**专利申请技术交底书**

**1、发明（或实用新型—以下同）的名称**

用于雷达引导目标追踪的卡尔曼滤波专用计算结构

**2、所属技术领域**

本发明涉及雷达引导目标跟踪技术领域，具体涉及一种用于雷达引导目标追踪的卡尔曼滤波专用计算结构。

**3、现有技术**

在目标追踪领域，可见光传感器因其高图像分辨率和直观的信息获取能力，被广泛应用于目标检测与追踪。然而，可见光传感器的视场有限，难以在大范围内高效捕获和连续跟踪目标，导致实际应用效果不佳。因此，当前技术常采用雷达引导可见光传感器进行目标追踪。雷达具有高精度测距和大范围扫描能力，能够为可见光传感器提供准确的目标位置估计，从而实现更有效的目标捕获和跟踪。引导过程流程图见附图1。

在这一融合方案中，卡尔曼滤波算法被广泛用于目标状态估计与轨迹预测。特别是在引入匀加加速模型的情况下，卡尔曼滤波能够显著提高目标跟踪精度。然而，匀加加速模型的引入导致卡尔曼滤波过程中的矩阵维度增大，尤其是在计算卡尔曼增益 时的矩阵求逆运算 成为计算瓶颈，极大地限制了算法的实时性和高效性。具体而言，卡尔曼增益的计算公式为：。其中，矩阵求逆运算 的计算复杂度通常为 （ 为矩阵的维度）。在十二维状态空间下，该运算的计算量非常巨大。

针对这一关键瓶颈，发明人曾在一专利[申请号：202411854505.4]中提出了一种针对特定形式矩阵 的高效求逆方法。该方法通过利用矩阵的稀疏性和对称性，提取卡尔曼滤波中预测方程与更新方程的特征，提前计算矩阵 ，从而优化计算效率。具体而言，该方法将高维矩阵求逆转化为 复杂度的元素四则运算，显著降低计算复杂度，并具备并行计算潜力，为卡尔曼滤波的加速提供了有效途径。

目前，针对卡尔曼滤波加速的硬件实现主要基于FPGA等可重构硬件平台，但现有技术仍存在以下不足：

1. 通用性矩阵运算优化的局限性：

现有硬件专用计算结构通常采用LU分解或Cholesky分解来加速矩阵求逆。这些方法在一般场景下具有较高的计算效率，但在卡尔曼滤波应用中并不完全适用。卡尔曼滤波中的矩阵求逆涉及协方差矩阵，通常具有特定的对称正定结构以及数据相关性，通用求逆算法在处理这些特性时会导致额外的计算开销和资源浪费。

2. 并行性利用不足：

现有硬件专用计算结构设计在矩阵运算（如乘法、加法、求逆）方面未能充分挖掘操作的并行性。例如，主流方案多基于行列块分割策略，未能充分利用矩阵的稀疏性或对称性特性，导致不必要的计算开销。此外，部分硬件架构采用流水线设计，但流水线的深度和颗粒度往往不足，矩阵更新时多个流水线阶段存在依赖，导致吞吐量下降。

3. 实时性不足：

在高精度需求场景（如匀加加速模型下的目标追踪）中，卡尔曼滤波对实时性的要求极高。滤波过程需要在每次传感器数据更新时完成状态预测和更新，并与外部控制系统协同工作，确保实时输出。现有硬件架构因计算延迟高、数据传输开销大，难以满足实时性要求。例如，处理12×12矩阵的单次滤波更新执行时间可能达数十微秒，无法满足每秒1000次状态更新的实时性需求。

综上，现有技术在基于卡尔曼滤波的目标追踪加速方面存在计算瓶颈突出、架构未优化、实时性不足等问题，急需一种高效、专用的硬件加速解决方案。

4、**发明的目的**

针对现有技术中卡尔曼滤波在雷达引导目标追踪系统中存在的计算性能瓶颈、架构优化不足及实时性不高等问题，本发明旨在提供一种高效、专用的卡尔曼滤波专用计算结构。通过创新的硬件架构设计与整体流程优化，实现卡尔曼滤波算法的加速，满足高精度和高实时性目标追踪的需求。具体而言，本发明的目的和主要创新点包括：

1. 高效矩阵求逆单元的设计

针对卡尔曼增益计算过程中矩阵求逆的高计算复杂度问题，设计了一种专用的矩阵求逆硬件单元。该单元采用分块矩阵分解与并行运算的优化方法，充分利用矩阵的稀疏性和对称性，显著降低求逆运算的延迟，提高计算效率，从而突破传统求逆算法在高维矩阵下的性能瓶颈。

2. 模块化并行计算架构

采用模块化设计策略，将卡尔曼滤波中的矩阵乘法、加法及求逆等核心操作进行流水线化处理，并结合高并行度的处理架构。通过多模块并行运行和流水化设计，实现卡尔曼滤波全过程的高效硬件加速，适应高动态场景下的大规模矩阵运算需求，提升系统整体运算吞吐量。

3. 优化的数据传输机制

设计了一种高效的数据流传输机制，采用块内总线（OMBus）和模块间总线（MIBus）相结合的方式，优化模块内部和模块之间的数据传输路径。引入AXI总线协议，支持多通道并行通信和突发传输模式，显著减少数据传输延迟，提升整体系统的实时性，确保高频实时计算需求得到充分满足。

4. 增强系统灵活性与扩展性

通过可配置脉动阵列矩阵乘法单元和模块接口标准化设计，实现系统的高度灵活性与可扩展性。根据不同应用场景和矩阵维度需求，灵活调整计算模块的配置和数量，既能在资源充足的情况下实现多模块并行加速，也能在资源受限时通过流水化设计节约硬件资源，确保系统在多种环境下的高效运行。

5. 提升系统协同性与稳定性

基于有限状态机（FSM）的控制单元设计，实现各模块之间的高效协同工作。通过精确的状态调度和信号控制，确保模块间的数据依赖关系得到正确处理，避免数据竞争和错误操作顺序，提高系统的稳定性和可靠性。同时，支持错误检测与反馈机制，增强系统在复杂计算过程中的容错能力。

综上所述，本发明通过专用硬件架构设计和优化的计算流程，显著提升了卡尔曼滤波算法在雷达引导目标追踪系统中的计算性能和实时性。该卡尔曼滤波专用计算结构不仅有效解决了现有技术中的计算瓶颈和实时性不足问题，还通过模块化设计和优化的数据传输机制，提供了高效、灵活、可靠的硬件解决方案，为复杂环境下的高精度目标追踪应用提供了有力支持。

**5、发明的内容（与权利要求书相呼应）**

为实现上述目的，本发明提出一种用于雷达引导可见光目标追踪的卡尔曼滤波专用计算结构，通过专用硬件架构解决矩阵求逆瓶颈，提高目标跟踪系统的实时性与精度。

**5.1整体架构及工作流程**

本发明专用计算结构的整体硬件架构见附图2，主要包括以下几个模块：卡尔曼增益计算模块（包括卡尔曼增益计算A单元[即矩阵求逆单元]、卡尔曼增益计算B单元、卡尔曼增益乘法单元）、状态预测模块、状态更新模块、协方差更新模块、延时缓存单元和控制单元。这些模块在控制单元的统一调度下协同工作，完成卡尔曼滤波各个阶段的计算任务。

该专用计算结构的工作流程流程图见附图3。工作流程为：

数据采集：雷达与可见光传感器同步采集目标的位置信息和图像数据。采集到的数据首先被存储在输入数据缓存模块中，作为后续处理的基础输入。

状态预测：状态预测模块依据卡尔曼滤波的状态转移方程，对目标的当前状态进行预测。此步骤利用先验状态向量和状态转移矩阵，生成预测状态向量，为后续的状态更新提供初步估计。

卡尔曼增益计算：矩阵求逆优化模块采用本发明设计的高效硬件算法，快速执行矩阵 的求逆运算。通过优化的矩阵求逆单元，该模块显著加速了卡尔曼增益 的计算，解决了传统算法在高维矩阵下的计算瓶颈问题。

状态与协方差更新：更新计算模块基于最新的观测数据 ​ 和计算得到的卡尔曼增益 ​，对状态向量 和协方差矩阵 ​ 进行更新。具体计算公式为：、。通过本发明的硬件专用计算结构，此更新过程能够在高并行度和低延迟下高效完成，确保系统能够实时响应动态环境变化。

结果输出：经过滤波处理后的目标状态向量 ​ 和协方差矩阵 ​被输出至可见光传感器。滤波后的数据用于实时引导和持续追踪目标，确保目标检测的精度与稳定性。结果输出模块通过高效的数据传输机制，将处理结果快速反馈至传感器系统，实现闭环控制。

**5.2模块功能描述和硬件架构设计**

**5.2.1卡尔曼增益计算模块**

卡尔曼增益计算模块分为卡尔曼增益计算A单元（即矩阵求逆单元）、卡尔曼增益计算B单元和卡尔曼增益乘法单元。卡尔曼增益计算模块设计思想见附图4。将两部分分别计算可以充分利用并行潜力，优化计算效率。该模块硬件架构见附图5。

* 计算内容：
* 模块组成：
  + 卡尔曼增益计算A单元即**矩阵求逆计算单元**，计算 ；
  + 卡尔曼增益计算B单元计算 ；
  + 卡尔曼增益乘法单元将卡尔曼增益A单元和卡尔曼增益B单元的输出进行相乘。
* 输入分配：

逆矩阵 的部分通过模块间总线（MIBus）三通道输入，由CEU（Calculate Element Unit-元素计算单元）构成，分别对逆矩阵的不同分块进行计算。 部分通过模块间总线（MIBus）四通道输入，由CMU（3\*3 Calculate Matrix Unit-3\*3矩阵计算单元）计算，使用宽度为64×3位、深度为16的FIFO，满足数据的分流需求和存储调度效率。

**现有问题：**

现有卡尔曼滤波专用计算结构存在矩阵求逆计算瓶颈，尤其在高精度匀加加速模型下，计算复杂度进一步提升。针对这一问题，本发明设计了一种高效的矩阵求逆单元，专为匀加加速模型优化，通过并行计算与串行调度相结合的策略，显著降低求逆操作的延迟，提高增益矩阵计算的性能与效率。

**矩阵求逆方法：**

本发明通过利用矩阵的稀疏性和对称性，提取卡尔曼滤波中预测方程与更新方程的特征，提前计算矩阵 ，从而优化计算效率。具体而言，该方法将高维矩阵求逆转化为 复杂度的双精度浮点元素四则运算，显著降低计算复杂度，并具备并行计算潜力，为卡尔曼滤波的加速提供了有效途径。

矩阵 具有如下形式：

其逆矩阵为如下形式：

通过提前计算，本发明提出的矩阵求逆单元能够以更低的硬件资源和更快的速度完成卡尔曼增益的计算，显著提高了卡尔曼滤波的实时性。本发明基于以上基础，并进一步结合高并行硬件设计，提出一种专用的卡尔曼滤波矩阵求逆单元。硬件架构示意图见附图6。

**具体实现：**

* CEU\_abc单元：负责计算矩阵元素 a、b、c，通过并行计算最大化吞吐量。
* CEU\_def单元和CEU\_xyz单元：分别负责计算矩阵元素 d、e、f和x、y、z，延续CEU\_abc单元的设计思路，实现高效并行计算。

**硬件架构：**

* 模块化设计，分为多个计算执行单元（CEUs），以并行方式运行，实现数据流的高效传递与计算，减少计算延迟。
* 集成双精度浮点运算IP核，包括乘法、减法、除法和取反运算单元，确保计算精度；
* 通过多通道总线实现与其他单元的高效通信，支持9×64的数据并行通信能力。

**CEU\_abc单元、CEU\_def单元和CEU\_xyz单元：**

由于 d、e、f和x、y、z的计算方式与a、b、c采用相同的数学逻辑，硬件实现上延续了专用计算模块（CEU）的架构设计思路。**数学计算公式如下：**

卡尔曼增益计算模块采用并行计算与串行调度相结合的策略，优化计算性能并确保结果的准确性。对于相互独立的操作，设计允许同一阶段的并行计算，以最大化吞吐量；对于存在数据依赖关系的计算，采用串行调度，确保每个计算阶段在前一个阶段的数据依赖关系满足后顺利推进，保证数据一致性和计算顺序。

此外，本发明支持灵活的多模块配置。在硬件资源充足的情况下，特别是在FPGA资源丰富的环境中，可以部署多个相同的计算模块并行运行。例如，可以通过部署9个计算模块并行生成所需的矩阵元素a、b、c、d、e、f，从而显著提升计算效率和系统吞吐量。对于资源受限的情况，可以采用流水化设计，通过时间分片逐步生成计算结果，从而降低对硬件资源的需求，尤其是在双精度浮点乘法IP的使用上，同时仍能保持较高的计算效率。

本发明设计的矩阵求逆单元，针对卡尔曼滤波种，通过复用数学推导中的特定公式，将复杂的矩阵运算转化为多个简化的元素级运算，减少了计算复杂度，提高运算并行性和模块实现效率。

**5.2.2状态更新模块**

状态更新模块接收量测向量 ，输入为先验状态向量 ​ 和卡尔曼增益矩阵 ，并进行预测计算，输出为预测状态向量 ​（包括专用计算结构输出和模块间输出）,计算公式为：。

该模块硬件架构示意图见附图7，其中量测向量 通过系统总线输入到状态预测模块。先验状态向量 通过模块内总线（OMBus，On-Module Bus）从前一个模块传输到状态更新模块内部。卡尔曼增益矩阵 通过模块间总线（MIBus，Module Interconnect Bus）输入到状态更新模块内部。矩阵乘法使用可配置脉动阵列矩阵乘法单元。

**5.2.3状态预测模块**

状态预测模块，输入为预测状态向量 和状态转移矩阵 ​，完成状态向量的更新，计算公式为：。

该模块硬件架构示意图见附图8，其中状态转移矩阵 通过模块内总线（OMBus）传输到 状态更新模块内， 通过模块间总线（MIBus）到模块内，该模块计算矩阵相乘使用可配置脉动阵列单元，完成计算后，输出到宽度为64位，深度为12的FIFO里。

**5.2.4协方差更新模块**

该模块硬件架构示意图见附图9，协方差更新模块通过高效硬件设计实现了协方差矩阵 的更新，其核心计算公式为：。

模块分别计算了 和 。前者通过Transpose-Bridge Double-out Unit提供的 及 ，利用可配置脉动阵列矩阵乘法单元内完成矩阵乘法；后者采用同样的思路，同样通过 Transpose-Bridge Double-out Unit 提供的 及 ，利用 CMU实现并行计算，显著提升了效率。

在计算最终协方差矩阵 时，模块采用 12×12 的双精度浮点加法器阵列，对上述两项结果及常数矩阵 进行高效加法操作。最终更新结果通过 OMBus 输出，为后续模块提供支持。

**5.2.5可配置脉动阵列矩阵乘法单元**

在卡尔曼滤波的计算中，由于涉及大量的固定矩阵乘法操作（[12,12]\*[12,6]，[12,6]\*[6,6],[12,12]\*[12,1],[12,6]\*[6,12]），采用可配置脉动阵列可以有效节约硬件资源，提高计算效率。脉动阵列（Systolic Array）是一种高效的并行计算架构，尤其适用于矩阵运算。数据在脉动阵列的处理单元（Processing Elements, PE）之间按照固定的路径和时间顺序流动。每个PE负责简单的计算（如乘法和加法），并将中间结果传递给相邻的PE，实现流水线式的数据处理。

输入矩阵 A 的数据按行加载至寄存器，经延迟计数器逐步输入至各列的处理单元；矩阵 B 的数据则按列加载至顶部寄存器，并同步输入至各行的处理单元。每个 PE 接收来自左侧或上方的数据，完成乘法操作后将中间结果传递至右侧或下方，逐步完成矩阵乘法运算。最终结果通过输出通道收集，形成完整的结果矩阵。该架构通过灵活配置和高效数据流动，实现了低延迟、高精度的矩阵乘法计算，适配多种维度和应用场景需求。

PE单元是可配置脉动阵列矩阵乘法单元的核心计算单元，其内部结构包括输入通道、乘法器IP核、寄存器、加法器IP核、输出通道和控制单元。输入通道用于接收来自相邻单元或外部的数据，乘法器IP核执行输入数据的乘法运算并生成部分积，寄存器存储部分积以供后续加法使用。加法器IP核将部分积与上方单元的累加结果相加，生成新的累加结果，并通过输出通道传递给相邻PE单元。控制单元采用有限状态机（FSM）实现，通过使能信号（en）和复位信号（rst）协调各模块的工作，确保数据的正确流动与计算结果的准确性。

脉动阵列的 PE 单元并行执行乘加运算，通过流式数据传输和同步机制显著提升矩阵乘法效率。其灵活的配置能力允许根据矩阵规模启用或关闭部分PE单元，实现资源节约与功耗降低。该设计特别适用于卡尔曼滤波中固定矩阵乘法的需求，确保高效、实时计算。

**5.2.6延时缓存单元**

延时缓存单元用于临时存储中间计算结果，即 ​ 和 ​，提供下一轮迭代的输入，保证数据流的连续性。实际硬件架构分别利用宽度64位、深度12和宽度64位、深度144的FIFO实现。

**5.2.7控制单元：有限状态机（FSM）**

专用计算结构的控制单元基于有限状态机（FSM）设计，调度各个模块的工作顺序和数据流。FSM 的设计保证了多模块间的协调和计算流程的高效性。通过有限状态机的精确控制，各模块可以在适当的时候并行处理数据，同时又能保证数据依赖关系的正确性。例如，在状态预测模块进行计算的同时，卡尔曼增益计算模块的A单元可以进行部分准备工作，一旦状态预测完成，卡尔曼增益计算可以迅速进行，减少了整体计算时间。

同时，有限状态机确保了模块间的状态互锁，避免了数据竞争和错误的操作顺序。每个模块只有在接收到正确的控制信号时才会开始工作，保证了数据的一致性和系统的稳定性。例如，协方差更新模块不会在卡尔曼增益未计算完成时开始工作，防止了错误数据的引入。

**5.3模块之间的数据传输**

本发明提出了一种高效的通信机制，用于满足卡尔曼滤波专用计算结构各模块之间快速、可靠的数据交互需求。通信机制主要由块内总线（OMBus）和模块间总线（MIBus）构成。OMBus用于模块内部子单元之间的数据传输，其设计重点在于满足短距离、低延迟的传输需求。例如，在卡尔曼增益计算模块内部，子单元之间需要频繁地交换中间计算结果，OMBus通过高速、并行的数据传输机制，保障了模块内的计算流畅性与实时性。而MIBus是连接各模块的数据通道，支持模块间数据的高效交互。它采用全双工通信模式，能够实现数据的双向并行传输，从而显著提升通信效率，避免瓶颈问题。例如，状态预测模块计算的结果可以通过MIBus高效传输至卡尔曼增益计算模块，为后续运算提供输入。

控制信号传递由控制单元完成，包括初始化信号的分发、同步控制以及错误处理与反馈机制。通过精准的同步信号协调，确保了多模块并行计算的步调一致，同时在出现异常时，快速反馈错误信息并调整系统状态，保证系统稳定性。

此外，本发明的通信机制中采用了多通道通信机制与模块接口标准化设计。多通道缓冲传输和分层通信使得模块间能够并行处理大规模矩阵数据，从而显著提升了矩阵运算的效率。例如，通过将大矩阵分块传输至不同模块进行并行运算，不仅减少了传输延迟，还优化了计算性能。模块接口标准化则确保了各模块之间的兼容性与扩展性。每个模块都采用标准化的输入接口、输出接口和控制接口，无需改变通信架构即可灵活增加或替换模块。

为了支持上述通信机制，本发明采用了AXI总线协议作为核心通信协议。AXI协议支持突发传输模式，可一次性传输多个数据块，大幅提升传输效率。其流水线架构不仅实现了高带宽传输，还降低了通信延迟。此外，AXI协议的主从设备架构设计使得模块间能够灵活扩展，无论是模块内部子单元还是模块间通信，都可以方便地接入AXI总线，适配了本发明模块化架构的需求。

通过以上通信机制的设计，本发明在数据传输效率、模块化扩展能力以及实时性支持方面均有显著提升，为卡尔曼滤波专用计算结构的高性能计算提供了可靠保障。

**6、发明的效果**

与现有技术相比，本发明具有以下有益效果：

1. 提高计算速度：

通过解决卡尔曼滤波矩阵求逆的瓶颈以及使用可配置脉动阵列矩阵乘法单元和并行计算架构，将多个计算执行单元以流水化方式运行，对输入矩阵进行分块处理和并行计算，大大提高了卡尔曼滤波算法的计算速度。例如，在12×12矩阵运算场景中，求逆运算的延迟大大降低，能够满足雷达引导目标追踪的实时性要求。

2. 增强计算精度：

采用双精度浮点运算模块，确保在复杂的矩阵运算过程中，如矩阵求逆、矩阵乘法和矩阵加法等操作的高精度，提高了目标追踪的准确性。

3. 提高系统灵活性：

可配置脉动阵列矩阵乘法单元的可重构解码器和可动态调整的脉动阵列单元，能够根据不同的矩阵维度和计算需求灵活调整运算规模，适应多种应用场景和数据类型。同时，使用不同深度和宽度的存储中间结果和数据，保证了系统的灵活性和扩展性。

4. 高效的数据传输和通信：

利用专用的通信机制，包括块内总线（OMBus）和模块间总线（MIBus），并采用AXI总线协议，实现了模块内部和模块间的高效数据传输。支持多通道并行通信和模块接口标准化，保证了数据传输的高效性、可靠性和实时性，避免了数据传输的瓶颈问题。

5. 良好的系统协同性：

基于有限状态机的控制单元，通过不同的状态和相应的控制信号，精确调度各个模块的工作顺序和数据流，实现了模块间的高效协同工作。确保系统在复杂计算过程中的稳定性和可靠性，支持高效并行处理和顺序执行，提高了系统的整体性能。系统整体运算效率提升了1倍，显著提高了目标跟踪的实时性和稳定性。

**7、附图及附图的简要说明**

为了更清楚地说明本发明实施方式或现有技术中的技术方案,下面将对实施方式中所需要使用的附图作简单地介绍。显而易见地,下面描述中的附图仅仅是本发明的一些实施方式,对于本领域普通技术人员而言,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。附图仅用于示出具体实施例的目的,而并不认为是对本发明的限制,在整个附图中,相同的参考符号表示相同的部件。

图 1 为雷达引导可见传感器流程图

图 2 为卡尔曼滤波专用计算结构整体架构示意图

图 3 为卡尔曼滤波专用计算结构工作流程图

图 4 为卡尔曼增益计算模块计算方法示意图

图 5 为卡尔曼增益计算模块硬件架构示意图

图 6 为卡尔曼增益计算A单元硬件架构示意图

图 7 为状态更新模块硬件架构示意图

图 8 为状态预测模块硬件架构示意图

图 9 为状态预测模块硬件架构示意图

图 10 为CEU\_abc 单元硬件架构示意图

图 11 为可配置脉动阵列乘法单元架构示意图

图 12 为PE结构图

图 13 为PE单元有限状态机

图 14 为卡尔曼滤波专用计算结构控制单元有限状态机

**8、实施实例**

为使本发明实施例的目的、技术方案和优点更加清楚,下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例, 都属于本发明保护的范围。

**具体实施方式：**

**8.1整体架构**

本发明专用计算结构的整体架构如附图2所示，主要包括以下几个模块：卡尔曼增益计算模块（包括卡尔曼增益乘法单元、卡尔曼增益计算A单元[即矩阵求逆单元]、卡尔曼增益计算B单元）、状态预测模块、状态更新模块、协方差更新模块、延时缓存单元和控制单元。这些模块在控制单元的统一调度下协同工作，完成卡尔曼滤波各个阶段的计算任务。

**8.2模块功能描述和硬件架构设计**

**8.2.1卡尔曼增益计算模块**

卡尔曼增益计算模块分为卡尔曼增益计算A单元、卡尔曼增益计算B单元和卡尔曼增益乘法单元。将两部分分别计算可以充分利用并行潜力，优化计算效率。

* 计算内容：
* 模块组成：
  + 卡尔曼增益计算A单元即**矩阵求逆计算单元**，计算 ；
  + 卡尔曼增益计算B单元计算 ；
  + 卡尔曼增益乘法单元将卡尔曼增益A单元和卡尔曼增益B单元的输出进行相乘。
* 输入分配：

逆矩阵 的部分通过模块间总线（MIBus）三通道输入，由CEU（Calculate Element Unit-元素计算单元）构成，分别对逆矩阵的不同分块进行计算。 部分通过模块间总线（MIBus）四通道输入，由CMU（Calculate Matrix Unit-矩阵计算单元）计算，使用宽度为64×3位、深度为16的FIFO，满足数据的分流需求和存储调度效率。

CMU（Caculate Matrix Unit-矩阵计算单元）是3\*3维度的矩阵计算单元，用来计算，而 的结果是 的第1-6列的元素， 同样也要用于协方差更新，所以这里采用宽度为64\*3位，深度为16的FIFO，既可以得到本模块需要使用的 也可以得到协方差更新模块使用的。

**现有问题：**

现有卡尔曼滤波专用计算结构存在矩阵求逆计算瓶颈，尤其在高精度匀加加速模型下，计算复杂度进一步提升。针对这一问题，本发明设计了一种高效的矩阵求逆单元，专为匀加加速模型优化，通过并行计算与串行调度相结合的策略，显著降低求逆操作的延迟，提高增益矩阵计算的性能与效率。

**矩阵求逆方法：**

本发明通过利用矩阵的稀疏性和对称性，提取卡尔曼滤波中预测方程与更新方程的特征，提前计算矩阵 ，从而优化计算效率。具体而言，该方法将高维矩阵求逆转化为 复杂度的元素四则运算，显著降低计算复杂度，并具备并行计算潜力，为卡尔曼滤波的加速提供了有效途径。

矩阵 具有如下形式：

其逆矩阵为如下形式：

通过提前计算，本发明提出的矩阵求逆单元能够以更低的硬件资源和更快的速度完成卡尔曼增益的计算，显著提高了卡尔曼滤波的实时性。本发明基于以上基础，并进一步结合高并行硬件设计，提出一种专用的卡尔曼滤波矩阵求逆单元。

**具体实现：**

* CEU\_abc单元：负责计算矩阵元素 a、b、c，通过并行计算最大化吞吐量。CEU\_abc硬件架构示意图见附图10；
* CEU\_def单元和CEU\_xyz单元：分别负责计算矩阵元素 d、e、f和x、y、z，延续CEU\_abc单元的设计思路，，CEU\_def、CEU\_xyz采用和CEU\_abc类似的硬件架构,实现高效并行计算。

**硬件架构：**

该模块的硬件架构示意图见附图6，展示了各个组成部分及其协同工作方式。

* 模块化设计，分为多个计算执行单元（CEUs），以并行方式运行，实现数据流的高效传递与计算，减少计算延迟。
* 集成双精度浮点运算模块，包括乘法、减法、除法和取反运算单元，确保计算精度。
* 通过多通道总线实现与其他单元的高效通信，支持9×64的数据并行通信能力。

**CEU\_abc、CEU\_def单元和CEU\_xyz单元：**

由于 d、e、f和x、y、z的计算方式与a、b、c采用相同的数学逻辑，硬件实现上延续了专用计算模块（CEU）的架构设计思路。**数学计算公式如下：**

卡尔曼增益计算模块采用并行计算与串行调度相结合的策略，优化计算性能并确保结果的准确性。对于相互独立的操作，设计允许同一阶段的并行计算，以最大化吞吐量；对于存在数据依赖关系的计算，采用串行调度，确保每个计算阶段在前一个阶段的数据依赖关系满足后顺利推进，保证数据一致性和计算顺序。

此外，本发明支持灵活的多模块配置。在硬件资源充足的情况下，特别是在FPGA资源丰富的环境中，可以部署多个相同的计算模块并行运行。例如，可以通过部署9个计算模块并行生成所需的矩阵元素a、b、c、d、e、f，从而显著提升计算效率和系统吞吐量。对于资源受限的情况，我们采用流水线设计，通过时间分片逐步生成计算结果，从而降低对硬件资源的需求，尤其是在双精度浮点乘法IP的使用上，同时仍能保持较高的计算效率。

本发明设计的矩阵求逆单元，针对卡尔曼滤波，通过复用数学推导中的特定公式，将复杂的矩阵运算转化为多个简化的元素级运算，减少计算复杂度，提高运算并行性和模块实现效率。

**8.2.2状态更新模块**

状态预测模块接收量测向量 ，输入为先验状态向量 ​ 和卡尔曼增益矩阵 ，并进行预测计算，输出为预测状态向量 ​（包括专用计算结构输出和模块间输出），计算公式为：。

**硬件架构：**

硬件架构示意图见附图7。

* 量测数据输入：量测向量 通过总线输入到状态预测模块。
* 数据传输：
  1. 先验状态向量 通过模块内总线（OMBus）传输到状态更新模块内部
  2. 卡尔曼增益矩阵 通过模块间总线（MIBus）输入到状态更新模块内部。
* 计算单元：
  1. 使用MUX选择器获取 （ 本质上是一个矩阵元素选择的过程）。
  2. 通过Neg取反器和加法器计算 。
  3. 使用可配置脉动阵列矩阵乘法单元计算 。
  4. 通过12个双精度浮点加法器计算 ，得到预测状态向量 。

**8.2.3状态预测模块**

状态更新模块，输入为预测状态向量 和状态转移矩阵 ​，完成状态向量的更新，计算公式为：。

状态转移矩阵 通过模块内总线（OMBus）传输到 状态更新模块内， 通过模块间总线（MIBus）到模块内，该模块计算矩阵相乘使用可配置脉动阵列单元。完成计算后，输出到宽度为64位，深度为12的FIFO里。

**硬件架构：**

硬件架构示意图见附图8。

* 状态转移矩阵 通过模块内总线（OMBus）传输到状态预测模块。
* 使用可配置脉动阵列单元进行矩阵乘法计算。
* 结果通过宽度为64位、深度为12的FIFO存储，为后续模块提供支持。

**8.2.4协方差更新模块**

协方差更新模块通过高效硬件设计实现了协方差矩阵 的更新，优化了数据传输与计算流程，整体架构计算并行度高，。其核心计算公式为：。

**硬件架构：**

硬件架构图示意图见附图9。

* Transpose-Bridge Double-out Unit

该单元负责矩阵 的数据格式转换，可同时输出矩阵 及其转置 。这种设计通过并行输出减少了串行数据处理的延迟，提高了计算效率。此外，利用存储资源换取时间的优化策略，为后续矩阵计算提供了快速输入支持。

* 矩阵计算单元（CMU）与 FIFO 设计

矩阵计算单元（Caculate Matrix Unit, CMU）为固定 3×3 维度的模块，专用于高效处理小型矩阵乘法任务，如计算 和 。由于 仅涉及 的前 1-6 列，而 用于协方差更新，因此设计了宽度为 64×3 位、深度为 16 的 FIFO，既满足了数据的分流需求，也提升了存储和调度效率。

FIFO 在存储和传输中间结果的同时，支持选择性读取模式，以满足并行计算对数据输入的需求，避免了冗余存取操作。

* 状态转移矩阵 的优化计算

针对矩阵 的计算，模块并未采用传统矩阵乘法和加法的方案，而是结合矩阵的稀疏性和特殊结构，直接通过选择器（Mux）和取反器（Negation Unit）对矩阵元素进行操作，并将结果动态组合成完整矩阵。这种设计显著降低了计算复杂度和资源开销。

* 矩阵运算与最终结果生成

模块分别计算了 和 。前者通过 Transpose-Bridge Double-out Unit 提供的 及 ，利用可配置阵列矩阵乘法单元内完成矩阵乘法；后者采用同样的思路，同样通过 Transpose-Bridge Double-out Unit 提供的 及 ，利用 CMU实现并行计算，显著提升了效率。

在计算最终协方差矩阵 时，模块采用 12×12 的双精度浮点加法器阵列，对上述两项结果及常数矩阵 进行高效加法操作。最终更新结果通过 OMBus 输出，为后续模块提供支持。

**8.2.5可配置脉动阵列矩阵乘法单元**

在卡尔曼滤波的计算中，由于涉及大量的固定矩阵乘法操作（[12,12]\*[12,12]，[12,6]\*[6,6],[12,12]\*[12,1],[12,6]\*[6,1], [12,6]\*[6,12]，[12,6]\*[6,6]），采用可配置脉动阵列可以有效节约硬件资源，提高计算效率。脉动阵列（Systolic Array）是一种高效的并行计算架构，尤其适用于矩阵运算。数据在脉动阵列的处理单元（Processing Elements,PE）之间按照固定的路径和时间顺序流动。每个PE负责简单的计算（如乘法和加法），并将中间结果传递给相邻的PE，实现流水线式的数据处理。

输入矩阵 A 的数据按行加载至寄存器，经延迟计数器逐步输入至各列的处理单元；矩阵 B 的数据则按列加载至顶部寄存器，并同步输入各行的处理单元。每个 PE 接收来自左侧或上方的数据，完成乘法操作后将中间结果传递至右侧或下方，逐步完成矩阵乘法运算。最终结果通过输出通道收集，形成完整的结果矩阵。该架构通过灵活配置和高效数据流动，实现了低延迟、高精度的矩阵乘法计算，适配多种维度和应用场景需求。

PE单元是可配置脉动阵列矩阵乘法单元的核心计算单元，其内部结构包括输入通道、乘法器IP核、寄存器、加法器IP核、输出通道和控制逻辑（FSM Control）。输入通道用于接收来自相邻单元或外部的数据，乘法器IP核执行输入数据的乘法运算并生成部分积，寄存器存储部分积以供后续加法使用。加法器IP核将部分积与上方单元的累加结果相加，生成新的累加结果，并通过输出通道传递给相邻PE单元。控制逻辑采用有限状态机（FSM）实现，通过使能信号（en）和复位信号（rst）协调各模块的工作，确保数据的正确流动与计算结果的准确性。

脉动阵列的 PE 单元并行执行乘加运算，通过流式数据传输和同步机制显著提升矩阵乘法效率。其灵活的配置能力允许根据矩阵规模启用或关闭部分PE单元，实现资源节约与功耗降低。该设计特别适用于卡尔曼滤波中固定矩阵乘法的需求，确保高效、实时计算。

**硬件架构：**

硬件架构示意图见附图11。

* 输入寄存器：存储矩阵 A 和 B 的数据。
* 可重构解码器：根据矩阵维度动态配置计算规模，通过控制信号（如 enb\_1、enb\_2\_6 等）选择启用的处理单元（Processing Elements，PE）列，以优化资源利用率和运算效率。
* 延迟计数器：精确控制数据在处理单元间的传递顺序，确保计算过程的同步性。
* 处理单元（PE）：核心计算单元，执行乘法累加（MAC）操作。
* 输出通道：输出计算结果至目标模块或存储设备。

**工作原理：**

* 输入矩阵A的数据按行加载至寄存器，经延迟计数器逐步输入至各列的处理单元。
* 矩阵B的数据按列加载至顶部寄存器，并同步输入各行的处理单元。
* 每个 PE 接收来自左侧或上方的数据，完成乘法操作后将中间结果传递至右侧或下方，逐步完成矩阵乘法运算。最终结果通过输出通道收集，形成完整的结果矩阵 。

**PE单元结构：**

硬件架构示意图见附图12。

* 输入通道：接收来自相邻单元或外部的数据。
* 乘法器 IP 核：执行输入数据的乘法运算，生成部分积。
* 寄存器：存储部分积以供后续加法使用。
* 加法器 IP 核：将部分积与上方单元的累加结果相加，生成新的累加结果。输出通道：传递累加结果至相邻 PE 单元。
* 控制单元：基于有限状态机（FSM）设计，通过使能信号（en）和复位信号（rst）协调各模块工作，确保数据的正确流动与计算结果的准确性。

**FSM 状态：**

PE单元有限状态机见附图13。

* IDLE（空闲）：待命和等待新任务的启动。
* INIT（初始化）：加载初始数据和配置控制信号。
* Mul（乘法）：执行乘法运算并生成部分积。
* Add（加法）：将部分积与累加结果相加。
* Send\_data（发送数据）：将计算结果发送至下一个 PE 单元或输出通道。
* Data\_Through（数据直通）：在特定条件下直接传递数据，无需进一步处理。End（结束）：完成当前运算任务并准备返回空闲状态。

**8.2.6延时缓存单元**

延时缓存单元用于临时存储中间计算结果，即 ​ 和 ​，提供下一轮迭代的输入，保证数据流的连续性。实际硬件架构分别利用宽度64位、深度12和宽度64位、深度144的FIFO实现。

**8.2.7控制单元：有限状态机（FSM）**

专用计算结构的控制单元基于有限状态机（FSM）设计，有限状态机见附图14，负责调度各个模块的工作顺序和数据流。FSM 的设计保证了多模块间的协调和计算流程的高效性。通过有限状态机的精确控制，各模块可以在适当的时候并行处理数据，同时又能保证数据依赖关系的正确性。例如，在状态预测模块进行计算的同时，卡尔曼增益计算模块的A单元可以进行部分准备工作，一旦状态预测完成，卡尔曼增益计算可以迅速进行，减少了整体计算时间。

同时，有限状态机确保了模块间的状态互锁，避免了数据竞争和错误的操作顺序。每个模块只有在接收到正确的控制信号时才会开始工作，保证了数据的一致性和系统的稳定性。例如，协方差更新模块不会在卡尔曼增益未计算完成时开始工作，防止了错误数据的引入。

**FSM 主要状态：**

* 初始化（Init）：完成系统寄存器清零和缓存清空操作；
* 状态预测（State\_Prediction）：根据先验状态和卡尔曼增益矩阵计算当前状态；
* 卡尔曼增益计算（Cal\_Kalman\_Gain）：执行逆矩阵计算、矩阵乘法和增益更新；
* 状态-协方差更新（State\_Covariance\_Update）：完成协方差矩阵的更新；
* 测量数据输入（Measure\_Data\_Input）：接收新数据，准备下一轮迭代；
* 状态与协方差输出（State\_Covariance\_Output）：将计算结果提供给外部；
* 结束（End）：进入待机模式，准备下一任务。

**状态转换条件：**

* Init → State\_Prediction：接收到初始化完成信号（Init\_Valid）；
* State\_Prediction → Cal\_Kalman\_Gain：状态预测完成信号（SP\_Done）；
* Cal\_Kalman\_Gain → State\_Covariance\_Update：卡尔曼增益计算完成信号（CKG\_Done）；
* State\_Covariance\_Update → Measure\_Data\_Input：状态-协方差更新完成信号（SCU\_Done）；
* Measure\_Data\_Input → State\_Prediction：测量数据输入完成信号（MDI\_Valid）；
* State\_Covariance\_Update → State\_Covariance\_Output：协方差更新完成信号（SCO\_Valid）；
* State\_Covariance\_Output → End：输出完成信号（End\_Valid）；
* End → Init：准备进入下一轮任务。

**8.3模块之间的数据传输**

本发明提出了一种高效的通信机制，用于满足卡尔曼滤波专用计算结构各模块之间快速、可靠的数据交互需求。通信机制由块内总线（OMBus）和模块间总线（MIBus）构成。OMBus用于模块内部子单元之间的数据传输，其设计重点在于满足短距离、低延迟的传输需求。例如，在卡尔曼增益计算模块内部，子单元之间需要频繁地交换中间计算结果，OMBus通过高速、并行的数据传输机制，保障了模块内的计算流畅性与实时性。而MIBus是连接各模块的数据通道，支持模块间数据的高效交互。它采用全双工通信模式，能够实现数据的双向并行传输，从而显著提升通信效率，避免瓶颈问题。MIBus还具备高带宽与多优先级调度能力，能够根据实时性需求动态分配带宽，以保证关键数据的优先传输。例如，状态预测模块计算的结果可以通过MIBus高效传输至卡尔曼增益计算模块，为后续运算提供输入。

控制信号传递由中央控制逻辑单元完成，包括初始化信号的分发、同步控制以及错误处理与反馈机制。通过精准的同步信号协调，确保了多模块并行计算的步调一致，同时在出现异常时，快速反馈错误信息并调整系统状态，保证系统稳定性。

此外，本发明的通信机制中采用了多通道通信机制与模块接口标准化设计。多通道缓冲传输和分层通信使得模块间能够并行处理大规模矩阵数据，从而显著提升了矩阵运算的效率。例如，通过将大矩阵分块传输至不同模块进行并行运算，不仅减少了传输延迟，还优化了计算性能。模块接口标准化则确保了各模块之间的兼容性与扩展性。每个模块都采用标准化的输入接口、输出接口和控制接口，无需改变通信架构即可灵活增加或替换模块。

为了支持上述通信机制，本发明采用了AXI总线协议作为核心通信协议。AXI协议支持突发传输模式，可一次性传输多个数据块，大幅提升传输效率。其流水线架构不仅实现了高带宽传输，还降低了通信延迟。此外，AXI协议的主从设备架构设计使得模块间能够灵活扩展，无论是模块内部子单元还是模块间通信，都可以方便地接入AXI总线，适配了本发明模块化架构的需求。