

近似电路的逻辑综合

钱炜慷

上海交通大学密西根学院

关键词：近似计算 近似电路 近似电路逻辑综合算法

近似计算与近似电路

人们观察到，目前许多应用并不需要一个完全准确的计算结果，拥有少量误差的结果也可以被接受^[1]。这些应用包括多媒体、信号处理、模式识别、机器学习和数据挖掘等。例如，在图像处理中，小部分像素值略有偏差不会被人察觉；在基于机器学习的分类应用中，数值计算结果略有偏差不会影响最后分类的准确性。在这一背景下，近似计算被提出，成为计算系统设计中的一个新兴范式。由于应用本身的容错性，并不要求计算系统实现与给定目标完全一致的计算，因此，可以通过牺牲一定的准确性来进一步降低系统的能耗，提升系统的性能。这一设计理念可以被应用于计算系统的各个设计层

次，包括应用算法、编译器、体系结构、电路等层次。

本文主要关注电路层次的近似，对应的电路称为近似电路。图1以2位乘法器为例说明了近似电路的效果^[2]。图1(a)是2位精确乘法器的真值表，其中乘法器的两个输入分别为 a_1a_0 和 b_1b_0 。图1(b)是该2位乘法器的一个电路实现。如果将 $a_1a_0=11$ ， $b_1b_0=11$ 时的输出由正确值1001改为111，而其他输入组合对应的输出不变，那么对应的真值表如图1(c)所示，对应的实现电路如图1(d)所示。可以看到，对乘法器的功能引入略微错误（仅6.25%的输入组合对应的计算结果发生了错误），可以使电路面积和延时都大大减小：电路所用的基本门的个数由8个变为5个，降幅达37.5%；延时由3个基本门的延时降为2个基本门的延时，降幅达33.3%。随着电路面积的大幅减小，电路的功耗也将会大幅降低。因此，通过恰当地引入略微错误，可以大幅降低电路设计关心的三大指标——面积、延时和功耗。因此，近似计算为设计高性能、低功耗的电路开辟了一条全新的路径。

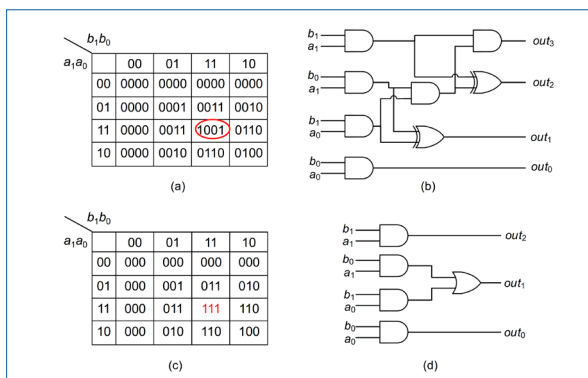


图1 2位精确乘法器及其近似电路。(a) 2位精确乘法器的真值表。其中 a_1a_0 和 b_1b_0 为乘法器的两个输入；(b) 2位精确乘法器的电路实现；(c) 一个2位近似乘法器的真值表。相对于精确乘法器，在 $a_1a_0=11$ 、 $b_1b_0=11$ 时的输出由1001变为了111；(d) 图(c)所示2位近似乘法器的电路实现

近似电路的优势与挑战

对于传统的数字电路设计，电路的面积和延时是一对矛盾体。因此，传统电路设计可以认为是在面积和延时构成的二维平面上寻找最优设计。如图2(a)所示，一旦设计落在了面积-延时帕累托(Pareto)最优前沿上，如果要进一步减小面积，则势必会增加电路延时，反之亦然。而近似电路的

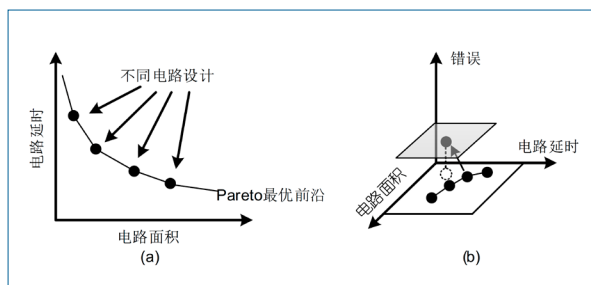


图2 数字电路的设计空间 (a) 传统精确电路的设计空间 (b) 面向可容错应用的近似电路的设计空间

优势在于为电路的设计空间开辟了一个维度——错误，因而大大提升了电路的设计空间。如图2(b)所示，面向可容错应用的近似电路设计是在一个由电路面积、延时和错误构成的三维空间中进行搜索，大大增加了电路设计的搜索范围。因而，极有可能找到不仅使面积降低，而且使延时减小的设计，如图1中的乘法器所示。

然而，近似电路设计面临的一大挑战是如何进行“恰当地”改变，使电路的硬件消耗大幅降低。该挑战存在的主要原因是用户在给定的错误限制下，可行的改变数目巨大。对于一个输入为 n 的电路，存在 2^n 个不同输入组合。假设允许改变不超过 r 个输入组合对应的输出，那么存在的可行改变个数为 $\binom{2^n}{r} = O(2^{nr})$ 。由此可见，可行改变个数与 n 和 r 的乘积呈指数关系。单纯的人为设计仅限于一些小规模的电路。

近似电路的逻辑综合

为解决以上挑战，有必要开发自动综合算法，让计算机自动地搜索最优的改变。这类方法被称为近似逻辑综合，其所考虑的基本问题如图3所示，即针对给定的优化目标，在给定错误约束下，优化某一给定电路，得到相应的近似电路。其中，优化目标包括电路的面积、延时、功耗等；错误约束针对一些具体的错误度量。常用的错误度量包括错误率和误差幅值^[1]。错误率是指导致输出产生错误的输入组合占所有输入组合的比例。误差幅值通常用于对算数运算电路错误的描述。它指的是近似电路

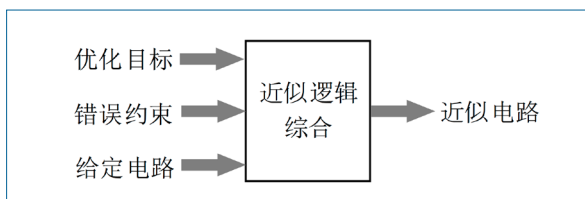


图3 近似逻辑综合问题示意图

输出所表示的二进制数相对正确数值的误差，可分为最大误差幅值与平均误差幅值，分别对应近似电路输出误差的最大值和平均值。以下将介绍近似逻辑综合中常用的一些技术与思想。关于近似逻辑综合更详细的综述，有兴趣的读者可以参考文献[3]。

在一些特殊的情况下，近似逻辑综合可以转换为传统的逻辑综合问题。这里一个代表性工作是SALSA^[4]，适用于最大误差幅值约束下的近似电路综合问题。它首先构建质量约束电路（quality constraint circuit），如图4所示，该电路由原始电路、近似电路和一个校验电路构成。校验电路以原始电路和近似电路的输出为输入，输出为单比特信号 Q 。对于任意输入组合，当且仅当近似电路相对原始电路的输出误差不超过用户给定的上限 K 时， Q 取值为1。由此，为设计最大误差幅值不超过 K 的近似电路，只需使最终质量约束电路输出恒为1。注意到近似电路的输出 PO_{approx} 为质量约束电路的中间节点，根据传统逻辑综合的知识，中间节点存在着所谓的可观性无关项（observability don't cares sets），即一些特别的输入组合，满足在这些输入组合下，中间节点取值发生改变不会影响电路最终的输出。因此，可以通过提取可观性无关项，再结合传统逻辑综合利用无关项来化简电路的方法，得到

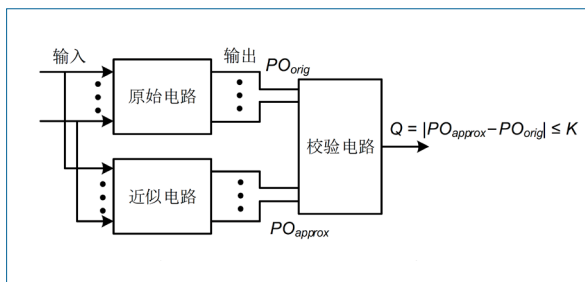


图4 质量约束电路，用于最大误差幅值约束下的近似电路综合

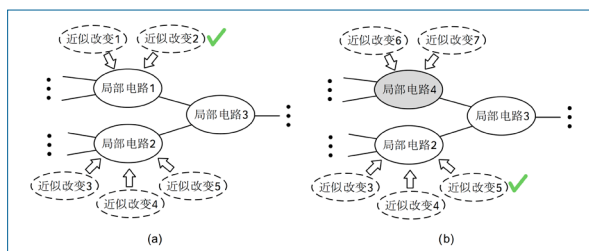


图5 基于局部近似改变的近似逻辑综合方法。(a) 产生局部近似改变并进行挑选 (b) 应用所选择的局部近似改变, 进行下一轮迭代

近似电路。

由于近似电路引入了传统逻辑综合不考虑的新指标——错误, 因此只在有限的场景下可以将近似电路综合问题转换为传统逻辑综合问题。学术界更普遍的做法是开发新的、更具针对性的方法。由于近似电路的设计空间巨大, 现在常用的方法是基于局部近似改变的迭代优化方法, 其主要思想如图5所示。在每一轮迭代中, 先针对各个局部电路, 取得一系列相应的局部近似改变, 然后从中挑选一部分应用于当前电路上。如此迭代, 直至电路的错误接近给定的错误上限。

根据以上基本框架, 现有的基于局部近似改变的近似逻辑综合方法主要围绕两个问题展开: (1) 有哪些有效的局部近似改变; (2) 如何从众多的局部近似改变中挑选合适的改变。

针对第一个问题, 一种最基本的改变是常值替换^[5], 即把一个信号线替换为常数0或1。该方法虽然会引入错误, 但是可以通过删除被替换信号对应的局部独立逻辑锥来减小电路面积, 如图6(a)所示。此外, 由于常值可以进一步向前或者向后传播(例如当与门的一个输入为0时, 其输出也变为常值0, 该常值可以进一步向后传播), 因此可以引起一系列的电路化简。然而, 由于每个信号线只有两种常值变换可选择, 基于常值变换的所有近似改变数目有限。为了提升近似逻辑综合的效果, 有必要增加备选局部近似改变的数目。一个改进是用电路中已有的一个信号替换另一个信号^[6], 如图6(b)所示。这样, 可以在引入一定错误的情况下, 实现被替换信号对应的局部独立逻辑锥的删除。这种方

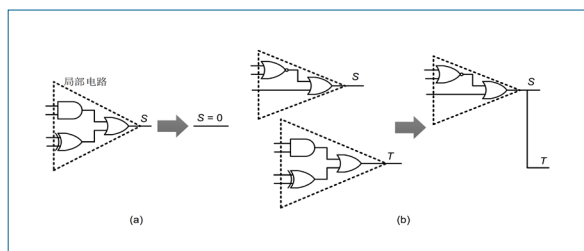


图6 局部近似改变示例。(a) 常值替换 (b) 已有信号替换

法大大增加了局部近似改变的选择。为进一步增加选择, 还可以构造一些当前电路中不存在的信号作为替换信号。例如, 可以借鉴传统逻辑综合中信号重替换(resubstitution)的思想, 使用电路中已有的若干信号, 并结合一些简单的逻辑替换生成另一个信号的原有局部电路^[7]。

针对第二个问题, 不少已有的方法采用贪婪思想, 即在每一轮对所有的局部近似改变进行评分, 然后从中挑选一个评分最高的。在这方面的一个设计细节是具体的评分策略。简单的策略只考虑局部近似改变引起的错误大小, 错误越小则评分越高^[7]。更全面的策略综合考虑局部近似改变引起的错误以及在电路硬件开销上的收益, 例如, 采用电路面积的减少量与引入错误量之比^[6]。

然而, 贪婪挑选方法的问题在于算法效率比较低, 花费大量时间评估每一轮所有可能的局部近似改变, 但最终只挑选一个应用于当前的电路上。为解决这一问题, 一些更高效的挑选策略被提出。例如, 面向错误率的约束可以构建一个背包问题来同时挑选多个局部近似改变, 从而提升算法的效率^[8]。其核心思想是将每个局部近似改变引起的错误视为物品的重量, 将每个改变对应的电路硬件开销的减少量视为物品的价值, 将给定的错误上限视为背包的容量。近似电路的综合可以认为是在给定错误上限的情况下, 挑选最优的局部近似改变的组合, 最大化地降低硬件开销。通过以上映射, 这一问题就自然地转化成经典的背包问题。再如, 当优化目标为电路延时, 可以构建一个网络流问题来同时挑选多个局部近似改变^[9]。为降低延时, 需要找到电路关键路径节点构成的图上的一个割集(cut), 通过

对该割集中的节点同时引入局部近似改变,使所有的关键路径同时缩短,以降低电路延时。由于不同的割集选择及其局部近似改变的选择会引起不同的总错误,为使总错误最小化,相关的选择问题可以转换为一个最小割问题,从而通过网络流算法得到求解。此外,提升效率的另一个方法是以一定的概率来选择每一个局部近似改变^[10]。在这一方法下,每次产生一个局部近似改变后,可以立即决定是否采用它,在一定程度上可避免生成众多选择但只采用一个的低效率问题。

总结与展望

为面向可容错应用设计高能效计算系统,一个潜在的突破方向是设计近似电路。然而,虽然近似电路的设计空间巨大,如何优化设计近似电路却极具挑战。为解决这一挑战,一个可行的技术路线是开发近似电路逻辑综合算法,借助计算机来自动、高效地进行设计空间的探索。现有的多数近似逻辑综合算法都是基于局部近似改变的迭代优化方法,其包含两个核心问题:局部近似改变的设计和挑选。本文针对这两个问题介绍了一些解决方案。

为进一步提升近似逻辑综合算法的性能,未来的探索方向包括:(1)针对多输出局部电路的近似改变。现有的多数局部近似改变只针对单输出的局部电路,而此类局部电路的数量较为有限。为提升可供选择的局部电路的数量,有必要针对更普遍的多输出局部电路,设计相应的局部近似改变方法。(2)对错误进行快速、准确的估计。相对于传统电路设计,错误是有关近似电路设计的一个新指标,对其的准确估计有利于进一步指导近似逻辑综合算法挑选出更好的局部近似改变。然而,评估所有局部近似改变的错误非常耗时。为兼顾近似逻辑综合算法的效率和性能,有必要对如何进行快速、准确的错误估计开展研究。■

致谢:本文由科技部国家重点研发计划课题2020YFB2205501资助。



钱炜慷

CCF 专业会员。上海交通大学密西根学院院长特聘副教授。主要研究方向为面向新兴计算范式的计算机辅助设计和数字电路设计。
qianwk@sjtu.edu.cn

参考文献

- [1] Han J, Orshansky M. Approximate computing: an emerging paradigm for energy-efficient design[C]// Proceedings of the European Test Symposium, 2013: 1-6.
- [2] Kulkarni P, Gupta P, Ercegovic M. Trading accuracy for power with an underdesigned multiplier architecture[C]// Proceedings of the International Conference on VLSI Design, 2011: 346-351.
- [3] Scarabottolo I, Ansaloni G, Constantinides G A, et al. Approximate logic synthesis: A survey[J]. Proceedings of the IEEE, 2021, 108(12): 2195-2213.
- [4] Venkataramani S, Sabne A, Kozhikkottu V, et al. SALSA: Systematic logic synthesis of approximate circuits[C]// Proceedings of the Design Automation Conference, 2012: 796-801.
- [5] Shin D, Gupta S. A new circuit simplification method for error tolerant applications[C] // Proceedings of the Design, Automation, and Test in Europe, 2011: 1-6.
- [6] Venkataramani S, Roy K, Raghunathan A. Substitute-and-simplify: A unified design paradigm for approximate and quality configurable circuits[C]// Proceedings of the Design, Automation, and Test in Europe, 2013: 1367-1372.
- [7] Meng C, Qian W, Mishchenko A. ALSRAC: Approximate logic synthesis by resubstitution with approximate care set[C]// Proceedings of the Design Automation Conference, 2020: 187:1-187:6.
- [8] Wu Y, Qian W. ALFANS: Multilevel approximate logic synthesis framework by approximate node simplification[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020, 39(7): 1470-1483.
- [9] Zhou Z, Yao Y, Huang S, et al. DALs: Delay-driven approximate logic synthesis[C]// Proceedings of the International Conference on Computer-Aided Design, 2018: 86:1-86:7.
- [10] Liu G, Zhang Z. Statistically certified approximate logic synthesis[C]// Proceedings of the International Conference on Computer-Aided Design, 2017: 344-351.