

华北电力大学

专业硕士学位论文

基于 FPGA 的模块化多电平换流器均压控制 策略研究

Research on Voltage Balancing Control Strategy of Modular Multilevel Converter Based on FPGA

王鑫艳

2024 年 5 月

国内图书分类号: TM743
国际图书分类号: 621.3

学校代码: 10079
密 级: 公开

专业硕士学位论文

基于 FPGA 的模块化多电平换流器均压控制 策略研究

硕 士 研 究 生: 王鑫艳

导 师: 刘崇茹教授

企 业 导 师: 李振动高级工程师

申 请 学 位: 能源动力硕士

专 业 领 域: 电气工程

学 习 方 式: 全日制

所 在 学 院: 电气与电子工程学院

答 辩 日 期: 2024 年 5 月

授予学位单位: 华北电力大学

Classified Index: TM743

U.D.C: 621.3

Thesis for the Professional Master's Degree

**Research on Voltage Balancing Control Strategy of
Modular Multilevel Converter Based on FPGA**

Candidate:	Wang Xinyan
Supervisor:	Prof. Liu Chongru
Enterprise mentor:	Li Zhendong
Professional Degree Applied for:	Master of Energy and Power Engineering
Speciality:	Electrical Engineering
Cultivation ways:	Full-time
School:	School of Electrical and Electronic Engineering
Date of Defence:	May, 2024
Degree-Conferring-Institution:	North China Electric Power University

摘 要

为了有效推动国家“双碳”战略的实施，大规模、远距离的能源传输成为关键。柔性直流输电技术作为高效、可靠的电力传输方式，在可再生能源并网和跨区域电网互联互通中发挥着重要作用。模块化多电平变流器（Modular Multilevel Converter, MMC）以其独特的优势成为柔性直流输电技术的核心。然而，随着 MMC 系统规模的日益扩大，单桥臂的子模块数量可达数百个，子模块电容电压不平衡的问题愈发显著。传统的均压策略受限于其高计算复杂度，难以快速完成大量子模块电压的排序，导致均压效果欠佳。这种不理想的均压效果可能引发三相桥臂输出电压不一致，进而诱发波动环流、加剧器件电流应力、增加系统功率损耗，最终对整个 MMC 系统的安全稳定运行构成威胁。针对上述问题，本文围绕模块化多电平换流器的均压控制策略进行研究，结合现场可编程门阵列（Field Programmable Gate Array, FPGA）的高度并行处理优势，提出了一种仅通过寻找极值而无需对所有子模块进行完整排序的高效均压控制策略。该策略通过优化排序算法，充分利用 FPGA 的并行处理能力，旨在实现 MMC 子模块电容电压的快速、准确均衡控制。论文的主要工作如下：

（1）本文深入探讨了分组并行极值排序算法的原理和实现机制，并通过数值计算方法探究了最优分组数与数据规模之间的关系。借助曲线拟合技术和对数模型的构建，本文得出一个具有广泛适用性的最优分组数确定公式，为均压控制策略的实际应用提供了理论依据。

（2）本文提出了一种基于并行分组极值排序算法的电容电压均衡控制策略。该策略在并行分组极值排序算法的基础上，通过设定电容电压上下限实现精准分组，避免了繁琐的子模块电压遍历过程。同时，引入子模块投切状态机制，在不牺牲均压效果的前提下有效降低开关频率，进而减少了换流器的功率损耗。基于 MATLAB/Simulink 平台进行了仿真验证，实验结果表明，该策略在稳态及暂态条件下均能实现快速且高效的电容电压平衡控制。

（3）本文基于 FPGA 实现了均压控制策略的硬件化。通过设计电压缓存模块、极值搜索模块以及触发信号生成模块，实现了对子模块电容电压数据的高效处理与均衡控制。测试结果表明，该策略在 FPGA 硬件环境下具有可行性，相较于现有研究，有效提升了计算速度，减少了资源消耗，具有实际应用价值。

关键词：模块化多电平换流器；电容电压平衡；现场可编程门阵列；排序算法

Abstract

In order to effectively promote the implementation of the national "double carbon" strategy, large-scale and long-distance energy transmission is the key. As an efficient and reliable power transmission method, flexible DC transmission technology plays an important role in renewable energy grid connection and cross-regional power grid interconnection. Modular multilevel converter (MMC) has become the core of flexible DC transmission technology with its unique advantages. However, with the increasing scale of MMC system, the number of sub-modules in a single bridge arm can reach hundreds, and the problem of sub-module capacitor voltage imbalance is becoming more and more significant. The traditional voltage balancing strategy is limited by its high computational complexity, and it is difficult to quickly complete the sorting of a large number of sub-module voltages, resulting in poor voltage balancing effect. This unsatisfactory voltage balancing effect may lead to inconsistent output voltage of the three-phase bridge arm, which in turn induces fluctuating circulation, aggravates device current stress, increases system power loss, and ultimately poses a threat to the safe and stable operation of the entire MMC system. In view of the above problems, this paper focuses on the voltage balancing control strategy of modular multilevel converter. Combined with the advantages of highly parallel processing of field programmable gate array (FPGA), an efficient voltage balancing control strategy is proposed, which only finds the extreme value without sorting all sub-modules. By optimizing the sorting algorithm and making full use of the parallel processing ability of FPGA, this strategy aims to realize the fast and accurate equalization control of MMC sub-module capacitor voltage. The main work of this paper is as follows:

(1) In this thesis, the principle and implementation mechanism of the grouping parallel extremum sorting algorithm are discussed in depth, and the relationship between the optimal number of groups and the data size is explored by numerical calculation. With the help of curve fitting technology and the construction of logarithmic model, this paper obtains a formula for determining the optimal number of groups with wide applicability, which provides a theoretical basis for the practical application of voltage balancing control strategy.

(2) In this thesis, a capacitor voltage balancing control strategy based on parallel

grouping extremum sorting algorithm is proposed. Based on the parallel grouping extremum sorting algorithm, this strategy achieves accurate grouping by setting the capacitor voltage threshold, which avoids the cumbersome sub-module voltage traversal process. At the same time, the sub-module switching state mechanism is introduced to effectively reduce the switching frequency without sacrificing the voltage balancing effect, thereby reducing the power loss of the converter. The simulation is carried out based on MATLAB/Simulink platform. The experimental results show that the strategy can achieve fast and efficient capacitor voltage balance control under steady-state and transient conditions.

(3) In this thesis, the hardware implementation of the voltage balancing control strategy based on FPGA is achieved. By designing the voltage buffer module, the extreme value search module and the trigger signal generation module, the efficient processing and equalization control of the sub-module capacitor voltage data are realized. The test results show that the strategy is feasible in the FPGA hardware environment. Compared with the existing research, it effectively improves the calculation speed, reduces the resource consumption, and has practical application value.

Keywords: modular multilevel converter (MMC), capacitor voltage balancing, field programmable gate array (FPGA), sort algorithm

目 录

第 1 章 绪论.....	1
1.1 课题研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 子模块电容电压均衡方法.....	2
1.2.2 FPGA 在实时仿真领域的应用	5
1.3 课题研究内容及章节安排	6
第 2 章 MMC 均压算法与 FPGA 工作原理	7
2.1 引言	7
2.2 基于排序算法的子模块电容电压均衡控制	7
2.2.1 子模块工作原理与电压波形生成机制	8
2.2.2 基于最近电平逼近调制的均压控制方法	10
2.3 现场可编程门阵列及其开发流程	12
2.3.1 FPGA 结构与性能分析	12
2.3.2 基于 Vivado 的开发流程	14
2.4 小结	16
第 3 章 适用于 FPGA 的新型排序算法	17
3.1 引言	17
3.2 基于分组并行的极值排序算法	17
3.2.1 传统排序算法对比分析.....	17
3.2.2 分组并行极值排序算法.....	21
3.2.3 最优分组数的参数设计.....	23
3.2.4 性能评估.....	26
3.3 考虑开关频率优化的新型排序算法	29
3.3.1 考虑实际运行需求的分组策略优化.....	29
3.3.2 考虑投切状态的开关频率优化.....	30
3.3.3 基于新型排序算法的均压控制策略.....	31
3.4 仿真验证.....	34
3.4.1 稳态实验.....	35
3.4.2 暂态实验.....	37
3.5 小结	38
第 4 章 均压算法在 FPGA 上的实现与验证	40
4.1 引言	40
4.2 约束设计与数据传输优化	40
4.2.1 设计选择.....	40

4.2.2 约束设计.....42

4.2.3 数据转存机制.....43

4.2.4 启动延时机制.....45

4.3 均压策略的 FPGA 实现46

4.3.1 电压缓存模块.....46

4.3.2 极值搜索模块.....50

4.3.3 触发信号生成模块.....53

4.4 实时仿真实验.....54

4.5 小结.....56

第 5 章 结论与展望57

5.1 总结.....57

5.2 展望.....58

参考文献.....59

第 1 章 绪论

1.1 课题研究背景及意义

为了有效推动国家“双碳”战略的实施，国家能源局近日发布了《2024 年能源工作指导意见》，明确了进一步优化国家能源结构，要求到 2024 年年底将我国的非化石能源发电装机占比提高到 55%左右，并确保风电、太阳能发电量占全国发电总量的比重超过 17%^[1]。然而，我国能源生产和消费中心的地理分布极不均衡，西部地区拥有丰富的风能和太阳能资源，而能源消费中心主要集中在东部地区。因此，实现大规模、远距离的能源传输成为国家能源战略的关键组成部分。

《新型电力系统发展蓝皮书》强调，为适应未来以高比例新能源并网为核心，特高压柔性直流技术的研发成为重点^[2]。柔性直流输电技术是一种高效、可靠的电力传输方式，特别适用于大规模可再生能源的远距离输送、跨区域电网的互联互通以及缓解清洁能源大规模并网消纳^[3]。模块化多电平变流器（Modular Multilevel Converter, MMC）以其高模块化设计、低开关频率和低谐波含量等优势^[4]，已成为柔性直流输电技术的核心。在我国，绝大多数已投运或在建的柔性直流输电工程均采用了 MMC 拓扑结构，如表 1-1 所示。

表 1-1 中国柔性直流输电工程

工程名称	投产年份	直流电流/kV	输电容量/MW
厦门柔性直流工程	2015	±320	1000
鲁西背靠背直流工程	2016	±350	1000
渝鄂异步联网工程	2019	±420	4×1250
张北直流电网工程	2020	±500	2×3000, 2×1500
昆柳龙直流工程	2020	±800	5000
如东海上风电工程	2021	±400	1100
白鹤滩输电工程	2022	±800	8000
广东电网东西异步联网工程	2022	±300	4×1500
阳江三山岛海上风电送出工程	—	±500	2000

MMC 的稳定运行依赖于散布在各子模块中的电容所提供的直流侧电压支撑，使得子模块电容电压的均衡控制成为其稳定运行的关键^[5]。随着 MMC 系统电压等级的提升，单个桥臂的子模块数量可达数百个，导致均压控制策略的

实现难度加大，子模块电容电压不平衡的问题愈发显著^[6]。为了响应电网负载和可再生能源输出的快速变化，均压控制策略需要具备高度的实时性和精确性。这意味着控制算法不仅需要快速执行，而且必须保证计算结果的准确，以防止电压不平衡导致的系统不稳定或设备损坏。

现场可编程门阵列（Field Programmable Gate Array, FPGA）以其高度的并行处理能力和深度流水线架构，为解决以上问题提供了新的技术路径^[7]。FPGA 能够同时处理多个运算任务，这使得它特别适合于执行均压控制算法中涉及的大规模并行计算^[8]。同时，FPGA 的深度流水线架构支持低延迟的数据处理，确保均压控制命令能够快速执行，满足系统对实时性的严格要求。然而，实现基于 FPGA 的均压算法面临资源限制、算法适配及调试的挑战，其有限的逻辑块、存储资源和 I/O 端口可能对复杂算法的应用构成约束，尤其在处理涉及数百子模块电容电压均衡时。此外，算法通过 FPGA 硬件描述语言描述带来的逻辑和性能优化问题也不容忽视。尽管基于 FPGA 的均压算法实现面临一系列技术挑战，但 FPGA 仍是实现高性能均压控制策略的理想平台。

因此，本文主要研究基于 FPGA 的模块化多电平换流器电容电压均衡策略，旨在通过最小的计算量快速准确地实现控制计算，对于提高 MMC 系统的响应速度和稳定性具有重要的实际意义。

1.2 国内外研究现状

1.2.1 子模块电容电压均衡方法

在高压直流传输系统中，模块化多电平转换器的子模块电容电压均衡控制是保证系统稳定运行的关键技术。目前的研究主要集中在通过改进拓扑结构和附加控制策略两方面来优化均压控制^[9]。

改进拓扑结构是指通过共享中间单元、引入二极管钳位机制或外加自供电等方式来实现均压功能的方法。文献[10]提出了一种改进的 MMC 拓扑，通过每相级联子模块共享中间单元来减少桥臂输出电平数，简化了均压控制，但仍需要辅助控制策略以实现全面的电容电压均衡。文献[11]提出了基于二极管钳位的拓扑结构，通过相邻子模块电容间串联二极管形成功率通道，并引入功率反馈控制策略，构建了可控闭环系统来有效钳制电容电压。尽管这一新拓扑减少了电压传感器数量，加快了系统响应速度，但相内电压平衡的实现依赖于辅助变压器。文献[12]针对混合型 HVDC 系统，通过混合半桥和全桥子模块的新型 MMC 拓扑实现了自均压功能，优化了成本和减少了运行损耗，但缺乏交流升

压功能。文献[13]基于自供电控制的外加均压拓扑,通过调整自供电源输入特性和控制信号的频率及占空比,在实现均压控制的基础上,减轻了计算负担,避免了对 IGBT 和其他复杂电路的依赖并降低了子模块有功功率损耗。文献[14]提出了一种自均压 MMC 拓扑结构,该拓扑通过子模块间的二极管和辅助电路元件实现电能的自由传递,实现自我均衡而无需外加控制策略。通过减半参与均压控制的电容数量,降低了系统的设计复杂度。这类方法在简化控制逻辑和提升系统响应速度方面取得了显著进展,但伴随着额外电路的增加,引入了更高的硬件复杂性和成本,因此在工程实践中往往不被优先考虑。

附加控制策略需要结合调制策略,通过控制 IGBT 的触发信号,确保参与导通的子模块电容电压共同维持 MMC 输出电压的稳定。根据调制策略的不同,主要可以分为结合载波层叠加 PWM 调制 (Carrier Phase Shifted SPWM, CPS-SPWM) 的均压策略和结合最近电平逼近调制 (Nearest Level Modulation, NLM) 的均压策略^[15]。

结合 CPS-SPWM 的均压策略根据设定的子模块开关频率与附加控制量生成调制波来控制子模块的导通时间,进而对电压进行动态调节,而无需对所有的电容电压进行排序^[16]。在此基础上,文献[17]根据桥臂能量的自然平衡状态,引入一个与桥臂电流相位相同的附加参考电压,实现调整桥臂内部能量均匀分布,进而实现电容电压均衡。但该方法基于桥臂电压均衡的假设条件,需额外的控制环节,增加了算法实现的复杂度。针对低频导致的畸变问题,文献[18]通过注入高频共模电压与高频环流电流,维持了电压平衡,实现 MMC 变频驱动,减少了谐波含量。文献[19]针对子模块的开关频率进行了参数设计,明确开关频率不应为电网基频的整数倍。在此基础上,文献[20]提出电压与电流谐波间的相互作用影响各子模块间的功率分配不均,而非整数调频比的选择有助于在同相桥臂的单元间提供更均匀的功率分配。针对混合型 MMC 均压问题,基于 CPS-SPWM 进行二次谐波注入在实现均压的同时可以有效减小输出电压畸变率^[21-23]。CPS-SPWM 策略有效降低了开关损耗并保证了 SM 损耗的一致性。然而,该策略不支持自动选择备用子模块。特别是在发生故障时,若备用子模块已经放电,系统无法快速灵活地执行故障子模块的替换,从而影响冗余模块备用的实施可行性^[24]。针对该问题,文献[25]提出了基于 CPS-SPWM 的新型冗余保护策略,根据子模块的运行状态进行载波动态分配,并引入电容电压优化控制来抑制电容电压波动。

结合 NLM 的均压策略通过子模块电容电压的排序情况来确定子模块的开关状态,进而通过控制子模块投入数量来实现电压均衡。这种方法简单可靠,

特别适合电平数高的场合，已应用在实际的柔性直流工程中^[26]。但随着子模块数量的增加，对大量子模块电容电压的排序环节限制了算法效率^[27]，而且子模块的频繁投切会导致高额的开关损耗，进而影响 IGBT 的使用寿命^[28, 29]。大量研究人员从不同角度对基于 NLM 的均压策略进行了优化^[30]。

针对降低算法复杂度方面，文献[31]采用了相比冒泡排序法时间复杂度更低的快速排序算法，而文献[32]采用了时间复杂度极低但空间复杂度极高的并行全比较排序算法。文献[33]将子模块随机分组，通过计算每组的能量平衡因子以及组内并行全比较排序得到子模块投切顺序。该方法虽然用于极低的时间复杂度，但由于随机分配可能导致部分子模块电压越限触发过(欠)压保护。除了结合最近电平调制和排序算法实现均压控制之外，还可以通过非排序方法实现。文献[34]提出了一种基于预测的均衡策略，按能量分配原则进行子模块触发实现电容均衡，但不适用于故障的情况。文献[35]提出了一种基于电压估计的控制策略，通过自适应线性神经元算法，减少了对电压测量与通信的依赖，但同样不适用于故障情况。文献[36]通过子模块与基准值的差值是否越界来判断是否需要重新生成触发脉冲，并依据偏差值的正负顺序投入子模块。文献[37]借鉴这种思想，提出了一种仅需一次排序过程就能实现均压的策略，该策略基于初步排序将子模块划分为两组，并通过比较各组子模块的电压与电容电压的平均值来决定其是否导通。然而，面对系统启动或功率突变引起的电压剧烈变化，该策略可能无法维持有效控制。

除了表征均衡效果的电压总体波动和电压的不平衡度外，器件的开关频率也是影响均压算法优劣的重要因素^[38]。降低 IGBT 开关频率的方法主要有引入保持因子法^[39]和预设最大电压偏差量^[40]。文献[41]为保持能量均衡，引入了互为倒数的双保持因子，分别用于与不同状态的子模块电压相乘，使得开关状态尽量维持不变。文献[42]在文献[44]的基础上，引入了可控频率的比例积分控制器，来保证系统的稳定运行。文献[43]在引入保持因子的基础上，采用了快速排序算法，实现了降低开关频率且简化计算的双重目的。但该方法在处理大规模子模块时运算量较大，需要通过反复测试来确定合适的保持因子。文献[44]在引入最大电压偏差量的基础上，对投切的子模块进行一定比例的轮换，虽然增加了一定的开关频率，但获得了更好的均压效果。实现该类方法的关键在于选择合适的偏差量整定值，过大或过小的都不适宜^[45]。此外，文献[46]提出了一种分层均压控制法，其核心思想是将子模块电压依据其大小进行分层投切，从而达到降低频率的目的。而文献[47]则在文献[46]的基础上进行了拓展，通过设置明确的投入优先级，将子模块细致地划分为优先投入、优先切除以及正常触发

三组，通过这种精细化的投切策略，进一步提升了降频效果。

除以上的均压策略外，还可以基于投切状态矩阵实现电容电压的自平衡^[48]，或者结合特定谐波消除的 PWM 调制策略与能量平衡思想^[49]。

综上所述，尽管目前的均压控制策略在实现电容电压均衡方面取得了一定的进展，但仍面临许多挑战，需要进一步的研究以优化这些控制策略，减少开关损耗，提高系统的整体性能和可靠性。

1.2.2 FPGA 在实时仿真领域的应用

FPGA 具有高度的并行处理能力和低延迟特性，在要求快速响应与高精度计算的场景中备受青睐，特别是在电力电子实时仿真领域^[50]。文献[51]详细论证了 FPGA 应用在实时仿真领域的可行性，而文献[52-54]则分别从不同系统、算法和平台等角度，论证了 FPGA 在实时仿真中的优势和应用潜力。基于此，研究者们积极探索基于 FPGA 的实时仿真技术，以满足电力系统动态特性的高精度仿真需求。

针对有源配电网，文献[55]提出了基于 FPGA 的光伏发电系统实时仿真方法。文献[56]提出了一种多层级并行暂态实时仿真系统架构，通过 FPGA 的数据并行和流水线并行，实现了有源配电网求解的高效实时仿真。文献[57]基于 FPGA 开发了包含小型分布式风力发电、光伏发电以及蓄电池储能单元的新能源低压直流配电系统的暂态实时仿真器

针对开关器件，为解决二值电阻开关模型的迭代计算问题，文献[58]提出了一种适用于高频电力电子变换器的实时仿真建模方法，并给出了相应的 FPGA 硬件实现方案。文献[59]基于 DSP-FPGA 架构，提出了一种初始误差修正算法，消除了由开关模型引入导致的虚拟功率损耗。文献[60]针对高瞬态分辨率的需求，提出了一种基于 FPGA 的高效 IGBT 行为模型，能够以 5 纳秒的步长获得精确的 IGBT 切换瞬态波形。

针对 MMC-HVDC，文献[61]提出了一种利用 FPGA 并行架构的高效建模方法，仅在单个 FPGA 中就能实现 1500 多个子模块的实时仿真。针对四端真双极的大规模 MMC-HVDC 系统，文献[62]提出了一种基于 FPGA 的并行多速率仿真方案，并设计了相应的主系统求解器和控制器的计算架构。

针对其他电力电子设备，文献[63]结合混合伴随电路建模方法和紧凑型电磁暂态程序算法，基于 FPGA 平台以 250ns 的步长实现了固态变压器的准确实时仿真，其开关频率为 50kHz，达到了亚微秒级别的实时仿真。文献[64]针对高频谐振变换器的实时仿真提出了一种基于 FPGA 直接映射的方法，该方法无需

迭代即可提供准确的仿真结果。

为满足仿真规模和复杂度日益增长的新型电力系统仿真需求,研究人员针对提高 FPGA 性能展开了研究。文献[65]提出了一种网络撕裂技术,允许对开关的子集进行独立处理,减轻了嵌入式内存需求并降低了计算负担。文献[66]结合单延时解耦技术和网络撕裂技术,实现了高开关频率和较小仿真步长的实时仿真建模。文献[67]针对单块 FPGA 逻辑资源受限,无法高效仿真大规模电力系统的问题,设计了一种多 FPGA 实时仿真架构,能够有效增加仿真计算能力并缩短仿真步长。

综上所述, FPGA 在电力电子实时仿真领域的应用已经取得了显著的进展。从算法优化到系统架构设计,从单一应用到多平台集成, FPGA 以其独特的优势为实时仿真技术的发展提供了强大的支持。

1.3 课题研究内容及章节安排

随着柔性直流输电技术的发展,传统的均压策略由于其计算复杂度高,难以快速完成大量子模块电压的排序,子模块电压均衡愈发困难。针对以上问题,本文提出了一种基于并行分组极值排序的均压策略,并进行了频率优化,在此基础上结合 FPGA 高度并行化与流水线架构的特点,针对 FPGA 进行了设计与实现,并验证了其在实际环境中的性能和效果。各章节主要内容如下:

第二章探讨了模块化多电平换流器和现场可编程逻辑门阵列的关键技术。首先介绍了 MMC 的拓扑结构及子模块工作原理,阐述了输出电压波形生成机制,详细分析了最近电平逼近调制策略。随后,分析了 FPGA 的核心原理、结构组成及在电力电子系统中的应用优势。通过全面分析 MMC 和 FPGA 的工作机制,为实现高效、可靠的均压控制策略奠定了理论基础。

第三章旨在解决柔性直流输电系统中均压控制策略的实时性和计算速度问题。传统均压策略在大规模 MMC 系统中存在效率不高的问题。为此,本章提出了一种基于并行分组极值排序算法的新型均压控制策略,有效简化了控制逻辑,并针对实际运行需求及降低开关频率的目标进行优化,有效减少了功率损耗。通过理论分析和仿真实验验证了该策略在时间和空间的优越性。

第四章旨在利用 FPGA 的并行计算优势实现子模块电容电压均衡控制。首先,引入了 FPGA 的硬件实现与性能优化方法,包括约束设计、数据转存与启动延时机制、流水线设计和分时复用设计。然后,对均压策略进行了顶层架构设计和功能模块设计,详细阐述了均压策略在 FPGA 上的具体实现过程,并进行了性能测试和仿真验证,证实了其在硬件条件下的可行性。

第 2 章 MMC 均压算法与 FPGA 工作原理

2.1 引言

基于电压源型换流器的柔性直流输电响应速度快、可控性好、运行方式灵活，被广泛用于长距离高压直流输电工程、大规模可再生能源并网。模块化多电平换流器作为柔性直流输电的关键技术之一，因其模块化程度高、可扩展性强和优异的电压输出质量，正逐步成为主流选择。同时，现场可编程逻辑门阵列以其卓越的可编程性和并行处理能力，在电力电子系统的控制与优化中扮演着至关重要的角色。尽管这些技术在理论和应用上都取得了显著进展，但在提升仿真精度和实时处理性能方面，仍存在挑战和研究空间。

本章旨在深入探讨 MMC 和 FPGA 的核心工作原理及关键技术。在 MMC 部分，首先通过分析子模块工作原理，阐释了输出电压波形生成机制。紧接着，详细分析了最近电平逼近调制策略，展示了其不同电平数下对输出波形及谐波含量的调节能力，并简要介绍了传统均压控制策略。在 FPGA 部分，详述 FPGA 的核心原理、结构组成及其在电力电子系统中的应用优势。同时，基于 Vivado 平台介绍 FPGA 从方案规划到硬件测试的全部开发流程。通过对 MMC 和 FPGA 的系统性分析，本章为后续的研究提供了坚实的理论基础，为实现高效、可靠的电力系统控制策略奠定了基础。

2.2 基于排序算法的子模块电容电压均衡控制

模块化多电平换流器的桥臂并非由多个开关器件直接串联构成的，而是采用了子模块（Sub-Module, SM）级联的方式，基于半桥子模块的 MMC 拓扑结构如图 2-1 所示。从图中可以看到，一个三相模块化多电平换流器共有 6 个桥臂，其中每个桥臂由 N 个半桥子模块和一个串联电抗器 L 组成，其中半桥子模块含一个直流电容、两组绝缘栅双极型晶体管（Insulate Gate Bipolar Transistor, IGBT）及反并联续流二极管。

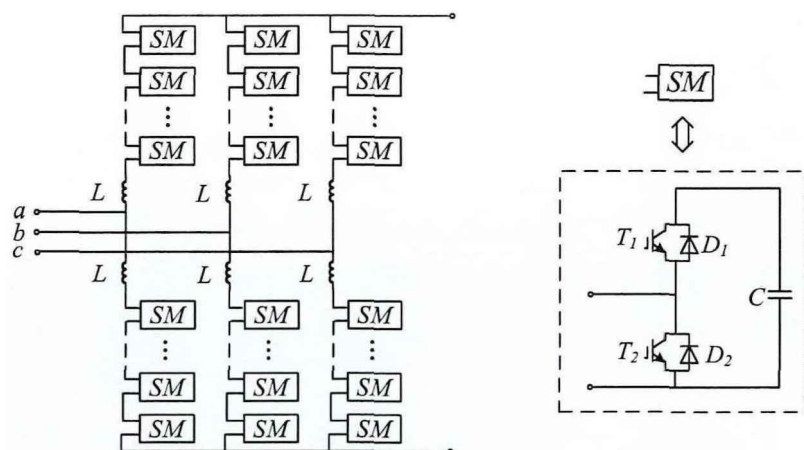


图 2-1 基于半桥子模块的 MMC 拓扑结构图

2.2.1 子模块工作原理与电压波形生成机制

IGBT 是一种高效、可靠的半导体开关器件，通过给 IGBT 施加不同的触发信号，可以实现子模块的状态切换。需要注意的是，为避免子模块电容发生直通故障，不可以对两个 IGBT 同时施加导通信号。因此，每个子模块有三种不同的开关状态：

（1）投入状态

为了方便描述桥臂电流的方向性，本文定义电流从 i 点流向 j 点为正方向。对上桥 T_1 施加导通信号，下桥 T_2 施加关断信号。当桥臂电流为正时，电流与上桥 T_1 反向，且由于 i 点电位高于 j 点电位， D_2 处于反向截止状态，因此电流流过电容，此时对电容进行充电；当桥臂电流为负时，由于上桥 T_1 使得 i 点电位高于 j 点电位，电流不能通过下桥 D_2 流出，而是通过上桥 T_1 流出，此时对电容进行放电。在上述两个状态下，若忽略电力电子器件压降，则子模块输出电压均为电容电压，该状态被称为投入状态，如图 2-2 所示。

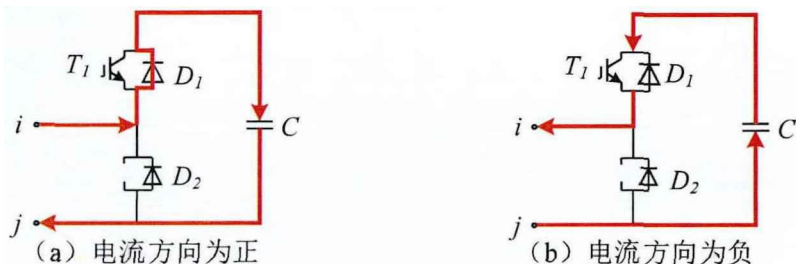


图 2-2 子模块投入状态

（2）切除状态

对上桥 T_1 施加关断信号，下桥 T_2 施加导通信号。当桥臂电流为正时，由于下桥 T_2 已经导通， i 、 j 两点近似等电位，电容电压使上桥二极管处于反向截

止状态，因此电流不能通过上桥 D_1 流出，而是通过下桥 T_2 流出，此时电流不经过电容。当桥臂电流为负时，上桥 D_1 处于反向截止状态，下桥 D_2 导通，由于电流方向与下桥 T_1 反向，因此电流通过下桥 D_2 流出，此时也电流不经过电容。在上述两个状态下，电流都不经过电容，则子模块输出电压为 0，该状态被称为切除状态，如图 2-3 所示。

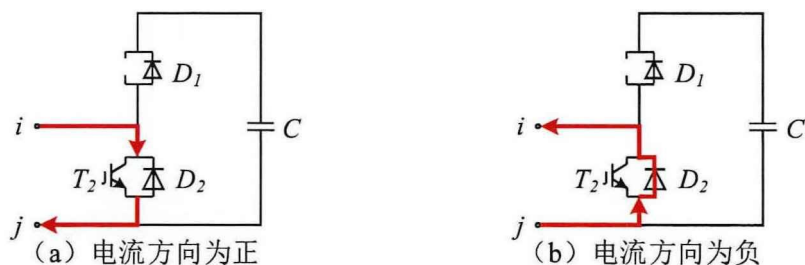


图 2-3 子模块切除状态

(3) 闭锁状态

对上桥 T_1 和下桥 T_2 都施加关断信号。当桥臂电流为正时，由于上、下桥 IGBT 都不导通，相当于 IGBT 支路都断开，而电流与上桥 D_1 同向，而与下桥 D_2 反向，此时电流流过电容，对电容进行充电，与投入状态类似。当桥臂电流为负时，电流与上桥 D_1 反向，而与下桥 D_2 同向，此时电流不流经电容，与切除状态类似。在上述两个状态下，上下两个 IGBT 都关断，该状态被称为闭锁状态，如图 2-4 所示。

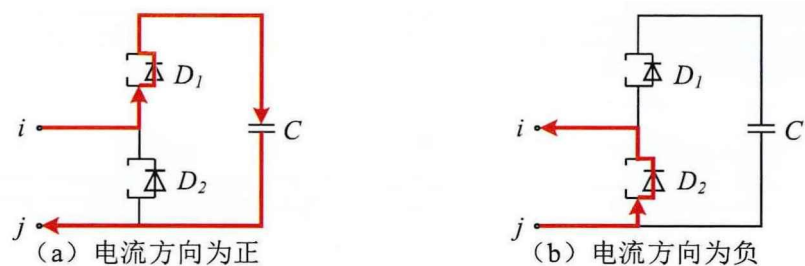


图 2-4 子模块闭锁状态

桥臂电流为正时，电流对投入状态的子模块充电，使得子模块电压升压，当电流反向，处于投入状态的子模块被放电，子模块电压降低。处于切除状态的子模块，由于电流并不经过电容，其电容电压不变。由于每个子模块相互独立，电容在充放电过程中存在的差异可能导致电容电压不平衡。

MMC 通过控制 IGBT 的开关状态，实现子模块的不同工作状态（投入、切除、闭锁）的切换，以及电容的充放电过程，进而调节输出电压的波形。假设子模块电容电压均衡，则每个子模块只存在两种电平，投入时子模块输出电压为全电压，切除时子模块电压为零。通过将所有已投入子模块的电平进行叠加，就能实现桥臂输出电压呈现阶梯波近似正弦。同时，为保证直流侧电压稳

定，上下桥臂叠加的电平应保持不变。以 A 相为例， p 、 d 分别表示上桥臂和下桥臂，MMC 多电平波形生成机制如图 2-5 所示。

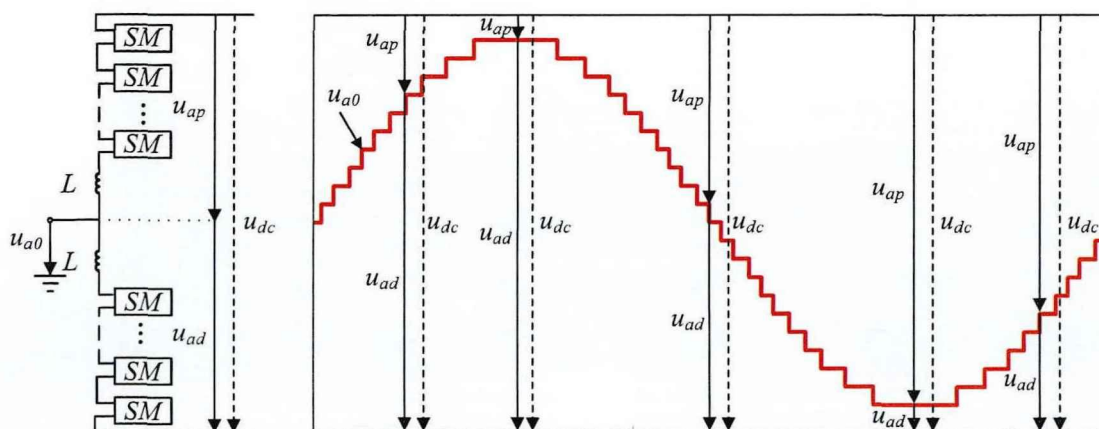


图 2-5 MMC 多电平波形生成机制

2.2.2 基于最近电平逼近调制的均压控制方法

为确定输出电压阶梯波的形式，需要对调制策略进行研究。最近电平逼近调制的原理与数模转换相似，在每个控制周期内，根据当前时刻的参考电压值，通过就近取整计算得到一个最接近参考电压的电平数^[68]，其原理图如图 2-6 所示，求取电压阶梯波的公式如式 (2-1) 所示。

$$u_{step} = U_c \cdot \text{round}\left[\frac{u_{ref}}{U_c}\right] \quad (2-1)$$

式中， U_c 为子模块额定电容电压， u_{ref} 为桥臂电压参考值， $\text{round}[x]$ 为就近取整函数，例如 $\text{round}[3.2]=3$ ， $\text{round}[3.6]=4$ 。

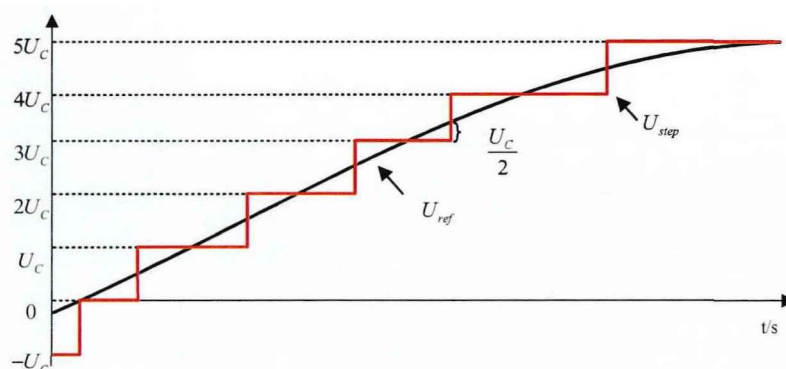


图 2-6 最近电平逼近调制原理图

电平分别为 5、21、201 的 NLM 输出波形如图 2-7 所示，可以看到输出电平数较低时，逼近误差较大，其谐波含量较高，电压质量较差。但随着电平数

的增加, 输出电压波形越来越逼近正弦调制波, 谐波含量显著降低。当电平数为 21 时, 总谐波失真率(Total Harmonic Distortion, THD)为 3.85%, 而当电平数达到 201 时, THD 仅为 0.36%。在实际工程应用中, 柔性直流输电工程的子模块数量往往较高, 一般在几百到几千之间。因此, NLM 调制方式不仅简单易于实现, 而且非常适用于大规模 MMC 系统。

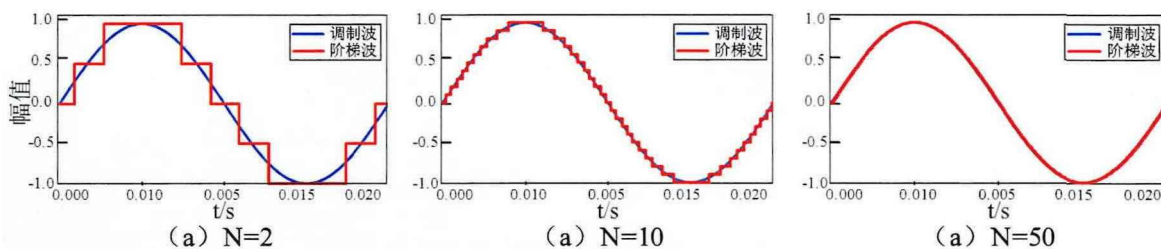


图 2-7 不同电平下的 NLM 输出波形

假设上下桥臂的总数为 N , 则 $n_p + n_d = N$, 即同相上、下桥臂投入的子模块总数恒为 N , 从而保证了直流侧电压恒定不变。根据式 (2-2), 即可计算出每个周期上、下桥臂所需投入的子模块个数。

$$\begin{cases} n_p = \frac{N}{2} - \text{round}\left[\frac{u_{ref}}{U_c}\right] \\ n_d = \frac{N}{2} + \text{round}\left[\frac{u_{ref}}{U_c}\right] \end{cases} \quad (2-2)$$

为确保系统运行效率并防止过调制区的潜在风险, 必须对换流器中上下桥臂的子模块投入个数进行严格控制, 维持在 0 至 N 的安全范围内。达到边界值的子模块数量可能会导致输出电压波形失真, 进而影响换流器的性能和运行特性。为了避免这种情况, 引入限幅环节, 以确保子模块投入数量符合要求。

此外, 最近电平调制方式实施的前提是子模块电容电压均衡。如果对子模块电容电压波动性不加以控制, MMC 将难以准确合成输出电压, 导致电压波形失真, 进而降低电能质量。因此, 该 NLM 常与电容电压均衡策略相结合, 其具体控制流程如图 2-8 所示。

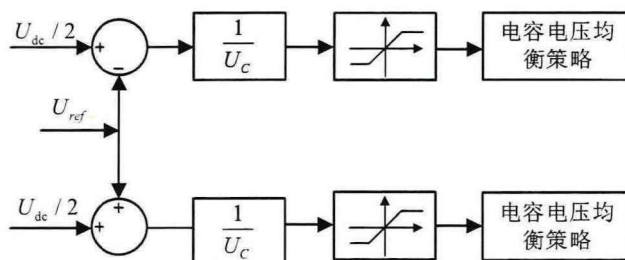


图 2-8 NLM 策略流程图

基于 NLM 的均压策略依据电容的储能特性, 对电压较高的子模块放电, 使其电压降低, 对电压较低子模块充电, 使其电压升高, 进而维持子模块电压在额定值附近波动。因此, 桥臂电流为正时, 投入电压最低的 n 个子模块, 切除电压最高的 $N-n$ 个子模块, 其中 n 为此时需投入的子模块个数; 同理, 桥臂电流为负时, 投入电压最高的 n 个子模块, 切除电压最低的 $N-n$ 个子模块。

以一个桥臂为例说明传统均压控制的具体步骤, 其流程如图 2-9 所示:

- 1) 经过最近电平逼近调制环节, 得到该时刻需要投入的子模块个数 n ;
- 2) 采用冒泡排序法对所有子模块电容电压进行由低到高的排序;
- 3) 根据桥臂电流方向和排序结果, 依据上述原则选择投切的子模块;
- 4) 依据子模块投切决策表生成触发脉冲。

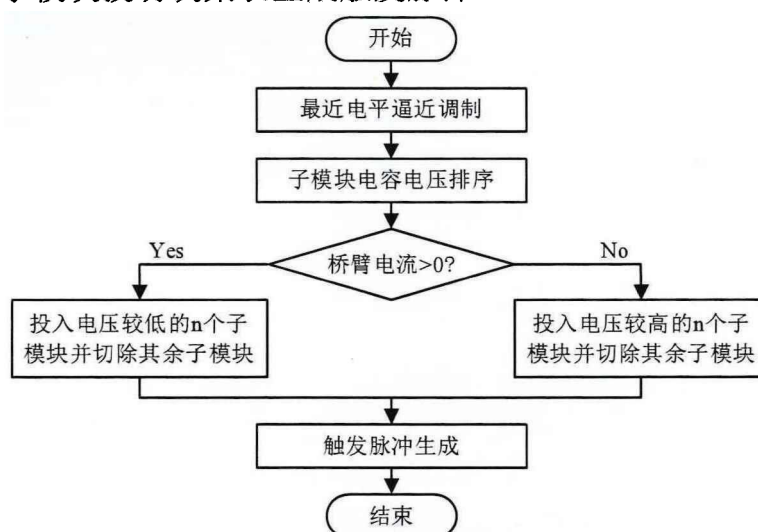


图 2-9 传统电容电压控制方法

在模块化多电平换流器的电容电压均衡控制策略中, 电容电压的排序过程占据了核心地位, 其计算消耗时间显著超过最近电平调制及触发信号生成环节所耗时间。随着 MMC 向更高电压和更大容量演进, 单桥臂内子模块数量增加导致排序计算时间急剧上升。因此, 具备高效并行处理能力的先进排序技术是提高阀级控制器控制精度和响应速度的关键。

2.3 现场可编程门阵列及其开发流程

2.3.1 FPGA 结构与性能分析

现场可编程门阵列是一种高度灵活的数字集成电路, 能够通过编程手段在现场配置以实现特定的逻辑功能。与其他可编程逻辑器件相比, FPGA 提供了更高的逻辑密度和复杂度, 允许实现更加复杂的数字系统和算法。与专用集成

电路相比，FPGA 不需要昂贵的制造过程和长时间的开发周期，就可以实现定制化的电路设计。用户可以利用硬件描述语言来描述他们想要实现的电路功能，然后通过编译这些描述并将其下载到 FPGA 芯片上，从而完成电路的配置。

顾名思义，现场可编程门阵列的核心特征包括其在现场的可编程能力以及其基于门阵列的结构。然而，FPGA 的内部构造并不仅限于简单的“门”逻辑，而是包括了更为复杂的可编程逻辑块和互连资源，如图 2-10 所示。

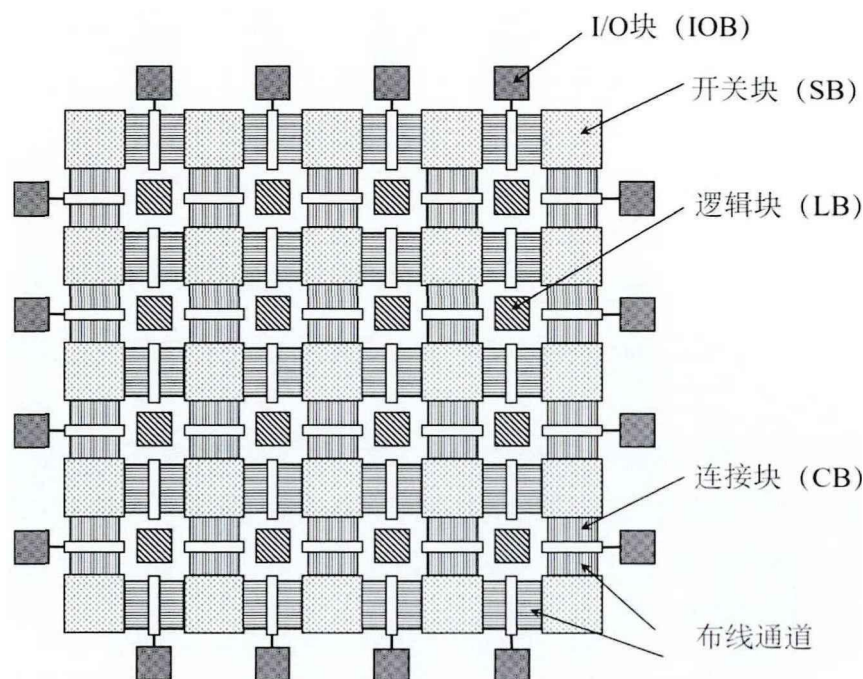


图 2-10 FPGA 结构

FPGA 由逻辑块、输入/输出块三大部分构成：

第一部分是逻辑块 (Logic Block, LB)，是实现各种逻辑电路功能的基本单元，负责实施各式各样的逻辑运算，其灵活的配置方式使得能够轻松适应不同的设计需求；

第二部分是输入/输出块 (Input/Output Block, IOB)，是 FPGA 与外界沟通的桥梁，提供器件引脚和内部逻辑阵列之间的连接，保障了内外信号的顺畅交流，并满足不同电气特性下的输入/输出功能要求；

第三部分是连接前两种元素的布线要素，包括布线通道、开关块 (Switch Block, SB) 和连接块 (Connection Block, CB)，可实现了逻辑块间以及逻辑块与输入/输出块之间的互联，为构建起具有复杂功能的系统提供了支撑。布线长度及制造工艺直接影响信号的传递效率和速度，是系统性能的关键决定因素。虽然具有上述几种要素就可以实现任意逻辑电路，但是在实际的 FPGA 中还有其他一些必要组成，如时钟数、配置、扫描链、测试电路等。此外，商用 FPGA

中还包含处理器、块存储器、乘法器等固定功能的硬核电路。

下面将深入分析源于 FPGA 独特硬件结构的优势及关键特性：

(1) 可编程性和灵活性

FPGA 的可编程性和灵活性指的是用户能够在现场通过编程来配置或修改内部的逻辑功能以适应不断变化的应用需求，而无需更换硬件本身。FPGA 内部包含大量可编程逻辑块，这些逻辑块可以通过用户定义的逻辑功能进行配置，从而实现从简单的逻辑门到复杂的数字系统的任意逻辑操作。FPGA 内部的互连资源也是完全可编程的，允许不同的逻辑块之间按需建立连接，支持不同配置的数据路径和信号处理流程。FPGA 的逻辑功能是通过硬件描述语言进行编程和配置的，这种语言提供了一种高级的、抽象的方式来描述电路的逻辑行为，能够轻松修改和更新 FPGA 的功能。

(2) 并行处理能力

FPGA 通过并行处理机制，能同时执行多个任务或数据流，减少了等待时间。FPGA 内部集成了大量可编程逻辑单元，这些单元可独立配置以执行特定任务，使得多个数据流能够并行处理，从而大幅提高处理速度和效率。以图像处理为例，FPGA 能同时处理多个像素或图像块，而传统 CPU 只能逐行扫描。此外，FPGA 的并行逻辑单元能执行硬件级别的并行操作，如数组运算和位操作，实现了更高的处理效率和速度。这种硬件并行处理使 FPGA 在处理高吞吐量、低延迟的应用中具备显著优势。

FPGA 凭借其可编程逻辑块、输入/输出块以及灵活的布线结构，展现出卓越的可编程性、灵活性和并行处理能力。这种独特的硬件结构使得 FPGA 能够针对各种高性能计算任务进行高效配置和优化，特别是在需要高并行计算性能和低延迟的实时应用场景。

2.3.2 基于 Vivado 的开发流程

Vivado 是 Xilinx 推出的高级综合和分析设计软件，采用了先进的综合算法，可以将高级硬件描述语言代码更高效地转换为 FPGA 上的逻辑表示。基于 Vivado 的 FPGA 开发流程包括方案规划、设计输入、功能仿真、综合优化、布局布线、时序分析和硬件测试等七个步骤。

(1) 方案规划

在开发设计的初始阶段，需要对项目进行方案设计和芯片选型。根据功能需求、性能目标、成本预算等需求分析，选择合理的设计方案和芯片型号。通常情况下，采纳自顶向下的设计方法，即将整个系统进行宏观划分，将其分

解为若干关键模块,再进一步将各模块进一步拆分为更细致的单元,如此反复,直至可映射为 EDA 工具库的元件。

(2) 设计输入

设计输入是指将所设计的系统转换为可以在 FPGA 上实现的具体过程。可以直接使用硬件描述语言 Verilog HDL 或 VHDL 编写代码描述硬件的行为和结构,也可以使用图形化的设计工具,如 Xilinx 的 Vivado IP Integrator,通过拖拽连接预配置的 IP 核来构建设计。这种方式提高了设计的速度和可靠性,适合于需要集成多个复杂 IP 核心的大型项目。此外,还可以使用 MATLAB/Simulink 等工具进行基于模型的设计,然后通过特定的转换工具将模型转换为 HDL 代码,但这种方法效率很低,且不易维护,不利于系统修改和拓展^[69]。因此,在实际开发中,直接编写 HDL 代码仍是主流做法,同时结合图形化设计工具以优化设计流程和提高项目效率。

(3) 功能仿真

功能仿真是指在实际编译之前,使用 Vivado 的仿真功能或调用第三方平台 ModelSim,编写测试用例,创建测试平台,生成模拟输入信号,然后观察 HDL 代码在这些输入信号下的响应,以验证其行为是否符合预期。如果发现错误,则返回设计输入修改代码。这种仿真仅验证设计的逻辑是否正确,确保没有功能性错误,不涉及时延信息。这样,即使在没有物理硬件的情况下,也可以对设计进行检验,能够在早期发现和修正错误,从而减少后期修改的成本。

(4) 综合优化

综合优化是指利用 Vivado 的综合工具将硬件描述语言代码转换为 FPGA 上的具体逻辑实现,也就是查找表、寄存器、I/O 端口等基本逻辑单元组成的逻辑连接,即网表文件。同时,在满足约束的前提下,综合工具还对 HDL 代码中的逻辑表达式和结构进行优化,合并简化逻辑运算,减少所需的逻辑资源。此外,还调整逻辑布局和管脚分配,优化数据路径,以满足设计的时序要求。

(5) 布局布线

布局与布线是指根据芯片的型号,将经过综合优化后的逻辑网表映射到该芯片上的实际物理资源上。布局环节专注于确定逻辑元件在 FPGA 芯片上的具体位置,目的是优化逻辑元件的分布。而布线环节则负责规划逻辑元件间的物理连接路径,确保设计满足所有电气和时序约束。这一过程大多由如 Vivado 等专业 FPGA 设计工具自动执行,这些工具采用高级算法自动进行布局布线,并允许用户对关键部位进行手动优化,以确保设计满足特定的需求和约束条件。

(6) 时序分析

时序分析，也被称作后仿真，是指在布局和布线完成后，分析设计是否满足所有的时序要求，包括设置时间、保持时间、数据链路延迟等关键时序参数，确保设计在指定的工作频率下能够稳定工作，没有数据竞争、迟滞等问题。Vivado 提供了强大的图形化界面和命令行工具，可以自动分析整个设计的时序性能，识别时序违规的地方并提供详细的报告。如果时序报告中存在违反时序约束的情况，则需要根据报告中的指示对设计进行修改和优化，然后重新进行综合、实现和时序分析，直到满足所有时序要求。

（7）硬件测试

在开发设计的最终阶段，将生成比特流文件并通过 Hardware Manager 烧录至 FPGA 芯片中，随后在实际环境下对硬件执行全面测试，以验证其功能是否符合预期。该测试还包括性能测试、稳定性和可靠性测试，旨在验证设计是否满足预定的工作频率和响应时间等性能指标，并确保其在长期运行或多变工作条件下保持稳定可靠。在测试过程中，利用集成逻辑分析器（Integrated Logic Analyzer, ILA）等高级调试工具实时监控和分析 FPGA 内部信号，有助于快速定位问题、验证设计假设及性能优化。

FPGA 开发设计是一个不断迭代和优化的过程，如果在硬件验证过程中发现问题，可能需要返回到之前的步骤，包括修改 HDL 代码、重新综合和布局布线等，直到设计完全满足需求。经过多次迭代设计且在真实硬件环境中彻底通过测试后，还需要完善设计文档，编写用户指南。

2.4 小结

本章深入探讨了模块化多电平换流器及现场可编程逻辑门阵列的工作原理与应用。通过分析 MMC 的子模块工作机制与输出电压波形的控制方法，围绕最近电平逼近调制及其均压策略进行了讨论，阐释了 NLM 在降低谐波含量和提升电能质量方面的有效性，并指出了在大规模 MMC 系统中电容电压均衡与高效处理技术对于维持系统性能的重要性。同时，对 FPGA 的结构和关键特性进行了详细讨论，包括其在电力电子系统中的可编程性、并行处理能力，并详细介绍了基于 Vivado 平台的开发流程。通过这一系列研究，本章不仅加深了对 MMC 和 FPGA 技术的理解，也为后续章节中更为复杂的应用场景和设计挑战奠定了基础。

第3章 适用于FPGA的新型排序算法

3.1 引言

随着柔性直流输电技术向高电压和大容量发展,子模块数量必须增加,以适应更高的耐压要求和改善交流侧的输出波形,这给实现有效的均压控制策略带来了挑战。传统的均压策略由于其计算复杂度高,难以快速完成大量子模块电压的排序,进而影响均压效果。不理想的均压效果可能导致三相桥臂输出电压不一致,诱发波动环流、加剧器件电流应力,增加系统功率损耗,从而威胁到整个MMC系统的安全稳定运行^[70]。因此,开发一种高效且适用于多电平系统的均压控制策略成为确保模块化多电平换流器系统性能和稳定性的关键。

事实上,在开关频率足够高时,每个控制周期内电平变化与模块数量的波动极为有限,仅需要对电压最高或者最低的模块进行投切操作。因此,只需获取子模块电压的极值,而无需关注子模块间详细的电压排列顺序。鉴于此,本章通过对比分析传统排序算法,结合同步全比较与桶排序提出了一种分组并行极值排序算法。然后结合该算法与最近电压逼近调制,针对实际运行需求及降低开关频率的目标进行优化,提出了一种适合并行处理的新型均压控制策略。该策略能够通过简化控制逻辑并减少对实时电压信息处理的需求,有效提高均压控制的效率,其并行化的设计能够充分利用FPGA的高并行处理能力。

3.2 基于分组并行的极值排序算法

3.2.1 传统排序算法对比分析

(1) 冒泡排序法

冒泡排序是一种基本的比较排序方法,它通过连续的两两比较和交换相邻元素来实现数据序列的有序化。每次遍历列表时,若前一个元素大于后一个元素,则交换它们的位置,此过程迭代进行,直至无需进一步交换,表明序列已达到完全有序状态。该算法之所以称为“冒泡”,是因为较小的元素会逐渐移至序列尾端,如同气泡上升至水面,如图3-1所示。对包含五个数值的数组进行排序需执行四轮主循环,每轮完成后,未排序元素中的最大值“冒泡”至其正确位置。随着每轮排序的结束,所需的比较和可能的交换次数减少,最终在第四轮仅需进行单一比较即可完成排序。事实上,该序列在第二轮的时候已经完成了排序。因此,如果发现在某个大循环中,一次交换都没有发生,则可以

提前结束冒泡排序。

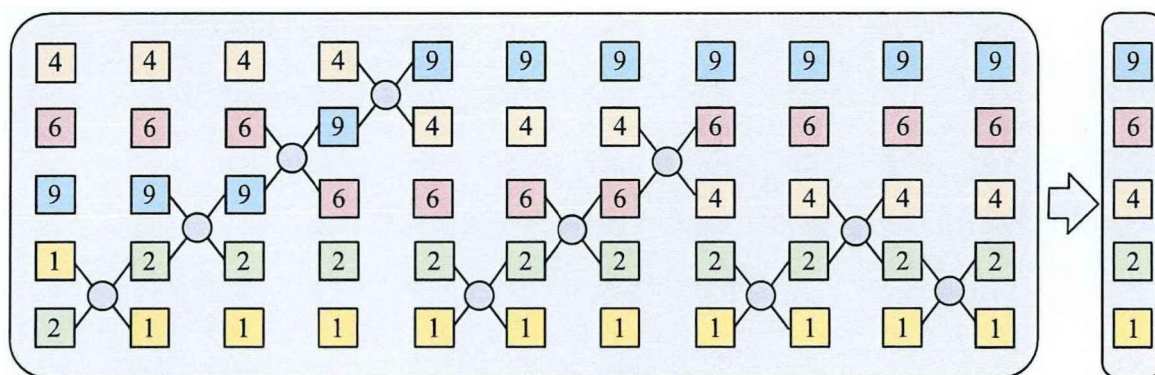


图 3-1 冒泡排序法

(2) 双调排序算法

由 Ken Batchter 提出的双调排序算法是一种基于比较排序的并行排序算法，特别适用于使用硬件进行排序的情况^[71]。双调排序是分治策略的一个应用，它将排序问题分解为形成双调序列的子问题，然后再对这些序列进行合并。双调序列是指一个先连续递增再连续递减的序列，例如[4, 7, 9, 8, 3]。

双调排序算法的基本步骤主要包括构造双调序列以及对双调序列进行分治递归排序两步。下面以一个 8 元素的数组为例，说明如何将初始无序序列转变为一个双调序列。首先将相邻两个元素进行比较排序并合并形成 4 个单调性相反的单调序列，然后两两序列合并，形成 2 个双调序列。随后，分别对这两个双调序列进行双调排序，一个子序列升序排列，另一个降序排列，形成两个长度为 4 相反单调性的单调序列。最后将两个单调序列合并，就能得到长度为 8 的一个双调序列，如图 3-2 所示。

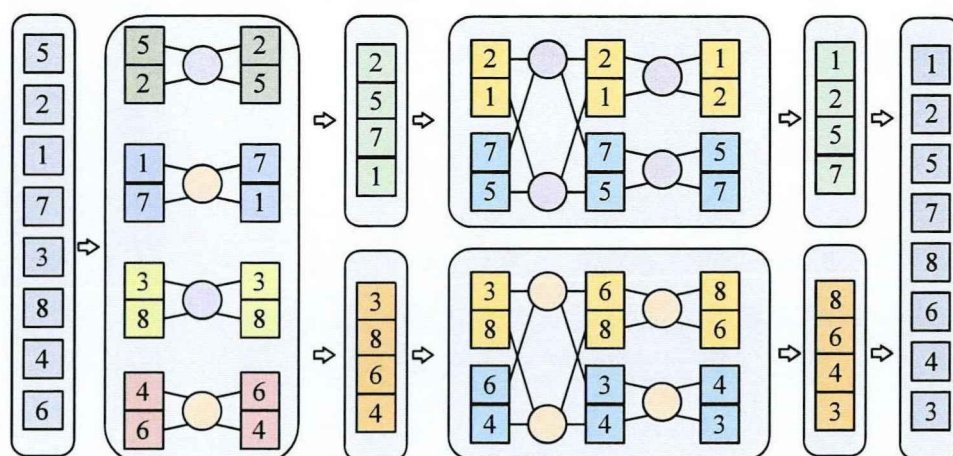


图 3-2 构建双调序列

双调序列进行分治递归排序的步骤如图 3-3 所示，每个“Batcher”方框代表比较-合并单元，负责执行双调合并操作，将相同位置的元素进行比较并根据

大小重新排序。将上个步骤获得的双调序列分割为两个长度等同、单调性相反的子序列。随后，同时比较这两个序列相同位置的元素，较小的元素移动到一个新的子序列中，较大的元素移动到另一个新的子序列中。此分割和比较过程递归进行，直到各个子序列的长度为 1，从而实现整个序列的排序。

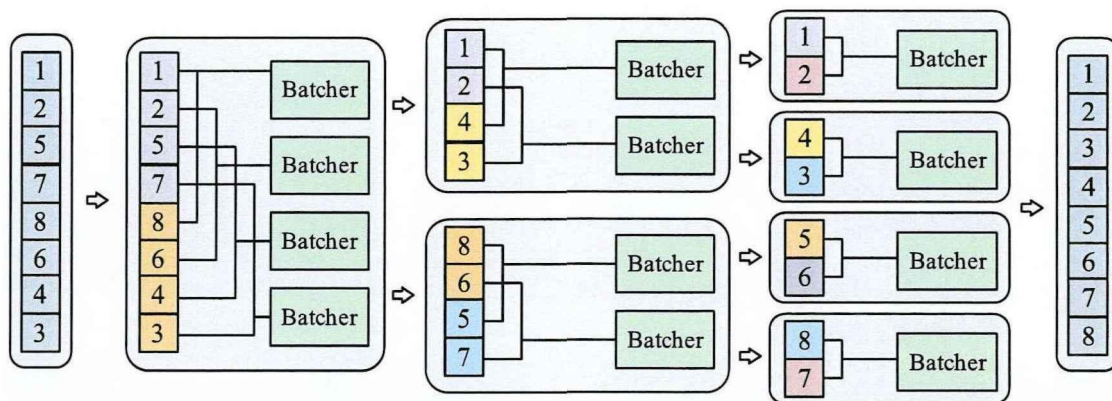


图 3-3 对双调序列进行双调排序

因为多个比较和交换操作可以同时进行，因此该算法具有良好的并行性，在 FPGA 上可以很好地利用它的并行性以提高排序的速度。根据上述的分析可知，双调排序算法要求输入数组的长度为 2 的幂次方，否则无法执行上述递归步骤。如果原数组长度不满足此条件，则必须通过填充零元素来扩充数组至最近的 2 的幂次方长度。这一过程是确保排序网络正确运行的必要步骤，但是可能导致空间浪费。若数据量小或接近 2 的幂次方时，这种浪费并不明显，额外空间使用量不显著。但随着待排序元素数增加，其占用的空间资源急剧上升。以一个包含 1025 个元素的数组为例，其必须扩充至 2048 个元素以满足算法要求，这将导致 1023 个冗余位置被填充，造成近一倍的空间冗余。在处理大量数据时，这种方法会大量占用计算机的存储空间，可能对系统资源造成显著压力。

(3) 并行全比较排序

并行全比较排序算法利用并发比较机制实现排序，通过同时比较数组中任意两个元素来优化处理时间^[72]。在此算法中，对数组中的每个元素执行全比较操作，即将每个元素与其它所有元素进行比较：若当前元素较大，则比较结果记作 1；反之，则记作 0。随后，对每个元素的比较结果进行累加，以此累加值作为该元素在新有序数组中的索引。这种方法根据元素的相对大小确定其最终排序位置，进而组成排序后的序列。该策略的核心在于“以空间换时间”，即通过增加并行处理的空间复杂度来实现排序速度的提升。

在并行全比较排序算法中，对于给定数组[4, 6, 9, 1, 2]，通过并行化比较每个元素与其余元素的大小，可以有效地确定每个元素在最终排序数组中的

索引位置,如图 3-4 所示。此过程首先涉及计算每个元素胜过其他元素的次数,这一计数反映了元素在排序后数组的位置。如表 3-1 所示,元素 A[2] (9) 作为最大值,其胜场次数最高,因此在新数组 B 中被分配到最末位置 (B[4]=9)。按此逻辑,元素 A[3] (1) 为最小值,被分配到新数组的首位 (B[0]=1)。最终,根据这些索引位置重构数组,得到有序序列[1, 2, 4, 6, 9]。

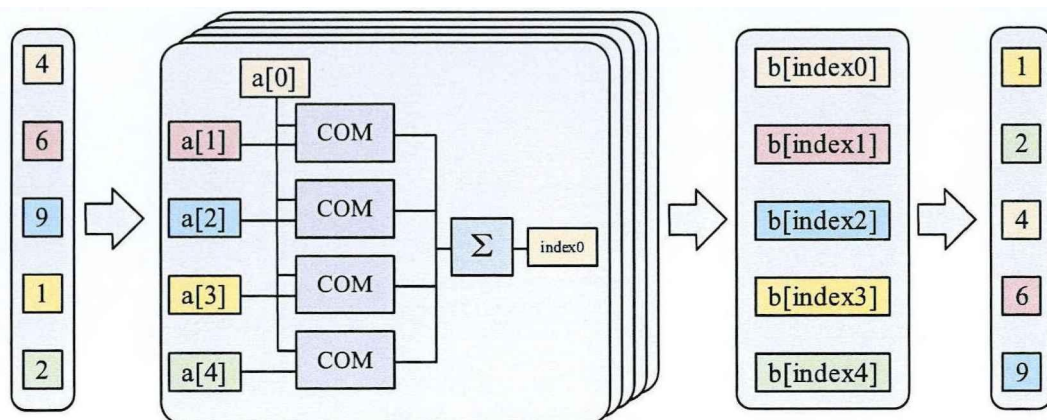


图 3-4 并行全比较排序

表 3-1 并行全比较得分表

	A[0]=4	A[1]=6	A[2]=9	A[3]=1	A[4]=2	index	赋值
A[0]=4	—	0	0	1	1	2	B[2]
A[1]=6	1	—	0	1	1	3	B[3]
A[2]=9	1	1	—	1	1	4	B[4]
A[3]=1	0	0	0	—	0	0	B[0]
A[4]=2	0	0	0	1	—	1	B[1]

根据上述分析可知,并行全比较排序算法通过执行数组中每个元素与其他元素的直接并行比较确定其排序位置,这一机制显著适合并行计算架构。在并行计算的框架下,元素间的比较和位置索引计算可以同时进行,从而有效提高排序效率。然而,算法的性能上限可用的处理器数量和并行处理能力的约束。为了完成比较操作,算法需部署大量比较器,导致空间复杂度显著增加。以 100 个元素的排序为例,若并发执行比较,则需配置接近 10000 个比较器。因此,在处理大规模数据集时,需要权衡算法的资源消耗与其并行加速效益。

(4) 桶排序

桶排序算法是基于分治策略的一种高效排序方法。该算法首先确定输入数据的最大值与最小值,根据最大值和最小值平均划分若干个预定的区间,即“桶”。每个桶代表一个值范围,通过映射函数将数据项分配到各个桶中,映射函数如式 (3-1) 所示。数据分配后,算法对每个桶内的数据进行独立排序。最后,通

过顺序合并各桶中的数据，形成一个有序序列。

$$f(x) = \begin{cases} \text{floor}[\frac{x - x_{\min}}{B_{size}} + 1], & x < x_{\max} \\ B_N, & x = x_{\max} \end{cases} \quad (3-1)$$

式中， $\text{floor}[x]$ 为向下取整函数， x 为待排的数据集； $x_{\min}$ 和 $x_{\max}$ 分别是 x 的最小、最大值； B_{size} 为桶间隔， B_N 为桶数。

以一个具体的数据集[24, 47, 66, 16, 12, 45, 35, 38, 31, 21, 26, 60]为例，说明桶排序的具体过程。在此数据集中，数据的最大值为66，最小值为12。按照数据范围，将数据分成3个桶：[12,32]，[33,54]，[55,66]。随后，数据根据映射函数分配到对应的桶中。在每个桶内对数据进行排序后，最终通过合并每个桶内排序后的数据，得到最终的有序序列。

桶排序算法通过将大规模数据集细分为多个小规模子集来优化排序过程，每个子集对应一个桶，使得对较小数据量的排序问题可以并行处理。该算法的并行性使其能够充分利用多硬件系统，加速整个排序操作。桶的尺寸和数量的动态可调性，可以根据数据分布进行优化，提高了算法的适应性与效率。

3.2.2 分组并行极值排序算法

事实上，在对MMC进行均压控制时并不需要对所有子模块的电容电压进行完整排序^[73]。由2.2.2节内容可知，当桥臂电流方向正时，只需选择电容电压最低的 n 个子模块进行投入；反之，若电流方向负，也仅需选择电容电压最高的 n 个子模块投入。这说明了采用NLM调制进行均压控制时仅需确定数列的极大值或者极小值，而不用得到完整的排序结果。因此，本文提出一种结合桶排序的分布式特性以及并行全比较排序的计算效率的算法，以优化寻找序列极值的过程。下面以寻找最小的 n 个数为例，阐述算法的设计思路。

首先，借鉴桶排序的思想，将数列映射至若干预定义的桶中。计算所消耗时间远大于逻辑算法。因此，为了提高计算效率，不再采用映射函数，而是通过比较每个桶的边界值的方式，进行映射。假设 n 满足式(3-2)，即前 k 个桶包含的元素总数小于 n ，而前 $k+1$ 个桶的总数则达到或超过 n 。那么可以通过对第 $k+1$ 个桶内元素的排序，确定最小的 n 个数。由前 k 个桶的元素以及第 $k+1$ 个桶内最小的 m 个元素，共同构成了所求的最小 n 个数据。

$$\sum_{i=1}^k N(i) < n \leq \sum_{i=1}^{k+1} N(i) \quad (3-2)$$

$$m = n - \sum_{i=1}^k N(i) \quad (3-3)$$

针对第 $k+1$ 个桶内的元素排序采用并行全比较排序算法，以便利用其分组后小数据量的特性实现并行化处理，从而提升计算效率。尽管并行全比较排序在处理大数据量时的资源占用率较高，但在经过分组后的小规模数据集上，其资源消耗相对较低。并行全比较排序的显著优势在于其能够在有限的时钟周期内迅速完成排序，适合于实时系统中的快速控制需求。

综上所述，本研究提出的算法结合了桶排序的高效数据分配机制与并行全比较排序的快速处理能力，为寻找极值提供了一种有效的解决方案。通过对小规模数据集的快速排序，该算法能够在保证计算效率的同时，减少资源的消耗，满足实时控制的需求。

以一个具体的数据集[24, 47, 60, 16, 69, 45, 35, 80, 31, 21, 26, 11, 12, 55, 68, 76, 86, 32, 72, 14]为例，说明分组并行极值排序算法如何寻找最小的 $n=12$ 个元素的具体实现过程，如图 3-5 所示。具体可分为以下五步：

1) 桶界定：遍历该数据集，找到最大值为 86，最小值为 11，因此数据范围为 75。据此，数据集被均匀地划分成 5 个桶，每个桶的跨度为 15，具体桶的区间定义为：[11, 26)，[26, 41)，[41, 56)，[56, 71)和[71, 86]。

2) 数据映射：通过比较边界值，将所有数据根据其值分配到对应范围的桶中。此过程涉及将数据点与桶区间边界进行比较。

3) 计数累积：在分桶过程中，统计每个桶内的数据点数量，并计算至第 i 个桶的累积数据点总量。

4) 识别关键桶与排序：通过比较累积数据点总数与目标数 n 的大小，确定关键桶 k ，使得前 k 个桶中的元素数量少于 n ，而前 $k+1$ 个桶的数量达到或超过 n 。随后，对第 $k+1$ 个桶（此例中为第 3 个桶）的数据点进行并行全比较排序。

5) 极值元素筛选：通过式 (3-3) 确定 $m=2$ 值，则从第 3 个桶中找到最小的 2 个数据点。最后将这 m 个数据点与前 k 个桶中的数据点合并，即可得到整个数据集中最小的 12 个数为[26, 16, 21, 26, 11, 12, 14, 35, 32, 47, 45]。

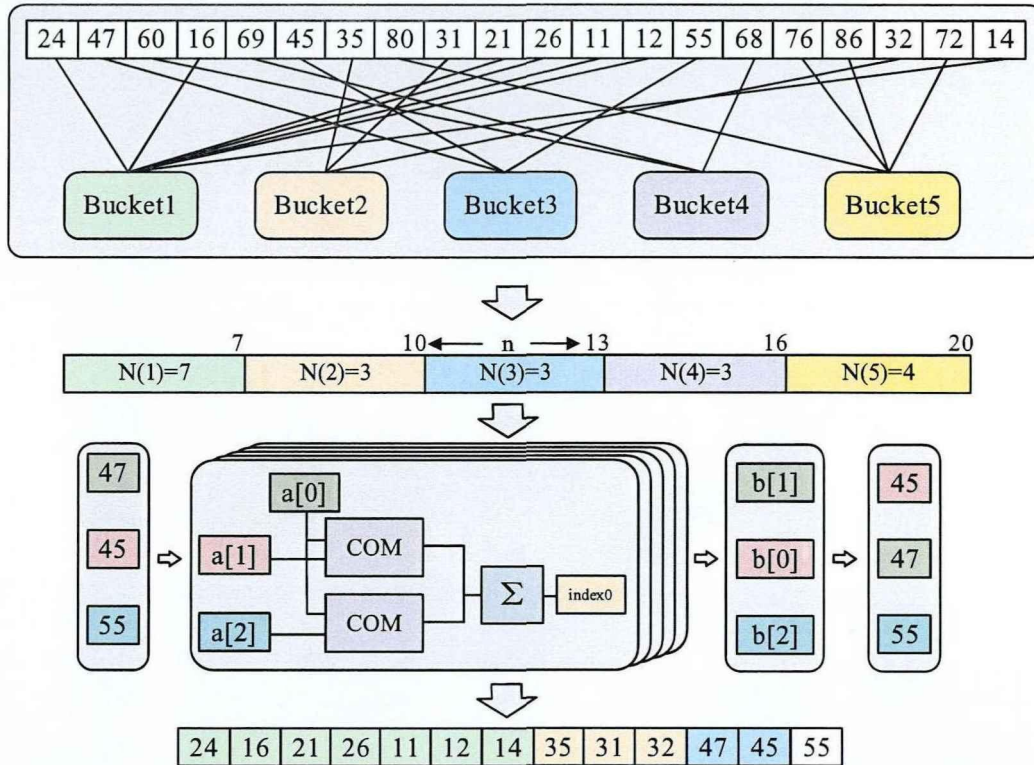


图 3-5 分组并行极值排序算法

3.2.3 最优分组数的参数设计

一般来说，排序算法的空间复杂度主要取决于所需比较器的数量，其所需比较器总数越少，则排序算法的所消耗的资源越少。假设待排序元素均匀分布，则每个分组内的元素数量为 n/L 。在分组阶段，以线性搜索法确定元素归属，最坏情况下需进行 $L-1$ 次边界值比较。因此，对于 n 个元素，总比较次数达到 $n \times (L-1)$ 。此外，根据 3.2.1 节分析，在并行全比较阶段，每个元素需同时与其他元素进行比较，则需要的比较器数量为 $(n/L) \times (n/L-1)$ 。综上，分组并行极值排序算法的总比较器需求量可表示为：

$$S = n \times (L-1) + \frac{n}{L} \times (\frac{n}{L} - 1) \quad (3-4)$$

总比较器需求量随元素个数和分组数变化的三维曲面图如图 3-6 所示，其中横轴表示“元素个数”，纵轴表示“分组数”，而曲面的高度则代表“总比较器的需求量”。从图中可以看出，分组数对需求量的影响十分显著，随着分组数的增加，需求量呈现出先减少后增加的趋势。这种趋势可以通过曲面图中的局部最小值区域来识别，该区域在图中以绿色标识，表示在特定的分组数下，总比较器的需求量达到最低点，从而反映出算法效率的最优状态。进一步地，

该局部最小值区域的存在表明，在给定的元素个数范围内，存在一个最优的分组数，使得比较器的需求量达到最小。这一现象与排序算法中比较器的分配策略和分组机制的效率密切相关。在最优分组数下，算法能够以最少的比较器资源完成排序任务，从而实现资源的最优分配。

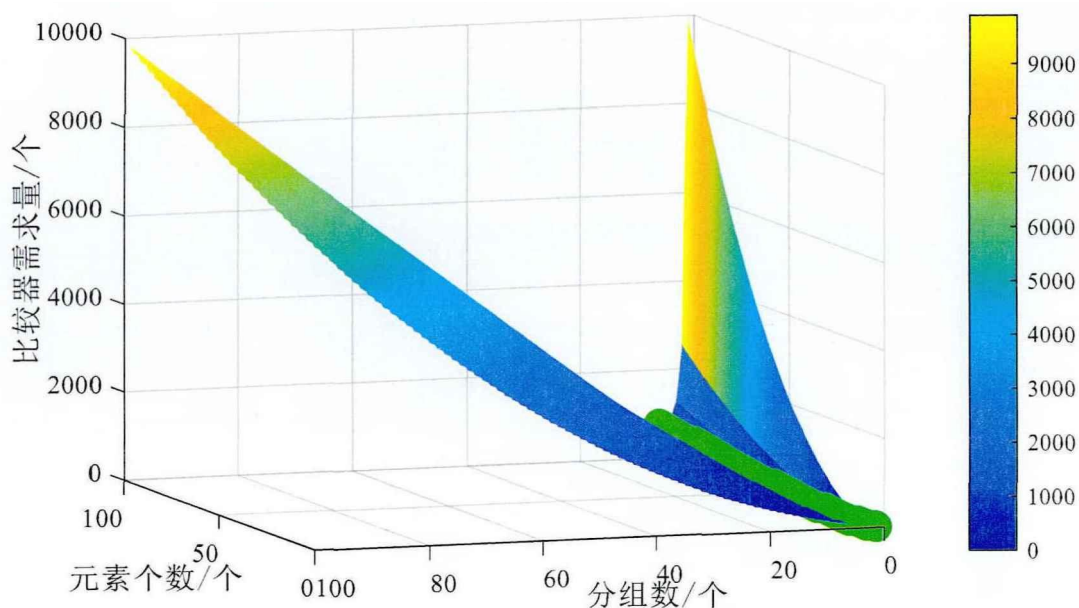


图 3-6 总比较器需求量的三维曲面图

为了确定每元素个数 n 下的最优分组数 L^* ，对比较器需求量函数 S 关于分组数 L 进行求导。这一过程旨在通过数学分析揭示在不同数据规模下，哪个分组数能够最小化比较器的需求量。具体地，对函数 S 执行偏导数运算，得到 dS/dL ，如式 (3-5) 所示。通过求解该导数等于零的方程，即可求得使比较器需求量达到最小值的 L 值，即最优分组数 L^* 。

$$\frac{dS}{dL} = n - \frac{2n^2}{L^3} + \frac{n}{L} \quad (3-5)$$

然而，由于目标函数的复杂性，这一直接方法无法求得显式解。鉴于此，转而考虑一种替代的求解策略。考虑到数据规模和分组数在实际应用中通常被限制为正整数，因此，采用逐个遍历搜索的方法来确定最优分组数。具体而言，针对每个数据规模遍历所有可能的分组数的整数值，并针对每个计算相应的目标函数的值。通过比较这些值，能够识别出使目标函数达到最小化的值，即最优分组数。这种方法虽然在计算上可能较为耗时，但它确保了在整数限制条件下，能够找到全局最优解。最终，得到了一个最优分组数与数据规模之间关系的对应表，如表 3-2 所示。该表格为不同数据规模下的最优分组策略提供了明确的指导，具有重要的实际应用价值。

表 3-2 最优分组数对应表

数据量	最优分组数	数据量	最优分组数
$1 \leq n < 3$	1	$139 \leq n < 213$	7
$3 \leq n < 9$	2	$213 \leq n < 310$	8
$9 \leq n < 23$	3	$310 \leq n < 432$	9
$23 \leq n < 47$	4	$432 \leq n < 582$	10
$47 \leq n < 85$	5	$582 \leq n < 764$	11
$85 \leq n < 139$	6	$764 \leq n < 980$	12

为提升计算结果的普遍适用性，本研究采用曲线拟合技术，以获得 L^* 与 n 间的近似函数关系，从而形成更广泛适用的表达式。观察曲线图像，明显可见在最优分组数存在跳变现象。通过遍历 L^* 与 n 的数据集，识别出 L^* 发生跳变的特定 n 值，并利用这些跳变点采用指数模型进行曲线拟合。进一步，依据跳变点的变化趋势，选用对数函数模型进行拟合，并执行向下取整处理，使拟合曲线更贴近原始的关系，形成拟合公式（3-6）。所得拟合曲线如图 3-7 所示。

$$L^* = \text{floor}[2.128 \ln n - 3.098] \quad (3-6)$$

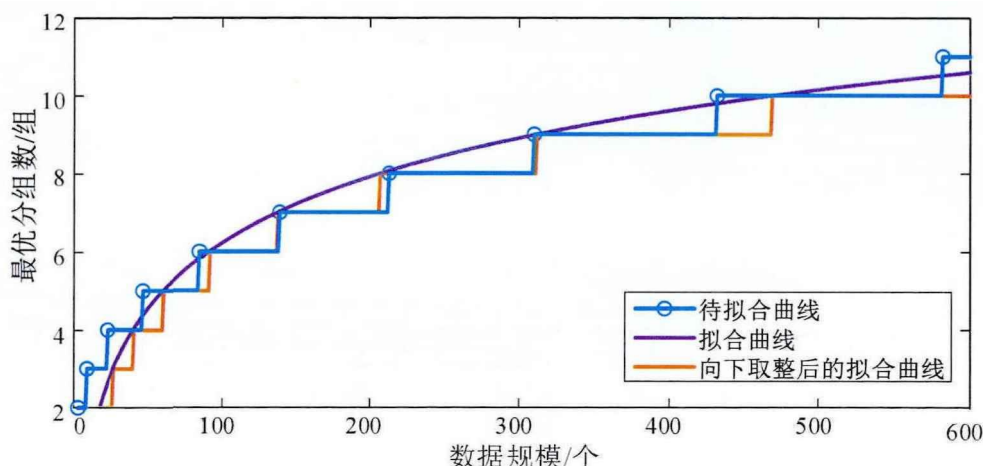


图 3-7 最优分组数的拟合曲线

为验证所得模型的准确性，采用均方根误差 (Root Mean Square Error, RMSE) 作为度量标准，对模型预测的 L^* 值与通过数值优化方法获得的实际 L^* 值之间的差异进行评估。RMSE 的计算公式如下：

$$RMSE = \sqrt{\frac{1}{600} \sum_{n=1}^{600} (L_n^* - L_n^*)^2} \quad (3-7)$$

式中， L_n^* 为模型预测的数据量为 n 时对应的最优分组数； L_n^* 为数值优化得到的

数据量为 n 时对应的最优分组数。

计算得到均方根误差为 $RMSE = 0.574 < 1$ ，这表明模型预测的 L^* 与通过数值优化获得的实际 L^* 值之间的偏差非常小，因此该模型具有很高的预测准确性。在此情况下，可以认为模型在统计意义上对于数据集具有良好的拟合度，能够根据数据规模可靠地预测最优分组数 L^* 。

3.2.4 性能评估

排序算法的性能评估一般包括空间复杂度和时间复杂度。

(1) 空间复杂度

正如 3.2.3 节所提到的排序算法的空间复杂度主要取决于所需比较器的数量。因此，将最优分组数的预测公式式 (3-6) 代入式 (3-4) 即可得到总比较器需求量随数据规模变化的曲线，如图 3-8 中蓝色曲线所示。

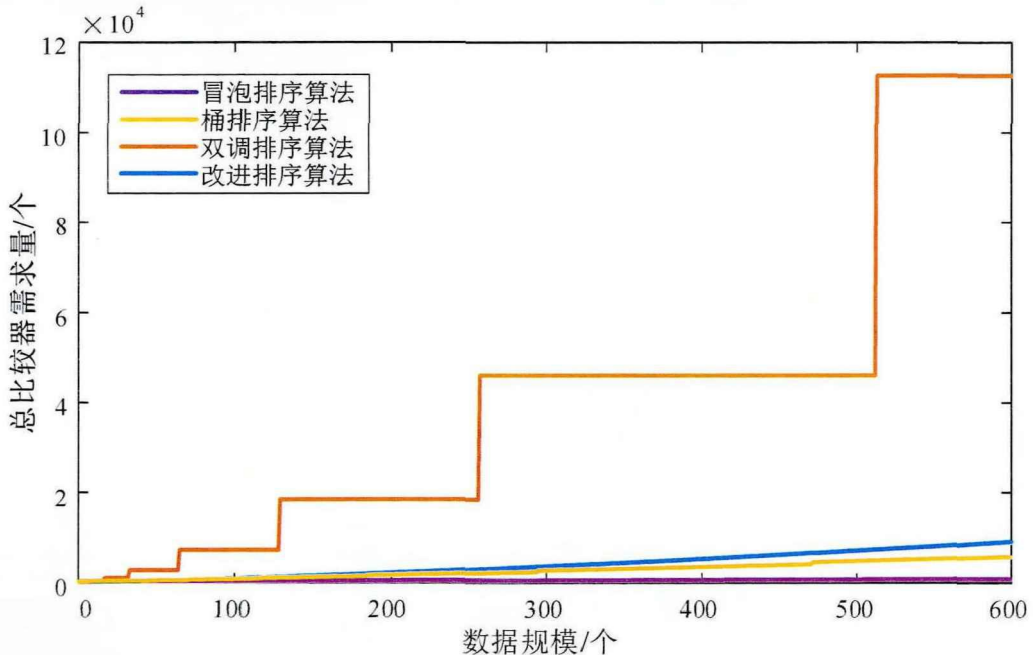


图 3-8 总比较器需求量对比图

冒泡排序法是传统的串行排序算法，仅消耗一个比较器，因此一般不讨论其空间复杂度。在并行排序算法中，当数据规模小于 30 时，双调排序算法和本文所提改进算法所需的总比较器需求量差别不大，双调排序算法略高于改进算法。随着数据规模的增大，双调排序算法因其固有的特性（数据量必须为 2 的幂次方，否则需要扩充至最近的 2 的幂次方），导致总比较器需求量呈阶梯式激增，造成资源浪费，而改进排序算法以近似线性增加，呈现出优越的空间性能，略高于基于冒泡排序的桶排序算法。

(2) 时间复杂度

在综合评估分组并行极值排序算法的时间复杂度时，根据其主要步骤，可以将整个算法分析划分为以下三个关键阶段：

1) 最值确定阶段

此阶段通过遍历整个数据集以识别数据的最大值和最小值，因此每个数据点都需要被处理一次。假设数据集有 n 个元素，由于遍历是顺序执行的，其时间复杂度可表示为：

$$T_1 = 2n \quad (3-8)$$

2) 数据映射阶段

此阶段涉及将数据集的各个元素根据其数值与预设桶间隔的关系分配至相应桶中。假设有 n 个数据点被分配至 L 个桶中，考虑到每个数据点需要与 $L-1$ 个边界值进行比较以确定其所属桶，因此，该过程的时间复杂度可表示为：

$$T_2 = n \times (L-1) \quad (3-9)$$

并行化技术在算法性能提升中发挥关键作用，特别是在处理大规模数据集时，能够显著减少算法的执行时间。在第一阶段，通过并行计算技术的应用，算法能够对数据集进行单次遍历的同时，有效地识别出数据集中的最大和最小值，减少了对数据的多次扫描需求。同样，在第二阶段，每个数据点与所有桶的边界值进行的比较也可以并行执行，也仅需对整个数据集执行一次遍历。因此，前两个阶段的时间复杂度可降为：

$$T_1 + T_2 = 2n \quad (3-10)$$

3) 桶内排序

在该算法中，该阶段只需对第 $k+1$ 个桶需要进行排序，因此桶的排序时间复杂度取决于并行全比较算法。假设第 $k+1$ 个桶内有 m 个元素与 p 个处理器。那么算法的时间复杂度将受限于比较操作次数与处理器数量的比例。对于 m 个元素的数组，总共需要进行 $m \times (m-1)$ 次比较。如果每个处理器同时执行比较操作，那么时间复杂度为：

$$T_3 = \frac{m \times (m-1)}{p} \quad (3-11)$$

考虑到分组后元素数 m 相对较小，且在假定数据均匀分布的前提下，理论上有足够的处理器以支持高效并行操作，时间复杂度可以达到常数。因此，第三阶段的时间复杂度对总体影响较小。

因此，整个算法的时间复杂度主要由前两个线性步骤决定，为：

$$T_{\Sigma}(n) \approx 2n \quad (3-12)$$

综上所述，分组并行极值排序算法的时间复杂度是 $O(n)$ 。

时间复杂度对比如图 3-9 所示，其中横坐标为数据数量，纵坐标为比较次数。从图中可见，随着数据量的增长，冒泡排序的比较次数急剧上升，呈现出二次方的增长趋势。在此基础上，桶排序通过有效的分组策略降低了比较次数，尽管如此，其性能提升相较于其他算法而言仍有限。双调排序的性能受限于其固有特性，在不满足数据规模要求时需通过填充极值进行调整，导致比较次数的非连续性跳变。而改进的分组并行极值排序算法呈现出了与数据量直接成线性关系的比较次数增长趋势，其时间复杂度的线性特征在各种数据规模下均优于其他算法，证明了其在效率上的显著优势。

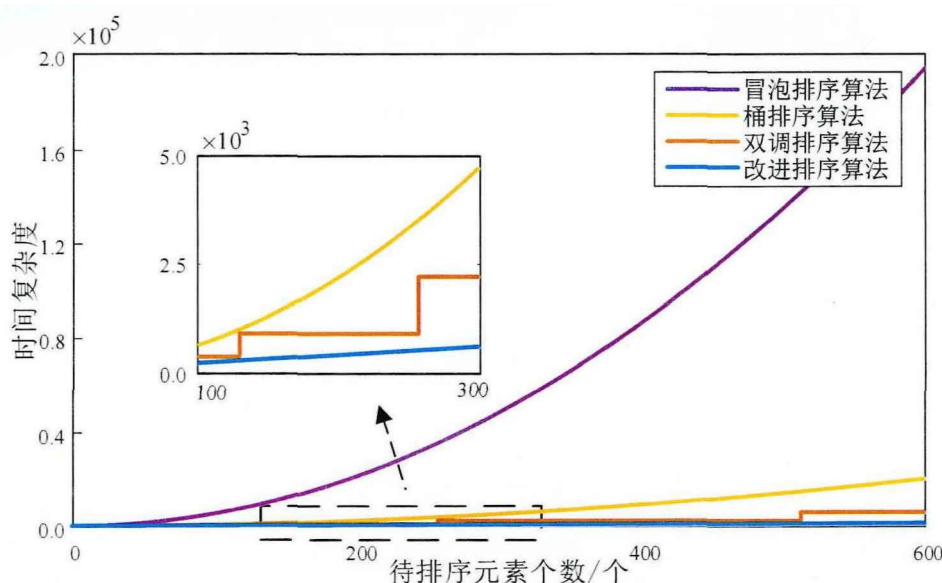


图 3-9 时间复杂度对比

在基于 CPU 的并行计算框架中，操作的并行化程度主要取决于处理器内的核心数，每个核心负责独立的计算任务。MATLAB 通过其 Parallel Computing Toolbox 提供了一套工具和函数，支持从串行到并行代码的转换。PCT 利用并行循环、并行任务及数据并行机制来提升多核心环境下的计算效率，并通过 spmd 块及在多个工作进程间的数据和计算分配，支持复杂任务的并行处理。

算法的运行时间可能会受到多种因素的影响，包括 CPU 性能、系统负载、并行工作进程的数量以及 MATLAB 的其他配置。因此，运行时间的测量可能会有所波动，应在在稳定条件下进行多次测量以获得平均值以确保结果的可靠性。本研究所使用的计算设备配备了第 12 代英特尔酷睿 i5-12400 处理器 (2.50GHz)，拥有 12 个 CPU 核心和 16.0GB 内存。在 MATLAB R2018a 环境

下,对改进的分组并行极值排序算法进行时间性能评估。通过生成包含 50 至 1000 个元素、元素随机分布于 $[1, 99]$ 区间的数组(分组数设为 10),并利用 tic 和 toc 函数记录代码执行时间,重复 100 次以计算平均运行时间。实验结果如图 3-10 所示,其中横坐标为随机数据量,纵坐标为平均运行时间。

结果表明,元素的分布对改进的分组并行极值排序算法的运行时间有所影响,但是运行时间与数据量之间基本呈线性关系,这与理论分析相吻合。这种线性关系的存在验证了并行处理技术在优化算法运行时间方面的有效性。

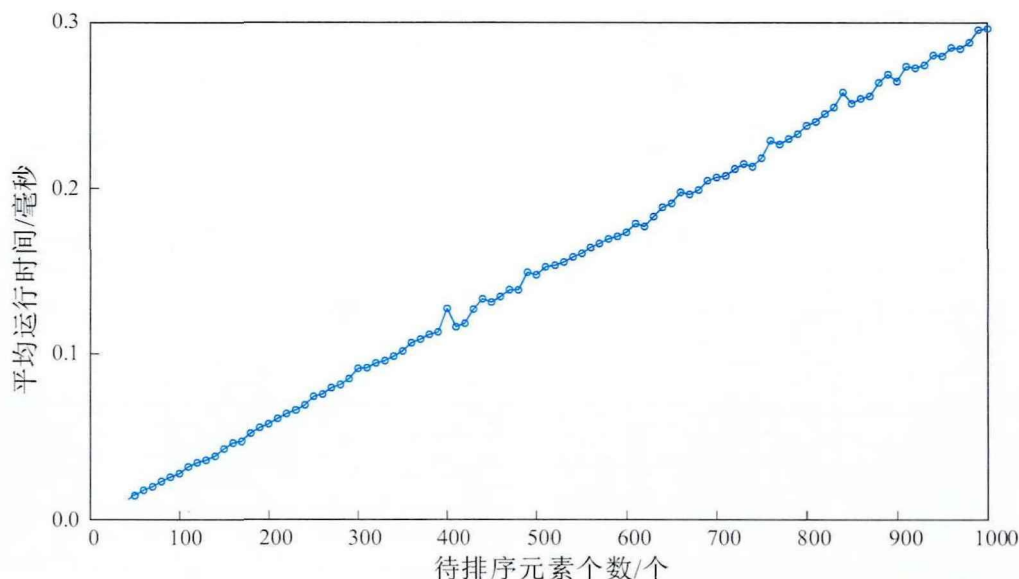


图 3-10 分组并行极值排序算法时间性能

综上所述,本研究提出的改进排序算法不仅在时间性能方面表现出较高的效率,而且在空间利用效率上也展现了显著的优势。与传统方法相比,它在处理大规模数据集时展现出更快的排序速度及更低的内存占用,有效减少了空间复杂度。这种时间与空间性能的双重优化,为处理复杂数据提供了一种更加高效且经济的解决方案,具有广泛的应用前景和实际价值。

3.3 考虑开关频率优化的新型排序算法

结合 3.2 节提出的新型均压算法与最近电平逼近调制,形成了一种新的均压控制策略。此策略融合了两种技术的优点,但在开关频率的优化方面尚有不足。鉴于此,本节将着重探讨如何根据实际运行需求对分组策略和投切逻辑进行细化优化,以提高系统的整体性能。

3.3.1 考虑实际运行需求的分组策略优化

针对所提策略中需遍历所有子模块电容电压以确定最大和最小值后进行分

组的步骤，引入了基于实际运行需求的改进策略。考虑到子模块电容电压在正常运行条件下趋于稳定，且主要分布在额定值附近^[74]，本优化方法通过预设电容电压的上下限值（ $V_{\min}$ 和 $V_{\max}$ ），直接对电容电压进行分组，从而避免了对所有子模块电容电压进行遍历的步骤。具体而言，电容电压低于下限 $V_{\min}$ 的子模块被划分为一组，高于上限 $V_{\max}$ 的为另一组，而位于 $[V_{\min}, V_{\max}]$ 区间内的电容电压则按照 L 的分组策略进行细分，如图 3-11 所示。

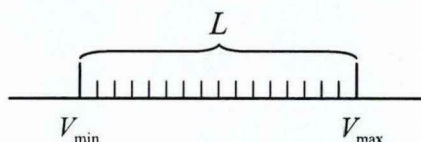


图 3-11 基于实际运行需求的改进分组

则子模块电容电压对应的分组序号为：

$$k(V_c) = \begin{cases} 0 & , V_c \leq V_{\min} \\ \text{ceil}\left[\frac{V_c - V_{\min}}{\Delta V^*}\right] + 1 & , V_{\min} < V_c < V_{\max} \\ L + 1 & , V_c \geq V_{\max} \end{cases} \quad (3-13)$$

式中， $\text{ceil}[x]$ 为向上取整函数； ΔV^* 为基于实际运行需求的改进分组后的桶间隔，如式（3-14）所示。

$$\Delta V^* = \frac{V_{\max} - V_{\min}}{L} \quad (3-14)$$

此优化不仅简化了电容电压的分组流程，提高了算法的效率，而且通过减少遍历过程，加快了电容电压均衡控制的响应速度。更重要的是，这种方法有助于维持子模块电容电压在其额定值附近运行，提高了系统的稳定性和可靠性。

3.3.2 考虑投切状态的开关频率优化

为确保各子模块电容电压的均衡，需要快速监控并调整子模块的导通状态，导致模块开关动作频繁，使得 MMC 存在较高的开关频率和损耗。因此，针对子模块电容电压的均衡与低开关频率之间存在的固有矛盾，需要相应的优化方法以实现电压平衡和降低开关频率的双重目标。

传统的开关频率优化方法通过应用双保持因子排序来降低频率，即在桥臂电流为正时，将投入状态子模块的电容电压乘以小于 1 的保持因子，而切除状态子模块电容电压乘以大于 1 的保持因子，以期维持子模块的当前状态。然而，这种方法虽可减少开关频率，却增加了计算复杂度，并需要大量试验来确定最

佳保持因子。不精确的因子设置可能导致电容电压的不均衡。因此，研究一种既能有效平衡电容电压又能减少开关频率，且计算复杂度低的优化策略，对于提高模块换流器的性能和可靠性至关重要。

本研究在所提均压策略的基础上，引入一种考虑前一操作周期开关状态的优化策略。该策略旨在维持子模块电容电压在安全区间 $[V_{\min}, V_{\max}]$ 内的同时，减少开关操作的频率以减少开关损耗。假设该时刻应投入的子模块个数为 N_{on} ，则该策略详细过程如下：

1) 当桥臂电流方向为正时，优先投入电容电压低于 $V_{\min}$ 的子模块。随后，按照电容电压由低至高的顺序，先投入前一步长已投入且处于 $[V_{\min}, V_{\max}]$ 区间内的子模块。如果这些子模块已全部投入而 $\sum n_k < N_{on}$ ，接着继续投入区间内剩余的子模块；若区间内子模块也已全部投入而子模块个数仍未达到 N_{on} ，则继续投入高于 $V_{\max}$ 的子模块。

2) 当桥臂电流方向为负时，首先投入电容电压高于 $V_{\max}$ 的子模块。接下来，按照电容电压由高至低的顺序，先投入前一步长已投入且位于 $[V_{\min}, V_{\max}]$ 区间内的子模块。若这些子模块已全部投入而 $\sum n_k < N_{on}$ ，则继续以从高到低的顺序投入区间内剩余的子模块；若 $[V_{\min}, V_{\max}]$ 区间内子模块均已投入而子模块个数仍未达到 N_{on} ，则转向投入低于 $V_{\min}$ 的子模块。

此策略通过在电容电压控制中引入子模块的前一状态，在计算复杂度未显著增加的情况下，不仅确保了电容电压的稳定运行在规定区间，还有效降低了开关频率，缓解了开关损耗问题。该方法实现了电容电压均衡与低开关频率之间的平衡，优化了模块换流器的运行效率和可靠性。

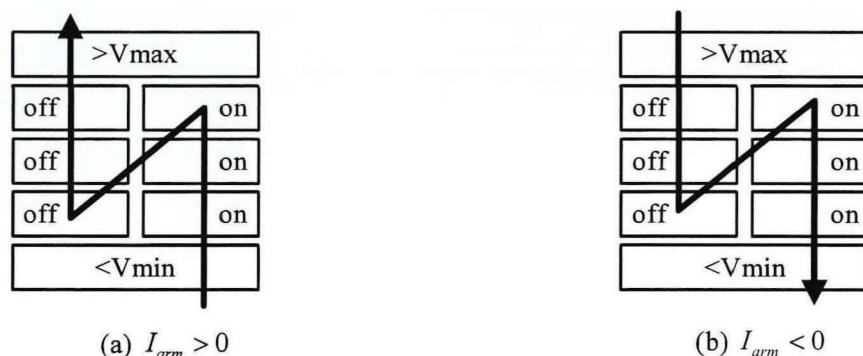


图 3-12 开关频率优化策略

3.3.3 基于新型排序算法的均压控制策略

结合上述针对均压应用场景的排序算法优化与最近电压逼近调制策略，提出一种基于新型排序算法的电容电压均衡控制策略，具体流程如下：

- 1) 利用最近电平逼近调制环节确定当前时刻所需投入的子模块数量 N_{on} 。
- 2) 根据 N_{on} 的大小做初步处理, 若 $N_{on} = N$ 则该投入桥臂所有子模块, 若 $N_{on} = 0$, 则切除该桥臂所有子模块, 而当 $0 < N_{on} < N$ 则进行下一步。
- 3) 采集子模块电容电压 $V_{C,i}$ 、桥臂电流 I_{arm} 。
- 4) 根据预设的子模块电容电压最高值 $V_{\max}$ 和最低值 $V_{\min}$, 计算每个桶的边界值。通过子模块电容电压与每个桶边界比较, 将其分配到对应的桶中, 其中边界值如式 (3-15) 所示。

$$\begin{cases} V_{bk} = V_{C\min} + k \times \Delta V \\ \Delta V = \frac{V_{\max} - V_{\min}}{\text{floor}[2.128 \ln N - 3.098]} \end{cases} \quad (3-15)$$

- 5) 根据桥臂电流方向确定需要投入或切除的电容电压最小的子模块个数 N_k 。当 $I_{arm} > 0$ 时, 选取电容电压最低的 N_{on} 个子模块进行充电, 即识别 N_{on} 个最小电容电压值, 当 $I_{arm} < 0$ 为负时, 选择电容电压最高的 N_{on} 个子模块执行放电操作, 即识别 $N - N_{on}$ 个最小电容电压值。根据 I_{arm} 计算 N_k :

$$N_k = \begin{cases} N_{on}, I_{arm} > 0 \\ N - N_{on}, I_{arm} < 0 \end{cases} \quad (3-16)$$

- 6) 统计各桶中的子模块数目, 计算累积数据量, 然后根据 N_k 计算关键桶的位置和需要投入的第 $k+1$ 个桶的子模块数量 m , 并在此桶中执行并行全比较排序, 找到 m 个极大值或极小值。

- 7) 整合各桶数据, 根据桥臂电流方向与所提投切原则选择 N_k 个子模块, 生成相应子模块开关信号。

基于新型排序算法的均压控制策略流程如图 3-13 所示。

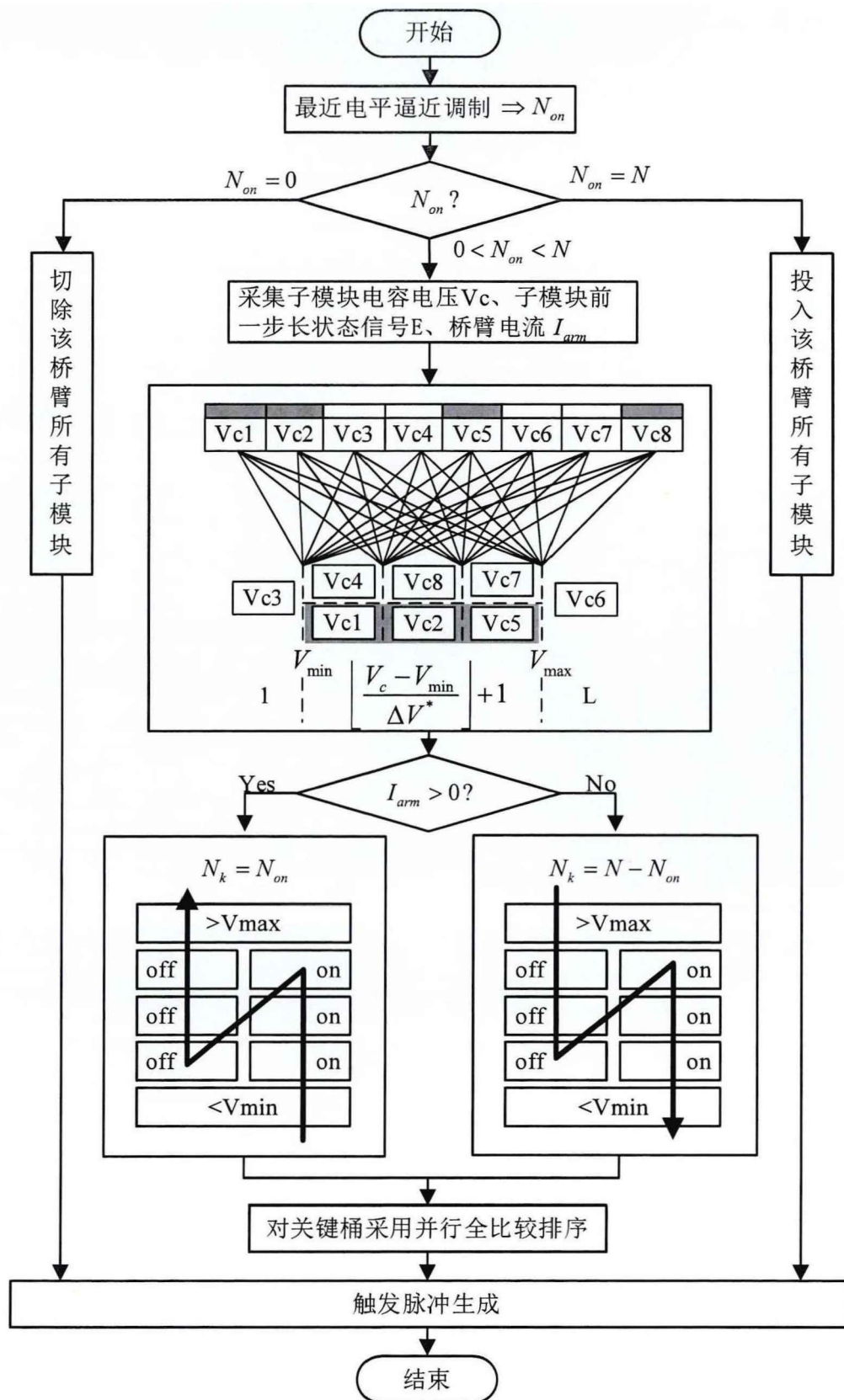


图 3-13 基于新型排序算法的均压控制策略

3.4 仿真验证

为验证所提均压策略的有效性，本文在 MATLAB/Simulink 平台上搭建了一个如图 3-14 所示的双端的模块化多电平换流器仿真模型，其中 MMC1 采用定直流电压和无功功率控制方式，MMC2 采用定有功功率和无功功率控制方式，并采用文献[75]所提的通用环流抑制策略，参数详细情况如表 3-3 所示。该仿真模型可以对所提均压策略进行全面评估，以验证其在实际应用场景中的性能和可行性。

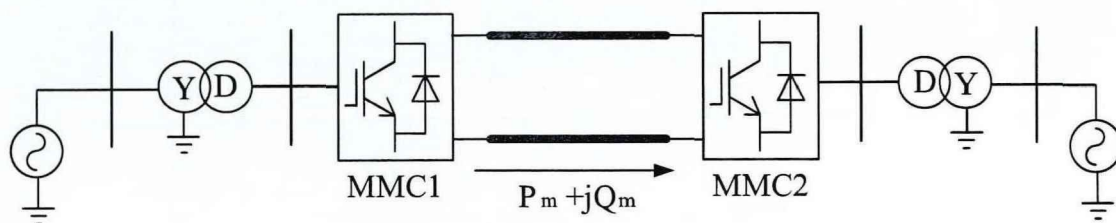


图 3-14 双端 MMC-HVDC 测试模型

表 3-3 仿真系统详细参数

模型参数	参数值
交流侧额定电压/kV	525
桥臂电感/H	0.14
单桥臂子模块个数	397
子模块电容/F	0.008
额定有功功率/MW	1250
额定无功功率/Mvar	0
直流侧额定电压/kV	±420
子模块额定电压/kV	2.1

为了评估所提策略的均压与降频性能，利用电容电压不平衡度与子模块平均投切频率作为算法效果的评价指标^[76]。

1) 电容电压不平衡度：在同一时刻下，桥臂子模块电容电压的最大值与最小值之间的差值与子模块额定电压的百分比。

$$\sigma = \frac{\max_{k \in [1, N]} [u_{C,k}] - \min_{k \in [1, N]} [u_{C,k}]}{V_{CN}} \times 100\% \quad (3-17)$$

式中， $u_{C,k}$ 为该时刻第 k 个子模块的电容电压值； V_{CN} 为子模块额定电压。

2) 子模块平均投切频率：在单位时间内，同一桥臂所有子模块投切状态改变的总次数与子模块的数量之比。

$$f_{av} = \frac{1}{2N} \sum_{k=1}^{2N} n_{sw,k} \quad (3-18)$$

式中， $n_{sw,k}$ 为单位时间内第 k 个子模块的状态切换总次数，投入和切除各算一次； N 为子模块数个数。

3.4.1 稳态实验

新型排序均压算法的具体实施流程如图 3-13 所示，在子模块总数 N 为 397 的条件下，应用式 (3-6) 计算得到最优分组数 $L^* = 9$ 。在上述搭建的 MMC-HVDC 双端系统上分别实现该新型排序均压算法与传统冒泡排序均压算法，稳态实验主要以 2 号 MMC 换流器中的数据进行分析。

A 相上桥臂的桥臂电流如图 3-15 所示，两种算法下的桥臂电流基本一致，呈现正弦波，未见明显畸变。这表明本文提出的均压策略虽在一定程度上降低了子模块电容电压一致性要求，但并不会影响环流抑制策略的有效性。

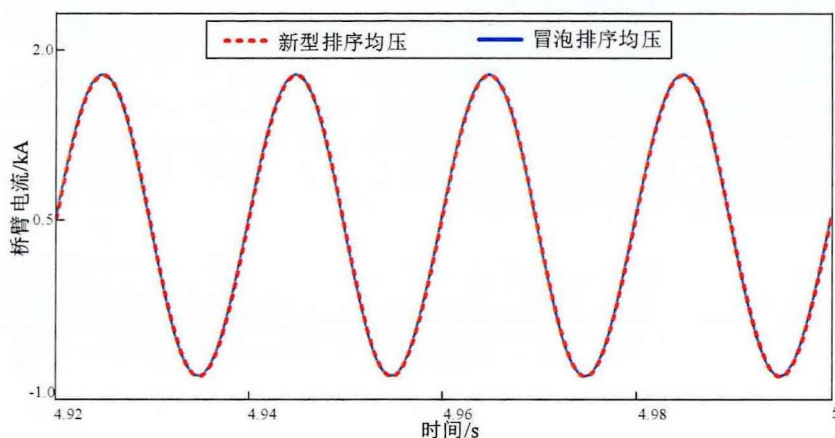


图 3-15 A 相上桥臂的桥臂电流

在 MM1 系统 A 相上桥臂中随机选取 32 个子模块，其电容电压波动曲线如图 3-16 所示。可以看到新型排序均压算法与冒泡排序均压算所得到的电容电压波动基本一致。

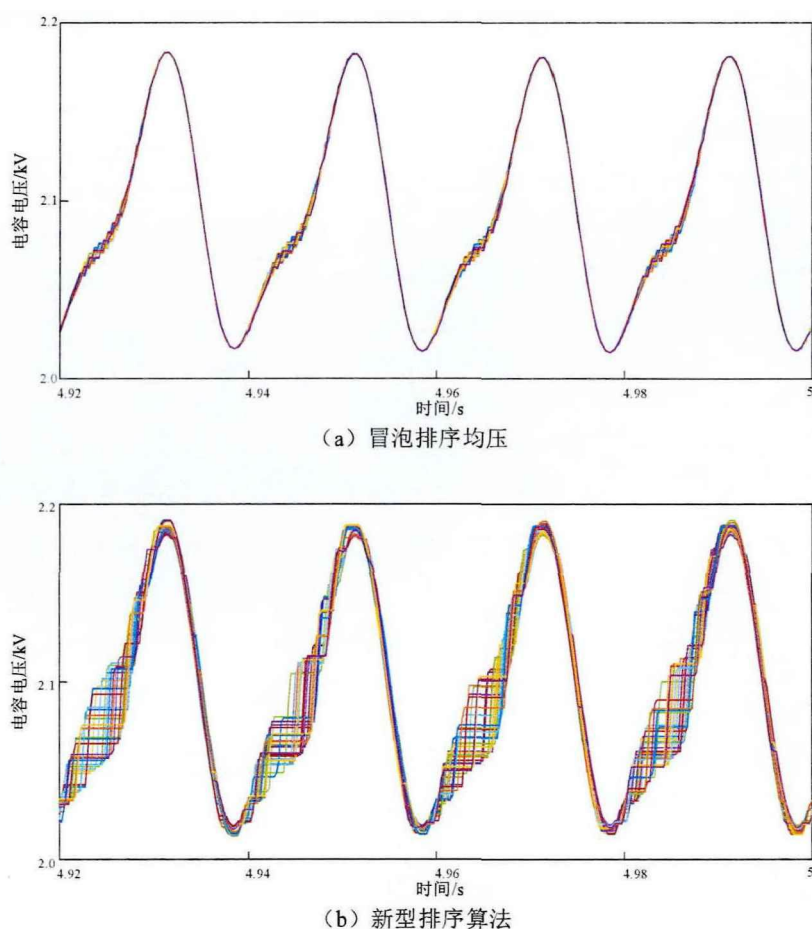


图 3-16 电容电压波动曲线

为进一步评价均压效果，根据式（3-17）可计算电容电压不平衡度，如图 3-17 所示。在稳态运行条件下，电容电压不平衡度在 0.5%至 4.5%的范围内波动，特别是当电容电压位于中间分组时，其波动性最大。这一现象与本研究提出的均压策略一致。经过均压策略处理，电容电压的不平衡度被有效控制在 5%以内，证明了该策略能够在不显著降低电容电压一致性的前提下，维持稳定性。

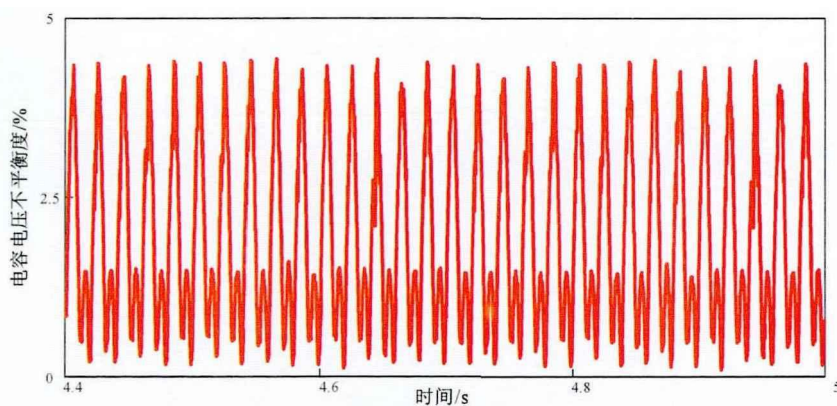


图 3-17 所提策略的电容电压不平衡度

第 16 个子模块 T_1 触发脉冲如图 3-18 所示。通过对比可见，在本文均压策略控制下，触发脉冲密度明显低于传统排序均压策略控制下的触发脉冲密度。为进一步说明，根据式（3-18）可计算出新型排序均压策略下的平均开关频率为 542Hz，而采用冒泡排序均压策略的平均开关频率为 1923Hz。可见，本文所提的基于新型排序的均压策略能显著减低子模块功率器件的平均开关频率。

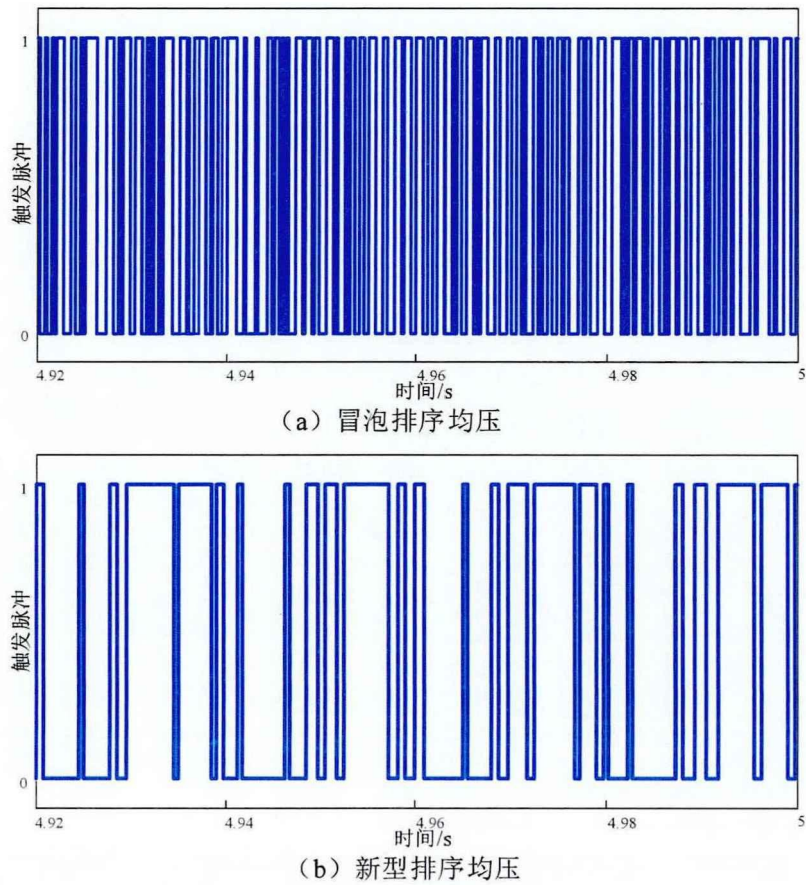


图 3-18 第 16 个子模块触发 T1 脉冲

3.4.2 暂态实验

为测试所提均压算法在暂态过程的均压效果，进行充电启动实验。采用文献[77]中的两阶段充电策略控制 MMC1 的充电启动过程，同时施加本文所提均压算法，验证其在充电启动过程中的适用性。充电策略控制参数见表 3-4。

表 3-4 充电策略控制参数

参数	参数值	描述
直流电压参考值/kV	600 800 (500kV/s)	0.5s 时切换
限流电阻/ Ω	500	串联在阀侧，0.5s 时退出
无功功率参考值/Mvar	0	—

1 号 MMC 换流器的直流母线实际电压与参考值如图 3-19 所示。

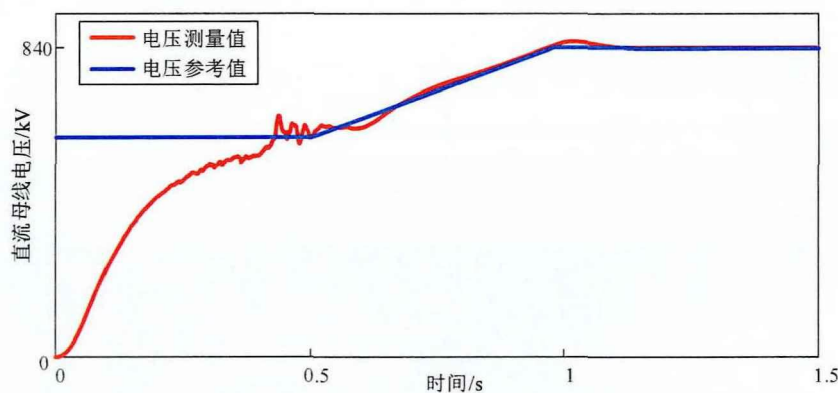


图 3-19 直流侧母线电压

图 3-20 为充电启动过程子模块电容电压的变化曲线。在换流器闭锁阶段，所有子模块电容电压平稳上升，且电压值基本一致；在换流器闭锁状态切换至解锁瞬间，电容电压逐渐发散，但波动幅度较小。随着电压参考值上升，子模块电压也逐渐上升至正常工作电压，证明了该策略在暂态工况下也能实现良好的均压效果。

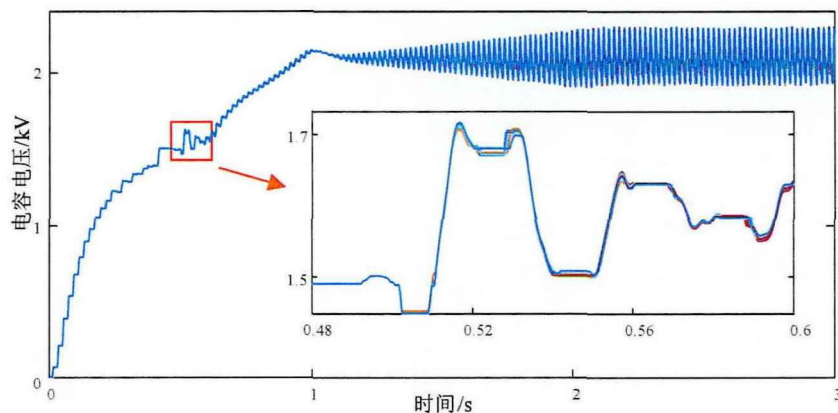


图 3-20 充电启动过程

与传统排序均压策略相比，本文提出的均压策略能够在保证较高电容电压一致性的前提下，显著降低子模块功率器件的开关频率，从而降低整个换流器的功率损耗。不论在稳态过程，还是在暂态过程，本文所提均压策略都能有效均衡各子模块电容电压，并具有较快的响应速度。

3.5 小结

本章深入探讨了针对 MMC 的新型均压控制策略，通过理论分析和仿真实验，证实了该策略在提高电容电压控制精度和系统响应速度方面的有效性。

(1) 本章针对只需确定序列极值的特定场景，提出了一种融合桶排序分布

特性和并行全比较排序计算效率的极值排序算法。通过与传统排序算法的比较，证实了新型算法在时间和空间复杂度上具有近似线性的显著优势。

(2) 探究了最优分组数与数据规模之间的关系。通过引入曲线拟合技术，结合对数模型和就近取整方法进行了精确拟合，得到了一个广泛适用的公式，该公式具有小于 1 的均方根误差，确保了预测的高可靠性。

(3) 提出了一种电容电压均衡控制策略，该策略通过设定阈值和子模块投切状态机制，在保持均压效果的同时降低了开关频率，减少了功率损耗。实验结果表明该策略在稳态和暂态条件下均能快速有效地实现电容电压平衡控制。

鉴于该均压策略高度并行性，本文将对其硬件设计与实现展开探究，具体内容将在下一章进行介绍。

第 4 章 均压算法在 FPGA 上的实现与验证

4.1 引言

随着柔性直流输电技术的不断发展，对于均压策略的实时性和计算速度提出了更高的要求。使用传统通用处理器实现均压策略时，其计算速度常受到逻辑处理器数量的限制，难以充分发挥策略的高度并行性。与传统通用处理器平台相比，FPGA 凭借其体积小、成本低、响应速度快等优势。使用 FPGA 实现均压策略，可以依靠 FPGA 的特性来发挥第 3 章所提均压策略的优势。

本章根据第 3 章所提均压策略，以 FPGA 为基础进行设计和实现。首先，对该均压策略的整体架构进行设计，包括电压缓存模块、极值搜索模块、触发信号生成模块。接着，利用 Verilog HDL 语言分别对这些功能模块进行独立设计与实现。随后，对各个模块进行集成和连接，以实现该策略的均压功能，并采用基于 FPGA 的 Vivado 软件对均压算法的性能进行了测试和分析。最后，基于以太网通信以及 FPGA 硬件平台，进行了均压策略的仿真测试，验证最终的仿真结果，证实了第 3 章中提出的均压策略在硬件条件下的可行性。

4.2 约束设计与数据传输优化

4.2.1 设计选择

硬件描述语言为 FPGA 提供了一种灵活、高效且精确的设计和实现方法，是现代电子设计自动化（Electronic Design Automation, EDA）流程中不可或缺的一部分。Verilog HDL 与 VHDL 均为硬件描述语言，广泛应用于电子系统的描述、设计与验证过程中，尤其在数字电路领域中具有重要作用。VHDL 的语法基于 Ada 语言，更加严格和形式化。这种严格性有助于避免设计中的模糊性，但同时也使得学习更加困难，编写的代码更加复杂。与 VHDL 不同，Verilog 的简洁语法与 C 语言相似，编写和理解硬件描述更为直观，这为具有开发者提供了更低的学习门槛和更短的设计周期。这种简洁性不仅加快了设计的初步开发和迭代过程，而且有助于提高整体开发效率。因此，本文设计选取 Verilog HDL 语言进行编程。

在 FPGA 硬件实现开发中，选择合适的芯片类型也是设计中一项很重要的工作，是整个设计的基础。Xilinx 作为 FPGA 技术的先驱之一，提供了强大的开发工具和资源，包括 Vivado Design Suite 和 Vitis 平台，这些工具支持高效的

设计流程、仿真和调试。Xilinx 的生态系统还包括大量的知识产权核 (Intellectual Property Core, IP Core)、参考设计和开发板，有助于加速设计周期和降低开发成本。因此，在本设计中采用 Xilinx 厂商提供的产品。Xilinx 的主要 FPGA 系列包括低端的 Artix 系列、中端的 Kintex 系列以及高端的 Virtex 系列。其中，Virtex-7 系列芯片不仅提供了大量的逻辑单元、DSP 和内存资源，能够实现复杂的数字逻辑和算法处理，而且支持多种高速串行通信协议，如 PCI Express、Gigabit Ethernet 和 SATA 等，以及高速 I/O 接口，适用于高数据吞吐量的应用，其性能特征表如表 4-1 所示。

表 4-1 Virtex-7 系列性能特征表

芯片型号	XC7VX330T	XC7VX415T	XC7VX485T	XC7VX1140T
系统门数	51000	64400	75900	178000
逻辑单元	326400	412160	485760	1139200
可配置逻辑块	408000	515200	607200	1424000
I/O 引脚	700	600	700	1100
DSP Slice	1120	2160	2800	3360
内存/kB	27000	31680	37080	67680

XC7VX485T 提供的逻辑资源、DSP 单元数量、内存容量及 I/O 数量，与本设计的具体需求更为匹配。在满足设计需求的同时，避免了资源的过度配置，提供了更好的成本效益比。鉴于此，在本设计中选用 Xilinx 公司的 XC7VX485T 芯片。基于该芯片的优异性能，本文采用 VC707 开发板作为实验平台，实际电路板如图 4-1 所示。相比单芯片，VC707 开发板提供丰富硬件接口和成熟测试环境，使得开发者更加专注于核心功能的实现和优化，而无需投入大量时间于基础硬件的搭建和调试，极大简化了开发流程。

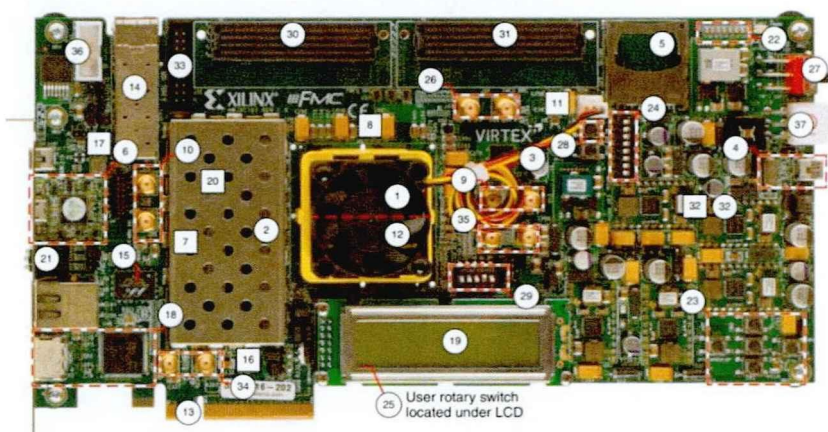


图 4-1 VC707 开发板

4.2.2 约束设计

在集成电路设计和现场可编程门阵列开发中，确保设计满足特定的物理和时序要求是至关重要的。设计约束作为设计过程中的一个关键组成部分，为实现设计在硬件上的正确性和性能优化提供了必要的指导。本节主要探讨了FPGA设计中涉及的设计约束，包括时钟定义、输入/输出（I/O）标准设置、高速接口布局以及时序约束等方面。

（1）时钟定义和约束

时钟是FPGA设计中的基础，正确定义时钟及其约束对于确保设计的时序正确性至关重要。通过create_clock命令定义了系统时钟User_SYS_CLK和GT参考时钟GT_REFCLK1，并分别指定了它们的周期为5.0ns和4.0ns，如表4-2所示。这一步骤确保了时钟信号的准确性和稳定性。

表 4-2 时钟约束设置

时钟名称	周期/ns	端口	描述
User_SYS_CLK	5.0	sys_clk_p	系统主时钟约束
GT_REFCLK1	4.0	GTXQ0_P	GT 参考时钟约束

（2）输入/输出（I/O）约束

为了保证信号的完整性和减少电磁干扰，对FPGA设计中的I/O端口进行了精确的物理位置和电气标准设置。通过set_property命令，分别为系统时钟、复位信号、管理数据输入/输出（MDIO）、管理数据时钟（MDC）、中断信号（INT）以及重置信号等端口指定了具体的物理位置（LOC）和I/O标准（如LVDS、LVCMOS18），如表4-3所示。这些设置有助于确保设计在实际硬件上能够按照预期工作。

表 4-3 I/O 约束设置

端口	LOC	IOSTANDARD	描述
sys_clk_p	E19	LVDS	系统时钟正端
sys_clk_n	E18	LVDS	系统时钟负端
rst	AV40	LVC MOS18	复位信号
mdio	AK33	LVC MOS18	管理数据输入/输出
mdc	AH31	LVC MOS18	管理数据时钟
int	AL31	LVC MOS18	中断信号
reset	AJ33	LVC MOS18	重置信号

(3) 高速接口布局约束

在处理高速串行通信接口时，对接收（RX）和发送（TX）通道进行了物理位置的精确指定，并为 SFP 模块的 TX_DISABLE 信号设置了 I/O 标准，如表 4-4 所示。这样的约束不仅对于保持信号的完整性至关重要，也对于优化设计的电磁兼容性（EMC）和减少串扰有着显著效果。

表 4-4 高速接口布局约束设置

端口	LOC	描述
GTXQ 组	A9、A10	GT 通道的负端 与 GT 通道正端
RXP 组	A6、B8、C6、D8	接收通道正端的 第 1 组到第 4 组
RXN 组	A5、B7、C5、D7	接收通道负端的 第 1 组到第 4 组
TXP 组	B4、C2、D4、E2	发送通道正端的 第 1 组到第 4 组
TXN 组	B3、C1、D3、E1	发送通道负端的 第 1 组到第 4 组
SFP_TX_DISABLE_N 组	L31、G39、N38、J42 、U31、T30、J30、P28	第 1 个到第 8 个 SFP 模 块的 TX_DISABLE 信号

(4) 时序约束

为了进一步优化设计的性能并确保时序正确性，本设计利用了一系列的时序约束指令，如 set_false_path，来指定时序分析中应该忽略的路径。时序约束主要包括了 set_false_path 和 set_max_delay，用于控制数据跨时钟域的延迟和抖动，以及处理特定的时序路径，以确保设计的时序正确性。

总之，设计约束的合理设置对于确保 FPGA 设计的成功非常关键。它不仅影响着设计的物理实现，还直接关联到设计的性能和可靠性。通过精确的时钟定义、I/O 约束设置、高速接口布局以及时序约束的应用，可以显著提高设计在实际硬件上的表现，确保设计的最终成功。

4.2.3 数据转存机制

在模块设计中，数据转存环节起着至关重要的作用，它确保了处理完毕的数据能够安全、有效地传递至模块的下一级。本节详细讨论了数据转存环节的实现机制及其在整个数据处理流程中的重要性。

数据转存环节主要涉及两个寄存器：is_r 和 valid_r。其中，is_r 寄存器用于暂存处理完毕的结果数据，而 valid_r 寄存器则标识这些数据的有效性。在初始时刻，is_r 和 valid_r 都被初始化为 0，表示没有有效的数据存储在 is_r 中。每当时钟信号 clk 的上升沿到达，就会执行以下操作：如果复位信号 rst 为高（即处于复位状态），则 is_r 被重置为 0，valid_r 被重置为 0，表示寄存器中没有有效的数据；当 valid1 信号为高时，表明有新的有效数据待转存，此时 is_r 将更新为最新的结果数据 result1，同时 valid_r 置为 1，标识数据为有效状态；若 valid1 信号未激活，则 valid_r 重置为 0，表示当前无有效数据传递，is_r 保持当前值不变。最终模块的输出 valid 为 valid_r 的当前值，输出 is 为 is_r 的当前值。数据转存环节伪代码如表 4-5 所示。

表 4-5 数据转存环节伪代码

数据转存实现步骤
输入：复位信号 rst，时钟信号 clk，计算结果 result1 初始化：定义一个 32 位的变量 is_r，初始值设为 0；定义一个布尔变量 valid_r，初始值为假（false） FOR 每当时钟信号 clk 上升沿到达 IF 复位信号被激活（即 rst 为真）： 将 is_r 设置为 0 将 valid_r 设置为假（表示没有有效数据） ELSE: IF valid1 信号为真（表示有有效的输入数据 result1）： 将 is_r 更新为 result1 的值 将 valid_r 设置为真（表示 is_r 中的数据现在是有效的） ELSE: 保持 is_r 不变 将 valid_r 设置为假（表示 is_r 中的数据不再有效） END FOR 输出：将 valid_r 赋值给数据使能信号 valid，将 is_r 赋值给数据信号 is

通过这种机制，模块能够在数据处理完成后，及时将结果数据及其有效性状态传递给后续处理流程，而无需外部干预。该机制特别适用于那些不会在每个时钟周期产生有效输出的算法场景。外部系统借助 valid_r 标志即可识别数据的有效性，避免无效数据的误用。该内置有效性指示简化了模块间的接口设计，提升了交互的效率。此外，将该代码段置于算法末端，可以保证在数据输出前，所有数据处理和逻辑判断均已完毕。这样做有助于确保数据的完整性和时效性，

同时降低了因为错误或过早读取数据而引发的风险。

综上所述，数据转存环节是整个数据处理模块中不可或缺的一环，它不仅确保了数据的安全传输和有效性控制，也为模块间的高效通信提供了保障。通过所设计的数据转存机制，本模块能够在保证高效率的同时，也确保了数据处理的准确性和系统的鲁棒性。因此，在每个模块末尾都应该添加数据转存部分。

4.2.4 启动延时机制

在数字电路和系统设计中，启动延时是一种常见的技术，用于在系统上电后引入一个预定时间的延迟，以确保系统稳定或等待外部条件成立。与数据转存相似，启动延时的处理主要集中在 valid 信号的生成上，目的是实现一个简单的计时功能。通过持续计数直到达到一个预设的阈值，之后激活 valid 信号。

板子上电后到上位机开始操作并实时采集数据之间需要一定的时间，而直接通过上位机进行复位操作可能引入不可预测的错误。此外，电源启动后，为确保电源稳定，通常也需要等待一段时间。因此，在本设计中，将系统的启动延迟设置为 20 秒，以确保系统稳定运行。启动延时的实现采用了一个 32 位的计数器 cnt_initial 和一个单比特的寄存器 signal_initial_r。这两个组件协同工作，通过以下步骤实现启动延时。

表 4-6 启动延时环节的伪代码

启动延时实现步骤
输入： 复位信号 rst，时钟信号 clk
初始化： 定义一个 32 位十进制的寄存器 cnt_initial，并将初始值设为 0；定义一个布尔变量 signal_initial_r，并将初始值设为假（false）
FOR 每当时钟信号 clk 上升沿到达
IF 复位信号被激活（即 rst 为真）：
将 cnt_initial 设置为 0，将 signal_initial_r 设置为假
ELSE
IF cnt_initial 的值小于 2,000,000,000：
将 cnt_initial 与 1 相加，然后将结果重新赋值给 cnt_initial
保持 signal_initial_r 为假
ELSE:
保持 cnt_initial 不变，将 signal_initial_r 设置为真（表示延时完成）
END FOR
输出： 将 signal_initial_r 的值赋给新的系统使能信号 signal_initial

1) 初始化和复位处理: 在时钟信号 `clk` 的上升沿到来时, 如果复位信号 `rst` 为高, 则计数器 `cnt_initial` 和信号寄存器 `signal_initial_r` 均被重置为初始状态。这确保了在系统需要重新启动时, 延时逻辑也能从初始状态开始。

2) 计数和延时实现: 在复位信号 `rst` 为假时, 系统开始计数。如果 `cnt_initial` 的值小于 2000000000 (对应于 20 秒的延时, 因时钟频率采用 100Mhz), 计数器的值递增, 并且 `signal_initial_r` 保持为 0, 表示定时过程尚未完成。此阶段, 系统处于等待状态, 不执行正常操作。

3) 延时完成与信号激活: 当计数器的值达到或超过 2000000000 时, `signal_initial_r` 被设置为 1, 此时表示定时完成。此信号的激活标志着系统可以开始执行后续操作。

4) 信号输出: `signal_initial_r` 的状态通过 `signal_initial` 输出, 供系统的其他模块使用。这样, 后续模块就可以根据 `signal_initial` 的状态判断系统是否已完成启动延时, 从而在适当的时间开始其操作。

这一机制确保了系统在上电后引入一个固定的启动延时, 为系统的稳定运行提供了额外的保障。它对于处理系统上电初期可能遇到的不稳定因素至关重要, 有助于确保系统能够在预定的、稳定的状态下开始执行其主要功能。

4.3 均压策略的 FPGA 实现

针对第三章所提均压策略, 本章基于硬件实现的逻辑进行了改进。在基于 FPGA 实现均压策略时, 主要分为三大部分: 电压缓存模块、极值搜索模块以及触发信号生成模块。电压缓存模块主要负责将以数据流形式传来的子模块电容电压数据进行预处理并进行缓存, 确保数据完整且易于被后续环节处理。极值搜索模块是本策略的核心实现环节, 通过高度并行的排序算法与查找表, 充分利用了 FPGA 的高度并行能力, 缩短了排序时间。触发信号生成模块是该策略的最终实施步骤, 利用子模块电压经过预处理后自带的位置信息作为索引生成相应的触发信号, 以调节相应的子模块来实现电压的均衡。

4.3.1 电压缓存模块

在计算机工程领域, 数值的表征与日常生活中的直观概念存在显著差异, 尤其在进行基于 FPGA 的设计时, 选择合适的数值表示方法成为设计优化的关键一环。数值表示法主要分为浮点数和定点数两种类型。浮点数动态范围更大, 能够表示非常大和非常小的数值, 适用于范围和精度需求变化很大的场合。然而, 浮点运算在硬件实现上由于其复杂性, 往往伴随较高的功耗和资源消耗。

与之对比，定点数表示法以其运算的简洁性和高效性，更加适用于对延迟和资源使用敏感的实时系统应用。由于电容电压波动范围不大，因此，本文选用定点数表示法来实现均压算法。

定点数通过固定长度的二进制格式预定整数位和小数位，以适应特定应用场景的需求。本研究面向基于 FPGA 的均压算法实现，对于模块电容电压的处理提出了在电容电压表示中附加位置信息和投切状态信息的方案。此策略基于以下考量：首先，通过位置信息的加入，能够直接利用排序结果产生触发信号，避免引入额外的位置标识信号，增强数据的可追踪性和处理效率；其次，考虑到并行全比较算法存在数值相同导致排序失败的潜在缺陷，位置信息的引入可作为辅助判断依据；最后，前一时刻投切状态的引入可以减少子模块开关频率。

定点数表示中位置信息的位宽选择依据子模块总数进行动态调整。例如，当子模块数量不超过 512 时，位置信息位宽设为 9 位；不超过 256 时，设为 8 位；以此类推，直至满足实际应用需求。通过第 3 章的仿真实验，可以知道子模块电容电压一般维持在约 2kV 的水平，实际工程中也是如此。由于技术的限制，目前市场上的国产 IGBT 产品耐压等级通常约为 1kV，而国际产品的耐压能力更高，一般可达 3kV。同时，对于超过此范围的应用，更高耐压级别的 IGBT 正在研发之中，预计最高耐压能力将达到 6kV。因此，本文选择 3 位整数位。同时，为了保持数据位宽一致，小数位数根据整数位和位置信息位的总和进行调整。针对 397 个子模块的排序情况，本研究设计的定点数格式为 3 位整数位，19 位小数位，9 位位置信息以及 1 位前一时刻投切状态信息。例如，第 32 个子模块的电容电压（值为 2.267kV）采用附加位置信息的电容电压定点数表示形式如图 4-2 所示。

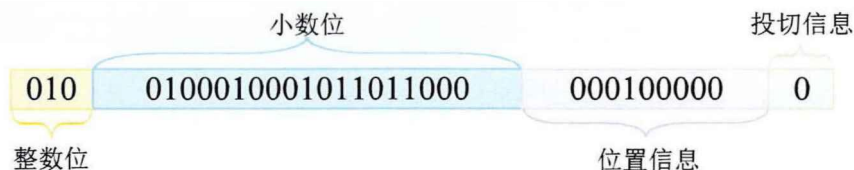


图 4-2 附加位置信息的电容电压定点数表示形式

通过上述步骤可以将电容电压数据统一至 32 位，显著提升了系统的通用性和灵活性。通过这种方式，无论子模块的具体数量如何变化，后续处理步骤可以无缝接收和处理这些 32 位的数据，无需对数据位宽进行任何调整。

在 MMC 的实时仿真中，子模块电容电压的计算结果需要实时传输至阀控，面对大量数据处理需求，本文引入缓存机制以临时存储并序列化数据。采用先入先出（First In First Out, FIFO）寄存器的缓存策略，具体流程如下：首先，

数据流进入 FPGA 后立即进行浮点到固定点的转换；其次，转换后的数据被送往位置信息拼接单元；最后，拼接好的数据存入 FIFO 等待后续处理。

(1) 固定点转换模块

为优化数据流的处理效率，引入了数据预处理环节，该单元负责将数据从浮点格式实时转换为固定点格式，支持多数据流的并行处理。采用 Floating-point IP 核实现 Float-to fixed 的功能，根据输入数据设置浮点数数位，一般是 IEEE 754 标准的单精度 32 位浮点数。根据上述分析选择 4 位整数位以及 19 位小数位，具体设置如图 4-3 所示。

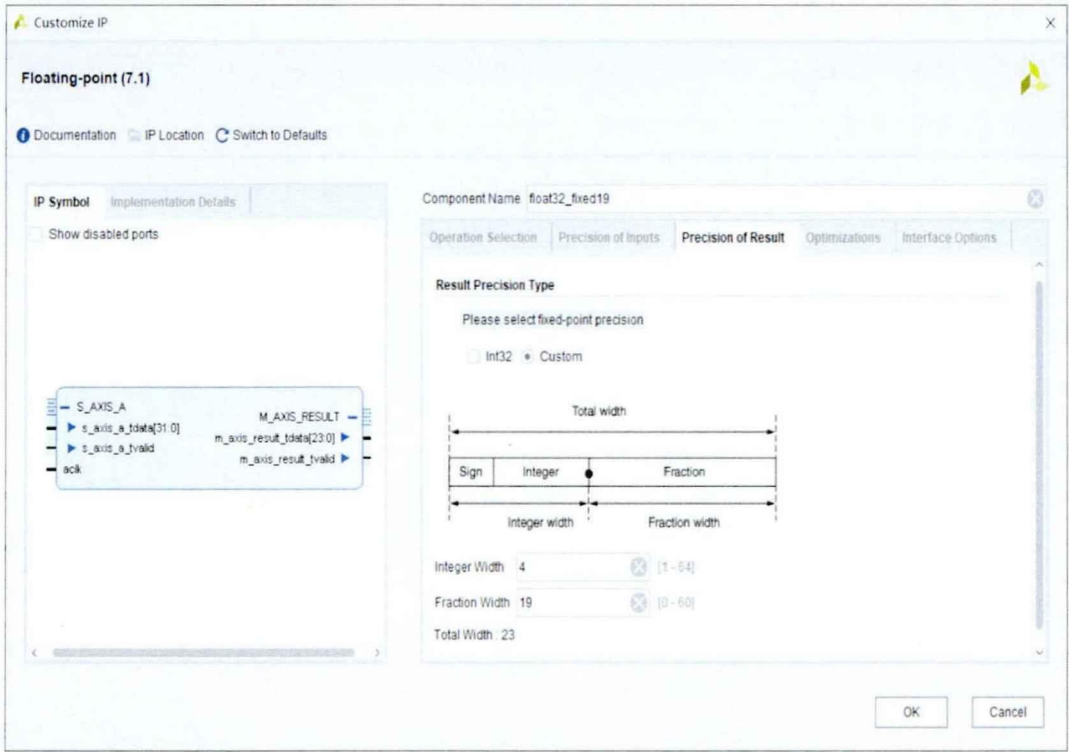


图 4-3 浮点数到定点数转换模块

(2) 信息拼接模块

该模块为进行位置信息的拼接，目的是为每个数据点即时分配明确的位置信息。这一步骤对于数据的后续排序和检索至关重要。具体而言，该模块接受一个 22 位的电容电压值作为输入，并且每个输入值都伴随着一个有效信号，指示当前的电容电压值已准备好被处理。该模块的输出是一个 32 位的数据字，其中包括原始的 22 位电容电压值及其对应的 9 位位置信息、1 位投切信息。位置信息是通过内部的 9 位计数器生成的，该计数器在每次处理一个电容电压值后递增，并在达到 396 后循环回 0，以匹配系统中总共 397 个电容的配置，该模块的伪代码如表 4-7 所示。该设计通过将电容电压值与其物理位置在芯片上的对应关系明确编码，有效地提高了数据的可管理性和后续处理的灵活性。

表 4-7 信息拼接的伪代码

信息拼接的实现

输入：时钟信号 clk，复位信号 rst，22 位电容电压值 vc，电容电压有效信号

vc_valid，投切信号 on

初始化：9 位寄存器 cnt，并将初始值设为 0；布尔变量 data_ready，初值为假

FOR 每当时钟信号 clk 上升沿到达

IF 复位信号被激活（即 rst 为真）：

将 cnt 设置为 0，将 data_ready 设置为假

ELSE：

IF 电容电压有效（即 voltage_valid 为真）：

拼接电容电压值和位置信息、投切信号到 uc (uc<= {vc, cnt, on};)

将 data_ready 设置为真（表示标记数据已准备）

IF cnt 等于 396（已读取完 397 个数）：

将 cnt 设置为 1（位置信息复位）

ELSE：

将 cnt 与 1 相加，然后将结果重新赋值给 cnt（更新位置信息）

ELSE 电容电压无效（即 voltage_valid 为假）：

将 data_ready 设置为假（无新数据时，数据未准备好）

输出：32 位包含电容电压值和位置信息的 uc，电容电压使能信息 data_ready

(3) FIFO 设计

FIFO 的深度对系统延迟和吞吐量有直接影响。一个较深的 FIFO 能够更好地应对突发的数据流高峰，但同时可能增加数据在队列中的等待时间。因此，深度设置需在满足实时性要求和处理高峰期数据流的前提下进行优化。根据本设计的应用场景，设置深度为 512，既能满足数据量的需求，又能避免资源的过度消耗，具体配置如图 4-4 所示。

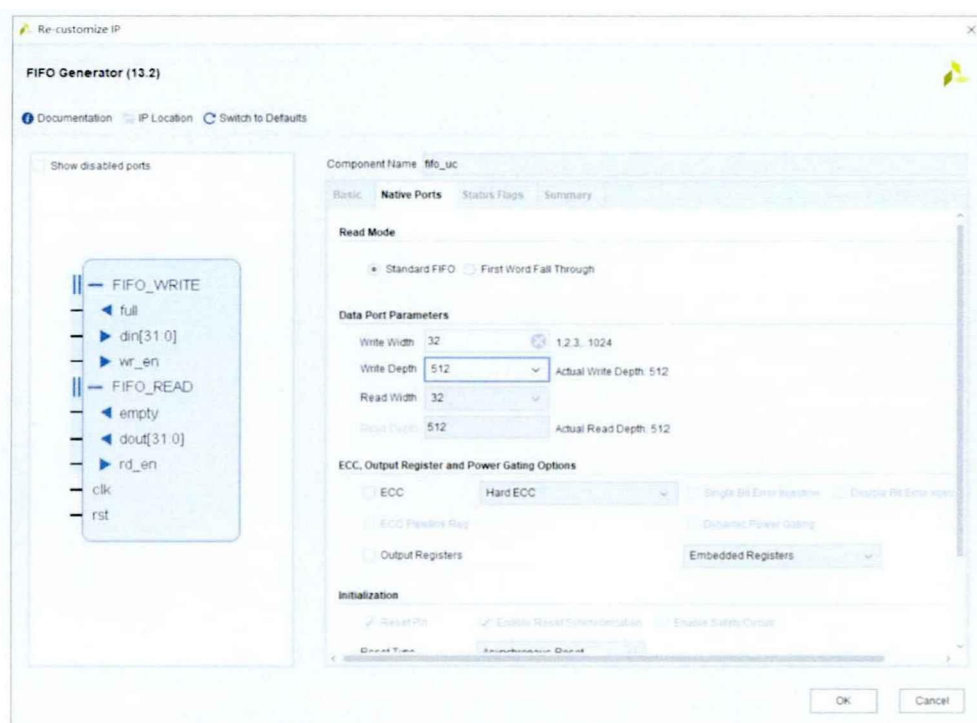


图 4-4 FIFO 缓存模块

4.3.2 极值搜索模块

(1) 电容电压的分组转存与累计

查找表 (Lookup Table, LUT) 是一种用于存储数据或计算结果的方法, 以便于快速检索。在硬件设计中, 查找表是实现逻辑函数的基本单元, 能够根据输入直接给出输出, 避免了复杂的计算过程。查找表可以根据自定义的比较器逻辑来实现分组的功能。在针对子模块电容电压排序的应用场景中, 可以通过预设定的边界值 (本设计根据最优分组数和电压阈值, 设计的边界值为[2.01, 2.03, 2.05, 2.07, 2.09, 2.11, 2.13, 2.15, 2.17, 2.19]) 来创建查找表, 然后使用这个查找表来快速决定任意输入值 (比如 2.267) 应该分配到哪个组中。这种方法非常适合于需要快速分类或分组输入数据的场景。

为了在记录每个组数量的同时, 将数据存储在与相应分组下的寄存器中, 以便之后的排序工作, 需要在原有的设计基础上增加数据存储的逻辑。在对 397 个子模块进行 9 分组排序的情况下, 每组约含 40 个 32 位数据元。此数据量相对较小, 故采用寄存器数组而非内部 RAM 作为更合理的存储方案。尽管寄存器占用较多逻辑资源, 但考虑到仅 40 个数据元的规模, 在大多数 FPGA 设计中是可接受的。更重要的是, 寄存器能提供快速数据访问, 对高速处理应用尤为重要。因此, 使用寄存器数组实现了资源与性能之间的有效平衡, 满足了高效处理的需求。

为确保在构建 group_cnt 时数据逻辑的清晰和简化逻辑判断，采取了分阶段方法。group_cnt 包含 20 个元素，分别代表 1 个 DOWN 组的计数、9 个 ON 组的小分组计数、9 个 OFF 组的小分组计数，以及 1 个 UP 组的计数。此策略通过减少需要跟踪的变量数量，有效地简化了逻辑结构。表 4-8 为基于查找表的数据分组与转存处理过程的伪代码。

表 4-8 基于查找表的数据分组与转存的伪代码

基于查找表的数据分组与转存实现

输入：clk, rst, data_in (32 位定点数输入数据), valid_in (有效数据标志)

参数定义：边界值数组 BOUNDARIES[0...8]，包含 9 个定点数边界值

FOR 每当接收到新数据 (data_in) 时：

IF reset 信号被触发：

 重置所有计数器和存储数组

ELSE, IF valid_in 信号为真：

IF data_in 小于 BOUNDARIES[0]：

 增加 down_cnt

 将 data_in 存储到 down_uc 中

ELSE, IF data_in 大于或等于 BOUNDARIES[9]：

 增加 up_cnt

 data_in 存储到 up_uc 中

ELSE：

FOR i 从 1 到 9：

IF data_in 小于 BOUNDARIES[i]：

IF data_in 的最低位为 0 (OFF 组)：

 增加 off_cnt[i-1]

 将 data_in 存储到 off_uc[i-1][off_cnt[i-1]-1]

ELSE (ON 组)：

 增加 on_cnt[i-1]

 将 data_in 存储到 on_uc[i-1][on_cnt[i-1]-1]

END FOR

输出数据：计数器序列 group_cnt，存储数组 down_uc on_uc off_uc up_uc

(2) 识别关键组

在关键组的识别过程中，采取了一种以计数为基础的策略，简化了操作流程。该过程中，使用单一计数变量 sum 来累计前 k+1 个分组的总计数，进而确

定所谓的关键组。当 sum 正好等于 n, 此时表明没有必要进行后续的全面比较, 因而引入 M_on 信号以指示进一步的排序操作可以省略。进一步地, 根据 Iarm 的符号 (正或负), 选择不同的统计起点和遍历方向, 以优化数据处理过程。这种方法增强了程序的逻辑性和可维护性, 伪代码如表 4-9 所示。

表 4-9 识别关键桶的伪代码

识别关键桶的实现
输入: 桥臂电流方向信息 Iarm, 子模块待投入个数 N_on, 组别计数器 group_cnt 初始化: 累计寄存器 sum 为 0, 关键组有效信号 k_valid 为 0(假), M_on 为 0 FOR 每当接收到分组过程结束时: IF Iarm 等于 1 (表示 Iarm>0, 遍历顺序: DOWN, ON, OFF, UP): FOR (i = 0; i <= 19; i = i + 1) 将 sum 和 group_cnt[i]相加, 并将新值赋值给 sum IF sum 大于等于 N_on: 设置 keyGroupIndex 为 i 设置 valid 为 1 (真) 将 sum 与 N_on 的差值赋值给 M_on ELSE (表示 Iarm<0, 遍历顺序: UP, OFF, ON, DOWN): FOR (i = 19; i >= 0; i = i - 1) 将 sum 和 group_cnt[i]相加, 并将新值赋值给 sum IF sum 大于等于 N_on: 设置 keyGroupIndex 为 i, 设置 valid 为 1 (真) 如果 sum 恰好等于 n, 则设置 sort_needed 为假 END FOR 输出: 关键组索引 keyGroupIndex, 关键组有效信号 k_valid, 排序个数 M_on

(3) 并行全比较排序

由于采用了寄存器数组存储数据的方法, 因此在排序阶段无需进行流式传输, 可以直接采用并行全比较排序。这一步骤基于每个子模块的得分进行排序, 并假设处理后的子模块电压不会出现得分相同的情况, 从而避免了排序冲突。以下是具体步骤简化说明:

- 1) 初始化: 如果系统处于重置状态, 首先将所有得分 score[] 设置为 0。
- 2) 得分计算: 通过双层循环遍历数据数组 data[]。对于每个元素 data[i], 比较其与数组中其他元素的大小, 若 data[i] 大于 data[j], 则 data[i] 的得分 score[i] 增加 1。这样计算后, 每个元素的得分反映了它大于数组中其他元素的总数。

3) 数据重新排序：根据得分重新赋值新数组。外层循环遍历所有得分，内层循环在得分数组 `score[]` 中寻找与当前得分相等的元素，并将这个元素赋值给新的排序后数组 `sorted_data[]`。即，当 `score[j]==i` 时，执行 `sorted_data[i]=data[j]`。

这种方法充分利用了并行处理的优势，有效加速了排序过程，而且通过预处理确保了得分的唯一性，避免了排序时的冲突

(4) 合并数据

根据 `M_on` 的值，选择 `sorted_data` 中前 `M_on` 个极大值或极小值，如果 `Iarm>0`，则选择极小值，若 `Iarm<0`，则选择极大值。这些数极大值或极小值与基础数组一起构成了新的子模块投切决策表，如表 4-10 所示。

表 4-10 `Iarm>0` 的子模块投切决策表

keyGroupIndex	基础数组	关键组的 <code>M_on</code> 个极小值
0	—	<code>down_uc</code>
1	<code>down_uc</code>	<code>on_uc</code>
2-10	<code>down_uc+ on_uc</code>	<code>off_uc</code>
11-19	<code>down_uc+ on_uc+ off_uc</code>	<code>up_uc</code>

4.3.3 触发信号生成模块

在控制系统的设计中，针对 397 个子模块的触发信号生成过程采用了一种基于寄存器数据解析的方法。触发脉冲信号的生成逻辑经过预处理后变得相对简单：只需识别寄存器中的第 23 位到第 31 位数据（`wire [8:0] index = down_uc[31:23]`），并将其作为索引赋值给相应的数组即可生成最终的触发信号（`gate[index] <= 1'b1`）。这些信号用一个深度为 397 位的 1 位数组 `gate` 表示，其中每一位代表一个子模块是否接收到触发信号。例如，若在基础组的寄存器中解析得到的索引值为 20，则将 `gate[20]` 设为 1，标志着第 20 个子模块接收到触发信号，即被投入。

为提高信号处理的准确性和效率，触发信号的生成分为两个阶段：首先，系统对基础组的子模块进行处理，`keyGroupIndex` 有效时，系统即可对基础组产生投入信号。在剩下的非关键组中，也可以同时生成切除信号。经过排序后，系统再对关键组的信号进行处理，针对关键组的 `M_on` 个极小的子模块进行生成触发脉冲的生成操作。这种分阶段的方法充分利用了 FPGA 的并行性，优化了触发信号的生成流程，并确保在复杂系统操作中，各子模块能够按照既定逻辑准确地接收到触发指令，从而提升了整个控制系统的响应速度和可靠性。

4.4 实时仿真实验

针对以上的设计,使用 Verilog HDL 编写所提均压策略的源代码,通过 Vivado 软件进行综合,并部署到 VC707 上。MMC 求解器和控制器通过 FPGA 内部连线连接,免除外部通信协议的需求。利用自主开发的硬核 TCP/IP 协议栈,实现了通过千兆以太网的实时数据传输,并通过上位机的 TCP Client 进行数据采集。数据处理和可视化则基于 MATLAB 完成。仿真平台如图 4-5 所示。

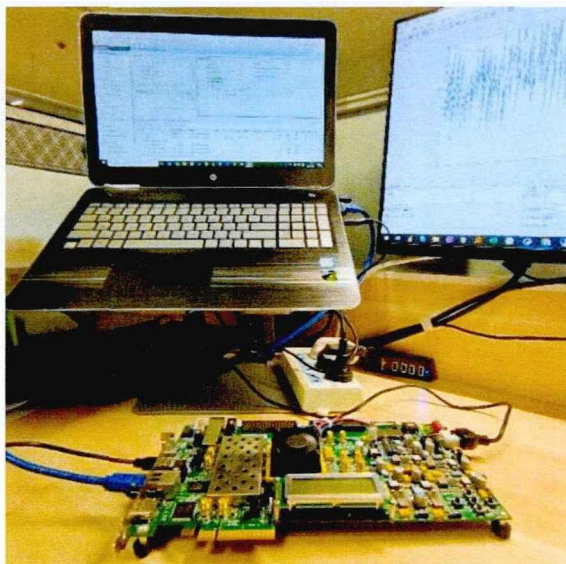
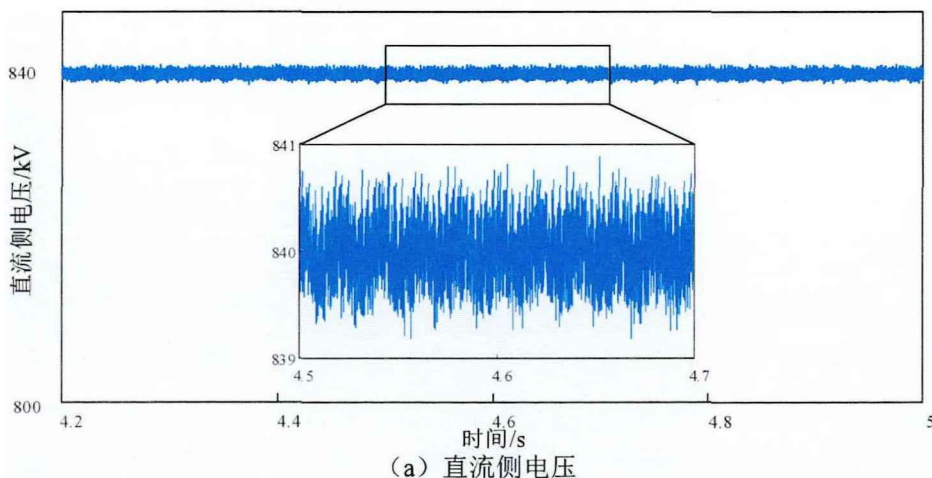


图 4-5 FPGA 实时仿真平台

在 MMC 求解器中搭建 397 子模块的双端 MMC 模型,具体数据参照 3.4 节参数设置,并进行稳态功率传输实验,基于 FPGA 实现的实时仿真结果如图 4-6 所示,验证了所提新型排序均压算法在 FPGA 的可行性。



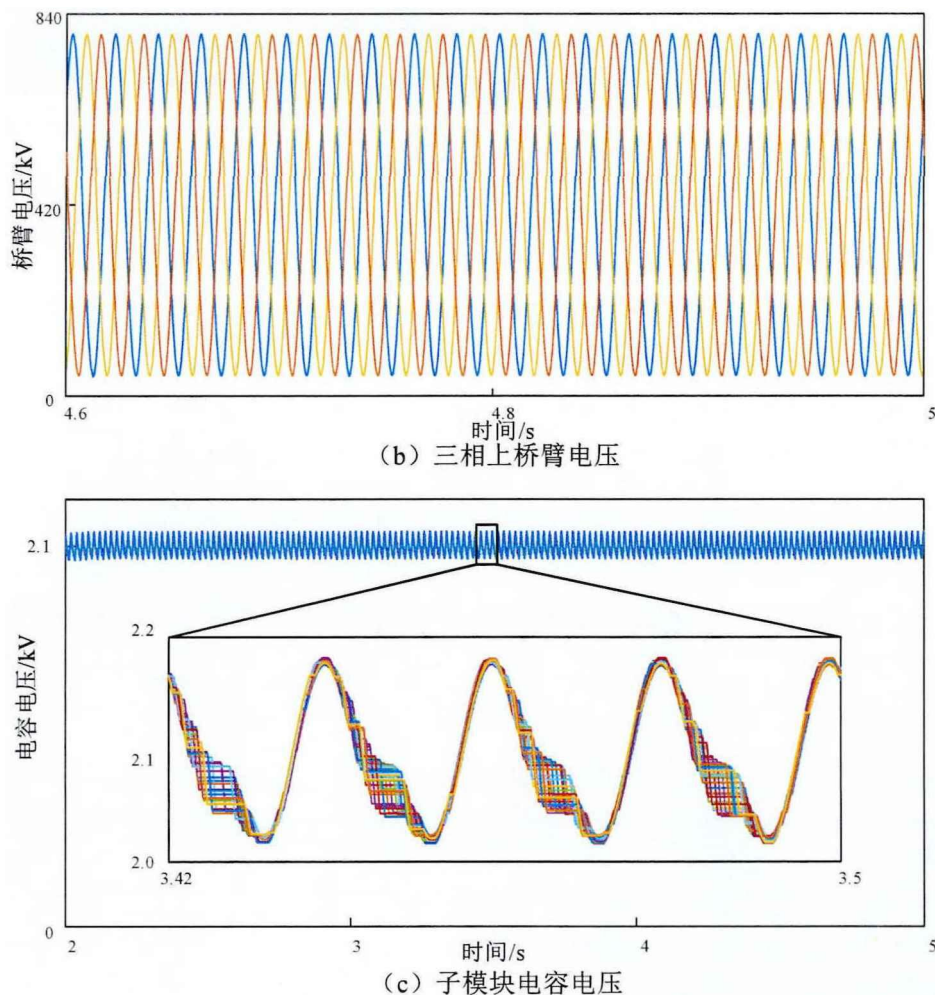


图 4-6 基于 FPGA 的实时仿真结果

为了评估所提出的均压排序算法在 FPGA 硬件资源和时间性能方面的性能，本研究利用 Vivado 软件对三种算法——本文提出的算法、桶排序算法和双调排序算法——的资源消耗进行了详细测试。测试指标涵盖四个主要方面：查找表（LUT）、寄存器（Register）、块级存储器（BRAM）以及数字信号处理器（DSP）资源。此外，时间性能的指标为基于 100MHz 时钟频率下完成一次排序操作所需的时间长度，测试结果如表 4-11 所示。

表 4-11 不同算法下 FPGA 资源占用及时间性能对比

算法	资源占用				时间性能
	LUT	Register	BRAM	DSP	Time
本文排序算法	33457	8166	94	28	6.67μs
桶排序算法	28646	7483	86	47	48.02μs
双调排序算法	202231	186930	83	69	6.98μs

通过对表中不同算法在 FPGA 资源占用及时间性能对比数据的分析,可以清晰地看到本文所提排序算法在资源占用和性能上的显著优势。

首先,从资源消耗的角度来看,排序算法在 FPGA 上主要消耗查找表和寄存器资源。因此,评估算法资源占用时,这两个指标成为了关键的考量点。本文排序算法的查找表和寄存器资源消耗分别为 33457 和 8166,虽然略高于桶排序算法的 28646 和 7483,但桶排序算法的时间性能却显著逊色,达到 $48.02\mu\text{s}$,远高于本文排序算法的 $6.67\mu\text{s}$ 。

另一方面,双调排序算法在时间性能上 ($6.98\mu\text{s}$) 与本文排序算法 ($6.67\mu\text{s}$) 相近,但其资源占用却显著高于本文排序算法,查找表和寄存器的使用量分别为 202231 和 186930,远超本文排序算法的资源需求。

综上所述,本文排序算法在保持较低资源占用的同时,实现了快速的时间性能。虽然在查找表和寄存器资源消耗上略高于桶排序算法,但其显著更快的时间性能弥补了这一点。同时,与双调排序算法相比,本文排序算法在时间性能相近的情况下,资源占用大大减少。因此,本文排序算法在 FPGA 资源受限的应用场景中具有明显的优越性和实用价值。

4.5 小结

本章主要围绕 FPGA 并行计算优势,对子模块电容电压均衡控制策略进行了硬件设计与仿真验证。首先,本文选定 VC707 套件和 Verilog HDL 语言作为硬件设计与实现的基础,并合理设置了约束条件,引入了数据转存与延时启动环节,确保数据的稳定与准确采集。

在硬件设计方面,本文构建了电压缓存模块、极值搜索模块以及触发信号生成模块。电压缓存模块负责预处理并缓存以数据流形式传输的子模块电容电压数据,为后续处理提供了完整且便捷的数据基础。极值搜索模块则充分利用 FPGA 的并行计算能力,通过并行排序算法与查找表技术,大幅提升了排序效率,进而优化了均压策略的执行速度。触发信号生成模块则基于预处理后的数据位置信息,精准生成触发信号,实现对子模块的精确控制,最终达成电压均衡的目标。

此外,本文搭建了基于硬件开发板与 PC 端上位机的测试验证平台,对所提均压控制策略进行了测试。结果表明,该策略在 FPGA 硬件环境下具有高度的可行性,显著提高了排序算法的计算速度,减少了控制链路的延迟。

第 5 章 结论与展望

5.1 总结

高效可靠的柔性直流输电技术在可再生能源并网与电网互联中发挥着不可或缺的作用。然而，随着模块化多电平换流器系统规模的日益扩大，子模块电容电压不平衡问题逐渐凸显，成为制约 MMC 系统性能提升和安全稳定运行的关键因素。针对这一问题，本文深入研究了模块化多电平换流器与现场可编程逻辑门阵列的工作原理及其应用，提出了针对大规模 MMC 的新型均压控制策略，并基于 FPGA 实现了该策略的硬件设计与仿真验证。论文的主要成果如下：

(1)提出了一种融合桶排序分布特性和并行全比较排序计算效率的极值排序算法。该算法通过优化传统电容电压均衡控制策略在处理速度方面的不足，显著提高了均压控制的实时性和准确性。通过与传统排序算法进行对比分析，本文证明了所提算法在时间和空间上均具有显著优势，特别适用于大规模 MMC 系统。

(2)解决了新型均压算法中最优分组数确定问题。通过数值计算方法与曲线拟合技术，本文成功推导出一个具有广泛适用性的最优分组数确定公式。这一公式为实际应用中确定最优分组数提供了理论依据，有助于实现更高效、更稳定的均压控制。

(3)实现了所提均压控制策略在 FPGA 上的硬件设计与仿真验证。通过设计电压缓存模块、极值搜索模块以及触发信号生成模块，实现了对子模块电容电压数据的高效处理与均衡控制。特别是极值搜索模块，利用 FPGA 的高度并行能力，通过并行排序算法与查找表，显著缩短了排序时间，并巧妙利用预处理后数据自带的位置信息作为索引，生成相应的触发信号，实现了对子模块的精准控制，从而达到了电压均衡的目标。这一改进为均压策略在实际硬件平台上的实现提供了有效支持，有助于提升电力系统的稳定性和可靠性。

综上所述，本论文在模块化多电平换流器均压控制策略及 FPGA 实现方面取得了一定成果，为相关领域的研究和应用提供了参考。通过优化排序算法和利用 FPGA 的并行处理能力，本文成功提出了一种高效、可靠的均压控制策略，并通过实验验证了其在实际应用中的可行性。这一研究不仅丰富了 MMC 均压控制策略的理论体系，还为柔性直流输电技术的发展提供了技术支持。对于推动柔性直流输电技术的广泛应用和促进国家“双碳”战略的实施具有重要意义。

5.2 展望

本文的研究工作还存在可优化或拓展之处,后续需进一步完善的方向包括:

(1) 本文实现了均压控制算法的源代码编写,但当前仅限于 397 个子模块的应用场景。可通过优化内部逻辑关系,将其封装为通用性更强、可配置性更高的 IP 核。后续工作应进一步完善 IP 核的输入接口与参数配置机制,以适应更多种类的子模块和灵活多变的配置需求,从而满足不同厂商和系统对均压控制策略的实际应用要求。

(2) 基于半桥子模块的柔性直流工程并不具备直流故障清除能力,而由半桥和全桥子模块按 1:1 混合的柔直系统在具备了直流故障穿越能力同时兼具了经济性。鉴于本文仅针对基于 HBSM-MMC 进行了验证,未来研究应拓展至混合半桥和全桥子模块的模块化多电平换流器,以全面评估算法的适用性。

(3) 随着 MMC 在远海送电等领域的应用日益广泛,受限于土地资源,其轻型化设计成为研究新热点。针对桥臂复用模块化多电平换流器,如何实现有效的均压控制是亟待解决的问题。因此,后续研究应聚焦于轻型化 MMC 的均压控制策略,以推动该技术在更多场景下的应用。

参考文献

- [1] 国家能源局. 国家能源局关于印发《2024 年能源工作指导意见》的通知[EB/OL]. (2024-03-18). [2024-04-03]. https://www.gov.cn/zhengce/zhengceku/202403/content_6941170.htm
- [2] 国家能源局. 国家能源局组织发布《新型电力系统发展蓝皮书》[EB/OL]. (2023-06-02). [2024-04-03]. <https://www.nea.gov.cn/2023-06/02/c1310724249.htm>
- [3] Saeedifard M, Iravani R. Dynamic performance of a modular multilevel back-to-back HVDC system[J]. IEEE Transactions on Power Delivery, 2010, 25(4): 2903-2912.
- [4] Xu J, Zhao C, Xiong Y, et al. Optimal design of MMC levels for electromagnetic transient studies of MMC-HVDC[J]. IEEE Transactions on Power Delivery, 2016, 31(4): 1663-1672.
- [5] 史书怀, 王丰, 卓放, 等. 模块化多电平换流器的电容均压算法比较及优化设计[J]. 电力系统自动化, 2017, 41(13): 150-155.
- [6] Li J, Konstantinou G, Wickramasinghe H R, et al. Operation and control methods of modular multilevel converters in unbalanced AC grids: A review[J]. IEEE Journal of Emerging and Selected Topics in Power Electronics, 2019, 7(2): 1258-1271.
- [7] 杨海钢, 孙嘉斌, 王慰. FPGA 器件设计技术发展综述[J]. 电子与信息学报, 2010, 32(03):714-727.
- [8] Li W, Gregoire L A, Belanger J. A modular multilevel converter pulse generation and capacitor voltage balance method optimized for FPGA implementation[J]. IEEE Transactions on Industrial Electronics, 2014, 62(5): 2859-2867.
- [9] 许建中, 李钰, 陆锋, 等. 降低 MMC 子模块电容电压纹波幅值的方法综述[J]. 中国电机工程学报, 2019, 39(2): 571-584.
- [10] 王奎, 郑泽东, 李永东. 新型模块化多电平变换器电容电压波动规律及抑制方法[J]. 电工技术学报, 2011, 26(5): 8-14.
- [11] Gao C, Jiang X, Li Y, et al. A DC-link voltage self-balance method for a diode-clamped modular multilevel converter with minimum number of voltage sensors[J]. IEEE Transactions on Power Electronics, 2013, 28(5): 2125-2139.
- [12] Jung J J, Cui S, Lee J H, et al. A new topology of multilevel VSC converter for a hybrid HVDC transmission system[J]. IEEE Transactions on Power Electronics, 2017, 32(6): 4199-4209.

- [13] Luo L, Zhang Y, Jia L, et al. A novel method based on self-power supply control for balancing capacitor static voltage in MMC[J]. IEEE Transactions on Power Electronics, 2018, 33(2): 1038-1049.
- [14] 张建坡, 崔涤穹, 田新成, 等. 自阻自均压模块化多电平换流器子模块拓扑及控制[J]. 电工技术学报, 2020, 35(18): 3917-3926.
- [15] 辛业春, 王朝斌, 李国庆, 等. 模块化多电平换流器子模块电容电压平衡改进控制方法[J]. 电网技术, 2014, 38(05): 1291-1296.
- [16] 赵昕, 赵成勇, 李广凯, 等. 采用载波移相技术的模块化多电平换流器电容电压平衡控制[J]. 中国电机工程学报, 2011, 31(21): 48-55.
- [17] 李笑倩, 宋强, 刘文华, 等. 采用载波移相调制的模块化多电平换流器电容电压平衡控制[J]. 中国电机工程学报, 2012, 32(9): 49-55.
- [18] 公铮, 伍小杰, 王钊, 等. 基于载波移相调制的模块化多电平变换器变频运行控制[J]. 中国电机工程学报, 2015, 35(11): 2822-2830.
- [19] Ilves K, Harnfors L, Norrga S, et al. Analysis and operation of modular multilevel converters with phase-shifted carrier PWM[J]. IEEE Transactions on Power Electronics, 2015, 30(1): 268-283.
- [20] Behrouzian E, Bongiorno M, Teodorescu R. Impact of switching harmonics on capacitor cells balancing in phase-shifted PWM-based cascaded H-bridge STATCOM[J]. IEEE Transactions on Power Electronics, 2017, 32(1): 815-824.
- [21] Hu P, Teodorescu R, Guerrero J M. Negative-sequence second-order circulating current injection for hybrid MMC under over-modulation conditions[J]. IEEE Journal of Emerging and Selected Topics in Power Electronics, 2020, 8(3): 2508-2519.
- [22] Cui S, Sul S K. A comprehensive DC short-circuit fault ride through strategy of hybrid modular multilevel converters (MMCs) for overhead line transmission[J]. IEEE Transactions on Power Electronics, 2016, 31(11): 7780-7796.
- [23] Dong Y, Tang J, Yang H, et al. Capacitor voltage balance control of hybrid modular multilevel converters with second-order circulating current injection[J]. IEEE Journal of Emerging and Selected Topics in Power Electronics, 2019, 7(1): 157-167.
- [24] 林周宏, 刘崇茹, 林雪华, 等. 基于载波移相调制的模块化多电平换流器冗余保护策略[J]. 电力系统自动化, 2015, 39(20): 109-115.
- [25] Li Z, Wang P, Zhu H, et al. An improved pulse width modulation method for chopper-cell-based modular multilevel converters[J]. IEEE Transactions on Power Electronics, 2012, 27(8): 3472-3481.

- [26] 汤广福, 庞辉, 贺之渊. 先进交直流输电技术在中国的发展与应用[J]. 中国电机工程学报, 2016, 36(07): 1760-1771.
- [27] 赵聪, 李耀华, 李子欣, 等. 适合于柔性高压直流输电系统的模块化多电平变流器的自适应均压方法[J]. 电网技术, 2017, 41(05): 1510-1518.
- [28] Wang S, Adam G P, Massoud A M, et al. Analysis and assessment of modular multilevel converter internal control schemes[J]. IEEE Journal of Emerging and Selected Topics in Power Electronics, 2020, 8(1): 697-719.
- [29] Wang J, Tang Y, Liu X. Arm current balancing control for modular multilevel converters under unbalanced grid conditions[J]. IEEE Transactions on Power Electronics, 2020, 35(3): 2467-2479.
- [30] 刘国伟, 姜齐荣, 魏应冬. 低频率工况下模块化多电平变流器电容电压平衡控制策略[J]. 电工技术学报, 2014, 29(08): 166-172.
- [31] 粟时平, 魏新伟, 牛鼎, 等. 模块化多电平换流器电容电压改进排序平衡方法[J]. 中国电机工程学报, 2017, 37(13): 3874-3882.
- [32] 常非, 杨中平, 陈俊, 等. 模块化多电平换流器电容电压平衡并行排序方法[J]. 高电压技术, 2016, 42(10): 3166-3171.
- [33] 樊强, 王乐, 冯谟可, 等. 适用于实时仿真的 MMC 子模块电容电压优化均衡方法[J]. 电力自动化设备, 2020, 40(11): 175-181.
- [34] Ilves K, Harnfors L, Norrga S, et al. Predictive sorting algorithm for modular multilevel converters minimizing the spread in the submodule capacitor voltages[J]. IEEE Transactions on Power Electronics, 2014, 30(1): 440-449.
- [35] Abdelsalam M, Marei M, Tennakoon S, et al. Capacitor voltage balancing strategy based on sub-module capacitor voltage estimation for modular multilevel converters[J]. CSEE Journal of Power and Energy Systems, 2016, 2(1): 65-73.
- [36] 喻锋, 王西田, 林卫星, 等. 一种快速的模块化多电平换流器电压均衡控制策略[J]. 中国电机工程学报, 2015, 35(04): 929-934.
- [37] 张婕, 曾国辉, 赵晋斌, 等. 基于改进冒泡排序的模块化多电平换流器电容电压均衡策略[J]. 电力系统保护与控制, 2020, 48(6): 92-99.
- [38] Li Y, Jones E A, Wang F. The impact of voltage-balancing control on switching frequency of the modular multilevel converter[J]. IEEE Transactions on Power Electronics, 2015, 31(4): 2829-2839.
- [39] 管敏渊, 徐政. MMC 型 VSC-HVDC 系统电容电压的优化平衡控制[J]. 中国电机工程学报, 2011, 31(12): 9-14.
- [40] 屠卿瑞, 徐政, 郑翔, 等. 一种优化的模块化多电平换流器电压均衡控制方法

- [J]. 电工技术学报, 2011, 26(05): 15-20.
- [41] 陆翌, 王朝亮, 彭茂兰, 等. 一种模块化多电平换流器的子模块优化均压方法[J]. 电力系统自动化, 2014, 38(03): 52-58.
- [42] 郭裕群, 赵成勇, 许建中, 等. 模块化多电平换流器子模块平均开关频率的精确控制方法[J]. 电力系统自动化, 2016, 40(19): 94-100.
- [43] Ding C, Zhang J, Cheng Y, et al. A enhanced control strategy for capacitance voltage equalization in modular multilevel converter[J]. Measurement and Control, 2024, 57(4): 391-403.
- [44] Li Z, Gao F, Xu F, et al. Power module capacitor voltage balancing method for a ± 350 -kV/1000-MW modular multilevel converter[J]. IEEE Transactions on Power Electronics, 2015, 31(6): 3977-3984.
- [45] Tu Q, Xu Z, Xu L. Reduced switching-frequency modulation and circulating current suppression for modular multilevel converters[J]. IEEE Transactions on Power Delivery, 2011, 26(3): 2009-2017.
- [46] 林周宏, 刘崇茹, 李海峰, 等. 模块化多电平换流器的子模块电容电压分层均压控制法[J]. 电力系统自动化, 2015, 39(07): 175-181.
- [47] 洪国巍. 模块化多电平换流器控制策略改进和控制参数优化[D]. 北京: 华北电力大学, 2017.
- [48] Liu Y, Peng F Z. A modular multilevel converter with self-voltage balancing part I: Mathematical proof[J]. IEEE Journal of Emerging and Selected Topics in Power Electronics, 2019, 8(2): 1117-1125.
- [49] Xin Y, Chen C, Sun L, et al. Module selection method with improved capacitor voltage balance for MMC using $(2N+1)$ SHEPWM[J]. International Journal of Electrical Power & Energy Systems, 2021, 133: 107294.
- [50] Mirzahosseini R, Iravani R, Zhang Y. An FPGA-based digital real-time simulator for hardware-in-the-loop testing of traveling-wave relays[J]. IEEE Transactions on Power Delivery, 2020, 35(6): 2621-2629.
- [51] Saad H, Ould-Bachir T, Mahseredjian J, et al. Real-time simulation of MMCs using CPU and FPGA[J]. IEEE Transactions on Power Electronics, 2015, 30(1): 259-267.
- [52] Guo X, Tang Y, Wu M, et al. FPGA-based hardware-in-the-loop real-time simulation implementation for high-speed train electrical traction system[J]. IET Electric Power Applications, 2020, 14(5): 850-858.
- [53] Guo P, Xu Q, Yue Y, et al. Hybrid model predictive control for modified modular multilevel switch-mode power amplifier[J]. IEEE Transactions on Power Electronics,

- 2021, 36(5): 5302-5322.
- [54] Hu J, Xiang M, Lin L, et al. Improved design and control of FBSM MMC with boosted AC voltage and reduced DC capacitance[J]. IEEE Transactions on Industrial Electronics, 2018, 65(3): 1919-1930.
- [55] 王成山, 丁承第, 李鹏, 等. 基于 FPGA 的光伏发电系统暂态实时仿真[J]. 电力系统自动化, 2015, 39(12): 13-20.
- [56] 李鹏, 王智颖, 王成山, 等. 基于多 FPGA 的有源配电网实时仿真器并行架构设计[J]. 电力系统自动化, 2019, 43(8): 174-182.
- [57] 王守相, 张春雨, 赵倩宇. 基于 FPGA 的新能源低压直流配电系统暂态实时仿真研究[J/OL]. 电工技术学报, 1-15 [2024-03-28]. DOI: 0.19595/j.cnki.1000-6753.tces.231035.
- [58] 孙宗辉, 郭希铮, 王诗楠, 等. 基于开关状态预测矫正的高频电力电子变换器实时仿真建模方法研究[J/OL]. 中国电机工程学报, 1-12 [2024-03-28]. DOI: 10.13334/j.0258-8013.pcsee.230781.
- [59] 王钦盛, 王灿, 潘学伟, 等. 基于 FPGA 的电力电子恒导纳开关模型修正算法及实时仿真架构[J]. 电力系统自动化, 2024, 48(01): 150-159.
- [60] Bai H, Liu C, Rathore A K, et al. An FPGA-based IGBT behavioral model with high transient resolution for real-time simulation of power electronic circuits[J]. IEEE Transactions on Industrial Electronics, 2018, 66(8): 6581-6591.
- [61] Ou K, Rao H, Cai Z, et al. MMC-HVDC simulation and testing based on real-time digital simulator and physical control system[J]. IEEE Journal of Emerging and Selected Topics in Power Electronics, 2014, 2(4): 1109-1116.
- [62] Liu C, Hou Y, Dong H, et al. Parallel multi-rate simulation scheme for modular multilevel converter-based high-voltage direct current with accurate simulation of high-frequency characteristics and field programmable gate array-based implementation[J/OL]. High Voltage, 1-15 [2024-3-20]. DOI: 10.1049/hve2.12418.
- [63] Xu J, Wang K, Wu P, et al. FPGA-based submicrosecond-level real-time simulation of solid-state transformer with a switching frequency of 50 kHz[J]. IEEE Journal of Emerging and Selected Topics in Power Electronics, 2021, 9(4): 4212-4224.
- [64] Chalanger H, Ould-Bachir T, Sheshyekani K, et al. A direct mapped method for accurate modeling and real-time simulation of high switching frequency resonant converters[J]. IEEE Transactions on Industrial Electronics, 2020, 68(7): 6348-6357.
- [65] Ould-Bachir T, Blanchette H F, Al-Haddad K. A network tearing technique for

- FPGA-based real-time simulation of power converters[J]. IEEE Transactions on Industrial Electronics, 2015, 62(6): 3409-3418.
- [66] Chalangar H, Ould-Bachir T, Sheshyekani K, et al. Methods for the accurate real-time simulation of high-frequency power converters[J]. IEEE Transactions on Industrial Electronics, 2021, 69(9): 9613-9623.
- [67] Milton M, Benigni A, Monti A. Real-time multi-FPGA simulation of energy conversion systems[J]. IEEE Transactions on Energy Conversion, 2019, 34(4): 2198-2208.
- [68] 管敏渊, 徐政, 屠卿瑞, 等. 模块化多电平换流器型直流输电的调制策略[J]. 电力系统自动化, 2010, 34(02): 48-52.
- [69] Montano F, Ould-Bachir T, David J P. An evaluation of a high-level synthesis approach to the FPGA-based submicrosecond real-time simulation of power converters[J]. IEEE Transactions on Industrial Electronics, 2017, 65(1): 636-644.
- [70] 何志兴, 罗安, 熊桥坡, 等. 模块化多电平变换器模型预测控制[J]. 中国电机工程学报, 2016, 36(05): 1366-1375.
- [71] 王宇, 刘崇茹, 凌博文, 等. 集成排序网络的模块化多电平换流器实时仿真模型[J]. 电网技术, 2020, 44(03): 1090-1098.
- [72] Dekka A, Wu B, Zargari N R, et al. Dynamic voltage balancing algorithm for modular multilevel converter: A unique solution[J]. IEEE Transactions on Power Electronics, 2015, 31(2): 952-963.
- [73] Liu H, Liu C. An MMC submodular capacitor voltage balancing scheme based on improved bucket sorting[C]. 2nd IEEE Conference on Energy Internet and Energy System Integration, Beijing, China, 2018: 1-6.
- [74] 王宇, 刘崇茹, 李庚银, 等. 适用于 FPGA 的模块化多电平换流器电容电压均衡控制方法[J]. 电力系统自动化, 2019, 43(08): 167-173.
- [75] 杨晓峰, 郑琼林. 基于 MMC 环流模型的通用环流抑制策略[J]. 中国电机工程学报, 2012, 32(18): 59-65+178.
- [76] 姜喜瑞, 贺之渊, 汤广福, 等. 基于禁忌搜索优化算法的高压大容量柔性直流输电电子模块电容电压平衡算法[J]. 中国电机工程学报, 2013, 33(21): 71-80+195.
- [77] 徐政. 柔性直流输电系统[M]. 北京: 机械工业出版社, 2017: 162-163.