Locate：使用近似加法器探索低功耗维特比解码器

Rajat Bhattacharjya，TCS 研究院，印度马哈拉施特拉邦孟买，bhattacharjya.rajat@tcs.com

Biswadip Maity，加利福尼亚大学尔湾分校，美国加利福尼亚州尔湾，maityb@uci.edu

Nikil Dutt，加利福尼亚大学尔湾分校，美国加利福尼亚州尔湾，dutt@uci.edu

arXiv:2304.03257v1 [cs.AR] 6 Apr 2023

摘要

维特比译码器广泛应用于通信系统、自然语言处理（NLP）及其他领域。维特比译码器虽然计算密集且功耗较高，但我们可以利用近似方法，在设计早期探索准确率、功耗和面积之间的权衡。我们提出 Locate，这是一个设计空间探索（DSE）框架，它在维特比译码器中计算密集且功耗关键的加-比-选单元（ACSU）中使用近似加法器。我们展示了 Locate 在早期探索两种应用——通信系统和 NLP——的准确率、功耗与面积权衡方面的效用，并给出一系列帕累托最优的设计配置。例如，在通信系统中，使用近似加法器后，面积降低 21.5%，功耗降低 31.02%，而在三种调制方案上的平均准确率仅损失 0.142%。类似地，在 NLP 场景下的词性标注器中，与在 ACSU 中使用超前进位加法器相比，15 种近似加法器中有 7 种可达到 100% 的准确率，平均节省 22.75% 的面积和 28.79% 的功耗。这些结果表明，Locate 可以与其他优化技术协同使用，从而提升维特比译码器在各种应用领域中的端到端效率。

CCS 概念

• **硬件** → **通信硬件、接口和存储**； • **计算方法** → 自然语言处理。

关键词

近似计算; 低功耗电路; 维特比译码器; 数字通信系统;绿色自然语言处理

1 引言

维特比译码器已广泛应用于各个领域——包括自然语言处理、语音识别，以及 LTE PDCCH、WLAN、CDMA和卫星通信等现代通信系统。随着时间的推移，鉴于维特比译码器在各个领域中的广泛应用，人们一直在持续研究并改进维特比译码器 [1–5]

维特比译码器通常计算成本高昂，并且难以在低功耗硬件中实现。我们来看两个示例：(1) 数字通信系统。

：在信道条件恶化的通信系统中，维特比译码器的功耗增长尤为显著 [1]。

论文已被 ACM GLSVLSI 2023 接收。作者版本仅供个人使用，禁止重新分发。最终版本将在 ACM GLSVLSI 2023 论文集中发布，网址为 https://doi.org/10.1145/3583781.3590314。

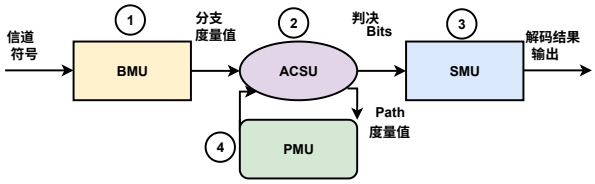


图 1：维特比译码器的功能框图。

研究表明，即使在 IEEE 802.11 a/g WLAN 标准所规定的理想条件下，具有 128 个状态、以 54 Mbps 运行的维特比译码器，其功耗也占收发器总功耗的 35% [2]（包括数字和模拟前端）。按每秒操作次数计算，这占所有数字处理复杂度的 76% [6]。(2) 自然语言处理（*NLP*）：由于需要考虑大量序列、语言本身复杂且难以确定最可能的序列，维特比译码在自然语言处理中颇具挑战。这导致计算复杂度和功耗都很高。因此，提出降低此类解码器功耗的方法变得愈发重要，这有助于提升它们在自然语言处理任务中的实用性。

以往研究提出了多种降低硬件维特比译码器功耗的方法。Abdallah 等人 [1] 提出了两种利用容错能力的低功耗译码器。Hegde 等人 [7] 采用电压过度缩放（VOS）和算法噪声容忍（ANT）来降低功耗并纠正时序错误。其他研究提出了缩小状态序列译码器 [8], 幸存存储器的大小 [9], 以及限制搜索网格 [10],。这些方法都需要在性能和准确率之间进行权衡。在这项工作中，我们重点通过利用加法器中的近似计算来降低功耗；加法器是维特比译码器中计算密集部分的关键组件。

具体而言，我们的*Locate* 设计空间探索（DSE）方法在维特比译码器的加比选单元（ACSU）中使用近似加法器（如图 1 中的单元 2 所示，并将在第 2 节中进一步说明）。众所周知，该 ACSU 是计算密集型内核，会导致较高的硬件复杂度和功耗；尤其是当约束长度增大时，因为这涉及大量网格状态和转移 [11]。通过在该单元中引入近似，*Locate*可以探索准确率、功耗和面积的帕累托最优设计点，使系统设计人员能够找到有意义的设计配置，在维持合理准确率的同时降低维特比译码器的功耗。我们通过实验评估和验证所提出的方法，获得最优设计点。对于数字通信系统，

所生成的低功耗译码器采用加法器 add12u_187 [12]，与在 ACSU 中使用超前进位加法器（CLA）的译码器相比，面积减少 21.5%，功耗降低 31.02%（在 BASK、BPSK 和 QPSK 三种调制方案上取平均），准确率损失为 0.142%。在自然语言处理（NLP）场景下的词性标注器中，我们观察到，在 15 种近似加法器中，有 7 种准确率达到 100%；与在 ACSU 中使用 CLA 的译码器相比，其平均面积减少 22.75%，功耗降低 28.79%。

2 背景

如图1所示，维特比译码器是用于解码卷积码或网格码的广泛使用的组件。其底层算法称为维特比算法，采用动态规划技术，并经常与马尔可夫源和隐马尔可夫模型（HMM）结合使用 [13]。维特比算法估计最可能的隐藏状态序列的概率，该序列称为维特比路径。维特比译码方法广泛应用于各种领域，包括数字通信、自然语言处理（NLP）、语音识别和DNA分析。

维特比译码器（图1）包含以下四个关键单元：分支度量单元（BMU）① 用于计算码字母表中的每个符号与接收符号之间的距离，并利用范数对距离进行标准化。这些标准化后的距离称为分支度量（BMs）。分支度量被输入加-比-选单元（ACSU）②，该单元递归计算路径度量（PMs），并为每种可能的状态转移生成判决位作为输出。随后，判决位被存储在幸存路径存储单元（SMU）③ 中并从中读取，以获得最终幸存路径上的解码源比特。当前迭代的路径度量存储在路径度量单元（PMU）④中。

鉴于维特比算法具有抗错误特性且应用范围广泛，可以探索近似计算技术，以进一步降低功耗并提高效率。鉴于算法和应用具有抗错误特性，近似计算已发展成为传统计算方法的一种流行替代方案 [14]。近似方法可以应用于不同的抽象层级，从完整系统层级 [15] 到软件层级 [。

16]和存储 [17, 18]。诸如加法器和乘法器之类的算术单元构成了所有此类近似系统中的基本构建-模块 [19–22]。

鉴于已确定 ACSU 是维特比译码器的功耗瓶颈，我们的 Locate 方法通过在功耗较高的 ACSU 内部硬件层面部署近似加法器，并探索准确率-功耗-面积设计空间，作为对既有优化工作的补充，其应用涵盖数字通信和自然语言处理。

3 方法论

图 2 概述了我们用于维特比译码器低功耗译码器探索的*Locate*方法。该方法包含三个主要步骤：软件层级的功能验证、硬件实现，以及设计空间探索，以生成一个解码器。具体步骤如下：

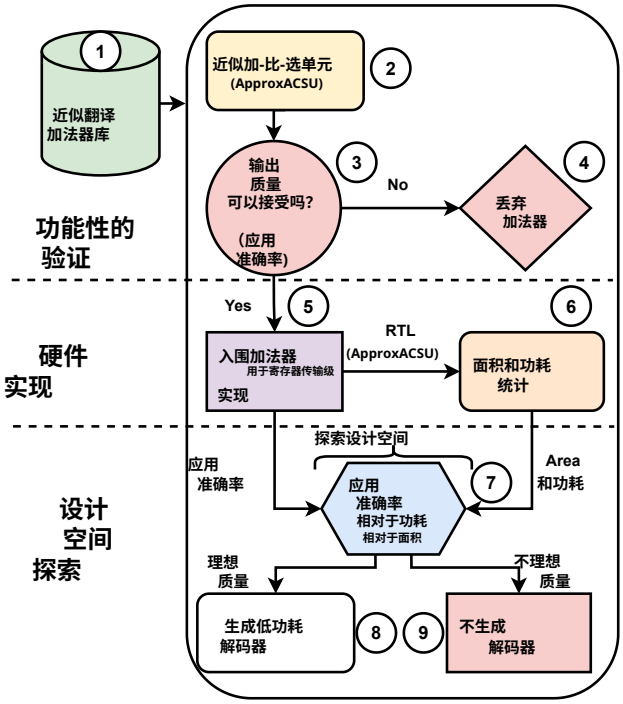


图 2：Locate 的方法论

3.1 功能验证

功能验证在软件层级使用 MAT- LAB/Python 进行，以评估近似对输出质量的影响。一个关键问题是在应用的质量窗口约束下，确定适合进行近似的模块以及这一选择背后的原因。

先前的研究表明，尽管功耗高，维特比译码器仍具有抗错误能力 [7],，使其成为近似技术的理想候选对象。然而，首先有必要确定维特比译码器（先前在图1中所示）内部的哪个特定单元最适合进行近似。为此，我们选择维特比译码器中的加-比-选单元（ACSU）。ACSU 以递归方式计算路径度量，从而针对每个可能的状态转移给出判决位作为输出。在ACSU内对运算进行近似是有利的，因为这减少了与递归相关的大量复杂计算。

本文研究了部署维特比译码器的两个主要应用：完整的端到端数字通信系统，以及自然语言处理领域的词性标注器。为确定该方法在 Locate 特定场景下的有效性，我们针对每种应用采用不同的硬件实现进行精度分析。该分析涉及使用来自 EvoApprox 库 [1] [12] 的各种近似加法器，对 ACSU② 中的加法运算进行近似。如果应用的输出质量满足所需的准确率要求③，则考虑在寄存器传输级中实现该库中的加法器；否则，将其排除在后续考量之外④。需要注意的是，并非所有加法器

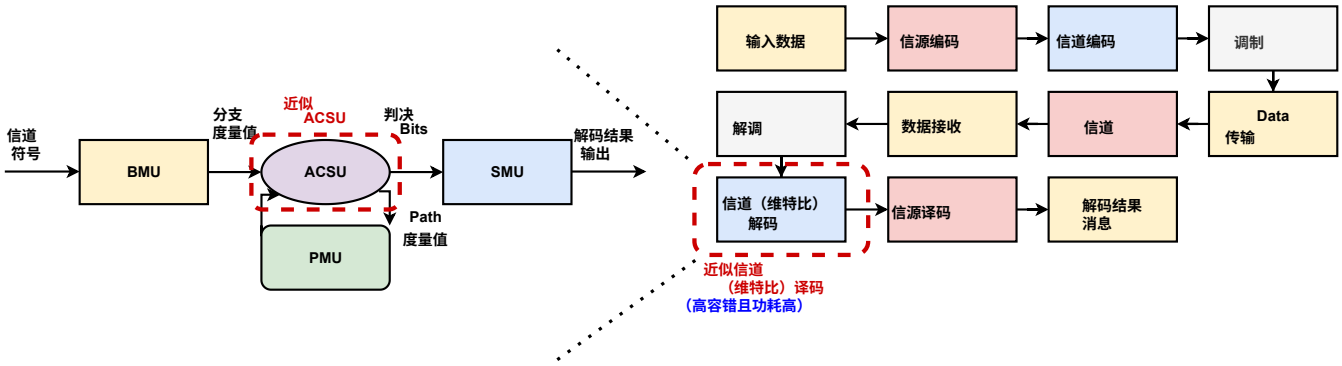


图 3：端到端数字通信系统：近似单元以红色突出显示。

适用于所有应用，因为它们对准确率度量值的影响各不相同。其中一些表现优于其他方法，因此需要进行全面的设计空间探索，以评估与该方法相关的各种准确率-硬件权衡。

为了评估近似加法器的准确率，考虑了各种度量值，例如平均绝对误差（MAE）、误差百分比（EP）、最坏情况绝对误差（WCE）、均方误差（MSE）、平均相对误差（MRE）等。 [23]需要注意的是，包括输入数据的分布和数字设计在内的各种因素都可能影响这些误差。因此，有必要研究这些因素，以便更深入地了解哪些电路适用于哪些应用，以及其中的原因。

3.2 硬件实现

为全面了解近似技术在降低功耗和面积方面的优势，必须对已识别的模块进行硬件实现 ⑤。为此，我们使用 Verilog 硬件描述语言实现 ACSU 的寄存器传输级，并在 ACSU 中集成各种近似加法器。我们使用 Synopsys Design Compiler (DC) 和 45 纳米 NanGate 开放单元库，对近似 ACSU 设计和精确 ACSU 设计进行综合。通过这一过程，我们可以获得这些设计相关的面积和功耗统计数据 ⑥。用于近似 ACSU 中加法运算的近似加法器选自 EvoApprox 库 [12]。接下来，我们通过改变近似加法器来探索维特比译码器的设计空间。

3.3 设计空间探索

设计空间探索（DSE）是一种用于电子设计自动化（EDA）的方法，旨在综合考虑各种设计约束和目标，为系统确定最佳设计点。在 Locate 中，我们通过探索维特比译码器在准确率、功耗和面积之间的权衡，开展设计空间探索。

在此前已确定准确率和硬件（面积、功耗）统计数据后，考虑设计点之间的权衡，识别帕累托最优设计点至关重要。帕累托最优设计点是指满足所有设计约束和目标的设计点，例如达到特定的准确率阈值，同时以最大限度降低功耗和面积利用率。为此，我们对维特比译码器进行设计空间探索（准确率 v/s 功耗 v/s 面积） ⑦，使用各种近似 ACSU，以获得帕累托最优设计点 ⑧。如果对准确率、功耗和面积之间权衡的评估结果不理想，则不使用与该设计点对应的加法器来生成译码器⑨。需要注意的是，这项选择9 与 4 中的选择是相互独立的。4 允许在软件层级完成«功能验证»后立即进行初步选择，而对于9，还必须完成硬件实现和设计空间探索，才能得出结论。这种两步方法通过筛除不符合预期标准的加法器，使我们能够确定最佳的低功耗译码器设计。

4 评估

为了评估在维特比译码器的ACSU 内近似加法操作的有效性，我们评估两个最先进的应用：端到端数字通信系统（第 4.1 节）和词性标注器（第 4.2 节），后者是自然语言处理中的一项关键任务。

这两个应用所采用的方法保持不变（详见第 3 节）。首先，我们在软件层级执行功能验证，这有助于我们验证对维特比译码器的 ACSU 内部加法运算进行近似的方法，从而将其纳入端到端系统中。其次，我们使用各种近似加法器对受影响的 ACSU 模块进行硬件评估。最后，我们执行设计空间探索，以找出帕累托最优设计。

4.1 数字通信系统

所考虑的数字通信系统如图3 所示。该系统来源于 [24]中提供的存储库。通信系统不同模块中所使用的技术列于表1中，整个模型的系统属性列于表 2 中。仅对信道解码器模块（维特比译码器）进行近似处理，而其余模块保持不变。针对信道解码器中的近似处理，开展了端到端性能分析。

表 1：通信系统模块及其属性

模块类型	所用技术
信源编码	霍夫曼编码
信道编码	卷积编码
调制	幅移键控（ASK）和 相移键控（PSK）
信道	加性白高斯噪声信道
解调	幅移键控（ASK）和 相移键控（PSK）解调
信道译码	维特比译码
信源译码	霍夫曼译码

总共从EvoApprox库中评估了14 个近似加法器 [12]，以对 ACSU 中的加法运算进行近似。这些加法器为：add12u_2UF、add12u_39N、add12u_0UZ、add12u_0Z5、add12u_0LN、add12u_187、add12u_0ZP、add12u_103、add12u_0AF、add12u_28B、add12u_4NT、add12u_50U、add12u_0C9、add12u_0AZ。我们在3种调-制方案下进行评估，即二进制幅移键控（BASK）、二进制相移键控（BPSK）和正交相移键控（QPSK）。

4.1.1 精度分析。 为了评估数字通信系统中各设计点的准确率，我们传输文本数据（653 个单词），并计算接收数据时出现的任何错误。错误以误码率（BER）的形式报告，使用 EvoApprox 库中的 8 个加法器的通信系统的 BER 与信噪比（SNR）（dB）之间的结果如图 4 所示，其中 SNR 为 X 轴，BER 为 Y 轴。每个加法器和调制方案的每个 SNR 对应的 BER 均取十几次运行结果的平均值。

我们得出了 4 个主要观察结果。首先，在 14 个加法器中，有 6 个会导致数据完全损坏，因此被认为不适用于该通信模型（故未显示在图 4 中）。其次，add12u_187 在所有信噪比水平下的平均误码率最低。

表 2：通信模型的系统属性

属性	Type
调制	BASK、BPSK、QPSK
每位采样数	40
比特率	1000
载波频率	1000 Hz
载波幅度	1 V
SNR	-15 至 10 dB
信道编码器的生成矩阵	[1 1 1; 1 0 1]
寄存器中的位移数量	1

这很有意思，因为其 MAE 为 0.24%，EP 为 49.22% [12]，而 add12u_2UF 的 MAE 和 EP 均为 0%。第三，与精确计算的情况相比，使用 add12u_187 时，误码率的准确率仅略微下降了 0.142%（对各种调制方案取平均）。这证明了使用近似加法器构建低功耗维特比译码器的可行性。第四，在负信噪比下的大多数情况下，BPSK 的误码率低于 QPSK 和 BASK，但在某些情况下 QPSK 表现更好。在负信噪比区域，BASK 的误码率高于 PSK 调制方案，这符合预期。

最后，我们得出结论：尽管某个特定电路可能有其自身相关的误差度量值，但在端到端系统中，使用 Locate 时，我们观察到的行为由于各模块之间的动态交互而不同于预期。

4.1.2 硬件评估。 我们按照第 3.2 节所述，使用 Synopsys DC 综合硬件，并获得如图 5 所示的全面面积和功耗统计数据。我们观察到，超前进位加法器消耗的面积和功耗最多，这不足为奇，因为超前进位电路本身结构复杂。统计数据还显示，在所有候选加法器中，add12u 28B 消耗的面积和功耗最少。不过，需要注意的是，add12u_28B 会导致数据完全损坏，因此不可行使用它来进行_

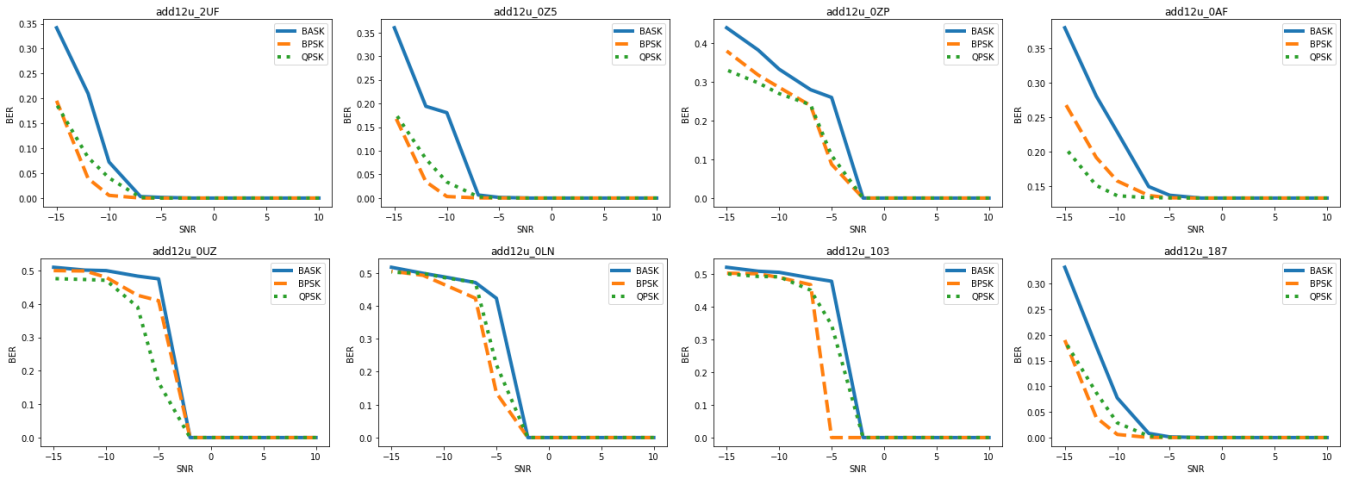


图 4：在完整数字通信系统中，维特比译码器使用不同近似加法器时的误码率与信噪比关系（文本数据传输与接收）

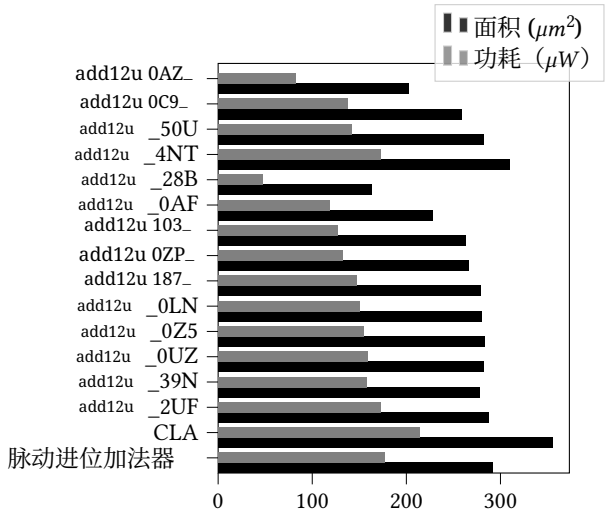


图 5：采用不同加法器的维特比译码器在数字通信系统中的面积和功耗统计

此类应用。接下来，利用硬件统计数据进行设计空间探索（DSE）。

4.1.3 设计空间探索（*DSE*）。准确率、面积和功耗之间的关系十分复杂：某些加法器可能具有出色的准确率，但在硬件层面可能需要消耗更多芯片面积和功耗；另一些加法器的芯片面积和功耗指标可能非常出色，但会严重降低应用准确率。因此，全面探索涉及多种因素的设计空间至关重要。

在实验中，我们探索了一个包含以下参数的三维设计空间：应用准确率（BER）、芯片面积和功耗。所观测到的 BER 来自采用三种调制方案——BASK、BPSK 和 QPSK——的端到端数字通信模型。面积和功耗统计数据则来自采用各种精确加法器和近似加法器的 ACSU。结果如图 6 所示。

• **BASK:** 图 6a 展示了 BASK 的设计空间。我们观察到，尽管超前进位加法器的 BER 最低，但面积和功耗都非常高。若所需 BER 为 <0.2，则有 6 个候选方案满足该条件。对于面积预算 <250 μm^2，我们有 3 个候选方案，但只有 1 个满足 BER 不高于 <0.2 的条件，即 add12u_0AF。同样，对于功耗预算 <140 μW，有 6 个候选方案满足该条件，但只有 2 个满足 BER 不高于 <0.2 的条件，分别是 add12u_0AF 和 add12u_0ZP。因此，图 6a 所示的三维空间为我们提供了根据设计者需求进行权衡、生成低功耗维特比译码器的机会。

• **BPSK:** 根据图 6b，我们可以分析 BPSK的设计空间。与 BASK 类似，如果所需的误码率为 <0.2，那么有6个候选项满足该标准。同时，对于特定的面积和功耗预算，结合误码率阈值，可以进行类似的分析，以筛选出加法器候选项，从而为低功耗维特比解码器实现最佳设计。

• **QPSK:** 对于 QPSK，图 6c 全面展示了适用于生成低功耗维特比译码器的加法器类型。可以看出，在功耗预算 < 130 μW 下，有 4 个加法器满足条件，分别为 add12u_103、add12u_0AF、add12u_28B、add12u_0AZ；然而，若要求 BER 阈值为 <0.2，则其中只有 add12u_0AF 同时满足这两个条件。因此，根据设计者/应用的需求，也可以针对 BER、面积和功耗的各种组合得出类似结论。

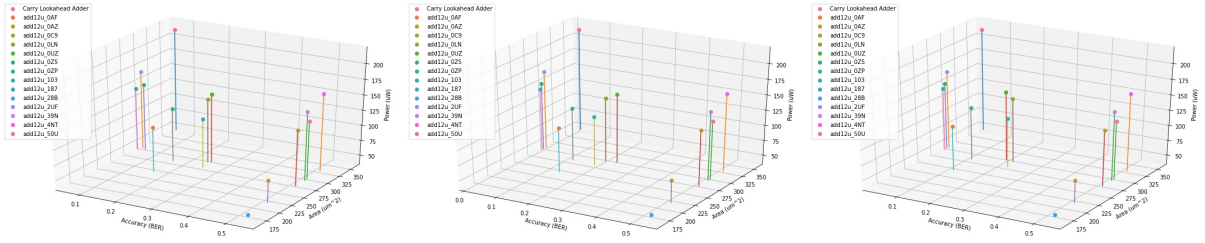
尽管我们得到了上述针对每种调制方案的结论，但需要注意的是，这仅对图 3 所示的通信模型及表 1 和 2 中提到的其属性有效。随着通信模型及其属性的变化，Locate 将有助于确定最优设计点。

4.2 绿色自然语言处理

为衡量近似方法在维特比译码器中更广泛的适用性，我们还使用近似维特比译码评估词性（POS）标注，以探索低功耗和绿色自然语言处理。

4.2.1 精度分析。为进行精度分析，我们使用 Python 在自然语言处理场景中实现了一个采用近似 ACSU（近似加法器）的词性标注系统。为测试我们的方法，我们选取了 3 个句子，分别包含 2、3 和 6 个单词。

我们从EvoApprox库中评估了15个近似加法器 [12]，以构建近似维特比译码器，为这些句子标注词性并报告准确率水平。经过分析，我们发现有7个加法器达到了 100% 的准确率，即 add16u_1A5,



(a) BASK 的设计空间探索　(b) BPSK 的 DSE　(c) QPSK 的设计空间探索

图 6：使用各种加法器和三种调制方案对数字通信系统中的维特比译码器进行设计空间探索：BASK、BPSK、QPSK

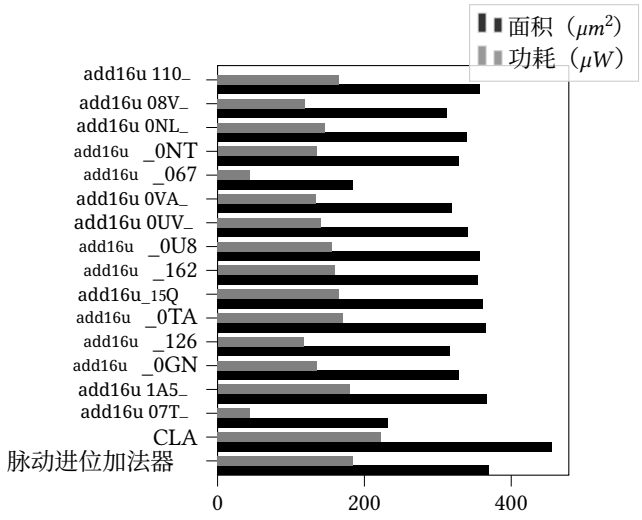


图 7：维特比译码器中使用各种加法器的词性标注器（自然语言处理）面积和功耗统计

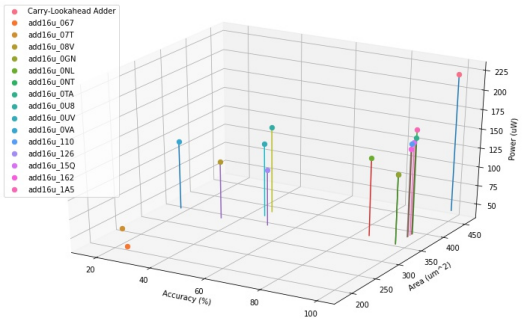


图 8：使用各种加法器的绿色自然语言处理维特比译码器设计空间探索

add16u_0GN、add16u_0TA、add16u_15Q、add16u_162、add16u_0NT 和 add16u_110。add16u_0NL 的准确率为 88.89%。其余加法器的准确率均低于 60%。

上述结果表明，对于词性标注这类应用，在许多情况下，使用近似加法器代替精确加法器仍可达到 100% 的准确率，从而支持采用近似方法。

4.2.2 硬件评估。 与第 4.1.2 节类似，硬件评估使用 EvoApprox 库中的 15 个 16 位近似加法器（如图 7 所示）进行。分析结果表明，add16u_07T 的功耗最低，为 44.195 μW。然而，add16u_07T 的准确率仅为 16.663%，不适用于实际情况。因此，为研究准确率与硬件之间的权衡，我们在下一小节中探索使用近似 ACSU 的词性标注器设计空间，以获得最优的解码器设计。

4.2.3 设计空间探索（*DSE*）。 图8展示了一个3D网格，其中包含每个设计点的准确率、面积和功耗。

目标是寻找最优设计点，以便为不同使用场景生成低功耗维特比译码器。从图 8 中我们可以得出结论：对于 $< 120\ \mu W$ 的功耗预算，有 4 个候选方案满足该标准，但其中没有任何一个的准确率达到 $> 60\%$。根据设计者/应用对特定准确率水平的要求，以及相应的功耗和面积预算，也可以得出类似的观察结果并进行权衡。

5 结论与未来工作

我们提出了 Locate，这是一种设计空间探索（DSE）方法，用于评估满足准确率和面积约束的帕累托最优低功耗维特比译码器。Locate 利用近似加法器执行维特比译码器中计算密集型 ACSU 内部的加法运算。我们通过两个涉及维特比译码的示例应用展示了 Locate 的实用性：一个端到端数字通信系统和一个自然语言处理（NLP）中的词性标注器。实验结果表明，Locate 能够找到满足质量约束、同时降低芯片面积和功耗的帕累托最优设计点。

不同应用领域对近似的敏感度各异，因此需要使用 Locate 这类框架进行全面的设计空间探索。例如，Locate 生成的设计空间表明，与 NLP 中的词性标注器相比，数字通信系统对近似带来的影响容忍度较低。这种差异取决于所涉及的通信模型和系统属性。设计点的探索因应用而异，通用方法未必总是最优；这一点可从数字通信中误码率（BER）仅损失 0.142% 的准确率，以及词性标注器所达到的高准确率看出。未来的工作将研究把 Locate 与 [3–5] 中所述的其他互补方法相结合，以进一步提高端到端系统中维特比译码的效率。

参考文献

[1] R. A. Abdallah 等。抗错误低功耗维特比译码器。载于 *ISLPED* 会议论文集，第 111–116 页，2008 年。[2] B. Bougard 等。无线局域网收发器的能耗可扩展性增强。*IEEE* 第五届 *SPAWC* 研讨会，第 449–453 页，2004 年。[3] S. K. Monfared 等。基于位切片的高吞吐量并行维特比算法。*ACM TOPC*, 8(4)，2021 年。[4] M. Ciaperoni 等。Sieve：一种空间高效的维特比译码算法。载于 *SIGMOD '22* 会议论文集，第 1136–1145 页，纽约，美国，2022 年。[5] P. P. Raj 等。用于硬件加速语音识别的缩减存储器维特比译码。*ACM TECS*, 21(3)，2022 年 5 月。[6] K. Masselos 等。无线通信系统的可重构性要求。载于 *IEEE* 异构可重构片上系统研讨会论文集，2002 年。[7] R. Hegde 等。软数字信号处理。*TVLSI*, 9(6)：813–823，2001 年。[8] J.B. Anderson 等。卷积码的缩减状态序列检测。*IEEE*信息论汇刊，40(3)：965–972，1994 年。[9] Y. Chang 等。一款 2 Mb/s、256 状态、10 mW、码率为 1/3 的维特比译码器。*IEEE*固态电路杂志，35(6)：826–834，2000 年。[10] F. Sun 等。用于并行高吞吐量有限搜索网格译码器的超大规模集成电路设计。*TVLSI*, 13(9)：1013–1022，2005 年。[11] D. Vaithiyanathan 等。采用流水线T算法的高性能维特比译码器加比选。《亚历山大工程学报》，54(3)：447–455，2015 年。[12] V. Mrazek 等。Evoapprox8b：用于电路设计和近似方法基准测试的近似加法器与乘法器库。载于 *DATE* 会议论文集，第 258–261 页，2017 年 3 月。[13] G.D. Forney。维特比算法。*IEEE* 会议论文集，61(3)：268–278，1973 年。[14] Jie Han 等。近似计算：一种新兴的能效设计范式。载于 *2013* 年第 *18* 届 *IEEE* 欧洲测试研讨会（*ETS*），第 1–6 页，2013 年。[15] A. Raha 等。迈向全系统能耗与精度权衡：近似智能相机系统案例研究。载于 *DAC*，第 1–6 页，2017 年。

[16] M. Samadi 等。Paraprox：面向数据并行应用的基于模式的近似。载于 *ASPLOS* 会议论文集，第 35–50 页，纽约，美国，2014 年。[17] B. Maity 等。自适应存储器近似：一种形式化控制理论方法。*IEEE* 嵌入式系统快报，12(2)：33–36，2020 年。[18] B. Maity 等。SEAMS：近似存储器层级的自优化运行时管理器。*ACM TECS*，20(5)，2021 年 7 月。[19] R. Bhattacharjya 等。一种带修正的近似进位估算并行加法器。载于 *GLSVLSI* 会议论文集，第 139–144 页，2020 年。[20] S. Ghabraei 等。Axcem：设计支持近似比较器的乘法器。《低功耗电子与应用》，2020 年 3 月。[21] R. Bhattacharjya 等。Rearm：一种用于图像处理的可重构近似舍入乘法器。载于 *2020* 年第 *24* 届国际超大规模集成电路设计与测试研讨会（*VDAT*），第 1–4 页，2020 年。[22] S. Belwal 等。ACLA：一种具有智能进位判断与校正功能的近似超前进位加法器。载于 *ISQED* 会议论文集，2021 年。[23] Z. Vasicek。近似电路精确分析的形式化方法。*IEEE Access*，7：177309–177331，2019 年。[24] M. M. Alam。A-Complete-Digital-Communication-System。https://github.com/MahmudulAlam/A-Complete-Digital-Communication-System。 [在线；访问于 11 月 18 日-2022]。