

硬核处理器系统 (HPS) SDRAM 控制器子系统为 ARM® Cortex™ A9 微处理器单元 (MPU) 子系统的外部 SDRAM、level 3 (L3) 互联和 FPGA 架构提供有效访问。SDRAM 控制器提供 FPGA 架构和 HPS 之间的接口。接口接受高级微控制器总线体系结构 (AMBA®) 高级可扩展接口 (AXI™) 和 Avalon® 存储器映射的 (Avalon-MM) 传输，将那些命令转换成 SDRAM 的正确命令，并且管理 SDRAM 访问的详细信息。

SDRAM 控制器子系统的功能

SDRAM 控制器子系统具有以下功能：

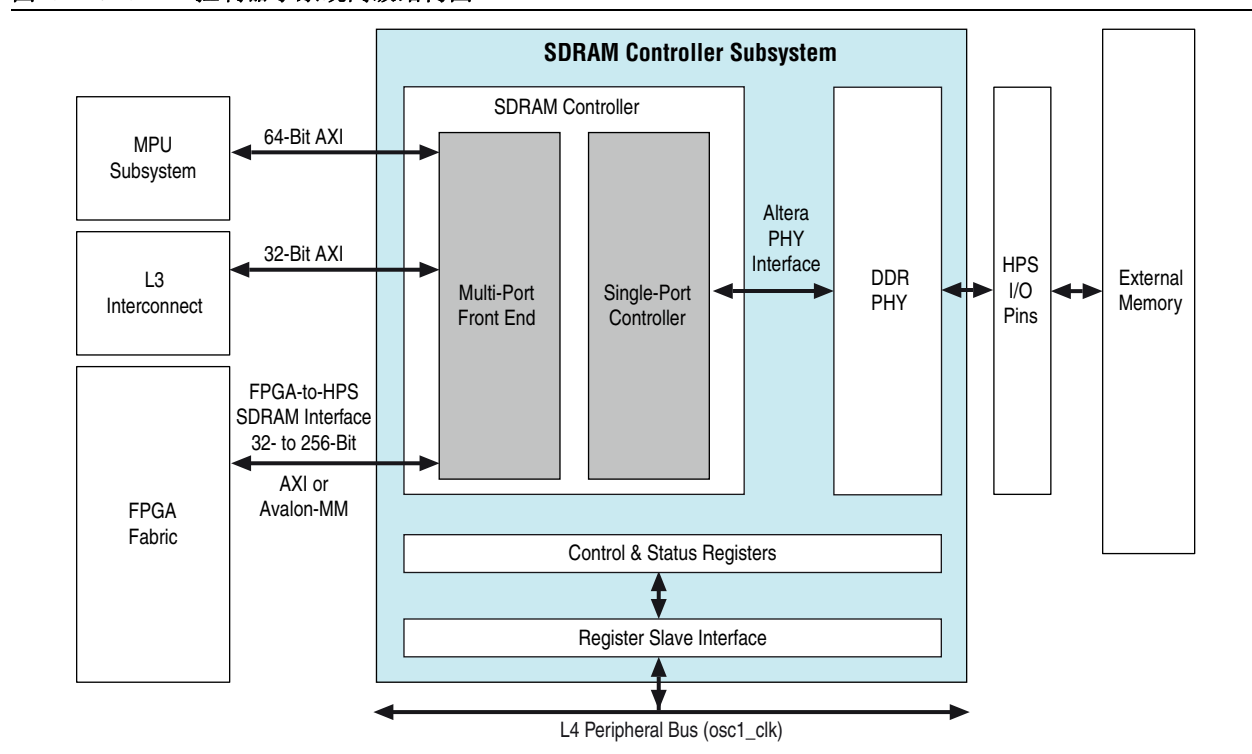
- 支持双数据速率 2 (DDR2)、DDR3 和低功耗 DDR2 (LPDDR2) SDRAM
- 用户可配置的时序参数
- 高达 4 Gb 密度部件
- 两个片选
- 集成的错误纠正代码 (ECC)、24- 和 40-bit 宽度
- 用户可配置的存储器宽度为 8、16、16+ECC、32、32+ECC
- 命令重新排序 (look-ahead bank 管理)
- 数据重新排序 (无序传输)
- 对于关闭页或有条件打开页访问，基于每端口的用户可控制的 bank 原则
- 具有绝对和相对优先级调度的用户可配置的优先级支持
- 通过使用 Avalon-MM 和 AXI 接口，高达 6 个端口和 256 位宽的数据宽度的灵活的 FPGA 架构接口配置
- 功耗管理支持自刷新、部分阵列自刷新 (PASR)、关断和 LPDDR2 深度关断

SDRAM 控制器子系统结构图和系统集成

SDRAM 控制器子系统连接 MPU 子系统、L3 互联的主开关和 FPGA 架构。存储器接口包含 SDRAM 控制器、物理层 (PHY)、控制和状态寄存器 (CSR) 以及它们相关的接口。

图 8-1 显示了 SDRAM 控制器子系统的高级结构图。

图 8-1. SDRAM 控制器子系统高级结构图



SDRAM 控制器

SDRAM 控制器提供高性能数据访问和运行时可编程性。通过将读和写处理组合在一起，控制器重新排序数据以减少行冲突和总线转换时间，从而支持有效传输模式减小的延迟。

SDRAM 控制器包含一个多端口前端（MPFE）和一个单端口控制器。MPFE 对单端口控制器提供多个独立的接口。单端口控制器与每个外部存储器件通信并且管理它们。要了解更多信息，请参考第 8-4 页的“存储控制器体系结构”。

DDR PHY

DDR PHY 在存储控制器和存储器件之间提供了一个物理层接口，它可以执行读和写存储操作。DDR PHY 具有数据流组件、控制组件和处理 SDRAM 接口时序校准的校准逻辑。

SDRAM 控制器子系统接口

以下的部分介绍 SDRAM 控制器子系统接口。

MPU 子系统接口

SDRAM 控制器通过专用的 64-bit AXI 接口连接到 MPU 子系统，并且在 mpu_l2_ram_clk 时钟域上操作。

L3 互联接口

SDRAM 控制器使用专用 32-bit AXI 接口连接到 L3 互联，并且在 l3_main_clk 时钟域上操作。

CSR 接口

CSR 接口与 level 4 (L4) 总线连接并且在 l4_sp_clk 时钟域上操作。MPU 子系统使用 CSR 接口配置控制器和 PHY，例如，设置存储器时序参数值或使存储器处于低功耗状态。CSR 接口也对控制器和 PHY 中的状态寄存器提供访问。

FPGA-to-HPS SDRAM 接口

FPGA-to-HPS SDRAM 接口提供 FPGA 架构中实现的主器件对 HPS 中的 SDRAM 控制器子系统的访问。接口具有 3 个端口类型，用于构建以下的 AXI 或 Avalon-MM 接口：

- 命令端口 — 发出读和写命令，并且用于接收写确认响应
- 64-bit 读数据端口 — 接收从存储器读返回的数据
- 64-bit 写数据端口 — 发送写数据

FPGA-to-HPS SDRAM 接口支持 6 个命令端口，对于 Avalon-MM 接口可以支持 6 个或对于 AXI 接口可以支持 3 个。每个命令端口可用于实现 AXI 的读或写命令端口，或用作 Avalon-MM 接口的一部分。AXI 和 Avalon-MM 接口可以配置成支持 32-、64-、128- 和 256-bit 数据。

表 8-1 列出了与 FPGA 连接的 FPGA-to-HPS SDRAM 控制器接口端口。

表 8-1. FPGA-to-HPS SDRAM 控制器端口类型

端口类型	数量
命令	6
64-bit 读数据	4
64-bit 写数据	4

FPGA-to-HPS SDRAM 控制器接口的配置具有以下特征：

- 在对架构所提供的端口的限定数内，Avalon-MM 接口和 AXI 接口可以根据架构逻辑的要求混合并且匹配。
- FPGA-to-HPS SDRAM 接口的每个 Avalon-MM 或 AXI 接口在一个独立的时钟域上操作。
- FPGA-to-HPS SDRAM 接口在 FPGA 配置期间进行配置。

表 8-2 显示了配置不同的总线协议所需的基于类型和数据宽度的端口数。

表 8-2. FPGA-to-HPS SDRAM 端口使用 (1/2)

总线协议	命令	读数据	写数据
32- 或 64-bit AXI	2 (1)	1	1
128-bit AXI	2 (1)	2 (2)	2 (2)
256-bit AXI	2 (1)	4 (2)	4 (2)
32- 或 64-bit Avalon-MM	1	1	1
128-bit Avalon-MM	1	2	2
256-bit Avalon-MM	1	4	4

表 8-2. FPGA-to-HPS SDRAM 端口使用 (2/2)

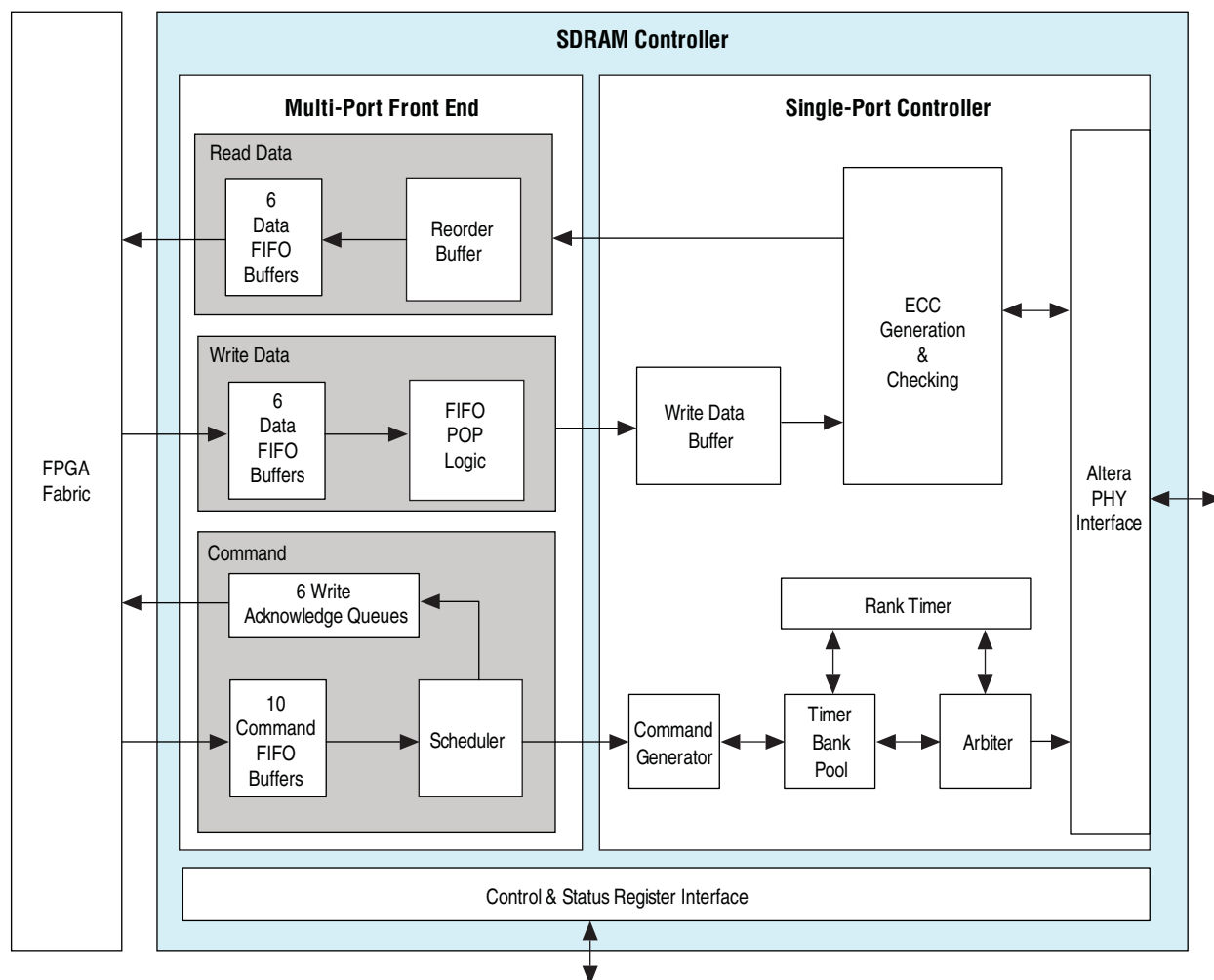
总线协议	命令	读数据	写数据
32- 或 64-bit Avalon-MM 只写 (write-only)	1	0	1
128-bit Avalon-MM 只写	1	0	2
256-bit Avalon-MM 只写	1	0	4
32- 或 64-bit Avalon-MM 只读	1	1	0
128-bit Avalon-MM 只读	1	2	0
256-bit Avalon-MM 只读	1	4	0
表 8-2 注释： (1) 因为 AXI 协议支持同时发出读写命令，所以需要两个 SDRAM 控制端口以便形成一个 AXI 接口。 (2) 因为数据端口的原始尺寸是 64 位，所以需要额外的读写端口以便形成一个 AXI 接口。			

存储控制器体系结构

SDRAM 控制器包含一个 MPFE、一个单端口控制器和一个到 CSR 的接口。

图 8-2 显示了 SDRAM 控制子系统的 SDRAM 控制器部分的结构图。

图 8-2. SDRAM 控制器结构图



MPFE

MPFE 负责规划来自配置接口的待定处理并且将规划的存储器处理发送到单端口控制器。MPFE 处理与单独的端口相关的所有功能。

MPFE 包含以下 3 个主要子模块。

命令模块

命令模块从 FPGA 架构和 HPS 接受读写传输。当命令 FIFO 缓冲器满时，命令模块通过让 ready 信号无效来实施反压。对于每个待定处理，命令模块计算在该传输上所需进行的下一个 SDRAM 突发。命令模块基于用户提供的配置、可用写数据和未分配的读数据空间规划待定 SDRAM 突发命令。

写数据模块

写数据模块将数据发送到单端口控制器。写数据模块对写数据保持写数据 FIFO 缓冲器和跨时钟域处理。写数据模块通知命令模块每个处理的待定写数据量，以便命令模块可以计算下一个 SDRAM 写突发的适用性。

读数据模块

读数据模块从单端口控制器接收数据。根据端口状态，读数据模块在其内部缓冲器中缓冲数据或将数据直接传递到时钟边界交叉 FIFO 缓冲器。读数据模块重新排序 Avalon-MM 端口的无序数据。

为了防止读 FIFO 缓冲器上溢，读数据模块通知命令模块可用缓冲器区域，以便命令模块的接收可以与读处理调度保持同步。

单端口控制器

单端口逻辑负责进行以下操作：

- 排列待定 SDRAM 突发
- 选择最有效的下一个待发的突发
- 保持 SDRAM 流水线满
- 确保满足所有 SDRAM 时序参数

传递到单端口逻辑的 SDRAM 中的单页面事物处理被保证按顺序执行，但是处理可以在页面之间被重新排序。每个 SDRAM 突发读或写都被转换成相应的 Altera PHY 接口 (AFI) 命令以便打开一个 bank 并在正确的行上进行处理（如果需要），执行读或写命令并且预充电 bank（如果需要）。

单端口逻辑实现命令重新排序（预见命令序列来查看使哪一个 bank 处于正确的状态以便使得读或写命令能够被执行）以及数据重新排序（即使数据传输以不同于从多端口逻辑接收它们的顺序被执行，但它使得数据传输能够被分配）。

命令生成器

命令生成器接受 MPFE 和内部 ECC 逻辑的命令，并且对 timer bank pool 提供这些命令。

Timer Bank Pool

timer bank pool 是一个并行队列，它和仲裁器一并进行操作以便让数据重新排序。timer bank pool 跟踪输入请求，确保满足所有的时序要求，并且从写数据缓冲器接收 write-data-ready 提示时，将请求传递到仲裁器。

仲裁器

仲裁器决定请求被传递到存储器件的顺序。当仲裁器接收到一个单个请求时，请求立即被传递。当接收到多个请求时，仲裁器使用仲裁规则以便决定将请求传递到存储器件的顺序。

Rank Timer

rank timer 执行以下的功能：

- 保持 rank 指定的时序信息

- 确保只有 4 个激活在指定的时序窗口内发生。
- 管理 read-to-write 和 write-to-read 总线转换时间
- 管理不同的 bank 之间的 time-to-activate 延迟

写数据缓冲器

写请求信号被置位时，写数据缓冲器接收 MPFE 的写数据并且将数据传递到 PHY。

ECC 模块

ECC 模块包含一个编码器和一个解码器 - 校正器 (decoder-corrector)，它可以检测到并且纠正单比特错误，以及检测到双比特错误。对于数据传输期间噪声或其它的损坏导致的单比特和双比特错误，ECC 模块可以纠正单比特错误并且检测到双比特错误。

AFI 接口

AFI 接口提供控制器和 PHY 之间的通信。

CSR 接口

CSR 接口可从 L4 总线访问。该接口使得 HPS MPU 和 FPGA 架构中执行的代码能够配置并且监控 SDRAM 控制器。

SDRAM 控制器子系统的功能说明

该部分提供 SDRAM 控制器子系统的功能说明。

MPFE 操作行为

该部分介绍 MPFE 的操作行为。

操作顺序

到达给定端口的相同 SDRAM 页面的请求以它们接收到的顺序被执行。除非第一个事务处理在第二个事务处理到达之前完成，否则到达不同端口的请求没有确定的执行顺序。

操作顺序在端口内定义并且执行，但不是在端口之间。单端口接收到的重叠地址的所有传输按顺序执行。不同端口接收到的传输没有保证的顺序，除非第二个传输在第一个传输完成之后出现。

Avalon-MM 不支持写确认。当一个端口被配置成支持 Avalon-MM 时，您应该从之前被写入的位置读取以便确保写操作已经完成。当端口被配置成支持 AXI 时，一旦写操作被确认，访问端口的主器件就可以安全地发出和写操作相同地址的读操作。要保持写延迟低，传输顺序一旦被保证，写操作就会被确认 — 意味着所有端口接收到的和写操作相同地址的所有操作都会在写操作后被执行。

要确保整个传输的延迟尽量低，那么当传输返回无序时，单端口逻辑可以将无序读数据返回到可以将其重新排序的多端口逻辑。大比例的传输将在端口之间重新排序并且传输仅在端口内排序。对于在端口之间而不是在端口内重新排序的传输，不需要进行任何的重新排序。消除不必要的重新排序会减少平均延迟。

多端口调度

多端口调度取决于两个因素，请求的绝对优先权和端口的权重。

绝对优先权的评估确保载有较高优先权传输的端口在载有较低优先权传输的端口之前被执行。调度器识别 8 个优先权等级 (0-7)，较高值代表较高优先权。例如，优先权等级 7 的所有传输在优先权等级 6 或低于 6 的传输之前被调度。

当端口载有相同绝对优先权的传输时，相对优先权根据端口权重而被决定。端口权重是一个 5 位值 (0-31)，并且由差值加权轮询 (deficit-weighted round robin, DWRR) 算法决定，它纠正过多执行的端口传输或过少执行的端口传输。每个端口都具有一个每周期被更新的相关权重，会加上用户配置的权重并且减去执行的传输量。最高权重的端口被认为是可用性最大的。

要确保高优先权传输快速的执行并且长和短突发在端口之间有效地交错，那么长于单一 SDRAM 突发的输入传输被调度为一系列的 SDRAM 突发，并且每个突发被单独地仲裁。

要确保当较高优先权端口垄断带宽时，较低优先权端口不会堆积大量运行的权重，那么控制器的 DWRR 权重只有当端口匹配调度的优先权时才被更新。因此，如果访问 3 个端口，2 个是优先权等级 7 并且 1 个是优先权等级 4，那么优先权等级为 7 的 2 个端口的权重被更新但是优先权等级为 4 的端口保持不变。

多端口调度在与 FPGA 架构和 HPS 内部连接的所有端口之间执行，以便决定下一个要执行哪一个传输。仲裁在 SDRAM 突发基础上执行以确保一个长传输不会锁定其它传输或导致高优先权端口的延迟明显地增加。

仲裁支持绝对和相对优先权。绝对优先权用于一个主器件总是高于或低于其它器件优先权的应用。相对优先权通过可编程的权重域支持，从而控制相同优先权的端口之间的调度。

调度器是连续工作的。只有当 SDRAM 突发的足够数据被接收到时，写操作才可以被调度。只有当足够内部存储器空余并且端口没有占用很多读缓冲器时，读操作才可以被调度。

传输进行时，多端口调度配置可以被更新。端口的优先权和权重都可以被更新而不会中断端口的传输。更新用于被更新的 10 个存储器时钟周期内的调度决策，所以可以根据需要频繁地更新优先权。

读数据处理

MPFE 包含一个所有端口共享的读缓冲器。如果端口能够接收返回的数据，那么读缓冲器被旁路。如果读传输的容量小于存储器接口宽度的两倍，那么缓冲器 RAM 不能够被旁路。

SDRAM 突发调度

SDRAM 突发调度识别访问相同 row/bank 组合 (称为打开页面访问) 的地址。页面的操作以单端口控制器接收它们的顺序被执行。

SDRAM 操作的选择的过程分为 2 个阶段。首先，每个待定传输必须等待其计时器适合执行。其次，传输对其它也适合执行的传输赢取仲裁。

以下规则控制传输仲裁：

- 高优先权操作优先于较低优先权操作
- 如果多个操作处于仲裁，那么读操作优先于写操作。
- 如果多个操作仍然存在，那么最早的操作被第一个执行。

如果 bank 闲置并且地址的高优先权传输的片选、行或列域不匹配单端口控制器中已存在的地址，那么该 bank 的 SDRAM 突发调度器中的高优先权传输立即赢得仲裁。如果 bank 没有闲置，那么该 bank 的其它操作直到高优先权操作完成才产生。如果片选、行和列域匹配一个较早传输，那么高优先权传输直到较早传输完成才产生。

时钟

MPFE 的 FPGA 架构端口可以在不同的频率被提供时钟。同步由 MPFE 中时钟域交叉逻辑保持。命令端口可以在不同的时钟域上操作，但是与给定的命令端口相关的数据端口必须连接与该命令端口相同的时钟。例如，如果命令端口与读和写端口成对以形成一个 Avalon-MM 接口，那么它必须以与其相关的数据端口的相同时钟频率操作。

单端口控制器操作行为

该部分介绍单端口控制器的操作行为。

SDRAM 接口

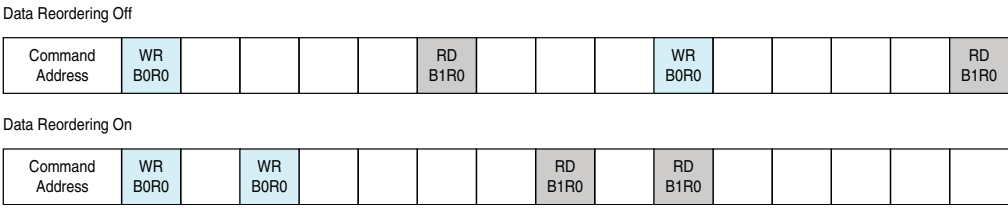
SDRAM 接口高达 40 bit 宽并且可以适应 8-bit、16-bit、16-bit plus ECC、32-bit 或 32-bit plus ECC 配置。SDRAM 接口支持 LPDDR2、DDR2 和 DDR3 存储器协议。

命令和数据重新排序

SDRAM 控制器的核心是一个命令和数据重新排序引擎。命令重新排序使得目前传输完成之前新的传输要进行的 bank 能够被打开。当为了提高 SDRAM 带宽的使用率而允许新顺序时，数据重新排序使得传输能够以接收它们的不同的顺序进行。相同 bank 和行的操作被执行以便确保影响相同地址的操作保存数据完整性。

图 8-3 显示了按顺序执行的写 / 读 / 写 / 读命令序列和按数据重新排序执行的相同命令序列的相对时序。数据重新排序使得写和读操作在突发中发生，而不会有总线转换时序延迟或 bank 重新分配。

图 8-3. 数据重新排序影响



对于每个时钟周期，SDRAM 控制器在所有待定行和列命令中调度。

Bank 原则

SDRAM 控制器的 bank 原则使得用户能够请求一个处理的 bank 在操作完成后仍然打开以便将来的访问不会延迟激活相同的 bank 和行组合。控制器仅支持 8 个同时打开的 bank，所以如果其它的操作需要 bank 资源，那么一个打开的 bank 可能被关闭。

当 SDRAM 突发传输被调度时，打开的 bank 资源被动态地分配。Bank 分配由控制器自动地请求 (当一个输入传输跨越多个 SDRAM 突发时) 或由扩展的命令接口自动地请求。当一个 bank 必须被重新分配时，可使用最近最少使用的打开的 bank 替代。

如果控制器决定下一个待定命令会导致 bank 请求不被执行，那么 bank 可能根据待定操作保持打开或关闭。到相同行地址的关闭 bank 的请求，因为在 timer bank pool 中的待定操作，所以会导致 bank 仍然打开。一个到相同 bank 但是不同行地址的使 bank 打开的请求，因为 bank 中的待定命令会导致一个预充电操作发生。

写合并

SDRAM 控制器合并来自端口的连续的突发的写操作，其中第二个突发的起始地址大于第一个突发的结束地址并且产生的突发长度不会上溢 11-bit 突发长度计数器。如果之前的总线命令在接收到新命令之前已经完成执行，那么写合并不会发生。

突发长度支持

控制器支持 2、4、8 和 16 的突发长度、非 ECC 操作的 8、16 和 32 bit 的数据宽度以及 ECC 使能的 24 和 40 宽度的操作。表 8-3 显示了每个突发长度的 SDRAM 的类型。

表 8-3. SDRAM 突发长度

突发长度	SDRAM
4	LPDDR2、DDR2
8	DDR2、DDR3、LPDDR2
16	LPDDR2

宽度匹配

SDRAM 控制器自动地执行数据宽度转换。

ECC

单端口控制器支持控制器计算的存储器 ECC。控制器 ECC 采用标准的 Hamming 逻辑来检测和纠正单位错误以及检测双位错误。控制器 ECC 可用于 16-bit 和 32-bit 宽度，每个都需要一个额外的 8 bit 的存储器，从而分别导致 24-bit 和 40-bit 的实际存储器宽度。

控制器 ECC 提供以下功能：

- 字节写操作 — 存储控制器执行 read-modify-write 操作以确保当字的位的子集被写入时 ECC 数据保持有效。如果整个字正在被写入（但是少于一个全突发）并且 DM 管脚被连接，那么无需读取并且只有该字被更新。如果控制器 ECC 被禁用，那么不会影响字节写操作的性能。
- ECC 写回 — 当读操作检测到一个可纠正的错误时，对于 read-modify-write 操作，存储器位置被调度以纠正单位错误。ECC 写回通过 ctrlcfg 寄存器中的 cfg_enable_ecc_code_overwrites 域使能和禁用。
- ECC 错误的提示 — 存储控制器对单位和双位错误提供中断。中断和错误的状态被记录在状态寄存器中，如下所示：
 - dramsts 寄存器记录中断状态。
 - dramintr 寄存器记录中断屏蔽。
 - sbecount 寄存器记录单位错误数。
 - dbecount 寄存器记录双位错误数。
 - erraddr 寄存器记录最新错误的地址。

字节写操作

ECC 使能的字节写操作被执行为一个 read-modify-write。通常操作仅使用 timer bank pool 中的单入口。控制器 ECC 使能的子字写使用两个入口。第一个操作为读，第二个操作为写。这两个操作通过地址相依性被传输到 timer bank pool，以便写操作直到读数据返回才能被执行。这一方法确保任何接下来的相同地址的操作（来自相同端口）都在写操作之后执行，因为它们在写操作后被排列在行列表中。

如果正在写入整个字（但是少于一个全突发），那么不需要读并且只有该字被更新。

ECC 写回

如果使能控制器 ECC 并且一个读操作导致了一个可纠正的 ECC 错误，那么控制器纠正存储器中的位置（如果写回使能）。纠正导致规划一个新的 read-modify-write。在该位置执行新的读操作以确保修改位置的写操作不会被重写。实际的 ECC 纠正操作被执行为一个 read-modify-write 操作。

ECC 错误的用户提示

以下方法向您提示一个 ECC 错误：

对于 MPU 子系统，一个中断信号提供提示并且 ECC 错误信息被存储在状态寄存器中。



要了解更多信息，请参考 *Cyclone V 器件手册* 第 3 卷的 *Cortex-A9 Microprocessor Unit SubSystem* 章节。

交错选项

控制器支持以下地址交错选项：

- 非交错
- 无片选交错的 Bank 交错
- 有片选交错的 Bank 交错

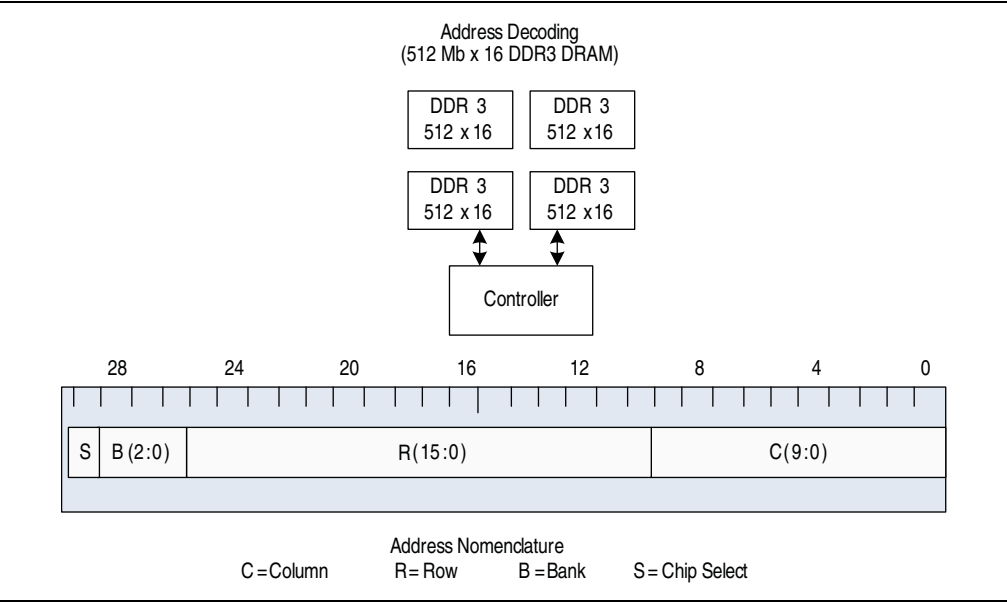
所有的交错实例都使用 512 megabits (Mb) x 16 DDR3 芯片并且被记录为字节地址。对于具有较小地址域的 RAM，域的顺序保持相同，但是宽度可能更改。

非交错

RAM 映射为非交错的。

图 8-4 显示了非交错的地址解码。

图 8-4. 非交错的地址解码

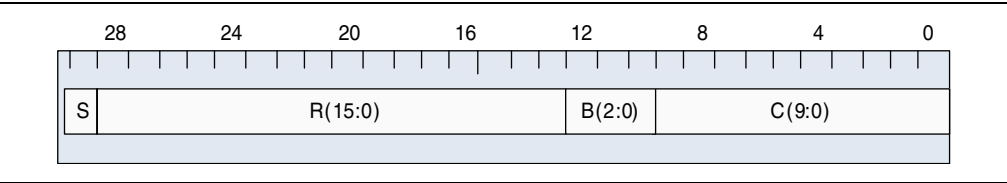


无片选交错的 Bank 交错

无片选交错的 Bank 交错交换非交错地址映射的 row 和 bank。这一交错支持较小数据结构在芯片的所有 bank 中分布。

图 8-5 显示了无片选交错的 bank 交错的地址解码。

图 8-5. 无片选交错的 Bank 交错的地址解码

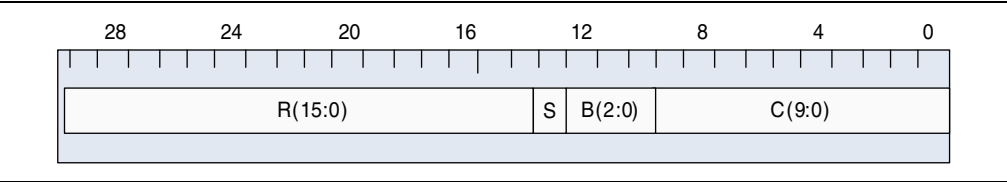


有片选交错的 Bank 交错

有片选交错的 Bank 交错将行地址移到顶部，然后依次移动片选、bank 和列地址。该交错支持较小的数据结构在多 bank 和芯片（通过访问 16 个 bank，可实现对存储器模块的多线程访问）中分布。当在芯片之间切换时，存储器时序被降低。

图 8-6 显示了有片选交错的 bank 交错的地址解码。


图 8-6. 有片选交错的 Bank 交错的地址解码



AXI 专有支持


单端口控制器支持 AXI 专有操作。控制器实现所有主器件共享的表，它可以存储高达 16 个待定写入。表项通道会分配好使读取操作都是独享的，同时会将任意主设备分配好使相同的地址都会成功写入。

任何不在列表中的同一种写操作返回一个独立的应答。如果当执行同一种读取时表是满的，那么表代替一个随机项。

 当使用 AXI 专有操作时，访问 Avalon-MM 接口的相同地址会导致不可预测的结果。

存储器保护

单端口控制器具有地址保护以便支持软件配置系统中所有主器件存储器的基本保护。如果系统完全使用 AXI 主器件进行设计，那么 TrustZone® 被支持。使用 Avalon-MM 的端口可以被配置为端口层保护。

 要了解关于 TrustZone® 的更多信息，请参考 ARM 网页 (www.arm.com)。

存储器保护以存储器中的物理地址为基础。可以设置规则来支持或禁止对一系列存储器的访问，或仅使能对一系列存储器（或两个存储器组合）的安全访问。

安全和非安全区域由分别包含 1 MB 边界的起始地址和结束地址的规则指定。您可以覆盖端口默认以及支持或不支持所有传输。

存储器保护表，通过 CSR 接口访问的内部表，包含允许或否定存储器访问的规则。您可以最多配置 20 个规则来控制存储器访问。表 8-4 列出了可以对每个规则指定的域。

表 8-4. 存储器保护表中规则的域

域	宽度	说明
有效	1	设置为 1 以激活规则。设置为 0 使规则无效。
端口屏蔽 ⁽¹⁾	10	指定规则适用的端口的设置，使用一个位代表一个端口，如下所示：位 0 到 5 对应于 FPGA 架构端口 0 到 5、位 6 对应于 AXI L3 开关读取、位 7 是 CPU 读取、位 8 是 L3 开关写入，以及位 9 是 CPU 写入。
TID_low ⁽¹⁾	12	该规则适用的低传输 ID 规则。如果它们大于或等于该值时，输入传输匹配。较小 TID 的端口使得 TID 移到较低位并且在顶部 0 加长。
TID_high ⁽¹⁾	12	该规则适用的高传输 ID 规则。如果它们少于或等于该值，那么输入传输匹配。
Address_low	12	指向 1MB 模块并且是较低地址。如果它们大于或等于该值，那么输入地址匹配。
Address_high	12	地址的上限。如果它们少于或等于该值，那么输入地址匹配。
保护	2	值 00 表示保护位没有被设置；值 01 设置保护位。没有设置 AXI 保护为已知值的系统应该对该域编程为其中一个保护值。
失败 / 支持	1	将该值设置为 1 以强制操作失败或成功。

表 8-4 注释：

(1) 虽然 TID 和端口屏蔽可能是冗余的，但在表中包含两者支持的可能规则压缩。如果连接到端口的主器件 没有连续的 TID，那么根据所需的规则数，一个基于端口的规则可能比基于 TID 的规则更有效。

一个端口具有支持或失败的默认访问状态，并且具有相反支持 / 失败值的规则可能覆盖默认状态。系统评估不符合存储器保护表中每个规则的每个传输。默认上支持访问的端口接收到的传输，只有当具有失败位的规则匹配传输时才会失败。相反的，默认上阻止访问的端口，只有当规则允许该传输通过时才会支持访问。

专有传输只有在读操作时才会进行安全性检查。只有当有效读取在内部专有表中被标记时，一个写操作才会发生。结果，执行同一种读然后再执行写的主器件，只有当同一种读成功时才能写入存储器。

TrustZone 配置的实例

对于 TrustZone 配置，存储器被划分为安全主器件访问的一系列存储器和非安全主器件访问的一系列存储器。两个存储器地址范围可能具有一系列重叠的存储器。

这一实例实现以下存储器配置：

- 2 GB 总 RAM 容量
- 0—512 MB 专用的安全区域
- 513—576 MB 共享区域
- 577—2048 MB 专用的非安全区域

在该实例中，每个端口在默认情况下配置以便禁止所有访问。表 8-5 显示了编程到存储器保护表的两个规则。

表 8-5. 对于实例配置的存储器保护表中的规则

规则 #	端口屏蔽	TID 低	TID 高	地址低	地址高	保护	失败 / 支持
1	0' b111111111	0	4095	0	576	b01	支持
2	0' b111111111	0	4095	512	2047	b00	支持

端口屏蔽值、TID 低和 TID 高应用于所有端口和端口内的所有传输。每个访问请求根据存储器保护表而被评估，并且将会失败（除非规则匹配支持一个传输成功地完成）。

表 8-6 显示了传输的样本集的结果。

表 8-6. 传输的样本集的结果

操作	源	地址	保护	结果	注释
读	CPU	4096	1	支持	与规则 1 匹配。
写	CPU	536、870、912 (512 MB)	1	支持	与规则 1 匹配。
写	与 L3 连接的主器件	605、028、350 (577 MB)	1	失败	不匹配规则 1(在地址域范围外)，不匹配规则 2(保护位不正确)。
读	与 L3 连接的主器件	4096	0	失败	不匹配规则 1(保护值错误)，不匹配规则 2(在地址范围外)。
写	CPU	536、870、912 (512 MB)	0	支持	匹配规则 2。
写	与 L3 连接的主器件	605、028、350 (577 MB)	0	支持	匹配规则 2。

如果您不想要存储器模块之间的任何重叠，那么可以将表 8-5 的两个规则中的地址范围指定为互相排斥。根据所需的 TrustZone 配置，您可以将规则添加到存储器保护表以创建被保护或不被保护空间的多个模块。

SDRAM 功耗管理

SDRAM 控制子系统支持以下 SDRAM 中的功耗节省功能：

- 局部阵列自刷新 (Partial array self-refresh, PASR)
- 关断
- LPDDR2 的深度关断

功耗节省模式由于一个用户命令而启动或从未激活而启动。

关断模式通过写入相应的控制寄存器而被启动。它强制 SDRAM burst-scheduling bank-management 逻辑关闭所有 bank 并且发出关断命令。对于指定的时钟周期数，当 SDRAM burst-scheduling 队列为空时，您可以编程控制器以使能关断。当接收到一个激活的 SDRAM 命令时，SDRAM 自动地重新激活。

其它的关断模式仅在用户控制下被执行。

DDR PHY

DDR PHY 在速度关键的命令路径中连接存储控制器和外部存储器件。

DDR PHY 实现以下的功能：

- 校准 —DDR PHY 支持 JEDEC 指定的步骤以便同步控制器和 SDRAM 芯片之间的存储器时序。校准算法在软件中实现。
- 存储器件初始化 —DDR PHY 执行模式寄存器写操作以便初始化器件。DDR PHY 在深度关断后处理重新初始化。
- 单数据速率到双数据速率转换。

时钟

所有时钟都被假设与 ddr_dqs_clk 存储器时钟是异步的。所有处理都被同步于存储器时钟域。

表 8-7 显示了 SDRAM 控制子系统时钟域。

表 8-7. SDRAM 控制子系统时钟域


时钟名称	说明
ddr_dq_clk	用于 PHY 的时钟
ddr_dqs_clk	用于 MPFE、单端口控制器、CSR 访问和 PHY 的时钟
ddr_2x_dqs_clk	用于 PHY 的时钟
l4_sp_clk	用于 CSR 接口的时钟
mpu_l2_ram_clk	用于 MPU 接口的时钟
l3_main_clk	用于 L3 接口的时钟
f2h_sdr_clk[5:0]	用于 FPGA 架构的 FPGA-to-HPS SDRAM 端口的 6 个单独的时钟

根据时钟关系，FPGA 架构将相应的时钟连接到写数据、读数据和构建的端口的命令端口。

 要了解更多信息，请参考 *Cyclone V 器件手册* 第 3 卷的 *Clock Manager* 章节。

复位

SDRAM 控制器子系统支持全复位（冷复位）和暖复位，可能会或不会保存存储器的内容。为了保存存储器内容，复位管理器可以请求单端口控制器在发出暖复位之前使 SDRAM 处于自刷新模式。如果存储器内容被保存，那么 PHY 和存储器时序逻辑不处于复位，但是控制器的其余部分会处于复位。

 要了解更多信息，请参考 *Cyclone V 器件手册* 第 3 卷的 *Reset Manager* 章节。

初始化

SDRAM 控制器子系统具有 CSR，它会控制控制器的操作，包括 DRAM 类型、DRAM 时序参数和相对端口优先权。它也具有取决于 FPGA 架构的一小组位，用于配置存储控制器和 FPGA 架构之间的端口；当您使用 Qsy 中的 HPS GUI 配置实现时可以使用这些位。

CSR 使用专用的从接口而被配置，可以提供对寄存器的访问。该区域控制所有 SDRAM 操作、MPFE 调度器配置和 PHY 校准。

FPGA 架构接口配置被编程进 FPGA 架构并且这些寄存器位的值可以由软件读取。可以在软件开发商无需了解 FPGA-to-HPS SDRAM 接口如何被配置的情况下对该端口进行配置。

协议详细信息

Avalon-MM 双向端口

Avalon-MM 双向端口为标准的 Avalon-MM 端口，用于分配读和写操作。每个配置的 Avalon-MM 双向端口包含表 8-8 中列出的信号。

表 8-8. Avalon-MM 双向端口信号

名称	位宽	方向	功能
clk	1	In	用于 Avalon-MM 接口的时钟
读	1	In	表示读传输
写	1	In	表示写传输
地址	32	In	传输的地址
readdata	32、64、128 或 256	Out	读数据返回
readdatavalid	1	Out	读数据返回的有效周期标记
writedata	32、64、128 或 256	In	传输的写数据
byteenable	4 (32-bit 数据)、8(64-bit 数据)、16(128-bit 数据)、32(256-bit 数据)	In	每个写字节的字节使能
waitrequest	1	Out	表示需要额外的周期以完成一个传输
burstcount	11	In	传输突发长度

读和写接口被配置成相同的尺寸。字节使能尺寸随着数据总线尺寸的扩展而扩展。

要了解关于 Avalon-MM 协议的更多信息，请参考 [Avalon Interface Specifications](#)。

Avalon-MM 写端口

Avalon-MM 写端口为标准的 Avalon-MM 端口，仅用于分配写操作。每个配置的 Avalon-MM 写端口包含表 8-9 中列出的信号。

表 8-9. Avalon-MM 写端口信号

名称	位宽	方向	功能
复位	1	In	复位
clk	1	In	时钟
写	1	In	表示写传输
地址	32	In	传输的地址
writedata	32、64、128 或 256	In	传输的写数据
byteenable	4 (32-bit 数据)、8 (64-bit 数据)、16 (128-bit 数据)、32 (256-bit 数据)	In	每个写字节的字节使能
waitrequest	1	Out	表示需要额外的周期以完成一个传输
burstcount	11	In	传输突发长度

要了解关于 Avalon-MM 协议的更多信息，请参考 [Avalon Interface Specifications](#)。

Avalon-MM 读端口

Avalon-MM 读端口是标准的 Avalon-MM 端口，仅用于分配读操作。每个配置的 Avalon-MM 读端口包含表 8-10 中列出的信号。

表 8-10. Avalon-MM 读端口信号

名称	位宽	方向	功能
reset	1	In	复位
clk	1	In	时钟
read	1	In	表示读传输
address	32	In	传输的地址
readdata	32、64、128 或 256	Out	读数据返回
readdatavalid	1	Out	对读数据返回标记有效周期
waitrequest	1	Out	表示需要额外的周期以完成一个传输。当需要延迟以接受读命令时用于读操作。
burstcount	11	In	传输突发长度

要了解关于 Avalon-MM 协议的更多信息，请参考 [Avalon Interface Specifications](#)。

AXI 端口

AXI 端口使用一个 AXI-3 接口。

要了解关于 AXI-3 接口的更多信息，请参考 ARM 网站 (www.arm.com) 的 *AMBA Open Specifications*。

要了解关于高性能 II 控制器 (HPC II) 中 AXI 接口端口的更多信息，请参考 *外部存储器接口手册* 中的 *Functional Description—HPC II Controller* 章节。

每个配置的 AXI 端口包含表 8-11 中列出的信号。每个 AXI 接口信号独立于其它接口的所有信号 (包括时钟和复位)。

表 8-11. AXI 端口信号 (1/2)

名称	位宽	方向	功能
ARESETn	1	In	复位
ACLK	1	In	时钟
写地址通道信号			
AWID	4	In	写识别标签
AWADDR	32	In	写地址
AWLEN	4	In	写突发长度
AWSIZE	3	In	传输容量的宽度
AWBURST	2	In	突发类型
AWREADY	1	Out	表示可以发出写命令
AWVALID	1	In	表示有效写命令
写数据通道信号			
WID	4	In	写数据传输 ID
WDATA	32、64、128 或 256	In	写数据
WSTRB	4、8、16、32	In	基于字节的写数据选通。对于 32-bit 宽到 256-bit 宽传输，每位宽度对应于 8 位宽传输。
WLAST	1	In	突发中的最后传输
WVALID	1	In	表明写数据 + 选通有效
WREADY	1	Out	表明准备发送写数据和选通
写响应通道信号			
BID	4	Out	写响应传输 ID
BRESP	2	Out	写响应状态
BVALID	1	Out	写响应有效信号
BREADY	1	In	写响应 ready 信号
读地址通道信号			
ARID	4	In	读识别标签
ARADDR	32	In	读地址
ARLEN	4	In	读突发长度
ARSIZE	3	In	传输容量宽度
ARBURST	2	In	突发类型
ARREADY	1	Out	表示准备发出读命令

表 8 - 11. AXI 端口信号 (2/2)

名称	位宽	方向	功能
ARVALID	1	In	表示有效读命令
读数据通道信号			
RID	4	Out	读数据传输 ID
RDATA	32、64、128 或 256	Out	读数据
RRESP	2	Out	读响应状态
RLAST	1	Out	突发中的最后传输
RVALID	1	Out	表示读数据有效
RREADY	1	In	读数据通道 ready 信号

SDRAM 控制器子系统编程模型

初始化

SDRAM 控制器配置使用 CSR 接口通过配置寄存器的软件编程而发生。SDRAM 控制器的初始化具有两个不同控制的单独区域。

时序参数

SDRAM 控制器支持在运行时可配置的一组完整的时序参数。


SDRAM 控制器地址映射和寄存器定义

 地址映射位于该手册卷附带的 [hps.html](#) 文件中。点击链接以打开文件。

要查看模块说明和基地址，找到并且点击链接以查看以下模块实例：

■ sdr

要查看寄存器和域说明，找到并且点击寄存器名称。寄存器地址是相对于每个模块实例的基地址的偏移。

 所有模块的基地址也在 *Cyclone V 器件手册* 第 3 卷的 *Introduction to the Hard Processor System* 章节中列出。

文档修订历史

表 8 - 12 显示了该文档的修订历史。

表 8 - 12. 文档修订历史

日期	版本	修订内容
2012 年 11 月	1.1	添加了地址映射和寄存器定义部分。
2012 年 1 月	1.0	首次发布。