

SpectreX 对 ADLL 的精准快速仿真

张艳维¹, 司强¹, 吕志军²

(1.北京兆芯电子科技有限公司, 北京 100094; 2.北京楷登信息技术有限公司, 北京 100013)

摘要: 2019年, Cadence公司推出了新的全精度仿真器 SpectreX, 它在保持 APS 同等精度的基础上, 能成倍提升仿真速度。先介绍 SpectreX 的简单原理和使用方法, 然后重点介绍如何使用 SpectreX 对 ADLL 进行精准快速仿真以及对其结果的比较与分析。

关键词: ADLL; SpectreX; 仿真; 精度; 速度; 多核

中图分类号: TN402

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.209802

中文引用格式: 张艳维, 司强, 吕志军. SpectreX 对 ADLL 的精准快速仿真[J]. 电子技术应用, 2020, 46(8): 60-63.

英文引用格式: Zhang Yanwei, Si Qiang, Lv Zhijun. Fast and accurate simulation with SpectreX in ADLL[J]. Application of Electronic Technique, 2020, 46(8): 60-63.

Fast and accurate simulation with SpectreX in ADLL

Zhang Yanwei¹, Si Qiang¹, Lv Zhijun²

(1. Beijing Zhaoxin Electronic Technology Co., Ltd., Beijing 100094, China

2. Cadence Design Systems, Inc., Beijing 100013, China)

Abstract: Cadence publicized a new simulator SpectreX which can speed up and keep accuracy in 2019. This article introduces the theory and usage of SpectreX firstly, then focuses on how can use SpectreX to simulate ADLL, compare and analyze the simulation results.

Key words: ADLL; SpectreX; simulate; accuracy; speed; multi-core

0 引言

Finfet 工艺后, 随着工艺尺寸不断缩小, CMOS 器件建模愈加复杂。进入后版图仿真, 寄生效应引起的仿真负荷也愈发巨大。以上两个因素会严重拖累模拟集成电路的仿真速度。

ADLL 电路是现代时钟系统的重要组成部分。对 ADLL 的仿真精度要求也随工艺尺寸缩小而愈加严格。在设计 ADLL 同时, 还需要考虑其他不断增多的物理效应, 最终留给设计工程师的仿真验证时间就非常有限了。

1 关于 ADLL 电路

DLL, 即延迟锁相环^[1], 是基于数字抽样方式, 在输入时钟和反馈时钟之间插入延迟, 使输入时钟和反馈时钟的上升沿一致来实现的。实现方式如图 1 所示。

DLL 就是在输入时钟和反馈时钟间插入延时脉冲, 直到这两个时钟上升沿对齐达到同步, DLL 才能锁定。这样两个时钟没有了差别, 也就补偿了时钟分配网络造成的时间延时, 有效改善了时钟源和负载之间延时。

所谓 ADLL, 即模拟延迟锁相环, 是将输入时钟经过 VCDL 单元进行延时 1T(周期)后反馈到相位检测单元, 经过 CP 后再去调整 VCDL 的偏置电压, 从而达到稳定

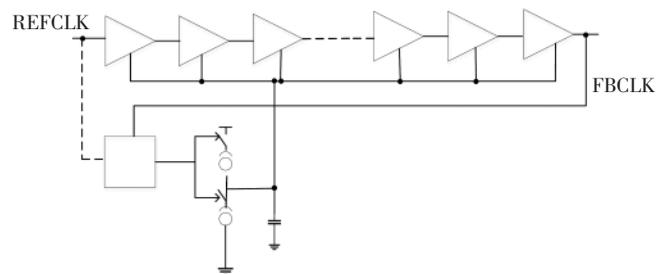


图 1 延迟锁相环的电路结构

延迟 1T 信号的目的。

ADLL 的精度测量指标, 输入时钟和反馈时钟上升沿之间的 jitter 越小越精确。

2 关于 SpectreX

SpectreX^[2]是 Cadence 公司在 2019 年底推出的新一代全精度仿真器。相对于 Cadence 上一代的仿真器 APS, SpectreX 可以在保持同样精度的基础上, 实现 3~10 倍的速度提升。

为什么会有如此明显的性能提升呢? 首先, 在建立电路的矩阵方程上, 针对先进工艺节点的电路, SpectreX 对于 MOS 器件和寄生参数的处理要比 APS 快 1.2 倍;

接下来,在解矩阵方程时,SpectreX 的新算法又比 APS 快 2~3 倍;对于多核 CPU 的更有效利用,又使得 SpectreX 得到 1.5~2 倍的速度提升。

同时,SpectreX 可以仿真的电路规模达到 APS 的 5 倍。这主要是因为采用了改进后的数据处理技术,从而提升内存的利用率,也就提升了对大规模后仿电路的可处理规模。

SpectreX 可以使用更多的 CPU 核,这些 CPU 核可以是单一服务器中的内核,也可以是不同服务器中的内核。

图 2 是 SpectreX 和老一代的仿真器 Spectre Classic 和 APS 的对比。可以看出,对比以前的两个仿真器,SpectreX 在仿真速度、可仿真电路规模和可使用 CPU 核数上都有了明显的提升。

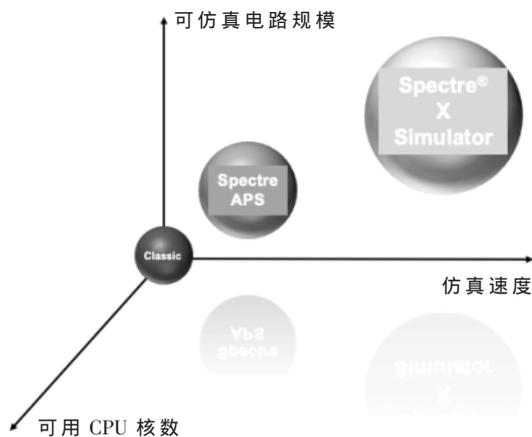


图 2 SpectreX 与 SpectreClassic 和 APS 的对比

3 SpectreX 的使用方法

SpectreX 的使用方法非常简单,只有 5 种设置可选,分别是: Cx、Ax、Mx、Lx 和 Vx。这 5 种设置的适用范围如表 1 所示。

表 1 SpectreX 的 5 种设置

设置	适用范围
Cx	最高精度,用于得到一个最精准结果
Ax	适用于高精度模拟电路仿真
Mx	适用于大多数模拟电路的仿真(默认)
Lx	适用于电源管理电路和其他精度要求不高的电路
Vx	适用于定制数字电路的仿真

图 3 所示为 SpectreX 的 5 种精度和 APS 的 3 种精度(conservative、moderate 和 liberal)在精度(x 轴)和速度(y 轴)的大致对应关系。不难看出,在同样的仿真精度上,SpectreX 的速度始终处于 APS 的上方。

在仿真命令中加入 +preset=cx/ax/mx/lx/vx,就可以选中不同的精度,同时启动 SpectreX,如下就是调用 SpectreX 的 mx 精度进行仿真的例子:

```
spectre +preset=mx adll_sim.sp
```

《电子技术应用》2020 年 第 46 卷 第 8 期

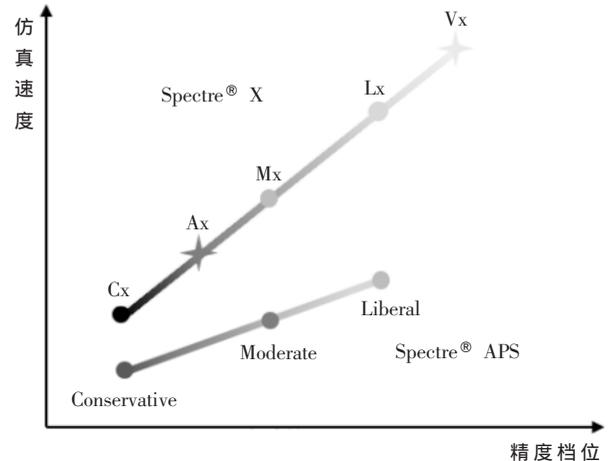


图 3 SpectreX 和 APS 精度设置的对应关系

另外,对于后仿网表,SpectreX 会自动进行寄生约简。寄生约简的精度也由 +preset=cx/ax/mx/lx/vx 决定:在 preset=cx 时,寄生约简的程度是最轻的,约简后网表的精度最高;在 preset=vx 时,寄生约简最为剧烈,约简后网表精度最低,从 Cx 经 Ax、Mx、Lx 到 Vx,约简程度依次增强。如果要取消约简,可以在命令中加入 +postpreset=off,如下:

```
spectre +preset=mx +postpreset=off adll_sim.sp
```

更高级的使用模式,是对于确定的 +preset,寄生的约简可以通过 +postpreset 在 Cx、Ax、Mx、Lx、Vx 中选择,如下命令行所示,就是对寄生约简设为 Cx,对仿真精度设为 Mx。

```
spectre +preset=mx +postpreset=cx adll_sim.sp
```

4 SpectreX 对 VCDL 的初次使用

第一次接触 SpectreX,我们用它仿真 ADLL 的核心 VCDL 电路。这是一个 16 nm 的 VCDL 后仿电路,有 480 万个节点,35 万个晶体管,1 550 万寄生电容,1 090 万寄生电阻。之前用 APS 仿真,需要 7~8 天的仿真时间。使用 SpectreX,稍作调试,用 lx 的精度就能得到比较理想的速度和精度。具体结果如表 2 所示。

表 2 使用 APS 和 SpectreX 对 VCDL 的仿真结果

工具	运行命令	仿真时间	延迟/ps
APS	++aps=mod	8d	3.6
SpectreX	preset=lx	1d15h	3.5

MT=8, 16 nm, 4.8M N, 352K MOS, 15.5M C, 10.9M R

可以看到,在和 APS 报出同样精度(延迟测量)的条件下,SpectreX 的速度是 APS 的 5 倍。在对 VCDL 的仿真中,SpectreX 的易用、快速、准确均表现突出。

5 SpectreX 对 ADLL 的后仿真

ADLL 是锁相环电路的核心部件。对于它的后仿,精度主要是看延迟的大小。延迟由电路设计和寄生阻容共同决定,它是有确定值的。仿真延迟过大,可能仿真精度不够;仿真延迟过小,可能是寄生约简过度。

首先,使用+APS的 moderate 精度进行仿真,其中寄生约简设为 hpa,这对 ADLL 是一个比较精确的设置。仿真得到 delay=-4.2 ps,仿真时间 2 天 20 小时 13 分。

依照上一节对 VCDL 的经验,首先使用 SpectreX 的 preset=lx 进行仿真,发现所得 delay 和 APS 的结果相差超过 20%,精度不合适。

简单作一下调整,将 SpectreX 的精度设为 prese=mx,就可以得到合适的结果:精度为 7.8%,仿真时间为 9 小时 36 分。

探索还没有结束,希望仿真精度能在 7.8% 的基础上进一步缩小,同时仿真时间不要比 9 小时 36 分慢一倍。因为 ADLL 设计重心是时序关系,它的仿真对寄生阻容更加敏感,所以尝试将 SpectreX 的寄生约简+postpreset 设为较高精度,对总体精度设置+preset 适度放松,得到了理想的结果。保持+postpreset=cx;+preset=lx 时,精度达到 5% 以内,仿真时间为 13 小时 51 分;+preset=mx 时,精度能达到 2% 以内,仿真时间为 17 小时 36 分。具体如表 3 所示。

表 3 使用 APS 和 SpectreX 对 ADLL 的仿真结果

工具	运行命令	仿真时间	延迟/ps	精度/1%
APS	+mod hpa	2d20h13min	-4.2	-
SpectreX	preset=lx postpreset=cx	13h51min(5X)	-4.0	-5
SpectreX	preset=mx postpreset=cx	17h36h(4X)	-4.3	2

MT=16, 7 nm, post, 341K N, 44K MOS, 246K C, 910K R

由以上数据可以看出,对于 ADLL,当要仿真速度更快一些时,可以使用 preset=lx, postpreset=cx,会有 5 倍的速度提升,精度 5% 以内;当要求精度更高一些是,使用 preset=mx, postpreset=cx,依然会有 4 倍的速度提升,精度达到 2%~3% 以内。

6 SpectreX 对 ADLLPI 的后仿真

ADLLPI 是以上一节中的 ADLL 电路为核心,添加了更多的接口电路构成的。所以 ADLLPI 的电路规模更大,但它的主要功能精度指标和 ADLL 是一致的。将 ADLL 的仿真设置应用在 ADLLPI 是个自然的想法。将“preset=lx postpreset=cx”和“preset=mx postpreset=cx”分别应用在 ADLLPI 的 SpectreX 仿真中,在保证精度的基础上,得到了预期的性能提升,如表 4 所示。

表 4 使用 APS 和 SpectreX 对 ADLLPI 的仿真结果

工具	运行命令	仿真时间	延迟/ps	精度/1%
APS	+mod noRCR	6d0h13min	-2.5	-
SpectreX	preset=lx postpreset=cx	1d0h36min(6X)	-2.9	13
SpectreX	preset=mx postpreset=cx	2d2h48h(3X)	-2.3	-8

MT=16, 7 nm, post, 405K N, 69K MOS, 404K C, 1.1M R

从以上结果可以看出,“preset=lx postpreset=cx”兼顾速度和精度,能够把仿真时间缩小 6 倍到 1 天;“pre-

set=mx postpreset=cx”精度更好一些,但是速度只比 APS 快了 3 倍。以上结果真的相对误差要比上一节中的 adll 大,这是因为 ADLLPI 中的基准延迟为 -2.5 ps,比 add 中的 -4.2 ps 要小 40%。从绝对误差来看,SpectreX 的仿真精度是满足要求的。

7 对 SpectreX 多核技术的尝试

在第 4 节~第 6 节中,在使用相同的 CPU 核数的条件下,SpectreX 比 APS 有了成倍的速度提升。在此基础上,如果把 SpectreX 使用的 CPU 核数进一步增加到 16 核、32 核甚至 64 核,仿真速度是否会有进一步提升呢?

以第 4 节中的 VCDL 为例,保持 SpectreX 的仿真设置不变,只截取其中一小段电路工作时间,分别使用 CPU 的 8 核、16 核、32 核和 64 核进行仿真,得到如表 5 结果。

表 5 使用 SpectreX 对 VCDL 的多核仿真结果

核数	仿真时间/h	CPU 核利用率/1%	最大内存/GB	步数
8T	4.2	690	19.9	144,768
16T	2.8	1 260	20.2	144,768
32T	1.7	2 160	19.9	144,768
64T	1.4	3 870	19.9	144,768

16 nm, 4.8M N, 352K MOS, 15.5M C, 10.9M R

可以看到,以上四个仿真的步数完全一致,说明四个瞬态仿真截止时间一致;最大内存使用基本一致,说明 CPU 核数的增加没有额外占用多余的内存资源;CPU 核的利用率是指仿真实际所用到的核数,它是随核数而不断增加的,说明多核技术被正常启动;仿真本身所用时间不断减少,对它的具体分析如图 4 所示。

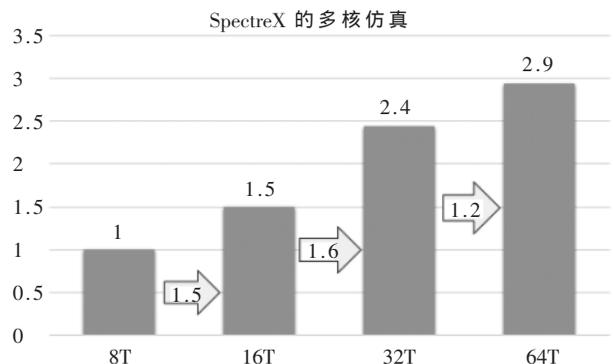


图 4 SpectreX 对 VCDL 的多核提速比较结果

在图 4 中,横坐标分别对应 8 核、16 核、32 核和 64 核仿真;纵坐标是将四个仿真的速度分别归一化处理:8 核的仿真速度设为 1,其他三种仿真速度分别是 1 的相应倍数。

可以看出,随着 CPU 核数的增加,仿真不断加速,最快的 64 核仿真能比 8 核仿真提速 2.9 倍。从 8 核开始,每一次加倍核数,获得的速度提升在 1.2 至 1.6 倍之间。8 核倍增到 16 核,16 核倍增到 32 核,提速都超过了

微电子技术 Microelectronic Technology

CadenceLIVE 2020 中国用户大会优秀论文

1.5 倍,超出预期。32 核提速到 64 核,加速 1.2 倍,也在预期之内,因为:

(1)测试所用的电路规模如表 5 最后一行所示。虽然这个电路规模不小,但其电路规模还没有足够大的可以更好地利用 64 核的资源;

(2)最为关键的,因为仿真用的设置是+preset=lx,这种设置对寄生电阻电容的约简比较剧烈,所以会更多地降低电路规模。如果采用 Mx 或是 Ax 的设置,相信 64 核的提速会更多。

以上都是在 1 台服务器上所做的多核仿真,可以看到 SpectreX 随 CPU 核数的增加,对仿真速度的提升是比较明显的。这种 1 台服务器上达到或超过 32 核的仿真技术,Cadence 称之为 HCC 技术,即 High Core Count 技术。一般适用于大规模模拟电路的仿真。

对一些超大规模电路,如果 HCC 技术还不能满足提速要求,可以用多台服务器去仿同一个电路,从而在 HCC 基础上进一步提升速度,即使用 Cadence 的 XDP 技术,即 X Distributed Processing 技术。目前没有对此技术进行测试。

8 结论

在本文列举的 VCDL、ADLL 和 ADLLPI 的电路仿真中,要达到比较精准的结果,以前都要花上 2~8 天的时间做仿真。有了 SpectreX,可以在保持精度的基础上,把仿真时间提高 4~6 倍,极大提升了工作效率。

对大多数电路,只需调整 preset 的 5 个选项 Cx、Ax、Mx、Lx、Vx,就可以得到比较理想的仿真精度和速度,比如本文中的 VCDL 电路;SpectreX 还提供调整寄生约简的设置参数 postlpreset,它和 preset 有同样的 5 个选

项,默认情况下 postlpreset 取值和 preset 相同,但 postlreset 可以和 preset 设置不同的值。对 ADLL 这样对时序精度要求较高的电路仿真,一般可以把 postlpreset 设得比较高,如 Cx,以保证寄生电路的精度;同时将 preset 的精度降低,如 Lx,从而得到较快的仿真速度。所以 SpectreX 的使用既可以在 5 个选项中简单选择,又提供进一步微调的空间,为不同层次的工程师提供了方便的使用性。

SpectreX 可以在一台服务器上使用达到或超过 32 核进行仿真,实践证明,对于 ADLL(VCDL)电路,随着 CPU 核数的倍增,SpectreX 的提速明显,且提度线性度好。

综上,SpectreX 不仅能提供快速精准的仿真,其使用方式也比较简单、灵活,支持多核仿真对仿真速度的进一步线性提升。SpectreX 不仅极大提升了 ADLL 电路的仿真验证效率,也值得更多工程师去了解使用。

参考文献

- [1] KIM C.Delayed locked loop design issues[C].ISOCC2004 Conference, 2004.
- [2] Spectre classic simulator, Spectre APS, Spectre X, and Spectre XPS User Guide, Product Version 19.1[Z].

(收稿日期:2020-06-22)

作者简介:

张艳维(1982-),女,硕士,主要研究方向:DLL,时钟信号。

司强(1987-),男,硕士,主要研究方向:DDR、高速电路接口。

吕志军(1977-),男,硕士,主要研究方向:Cadence Virtuoso 前端电路设计平台和仿真工具的技术支持。

(上接第 59 页)

同时将该流程应用到规模更大、设计复杂度更高的 CPU 设计中,同样得到了很好的性能优化。最终确定了两款模块芯片均采用 Innovus 机器学习进行延时优化的物理设计流程,实现 setup 和 hold 均比较容易收敛的同时减小功耗的结果。

参考文献

- [1] BISHOP C M.Pattern Recognition and machine learning (information science and statistics)[M].Springer, 2006.
- [2] YU B,PAN D Z,MATSUNAWA T,et al.Machine learning and pattern matching in physical design[C].Proceedings of the IEEE/ACM Asian and South Pacific Design Automation Conference(ASPAC), 2015.
- [3] LI B,FRANZON P D.Machine learning inphysical design[C].Proceedings of the IEEE Conference on Electrical Performance of Electronic Packaging andSystems(EPEPS), 2016.
- [4] Cliff Hou.A smart design paradigm for smart chips[R]. 2017 IEEE International Solid-State Circuits Conference,

2017.

- [5] Bai Lida, Chen Lan.Machine-learning-based early-stage timing prediction in SoC physical design[C].2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology(I CSI CT), 2018.
- [6] 边少鲜, David He, 宋晓琨, 等.基于 Innovus 提升芯片性能的物理实现方法[J].电子技术应用, 2019, 45(8): 48-52.
- [7] 林铭波.超大规模集成电路系统导论[M].北京:中国工信出版集团, 2015.

(收稿日期:2020-06-22)

作者简介:

边少鲜(1991-),女,本科,工程师,主要研究方向:数字后端物理实现。

栾晓琨(1981-),男,硕士,架构师,主要研究方向:数字后端物理实现。

蒋剑锋(1988-),男,硕士,工程师,主要研究方向:数字后端物理实现。

版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所