

工程师学院硕士专业学位论文开题报告



**中文论文题目 ： 基于存内计算的快速傅里叶**

**变换设计**

**英文论文题目 ： Fast Fourier Transform Design**

**Based on Computing in Memory**

姓 名： 刘润

校内导师： 黄科杰

校外合作导师： 陈华锋

专业学位领域： 集成电路工程

日期： 2020年11月10日

摘要

快速傅里叶分析技术（FFT）广泛应用于当代工程，科学和数学领域，并在以神经网络为代表的人工智能技术领域也占用重要地位，例如在语音关键词检测（KWS）领域，FFT就是语音信号特征提取的重要步骤。传统的冯诺依曼架构对于高并行，高数据吞吐的FFT计算不仅会产生严重的“内存墙”效应，还会产生大量能耗，限制硬件效能。而存内计算(CIM)技术则突破了传统的冯诺依曼瓶颈，实现了存储单元和逻辑单元的融合，故基于存内计算技术的低功耗FFT设计方案是很有前景的。同时，对于关键词检测来说，大部分研究集中于如何使用存内计算实现后端神经网络分类器，而较少考虑将其应用于前端特征提取部分，尤其是FFT的设计。所以为了进一步降低关键词检测芯片的设计功耗，FFT使用存内计算实现是很用意义的。

关键词：快速傅里叶变换，关键词检测，存内计算

Abstract

Fast Fourier Transform (FFT) is widely used in the fields of contemporary engineering, science and mathematics, it also occupies an important position in the artificial intelligence field represented by neural network. For example, in the field of keyword spotting (KWS), FFT is an important step of speech signal feature extraction. Traditional von Neumann architecture for high parallel, high data throughput FFT computing will not only produce a serious "memory wall" effect, but also produce a lot of energy consumption, limiting the hardware performance. Computing in memory (CIM) technology breaks through the bottleneck of traditional von Neumann and realizes the integration of storage unit and logic unit. Therefore, low power FFT design scheme based on CIM technology is very promising. At the same time, for KWS technology, most of the research focuses on how to use CIM technology to implement the back-end neural network classifier, and less attention is paid to the application of CIM technology to the front-end feature extraction part, especially the design of FFT. Therefore, in order to further reduce the design power consumption of KWS chip, it is also very useful to using CIM technology to realize FFT design

Keywords: Fast Fourier Transform, Keyword Spotting, Computing in Memory

目录

摘要………………………………………………………………………1

目录………………………………………………………………………2

1. 研究背景与研究意义 …………………………………………3
   1. 研究背景…………………………………………………3
   2. 研究意义…………………………………………………5
2. 研究现状与趋势 ………………………………………………5

2.1 研究现状…………………………………………………5

2.2 研究趋势…………………………………………………6

2.3 研究目的…………………………………………………7

1. 研究内容 ………………………………………………………7

3.1 设计创新…………………………………………………7

3.1.1 基于CIM的FFT设计……………………………7

3.1.2 FFT混合基分解…………………………………8

3.2 计算过程…………………………………………………9

3.2.1 理论计算过程……………………………………9

3.2.2 FFT存内计算方法………………………………11

3.2.3 性能参数估计 …………………………………14

3.3 研究计划 ………………………………………………15

参考文献 ………………………………………………………………16

基于存内计算的快速傅里叶变换设计

刘润 21860268 工程（集成电路工程）

1. 研究背景与意义

1.1 研究背景

傅里叶分析是一种广泛应用于工程，科学和数学等领域的将信号从原始域变换到频率域的分析方法，快速傅里叶变换（FFT）是其在实际中常用到的快速计算方法。在人工智能研究大发展的今天，以神经网络技术为代表的人工智能技术推动了各个领域的研究发展[1][2][3][4]，而在语音关键词检测(KWS)领域，深度神经网络技术得到了广泛应用并取得了良好的性能指标，例如，Chen[5]首先将深度神经网络应用到关键词检测任务，开创了这一研究方法的先河，随后卷积神经网络(CNN)[6]，卷积循环神经网络(CRNN)[7]，深度可分离卷积神经网络(DSCNN)[8]和深度卷积残差网络(ResNet)[9]等新兴轻量级神经网络模型也被应用到了关键词检测任务中。在网络模型快速更新换代的同时，为了得到高算力，高效能，低存储的模型推理计算，各种专用神经网络计算芯片也应运而生。针对于关键词检测任务，Yin[10]设计了一款基于优化二值卷积神经网络的语音处理芯片，最高识别准确率达到了98.6%，并且功耗仅为141uW; 同时一款基于65纳米技术的SoC芯片[11]将关键词检测任务设计功耗降低到了18uW;而最近Shan[12]设计了一款基于28nm CMOS制程工艺的专用芯片以进行模型推理，该设计采用深度可分离卷积神经网络模型，对于单个词的识别精度达到98%，其设计功耗更是低至510nW。

在传统的冯洛伊曼计算架构中，存储模块和计算模块是分离的，这种结构会产生严重的“内存墙”效应。

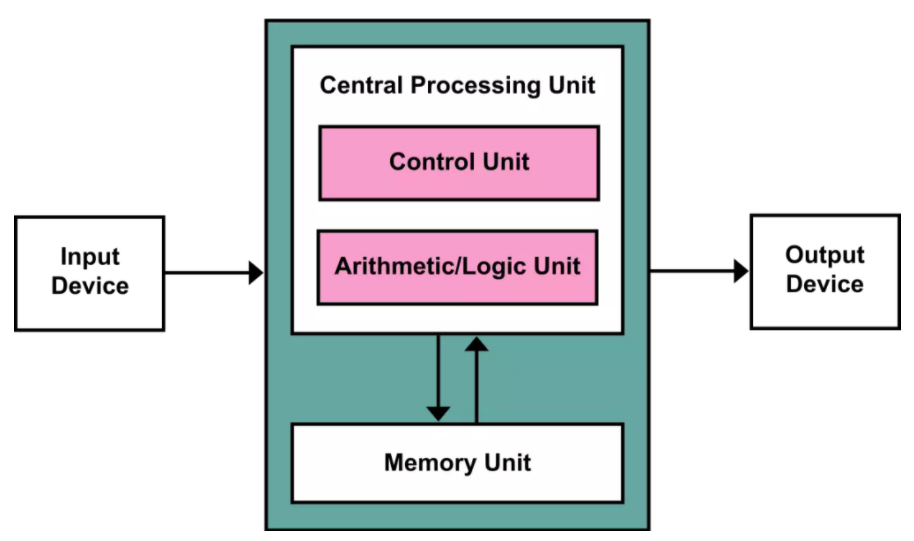


图1：经典冯诺依曼计算结构

进入AI时代后，大数据量，大计算量的计算特点不仅使冯诺依曼架构芯片运行速度缓慢，更严重的是会产生大量的能耗。存内计算技术[13]是一种采用新兴存储单元同时实现存储和计算的一种技术，基于存内计算技术设计的芯片架构，不仅实现了存储和计算的同一性，消除了“内存墙”效应，降低了功耗，并且其本身乘加和累计的计算模式也很适合现代以卷积神经网络为代表的AI模型的计算与处理[14][15]。

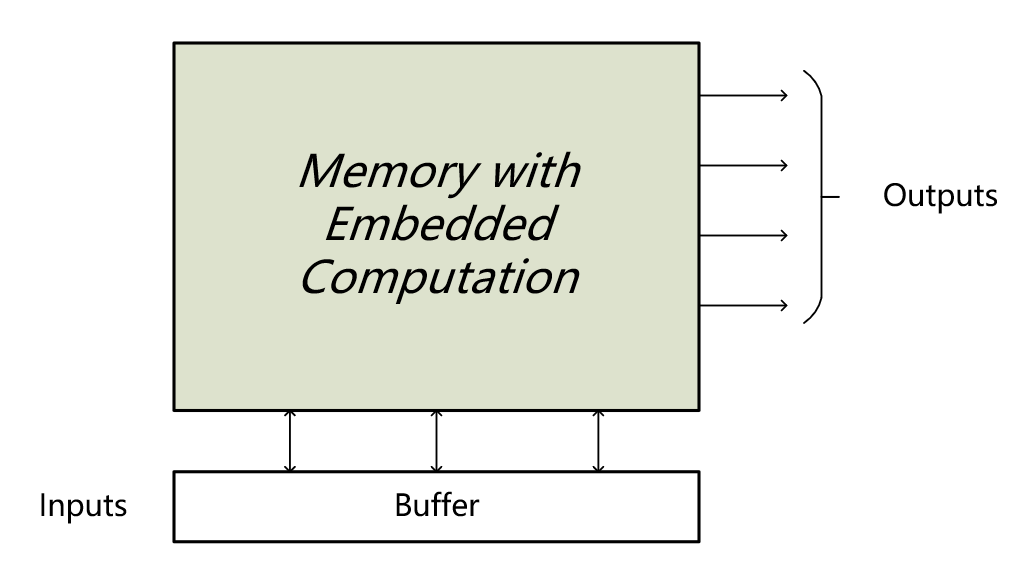


图2：存内计算设计结构

在这种情况下，存内计算技术也被应用到关键词检测领域的芯片设计中，而其主要的目的是为了破除冯诺依曼架构中存在的“存储墙”效应，以便得到极低功耗的关键词检测芯片。M.Price[16]实现了一种基于存内计算的可裁剪的语音神经网

络处理器来进行语音识别；而这篇工作[17]中，在移动设备的存储器中实现了288uW的可编程的深度学习处理器；或者使用新型的存储单元XNOR-SRAM来实现三值或者二值网络[18];为了得到处理速度和处理功耗的双向性能提升，可以采用多存内计算核的方式[19]来设计芯片；也可以采用新型存储器件线性可编程RAM器件来构成存内计算的基本单元来实现关键词检测任务[20]。在实际应用场景中的关键词检测任务，往往需要随时检测是否有相应词语语音的输入，也就是需要长时间处于任务开启状态，而存内计算的低功耗特性很适合用于对功耗敏感的终端设备中实现关键词检测任务。

1.2 研究意义

在各种采用存内计算技术设计的关键词检测芯片中，大部分的研究集中于后端神经网络分类器的实现，然而在关键词检测模型中，前端特征提取部分也占有重要地位[21]。一方面特征提取结果的好坏直接影响着后端神经网络分类的精度，另一方面特征提取中对于语音信号进行时域到频域的转换时，也会产生大量功耗和中间存储空间，尤其是其中的FFT过程。所以，对于特征提取的优化是进一步降低KWS芯片功耗和存储空间的重要方面，而其中关键的优化目标是降低特征提取中产生大数据量计算和功耗的FFT过程。

针对于特征提取过程中FFT计算是大量矩阵运算的实质，FFT计算使用存内计算技术是很适合的。对于特征提取和神经网络卷积同时使用了存内计算技术的KWS芯片来说，进一步大幅度地降低功耗设计可以预见的，这也是关键词检测设计方案优化的重要方向。同时，对于其他采用了将数据从原始域转换到频率域的神经网络模型[22][23][24]来说，使用存内计算技术优化其转换方式，也是其硬件实现优化的一个可行方向。

2. 研究现状及趋势

2.1 研究现状

最近以来，FFT的研究主要集中于使用各种方法提升计算速率和减少硬件成本。例如使用单端口存储器将存储器合并为4个存储器[32]，进行任意基2FFT计算，以使得输入输出可以与内部处理相同的并行度运行，提升吞吐率；或者使用电容作为存储单元设计异步随机加法器以改善FFT设计的面积[33]；亦或是使用低秩张量逼近简化FFT计算过程[34]；使用张量分解存储旋转因子矩阵，减少内存消耗[35]。通过诸如张量分解和调整存内结构的方法优化FFT计算，但是对于使用新型存内计算技术优化FFT计算的研究并没有特别多的展现。基于存内计算的芯片设计研究得到越来越多的重视，很多芯片的设计兼顾了算力的提升和功耗的降低。基于存内计算技术设计的芯片既可以实现机器学习任务的训练过程[25]，也可以在边缘设备中实现推理功能[26]；既可以运行多比特的网络模型[29]，也可以运行轻量级的二值网络[27][28]；既可以在存内计算模块完成整体模型计算，也可以只专注于网络中常用的乘加操作[30]。针对于关键词检测任务，如1.2节所述，大多数研究侧重于后端神经网络分类在存内计算模块中的实现。表1中所列是使用了存内计算技术实现关键词检测模型的芯片设计，其中大部分都忽视了使用存内计算实现前端特征提取尤其是FFT计算的可能性，少部分使用了存内计算技术计算FFT的芯片也只是使用了离散傅里叶变换（DFT）矩阵计算模式来进行实现，没有考虑对于FFT设计进行进一步的优化。

表1：关键词检测存内计算研究对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ISSCC 2017  [16] | ISSCC 2017  [17] | VLSI 2018  [18] | VLSI 2019  [19] | IEDM 2020  [20] |
| Technology | 65nm | 40nm | 28nm | 65nm | 90nm |
| Memristor | 5.84Mb SRAM | 144KB SRAM | 27KB SRAM | 10KB SRAM+64Kb CIM-SRAM | PLRAM |
| Accuracy | 92% | - | 95% | 95% | >90% |
| Throughout  (TOPS/W) | - | 0.318@0.65V,  3.9MHz | 54.4@0.57V,  2.5MHz | 11.7 @ 0.9V, 75MHz | 11 |
| Neural Energy Efficiency  (pJ/Neuron) | 36@0.9V,10.2MHz | 2540@ 0.65V,  3.9MHz | 2.46@0.57V,  2.5MHz | 5.1 @ 0.9V, 75MHz | - |
| Bits | - | - | - | 3bits | 7bits |

2.2 研究趋势

由于传统的DFT矩阵计算模式中旋转因子的矩阵比较大，而如果不考虑对于DFT进行相应的优化，很难使用存内计算技术直接进行矩阵计算。在关键词检测模型中，特征提取的FFT点数一般都为512点，如果直接进行矩阵计算，使用的旋转因子矩阵则是512\*512，这对于存内计算模块来说具有相当大的设计和制造压力。对于高点数FFT计算，寻求基于FFT分解的方法，将大型矩阵计算分解为多个小型矩阵计算是比较合理的，而这对于有效降低存内计算资源，提升芯片整体效能也有很大的好处。基于这种设计理念，语音信号的处理在未来不仅仅可以采用存内计算技术计算后端神经网络，更可以将前端特征提取部分的计算也使用该项技术以达到大幅度降低功耗的目的。与此同时，针对于FFT的优化设计研究也可以成为一个重点，通过传统的FFT分解方式，甚至是基于编译指令控制流程的方式，基于合理软件调度，增加存内单元可重用性的方式，都会是未来关于关键词检测芯片设计的研究方向。

2.3 研究目的

基于FFT应用广泛，矩阵计算和计算量较大的特点，与存内计算技术相结合，以存内计算的乘加模式实现矩阵运算，进而减少数据的访存次数，并以此来降低设计功耗。而本设计的研究目的也就是探索两者结合的方案可行性，进而设计具体的低功耗可执行方案，最后基于关键词检测的特定任务来进行相关优化，以达到大幅度降低关键词检测芯片设计的功耗的目的。

3. 研究内容

3.1. 设计创新

3.1.1. 基于CIM的FFT设计

由前文可得，使用基于存内计算技术实现FFT计算的设计较少，针对于关键词检测任务而特定设计的FFT计算更少。针对于FFT大计算量大数据量的特点，使用存内计算技术实现FFT可以有效降低设计功耗，对于需要长时间待机工作的关键词检测芯片的功耗降低可以做出很好的贡献。于此同时，存内计算模式和FFT矩阵相乘的乘加模式很匹配，将FFT计算与存内计算结合是非常合适的。

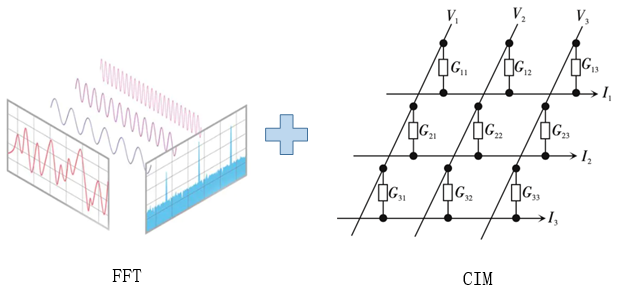


图3：FFT和CIM结合示意图

3.1.2. FFT混合基分解

传统FFT当计算点数较大时，使用矩阵乘法形式计算十分不便，尤其是使用存内计算技术。对于大点的FFT计算，可以采用混合基分解形式将一个大型旋转因子矩阵分解为多个小型旋转因子矩阵，从而在存内计算模块中友好地实现。对于N点DFT，如果N是一个复合数，它可以分级为成为一些因子的乘积，则可以使用FFT的一般算法，即混合基FFT算法，而常用的基-2算法是这种一般算法的特例[31]。

若N可以表示为复合数，则对于的任何一个正整数n，都可按照L基，，…, 表示为多基多进制形式,

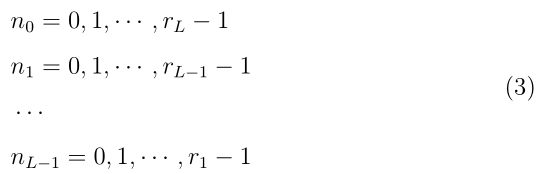
这一多基多进制所代表的数值为



其倒位序形式为=，它所代表的数值为：



在这一多基多进制的表示中：

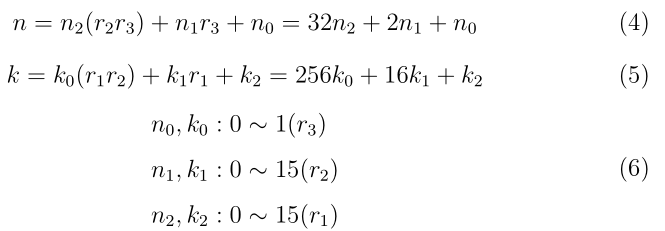


在关键词检测模型中，语音信号的特征提取中使用的FFT一般为512点，所以本设计以512点FFT进行混合基分解。512点FFT可以分解为多个小点FFT的级联计算，需要选择合适的级联数来减小功耗和资源的使用。根据文献[36]可以得到256\*256矩阵的存内计算核的功耗，据此下表列出了三种级联方式以及不同方式所对应的功耗和资源利用情况。

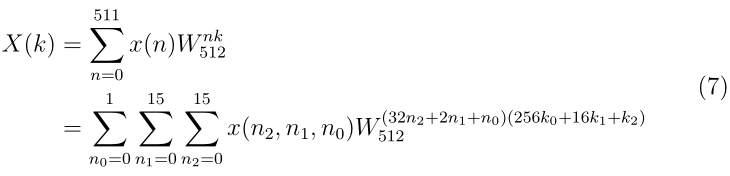
表2：不同级联方式的功耗和资源消耗情况

|  |  |  |
| --- | --- | --- |
| 级联方式 | 功耗 | 存内资源 |
| 16\*32 | 28.88mW | 32KB |
| 16\*16\*2 | 21.66mW | 24KB |
| 8\*8\*8 | 36.10mW | 40KB |

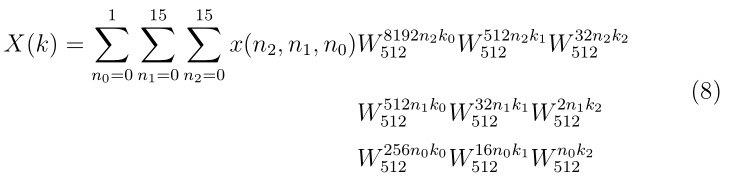
从上表可得，将512点FFT分级为16\*16\*2三级从功耗和资源利用方面来看是比较合适的。将512点FFT分解为16\*16\*2三级，即N=16\*16\*2，则，，，，则有：

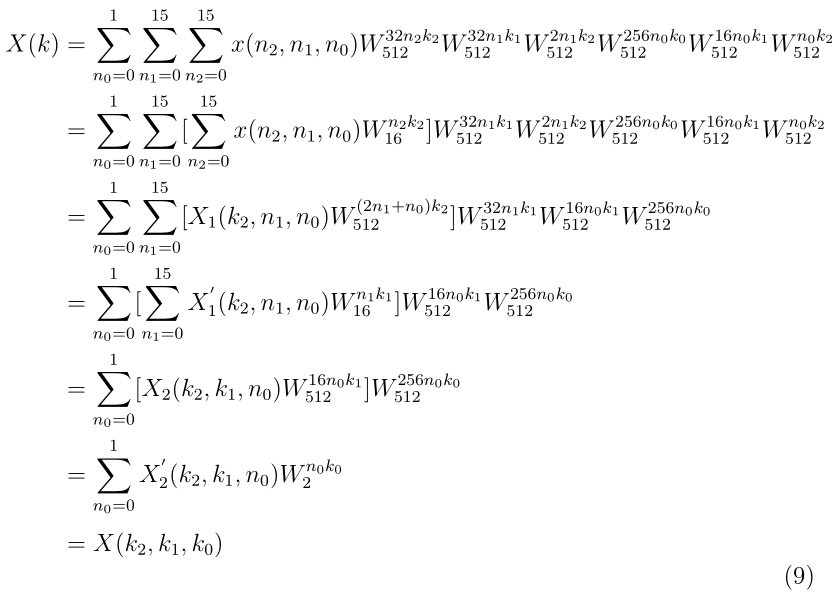


所以根据DFT公式，则将公式(4)~(6)带入可得：

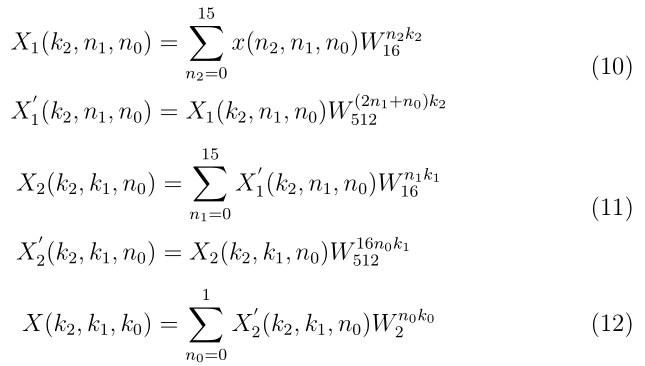


可以得到





并且有



综上可以看出，512点FFT可以拆解为16\*16\*2的三级FFT的级联，公式(10)为第一级16点FFT，公式(11)为第二级16点FFT，公式(12)为第三级2点FFT。

3.2. 计算过程

3.2.1. 理论计算过程

根据3.2.1小节中的公式，可以得出结论，512点FFT计算可以分解为三级较小规模的FFT计算，通过公式(4)~(6)可以得知原始数据的输入次序和计算得到数据的输出次序。由于第三级2点FFT可以使用简单加减计算实现，所以设计重点在于第一级和第二级16点FFT的实现。原始数据为x(0)，x(1)，…, x(510)，x(511)，输入时的读入次序为。固定和，在的范围内进行变化，就可以得到一次FFT16计算所需要的原始数据的排列次序。

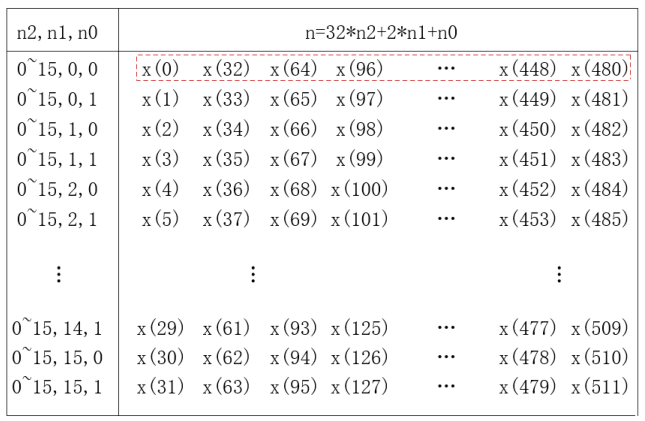
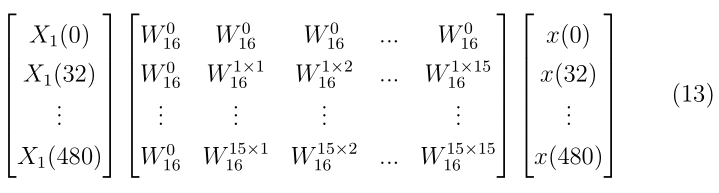
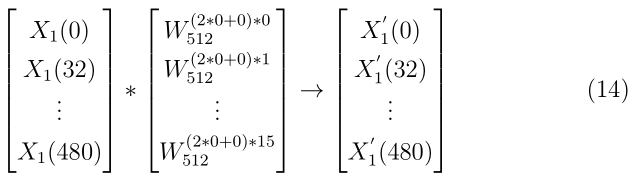


图4：第一级FFT16输入次序

原始数据以行为单位并行输入16个数据，作为第一级FFT计算。根据式(10)可以得知，第一级FFT16并行输入16个数据之后，和存储在FLASH模块中的旋转因子相乘，经过ADC采样之后，得到，再与相乘后完成第一级FFT计算。所以对于第一级FFT16来说，以第一行数据为例，其矩阵运算模式为：



再将得到的向量与向量元素对应相乘即可。



重复上述过程，即可完成第一级FFT16计算。随后在第二级FFT16计算中，还是按照公式进行，只不过此时信号输入数据为第一级FFT16的计算结果，并且每一次计算的次序按照进行，即固定和，在的范围内进行变化。将红框内的数据并行输入，进行第二级FFT16的计算

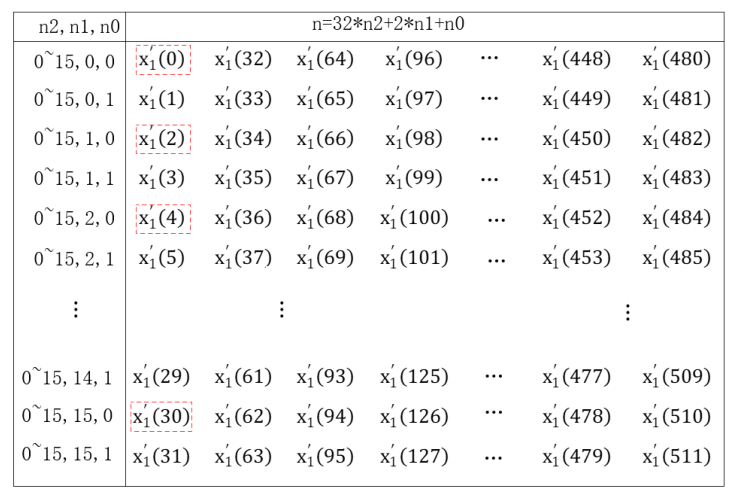
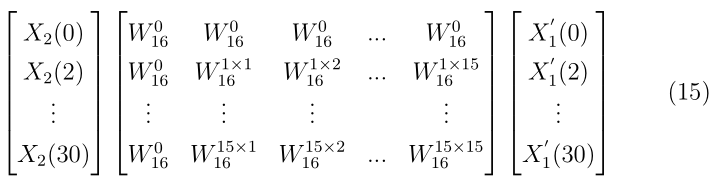
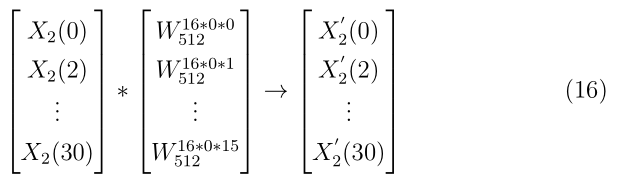


图5：第二级FFT16输入次序





重复以上过程，即可完成第二级FFT16的计算过程。

3.2.2. FFT存内计算方法

存内计算模式对于矩阵乘加计算比较友好，适合FFT这样的矩阵乘法运算。仍然以第一级FFT16为例，其第一行原始信号数据输入后与旋转因子矩阵第一行元素对应相乘相加，得到第一个FFT16模块计算结果，其过程在FLASH块中的计算过程为下图所示。

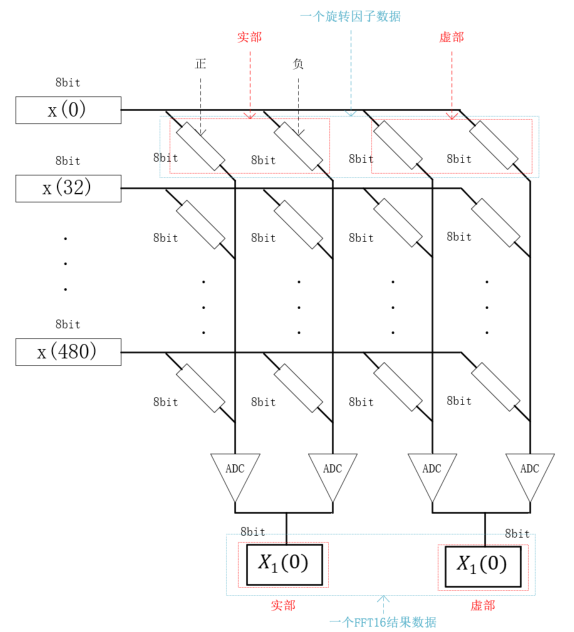


图6：一个FFT16数据在CIM中的计算

所以对于一个完整的FFT16模块来说，其在存内计算块中的计算过程可以如下如所示。

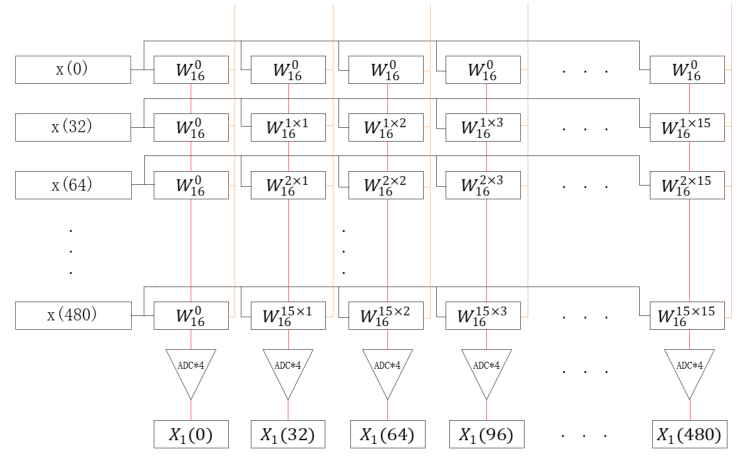


图7：一个FFT16模块在CIM中的计算

当有多个FFT16模块在FLASH Core中同时存在时，其结构图如下图所示。存储\_1中存储的就是需要进行FFT计算的原始数据和在分级计算中产生的中间数据。从存储\_1模块中每次并行输入16个数据进入到一个FFT16模块块中进行计算，在FLASH Core中完成计算之后并行输入16个模拟结果经过ADC转换成为数字结果。由于矩阵乘法和向量对应相乘的旋转因子可以进行融合，所以产生的数字结果不需要单独的乘法器矩阵再次进行向量对应相乘。地址生成模块控制着进入FLASH Core中的数据的地址。通过地址的生成和数据的并行输入输出，在重复32次16点FFT计算后，可以完成第一级FFT16的计算。第二级FFT16的计算过程和第一级类似，所不同的只是输入数据的次序不同而已。

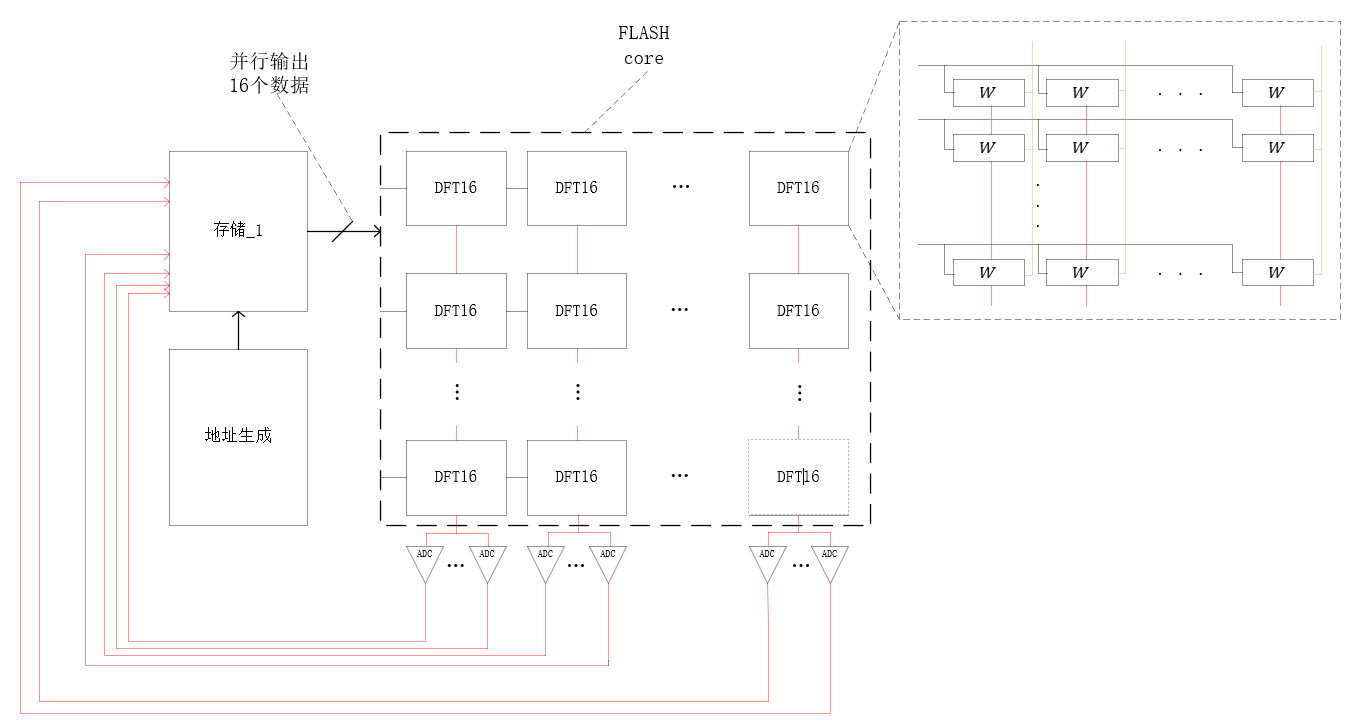


图8：多个DFT16模块在CIM中计算的示意图

由此，我们也可以得到第一级FFT16的计算时序图，当重复完成上述时序之后，可以得到第二级FFT16的时序，从而得到在存内计算块中FFT计算的完成时序。

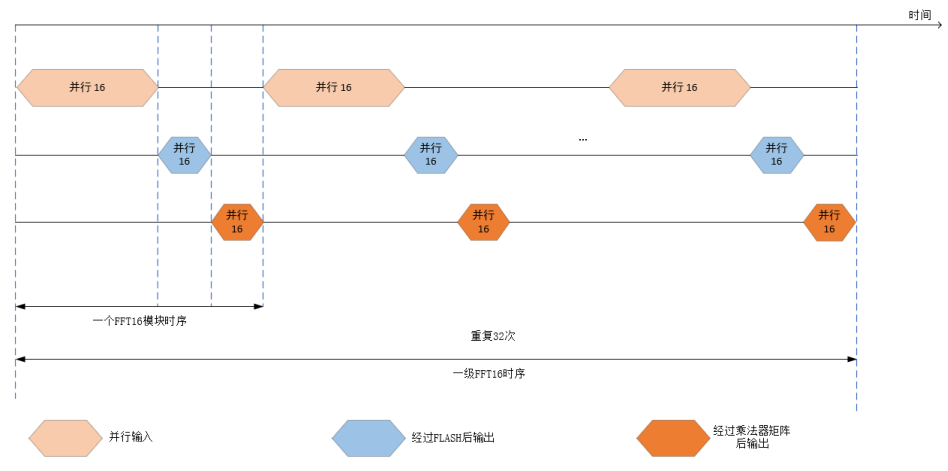


图9：一级FFT16计算时序示意图

3.2.3.性能参数估计

针对于本设计，可以通过数据计算量和存内计算核的性能参数进行整体设计的性能估计。根据文献[36]可得在一个256\*256的存内计算核中，大概的功耗是3.61mW，则根据以上设计，需要6个256\*256的存内计算核，则理论功耗为21.66mW。同样，根据存内计算核大小和计算流程设计，可以得到需要的存内计算块大小约为24KB。综上，可以与其他存内计算设计进行比较，比较的参数结果可见下表。

表3：不同设计性能估计比较

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ISSCC 2017  [17] | VLSI 2018  [18] | VLSI 2019  [19] | This  Work |
| Memristor | 144KB SRAM | 27KB SRAM | 10KB SRAM+64Kb CIM-SRAM | 24KB CIM |
| Power | 131.45mW | 14.29mW | 11.47mW | 21.66mW |
| Bits | - | - | 3bits | 8bits |

3.3. 研究计划

根据设计规划与时间要求，列出以下研究计划表格，以期在规定毕业时间内完成所有相关设计和论文撰写。

表3：研究计划时间表

|  |  |
| --- | --- |
| 时间 | 完成内容 |
| 2020/09-2020/10 | 理论计算，方案可行性论证 |
| 2020/11-2020/12 | FFT16模块设计，存储模块设计，地址生成模块设计，分立模块组合，顶层模块设计 |
| 2020/12 | 大论文写作  小论文写作和投稿 |
| 2021/01 | 大论文送审 |
| 2021/02 | 小论文修改，设计补充 |
| 2021/03 | 毕业答辩 |

参考文献

1. Jianqing Fan, Cong Ma, Yiqiao Zhong. “A Selective Overview of Deep Learning.”2019, arXiv:1904.05526v2.[Online].Available: https://arxiv.org/pdf/1904.05526.pdf
2. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pages 770–778.
3. David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. Nature, 550(7676):354, 2017.
4. Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144
5. Guoguo Chen, C.Parada, G.Heigold, “Small-FootPrint Keyword Spotting Using Deep Neural Network,”2014, IEEE International Conference on Acoustic, Speech and Signal Processing(ICASSP)
6. T.N.Sainath, C.Parada, “Convolutional Neural Networks for Small-footprint Keyword Spotting,”in Proceeding of Interspeech, 2015, pp.1478-1482
7. S.O.Arik, M.Kliegl, R.Child, et al, “Convolutional recurrent neural networks for small-footprint keyword spotting.”2017,

arXiv:1703.05390.Available: <https://arxiv.org/abs/1703.05390>

1. Yundong Zhang,N.Suda,Liangzhen Lai,et al.“Hello Edge:Keyword Spotting on Microcontrollers.”2018,arXiv:1711.07128.Available: <https://arxiv.org/abs/1711.07128>
2. R.Tang,J.Lin,“Deep Residual Learning for Small-footprint Keyword Spotting”[J].IEEE International Conference on Acoustic,Speech and Signal Processing(ICASSP),2018.
3. Shouyi Yin,Peng Ouyang,Shixuan Zheng,et al.“A 141 uW, 2.46 pJ/Neuron Binarized Convolutional Neural Network based Self-learning Speech Recognition Processor in 28nm CMOS”[J].IEEE Symposium on VLSI Circuit(VLSI),2018.
4. J.S.P Giraldo,S.Lauwereins,K.Badami,et al.“18μW SoC for near-microphone Keyword Spotting and Speaker Verification”[J]. IEEE Symposium on VLSI Circuit(VLSI),2019.
5. Weiwei Shan,Minhao Yang,Jiaming Xu,et al.“A 510nW 0.41V Low-Memory Low-Computation Keyword-Spotting Chip Using Serial FFT-Based MFCC and Binarized Depthwise Separable Convolutional Neural Network in 28nm CMOS”[J].IEEE International Solid-State Circuit Conference(ISSCC),2020.
6. G.W.Burr,R.M.Shelby,A.Sebastian,et al.“Neuromorphic computing using non-volatile memory”[J].Advances in Physics,2017,Volume 2,pp:89-124
7. Xin Si,Cheng Xin,Jianwei Su,et al.“Circuit Design Challenges in Computing-in-Memory for AI Edge Devices”[J].IEEE International Conference on ASIC(ASICON),2019.
8. Weihao Chen,Winsan Khwa,Junyi Li,et al.“Circuit design for beyond von Neumann applications using emerging memory:From nonvolatile logics to neuromorphic computing”[J].International Symposium on Quality Electronic Design(ISQED),2017.
9. M.Price,J.Glass,A.P.Chandarkasan,et al.“A Scalable Speech Recognizer with Deep-Neural-Network Acoustic Models and Voice

Activated Power Gating”[J].IEEE International Solid-State Circuit Conference(ISSCC),2017.

1. Suyang Bang,Jingcheng Wang,Ziyun Li,et al.“A 288μW Programmable Deep-Learning Processor with 270KB On-Chip Weight Storage Using

Non-Uniform Memory Hierarchy for Mobile Intelligence”[J]. IEEE International Solid-State Circuit Conference(ISSCC),2017.

1. Shihui Yin,Zhewei Jiang,Jae-Sun Seo,et al.“XNOR-SRAM: In-Memory Computing SRAM Macro for Binary/Ternary Deep Neural Networks”[J]. IEEE Journal of Solid-State Circuits,2020,VOL.55,NO.6.
2. Ruiqi Guo,Yonggang Liu,Shixuan Zheng,et al.“A 5.1pJ/Neuron 127.3us/Inference RNN-based Speech Recognition Processor using 16 Computing-in-Memory SRAM Macros in 65nm CMOS”[J]. IEEE Symposium on VLSI Circuit(VLSI),2019.
3. Shifan Gao,Guangjun Yang,Xiang Qiu,et al.“Programmable Linear RAM: A New Flash Memory-based Memristor for Artificial Synapses and Its Application to Speech Recognition System”[C].IEEE International Electron Devices Meeting(IEDM),2019.
4. R.Alvarez,H.J.Park.“End-to-End Streaming Keyword Spotting.”[C].IEEE International Conference on Acoustics,Speech and Signal Processing(ICASSP),2019.
5. A.Lavin,S.Gray.“Fast Algorithms for Convolutional Neural Networks”[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2016,pp.4013-4021.
6. N. Nguyen-Thanh, H. Le-Duc, D.-T. Ta, V.-T. Nguyen, “Energy efficient techniques using FFT for deep convolutional neural networks,”[C] International Conference on Advanced Technologies for Communications (ATC),2016,pp.231–236.
7. J. Lin,Y. Yao, “A Fast Algorithm for Convolutional Neural Networks Using Tile-based Fast Fourier Transforms,”[J] Neural Process. Lett.,vol.50(2),2019,pp.1951–1967.
8. S.K.Gonugondla,Mingu Kuang,N.Shanbhag.“A 42pJ/Decision 3.12TOPS/W Robust In-Memory Machine Learning Classifier with On-Chip Training”[J]. IEEE International Solid-State Circuit Conference(ISSCC),2018.
9. W.S.Khwa,Jia-Jing Chen,Jia-Fang Li,et al.“A 65nm 4Kb Algorithm-Dependent Computing-in-Memory SRAM Unit-Macro with 2.3ns and 55.8TOPS/W Fully Parallel Product-Sum Operation for Binary DNN Edge Processors”[J]. IEEE International Solid-State Circuit Conference(ISSCC),2018.
10. A.Biswas,A.P.Chandrakasan.“Conv-RAM: An Energy-Efficient SRAM with Embedded Convolution Computation for Low-Power CNN-Based

Machine Learning Applications”[J]. IEEE International Solid-State Circuit Conference(ISSCC),2018.

1. Jun Yang,Yuyao Kong,Zhen Wang,et al.“Sandwich-RAM: An Energy-Efficient In-Memory BWN Architecture with Pulse-Width Modulation”[J]. IEEE International Solid-State Circuit Conference(ISSCC),2019.
2. Xin Si,Jiajing Chen,Yungning Tu,et al.“A Twin-8T SRAM Computation-In-Memory Macro for Multiple-Bit CNN-Based Machine Learning”[J].IEEE International Solid-State Circuit Conference(ISSCC),2019.
3. Xin Si,Yungning Tu,Wei-Hsing Huang,et al.“A 28nm 64Kb 6T SRAM Computing-in-Memory Macro with 8b MAC Operation for AI Edge Chips”[J]. IEEE International Solid-State Circuit Conference(ISSCC),2020.
4. 程佩青.“数字信号处理教程”[M].北京：清华大学出版社，2007年：pp162.
5. Jian Wang,Xiaobin Li,Guangteng Fan,et al.“A Parallel Radix- 2 k FFT Processor Using Single-Port Merged-Bank Memory”[C].IEEE International Symposium on Circuit and Systems(ISCAS),2019.
6. P.Gonzalez-Guerrero,Xinfei Guo,M.R.Stan.“ASC-FFT: Area-Efficient Low-Latency FFT Design Based on Asynchronous Stochastic Computing”[C].IEEE Latin American Symposium on Circuit Systems(LASCAS),2019.
7. J.Vondrejc,D.Liu,M.Ladecky,et al.“FFT-based homogenization accelerated by low-rank tensor approximations.” 2019, arXiv: 1902.07488. Available: <https://arxiv.org/abs/1902.07455>
8. Cheng Qian,Zhuotong Chen,A.C.Yucel.“Tensor Decompositions for Reducing the Memory Requirement of Translation Operator Tensors in

FMM-FFT Accelerated IE Solvers”[C].International Applied Computational Electromagnetics(ACES),2019.

1. Yewei Zhang,Kejie Huang,Rui Xiao,et al.“An 8-bit In Resistive Memory Computing Core with Regulated Passive Neuron and Bit Line Weight Mapping.”2020,arXiv:2008.11669.Available: <https://arxiv.org/abs/2008.11669>