一种基于 3D NAND 存储器的存算一体架构及其系统技术协同优化仿真

郑好[1)2)] 刘慧雯[3)] 方语萱[3)] 范冬宇[3)] 韩玉辉[3)]
侯春源[3)] 刘威[3)] 夏志良[3)†] 霍宗亮[1)3)‡]

1) (中国科学院微电子研究所, 北京 100029)

2) (中国科学院大学, 北京 100049)

3) (长江存储科技控股有限责任公司, 武汉 430070)

(2025 年 7 月 8 日收到; 2025 年 10 月 1 日收到修改稿)

随着 ChatGPT 等大语言模型的发展, 产业界对硬件的算力、容量和功耗提出了新的需求. 存算一体 (compute-in-memory, CIM) 技术相较于传统近存计算, 减少了数据搬移, 显著降低功耗. 而在众多存储器中, 3D NAND 闪存因其成熟的工艺制造技术和超高容量, 是最有可能实现大模型本地部署的候选方案. 然而, 目前针对 3D NAND 闪存 CIM 芯片的研究大多停留在学术研究阶段, 未基于产品级 3D NAND 闪存芯片进行系统性的 CIM 架构设计和大模型功能验证. 对此, 本文搭建了基于 PyTorch 框架的大语言模型仿真器平台来评估系统架构的性能, 并提出了一种基于源线背面切分工艺的通用 3D NAND 架构. 该架构通过改动 3D NAND 的源线制造工艺以支持 CIM 计算, 工艺成本极低, 可供产业界快速迭代, 并完善了相应的映射算法和流水线设计. 最后通过仿真器平台对所提出的架构在电流分布和量化的影响下进行了性能评估, 仿真结果表明所设计的产品级 3D NAND 芯片可以在 GPT-2-124M 大模型上做到 20 tokens/s 的生成速度和 5.93 TOPS/W 的能效比, 在 GPT-2-355M 大模型上做到 8.5 tokens/s 的生成速度和 7.17 TOPS/W 的能效比.

关键词: 3D NAND, 存算一体, 硬件加速

PACS: 85.40.–e **DOI:** 10.7498/aps.74.20250891

CSTR: 32037.14.aps.74.20250891

1 引 言

人工智能 (artificial intelligence, AI) 的快速发展开启了大规模深度学习模型的新时代. 以 OpenAI 的 GPT 系列的大语言模型为例的诸多大模型如雨后春笋般涌现, 这些大模型在自然语言处理、计算机视觉等领域展现了前所未有的能力, 推动了各行业的变革. 然而, 大模型的部署、训练和推理任务仍然面临着巨大的计算挑战, 包括海量数据处理需求、高内存容量要求以及显著的能耗问题. 因此, 亟需创新的硬件解决方案, 以满足这些模型对性能、容量和能效的需求. 存算一体 (compute-in-memory, CIM) 技术作为一种新型的解决方案, 展现出巨大的潜力. 与传统冯·诺依曼架构相比, CIM 技术通过将计算直接集成到存储阵列中, 避免了内存与处理器之间频繁的数据搬运, 从而显著降低数据传输开销, 提高能效, 并支持大规模并行处理. 在众多可以实现 CIM 的存储器中, 许多非易失性存储器如 FeFET[1,2], MRAM[3,4], SRAM[5,6]

† 通信作者. E-mail: albert_xia@ymtc.com

‡ 通信作者. E-mail: zongliang_huo@ymtc.com

以及 RRAM[7–9] 等已经在学术界被广泛研究, 但受限于目前工艺或者器件本身特性, 这些非易失性存储器的存储容量都无法满足大语言模型的参数需求. 而在传统非易失性存储器中, 3D NAND 闪存虽然存在存储单元擦写次数受限的问题, 但是在大模型推理任务下权重固定后只需要少量校准写入, 极大降低了对单元擦写次数的需求, 同时因其极其成熟的制造工艺和目前非易失性存储器中最高的存储密度和成本效益, 综合下来是实现 CIM 推理芯片商业化的有力候选方案.

尽管如此, 目前基于 3D NAND 闪存的 CIM 推理芯片研究仍停留在学术阶段[10–13]. 并且, 考虑到商业化成本和 3D NAND 本身的器件特性, 大多数都只集中于小规模原型设计, 缺乏针对大语言模型的系统性架构探索和功能验证[14]. 为填补这一空白, 迫切需要开发适用于大语言模型的 3D NAND 闪存的 CIM 架构, 并配套高效的数据映射算法和系统级评估框架. 本文旨在应对这些挑战, 搭建了基于 PyTorch 的 3D NAND 大语言模型系统级仿真平台, 结合所提出的基于 3D NAND 闪存的全新架构, 通过特殊的工艺架构设计、数据量化映射, 流水线设计和硬件算法逻辑优化完成了全面的系统解决方案和性能评估. 最后, 本文在基于 Transformer 模型的 GPT-2-124M 模型上完成对推理时间, 功耗和误差容忍度的仿真, 以及在 GPT-2-355M 上对推理时间、功耗的仿真. 结果表明本文的系统解决方案具备较高的 AI 算力和高功耗比的发展潜力, 为未来基于 3D NAND 的存算一体芯片商业化提供方向指导.

2 Transformer 模型

Transformer 模型是由 Vaswani 等[15] 在 2017 年提出的一种深度学习模型, 目前已广泛应用于自然语言处理 (natural language processing, NLP) 任务. 与传统的循环神经网络 (recurrent neural network, RNN) 不同, Transformer 不依赖于序列顺序处理输入, 而是采用自注意力机制 (self-attention) 来捕捉输入序列中各个元素之间的关系. 在 Transformer 模型被提出之后, 各种大语言模型开始不断涌现, 其中 GPT-2 (generative pre-trained transformer 2) 是由 OpenAI 于 2019 年推出的一种语言生成模型. 其通过无监督学习在大规模文本数据上进行预训练, 在多个自然语言处理任务, 例如文本生成、翻译、问答系统等多个领域中表现出色, 展现了强大的生成能力. GPT-2[16] 只使用 Transformer 架构中的解码器部分进行语言生成, 其算法如图 1(a) 所示. 在每一次前向推理计算中, 输入经过词嵌入层转换为向量表示, 然后通过位置嵌入添加位置信息并对结果归一化, 接着归一化的输入向量进入掩码多头自注意力计算模块, 如图 1(b) 所示, 其中自注意力通过 (1) 式计算:

$$\text{Attention}\left(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}\right)=\frac{\text{softmax}\left(\boldsymbol{Q}\cdot\boldsymbol{K}^{\mathrm{T}}\right)}{\sqrt{dk}}\cdot\boldsymbol{V}, \tag{1}$$

$$\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}=\text{Input}\cdot\boldsymbol{W}^{\boldsymbol{QKV}}, \tag{2}$$

其中, $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ 分别表示 Query, Key 和 Value 矩阵, 由输入向量和 QKV 投影矩阵 $\boldsymbol{W}^{\boldsymbol{QKV}}\in\mathbb{R}^{\text{dmodel}\times\text{dmodel}}$ 计算得到; dk 是 $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ 的维度. 同时, 由于是多头自注意力计算, 还需要对 $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ 进行拆分得到 $\boldsymbol{Q}_i,\boldsymbol{K}_i,\boldsymbol{V}_i$ 来分析词语之间不同维度的关联性, 具体表示为

$$\begin{aligned}&\text{MultiHead}\left(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}\right)\\&=\text{Concat}(\text{head}_1\cdots\text{head}_n)\cdot\boldsymbol{W}^{\boldsymbol{O}},\end{aligned} \tag{3}$$

$$\text{head}_i=\text{Attention}\left(\boldsymbol{Q}_i,\boldsymbol{K}_i,\boldsymbol{V}_i\right), \tag{4}$$

其中 $\boldsymbol{W}^{\boldsymbol{O}}$ 为多头融合投影矩阵, 用来整合不同注意力头的信息, $\boldsymbol{W}^{\boldsymbol{O}}\in\mathbb{R}^{\text{dmodel}\times\text{dmodel}}$. 掩码多头自注意力计算得到的结果交由两层全连接层网络处理, 中间对第一层的输出采用 GELU(Gaussian error linear unit) 激活函数. 全连接层计算过程如图 1(c) 所示, 可以表示为

$$\text{FFN}\left(\boldsymbol{X}\right)=\text{GELU}\left(\boldsymbol{X}\cdot\boldsymbol{W}_1+\boldsymbol{b}_1\right)\cdot\boldsymbol{W}_2+\boldsymbol{b}_2, \tag{5}$$

其中 $\boldsymbol{X}$ 为全连接层输入; $\boldsymbol{b}_1$ 和 $\boldsymbol{b}_2$ 均为偏置向量; 权重 $\boldsymbol{W}_1\in\mathbb{R}^{\text{dmodel}\times\text{dff}}$, $\boldsymbol{W}_2\in\mathbb{R}^{\text{dff}\times\text{dmodel}}$; dmodel 为模型的隐藏层维度; dff 为全连接层的隐藏层维度, 一般取 dmodel 的 4 倍. 除以上计算, 整个前向推理计算流程还包括了残差求和多次的归一化. 对于 GPT-2-124M 模型, 在反复 12 次前向计算后通过模型头完成最终输出. 本文将基于搭建的仿真平台通过系统技术协同优化 (system technology co-optimization, STCO) 逐步完善系统架构, 并在 GPT-2-124M 大模型上完成前向推理计算过程的性能评估.

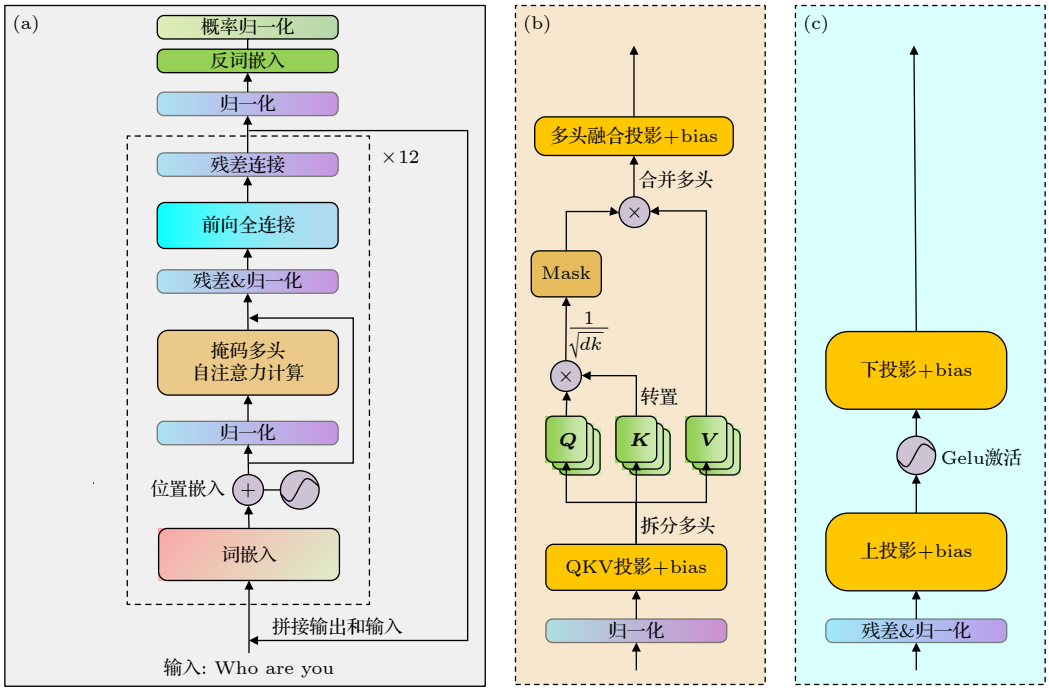

图 1 GPT-2-124M 模型算法示意图 (a) GPT-2-124M 整体算法流程; (b) 掩码多头注意力计 算流程; (c) 前向全连接层计算流程

Fig. 1. Schematic diagram of the GPT-2-124M model algorithm: (a) Overall algorithm flow of GPT-2-124M; (b) masked multi-head attention computation flow; (c) forward fully connected layer computation flow.

3 仿真平台

本研究构建了一套面向 Transformer 推理场景的 3D NAND 存算一体行为级仿真平台. 平台以 Python 为开发语言, 当前版本聚焦 GPT-2 系列模型推理过程的软硬件协同评估. 其总体结构建立在 PyTorch 生态之上. 在保留原生张量表达、自动求导与模块封装接口的同时, 新增硬件模拟子层, 用以替换模型中若干具有高度矩阵乘-加特征的算子 (例如多头注意力中的线性投影与前馈网络的全连接层), 将其计算负载映射到软件定义的 3D NAND 阵列中, 实现对存算一体“原位”矩阵向量乘过程的可控复现与评估. 平台可在同一实验配置下输出纯软件与硬件加速两种方案的对比结果, 支持对性能增益与精度损失的量化分析. 平台的硬件模拟部分对 3D NAND 结构进行了层次化抽象, 包括 3D NAND 阵列结构的系统化建模; 权重的分块、量化与地址映射策略; 对阵列单元的编程、读出与感测时序行为建模.

分析评估模块则围绕性能、功耗与计算误差, 具体如下.

1) 计算时间建模: 将总计算时间分解为算子层的计算时间和反量化层的计算时间, 忽略注意力计算的时间 (和 KV 缓存强相关) 和激活层计算时间.

2) 功耗估计: 统计系统的背景功耗、预充电功耗、块切换功耗、读出功耗和外围电路功耗.

3) 精度与误差: 在算子级提供平均误差、相对误差、余弦相似度等指标, 并在模型级计算输出概率分布, 以评估硬件噪声和量化近似对最终输出概率分布的影响.

为提高行为级结果可信度, 核心电学与时序参数已依据现有 3D NAND 产品数据范围进行初步校准, 例如编程窗口大小、典型读写时间数量级与读写电路数量级功耗, 从而保证在性能与误差数量级上的合理一致性.

为了提升实验效率, 平台实现了交互式参数管理. 用户可通过可视化前端界面快速设定硬件与模型映射参数, 启动多组仿真任务, 并自动记录实验数据、日志与作图结果. 有助于系统性比较不同量化位宽、阵列规模、映射方案或调度策略的综合性能-能效-精度曲线. 平台目前的主要参数类别如表 1 所列, 分为五大类:

1) 硬件架构参数. 平面、层和块的数量、切分参数、ADC(analog-to-digital converter) 分辨率与复用比等.

表 1 仿真平台部分参数表

Table 1. Partial parameter table of the simulation platform.

参数名	功能	参数名	功能
Quantization bits	量化数	Block setup time	时间常数
Current mean /Scale	器件开态电流分布均值/标准差	WL switch time	时间常数
Blocks/Operation	单次计算操作的Block数量	TSG switch time	时间常数
Max current sum	单次计算求和的电流数	BL switch time	时间常数
Symmetric mode	是否采用对称量化	TIA conversion time	时间常数
ADC multiplexing factor	ADC复用数	ADC conversion time	时间常数
X path current	横向通道电流	Planes/Die	硬件常数
Y Path current	纵向通道电流	Layers/Die	硬件常数
Vcc	电压	Blocks/Plane	硬件常数
Background current	背景电流	TSGs/Block	硬件常数
Num of TIAs	TIA数量	Bit lines/Plane	硬件常数

2) 硬件时序参数. 编程读出时间、预充电时间、跨层级调度时间等.

3) 硬件电学参数. 工作电压、横/纵向阵列电流、背景电流等.

4) 模型生成参数. 序列长度、采样策略等.

5) 系统操作参数. 权重常驻/按需加载策略、量化策略等.

4 3D NAND 推理单元 STCO 优化设计

图 2 展示了由余诗孟课题组 [12] 提出的具有 CIM 功能的 3D NAND 闪存的向量矩阵乘法操作 (vector-matrix multiplication, VMM) 及其电路示意图, 其采用位线 (bit line, BL) 作为输入, 源线 (source line, SL) 作为输出. 图中 $W_{i,\mathrm{MSB}}$, $W_{i,\mathrm{LSB}}$ 分别为在 2 bit 精度表示下的权重 W_i 的最高位 bit (most significant bit, MSB) 和最低位 bit (least significant bit, LSB). 由于产品级的 3D NAND 单个块 (block) 的 SL 在电信上互连, 他们定制化了较小的块来保证输出的并发度, 这对于现有的成熟 3D NAND 工艺无疑是增成本降效率的. 同时, 由于时代局限性, 包括余诗孟课题组在内的大部分研究 [13,17–19] 在设计架构时都没有考虑大语言模型的应用, 基本只针对小模型优化了映射模式. 而在大语言模型应用下, 需要采用产品级的 3D NAND 才能承载大模型的参数量. 相较于定制化的 3D NAND 尺寸, 产品级 3D NAND 的块面积更大, 单个块的 TSG(top select gate) 数量一般大于 8, 每个 TSG 所在的 BL 数量一般为 128 kB, 即 131072

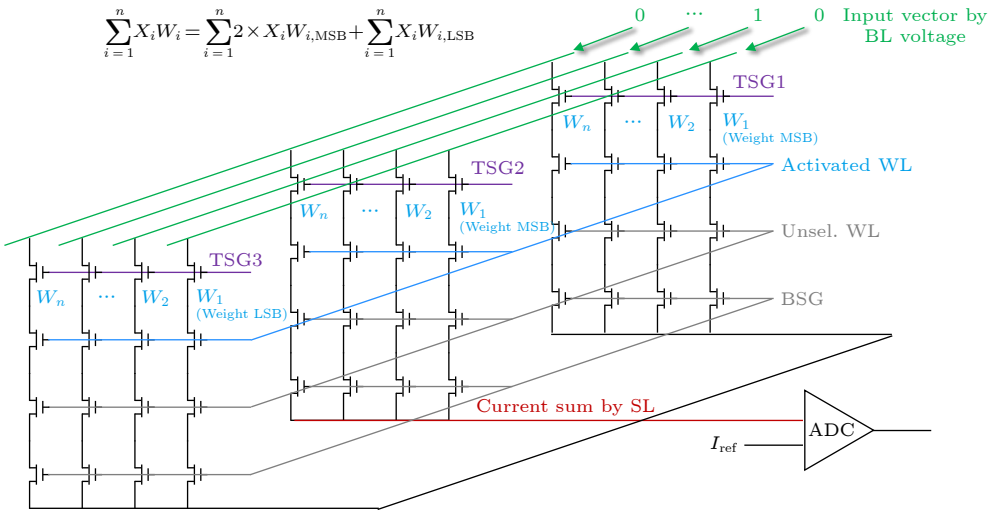

图 2 3D NAND 闪存的向量矩阵乘法操作及其电路示意图

Fig. 2. Vector-matrix multiplication operation and its circuit schematic of 3D NAND flash memory.

个 BL. 单个芯片采用多 Plane 设计, 每个 Plane 的块数> 200, 芯片的总层数>100 层. 对此, 针对大语言模型的 3D NAND 架构需要重新考虑之前的工作中一些设计是否合理, 例如:

1) 定制的 3D NAND Block 模块中极低的开态电流和关态电流很难在产品级 3D NAND 中实现, 目前产品级 3D NAND 开启电流基本大于 100 nA.

2) 十进制的 BL 映射方式占用了大量的 BL, n bit 的输入需要 $2n-1$ 个 BL, 而大模型的隐藏层维度一般都大于 512, 映射 BL 负载过大. 权重映射方式需要 2^n-1 个 Page 来存储权重, 所需 Page 数量随着量化数指数增大, 单个乘积运算消耗的存储单元总数达到 $(2n-1)\times(2^n-1)$, 存储负载极大.

对于数据流, 目前 3D NAND 的 3 种输入模式 TSG/WL(word line)/BL[12,13] 的研究已经相当成熟. 本文在 BL 输入, SL 输出的架构基础上提出一种创新的通用 3D NAND-SS(NAND based on SL slicing) 架构用于承载更大尺寸的模型计算, 同时提出了一种更加灵活的映射方式减少存储负载, 并解决了 SL 并发度过小的问题, 最后针对具体问题进行系统设计协同优化.

4.1 3D NAND-SS 架构概览

如图 3(a) 所示, 整个 3D NAND-SS 芯片分为 3D NAND-SS 阵列、感测电路 (sensing circuit, SC)、加法和移位电路 (adder & shifter)、寄存器 (register)、I/O 接口 (I/O interface) 和控制器 (controller) 六部分. 图 3(b) 为控制器的调度逻辑, 负责对系统逻辑完成整体调度, 具体如下: 1) 3D NAND-SS 阵列存储模型参数完成原位 VMM 计算; 2) 模拟电流量经 SC 和跨阻放大器 (transimpedance amplifier, TIA) 转换后再由 ADC 数字化; 3) 数字量经过数字移位和数字加法电路完成部分和的相加, 期间需要寄存器辅助中间量的暂存; 4) 最终计算结果通过 I/O 接口和外部硬件完成数据交换.

4.2 SL 切分方案

如图 3(c) 所示, 3D NAND-SS 采用标准的 3D NAND 阵列工艺和外围电路设计, 但是在背面沟道的 N 阱区域制造工艺中利用反应离子刻蚀技术进行刻蚀, 分割单个 Block 的源线 N 型多晶硅区域形成多个小的 SL 分区 (以下简记为分区), SL 分区的具体形貌如图 4 所示结构. 然后在一片单独的感测晶圆上布置 ADC 和 TIA 等外围电路, 最后通过晶圆键合技术将感测晶圆的正面和阵列晶圆的背面完成键合. 其中, 感测晶圆上的电路负责对 3D NAND 阵列所产生的模拟电流求和值 (简记为求和值) 进行模数转换, 并通过移位累加实现部分和融合. 通过这样的架构设计, 可以在现有成熟 3D NAND 的制备工艺下仅增加单步刻蚀工艺即可实现 CIM 功能, 同时, 架构中的每个 Block 可以独立向外输出多个分区的求和值, 解决了之前每个 Block 只有一个 SL 并发度的问题, 极大地扩展了算法映射的灵活度.

然而, 对 SL 区域进行切分涉及沿 WL 方向的横向和沿 BL 方向的纵向两个方向, 需要从以下 3 个维度来进行整体考虑:

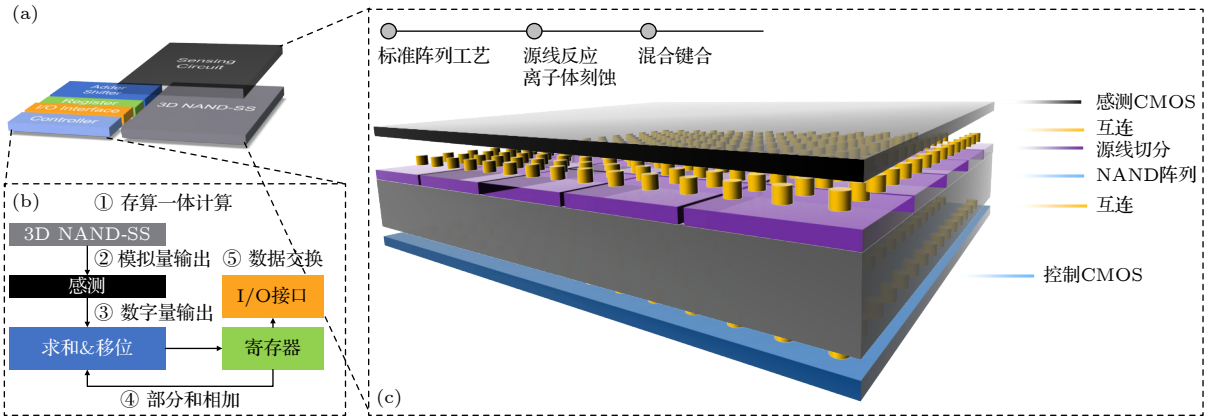

图 3 3D NAND-SS 架构完整示意图 (a) 系统架构示意图; (b) 控制器数据指令流; (c) 3D NAND-SS 阵列示意图

Fig. 3. The 3D NAND-SS architecture: (a) System architecture diagram; (b) controller data and instruction flow; (c) 3D NAND-SS array diagram.

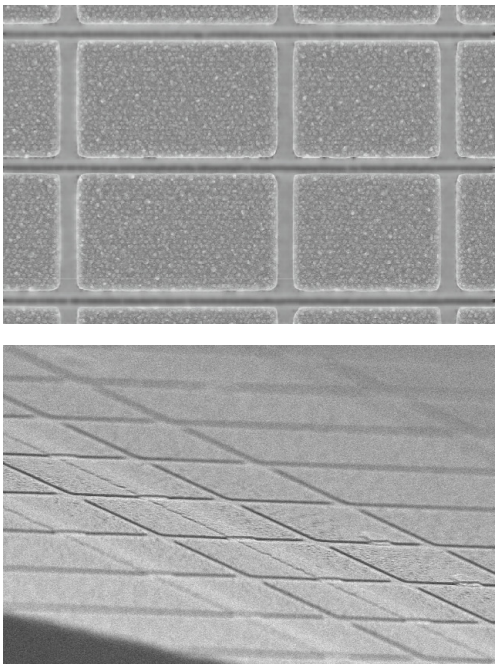

图 4 单个 Block 的源线 N 型多晶硅区域经切分后形成的 SL 分区俯视图/斜视图

Fig. 4. Top and oblique views of the SL partitions formed by segmenting the source-line N-type polysilicon region in a single block.

1) 横向和纵向的切分数不能太稀疏, 否则并行度较低, 对模型权重的映射相当不友好. 极限情况, 切分数为 1 时, 就退化到整体 Block 的情况.

2) 横向和纵向的切分数不能太密集, 否则会使得大量沟道孔失去电信接触从而失去作用, 从而导致阵列单元的浪费率较高, 同时还会增加 ADC 和 TIA 电路的数量, 带来额外的功耗和面积负载.

3) 为了保证分区的求和值是可以被分辨的, 切分后每个分区在横向的沟道孔数存在一个最大值, 即 ADC 最大分辨率.

基于产品级 128 层 3D NAND 的参数, 可以分析得到不同切分情况下的阵列浪费率 L, 具体分析公式如下:

$$L = 1 - \left(\left\lfloor \frac{B_{\text{plane}} - R}{R + B_{\text{lost}}} \right\rfloor + 1 \right) \times \frac{R}{B_{\text{plane}}}, \tag{6}$$

其中, B_{lost} 代表每次纵向切分会损失的 BL 数量, 根据具体工艺而定; B_{plane} 为单个平面的 BL 数量; R 为 ADC 最大分辨率. 公式中没有加入横向切分的分析, 因为对于横向切分维度, Block 之间有足够的空间用于切分, 在每个 Block 之间进行切分是最合理的, 这样横向的阵列损失率可以降到 0, 不会损失沟道孔. 最后, 在本文的工艺条件下将参数代入公式可得阵列浪费率为 3%, 在合理的成本开销范围内. 在其余结构参数上, 如层数, Block 的大小和 VMM 计算时的电压建立时间等参数, 本文均和产品级 3D NAND 保持一致以最小化存储阵列的改动. 综上, 3D NAND-SS 具体结构参数如表 2所列.

表 2 3D NAND-SS 硬件参数

Table 2. 3D NAND-SS hardware configuration.

硬件参数名	值	硬件参数名	值
Plane数每芯片	4	Layer数每芯片	32
Block数每Plane	216	纵向切分数	216
TSG数每Block	10	横向切分数	1024
BL数每Plane	131072	ADC最大分辨率	128

*缩减层数用于简化仿真, 实际产品为128层

4.3 VMM 实现方案

为了在 3D NAND-SS 架构上实现 VMM 计算, 需要将输入和输出进行动态量化, 使得 3D NAND-SS 可以根据基尔霍夫电流定律对量化后的输入和权重完成模拟 VMM 计算. 本文采用 SLC (single-level cell) 单元来存储 1 bit 的信息, 根据每个单元的阈值电压高低反映了 0 和 1, 低阈值电压的单元代表 1, 高阈值电压的单元代表 0, 并采用 4 个或 8 个单元来表示完整的 INT4 或者 INT8 量化权重. 而后使用二进制的 BL 输入方式来表示输入, 即 n bit 的输入用 n 个 BL 来表示, 同样用电压高低来表示 1 和 0. 相较于十进制的 BL 输入方式, 通过此映射方式可以允许更大维度的输入向量. 得益于 3D NAND 原有的操作机制, 可以通过 TSG 和 BL 来交叉选中任意 Block 中的任意沟道孔, 而后通过 WL 来选中需要激活的层, 其余 WL 加上 Pass 电压即可访问被选中沟道孔中被激活层的单元所存储的权重.

在以往的映射算法 [17,19–21] 中, 考虑有符号数的输入和权重映射时都需要复制一份存储单元来实现正负突触的概念, 这极大地浪费了存储空间. 因此, 本文设计了一套映射算法, 将权重和输入按照二进制位的幂次高低顺序拆分到不同的分区中, 通过交叉选中来进行两个二进制有符号数的按位乘法计算. 在写入和计算过程中, 都需要按照规定顺序: 在整个 LLM(large language model, LLM) 计算过程中的固定权重按照 Plane, TSG, Block 和 Layer 的顺序被依次写入到阵列单元中. 例如,

当 (Layer_0, Block_0, TSG_0, Plane_0) 所有 BL 位置对应的单元全都被写入权重后, 需要切换到 (Layer_0, Block_0, TSG_0, Plane_1) 继续写入, 计算时也以相同的顺序依次计算. 以一个较小的 VMM 计算来展示我们的映射算法, 如图 5 所示, 以 INT4量化的计算过程为例进行算法示意.

假设输入和权重的维度都是 2, 首先将输入 (7 –8) 和权重 (3 7) 量化为 INT4, 即 (0111 1000) 和 (0011 0111), INT4 输入按照二进制位的幂次高低顺序拆分到横向的 4 个不同的分区中的 BL_0 至 BL_1 位置, 这 4 个不同分区从左往右所代表的二进制位的幂次逐步增大, 而后INT4 权重先后按照二进制位的幂次高低顺序拆分到纵向的 4 个不同的分区 (即 Block_0 至 Block_1)的 TSG_0 位置, 如图 5 中黑色圆形所示, 每个权重都会沿着横向复制 4 次以保证输入的每个位都与权重的所有位完成一次按位乘法. 权重和输入映射完毕后, 16 个分区同时开始计算, 每个分区中所有被激活的 BL 都将和阵列单元进行计算, 最后在 SL 汇总电流值. 高压的 BL 和低阈值电压的阵列单元产生较大电流, 代表计算结果为 1, 而其余 3 种情况均产生较小电流, 代表计算结果为 0. 最终, 16 个求和完成的电流值将经由 TIA 和 ADC电路转换为数字量, 再根据移位加权表中每个分区的加权数值进行有符号数的数字电路移位运算, 最后在加法器中将移位后的值求和得到最终计算结果. 通过该计算方法, 可以将任意有符号数的 VMM 计算转换为适配 3D NAND-SS 的二进制按位计算. 该映射方法下的计算过程的阵列示意图如图 6 所示, (7) 式为映射后的计算公式:

$$O=\sum_{j=0}^{N-1}\sum_{i=0}^{N-1}\sum_{c=0}^{C_{\mathrm{m}}-1}X_{j,c}W_{i,c}2^{j+i}(-1)^{\left\lfloor\frac{j+1}{N}\right\rfloor\left\lfloor\frac{i+1}{N}\right\rfloor}. \quad (7)$$

其中, $X_{j,c}$ 表示纵向第 j 个分区的第 c 根 BL 上的输入, $W_{i,c}$ 表示横向第 i 个分区的第 c 根 BL 上的权重, N 为量化位数, C_{m} 为 ADC 最大分辨率. 因为引入了分区计算的概念, 与使用传统 3D NAND 的单层阵列完成相同映射时, 小于 0.4% 的单层阵列激活效率相比, 我们的方案可以提供 10% 的单层阵列激活效率, 极大提升了阵列的有效算力. 同时, 每个分区都需要一个 TIA 电路完成电流电压转化以供 ADC 进行数模转化, 因此算力和 ADC 最大分辨率以及 TIA 电路总数强相关, 具体如 (8) 式所示:

$$\mathrm{TOPS}=\frac{R\times M_{\mathrm{TIA}}\times 2}{N\times T}, \quad (8)$$

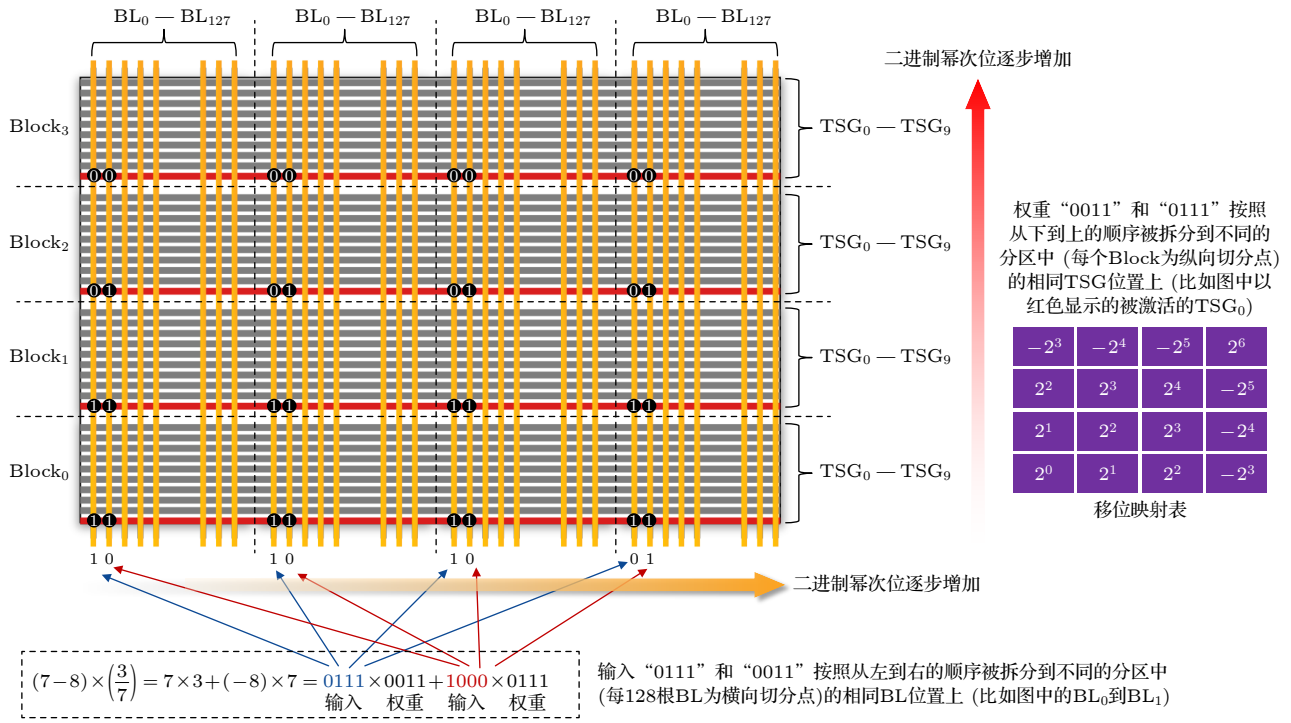

图 5 3D NAND-SS 输入和权重映射方案, 输入和权重均被量化成多个 1 bit 数据沿着横纵两个方向排布

Fig. 5. The 3D NAND-SS input and weight mapping scheme: Both inputs and weights are quantized into multiple 1-bit data arranged in horizontal and vertical directions.

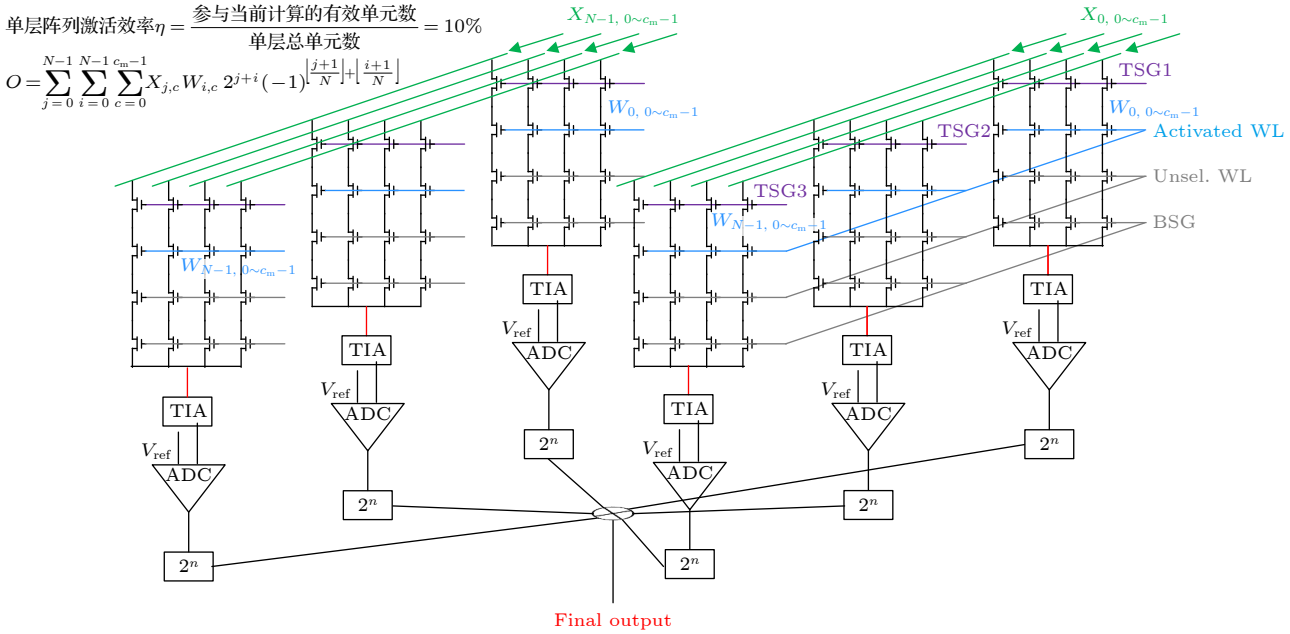

图 6 3D NAND-SS 计算过程阵列示意图

Fig. 6. 3D NAND-SS computational process array schematic diagram.

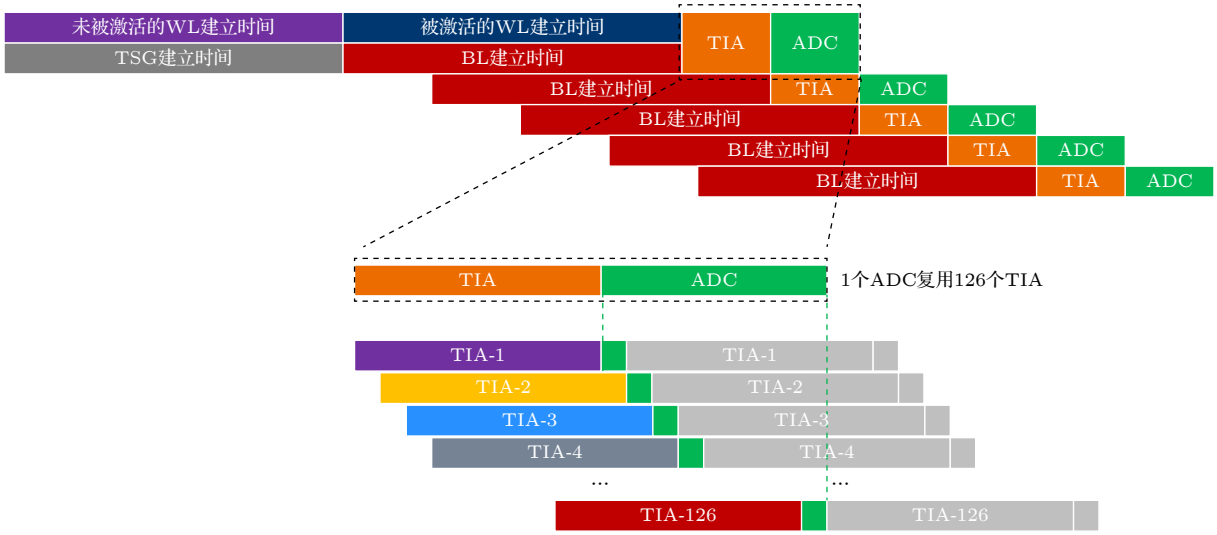

图 7 3D NAND-SS 计算流水线设计, 处理多 TIA 复用 ADC 的情况

Fig. 7. The 3D NAND-SS array computation pipeline design for handling multiple TIA multiplexed ADCs.

其中T代表一次完整计算的时间; M_{TIA}表示TIA电路的数量, 即一次可以转化多少个分区的计算结果; R表示单个分区在X向的沟道孔数, 即ADC的分辨率, 对于输入维度大于R的情况, 需要将输入拆分成多个维度等于R的输入, 分开计算后最后再相加一次; 2为本征操作数, 表示一次操作包含了一次乘法和一次加法.

4.4 流水线设计

如果在某一次VMM计算中, 同时激活了所有的Block和BL, 则会同时输出几十万个分区的求和值, 受限芯片面积, TIA电路的数量无法同时对如此多的求和值完成转换, 因此引入流水线设计是相当必要的. 如图7所示, 结合在4.2节中提到的权重和输入的映射方法, 我们设计了一套遍历3D NAND-SS芯片单层所有Block的流水线方案, 单步只处理等同于TIA电路数量的分区的求和值.

同时, 在STCO设计过程中, 我们注意到ADC和TIA电路的工作速度存在较大差异, 这会导致ADC在整体流水线中经常处于空负载的情况, 因

此可以通过减少 ADC 的数量并增加 TIA 的数量来形成子流水线. 子流水线可以使得单个 ADC 电路同时负责多个 TIA 电路输出结果的转化, 保证单位面积内的算力最大化. 本文所采用的 ADC 工作频率为 500 MHz, TIA 工作频率为 4 MHz, 相差 125 倍, 采用子流水线可以使得单个 ADC 负责处理 125+1 个 TIA 的计算结果.

5 系统级仿真和评估

选用 GPT-2-124M 模型, 总参数量为 124M, 在 INT8 量化下进行了前向推理仿真, 分析系统的计算时间和功耗.

在仿真器中, 由于大语言模型算法的特殊性, 实际上通过 3D NAND-SS 架构实现 VMM 计算加速的模块只有 QKV 的线性投影层、多层感知机 (multi-layer perceptron, MLP) 模块的线性投影层以及 MLP 中下投影层需要的解量化计算, 其余计算包括嵌入、激活、归一化模块、残差连接以及模型头交由 CPU 或 GPU 计算, 因此我们建议嵌入层和模型头都使用内存寻址器 [1,22,23] 来进行寻址计算, 因为它们的本质都是查找表而不是 VMM 计算. GPT-2-124M 模型的权重参数信息如表 3 所列.

5.1 量化选择和电流分布影响

图 8 所示为仿真平台的量化选择和电流分布影响仿真结果, 其中图 8(a) 为不同量化 bit 数下的模型首个 token 的输出分布概率. 我们在不同上下

表 3 GPT-2-124M 模型参数

Table 3. GPT-2-124M model parameters.

模型层名	计算硬件	参数形状	参数量(INT8)
嵌入层	CPU/GPU	(50256, 768)	—
QKV投影层	3D NAND-SS	(768, 2304)	13.5 MB
注意力矩阵>计算	CPU/GPU	(序列长度, 768)	—
注意力矩阵投影	3D NAND-SS	(768, 768)	4.5 MB
多层感知机上投影层	3D NAND-SS	(768, 3072)	18 MB
激活函数	CPU/GPU	—	—
多层感知机下投影层	3D NAND-SS	(3072, 768)	18 MB
多层感知机反量化层	3D NAND-SS	(3072, 768)	18 MB
归一化	CPU/GPU	—	—
残差连接	CPU/GPU	—	—
模型头	CPU/GPU	(768, 50256)	—

注: 仅显示单个注意力模块的参数数量. 在实际算法中, 注意力模块通常是多层的. 对于GPT-2-124M模型, 注意力模块有12层.

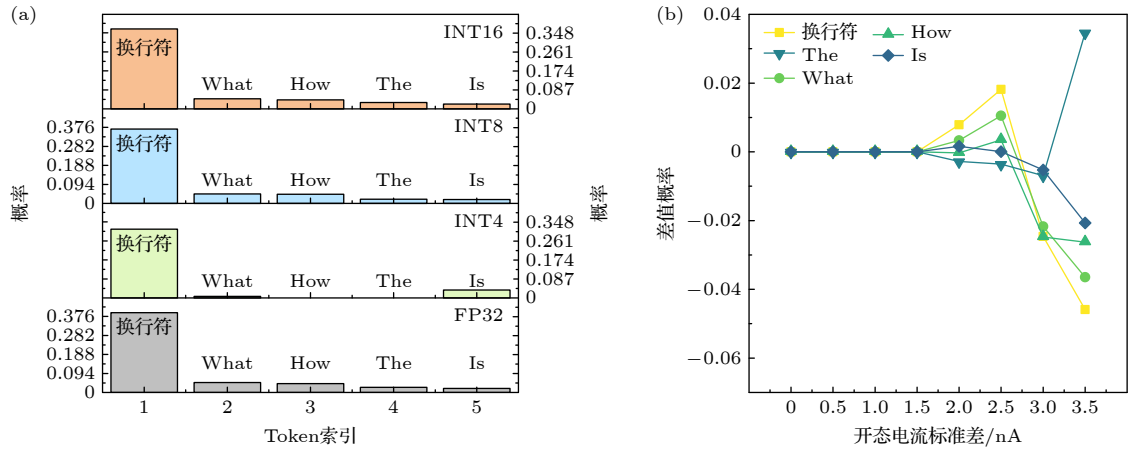

图 8 (a) 不同量化数下模型 token 概率 (提示词: “How is the weather today?”); (b) 不同开态电流分布下 GPT-2-124M 模型输出差值概率分布

Fig. 8. (a) Token probabilities of the model under different quantization bit widths (prompt: “How is the weather today?”); (b) output probability distribution of the GPT-2-124M model under different open-state current distributions.

文提示上, 对 INT4/INT8/INT16/FP32 进行下一 token 分布的 Top-1(最大概率 token) 一致率和 Top-k (k = 5, 10) 做了交集比对. 结果表明 INT8 量化在 GPT2-124 M 上已可接近 FP32 的推理效果, 继续提升至 INT16 边际收益有限. 图 8(b) 显示了不同的器件电流分布对 token 输出概率分布和标准概率分布差值的影响. 结果表示在单次求和 128 个单元电流的情况下, 开态电流均值为 160 nA 时, 开态电流标准差≤1.5 nA 即可保证差值为 0, 即使分布展宽到 2.5 nA, 模型网络本身的鲁棒性也可以保证 token 预测的相对顺序不变.

5.2 误 差

我们在实际进行 VMM 计算时, 对输入和权重采用动态量化, 这会引入量化误差从而直接影响最后的结果. 特别是在第 2 个全连接层之前, 由于 GELU 激活函数改变了输入的分布, 将输入的负数部分进行了压缩, 输入数据的中心点会出现明显的偏移, 此时如果仍然对输入采用对称量化会引入较大的误差. 但是, 非对称量化又会引入额外的解量化计算步骤, 为了避免数据的不必要搬移, 并最小化对激活输入使用非对称量化的硬件开销, 同样可以通过 3D NAND-SS 而不是 GPU/CPU 来计算解量化过程.

解量化输出 D 的计算步骤如 (9) 式所示:

$$D = (\boldsymbol{s}_{\text{in}} \cdot \boldsymbol{s}_{\text{w}}) \odot (\boldsymbol{M} - \boldsymbol{o}_{\text{in}} \cdot \boldsymbol{m}_{\text{w}}), \tag{9}$$

其中, $\boldsymbol{m}_{\text{w}}$ 为量化后的多层感知机下投影层权重矩阵在列方向上的求和向量, $\boldsymbol{s}_{\text{in}}$ 和 $\boldsymbol{s}_{\text{w}}$ 为输入和权重量化的缩放因子向量, o_{in} 为输入非对称量化的零点偏移向量, $\boldsymbol{M}$ 为量化后的输入和权重的计算输出向量. (9) 式中, 只有 $\boldsymbol{m}_{\text{w}}$ 向量需要额外计算, 通过复制一份多层感知机下投影层的权重, 并将输入设置为全 1 向量, 即可等效为在列方向上的求和计算. 如图 9 所示, 在 GPT-2-124M 模型前 12 个注意力模块的第 2 个全连接层中, 对 (INT4/INT8/INT16)×(对称/非对称) 组合测得的平均误差显示: INT8 条件下采用非对称量化较对称量化平均误差下降约 39.1%; INT4 下降 17.8%; INT16 下降 2.2%. 表明非对称量化在中等比特宽 (INT8) 时收益最显著, 高比特 (INT16) 下边际收益减弱, 而低比特 (INT4) 受基准量化误差主导.

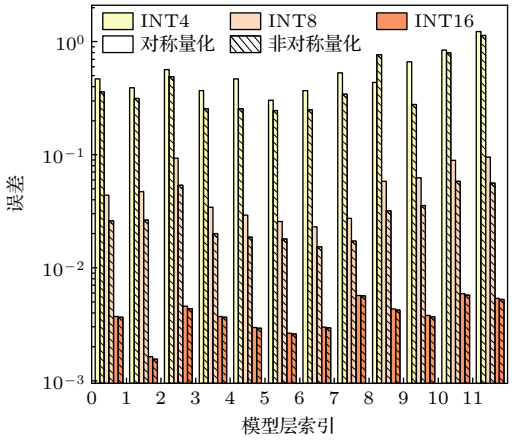

图 9 采用非对称量化和对称量化下的多层感知机下投影层计算误差对比图

Fig. 9. Comparison of computation errors in the MLP down-projection layer under asymmetric and symmetric quantization.

5.3 时间和功耗

根据我们选用的产品级 3D NAND 参数, 3D NAND-SS 的操作电压建立时间和硬件电流参数如表 4 所列, 其中 WL 的建立时间和同时开启的 Block 数量 b_{num} 有关.

表 4 3D NAND-SS 计算时间仿真参数

Table 4. Simulation parameters for 3D NAND-SS computation time.

参数名	值	参数名	值
Block 建立时间/μs	$7b_{\text{num}}$	X通路电流/mA	96.732
BL切换时间/μs	13	Y通路电流/nA	150
WL切换时间/μs	2	V_{cc}/V	2.5
TSG切换时间/μs	0.8	ADC+TIA功率/mW	0.5
TIA 转换时间/μs	0.25	—	—
ADC转换时间/μs	0.002	—	—

注: X通路电流指在单个Plane中建立一个Block的所有WL电压所需时间内的平均电流; Y通路电流指在单个Plane中建立一个BL所需时间的平均电流.

在 GPT-2-124M 模型下处理 10 个 tokens 的时间组成如图 10(a) 所示. 全连接层的上下投影层计算时间最长, 因为权重参数主要分布在全连接层. 整体计算时间为 497 ms, 平均 49.7 ms 处理一个 token, 单位时间的 token 生成速率可达 20 tokens/s. 同时也在 GPT-2-355M 模型中进行时间和功耗分析, 生成速率则为 8.5 tokens/s, 但是能耗比 124M 模型提升了 21%, 展现出架构对大模型的优化潜力.

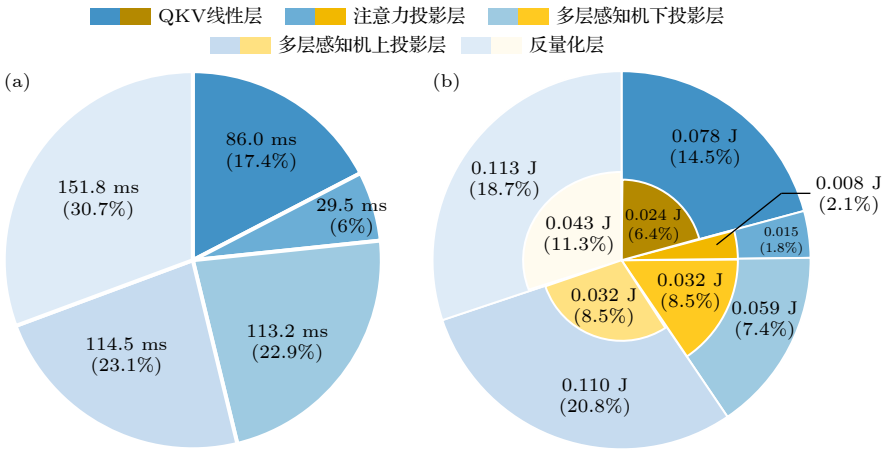

图 10 (a) The 3D NAND-SS 架构生成 10 tokens 的时间组成; (b) 3D NAND-SS 架构生成 10 tokens 的功耗组成

Fig. 10. (a) Time composition for generating 10 tokens in the 3D NAND-SS architecture; (b) power composition for generating 10 tokens in the 3D NAND-SS architecture.

整个系统的功耗主要可以分为 3 个部分, 如图 10(b) 所示. 第 1 部分是正常权重读写和 CIM 计算所产生的阵列功耗 (冷色区块); 第 2 部分是 TIA 和 ADC 模拟电路的感测功耗 (暖色区块); 最后一部分是移位和求和数字电路的功耗, 其中移位和求和数字电路的功耗相较于模拟电路功耗较小, 在仿真中被忽略. 从功耗组成中可以看到感测功耗占总功耗的 1/3 左右, 在未来有很大的优化空间.

综上, 我们通过搭建的仿真器对 3D NAND-SS 的整体性能进行分析. 由于缺乏针对大模型的 3D NAND 架构研究, 选用负载 ResNet-18 模型的 3D NAND 架构和本文架构进行比较. 如表 5 所列, 本方案通过源线切分、分区映射和流水线调度, 使阵列在不改变单芯片存储物理容量前提下, 实现了灵活的大语言模型算子映射, 具备在大模型下的功耗比优化潜力; 并围绕注意力与前馈结构进行了量化与误差抑制优化, 从而提升对大语言模型的适配效率.

表 5 综合对比

Table 5. Benchmark.

器件技术节点	32 nm 3D NAND[11]	40 nm 3D NAND-SS	40 nm 3D NAND-SS
ADC精度/bit	7	7	7
Cell精度/bit	1	1	1
面积/mm^2	17.91	40	40
容量利用率/%	33.5 @INT8	17 @INT8	60 @INT8
算力/TOPS	0.0018	4.57	4.57
能耗比/(TOPS·W^{-1})	12.95 @INT8	5.93 @INT8	7.17 @INT8
负载模型	ResNet-18	GPT-2-124M	GPT-2-355M

6 结 论

本文针对大语言模型本地化部署的算力与能效挑战, 提出了一种基于产品级 3D NAND 闪存的存算一体系统解决方案, 并结合自研仿真平台进行了完整性能分析.

1) 通过引入 SL 背面切分工艺 (3D NAND-SS 架构), 在现有工艺基础上仅增加一步刻蚀步骤, 即可实现 Block 的多分区并行计算能力. 该设计将阵列浪费率控制在 3% 以内, 并通过混合键合技术集成高密度 ADC/TIA 电路, 显著提升了硬件资源利用率和大模型权重映射的灵活性.

2) 开发了基于二进制有符号数量化的 VMM 映射算法, 结合分级流水线设计, 降低了计算时间和功耗. 仿真结果表明, 该架构可实现 4.57 TOPS 的峰值算力, 单芯片推理 INT8 GPT-2-124M 模型的速度达到 20 tokens/s, 功耗比达 5.93 TOPS/W; 推理 INT8GPT-2-355M 模型的速度达 8.5 tokens/s, 功耗比达 7.17 TOPS/W, 具备在大模型下的功耗比优化潜力. 模型推理精度基本只受限于量化误差, 可用于正常的推理任务应用.

3) 采用非对称量化映射方案进一步提升了推理精度. 并通过对电流分布和量化误差的系统级仿真验证, 证实了架构在 160 nA 平均开启电流、$\sigma < 2.5$ nA 范围内的可靠性.

架构本身可通过现有 NAND 芯片封装技术封装多个 NAND 芯片实现多通道计算, 大幅度提升算力并承载更大的模型. 本文首次在国产 3D NAND

芯片上完成大语言模型的系统验证, 提出的架构设计为 3D NAND 存算一体芯片的量产化提供了可行路径. 考虑到目前的仿真器只能计算 3D NAND 阵列部分的相关性能, 没有做到完整的系统仿真, 后续研究将聚焦在 3D NAND 的器件级性能优化和仿真器算法完善上, 完整仿真器算法已发布在 GitHub 社区, 社区链接如下: https://github.com/zhenghao21/CIM-Simulator-based-3D-NAND.

参考文献

[1] Singh Parihar S, Kumar S, Chatterjee S, Pahwa G, Singh Chauhan Y, Amrouch H 2025 *IEEE J. Explor. Solid-State Comput. Devices Circuits* **11** 34

[2] Molom-Ochir T, Taylor B, Li H, Chen Y R 2025 *IEEE Trans. Circuits Syst. I* **72** 3971

[3] Wu B, Lv X R, Yu T Y, Chen K, Liu W Q 2025 *IEEE Nanotechnol. Mag.* **3** 19

[4] Li H W, Yao E Y, Qin P, Jiang S 2025 *IEEE Trans. Magn.* **61** 3401306

[5] Khwa W S, Wen T H, Hsu H H, Huang W H, Chang Y C, Chiu T C, Ke Z E, Chin Y H, Wen H J, Hsu W T, Lo C C, Liu R S, Hsieh C C, Tang K T, Ho M S, Lele A S, Teng S H, Chou C C, Chih Y D, Chang T Y J, Chang M F 2025 *Nature* **639** 617

[6] Sharma V, Zhang X, Dhakad N S, Kim T T H 2025 *IEEE Trans. Circuits Syst. I* **72** 5696

[7] Liu S Q, Wei S T, Yao P, Wu D, Jie L, Pan S N, Tang J S, Gao B, Qian H, Wu H Q 2025 *J. Semicond.* **46** 062304

[8] Chang S H, Yen R H, Liu C N 2025 *ACM J. Emerg. Technol. Comput. Syst.* **21** 4

[9] Zhang Y Q, Wang J J, Lv Z Y, Han S T 2022 *Acta Phys. Sin.* **71** 148502 (in Chinese) [张宇琦, 王俊杰, 吕子玉, 韩素婷 2022 物理学报 **71** 148502]

[10] Shim W, Yu S 2021 *IEEE J. Explor. Solid-State Comput. Devices Circuits* **7** 1

[11] Hong Y, Kim M, Kim C 2025 techrxiv: 174439324.42202505

[12] Shim W, Yu S M 2021 *IEEE J. Explor. Solid-State Comput. Devices Circuits* **7** 61

[13] Shim W, Yu S M 2021 *IEEE Electron Device Lett.* **42** 160

[14] Chen Y Y, He Y H, Miao X S, Yang D H 2022 *Acta Phys. Sin.* **71** 210702 (in Chinese) [陈阳洋, 何毓辉, 缪向水, 杨道虹 2022 物理学报 **71** 210702]

[15] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I 2017 *Proceedings of the 31st International Conference on Neural Information Processing Systems* Long Beach, California, USA, December 4–9, 2017 p6000

[16] Hanna M, Liu O, Variengien A 2023 *Adv. Neural Inf. Process. Syst.* **36** 76033

[17] Lue H T, Hsu P K, Wei M L, Yeh T H, Du P Y, Chen W C, Wang K C, Lu C Y 2019 *2019 IEEE International Electron Devices Meeting (IEDM)* San Francisco, USA, December 9–11, 2019 p38.1.1

[18] Kim M, Liu M, Everson L, Park G, Jeon Y, Kim S, Lee S, Song S, Kim C H 2019 *2019 IEEE International Electron Devices Meeting (IEDM)* San Francisco, USA, December 9–11, 2019 p38.3.1

[19] Kang M, Kim H, Shin H, Sim J, Kim K, Kim L S 2022 *IEEE Trans. Comput.* **71** 1291

[20] Lee S T, Yeom G, Yoo H, Kim H S, Lim S, Bae J H, Park B G, Lee J H 2021 *IEEE Trans. Electron Devices* **68** 3365

[21] Lee S T, Lee J H 2020 *Front. Neurosci.* **14** 571292

[22] Wong R, Kim N, Higgs K, Agarwal S, Ipek E, Ghose S, Feinberg B 2024 arXiv: 2403.06938 [cs.AR]

[23] Hsieh C C, Lue H T, Li Y C, Hung S N, Hung C H, Wang K C, Lu C Y 2023 *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)* Kyoto, Japan, June 11–16, 2023 p1

A compute-in-memory architecture and system-technology codesign simulator based on 3D NAND flash

ZHENG Hao [1)2)] LIU Huiwen [3)] FANG Yuxuan [3)] FAN Dongyu [3)]
HAN Yuhui [3)] HOU Chunyuan [3)] LIU Wei [3)] XIA Zhiliang [3)†]
HUO Zongliang [1)3)‡]

1) (*Institute of Microelectronics, Chinese Academy of Sciences, Beijing* 100029, *China*)
2) (*University of Chinese Academy of Sciences, Beijing* 100049, *China*)
3) (*Yangtze Memory Technology Corp, Wuhan* 430070, *China*)

(Received 8 July 2025; revised manuscript received 1 October 2025)

Abstract

The rapid advancement of large language models (LLM) such as ChatGPT has imposed unprecedented demands on hardware in terms of computational power, memory capacity, and energy efficiency. Compute-in-memory (CIM) technology, which integrates computing directly into memory arrays, has become a promising solution that can overcome the data movement bottlenecks of traditional von Neumann architectures, significantly reduce power consumption and achieve large-scale parallel processing. Among various non-volatile memory candidates, 3D NAND flash stands out due to its mature manufacturing process, ultrahigh density, and cost-effectiveness, making it a strong contender for commercial CIM deployment and local inference of large models.

Despite these advantages, most of existing researches on 3D NAND-based CIM remain at an academic level, focusing on theoretical designs or small-scale prototypes, with little attention paid to system-level architecture design and functional validation using product-grade 3D NAND chips for LLM applications. To address this gap, we propose a novel CIM architecture based on 3D NAND flash, which utilizes a source line (SL) slicing technique to partition the array and perform parallel computation at minimal manufacturing cost. This architecture is complemented by an efficient mapping algorithm and pipelined dataflow, enabling system-level simulation and rapid industrial iteration.

We develop a PyTorch-based behavioral simulator for LLM inference on the proposed hardware, evaluating the influences of current distribution and quantization on system performance. Our design supports INT4/INT8 quantization and employs dynamic weight storage logic to minimize voltage switching overhead, and is further optimized through hierarchical pipelining to maximize throughput under hardware constraints.

Simulation results show that our simulation-grade 3D NAND compute-in-memory chip reaches generation speeds of 20 tokens/s with an energy efficiency of 5.93 TOPS/W on GPT-2-124M and 8.5 tokens/s with 7.17 TOPS/W on GPT-2-355M, respectively, while maintaining system-level reliability for open-state current distributions with $\sigma < 2.5$ nA; in INT8 mode, quantization error is the dominant accuracy bottleneck.

Compared with previous CIM solutions, our architecture supports larger model loads, higher computational precision, and significantly reduced power consumption, as evidenced by comprehensive benchmarking. The SL slicing technique keeps array wastage below 3%, while hybrid wafer-bonding integrates high-density ADC/TIA circuits to improve hardware resource utilization.

This work represents the first system-level simulation of LLM inference on product-grade 3D NAND CIM hardware, providing a standardized and scalable reference for future commercialization. The complete simulation framework is released on GitHub to facilitate further research and development. Future work will focus on device-level optimization of 3D NAND and iterative improvements of the simulator algorithm.

Keywords: 3D NAND, compute-in-memory, hardware acceleration

PACS: 85.40.–e **DOI:** 10.7498/aps.74.20250891

CSTR: 32037.14.aps.74.20250891

† Corresponding author. E-mail: albert_xia@ymtc.com
‡ Corresponding author. E-mail: zongliang_huo@ymtc.com

物理学报 Acta Physica Sinica
Chinese Physical Society
Institute of Physics, CAS

一种基于3D NAND存储器的存算一体架构及其系统技术协同优化仿真

郑好 刘慧雯 方语萱 范冬宇 韩玉辉 侯春源 刘威 夏志良 霍宗亮

A compute-in-memory architecture and system-technology codesign simulator based on 3D NAND flash

ZHENG Hao LIU Huiwen FANG Yuxuan FAN Dongyu HAN Yuhui HOU Chunyuan LIU Wei XIA Zhiliang HUO Zongliang

引用信息 Citation: Acta Physica Sinica, 74, 248502 (2025) DOI: 10.7498/aps.74.20250891
CSTR: 32037.14.aps.74.20250891
在线阅读 View online: https://doi.org/10.7498/aps.74.20250891
当期内容 View table of contents: http://wulixb.iphy.ac.cn
