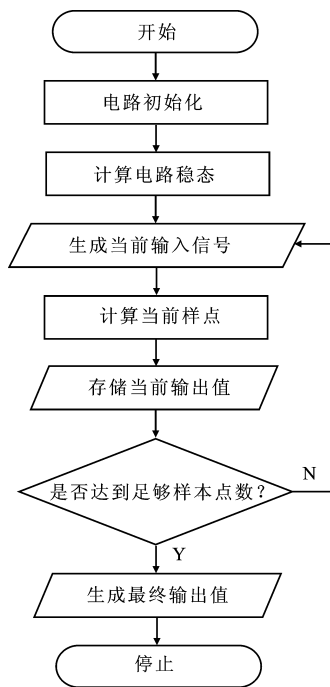


图 1 抽象 QCA 电路仿真过程

## 3 仿真软件框架的可扩展性及实际案例

### 3.1 仿真软件框架的可扩展性

如前所述,每个特定的算法都伴随着一个选项类,这个选项类可以通过相应的类型萃取类来访问.选项类的数据具有不同的来源,一些表示 QCA 器

件中的物理参数,如材料常数,而另一些表示用户的需求,如样本数量.通过综合考虑所有这些因素,框架获得了非常灵活的配置功能.

在不同的算法中,仿真过程的某些方面可能会发生改变.如在文献[13,14]中,定义了一些新的时钟方案;在文献[10]中具有不同的电路驱动方式.在QCADesigner的原始实现中,需要修改源代码来适应这些行为的变化,但是在提出的框架中,这些行为的变化可以通过提供新的类型(例如不同的生成器类型)完成相关功能.

此外,仿真过程本身是可扩展的.在理想情况下,每当提出一个新的仿真策略时,惟一需要做的工作就是子类化这个框架基类,并在这个基类中重新实现接口.

### 3.2 可扩展性案例——QCA 电路功耗分析算法

该仿真框架的真正强大之处在于它能够加速新算法的开发,以文献[12]中的功耗分析算法为例,下面演示了利用该仿真框架实现此算法的方法.要实现此算法,只需要定义四个新类.

第一个类是一个新的选项类.为了计算时钟周期中所释放的能量,除了相干矢量算法的原始参数外,还必须在选项类中加入一些额外的参数,因此该选项类可以简单的从“CoherenceOption”类派生得到.

第二个类是一个新的时钟方案生成器类.根据<sup>[12]</sup>中的要求,该类需要具有高斯斜率的时钟信号,所以需要提供一个名为“GuassianGammarGenerator”的新生成器类.

输入元胞的极化值跟时钟信号的相位相关联,其在整个时钟周期内将不再是恒定不变的,因此需要提供了一个名为“PhasedInputGenerator”的新输入生成器类.

最后一个类即仿真算法类本身,其大部分代码来自“CoherenceVectorSimulationEngine”类,只需重载“run\_simulation”函数来实现功耗计算.

在所有这些类都准备好之后,注册系统会将它们合并到仿真框架中,所以没有必要重写整个软件.

### 3.3 实验结果

功耗分析算法的实验结果如表 1 所示,表中电路与文献[10]中相同,仿真能量耗散的计算误差接近 1%,与文献[10]中相同.由于本算法基于仿真软件框架而来,不需要重写整个软件,提高了工作效率,从而验证了该仿真软件框架的可扩展性.

## 4 结束语

由于 QCA 电路的仿真算法研究尚处于起步阶段,一个具有良好扩展性的软件框架将加快相关领域研究和工具软件的开发.本文提出了一种用于 QCA 电路的仿真软件框架,利用该框架实现了双稳态和相干矢量仿真算法,实验结果表明该方案具有更高的性能.为了验证该框架的可扩展性,在其基础上实现了一个功耗分析算法,结果表明该方案具有更好的可扩展性.

表 1 QCA 连接线和基础逻辑门的功耗分析

| 电路名称     | 环境耗散 (eV) | 时钟 (eV)  | 输入输出 (eV) | 误差 (eV)   | 误差/环境(%) |
|----------|-----------|----------|-----------|-----------|----------|
| W4-1     | 1.452E-3  | 4.754E-5 | -1.514E-3 | -1.447E-5 | 0.996    |
| W8-2     | 2.762E-4  | 1.117E-3 | -1.396E-3 | -2.459E-6 | 0.890    |
| W16-2    | 4.832E-4  | 9.081E-4 | -1.395E-3 | -3.832E-6 | 0.783    |
| W16-4    | 3.916E-4  | 9.055E-4 | -1.300E-3 | -2.967E-6 | 0.758    |
| inverter | 5.276E-4  | 2.782E-4 | -8.108E-4 | -4.920E-6 | 0.932    |
| majority | 2.024E-3  | 2.626E-4 | -2.308E-3 | -2.207E-5 | 1.091    |
| and      | 1.927E-3  | 1.846E-3 | -2.130E-3 | -1.843E-5 | 0.956    |
| nand     | 1.598E-3  | 3.034E-4 | -1.916E-3 | -1.473E-5 | 0.922    |
| or       | 1.916E-3  | 1.241E-4 | -2.059E-3 | -1.892E-5 | 0.988    |
| nor      | 1.605E-3  | 2.344E-4 | -1.855E-3 | -1.531E-5 | 0.954    |

### 参考文献:

[1] CAVIN R K, LUGLI P, ZHIRNOV V V. Science and engineering beyond moore’s law[J]. Proceedings of the IEEE, 2012, 100 (Special Centennial Issue): 1720-1749.

[2] LENT C S, TOUGAW P D. A device architecture for computing with quantum dots[J]. Proceedings of the IEEE, 1997, 85(4):541-557.

[3] AMLANI I, ORLOV A O, TOTH G, et al. Digital logic gate using quantum-dot cellular automata [J]. Science, 1999, 284(5412):289-291.

[4] BLAIR E P, LENT C S. An architecture for molecular computing using quantum-dot cellular automata[C]// IEEE-NANO Third IEEE Conference on Nanotechnology. San Francisco CA: IEEE, 2003: 402-405.

[5] PUDI V, SRIDHARAN K. A bit-serial pipelined architecture for high-performance DHT computation in quantum-dot cellular automata[J]. IEEE Transactions on Very Large Scale Integration Systems, 2015, 23 (10):2352-2356.

[6] WALUS K, DYSART T, JULLIEN G, et al. QCADesigner: A rapid design and simulation tool for

quantum-dot cellular automata[J]. IEEE Transactions on Nanotechnology, 2004, 3(1): 26-31.

[7] TOUGAW P D, LENT C S. Logical devices implemented using quantum cellular automata[J]. Journal of Applied Physics, 1994, 75(3):1818-1825.

[8] TOUGAW P D, LENT C S. Dynamic behavior of quantum cellular automata [J]. Journal of Applied Physics, 1996, 80(8):4722-4736.

[9] TOTH G, LENT C S. Role of correlation in the operation of quantum-dot cellular automata[J]. Journal of Applied Physics, 2001,89(12): 7943-7953.

[10] TORRES F S, WILLE R, NIEMANN P, et al. An energy-aware model for the logic synthesis of quantum-dot cellular automata[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(12):3031-3041.

[11] KARIM F, WALUS K. Efficient simulation of correlated dynamics in quantum-dot cellular automata (QCA) [J]. IEEE Transactions on Nanotechnology, 2014, 13(2):294-307.

[12] COPLIEN J. Curiously recurring template patterns [J]. C++ Report, 1995, 7(2):24-27.

[13] CAMPOS C A T, MARCIANO A L, VILELA NETO O P, et al. USE: A universal, scalable, and efficient clocking scheme for QCA[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2016, 35(3):513-517.

[14] TASKIN B, HONG B. Improving line-based QCA memory cell design through dual phase clocking[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2008, 16(12):1648-1656.

作者简介:

肖贞杰 男,(1992-),硕士研究生. 研究方向为量子电路设计.

彭 斐 男,(1985-),博士研究生. 研究方向为电子设计自动化.

徐 栋 男,(1993-),硕士研究生. 研究方向为数字集成电路设计.

邝 锐 男,(1994-),硕士研究生. 研究方向为量子电路设计.

解光军(通讯作者) 男,(1970-),博士,教授,博士生导师. 研究方向为微纳电路与系统、集成电路设计.

E-mail: gjxie8005@hfut.edu.cn

(上接第 5 页)

[4] NAYAK N G, DÜRR F, ROTHERMEL K. Time-sensitive Software-defined Network (TSSDN) for Real-time Applications[C]// Proceedings of the 24th International Conference on Real-Time Networks and Systems. Brest, France: ACM, 2016:193-202.

[5] 杨吉喆,王玲玲,陆建德. 基于 Octeon 多核网络处理器的 IPv6 联动 IPS 研究与设计 [J]. 微电子学与计算机, 2011, 28 (12): 79-83.

[6] 吴霖. 基于多核网络处理器的 P2P 高速流量识别系统的设计与实现 [D]. 苏州: 苏州大学, 2010.

[7] 基于多核网络处理器的流控技术研究 [D]. 北京: 北京邮电大学, 2014.

[8] IEEE. 802.1Qbv-2015. IEEE Standard for Local and metropolitan area networks -Bridges and Bridged Networks-Amendment 25: Enhancements for Scheduled Traffic [S]. New York, NY, USA: IEEE, 2016.

作者简介:

贾正义 男,(1991-),博士研究生. 研究方向为软件定义网络、时间敏感网络. E-mail: jiazy@dsp.ac.cn

陈 晓 男,(1964-),硕士,研究员. 研究方向可编程网络、软件定义网络.

王劲林 男,(1964-),硕士,研究员. 研究方向为 5G 网络与信息安全、时间敏感网络.

叶晓舟 男,(1980-),博士,研究员. 研究方向 5G 网络与信息安全.