第十届
全国大学生集成电路创新创业大赛

报告类型：系统设计报告

参赛杯赛：龙芯中科杯

作品名称：基于龙芯 SoC 的后量子密码与矩阵运算加速系统设计与验证

队伍编号：CICC1008439

团队名称：碳基生物

基于龙芯 SoC 的后量子密码与矩阵运算加速系统设计与验证

碳基生物

队伍编号：CICC1008439

目录

1 系统概述

1.1 设计目标

本设计基于龙芯中科提供的 LoongArch32r 软核处理器及 AXI 总线 SoC 框架，构建了一个集成多种硬件加速器的异构计算系统。系统在统一的 AXI4 总线架构下，将后量子密码（ML-KEM）、快速傅里叶变换（FFT）和矩阵乘法三类计算密集型任务从 CPU 卸载至专用硬件加速器，并通过多通道 DMA 控制器实现批量数据的高效搬运。设计同时完成了龙芯中科杯阶段任务三（矩阵乘法器 IP 设计）的要求。

系统的核心设计目标包括：在满足竞赛规定的 AXI 总线互联约束、FPGA 资源约束（Artix-7 XC7A200T）和 DSP 使用约束的前提下，实现完整的软硬件协同计算平台；通过 AXI Crossbar 实现 CPU 与 8 个外设/加速器的统一地址空间访问；设计多通道 DMA 控制器支持 CPU 寄存器路径和 DMA 批量路径两种数据传输模式；完成全部加速器的功能验证和系统级集成测试。

1.2 软硬件功能划分

系统的软硬件划分遵循以硬件加速计算密集操作、以软件管理控制流和数据调度的原则。表**??**列出了各子系统在软硬件之间的功能分配。

表 1: 软硬件功能划分

功能	**硬件实现（本设计贡献）**	**软件管理**
ML-KEM 密钥生成/封装/解封装	`mlkem_top` 全硬件加速 + DMA 批量数据搬运	参数与种子配置、操作触发、结果校验
FFT 1024 点变换	FFT SDF 流水线	波形数据写入、模式配置、启动触发
矩阵乘法 C=A×B	`matmul_axi_slave` 双 Bank 加速器	矩阵布局配置、计算触发、CRC 校验
matmul 批量调度	`cdma_matmul_bridge` 硬件序列控制器	源/目的基址与组数一次性配置
DMA 数据搬运	4 通道 DMA 控制器（mem2mem/mem2ep/ep2mem）	传输描述符配置、完成轮询

CPU 负责系统的整体调度：配置各加速器的控制寄存器、触发计算启动、轮询或响应中断以获知完成状态、以及通过串口输出测试结果。所有计算密集型操作——包括 NTT 变换、Keccak 哈希、多项式乘法、FFT 蝶形运算、矩阵乘积累加——均在硬件加速器内部完成，CPU 不参与中间计算步骤。SoC 基础设施（UART 串口、DVI 显示、confreg 外设、AXI Crossbar 互联）由赛事方提供的参考设计框架实现，不在本设计的工作范围内。

2 系统架构设计

2.1 总体架构

图??给出了 SoC 系统的顶层架构。系统以 LoongArch32r CPU 核心为主控制器，通过 AXI4 总线经时钟域交叉桥接模块至系统总线域（50 MHz），再由 AXI 交叉开关将 CPU 和 DMA 两个主设备的访问请求按地址高位译码分发至 8 个从设备。

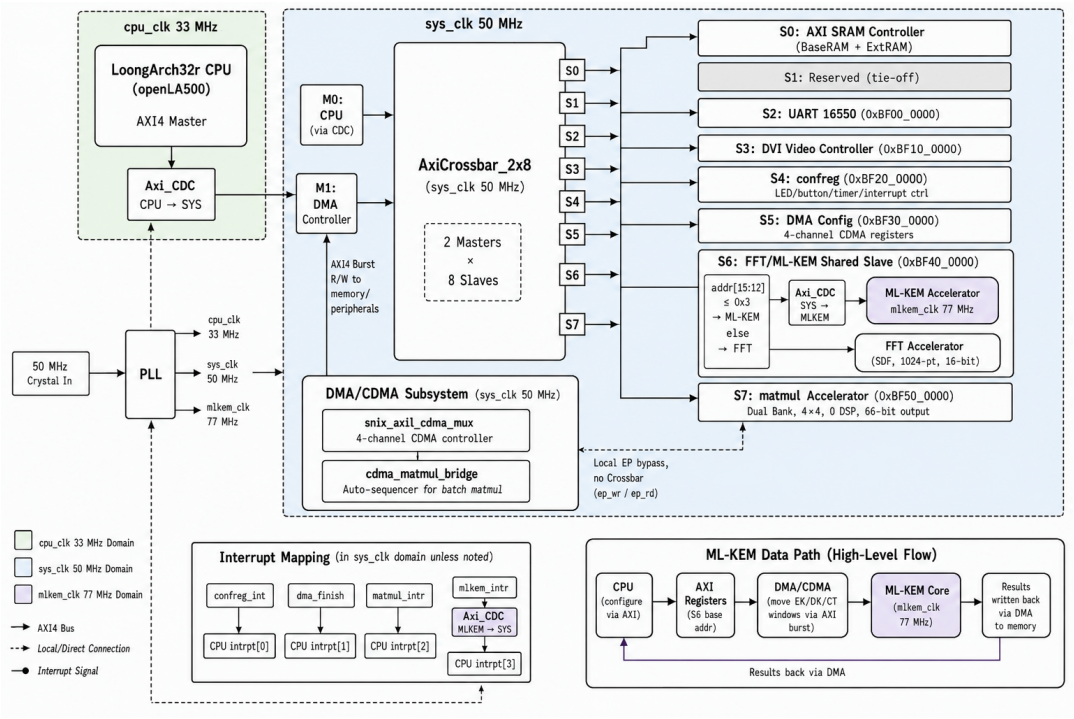

图 1: SoC 系统顶层架构

2.2 地址空间分配

系统地址空间采用统一编址，CPU 通过非缓存虚拟地址段（0xBFxxxxxx）访问各外设，对应物理地址段为 0x1Fxxxxxx。表??列出各从设备的地址窗口。

表 2: 系统地址空间映射

端口	物理地址	虚拟地址	外设
Slave 0	0x1C00_0000–0x1C7F_FFFF	—	BaseRAM + ExtRAM
Slave 2	0x1F00_0000–0x1F0F_FFFF	0xBF00_0000	UART 16550
Slave 3	0x1F10_0000–0x1F1F_FFFF	0xBF10_0000	DVI 视频控制器
Slave 4	0x1F20_0000–0x1F2F_FFFF	0xBF20_0000	confreg（GPIO/定时器/中断）
Slave 5	0x1F30_0000–0x1F3F_FFFF	0xBF30_0000	DMA/CDMA 配置空间
Slave 6	0x1F40_0000–0x1F4F_FFFF	0xBF40_0000	FFT + ML-KEM 共享窗口
Slave 7	0x1F50_0000–0x1F5F_FFFF	0xBF50_0000	矩阵乘法加速器

Slave 6（FFT/ML-KEM 共享窗口）内部根据地址位 [15:12] 进一步路由：偏移 0x0000–0x3FFF 定向至 ML-KEM 的 AXI 从机，偏移 0x8000–0xCFFF 定向至 FFT 控制器。Slave

5（DMA 配置窗口）内部根据地址位 [11:8] 分发：0x000–0x1FF 区间通往 DMA 通道寄存器，0x200–0x2FF 区间通往 matmul 序列控制器的配置寄存器。

2.3 中断系统

系统中断链路由外设配置与中断控制模块集中管理，6 个中断源经优先级编码后送入 CPU。中断映射关系如表**??**所示。

表 3: 中断映射

中断位	来源	触发条件
intrpt[0]	confreg_int	定时器/按键/UART/FFT 任一中断
intrpt[1]	dma_finish	DMA 传输完成或 matmul 序列完成
intrpt[2]	matmul_intr	矩阵乘法计算完成
intrpt[3]	mlkem_intr	ML-KEM 操作完成

2.4 时钟与复位架构

系统采用单一外部晶振（50 MHz）经 PLL 生成三个时钟域。CPU 核心运行于约 33 MHz（cpu_clk），AXI 总线、DMA 控制器以及 matmul 和 FFT 等外设统一运行于 50 MHz（sys_clk）。ML-KEM 加速器通过 PLL 独立获取约 77 MHz 的高频时钟（mlkem_clk），专用于其高吞吐计算通路。ML-KEM 的 AXI 从机接口内部通过跨时钟域桥接模块实现系统总线与高速计算域之间的同步，将 CPU 的寄存器读写请求安全传递至高频时钟域，中断信号同样经过跨域同步后送入 CPU 中断系统。三个时钟域之间通过异步时钟组约束正确隔离，各域配备独立的复位同步器，以 PLL 锁定信号作为复位释放的使能条件。

3 核心子系统设计

3.1 DMA 与 CDMA 数据搬运子系统

DMA 子系统是本设计中连接 CPU、外部存储器和加速器的数据搬运中枢，由中央 DMA 控制器、matmul 自动序列控制器和 AXI-Lite 地址分流逻辑三个模块协同构成。

中央 DMA 控制器（snix_axil_cdma_mux）以 4 通道配置例化，每通道配备独立的源地址、目的地址、传输长度和模式寄存器组，寄存器组基地址以 0x40 为步长间隔排列。控制器向外暴露两组数据接口：AXI4 主设备接口连接至 Crossbar 的 Master 1 端口，用于在系统存储器和外设之间发起标准 AXI Burst 读写传输；本地 Endpoint 接口以 valid/ready 握手协议直接连接至 matmul 加速器的内部缓冲区，实现低延迟的本地数据搬运，该通道完全绕开 Crossbar 的路由延迟。

matmul 自动序列控制器（cdma_matmul_bridge）使批量矩阵运算的数据调度脱离了 CPU 的逐次寄存器操作。CPU 通过其 AXI-Lite 配置接口一次性设置源基地址、目的基地址、待处理的矩阵组数和各组的数据跨度后，控制器内部的硬件状态机自动循环执行以下操

作序列：向 DMA 发起从 ExtRAM 读取一组源矩阵数据至 matmul 输入 Bank 的请求；等待 DMA 返回读完成信号后触发 matmul 计算启动脉冲；轮询 matmul 的完成状态后向 DMA 发起将结果 Bank 数据写回 ExtRAM 的请求；递增矩阵组索引后进入下一轮循环。控制器的序列完成中断信号与 DMA 的全局传输完成信号经逻辑或合并后送入 CPU 中断系统。

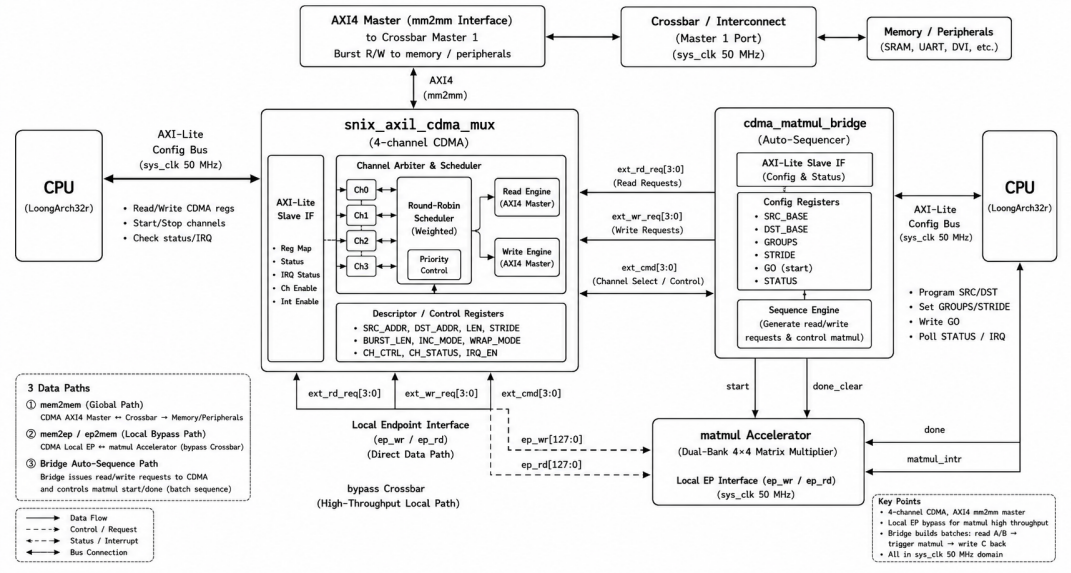

图 2: DMA/CDMA 数据搬运子系统架构

表**??**列出 DMA 通道和 matmul 序列控制器的核心寄存器。

表 4: DMA/CDMA 核心寄存器

偏移（通道 0）	**名称**	**功能**
4 通道 DMA（snix_axil_cdma_mux），通道 n 基址 = 0xBF300000 + n×0x40		
0x00	SRC_ADDR	源地址
0x04	DST_ADDR	目的地址
0x08	LENGTH	传输字节数
0x10	CTRL	bit0 启动，bit[5:3]=AxSIZE，bit[13:6]=AxLEN
0x14	STATUS	bit0=busy(RO)，bit1=done(W1C)
0x18	MODE	0=mem2mem，1=mem2ep，2=ep2mem
0x1C	EP_CFG	[3:0]=bank，[15:8]=word offset，[19:16]=ep id
matmul 序列控制器（cdma_matmul_bridge），基址 = 0xBF300200		
0x200	SEQ_CTRL	bit0=start
0x204	SEQ_STATUS	bit0=busy(RO)，bit1=done(W1C)
0x208	SEQ_SRC_BASE	输入矩阵组物理源基址
0x20C	SEQ_DST_BASE	输出矩阵组物理目的基址
0x210	SEQ_GROUPS	自动处理的矩阵组数
0x214	SEQ_MAT_BASE	matmul AXI slave 基址（固定 0x1F500000）
0x218	SEQ_SRC_STRIDE	相邻输入组间隔（默认 128 B）
0x21C	SEQ_DST_STRIDE	相邻输出组间隔（默认 192 B）

3.2 矩阵乘法加速器

矩阵乘法加速器根据赛事任务要求设计，完成最大 4 阶矩阵乘法运算 $C = A \times B$。加速器内部包含双 Bank 结构以实现流水化操作：一个 Bank 执行乘法计算时，另一个 Bank 可通过 DMA 预装载下一组输入数据。

每个 Bank 包含独立的 A 矩阵缓冲区（16 个 32-bit 字）、B 矩阵缓冲区（16 个 32-bit 字）和 C 结果缓冲区（16 个 66-bit 元素拆分为 48 个 32-bit 字）。计算阵列由 4 个并行乘法器实例组成，每个实例负责一行结果向量的点积累加。乘法器完全采用 LUT 和 CARRY4 逻辑资源实现 32 乘 32-bit 无符号乘法，严格遵循赛事不得使用 DSP 资源的约束。

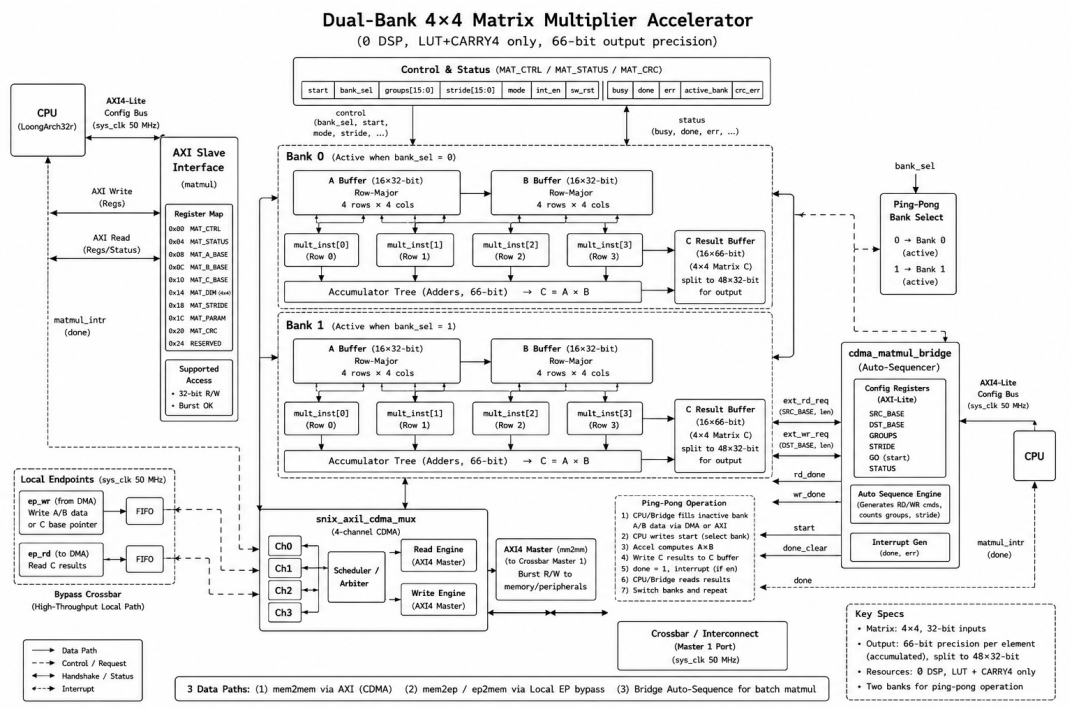

图 3: 矩阵乘法加速器架构

加速器的控制寄存器接口如表**??**所示。

表 5: matmul 加速器寄存器映射

偏移	名称	功能
0x000	MAT_CTRL	bit0 启动 Bank0，bit1 启动 Bank1
0x004	MAT_STATUS	bit0=busy(RO)，bit1=done(W1C)，bit2=error(W1C)
0x010	MAT_CRC	CRC32 累积值（^0xFFFFFFFF），写清
0x100–0x13C	B0_A[0..15]	Bank0 A 矩阵（16×32-bit）
0x140–0x17C	B0_B[0..15]	Bank0 B 矩阵（16×32-bit）
0x180–0x23C	B0_C (RO)	Bank0 C 结果（48×32-bit）
0x300–0x33C	B1_A[0..15]	Bank1 A 矩阵
0x340–0x37C	B1_B[0..15]	Bank1 B 矩阵
0x380–0x43C	B1_C (RO)	Bank1 C 结果

3.3 FFT 加速器

FFT 加速器基于单路径延迟反馈（Single-path Delay Feedback，SDF）架构实现 1024 点 16-bit 定点 FFT/IFFT。其流水线由 5 级 SDF 单元级联构成，每级包含一个基-2 蝶形单元、一个复数乘法器和一个深度逐级减半的延迟缓冲器。旋转因子由片上 ROM 提供。加速器通过 AXI 从机封装接口挂载于总线，提供输入实部和虚部的 BRAM 写窗口（各 1024 个 16-bit 字）、输出 BRAM 读窗口以及控制和状态寄存器。输入输出数据通过位反转地址索引实现算法要求的序映射。

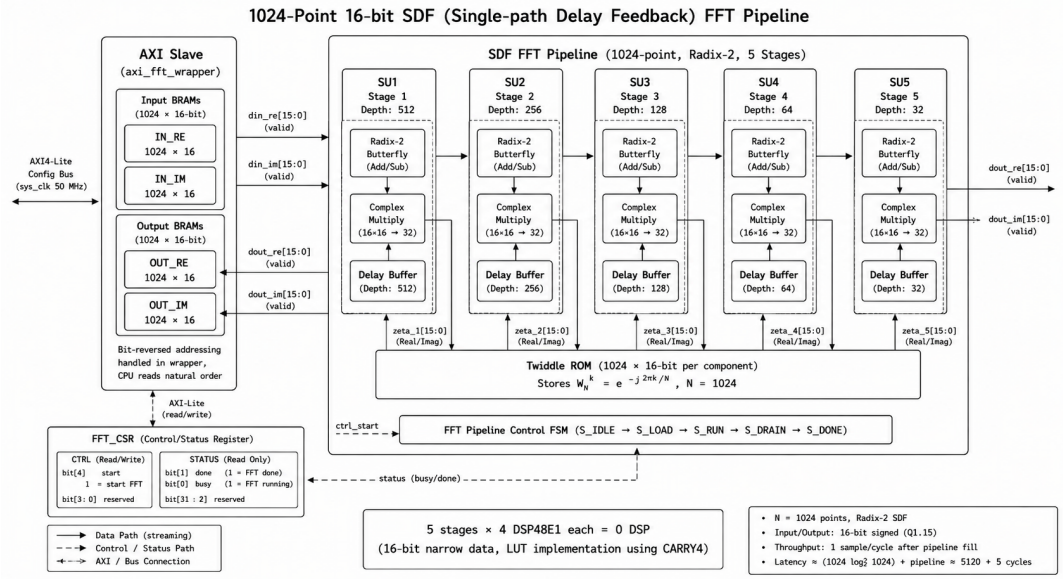

图 4: FFT 加速器 SDF 流水线架构

3.4 ML-KEM 后量子密码加速器

ML-KEM 后量子密码加速器是本系统中最复杂的单一加速器，完整的 IP 设计已在《*ML-KEM 后量子密钥封装硬件 IP 核设计与验证*》报告中详述，此处仅概述其在系统集成层面的关键特性。

该加速器核心在单一模块内支持 FIPS 203 标准全部三种操作和三种安全等级，通过运行时端口实现动态选择。加速器通过 AXI 从机接口挂载于总线，提供内存映射寄存器用于控制、状态查询和逐字节数据装载与回读。三个直接存储器窗口（EK、DK、CT）支持 DMA 批量数据搬运。

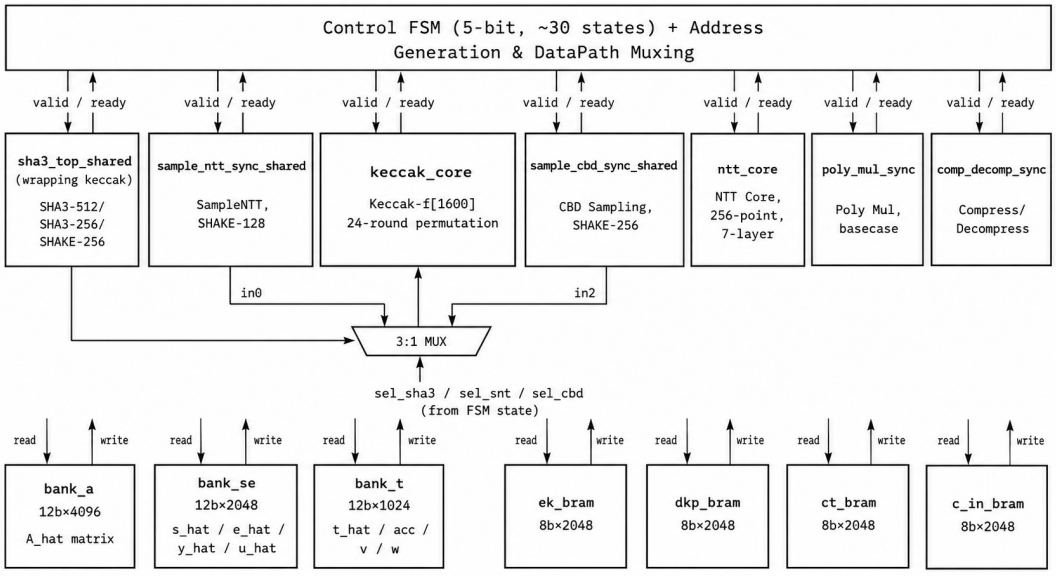

图 5: ML-KEM 加速器内部模块层次结构

4 芯片前端设计与验证方案

4.1 SoC 集成设计方法

本设计采用增量集成的设计方法学。各加速器 IP 首先在独立仿真环境中完成模块级功能验证，确认算法正确性和接口协议合规性后，再逐步嵌入 SoC 顶层。集成过程涵盖五个顺序步骤。

第一步是 AXI 从机接口适配。三个加速器各自通过对应的 AXI 从机包装器挂载至总线——矩阵乘法器提供控制寄存器和双 Bank 数据窗口，ML-KEM 通过从机接口暴露内存映射寄存器和 DMA 窗口，FFT 通过封装接口提供输入输出 BRAM 和状态寄存器。第二步是交叉开关端口分配，根据地址空间规划在顶层集成文件中完成各 AXI 从机与交叉开关对应端口的信号连接。FFT 和 ML-KEM 因共享同一从机端口，通过内部地址译码逻辑实现基于地址位的二次路由。第三步涉及中断链路的连接，各加速器的完成或错误信号根据中断映射表连接至中断控制器的对应输入位或 CPU 的直接中断输入。第四步完成时钟与复位的集成，确保各加速器的时钟信号通过 PLL 的对应输出端获取，复位信号通过复位同步器产生与各自时钟域同步的释放时序，避免异步复位引起的亚稳态问题。第五步为软件 BSP 适配，在 SDK 中为每个加速器添加基地址宏定义、寄存器偏移量枚举和基本的读写封装函数。

4.2 DMA 数据路径验证

DMA 子系统的验证覆盖三条独立的数据搬运路径，每条路径侧重于不同的硬件交互场景。

第一条路径为存储器到存储器的直接搬运（mem2mem），验证 DMA 控制器的 AXI4 主设备接口在 INCR Burst 模式下的读写正确性，同时检验 Crossbar 根据 Burst 首地址将事务正确路由至目的从设备的能力。测试通过向 ExtRAM 的不重叠源区和目的区写入已知

数据模式、触发 DMA 传输后逐字比对来完成。第二条路径涉及 DMA 与 matmul 加速器之间的本地 Endpoint 通信（mem2ep 与 ep2mem）。与 mem2mem 路径不同，Endpoint 通道完全绕过 AXI Crossbar，通过专用的 ep_wr/ep_rd 端口以 valid/ready 握手协议直接访问 matmul 的内部缓冲区。该路径的验证重点在于确保 DMA 侧和 matmul 侧的流控握手在所有传输长度下均不会出现数据丢失或死锁。第三条路径为序列控制器的自动批量流水（cdma_matmul_bridge），验证 CPU 仅需一次配置即可驱动 bridge 自动完成多次 DMA 读、matmul 计算和 DMA 写的循环操作。该路径的验证关注多组连续处理场景下的状态机转移正确性、传输完成中断的准确触发时机以及异常条件下的错误恢复行为。

4.3 系统级仿真验证

系统级验证在 Vivado XSIM 仿真器上通过 mycpu_tb 测试平台执行。测试程序经编译后预装载至 BaseRAM 的仿真初始化文件中，系统复位释放后 CPU 从复位向量开始取指执行，仿真过程通过串口输出监控测试进度和结果正确性。系统级验证覆盖四类典型使用场景。

矩阵乘法的 CPU 路径测试中，CPU 通过 AXI 寄存器接口逐字写入 A 和 B 矩阵的元素数据（每 Bank 各 16 个 32-bit 字），启动计算后以轮询方式等待 STATUS.done 标志置位，随后从 C 结果缓冲区回读 48 个 32-bit 字并进行 CRC32 完整性校验。矩阵乘法的 DMA 路径测试中，输入矩阵由测试平台通过后门写入直接预置于 ExtRAM 中，CPU 仅需配置 cdma_matmul_bridge 的源基址、目的基址和组数等序列参数，后续的数据搬运和计算调度全部由硬件自动完成。ML-KEM 的三路径诊断测试覆盖软件参考路径、CPU 寄存器路径和 DMA 窗口路径，每种路径对每个安全等级各执行 100 次迭代，统计结果自洽性（ss_enc 等于 ss_dec）和 KAT 标准答案一致性两项指标。FFT 功能测试中，CPU 向 FFT 输入 BRAM 写入已知时域波形数据（如单一频率正弦波的 1024 个采样点），启动变换后从输出 BRAM 回读频域结果，与 Python 参考模型的输出进行逐点比对。

5 仿真测试结果与分析

5.1 矩阵乘法测试结果

矩阵乘法加速器在 SoC 仿真环境中完成了完整功能验证，CRC32 校验值与参考模型完全一致。表**??**列出批量矩阵运算的时序分解。

表 6: 矩阵乘法测试结果

阶段	**CPU 周期**	**说明**
初始化 (PROLOGUE)	26,580	程序启动与运行环境初始化
批量计算循环 (LOOP)	171,840	10 组矩阵的 DMA 搬运 + 计算 + 写回
每组平均	17,184	含 DMA 读、matmul 计算、DMA 写
收尾 (EPILOGUE)	14,640	结果校验与串口输出
总计	**213,060**	约 6.46 ms（@33 MHz CPU）

5.2 ML-KEM 系统级测试结果

ML-KEM 加速器在 SoC 集成环境中完成了软件参考（SW）、CPU 寄存器路径和 DMA 窗口路径各 100 次迭代的系统级诊断测试。表**??**列出三种安全等级下各路径完成一轮完整往返（KeyGen+Encaps+Decaps）的平均 CPU 周期数、耗时和加速比。

表 7: ML-KEM 系统级测试结果（100 次迭代均值）

安全等级	SW (cycles)	CPU (cycles)	DMA (cycles)	CPU 加速比	DMA 加速比
ML-KEM-512	2,594,034	200,471	52,995	12.94×	48.95×
ML-KEM-768	4,065,123	296,193	80,648	13.72×	50.41×
ML-KEM-1024	6,041,204	404,476	113,273	14.94×	53.33×
全部	**PASS**	**PASS**	**PASS**	—	—

DMA 路径相对于纯软件实现取得了最高 53.33× 的加速比，相对于 CPU 寄存器路径额外贡献了约 3.6–3.8× 的效率提升。全部 360 个 KAT 测试用例逐字节比对 NIST 标准答案通过，Decaps 隐式拒绝路径同样验证正确。

5.3 FFT 测试结果

FFT 加速器在 SoC 环境中完成了 CPU 寄存器路径和 DMA 路径两种模式的功能与性能测试。测试使用单一频率正弦波作为输入（1024 点采样），硬件输出与软件参考 FFT 的逐 bin 比对结果验证了变换的正确性。表**??**列出两种数据路径的性能对比。

表 8: FFT 加速器测试结果

测试模式	耗时 (ms)	加速比	最大误差
CPU 寄存器路径	1.724	142.55×	1 (Re) / 2 (Im)
DMA 路径	0.293	838.32×	1 (Re) / 2 (Im)
软件参考 (33 MHz CPU)	245.6	—	—

CPU 寄存器路径中，CPU 逐字通过 AXI 寄存器写入 1024 个实部和虚部样本，启动 FFT 后轮询完成状态，再逐字回读输出结果，总耗时约 1.72 ms（硬件计算 + 寄存器 I/O），加速比为 143×。DMA 路径中，输入和输出数据全部通过 DMA 的 mem2ep 和 ep2mem 通道批量搬运，绕过了逐字 AXI 寄存器访问的开销，总耗时降至 0.29 ms，加速比达到 838×。两种路径的硬件输出与软件参考 FFT 的最大偏差仅为 1（实部）和 2（虚部），验证了 16-bit 定点 FFT 硬件实现的高精度。

5.4 系统时序收敛

完整 SoC 在 Xilinx Artix-7 XC7A200T-1 器件上完成布局布线，三个时钟域均通过时序收敛。主要时序指标如表**??**所示。

表 9: SoC 布局布线后时序收敛结果

时钟域	频率	Setup WNS	违例端点
cpu_clk	33 MHz	+0.737 ns	0 / 17,197
mlkem_clk	77 MHz	+0.070 ns	0 / 31,646
sys_clk	50 MHz	+0.125 ns	0 / 22,307

5.5 系统资源利用

SoC 整体资源利用见表**??**。数据来自 Vivado 2019.2 布局布线后报告(器件 XC7A200T-1，无 DSP 约束配置)。

表 10: SoC 整体资源利用

资源	已用	可用	利用率
Slice LUTs（Total）	71,196	133,800	53.2%
Slice LUTs（Logic）	69,453	133,800	51.9%
Slice Registers	39,690	267,600	14.8%
Block RAM Tile	28.0	365	7.7%
RAMB36E1	10	365	2.7%
RAMB18E1	36	730	4.9%
DSP48E1	0	740	0.0%
BUFG	4	32	12.5%

表**??**给出按模块层次分解的资源利用明细，从中可以清晰看出各子系统在总体面积中的贡献比例。

表 11: 主要模块资源利用明细

模块	LUT	FF	BRAM	占比
ML-KEM	21,697	15,826	2×36 + 14×18	30.5%
FFT	6,949	1,083	8×36 + 4×18	9.8%
矩阵乘法	10,204	3,239	—	14.3%
CPU (core_top)	8,553	5,591	20×18	12.0%
DMA/CDMA	1,558	1,476	1×18	2.2%
Crossbar + CDC + 适配器	2,102	621	—	3.0%
matmul 序列桥	247	578	—	0.3%
UART + confreg + DVI + SRAM	1,711	1,222	—	2.4%
顶层胶合逻辑	812	4	—	1.1%
合计	**71,165**	**39,682**	**10×36 + 31×18**	**100%**

注：Vivado 层次化报告中各子模块合计可能略小于顶层总数，原因在于跨层次 LUT 合并（cross-hierarchy LUT combining）机制。表中顶层总计为 71,196 LUT。

从模块资源分布可以得出以下观察。ML-KEM 加速器消耗 30.5% 的 LUT，其 ntt_core 采用双蝶形加 4 路 Banked BRAM 架构，poly_mul_sync 的系数和中间数据缓冲亦通过

BRAM 实现，有效控制了 LUT 开销。矩阵乘法加速器以 14.3% 的 LUT 占比位居第二，其 4 个并行 32×32-bit 流水线乘法器（每实例约 1,000 LUT）是面积的主要贡献者。CPU 核心消耗 12.0% 的 LUT，其中 BTB（分支目标缓冲器）和 TLB（地址转换旁路缓冲器）合占约 3,000 LUT。FFT 加速器虽仅占 9.8% 的 LUT，但消耗了最多的 BRAM 资源——8 个 RAMB36E1 用于 5 级 SDF 架构中的延迟缓冲和旋转因子 ROM 存储。

DSP 使用量为零体现了竞赛规则约束和设计策略的一致性——ML-KEM 不使用 DSP、矩阵乘法受竞赛条款限制、FFT 采用 16-bit 窄数据宽度，三者共同导致了全 LUT 乘法映射的设计决策。

5.6 功耗分析

布局布线后系统功耗的层次化分解如表**??**所示。

表 12: 系统功耗层次化分解

模块	**动态功耗 (W)**	**占动态功耗比**
ML-KEM（u_mlkem）	0.218	40.0%
PLL 时钟生成器	0.138	25.3%
矩阵乘法（u_matmul）	0.081	14.9%
FFT（u_FFT）	0.027	5.0%
CPU（u_cpu）	0.025	4.6%
DMA/CDMA	0.008	1.5%
其它外设与总线	0.048	8.8%
动态功耗合计	**0.545**	100%
器件静态功耗	0.141	—
总片上功耗	**0.686**	—

系统总功耗为 0.686 W，其中动态功耗占 79.4%（0.545 W），静态功耗占 20.6%（0.141 W）。ML-KEM 是最大的动态功耗来源（40.0%），与资源利用分布一致。PLL 以 25.3% 的比例位居第二，反映了其驱动多时钟域（cpu_clk、sys_clk）的固有功耗开销。按资源类型划分，信号翻转功耗（0.194 W，占动态功耗 35.6%）和 Slice Logic 功耗（0.109 W，20.0%）是两类最大的器件级功耗来源。

5.7 设计复杂度分析

表**??**列出主要模块的 Rent 指数和平均扇出数。Rent 指数是衡量模块内部互联复杂度的指标，数值越低表示结构越规整、局部互联越密集。

ML-KEM 的 Rent 指数（0.41）在主要模块中最低，表明其数据通路具有高度规整的局部互联特性——多项式系数在相邻流水级之间传递，NTT 蝶形单元按固定模式访问系数寄存器，这种结构化特征有利于布局布线的时序收敛。相比之下，矩阵乘法器和 CPU 的 Rent 指数较高（0.59–0.60），反映其控制逻辑的随机互联特性。FFT/ML-KEM 共享从机以 87,846 个实例的规模（占全系统 66.7% 的 LUT 实例数）成为物理设计中最庞大的层次化模块。

表 13: 模块设计复杂度（Rent 指数）

模块	Rent 指数	平均扇出	总实例数
矩阵乘法器	0.60（Normal）	3.76	15,711
CPU	0.59（Normal）	3.96	15,822
UART	0.55（Normal）	3.54	1,296
DMA/CDMA	0.46（Low）	3.31	3,435
ML-KEM	0.41（Low）	3.78	87,846

5.8 高扇出网络分析

表??列出布局布线后扇出最高的前 5 个网络，这些网络是潜在的时序瓶颈和布线拥塞来源。

表 14: 高扇出网络（Top 5）

网络名称	扇出	驱动源
sys_clk（全局时钟）	35,652	BUFG
ML-KEM ek_bram 延迟寄存器	24,670	LUT5
CPU 复位网络	5,108	LUT1
ML-KEM poly_mul 计数器	3,758	FDCE
matmul FSM 独热状态	2,151	FDRE

全局时钟网络（sys_clk，BUFG 驱动）以 35,652 个负载位居首位，在当前 50 MHz 工作频率下不构成时序问题。ML-KEM 内部有两个高扇出网络进入前 5 名：系数存储 BRAM 的延迟寄存器链（24,670 扇出）源于字节编码器中一个组合使能信号被复制到大量并行路径；多项式乘法模块的加载计数器（3,758 扇出）属于多路复用地址总线的经典高扇出场景。这些网络在当前工作条件下均未产生时序违例。

5.9 验证结果汇总

表??给出系统级验证的覆盖范围和结果汇总。

表 15: 系统级验证结果汇总

验证项	测试路径	迭代次数	通过率	关键指标
矩阵乘法	CPU 寄存器路径	—	100%	CRC32 校验正确
矩阵乘法	DMA 序列控制器	10 组	100%	总计 213,060 周期
ML-KEM	SW/CPU/DMA 三路径	100/级	100%	KAT 逐字节比对
FFT	CPU 寄存器路径	1 次	100%	max error 2, 143× 加速比
FFT	DMA 路径	1 次	100%	max error 2, 838× 加速比
DMA	mem2mem	多次	100%	数据逐字比对
DMA	mem2ep/ep2mem	多次	100%	matmul Buffer 读写正确
DMA	序列桥批量流水	10 组	100%	全自动循环完成

6 结论

本设计在龙芯 SoC 平台上成功构建了一个集成多种硬件加速器的异构计算系统。

在 AXI 总线集成架构方面，以 AXI 交叉开关为核心互连部件，实现了 CPU 和 DMA 双主设备对 8 个从设备的统一地址空间访问，涵盖外部 SRAM 控制器、UART 串口、DVI 显示控制器、GPIO 与定时器、DMA 配置空间、FFT 与 ML-KEM 共享加速窗口和矩阵乘法器。共享从机的内部二次路由设计——FFT 与 ML-KEM 通过地址高位区分并共享同一端口——在不增加交叉开关端口数量的前提下实现了两种加速器的无缝共存，体现了在资源受限场景下灵活地址规划的工程价值。

在 DMA 数据搬运体系方面，4 通道 CDMA 控制器与 matmul 自动序列控制器的协同设计显著降低了 CPU 在批量计算任务中的参与度。序列控制器将逐组的源数据读取、矩阵计算触发、结果写回和组索引递增等操作固化为硬件状态机的自动循环，CPU 仅需在批量任务启动前一次性配置 6 个序列参数。三条数据路径——mem2mem 直接存储搬运、本地 Endpoint 加速器旁路通信和序列控制器的全自动批量流水——覆盖了 SoC 中所有的数据传输场景。

在赛事任务完成度方面，矩阵乘法加速器严格遵循赛事规范：不使用 DSP 资源而全部以 LUT 逻辑实现 32 乘 32-bit 乘法；支持 4 乘 4 最大矩阵规模并兼容更小的非方阵规模（通过填零对齐）；输出结果按 66-bit 精度拆分为 3 个 32-bit 字的标准格式；输入输出数据布局完全符合 128 字节源组和 192 字节结果组的赛事定义。CRC32 校验、批量序列测试和系统级仿真验证共同确认了加速器设计的完整正确性。

在后量子密码硬件集成方面，ML-KEM 加速器通过共享 AXI 从机与 FFT 复用 Slave 6 端口，在不增加 Crossbar 端口数量的前提下完整实现了 FIPS 203 标准的全部三种操作和三种安全等级的硬件加速。三路径诊断测试（软件参考、CPU 寄存器、DMA 窗口各 100 次迭代，覆盖 ML-KEM-512/768/1024 三种等级）全部通过，验证了 IP 核在真实 SoC 环境中的功能正确性和软硬件接口的长期运行稳定性。

参考文献

[1] 龙芯中科技术有限公司. 龙芯杯全国大学生系统能力培养大赛——实验平台开发手册. 2024.

[2] National Institute of Standards and Technology. *FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard*. August 2024.

[3] Xilinx Inc. *7 Series FPGAs Data Sheet: Overview (DS180)*. v2.6, 2020.

[4] Xilinx Inc. *Vivado Design Suite User Guide: Synthesis (UG901)*. v2019.2, 2019.

[5] ARM Ltd. *AMBA AXI and ACE Protocol Specification*. Issue H, 2013.