既快又准并且低开销！一作亲解MICRO 2021最佳论文：一种自动化功耗模拟架构

本文作者：我在思考中 2021-11-22 10:39

导语：APOLLO让每个cycle都能得到一个准确的power。

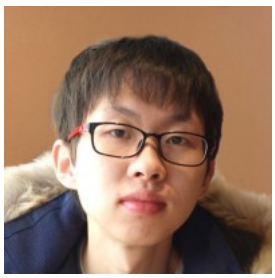
由于摩尔定律效用放缓，在设计芯片时，伴随着性能的提升，功耗也与日俱增。为了更加了解功耗，就要对出现的各种问题进行模拟，而真实模拟代价太大。就在这时，APOLLO应运而生，在芯片设计和运行时期，都能够对功耗进行既快又准确地预测。

作者 | 谢知遥
整理 | 王晔
编辑 | 青暮

第54届IEEE/ACM计算机体系结构顶会MICRO 2021于2021年10月16-20日作为全球在线活动举办。希腊雅典作为主办城市进行转播。

IEEE/ACM 微体系结构国际研讨会（IEEE/ACM International Symposium on Microarchitecture）是介绍和讨论先进计算和通信系统创新微架构思想和技术的主要论坛。本次研讨会汇集了与微架构、编译器、芯片和系统等相关领域的研究人员，就传统微结构主题和新兴研究领域进行技术交流。

来自杜克大学的谢知遥介绍了他们团队的最新工作《 APOLLO: An Automated Power Modeling Framework for Runtime Power Introspection in High-Volume Commercial Microprocessors 》，该论文获得了MICRO2021最佳论文奖（Best Paper Award）。

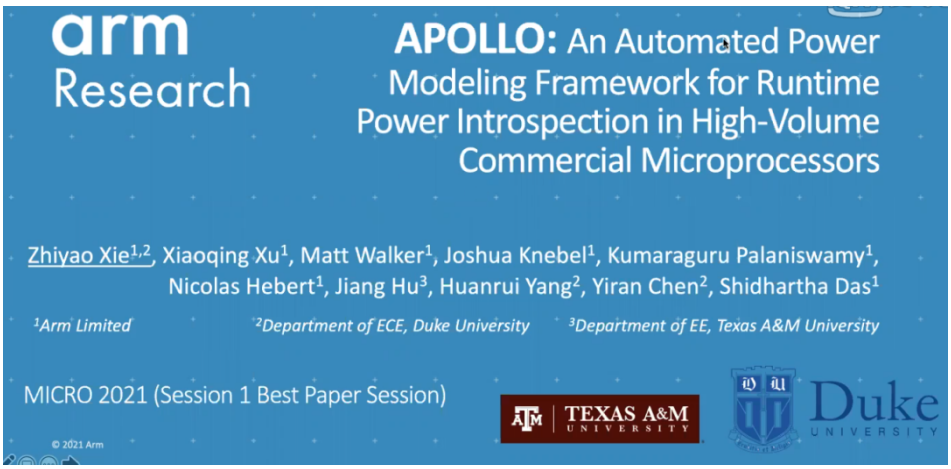


我在思考中
编辑
发私信

当月热门文章
吴恩达：告别，大数据
本科学历马斯克当选美国工程院士！张宏江、萨蒂亚：“我们都美好的未来”
清华博士后用10分钟讲解AlphaCode背后的技术原理，原程序员不是那么容易被取代的！
中国首次！[illegible]佳论文奖，中文获得[illegible]奖」
突发！TensorFlow技术主管皮特登离职，重返斯坦福读博：我在歌“太难了
0
最新文章
谷歌 AI 加入蛋白质解析大军 ProtENN 模型助增 680 万个白质注释词条，登顶 Nature 刊
参数量翻了10倍！Meta AI 祭100亿参数的“新SEER”，为宇宙铺路
普林斯顿邓嘉学生亲述：一定博士学位？不，我本科生也能大厂当应用科学家

谢知遥是杜克大学计算机工程专业的博士生、 致力于EDA/VLSI 设计机器学习算法，擅长机器学习、电子设计自动化、VLSI设计、编程。

他的导师是陈怡然教授。陈怡然教授是杜克大学电子与计算机工程系教授，计算进化智能中心主任，致力于新型存储器及存储系统，机器学习与神经形态计算，以及移动计算系统等方面的研究。



他们的工作APOLLO是针对于现代化的商业CPU或Micro processors所研发的一个自动化的功耗模拟架构（Power-Modeling Framework）。

AI科技评论有幸邀请到谢知遥，为我们亲自解读这篇论文的来龙去脉。

以下，AI科技评论对谢知遥的分享进行了不改变原意的整理：

1

原因及目的

该工作是在CPU设计或运行中所遇到的现实性问题的基础之上进行研究的。

首先第一个也是最大的问题。在CPU设计时期需要对power有更多的了解，而我们现在对power了解是不够的。这取决于设计时的trade off，即权衡或取舍。芯片设计最大的一个trade off是performance and power，即要好的性能，还是要低的功耗。

设计师在设计每一代芯片时都要提升芯片的性能，通常反应在提升IPC或者最大频率等方面。在过去几十年间，因为摩尔定律，性能的提升较为容易。

但由于摩尔定律效用放缓，导致性能提升变得不再那么容易。在这种情况下，设计师就需要在微架构上有更多的创新，但在这个过程中，伴随运行速度的增加，功耗往往也不断增加。

另一方面输电资源（power delivery sources）技术的发展非常缓慢。首先输电线上的电阻很大，导致不能提供足够的power。另外封装技术有限，封装上面的电感（inductance）会导致无法提供所需的快速变化的电流或power。

power和电流通常成正比，因此很难得到一个快速变化的电流。要一瞬间电流突然增大，只能慢慢的增大，不能一瞬间增大那么多。

结合两方面因素，促使我们不仅想要在设计时对功耗有更多的了解，而且在运行中要对power进行管理，而不能出现很多不想要的情况。

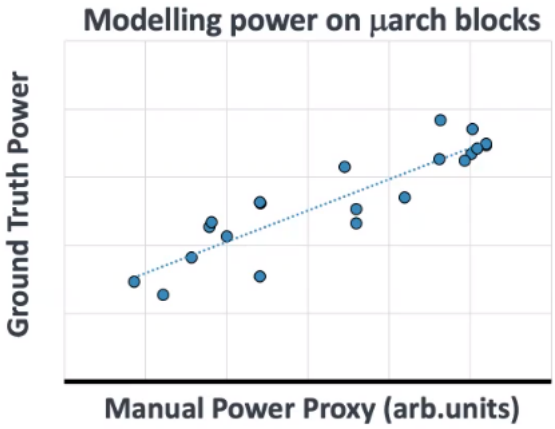
数据集拥有自己的世界观？不其实还是人的世界观

斯隆奖新晋得主宋舒然：从视出发，打造机器人之「眼」

AAAI 2022大奖出炉！中科院州扑克程序AlphaHoldem获论文奖

热门搜索

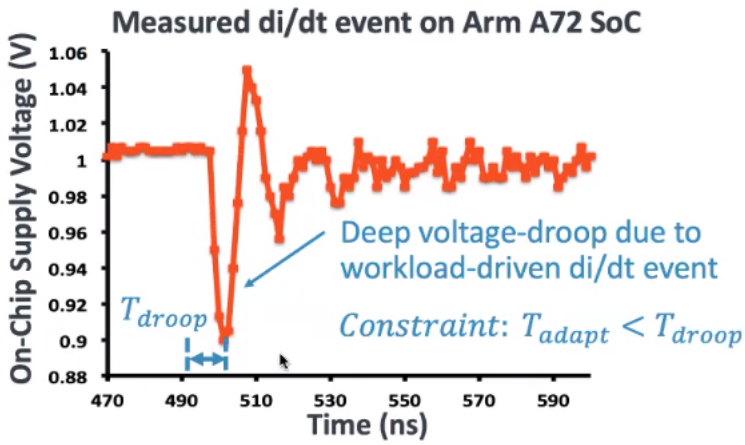
智慧城市 移动应用 发布会 迅雷 小鹏汽车 视频 yahoo 傅盛 数据安全 Alphabet 耳机



运行管理中最常见问题在于peak power mitigation。最大功耗有一个阈值，如果超过了阈值，就需要进行管理，使功耗压降低，否则会出现一系列的问题。管理power的峰值通常要准确实时计算power。在CPU运行时，根据power的计算减少给定CPU的指令，随之功耗就会降低。

但现在在设计CPU时，很多情况下都是人工在芯片上找能够模拟功耗的信号，这种方式不仅困难而且非常不准确。

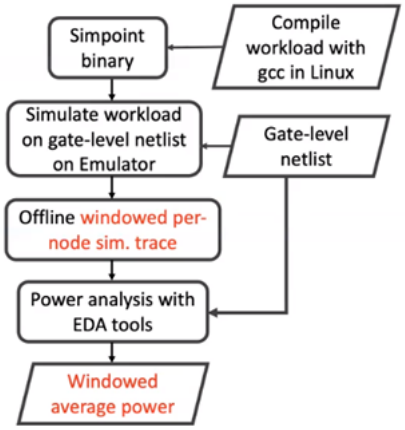
此外，更重要的一个问题是快速电流的变化（或者power的变化）会导致一个很快的电压降叫做voltage-droop。



图注：电流的快速变化导致的电压的急剧变化

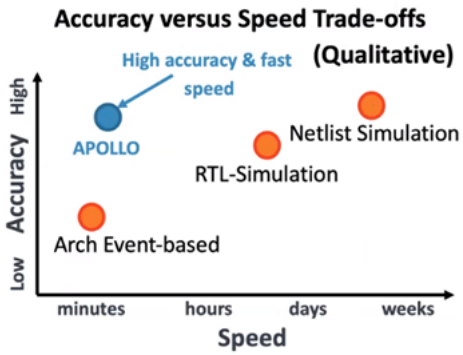
如图所示，起初电压保持不变，假设这一段时间CPU处于睡眠状态，没有执行任何指令。然后突然运行一个很大的程序，此时功耗和电流会突然增大。di/dt(即电流对时间求导)电流的变化量也会变得非常大。此时voltage-droop从1伏变成0.9伏，这会造成很多问题。要避免这个问题也并非容易，由于发生时间非常短暂，因此对应的处理策略也必须要在极短的时间内将其控制住。

既然输电上存在这么多问题，因此在芯片设计时，就要充分模拟芯片CPU上会遇到的各种问题。但如果要做到真实模拟代价是非常大。



工业界标准的 Power模拟流程

上图所示的模式是非常准确的，但可能需要花费几周时间，并且非常昂贵，反复花几周时间进行模拟是非常困难的。即使花费了几周时间，拿到了准确的power，但得到的power是平均power，这中间可能存在几千甚至几百万个周期，一个平均power是不够用的。我们还关心最大power、一瞬间的最大power、快速变化时power的变化等等。



不同类型的power simulation的方法

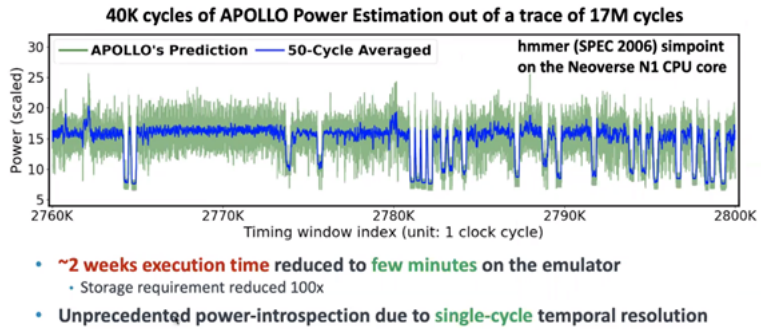
Netlist Simulation是上述介绍的最准确的，但可能需要花费几周时间。APOLLO位于蓝点位置，在保持速度快的同时，准确率很高（虽然不是最准确但准确率可达90%）。

— 2 —

APOLLO优异性质概括

- 首先，它在设计和运行时，都能够对power进行既快又准地预测。在商业化的CPU上能够做到90%~95%的正确率，我们把它在Neoverse N1 CPU上进行实现，我们发现它面积的overhead只有0.2%。
- 其次，对于任何一个设计该模型都可以自动生成。
- 不仅如此，每个cycle都能得到一个准确的power，时间分辨率非常好。
- 而且我们认为APOLLO模型可以延展到更高层次的模拟。

预测结果实例

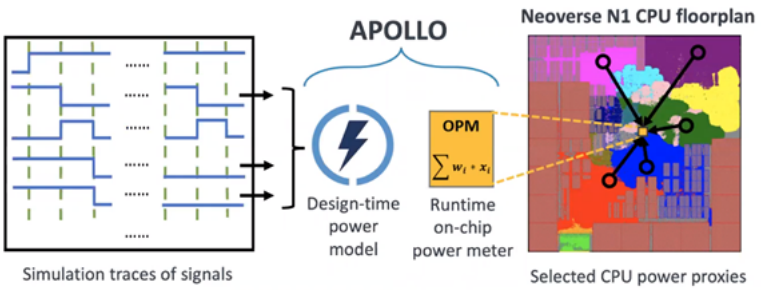


我们在Neoverse N1商业化的CPU上面，运行了一个workload。这个workload非常大，一共有1700万个时钟周期。我们对这1700万个时钟周期的每一个cycle都进行预测，上图展示的是4万个。在工业界用传统的方法可能需要两个星期的时间，而用我们的方法的，几分钟就可以做完。

准确率高、速度快的同时，对存储的要求减少了100倍以上，只需要存我们感兴趣的信号，这也是一个非常大的提升。保持这样的速度、准确度，得到每个周期的power这在之前的工作中几乎是做不到的。

3

APOLLO的组成部分



图注：APOLLO的组成部分

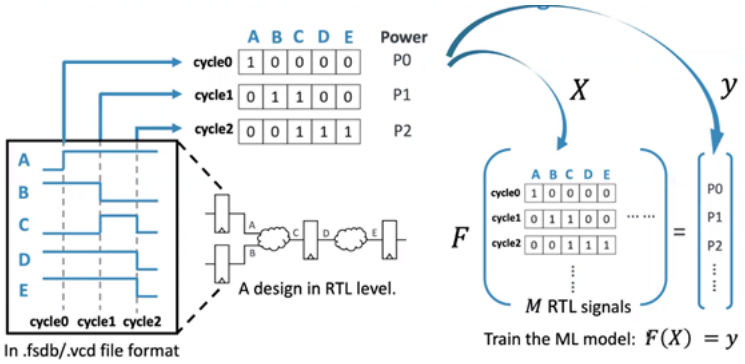
APOLLO由两大部分组成。

在设计时，它是一个又快又准的 power 模型。如图所示假如对信号模拟追踪，所有信号都在不停的运动，根据这些可以得到一个准确的power估计。

在CPU运行时，它就会成为一个片上功率表（on-chip power meter）。我可以直接把它做的到CPU里面变成CPU的一个模块，相当于一个监测工具，可以每时每刻提供CPU的功耗。

4

研究方法



如图，对于任何一个design我们得到的都是RTL level。然后运行一些程序，就会得到一个fsdb/VCD 文件，得知每个信号在每一个周期的一些信息，这是最基本的input。

基于此，每个cycle就可以进行这样处理。每个cycle中，对每个信号（ABCDE）用1表示它翻转了，0表示没有翻转，要翻转就肯定会有功耗。这是cycle0，同样可以得到cycle1、cycle2等等，翻转活动就是模型的输入，然后来预测功耗。

如图，得到的矩阵的宽度是M， M表示design里面一共有M个signal，因此一共有M个输入，每个cycle就是一个sample。接着每个cycle都会做power simulation，得到最准确的power（p0、p1、p2……），将此作为一个vector。vector也是从p0开始的准确的功耗，有x、y，有输入有label，就可以训练一个machine learning模型，得出F（x）=y。

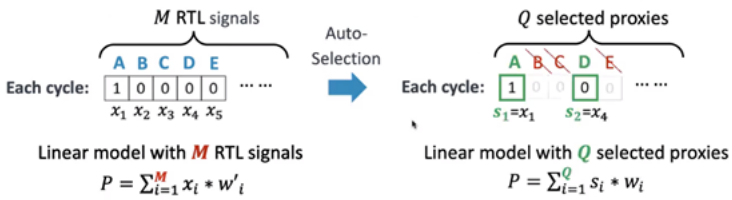
我们想要做的是训练出既准确又效率高的F。强调一点，我们的工作始终主要关注的是动态的power。由于当代CPU都非常复杂，并不是那么容易做，因此我们就要简化F模型。

核心思想

开始我们认为一个线性的模型，就已经足够提供既准确又快的power的估计。我们对动态的功耗进行模拟，计算的是电容的充放电，把所有的充放电的电容加起来得到总电容，然后乘以电压的平方，就是cycle的功耗。因此它本身就是一个线性模型，我们认为当然也可以用一个线性模型来模拟总功耗的过程。

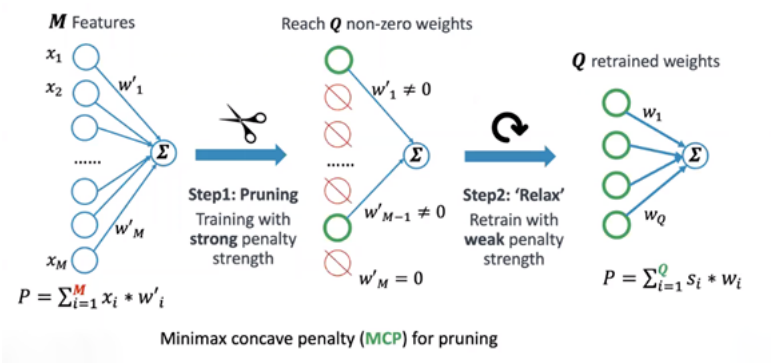
但是即使我们有一个线性模型，但这个线性模型还是M个input，M依然非常大，还是很复杂。

我们的第二个核心的思想是：一小部分cycle就能够提供足够的信息。因为很多信号都是相关的并不是完全相互独立，很多信号甚至完全一样。只需要看一部分最有代表性的信号，就足够作为模型的输入。



因此我们从M个信号中自动选取Q个有代表性的信号，我们把它叫做power proxies，然后让Q远小于M，这样模型就会变得很简单。

具体做法



我们用一种叫做剪枝的算法——**pruning**，比如开始是一个linear model，在 Linear model上面还要加一个penalty term，这个penalty term会惩罚所有的weight，如果weight过大，loss就会增加，使weight减少。这样就可以让绝大部分weight变为0，剩下则是不是0的weight，我认为这些不是零的weight很重要。

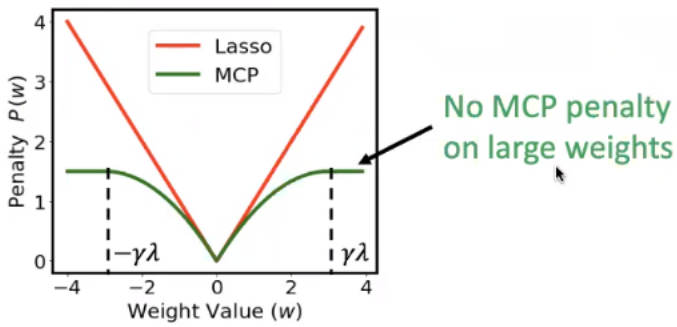
即使加了penalty之后，weight还必须要不是0，将不是0的weight保留，对应的信号就是要选取的信号。

在选取的过程中，会加一个非常强的penalty strength，使99.9%的weight全都变成0，这样可以使选取的信号最具有代表性。对penalty加的是一个叫做Minimax concave penalty(MCP)，用于剪枝算法。

选取有代表性的信号，基于这些信号，重新训练一个线性的模型，这个线性的模型就是最终的模型。这是第一步，也是最重要的一步。

选用 MCP算法的原因

在剪枝的时候，选用的是 MCP算法，而不是很多人熟悉的Lasso或是其它的。是因为要让选取的Q远小于M，penalty实际上就要加的非常大，因此惩罚很大。



图注：对不同的weight，Lasso和MCP的惩罚

如图所示，Lasso很简单，它是一视同仁的，weight越大，惩罚就越大。如果这样就相当于所有的weight都在被惩罚。这会导致，在惩罚性很大的情况下，即使那些不是0的weight，也会被压在一个非常小的值，模型就会变得不准确。由此基于一个不准确的模型，选出来的信号我们认为也是不准确的。

为了避免这种情况，所以我们使用了**MCP**。而使用MCP，当weight大到一定程度时，不会继续增大penalty。用MCP训练的模型，在整个训练过程中准确率都是比较高的，基于准确的模型做的剪枝，我们认为也是比较准确的。

另外我们观察到MCP选择的信号，彼此之间的相关性更小，这说明我们选的信号是有代表性的。

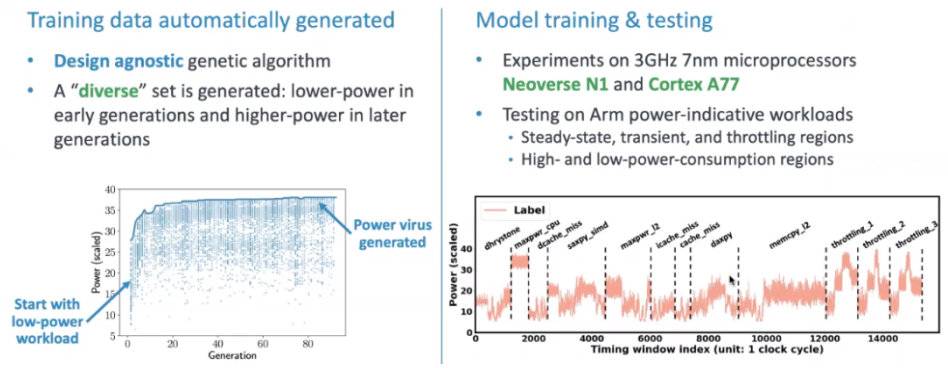
全自动机器生成的基本算法

除了APOLLO的算法之外，我们还有一套算法来提供训练数据来源。我们用纯机器自动生成很多workload，基于这些workload，来生成上述的input x 、label y等等，workload的生成有一套遗传算法。

开始有一些随机 workload，由于是随机生成的，因此它的功耗比较低。我们选取里面功耗高的做crossover或mutate，这就是遗传算法基本操纵。然后生成一些更高功耗的workload，一代又一代功耗会不断增加。

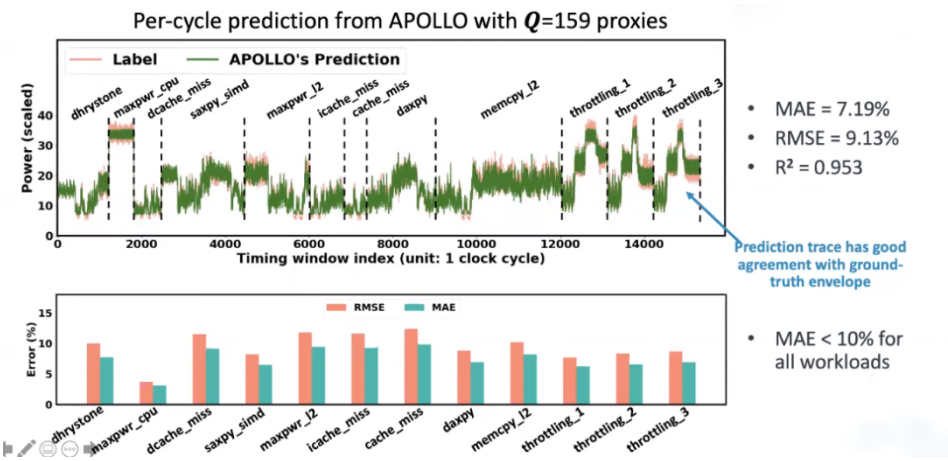
最后生成的workload，我们把它叫做power virus，它们的功耗非常高。这样我们就既得到了低功耗的workload，又有高功耗的，把两个掺在一起，训练数据就很全面了，就能够很准确的训练模型，这是我们全自动机器生成的一个基本算法。

实验结果



首先我们的实验是基于Neoverse N1和Crotex A77这两个CPU来做的，因此我们既测了服务器端，又测了移动端的CPU，让保证它在所有的CPU上都有很好的表现。

测试的时候也需要workload，这些workload是工程师手动写出来的，非常具有代表性。我们选选择了12个，既有有低功耗也有高功耗，还有快速变化的和保持不变的，覆盖了各种类型。



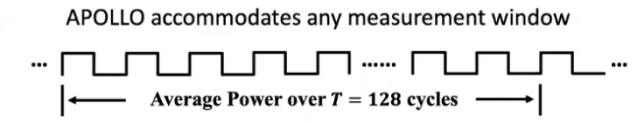
预测的结果：粉色的是真实的值，绿色的是预测的值

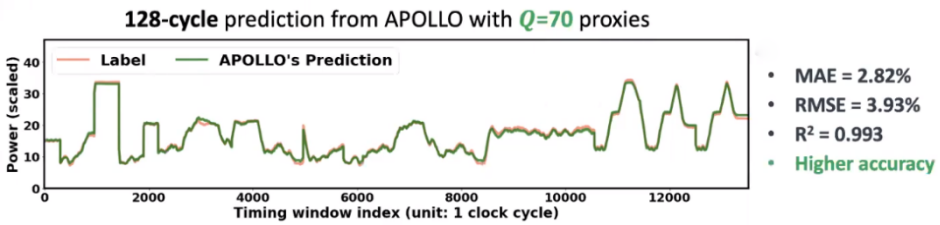
结果表明，预测的结果和真实的值具有很明显的相关性，匹配度很高。

我们测了它的error，MAE(mean absolute error)和RMSE(这两个值是越小越好)小于10%，（该值越大越好）高于0.95，说明准确率非常高。

同时我们计算了每个workload的MAE,发现所有类型的workload的MAE都少于10%，这说明了它的准确性。并且即使是7%的错误，也是由于清晰度太高，导致每个cycle之间有一个小错误这个是很难避免的。如果从一个更大的measurement window来算平均power，就会更准确。

事实上，APOLLO可以对任何一个measurement window进行计算，而不仅仅是 per-cycle。



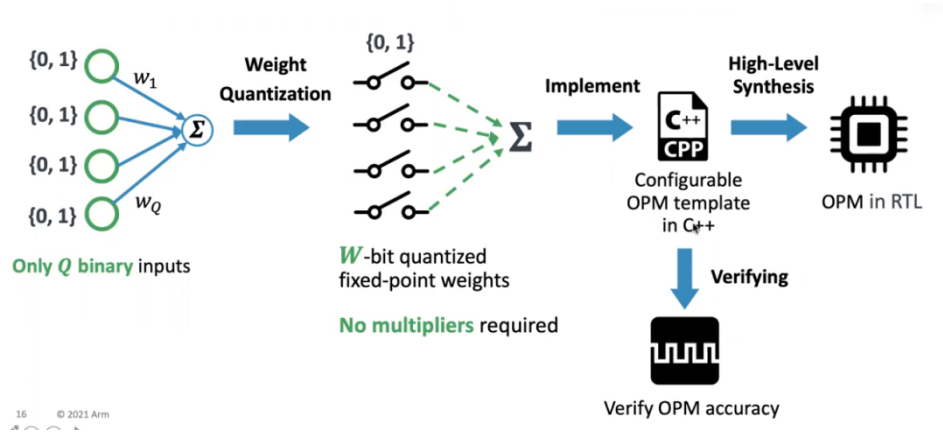


如上图，如果现在不需要per-cycle，只要一个average power ，over128个cycle，在这种情况下，只需要70个input，就可以做出一个准确的预测。预测结果error小于3%，如果能够容忍一个更大的measurement window，准确度将会几乎接近100%，因此在降低条件的情况下，它的性能可以有进一步的提升。

将APOLLO植入CPU

考虑到它的input数量少，同时模型简单、准确度高，因此我们要把它做到CPU里面。

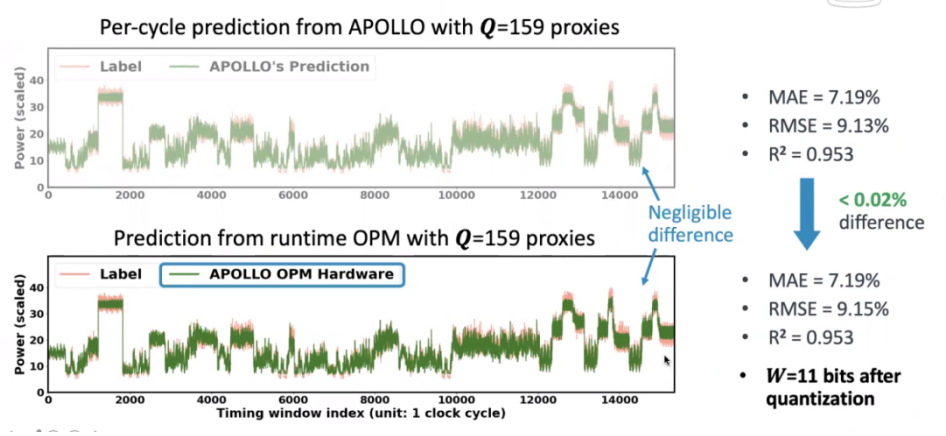
首先有Q个输入作为input，输入全都是0或者1，因此这个模型里面不需要乘法器，这样可以节省很大一笔开销。



同时weight作为quantization，不需要64位的weight那么准，只要需要十几位的weight，就可以很准确，因此开销又变得小了。

基于这个模型，用c++就可以很简单实现这个OMP模型，然后基于 C++的template，进行Hign-Level Synthesis，获得 design的RTL，如果这个RTL 可以和CPU的RTL合在一起，然后我们去做 tape out，这是一个最基本的思路，而流程本身也很简单。

同时基于C++的硬件设计，还可以verifying，可以验证硬件设计也是准确的。

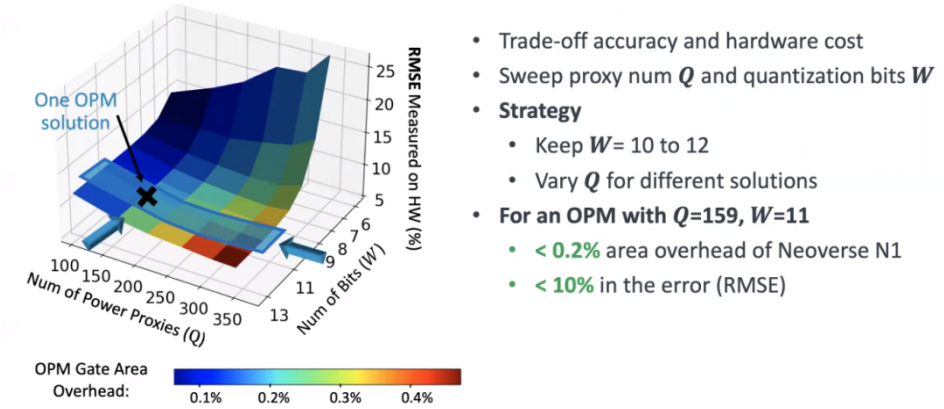


上面的图是APOLLO在软件上运行的结果。下面是硬件设计做的verification

如图所示，验证的结果两张图几乎是没有任何区别的，计算后区别小于0.02%，肉眼几乎不可见。

但注意下面这张图首先没有乘法器，另外它的weight现在不是64位，只有11位。在硬件已经优化的情况下它几乎没有准确率的损失，这说明硬件设计非常好。

硬件一定有trade off，在accuracy和hardware cost之间寻求一个平衡，因此我们计算了一下它到底是如何trade off的，然后来辅助我们设计一个这样的模块。



如图所示，我们用y轴来表示它的accuracy in error，然后用这个颜色来表示它在硬件上的代价(area overhead)，即占CPU比例是多少。

首先可以改变input的数量，另外一方面可以改变 quantization bits，我们改变这两个值观察它对accuracy和area overhead的trade off。

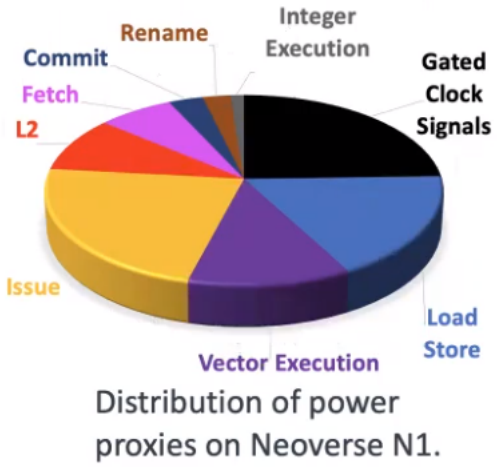
如上图，测量的结果中每个点都会有一个accuracy对应的hardware cost。当W继续小于10时，area会飞快的上升，即quantization 加的太大了，已经使原来的X扭曲掉了。所以quantization不能加的过大，并且W没必要大于12。因此我们策略是保持 W在10~12之间。

如果需要不同的solution，可以改变Q。比如我们根据这个策略，我们现在选到1个solution。如上图，OPM的Q是159，weight是11位，error大概是10%，在Neoverse N1上它的area overhead小于0.2%。所以我们认为它的实现代价非常低，并且准确率足够高，因此我们认为这是一个非常不错的solution。

所以到现在我已经介绍了它在设计时期，作为一个软件的准确率，和它在片上作为一个硬件的准确率以及实现的代价。

潜在应用

它开启了一些新的应用领域。举两个例子：

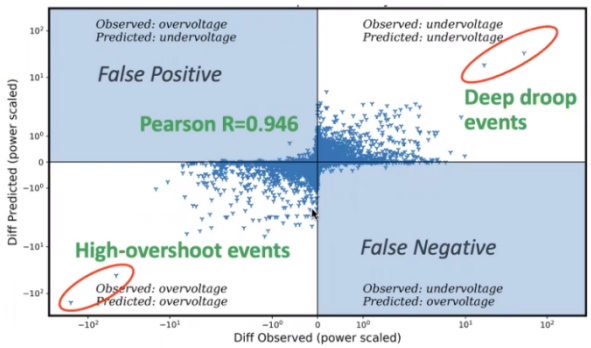


CPU中选取的信号的来源

在设计时期它可以给设计师很多反馈，如上图可以帮助设计师来了解 CPU里面功耗的组成。

为了进一步利用这个性质，我们可以允许CPU的设计师或架构师，自己限制来源范围，从里面找最有代表性的信号等，可以使设计师更容易理解这些信号。通过这种方法，这个模型的可解释性就变得更强，然后更能够辅助设计师来进行设计的决策。这当然这个是有一定代价的，如果限制了输入，它的准确率会有一定的下降，但下降非常少。

那么另外一个应用是上面所讲的voltage-droop电压降的问题，面对这个问题也可以用OMP来解决。



- OPM-generated current readings are differentiated to obtain di/dt events
- Excellent correlation is obtained for deep droop and deep overshoot events

上图是用OPM来预测di/dt的值，横坐标是我们测到的真实值，纵坐标是预测的值。当di/dt是正的时候，电流和power需求在不断的增加，那么这个时候有一个voltage-droop，电流需求增加，它的电压就会突然下降。当然，如果电流需求突然减少，它电压就会突然上升。

相当于我们有四个象限，如图两个蓝色区域预测和实际值完全相反，这两个是错的预测。而这两个错的预测的区域，几乎没有点是落在这个地方，就说明预测错的很少。而在预测对的区域里面，我们的预测非常准的。

因此我们的OPM可以在实际芯片运行的时候来指导我们去处理这些情况，因为它可以准确的预测。

中间很多的这些点，大家可能认为它的correlation看起来并不好。但请注意，我们的横轴和纵轴都是log scale，并不是linear scale，其实中间这个点它的值是非常小的，我们只是主动的去把它放大，把这些correlation不好的地方让大家去看清楚一些，实际上这些值非常小，所以实际上运行的时候影响是不大的。这点我们也可以从pearson simulation看出来，pearson只有0.946，这说明我们的预测是非

常准确的，因此我们认为我们的这个模型可以用于voltage-droop的motivation。同时大家注意这是CPU内部主动避免这个行为，相当于是预防。因此就比再加一套电路去阻止它会有效得多。

6
总结

- 快速的power-madelling对设计和部署CPU产生了实质性的影响
- 该方法与micro-architecture无关，且是自动化的,可以扩展到多个计算解决方案--CPU、GPU、NPU，甚至是子块。
- 潜在应用范围：从多核SoC中的power/thermel管理扩展到CPU驱动的主动降压缓解。
- ML/Data-Science方法是在设计中的许多方面拥有巨大潜力。

雷峰网(公众号：雷峰网)

相关文章

杜克大学　MICRO 2021最佳论文　谢知遥

与Jeff Dean聊ML for EDA，最佳论文花落伯克

重磅！2021 ACM 杰出科学家名单出炉：安波、虞晶

2021年IEEE Fellow名单公布，华人占比三成！生物医

MIT 博士蝉联金奖，北大获奖人数多达22人！阿里巴