

Intel® Arria® 10 内核架构和通用 I/O 手册



A10-HANDBOOK | 2017.05.08

官网最新文档: [PDF](#) | [HTML](#)

内容

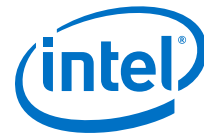
1. Arria® 10 器件中的逻辑阵列模块与自适应逻辑模块.....	7
1.1. LAB.....	7
1.1.1. MLAB.....	8
1.1.2. 本地和直链(Direct Link)互联	9
1.1.3. 共享算术链和进位链互联	10
1.1.4. LAB 控制信号.....	11
1.1.5. ALM 资源	12
1.1.6. ALM 输出	13
1.2. ALM 操作模式	14
1.2.1. 正常模式.....	14
1.2.2. 扩展 LUT 模式.....	17
1.2.3. 算术模式	18
1.2.4. 共享算术模式	20
1.3. LAB 功耗管理技术	21
1.4. 文档修订历史.....	21
2. Arria 10 器件中的嵌入式存储器模块.....	22
2.1. 嵌入式存储器类型.....	22
2.1.1. Arria 10 器件中的嵌入式存储器性能.....	23
2.2. Arria 10 器件的嵌入式存储器设计指南.....	23
2.2.1. 考虑存储器模块选择.....	23
2.2.2. 指南：实现外部冲突消解.....	24
2.2.3. 指南：定制 Read-During-Write 行为.....	24
2.2.4. 指南：考虑上电状态和存储器初始化.....	27
2.2.5. 指南：控制时钟来降低功耗.....	28
2.3. 嵌入式存储器特性.....	28
2.4. 嵌入式存储器模式.....	29
2.4.1. 单端口模式的嵌入式存储器配置.....	30
2.4.2. 双端口模式的嵌入式存储器配置.....	30
2.5. 嵌入式存储器时钟模式.....	32
2.5.1. 每种存储器模式的时钟模式.....	32
2.5.2. 时钟模式中的异步清零.....	32
2.5.3. 同步读/写中的输出读数据.....	33
2.5.4. 时钟模式的独立时钟使能.....	33
2.6. 嵌入式存储器模块中的奇偶校验位.....	33
2.7. 嵌入式存储器模块中的字节使能.....	33
2.7.1. 存储器模块中的字节使能控制.....	34
2.7.2. 数据字节输出.....	34
2.7.3. RAM 模块操作.....	34
2.8. 存储器模块 Packed 模式支持.....	35
2.9. 存储器模块地址时钟使能支持.....	35
2.10. 存储器模块异步清零.....	36
2.11. 存储器模块纠错码支持.....	37
2.11.1. 纠错码真值表.....	38

2.12. 文档修订历史.....	38
3. Arria 10 器件中的精度可调 DSP 模块.....	40
3.1. Arria 10 器件中支持的操作模式.....	40
3.1.1. 特性.....	41
3.2. 资源.....	42
3.3. 设计考量.....	43
3.3.1. 操作模式.....	44
3.3.2. 用于定点运算的内部系数和预加器.....	45
3.3.3. 用于定点运算的累加器.....	45
3.3.4. Chainout 加法器.....	45
3.4. 模块体系结构.....	45
3.4.1. 输入寄存器组(Input Register Bank).....	47
3.4.2. 流水线寄存器.....	49
3.4.3. 定点运算的预加器.....	49
3.4.4. 定点运算的内部系数.....	50
3.4.5. 乘法器.....	50
3.4.6. 加法器.....	50
3.4.7. 用于定点运算的累加器和 Chainout 加法器.....	50
3.4.8. 用于定点运算的脉动寄存器.....	51
3.4.9. 用于定点运算的双倍累加寄存器.....	51
3.4.10. 输出寄存器组(Output Register Bank).....	51
3.5. 操作模式说明.....	52
3.5.1. 定点运算的操作模式.....	53
3.5.2. 浮点运算的操作模式.....	59
3.6. 文档修订历史.....	66
4. Arria 10 器件中的时钟网络和 PLL.....	68
4.1. 时钟网络.....	68
4.1.1. Arria 10 器件中的时钟资源.....	69
4.1.2. 层次化时钟网络.....	71
4.1.3. 时钟网络类型.....	72
4.1.4. 时钟网络源.....	76
4.1.5. 时钟控制模块.....	77
4.1.6. 时钟断电.....	80
4.1.7. 时钟使能信号.....	80
4.2. Arria 10 PLL.....	81
4.2.1. PLL 使用.....	82
4.2.2. PLL 体系结构.....	83
4.2.3. PLL 控制信号.....	83
4.2.4. 时钟反馈模式.....	84
4.2.5. 时钟倍频与分频.....	85
4.2.6. 可编程相移.....	85
4.2.7. 可编程占空比.....	86
4.2.8. PLL 级联.....	86
4.2.9. 参考时钟源.....	86
4.2.10. 时钟切换.....	87
4.2.11. PLL 重配置和动态相移.....	91

4.3. 文档修订历史.....	91
5. Arria 10 器件的 I/O 和高速 I/O.....	94
5.1. Arria 10 器件中的 I/O 和差分 I/O 缓冲.....	95
5.2. Arria 10 器件中的 I/O 标准和电平.....	96
5.2.1. Arria 10 器件的 FPGA I/O 所支持的 I/O 标准.....	96
5.2.2. Arria 10 器件的 HPS I/O 所支持的 I/O 标准.....	97
5.2.3. Arria 10 器件中的 I/O 标准电平.....	98
5.2.4. Arria 10 器件中的 MultiVolt I/O 接口.....	99
5.3. Arria 10 器件的 Intel FPGA I/O IP 内核.....	99
5.4. Arria 10 器件的 I/O 资源.....	99
5.4.1. Arria 10 器件的 GPIO Bank、SERDES 和 DPA 位置.....	100
5.4.2. Arria 10 器件的 GPIO 缓冲和 LVDS 通道.....	105
5.4.3. Arria 10 器件的 I/O Bank 组.....	108
5.4.4. Arria 10 器件的 I/O 纵向移植.....	114
5.5. Arria 10 器件的体系结构和 I/O 的一般功能.....	115
5.5.1. Arria 10 器件中的 I/O 单元结构.....	115
5.5.2. Arria 10 器件的 I/O 管脚特性.....	117
5.5.3. Arria 10 器件的可编程 IOE 的特性.....	118
5.5.4. Arria 10 器件的片上 I/O 匹配.....	123
5.5.5. Arria 10 器件的外部 I/O 匹配.....	132
5.6. Arria 10 器件的高速源同步 SERDES 和 DPA.....	140
5.6.1. SERDES 电路	141
5.6.2. Arria 10 器件中支持的 SERDES I/O 标准.....	142
5.6.3. Arria 10 器件的差分发送器.....	144
5.6.4. Arria 10 器件中的差分接收器.....	145
5.6.5. Arria 10 器件的 PLL 和时钟.....	152
5.6.6. Arria 10 器件的时序和优化.....	163
5.7. 在 Arria 10 器件中使用 I/O 和高速 I/O.....	168
5.7.1. Arria 10 器件的 I/O 和高速 I/O 通用指南.....	168
5.7.2. 混合电压参考和非电压参考 I/O 标准.....	170
5.7.3. 指南：上电顺序期间不可驱动 I/O 管脚.....	171
5.7.4. 指南：在 HPS 共享的 I/O Bank 中使用 I/O 管脚.....	171
5.7.5. 指南：最大化 DC 电流限制.....	172
5.7.6. 指南：Altera LVDS SERDES IP 内核实例.....	172
5.7.7. 指南：Soft-CDR 模式的 LVDS SERDES 管脚对.....	172
5.7.8. 指南：Arria 10 GPIO 性能的最小化高抖动的影响.....	173
5.7.9. 指南：外部存储器接口 I/O Bank 2A 的使用.....	173
5.8. 文档修订历史.....	174
6. Arria 10 器件的外部存储器接口.....	178
6.1. Arria 10 外部存储器接口解决方案的关键特性.....	178
6.2. Arria 10 器件支持的存储器标准.....	178
6.3. Arria 10 器件的外部存储器接口宽度.....	180
6.4. Arria 10 器件的外部存储器接口 I/O 管脚.....	180
6.4.1. 指南：外部存储器接口 I/O Bank 2A 的使用.....	181
6.5. Arria 10 器件封装支持的存储器接口.....	182
6.5.1. 含有 ECC 的 DDR3 x40 在 Arria 10 中的封装支持.....	183

6.5.2. Single 和 Dual-Rank 的 DDR3 x72(含有 ECC)在 Arria 10 中的封装支持.....	185
6.5.3. 含有 ECC 的 DDR4 x40 在 Arria 10 中的封装支持.....	187
6.5.4. Single-Rank 含有 ECC 的 DDR4 x72 在 Arria 10 中封装支持.....	189
6.5.5. Dual-Rank 含有 ECC 的 DDR4 x72 在 Arria 10 中的封装支持.....	191
6.5.6. Arria 10 的 HPS 外部存储器接口连接.....	192
6.6. Arria 10 器件支持的外部存储器接口 IP.....	196
6.6.1. Ping Pong PHY IP.....	197
6.7. Arria 10 器件的外部存储器接口体系结构.....	197
6.7.1. I/O Bank.....	198
6.7.2. I/O AUX.....	206
6.8. 文档修订历史.....	208
7. Arria 10 器件中的配置、设计安全和远程系统更新.....	210
7.1. 增强配置和通过协议配置(Configuration via Protocol).....	210
7.2. 配置方案.....	211
7.2.1. 主动串行配置.....	211
7.2.2. 被动串行配置.....	220
7.2.3. 快速被动并行配置.....	224
7.2.4. JTAG 配置.....	228
7.3. 配置详细信息.....	231
7.3.1. MSEL 管脚设置.....	231
7.3.2. CLKUSR.....	232
7.3.3. 配置序列.....	233
7.3.4. 配置时序波形.....	236
7.3.5. 估算配置时间.....	239
7.3.6. 器件配置管脚.....	240
7.3.7. 配置数据压缩.....	242
7.4. 使用主动串行方案升级远程系统.....	243
7.4.1. 配置映像.....	244
7.4.2. 远程更新模式中的配置序列.....	245
7.4.3. 远程系统更新电路.....	247
7.4.4. 使能远程系统更新电路.....	247
7.4.5. 远程系统更新寄存器.....	247
7.4.6. 远程系统更新状态机.....	249
7.4.7. 用户看门狗定时器 (User Watchdog Timer)	249
7.5. 设计安全.....	250
7.5.1. 安全密钥类型.....	251
7.5.2. 安全模式.....	251
7.5.3. Arria 10 Qcrypt 安全工具.....	253
7.5.4. 设计安全实现步骤.....	253
7.6. 文档修订历史.....	254
8. Arria 10 器件的 SEU 缓解	256
8.1. SEU 缓解概述.....	256
8.1.1. SEU 缓解应用.....	256
8.1.2. 配置 RAM.....	256
8.1.3. 嵌入式存储器.....	257
8.2. Arria 10 缓解技术.....	257

8.2.1. 存储器模块纠错码支持.....	257
8.2.2. CRAM 的错误检测和纠正.....	258
8.3. 规范.....	265
8.3.1. 错误检测频率.....	265
8.3.2. 错误检测时间.....	266
8.3.3. EMR 更新间隔.....	266
8.3.4. 纠错时间.....	268
8.4. 文档修订历史.....	268
9. Arria 10 器件中的 JTAG 边界扫描测试.....	270
9.1. BST 操作控制	270
9.1.1. IDCODE	270
9.1.2. 所支持的 JTAG 指令	271
9.1.3. JTAG 安全模式	272
9.1.4. JTAG 专用指令	273
9.2. JTAG 操作的 I/O 电压	273
9.3. 执行 BST	273
9.4. 使能和禁用 IEEE Std. 1149.1 BST 电路	274
9.5. IEEE Std. 1149.1 边界扫描测试指南.....	275
9.6. IEEE Std. 1149.1 边界扫描寄存器	275
9.6.1. Arria 10 器件 I/O 管脚的边界扫描单元.....	276
9.6.2. IEEE Std. 1149.6 边界扫描寄存器.....	278
9.7. 文档修订历史.....	279
10. Arria 10 器件中的电源管理.....	280
10.1. 功耗.....	280
10.1.1. 动态功耗公式.....	280
10.2. 功耗降低技术.....	281
10.2.1. SmartVID	281
10.2.2. 可编程电源技术.....	281
10.2.3. 低静态功耗器件等级.....	282
10.2.4. SmartVID 功能实现.....	282
10.3. 电源感应线 (Power Sense Line)	284
10.4. 电压传感器.....	284
10.4.1. 外部模拟信号的输入信号范围.....	286
10.4.2. 在 Arria 10 器件中使用电压传感器.....	286
10.5. 温度感应二极管.....	290
10.5.1. 内部温度感应二极管.....	290
10.5.2. 外部温度感应二极管.....	291
10.6. 上电复位电路.....	292
10.6.1. POR 电路监控和未监控电源.....	294
10.7. 上电和断电序列.....	294
10.8. 电源设计.....	297
10.9. 文档修订历史.....	298



1. Arria® 10 器件中的逻辑阵列模块与自适应逻辑模块

逻辑阵列模块 (LAB) 由称作自适应逻辑模块 (ALM) 的基本构造模块组成, 通过配置这些模块, 能够实现逻辑功能、算术功能以及寄存器功能。

您可以将 Arria® 10 器件中 1/4 的 LAB 用作存储器逻辑阵列模块 (MLAB)。

Quartus® Prime 软件和所支持的第三方综合工具, 与参数化功能 (例如参数化模块库 (LPM)) 一起, 自动为常用功能 (例如: 计数器、加法器、减法器 and 算术功能) 选择合适的模式。

本章节涵盖以下两方面内容:

- LAB
- ALM 操作模式

相关链接

[Arria 10 器件手册: 已知问题](#)

列出了对 Arria 10 器件手册章节的计划更新。

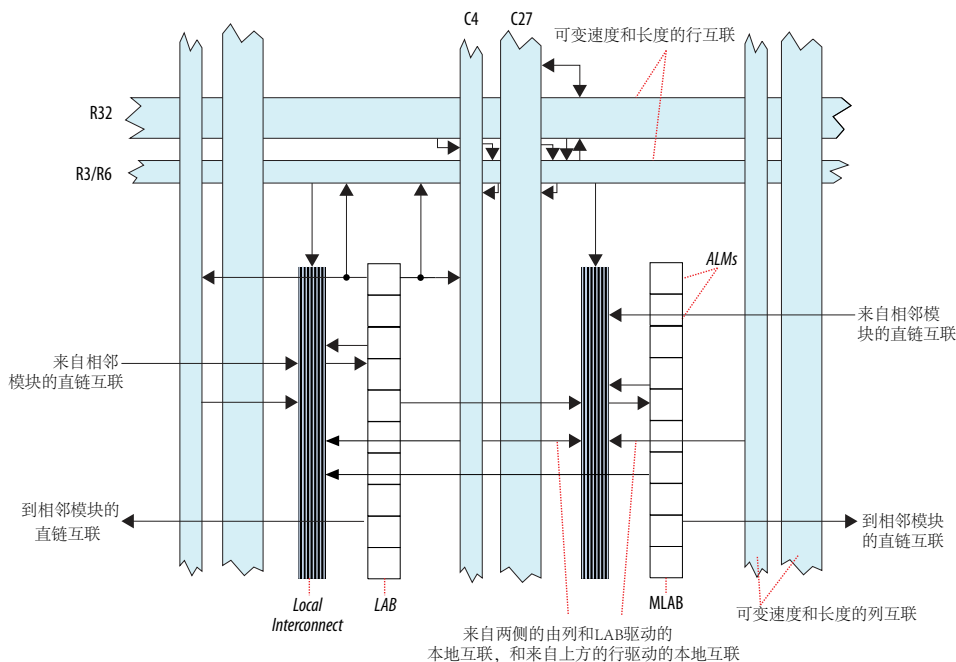
1.1. LAB

LAB 是由一组逻辑资源组成的可配置逻辑模块。每个 LAB 均包含专用逻辑, 用来将控制信号驱动到它的 ALM 中。

MLAB 是 LAB 的超集, 包含了 LAB 的所有特性。

图 1. Arria 10 器件中的 LAB 结构和互联概况

此图显示了基于 LAB 互联的 Arria 10 LAB 和 MLAB 结构。



1.1.1. MLAB

每个 MLAB 均支持最大 640 bits 的简单双端口 SRAM。

您可以将 MLAB 中的每一个 ALM 配置成 32 (深度) × 2 (位宽) 存储器模块，生成一个 32 (深度) × 20 (位宽) 简单双端口 SRAM 模块。

MLAB 支持使用 supports the following 64-deep modes in soft implementation using the Quartus Prime 软件执行以下 64-deep 模式：

- 64 (深度) × 8 (位宽)
- 64 (深度) × 9 (位宽)
- 64 (深度) × 10 (位宽)

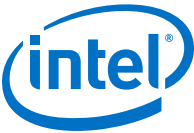
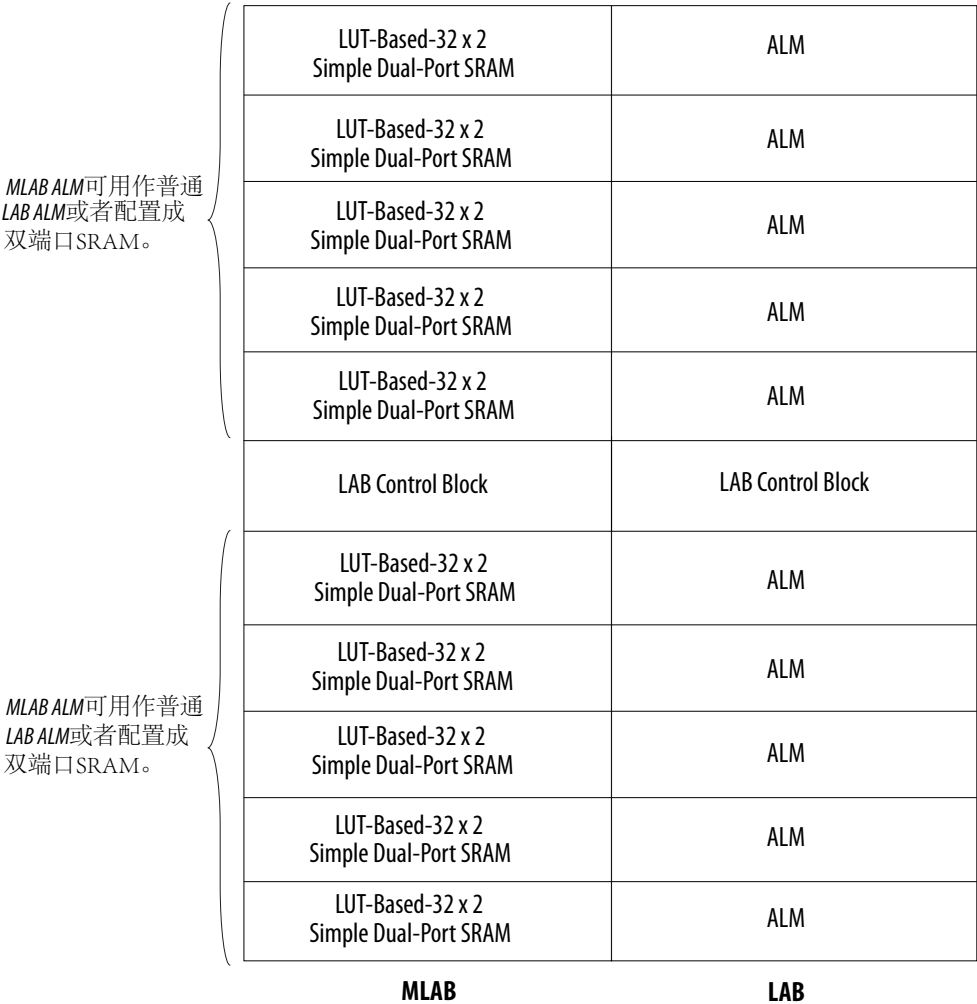


图 2. Arria 10 器件的 LAB 和 MLAB 结构



1.1.2. 本地和直链(Direct Link)互联

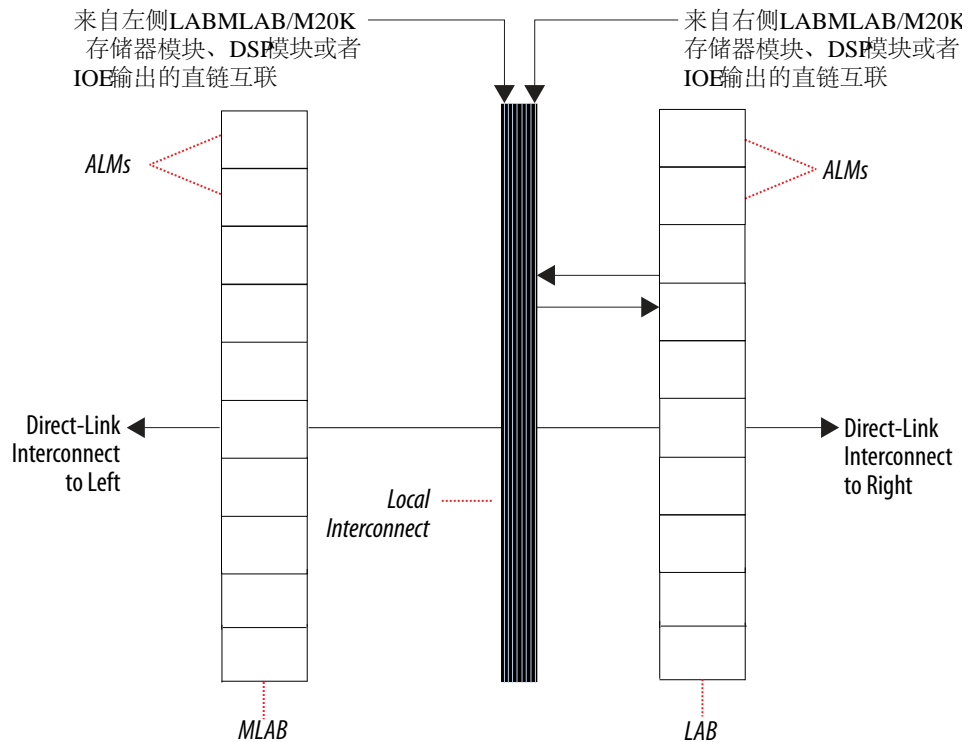
每个 LAB 能驱动 40 个 ALM 输出。两组 20 个 ALM 输出能通过直链互联直接驱动毗邻的 LAB。

直链互联功能最大限度地降低了行列互联的使用，从而提供了更高的性能和更大的灵活性。

本地互联通过使用相同 LAB 中的行列互联以及 ALM 输出来驱动相同 LAB 中的 ALM。

相邻的 LAB、MLAB、M20K 模块或者左/右侧的数字信号处理(DSP)模块也能够通过直链连接来驱动 LAB 的本地互联。

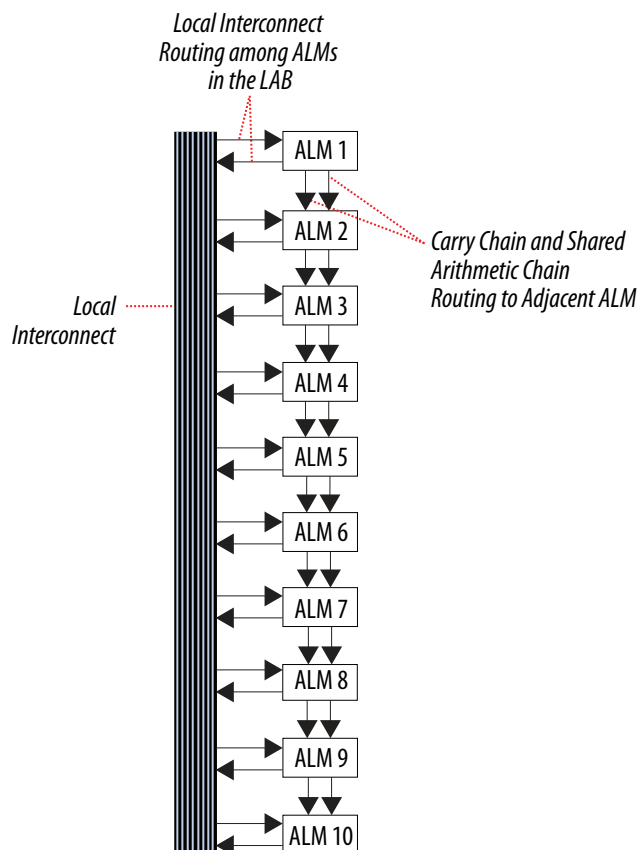
图 3. Arria 10 器件的 LAB 快速本地和直链互联



1.1.3. 共享算术链和进位链互联

ALM 之间有两条专用路径—进位链和共享算术链。Arria 10 器件在 LAB 中包含一个增强的互联结构，实现高效算术功能的共享算术链以及进位链的布线。这些 ALM-to-ALM 连接会旁路本地互联。Quartus Prime Compiler 自动利用这些资源来提高利用率及性能。

图 4. 共享算术链和进位链互联



1.1.4. LAB 控制信号

每个 LAB 均包含专用逻辑将控制信号驱动到它的 ALM 中，并且包含两个独立的时钟源和三个时钟使能信号。

LAB 控制模块使用两个时钟源和三个时钟使能信号最多可生成三个时钟。一个反向的时钟源可看作是一个独立的时钟源。每个时钟和时钟使能信号都是相连的。

置低时钟使能信号会关闭相应的 LAB 范围(LAB-wide)时钟。

LAB 行时钟[5..0]和 LAB 本地互联生成 LAB 范围控制信号。MultiTrack 互联所固有的低偏移不但实现了数据的分布，也实现了时钟与控制信号的分布。MultiTrack 互联包含不同长度和速度的连续的，性能优化的布线，用户设计内和设计间的模块互联。

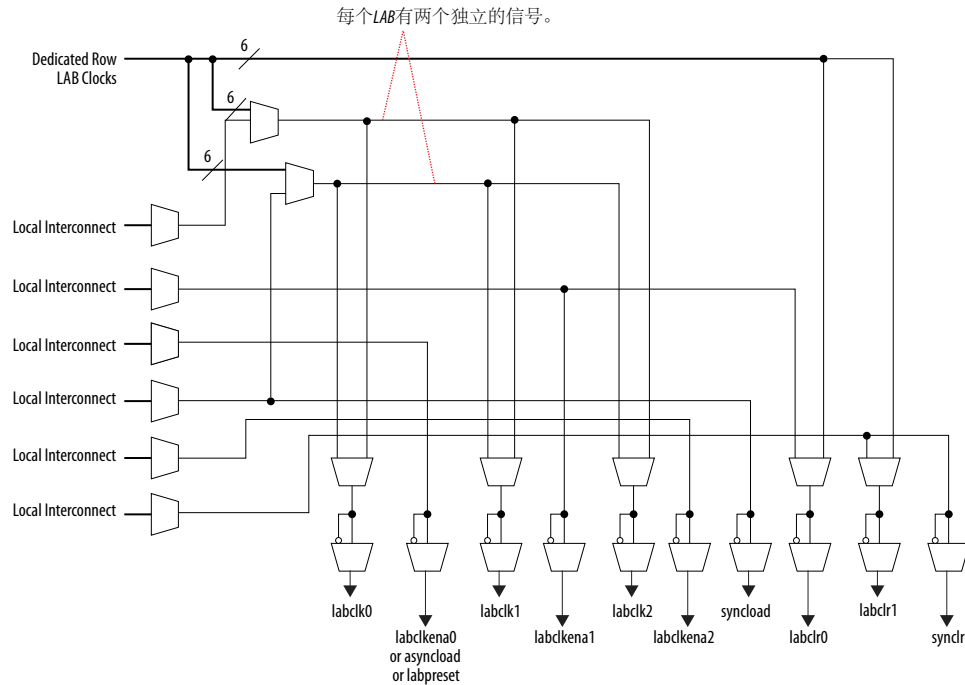
清零和预置逻辑控制

LAB 范围信号控制寄存器清零信号的逻辑。ALM 直接支持异步清零功能。通过 Quartus Prime 软件的 **NOT-gate push-back** 逻辑来实现寄存器预置。每个 LAB 支持两个清零操作。

Arria 10 器件提供了一个对器件中所有寄存器进行复位的器件范围(device-wide)复位管脚 (DEV_CLRn)。编译前，在 Quartus Prime 中使能 DEV_CLRn 管脚。这一器件范围复位信号覆盖所有其它控制信号。

图 5. Arria 10 器件的 LAB 范围控制信号

此图显示了 LAB 中的时钟源和时钟使能信号。



1.1.5. ALM 资源

每个 ALM 均包含多种基于 LUT 的资源，可以在两个组合自适应 LUT (ALUT)和四个寄存器之间进行划分。

通过使用两个组合 ALUT 的八个输入，一个 ALM 能够实现两种功能的各种组合。这一自适应性使 ALM 完全向后兼容四输入 LUT 体系结构。一个 ALM 也能够实现任何高达 6 输入或是某些 7 输入的功能。

一个 ALM 包含四个可编程寄存器。每个寄存器包含如下端口：

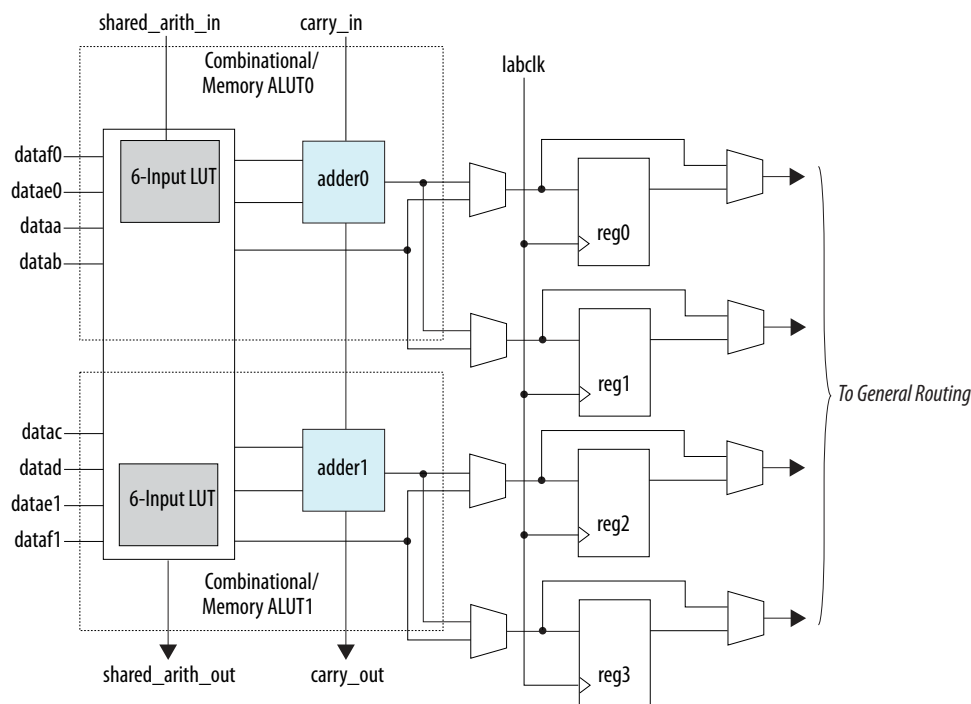
- 数据
- 时钟
- 同步和异步清零
- 同步加载

全局信号，通用 I/O (GPIO)管脚或者任何内部逻辑都可以驱动 ALM 寄存器的时钟使能信号，时钟和清零控制信号。

对于组合功能，寄存器被旁路，LUT 的输出直接驱动到 ALM 的输出。

注意: Quartus Prime 软件自动配置 ALM 以优化性能。

图 6. Arria 10 器件的 ALM 高级结构图



1.1.6. ALM 输出

每个 ALM 中的通用布线输出驱动本地、行和列布线资源。两个 ALM 输出能够驱动行、列或者直链布线连接。

LUT、加法器或者寄存器输出能驱动 ALM 输出。LUT 或加法器能够驱动一个输出，而寄存器驱动另一个输出。

寄存器封装(Register Packing)通过将无关的寄存器和组合逻辑封装在一个 ALM 中来提高器件利用率。改善布局布线的另一种机制是支持寄存器输出驱动回相同 ALM 的 LUT 中，使寄存器与其本身的扇出 LUT 封装在一起。ALM 也能够驱动寄存的以及未寄存的 LUT 或者加法器输出。

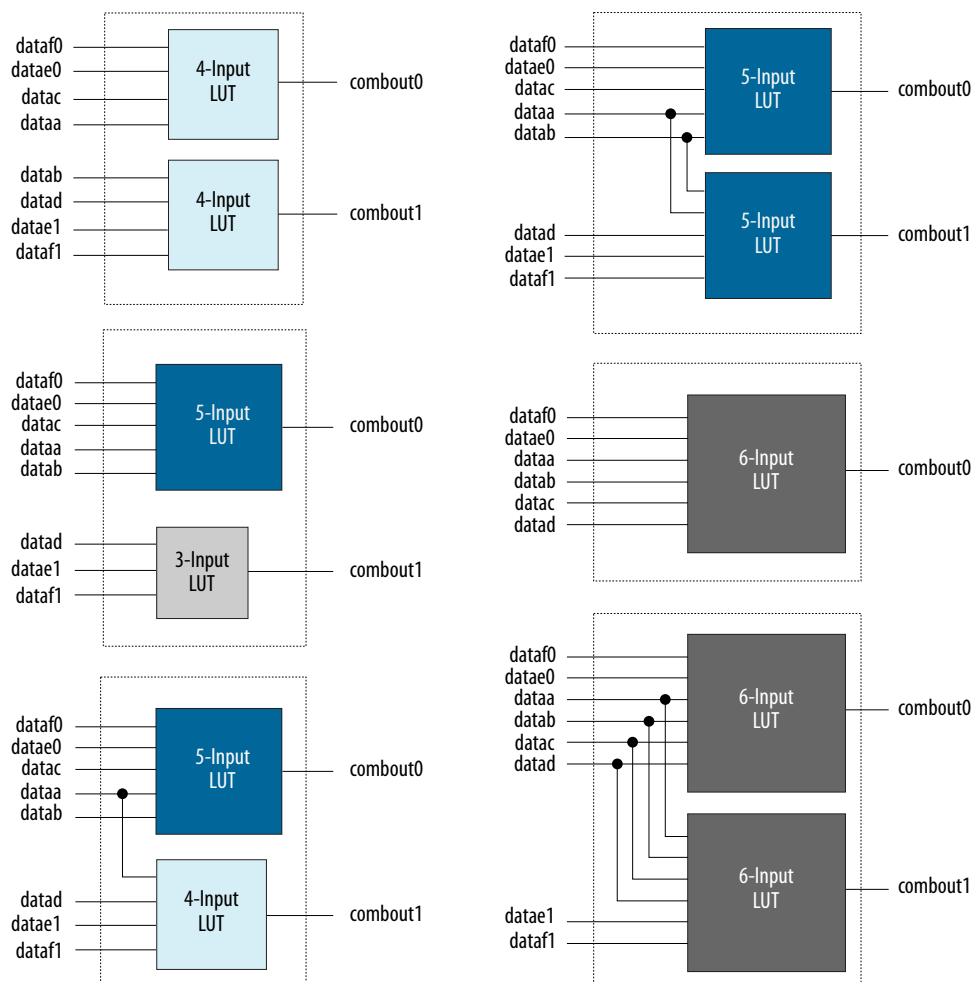
[illegible]

- 正常模式
- 扩展 LUT 模式
- 算术模式
- 共享算术模式

Quartus Prime Compiler 自动选择到 LUT 的输入。正常模式下的 ALM 支持寄存器封装。

图 8. 正常模式下的 ALM

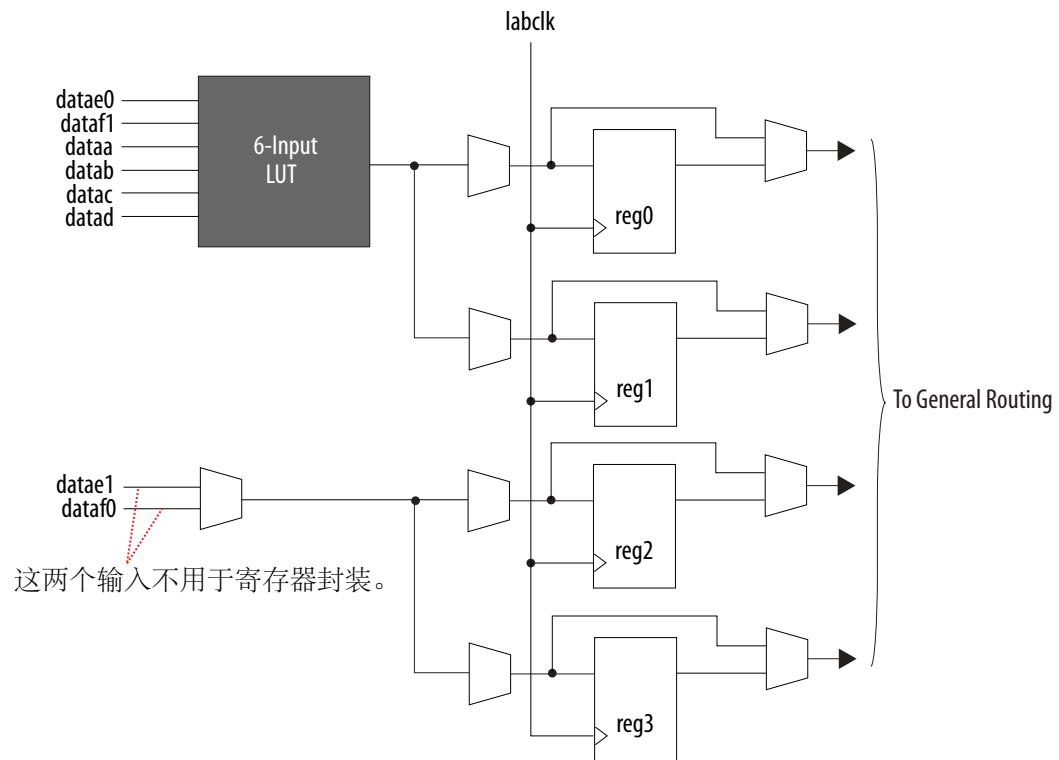
支持具有更少输入(少于图中所示的)的功能组合。例如，支持以下数量输入的功能组合：4 和 3，3 和 3，3 和 2，5 和 2。



对于 2 个 5 输入的功能封装进一个 ALM 中的情况，这些功能必须至少有两个通用输入，分别是 dataa 和 datab。4 输入功能与 5 输入功能的组合要求一个通用输入(dataa 或 datab)。

在一个 ALM 中实现 2 个 6 输入的功能的情况下，4 个输入必须是共享的，并且组合功能必须相同。在资源使用很少的器件中，Quartus Prime 软件可能会将可以在一个 ALM 中实现的功能放到不同的 ALM 中实现，从而实现最佳性能。当器件变得越来越满时，Quartus Prime 软件将自动使用 Arria 10 ALM 的全部潜能。Quartus Prime Compiler 自动搜索使用通用输入的功能或者完全独立的功能，将这些功能放置在一个 ALM 中，从而更加有效地利用器件资源。此外，通过设置位置约束，您也能够手动控制资源的使用。

图 9. 正常模式中的输入功能



您可以使用下面输入实现任何 6 输入功能：

- dataa
- datab
- datac
- datad
- datae0 和 dataf1 或 datae1 和 dataf0

如果使用 datae0 和 dataf1 输入，那么能得到下面输出：

- 驱动到 register0 的输出，或者 register0 被旁路
- 驱动到 register1 的输出，或者 register1 被旁路

您可以使用 datae1 或 dataf0 中可用的输入作为到 register2 或 register3 的封装寄存器输入。

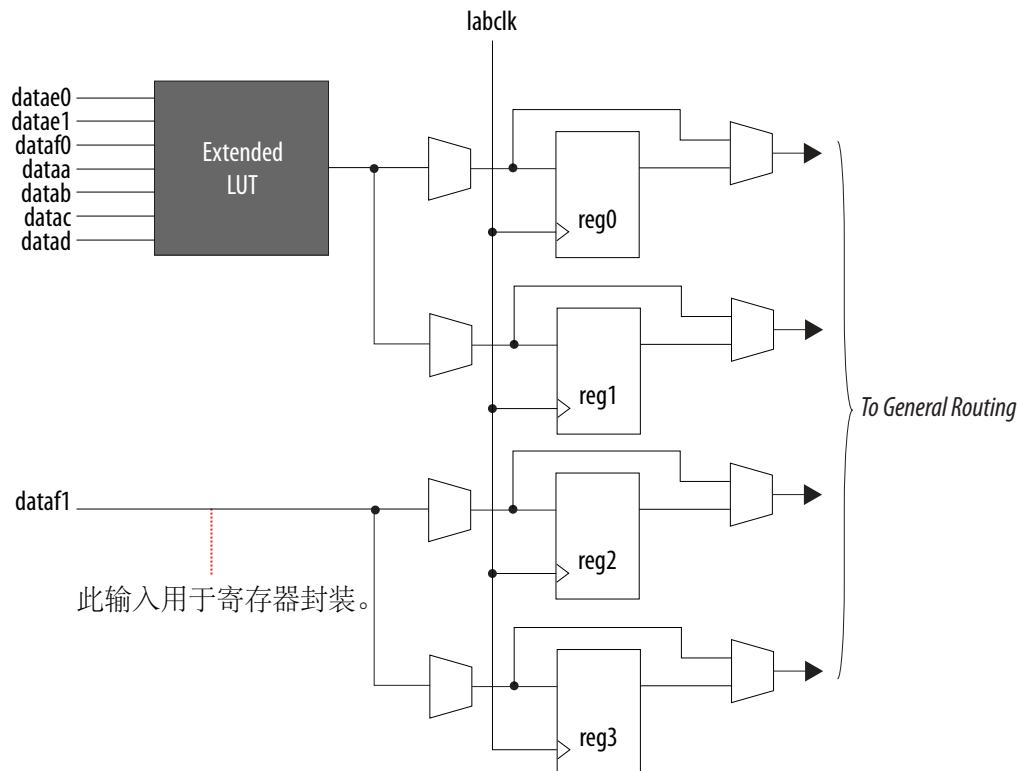
如果使用 datae1 和 dataf0 输入，那么您可以得到下面输出：

- 驱动到 register2 的输出，或者 register2 被旁路
- 驱动到 register3 的输出，或者 register3 被旁路

您可以使用 datae0 或 dataf1 中可用的输入作为到 register0 或 register1 的封装寄存器输入。

1.2.2. 扩展 LUT 模式

图 10. Arria 10 器件的扩展 LUT 模式中所支持的 7 输入功能的模板



您可以使用 `dataf0` 输入作为到 `register0` 或 `register1` 的封装寄存器输入。

1.2.3. 算术模式

算术模式中的 ALM 使用两组两个 4 输入 LUT 连同两个专用全加器。

专用加法器使 LUT 能够执行预加器逻辑；因此，每一个加法器能够将两个 4 输入功能的输出相加。

ALM 支持同时使用加法器的进位输出和组合逻辑输出。在此操作中，加法器输出被忽略。

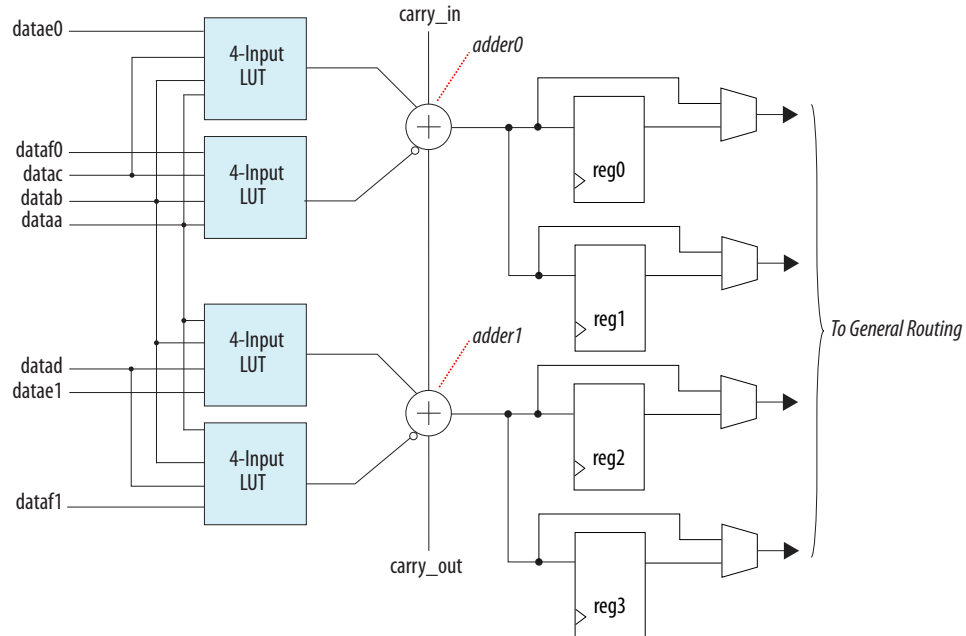
对于可以使用此模式的功能，使用带组合逻辑输出的加法器将节省高达 50% 的资源。

此外，算术模式还支持时钟使能、计数器使能、同步上下控制、加减控制、同步清零以及同步加载。

LAB 本地互联数据输入生成时钟使能、计数器使能、同步上/下和加/减控制信号。对于一个 ALM 中的 4 个 LUT 之间共享的输入而言，这些控制信号是很好的选择。

同步清零和同步加载选项是影响 LAB 中所有寄存器的 LAB 范围信号。您可以单独对每个寄存器禁用或使能这些信号。Quartus Prime 软件自动将计数器未使用的寄存器布局到其它的 LAB 中。

图 11. Arria 10 器件中算术模式的 ALM



进位链 (Carry Chain)

进位链提供了算术或者共享算术模式中专用加法器之间的快速进位功能。

Arria 10 器件中的 2-bit 进位选择功能使 ALM 中的进位链传播延迟减少了一半。进位链能够开始于 LAB 中的第一个 ALM 或者第五个 ALM。最终的 `carry-out` 信号会被传输到 ALM 中，在此 ALM 中驱动到本地、行或列互联。



当实现高扇入（high fan-in）算术功能时，要避免器件中一小块区域内的布线拥塞，连接到下一个 LAB 之前，LAB 能够支持仅使用 LAB 的上半部或者下半部的进位链。这样，LAB 中的剩余的另一半 ALM 可用于实现正常模式中较窄扇入的功能。使用第一个 LAB 中的上面五个 ALM 的进位链进位到列中下一个 LAB 中的 ALM 的上半部。使用第一个 LAB 中的下面五个 ALM 的进位链传输至列中下一个 LAB 中的 ALM 的下半部。您可以旁路 LAB 列的上半部和 MLAB 列的下半部。

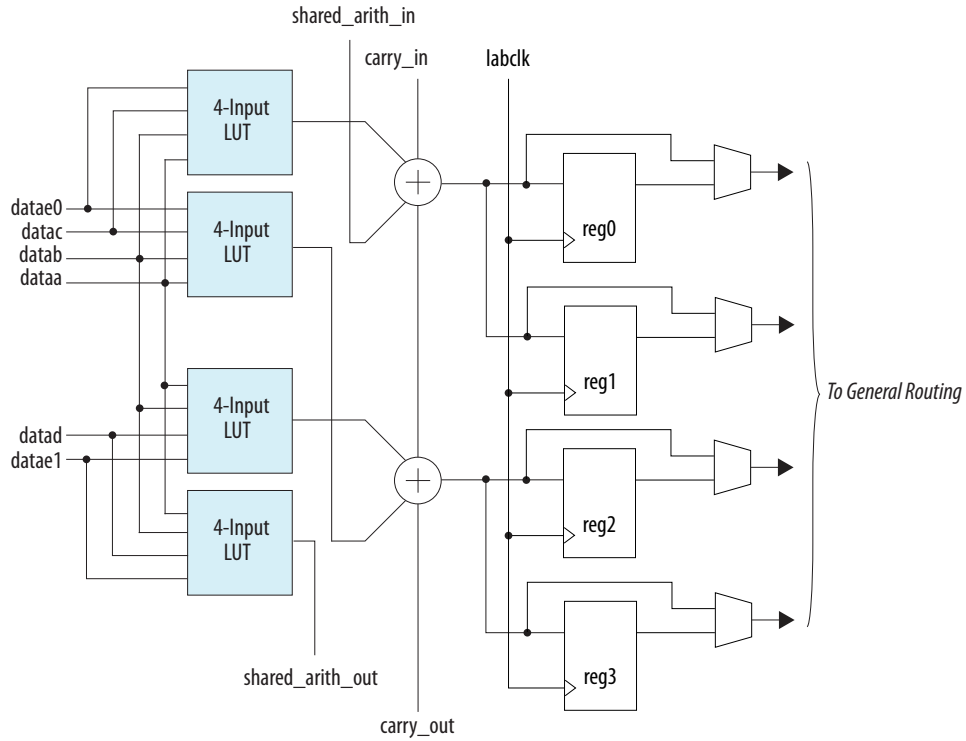
Quartus Prime Compiler 通过自动将 LAB 连接在一起创建超过 20 个 ALM 长的进位链（算术或者共享算术模式的 10 个 ALM）。要增强布局布线，长进位链纵向运行，实现到 TriMatrix 存储器和 DSP 模块的快速横向连接。进位链能够连续整列。

1.2.4. 共享算术模式

共享算术模式的 ALM 能够在 1 个 ALM 中实现 3 输入加法运算。

此模式配置具有四个 4 输入 LUT 的 ALM。每个 LUT 计算三个输入的和，或者计算三个输入的进位。通过使用称作共享算术链的专用连接将进位计算的输出传送到下一个加法器。

图 12. Arria 10 器件中共享算术模式的 ALM



共享算术链

增强算术模式中的共享算术链使 ALM 能够实现 4 输入加法。这样可以显著降低用于实现大型加法器树或者相关器功能的必要资源。

共享算术链可以开始于 LAB 中的第一个或者第六个 ALM。

与进位链类似，交替 LAB 列中的共享算术链的上半部和下半部能够被旁路。这一性能使得共享算术链通过 LAB 中的一半 ALM 进行级联，而剩余的一半用于较窄扇入功能。在每个 LAB 中，列的上半部是可旁路的；而在 MLAB 中，列的下半部是可旁路的。

Quartus Prime Compiler 通过自动将 LAB 连接在一起创建超过 20 个 ALM 长的共享算术链(算术或者共享算术模式中的 10 个 ALM)。要增强布局布线，长共享算术链纵向运行，实现到 TriMatrix 存储器和 DSP 模块的快速横向连接。共享算术链能够连续整列。



1.3. LAB 功耗管理技术

下面的技术用于管理 LAB 中的静态及动态功耗：

- Arria 10 LAB 运行在高性能模式或低功耗模式。Quartus Prime 软件根据您的设计自动优化 LAB 功耗模式。
- 时钟，特别是 LAB 时钟，消耗大量的动态功耗。每个 LAB 的时钟和时钟使能信号都是连接在一起的，由共享的选通时钟进行控制。使用全 LAB 时钟使能信号来选通 LAB 范围时钟，而无需禁用整个时钟树。在您的寄存逻辑的 HDL 代码中使用时钟使能结构。

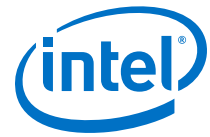
相关链接

[Quartus Prime 手册，功耗优化章节](#)

提供关于在 LAB 中实现静态和动态功耗的详细信息。

1.4. 文档修订历史

日期	版本	修订内容
2017 年 3 月	2017.03.15	重命名为 Intel。
2016 年 10 月	2016.10.31	对 LAB 控制信号部分中的时钟源添加了描述。
2015 年 11 月	2015.11.02	将 <i>Quartus II</i> 更改成 <i>Quartus Prime</i> 。
2013 年 12 月	2013.12.02	首次发布。



2. Arria 10 器件中的嵌入式存储器模块

器件中的嵌入式存储器模块具有高度灵活性，旨在提供一个最佳数量的小型 and 大型存储器阵列，以满足您的设计要求。

相关链接

[Arria 10 器件手册：已知问题](#)

列出了 *Arria 10 器件手册* 章节的计划更新。

2.1. 嵌入式存储器类型

Arria 10 器件包含两种存储器模块类型：

- 20 Kb M20K 模块—专用存储器资源模块。M20K 模块是较大型存储器阵列的最佳选择，并提供大量独立端口。
- 640 比特存储器逻辑阵列模块(MLAB)—增强型存储器模块由双功能逻辑阵列模块(LAB)配置而成。MLAB 是宽而浅型存储器阵列的最佳选择。优化 MLAB 以实现数字信号处理(DSP)应用的移位寄存器宽而浅型 FIFO 缓冲器和滤波延迟线。每个 MLAB 由 10 个自适应逻辑模块(ALM)组成。在 Arria 10 器件中，这些 ALM 可配置成 10 个 32 x 2 模块，这样每个 MLAB 有 1 个 32 x 20 简单双端口 SRAM 模块。

相关链接

[嵌入式单元\(EC\)](#)

提供了关于嵌入式单元的信息



2.1.1. Arria 10 器件中的嵌入式存储器性能

表 1. Arria 10 器件中的嵌入式存储器性能和分布

器件系列	型号	M20K		MLAB		总 RAM Bit(Kb)
		模块数	RAM Bit (Kb)	模块数	RAM Bit (Kb)	
Arria 10 GX	GX 160	440	8,800	1,680	1,050	9,850
	GX 220	587	11,740	2,703	1,690	13,430
	GX 270	750	15,000	3,922	2,452	17,452
	GX 320	891	17,820	4,363	2,727	20,547
	GX 480	1,431	28,620	6,662	4,164	32,784
	GX 570	1,800	36,000	8,153	5,096	41,096
	GX 660	2,131	42,620	9,260	5,788	48,408
	GX 900	2,423	48,460	15,017	9,386	57,846
	GX 1150	2,713	54,260	20,774	12,984	67,244
Arria 10 GT	GT 900	2,423	48,460	15,017	9,386	57,846
	GT 1150	2,713	54,260	20,774	12,984	67,244
Arria 10 SX	SX 160	440	8,800	1,680	1,050	9,850
	SX 220	587	11,740	2,703	1,690	13,430
	SX 270	750	15,000	3,922	2,452	17,452
	SX 320	891	17,820	4,363	2,727	20,547
	SX 480	1,431	28,620	6,662	4,164	32,784
	SX 570	1,800	36,000	8,153	5,096	41,096
	SX 660	2,131	42,620	9,260	5,788	48,408

2.2. Arria 10 器件的嵌入式存储器设计指南

为确保设计成功，有几方面需要考虑。这些设计指南适用于该器件系列的所有型号，除非另有说明。

2.2.1. 考虑存储器模块选择

Quartus Prime 根据用户存储器设计的速度与大小，自动划分实现时存储器模块的数量与配置方式。例如，为提高设计性能，Quartus Prime 软件可能将由 1 块 RAM 实现的存储器设计扩展为由多块 RAM 来实现。

要手动分配给存储器一个指定模块大小，需要使用参数编辑器中的 RAM IP 内核。

对于 MLAB，您可以使用 Quartus Prime 软件通过仿真来实现单端口 SRAM。仿真能使额外的逻辑资源使用的最少。

由于 MLAB 的复用体系结构，只有数据输入寄存器，输出寄存器和写地址寄存器在模块中可用。MLAB 从 ALM 获取读地址寄存器。

注意: 对于 Arria 10 器件，Resource Property Editor 和 TimeQuest Timing Analyzer 报告 M20K 模块的位置为 EC_X<number>_Y<number>_N<number>，尽管允许的分配位置是 M20K_X<number>_Y<number>_N<number>。Embedded Cell (EC)是 M20K 模块的子位置。

2.2.2. 指南：实现外部冲突消解

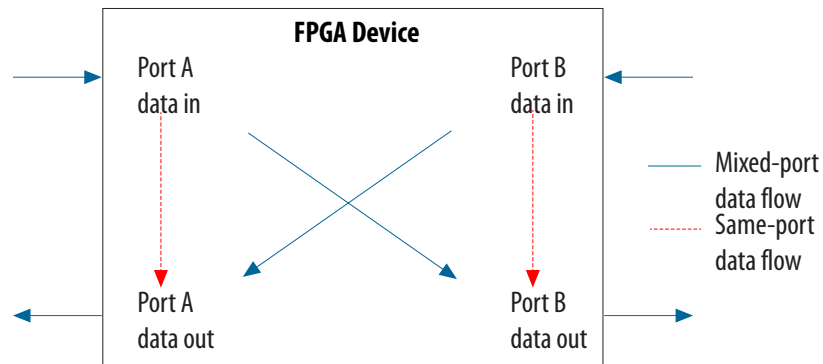
在真双端口 RAM 模式中可以同时对同一存储器位置执行两个写操作。然而，存储器模块内部没有针对同时对同一地址执行两个写操作的冲突消解电路。要避免写入未知数据到地址中，就需要从存储器模块的外部实现冲突消解逻辑。

2.2.3. 指南：定制 Read-During-Write 行为

定制存储器模块的 read-during-write 行为以满足您的设计要求。

图 13. Read-During-Write 数据流程

此图显示了两种类型的 read-during-write 操作—相同端口和混合端口之间的差异。



2.2.3.1. 相同端口 Read-During-Write 模式

相同端口 Read-During-Write 模式应用于单端口 RAM 或者真双端口 RAM 的同一端口。

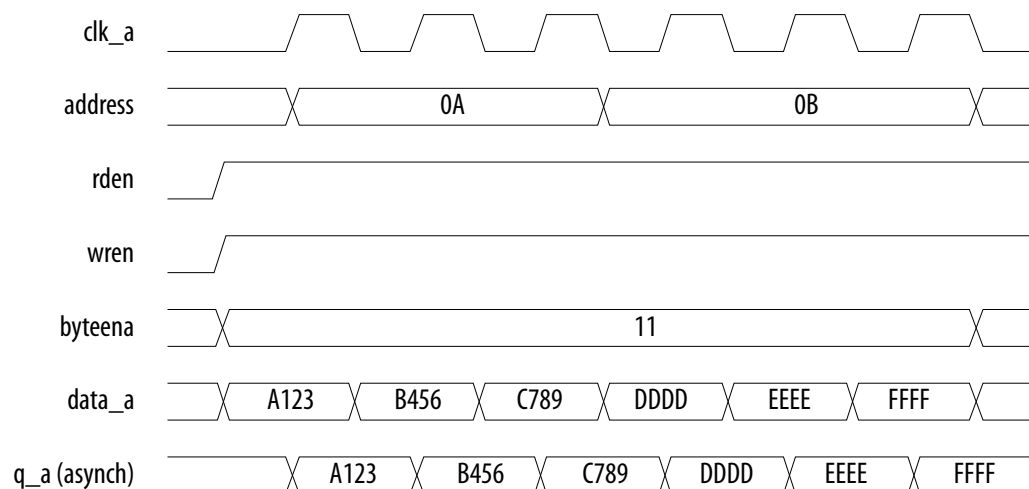
表 2. 相同端口 Read-During-Write 模式下嵌入式存储器模块的输出模式

此表列出了选择相同端口 read-during-write 模式的嵌入式存储器模块时的可用输出模式。

输出模式	存储器类型	说明
"new data" (直通)	M20K	新数据出现在同一时钟周期的上升沿，在此时钟周期上写入新数据。
"don't care"	M20K, MLAB	RAM 对 read-during-write 操作输出"don't care"值。

图 14. 相同端口 Read-During-Write: New Data 模式

此图显示了“new data”模式下相同端口 read- during- write 行为的功能波形样本。



2.2.3.2. 混合端口 Read-During-Write 模式

混合端口 read-during-write 模式应用于简单和真双端口 RAM 模式，其中两个端口使用同一时钟对同一存储器地址执行读写操作，一个端口读数据，一个端口写数据。

表 3. 混合端口 Read-During-Write 模式下 RAM 的输出模式

输出模式	存储器类型	说明
"new data"	MLAB	对不同端口的 read-during-write 操作会导致 MLAB 寄存输出在数据被写入到 MLAB 存储器后的下一个时钟上升沿显示为“new data”。 仅当输出被寄存时才可使用此模式。
"old data"	M20K, MLAB	对不同端口的 read-during-write 操作会导致 RAM 输出端在相应地址上显示“old data”值。 对于 MLAB，仅当输出被寄存时才可使用此模式。
"don't care"	M20K, MLAB	RAM 输出“don' t care”或者“unknown”值。 <ul style="list-style-type: none"> 对于 M20K 存储器，Quartus Prime 软件不分析读写操作之间的时序。 对于 MLAB，默认情况下 Quartus Prime 软件分析读写操作之间的时序。要禁用此行为，需要开启 Do not analyze the timing between write and read operation. Metastability issues are prevented by never writing and reading at the same address at the same time 选项。
"constrained don't care"	MLAB	RAM 输出“don' t care”或“unknown”值。Quartus Prime 软件分析 MLAB 中的读写操作之间的时序。

图 15. 混合端口 Read-During-Write: New Data 模式

此图显示了“new data”模式的混合端口 read- during-write 行为的功能波形样例。

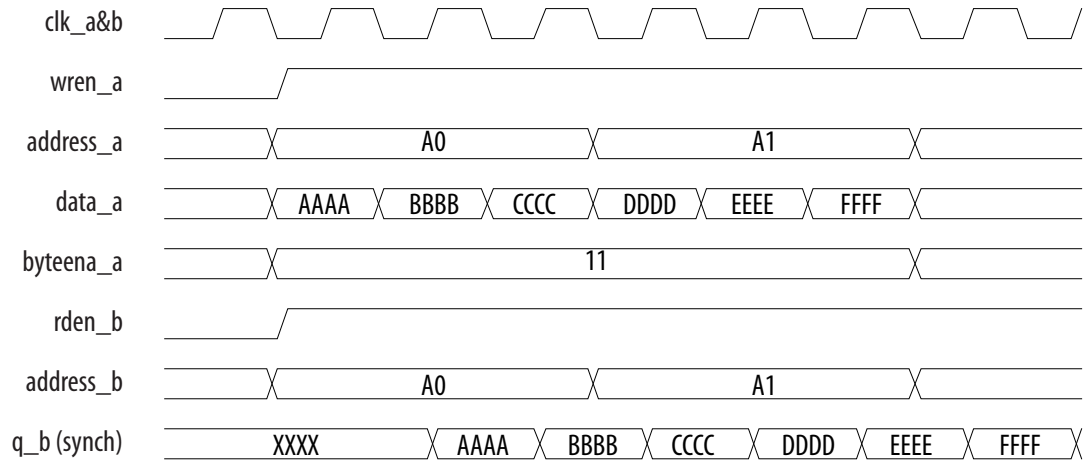
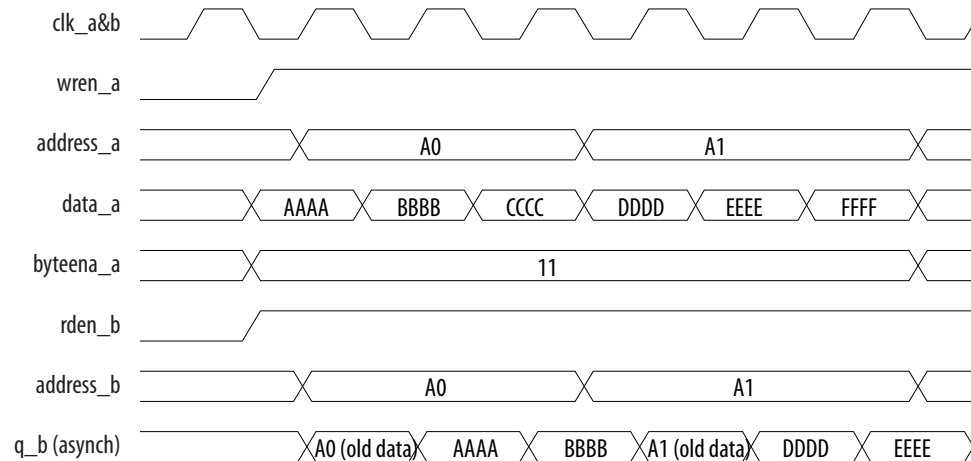


图 16. 混合端口 Read-During-Write: Old Data 模式

此图显示了“old data”模式的混合端口 read- during-write 行为的功能波形样例。



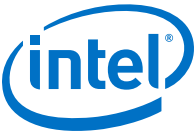
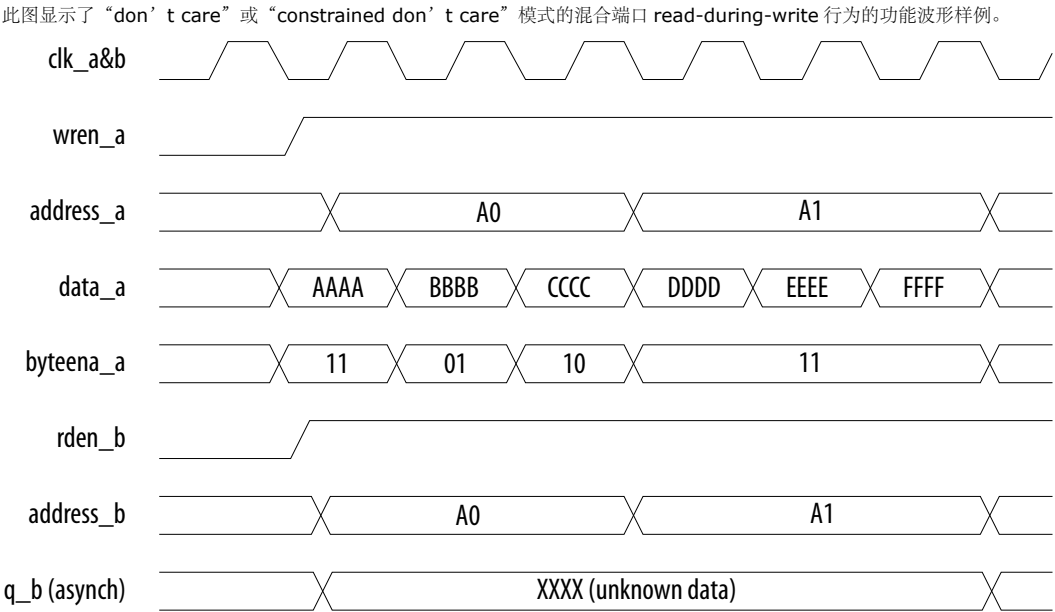


图 17. 混合端口 Read-During-Write: Don' t Care 或 Constrained Don' t Care 模式



在双端口 RAM 模式中，如果输入寄存器具有相同时钟，那么就支持混合端口 read-during-write 操作。

相关链接

嵌入式存储器(RAM: 1-PORT、RAM: 2-PORT、ROM: 1-PORT 和 ROM: 2-PORT)用户指南
提供了关于用于控制 read-during-write 行为的 RAM IP 内核的详细信息。

2.2.4. 指南：考虑上电状态和存储器初始化

如果您正在设计求得上电初始值的逻辑，那么需要考虑不同类型存储器模块的上电状态。

表 4. 嵌入式存储器模块的初始上电值

存储器类型	输出寄存器	上电值
MLAB	使用	零(清零)
	旁路	读存储器内容
M20K	使用	零(清零)
	旁路	零(清零)

默认情况下，Quartus Prime 软件将 Arria 10 器件中的 RAM 单元初始化成零，除非指定一个 .mif 文件。

所有存储器模块都支持通过 .mif 的初始化。您可以在 Quartus Prime 软件中创建 .mif 文件，当在设计中例化一个存储器时，可以指定同 RAM IP 内核的使用。即使存储器被预初始化(例如：使用 .mif)，它也仍然通过其输出清零上电。

相关链接

- 嵌入式存储器(RAM: 1-PORT、RAM: 2-PORT、ROM: 1-PORT 和 ROM: 2-PORT)用户指南提供了关于 .mif 文件的详细信息。
- Quartus Prime 手册第一卷：设计和综合提供了关于 .mif 文件的详细信息。

2.2.5. 指南：控制时钟来降低功耗

在您的设计中降低每个存储器模块的 AC 功耗：

- 使用 Arria 10 存储器模块时钟使能，使您能够控制每个存储器模块的时钟。
- 使用读使能信号来确保读操作仅在必要时出现。如果您的设计不要求 read-during-write，那么在写操作期间或者无存储器操作期间，可以通过置低读使能信号以降低功耗。
- 使用 Quartus Prime 软件自动将未使用的存储器模块置于低功耗状态来降低静态功耗。

2.3. 嵌入式存储器特性

表 5. Arria 10 器件的存储器特性

此表汇总了嵌入式存储器模块所支持的特性。

特性	M20K	MLAB
最大操作频率	730 MHz	700 MHz
RAM 总位数(包括奇偶校验位)	20,480	640
奇偶校验位	支持	—
字节使能	支持	支持
Packed 模式	支持	—
地址时钟使能	支持	—
简单双端口混合位宽	支持	—
真双端口混合位宽	支持	—
FIFO 缓存混合位宽	支持	—
存储器初始化文件(.mif)	支持	支持
混合时钟模式	支持	支持
完全同步存储器	支持	支持
异步存储器	—	仅用于直通(flow-through)读存储器操作。
上电状态	输出端口清零。	<ul style="list-style-type: none"> • 寄存的输出端口—清零。 • 未寄存的输出端口—读存储器内容。
异步清零	输出寄存器和输出锁存器	输出寄存器和输出锁存器
读/写操作触发	时钟上升沿	时钟上升沿
继续...		



特性	M20K	MLAB
相同端口 read-during-write	输出端口设为"new data"或"don't care"。	输出端口设为"don't care"。
混合端口 read-during-write	输出端口设为"old data"或"don't care"。	输出端口设为"old data", "new data", "don't care"或者"constrained don't care"。
ECC 支持	使用 Quartus Prime 的软核 IP 支持。 x32 宽的简单双端口模式的内置支持。	使用 Quartus Prime 的软核 IP 支持。

相关链接

嵌入式存储器(RAM: 1-PORT、RAM: 2-PORT、ROM: 1-PORT 和 ROM: 2-PORT)用户指南
提供了关于嵌入式存储器特性的详细信息。

2.4. 嵌入式存储器模式

表 6. 嵌入式存储器模块中支持的存储器模式

此表列出并描述了 Arria 10 嵌入式存储器模块中所支持的存储器模式。

存储器模式	M20K Support	MLAB 支持	说明
单端口 RAM	Yes	Yes	一次只能执行一个读或一个写操作。 使用读使能端口控制写操作期间的 RAM 输出端口行为： <ul style="list-style-type: none"> 保留最近有效读使能期间保持的之前值—创建一个读使能端口并通过置低此端口执行写操作。 显示正在写入的新数据，该地址上的旧数据，或者"Don't Care"值(当 read-during-write 出现在同一地址上)—不要创建 read-enable 信号，或者在写操作期间启用读使能。
简单双端口模式	Yes	Yes	你可以对不同位置同时执行读写操作，端口 A 进行写操作，端口 B 进行读操作。
真双端口 RAM	Yes	—	您可以执行两个端口操作的任意组合：在两个不同时钟频率上的两个读操作，两个写操作，或者一个读操作和一个写操作。
移位寄存器(shift-register)	Yes	Yes	存储器模块可用作移位寄存器以节省逻辑单元和布线资源。 这在要求本地数据存储(例如：有限脉冲响应(FIR)滤波器、伪随机数生成器、多通道滤波和自相关和互相关函数)的 DSP 应用中很有用。传统上，使用标准触发器(flip-flop)实现本地数据存储，使用触发器实现大型移位寄存器会消耗大量逻辑资源。 移位寄存器的大小($w \times m \times n$)是由输入数据位宽(w)、抽头(tap)长度(m)和抽头数量(n)决定。通过级联存储器模块，能够实现更大的移位寄存器。
ROM	Yes	Yes	存储器模块可用作 ROM。 <ul style="list-style-type: none"> 使用.mif 或.hex 初始化存储器模块的 ROM 数据。 ROM 的地址行在 M20K 模块中寄存；然而，它们在 MLAB 中可以是未寄存的。 输出可以是寄存的或者是未寄存的。 输出寄存器能够被异步清零。 ROM 的读操作与单端口 RAM 配置的读操作相同。
FIFO	Yes	Yes	存储器模块用作 FIFO 缓存。使用 SCFIFO 和 DCFIFO megafunctions 实现您设计中的单时钟和双时钟异步 FIFO 缓存。 对于使用小而浅的 FIFO 缓存的设计而言，MLAB 是 FIFO 模式的最理想选择。然而，MLAB 不支持混合宽度 FIFO 模式。

警告: 为避免损坏存储器数据，在读写操作期间不要违规任何存储器模块输入寄存器上的建立和保持时间。这一注意事项适用于单端口 RAM，简单双端口 RAM，真双端口 RAM 或者 ROM 模式的存储器模块。

相关链接

- [嵌入式存储器\(RAM: 1-PORT、RAM: 2-PORT、ROM: 1-PORT 和 ROM: 2-PORT\)用户指南](#) 提供了关于存储器模式的详细信息。
- [基于 RAM 的移位寄存器\(ALTSHIFT_TAPS\)宏功能用户指南](#) 提供了关于实现移位寄存器模式的详细信息。
- [SCFIFO 和 DCFIFO IP 内核用户指南](#) 提供了关于实现 FIFO 缓存的详细信息。

2.4.1. 单端口模式的嵌入式存储器配置

表 7. Arria 10 器件的单端口嵌入式存储器配置

下表列出了单端口 RAM 和 ROM 模式所支持的最高配置。

存储器模块	深度(bits)	可编程宽度
MLAB	32	x16, x18 或 x20
	64 ⁽¹⁾	x8, x9, x10
M20K	512	x40, x32
	1K	x20, x16
	2K	x10, x8
	4K	x5, x4
	8K	x2
	16K	x1

2.4.2. 双端口模式的嵌入式存储器配置

表 8. 简单双端口 RAM 模式的存储器配置

下表列出了简单双端口 RAM 模式下的存储器配置。仅在 M20K 模块中支持混合宽度配置。

读端口	写端口									
	16K×1	8K×2	4K×4	4K×5	2K×8	2K×10	1K×16	1K×20	512×32	512×40
16K×1	Yes	Yes	Yes	—	Yes	—	Yes	—	Yes	—
8K×2	Yes	Yes	Yes	—	Yes	—	Yes	—	Yes	—
4K×4	Yes	Yes	Yes	—	Yes	—	Yes	—	Yes	—
4K×5	—	—	—	Yes	—	Yes	—	Yes	—	Yes
2K×8	Yes	Yes	Yes	—	Yes	—	Yes	—	Yes	—
2K×10	—	—	—	Yes	—	Yes	—	Yes	—	Yes
1K×16	Yes	Yes	Yes	—	Yes	—	Yes	—	Yes	—
继续...										

(1) 通过软件仿真进行支持，使用额外的 MLAB 模块。



读端口	写端口									
	16K×1	8K×2	4K×4	4K×5	2K×8	2K×10	1K×16	1K×20	512×32	512×40
1K×20	—	—	—	Yes	—	Yes	—	Yes	—	Yes
512×32	Yes	Yes	Yes	—	Yes	—	Yes	—	Yes	—
512×40	—	—	—	Yes	—	Yes	—	Yes	—	Yes

表 9. 真双端口模式的存储器配置

下表列出了真双端口 RAM 模式的存储器配置。仅在 M20K 模块中支持混合宽度配置。

端口 A	端口 B							
	16K×1	8K×2	4K×4	4K×5	2K×8	2K×10	1K×16	1K×20
16K×1	Yes	Yes	Yes	—	Yes	—	Yes	—
8K×2	Yes	Yes	Yes	—	Yes	—	Yes	—
4K×4	Yes	Yes	Yes	—	Yes	—	Yes	—
4K×5	—	—	—	Yes	—	Yes	—	Yes
2K×8	Yes	Yes	Yes	—	Yes	—	Yes	—
2K×10	—	—	—	Yes	—	Yes	—	Yes
1K×16	Yes	Yes	Yes	—	Yes	—	Yes	—
1K×20	—	—	—	Yes	—	Yes	—	Yes

2.5. 嵌入式存储器时钟模式

本节介绍了 Arria 10 存储器模块的时钟模式。

警告: 为避免损坏存储器中的数据，在读写操作期间不要违反任何存储器模块输入寄存器上的建立和保持时间。

2.5.1. 每种存储器模式的时钟模式

表 10. 每种存储器模式支持的存储器模块时钟模式

时钟模式	存储器模式				
	单端口	简单双端口	真双端口	ROM	FIFO
单一时钟模式	Yes	Yes	Yes	Yes	Yes
读/写时钟模式	—	Yes	—	—	Yes
输入/输出时钟模式	Yes	Yes	Yes	Yes	—
独立时钟模式	—	—	Yes	Yes	—

注意: 在 MLAB 模块的写地址，字节使能和数据输入寄存器上不支持时钟使能信号。

2.5.1.1. 单一时钟模式

在单一时钟模式中，单一时钟与时钟使能一起用于控制存储器模块的所有寄存器。

2.5.1.2. 读/写时钟模式

在读/写时钟模式中，读写端口分别使用单独的时钟。读时钟控制数据输出、读地址和读使能寄存器。写时钟控制数据输入、写地址、写使能和字节使能寄存器。

2.5.1.3. 输入/输出时钟模式

在输入/输出时钟模式中，输入和输出端口分别使用单独的时钟。输入时钟控制所有与存储器模块数据输入相关的寄存器，包括数据、地址、字节使能、读使能和写使能。输出时钟控制数据输出寄存器。

2.5.1.4. 独立时钟模式

在独立时钟模式中，每一个端口（端口 A 与端口 B）分别使用单独的时钟。clock A 控制端口 A 侧上的所有寄存器，而 clock B 则控制端口 B 侧上的所有寄存器。

注意: 您可以对不同的输入和输出寄存器创建独立时钟使能，以控制特定寄存器的关闭,从而节省功耗。在配置 Clock enable 的页面中，点击 **More Options** (clock enable 选项旁)设置所需的独立时钟使能。

2.5.2. 时钟模式中的异步清零

在所有的时钟模式中，异步清零仅用于输出锁存器和输出寄存器。对于独立时钟模式，两个端口都适用。



2.5.3. 同步读/写中的输出读数据

如果使用读/写时钟模式对同一地址同时执行读写操作，那么输出读数据将是未知的。如果要求输出读数据是一个已知值，那么需要使用 **single-clock** 或 **input/output clock** 模式并在 **IP core parameter editor** 中选择相应的 **read-during-write** 行为。

2.5.4. 时钟模式的独立时钟使能

以下时钟模式支持独立时钟使能：

- 读/写时钟模式—支持读写时钟。
- 独立时钟模式—支持两个端口的寄存器。

要降低功耗，您可以使用时钟使能来控制指定寄存器的关闭。

相关链接

指南：控制时钟来降低功耗 (第 28 页)

2.6. 嵌入式存储器模块中的奇偶校验位

以下描述了 M20K 模块的奇偶校验位支持：

- 在 5、10、20 和 40 位长度的数据中，奇偶校验位是每 4 个数据位关联的第 5 个位(bit 4、9、14、19、24、29、34 和 39)。
- 在非奇偶校验数据位宽中，奇偶校验位在读或写操作期间被跳过。
- 在奇偶校验位上不执行奇偶校验功能。

2.7. 嵌入式存储器模块中的字节使能

嵌入式存储器模块支持字节使能控制：

- 字节使能通过屏蔽部分输入数据，实现仅写入数据中的指定字节。未被写入的字节保留之前写入的值。
- 写使能(**wren**)信号与字节使能(**byteena**)信号一起控制 RAM 模块上的写操作。默认情况下，**byteena** 信号是高电平(使能)，仅使用 **wren** 信号控制写操作。
- 字节使能寄存器没有 **clear** 端口。
- 如果使用奇偶校验位，在 M20K 模块上，字节使能功能控制 8 个数据位和 2 个奇偶校验位；在 MLAB 上，字节使能功能控制最宽模式的全部 10 位。
- 字节使能运行在一位热码方式(**one-hot fashion**)。 **byteena** 信号的 LSB 对应于数据总线的 LSB。
- 字节使能信号为高电平有效(**active high**)。

2.7.1. 存储器模块中的字节使能控制

表 11. x20 数据位宽的 byteena 控制

byteena[1:0]	写入的数据比特	
11 (默认)	[19:10]	[9:0]
10	[19:10]	—
01	—	[9:0]

表 12. x40 数据位宽的 byteena 控制

byteena[3:0]	写入的数据比特			
1111 (默认)	[39:30]	[29:20]	[19:10]	[9:0]
1000	[39:30]	—	—	—
0100	—	[29:20]	—	—
0010	—	—	[19:10]	—
0001	—	—	—	[9:0]

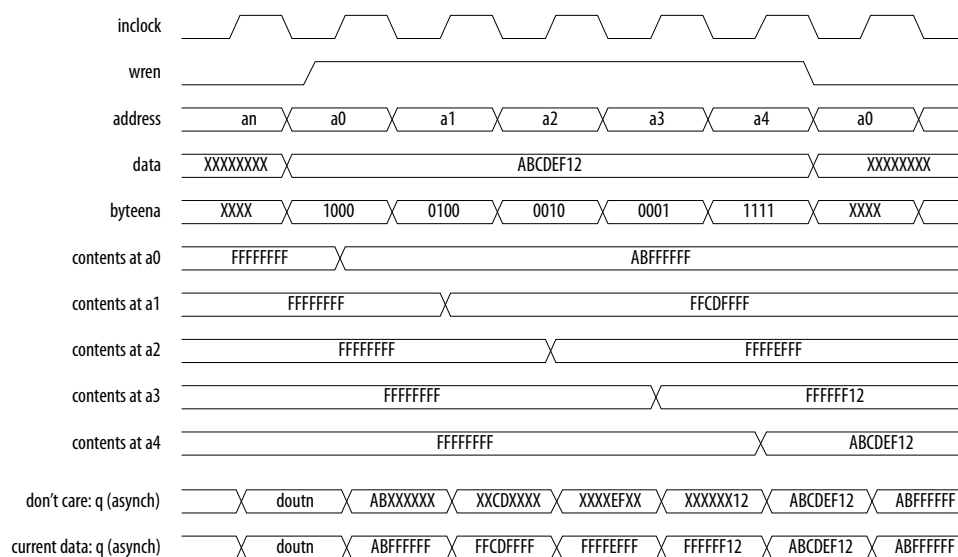
2.7.2. 数据字节输出

在 M20K 模块或者 MLAB 中，当 byte-enable 比特设为 0 时，嵌入式存储器 IP 会将相应的数据字节输出设为“don't care”。您必须确保始终选择 **Get X's for write masked bytes instead of old data when byte enable** 选项。

2.7.3. RAM 模块操作

图 18. 字节使能功能波形

此图显示了 wren 和 byteena 信号是如何控制 RAM 模块的操作。



2.8. 存储器模块 Packed 模式支持

M20K 存储器模块支持 packed 模式。

packed 模式的特性是将两个独立的单端口 RAM 封装进一个存储器模块中。适用于 packed 模式时，Quartus Prime 软件通过将物理 RAM 模块置于真双端口模式并通过地址的最高有效位来区分两个逻辑 RAM 的方式自动实现 packed 模式。每个单端口 RAM 的容量一定不能超过目标模块容量的一半。

2.9. 存储器模块地址时钟使能支持

嵌入式模块支持地址时钟使能，当信号使能时(addressstall = 1)，地址时钟使能会保持之前的地址值。当在双端口模式下配置存储器模块时，每个端口都有各自独立的地址时钟使能。地址时钟使能信号的默认值为低电平(禁用的)。

图 19. 地址时钟使能

此图显示地址时钟使能结构图。端口名 addressstall 是指地址时钟使能。

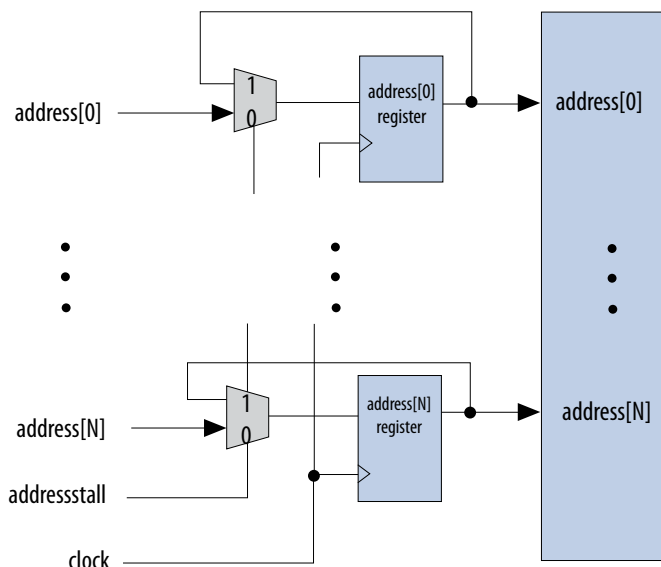


图 20. 读周期中的地址时钟使能波形

此图显示了读周期中的地址时钟使能波形。

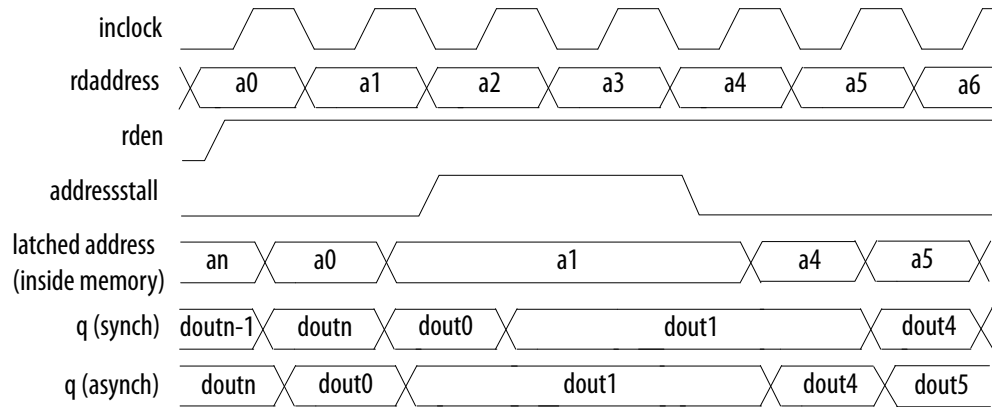
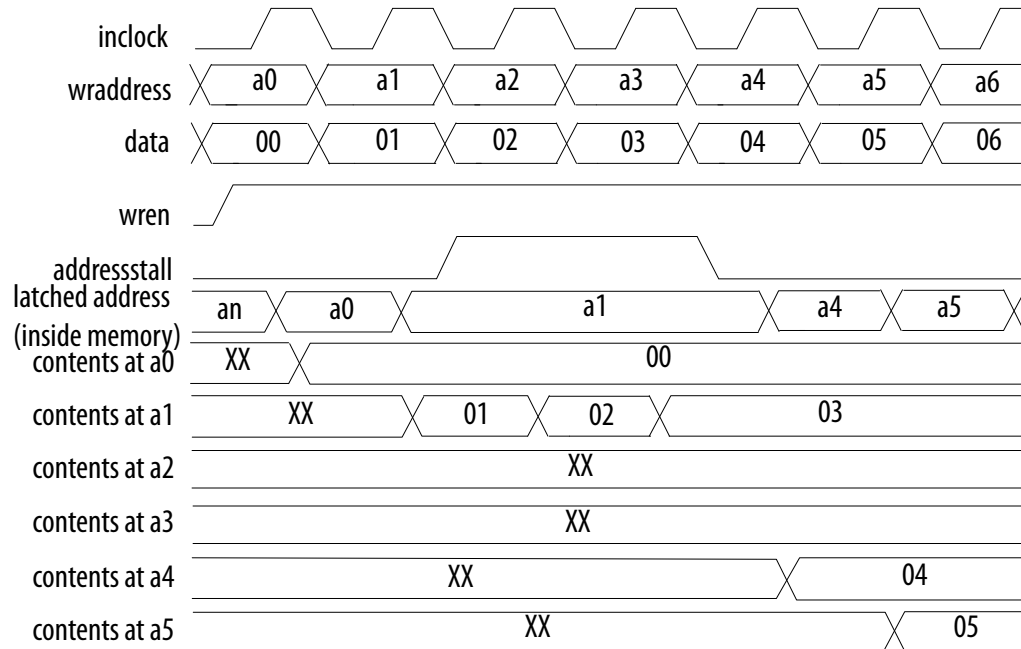


图 21. 写周期中的地址时钟使能波形

此图显示了写周期中地址时钟使能波形。



2.10. 存储器模块异步清零

M20K 存储器模块支持输出锁存器和输出寄存器异步清零。如果您的 RAM 不使用输出寄存器，那么使用输出锁存器异步清零对 RAM 的输出清零。

清零(**clear**)是一个异步信号，并且随时生成。内部逻辑扩展清零脉冲，直到输出时钟的下一个上升沿。当清零被置位时，输出被清零，并保持清零状态直到下一个读周期。

图 22. Arria 10 器件中的输出锁存器清零 (非 ECC 模式)

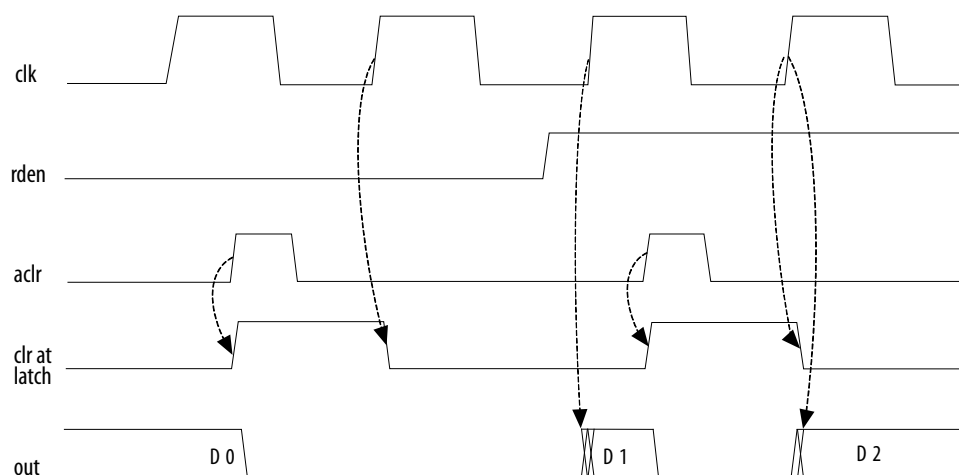
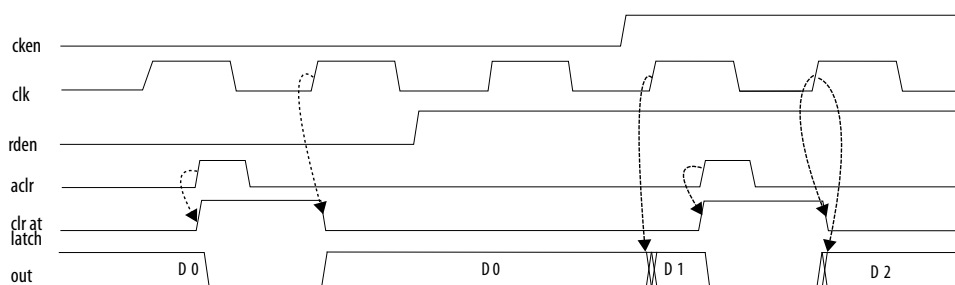


图 23. Arria 10 器件中的输出锁存器清零 (ECC 模式)



2.11. 存储器模块纠错码支持

ECC 使您能够检测并纠正存储器输出上的数据错误。ECC 能够执行 32-bit 字的单一错误纠正，双邻错误纠正和三邻错误检测。然而，ECC 无法检测四个或更多错误。

当 M20K 模块处于 $\times 32$ 宽简单双端口模式中时，它具有对 ECC 的内置支持：

- 当使用 ECC 时，M20K 要比非 ECC 简单双端口模式运行的慢。然而，您可以在输出解码器之前使能可选的 ECC 流水线寄存器，实现比非流水线模式更高的性能，但要以一个周期的延迟为代价。
- M20K ECC 状态与两个 ECC 状态标志信号（e（错误）和 ue（不可纠正的错误））进行通信。状态标志是存储器模块的常规输出的一部分。当使用 ECC 时，您不能访问两个奇偶校验位，因为 ECC 状态标志将它们替换了。

相关链接

[存储模块纠错编码支持](#)

2.11.1. 纠错码真值表

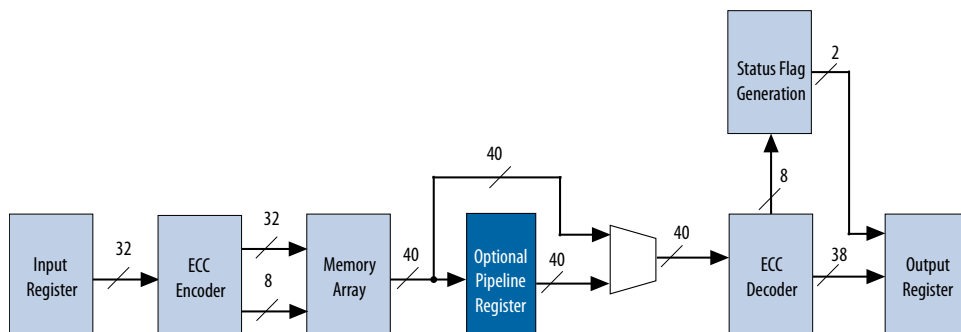
表 13. ECC 状态标志真值表

e (错误) eccstatus[1]	ue (不可纠正的错误) eccstatus[0]	状态
0	0	无错误
0	1	非法
1	0	出现了可纠正的错误，并且在输出上已纠正；然而，存储器阵列尚未更新。
1	1	出现了不可纠正的错误，并且不可纠正的数据出现在输出上。

如果使用 ECC，那么：

- 不能使用字节使能功能。
- 不支持 Read-during-write 旧数据模式。

图 24. M20K 存储器的 ECC 结构图



2.12. 文档修订历史

日期	版本	修订内容
2017 年 3 月	2017.03.15	<ul style="list-style-type: none"> • 重命名为 Intel。 • 在“Arria 10 器件的存储器特性”表中，删除了 Error Correction Code (ECC) 支持下的 MLAB 的奇偶校验位支持。 • 在奇偶校验位主题中删除了 MLAB 的奇偶校验位支持。
2016 年 10 月	2016.10.31	<ul style="list-style-type: none"> • 删除了对 MLAB 模块的地址时钟使能支持。
2015 年 12 月	2015.12.14	<ul style="list-style-type: none"> • 将 Arria 10 GX 660 的 M20K 存储器模块的数量从 2133 更新成 2131，将 RAM 总比特数从 48,448 Kb 更正成 48,408 Kb。
2015 年 11 月	2015.11.02	<ul style="list-style-type: none"> • 更新了“单端口模式的嵌入式存储器配置”和“双端口模式的嵌入式存储器配置”章节。 • 更新了“数据字节输出”部分中的描述。 • 更新了“嵌入式存储器性能和发布”表。 • 将 <i>Quartus II</i> 更改为 <i>Quartus Prime</i>。
2015 年 6 月	2015.06.15	更新了链接。

继续...



日期	版本	修订内容
2015 年 5 月	2015.05.04	<ul style="list-style-type: none"> 将 Mega Wizard Plug-In manager 更新成 IP Core parameter editor。 将 Megafunction 更新成 IP core。
2014 年 8 月	2014.08.18	<ul style="list-style-type: none"> 新增了 ECC 模式中的输出锁存器清零的时序图。 添加了一个注释以阐明：对于 Arria 10 器件，Resource Property Editor 和 TimeQuest Timing Analyzer 报告 M20K 的位置为 EC_X<number>_Y<number>_N<number>。 更新了 Arria 10 GX 660 和 Arria 10 SX 660 的 M20K 中的 RAM 比特值。
2013 年 12 月	2013.12.02	首次发布。

3. Arria 10 器件中的精度可调 DSP 模块

本章描述了 Arria 10 器件中的精度可调数字信号处理(DSP)如何被优化以支持高性能 DSP 应用中的更高比特精度。

3.1. Arria 10 器件中支持的操作模式

表 14. Arria 10 器件中精度可调 DSP 模块支持的操作模式组合及特性。

精度可调 DSP 模块资源	操作模式	支持的操作实例	预加器支持	系数支持	输入级联支持	Chainin 支持	Chainout 支持
1 个精度可调 DSP 模块	定点独立 18 x 19 乘法运算	2	Yes	Yes	Yes ⁽²⁾	No	No
	定点独立 27 x 27 乘法运算	1	Yes	Yes	Yes ⁽³⁾	Yes	Yes
	定点两个 18 x 19 乘法加法器模式	1	Yes	Yes	Yes ⁽²⁾	Yes	Yes
	与 36-bit 输入相加的定点 18 x 18 乘法加法器	1	No	No	No	Yes	Yes
	定点 18 x 19 脉动模式	1	Yes	Yes	Yes ⁽²⁾	Yes	Yes
1 个精度可调 DSP 模块	浮点乘法模式	1	No	No	No	No	Yes
	浮点加法或减法模式	1	No	No	No	No	Yes
	浮点乘法加法或减法模式	1	No	No	No	Yes	Yes
	浮点乘法累加模式	1	No	No	No	No	Yes
	浮点矢量一模式	1	No	No	No	Yes	Yes
	浮点矢量二模式	1	No	No	No	Yes	Yes
2 个精度可调 DSP 模块	复合 18x19 乘法	1	No	No	Yes	No	No

(2) 预加器的两个输入的最大宽度都是 18-bit。当输入级联用作其中一个预加器输入时，输入级联的最大宽度为 18-bit。

(3) 当使能预加器功能时，不支持输入级联。

Intel Corporation. All rights reserved. Agilex, Altera, Arria, Cyclone, Enpirion, Intel, the Intel logo, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

表 15. Arria 10 器件中精度可调 DSP 模块支持的操作模式组合及动态控制特性。

精度可调 DSP 模块资源	操作模式	Dynamic ACCUMULATE	Dynamic LOADCONST	Dynamic SUB	Dynamic NEGATE
1 个精度可调 DSP 模块	定点独立 18 x 19 乘法	No	No	No	No
	定点独立 27 x 27 乘法	Yes	Yes	No	Yes
	定点两个 18 x 19 乘法加法器模式	Yes	Yes	Yes	Yes
	与 36-bit 输入相加的定点 18 x 18 乘法加法器	Yes	Yes	Yes	Yes
	定点 18 x 19 脉动模式	Yes	Yes	Yes	Yes
	浮点乘法模式	No	No	No	No
	浮点加法或减法模式	No	No	No	No
	浮点乘法加法或减法模式	No	No	No	No
	浮点乘法累加模式	Yes	No	No	No
	浮点矢量一模式	No	No	No	No
	浮点矢量二模式	No	No	No	No
2 个精度可调 DSP 模块	复合 18 x 19 乘法	No	No	No	No

3.1.1. 特性

Arria 10 精度可调 DSP 块支持定点运算和浮点运算。

定点运算特性：

- 高性能、功耗优化和全寄存模式的乘法操作
- 18-bit 和 27-bit 字长
- 每个 DSP 模块中有两个 18x19 乘法器或一个 27 x 27 乘法器
- 内置加法，减法和 64-bit 双倍累加寄存器，用于综合乘法结果
- 当禁用预加器时级联 19-bit 或 27-bit，当预加器用于形成滤波应用的抽头延迟线时级联 18-bit
- 级联 64-bit 输出总线，用以在没有外部逻辑支持的情况下将输出结果从一个模块传播至下一个模块
- 对称滤波器的 19-bit 和 27-bit 模式中支持的硬核预加器
- 用于滤波器实现的 18-bit 和 27-bit 模式下的内部系数寄存器块
- 带分布式输出加法器的 18-bit 和 27-bit 脉动有限脉冲响应(FIR)滤波器
- 四舍五入支持

浮点运算特性：

- 支持乘法，加法，乘加和乘减的全强化体系结构
- 具有累加功能的乘法和动态累加器复位控制
- 具有级联求和功能的乘法
- 具有级联减法功能的乘法

- 复合乘法
- 直接矢量点积
- 脉动 FIR 滤波器

相关链接

- [Arria 10 器件手册：已知问题](#)
列出了 *Arria 10 器件手册* 中章节的计划更新。
- [Arria 10 器件概述 - 精度可调 DSP 模块](#)
提供关于每个 Arria 10 器件中乘法器数量的详细信息。

3.2. 资源

表 16. Arria 10 器件中的定点运算的资源

下表列出了每种 Arria 10 器件系列的精度可调 DSP 资源，以比特精度表示。

器件系列	器件种类	精度可调 DSP 模块	独立输入和输出乘法操作数		18 x 19 乘法加法器和模式	18 x 18 与 36 bit 输入相加的乘法加法器
			18 x 19 乘法器	27 x 27 乘法器		
Arria 10 GX	GX 160	156	312	156	156	156
	GX 220	192	384	192	192	192
	GX 270	830	1,660	830	830	830
	GX 320	984	1,968	984	984	984
	GX 480	1,368	2,736	1,368	1,368	1,368
	GX 570	1,523	3,046	1,523	1,523	1,523
	GX 660	1,687	3,374	1,687	1,687	1,687
	GX 900	1,518	3,036	1,518	1,518	1,518
	GX 1150	1,518	3,036	1,518	1,518	1,518
Arria 10 GT	GT 900	1,518	3,036	1,518	1,518	1,518
	GT 1150	1,518	3,036	1,518	1,518	1,518
Arria 10 SX	SX 160	156	312	156	156	156
	SX 220	192	384	192	192	192
	SX 270	830	1,660	830	830	830
	SX 320	984	1,968	984	984	984
	SX 480	1,368	2,736	1,368	1,368	1,368
	SX 570	1,523	3,046	1,523	1,523	1,523
	SX 660	1,687	3,374	1,687	1,687	1,687



表 17. Arria 10 器件中的浮点运算的资源

下表列出了每种 Arria 10 器件系列的精度可调 DSP 的资源，以比特精度表示。

器件系列	器件种类	精度可调 DSP 模块	单精度浮点乘法模式	单精度浮点加法模式	单精度浮点乘法累加模式	Peak 每秒十亿浮点运算 (GFLOP)
Arria 10 GX	GX 160	156	156	156	156	140
	GX 220	192	192	192	192	173
	GX 270	830	830	830	830	747
	GX 320	984	984	984	984	886
	GX 480	1,369	1,368	1,368	1,368	1,231
	GX 570	1,523	1,523	1,523	1,523	1,371
	GX 660	1,687	1,687	1,687	1,687	1,518
	GX 900	1,518	1,518	1,518	1,518	1,366
	GX 1150	1,518	1,518	1,518	1,518	1,366
Arria 10 GT	GT 900	1,518	1,518	1,518	1,518	1,366
	GT 1150	1,518	1,518	1,518	1,518	1,366
Arria 10 SX	SX 160	156	156	156	156	140
	SX 220	192	192	192	192	173
	SX 270	830	830	830	830	747
	SX 320	984	984	984	984	886
	SX 480	1,369	1,368	1,368	1,368	1,231
	SX 570	1,523	1,523	1,523	1,523	1,371
	SX 660	1,687	1,687	1,687	1,687	1,518

3.3. 设计考量

在您的设计中需要考虑以下几点：

表 18. 设计考量

DSP 实现	定点运算	浮点运算
设计单元	<ul style="list-style-type: none"> 操作模式 内部系数和预加器 累加器 Chainout 加法器 	<ul style="list-style-type: none"> 操作模式 Chainout 加法器

Quartus Prime 软件提供以下设计模板用于实现 Arria 10 器件中的 DSP 模块。

表 19. Arria 10 器件中的 DSP 设计模板

操作模式	可用的设计模板
18 x 18 Independent Multiplier Mode	Single Multiplier with Preadder and Coefficient
27 x 27 Independent Multiplier Mode	<ul style="list-style-type: none"> M27x27 with Dynamic Negate M27x27 with Preadder and Coefficient M27x27 with Input Cascade, Output Chaining, Accumulator, Double Accumulator and Preload Constant
Multiplier Adder Sum Mode	<ul style="list-style-type: none"> M18x19_sumof2 with Dynamic Sub and Dynamic Negate M18x19_sumof2 with Preadder and Coefficient M18x19_sumof2 with Input Cascade, Output Chaining, Accumulator, Double Accumulator and Preload Constant
18 x 19 Multiplication Summed with 36-Bit Input Mode	<ul style="list-style-type: none"> M18x19_plus36 with Dynamic Sub and Dynamic Negate M18x19_plus36 with Input Cascade, Output Chaining, Accumulator, Double Accumulator and Preload Constant
18-bit Systolic FIR Mode	<ul style="list-style-type: none"> M18x19_systolic with Preadder and Coefficient M18x19_systolic with Input Cascade, Output Chaining, Accumulator, Double Accumulator and Preload Constant

通过执行下面步骤可以获得设计模板：

1. 在 Quartus Prime 软件中，打开一个新的 Verilog HDL 或 VHDL 文件。
2. 在 Edit 标签中，点击 **Insert Template**。
3. 在 Insert Template 窗口提示中，根据喜好的设计语言点击 **Verilog HDL** 或者 **VHDL**。
4. 点击 **Full Designs** 展开选项。
5. 在选项中，点击 **Arithmetic > DSP Features > DSP Features for 20-nm Device**。
6. 选择符合您设计要求的设计模板，点击 **Insert to append the design template to a new .v or .vhd file**。

3.3.1. 操作模式

Quartus Prime 软件包括用于控制乘法器操作模式的 IP 内核。在 IP Catalog 中输入参数设置后，Quartus Prime 软件将自动配置精度可调 DSP 模块。

也可以使用 Intel® FPGA 的 DSP Builder 和 OpenCL™ 实现精度可调 DSP 模块。

表 20. 操作模式

定点运算	浮点运算
<p>Intel 提供两种方法在一个设计中实现 Arria 10 精度可调 DSP 模块的各种模式—使用 Quartus Prime DSP IP core 和 HDL inferring。</p> <p>在定点运算实现中，Arria 10 精度可调 DSP 模块支持下面的 Quartus Prime IP 内核：</p> <ul style="list-style-type: none"> ALTERA_MULT_ADD ALTMULT_COMPLEX Arria 10 Native Fixed Point DSP IP core 	<p>Intel 提供一种方法，在设计中实现多种模式的 Arria 10 精度可调 DSP 模块—使用 Quartus Prime DSP IP 内核。</p> <p>在浮点运算实现中，Arria 10 精度可调 DSP 模块支持下面的 Quartus Prime IP 内核：</p> <ul style="list-style-type: none"> ALTERA_FP_FUNCTIONS Arria 10 Native Floating Point DSP IP core

相关链接

- Introduction to Intel FPGA IP Cores



- [Integer Arithmetic Megafunctions User Guide](#)
- [Floating-Point Megafunctions User Guide - ALTERA_FP_FUNCTIONS IP Core](#)
- [Quartus Prime Software Help](#)
- [Arria 10 Native Fixed Point DSP IP User Guide](#)

3.3.2. 用于定点运算的内部系数和预加器

当对预加法器功能使能输入寄存器时，这些输入寄存器必须要有相同的时钟设置。

当预加器功能使能时，输入级联支持仅用于 18-bit 模式。

在 18-bit 和 27-bit 模式中，你可以独立使用系数功能和预加器功能。

当在 18-bit 模式下使能内部系数功能时，必须同时使能顶部以及底部系数。

当在 18-bit 模式下使能预加法器功能时，必须同时使能顶部以及底部预加法器。

3.3.3. 用于定点运算的累加器

通过使能位于输出寄存器组与累加器之间的 64-bit 双倍累加寄存器，Arria 10 器件中的累加器支持双倍累加。

3.3.4. Chainout 加法器

表 21. Chainout 加法器

定点运算	浮点运算
您可以使用输出链式路径将另一个 DSP 模块的结果相加。	您可以使用输出链式路径将另一个 DSP 模块的结果相加。 支持某些操作模式： <ul style="list-style-type: none"> • 乘加或乘减模式 • 矢量一模式 • 矢量二模式

3.4. 模块体系结构

Arria 10 精度可调 DSP 模块由下面的单元组成：

表 22. 模块体系结构

DSP 实现	定点运算	浮点运算
模块体系结构	<ul style="list-style-type: none"> • 输入寄存器组 • 流水线寄存器块 • 预加器 • 内部系数 • 乘法器 • 加法器 • 累加器和 chainout 加法器 • 脉动寄存器 • 双倍累加寄存器 • 输出寄存器组 	<ul style="list-style-type: none"> • 输入寄存器组 • 流水线寄存器 • 乘法器 • 加法器 • 累加器和 chainout 加法器 • 输出寄存器组

如果精度可调 DSP 模块没有配置成定点运算脉动 FIR 模式，那么两个脉动寄存器都被旁路。

图 25. Arria 10 器件中定点运算 18 x 19 模式的精度可调 DSP 模块体系结构

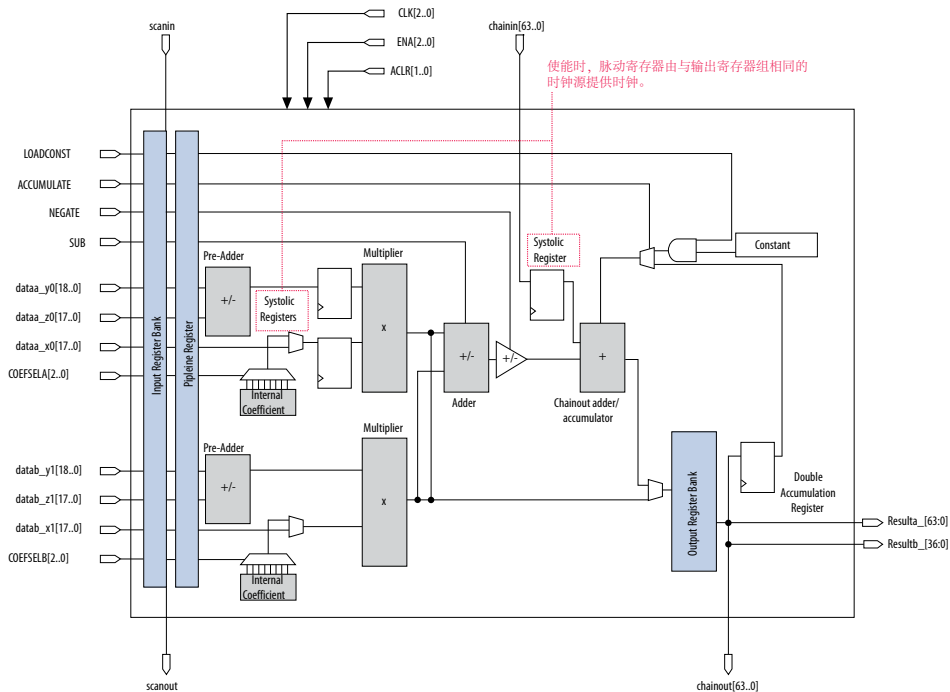


图 26. Arria 10 器件中定点运算 27 x 27 模式的精度可调 DSP 模块体系结构

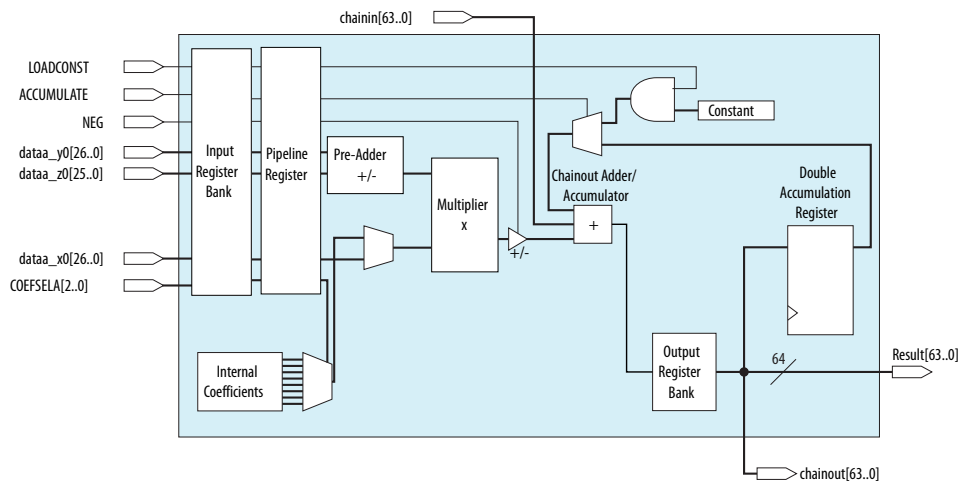
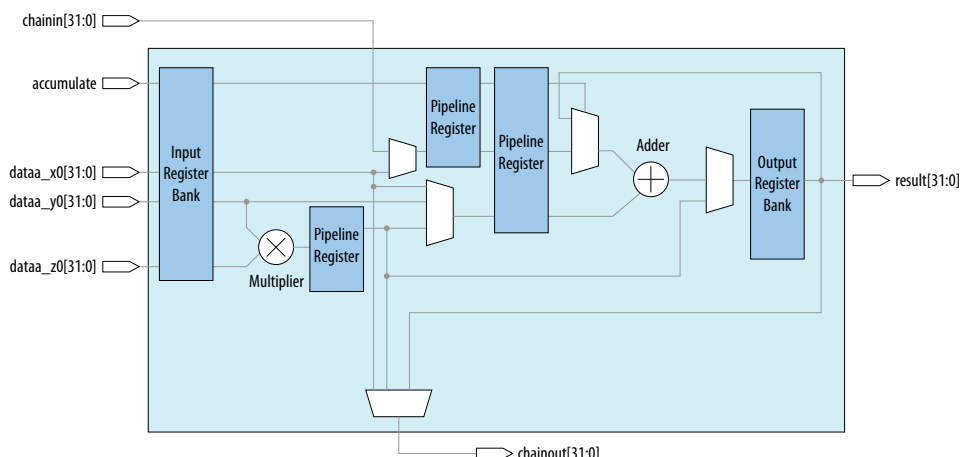


图 27. Arria 10 器件中浮点运算的精度可调 DSP 模块体系结构



3.4.1. 输入寄存器组(Input Register Bank)

表 23. 输入寄存器组

定点运算	浮点运算
<ul style="list-style-type: none"> 数据 动态控制信号 两组延迟寄存器 	<ul style="list-style-type: none"> 数据 动态 ACCUMULATE 控制信号

DSP 模块中所有寄存器都是正边沿触发并在上电时清零。每个乘法器操作数都能够驱动输入寄存器，或者直接驱动乘法器而旁路输入寄存器。

下面的精度可调 DSP 模块信号控制精度可调 DSP 模块中的输入寄存器：

- CLK[2..0]
- ENA[2..0]
- ACLR[0]

在定点运算 18 x 19 模式中，当使用输入级联及 chainout 功能时，您可以使用延迟寄存器来平衡延迟要求。

抽头延迟线功能使您能够从一般布线或者从级联链驱动乘法器输入的顶相(top leg)，18 x 19 模式的 dataa_y0 和 datab_y1，以及仅 27 x 27 模式的 dataa_y0。

3.4.1.1. 用于定点运算的两组延迟寄存器

能够在定点运算 18 x 19 模式下使用的两个延迟寄存器以及输入级联链是顶部延迟寄存器和底部延迟寄存器。在 18 x 19 multiplication summed with 36-bit input 模式和 27 x 27 模式下不支持延迟寄存器。

图 28. Arria 10 器件中定点运算 18 x 19 模式下的一个精度可调 DSP 模块的输入寄存器

此图仅显示数据寄存器，没有显示控制信号的寄存器。

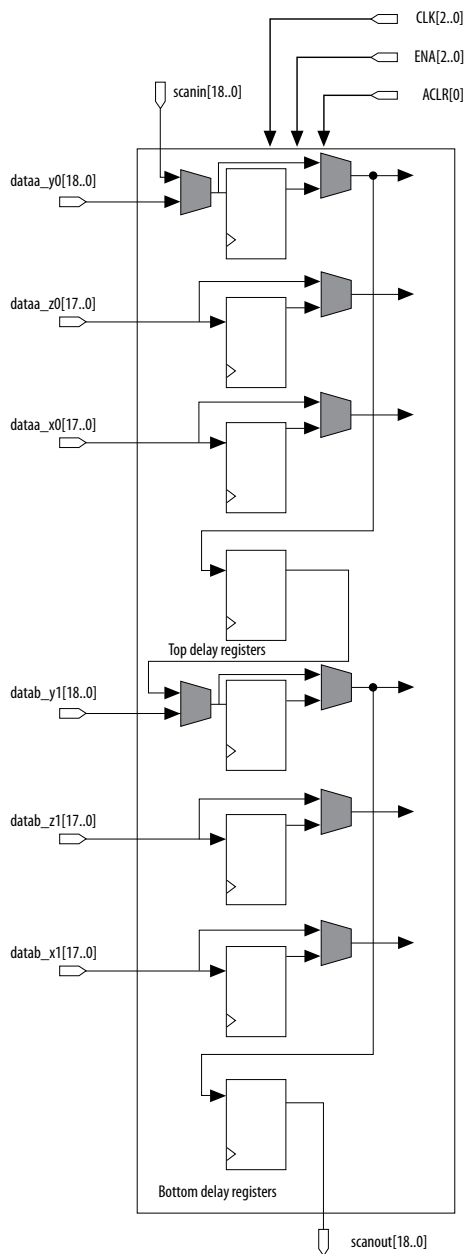
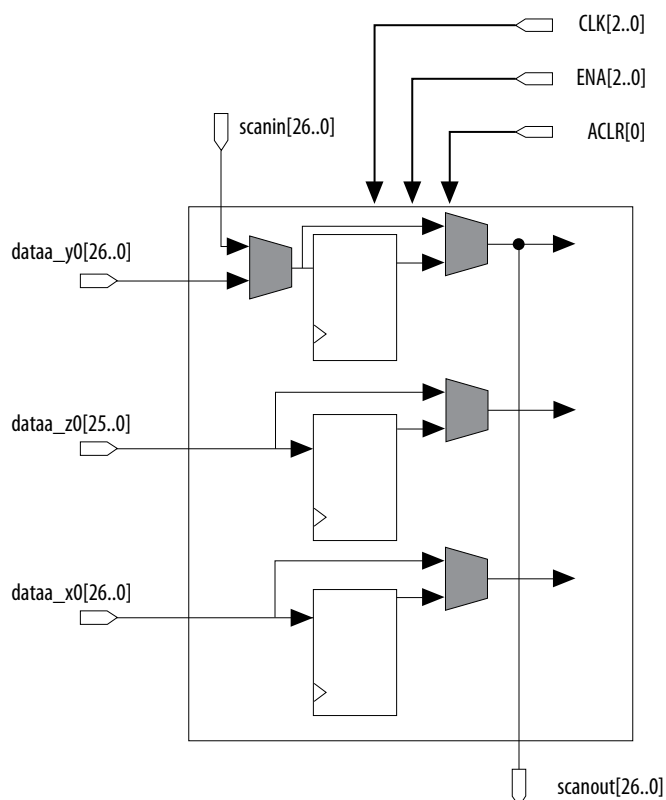


图 29. Arria 10 器件中定点运算 27 x 27 模式下的一个精度可调 DSP 模块的输入寄存器

此图仅显示数据寄存器，没有显示控制信号的寄存器。



3.4.2. 流水线寄存器

流水线寄存器用于获得最大的 **Fmax** 性能。如果不需要高 **Fmax**，那么流水线寄存器可以被旁路。

下面精度可调 DSP 模块信号控制精度可调 DSP 模块中的流水线寄存器：

- CLK[2..0]
- ENA[2..0]
- ACLR[1]

浮点运算有流水线寄存器的 2 个延迟层，可以作以下用途：

- 旁路流水线寄存器的所有延迟层
- 使用流水线寄存器的其中一个延迟层
- 使用流水线寄存器的两个均延迟层

3.4.3. 定点运算的预加器

每个精度可调 DSP 模块有两个 19-bit 预加器。这些预加器可以配置成以下配置：

- 两个独立的 19-bit 预加器
- 一个 27-bit 预加器

预加器支持以下输入配置的加减运算：

- 18 x 19 模式的 18-bit (有符号或无符号)加法或减法
- 27 x 27 模式的 26-bit 加法或减法

当使用同一个 DSP 模块中的两个预加器时，它们必须共享相同的操作类型（加法或减法）。

3.4.4. 定点运算的内部系数

Arria 10 精度可调 DSP 模块具有从动态输入或者内部系数中选择被乘数的灵活性。

对于 18-bit 和 27-bit 模式的被乘数，内部系数最多支持 8 个常数系数。内部系数功能使能时，COEFSELA/COEFSELB 用于控制系数多路复用器的选择。

3.4.5. 乘法器

一个精度可调 DSP 模块可根据乘法器的数据位宽及实现同时执行多个乘法运算。

每个精度可调 DSP 模块中有两个乘法器。这两个乘法器可配置成下面几种操作模式：

表 24. 操作模式

定点运算	浮点运算
<ul style="list-style-type: none"> • 一个 27 x 27 乘法器 • 两个 18 (有符号)/(无符号) x 19 (有符号)乘法器 	一个浮点运算单精度乘法器

相关链接

[操作模式说明 \(第 52 页\)](#)

提供关于乘法器操作模式的详细信息。

3.4.6. 加法器

根据不同的操作模式，您可以按如下使用加法器：

- 一个 55-bit 或 38-bit 加法器
- 一个浮点运算单精度加法器

DSP 实现	使用动态 SUB 端口的加法	使用动态 SUB 端口的减法
定点运算	Yes	Yes
浮点运算	No	No

3.4.7. 用于定点运算的累加器和 Chainout 加法器

Arria 10 精度可调 DSP 模块支持用于定点运算的一个 64-bit 累加器和一个 64-bit 加法器。

下面信号能够动态控制累加器功能：

- NEGATE
- LOADCONST
- ACCUMULATE

通过使能位于输入寄存器块与累加器之间的 64-bit 双倍累加寄存器，累加器支持双倍累加。

在两个定点算术独立 18 x 19 模式中不支持累加器和 chainout 加法器功能。

表 25. 累加器功能和动态控制信号

此表列出了动态信号设置和每种功能的描述。在此表中，X 代表"don't care"。

功能	说明	NEGATE	LOADCONST	ACCUMULATE
Zeroing(归零)	禁用累加器。	0	0	0
Preload(预加载)	结果始终与预加载值相加。在 64-bit 预加载值中，只有一个比特的值能为“1”。它可用作舍入 DSP 结果到 64-bit 结果的任何位置。	0	1	0
Accumulation(累加)	将当前结果与之前累加结果相加。	0	X	1
Decimation + Accumulate(抽取+累加)	此功能将当前结果转换成二补数，然后与之前结果相加。	1	X	1
Decimation + Chainout Adder(抽取+Chainout 加法器)	此功能获取当前结果并将其转换成二补数，然后与之前 DSP 模块的输出相加。	1	0	0

3.4.8. 用于定点运算的脉动寄存器

每个精度可调 DSP 模块有两个脉动寄存器。如果精度可调 DSP 模块没有配置成定点运算脉动 FIR 模式，那么这两个脉动寄存器都被旁路。

第一组脉动寄存器包括 18-bit 和 19-bit 寄存器，分别用于寄存顶部乘法器的 18-bit 和 19-bit 输入。

第二组脉动寄存器用于延迟前一个精度可调 DSP 模块的 chainin 输入。

您必须使用与输出寄存器块相同的时钟源对所有的脉动寄存器提供时钟。必须开启输出寄存器。

3.4.9. 用于定点运算的双倍累加寄存器

双倍累加寄存器是累加器反馈路径中的一个额外寄存器。使能双倍累加寄存器将导致累加器反馈路径中的一个额外时钟周期。

此寄存器具有与输出寄存器块相同的 CLK, ENA 和 ACLR 设置。

通过使能此寄存器能够有两个使用相同数量精度可调 DSP 模块的累加器通道。这在处理交错复杂数据(I, Q)时很有用。

3.4.10. 输出寄存器组(Output Register Bank)

时钟信号的正边沿触发 74-bit 可旁路输出寄存器块，并在上电后清零。

下面的 DSP 模块信号控制 DSP 模块中的输出寄存器：

- CLK[2..0]
- ENA[2..0]
- ACLR[1]

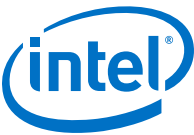


3.5. 操作模式说明

这一部分描述如何配置 Arria 10 精度可调 DSP 模块来有效支持定点运算和浮点运算操作模式：

表 26. 操作模式

定点运算	浮点运算
<ul style="list-style-type: none">独立乘法器模式乘法加法器求和模式独立复数乘法器与 36-Bit 输入相加的 18 x 18 乘法模式脉动 FIR 模式	<ul style="list-style-type: none">乘法模式加法或减法模式乘加或乘减模式乘法累加模式矢量一模式矢量二模式直接矢量点积复数乘法



3.5.1. 定点运算的操作模式

3.5.1.1. 独立乘法器模式

在独立输入与输出乘法器模式中，精度可调 DSP 模块执行通用乘法器的单独乘法运算操作。

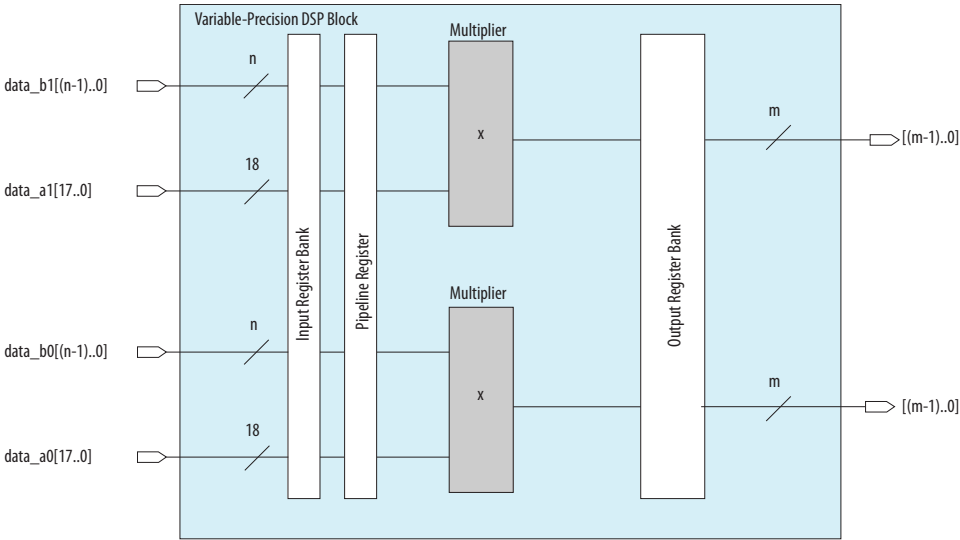
配置	每个模块的乘法器数量
18 (有符号) x 19 (有符号)	2
18 (无符号) x 18 (无符号)	2
27 (有符号或无符号) x 27 (有符号或无符号)	1

3.5.1.1.1. 18 x 18 或者 18 x 19 独立乘法器

图 30. Arria 10 器件中每个精度可调 DSP 模块有两个 18 x 18 或者 18 x 19 独立乘法器

在此图中，变量定义如下：

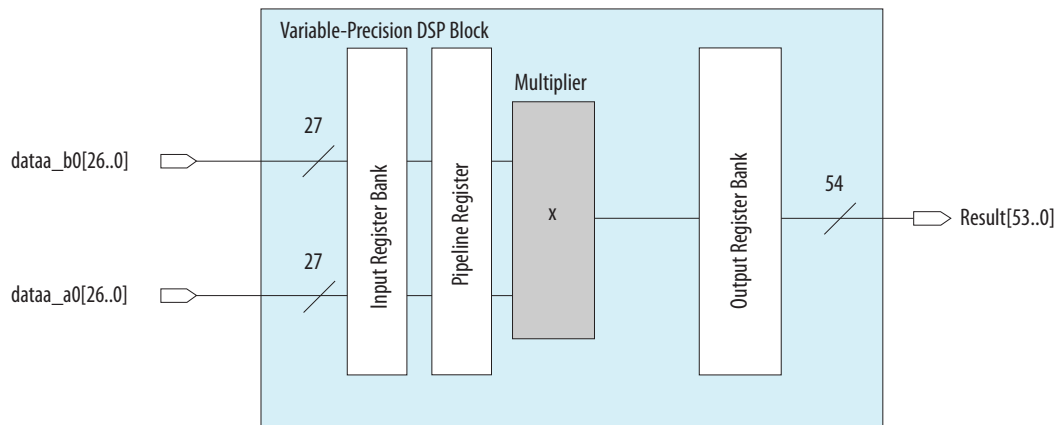
- 对于 18 x 19 操作数， $n = 19$ 和 $m = 37$
- 对于 18 x 18 操作数， $n = 18$ 和 $m = 36$



3.5.1.1.2. 27 x 27 独立乘法器

图 31. Arria 10 器件每个精度可调 DSP 模块一个 27 x 27 独立乘法器模式

在此模式中，当与 chainout 加法器或累加器结合时，result 能够高达 64 bit。



3.5.1.2. 独立复合乘法器

Arria 10 器件支持使用两个定点运算乘法加法器模式的 18 x 19 复合乘法器模式

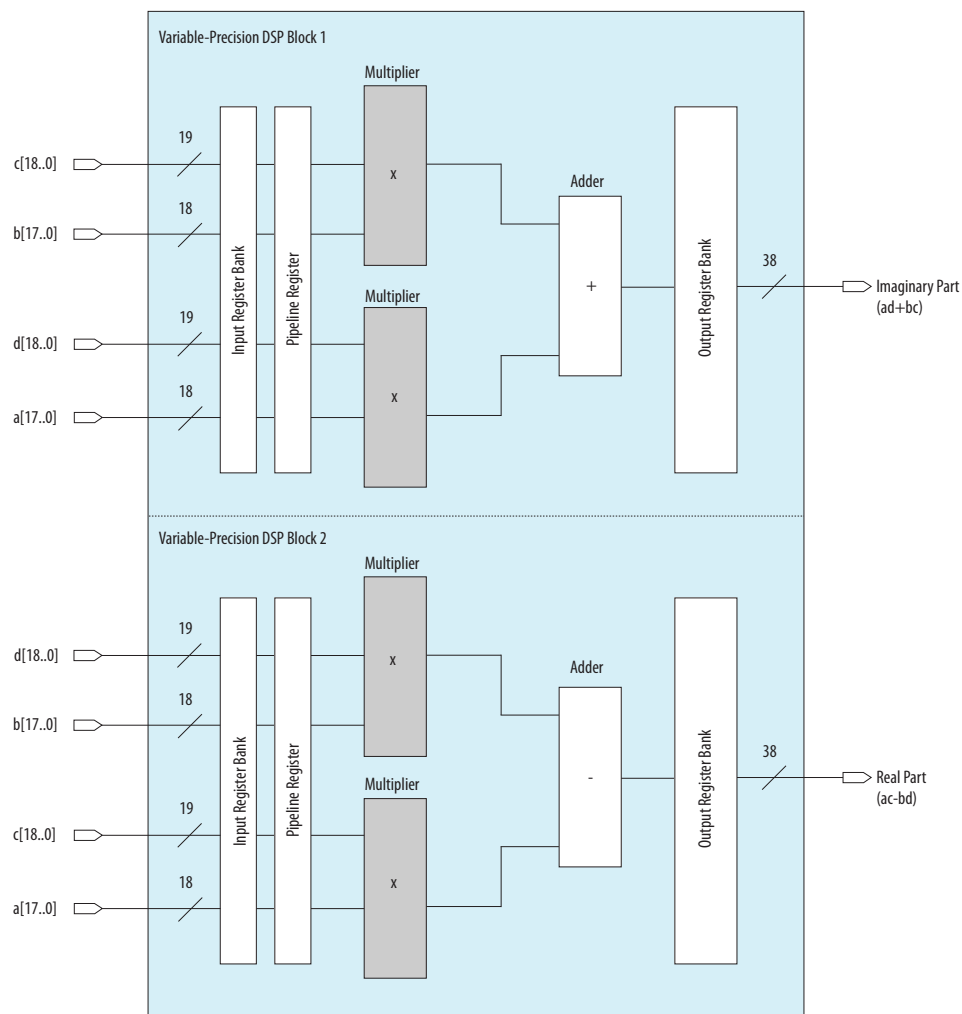
图 32. 复合乘法公式示例

$$(a + jb) \times (c + jd) = [(a \times c) - (b \times d)] + j[(a \times d) + (b \times c)]$$

虚部 $[(a \times d) + (b \times c)]$ 在第一个精度可调 DSP 模块中实现，而实部 $[(a \times c) - (b \times d)]$ 在第二个精度可调 DSP 模块中实现。

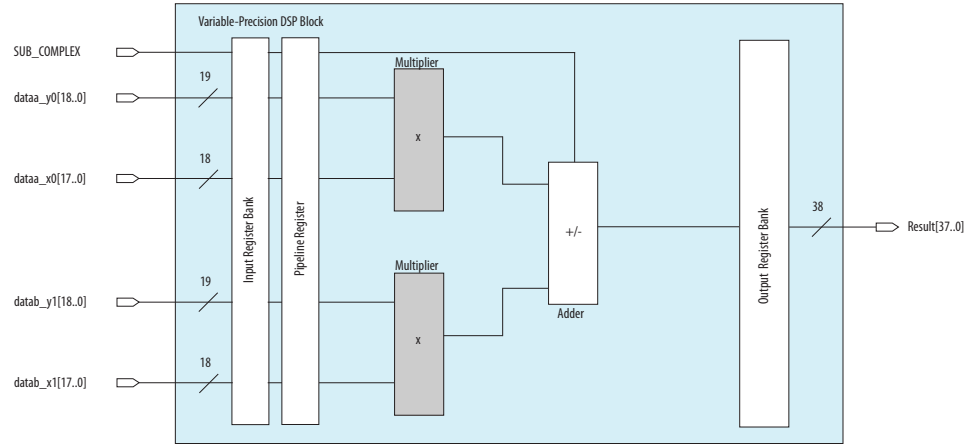
3.5.1.2.1. 18 x 19 复合乘法器

图 33. Arria 10 器件中一个具有两个精度可调 DSP 模块的 18 x 19 复合乘法器



3.5.1.3. 乘法加法器求和模式

图 34. Arria 10 器件基于一个精度可调 DSP 模块的两个 18 x 19 乘法器的和

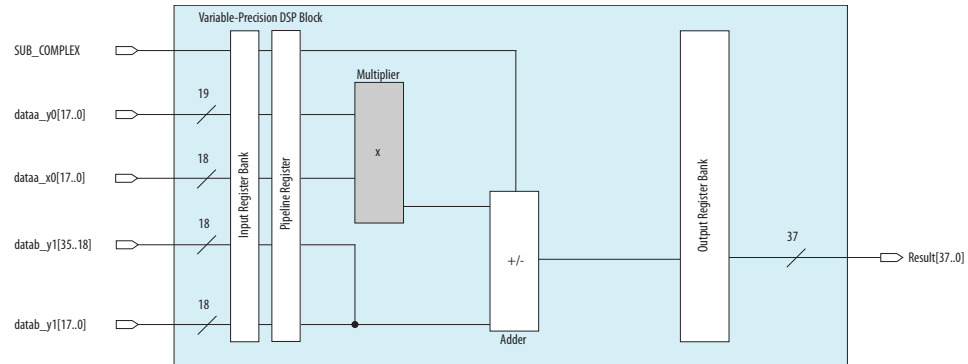


3.5.1.4. 与 36-Bit 输入相加的 18 x 19 乘法运算

Arria 10 精度可调 DSP 模块支持一个与 36-bit 输入相加的 18 x 19 乘法运算。

使用顶部乘法器对 18 x 19 乘法提供输入，而底部乘法器被旁路。datab_y1[17..0] 和 datab_y1[35..18] 信号级联生成一个 36-bit 输入。

图 35. Arria 10 器件中与 36-Bit 输入相加的 18 x 19 乘法模式



3.5.1.5. 脉动 FIR 模式

FIR 滤波器的基本结构包括一系列乘法运算和其后的一个加法运算。

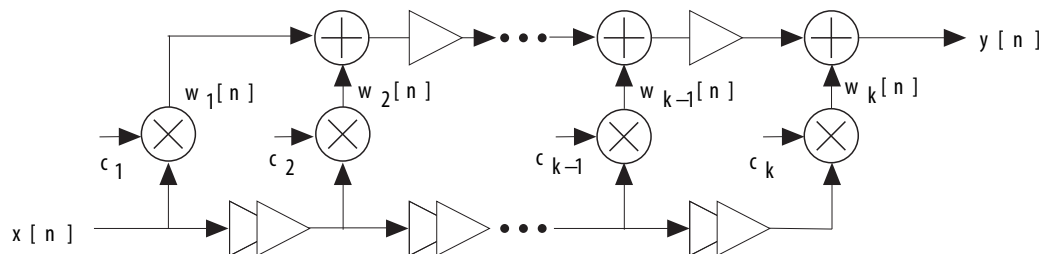
图 36. 基本 FIR 滤波器公式

$$y[n] = \left(\sum_{i=1}^k w_i[n - k + i] + w_1[n - k + 2] \right)$$

Where i start from 1, $w_i[n] = c_i x[n - 2i + 2]$

根据抽头数量和输入大小，链接大量加法器能够导致相当大的延迟。要解决该延迟性能问题，使用脉动形式(systolic form)与每个抽头中的额外延迟单元以增加延迟为代价来提高性能。

图 37. 脉动 FIR 滤波器等效电路



Arria 10 精度可调 DSP 模块支持以下脉动 FIR 结构：

- 18-bit
- 27-bit

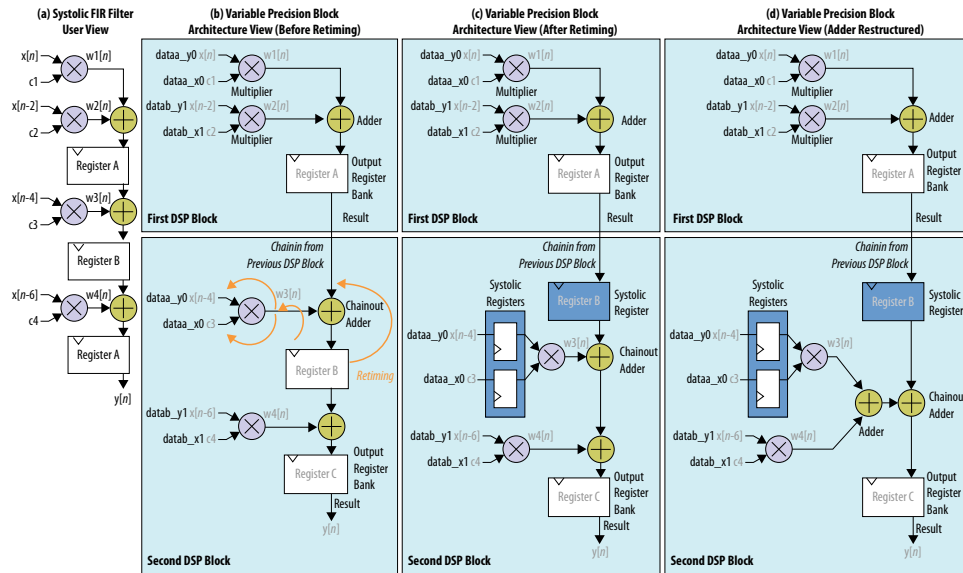
在脉动 FIR 模式中，乘法器的输入来自四组不同的数据源：

- 两个动态输入
- 一个动态输入和一个系数输入
- 一个系数输入和一个预加器输出
- 一个动态输入和一个预加器输出

3.5.1.5.1. 映射脉动模式用户视图到精度可调模块体系结构视图

下图显示了通过使用 Arria 10 精度可调 DSP 模块(d) 重新时序化寄存器和重构加法器能够实现脉动 FIR 过滤器(a)的用户视图。如(b)中所示，在 chainin, dataa_y0 和 dataa_x0 输入路径上 Register B 能够重新时序化到脉动寄存器。(c)中显示了寄存器重新时序化的最终结果。如(d)中所示，chainout 加法器将通过重新结构化加法器输入和位置得到的两个乘法器结果的和与 chainin 输入相加。

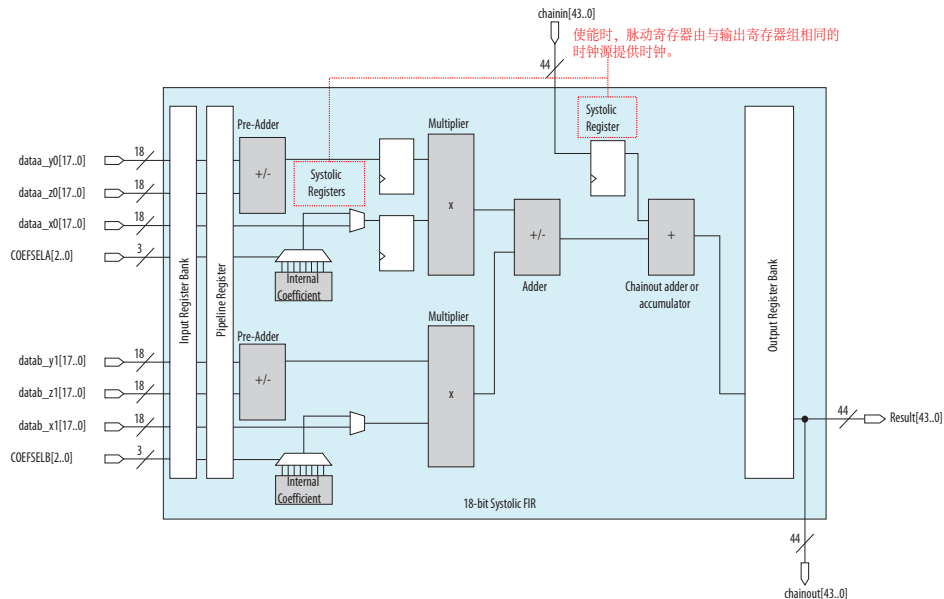
图 38. 映射脉动模式用户视图到精度可调模块体系结构视图



3.5.1.5.2. 18-Bit 脉动 FIR 模式

在 18-bit 脉动 FIR 模式中，加法器配置成双 44-bit 加法器，因此当使用 18×19 操作模式时会产生 7 bits 成本(overhead)，从而产生 37-bit 结果。这使得总共 16 个 18×19 乘法器或者 Arria 10 精度可调 DSP 模块能够级联在一起组成一个脉动 FIR 结构。

图 39. Arria 10 器件的 18-Bit 脉动 FIR 模式

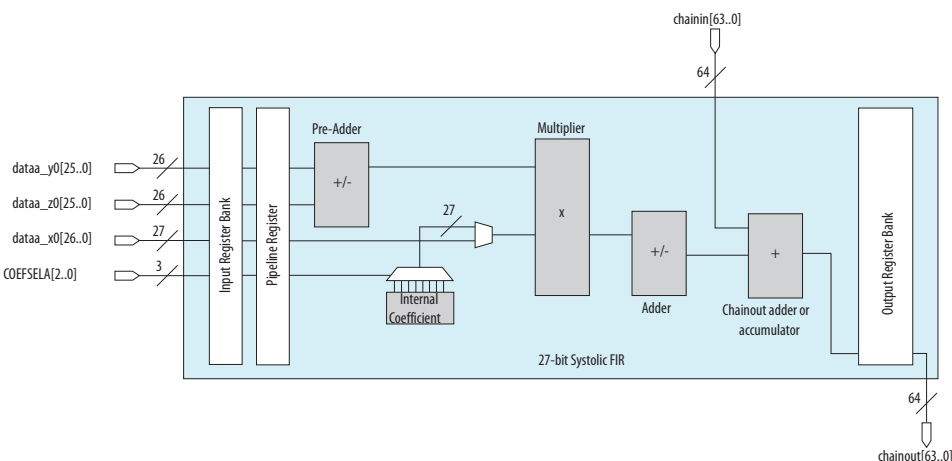


3.5.1.5.3. 27-Bit 脉动 FIR 模式

在 27-bit 脉动 FIR 模式中，chainout 加法器或累加器配置成 64-bit 操作，当使用 27-bit 数据 (54-bit 乘积) 时提供 10 比特成本(overhead)。这使得总共 11 个 27 x 27 乘法器或者 11 个 Arria 10 精度可调 DSP 模块级联在一起组成一个脉动 FIR 结构。

27-bit 脉动 FIR 模式支持每个 DSP 模块一阶脉动滤波器(one stage systolic filter)的实现。在此模式中不需要脉动寄存器。

图 40. Arria 10 器件的 27-Bit 脉动 FIR 模式



3.5.2. 浮点运算的操作模式

3.5.2.1. 单一浮点运算功能

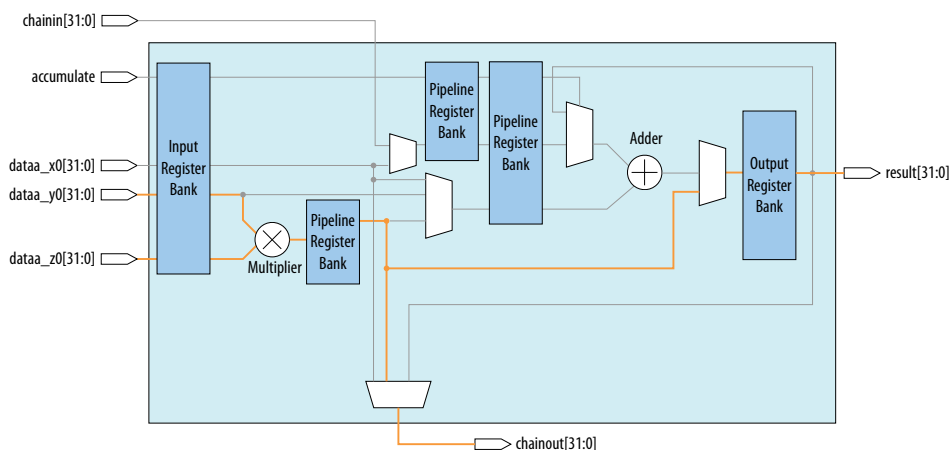
一个浮点运算 DSP 可以执行：

- 乘法模式
- 加法或减法模式
- 乘法累加模式

3.5.2.1.1. 乘法模式

此模式使您能够应用基本的浮点乘法($y*z$)。

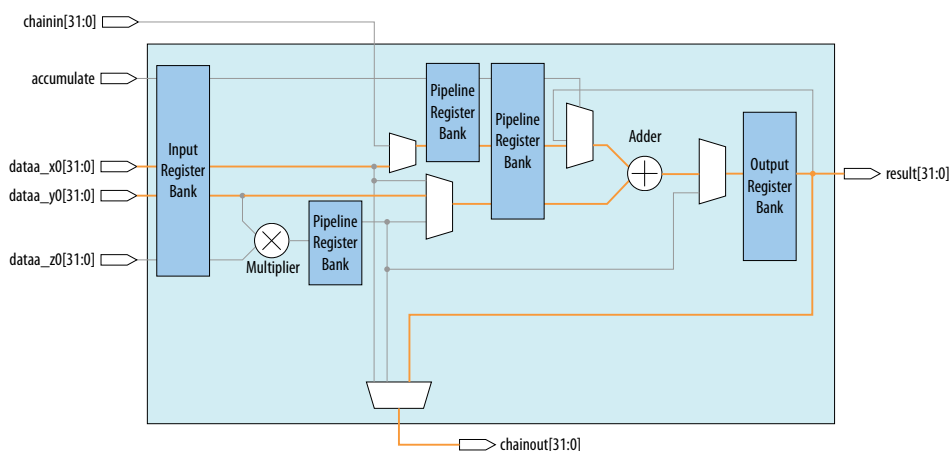
图 41. Arria 10 器件的乘法模式



3.5.2.1.2. 加法或减法模式

此模式使您能够应用基本的浮点加法($x+y$)或者基本的浮点减法($y-x$)。

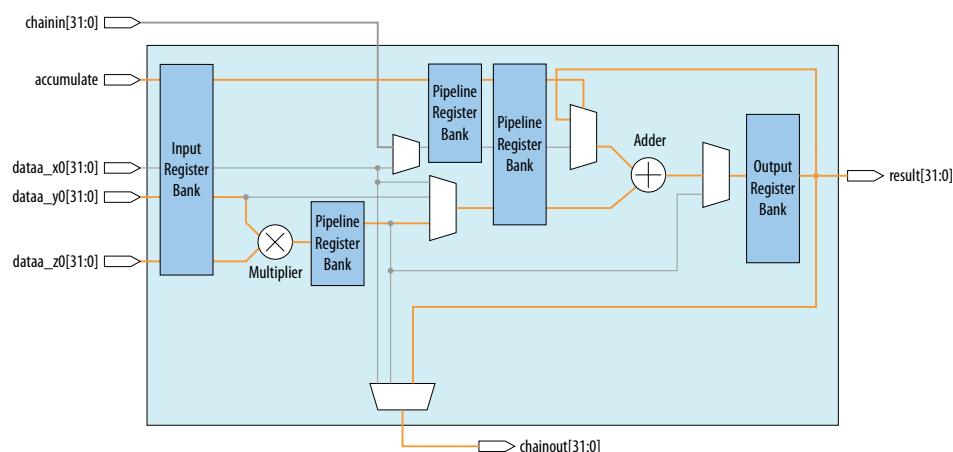
图 42. Arria 10 器件的加法或减法模式



3.5.2.1.3. 乘法累加模式

此模式先执行浮点乘法，再执行与乘积的浮点加法{ $((Y * Z) + ACC)$ 或 $((Y * Z) - ACC)$ }。

图 43. Arria 10 器件的乘法累加模式



3.5.2.2. 多浮点运算功能

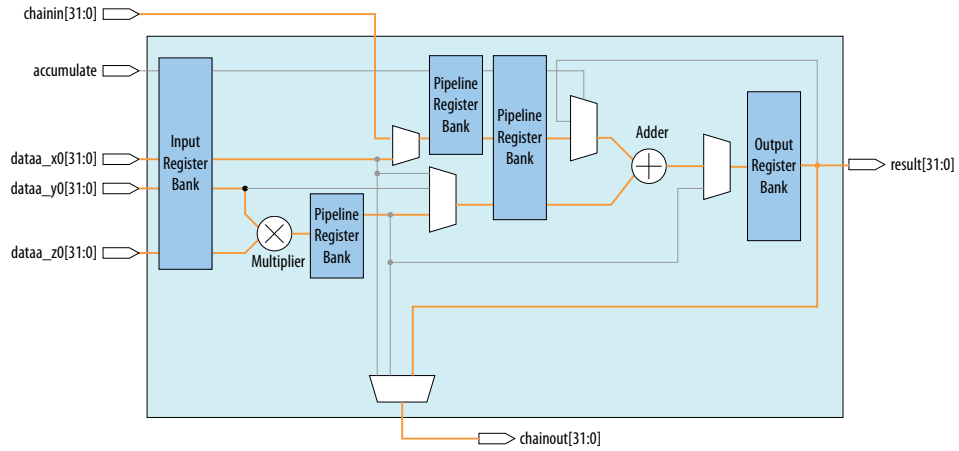
两个或以上浮点运算 DSP 可以执行：

- 乘加或乘减模式，如果 chainin 参数关闭，此模式使用单浮点运算 DSP
- 矢量一模式
- 矢量二模式
- 直接矢量点积
- 复数乘法

3.5.2.2.1. 乘加或乘减模式

此模式先执行浮点乘法，再执行浮点加法或浮点减法{ $((y*z) + x)$ 或 $((y*z) - x)$ }。chainin 参数用于使能多链路(multiple-chain)模式。

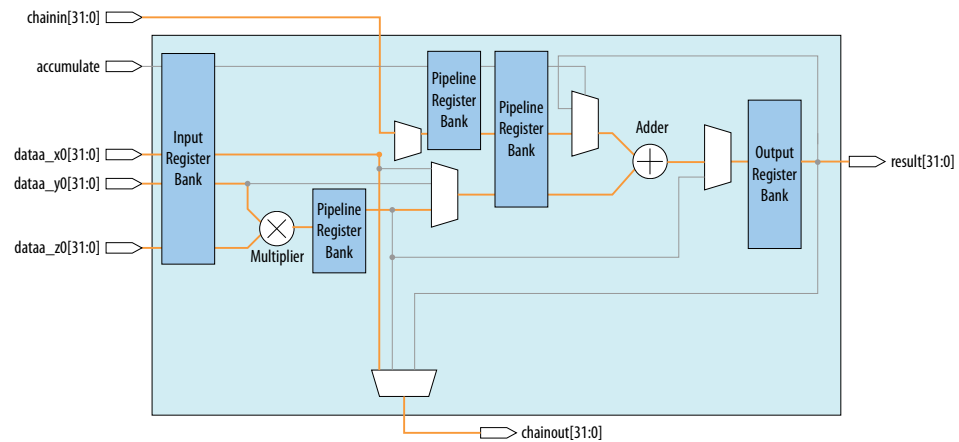
图 44. Arria 10 器件的乘加或乘减模式



3.5.2.2.2. 矢量一模式

此模式先执行浮点乘法，再执行与之前可变 DSP 模块的 chainin 输入的浮点加法。输入 x 直接加入到 chainout。(result = Y*Z + chainin, 其中 chainout = x)

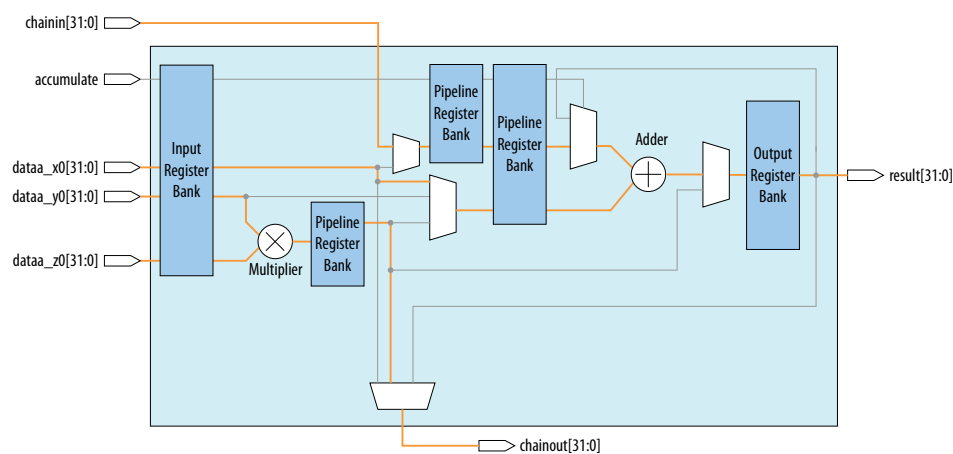
图 45. Arria 10 器件的矢量一模式



3.5.2.2.3. 矢量二模式

此模式执行浮点乘法，乘积被直接加入到 chainout。然后作为输出结果将之前可变 DSP 模块的 chainin 输入与 input x 相加。(result = x + chainin, 其中 chainout = y*z)

图 46. Arria 10 器件的矢量二模式

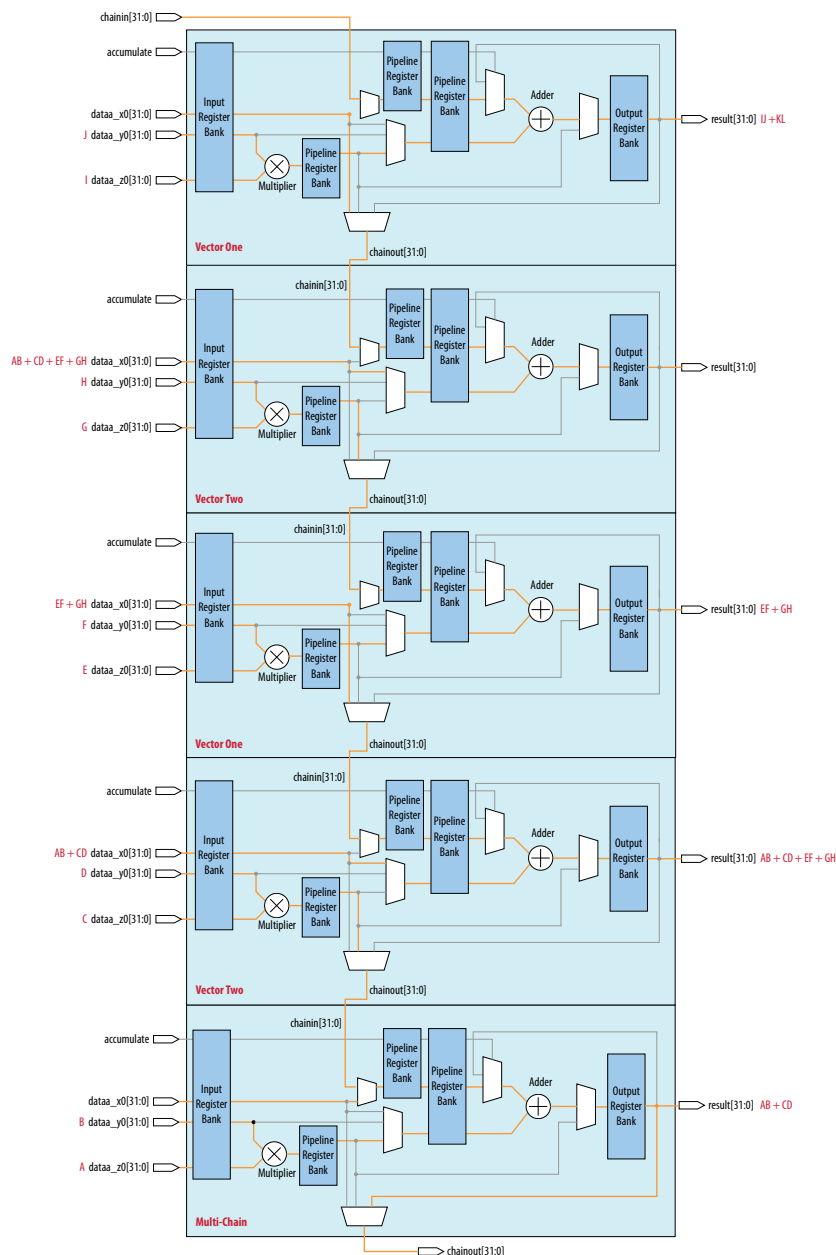


3.5.2.2.4. 直接矢量点积

在下图中，通过设置以下 DSP 模式，直接矢量点积由几个 DSP 模块实现：

- chainin 参数使能的乘加和乘减模式
- 矢量一
- 矢量二

图 47. 直接矢量点积



3.5.2.2.5. 复合乘法

Arria 10 器件使用四个 Arria 10 精度可调 DSP 模块来支持浮点运算单精度复数乘法器。

图 48. 复合乘法公式示例

$$(a + jb) \times (c + jd) = [(a \times c) - (b \times d)] + j[(a \times d) + (b \times c)]$$

虚部 $[(a \times d) + (b \times c)]$ 在头两个精度可调 DSP 模块中实现，而实部 $[(a \times c) - (b \times d)]$ 在第二个精度可调 DSP 模块中实现。

图 49. 实结果的复合乘法

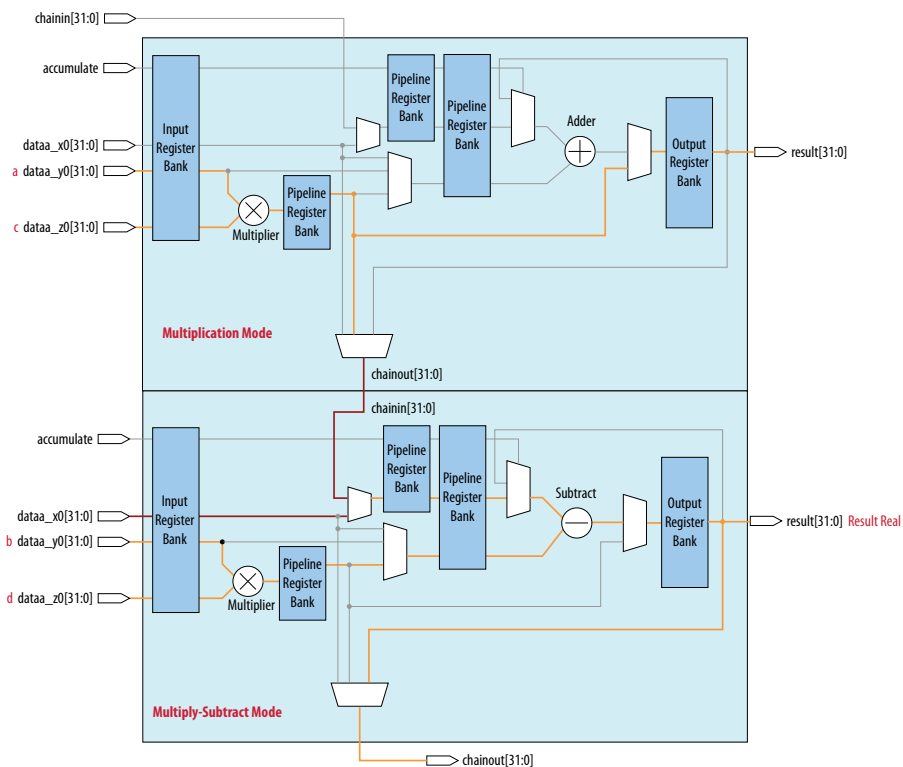
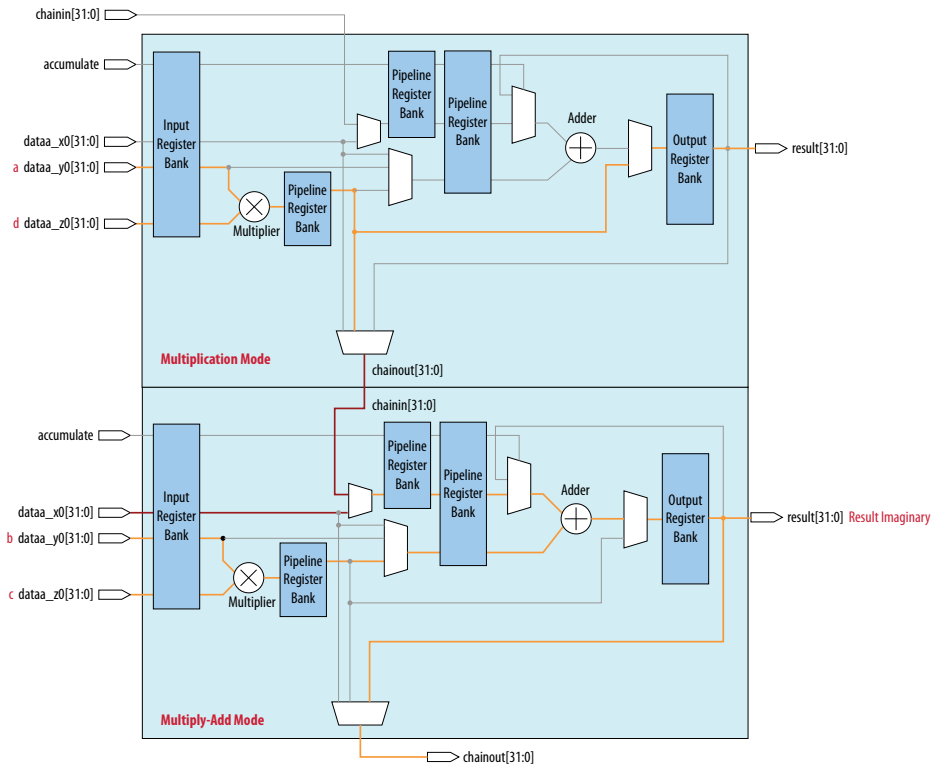
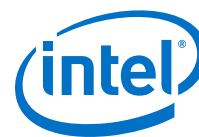


图 50. 虚结果的复合乘法

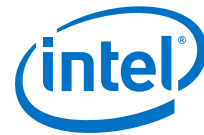


3.6. 文档修订历史

日期	版本	修订内容
2017 年 3 月	2017.03.15	<ul style="list-style-type: none"> 重命名为 Intel。 将减法 $x-y$ 更改成 $y-x$。
2015 年 12 月	2015.11.14	<ul style="list-style-type: none"> 在浮点运算资源的列表中，将 Arria 10 GX 660 的 DSP 模块数量从 1688 更改成 1687。
2015 年 11 月	2015.11.02	<ul style="list-style-type: none"> 更新了 Arria 10 器件中乘法器数量 表中的 Arria 10 GX 320、GX 480、GX 660、SX 320、SX 480 和 SX 660 器件的资源数。 更新了“输入寄存器组”表，指明浮点运算的动态控制信号的输入寄存器组只适用于 Dynamic ACCUMULATE 控制信号。 阐明了 18 x19 脉动 FIR 模式有 7-bits 成本和 37-bits 结果。 更新了 18-bit 和 27-bit 脉动 FIR 模式所支持的级联 DSP 模块的数量。 将 <i>Quartus II</i> 更改成 <i>Quartus Prime</i>。
2015 年 5 月	2015.05.04	<ul style="list-style-type: none"> 在 Arria 10 器件精度可调 DSP 模块的支持组合操作模式和特性表中更新了对所有浮点模式的 Chainin 和 Chainout 支持。 增添了获取独立乘法器模式，乘法加法器求和模式和脉动 FIR 模式的设计模板的步骤。 增添了操作模式下的 Arria 10 Native Floating Point DSP IP 内核表。
继续...		



日期	版本	修订内容
2015 年 1 月	2015.01.23	<ul style="list-style-type: none"> 增添了关于原语 DSP 的信息。 使用列名支持的操作实例更新了 Arria 10 器件精度可调 DSP 模块的支持组合操作模式和特性表。 更新了 Arria 10 器件中乘法器数量表中的单精度浮点加法器的资源。 在定点运算的累加器部分删除了双倍累加寄存器在编程文件中静态地设置这一陈述。 在浮点运算的 Quartus II DSP IP 列表中添加了 ALTERA_FP_FUNCTIONS。 对定点运算中延迟寄存器支持的操作模式增添了阐明信息。 增添了阐明信息：如果正使用这些特性，那么必须使能顶部和底部内部系数以及预加法器。
2014 年 8 月	2014.08.18	<ul style="list-style-type: none"> 增添了浮点运算。 增添了 Dynamic ACCUMULATE, Dynamic LOADCONST, Dynamic SUB, Dynamic NEGATE 到精度可调 DSP 模块可选模式。 在输入级联链上增添了顶部延迟寄存器和底部延迟寄存器。 增添了精度可调 DSP 模块信号，控制精度可调 DSP 模块中的流水线寄存器。 增添了使用同一 DSP 模块中的两个预加器的条件，它们必须共享相同的操作类型（加法或减法）。 更新了 55-bit 加法器。 增添了 38-bit 加法器。 更新了两个 18×19 模式，其中加法器被旁路。 将 Decimation 更新成 Decimation + Accumulate。 新增了 Decimation + Chainout Adder，用于累加器功能和动态控制信号。 新增了每个模块一个乘法器的 27（符号或无符号）× 27（符号或无符号）配置。 在基于一个精度可调 DSP 模块的两个 18 × 19 乘法器的和以及 36-Bit 输入相加的 18 × 19 乘法模式框图中去掉了 chainout 加法器或者累加器。 更新了基本 FIR 滤波器公式。 新增了映射脉动用户视图到精度可调模块体系结构视图。 新增了 27-bit 脉动 FIR 模式不需要脉动寄存器的信息。
2013 年 12 月	2013.12.02	首次发布。



4. Arria 10 器件中的时钟网络和 PLL

本章节介绍了 Arria 10 器件中不同结构的时钟网络与锁相环(PLL)的高级特性。Quartus Prime 软件不需要外部器件来使能 PLL 及其功能。

相关链接

[Arria 10 器件手册：已知问题](#)

列出了对 Arria 10 器件手册章节的计划更新。

4.1. 时钟网络

Arria 10 器件包括具有下面层次结构的时钟网络：

- 全局时钟(GCLK)网络
- 局域时钟(RCLK)网络
- 外围时钟 (PCLK) 网络
 - 小型外围时钟 (SPCLK) 网络
 - 大型外围时钟 (LPCLK) 网络

4.1.1. Arria 10 器件中的时钟资源

表 27. Arria 10 器件中的时钟资源

时钟输入管脚		
器件	可用的资源数量	时钟资源的来源
<ul style="list-style-type: none"> 10AS016 10AS022 10AX016 10AX022 	<ul style="list-style-type: none"> HSSI: 4 个差分对 I/O: 32 个单端或 16 个差分对 	对于高速串行接口(HSSI): REFCLK_GXB[L,R][1:4][C,D,E,F,G,H,I,J]_CH[B,T][p,n]管脚 对于 I/O: CLK_[2,3][A..L]_[0,1][p,n]管脚
<ul style="list-style-type: none"> 10AS027 10AS032 10AX027 10AX032 	<ul style="list-style-type: none"> HSSI: 8 个差分对 I/O: 32 个单端或 16 个差分对 	
<ul style="list-style-type: none"> 10AS048 10AX048 	<ul style="list-style-type: none"> HSSI: 12 个差分对 I/O: 48 个单端或 24 个差分对 	
<ul style="list-style-type: none"> 10AS057 10AS066 10AX057 10AX066 	<ul style="list-style-type: none"> HSSI: 16 个差分对 I/O: 64 个单端或 32 个差分对 	
<ul style="list-style-type: none"> 10AT090 10AT115 10AX090 10AX115 	<ul style="list-style-type: none"> HSSI: 32 个差分对 I/O: 64 个单端或 32 个差分对 	

GCLK 网络		
器件	可用的资源数量	时钟资源的来源
全部	32	<ul style="list-style-type: none"> 每个通道的物理介质附加子层 (PMA) 和物理编码子层 (PCS) TX 和 RX 时钟 每个通道的 PMA 和 PCS TX 和 RX 分频时钟 Hard IP 内核时钟输出信号 DLL 时钟输出 小数分频 PLL (fPLL) 和 I/O PLL C 计数器输出 用于反馈的 I/O PLL M 计数器输出 REFCLK 和时钟输入引脚 内核信号 相对对齐计数器输出

RCLK 网络		
器件	可用的资源数量	时钟资源的来源
<ul style="list-style-type: none"> 10AS016 10AS022 10AS027 10AS032 10AX016 10AX022 10AX027 10AX032 	8	<ul style="list-style-type: none"> 每个通道的物理介质附加子层 (PMA) 和物理编码子层 (PCS) TX 和 RX 时钟 每个通道的 PMA 和 PCS TX 和 RX 分频时钟 Hard IP 内核时钟输出信号 DLL 时钟输出 fPLL 和 I/O PLL C 计数器输出 用于反馈的 I/O PLL M 计数器输出

继续...



RCLK 网络		
器件	可用的资源数量	时钟资源的来源
<ul style="list-style-type: none"> 10AS048 10AX048 	12	<ul style="list-style-type: none"> REFCLK 和时钟输入引脚 内核信号 相位对齐计数器输出
<ul style="list-style-type: none"> 10AS057 10AS066 10AX057 10AX066 10AT090 10AT115 10AX090 10AX115 	16	

SPCLK 网络		
器件	可用的资源数量	时钟资源的来源
<ul style="list-style-type: none"> 10AS016 10AS022 10AX016 10AX022 10AS027 10AS032 10AX027 10AX032 	144	<p>对于 HSSI:</p> <ul style="list-style-type: none"> 每个通道上的物理介质附加子层(PMA)和物理编码子层(PCS) TX 和 RX 时钟 每个通道上的 PMA 和 PCS TX 和 RX 分频时钟 Hard IP 内核时钟输出信号 DLL 时钟输出 fPLL C 计数器输出 REFCLK 和时钟输入引脚 核心信号 <p>对于 I/O:</p> <ul style="list-style-type: none"> DPA 输出(LVDS I/O only) I/O PLL C 和 M 计数器输出 时钟输入引脚 核心信号 相位对齐计数器输出
<ul style="list-style-type: none"> 10AS048 10AX048 	216	
<ul style="list-style-type: none"> 10AS057 10AS066 10AX057 10AX066 	288	
<ul style="list-style-type: none"> 10AT090 10AT115 10AX090 10AX115 	384	

LPCLK 网络		
器件	可用的资源数量	时钟资源的来源
<ul style="list-style-type: none"> 10AS016 10AS022 10AX016 10AX022 10AS027 10AS032 10AX027 10AX032 	24	<p>对于 HSSI:</p> <ul style="list-style-type: none"> 每个通道上的物理介质附加子层(PMA)和物理编码子层(PCS) TX 和 RX 时钟 每个通道上的 PMA 和 PCS TX 和 RX 分频时钟 Hard IP 内核时钟输出信号 DLL 时钟输出 fPLL C 和 M 计数器输出 REFCLK 和时钟输入引脚 内核信号
<ul style="list-style-type: none"> 10AS048 10AX048 	36	

继续...



LPCLK 网络		
器件	可用的资源数量	时钟资源的来源
<ul style="list-style-type: none"> 10AS057 10AS066 10AX057 10AX066 	48	对于 I/O: <ul style="list-style-type: none"> DPA 输出(LVDS I/O only) I/O PLL C 和 M 计数器输出 时钟输入管脚 内核信号 相位对齐计数器输出
<ul style="list-style-type: none"> 10AT090 10AT115 10AX090 10AX115 	64	

关于时钟输入管脚连接的详细信息，请参考管脚连接指南。

相关链接

- 指南：支持 I/O PLL 参考时钟输入管脚的 I/O 标准 (第 170 页)
- Arria 10 器件系列管脚连接指南
- 指南：支持 I/O PLL 参考时钟输入管脚的 I/O 标准 (第 170 页)

4.1.2. 层次化时钟网络

Arria 10 器件包括 3 层时钟网络结构。层次结构的顺序如下：

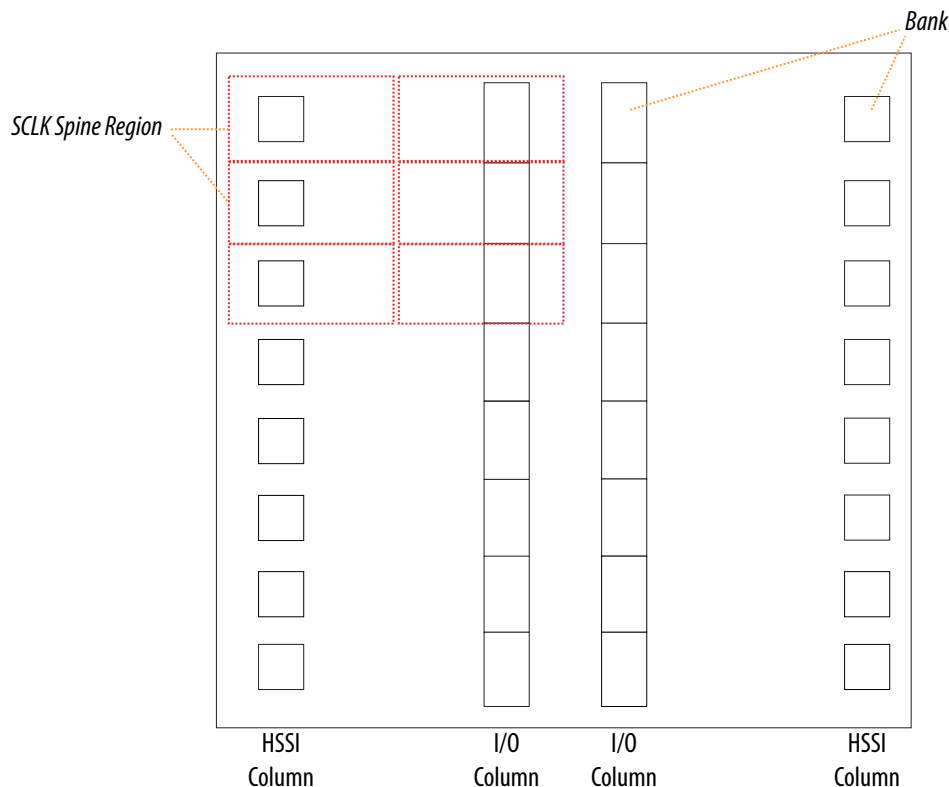
1. GCLK、RCLK、PCLK 和 GCLK 与 RCLK 的反馈时钟
2. 部分时钟 (SCLK)
3. 行时钟

每个 HSSI 和 I/O 列包含时钟驱动器，驱动共享总线到相应的 GCLK，RCLK 和 PCLK 时钟网络。

在每个时钟连接到每个 HSSI 或 I/O bank 的时钟布线之前，Arria 10 时钟网络(GCLK, RCLK 和 PCLK)通过 SCLK 进行布线。Quartus Prime 软件基于 GCLK，RCLK 和 PCLK 网络自动布线 SCLK。

每个 SCLK 脊柱都有相同高度，以匹配 HSSI 和 I/O 组的高度。一个器件中的 SCLK 脊柱数量取决于 HSSI 和 I/O 组的数量。

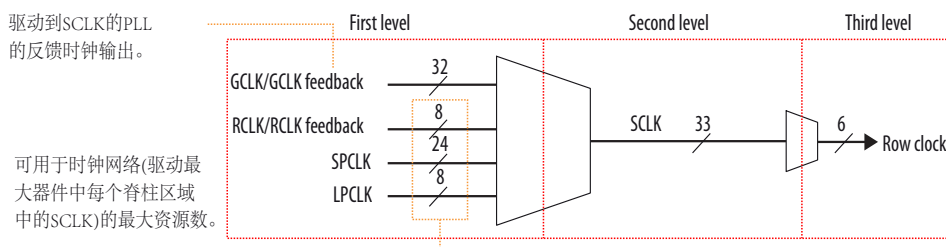
图 51. Arria 10 器件的 SCLK 脊柱区域



Arria 10 器件在每个 SCLK 脊柱区域提供 33 个 SCLK 网络。SCLK 网络能够驱动每行时钟域中的六行时钟。行时钟是内核功能模块，PLL 和器件的 I/O 接口，HSSI 接口的时钟资源。六个唯一的信号能够布线到每个行时钟域。驱动每个 SCLK 的多路复用器的连接模式将时钟源限制在 SCLK 脊柱域。每个 SCLK 能够从 GCLK、RCLK、LPCLK 或 SPCLK 行选择时钟资源。

下图显示了每个 SCLK 脊域中 GCLK、RCLK、PCLK 或者 GCLK 与 RCLK 反馈时钟网络驱动的 SCLK。GCLK、RCLK、PCLK 和 GCLK 与 RCLK 反馈时钟共享相同的 SCLK 布线资源。要确保在 Quartus Prime 中成功布线，时钟资源的总数一定不要超过每个 SCLK 脊域中的 SCLK 数量的限制。

图 52. SCLK 脊柱中层次化时钟网络



4.1.3. 时钟网络类型

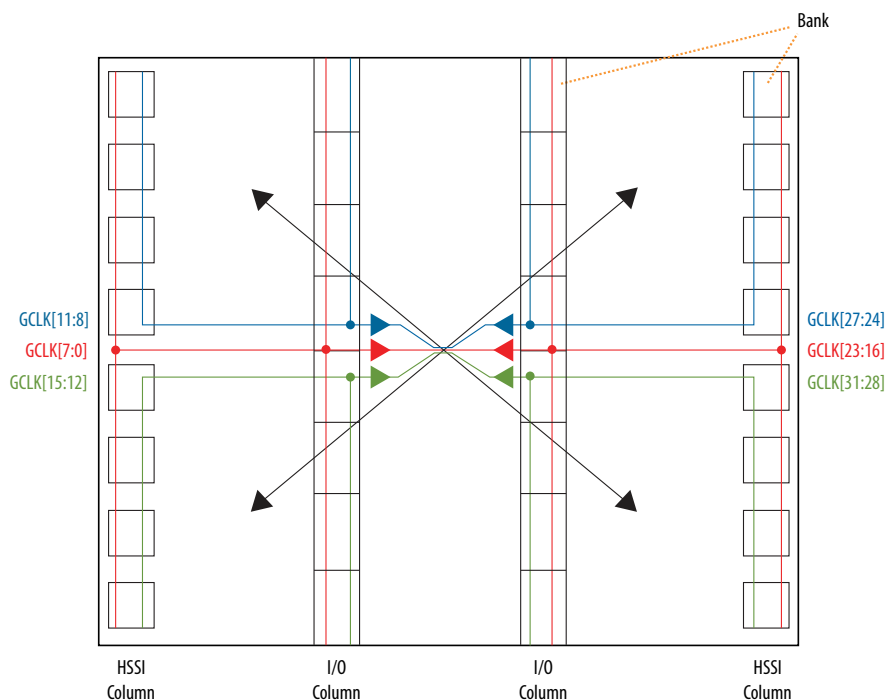
4.1.3.1. 全局时钟网络

GCLK 网络可用作功能模块的低偏斜时钟源，例如：自适应逻辑模块(ALM)、数字信号处理(DSP)、嵌入式存储器以及 PLL。Arria 10 I/O 单元(IOE)和内部逻辑也能够通过使用 GCLK 资源，作为全局时钟或者高扇出控制信号，例如：同步或异步清零与时钟使能信号。

Arria 10 器件提供 GCLK，能够驱动整个器件。GCLK 覆盖器件中的每个 SCLK 脊柱区域。通过脉动 GCLK 网络图中指示的方向对每个 GCLK 进行访问。

图 53. Arria 10 器件中的脉动 GCLK 网络

此图是硅晶片的顶视图，对应于器件封装的反向图。



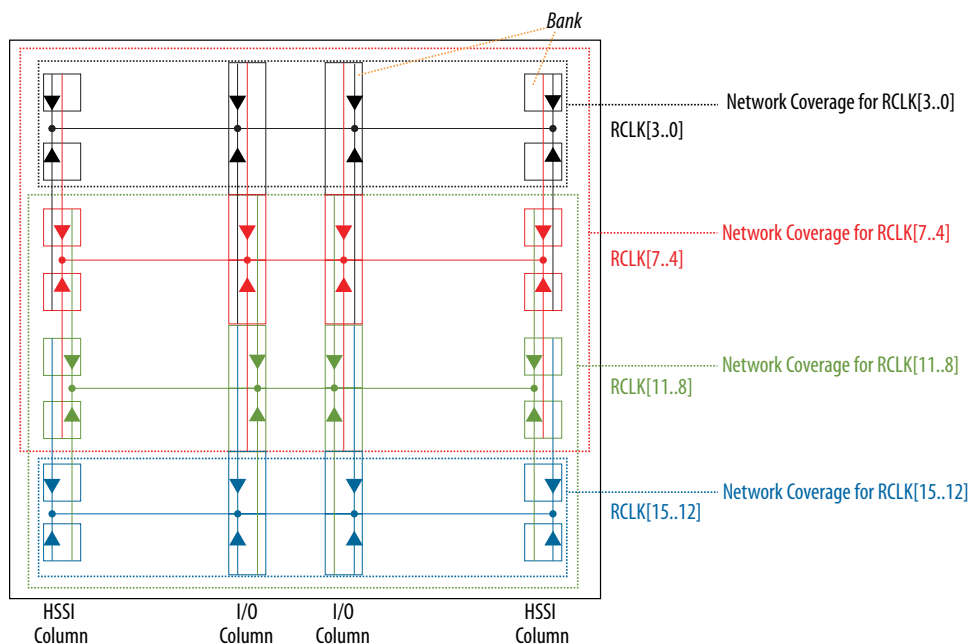
4.1.3.2. 局域时钟网络

RCLK 网络对单一 RCLK 域中包含的逻辑提供低时钟插入延迟和偏斜。Arria 10 IOE 和指定象限中的内部逻辑也能够驱动 RCLK 来创建内部生成的局域时钟和其它高扇出控制信号。

Arria 10 器件提供能够水平驱动芯片的 RCLK。RCLK 覆盖器件同一行中的全部 SCLK 脊域。顶部和底部 HSSI 和 I/O 组具有垂直覆盖 2 行的 RCLK。其它中等 HSSI 和 I/O 组具有垂直覆盖 6 行的 RCLK。下图显示了 RCLK 网络覆盖区域。

图 54. Arria 10 器件中的 RCLK 网络

此图是硅晶片的顶视图，对应于器件封装的反向图。



4.1.3.3. 外设时钟网络

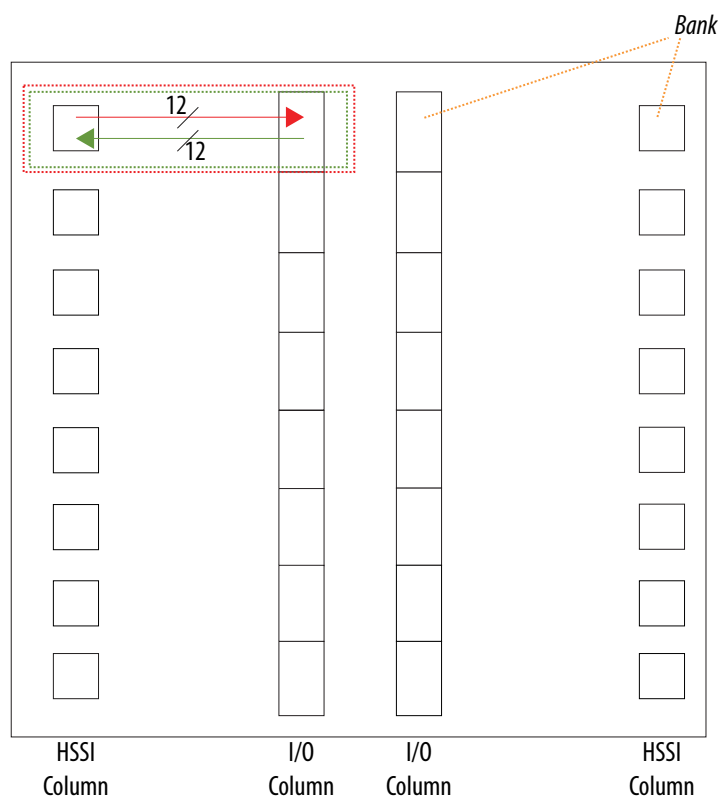
PCLK 网络提供最低的插入延迟以及与 RCLK 网络相同的偏斜。

小型外设时钟网络

每个 HSSI 或者 I/O bank 含有 12 个 SPCLK。SPCLK 在 HSSI bank 中覆盖一个 SCLK 脊域，以及在同一行中彼此相邻的 I/O bank 中覆盖一个 SCLK 脊域。

图 55. Arria 10 器件的 SPCLK 网络

此图是硅晶片的顶视图，对应于器件封装的反向图。

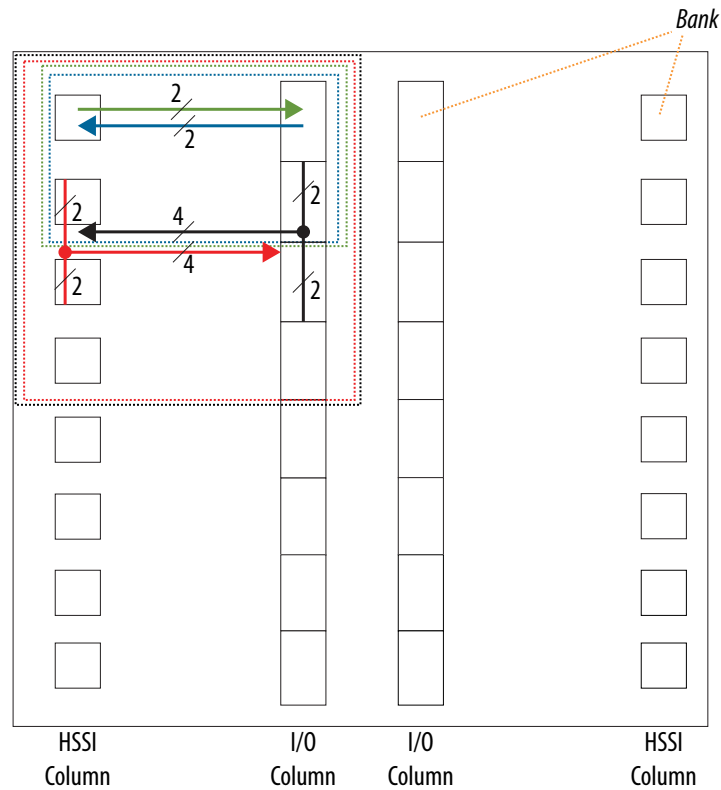


大型外设时钟网络

每个 HSSI 或 I/O bank 含有 2 个 LPCLK。LPCLK 比 SPCLK 具有更大的网络覆盖范围。LPCLK 在 HSSI bank 中覆盖一个 SCLK 脊区，以及在同一行中彼此相邻的 I/O bank 中覆盖一个 SCLK 脊域。顶部和底部 HSSI 以及 I/O bank 含有覆盖垂直两行的 LPCLK。其它中间的 HSSI 和 I/O bank 含有覆盖垂直四行的 LPCLK。

图 56. Arria 10 器件的 LPCLK 网络

此图是硅晶片的顶视图，对应于器件封装的反向图。



4.1.4. 时钟网络源

本节介绍了驱动的 GCLK, RCLK 和 PCLK 网络的时钟网络源。

4.1.4.1. 专用时钟输入管脚

专用时钟输入管脚的来源如下：

- fPLL—
来自 HSSI 列的 $\text{REFCLK_GXB}[L,R][1:4][C,D,E,F,G,H,I,J]_CH[B,T][p,n]$
- I/O PLL—来自 I/O 列的 $\text{CLK}_{[2,3][A..L]}_{[0,1][p,n]}$

您可以将专用时钟输入管脚用于高扇出控制信号，例如：异步清零、预置和时钟使能信号，以及 GCLK 或者 RCLK 网络的协议信号。

专用时钟输入管脚可以是 I/O PLL 的差分时钟或者单端时钟。当专用时钟输入管脚用作单端时钟输入时，只有 $\text{CLK}_{[2,3][A..L]}_{[0,1][p,n]}$ 管脚具有到 PLL 的专用连接。fPLL 仅支持差分时钟输入。

通过全局或局域时钟驱动 PLL 会导致 PLL 输入上的更高抖动，并且 PLL 将无法对全局或局域时钟进行完全补偿。Intel 建议使用专用时钟输入管脚来实现驱动 PLL 的最佳性能。

相关链接

指南：支持 I/O PLL 参考时钟输入管脚的 I/O 标准 (第 170 页)

4.1.4.2. 内部逻辑

您可以使用内核布线驱动每个 GCLK 和 RCLK 网络，以使内部逻辑能够驱动高扇出、低偏斜信号。

4.1.4.3. DPA 输出

每个 DPA 都能够驱动 PCLK 网络。

4.1.4.4. HSSI 时钟输出

HSSI 时钟输出能够驱动 GCLK, RCLK 和 PCLK 网络。

4.1.4.5. PLL 时钟输出

fPLL 和 I/O PLL 时钟输出能够驱动所有时钟网络。

4.1.5. 时钟控制模块

每个 GCLK, RCLK 和 PCLK 网络都有各自的时钟控制模块。时钟控制模块具有以下特性：

- 时钟源选择(动态选择仅用于 GCLK)
- 时钟断电(静态或动态时钟使能或禁用仅用于 GCLK 和 RCLK)

相关链接

时钟控制模块(ALTCLKCTRL) IP 内核用户指南

提供了关于 ALTCLKCTRL IP 内核和时钟乘法方案的详细信息。

4.1.5.1. Arria 10 器件中的管脚映射

表 28. HSSI 列的时钟输入管脚, PLL 计数器输出和时钟控制模块输入之间的映射

时钟	由...提供
inclk[0]	相邻 fPLL 的 PLL 计数器 C0 和 C2。
inclk[1]	相邻 fPLL 的 PLL 计数器 C1 和 C3。
inclk[2]和 inclk[3]	同一 HSSI bank 上的任意两个专用时钟管脚。

表 29. I/O 列的时钟输入管脚, PLL 计数器输出和时钟控制模块输入之间的映射

一个计数器只能分配给一个 inclk。

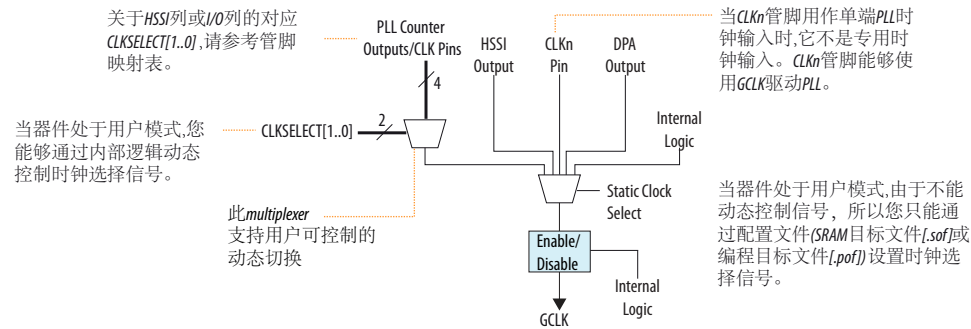
时钟	由...提供
inclk[0]	CLK_[2,3][A..L]_0p 或者相邻 I/O PLL 的任意计数器。
inclk[1]	CLK_[2,3][A..L]_0n 或者相邻 I/O PLL 的任意计数器。
inclk[2]	CLK_[2,3][A..L]_1p 或者相邻 I/O PLL 的任意计数器。
inclk[3]	CLK_[2,3][A..L]_1n 或者相邻 I/O PLL 的任意计数器。

4.1.5.2. GCLK 控制模块

通过静态或动态地使用内部逻辑来驱动多路复用器选择输入，可以为 GCLK 选择模块选择时钟源。

当动态地选择时钟源时，您可以选择 PLL 输出(例如 C0 或 C1)，或者选择一组时钟管脚或者 PLL 输出。

图 57. Arria 10 器件的 GCLK 控制模块



您可以通过 Quartus Prime 软件，使用 ALTCLKCTRL IP 内核对 GCLK 网络多路复用器的输入时钟源和 clkena 信号进行设置。

当使用 ALTCLKCTRL IP 内核动态选择时钟源时，使用 CLKSELECT[0..1] 信号选择输入。

注意: 您只能切换同一 I/O 或 HSSI 组中的专用时钟输入。

相关链接

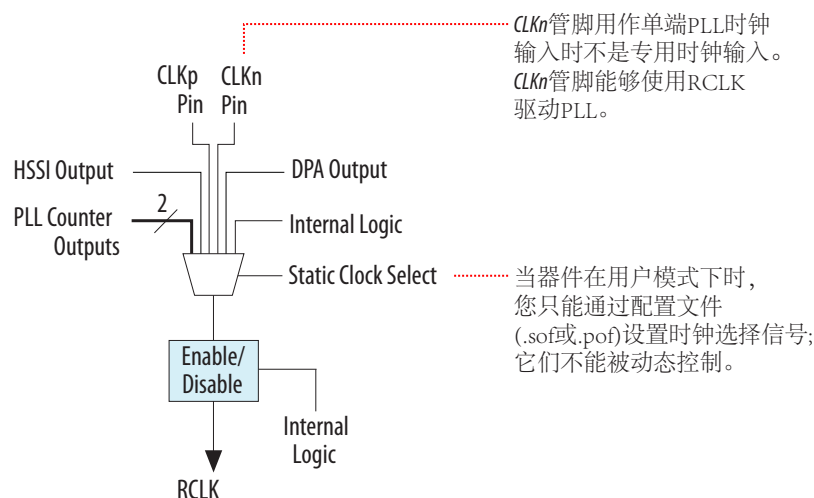
[Arria 10 器件中的管脚映射 \(第 77 页\)](#)

提供了 HSSI 列和 I/O 列中时钟输入管脚，PLL 计数器输出和时钟控制模块输入之间的映射。

4.1.5.3. RCLK 控制模块

您只能通过使用由 Quartus Prime 生成的配置文件(.sof 或 .pof)中的配置位设置来对 RCLK 选择模块的时钟源选择进行控制。

图 58. Arria 10 器件的 RCLK 控制模块



您可以通过 Quartus Prime 软件，使用 ALTCLKCTRL IP 内核来对 RCLK 网络的输入时钟源和 `clkena` 信号进行设置。

4.1.5.4. PCLK 控制模块

PCLK 控制模块驱动 SPCLK 以及 LPCLK 网络。

要驱动 HSSI PCLK，需要选择 HSSI 输出，fPLL 输出或者时钟输入管脚。

要驱动 I/O PCLK，需要选择 DPA 时钟输出，I/O PLL 输出或时钟输入管脚。

图 59. Arria 10 器件的 HSSI 列的 PCLK 控制模块

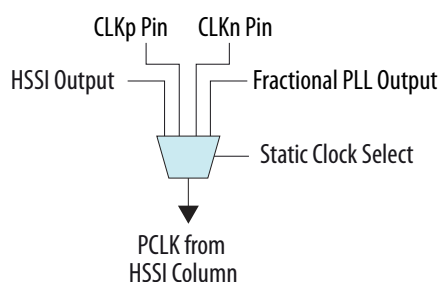
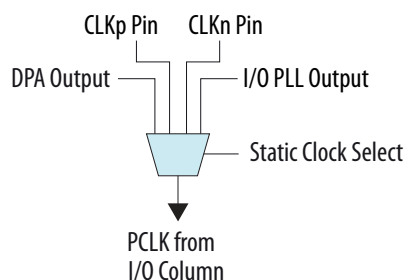


图 60. Arria 10 器件的 I/O 列的 PCLK 控制模块



您可以通过 Quartus Prime 软件，使用 ALTCLKCTRL IP 内核来对 PCLK 网络的输入时钟源和 `clkena` 信号进行设置。

4.1.6. 时钟断电

您可以使用静态和动态方法对 GCLK 和 RCLK 时钟网络进行断电。

当一个时钟网络断电时，由时钟网络提供的所有逻辑均处于关闭状态，从而降低了器件的总功耗。通过 Quartus Prime 生成的配置文件 (`.sof` 或 `.pof`) 中的配置位设置，未使用的 GCLK、RCLK 和 PCLK 网络会自动断电。

动态时钟使能或者禁用功能使内部逻辑能够对 GCLK 和 RCLK 网络进行同步上电或者断电。此功能独立于 PLL，并直接应用于时钟网络。

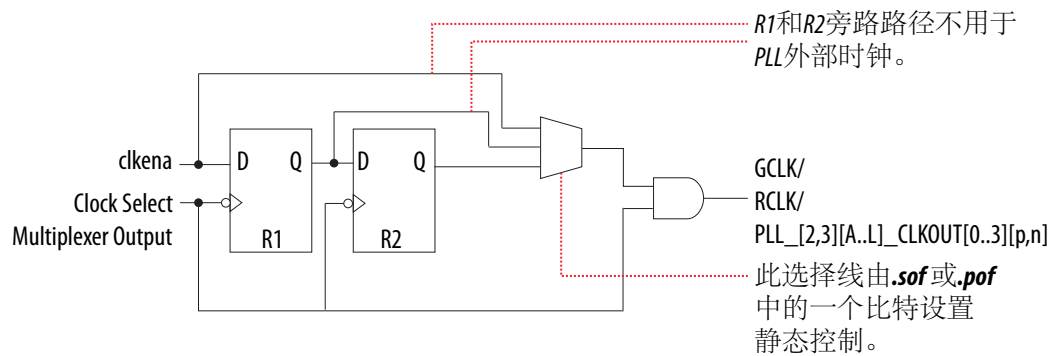
注意: 您不能动态使能或禁用驱动 PLL 的 GCLK 或 RCLK 网络。当内核频率很高时，动态门控大型时钟可能会影响芯片性能。

4.1.7. 时钟使能信号

如果 GCLK 或 RCLK 输出驱动 PLL 的输入，那么不能使用时钟控制模块的时钟使能和禁用电路。

图 61. 使用时钟使能和禁用电路的 `clkena` 实现

此图显示了时钟控制模块的时钟使能和禁用电路的实现。



在时钟网络级支持 `clkena` 信号，而不是在 PLL 输出计数器级。这样即使在没有使用 PLL 的时候也能够关断 (gate off) 时钟。您也可以使用 `clkena` 信号控制 PLL 的专用外部时钟。

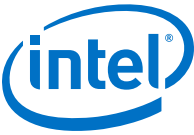
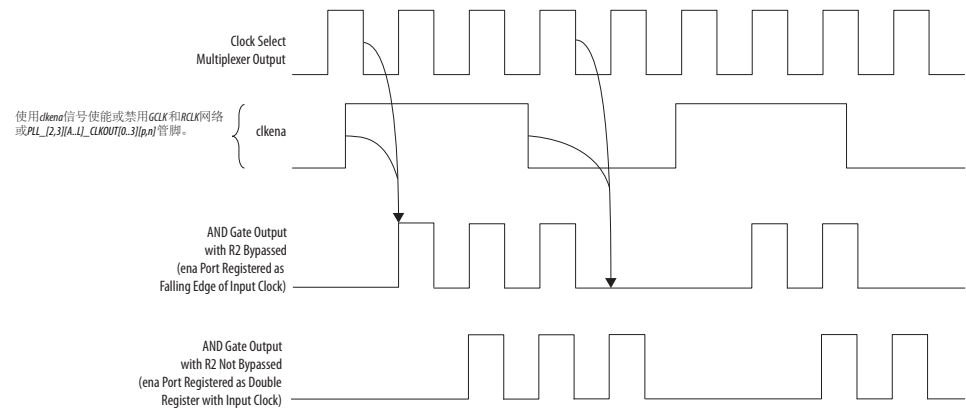


图 62. clkena 信号的实例

此图显示了一个时钟输出使能的波形实例。clkena 信号同步到时钟输出的下降沿。



Arria 10 器件有一个额外的亚稳态寄存器，用于 GCLK 和 RCLK 网络的异步使能或者禁用。您可以在 Quartus Prime 中有选择性地旁路这一寄存器。

由于与回路相关的计数器不会受到影响，因此 PLL 能够独立于 clkena 信号，并保持在锁定状态。这一特性对于要求低功耗或睡眠模式的应用非常有用。如果系统在重新同步过程中不能承受频率过冲，那么 clkena 信号也能够禁用时钟输出。

4.2. Arria 10 PLL

PLL 对器件时钟管理、外部系统时钟管理以及高速 I/O 接口提供了可靠的时钟管理与综合。

Arria 10 器件系列包含以下 PLL：

- fPLL—可用作小数 PLL 或整数 PLL
- I/O PLL—仅用作整数 PLL

fPLL 位于 HSSI Bank 中，与收发器模块相邻。每个 HSSI bank 包含两个 fPLL。您可以在传统的整数模式或小数模式下单独配置每个 fPLL。在小数模式下，fPLL 可以使用三阶 delta-sigma 调制进行操作。每个 fPLL 有四个 C 计数器输出和一个 L 计数器输出。

I/O PLL 位于 I/O bank 中硬核存储控制器和 LVDS 串行器/解串器(SERDES)模块旁边。每个 I/O bank 包含一个 I/O PLL。I/O PLL 能够运行在传统整数模式。每个 I/O PLL 有九个 C 计数器输出。在某些特定封装中，您可以使用那些没有在设计中绑定的 I/O bank 中的 I/O PLL。这些 I/O PLL 必须从 FPGA 内核获得时钟源，或者通过一个专用级联连接从同一 I/O 列中的另一个 I/O PLL 获得参考时钟源。

Arria 10 器件在密集度最大情况下最多有 32 个 fPLL 和 16 个 I/O PLL。Arria 10 PLL 具有不同的内核模拟结构和特性支持。

表 30. Arria 10 器件的 PLL 特性

特性	小数分频 PLL	I/O PLL
整数模式	Yes	Yes
小数分频模式	Yes	—
继续...		

特性	小数分频 PLL	I/O PLL
C 输出计数器	4	9
M 计数器分频因子	8 到 127	4 到 160
N 计数器分频因子	1 到 32	1 到 80
C 计数器分频因子	1 到 512	1 到 512
L 计数器分频因子	1, 2, 4, 8	—
专用外部时钟输出	-	Yes
专用时钟输入管脚	Yes	Yes
外部反馈输入管脚	—	Yes
展频输入时钟跟踪 ⁽⁴⁾	Yes	Yes
源同步补偿	—	Yes
直接补偿	Yes	Yes
正常补偿	—	Yes
零延迟缓存补偿	—	Yes
外部反馈补偿	—	Yes
LVDS 补偿	—	Yes
反馈补偿绑定	Yes	—
压控振荡器(VCO)输出驱动 DPA 时钟	—	Yes
相移分辨率 ⁽⁵⁾	72 ps	78.125 ps
可编程占空比	固定的 50% 占空比	Yes
断电模式	Yes	Yes

4.2.1. PLL 使用

fPLL 被优化用作收发器发送 PLL 和综合参考时钟频率。fPLL 可用作：

- 降低电路板上所需要的振荡器数量
- 可以减少 FPGA 所使用的时钟管脚，因为一个参考时钟源可以产生多种不同频率的时钟输出
- 补偿时钟网络延迟
- 收发器的传输时钟

I/O PLL 被优化与存储器接口和 LVDS SERDES 一起使用。I/O PLL 可用作：

- 降低电路板上所需要的振荡器数量
- 可以减少 FPGA 所使用的时钟管脚，因为一个参考时钟源可以产生多种不同频率的时钟输出
- 简化外部存储器接口和高速 LVDS 接口的设计

⁽⁴⁾ 所提供的输入时钟抖动在输入抖动容限规范之内。

⁽⁵⁾ 最小相移等于 VCO 周期除以 4 (对于 fPLL)或除以 8(对于 I/O PLL)。对于度数递增，Arria 10 器件能够以至少 45°(对于 I/O PLL)或者 90° (对于 fPLL)递增所有输出频率。更小的度数递增有可能受到频率和分频系数的限制。

- I/O PLL 与 I/O 的紧密耦合使时序收敛更容易
- 补偿时钟网络延迟
- 零延迟缓存

4.2.2. PLL 体系结构

图 63. Arria 10 器件的小数分频 PLL 高级结构图

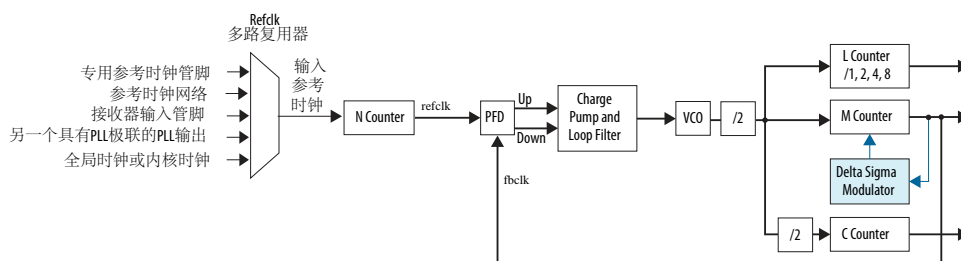
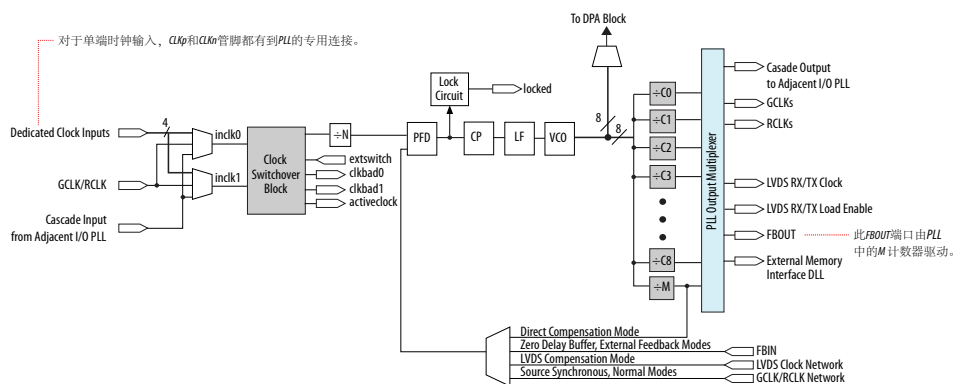


图 64. Arria 10 器件的 I/O PLL 高级结构图



4.2.3. PLL 控制信号

您可以使用复位信号控制 PLL 操作和重新同步，使用锁定信号观测 PLL 的状态。

4.2.3.1. 复位 (reset)

每个 PLL 的 IP 内核的复位信号端口如下：

- fPLL—pll_powerdown
- I/O PLL—reset

复位信号是每个 PLL 的复位或者重新同步输入。器件输入管脚或者内部逻辑能够驱动这些输入信号。

当复位信号被驱高时，PLL 计数器复位，对 PLL 输出清零，使 PLL 处于失锁状态。VCO 然后恢复为默认设置。当复位信号再次被驱低时，PLL 在重新锁定的同时将重新同步到它的输入。

每次 PLL 失锁后必须置位复位信号以保证 PLL 输入与输出时钟之间的正确相位关系。您可以使用 Quartus Prime 参数编辑器将 PLL 设置成失锁 (loss-of-lock) 后自动自复位(self-reset)。

当下面其中一个条件为真时就必须包括复位信号：

- 设计中使能了 PLL 重配置或者时钟切换功能
- 失锁(loss-of-lock)情况过后,必须维持 PLL 输入与输出时钟之间的相位关系

注意:

- 如果上电后 PLL 的输入时钟没有翻转或者不稳定，那么要在输入时钟变得稳定并在规格范围内后位置复位信号。
- 对于 fPLL，器件上电后，当 fPLL 上电校准完成时(pll_cal_busy 信号置低)，您必须复位 fPLL。

4.2.3.2. 锁定 (locked)

每个 PLL 的 IP 内核的锁定信号端口如下：

- fPLL—pll_locked
- I/O PLL—locked

锁定检测电路(lock detection circuit)提供了一个到内核逻辑的信号，表明反馈时钟在相位以及频率上已经锁定到参考时钟。

4.2.4. 时钟反馈模式

时钟反馈模式对时钟网络延迟进行补偿，以对齐 PLL 时钟输入上升沿和时钟输出的上升沿。对您设计中的时序关键时钟路径选择适当类型的补偿。

并非总是需要 PLL 补偿。PLL 应该在直接模式下(无补偿)配置，除非需要补偿。直接模式提供最佳的 PLL 抖动性能，并避免不必要地消耗补偿时钟资源。

默认的时钟反馈模式是直接补偿模式。

fPLL 支持以下时钟反馈模式：

- 直接补偿
- 反馈补偿绑定

I/O PLL 支持以下时钟反馈模式口：

- 直接补偿
- 正常补偿
- 源同步补偿
- LVDS 补偿
- 零延迟缓存(ZDB)补偿
- 外部反馈(EFB)补偿

相关链接

- [Altera I/O 锁相环\(Altera IOPLL\) IP 内核用户指南](#)
提供了关于 I/O PLL 操作模式的更多信息。
- [PLL 反馈和级联时钟网络，Arria 10 收发器 PHY 用户指南](#)
提供了关于 fPLL 操作模式的更多信息。



4.2.5. 时钟倍频与分频

一个 Arria 10 PLL 输出频率通过整数模式的缩放因子 $M/(N \times C)$ 与其输入参考时钟源相关联。输入时钟与预缩放因子 N 相除，然后乘以反馈因子 M 。控制环路驱动 VCO 以匹配 $f_{in} \times (M/N)$ 。

Quartus Prime 软件根据输入到 Altera IOPLL IP core for I/O PLL 和 Arria 10 FPLL IP core for fPLL 的输入频率、倍频和分频值来自动选择相应的缩放因子。

预缩放计数器 N 和乘法计数器 M

每个 PLL 有一个预缩放计数器 N 和一个乘法计数器 M 。由于 M 和 N 计数器仅用于分频，因此它们不使用占空比控制。

后缩放计数器 C

每一个输出端口有一个单独的后缩放计数器 C 。对于具有不同频率的多个 C 计数器输出，VCO 的值被设为输出频率的最小公倍数，以满足其频率规格。例如，如果一个 I/O PLL 所要求的输出频率是 55 MHz 和 100 MHz，那么 Quartus Prime 会将 VCO 频率设置为 1.1 GHz (55 MHz 和 100 MHz 在 VCO 频率范围内的最小公倍数)。然后，后缩放计数器 C 会降低每个输出端口的 VCO 频率。

后缩放计数器， L

fPLL 有一个额外的后缩放计数器 L 。 L 计数器使用 $M/(N \times L)$ 缩放因子来综合其时钟源的频率。 L 计数器生成一个差分时钟对 (0 度和 180 度)，并驱动 HSSI 时钟网络。

Delta-Sigma Modulator (Delta-Sigma 调制器)

delta-sigma modulator (DSM) 与 M 乘法计数器一起用于使 PLL 运行在小数分频模式。DSM 从周期到周期动态地修改 M 计数器因子。不同的 M 计数器因子允许 "average" M 计数器因子是一个非整形。

小数分频模式 (Fractional Mode)

在小数分频模式下， M 计数器值等于 M 反馈因子与小数分频值的和。小数分频值等于 $K/2^{32}$ ，其中 K 是 0 到 $(2^{32} - 1)$ 之间的整数。

整数模式 (Integer Mode)

对于运行在整数模式的 PLL， M 是一个整数值，DSM 是禁用的。

I/O PLL 只能运行在整数模式。

相关链接

- [Altera I/O 锁相环 \(Altera IOPLL\) IP 内核用户指南](#)
提供了关于 Quartus Prime 软件中 I/O PLL 软件支持的详细信息。
- [PLL 和时钟网络章节, Arria 10 收发器 PHY 用户指南](#)
提供了关于 Quartus Prime 软件中 fPLL 软件支持的详细信息。

4.2.6. 可编程相移

可编程相移特性使 fPLL 以及 I/O PLL 都能够生成具有固定相移的输出时钟。

PLL 的 VCO 频率决定了相移的精度。最小相移增量为 1/8 (I/O PLL) 或 1/4 (fPLL) 的 VCO 周期。例如, 如果 I/O PLL 使用 1000 MHz 的 VCO 频率运行, 125 ps 的相移步长是可能的。

Quartus Prime 根据输入到 IP 内核的用户指定相移值自动调整 VCO 频率。

4.2.7. 可编程占空比

可编程占空比功能使 I/O PLL 能够生成具有可变占空比的时钟输出。只有 I/O PLL post-scale 计数器 C 支持此功能。fPLL 不支持可编程占空比功能, 只有固定的 50% 占空比。

I/O PLL C 计数器的值决定了占空比的精度。该精度定义为 50% 除以缩放计数器的值。例如, 如果 C0 计数器是 10, 则 5% 的步长可用于 5% 到 90% 之间的占空比选择。如果 I/O PLL 在外部反馈模式下, 那么需要对驱动 fbin 管脚的计数器设置 50% 的占空比。

Quartus Prime 根据输入到 IP 内核的用户所需的占空比自动调整 VCO 频率。

可编程占空比与可编程相移的结合可以生成精确的非重叠时钟。

4.2.8. PLL 级联

Arria 10 器件支持 PLL 到 PLL (PLL-to-PLL) 级联, 最多可级联 2 个 PLL。级联的 PLL 必须是毗邻的 PLL。与单一 PLL 相比, PLL 级联综合更多的输出时钟频率。

如果设计中存在级联 PLL, 那么源(上游)PLL 必须采用低带宽设置, 而目标(下游)PLL 必须采用高带宽设置。级联期间, 源 PLL 的输出充当目标 PLL 的参考时钟(输入)。具有级联关系的不同 PLL 的带宽设置必须不同。如果具有级联关系的不同 PLL 的带宽设置相等, 那么这些级联 PLL 可能会在某些频率放大相位噪声。

Arria 10 器件仅支持内核应用的 I/O-PLL-to-I/O-PLL 级联。在此模式下, 上游 I/O PLL 和下游 I/O PLL 必须位于同一 I/O 列中。

Arria 10 fPLL 不支持内核应用的 PLL 级联模式。

相关链接

- [Altera I/O 锁相环\(Altera IOPLL\) IP 内核用户指南](#)
提供了关于 Quartus Prime 软件中 I/O PLL 级联的详细信息。
- [实现 PLL 级联, Arria 10 收发器 PHY 用户指南](#)
提供了关于 Quartus Prime 软件中 fPLL 级联的详细信息。

4.2.9. 参考时钟源

有三种可能的 I/O PLL 的参考时钟源。时钟可以来自专用管脚, 内核时钟网络或者专用级联网络。

Intel 建议使用专用管脚提供 I/O PLL 参考时钟(如果可能)。如果您想对 PLL 参考时钟使用一个非专用管脚, 那么必须要在 Quartus Prime 软件中明确地将时钟提升到一个全局信号。

您可以提供两种 I/O PLL 的参考时钟。

- 两种参考时钟都能够来自专用管脚。
- 只有一种参考时钟能够来自内核时钟。
- 只有一种参考时钟能够来自专用级联网络。

4.2.10. 时钟切换

时钟切换功能使 PLL 能够在两个参考输入时钟之间进行切换。此功能用于时钟备份或双时钟域的应用，在此应用中如果前一个时钟停止运行，那么系统就开启备份时钟。当时钟不再翻转或者不再基于用户控制信号 `extswitch` 时，设计能够自动执行时钟切换。

Arria 10 PLL 支持以下时钟切换模式：

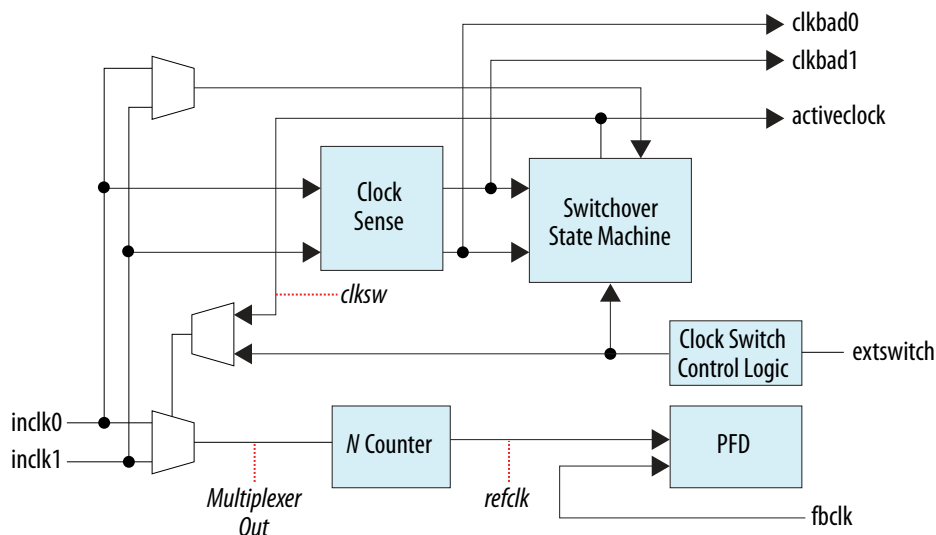
- 自动切换—时钟检测电路监控当前的参考时钟。如果当前参考时钟停止翻转，那么该参考时钟会自动在 `inclk0` 或者 `inclk1` 时钟之间切换。
- 手动时钟切换—使用 `extswitch` 信号控制时钟切换。当 `extswitch` 信号脉冲保持至少三个时钟周期的低电平以切换 `inclk` 时，PLL 的参考时钟会从 `inclk0` 切换到 `inclk1`，反之亦然。
- 通过手动覆盖的自动切换(**automatic switchover with manual override**)—此模式是自动切换和手动时钟切换的综合。当 `extswitch` 信号变低时，它将覆盖自动时钟切换功能。只要 `extswitch` 信号为低，进一步的切换操作就会被阻止。

4.2.10.1. 自动时钟切换

Arria 10 PLL 支持一个完全可配置的时钟切换功能。

图 65. 自动时钟切换电路结构图

此图显示了内置在 PLL 中的自动切换电路的结构图。



当前参考时钟不存在时，时钟检测模块会自动切换到 PLL 参考的备用时钟。在您的设计中，通过连接备用时钟到 PLL 的 `inclk1` 端口可以选择一个时钟源作为备用时钟。

时钟切换电路发出三个状态信号—`clkbad0`、`clkbad1` 和 `activeclock`—来自 PLL 在逻辑阵列中实现定制切换的电路。

在自动切换模式中，`clkbad0` 和 `clkbad1` 信号表明这两个时钟输入的状态。当这两个信号置位时，时钟检测模块检测到相应时钟输入已经停止翻转。如果 `inclk0` 和 `inclk1` 之间的频差大于 20%，那么这两个信号无效。

activeclock 信号表明两个时钟输入(inclk0 或者 inclk1) 的哪一个被选作 PLL 的参考时钟。当这两个时钟输入之间的频差大于 20%, activeclock 信号是唯一有效的状态信号。

当 PLL 的当前参考时钟停止翻转时, 使用切换电路在 inclk0 与 inclk1 之间自动切换。当两个时钟中的一个无效而另一个有效时, 您可以在 inclk0 与 inclk1 之间来回切换任意次数。

例如, 在要求备份时钟(其频率与参考时钟频率相同)的应用中, 切换状态机会生成一个 clksw 信号, 用于控制多路复用器选择。在此情况下, inclk1 变成 PLL 的参考时钟。

当使用自动时钟切换模式时, 必须满足下列要求:

- 当配置 FPGA 时, 这两个时钟输入都必须运行。
- 两个时钟输入的周期差异不能超过 20%。

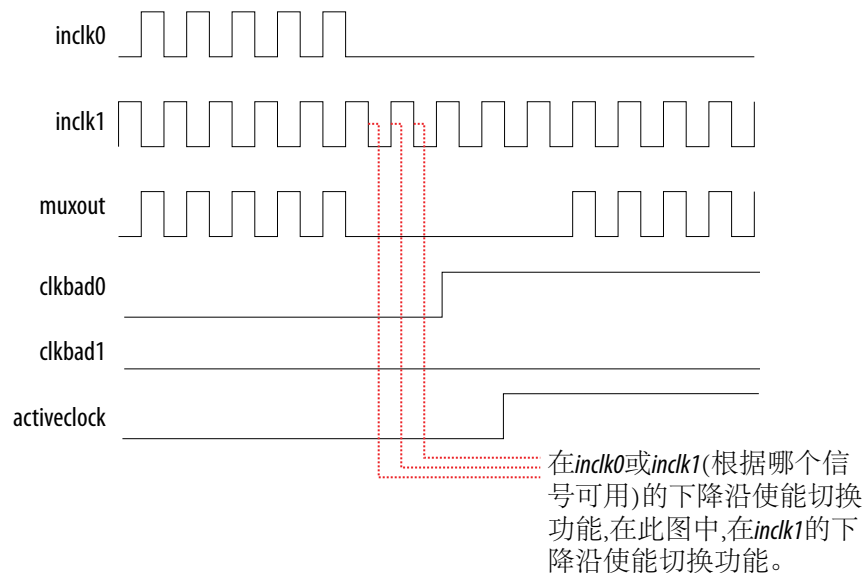
输入时钟必须满足输入抖动规范, 以确保该状态信号的正确操作。输入时钟中的毛刺可能会导致输入时钟之间的频差超过 20%。

如果当前时钟输入停止翻转, 而另一个时钟也没有翻转, 那么将不会启用切换功能, 并且 clkbad[0..1] 信号是无效的。如果两个时钟输入的频率不同, 但它们的周期差在 20% 以内, 那么时钟检测模块会检测到时钟何时停止翻转。然而, PLL 在切换完成后可能会失锁, 从而需要时间重新锁定。

注意: 使用时钟切换功能时, 您必须通过复位信号对 PLL 进行复位, 以保持 PLL 输入与输出时钟之间的相位关系。

图 66. 失锁检测后的自动切换

此图显示了自动切换模式的切换功能实例波形。在此实例中, inclk0 信号保持在低电平。在 inclk0 信号保持在低电平大概两个时钟周期后, 时钟检测电路驱高 clkbad0 信号。由于参考时钟信号(inclk0)没有翻转, 因此切换状态机通过 extswitch 信号控制多路复用器, 以切换到备用时钟 inclk1。



4.2.10.2. 手动覆盖的自动切换(Automatic Switchover with Manual Override)

在手动覆盖的自动切换模式中，`extswitch` 信号可用于用户或系统控制的切换情况。此模式可用于相同频率切换或者不同频率的输入之间的切换。

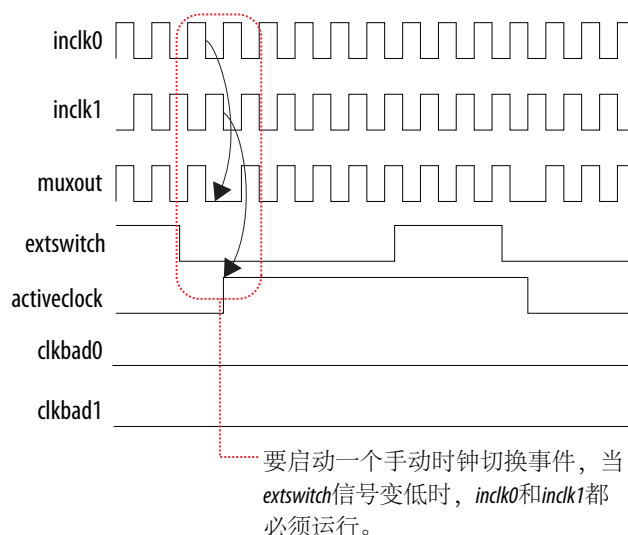
例如，如果 `inclk0` 是 66 MHz，`inclk1` 是 200 MHz，那么必须使用 `extswitch` 信号控制切换。自动时钟检测电路不能监控频差大于 100% (2×) 的时钟输入(`inclk0` 和 `inclk1`) 频率。

当时钟来自背板上的多个板卡，并且需要在操作频率之间系统的控制切换时，该功能是非常有用的。

您必须选择备用时钟频率和设置 M、N、C、L 和 K 计数器，以便 VCO 运行在建议的操作频率范围内。如果指定组合的 `inclk0` 和 `inclk1` 频率不能满足这一要求，那么 Altera IOPLL (I/O PLL) 和 Arria 10 FPLL (fPLL) 参数编辑器会通知您。

图 67. 使用 `extswitch` (手动) 控制的时钟切换

此图显示一个由 `extswitch` 信号控制的时钟切换波形。在此情况中，两个时钟源都是可用的，并且 `inclk0` 被选作参考时钟。当 `extswitch` 信号变低时开始切换流程。在 `inclk0` 的下降沿，计数器的参考时钟 `muxout` 关断(gated off)以防止产生时钟毛刺。在 `inclk1` 的下降沿，作为 PLL 参考，参考时钟多路复用器从 `inclk0` 切换到 `inclk1`。 `activeclock` 信号的变化指示当前哪个时钟驱动 PLL。



在手动切换的自动覆盖模式中，`extswitch` 信号从逻辑高跳变到逻辑低后，`activeclock` 信号反转。由于两个时钟在手动切换期间仍然可用，因此 `clkbad` 信号不会变高。由于切换电路是上升沿敏感的，因此 `extswitch` 信号的上升沿不会导致电路从 `inclk1` 切换回 `inclk0`。当 `extswitch` 信号再次变低，重复整个过程。

只有当目的时钟可用时，`extswitch` 信号和自动切换才能工作。如果目的时钟不可用，那么状态机会一直等待，直到该时钟可用。

相关链接

- [Altera I/O 锁相环\(Altera IOPLL\) IP 内核用户指南](#)
提供了关于 Quartus Prime 软件中 I/O PLL 软件支持的详细信息。
- [PLL 和时钟网络章节，Arria 10 收发器 PHY 用户指南](#)
提供了关于 Quartus Prime 软件中 fPLL 软件支持的详细信息。

4.2.10.3. 手动时钟切换

在手动时钟切换模式中，`extswitch` 信号控制选择 `inclk0` 还是 `inclk1` 作为 PLL 的输入时钟。默认情况下选择 `inclk0`。

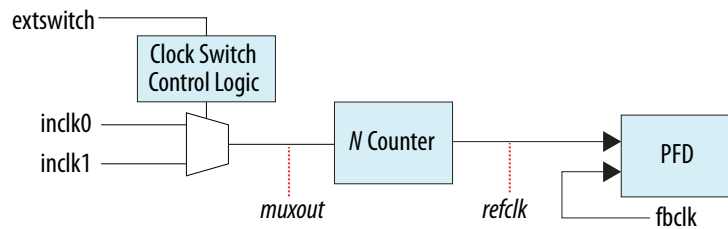
当 `extswitch` 信号从逻辑高电平跳变到逻辑低电平，并保持至少三个 `inclk` 周期的高电平以切换到 `inclk` 时，一个时钟切换事件被启动。

您必须将 `extswitch` 信号再拉回到高电平才能执行另一个时钟切换事件。如果不需要另一个时钟切换事件，那么在初始切换后保持 `extswitch` 处于逻辑低电平的状态。

拉低 `extswitch` 信号至少三个 `inclk` 周期，以切换 `inclk` 来执行另一个切换事件。

如果 `inclk0` 和 `inclk1` 的频率不同并且一直运行，那么 `extswitch` 信号最短的高电平时间一定要大于或等于 `inclk0` 和 `inclk1` 中较低频率的三个时钟周期。

图 68. Arria 10PLL 中的手动时钟切换电路



通过在 Altera IOPLL (for I/O PLL) 和 Arria 10 FPLL (for fPLL) IP 内核中指定切换延迟，您可以延迟时钟切换操作。指定切换延迟时，`extswitch` 信号必须保持高电平至少三个 `inclk` 周期以切换 `inclk` 并加上已经指定的延迟周期数以启动时钟切换。

相关链接

- [Altera I/O 锁相环\(Altera IOPLL\) IP 内核用户指南](#)
提供了关于 Quartus Prime 软件中 I/O PLL 软件支持的详细信息。
- [PLL 和时钟网络章节, Arria 10 收发器 PHY 用户指南](#)
提供了关于 Quartus Prime 软件中 FPLL 软件支持的详细信息。

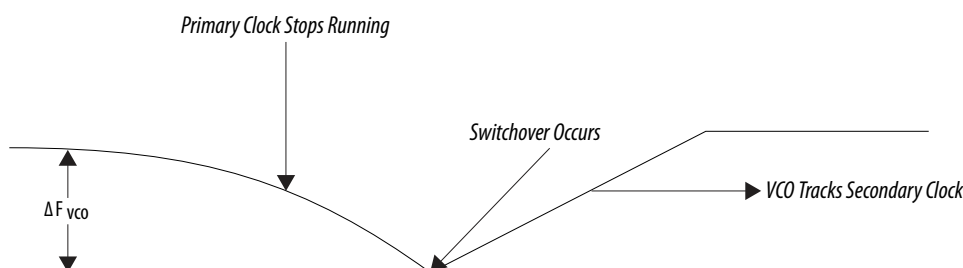
4.2.10.4. 指南

当在 Arria 10PLL 中实现时钟切换时，需要遵循下面的指导原则：

- 自动时钟切换要求 `inclk0` 和 `inclk1` 频率偏差保持在 20% 以内，否则将导致 `clkbad0` 和 `clkbad1` 信号无法正常运行。
- 使用手动时钟切换时，`inclk0` 和 `inclk1` 之间的差异能够大于 100% (2×)。然而，两个时钟源的频差和相差都将可能导致 PLL 失锁。复位 PLL 确保了在输入和输出时钟之间保持正确的相位关系。
- 当 `extswitch` 信号变低以启动手动时钟切换事件时，`inclk0` 和 `inclk1` 都必须运行，否则会导致时钟切换操作无法正常进行。
- 要求时钟切换功能以及低频率漂移的应用必须使用低带宽 PLL。当参考输入时钟变化时，低带宽 PLL 要比高带宽 PLL 反应慢。发生切换时，与高带宽 PLL 相比，低带宽 PLL 更慢地传播时钟停止到输出。然而，要知道低带宽 PLL 也会增加锁定时间。

- 切换发生后，PLL 在一个有限的重同步周期锁定到一个新的时钟。PLL 重新锁定所需要的时间取决于 PLL 配置。
- 在您的设计中，PLL 的输入时钟与输出时钟之间的相位关系是非常重要的。时钟切换完成后要置位复位信号至少 10 ns。等待锁定的信号变高并且稳定后，再重新使能 PLL 的输出时钟。
- 显示了当前时钟丢失时，VCO 频率逐渐下降，然后在 VCO 锁定到备用时钟时又回升的情况。

图 69. VCO 切换操作频率



4.2.11. PLL 重配置和动态相移

fPLL 和 I/O PLL 支持具有以下功能的 PLL 重配置和动态相移：

- PLL 重配置—重配置 M、N 和 C 计数器。能够重配置小数分频设置(fPLL)。
- 动态相移—执行正或负相移。fPLL 仅支持一个动态相移操作中一个相位步进，其中每个相位步进等于 1/4 VCO 周期。I/O PLL 支持一个动态相移操作中多个相位步进，其中每个相位步进等于 1/8 VCO 周期。

相关链接

- [AN 728: Arria 10 器件的 I/O PLL 重配置和动态相移](#)
提供了关于在 PLL Reconfig IP 内核中实现 I/O PLL 重配置和在 IOPLL IP 内核中实现 I/O PLL 动态相移的详细信息。
- [使用 PLL 和时钟网络, Arria 10 收发器 PHY 用户指南](#)
提供了关于在 Quartus Prime 软件中实现 fPLL 重配置的详细信息。

4.3. 文档修订历史

日期	版本	修订内容
2017 年 5 月	2017.05.08	<ul style="list-style-type: none"> • 更新了关于 PLL 级联的信息。 • 删除了所有的“初步”标识。
2017 年 3 月	2017.03.15	重命名为 Intel。
2016 年 10 月	2016.10.31	<ul style="list-style-type: none"> • 将时钟切换控制信号从 clkswitch 修改成 extswitch。 • 将“手动时钟切换”部分中的时钟切换控制信号更新成了低电平有效(active low)。
2016 年 5 月	2016.05.02	<ul style="list-style-type: none"> • 更新了“Arria 10 器件中的时钟资源”表。 <ul style="list-style-type: none"> — 更新了用于 HSSI 的资源数。 — 删除了 fPLL M 计数器输出。 • 更新了专用时钟输入管脚的说明。 • 更新了“时钟断电”部分中的注释。 • 更新了“Arria 10 PLL”部分中 fPLL 的说明。

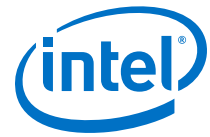
继续...



日期	版本	修订内容
		<ul style="list-style-type: none">更新了“Arria 10 的小数分频 PLL 的高级结构图”。删除了“Arria 10 器件的 I/O PLL 高级结构图”中的专用 refclk 输入。更新了 Arria 10 器件所支持的 PLL 级联模式。增添了“参考时钟源”部分。
2015 年 11 月	2015.11.02	<ul style="list-style-type: none">更新了层次化时钟网络部分中的描述：Arria 10 器件在 SCLK 脊柱域提供高达 33 个 SCLK 网络。更新了 Arria 10 器件的 GCLK 控制模块图。在 GCLK 控制模块部分删除了以下描述：时钟管脚的输入驱动复用器的 inc1k[0..1]端口，PLL 输出驱动 inc1k[2..3]端口。在 Arria 10 PLL 部分添加了关于 I/O PLL 的说明。更新了 Arria 10 器件表中的 PLL 特性。<ul style="list-style-type: none">将特性“整数和小数 PLL”更新成“整数和小数模式”。将 fPLL 的 M 计数器分频因子从“1 to 320”更新成“8 to 127”。将 I/O PLL 的 M 计数器分频因子从“1 to 512”更新成“4 to 160”。将 fPLL 的 N 计数器分频因子从“1 to 512”更新成“1 to 80”。将 fPLL 的 C 计数器分频因子从“1 to 320”更新成“1 to 512”。删除了 fPLL 中的正常补偿支持。将“小数 PLL 绑定补偿”变更成“反馈补偿绑定”。将 fPLL 的相移分辨率从 41.667 ps 更新成 72 ps。更新了 Arria 10 器件的小数分频 PLL 高级结构图中的补偿模式。更新了 fPLL 的时钟反馈模式。<ul style="list-style-type: none">删除了正常补偿。将 fPLL 绑定补偿更改成反馈补偿绑定。更新了“PLL 重配置和动态相移”部分中关于动态相移的描述。将 Quartus II 更改成 Quartus Prime。
2015 年 5 月	2015.05.04	<ul style="list-style-type: none">将 SCLK Spine 中层次化时钟网络图中的 RCLK/RCLK 反馈数量从 12 更新成 8。添加了对全局时钟网络部分的描述：通过 Symbolic GCLK Networks 图中指示的方向访问每个 GCLK。在时钟网络源部分中将 HSSI 输出更新为 HSSI 时钟输出。在 PLL 时钟输出部分指定 fPLL 和 I/O PLL 时钟输出能够驱动所有时钟网络。增添了对 PLL 级联带宽要求和 PLL 级联模式的描述。在 PLL 控制信号(复位)部分添加了 fPLL 复位要求的一个注释。
2015 年 1 月	2015.01.23	<ul style="list-style-type: none">更新了专用时钟输入管脚用作单端时钟输入时，具有到 I/O PLL (CLK_[2,3][A..L]_[0,1][p,n]) 的专用连接。删除了专用时钟输入管脚 CLK_[2,3][A..L]_[0,1]n，这些管脚通过全局或局域时钟网络驱动 I/O PLL，没有到 I/O PLL 的专用布线路径。删除了时钟网络源部分中内部逻辑的一个注释。删除的注释：内部生成的 GCLK 或 RCLK 不能驱动 Arria 10 PLL。PLL 的输入时钟必须来自专用时钟输入管脚，PLL 驱动的 GCLK 或者 PLL 驱动的 RCLK。增添了 HSSI 和 I/O 列的时钟控制模块管脚映射表。更新了 Arria 10 器件的小数分频 PLL 高级结构图。将专用时钟输入注释中的 CLKp 更改成 REFCLK_GXBp，CLKn 更改成 REFCLK_GXBn。更新了 Arria 10 器件的 I/O PLL High-Level Block Diagram 中的专用时钟输入的注释，因为所有四个时钟输入都能够用作 I/O PLL 的专用时钟输入。注释从“对于单端时钟输入，只有 CLKp 管脚有一个到 PLL 的专用连接。如果您使用 CLKn 管脚，那么就要使用一个全局或局域时钟。”更改成“对于单端时钟输入，CLKp 和 CLKn 管脚都有到 PLL 的专用连接。”增添了 PLL 级联信息。阐明当复位信号再次驱低时，PLL 重新锁定的同时会重新同步到它的输入时钟源。
继续...		



日期	版本	修订内容
		<ul style="list-style-type: none"> • 添加了时钟反馈模式的描述：时钟反馈模式对时钟网络延迟进行补偿，以对齐 PLL 时钟输入上升沿和时钟输出的上升沿。对您设计中的时序关键时钟路径选择适当类型的补偿。并非总是需要 PLL 补偿。PLL 应该在直接模式下(无补偿)配置，除非需要补偿。直接模式提供最佳的 PLL 抖动性能，并避免不必要地消耗补偿时钟资源。 • 将时钟切换 clkswitch 信号从正触发器更新成负触发器。 • 对下面文档添加了链接： <ul style="list-style-type: none"> — Altera I/O 锁相环(Altera IOPLL) IP 内核用户指南—提供关于 Quartus Prime 软件中 I/O PLL 软件支持的更多信息。 — Arria 10 收发器 PHY 用户指南中的 PLL 和时钟网络章节—提供关于 Quartus Prime 中 fPLL 软件支持的更多信息。 — Arria 10 器件的 I/O PLL 重配置和动态相移—提供关于在 Altera PLL Reconfig IP 内核中实现 I/O PLL 重配置和在 Altera IOPLL IP 内核中实现 I/O PLL 动态相移的更多信息。
2014 年 8 月	2014.08.18	<ul style="list-style-type: none"> • 更新了 HSSI 组中的专用时钟输入管脚名称。 • 更新了层次化时钟网络部分的描述。 • 更新了专用时钟输入管脚部分的描述。 • 移除了内部逻辑部分中的 PCLK 网络。 • 更新了“PCLK 控制模块”部分中的描述。 • 更新了下图： <ul style="list-style-type: none"> — Arria 10 器件 HSSI 列的 PCLK 控制模块 — Arria 10 器件 I/O 行的 PCLK 控制模块 • 移除了 IQTXRXCLK 补偿模式。 • 更新了小数分频 PLL 和 I/O PLL 高级结构图。 • 更新了手动时钟切换的描述。 • 更新了 PLL 重配置的描述。
2013 年 12 月	2013.12.02	首次发布。



5. Arria 10 器件的 I/O 和高速 I/O

Arria 10 I/O 支持以下功能:

- 单端、非参考电压和参考电压 I/O 标准
- 低电压差分信号(LVDS)、RSDS、mini-LVDS、HSTL、HSUL、SSTL 和 POD I/O 标准
- 串行器/解串器(SERDES)
- 可编程输出电流强度
- 可编程摆率
- 可编程总线保持
- 可编程弱上拉电阻
- DDR4 和 LVDS 标准的可编程预加重
- 可编程 I/O 延迟
- 可编程差分输出电压(V_{OD})
- 开漏输出
- 带和不带校准的片上串行匹配(R_S OCT)
- 片上并行匹配(R_T OCT)
- 片上差分匹配(R_D OCT)
- 动态断电的 HSTL 和 SSTL 输入缓冲
- 所有 I/O bank 的动态片上并行匹配
- DDR4 校准内部生成的 V_{REF}

注意: 本章节中的信息适用于所有 Arria 10 系列, 除非另有说明。

相关链接

[Arria 10 器件手册: 已知问题](#)

列出了对 *Arria 10 器件手册* 章节规划的更新。



5.1. Arria 10 器件中的 I/O 和差分 I/O 缓冲

通用 I/O (GPIO) 由 LVDS I/O 和 3 V I/O bank 组成:

- **LVDS I/O bank** — 支持高达 1.8 V 的差分 and 单端 I/O 标准。LVDS I/O 管脚形成真差分 LVDS 通道的管脚对。每对支持两个管脚之间的并行输入/输出匹配。可将每个 LVDS 通道用作 transmitter only 或 receiver only。每个 LVDS 通道支持带 DPA 电路的发送 SERDES 和接收 SERDES。例如: 如果将 72 个通道中的 30 个通道用作发送器, 那么可将剩下的 42 个通道用作接收器。
- **3 V I/O bank** — 仅支持高达 3 V 的单端 I/O 标准。每个相邻的 I/O 对也支持差分 SSTL 和差分 HSTL I/O 标准。3 V I/O 的单端输出支持所有可编程的 I/O 单元(IOE)功能, 除了:
 - 可编程预加重
 - R_D 片上匹配(OCT)
 - 校准 R_S 和 R_T OCT
 - 内部 V_{REF} 生成

Arria 10 器件支持所有 LVDS I/O bank 中的 LVDS:

- 所有的 LVDS I/O bank 支持 R_D OCT 的真 LVDS 输入和真 LVDS 输出缓冲器。
- 该器件不支持仿 LVDS 通道。
- 该器件支持驱动 SERDES 的 I/O PLL 的单端 I/O 参考时钟。

相关链接

- [Arria 10 GX 封装的 FPGA I/O 资源](#) (第 105 页)
列出了 Arria 10 GX 封装中可用的 3 V 和 LVDS I/O 缓冲的数量。
- [Arria 10 GT 封装的 FPGA I/O 资源](#) (第 106 页)
列出了 Arria 10 GT 封装中可用的 3 V 和 LVDS I/O 缓冲的数量。
- [Arria 10 SX 封装的 FPGA I/O 资源](#) (第 107 页)
列出了 Arria 10 SX 封装中可用的 3 V 和 LVDS I/O 缓冲的数量。

5.2. Arria 10 器件中的 I/O 标准和电平

Arria 10 器件系列包含 FPGA 和 SoC 器件。除了 FPGA I/O 缓冲器，Arria 10 SoC 器件也含有支持差分 I/O 标准的 HPS I/O 缓冲器。

5.2.1. Arria 10 器件的 FPGA I/O 所支持的 I/O 标准

表 31. Arria 10 器件的 FPGA I/O 中所支持的 I/O 标准

I/O 标准	器件类型支持	I/O 缓冲类型支持		应用	支持标准
		LVDS I/O	3V I/O		
3.0 V LVTTTL/3.0 V LVCMOS	仅支持 3 V I/O bank 的器件。请参考相关信号。	No	Yes	通用	JESD8-B
2.5 V LVCMOS	仅支持 3 V I/O bank 的器件。请参考相关信息。	No	Yes	通用	JESD8-5
1.8 V LVCMOS	全部	Yes	Yes	通用	JESD8-7
1.5 V LVCMOS	全部	Yes	Yes	通用	JESD8-11
1.2 V LVCMOS	全部	Yes	Yes	通用	JESD8-12
SSTL-18 Class I 和 Class II	全部	Yes	Yes	DDR2	JESD8-15
SSTL-15 Class I 和 Class II	全部	Yes	Yes	DDR3	—
SSTL-15	全部	Yes	Yes	DDR3	JESD79-3D
SSTL-135 Class I 和 Class II	全部	Yes	Yes	DDR3L	—
SSTL-125 Class I 和 Class II	全部	Yes	Yes	DDR3U	—
SSTL-12 Class I 和 Class II	全部	Yes	No	RLDRAM 3	—
POD12	全部	Yes	No	DDR4	JESD8-24
1.8 V HSTL Class I 和 Class II	全部	Yes	Yes	DDR II+、QDR II+ 和 RLDRAM 2	JESD8-6
1.5 V HSTL Class I 和 Class II	全部	Yes	Yes	DDR II+、QDR II+、QDR II 和 RLDRAM 2	JESD8-6
1.2 V HSTL Class I 和 Class II	全部	Yes	Yes	通用	JESD8-16A
HSUL-12	全部	Yes	Yes	LPDDR2	—
差分 SSTL-18 Class I 和 Class II	全部	Yes	Yes	DDR2	JESD8-15
差分 SSTL-15 Class I 和 Class II	全部	Yes	Yes	DDR3	—
差分 SSTL-15	全部	Yes	Yes	DDR3	JESD79-3D
差分 SSTL-135 Class I 和 Class II	全部	Yes	Yes	DDR3L	—
差分 SSTL-125 Class I 和 Class II	全部	Yes	Yes	DDR3U	—
差分 SSTL-12 Class I 和 Class II	全部	Yes	No	RLDRAM 3	—
差分 POD12	全部	Yes	No	DDR4	JESD8-24

继续...



I/O 标准	器件类型支持	I/O 缓冲类型支持		应用	支持标准
		LVDS I/O	3V I/O		
差分 1.8 V HSTL Class I 和 Class II	全部	Yes	Yes	DDR II+、QDR II+ 和 RLD RAM 2	JESD8-6
差分 1.5 V HSTL Class I 和 Class II	全部	Yes	Yes	DDR II+、QDR II+、QDR II 和 RLD RAM 2	JESD8-6
差分 1.2 V HSTL Class I 和 Class II	全部	Yes	Yes	通用	JESD8-16A
差分 HSUL-12	全部	Yes	Yes	LPDDR2	—
LVDS	全部	Yes	No	SGMII、SFI 和 SPI	ANSI/TIA/EIA-644
Mini-LVDS	全部	Yes	No	SGMII、SFI 和 SPI	—
RSDS	全部	Yes	No	SGMII、SFI 和 SPI	—
LVPECL	全部	Yes	No	SGMII、SFI 和 SPI	—

相关链接

- [Arria 10 GX 封装的 FPGA I/O 资源](#) (第 105 页)
列出了 Arria 10 GX 封装中可用的 3 V 和 LVDS I/O 缓冲的数量。
- [Arria 10 GT 封装的 FPGA I/O 资源](#) (第 106 页)
列出了 Arria 10 GT 封装中可用的 3 V 和 LVDS I/O 缓冲的数量。
- [Arria 10 SX 封装的 FPGA I/O 资源](#) (第 107 页)
列出了 Arria 10 SX 封装中可用的 3 V 和 LVDS I/O 缓冲的数量。

5.2.2. Arria 10 器件的 HPS I/O 所支持的 I/O 标准

表 32. Arria 10 SX 器件的 HPS I/O 中所支持的 I/O 标准

I/O 标准	应用	支持标准
3.0 V LVTTL/3.0 V LVCMOS	通用	JESD8-B
2.5 V LVCMOS	通用	JESD8-5
1.8 V LVCMOS	通用	JESD8-7

5.2.3. Arria 10 器件中的 I/O 标准电平

表 33. Arria 10 I/O 标准电平

该表列出了 Arria 10 器件所支持的每种 I/O 标准的典型电源。

I/O 标准	V _{CCIO} (V)		V _{CCPT} (V) (预驱动电压)	V _{REF} (V) (输入参考电压)	V _{TT} (V) (板级匹配电压)
	输入 ⁽⁶⁾	输出			
3.0 V LVTTTL/3.0 V LVCMOS	3.0/2.5	3.0	1.8	—	—
2.5 V LVCMOS	3.0/2.5	2.5	1.8	—	—
1.8 V LVCMOS	1.8	1.8	1.8	—	—
1.5 V LVCMOS	1.5	1.5	1.8	—	—
1.2 V LVCMOS	1.2	1.2	1.8	—	—
SSTL-18 Class I 和 Class II	V _{CCPT}	1.8	1.8	0.9	0.9
SSTL-15 Class I 和 Class II	V _{CCPT}	1.5	1.8	0.75	0.75
SSTL-15	V _{CCPT}	1.5	1.8	0.75	0.75
SSTL-135 Class I 和 Class II	V _{CCPT}	1.35	1.8	0.675	—
SSTL-125 Class I 和 Class II	V _{CCPT}	1.25	1.8	0.625	—
SSTL-12 Class I 和 Class II	V _{CCPT}	1.2	1.8	0.6	—
POD12	V _{CCPT}	1.2	1.8	0.84	1.2
1.8 V HSTL Class I 和 Class II	V _{CCPT}	1.8	1.8	0.9	0.9
1.5 V HSTL Class I 和 Class II	V _{CCPT}	1.5	1.8	0.75	0.75
1.2 V HSTL Class I 和 Class II	V _{CCPT}	1.2	1.8	0.6	0.6
HSUL-12	V _{CCPT}	1.2	1.8	0.6	—
差分 SSTL-18 Class I 和 Class II	V _{CCPT}	1.8	1.8	—	0.9
差分 SSTL-15 Class I 和 Class II	V _{CCPT}	1.5	1.8	—	0.75
差分 SSTL-15	V _{CCPT}	1.5	1.8	—	0.75
差分 SSTL-135 Class I 和 Class II	V _{CCPT}	1.35	1.8	—	0.675
差分 SSTL-125 Class I 和 Class II	V _{CCPT}	1.25	1.8	—	0.625
差分 SSTL-12 Class I 和 Class II	V _{CCPT}	1.2	1.8	—	0.6
差分 POD12	V _{CCPT}	1.2	1.8	—	1.2
差分 1.8 V HSTL Class I 和 Class II	V _{CCPT}	1.8	1.8	—	0.9
差分 1.5 V HSTL Class I 和 Class II	V _{CCPT}	1.5	1.8	—	0.75
差分 1.2 V HSTL Class I 和 Class II	V _{CCPT}	1.2	1.8	—	0.6
差分 HSUL-12	V _{CCPT}	1.2	1.8	—	—
LVDS	V _{CCPT}	1.8	1.8	—	—

继续...

⁽⁶⁾ SSTL、HSTL、差分 SSTL、差分 HSTL、POD、差分 POD、LVDS、RSDS、Mini-LVDS、LVPECL、HSUL 和差分 HSUL 的输入由 V_{CCPT} 供电。



I/O 标准	V _{CCIO} (V)		V _{CCPT} (V) (预驱动电压)	V _{REF} (V) (输入参考电压)	V _{TT} (V) (板级匹配电压)
	输入 ⁽⁶⁾	输出			
Mini-LVDS	V _{CCPT}	1.8	1.8	—	—
RSDS	V _{CCPT}	1.8	1.8	—	—
LVPECL (仅差分时钟输入)	V _{CCPT}	—	1.8	—	—

相关链接

- 指南: 遵守 3.0 V 连接的器件绝对最大额定值 (第 169 页)
- 指南: VREF 资源和 VREF 管脚 (第 169 页)

5.2.4. Arria 10 器件中的 MultiVolt I/O 接口

MultiVolt I/O 接口功能支持所有封装的 Arria 10 器件与不同供电电压系统的连接:

- Arria 10 器件中的每个 I/O bank 都有其自身 V_{CCIO} 供电, 但仅支持一个 V_{CCIO} 电压。
- 所支持的 V_{CCIO} 电压是 1.2 V、1.25 V、1.35 V、1.5 V、1.8 V、2.5 V 或者 3.0 V。
- 2.5 V 和 3.0 V V_{CCIO} 仅在 3 V I/O 缓冲类型中受到支持。
- I/O 缓冲由 V_{CC}、V_{CCPT} 和 V_{CCIO} 供电。

5.3. Arria 10 器件的 Intel FPGA I/O IP 内核

I/O 系统受到几个 Intel FPGA I/O IP 内核的支持。

- Altera GPIO — 支持 GPIO 组件的操作。
- Altera LVDS SERDES — 支持高速源同步 SERDES 的操作。
- Altera OCT — 支持 OCT 校准模块。
- Altera PHYlite — 支持 strobe-based 采集 I/O 单元的动态 OCT 和 I/O 延迟。

相关链接

- PHYlite 存储器 IP 内核用户指南
- Altera GPIO IP 内核用户指南
- Altera OCT IP 内核用户指南
- Altera LVDS SERDES IP 内核用户指南

5.4. Arria 10 器件的 I/O 资源

Arria 10 器件的 GPIO Bank、SERDES 和 DPA 位置 (第 100 页)

Arria 10 器件的 GPIO 缓冲和 LVDS 通道 (第 105 页)

Arria 10 器件的 I/O Bank 组 (第 108 页)

Arria 10 器件的 I/O 纵向移植 (第 114 页)

⁽⁶⁾ SSTL、HSTL、差分 SSTL、差分 HSTL、POD、差分 POD、LVDS、RSDS、Mini-LVDS、LVPECL、HSUL 和差分 HSUL 的输入由 V_{CCPT} 供电。

5.4.1. Arria 10 器件的 GPIO Bank、SERDES 和 DPA 位置

I/O bank 位于 I/O 列。每个 I/O bank 包含其自身的 PLL、DPA 和 SERDES 电路。

要了解关于每种器件封装中可用的模块化 I/O bank 的详细信息，请参考相关信息。

图 70. Arria 10 GX 160 和 GX 220 器件的 I/O Bank

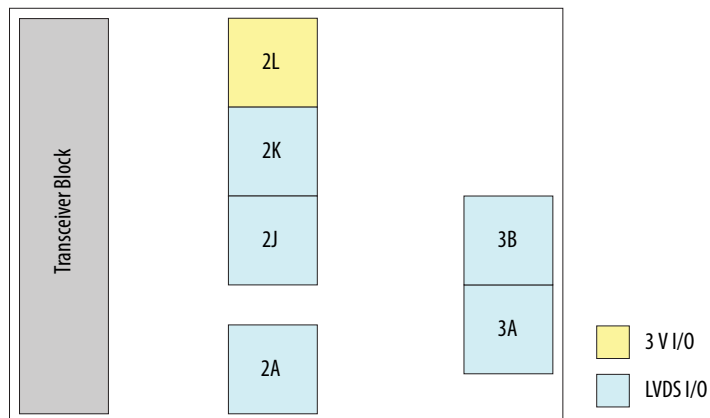


图 71. Arria 10 SX 160 和 SX 220 器件的 I/O Bank

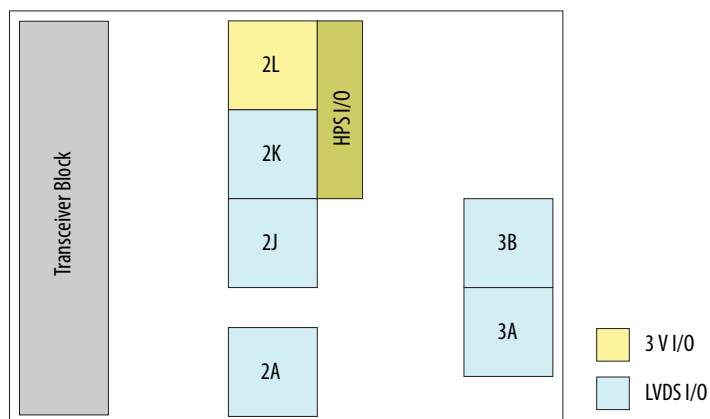


图 72. Arria 10 GX 270 和 GX 320 器件的 I/O Bank

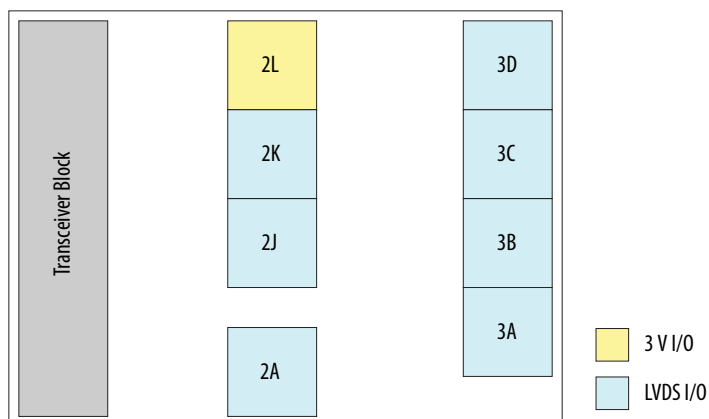


图 73. Arria 10 SX 270 和 SX 320 器件的 I/O Bank

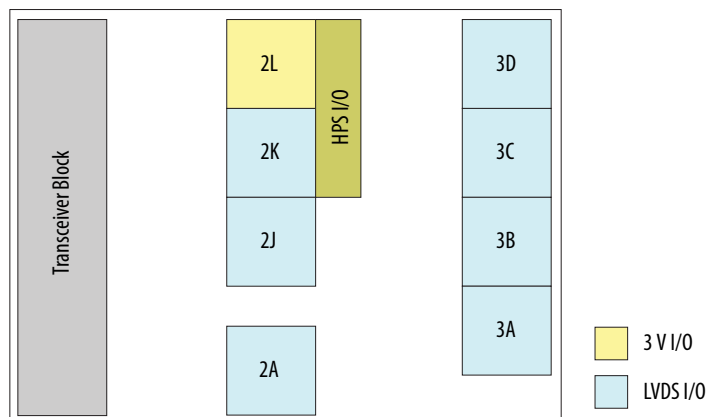


图 74. Arria 10 GX 480 器件的 I/O Bank

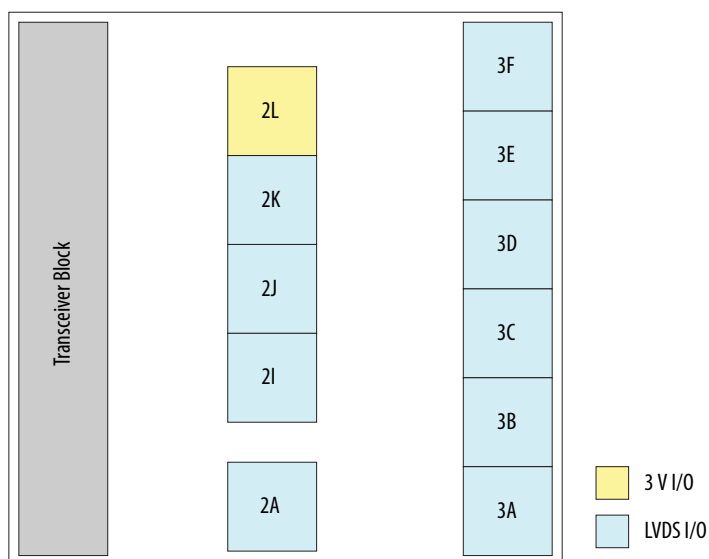


图 75. Arria 10 SX 480 器件的 I/O Bank

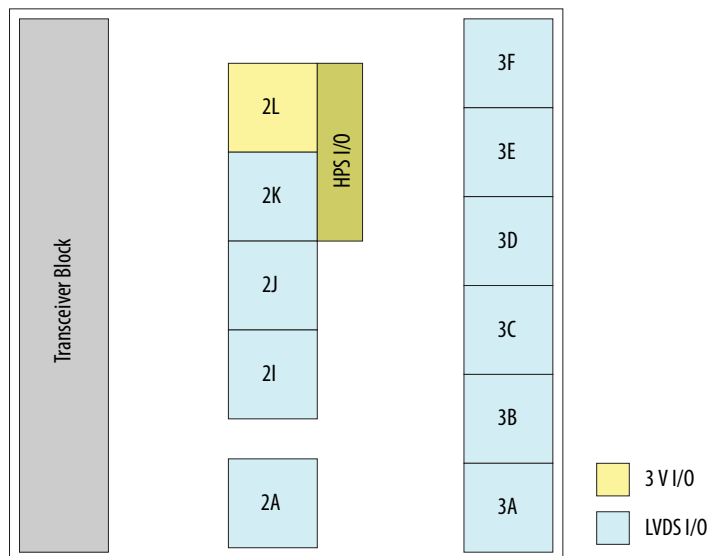


图 76. Arria 10 GX 570 和 GX 660 器件的 I/O Bank

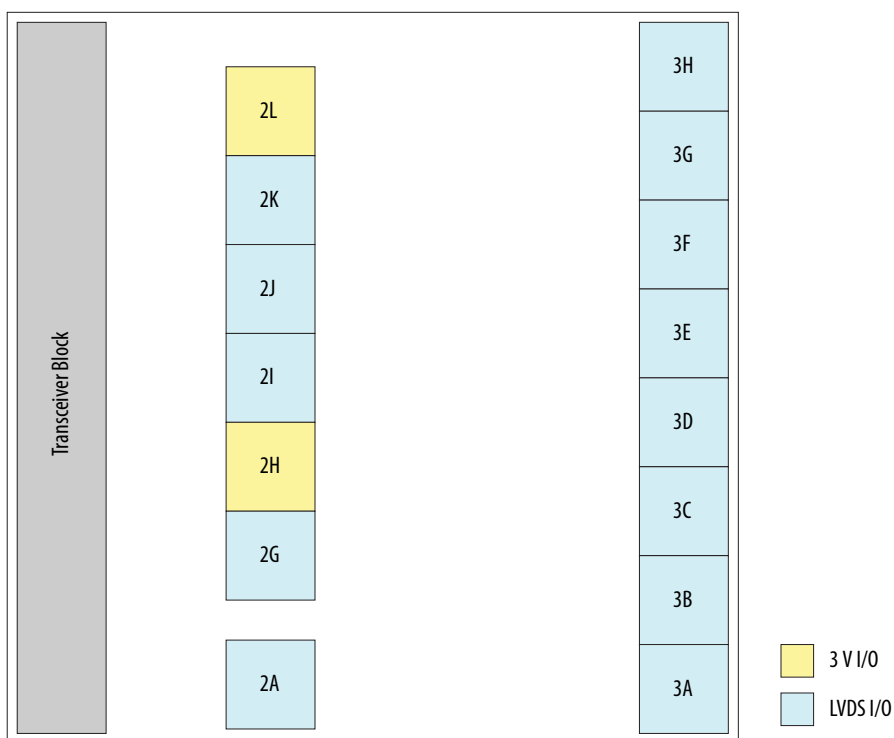


图 77. Arria 10 SX 570 和 SX 660 器件的 I/O Bank

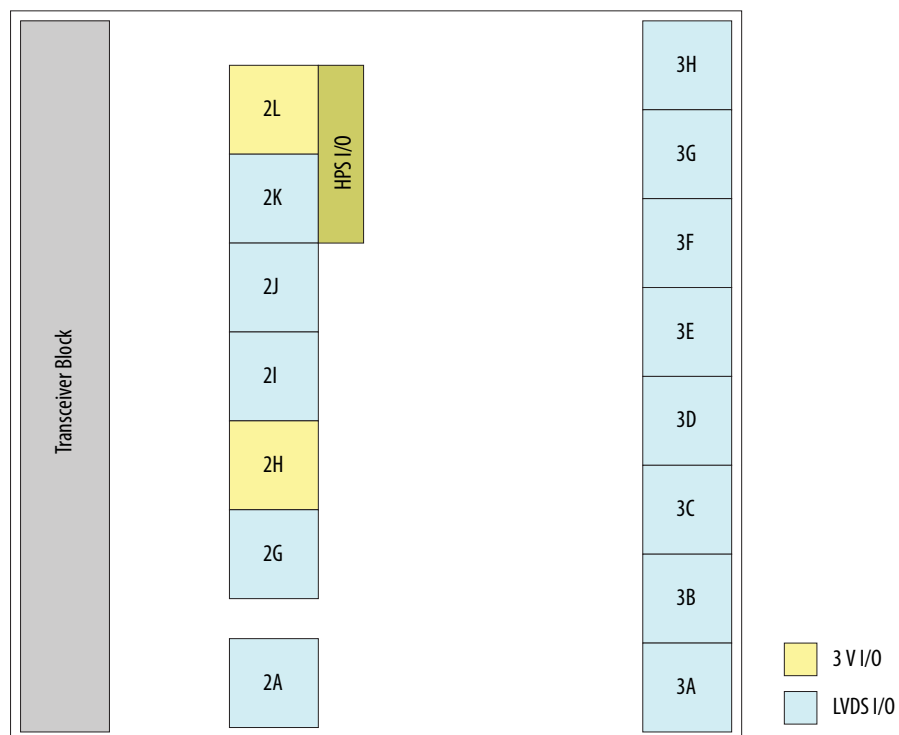
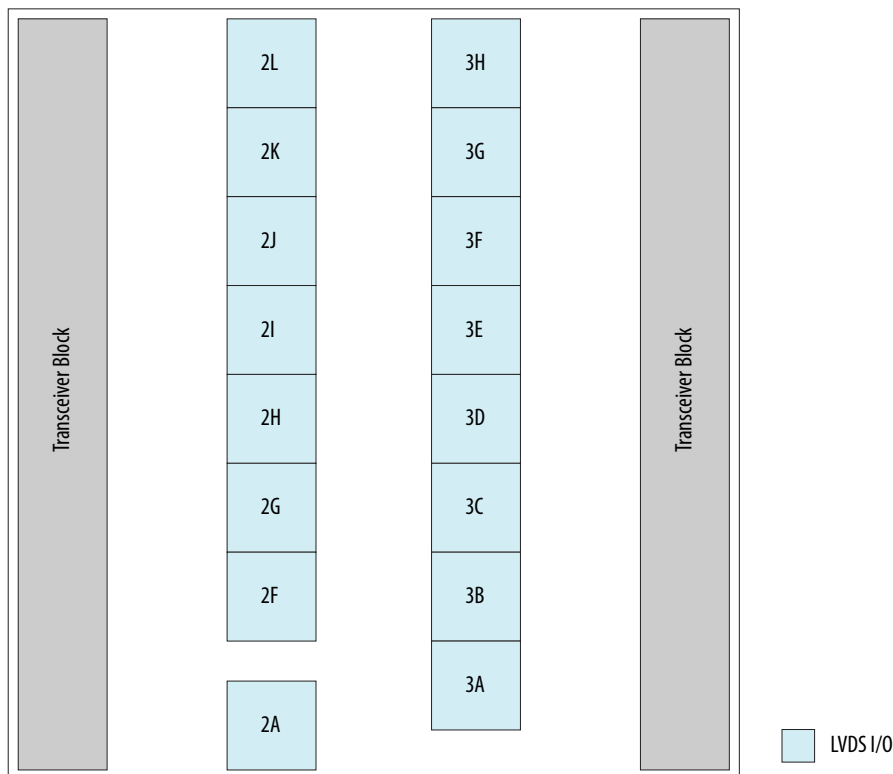


图 78. Arria 10 GX 900、GX 1150、GT 900 和 GT 1150 器件的 I/O Bank



相关链接

- [器件收发器的布局](#)
提供 Arria 10 器件中关于收发器 bank 的更多信息。
- [Arria 10 GX 器件的模块化 I/O Bank](#) (第 108 页)
列出了每个 Arria 10 GX 封装中可用的 I/O bank 的 I/O 管脚数。
- [Arria 10 GT 器件的模块化 I/O Bank](#) (第 111 页)
列出了每个 Arria 10 GT 封装中可用的 I/O bank 的 I/O 管脚数。
- [Arria 10 SX 器件的模块化 I/O Bank](#) (第 112 页)
列出了每个 Arria 10 SX 封装中可用的 I/O bank 的 I/O 管脚数。
- [Arria 10 GX 封装的 FPGA I/O 资源](#) (第 105 页)
列出了 Arria 10 GX 封装中可用的 3 V 和 LVDS I/O 缓冲的数量。
- [Arria 10 GT 封装的 FPGA I/O 资源](#) (第 106 页)
列出了 Arria 10 GT 封装中可用的 3 V 和 LVDS I/O 缓冲的数量。
- [Arria 10 SX 封装的 FPGA I/O 资源](#) (第 107 页)
列出了 Arria 10 SX 封装中可用的 3 V 和 LVDS I/O 缓冲的数量。
- [Arria 10 器件管脚输出\(Pin-Out\)](#)
提供了每种 Arria 10 器件的管脚输出文件。对于 SoC 器件，管脚输出文件也列出了 FPGA 架构和 HPS 共享的 I/O bank。
- [Altera GPIO IP 内核用户指南](#)
- [Arria 10 器件的 PLL 和时钟](#) (第 152 页)



5.4.2. Arria 10 器件的 GPIO 缓冲和 LVDS 通道

5.4.2.1. Arria 10 GX 封装的 FPGA I/O 资源

表 34. Arria 10 GX 器件的 GPIO 缓冲和 LVDS 通道

- U19 封装是一种 0.8 mm 间距的焊球栅阵列。所有其它的封装是 1.0 mm 间距的焊球栅阵列。
- LVDS 通道数不包括专用的时钟管脚。

产品系列	封装		GPIO			LVDS 通道
	代码	类型	3 V I/O	LVDS I/O	总数量	
GX 160	U19	484-pin UBGA	48	148	196	74
	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	240	288	120
GX 220	U19	484-pin UBGA	48	148	196	74
	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	240	288	120
GX 270	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	312	360	156
	F34	1,152-pin FBGA	48	336	384	168
	F35	1,152-pin FBGA	48	336	384	168
GX 320	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	312	360	156
	F34	1,152-pin FBGA	48	336	384	168
	F35	1,152-pin FBGA	48	336	384	168
GX 480	F29	780-pin FBGA	48	312	360	156
	F34	1,152-pin FBGA	48	444	492	222
	F35	1,152-pin FBGA	48	348	396	174
GX 570	F34	1,152-pin FBGA	48	444	492	222
	F35	1,152-pin FBGA	48	348	396	174
	NF40	1,517-pin FBGA	48	540	588	270
	KF40	1,517-pin FBGA	96	600	696	300
GX 660	F34	1,152-pin FBGA	48	444	492	222
	F35	1,152-pin FBGA	48	348	396	174
	NF40	1,517-pin FBGA	48	540	588	270
	KF40	1,517-pin FBGA	96	600	696	300
GX 900	F34	1,152-pin FBGA	0	504	504	252
	NF40	1,517-pin FBGA	0	600	600	300
	RF40	1,517-pin FBGA	0	342	342	154
	NF45	1,932-pin FBGA	0	768	768	384

继续...

产品系列	封装		GPIO			LVDS 通道
	代码	类型	3 V I/O	LVDS I/O	总数量	
	SF45	1,932-pin FBGA	0	624	624	312
	UF45	1,932-pin FBGA	0	480	480	240
GX 1150	F34	1,152-pin FBGA	0	504	504	252
	NF40	1,517-pin FBGA	0	600	600	300
	RF40	1,517-pin FBGA	0	342	342	154
	NF45	1,932-pin FBGA	0	768	768	384
	SF45	1,932-pin FBGA	0	624	624	312
	UF45	1,932-pin FBGA	0	480	480	240

相关链接

- [Arria 10 GX 器件的模块化 I/O Bank](#) (第 108 页)
列出了每个 Arria 10 GX 封装中可用的 I/O bank 的 I/O 管脚数。
- [Arria 10 器件的 FPGA I/O 所支持的 I/O 标准](#) (第 96 页)
- [Arria 10 器件的 GPIO Bank、SERDES 和 DPA 位置](#) (第 100 页)
- [Arria 10 器件中的 I/O 和差分 I/O 缓冲](#) (第 95 页)

5.4.2.2. Arria 10 GT 封装的 FPGA I/O 资源

表 35. Arria 10 GT 器件的 GPIO 缓冲和 LVDS 通道

- SF45 封装是一种 1.0 mm 间距的焊球栅阵列。
- LVDS 通道数不包括专用的时钟管脚。

产品系列	封装		GPIO 缓冲			LVDS 通道
	代码	类型	3 V I/O	LVDS I/O	总数量	
GT 900	SF45	1,932-pin FBGA	0	624	624	312
GT 1150	SF45	1,932-pin FBGA	0	624	624	312

相关链接

- [Arria 10 GT 器件的模块化 I/O Bank](#) (第 111 页)
列出了每个 Arria 10 GT 封装中可用的 I/O bank 的 I/O 管脚数。
- [Arria 10 器件的 FPGA I/O 所支持的 I/O 标准](#) (第 96 页)
- [Arria 10 器件的 GPIO Bank、SERDES 和 DPA 位置](#) (第 100 页)
- [Arria 10 器件中的 I/O 和差分 I/O 缓冲](#) (第 95 页)



5.4.2.3. Arria 10 SX 封装的 FPGA I/O 资源

表 36. Arria 10 SX 器件的 GPIO 缓冲和 LVDS 通道

- U19 封装是一种 0.8 mm 间距的焊球栅阵列。所有其它的封装是 1.0 mm 间距的焊球栅阵列。
- LVDS 通道数不包括专用的时钟管脚。

产品系列	封装		GPIO 缓冲			LVDS 通道
	代码	类型	3 V I/O	LVDS I/O	总数量	
SX 160	U19	484-pin UBGA	48	148	196	74
	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	240	288	120
SX 220	U19	484-pin UBGA	48	148	196	74
	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	240	288	120
SX 270	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	312	360	156
	F34	1,152-pin FBGA	48	336	384	168
	F35	1,152-pin FBGA	48	336	384	168
SX 320	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	312	360	156
	F34	1,152-pin FBGA	48	336	384	168
	F35	1,152-pin FBGA	48	336	384	168
SX 480	F29	780-pin FBGA	48	312	360	156
	F34	1,152-pin FBGA	48	444	492	222
	F35	1,152-pin FBGA	48	348	396	174
SX 570	F34	1,152-pin FBGA	48	444	492	222
	F35	1,152-pin FBGA	48	348	396	174
	NF40	1,517-pin FBGA	48	540	588	270
	KF40	1,517-pin FBGA	96	600	696	300
SX 660	F34	1,152-pin FBGA	48	444	492	222
	F35	1,152-pin FBGA	48	348	396	174
	NF40	1,517-pin FBGA	48	540	588	270
	KF40	1,517-pin FBGA	96	600	696	300

相关链接

- [Arria 10 SX 器件的模块化 I/O Bank](#) (第 112 页)
列出了每个 Arria 10 SX 封装中可用的 I/O bank 的 I/O 管脚数。
- [Arria 10 器件的 FPGA I/O 所支持的 I/O 标准](#) (第 96 页)
- [Arria 10 器件的 GPIO Bank、SERDES 和 DPA 位置](#) (第 100 页)
- [Arria 10 器件中的 I/O 和差分 I/O 缓冲](#) (第 95 页)

5.4.3. Arria 10 器件的 I/O Bank 组

Arria 10 器件中的 I/O 管脚以模块化 I/O bank 组的形式进行组织：

- 模块化 I/O bank 有独立的供电，使得每个 bank 可以支持不同的 I/O 标准。
- 每个模块化 I/O bank 可以支持使用相同电压的多个 I/O 标准。

相关链接

- [Arria 10 GX 器件的模块化 I/O Bank](#) (第 108 页)
- [Arria 10 GT 器件的模块化 I/O Bank](#) (第 111 页)
- [Arria 10 SX 器件的模块化 I/O Bank](#) (第 112 页)

5.4.3.1. Arria 10 GX 器件的模块化 I/O Bank

下表列出了可用的 I/O bank、每个 bank 中 I/O 管脚的总数量以及每种产品系列和 Arria 10 GX 器件系列的器件封装的 I/O 管脚的总数量。

表 37. Arria 10 GX 160 和 GX 220 器件的模块化 I/O Bank

产品系列		GX 160			GX 220		
封装		U19	F27	F29	U19	F27	F29
I/O Bank	2A	48	48	48	48	48	48
	2J	48	48	48	48	48	48
	2K	48	48	48	48	48	48
	2L	48	48	48	48	48	48
	3A	—	48	48	—	48	48
	3B	4	—	48	4	—	48
总数量		196	240	288	196	240	288

表 38. Arria 10 GX 270 和 GX 320 器件的模块化 I/O Bank

产品系列		GX 270				GX 320			
封装		F27	F29	F34	F35	F27	F29	F34	F35
I/O Bank	2A	48	48	48	48	48	48	48	48
	2J	48	48	48	48	48	48	48	48
	2K	48	48	48	48	48	48	48	48
	2L	48	48	48	48	48	48	48	48
	3A	48	48	48	48	48	48	48	48
	3B	—	48	48	48	—	48	48	48
	3C	—	48	48	48	—	48	48	48
	3D	—	24	48	48	—	24	48	48
总数量		240	360	384	384	240	360	384	384



表 39. Arria 10 GX 480 器件的模块化 I/O Bank

产品系列		GX 480		
封装		F29	F34	F35
I/O Bank	2A	48	48	48
	2I	—	12	12
	2J	48	48	48
	2K	48	48	48
	2L	48	48	48
	3A	48	48	48
	3B	48	48	48
	3C	48	48	48
	3D	24	48	48
	3E	—	48	—
	3F	—	48	—
总数量		360	492	396

表 40. Arria 10 GX 570 和 GX 660 器件的模块化 I/O Bank

产品系列		GX 570				GX 660			
封装		F34	F35	NF40	KF40	F34	F35	NF40	KF40
I/O Bank	2A	48	48	48	48	48	48	48	48
	2G	—	—	—	24	—	—	—	24
	2H	—	—	—	48	—	—	—	48
	2I	12	12	12	48	12	12	12	48
	2J	48	48	48	48	48	48	48	48
	2K	48	48	48	48	48	48	48	48
	2L	48	48	48	48	48	48	48	48
	3A	48	48	48	48	48	48	48	48
	3B	48	48	48	48	48	48	48	48
	3C	48	48	48	48	48	48	48	48
	3D	48	48	48	48	48	48	48	48
	3E	48	—	48	48	48	—	48	48
	3F	48	—	48	48	48	—	48	48
	3G	—	—	48	48	—	—	48	48
	3H	—	—	48	48	—	—	48	48
总数量		492	396	588	696	492	396	588	696

表 41. Arria 10 GX 900 器件的模块化 I/O Bank

产品系列		GX 900					
封装		F34	NF40	RF40	NF45	SF45	UF45
I/O Bank	2A	48	48	48	48	48	48
	2F	—	—	48	48	—	—
	2G	—	—	—	48	—	—
	2H	—	—	—	48	—	—
	2I	24	24	—	48	48	48
	2J	48	48	—	48	48	48
	2K	48	48	48	48	48	48
	2L	48	48	48	48	48	48
	3A	48	48	28	48	48	48
	3B	48	48	27	48	48	48
	3C	48	48	—	48	48	48
	3D	48	48	—	48	48	48
	3E	48	48	—	48	48	48
	3F	48	48	—	48	48	—
	3G	—	48	47	48	48	—
	3H	—	48	48	48	48	—
总数量		504	600	342	768	624	480

表 42. Arria 10 GX 1150 器件的模块化 I/O Bank

产品系列		GX 1150					
封装		F34	NF40	RF40	NF45	SF45	UF45
I/O Bank	2A	48	48	48	48	48	48
	2F	—	—	48	48	—	—
	2G	—	—	—	48	—	—
	2H	—	—	—	48	—	—
	2I	24	24	—	48	48	48
	2J	48	48	—	48	48	48
	2K	48	48	48	48	48	48
	2L	48	48	48	48	48	48
	3A	48	48	28	48	48	48
	3B	48	48	27	48	48	48
	3C	48	48	—	48	48	48
	3D	48	48	—	48	48	48
	3E	48	48	—	48	48	48
继续...							



产品系列		GX 1150					
封装		F34	NF40	RF40	NF45	SF45	UF45
	3F	48	48	—	48	48	—
	3G	—	48	47	48	48	—
	3H	—	48	48	48	48	—
总数量		504	600	342	768	624	480

相关链接

- [Arria 10 器件的 GPIO Bank、SERDES 和 DPA 位置 \(第 100 页\)](#)
- [Arria 10 GX 封装的 FPGA I/O 资源 \(第 105 页\)](#)
- [Arria 10 器件的 I/O Bank 组 \(第 108 页\)](#)
- [指南: Altera LVDS SERDES IP 内核实例 \(第 172 页\)](#)

5.4.3.2. Arria 10 GT 器件的模块化 I/O Bank

下表列出了可用的 I/O bank、每个 bank 中 I/O 管脚的总数量以及每种产品系列和 Arria 10 GT 器件系列的器件封装的 I/O 管脚的总数量。

表 43. Arria 10 GT 900 和 GT 1150 器件的模块化 I/O Bank

产品系列		GT 900	GT 1150
封装		SF45	SF45
I/O Bank	2A	48	48
	2I	48	48
	2J	48	48
	2K	48	48
	2L	48	48
	3A	48	48
	3B	48	48
	3C	48	48
	3D	48	48
	3E	48	48
	3F	48	48
	3G	48	48
	3H	48	48
总数量		624	624

相关链接

- [Arria 10 器件的 GPIO Bank、SERDES 和 DPA 位置 \(第 100 页\)](#)
- [Arria 10 GT 封装的 FPGA I/O 资源 \(第 106 页\)](#)
- [Arria 10 器件的 I/O Bank 组 \(第 108 页\)](#)

- 指南: Altera LVDS SERDES IP 内核实例 (第 172 页)

5.4.3.3. Arria 10 SX 器件的模块化 I/O Bank

下表列出了可用的 I/O bank、每个 bank 中 I/O 管脚的总数量以及每种产品系列和 Arria 10 SX 器件系列的器件封装的 I/O 管脚的总数量。

表 44. Arria 10 SX 160 和 SX 220 器件的模块化 I/O Bank

产品系列		SX 160			SX 220		
封装		U19	F27	F29	U19	F27	F29
I/O Bank	2A	48	48	48	48	48	48
	2J	48	48	48	48	48	48
	2K	48	48	48	48	48	48
	2L	48	48	48	48	48	48
	3A	—	48	48	—	48	48
	3B	4	—	48	4	—	48
总数量		196	240	288	196	240	288

表 45. Arria 10 SX 270 和 SX 320 器件的模块化 I/O Bank

产品系列		SX 270				SX 320			
封装		F27	F29	F34	F35	F27	F29	F34	F35
I/O Bank	2A	48	48	48	48	48	48	48	48
	2J	48	48	48	48	48	48	48	48
	2K	48	48	48	48	48	48	48	48
	2L	48	48	48	48	48	48	48	48
	3A	48	48	48	48	48	48	48	48
	3B	—	48	48	48	—	48	48	48
	3C	—	48	48	48	—	48	48	48
	3D	—	24	48	48	—	24	48	48
总数量		240	360	384	384	240	360	384	384

表 46. Arria 10 SX 480 器件的模块化 I/O Bank

产品系列		SX 480		
封装		F29	F34	F35
I/O Bank	2A	48	48	48
	2I	—	12	12
	2J	48	48	48
	2K	48	48	48
	2L	48	48	48
	3A	48	48	48
继续...				



产品系列		SX 480		
封装		F29	F34	F35
	3B	48	48	48
	3C	48	48	48
	3D	24	48	48
	3E	—	48	—
	3F	—	48	—
总数量		360	492	396

表 47. Arria 10 SX 570 和 SX 660 器件的模块化 I/O Bank

产品系列		SX 570				SX 660			
封装		F34	F35	NF40	KF40	F34	F35	NF40	KF40
I/O Bank	2A	48	48	48	48	48	48	48	48
	2G	—	—	—	24	—	—	—	24
	2H	—	—	—	48	—	—	—	48
	2I	12	12	12	48	12	12	12	48
	2J	48	48	48	48	48	48	48	48
	2K	48	48	48	48	48	48	48	48
	2L	48	48	48	48	48	48	48	48
	3A	48	48	48	48	48	48	48	48
	3B	48	48	48	48	48	48	48	48
	3C	48	48	48	48	48	48	48	48
	3D	48	48	48	48	48	48	48	48
	3E	48	—	48	48	48	—	48	48
	3F	48	—	48	48	48	—	48	48
	3G	—	—	48	48	—	—	48	48
	3H	—	—	48	48	—	—	48	48
总数量		492	396	588	696	492	396	588	696

相关链接

- [Arria 10 器件的 GPIO Bank、SERDES 和 DPA 位置 \(第 100 页\)](#)
- [Arria 10 SX 封装的 FPGA I/O 资源 \(第 107 页\)](#)
- [Arria 10 器件的 I/O Bank 组 \(第 108 页\)](#)
- [指南: Altera LVDS SERDES IP 内核实例 \(第 172 页\)](#)

5.4.4. Arria 10 器件的 I/O 纵向移植

图 79. Arria 10 产品系列之间的移植能力

- 箭头表示移植路径。包含在每条纵向移植路径中的器件呈阴影。相同路径中有更少资源的器件呈更浅的阴影。
- 为了在同一移植路径的产品系列中实现完整的 I/O 移植，可限制 I/O 和收发器的使用，以最低的 I/O 和收发器数量相匹配的产品系列。
- 源器件中的一个 LVDS I/O bank 可能被映射到目标器件的 3 V I/O bank。要使用高于 533 MHz 的存储器接口时钟频率，请为外部存储器接口管脚分配在两种器件中都是 LVDS I/O 的 bank。
- 在同一封装类型的一些产品系列之间可能会有 0.15 mm 的封装高度差。
- 一些移植路径没有显示在 Quartus Prime 软件的 Pin Migration View 中。

器件系列	产品型号	封装										
		U19	F27	F29	F34	F35	KF40	NF40	RF40	NF45	SF45	UF45
Arria 10 GX	GX 160	↑	↑	↑								
	GX 220	↓	↑	↑								
	GX 270		↑	↑	↑	↑						
	GX 320		↓	↑	↑	↑						
	GX 480			↓	↑	↑	↑	↑				
	GX 570				↑	↑	↑	↑	↑	↑	↑	↑
	GX 660				↑	↑	↑	↑	↑	↑	↑	↑
	GX 900				↑			↑	↑	↑	↑	↑
	GX 1150				↑			↑	↑	↑	↑	↑
Arria 10 GT	GT 900										↓	
	GT 1150											↓
Arria 10 SX	SX 160	↑	↑	↑								
	SX 220	↓	↑	↑								
	SX 270		↑	↑	↑	↑						
	SX 320		↓	↑	↑	↑						
	SX 480			↓	↑	↑	↑	↑				
	SX 570				↑	↑	↑	↑	↑	↑	↑	↑
	SX 660				↑	↑	↑	↑	↑	↑	↑	↑

注意: 要验证管脚移植能力，请使用 Quartus Prime 软件 Pin Planner 的 Pin Migration View。

相关链接

- [验证管脚移植兼容性](#) (第 114 页)
- [移植分配到另一个目标器件](#)
提供了关于纵向 I/O 移植的更多信息。

5.4.4.1. 验证管脚移植兼容性

可使用 Quartus Prime 软件 Pin Planner 中的 **Pin Migration View** 窗口辅助验证管脚分配是否成功移植到其他器件中。可在使用同一器件封装时，纵向移植到具有不同密度的器件，或在不同密度和球数的封装间进行移植。

1. 打开 **Assignments > Pin Planner** 并创建管脚分配。
2. 如有需要，可执行下列选项之一，使用设计中的节点名称填充 Pin Planner:
 - Analysis & Elaboration (分析&拟订)
 - Analysis & Synthesis (分析&综合)



- Fully compile the design (完全编译设计)
- 3. 然后，在菜单上，点击 **View > Pin Migration View**。
- 4. 选择或更改移植器件：
 - a. 点击 **Device** 打开 **Device** 对话框。
 - b. **Migration compatibility** 下点选 **Migration Devices**。
- 5. 显示关于该管脚更多的信息：
 - a. 右键点击 **Pin Migration View** 窗口中任意位置，并选择 **Show Columns**。
 - b. 然后，点击要显示的管脚功能。
- 6. 如果仅查看至少一个移植器件中的管脚与移植结果中相应管脚的不同功能，则打开 **Show migration differences**。
- 7. 点击 **Pin Finder** 打开 **Pin Finder** 对话框，以查找并突出显示具有特定功能的管脚。
如果仅在 **Pin Finder** 对话框中查看因最近查询而突显的管脚，则打开 **Show only highlighted pins**。
- 8. 将管脚移植信息导出到 Comma-Separated Value 文件 (.csv)，请点击 **Export**。

相关链接

- [Arria 10 器件的 I/O 纵向移植](#) (第 114 页)
- [移植分配到另一个目标器件](#)
提供了关于纵向 I/O 移植的更多信息。

5.5. Arria 10 器件的体系结构和 I/O 的一般功能

[Arria 10 器件中的 I/O 单元结构](#) (第 115 页)

[Arria 10 器件的 I/O 管脚特性](#) (第 117 页)

[Arria 10 器件的可编程 IOE 的特性](#) (第 118 页)

[Arria 10 器件的片上 I/O 匹配](#) (第 123 页)

[Arria 10 器件的外部 I/O 匹配](#) (第 132 页)

5.5.1. Arria 10 器件中的 I/O 单元结构

在 Arria 10 器件中的 I/O 单元 (IOE) 包含双向 I/O 缓冲器和 I/O 寄存器，以支持完全嵌入式双向单倍数据速率 (SDR) 或双倍数据速率 (DDR) 的传送。

IOE 位于 Arria 10 器件内核架构的 I/O 列中。

Arria 10 SX 器件还具有用于 HPS 的 IOE。

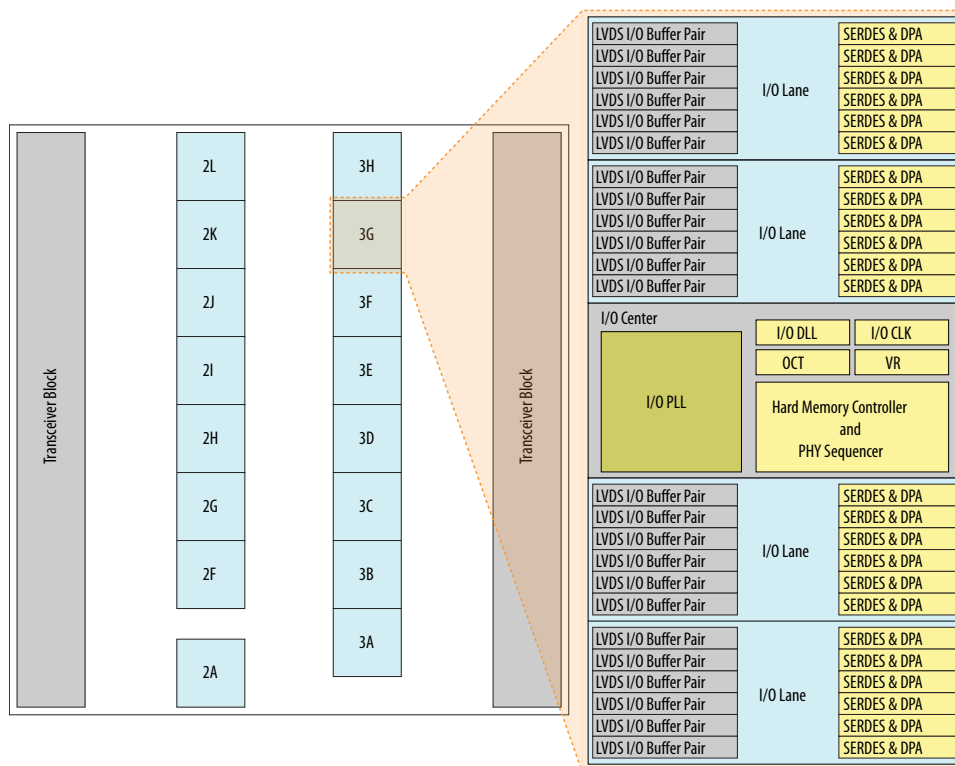
GPIO IOE 寄存器由 DDR 寄存器、半速率寄存器以及输入、输出和输出使能 (OE) 路径的发送器延迟链组成：

- 可从组合式路径或寄存路径中获取数据。
- 只有内核时钟为数据提供时钟。
- 从内核布线的半速率时钟对半速率寄存器提供时钟。
- 内核的全速率时钟对全速率寄存器提供时钟。

5.5.1.1. Arria 10 器件的 I/O Bank 体系结构

在每个 I/O bank 中，有 4 个 I/O 通道，每个通道含有 12 个 I/O 管脚。除了 I/O 通道，每个 I/O bank 也包含专用的电路，包括 I/O PLL、DPA 模块、SERDES、硬核存储控制器和 I/O 序列器。

图 80. I/O Bank 结构



相关链接

指南: [VREF 资源](#)和 [VREF 管脚](#) (第 169 页)
介绍了与 I/O 通道相关的 VREF 限制。

5.5.1.2. Arria 10 器件的 I/O 缓冲器和寄存器

I/O 寄存器由用于处理管脚至内核的数据的输入路径、用于处理内核至管脚的数据的输出路径和用于处理 OE 信号至输出缓冲的输出使能(OE)路径组成。这些寄存器实现更快的源同步(source-synchronous)寄存器到寄存器(register-to-register)的传输和重同步。

输入和输出路径包含下面的模块:

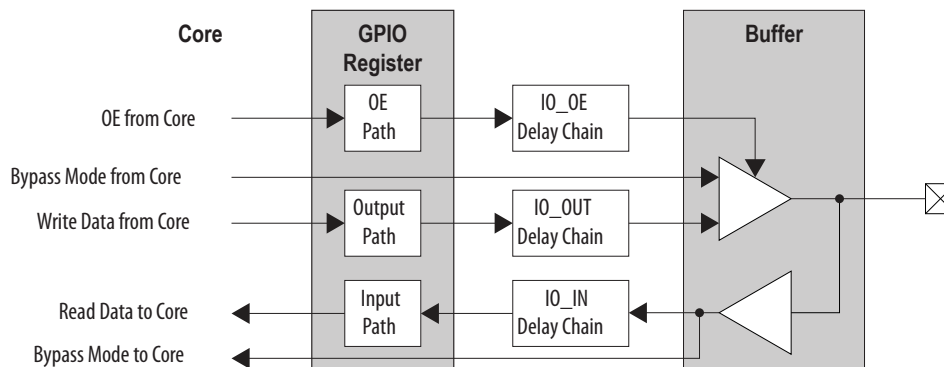
- 输入寄存器 — 支持从外设到内核传输半/全速率数据，并支持从 I/O 缓冲器中采集双倍或单倍数据速率的数据。
- 输出寄存器 — 支持从内核到外设传输半/全速率数据，并支持将双倍或单倍数据速率的数据传输到 I/O 缓冲器。
- OE 寄存器 — 支持从内核到外设传输半速率或全速率数据，并支持将单速率的数据传输到 I/O 缓冲器。

输入和输出路径也支持下面的功能：

- 时钟使能。
- 异步或同步复位。
- 输入和输出路径的旁路模式。
- 输入和输出路径上的延迟链。

图 81. Arria 10 器件的 IOE 结构

该图显示了 Arria 10 FPGA 的 IOE 结构。



5.5.2. Arria 10 器件的 I/O 管脚特性

[开漏输出 \(第 117 页\)](#)

[总线保持电路 \(第 117 页\)](#)

[弱上拉电阻 \(第 118 页\)](#)

5.5.2.1. 开漏输出

每个 I/O 管脚的可选开漏输出相当于一个集电极开路输出。如果它被配置为开漏，那么输出逻辑值为高阻或者逻辑低电平。

使用一个外部电阻将信号上拉到逻辑高电平。

5.5.2.2. 总线保持电路

每个 I/O 管脚提供一个仅在配置完成后才有效的可选总线保持功能。当器件进入用户模式时，总线保持电路采集配置最后出现的在管脚上的值。

总线保持电路使用一个额定阻值(R_{BH})大约为 7 k Ω 的电阻，将信号电平弱拉至管脚最后驱动的状态。总线保持电路将保持该管脚的状态直到出现下一个输入信号。由此，当总线处于三态时，您不需要外部上拉或者下拉电阻来保持信号电平。

对于每个 I/O 管脚，可以单独地指定总线保持电路将非驱动管脚拉离输入阈值电压 — 因为噪声能够导致意外的高频切换。为了防止过度驱动信号，总线保持电路驱动的 I/O 管脚的电压电平低于 V_{CCIO} 电平。

如果使能了总线保持功能，那么将不能使用可编程上拉选项。要配置差分信号的 I/O 管脚，请禁用总线保持功能。

5.5.2.3. 弱上拉电阻

每个 I/O 管脚在用户模式期间都提供了一个可选的可编程上拉电阻。该上拉电阻，通常为 25 k Ω ，将 I/O 微弱地保持到 V_{CCIO} 电平。

Arria 10 器件仅在用户 I/O 管脚上支持可编程的弱上拉电阻，但在专用配置管脚、专用时钟管脚或者 JTAG 管脚上不支持。

如果使能该选项，那么不能使用总线保持功能。

5.5.3. Arria 10 器件的可编程 IOE 的特性

表 48. Arria 10 可编程 IOE 功能设置和约束名称

性能	设置	条件	Quartus Prime 约束名称
摆率控制	0 (慢), 1 (快速)。默认值是 1。	使用 R _S OCT 功能时禁用。	SLEW_RATE
I/O 延迟	请参考器件手册	—	INPUT_DELAY_CHAIN OUTPUT_DELAY_CHAIN
开漏输出	On, Off。默认值是 Off。	—	AUTO_OPEN_DRAIN_PINS
总线保持	On, Off。默认值是 Off。	使用弱上拉电阻功能时禁用。	ENABLE_BUS_HOLD_CIRCUITRY
弱上拉电阻	On, Off。默认值是 Off。	使用总线保持功能时禁用。	WEAK_PULL_UP_RESISTOR
预加重	0 (禁用), 1 (使能), 默认为 1。	—	PROGRAMMABLE_PREEMPHASIS
差分输出电压	0 (低), 1 (中低), 2 (中高), 3 (高)。默认为 2。	—	PROGRAMMABLE_VOD

表 49. Arria 10 可编程 IOE 特性 I/O 缓冲器类型和 I/O 标准支持

该表列出了支持可编程 IOE 功能的 I/O 缓冲器类型和 I/O 标准。有关每种 I/O 缓冲器类型可用的 I/O 标准的更多信息，请参考相关的信息。

性能	I/O 缓冲器类型支持			I/O 标准支持
	LVDS I/O	3 V I/O	HPS I/O (仅适用于 SoC 器件)	
摆率控制	Yes	Yes	Yes	<ul style="list-style-type: none">• 3.0 V LVTTTL• 1.2 V、1.5 V、1.8 V、和 3.0 V LVCMOS• SSTL-18、SSTL-15、SSTL-135、SSTL-125 和 SSTL-12• 1.2 V、1.5 V 和 1.8 V HSTL• HSUL-12• POD12• 差分 SSTL-18、差分 SSTL-15、差分 SSTL-135、差分 SSTL-125 和差分 SSTL-12• 差分 1.2 V、1.5 V 和 1.8 V HSTL• 差分 HSUL-12
I/O 延迟	Yes	Yes	—	
开漏输出	Yes	Yes	Yes	
总线保持	Yes	Yes	Yes	

继续...



性能	I/O 缓冲器类型支持			I/O 标准支持
	LVDS I/O	3 V I/O	HPS I/O (仅适用于 SoC 器件)	
弱上拉电阻	Yes	Yes	Yes	
预加重	Yes	—	—	<ul style="list-style-type: none"> • LVDS • RSDS • Mini-LVDS • LVPECL • 差分 POD12
差分输出电压	Yes	—	—	<ul style="list-style-type: none"> • LVDS • RSDS • Mini-LVDS • LVPECL

相关链接

- [可编程 IOE 延迟](#)
- [可编程电流强度](#) (第 119 页)
- [可编程的输出摆率控制](#) (第 121 页)
- [可编程 IOE 延迟](#) (第 121 页)
- [可编程开漏输出](#) (第 121 页)
- [可编程预加重](#) (第 121 页)
- [可编程差分输出电压](#) (第 122 页)
- [Arria 10 器件的 FPGA I/O 所支持的 I/O 标准](#) (第 96 页)
列出了 LVDS I/O 和 3 V I/O 缓冲支持的 I/O 标准。
- [Arria 10 器件的 HPS I/O 所支持的 I/O 标准](#) (第 97 页)
列出了 HPS I/O 缓冲器支持的 I/O 标准。

5.5.3.1. 可编程电流强度

您可以通过修改电流驱动强度来减少远距离传输线路或者传统背板造成的高信号衰减影响。

注意: 要使用可编程电流强度，必须在 Quartus Prime 软件中指定电流强度约束。如果没有明确的约束，Quartus Prime 软件就会使用这些预定义的默认值：

- 所有 HSTL 和 SSTL Class I 以及所有非电压参考 I/O 标准 — 50 Ω 无校准的 R_S OCT
- 所有 HSTL 和 SSTL Class II I/O 标准 — 25 Ω 无校准的 R_S OCT
- POD12 I/O 标准 — 34 Ω 无校准的 R_S OCT

表 50. Arria 10 器件的可编程电流强度设置

每个 Arria 10 器件 I/O 管脚上的输出缓冲对于下表列出的 I/O 标准有一个可编程电流强度控制。

I/O 标准	I _{OH} / I _{OL} 电流强度设置(mA)或者 DDR3 OCT 设置 (Ω) (默认设置以粗体表示)	在 HPS 中支持 (仅适用于 SoC 器件) (7)
3.0 V LVTTTL/3.0 V CMOS	16, 12 , 8, 4	16, 12 , 8, 4
2.5 V LVCMOS	16, 12 , 8, 4	16, 12 , 8, 4
1.8 V LVCMOS	12 , 10, 8, 6, 4, 2	12 , 10, 8, 6, 4, 2
1.5 V LVCMOS	12 , 10, 8, 6, 4, 2	12 , 10, 8, 6, 4, 2
1.2 V LVCMOS	8 , 6, 4, 2	—
SSTL-18 Class I	12, 10, 8 , 6, 4	12, 10, 8 , 6, 4
SSTL-18 Class II	16	8, 16
SSTL-15 Class I	12, 10, 8 , 6, 4	12, 10, 8 , 6, 4
SSTL-15 Class II	16	8, 16
SSTL-135 Class I	12, 10, 8 , 6, 4	—
SSTL-135 Class II	16	—
SSTL-125 Class I	12, 10, 8 , 6, 4	—
SSTL-125 Class II	16	—
SSTL-12 Class I	12, 10, 8 , 6, 4	—
SSTL-12 Class II	16	—
POD12	16, 12, 10, 8 , 6, 4	—
1.8 V HSTL Class I	12, 10, 8 , 6, 4	12, 10, 8 , 6, 4
1.8 V HSTL Class II	16	16
1.5 V HSTL Class I	12, 10, 8 , 6, 4	12, 10, 8 , 6, 4
1.5 V HSTL Class II	16	16
1.2 V HSTL Class I	12, 10, 8 , 6, 4	—
1.2 V HSTL Class II	16	—
SSTL-135 Class I	12, 10, 8 , 6, 4	—
差分 SSTL-135 Class II	16	—
差分 SSTL-125 Class I	12, 10, 8 , 6, 4	—
差分 SSTL-125 Class II	16	—
差分 SSTL-12 Class I	12, 10, 8 , 6, 4	—
差分 SSTL-12 Class II	16	—
差分 POD12	16, 12, 10, 8 , 6, 4	—

注意: Intel 建议通过执行 IBIS 或者 SPICE 仿真来确定用于特定应用的最佳电流强度设置。

(7) HPS 的可编程电流强度信息是初步信息。



5.5.3.2. 可编程的输出摆率控制

每个普通和双功能 I/O 管脚的输出缓冲中可编程的输出摆率控制可进行如下配置：

- 快速摆率 — 对高性能系统提供高速跳变。
- 慢速摆率 — 有助于降低系统的噪声和串扰，但会在上升和下降沿上添加微小的延迟。

由于每个 I/O 管脚包含一个摆率控制，因此您可以逐一对管脚指定摆率。

注意： Intel 建议通过执行 IBIS 或者 SPICE 仿真来确定用于特定应用的最佳摆率设置。

5.5.3.3. 可编程 IOE 延迟

通过启用可编程 IOE 延迟来确保零保持时间，最小化建立时间，或者增加时钟到输出 (clock-to-output) 时间。这有助于增加读写时序裕量，因为它最小化了总线中信号之间的不确定性。

每个管脚从管脚到输入 (pin-to-input) 寄存器都有一个不同的输入延时，或者从寄存器到输出 (register-to-output) 管脚都有一个不同的输出延时，来保证进出器件的一组总线中的信号具有相同的延时。

- 在输出和 OE 路径中，存在 50 ps 延时和最大 800 ps 的输出和 OE 延时。
- 在输入路径中，有两个 50 ps 增量延时和最大 3.2 ns 的输入延时。

要了解关于可编程 IOE 延迟规范的详细信息，请参阅器件数据表。

相关链接

[可编程 IOE 延迟](#)

5.5.3.4. 可编程开漏输出

当 logic-to-pin 处于高电平时，开漏输出对输出提供高阻抗状态。如果 logic-to-pin 处于低电平，输出则为低阻抗状态。

可以附加几个开漏输出到线上。这种连接方式与一个逻辑 OR 功能相似，通常被称为 active-low wired-OR 电路。如果至少一个输出处于逻辑 0 状态 (active)，那么电路会吸收电流并将电线带至低电平。

如果连接多个器件至一个总线，就可以使用开漏输出。例如，可以将开漏输出用于系统级控制信号，该系统级控制信号可以被任何器件置位或者作为一个中断信号。

可以使用下面的其中一种方法使能开漏输出分配：

- 使用 OPNDRN 原语设计三态缓冲器。
- 打开 Quartus Prime 软件中的 **Auto Open-Drain Pins** 选项。

虽然没有使能这一选项分配也可以设计开漏输出，但是也将无法使用 I/O 缓冲器的开漏输出功能。I/O 缓冲器中的开漏输出功能提供了 OE 到输出的最佳传播延时。

5.5.3.5. 可编程预加重

V_{OD} 设置和驱动器输出阻抗对高速传输信号的输出电流限制进行设置。在高频率时，在下一个沿到达前，摆率或许不够快达到 V_{OD} 电平，从而产生一个固定模式抖动 (pattern-dependent jitter)。通过预加重，信号变化时，输出电流能迅速提升，从而增大输出摆率。

预加重提升输出信号高频分量的振幅，从而有助于补偿传输线上的频率相关衰减。与信号反射导致的过冲不同，由额外电流导致的过冲仅发生在状态发生变化切换期间来增大输出摆率，并且没有振铃。所需的预加重数量取决于传输线上的高频分量衰减。

图 82. 可编程预加重

该图显示了带预加重的 LVDS 输出。

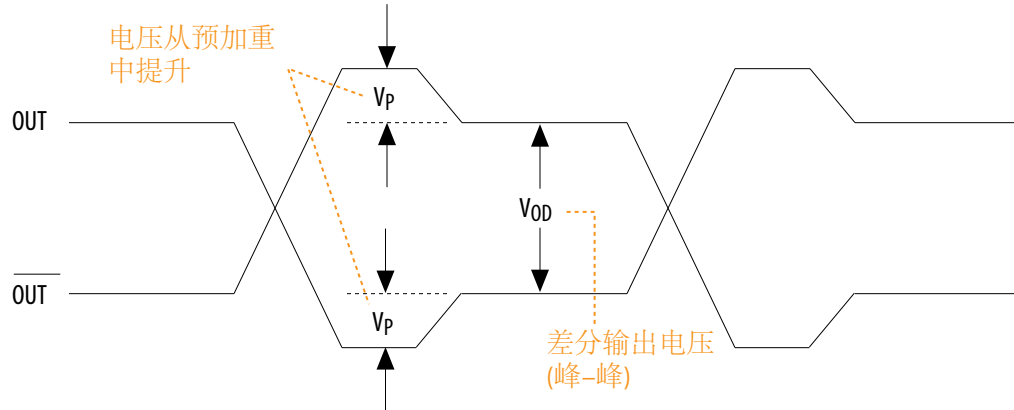


表 51. Quartus Prime 软件 Assignment Editor — 可编程预加重

该表列出了 Quartus Prime 软件 Assignment Editor 中可编程预加重的约束名及其可能的值。

域	约束
To	tx_out
Assignment name	可编程预加重
Allowed values	0 (禁用), 1 (使能), 默认为 1。

5.5.3.6. 可编程差分输出电压

可编程 V_{OD} 设置使您能够调节输出眼高，以优化走线长度及功耗。较高的 V_{OD} 摆动可提高接收器端的电压容限，而较小的 V_{OD} 摆动可降低功耗。通过修改 Quartus Prime 软件 Assignment Editor 中的 V_{OD} 设置，能够静态地调节差分信号的 V_{OD} 。

图 83. 差分 V_{OD}

该图显示了差分 LVDS 输出的 V_{OD} 。

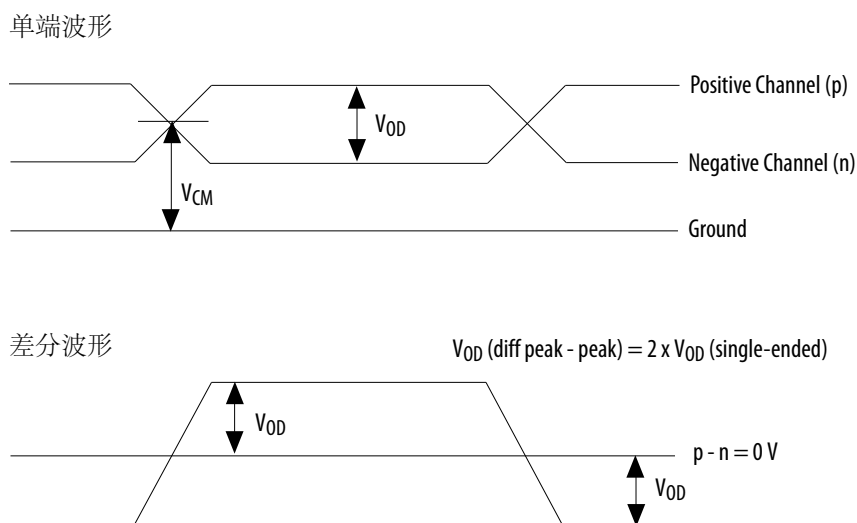


表 52. Quartus Prime 软件的 Assignment Editor — 可编程 V_{OD}

该表列出了 Quartus Prime 软件 Assignment Editor 中可编程 V_{OD} 的约束名以及可能的值。"0"仅适用于 RSDS 和 mini-LVDS I/O 标准，不适用于 LVDS I/O 标准。

域	约束
To	tx_out
Assignment name	可编程差分输出电压(V_{OD})
Allowed values	0 (低), 1 (中低), 2 (中高), 3 (高)。默认为 2。

5.5.4. Arria 10 器件的片上 I/O 匹配

串行(R_S)和并行(R_T) OCT 提供了 I/O 阻抗匹配和匹配性能。OCT 维持信号质量，节省电路板空间，并降低外部组件成本。

Arria 10 器件支持所有 FPGA 和 HPS I/O bank 中的 OCT。对于 3 V 和 HPS I/O，I/O 仅支持不带校准的 OCT。

图 84. 单端匹配(R_S 和 R_T)

下图显示了 Arria 10 器件所支持的单端匹配方案。 R_{T1} 和 R_{T2} 动态地进行并行匹配，并且仅在器件接收时被使能。在双向应用中， R_{T1} 和 R_{T2} 在器件接收时自动打开，并且在器件驱动时自动关闭。

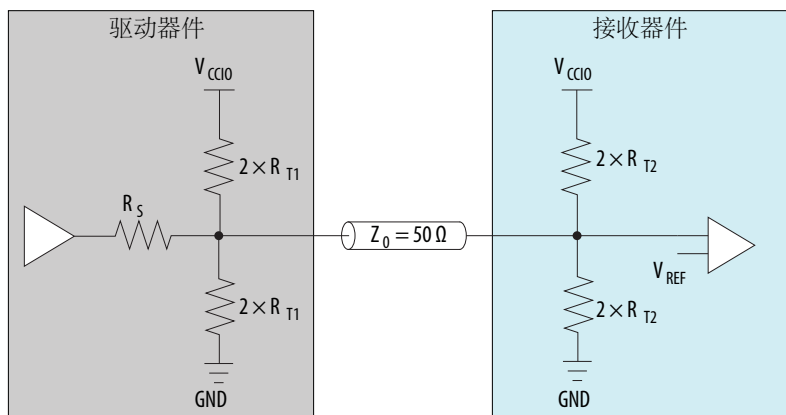


表 53. Arria 10 器件中支持的 OCT 方案

方向	OCT 方案	I/O 类型支持		
		LVDS I/O	3 V I/O	HPS I/O
输出	带校准的 R_S OCT	Yes	—	—
	无校准的 R_S OCT	Yes	Yes	Yes
输入	带校准的 R_T OCT	Yes	—	—
	R_D OCT (仅适用于 LVDS I/O 标准)	Yes	—	—
双向	动态 R_S OCT 和 R_T OCT	Yes	Yes	Yes

相关链接

- [Altera OCT IP 内核用户指南](#)
- [Arria 10 器件中不带校准的 \$R_S\$ OCT](#) (第 124 页)
- [Arria 10 器件中带校准的 \$R_S\$ OCT](#) (第 126 页)
- [Arria 10 器件中带校准的 \$R_T\$ OCT](#) (第 128 页)
- [动态 OCT](#) (第 130 页)
- [差分输入 \(\$R_D\$ OCT\)](#) (第 131 页)
- [Arria 10 器件中的 OCT 校准模块](#) (第 132 页)

5.5.4.1. Arria 10 器件中不带校准的 R_S OCT

Arria 10 器件支持单端和电压参考 I/O 标准中的 R_S OCT。不带校准的 R_S OCT 仅支持输出。



表 54. 不带校准的 R_S OCT 的可选 I/O 标准

该表列出了差分 I/O 标准上未校准 OCT 的输出匹配设置。

I/O 标准	器件类型支持	未校准的 OCT (输出)
		R_S (Ω)
3.0 V LVTTTL/3.0 V LVCMOS	GX, SX	25/50
2.5 V LVCMOS	GX, SX	25/50
1.8 V LVCMOS	全部	25/50
1.5 V LVCMOS	全部	25/50
1.2 V LVCMOS	全部	25/50
SSTL-18 Class I	全部	50
SSTL-18 Class II	全部	25
SSTL-15 Class I	全部	50
SSTL-15 Class II	全部	25
SSTL-15	全部	34、40
SSTL-135	全部	34、40
SSTL-125	全部	34、40
SSTL-12	全部	40, 60, 120, 240
POD12	全部	34、40、48、60
1.8 V HSTL Class I	全部	50
1.8 V HSTL Class II	全部	25
1.5 V HSTL Class I	全部	50
1.5 V HSTL Class II	全部	25
1.2 V HSTL Class I	全部	50
1.2 V HSTL Class II	全部	25
HSUL-12	全部	34.3, 40, 48, 60, 80
差分 SSTL-18 Class I	全部	50
差分 SSTL-18 Class II	全部	25
差分 SSTL-15 Class I	全部	50
差分 SSTL-15 Class II	全部	25
差分 SSTL-15	全部	34、40
差分 SSTL-135	全部	34、40
差分 SSTL-125	全部	34、40
差分 SSTL-12	全部	40, 60, 120, 240
差分 POD12	全部	34、40、48、60
差分 1.8 V HSTL Class I	全部	50
差分 1.8 V HSTL Class II	全部	25

继续...

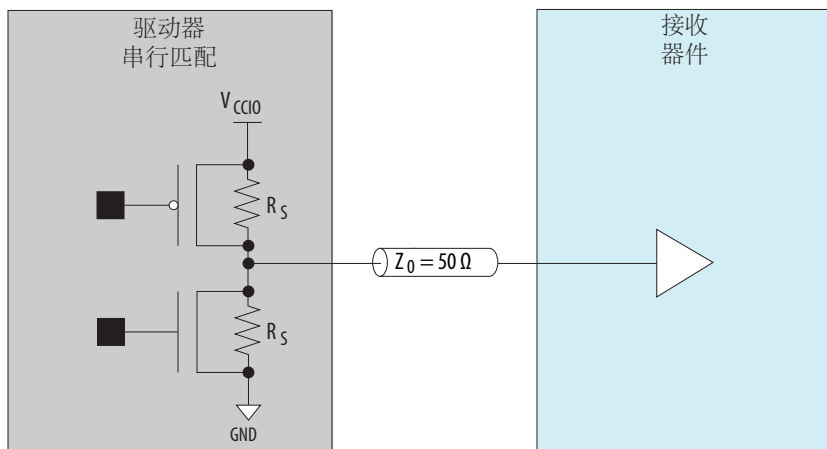
I/O 标准	器件类型支持	未校准的 OCT (输出)
		R_S (Ω)
差分 1.5 V HSTL Class I	全部	50
差分 1.5 V HSTL Class II	全部	25
差分 1.2 V HSTL Class I	全部	50
差分 1.2 V HSTL Class II	全部	25
差分 HSUL-12	全部	34.3, 40, 48, 60, 80

驱动器阻抗匹配对 I/O 驱动器提供受控输出阻抗，其高度匹配传输线路阻抗。这样，极大地减少 PCB 走线上的信号反射。

如果选择匹配阻抗，就不能选择电流强度。

图 85. 不带校准的 R_S OCT

该图显示了 R_S 作为输出晶体管固有的阻抗。



相关链接

Arria 10 器件的片上 I/O 匹配 (第 123 页)

5.5.4.2. Arria 10 器件中带校准的 R_S OCT

Arria 10 器件在所有 LVDS I/O bank 中支持带校准 R_S OCT。

表 55. 带校准的 R_S OCT 的可选 I/O 标准

该表列出了差分 I/O 标准上校准 OCT 的输出匹配设置。

I/O 标准	器件类型支持	校准的 OCT (输出)	
		R_S (Ω)	RZQ (Ω)
1.8 V LVCMOS	全部	25, 50	100
1.5 V LVCMOS	全部	25, 50	100
1.2 V LVCMOS	全部	25, 50	100
继续...			



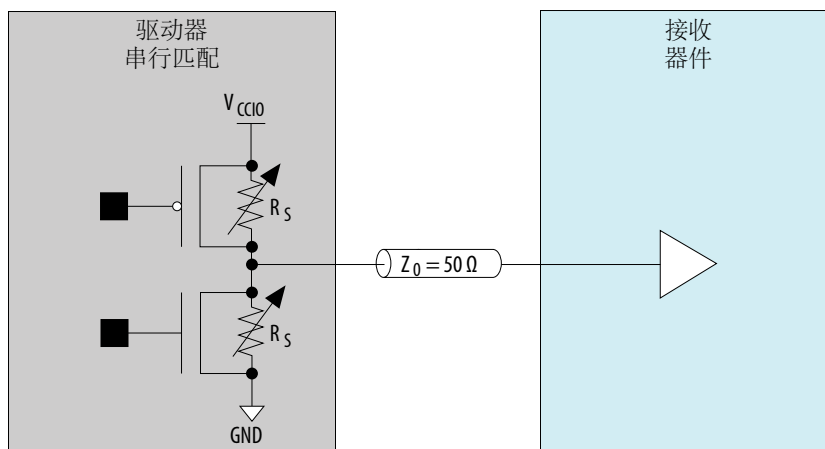
I/O 标准	器件类型支持	校准的 OCT (输出)	
		R _S (Ω)	RZQ (Ω)
SSTL-18 Class I	全部	50	100
SSTL-18 Class II	全部	25	100
SSTL-15 Class I	全部	50	100
SSTL-15 Class II	全部	25	100
SSTL-15	全部	25, 50	100
		34, 40	240
SSTL-135	全部	34, 40	240
SSTL-125	全部	34, 40	240
SSTL-12	全部	40, 60, 120, 240	240
POD12	全部	34, 40, 48, 60	240
1.8 V HSTL Class I	全部	50	100
1.8 V HSTL Class II	全部	25	100
1.5 V HSTL Class I	全部	50	100
1.5 V HSTL Class II	全部	25	100
1.2 V HSTL Class I	全部	50	100
1.2 V HSTL Class II	全部	25	100
HSUL-12	全部	34, 40, 48, 60, 80	240
差分 SSTL-18 Class I	全部	50	100
差分 SSTL-18 Class II	全部	25	100
差分 SSTL-15 Class I	全部	50	100
差分 SSTL-15 Class II	全部	25	100
差分 SSTL-15	全部	25, 50	100
		34, 40	240
差分 SSTL-135	全部	34, 40	240
差分 SSTL-125	全部	34, 40	240
差分 SSTL-12	全部	40, 60, 120, 240	240
差分 POD12	全部	34, 40, 48, 60	240
差分 1.8 V HSTL Class I	全部	50	100
差分 1.8 V HSTL Class II	全部	25	100
差分 1.5 V HSTL Class I	全部	50	100
差分 1.5 V HSTL Class II	全部	25	100
差分 1.2 V HSTL Class I	全部	50	100
差分 1.2 V HSTL Class II	全部	25	100
差分 HSUL-12	全部	34, 40, 48, 60, 80	240

R_S OCT 校准电路将 I/O 缓冲器的总阻抗与连接到 RZQ 管脚的外部参考电阻进行比较，并且动态地使能或者禁用晶体管直到它们匹配。

校准发生在器件配置的最后阶段。当校准电路找到正确的阻抗时，它会掉电并停止更改驱动器的特性。

图 86. 带校准的 R_S OCT

该图显示了 R_S 作为输出晶体管固有的阻抗。



相关链接

Arria 10 器件的片上 I/O 匹配 (第 123 页)

5.5.4.3. Arria 10 器件中带校准的 R_T OCT

Arria 10 器件支持所有 LVDS I/O bank(而不是 3 V I/O bank)中带校准的 R_T OCT。带校准的 R_T OCT 仅适用于输入和双向管脚的配置。输出管脚配置不支持带校准的 R_T OCT。如果使用 R_T OCT，那么 bank 上的 V_{CCIO} 必须与使能 R_T OCT 的管脚的 I/O 标准相匹配。

表 56. 带校准的 R_T OCT 的可选 I/O 标准

该表列出了差分 I/O 标准上校准 OCT 的输入匹配设置。

I/O 标准	器件类型支持	校准的 OCT (输入)	
		R_T (Ω)	RZQ (Ω)
SSTL-18 Class I	全部	50	100
SSTL-18 Class II	全部	50	100
SSTL-15 Class I	全部	50	100
SSTL-15 Class II	全部	50	100
SSTL-15	全部	30, 40, 60, 120	240
SSTL-135	全部	30, 40, 60, 120	240
SSTL-125	全部	30, 40, 60, 120	240
SSTL-12	全部	60, 120	240
POD12	全部	34, 40, 48, 60, 80, 120, 240	240

继续...

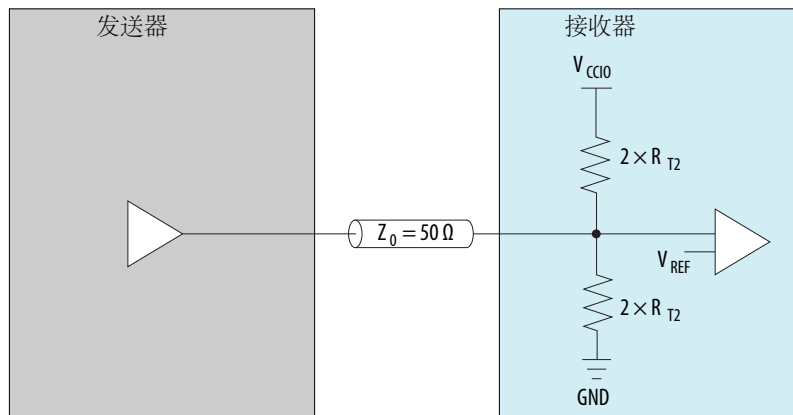


I/O 标准	器件类型支持	校准的 OCT (输入)	
		$R_T (\Omega)$	$R_{ZQ} (\Omega)$
1.8 V HSTL Class I	全部	50	100
1.8 V HSTL Class II	全部	50	100
1.5 V HSTL Class I	全部	50	100
1.5 V HSTL Class II	全部	50	100
1.2 V HSTL Class I	全部	50	100
1.2 V HSTL Class II	全部	50	100
差分 SSTL-18 Class I	全部	50	100
差分 SSTL-18 Class II	全部	50	100
差分 SSTL-15 Class I	全部	50	100
差分 SSTL-15 Class II	全部	50	100
差分 SSTL-15	全部	30, 40, 60, 120	240
差分 SSTL-135	全部	30, 40, 60, 120	240
差分 SSTL-125	全部	30, 40, 60, 120	240
差分 SSTL-12	全部	60, 120	240
差分 POD12	全部	34, 40, 48, 60, 80, 120, 240	240
差分 1.8 V HSTL Class I	全部	50	100
差分 1.8 V HSTL Class II	全部	50	100
差分 1.5 V HSTL Class I	全部	50	100
差分 1.5 V HSTL Class II	全部	50	100
差分 1.2 V HSTL Class I	全部	50	100
差分 1.2 V HSTL Class II	全部	50	100

R_T OCT 校准电路将 I/O 缓冲器的总阻抗与连接到 R_{ZQ} 管脚的外部电阻进行比较。该电路动态地使能或者禁用晶体管直到 I/O 缓冲器的总阻抗与外部电阻相匹配。

校准产生在器件配置的最后阶段。当校准电路找到正确的阻抗时，它会掉电并停止更改驱动器的特性。

图 87. 带校准的 R_T OCT



相关链接

Arria 10 器件的片上 I/O 匹配 (第 123 页)

5.5.4.4. 动态 OCT

根据数据方向对信号完整性进行优化，动态 OCT 对于匹配高性能双向路径是非常有用的。动态 OCT 也有助于节省功耗，因为器件匹配是仅在输入操作期间开启的内部 — 匹配，从而使用更少的静态功耗。

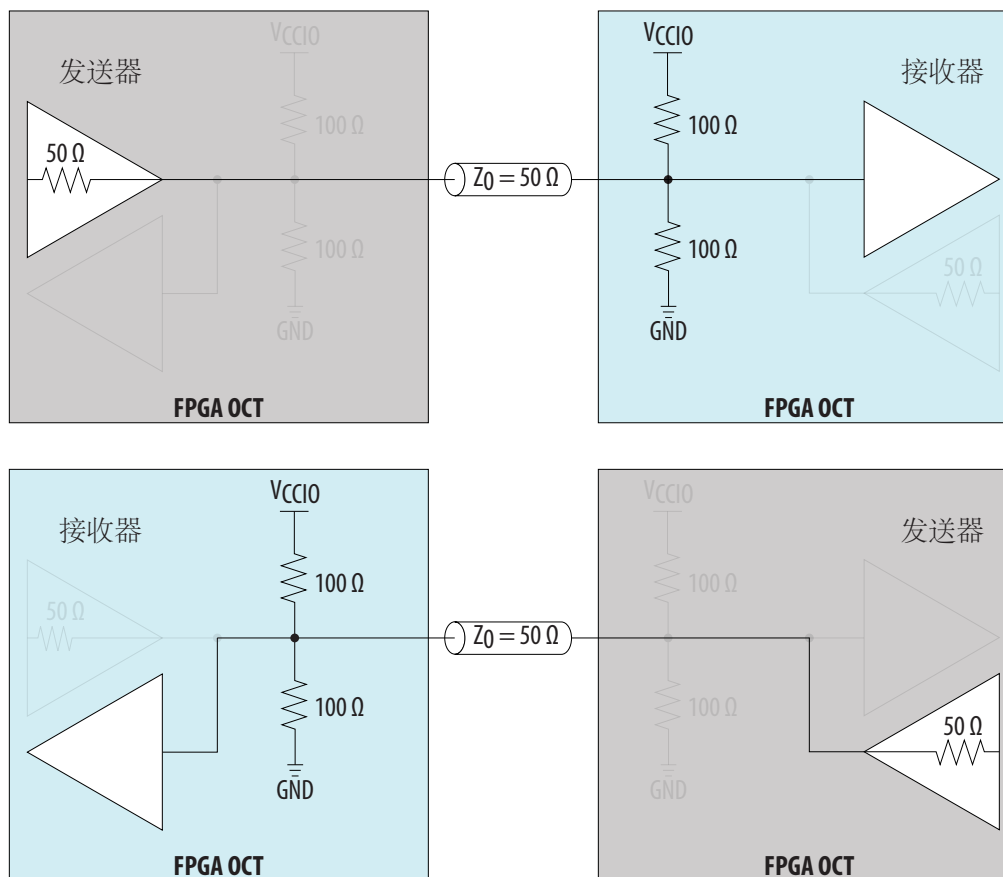
注意: 如果将 HSUL-12、SSTL-12、SSTL-15、SSTL-135 以及 SSTL-125 I/O 标准和 DDR3 存储器接口一起使用，Intel 建议将 OCT 和这些 I/O 标准一起使用来节省电路板空间和成本。OCT 减少了外部匹配电阻使用的数量。

表 57. 基于双向 I/O 的动态 OCT

使能或禁用动态 R_T OCT 或者 R_S OCT 是基于双向 I/O 用作接收器或是驱动器而定的。

动态 OCT	双向 I/O	状态
动态 R_T OCT	作为接收器使用	使能
	作为驱动器使用	禁用
动态 R_S OCT	作为接收器使用	禁用
	作为驱动器使用	使能

图 88. Arria 10 器件中的动态 R_T OCT



相关链接

Arria 10 器件的片上 I/O 匹配 (第 123 页)

5.5.4.5. 差分输入 (R_D OCT)

Arria 10 器件中的所有 I/O 管脚和专用时钟输入管脚都支持片上差分匹配, R_D OCT。Arria 10 器件在每个差分接收通道上均提供一个 100 Ω 片上差分匹配选项以用于 LVDS 标准。

您可以在 Quartus Prime 软件的 Assignment Editor 中使能片上匹配。

图 89. 片上差分 I/O 匹配

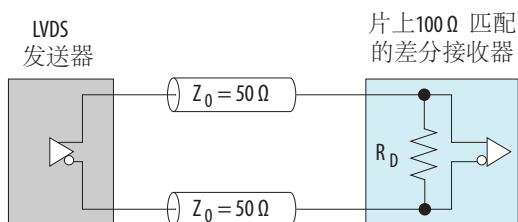


表 58. Quartus Prime 软件的 Assignment Editor — 片上差分匹配

该表列出 Quartus Prime 软件 Assignment Editor 中片上差分匹配的约束名称。

域	约束
To	rx_in
Assignment name	Input Termination
Value	Differential

相关链接

[Arria 10 器件的片上 I/O 匹配 \(第 123 页\)](#)

5.5.4.6. Arria 10 器件中的 OCT 校准模块

可使用相同 I/O 列中的任何 I/O bank 的 OCT 校准模块校准 OCT。包含 OCT 校准模块的 I/O bank 必须具有与 I/O bank 的 OCT 相同的 V_{CCIO} 。

如果 I/O 标准使用相同的 V_{CCIO} 电源电压，则可将同一 I/O bank 中的 R_S 和 R_T OCT 用于不同的 I/O 标准。您不能对同一 I/O 缓冲器进行 R_S OCT 和可编程电流强度配置。

OCT 校准处理使用给定 I/O bank 中每个校准模块的可用 RZQ 管脚进行串行和并行校准匹配：

- RZQ 管脚中每个 OCT 校准模块含有一个相关联的 240 Ω 外部基准电阻。
- 通过外部 100 Ω 或 240 Ω 电阻器（取决于 R_S 或 R_T OCT 值）将 RZQ 管脚连接到 GND。
- RZQ 管脚与其所在的 I/O bank 共享相同 V_{CCIO} 电源电压。
- 如果您不使用校准电路，则 RZQ 管脚为两用 I/O 管脚，并用作通用 I/O 管脚。

除专用配置管脚以外，Arria 10 器件支持所有 LVDS I/O 管脚上的已校准 R_S 和已校准 R_T OCT。

相关链接

- [Altera OCT IP 内核用户指南](#)
- [Arria 10 器件的片上 I/O 匹配 \(第 123 页\)](#)

5.5.5. Arria 10 器件的外部 I/O 匹配

表 59. 不同 I/O 标准的外部匹配方案

I/O 标准	外部匹配方案
2.5 V LVCMOS	不要求外部匹配
1.8 V LVCMOS	
1.5 V LVCMOS	
1.2 V LVCMOS	
SSTL-18 Class I	单端 SSTL I/O 标准匹配
SSTL-18 Class II	
SSTL-15 Class I	
SSTL-15 Class II	
继续...	



I/O 标准	外部匹配方案
SSTL-15 ⁽⁸⁾	不要求外部匹配
SSTL-135 ⁽⁸⁾	
SSTL-125 ⁽⁸⁾	
SSTL-12 ⁽⁸⁾	
POD12	单端 POD I/O 标准匹配
差分 SSTL-18 Class I	差分 SSTL I/O 标准匹配
差分 SSTL-18 Class II	
差分 SSTL-15 Class I	
差分 SSTL-15 Class II	
差分 SSTL-15 ⁽⁸⁾	不要求外部匹配
差分 SSTL-135 ⁽⁸⁾	
差分 SSTL-125 ⁽⁸⁾	
差分 SSTL-12 ⁽⁸⁾	
差分 POD12	差分 POD I/O 标准匹配
1.8 V HSTL Class I	单端 HSTL I/O 标准匹配
1.8 V HSTL Class II	
1.5 V HSTL Class I	
1.5 V HSTL Class II	
1.2 V HSTL Class I	
1.2 V HSTL Class II	
HSUL-12	不要求外部匹配
差分 1.8 V HSTL Class I	差分 HSTL I/O 标准匹配
差分 1.8 V HSTL Class II	
差分 1.5 V HSTL Class I	
差分 1.5 V HSTL Class II	
差分 1.2 V HSTL Class I	
差分 1.2 V HSTL Class II	
差分 HSUL-12	不要求外部匹配
LVDS	LVDS I/O 标准匹配
RSDS	RSDS/mini-LVDS I/O 标准匹配
Mini-LVDS	
LVPECL	差分 LVPECL I/O 标准匹配

注意: Intel 建议通过执行 IBIS 或者 SPICE 仿真来确定用于特定应用的最佳匹配方案。

⁽⁸⁾ Intel 建议将 OCT 和这些 I/O 标准一起使用来节省电路板空间和成本。OCT 减少了外部匹配电阻使用的数量。

5.5.5.1. 单端 I/O 匹配

基准电压 I/O 标准需要一个输入 V_{REF} 以及一个匹配电压 (V_{TT})。接收器件的基准电压追踪发送器件的匹配电压。

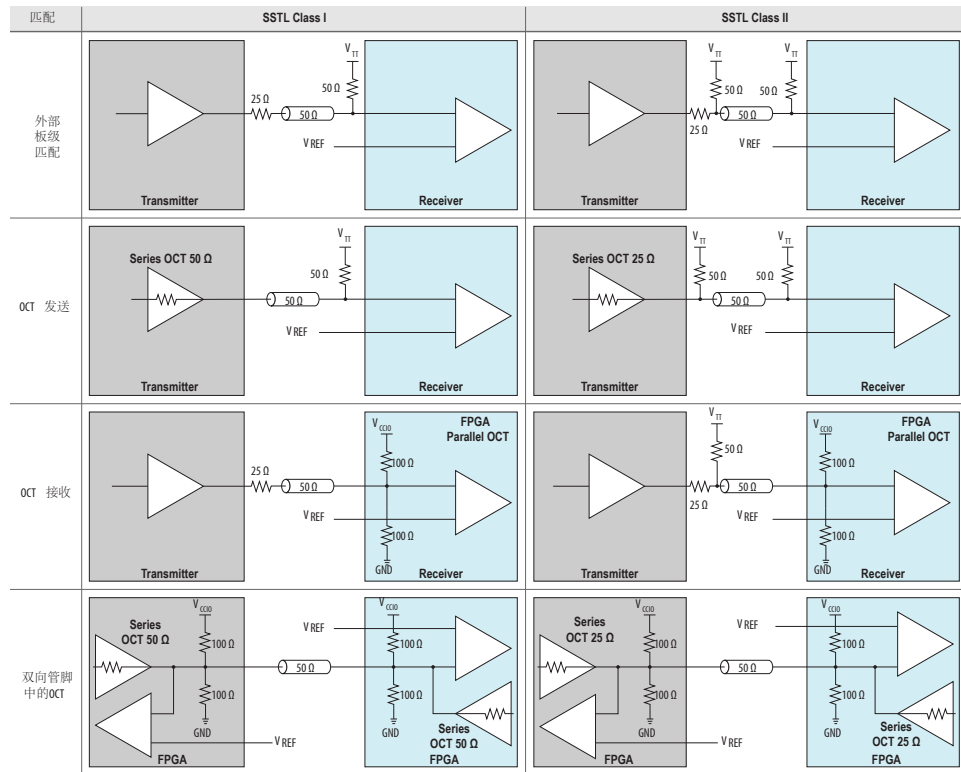
所支持的 I/O 标准，例如 SSTL-12、SSTL-125、SSTL-135 和 SSTL-15，通常不需要外部电路板匹配。

Intel 建议将 OCT 和这些 I/O 标准一起使用来节省电路板空间和成本。OCT 减少了外部匹配电阻使用的数量。

注意: 不能同时使用 R_S 和 R_T OCT。请参阅相关信息了解更多信息。

图 90. SSTL I/O 标准匹配

该图显示 Arria 10 器件上 SSTL I/O 匹配的详细内容。



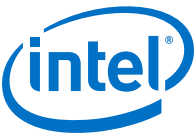


图 91. HSTL I/O 标准匹配

该图显示 Arria 10 器件上 HSTL I/O 匹配的详细内容。

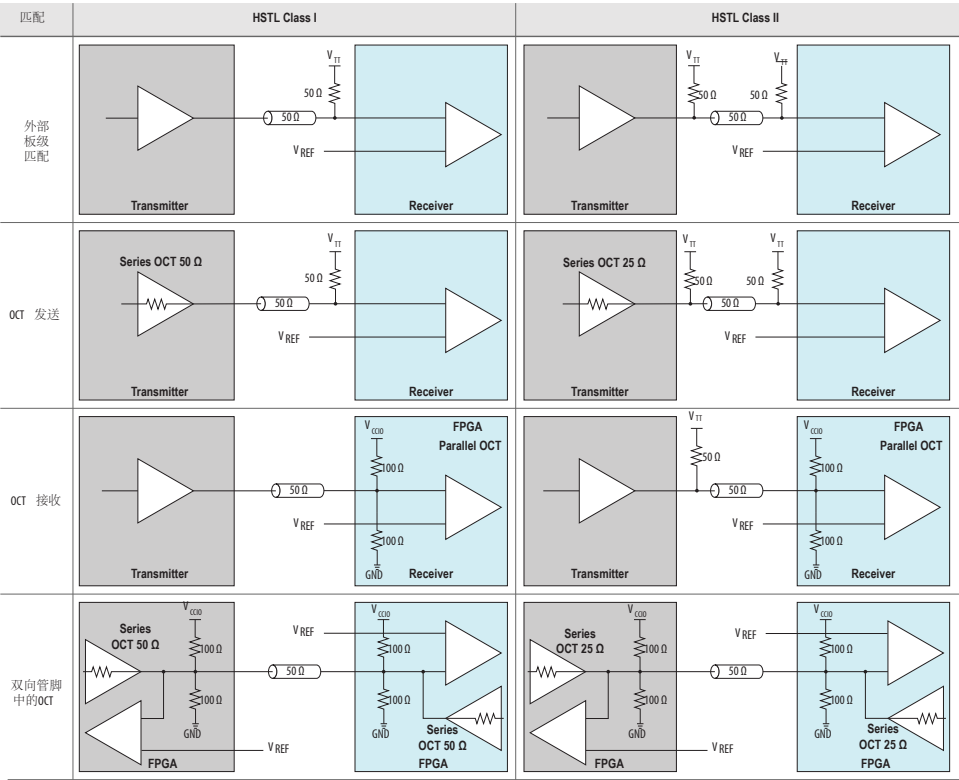
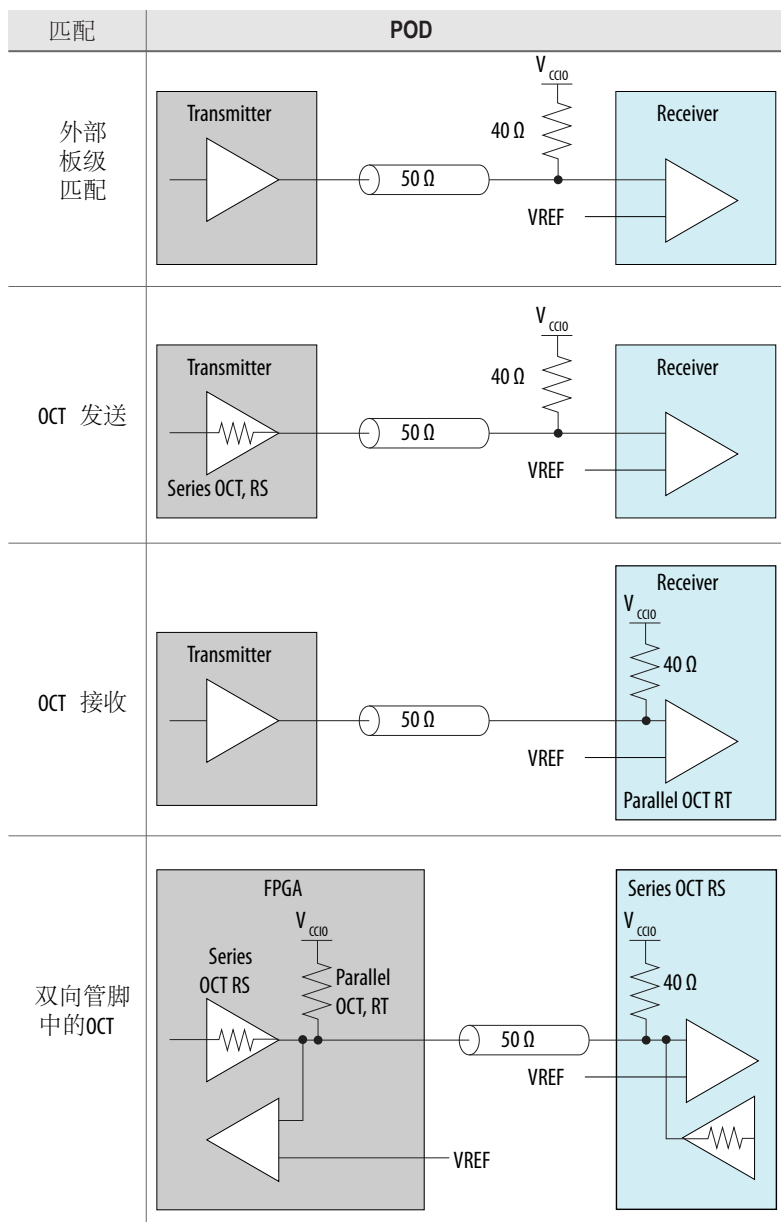


图 92. POD I/O 标准匹配

该图显示 Arria 10 器件上 POD I/O 匹配的详细内容。



相关链接

动态 OCT (第 130 页)

5.5.5.2. Arria 10 器件的差分 I/O 匹配

I/O 管脚成对排列以支持差分标准。每个 I/O 管脚对可支持差分输入和输出缓冲。

支持的 I/O 标准，例如 Differential SSTL-12、Differential SSTL-15、Differential SSTL-125 和 Differential SSTL-135，通常不需要外部电路板匹配。

Intel 建议将 OCT 和这些 I/O 标准一起使用来节省电路板空间和成本。OCT 减少了外部匹配电阻使用的数量。

相关链接

- [差分 HSTL、SSTL、HSUL 和 POD 匹配 \(第 137 页\)](#)
- [LVDS、RSDS 和 Mini-LVDS 匹配 \(第 139 页\)](#)
- [LVPECL 匹配 \(第 139 页\)](#)

5.5.5.2.1. 差分 HSTL、SSTL、HSUL 和 POD 匹配

差分 HSTL、SSTL、HSUL 和 POD 输入使用 LVDS 差分输入缓冲。不过 R_D 支持仅在 I/O 标准是 LVDS 时可用。

差分 HSTL、SSTL、HSUL 和 POD 输入不是真差分输出。这些 I/O 标准使用两个单端输出，其中第二个输出编程为反转输出。

图 93. 差分 SSTL I/O 标准匹配

该图显示了 Arria 10 器件上差分 SSTL I/O 标准匹配的详细内容。

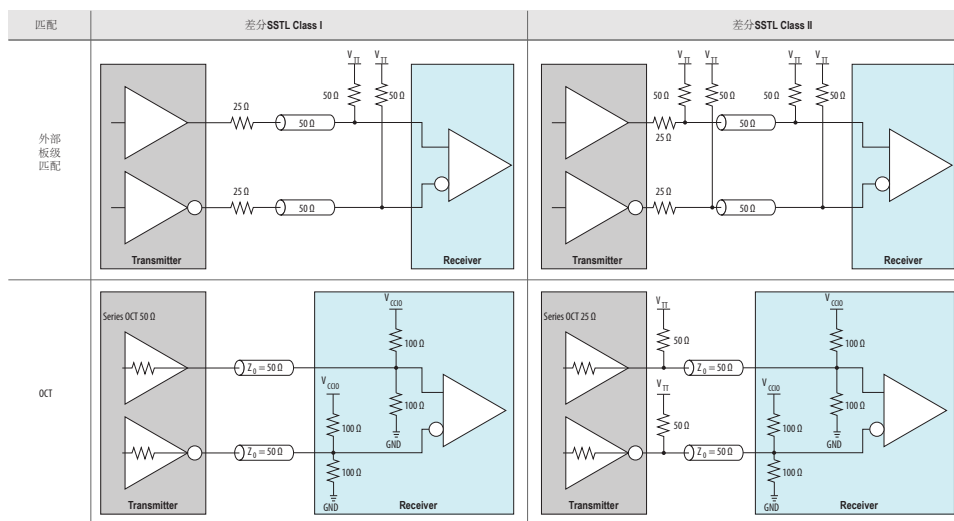


图 94. 差分 HSTL I/O 标准匹配

该图显示了 Arria 10 器件上差分 HSTL I/O 标准匹配的详细内容。

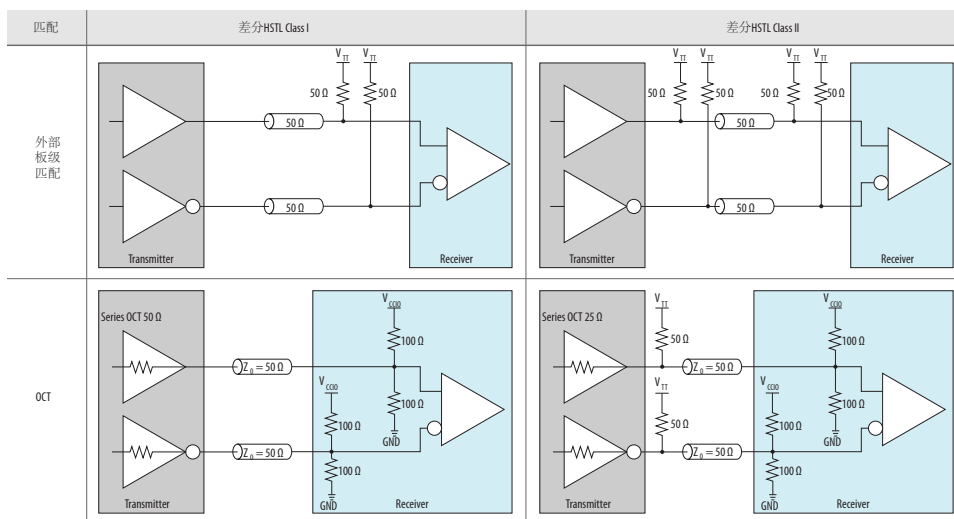
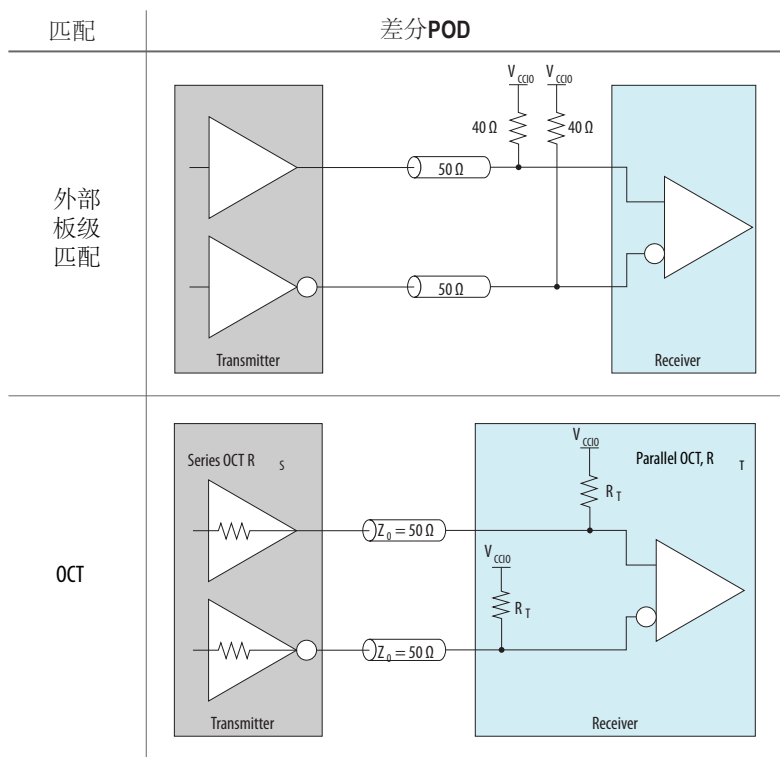


图 95. 差分 POD I/O 标准匹配

该图显示了 Arria 10 器件上差分 POD I/O 标准匹配的详细内容。



相关链接

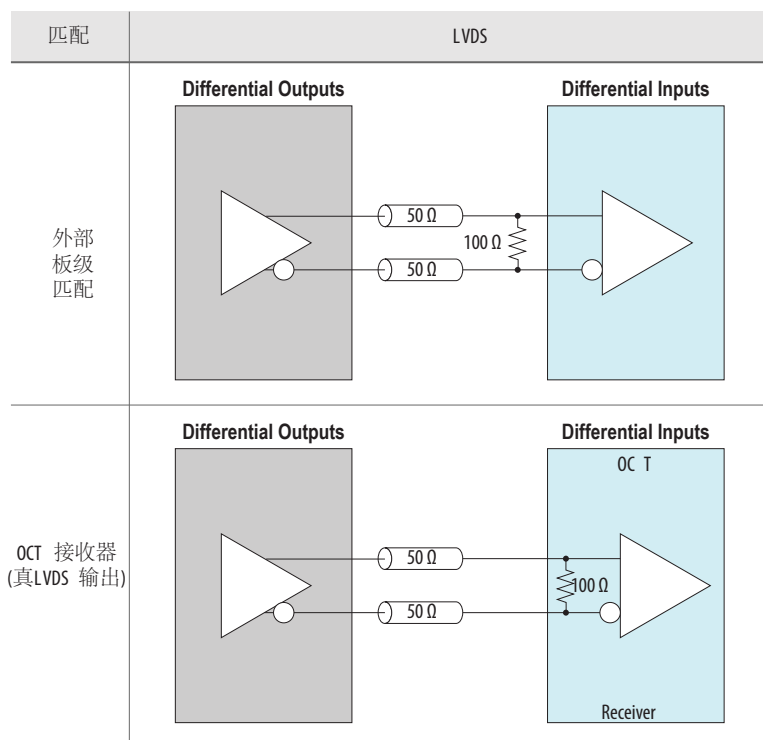
Arria 10 器件的差分 I/O 匹配 (第 136 页)

5.5.5.2.2. LVDS、RSDS 和 Mini-LVDS 匹配

所有 I/O bank 都具有支持真 LVDS、RSDS 和 mini-LVDS I/O 标准的专用电路，通过使用真 LVDS 输出缓冲器而非电阻器网络。

图 96. LVDS I/O 标准匹配

该图显示 LVDS I/O 标准匹配。片上差分电阻在所有 I/O bank 中可用。



相关链接

- 差分 I/O 标准规范
- 国家半导体公司 (www.national.com)
要了解关于 RSDS I/O 标准的详细信息，请参考国家半导体公司网页上的 *RSDS 规范* 部分。
- Arria 10 器件的差分 I/O 匹配 (第 136 页)

5.5.5.2.3. LVPECL 匹配

Arria 10 器件仅在输入时钟管脚中支持 LVPECL I/O 标准：

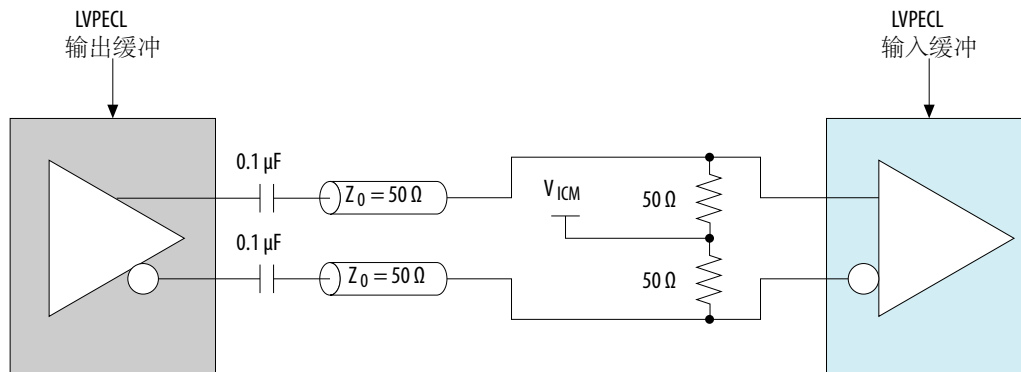
- 使用 LVDS 输入缓冲器支持 LVPECL 输入操作。
- 不支持 LVPECL 输出操作。

如果输出缓冲器的 LVPECL 共模电压与 LVPECL 输入共模电压不匹配，就使用交流(AC)耦合。

注意：

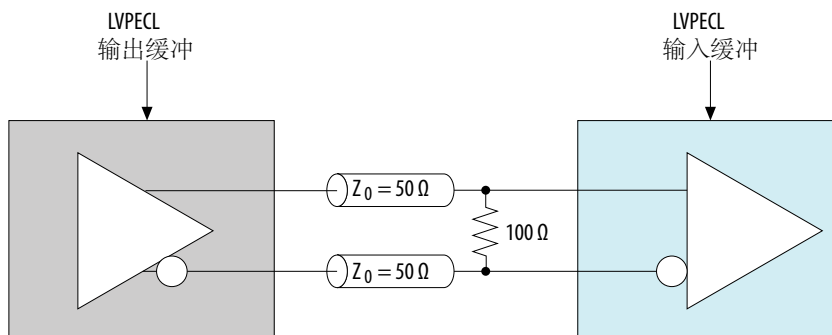
Intel 建议使用 IBIS 模式来验证 LVPECL AC/DC 耦合匹配。

图 97. LVPECL 交流耦合匹配



如果 LVPECL 输出共模电压在 Arria 10 LVPECL 输入缓冲器规格范围之内，那么直流(DC)耦合 LVPECL 可用。

图 98. LVPECL 直流耦合匹配



有关 V_{ICM} 规范的信息，请参考器件数据表。

相关链接

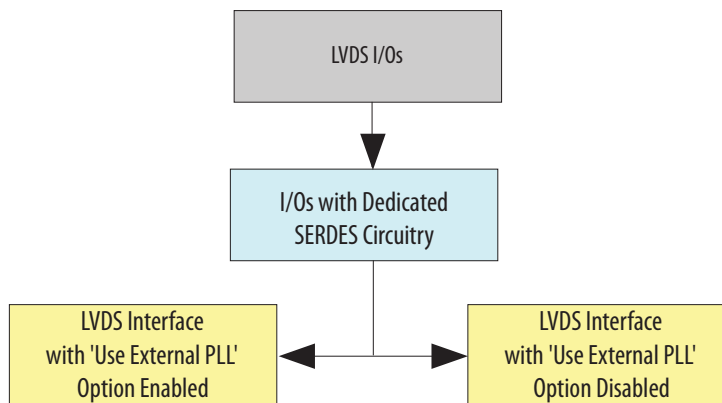
- [差分 I/O 标准规范](#)
- [Arria 10 器件的差分 I/O 匹配](#) (第 136 页)

5.6. Arria 10 器件的高速源同步 SERDES 和 DPA

Arria 10 器件中的高速差分 I/O 接口和 DPA 功能对单端 I/O 提供优势，并在实现总体系统带宽中发挥了作用。Arria 10 器件支持 LVDS、mini-LVDS 和 RSDS 差分 I/O 标准。

图 99. 高速差分 I/O 支持的 I/O Bank

下图显示了 Arria 10 器件中高速差分 I/O 支持的 I/O bank。



相关链接

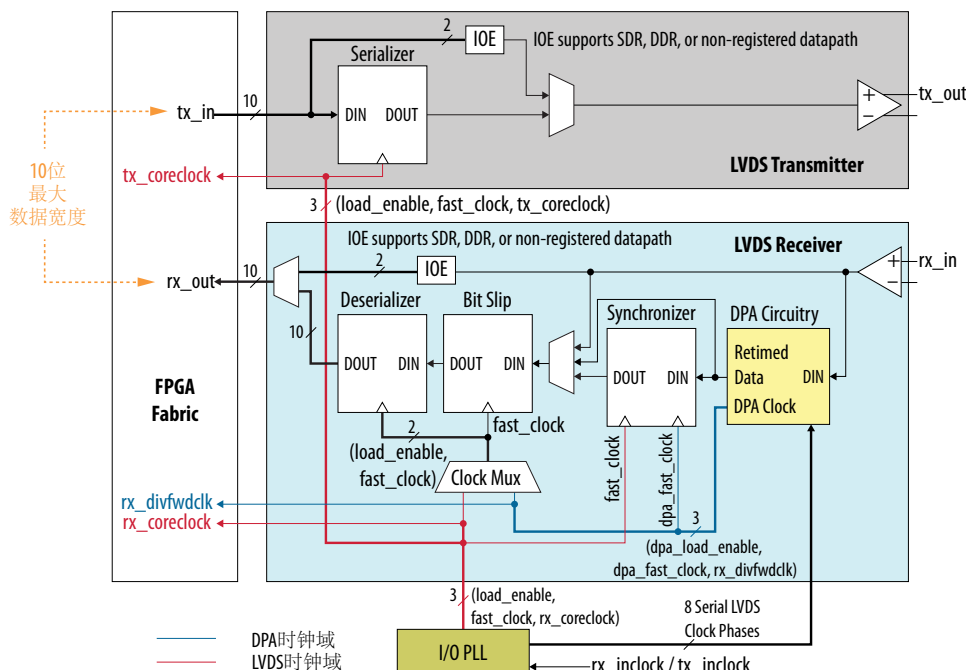
- [Arria 10 器件的 FPGA I/O 所支持的 I/O 标准](#) (第 96 页)
提供了关于所支持的差分 I/O 标准的信息。
- [Arria 10 器件的 GPIO Bank、SERDES 和 DPA 位置](#) (第 100 页)
- [Arria 10 GX 封装的 FPGA I/O 资源](#) (第 105 页)
提供了 LVDS 通道数。
- [Arria 10 GT 封装的 FPGA I/O 资源](#) (第 106 页)
提供了 LVDS 通道数。
- [Arria 10 SX 封装的 FPGA I/O 资源](#) (第 107 页)
提供了 LVDS 通道数。
- [Altera LVDS SERDES IP 内核用户指南](#)

5.6.1. SERDES 电路

Arria 10 器件的每个 LVDS I/O 通道含有内置串行器/解串器(SERDES)电路，支持高速 LVDS 接口。通过对 SERDES 电路进行配置可以支持源同步通信协议，如 RapidIO®、XSBI、串行外设接口(SPI)以及异步协议。

图 100. SERDES

该图显示了发送器和接收器数据路径的连接信号的 LVDS SERDES 电路的发送器和接收器的结构图。它显示了发送器和接收器之间一个共享的 PLL。如果发送器和接收器没有共享同一个 PLL，则需要两个 I/O PLL。在单数据速率(SDR)和双数据速率(DDR)模式中，数据位宽分别为 1 和 2 位。



Altera LVDS SERDES 发送器和接收器需要来自 I/O PLL 的多种时钟和加载使能信号。Quartus Prime 软件会自动配置 PLL 设置，并根据输入参考时钟和所选择的数据速率生成多种时钟和加载使能信号。

注意: 要了解 Arria 10 器件所支持的最高数据速率的更多信息，请参考器件概述。

相关链接

- [Arria 10 器件概述，特性汇总](#)
- [指南：将整数分频 PLL 模式中的 PLL 用于 LVDS \(第 154 页\)](#)

5.6.2. Arria 10 器件中支持的 SERDES I/O 标准

这些表格列出了 SERDES 接收器的发送器以及相应的 Quartus Prime 软件约束值支持的 I/O 标准。SERDES 接收器和发送器也支持所有差分 HSTL、差分 HSUL 和差分 SSTL I/O 标准。

表 60. SERDES 接收器 I/O 标准支持

I/O 标准	Quartus Prime 软件的约束(Assignment)值
真 LVDS	LVDS
差分 1.2 V HSTL Class I	Differential 1.2-V HSTL Class I
差分 1.2 V HSTL Class II	Differential 1.2-V HSTL Class II
差分 HSUL-12	Differential 1.2-V HSUL
继续...	



I/O 标准	Quartus Prime 软件的约束(Assignment)值
差分 SSTL-12	Differential 1.2-V SSTL
差分 SSTL-125	Differential 1.25-V SSTL
差分 SSTL-135	Differential 1.35-V SSTL
差分 1.5 V HSTL Class I	Differential 1.5-V HSTL Class I
差分 1.5 V HSTL Class II	Differential 1.5-V HSTL Class II
差分 SSTL-15	Differential 1.5-V SSTL
差分 SSTL-15 Class I	Differential 1.5-V SSTL Class I
差分 SSTL-15 Class II	Differential 1.5-V SSTL Class II
差分 1.8 V HSTL Class I	Differential 1.8-V HSTL Class I
差分 1.8 V HSTL Class II	Differential 1.8-V HSTL Class II
差分 SSTL-18 Class I	Differential 1.8-V SSTL Class I
差分 SSTL-18 Class II	Differential 1.8-V SSTL Class II
差分 POD12	Differential 1.2-V POD

表 61. SERDES 发送器 I/O 标准支持

I/O 标准	Quartus Prime 软件的约束(Assignment)值
真 LVDS	LVDS
差分 1.2 V HSTL Class I	Differential 1.2-V HSTL Class I
差分 1.2 V HSTL Class II	Differential 1.2-V HSTL Class II
差分 HSUL-12	Differential 1.2-V HSUL
差分 SSTL-12	Differential 1.2-V SSTL
差分 SSTL-125	Differential 1.25-V SSTL
差分 SSTL-135	Differential 1.35-V SSTL
差分 1.5 V HSTL Class I	Differential 1.5-V HSTL Class I
差分 1.5 V HSTL Class II	Differential 1.5-V HSTL Class II
差分 SSTL-15	Differential 1.5-V SSTL
差分 SSTL-15 Class I	Differential 1.5-V SSTL Class I
差分 SSTL-15 Class II	Differential 1.5-V SSTL Class II
差分 1.8 V HSTL Class I	Differential 1.8-V HSTL Class I
差分 1.8 V HSTL Class II	Differential 1.8-V HSTL Class II
差分 SSTL-18 Class I	Differential 1.8-V SSTL Class I
差分 SSTL-18 Class II	Differential 1.8-V SSTL Class II
差分 POD12	Differential 1.2-V POD
mini-LVDS	mini-LVDS
RSDS	RSDS

5.6.3. Arria 10 器件的差分发送器

Arria 10 发送器包含专用的电路来支持高速差分信号。差分发送器缓冲支持以下功能：

- LVDS 信号可驱动 LVDS、mini-LVDS 和 RSDS 信号
- 可编程 V_{OD} 和可编程预加重

表 62. 差分发送器的专用电路和功能

专用电路/功能	说明
差分 I/O 缓冲器	支持 LVDS、mini-LVDS 和 RSDS
SERDES	高达 10 位宽的串行器
锁相环(PLL)	同步加载和移位寄存器
可编程 V_{OD}	静态
可编程预加重	提升输出电流

相关链接

指南：将整数分频 PLL 模式中的 PLL 用于 LVDS (第 154 页)

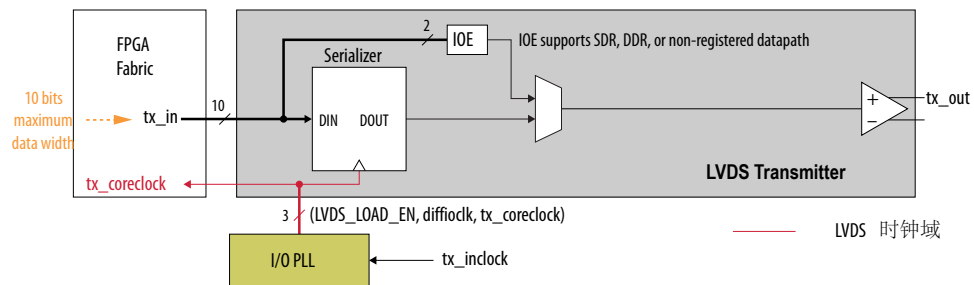
5.6.3.1. Arria 10 器件的发送器模块

专用电路由真差分缓冲器、串化器，以及在发送器与接收器之间可共享的 I/O PLL 所组成。串化器从 FPGA 逻辑中获取高达 10 位宽的并行数据，然后将此数据同步到加载寄存器，在发送此数据到差分缓冲器之前，使用由 I/O PLL 同步的移位寄存器将其串化。并行数据的 MSB 首先被发送。

注意：要驱动 LVDS 通道，必须使用整数分频 PLL 模式中的 PLL。

图 101. LVDS 发送器

该图显示了发送器的结构图。在 SDR 和 DDR 模式中，数据位宽分别为 1 和 2 位。



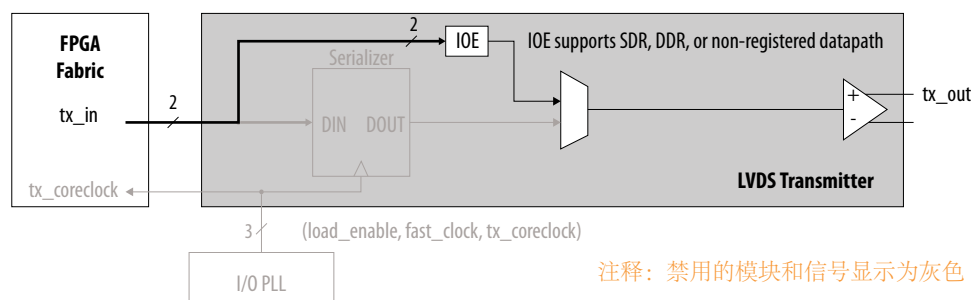
5.6.3.2. DDR 和 SDR 的串化器旁路操作

I/O 单元(IOE)包含可在 DDR 或 SDR 模式中操作的两个数据输出寄存器。

可以旁路串化器，来支持 DDR (x2)和 SDR (x1)的操作，分别实现串化因子 2 和 1。解串器旁路通过 Altera GPIO IP 内核受到支持。

图 102. 串行器旁路

下图显示了串行器的旁路路径。在 DDR 模式中，tx_inclock 对 IOE 寄存器提供时钟。在 SDR 模式中，数据直接通过 IOE。在 SDR 和 DDR 模式中，IOE 的数据位宽分别为 1 和 2 位。



5.6.4. Arria 10 器件中的差分接收器

此接收器具有一个差分缓冲和 I/O PLL，可在发送器与接收器之间共享、一个 DPA 模块、一个同步器、一个数据重对齐模块和一个解串器。差分缓冲器可接收 LVDS、mini-LVDS 和 RSDS 信号电平。您可以在 Quartus Prime 软件的 Assignment Editor 中静态地将接收器管脚的 I/O 标准设置成 LVDS、SLVS、mini-LVDS 或者 RSDS。

注意：要驱动 LVDS 通道，必须使用整数分频 PLL 模式中的 PLL。

表 63. 差分接收器的专用电路和特性

专用电路/特性	说明
差分 I/O 缓冲器	支持 LVDS、mini-LVDS 和 RSDS
SERDES	高达 10 位宽的解串器
锁相环(PLL)	生成不同相位的时钟用于数据同步器
数据重新对齐(Bit slip)	位延迟插入到串行数据
DPA	选择最接近串行数据相位的相位
同步器(FIFO 缓冲器)	对数据与接收器的输入参考时钟之间的相位差异进行补偿
偏移调整	手动
片上匹配(OCT)	在 LVDS I/O 标准中 100 Ω

相关链接

指南：将整数分频 PLL 模式中的 PLL 用于 LVDS (第 154 页)

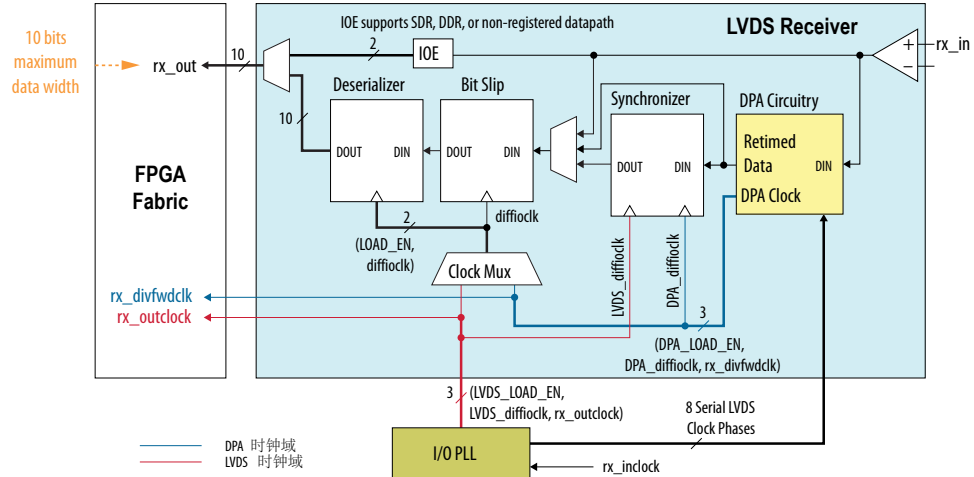
5.6.4.1. Arria 10 器件的接收器模块

Arria 10 差分接收器具有以下几个硬件模块：

- DPA 模块
- 同步器
- 数据重对齐模块(bit slip)
- 解串器

图 103. 接收器结构图

该图显示了接收器的硬件结构。在 SDR 和 DDR 模式中，来自 IOE 的数据宽度分别是 1 和 2 位。解串器包括移位寄存器和并行加载寄存器，最多发送 10 位到内部逻辑。

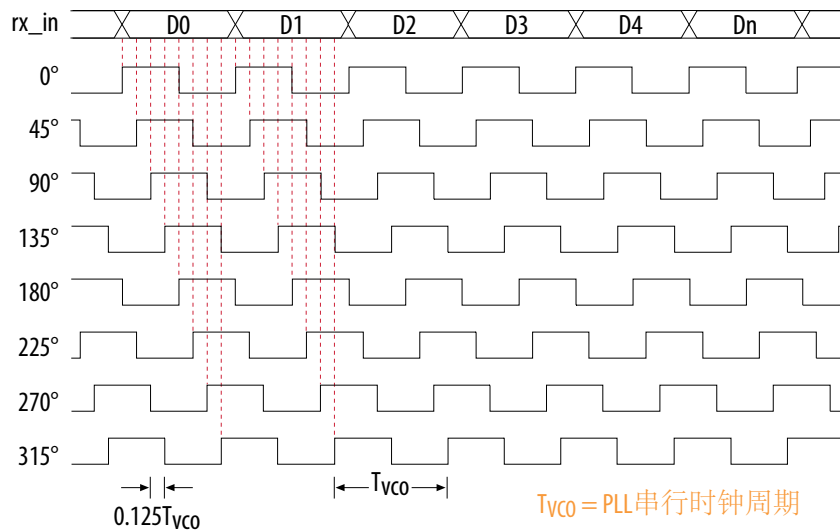


5.6.4.1.1. DPA 块

DPA 块接收来自差分输入缓冲器的高速串行数据，并从 I/O PLL 生成的 8 个相位中选择其一对数据进行采样。DPA 选择最靠近串行数据相位的相位。接收数据与所选相位之间的最大相位偏移为 1/8 单位间隔 (UI)⁽⁹⁾，也是 DPA 的最大量化误差。时钟的 8 个相位被平均分配，提供 45°分辨率。

图 104. DPA 时钟相位到串行数据时序的关系

该图显示了 DPA 时钟和输入串行数据之间可能的相位关系。



DPA 块持续监控输入串行数据的相位，并在需要的情况下选择新的相位。通过置位可选的 rx_dpa_hold 端口，可以防止 DPA 选择新的时钟相位，此方法适用于所有通道。

⁽⁹⁾ 单位间隔是以串行数据速率（快速时钟）运行的时钟的周期。



DPA 电路不需要固定的训练码型以锁定到 8 个相位中的最佳相位。复位或上电后，DPA 电路需要已接收数据上的跳变以锁定到最佳相位。可选的输出端口 `rx_dpa_locked` 可用于标示上电或者复位后初始 DPA 锁定条件已到达最佳相位。使用数据检查器，例如：循环冗余校验（CRC）或对角交叉存取奇偶校验（DIP-4）验证数据。

独立复位端口 `rx_dpa_reset` 可用于复位 DPA 电路。复位后必须重新训练 DPA 电路。

注意: Non-DPA 模式下 DPA 块被旁路。

相关链接

指南：将整数分频 PLL 模式中的 PLL 用于 LVDS (第 154 页)

5.6.4.1.2. 同步器

同步器是 1 位宽和 6 位深 FIFO 缓冲器，用于对 DPA 块选择的最佳时钟 `dpa_fast_clock` 与 I/O PLL 生成的 `fast_clock` 之间的相位差进行补偿。同步器仅补偿相位差，不能对数据与接收器的输入参考时钟之间的频率差异进行补偿。

可选端口 `rx_fifo_reset` 可用于内部逻辑复位同步器。当 DPA 首次锁定到输入数据时，同步器被自动复位。当数据检查器标示已接收数据损坏时，Intel 建议使用 `rx_fifo_reset` 复位同步器。

注意: 同步器电路在 non-DPA 和 soft-CDR 模式下被旁路。

相关链接

指南：将整数分频 PLL 模式中的 PLL 用于 LVDS (第 154 页)

5.6.4.1.3. 数据重对齐模块(Bit Slip)

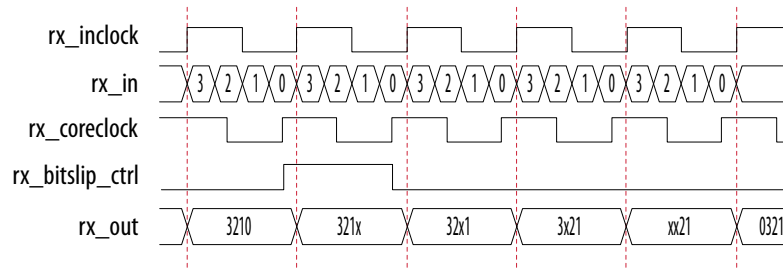
在发送数据中的偏移连同由链路添加的偏移一起会导致所接收串行数据流通道至通道的偏移。如果使能 DPA 模块，那么接收数据被每个通道中的不同时钟相位所采样。这种差异可能会导致接收数据在通道之间未被对齐。为了对该通道到通道的偏移进行补偿，并且在每个通道上建立正确的接收字边界，每个接收通道有一个专用的数据重对齐电路通过将延迟位插入到串行流来重新对齐数据。

可选端口 `rx_bitslip_ctrl` 可以独立地通过内部逻辑控制每个接收器的位元插入。数据会在 `rx_bitslip_ctrl` 的上升沿滑一个位。对 `rx_bitslip_ctrl` 信号的要求包括以下几项：

- 最小脉冲宽度是逻辑阵列中并行时钟的一个周期。
- 脉冲之间的最小低电平时间是并行时钟的一个周期。
- 该信号是一个边沿触发信号。
- 有效数据在 `rx_bitslip_ctrl` 的上升沿之后的并行时钟的四个周期后可用。

图 105. 数据重对齐时序

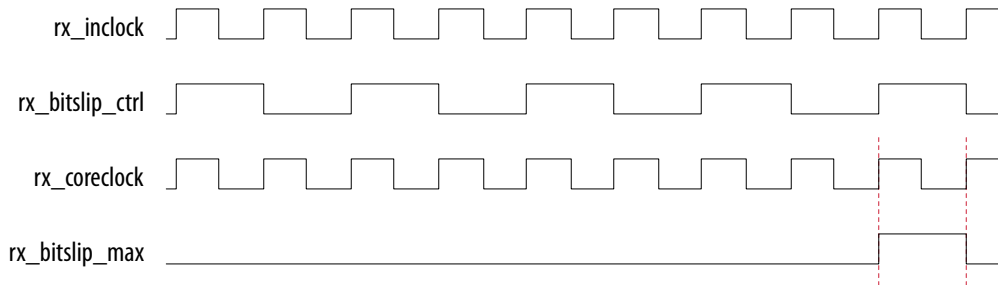
该图显示了一个 bit slip 脉冲(解串因子设为 4)之后的接收器输出(rx_out)。



数据重对齐电路有一个位滑翻转值设置成解串因子。每个通道都有一个可选的状态端口 **rx_bitslip_max** 送到 FPGA 逻辑, 来指示预设翻转点到达的时间。

图 106. 接收器数据重对齐翻转

该图显示了翻转出现前的一个 4 位周期的预设值。rx_bitslip_max 信号脉冲一个 rx_coreclock 周期来表明已经出现翻转。



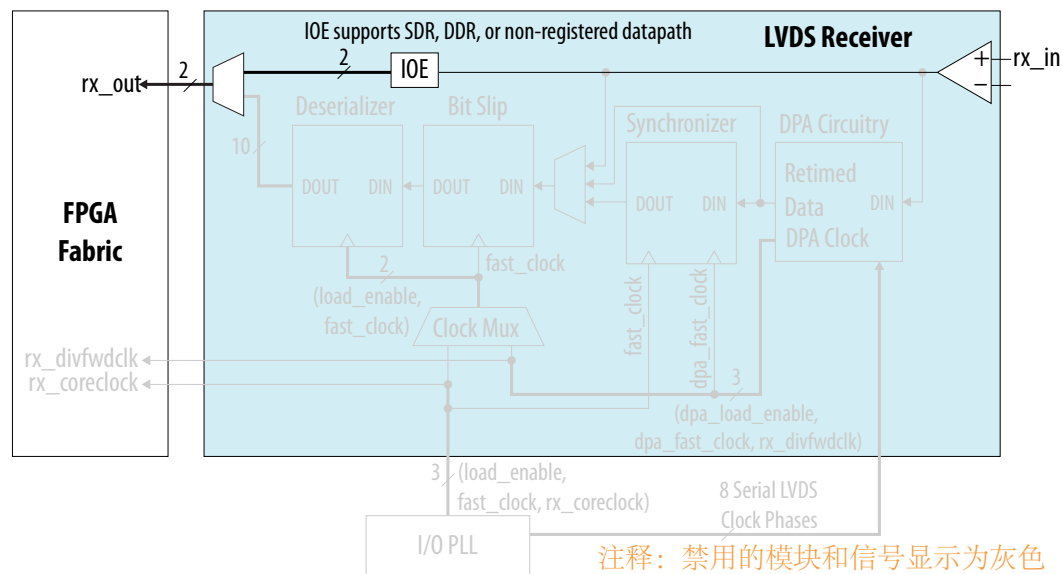
5.6.4.1.4. 解串器

通过使用 Quartus Prime 软件, 您可以将解串因子静态地设置成 x3、x4、x5、x6、x7、x8、x9 或者 x10。

IOE 包含两个可在 DDR 或者 SDR 模式中运行的数据输入寄存器。可以旁路解串器以支持 DDR (x2)和 SDR (x1)的操作。这个解串器旁路通过 Altera GPIO IP 内核受到支持。

图 107. 解串器旁路

该图显示了解串器的旁路路径。



当旁路解串器时，不可使用 DPA 和数据重对齐电路。

5.6.4.2. Arria 10 器件中的接收器模式

Arria 10 器件支持如下接收器模式:

- Non-DPA 模式
- DPA 模式
- Soft-CDR 模式

注意: 如果使用 **DPA** 模式, 则请按照建议的初始化和复位流程。建议的流程确保 **DPA** 电路可从 **PLL** 中检测到最佳相位抽头以采集接收器上的数据。

相关链接

建议的初始化和复位流程

提供了初始化和复位 LVDS SERDES IP 内核所建议的步骤。

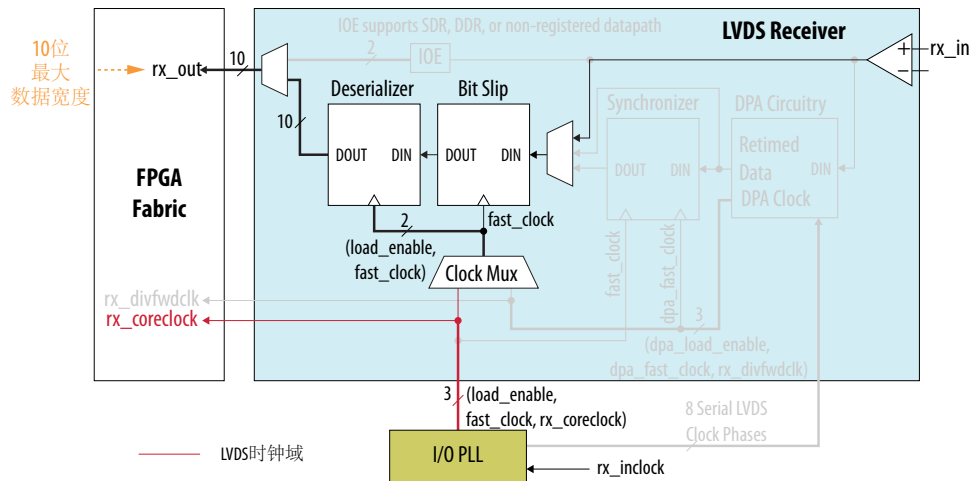
5.6.4.2.1. Non-DPA 模式

non-DPA 模式禁用 DPA 和同步器模块。输入串行数据被 I/O PLL 产生的串行 fast_clock 时钟的上升沿上所寄存。

通过使用 Quartus Prime 参数编辑器来选择 rising edge 选项。由 I/O PLL 生成的 fast_clock 时钟对数据重对齐和解串器模块提供时钟。

图 108. Non-DPA 模式的接收器数据通路

该图显示了 non-DPA 数据通路的结构图。在 SDR 和 DDR 模式中，来自 IOE 的数据宽度分别是 1 和 2 位。



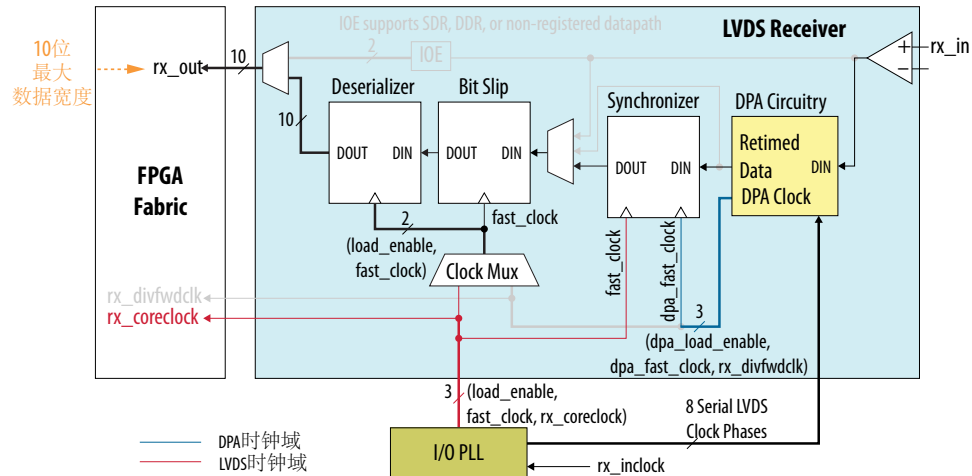
注释：禁用的模块和信号显示为灰色

5.6.4.2.2. DPA 模式

DPA 模块从 8 个由 I/O PLL 发送的快速时钟中选择最佳时钟(dpa_fast_clock)。该串行 dpa_fast_clock 时钟用于写入串行数据到同步器。串行 fast_clock 时钟用于从同步器中读取串行数据。在数据重对齐和解串器模块中使用同一个 fast_clock 时钟。

图 109. DPA 模式中的接收器数据通路

该图显示了 DPA 模式的数据通路。图中显示所有接收器硬核模块是有效的。在 SDR 和 DDR 模式下，来自 IOE 的数据宽度分别是 1 和 2 位。



注释：禁用的模块和信号显示为灰色

注意：在 DPA 模式中，必须将 LVDS 实例的所有接收器通道布局在一个 I/O bank。因为每个 I/O bank 最多含有 24 对 LVDS I/O 缓冲对，每个 LVDS 实例最多可以支持 24 个 DPA 通道。

相关链接

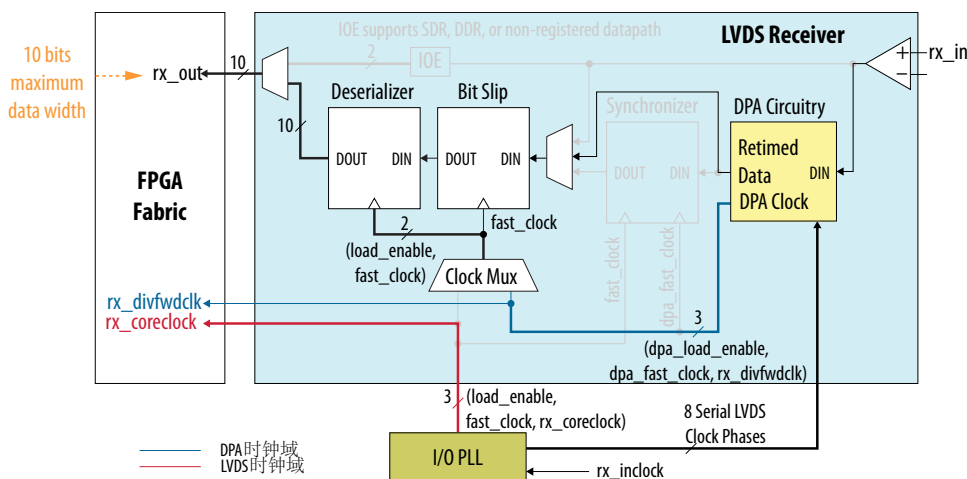
- 指南：将整数分频 PLL 模式中的 PLL 用于 LVDS (第 154 页)
- Arria 10 器件的接收器模块 (第 145 页)
列出并说明了接收器硬件的模块。

5.6.4.2.3. Soft-CDR 模式

Arria 10 LVDS 通道提供 soft-CDR 模式以支持 GbE 和 SGMII 协议。接收器 PLL 使用本地时钟源作为参考。

图 110. Soft-CDR 模式中的接收器数据通路

该图显示了 soft-CDR 模式的数据通路。在 SDR 和 DDR 模式下，来自 IOE 的数据宽度分别是 1 和 2 位。



注释：禁用的模块和信号显示为灰色

在 soft-CDR 模式中，同步器模块没有被启用。DPA 电路选择最佳的 DPA 时钟相位来对数据进行采样。该时钟被用于 bit-slip 操作和解串化。DPA 模块也将所选择的 DPA 时钟(由解串因子分频，称为 rx_divfwdclk)连同解串的数据一起转送到 FPGA 逻辑。该时钟信号被放置到外围时钟(PCLK)网络上。

如果使用 soft-CDR 模式，那么在训练 DPA 后，请不要置位 rx_dpa_reset 端口。DPA 将持续从 PLL 中选择新的相位抽头，以跟踪参考时钟与输入数据之间的百万分率(PPM)差异。

在 Arria 10 器件系列中，您可以在 soft-CDR 模式中使用每一个 LVDS 通道，并使用 PCLK 网络驱动 FPGA 架构。在 soft-CDR 模式中，rx_dpa_locked 信号无效，因为 DPA 持续改变其相位以跟踪上游发送器与本地接收器输入参考时钟之间的 PPM 差异。不过，可以使用 rx_dpa_locked 信号来确定初始 DPA 锁定条件表明 DPA 已经选到最佳抽头来采集数据。在 soft-CDR 模式下操作时，预计会置低 rx_dpa_locked 信号。并行时钟，rx_coreclock，由 I/O PLL 生成，也被转送到 FPGA 逻辑。

注意：

在 soft-CDR 模式中，必须将 LVDS 实例的所有接收器通道布局在一个 I/O bank。因为每个 I/O bank 最多含有 12 个 PCLK 资源，每个 LVDS 实例最多可以支持 12 个 soft-CDR 通道。

相关链接

- 指南：Soft-CDR 模式的 LVDS SERDES 管脚对 (第 172 页)

- [外设时钟网络](#) (第 74 页)
提供了关于 PCLK 网络的详细信息。

5.6.5. Arria 10 器件的 PLL 和时钟

要生成并行时钟(rx_coreclock 和 tx_coreclock)和高速时钟(fast_clock), Arria 10 器件在高速差分 I/O 接收器和发送器通道中提供 I/O PLL。

相关链接

- [Arria 10 器件的 GPIO Bank、SERDES 和 DPA 位置](#) (第 100 页)
- [时钟差分发送器](#) (第 152 页)
- [时钟差分接收器](#) (第 153 页)
- [指南: 将整数分频 PLL 模式中的 PLL 用于 LVDS](#) (第 154 页)
- [指南: 通过使用 PLL 的高速时钟仅对 LVDS SERDES 提供时钟](#) (第 154 页)
- [指南: 差分通道的管脚布局](#) (第 154 页)
- [External PLL 模式的 LVDS 接口](#) (第 157 页)
- [指南: 支持 I/O PLL 参考时钟输入管脚的 I/O 标准](#) (第 170 页)

5.6.5.1. 时钟差分发送器

I/O PLL 生成加载使能(load_enable)信号以及 fast_clock 信号(时钟运行在串行数据速率), 对加载和移位寄存器提供时钟。通过使用 Quartus Prime 软件, 您可以将串化因子静态地设置成 x3、x4、x5、x6、x7、x8、x9 或者 x10。加载使能信号源自串化因子的设置。

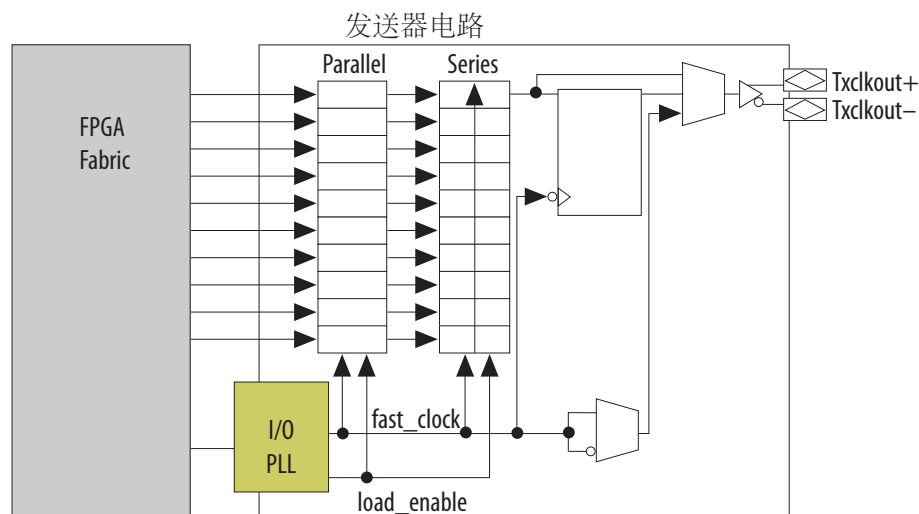
您可以配置任意的 Arria 10 发送器数据通道来生成源同步发送器输出时钟。这种灵活性允许将时钟输出布局在数据输出附近, 从而简化板级布线复杂度并降低时钟到数据偏移。

不同的应用经常会需要指定的时钟到数据(clock-to-data)对齐或者指定的数据速率到时钟速率(data-rate-to-clock-rate)因子。您可以在 Quartus Prime 参数编辑器中静态地指定这些设置:

- 发送器能够输出与数据具有相同速率的时钟信号 — 等同每种速度等级器件支持的最大输出时钟频率。
- 输出时钟也能够被因子 1、2、4、6、8 或者 10 整除, 这取决于串化因子。
- 与数据相关的时钟相位可被设置为 0° 或者 180°(边沿或者中间对齐)。I/O PLL 对 45° 递增的其它相移提供额外的支持。

图 111. 时钟输出模式中的发送器

该图显示了时钟输出模式中的发送器。在时钟输出模式中，可以将 LVDS 通道用作时钟输出通道。



相关链接

- 指南：将整数分频 PLL 模式中的 PLL 用于 LVDS (第 154 页)
- Arria 10 器件的 PLL 和时钟 (第 152 页)

5.6.5.2. 时钟差分接收器

I/O PLL 接收外部时钟输入，并生成同一时钟的不同相位。DPA 块自动从 I/O PLL 中选择一个时钟，并将每个通道上的输入数据对齐。

同步器电路是一个 1 位宽乘 6 位深的 FIFO 缓冲器，它对 DPA 块与数据重对齐块之间的相位差进行补偿。如有需要，则用户控制的数据重对齐电路在串行位流中插入单个位延迟，将字对齐到边界。解串器包括移位寄存器和并行加载寄存器，并最多发送 10 位到内部逻辑。

连接到发送器和接收器 LVDS 通道的物理介质可能会导致串行数据与源同步时钟之间的偏斜。每个 LVDS 通道与时钟之间的瞬间偏斜也因为通过接收器所观察到数据和时钟信号上的抖动情况而异。三种不同的模式 — non-DPA、DPA 和 soft-CDR — 提供了不同选项以克服源同步时钟 (non-DPA, DPA) /参考时钟 (soft-CDR) 与串行数据之间的偏斜。

Non-DPA 模式支持静态选择源同步时钟与已接收串行数据之间的最佳相位以对偏斜进行补偿。DPA 模式中，DPA 电路自动选择最佳相位，以补偿源同步时钟和接收串行数据之间的偏斜。Soft-CDR 模式对芯片到芯片的同步和异步应用程序，以及 SGMII 协议的短距离板级到板级应用程序提供机会。

注意: 仅 non-DPA 模式需要手动偏斜调整。

相关链接

- 指南：将整数分频 PLL 模式中的 PLL 用于 LVDS (第 154 页)
- Arria 10 器件的 PLL 和时钟 (第 152 页)

5.6.5.2.1. 指南：跨越多个 I/O Bank 的时钟 DPA 接口

使用超过 24 个通道的 DPA 接口跨越多个 I/O bank。Intel 建议使用专用的 `refclk` 管脚来驱动 DPA 接口中每个 I/O bank 的 I/O PLL。按照这一建议来实现器件表中所列的最大 DPA LVDS 规范。

相关链接

[高速 I/O 规范](#)

5.6.5.2.2. 指南：DPA 或 Non-DPA 接收器的 I/O PLL 参考时钟源

DPA 或 non-DPA LVDS 接收器的 I/O PLL 的参考时钟必须来自 I/O bank 内的专用参考时钟管脚。

注意：这一要求不适用于 LVDS 发送器。

5.6.5.3. 指南：将整数分频 PLL 模式中的 PLL 用于 LVDS

每个 I/O bank 有其自身的 PLL (I/O PLL) 来驱动 LVDS 通道。这些 I/O PLL 仅在整数模式下运行。

相关链接

[Arria 10 器件的 PLL 和时钟 \(第 152 页\)](#)

5.6.5.4. 指南：通过使用 PLL 的高速时钟仅对 LVDS SERDES 提供时钟

从 PLL 生成的高速时钟仅用于对 LVDS SERDES 电路提供时钟。不可使用高速时钟驱动其它逻辑，因为用于驱动内核逻辑的允许频率受到 PLL F_{OUT} 规范的限制。

要了解关于 F_{OUT} 规范的详细信息，请参阅器件数据表。

相关链接

- [PLL 规范](#)
- [Arria 10 器件的 PLL 和时钟 \(第 152 页\)](#)

5.6.5.5. 指南：差分通道的管脚布局

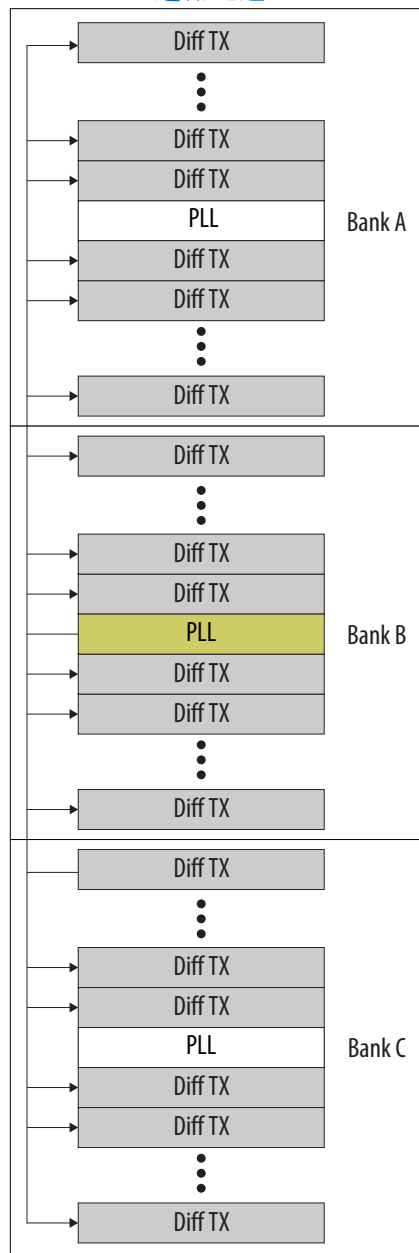
每个 I/O bank 包含其自身的 PLL。I/O bank PLL 能够驱动相同 bank 中的所有接收器和发送器通道，以及相邻 I/O bank 中的发送器通道。不过，I/O bank PLL 无法驱动另一个 I/O bank 的接收器通道或非相邻 I/O bank 中的发送器。

PLL 驱动差分发送器通道

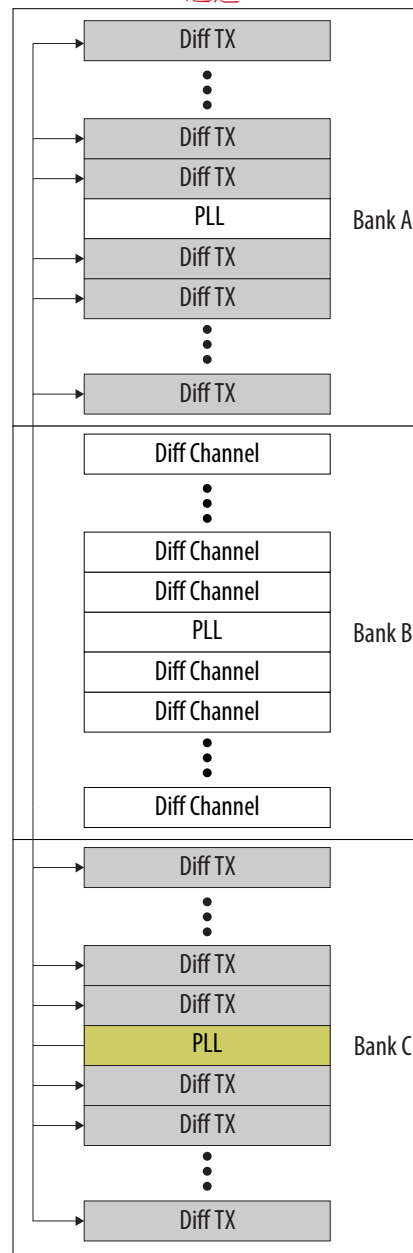
对于差分发送器，PLL 能够驱动自身 I/O bank 和相邻 I/O bank 中的差分发送器通道。不过，PLL 无法驱动非相邻 I/O bank 中的通道。

图 112. PLL 驱动差分发送器通道

有效：PLL 驱动相邻 bank 中的发送器通道



无效：PLL 驱动跨 bank 中的发送器通道



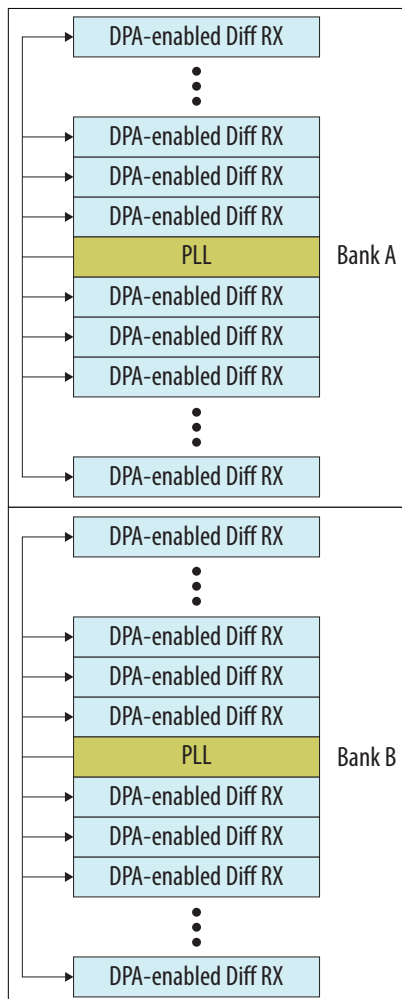
PLL 驱动 DPA-Enabled 差分接收器通道

对于差分接收器，PLL 能够驱动相同 I/O bank 中的所有通道，但无法跨过 bank 进行驱动。

I/O 模块中的每个差分接收器都有一个专用 DPA 电路，使时钟的相位对齐到其相关通道的数据相位。如果在一个 bank 中使能了 DPA 通道，那么在此 bank 中就可以同时使用单端 I/O 和差分 I/O 标准。

DPA 的使用增加了对高速差分接收器通道布局的一些限制。Quartus Prime 编译器自动检查设计，并且在布局指南违规时，发出错误信息警告。遵循指南以确保合适的高速 I/O 操作。

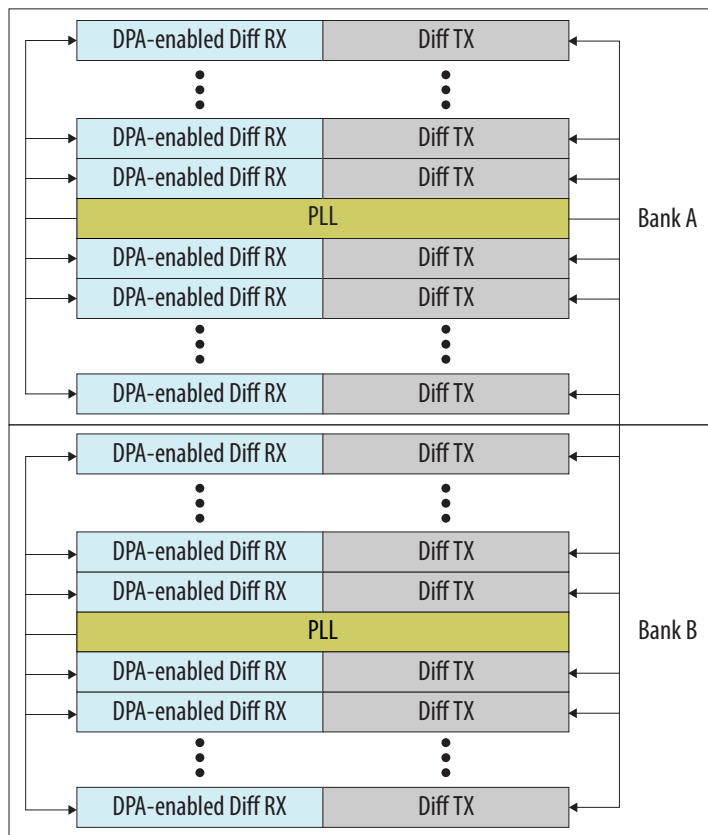
图 113. PLL 驱动 DPA-Enabled 差分接收器通道



交错的 PLL 驱动 DPA-Enabled 差分发送器和接收器通道

如果在 bank 中同时使用差分发送器通道和 DPA-enabled 接收器通道，可以将 bank 中 I/O PLL 驱动接收器通道与相邻 bank 中 I/O PLL 驱动发送器通道一起交错。

图 114. 交错的 PLL 驱动 DPA-Enabled 差分发送器和接收器通道



相关链接

Arria 10 器件的 PLL 和时钟 (第 152 页)

5.6.5.6. External PLL 模式的 LVDS 接口

Altera LVDS SERDES IP 内核参数编辑器提供了一个 **Use External PLL** 选项来实现 LVDS 接口的选项。通过使能此选项，您能够控制 PLL 设置，例如动态重配置 PLL 以支持不同的数据速率，动态相移以及其它设置。您也必须例化一个 Altera IOPLL IP 内核，来生成各种时钟和加载使能信号。

如果对 Altera LVDS SERDES 发送器和接收器使能 **Use External PLL** 选项，那么需要以下来自 Altera IOPLL IP 内核的信号：

- 到 Altera LVDS SERDES 发送器和接收器的 SERDES 的串行时钟输入信号
- 到 Altera LVDS SERDES 发送器和接收器的 SERDES 的加载使能信号
- 用于对发送器 FPGA 架构逻辑提供时钟的并行时钟以及用于接收器的并行时钟
- Altera LVDS SERDES 接收器的异步 PLL 复位端口
- Altera LVDS SERDES 接收器的 DPA 和 soft-CDR 模式的 PLL VCO 信号

IP 参考编辑器的 **Clock Resource Summary** 选项卡提供了上述列表中信号的详细信息。

相关链接

- [Altera LVDS SERDES IP 内核用户指南](#)
- [Arria 10 器件的 PLL 和时钟 \(第 152 页\)](#)
- [Altera IOPLL 信号接口与 Altera LVDS SERDES IP 内核 \(第 158 页\)](#)
- [External PLL 模式的 Altera IOPLL 参数值 \(第 159 页\)](#)
- [Altera IOPLL 和 Altera LVDS SERDES 之间的连接 \(第 161 页\)](#)

5.6.5.6.1. Altera IOPLL 信号接口与 Altera LVDS SERDES IP 内核

表 64. Altera IOPLL 和 Altera LVDS SERDES IP 内核之间的信号接口

该表列出了 Altera IOPLL IP 内核的输出端口与 Altera LVDS SERDES 发送器和接收器的输入端口之间的信号接口。

从 Altera IOPLL IP 内核	至 Altera LVDS SERDES 发送器	至 Altera LVDS SERDES 接收器
Lvsds_clk[0] (串行时钟输出信号) <ul style="list-style-type: none"> • 使用 PLL 中的 outclk0 配置该信号。 • 对 Access to PLL LVDS_CLK/LOADEN output port 设置选择 Enable LVDS_CLK/LOADEN 0 或 Enable LVDS_CLK/LOADEN 0 & 1 选项。大多数情况下, 选择 Enable LVDS_CLK/LOADEN 0。 串行时钟输出只能驱动 Altera LVDS SERDES 发送器的接收器上的 ext_fclk。该时钟不能驱动内核逻辑。	ext_fclk (到发送器的串行时钟输入)	ext_fclk (串行时钟输入到接收器)
loaden[0] (加载使能输出) <ul style="list-style-type: none"> • 使用 PLL 中的 outclk1 配置该信号。 • 对于 Access to PLL LVDS_CLK/LOADEN output port 设置, 选择 Enable LVDS_CLK/LOADEN 0 或者 Enable LVDS_CLK/LOADEN 0 & 1 选项。在大多数情况下, 选择 Enable LVDS_CLK/LOADEN 0。 	ext_loaden (到发送器的加载使能信号)	ext_loaden (用于解串器的加载使能信号)
outclk2 (并行时钟输出)	ext_coreclock (并行内核时钟)	ext_coreclock (并行内核时钟)
locked	—	pll_areset (异步 PLL 复位端口)
phout[7:0] <ul style="list-style-type: none"> • 该信号只有 DPA 或者 soft-CDR 模式中的 LVDS 接收器需要。 • 启用 PLL 中的 Specify VCO frequency 来配置该信号, 并指定 VCO frequency 的值。 • 启用 Enable access to PLL DPA output port。 	—	ext_vcoph 该信号只有 DPA 或者 soft-CDR 模式中的 LVDS 接收器需要。

注意: soft SERDES 的使用对时钟有不同的要求。

相关链接

- [Altera LVDS SERDES IP 内核用户指南](#)
提供了更多关于 soft SERDES 的不同时钟要求的信息。
- [External PLL 模式的 LVDS 接口 \(第 157 页\)](#)

5.6.5.6.2. External PLL 模式的 Altera IOPLL 参数值

下面的实例显示了使用 Altera IOPLL IP 内核对 Altera LVDS SERDES 生成输出时钟的时钟要求。该实例相移的设定是基于时钟和数据在器件的管脚处是边沿对齐的前提。

注意: 对于其它情况下的时钟和数据相位设置，Intel 建议无需使用 external PLL mode 选项，先例化 Altera LVDS SERDES 接口。然后在 Quartus Prime 软件中编译 IP 内核，注意每个时钟输出的频率、相移和占空比设置。在 Altera IOPLL IP 内核参数编辑器中输入这些设置，然后将相应的输出连接到 Altera LVDS SERDES IP 内核。

表 65. 实例：使用 Altera IOPLL IP 内核生成输出时钟(不使用 DPA 和 Soft-CDR 模式)

该表列出了在 Altera IOPLL 参数编辑器中可以设定的参数值，在不使用 DPA 和 soft-CDR 模式的情况下，使用 Altera IOPLL IP 内核生成三个输出时钟。

参数	outclk0 (作为 lvds_clk[0] 连接到 Altera LVDS SERDES 发送器和接收器的 ext_fclk 端口)	outclk1 (作为 loaden[0] 连接到 Altera LVDS SERDES 发送器和接收器的 ext_loaden 端口)	outclk2 (作为发送器和接收器的并行数据寄存器的核心时钟使用，并连接到 Altera LVDS SERDES 的 ext_coreclock 端口)
频率	数据速率	数据速率/串行因子	数据速率/串行因子
相移	180°	$[(\text{解串因子} - 1) / \text{解串因子}] \times 360^\circ$	180/串行因子 (outclk0 相移除以串行因子)
占空比	50%	100/串行因子	50%

相移的计算，使用 RSKM 方程，假设输入时钟和串行数据是边沿对齐的。引进一个 180°相移到采样时钟(c0)确保了输入数据相对 outclk0 是中央对齐的，如下图所示。

图 115. External PLL 接口信号的相位关系

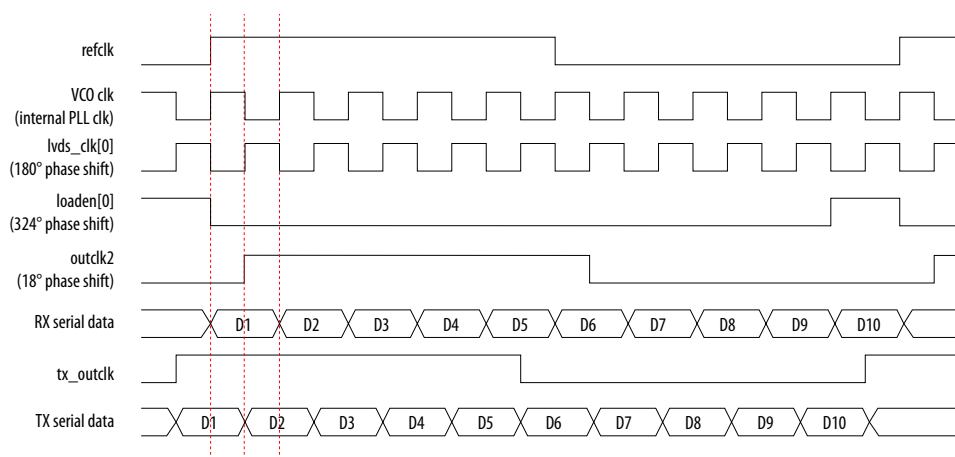


表 66. 实例：使用 Altera IOPLL IP 内核生成输出时钟 (使用 DPA 和 Soft-CDR 模式)

该表列出了在 Altera IOPLL 参数编辑器中可以设定的参数值，在使用 DPA 和 soft-CDR 模式的情况下，使用 Altera IOPLL IP 内核生成四个输出时钟。Altera IOPLL 的 locked 输出端口必须反转，并且在使用 DPA 和 soft-CDR 模式的情况下，连接到 Altera LVDS SERDES IP 内核的 pll_areset 端口。

参数	outclk0 (作为 lvds_clk[0] 连接到 Altera LVDS SERDES 发送器或接收器的 ext_fclk 端口)	outclk1 (作为 loaden[0] 连接到 Altera LVDS SERDES 发送器或接收器的 ext_loaden 端口)	outclk2 (作为发送器和接收器的并行数据寄存器的内核时钟使用，并连接到 Altera LVDS SERDES 的 ext_coreclock 端口)	VCO 频率 (作为 phout[7:0] 连接到 Altera LVDS SERDES 的 ext_vcoph[7:0] 端口)
频率	数据速率	数据速率/串化因子	数据速率/串化因子	数据速率
相移	180°	$[(\text{解串因子} - 1) / \text{解串因子}] \times 360^\circ$	180/串化因子 (outclk0 相移除以串化因子)	—
占空比	50%	100/串化因子	50%	—

表 67. 实例：对与接收器通道共享的跨越多个 Bank 的发送器使用共享的 Altera IOPLL IP 内核生成输出时钟 (使用 DPA 和 Soft-CDR 模式)

该表列出了在 Altera IOPLL 参数编辑器中可以设定的参数值，使用 Altera IOPLL IP 内核生成六个输出时钟。如果在 DPA 和 soft-CDR 模式中使用与接收器通道共享的跨越多个 bank 的发送器通道，就使用这些设置。Altera IOPLL 的 locked 输出端口必须反转，并且在使用 DPA 和 soft-CDR 模式的情况下，连接到 Altera LVDS SERDES IP 内核的 pll_areset 端口。

参数	outclk0 (作为 lvds_clk[0] 连接到 Altera LVDS SERDES 接收器的 ext_fclk 端口)	outclk1 (作为 loaden[0] 连接到 Altera LVDS SERDES 接收器的 ext_loaden 端口)	outclk4 (作为发送器和接收器的并行数据寄存器的内核时钟使用，并连接到 Altera LVDS SERDES 的 ext_coreclock 端口)	VCO 频率 (作为 phout[7:0] 连接到 Altera LVDS SERDES 的 ext_vcoph[7:0] 端口)
	outclk2 (作为 lvds_clk[1] 连接到 Altera LVDS SERDES 发送器的 ext_fclk 端口)	outclk3 (作为 loaden[1] 连接到 Altera LVDS SERDES 发送器的 ext_loaden 端口)		
频率	数据速率	数据速率/串化因子	数据速率/串化因子	数据速率
相移	180°	$[(\text{解串因子} - 1) / \text{解串因子}] \times 360^\circ$	180/串化因子 (outclk0 相移除以串化因子)	—
占空比	50%	100/串化因子	50%	—

相关链接

- Non-DPA 模式的接收器偏移裕量 (第 165 页)
RSKM 方程式用于相移计算。
- External PLL 模式的 LVDS 接口 (第 157 页)

5.6.5.6.3. Altera IOPLL 和 Altera LVDS SERDES 之间的连接

图 116. LVDS 连接 Altera IOPLL IP 内核(不使用 DPA 和 Soft-CDR 模式)

该图显示了不使用 DPA 和 soft-CDR 模式的情况下，Altera IOPLL 与 Altera LVDS SERDES IP 内核之间的连接。

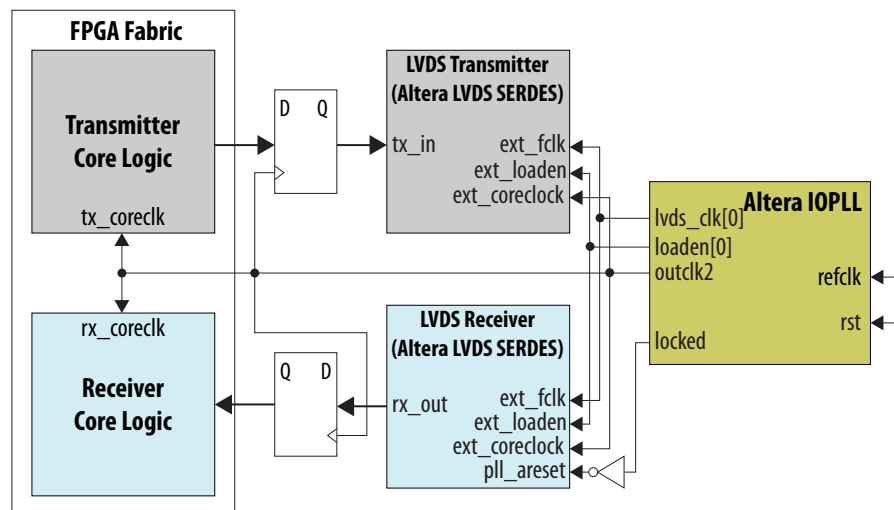


图 117. LVDS 连接 Altera IOPLL IP 内核(使用 DPA 模式)

该图显示了在使用 DPA 模式的情况下，Altera IOPLL 和 Altera LVDS SERDES IP 内核之间的连接。反转 locked 输出端口，并将其连接到 pll_areset 端口。

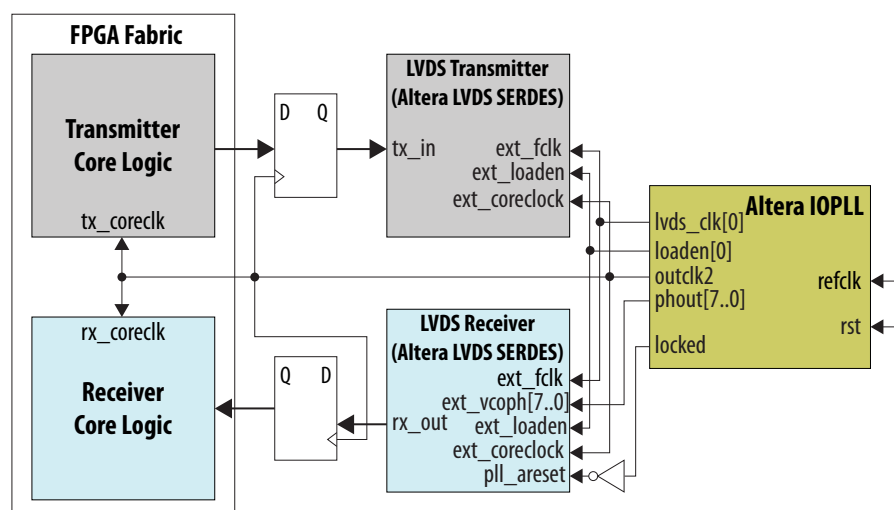


图 118. 对与接收器通道共享的跨越多个 Bank 的接收器通道使用共享的 I/O PLL，LVDS 连接 Altera IOPLL IP 内核(使用 DPA 模式)

该图显示了在使用跨越多个 bank 的发送器通道并与 DPA 接收器通道共享，使用共享的 I/O PLL 的情况下，Altera IOPLL 和 Altera LVDS SERDES IP 内核之间的连接。

- 将 I/O PLL lvds_clk[1] 和 loaden[1] 端口连接到 LVDS 发送器的 ext_fclk 和 ext_loaden 端口。
- 将 I/O PLL lvds_clk[0] 和 loaden[0] 端口连接到 LVDS 接收器的 ext_fclk 和 ext_loaden 端口。
- 反转 locked 输出端口，并将其连接到 pll_areset 端口。

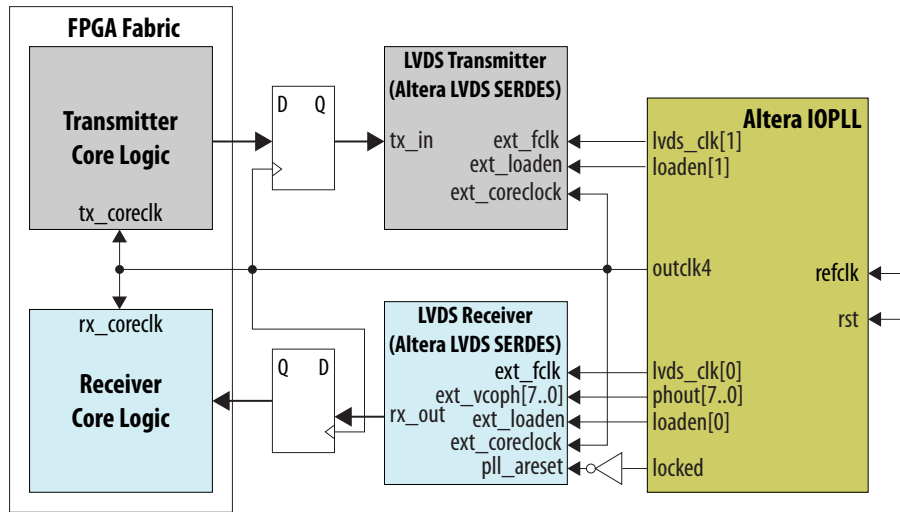
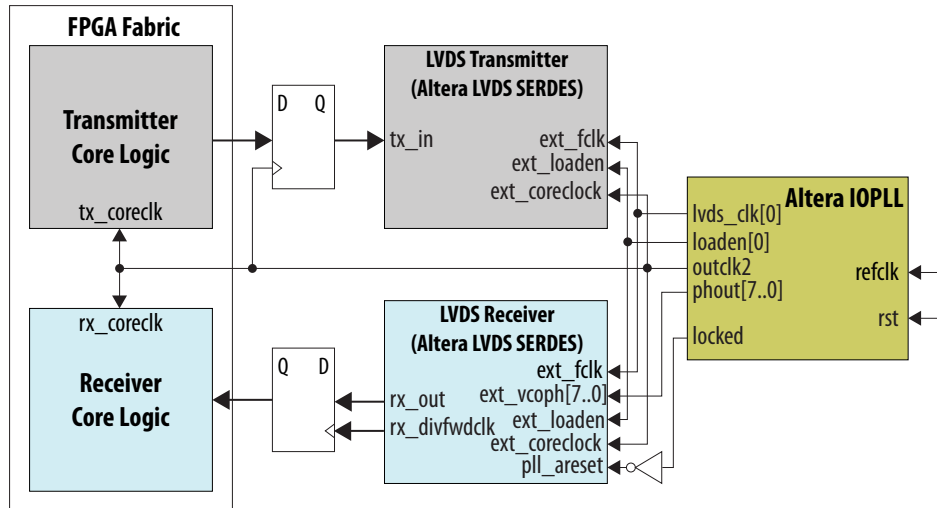


图 119. LVDS 连接 Altera IOPLL IP 内核(使用 Soft-CDR 模式)

该图显示了在使用 soft-CDR 模式的情况下，Altera IOPLL 和 Altera LVDS SERDES IP 内核之间的连接。反转 locked 输出端口，并将其连接到 pll_areset 端口。



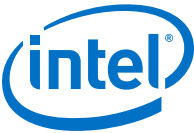


图 120. 对与接收器通道共享的跨越多个 Bank 的发送器通道使用共享的 I/O PLL，LVDS 连接 Altera IOPLL IP 内核(使用 Soft-CDR 模式)

该图显示了在使用跨越多个 bank 的发送器通道并与 soft-CDR 接收器通道共享，使用共享的 I/O PLL 的情况下，Altera IOPLL 和 Altera LVDS SERDES IP 内核之间的连接。

- 将 I/O PLL lvds_clk[1] 和 loaden[1] 端口连接到 LVDS 发送器的 ext_fclk 和 ext_loaden 端口。
- 将 I/O PLL lvds_clk[0] 和 loaden[0] 端口连接到 LVDS 接收器的 ext_fclk 和 ext_loaden 端口。
- 反转 locked 输出端口，并将其连接到 pll_areset 端口。

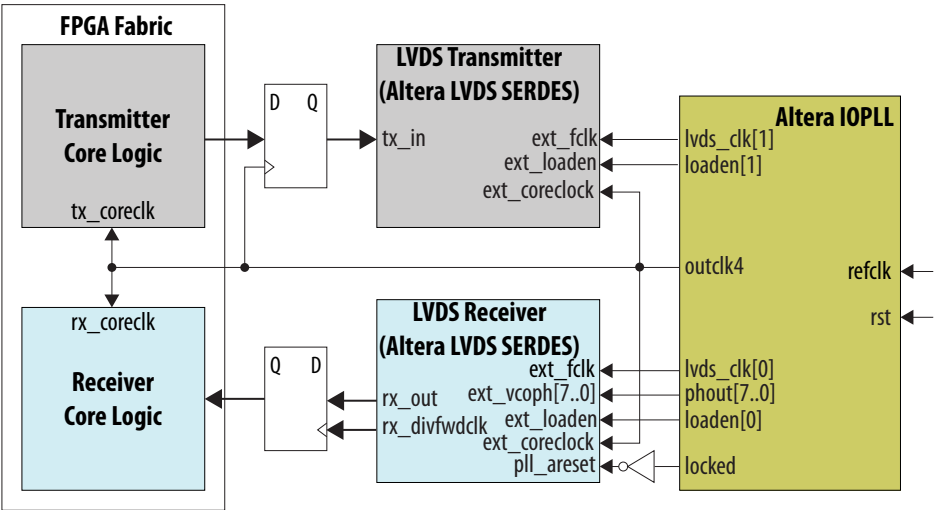


表 68. PLL 模式设置以生成 Altera IOPLL IP 内核

生成 Altera IOPLL IP 内核时，对相应的 LVDS 功能模式使用下表中的 PLL 设置。

LVDS 功能模式	PLL 设置
TX, RX DPA, RX Soft-CDR	Direct 模式
RX non-DPA	LVDS 补偿模式

在 external PLL 模式中，ext_coreclock 端口在 LVDS LVDS IP 内核中自动被使能。如果此端口没有如之前的图表所示被连接，那么 Quartus Prime 编译器输出错误信息。

相关链接

[External PLL 模式的 LVDS 接口 \(第 157 页\)](#)

5.6.6. Arria 10 器件的时序和优化

5.6.6.1. 源同步时序规划

本部分中的内容对 Arria 10 器件系列中源同步信号的时序规划、波形以及规范作了介绍。

LVDS I/O 标准使能了 high-speed 数据传输，实现更高的系统整体性能。要想利用快速的系统性能，必须分析这些高速信号的时序。对差分模块的时序分析不同于传统的同步时序分析技术。

源同步时序分析是基于数据与时钟信号之间的偏移，而不是基于时钟到输出的建立时间。高速差分数据传输需要使用 IC 供应商所提供的时序参数，并且会受到板级偏移、电缆偏移以及时钟抖动的严重影响。

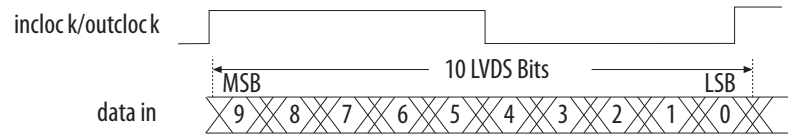
这一部分定义了 Arria 10 器件系列中的源同步差分数据定向的时序参数、时序规划，以及如何通过使用这些时序参数来决定设计的最佳性能。

5.6.6.1.1. 差分数据定向

外部时钟和输入数据之间存在一定关系。对于运行在 1 Gbps 和串化因子为 10 的操作，外部时钟乘以 10。您可以在 PLL 中设置相位对齐以符合每个数据位元的采样窗口。数据在被乘时钟的下降沿时被采样。

图 121. Quartus Prime 软件中的位定向 (Bit Orientation)

该图显示了 x10 模式中的数据位定向。



5.6.6.1.2. 差分 I/O 位位置

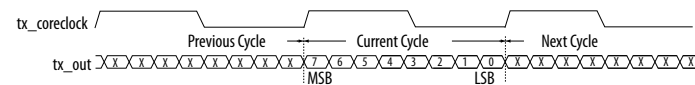
数据同步对于成功的高频数据传输是必要的。

图 122. 一个差分通道的位顺序和字边界

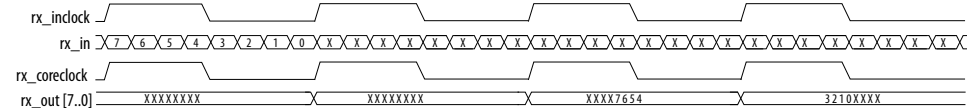
该图显示了通道操作的数据位元定向，它基于以下几个条件：

- 串化因子等于时钟倍频因子。
- 相位对齐使用边沿对齐。
- 该操作在硬核 SERDES 中实现。

发送器通道操作 (x8模式)



接收器通道操作 (x8模式)



注释：这些波形仅是功能波形，并不会传达时序信息

对于其它的串化因子，使用 Quartus Prime 在字中查找位位置。

差分位命名约定

表 69. 差分位命名

该表格列出 18 个差分通道的差分位命名约定。MSB 和 LSB 位置随着系统中所使用通道数量而增多。

接收器通道数据的数量	内部 8 位并行数据	
	MSB 位置	LSB 位置
1	7	0
2	15	8
3	23	16
继续...		



接收器通道数据的数量	内部 8 位并行数据	
	MSB 位置	LSB 位置
4	31	24
5	39	32
6	47	40
7	55	48
8	63	56
9	71	64
10	79	72
11	87	80
12	95	88
13	103	96
14	111	104
15	119	112
16	127	120
17	135	128
18	143	136

5.6.6.1.3. 发送器通道至通道偏移

接收器偏移裕量计算会使用发送器通道至通道偏移(TCCS) — 是源同步差分接口中基于 Arria 10 发送器的一个重要参数:

- TCCS 是最快和最慢数据输出跳变之间的差异, 包括 T_{CO} 的变化和时钟偏移。
- 在 LVDS 发送器中, TimeQuest 时序分析器在 Quartus Prime 编译报告的 TCCS 报告 (report_TCCS) 中提供一个 TCCS 值, 它显示了串行输出端口的 TCCS 值。
- 可以从器件数据表中获得 TCCS 值。

在 Arria 10 器件中, 必须执行 PCB 走线补偿来调整每个 LVDS 通道的走线长度, 当与数据速率高于 840 Mbps 的 non-DPA 接收器连接时, 可以改善通道至通道的偏移。Quartus Prime 软件的 Fitter Report 面板为 Arria 10 器件报告了必须添加到每条走线上的延迟数。可使用 LVDS Transmitter/Receiver Package Skew Compensation 面板发布建议的走线延迟数, 并手动补偿 PCB 电路板走线上的偏移, 来减少通道至通道的偏移, 从而满足 LVDS 通道之间的时序规划。

相关链接

- [高速 I/O 规范](#)
- [Altera LVDS SERDES IP 内核用户指南](#)
提供了关于 LVDS Transmitter/Receiver Package Skew Compensation 报告面板的详细信息。

5.6.6.1.4. Non-DPA 模式的接收器偏移裕量

不同模式的 LVDS 接收器使用不同的规范, 有助于评估正确采样所接收串行数据的能力。

- 在 DPA 模式中，使用 DPA 抖动容限而不是接收器偏移裕量(RSKM)。
- 在 non-DPA 模式中，将 RSKM、TCCS 和采样窗口(SW)规格用于接收数据通路中的高速源同步差分信号。

相关链接

- [Altera LVDS SERDES IP 内核用户指南](#)
提供了关于 LVDS Transmitter/Receiver Package Skew Compensation 报告面板的详细信息。
- [Quartus Prime TimeQuest 时序分析器](#)
提供了关于.sdc 命令和 TimeQuest 时序分析器的更多信息。

RSKM 方程

RSKM 方程反映 RSKM、TCCS 和 SW 之间的关系。

图 123. RSKM 方程

$$RSKM = \frac{TUI - SW - TCCS}{2}$$

方程中采用的约定：

- **RSKM** — 接收器时钟输入和数据输入采样窗口之间的时序裕量，以及内核噪声和 I/O 开关噪声诱发的抖动。
- **时间单位间隔 (TUI)** — 串行数据的时间周期。
- **SW** — 输入数据必须保持稳定以确保 LVDS 接收器成功进行数据采样的时间周期。SW 是一种器件属性，并因器件速度等级不同而异。
- **TCCS** — 由相同 PLL 驱动的跨通道间最快与最慢输出边沿时序差。该 TCCS 测量包括 t_{CO} 系列、时钟和时钟偏斜。

注意：如果有其他电路板通道至通道偏斜，可考虑接收器通道至通道总偏斜 (RCCS)，而非 TCCS。总 $RCCS = TCCS + \text{电路板通道至通道偏斜}$ 。

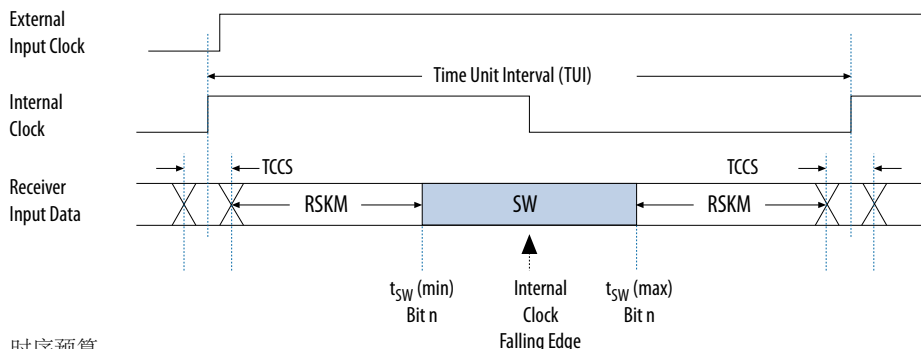
必须根据数据速率和器件计算 RSKM 值，以确定 LVDS 接收器是否能进行数据采样：

- 扣除发送器抖动后，正 RSKM 值表示 LVDS 接收器可正确采样数据。
- 扣除发送器抖动后，负 RSKM 值表示 LVDS 接收器无法正确采样数据。

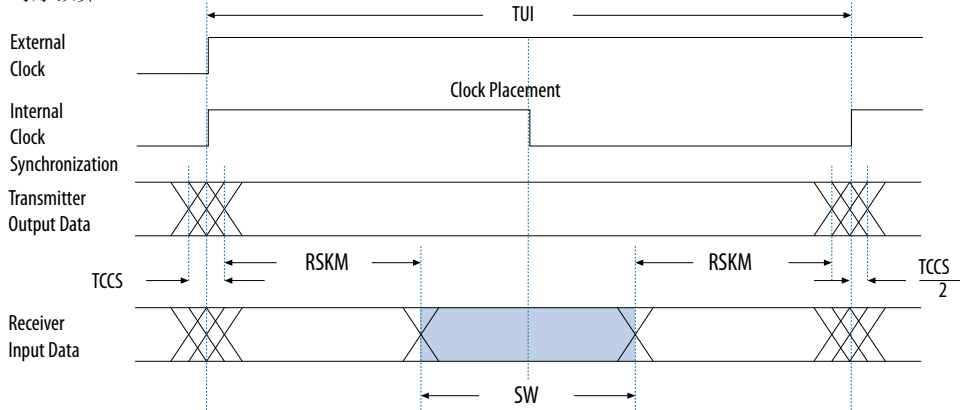
图 124. Non-DPA 模式的差分高速时序结构图和时序预算

该图显示为 RSKM、TCCS 和接收器 SW 之间的关系。

时序图



时序预算



LVDS 接收器的 RSKM 报告

对于 LVDS 接收器，Quartus Prime 提供一个 RSKM 报告，显示 non-DPA LVDS 模式下的 SW、TUI 和 RSKM 值。

- 要生成 RSKM 报告，可运行 TimeQuest 时序分析器中的 `report_RSKM` 命令。RSKM 报告在 Quartus Prime 编译报告的 TimeQuest 时序分析器部分。
- 要获得更真实的 RSKM 值，通过 TimeQuest 时序分析器的约束菜单(constraints menu)将输入延迟分配给 LVDS 接收器。输入延迟根据 LVDS 接收器端口上数据到达的时间而决定(相对参考时钟)。
- 如果在设置参数的 **Set Input Delay** 选项中设置输入延迟，那么将输入时钟名称设置成驱动 LVDS 接收器的源同步时钟的参考时钟。
- 如果在 TimeQuest 时序分析器中没有设置任何输入延迟，那么接收器通道至通道偏移默认为零。
- 您也可以使用 `set_input_delay` 命令在 Synopsys Design Constraint 文件(.sdc)中直接设置输入延迟。

通过 TimeQuest 时序分析器将 Input Delay 分配给 LVDS Receiver

要获得 RSKM 值，在 TimeQuest 时序分析器的约束菜单(constraints menu)中，将适当的输入延迟分配到 LVDS 接收器。

1. 在 TimeQuest 时序分析器的菜单上，选择 **Constraints > Set Input Delay**。
2. 在 **Set Input Delay** 窗口中，使用下拉菜单选择所需的时钟。时钟名称必须参照用于驱动 LVDS 接收器的源同步时钟。
3. 点击 **Browse** (在 **Targets** 栏旁边)。
4. 在 **Name Finder** 窗口中，点击 **List** 查阅所有可用端口的列表。根据所设置的输入延迟，选择在 LVDS 接收器串行输入端口，并点击 **OK**。
5. 在 **Set Input Delay** 窗口中，在 **Input delay** 选项和 **Delay value** 栏中设置相应的值。
6. 点击 **Run**，合并这些值到 TimeQuest 时序分析器中。
7. 重复执行从 1 (第 168 页) 开始的这些步骤，来分配相应的延迟到所有的 LVDS 接收器输入端口。如果已经分配了 **Input Delay**，并且需要对此端口添加更多的延迟，那么勾选 **Add Delay** 选项。

实例：RSKM 计算

该实例显示了 FPGA 器件在 200 ps 板级通道至通道偏移的 1 Gbps 数据速率上的 RSKM 计算。

- $TCCS = 100 \text{ ps}$ (待定属性)
- $SW = 300 \text{ ps}$ (待定属性)
- $TUI = 1000 \text{ ps}$
- 总的 $RCCS = TCCS + \text{板级通道至通道偏移} = 100 \text{ ps} + 200 \text{ ps} = 300 \text{ ps}$
- $RSKM = (TUI - SW - RCCS) / 2 = (1000 \text{ ps} - 300 \text{ ps} - 300 \text{ ps}) / 2 = 200 \text{ ps}$

如果扣除发送器抖动后 RSKM 大于 0 ps，那么 non-DPA 接收器将会正常工作。

5.7. 在 Arria 10 器件中使用 I/O 和高速 I/O

5.7.1. Arria 10 器件的 I/O 和高速 I/O 通用指南

为确保设计成功，有几方面需要考虑。这些设计指南适用于该器件系列的所有型号，除非另有说明。

指南：VREF 资源和 VREF 管脚 (第 169 页)

指南：遵守 3.0 V 连接的器件绝对最大额定值 (第 169 页)

指南：支持 I/O PLL 参考时钟输入管脚的 I/O 标准 (第 170 页)



5.7.1.1. 指南: V_{REF} 资源和 V_{REF} 管脚

对于 Arria 10 器件, 考虑下列的 V_{REF} 管脚指南:

- Arria 10 器件支持内部和外部 V_{REF} 源。可以通过带校准的内部 V_{REF} 来支持 DDR4 使用 POD12 I/O 标准。
 - 每个 I/O bank 有一个外部 V_{REF} 管脚, 对相同 bank 中的所有 I/O 提供一个外部 V_{REF} 源。
 - bank 中的每个 I/O 通道也有其自身的内部 V_{REF} 生成器。可以独立地配置每个 I/O 通道, 来使用其内部 V_{REF} 或者 I/O bank 的外部 V_{REF} 源。相同 I/O 通道的 I/O 管脚将会使用同一个 V_{REF} 源。
- 您可以对 V_{REF} 管脚附近的输入、输出或者双向管脚布局任何组合。没有 V_{REF} 管脚布局限制。
- V_{REF} 管脚专用于单端 I/O 标准。无法将 V_{REF} 管脚用作用户 I/O。

要了解关于 V_{REF} 管脚电容的详细信息, 请参阅器件手册。

相关链接

- [Arria 10 器件中的 I/O 标准电平 \(第 98 页\)](#)
- [管脚电容](#)
- [单端 I/O 标准规范](#)
- [单端 SSTL, HSTL 和 HSUL I/O 参考电压规范](#)
- [单端 SSTL, HSTL 和 HSUL I/O 标准信号规范](#)
- [Arria 10 器件的 I/O Bank 体系结构 \(第 116 页\)](#)

5.7.1.2. 指南: 遵守 3.0 V 连接的器件绝对最大额定值

为了确保器件的可靠性和正常运行, 当器件用作 3.0 V I/O 接口时, 一定不要超过器件的绝对最大额定值。要了解关于绝对最大额定值和跳变过程中允许的最大过冲的详细信息, 请参考器件手册。

提示: 执行 IBIS 或者 SPICE 仿真来确定过冲和下冲电压均在规范范围内。

单端发送器应用

如果将 Arria 10 器件作为发送器使用时, 请使用较慢的摆率和串行匹配来限制 I/O 管脚上的过冲和下冲。传输线效应导致接收器上有较大电压偏差, 这与驱动器和传输线之间的阻抗不匹配有关。通过匹配驱动器的阻抗与传输线的特征阻抗能够显著地降低过冲电压。您可以使用位于驱动器附近的串行匹配电阻来匹配总驱动器阻抗与传输线阻抗。

单端接收器应用

如果使用 Arria 10 器件作为接收器, 那么可使用外部钳位二极管来限制 I/O 管脚上的过冲和下冲电压。

使用 3.0 V bank 电源电压 (V_{CCIO}) 和 1.8 V 的 V_{CCPT} 电压来支持 3.0 V I/O 标准。在此方法中, 钳位二极管能够充分地钳位过冲电压, 使其保持在 DC 和 AC 输入电压规格内。钳位电压表示成 V_{CCIO} 和二极管正向电压的总和。

相关链接

- [Arria 10 器件中的 I/O 标准电平 \(第 98 页\)](#)

- 绝对最大额定值
- 允许的最大过冲和下冲电压

5.7.1.3. 指南：支持 I/O PLL 参考时钟输入管脚的 I/O 标准

I/O PLL 参考时钟(REFCLK)输入管脚仅支持以下的 I/O 标准：

- 单端 I/O 标准
- LVDS

Arria 10 器件使用 LVDS 输入缓冲支持差分 HSTL 和差分 SSTL 的输入操作。要支持差分 HSTL 或差分 SSTL 信号的电气规范，将 LVDS I/O 标准分配到 Quartus Prime 软件的 REFCLK 管脚。

5.7.2. 混合电压参考和非电压参考 I/O 标准

每个 I/O bank 可同时支持多个 I/O 标准。以下部分提供了器件中混合非电压参考和电压参考 I/O 标准的指南。

5.7.2.1. 非电压参考 I/O 标准

如果 I/O 标准支持 V_{CCIO} 电平的 I/O bank，那么一个 I/O bank 能够同时支持具有不同 I/O 标准分配的任意数量的输入信号。

对于输出信号，单一 I/O bank 支持与 V_{CCIO} 相同电压驱动的非电压参考输出信号。由于一个 I/O bank 仅能有一个 V_{CCIO} 值，它仅能为非电压参考信号驱动该值。

例如，一个 2.5 V V_{CCIO} 设置的 I/O bank 能够支持 2.5 V 的标准输入和输出，以及仅支持 3.0 V LVCMOS 的输入。

5.7.2.2. 电压参考 I/O 标准

为了适应电压参考 I/O 标准：

- 每 Arria 10 FPGA I/O bank 包含一个专用的 V_{REF} 管脚。
- 每个 bank 只能有一个 V_{CCIO} 电压电平和一个电压参考(V_{REF})电平。

电压参考输入缓冲由 V_{CCPT} 供电。因此，在下列情况下，采用单端或者差分标准的 I/O bank 能够支持不同的电压参考标准：

- V_{REF} 是相同的电平。
- 禁用了片上并行匹配 (R_T OCT)

如果使能 R_T OCT，那么输入标准的电压和 bank 的 V_{CCIO} 必须匹配。

这一特性使您能够将电压参考输入信号布局在 2.5 V 或是更低的 V_{CCIO} 的 I/O bank 中。例如，可以将 HSTL-15 输入管脚布局在 2.5 V V_{CCIO} 的 I/O bank 中。不过，使能 R_T OCT 的电压参考输入要求 I/O bank 的 V_{CCIO} 与输入标准的电压相匹配。当 V_{CCIO} 为 2.5 V 时，HSTL-15 I/O 标准不支持 R_T OCT。

5.7.2.3. 混合电压参考和非电压参考标准概述

通过单独应用每一种规则，一个 I/O bank 能够支持电压参考管脚和非电压参考管脚。



例如：

- 使用 1.8 V V_{CCIO} 和 0.9 V V_{REF} 的 I/O bank 可以支持 SSTL-18 输入和输出，以及 1.8 V 输入和输出。
- 使用 1.5 V V_{CCIO} 和 0.75 V V_{REF} 的 I/O bank 可以支持 1.5 V 标准、1.8 V 输入 (但没有输出) 以及 1.5 V HSTL I/O 标准。

5.7.3. 指南：上电顺序期间不可驱动 I/O 管脚

Arria 10 I/O 缓冲由 V_{CC} 、 V_{CCPT} 和 V_{CCIO} 供电。

由于 Arria 10 器件不支持热插拔，上电和断电期间，不可外部驱动 I/O 管脚。这包括所有 I/O 管脚，包含 FPGA 和 HPS I/O。坚持这条指导原则：

- 避免超过 I/O 管脚电流：
 - 超过 I/O 管脚电流会影响器件的寿命和可靠性。
 - 超过 3 V I/O 管脚上的电流会损坏 Arria 10 器件。
- 实现最低的电流消耗，并避免在上电或断电期间的 I/O 故障。
- 避免在 2.5 V 或 3 V 操作中对 3 V I/O 缓冲器的永久损坏。

相关链接

[上电和断电序列](#) (第 294 页)

5.7.4. 指南：在 HPS 共享的 I/O Bank 中使用 I/O 管脚

在 Arria 10 SX 器件中，模块化 I/O bank 2K、2J 和 2I，通过一个专用的 HPS 外部存储器接口，将 HPS 连接到 SDRAM 器件。

每个 I/O bank 含有 4 个通道：

- Lane 3—IO[47..36]
- Lane 2—IO[35..24]
- Lane 1—IO[23..12]
- Lane 0—IO[11..0]

当系统中不包括任何 HPS 外部存储器接口时，可以将 Arria 10 SX 器件中的 bank 2K、2J 和 2I 用作 FPGA GPIO。

当系统中包括 HPS 外部存储器接口时，如果想要使用 FPGA GPIO 的 bank 2K、2J 和 2I 中未使用的管脚，那么需要遵循这些规范：

- Bank 2K 用于 SDRAM ECC、地址和命令信号：
 - Lane 3 用于 SDRAM ECC 信号。仅可以将该通道中所剩余的管脚用于 FPGA 输入。
 - Lanes 2、1 和 0 用于 SDRAM 地址和命令信号。可将这些通道中所剩余的管脚用于 FPGA 输入和输出。
- Bank 2J 用于 SDRAM 数据信号[31..0]，bank 2I 用于 SDRAM 数据信号[63..32]。
 - 16 位数据宽度 — bank 2J 的两个通道用于数据。仅可以将这两个数据通道所剩余的管脚用作 FPGA 输入。可将另外两个 bank 2J 的通道，以及 bank 2I 的所有通道的管脚用作 FPGA 输入和输出。
 - 32 位数据宽度 — 仅可以将 bank 2J 的所有通道所剩余的管脚用作 FPGA 输入。可将 bank 2I 的所有通道的管脚用作 FPGA 输入和输出。
 - 64 位数据宽度 — 仅可以将 bank 2J 和 2I 的所有通道所剩余的管脚用作 FPGA 输入。

5.7.5. 指南：最大化 DC 电流限制

对于 Arria 10 器件的任何数量连续的 I/O 管脚的最大化 DC 电流没有限制。

Arria 10 器件符合 V_{CCIO} Electro-Migration (EM) 规则和所有 I/O 标准驱动强度设置的 IR 降压目标 — 确保了器件生命周期的可靠性。

5.7.6. 指南：Altera LVDS SERDES IP 内核实例

在 DPA 或 soft-CDR 模式下，仅能对每个 I/O bank 例化一个 Altera LVDS SERDES IP 内核实例。

相关链接

- [Arria 10 GX 器件的模块化 I/O Bank \(第 108 页\)](#)
- [Arria 10 GT 器件的模块化 I/O Bank \(第 111 页\)](#)
- [Arria 10 SX 器件的模块化 I/O Bank \(第 112 页\)](#)

5.7.7. 指南：Soft-CDR 模式的 LVDS SERDES 管脚对

在 soft-CDR 模式中仅能使用指定 LVDS 管脚对。请参阅每个器件的管脚说明 (pinout) 文件以确定支持 soft-CDR 模式的 LVDS 管脚对。

相关链接

- [Arria 10 器件管脚输出\(Pin-Out\)](#)
提供了每种 Arria 10 器件的管脚输出文件。对于 SoC 器件，管脚输出文件也列出了 FPGA 架构和 HPS 共享的 I/O bank。
- [Soft-CDR 模式 \(第 151 页\)](#)
- [外设时钟网络 \(第 74 页\)](#)
提供了关于 PCLK 网络的详细信息。



5.7.8. 指南: Arria 10 GPIO 性能的最小化高抖动的影响

在 Arria 10 设计流程中, 按照这一指南可以最小化对 GPIO 性能的抖动影响。

- 使用 Intel PDN 工具 2.0 执行电源配送网络分析。这一分析帮助您设计一个带有必要的去耦电容的更可靠且更高效的电源配送网络。使用 Arria 10 早期功耗估算器(EPE)来确定 V_{CC} 和其它电源供应的电流要求。基于所有电源供应轨, 尤其是 V_{CC} 电源轨的电流要求来执行 PDN 分析。
- 将电压稳压器与远程检测管脚一起使用来补偿在保持内核性能的情况下, 与 PCB 相关的 DC IR 压降和 V_{CC} 电源中的器件封装。有关差分远程传感器管脚的 V_{CC} 电源的连接指南的详细信息, 请参考管脚连接指南。
- 输入时钟抖动必须符合 Arria 10 PLL 输入时钟周期到周期的抖动规范, 从而产生低 PLL 输出时钟抖动。必须提供抖动小于 120 ps 的干净时钟源。有关所建议的操作条件的详细信息, 请参考器件数据表中的 PLL 规范。
- 使用专用 PLL 时钟输出管脚来发送时钟信号, 实现更好的抖动性能。每个 I/O bank 中的 I/O PLL 支持两个专用的时钟输出管脚。可以将 PLL 专用时钟输出管脚作为 FPGA 的参考时钟源使用。为了实现最佳的抖动性能, 可以提供外部干净的时钟源。有关 PLL 专用时钟输出管脚的抖动规范的详细信息, 请参考器件数据表。
- 如果 GPIO 运行在高于 250 MHz 的频率上, 则使用匹配 I/O 校准。SSTL、HSTL、POD 和 HSUL I/O 标准是匹配的 I/O 标准。Intel 建议您将 HSUL I/O 标准用以较短走线或者互联小于两英寸的参考长度。
- 使用 Altera PHYLite 的并行接口 IP 内核, 实现 GPIO 或者源同步 I/O 接口。Intel 建议如果无法收敛 GPIO 的时序或者数据速率大于 200 Mbps 的源同步 I/O 接口, 就使用 Altera PHYLite 的并行接口 IP 内核。有关将 Altera GPIO IP 内核移植到 Altera PHYLite 的并行接口 IP 内核的指导原则, 请参考相关的信息。
- 使用小型外设时钟(SPCLK)网络。SPCLK 网络用于高速 I/O 接口, 并且提供最小的插入延迟。下面列出了时钟网络的时钟插入延迟的排列, 从最大到最小进行排列:
 - 全局时钟网络(GCLK)
 - 区域时钟网络(RCLK)
 - 大型外设时钟网络(LPCLK)
 - SPCLK

相关链接

- [Arria 10 GX、GT 和 SX 器件系列管脚连接指南](#)
- [Arria 10 器件数据手册](#)
- [GPIO 至 PHYLite 设计移植指南](#)

5.7.9. 指南: 外部存储器接口 I/O Bank 2A 的使用

除了通用 I/O 用法之外, Arria 10 器件也将 I/O bank 2A 用于器件配置相关的操作。由于与配置相关的使用, 因此将 I/O bank 2A 用于外部存储器接口时必须遵循几条准则。



- 即使配置完成，也不要将配置相关操作需要的 I/O bank 2A 的管脚用作外部存储器接口管脚。
例如：
 - 用于 Fast Passive Parallel (FPP，快速被动并行) 配置总线的管脚
 - 用于 Partial Reconfiguration (部分重配置) 控制信号的管脚
- 确保外部存储器接口 I/O 电压与配置 I/O 电压兼容。
- 运行 Quartus Prime Fitter 确定您器件中外部存储器接口的管脚布局是否有效。

关于配置管脚的更多信息，请参阅器件 pin-out 文件中的"配置功能" (Configuration Function) 列。

相关链接

- [Arria 10 器件管脚输出\(Pin-Out\)](#)
提供了每种 Arria 10 器件的管脚输出文件。对于 SoC 器件，管脚输出文件也列出了 FPGA 架构和 HPS 共享的 I/O bank。
- [配置方案](#) (第 211 页)
- [器件配置管脚](#) (第 240 页)
- [用于配置管脚的 I/O 标准和驱动强度](#) (第 241 页)
- [Arria 10 器件封装支持的存储器接口](#) (第 182 页)

5.8. 文档修订历史

日期	版本	修订内容
2017 年 5 月	2017.05.08	<ul style="list-style-type: none">• 更新了纵向移植表，移除了 Arria 10 GX 和 Arria 10 SX 器件系列之间的纵向移植。• 更新了 External PLL 模式的 LVDS 接口，阐明 IP 内核参数编辑器的 Clock Resource Summary 选项卡提供了 IP 内核中所需信号的详细信息。• 更新了列出 I/O 缓冲器类型和 I/O 标准支持的可编程 IOE 功能的表格。• 删除了全部的"初始化"标记。
2017 年 3 月	2017.03.15	重命名为 Intel。
2016 年 10 月	2016.10.31	<ul style="list-style-type: none">• 添加了有关在 Quartus Prime 软件中没有专门分配电流强度情况下的默认预定义电流强度的信息。• 更新了有关 OCT 校准模块的主题，验证通过使用同一 I/O 列的任何 I/O bank 中的 OCT 校准模块可以校准 OCT 的主题。• 移除 Arria 10 GX 器件系列产品中的 F36 封装。• 更新了有关 non-DPA 模式的接收器偏移裕量的主题，声明 TCCS 和 RCCS 在计算 RSKM 值的使用情况。• 更新了有关上电顺序期间不可驱动 I/O 管脚以强调超过 I/O 管脚电流会影响器件的可靠性并损坏器件的指南。
6 月 13 日	2016.06.13	<ul style="list-style-type: none">• 更新了 I/O 纵向移植图，添加了 SX 570 和 SX 660 器件的 KF40 封装。• 更新了列出 I/O 标准电压电平的表格，添加 2.5 V 输入到 3.0 V LVTTTL/3.0 V LVCMOS，以及添加 3.0 V 输入到 2.5 V LVCMOS。
2016 年 5 月	2016.05.02	<ul style="list-style-type: none">• 移除了 Arria 10 GT 器件系列产品中的 NF40 和 UF45 封装。• 通过从 NF45 至 SF45 更新封装，更正了 Arria 10 GT 1150 器件的模块化 I/O bank 的信息。• 更新了列出 I/O 标准的表格，以阐明 SSTL-12、SSTL-125、STL-135、差分 SSTL-12、差分 SSTL-125 和差分 SSTL-135 I/O 标准支持的 Class I 和 Class II。• 更正了列出可编程 IOE 功能的表格，以移除 3 V I/O bank 支持的差分输出电压。
继续...		



日期	版本	修订内容
		<ul style="list-style-type: none"> 更新了可编程电流强度列表，以添加对 SSTL-135、SSTL-125、SSTL-12、POD-12、差分 SSTL-135、差分 SSTL-125、差分 SSTL-12 和差分 POD12 I/O 标准的支持。 对 SSTL-12 和差分 SSTL-12 I/O 标准添加了 120 Ω OCT 选项。 添加了有关使用超过 24 个通道的时钟 DPA 接口的指南。 添加了有关 I/O PLL 参考时钟源的指南。 添加了有关 I/O PLL 参考时钟输入管脚支持的 I/O 标准的指南。 添加了有关在 HPS 共享的 I/O bank 中使用 I/O 管脚的指南。 更新了最大化 DC 电流限制指南主题以指定对任何数量连续的 I/O 管脚没有限制。 更新了有关与 external PLL 模式一起使用 LVDS 接口的主题。更新添加了有关使用跨越多个 bank 的发送器通道以及共享 DPA 和 soft-CDR 模式的接收器通道的实例和连接图。 移除了使用外部存储器接口的 I/O bank 2A 的限制，并添加了使用外部存储器接口的 I/O bank 2A 的指南。
2015 年 12 月	2015.12.14	<ul style="list-style-type: none"> 对列出 I/O 标准电压支持的表格进行更新，将 3.0 V V_{CCIO} 输入从 2.5 V I/O 标准中移除。 更新了有关 MultiVolt I/O 接口的内容，将 V_{CCP} 更新为 V_{CC}。 更正了概述和可编程 IOE 功能表格中开漏输出、总线保持和弱上拉电阻功能所支持的 I/O 标准。 更新了有关数据重对齐模块(位滑)的内容，以指定有效数两据在 rx_bitslip_ctrl 的上升沿后的四个并行时钟周期后可用。之前，有效数据在两个并行时钟周期后才可用。 更新了有关器件的外部 I/O 匹配的内容以添加关于将 OCT 用于 SSTL-12 和差分 SSTL-12 I/O 标准的附注，并注释执行 IBIS 或 SPICE 仿真的建议。 更新了有关未校准的 R_S OCT 的内容： <ul style="list-style-type: none"> 更新 SSTL-15 的 R_S 值，移除 25 Ω 和 50 Ω。 添加了差分 SSTL-15、差分 SSTL-135、差分 SSTL-125、差分 SSTL-12、差分 POD12 和差分 HSUL-12 I/O 标准。 更新了有关校准的 R_S OCT 的内容以添加差分 POD12 I/O 标准。 更新了有关校准的 R_T OCT 的内容以移除 20 Ω R_T OCT 支持以及添加差分 POD12 I/O 标准。 从列出 SERDES 接收器和发送器 I/O 标准支持的表格中移除了差分 SSTL-2 Class I 和 Class II I/O 标准。 更新了有关混合电压参考和非电压参考 I/O 标准的电压参考 I/O 标准的内容。 添加了设计指南以最小化 GPIO 性能上的高抖动影响。 对下面的信号名称进行了更新：将 <ul style="list-style-type: none"> dpa_diffioclck 更新成 dpa_fast_clock 将 dpa_load_en 更新成 dpa_load_enable
2015 年 11 月	2015.11.02	<ul style="list-style-type: none"> 更新了有关 SDR 和 DDR 的串化器旁路操作的内容，以指定串化器旁路通过 Altera GPIO IP 内核受到支持。 在有关 DPA 模块的内容中添加了单位间隔(UI)定义的附注。 更新了有关数据重对齐模块(位滑)的内容。这个位滑翻转值现在自动设置成解串因子。 更新了有关解串器的内容以指定解串器旁路通过 Altera GPIO IP 内核受到支持。 更新了有关 PLL 和时钟的内容，将并行时钟 rx_outclock 和 tx_outclock 更正成 rx_coreclock 和 tx_coreclock。 更新了将整数模式中的 PLL 用于 LVDS 的内容，以声明 I/O PLL 操作仅在整数模式下进行。

继续...



日期	版本	修订内容
		<ul style="list-style-type: none">对下面的端口/信号名称进行了更新：将<ul style="list-style-type: none">rx_dpll_hold 更新成 rx_dpa_holdrx_reset 更新成 rx_dpa_resetrx_channel_data_align 更新成 rx_bitslip_ctrlrx_cda_max 更新成 rx_bitslip_maxrx_outclock 更新成 rx_coreclocklvds_diffioclk 和 diffioclk 更新成 fast_clocklvds_load_en 和 load_en 更新成 load_enable对差分通道的管脚布局的内容进行了更新：<ul style="list-style-type: none">对 PLL 驱动交错的差分发送器和 DPA-enabled 接收器通道提高清晰度。移除了有关 bank 布局和 SDR I/O 的附注。更新了有关 external PLL 模式下 Altera IOPLL 和 Altera LVDS SERDES IP 内核之间的信号接口的内容。更新了有关 external PLL 模式下 Altera IOPLL IP 内核参数值的内容：<ul style="list-style-type: none">outclk0 的相移从 -180° 更新成 180°outclk2 的相移从 -180/串行因子更新成 180/串行因子 (-18° 更新成 18°)在有关 non-DPA 模式下接收器偏移裕量的内容中更新了 RSKM 方程的 RSKM 定义。将 Quartus II 实例更改成 Quartus Prime。
2015 年 6 月	2015.06.15	更正纵向移植图中 Arria 10 GT 产品系列的标签。
2015 年 5 月	2015.05.04	<ul style="list-style-type: none">更新了有关 I/O 和差分 I/O 缓冲器主题的声明以提高清晰度。更新了 Arria 10 GX 160、GX 220、SX 160 和 SX 220 器件的 U19 封装的 I/O 资源信息：<ul style="list-style-type: none">更新了 LVDS I/O 数据，从 144 更新至 148更新了 GPIO 的总数，从 192 至 196更新了 LVDS 通道的数量，从 72 至 74在图中添加了 bank 3A，并移除了 bank 3C 以及相关的模块化 I/O bank 表更新了显示 IOE 结构的图，以阐明延迟链是分开的。针对 Arria 10 GX 270、GX 320、SX 270 和 SX 320 器件的 F27 封装，更新了 banks 3A (从 null 至 48) 以及 3B (从 48 至 null) 的模块化 I/O。
2015 年 1 月	2014.01.23	<ul style="list-style-type: none">添加了有关可编程开漏输出的主题。重新构建有关差分通道的管脚布局的主题以增加清晰度。更正了指定 DPA-enabled 发送器通道的内容。发送器通道没有 DPA 模式。添加了有关对每个 I/O bank 仅例化一个 Altera LVDS SERDES IP 内核实例的指南。添加了有关在 soft-CDR 模式中仅使用指定的 LVDS 管脚对的指南。更新了介绍 external PLL 的 LVDS 接口的使用部分：<ul style="list-style-type: none">更新了有关 Altera IOPLL 和 Altera LVDS SERDES IP 内核中所需信号的信息。更新了实例的参数值，使用 Altera IOPLL IP 内核生成输出时钟。更新了 external PLL 接口信号的 LVDS 时钟相位关系图。更新了显示 Altera IOPLL 和 Altera LVDS SERDES IP 内核之间的连接的结构图。添加了脚注以声明可对 LVDS 和 POD12 I/O 标准使用预加重。POD12 I/O 标准支持 DDR4。
2014 年 8 月	2014.08.18	<ul style="list-style-type: none">更新了有关可编程 IOE 功能支持的 3 V I/O bank 的说明。添加了说明以声明除了 FPGA I/O 缓冲器，Arria 10 SoC 器件也含有支持差分 I/O 标准的 HPS I/O 缓冲器。在每个 I/O bank 位置图中分开 I/O bank 2A，来表示它没有连续其它的 I/O bank。更新了 LVDS I/O 和 SERDES 电路说明来阐明每个 LVDS 通道都含有内置的发送器 SERDES 和接收器 SERDES。

继续...



日期	版本	修订内容
		<ul style="list-style-type: none"> 移除了片上钳位二极管的参考。Arria 10 器件没有片上钳位二极管。使用外部钳位二极管(如果适用)。 添加了相关信息链接到 Arria 10 收发器 PHY 用户指南，介绍了收发器 I/O bank 的位置。 更新了 I/O 纵向移植图以显示 Arria 10 GX 和 Arria 10 SX 器件之间的纵向移植。 将"宏功能"的所有参考更新为"IP 内核"。 将"MegaWizard Plug-in Manager"的所有参考更新为"参数编辑器"。 将 Altera PLL IP 内核的所有参考更新为 Altera IOPLL IP 内核。 对于将 LVDS 接口与 External PLL 模式一起使用的信号名称进行了更新：将 <ul style="list-style-type: none"> tx_inclock 和 rx_inclock 更新成 ext_fclk 将 tx_enable 和 rx_enable 更新成 ext_loaden 将 rx_dpaclock 更新成 ext_vcoph[7..0] 将 rx_synclock 更新成 ext_coreclock
2013 年 12 月	2013.12.02	首次发布。

6. Arria 10 器件的外部存储器接口

Arria 10 外部存储器接口高效的体系结构使您能够适配小模块化 I/O bank 结构中广泛的宽外部存储器接口。这种功能使能了对高水平系统带宽的支持。

与上一代 Arria 器件相比，新的体系结构和解决方案提供了以下优势：

- 在控制器以及从控制器到 PHY 预收敛时序。
- 简易的管脚布局。

为了最佳性能和灵活性，体系结构对主接口提供硬核存储控制器和硬核 PHY。

相关链接

- [Arria 10 器件手册：已知问题](#)
列出了对 *Arria 10 器件手册* 章节规划的更新。
- [Arria 10 FPGA 和 SoC 外部存储器资源](#)
提供关于 Arria 10 外部存储器解决方案的更多资源。
- [外部存储器接口规范估算器](#)
提供可以查找的参数的工具，并对比 Intel FPGA 中所支持的外部存储器接口的性能。

6.1. Arria 10 外部存储器接口解决方案的关键特性

- 该解决方案对多种协议提供完全的硬核外部存储器接口。
- 在内核架构中而不是在器件外设的 I/O bank 中混合的 I/O 列的器件功能。
- 单一硬核 Nios® II 模块校准 I/O 列中的所有存储器接口。
- I/O 列由 I/O 模块(称为 I/O bank)相结合而成。
- 每个 I/O bank 包含专用的整数 PLL (IO_PLL)、硬核存储控制器和延迟锁相环。
- PHY 时钟树与上一代 Arria 器件相比较短，并且仅跨越一个 I/O bank。
- 跨越多个 I/O bank 的接口要求多个 PLL 使用一个平衡的参考时钟网络。

相关链接

[Arria 10 器件的外部存储器接口体系结构](#) (第 197 页)
提供了关于 I/O 列和 I/O bank 体系结构的详细信息。

6.2. Arria 10 器件支持的存储器标准

I/O 旨在对现有和即将推出的外部存储器标准提供高性能支持。



表 70. 硬核存储控制器支持的存储器标准

该表列出了硬核存储控制器的综合性能。关于具体信息，请参考外部存储器接口规范评估和 Arria 10 器件数据表。

存储器标准	速率支持	Ping Pong PHY 支持	最大频率 (MHz)
DDR4 SDRAM	1/4 速率	Yes	1,067
		—	1,200
DDR3 SDRAM	1/2 速率	Yes	533
		—	667
	1/4 速率	Yes	1,067
		—	1,067
DDR3L SDRAM	1/2 速率	Yes	533
		—	667
	1/4 速率	Yes	933
		—	933
LPDDR3 SDRAM	1/2 速率	—	533
	1/4 速率	—	800

表 71. 软核存储控制器支持的存储器标准

存储器标准	速率支持	最大频率 (MHz)
RLDRAM 3 ⁽¹⁰⁾	1/4 速率	1,200
QDR IV SRAM ⁽¹⁰⁾	1/4 速率	1,067
QDR II SRAM	全速率	333
	1/2 速率	333
QDR II+ SRAM	全速率	333
	1/2 速率	550
QDR II+ Xtreme SRAM	全速率	333
	1/2 速率	633

表 72. HPS 硬核存储控制器支持的存储器标准

硬核处理器系统(HPS)仅在 Arria 10 SoC 器件中可用。

存储器标准	速率支持	最大频率 (MHz)
DDR4 SDRAM	1/2 速率	1,200
DDR3 SDRAM	1/2 速率	1,067
DDR3L SDRAM	1/2 速率	933

⁽¹⁰⁾ Arria 10 器件通过使用带软核存储控制器的硬核 PHY 来支持此外部存储器接口。

相关链接

- [外部存储器接口规范估算器](#)
提供可以查找的参数的工具，并对比 Intel FPGA 中所支持的外部存储器接口的性能。
- [Ping Pong PHY IP \(第 197 页\)](#)
提供了 Ping Pong PHY 的简介。
- [Arria 10 器件数据表 - 硬核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。
- [Arria 10 器件数据表 - 软核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。

6.3. Arria 10 器件的外部存储器接口宽度

Arria 10 器件可支持以下外部存储器接口宽度：

- DDR4 和 DDR3 支持多达 x144 个接口
- RLDRAM 3 和 QDR II+ Xtreme 支持多达 x72 个接口

表 73. 接口宽度要求的 I/O Bank

该表列出了需要支持不同外部存储器接口宽度的 I/O 的数量。必须使用相同 I/O 列中的 I/O bank 来实现每个单一的存储器接口。

该表是一种指南，它代表这些接口宽度最坏的情况。某些接口可以通过较少的 I/O 来实现，并且不会占用全部的 I/O bank。

除了 DDR4 接口，如果 address/command 管脚超过 36 个，就需要比表中所列的数量多一个 I/O bank。对于 DDR4 接口，如果 address/command 管脚超过 37 个，就需要额外的 I/O bank。

接口宽度	I/O Bank 的数量
x8	1
x16, x24, x32, x40	2
x48, x56, x64, x72	3
x80, x88, x96, x104	4
x112, x120, x128, x136	5
x144	6

6.4. Arria 10 器件的外部存储器接口 I/O 管脚

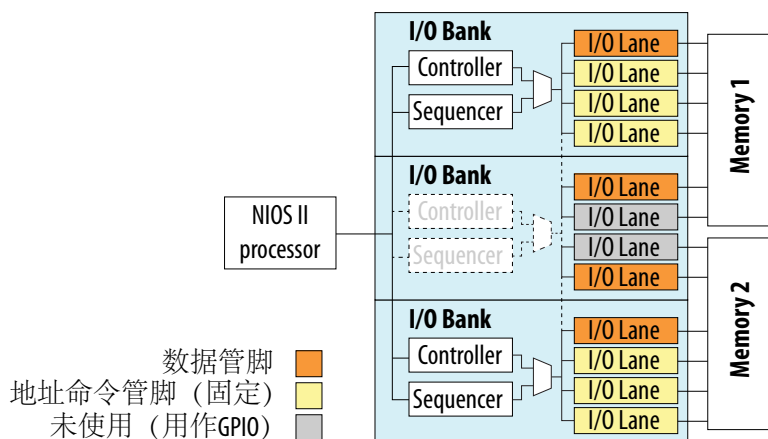
存储器接口电路可用于每个 I/O bank。Arria 10 器件功能对差分读取数据选通和时钟操作采用差分输入缓冲器。

I/O bank 中的控制器和定序器仅可以驱动 address command (A/C)管脚来固定位于相同 I/O bank 的 I/O 通道。A/C 管脚的最小要求是三个通道。不过，I/O bank 的控制器和定序器可以驱动数据组到相邻 I/O bank 的 I/O 通道(上面和下面)。

不用于存储器接口功能的管脚可用作通用 I/O (GPIO)管脚。

图 125. I/O Bank 接口共享

该表显示了两个 x16 接口共享三个 I/O bank 的实例。



相关链接

[Arria 10 器件的外部存储器接口体系结构](#) (第 197 页)
提供了关于 I/O 列和 I/O bank 体系结构的详细信息。

6.4.1. 指南：外部存储器接口 I/O Bank 2A 的使用

除了通用 I/O 用法之外，Arria 10 器件也将 I/O bank 2A 用于器件配置相关的操作。由于与配置相关的使用，因此将 I/O bank 2A 用于外部存储器接口时必须遵循几条准则。

- 即使配置完成，也不要将配置相关操作需要的 I/O bank 2A 的管脚用作外部存储器接口管脚。例如：
 - 用于 Fast Passive Parallel (FPP，快速被动并行) 配置总线的管脚
 - 用于 Partial Reconfiguration (部分重配置) 控制信号的管脚
- 确保外部存储器接口 I/O 电压与配置 I/O 电压兼容。
- 运行 Quartus Prime Fitter 确定您器件中外部存储器接口的管脚布局是否有效。

关于配置管脚的更多信息，请参阅器件 pin-out 文件中的“配置功能” (Configuration Function) 列。

相关链接

- [Arria 10 器件管脚输出\(Pin-Out\)](#)
提供了每种 Arria 10 器件的管脚输出文件。对于 SoC 器件，管脚输出文件也列出了 FPGA 架构和 HPS 共享的 I/O bank。
- [配置方案](#) (第 211 页)
- [器件配置管脚](#) (第 240 页)
- [用于配置管脚的 I/O 标准和驱动强度](#) (第 241 页)
- [Arria 10 器件封装支持的存储器接口](#) (第 182 页)

6.5. Arria 10 器件封装支持的存储器接口

注意: I/O bank 中 I/O 管脚的数量, 以及 I/O bank 的可用性, 在器件封装中有所不同。只有 48 个 I/O 管脚的 I/O bank 对外部存储器接口可用。有关可用于每种器件封装的 I/O bank 以及 I/O bank 连续位置的详细信息, 请参考相关信息。

[含有 ECC 的 DDR3 x40 在 Arria 10 中的封装支持 \(第 183 页\)](#)

[Single 和 Dual-Rank 的 DDR3 x72\(含有 ECC\)在 Arria 10 中的封装支持 \(第 185 页\)](#)

[含有 ECC 的 DDR4 x40 在 Arria 10 中的封装支持 \(第 187 页\)](#)

[Single-Rank 含有 ECC 的 DDR4 x72 在 Arria 10 中封装支持 \(第 189 页\)](#)

[Dual-Rank 含有 ECC 的 DDR4 x72 在 Arria 10 中的封装支持 \(第 191 页\)](#)

[Arria 10 的 HPS 外部存储器接口连接 \(第 192 页\)](#)

相关链接

- [Arria 10 器件的 GPIO Bank、SERDES 和 DPA 位置 \(第 100 页\)](#)
- [Arria 10 GX 器件的模块化 I/O Bank \(第 108 页\)](#)
- [Arria 10 GT 器件的模块化 I/O Bank \(第 111 页\)](#)
- [Arria 10 SX 器件的模块化 I/O Bank \(第 112 页\)](#)
- [指南: 外部存储器接口 I/O Bank 2A 的使用 \(第 173 页\)](#)



6.5.1. 含有 ECC 的 DDR3 x40 在 Arria 10 中的封装支持

要支持一个含有 ECC 的 DDR3 x40 接口(32 位数据 + 8 位数据), 需要两个 I/O bank。

表 74. 每种器件封装支持的 **DDR3 x40** 接口(含有 **ECC**)的数量(不包含 **HPS** 实例)

注意: 对于一些器件封装, 可以将 3 V I/O bank 用于外部存储器接口。不过, 最大化存储器接口时钟频率被限制在 533 MHz。要使用更高的存储器接口时钟频率, 从外部存储器接口中排除 3 V I/O bank。

产品系列	封装										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
GX 160	1	1	2	—	—	—	—	—	—	—	—
GX 220	1	1	2	—	—	—	—	—	—	—	—
GX 270	—	1	2	3	3	—	—	—	—	—	—
GX 320	—	1	2	3	3	—	—	—	—	—	—
GX 480	—	—	2	4	3	—	—	—	—	—	—
GX 570	—	—	—	4	3	5	6 ⁽¹¹⁾	—	—	—	—
GX 660	—	—	—	4	3	5	6 ⁽¹¹⁾	—	—	—	—
GX 900	—	—	—	4	—	5	—	1	7	6	4
GX 1150	—	—	—	4	—	5	—	1	7	6	4
GT 900	—	—	—	—	—	—	—	—	—	6	—
GT 1150	—	—	—	—	—	—	—	—	—	6	—
SX 160	1 ⁽¹²⁾	1 ⁽¹²⁾	2 ⁽¹²⁾	—	—	—	—	—	—	—	—
SX 220	1 ⁽¹²⁾	1 ⁽¹²⁾	2 ⁽¹²⁾	—	—	—	—	—	—	—	—
SX 270	—	1 ⁽¹²⁾	2 ⁽¹²⁾	3 ⁽¹²⁾	3 ⁽¹²⁾	—	—	—	—	—	—
SX 320	—	1 ⁽¹²⁾	2 ⁽¹²⁾	3 ⁽¹²⁾	3 ⁽¹²⁾	—	—	—	—	—	—
SX 480	—	—	2 ⁽¹²⁾	4 ⁽¹²⁾	3 ⁽¹²⁾	—	—	—	—	—	—
SX 570	—	—	—	4 ⁽¹²⁾	3 ⁽¹²⁾	5 ⁽¹²⁾	6 ⁽¹¹⁾ (12)	—	—	—	—
SX 660	—	—	—	4 ⁽¹²⁾	3 ⁽¹²⁾	5 ⁽¹²⁾	6 ⁽¹¹⁾ (12)	—	—	—	—

⁽¹¹⁾ 这个数字包括将 3 V I/O bank 用于外部存储器接口。否则, 外部存储器接口的数量可能被减至 1 个。

⁽¹²⁾ 这个数字包括 HPS 共享的 I/O bank 来实现内核 EMIF 配置。

表 75. 每种器件封装支持的 DDR3 x40 接口(含有 ECC)的数量(包含 HPS 实例)

该表显示的所支持的接口数量不包括 HPS 连接到外部 SDRAM 的接口。FPGA 内核中的主端口通过 HPS 中可配置的 FPGA-to-SDRAM 桥接端口可以访问 HPS 连接的外部存储器接口。

注意: 对于一些器件封装，可以将 3 V I/O bank 用于外部存储器接口。不过，最大化存储器接口时钟频率被限制在 533 MHz。要使用更高的存储器接口时钟频率，从外部存储器接口中排除 3 V I/O bank。

产品系列	封装										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
SX 160	0	0	1	—	—	—	—	—	—	—	—
SX 220	0	0	1	—	—	—	—	—	—	—	—
SX 270	—	0	1	2	2	—	—	—	—	—	—
SX 320	—	0	1	2	2	—	—	—	—	—	—
SX 480	—	—	1	3	2	—	—	—	—	—	—
SX 570	—	—	—	3	2	4	4 ⁽¹³⁾	—	—	—	—
SX 660	—	—	—	3	2	4	4 ⁽¹³⁾	—	—	—	—

相关链接

- [器件种类和应用](#)
提供了关于器件封装，例如：类型、大小和管脚数的详细信息。
- [Arria 10 器件数据表 - 硬核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。
- [Arria 10 器件数据表 - 软核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。

⁽¹³⁾ 这个数字包括将 3 V I/O bank 用于外部存储器接口。否则，外部存储器接口的数量可能被减至 1 个。



6.5.2. Single 和 Dual-Rank 的 DDR3 x72(含有 ECC)在 Arria 10 中的封装支持

要支持 single 和 dual-rank 含有 ECC (64 位数据 + 8 位 ECC)的 DDR3 x72 接口，需要三个 I/O bank。

表 76. 每种器件封装支持的 Single 和 Dual-rank 的 DDR3 x72 接口(含有 ECC)的数量 (不包含 HPS 实例)

注意: 对于一些器件封装，可以将 3 V I/O bank 用于外部存储器接口。不过，最大化存储器接口时钟频率被限制在 533 MHz。要使用更高的存储器接口时钟频率，从外部存储器接口中排除 3 V I/O bank。

产品系列	封装										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
GX 160	1 ⁽¹⁴⁾	1 ⁽¹⁴⁾	1 ⁽¹⁴⁾	—	—	—	—	—	—	—	—
GX 220	1 ⁽¹⁴⁾	1 ⁽¹⁴⁾	1 ⁽¹⁴⁾	—	—	—	—	—	—	—	—
GX 270	—	1 ⁽¹⁴⁾	2 ⁽¹⁴⁾	2 ⁽¹⁴⁾	2 ⁽¹⁴⁾	—	—	—	—	—	—
GX 320	—	1 ⁽¹⁴⁾	2 ⁽¹⁴⁾	2 ⁽¹⁴⁾	2 ⁽¹⁴⁾	—	—	—	—	—	—
GX 480	—	—	2 ⁽¹⁴⁾	3 ⁽¹⁴⁾	2 ⁽¹⁴⁾	—	—	—	—	—	—
GX 570	—	—	—	3 ⁽¹⁴⁾	2 ⁽¹⁴⁾	3 ⁽¹⁴⁾	3	—	—	—	—
GX 660	—	—	—	3 ⁽¹⁴⁾	2 ⁽¹⁴⁾	3 ⁽¹⁴⁾	3	—	—	—	—
GX 900	—	—	—	3	—	3	—	0	4	3	2
GX 1150	—	—	—	3	—	3	—	0	4	3	2
GT 900	—	—	—	—	—	—	—	—	—	3	—
GT 1150	—	—	—	—	—	—	—	—	—	3	—
SX 160	1 ⁽¹⁴⁾ (15)	1 ⁽¹⁴⁾ (15)	1 ⁽¹⁴⁾ (15)	—	—	—	—	—	—	—	—
SX 220	1 ⁽¹⁴⁾ (15)	1 ⁽¹⁴⁾ (15)	1 ⁽¹⁴⁾ (15)	—	—	—	—	—	—	—	—
SX 270	—	1 ⁽¹⁴⁾ (15)	2 ⁽¹⁴⁾ (15)	2 ⁽¹⁴⁾ (15)	2 ⁽¹⁴⁾ (15)	—	—	—	—	—	—
SX 320	—	1 ⁽¹⁴⁾ (15)	2 ⁽¹⁴⁾ (15)	2 ⁽¹⁴⁾ (15)	2 ⁽¹⁴⁾ (15)	—	—	—	—	—	—
SX 480	—	—	2 ⁽¹⁴⁾ (15)	3 ⁽¹⁴⁾ (15)	2 ⁽¹⁴⁾ (15)	—	—	—	—	—	—
SX 570	—	—	—	3 ⁽¹⁴⁾ (15)	2 ⁽¹⁴⁾ (15)	3 ⁽¹⁴⁾ (15)	3 ⁽¹⁵⁾	—	—	—	—
SX 660	—	—	—	3 ⁽¹⁴⁾ (15)	2 ⁽¹⁴⁾ (15)	3 ⁽¹⁴⁾ (15)	3 ⁽¹⁵⁾	—	—	—	—

⁽¹⁴⁾ 这个数字包括将 3 V I/O bank 用于外部存储器接口。否则，外部存储器接口的数量可能被减至 1 个。

⁽¹⁵⁾ 这个数据包括 HPS 共享的 I/O bank 来实现内核 EMIF 配置。

表 77. 每种器件封装支持的 Single 和 Dual-rank 的 DDR3 x72 接口(含有 ECC)的数量 (包含 HPS 实例)

该表显示的所支持的接口数量不包括 HPS 连接到外部 SDRAM 的接口。FPGA 内核中的主端口通过 HPS 中可配置的 FPGA-to-SDRAM 桥接端口可以访问 HPS 连接的外部存储器接口。

注意: 对于一些器件封装，可以将 3 V I/O bank 用于外部存储器接口。不过，最大化存储器接口时钟频率被限制在 533 MHz。要使用更高的存储器接口时钟频率，从外部存储器接口中排除 3 V I/O bank。

产品系列	封装										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
SX 160	0	0	0	—	—	—	—	—	—	—	—
SX 220	0	0	0	—	—	—	—	—	—	—	—
SX 270	—	0	1 ⁽¹⁶⁾	1 ⁽¹⁶⁾	1 ⁽¹⁶⁾	—	—	—	—	—	—
SX 320	—	0	1 ⁽¹⁶⁾	1 ⁽¹⁶⁾	1 ⁽¹⁶⁾	—	—	—	—	—	—
SX 480	—	—	1 ⁽¹⁶⁾	2 ⁽¹⁶⁾	1 ⁽¹⁶⁾	—	—	—	—	—	—
SX 570	—	—	—	2 ⁽¹⁶⁾	1 ⁽¹⁶⁾	2 ⁽¹⁶⁾	2	—	—	—	—
SX 660	—	—	—	2 ⁽¹⁶⁾	1 ⁽¹⁶⁾	2 ⁽¹⁶⁾	2	—	—	—	—

相关链接

- [器件种类和应用](#)
提供了关于器件封装，例如：类型、大小和管脚数的详细信息。
- [Arria 10 器件数据表 - 硬核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。
- [Arria 10 器件数据表 - 软核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。

⁽¹⁶⁾ 这个数字包括将 3 V I/O bank 用于外部存储器接口。否则，外部存储器接口的数量可能被减至 1 个。



6.5.3. 含有 ECC 的 DDR4 x40 在 Arria 10 中的封装支持

要支持一个含有 ECC 的 DDR4 x40 接口(32 位数据 + 8 位 ECC)，需要两个 I/O bank。

表 78. 每种器件封装支持的 DDR4 x40 接口(含有 ECC)的数量(不包含 HPS 实例)

产品系列	封装										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
GX 160	1	1	2	—	—	—	—	—	—	—	—
GX 220	1	1	2	—	—	—	—	—	—	—	—
GX 270	—	1	2	3	3	—	—	—	—	—	—
GX 320	—	1	2	3	3	—	—	—	—	—	—
GX 480	—	—	2	4	3	—	—	—	—	—	—
GX 570	—	—	—	4	3	5	5	—	—	—	—
GX 660	—	—	—	4	3	5	5	—	—	—	—
GX 900	—	—	—	4	—	5	—	1	7	6	4
GX 1150	—	—	—	4	—	5	—	1	7	6	4
GT 900	—	—	—	—	—	—	—	—	—	6	—
GT 1150	—	—	—	—	—	—	—	—	7	6	—
SX 160	1 ⁽¹⁷⁾	1 ⁽¹⁷⁾	2 ⁽¹⁷⁾	—	—	—	—	—	—	—	—
SX 220	1 ⁽¹⁷⁾	1 ⁽¹⁷⁾	2 ⁽¹⁷⁾	—	—	—	—	—	—	—	—
SX 270	—	1 ⁽¹⁷⁾	2 ⁽¹⁷⁾	3 ⁽¹⁷⁾	3	—	—	—	—	—	—
SX 320	—	1 ⁽¹⁷⁾	2 ⁽¹⁷⁾	3 ⁽¹⁷⁾	3 ⁽¹⁷⁾	—	—	—	—	—	—
SX 480	—	—	2	4 ⁽¹⁷⁾	3 ⁽¹⁷⁾	—	—	—	—	—	—
SX 570	—	—	—	4 ⁽¹⁷⁾	3 ⁽¹⁷⁾	5 ⁽¹⁷⁾	6 ⁽¹⁸⁾ (17)	—	—	—	—
SX 660	—	—	—	4 ⁽¹⁷⁾	3 ⁽¹⁷⁾	5 ⁽¹⁷⁾	6 ⁽¹⁸⁾ (17)	—	—	—	—

⁽¹⁷⁾ 这个数据包括 HPS 共享的 I/O bank 来实现内核 EMIF 配置。

⁽¹⁸⁾ 这个数字包括将 3 V I/O bank 用于外部存储器接口。否则，外部存储器接口的数量可能被减至 1 个。

表 79. 每种器件封装支持的 DDR4 x40 接口(含有 ECC)的数量(包含 HPS 实例)

该表显示的所支持的接口数量不包括 HPS 连接到外部 SDRAM 的接口。FPGA 内核中的主端口通过 HPS 中可配置的 FPGA-to-SDRAM 桥接端口可以访问 HPS 连接的外部存储器接口。

产品系列	封装										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
SX 160	0	0	1	—	—	—	—	—	—	—	—
SX 220	0	0	1	—	—	—	—	—	—	—	—
SX 270	—	0	1	2	2	—	—	—	—	—	—
SX 320	—	0	1	2	2	—	—	—	—	—	—
SX 480	—	—	1	3	2	—	—	—	—	—	—
SX 570	—	—	—	3	2	4	4	—	—	—	—
SX 660	—	—	—	3	2	4	4	—	—	—	—

相关链接

- [器件种类和应用](#)
提供了关于器件封装，例如：类型、大小和管脚数的详细信息。
- [DDR4 的外部存储器接口实现的实例](#)
- [Arria 10 器件数据表 - 硬核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。
- [Arria 10 器件数据表 - 软核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。



6.5.4. Single-Rank 含有 ECC 的 DDR4 x72 在 Arria 10 中封装支持

要支持 single-rank 的 DDR4 x72 接口(64 位数据 + 8 位 ECC)，需要三个 I/O bank。

表 80. 每种器件封装支持的 Single-Rank 的 DDR4 x72 接口(含有 ECC)的数量(不包含 HPS 实例)

产品系列	封装										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
GX 160	0	0	0	—	—	—	—	—	—	—	—
GX 220	0	0	0	—	—	—	—	—	—	—	—
GX 270	—	0	1	1	1	—	—	—	—	—	—
GX 320	—	0	1	1	1	—	—	—	—	—	—
GX 480	—	—	1	2	1	—	—	—	—	—	—
GX 570	—	—	—	2	1	2	3	—	—	—	—
GX 660	—	—	—	2	1	2	3	—	—	—	—
GX 900	—	—	—	3	—	3	—	0	4	3	2
GX 1150	—	—	—	3	—	3	—	0	4	3	2
GT 900	—	—	—	—	—	—	—	—	—	3	—
GT 1150	—	—	—	—	—	—	—	—	—	3	—
SX 160	0	0	0	—	—	—	—	—	—	—	—
SX 220	0	0	0	—	—	—	—	—	—	—	—
SX 270	—	0	1 ⁽¹⁹⁾	1 ⁽¹⁹⁾	1 ⁽¹⁹⁾	—	—	—	—	—	—
SX 320	—	0	1 ⁽¹⁹⁾	1 ⁽¹⁹⁾	1 ⁽¹⁹⁾	—	—	—	—	—	—
SX 480	—	—	1 ⁽¹⁹⁾	2 ⁽¹⁹⁾	1 ⁽¹⁹⁾	—	—	—	—	—	—
SX 570	—	—	—	2 ⁽¹⁹⁾	1 ⁽¹⁹⁾	2 ⁽¹⁹⁾	3 ⁽¹⁹⁾	—	—	—	—
SX 660	—	—	—	2 ⁽¹⁹⁾	1 ⁽¹⁹⁾	2 ⁽¹⁹⁾	3 ⁽¹⁹⁾	—	—	—	—

⁽¹⁹⁾ 这个数字包括 HPS 共享的 I/O bank 来实现内核 EMIF 配置。

表 81. 每种器件封装支持的 Single-Rank 的 DDR4 x72 接口(含有 ECC)的数量(包含 HPS 实例)

该表显示的所支持的接口数量不包括 HPS 连接到外部 SDRAM 的接口。FPGA 内核中的主端口通过 HPS 中可配置的 FPGA-to-SDRAM 桥接端口可以访问 HPS 连接的外部存储器接口。

产品系列	封装										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
SX 160	0	0	0	—	—	—	—	—	—	—	—
SX 220	0	0	0	—	—	—	—	—	—	—	—
SX 270	—	0	1	1	1	—	—	—	—	—	—
SX 320	—	0	1	1	1	—	—	—	—	—	—
SX 480	—	—	1	2	1	—	—	—	—	—	—
SX 570	—	—	—	2	1	2	2	—	—	—	—
SX 660	—	—	—	2	1	2	2	—	—	—	—

相关链接

- [器件种类和应用](#)
提供了关于器件封装，例如：类型、大小和管脚数的详细信息。
- [DDR4 的外部存储器接口实现的实例](#)
- [Arria 10 器件数据表 - 硬核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。
- [Arria 10 器件数据表 - 软核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。



6.5.5. Dual-Rank 含有 ECC 的 DDR4 x72 在 Arria 10 中的封装支持

要支持 dual-rank 含有 ECC 的 DDR4 x72 接口(64 位数据 + 8 位 ECC)，需要 3.25 I/O bank(在相邻的 I/O bank 三个 I/O bank 和一个 I/O 通道)。

表 82. 每种器件封装支持的 Dual-Rank 的 DDR4 x72 接口(含有 ECC)的数量(不包含 HPS 实例)

产品系列	封装										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
GX 160	0	0	0	—	—	—	—	—	—	—	—
GX 220	0	0	0	—	—	—	—	—	—	—	—
GX 270	—	0	1	1	1	—	—	—	—	—	—
GX 320	—	0	1	1	1	—	—	—	—	—	—
GX 480	—	—	1	1	1	—	—	—	—	—	—
GX 570	—	—	—	1	1	2	2	—	—	—	—
GX 660	—	—	—	1	1	2	2	—	—	—	—
GX 900	—	—	—	2	—	3	—	0	4	3	2
GX 1150	—	—	—	2	—	3	—	0	4	3	2
GT 900	—	—	—	—	—	—	—	—	—	3	—
GT 1150	—	—	—	—	—	—	—	—	—	3	—
SX 160	0	0	0	—	—	—	—	—	—	—	—
SX 220	0	0	0	—	—	—	—	—	—	—	—
SX 270	—	0	1 ⁽²⁰⁾	1 ⁽²⁰⁾	1 ⁽²⁰⁾	—	—	—	—	—	—
SX 320	—	0	1 ⁽²⁰⁾	1 ⁽²⁰⁾	1 ⁽²⁰⁾	—	—	—	—	—	—
SX 480	—	—	1 ⁽²⁰⁾	1 ⁽²⁰⁾	1 ⁽²⁰⁾	—	—	—	—	—	—
SX 570	—	—	—	1 ⁽²⁰⁾	1 ⁽²⁰⁾	2 ⁽²⁰⁾	2 ⁽²⁰⁾	—	—	—	—
SX 660	—	—	—	1 ⁽²⁰⁾	1 ⁽²⁰⁾	2 ⁽²⁰⁾	2 ⁽²⁰⁾	—	—	—	—

⁽²⁰⁾ 这个数字包括 HPS 共享的 I/O bank 来实现内核 EMIF 配置。

表 83. 每种器件封装支持的 Dual-Rank 的 DDR4 x72 接口(含有 ECC)的数量(包含 HPS 实例)

该表显示的所支持的接口数量不包括 HPS 连接到外部 SDRAM 的接口。FPGA 内核中的主端口通过 HPS 中可配置的 FPGA-to-SDRAM 桥接端口可以访问 HPS 连接的外部存储器接口。

产品系列	封装										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
SX 160	0	0	0	—	—	—	—	—	—	—	—
SX 220	0	0	0	—	—	—	—	—	—	—	—
SX 270	—	0	1	1	1	—	—	—	—	—	—
SX 320	—	0	1	1	1	—	—	—	—	—	—
SX 480	—	—	1	1	1	—	—	—	—	—	—
SX 570	—	—	—	1	1	2	2	—	—	—	—
SX 660	—	—	—	1	1	2	2	—	—	—	—

相关链接

- [器件种类和应用](#)
提供了关于器件封装，例如：类型、大小和管脚数的详细信息。
- [DDR4 的外部存储器接口实现的实例](#)
- [Arria 10 器件数据表 - 硬核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。
- [Arria 10 器件数据表 - 软核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。

6.5.6. Arria 10 的 HPS 外部存储器接口连接

必须把 Arria 10 外部存储器接口用作 HPS Qsys IP 组件，以便将外部 SDRAM 连接到 HPS。可以例化除 HPS Qsys 组件外的 Qsys 子系统中 HPS 组件的 Arria 10 外部存储器接口。必须将 HPS 组件的 EMIF 管道连接到 Arria 10 外部存储器接口的 HPS 的 EMIF 管道，以便将 HPS 连接到外部 SDRAM 存储器。

HPS 存储器接口被固定在 x40 宽度的 I/O Bank 2K 和 2J，以及 x64/x72 宽度的 2K、2J 和 2I。当外部 SDRAM 存储器连接到 HPS 时，对于 I/O bank (2K、2J、2I)中 FPGA 内核对 HPS 存储器接口的利用未使用的 I/O 的适用性有限制。

当 HPS 连接到外部 SDRAM 存储器时，没有其它的 Arria 10 外部存储器接口 IP 实例可被布局在同一 I/O 列。

相关链接

[External Memory Interface Handbook Volume 3: Reference Material - Functional Description - HPS Memory Controller](#)

有关 Arria 10 EMIF 硬核处理器子系统限制和布局的详细信息。

6.5.6.1. HPS 含有 ECC 的 DDR3 x40 在 Arria 10 中的封装支持

要支持 HPS 中含有 ECC 的一个 DDR3 x40 接口 (32 位数据 + 8 位 ECC)，需要使用 DDR 列中 3 V I/O bank 顶部下面的两个 I/O bank。



表 84. 每种器件封装支持的 HPS 的 DDR3 x40 接口(含有 ECC)的数量

该表列出了仅支持 HPS 的外部存储器接口的数量。

产品系列	封装						
	U19	F27	F29	F34	F35	NF40	KF40
SX 160	1	1	1	—	—	—	—
SX 220	1	1	1	—	—	—	—
SX 270	—	1	1	1	1	—	—
SX 320	—	1	1	1	1	—	—
SX 480	—	—	1	1	1	—	—
SX 570	—	—	—	1	1	1	1
SX 660	—	—	—	1	1	1	1

相关链接

- [器件种类和应用](#)
提供了关于器件封装，例如：类型、大小和管脚数的详细信息。
- [Arria 10 器件数据表 - 硬核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。
- [Arria 10 器件数据表 - 软核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。

6.5.6.2. HPS 的 Single 和 Dual-Rank 含有 ECC 的 DDR3 x72 在 Arria 10 中的封装支持

要支持 HPS 中含有 ECC (64 位数据 + 8 位 ECC) single 和 dual-rank 的一个 DDR3 x72 接口, 需要使用 DDR 列 3 V I/O bank 顶部下面的三个 I/O bank。

表 85. 每种器件封装支持的 HPS 的 Single 和 Dual-Rank 的 DDR3 x72 接口(含有 ECC)的数量

该表列出了仅支持 HPS 的外部存储器接口的数量。

产品系列	封装						
	U19	F27	F29	F34	F35	NF40	KF40
SX 160	0	0	0	—	—	—	—
SX 220	0	0	0	—	—	—	—
SX 270	—	0	0	0	0	—	—
SX 320	—	0	0	0	0	—	—
SX 480	—	—	0	0	0	—	—
SX 570	—	—	—	0	0	0	1
SX 660	—	—	—	0	0	0	1

相关链接

- [器件种类和应用](#)
提供了关于器件封装, 例如: 类型、大小和管脚数的详细信息。
- [Arria 10 器件数据表 - 硬核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。
- [Arria 10 器件数据表 - 软核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。

6.5.6.3. HPS 含有 ECC 的 DDR4 x40 在 Arria 10 中的封装支持

要支持 HPS 中含有 ECC 的一个 DDR4 x40 接口 (32 位数据 + 8 位 ECC), 需要使用 DDR 列中 3 V I/O bank 顶部下面的两个 I/O bank。

表 86. 每种器件封装支持的 HPS 的 DDR4x40 接口(含有 ECC)的数量

该表列出了仅支持 HPS 的外部存储器接口的数量。

产品系列	封装						
	U19	F27	F29	F34	F35	NF40	KF40
SX 160	1	1	1	—	—	—	—
SX 220	1	1	1	—	—	—	—
SX 270	—	1	1	1	1	—	—
SX 320	—	1	1	1	1	—	—
SX 480	—	—	1	1	1	—	—
SX 570	—	—	—	1	1	1	1
SX 660	—	—	—	1	1	1	1



相关链接

- [器件种类和应用](#)
提供了关于器件封装，例如：类型、大小和管脚数的详细信息。
- [Arria 10 器件数据表 - 硬核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。
- [Arria 10 器件数据表 - 软核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。

6.5.6.4. HPS 的 Single-Rank 含有 ECC 的 DDR4 x72 在 Arria 10 中的封装支持

要支持 HPS 中含有 ECC (64 位数据 + 8 位 ECC) single-rank 的一个 DDR4 x72 接口，需要使用 DDR 列 3 V I/O bank 顶部下面的三个 I/O bank。

表 87. 每种器件封装支持的 HPS 的 Single-Rank 的 DDR4 x72 接口(含有 ECC)的数量

该表列出了仅支持 HPS 的外部存储器接口的数量。

产品系列	封装						
	U19	F27	F29	F34	F35	NF40	KF40
SX 160	0	0	0	—	—	—	—
SX 220	0	0	0	—	—	—	—
SX 270	—	0	0	0	0	—	—
SX 320	—	0	0	0	0	—	—
SX 480	—	—	0	0	0	—	—
SX 570	—	—	—	0	0	0	1
SX 660	—	—	—	0	0	0	1

相关链接

- [器件种类和应用](#)
提供了关于器件封装，例如：类型、大小和管脚数的详细信息。
- [Arria 10 器件数据表 - 硬核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。
- [Arria 10 器件数据表 - 软核存储控制器支持的存储器标准](#)
对每种器件速率等级支持的存储器接口时钟频率提供信息。

6.6. Arria 10 器件支持的外部存储器接口 IP

表 88. 每种存储器标准支持的 Intel FPGA IP 的类型

该表列出了 Intel 所提供的存储控制器 IP。您也可以将自身的软核存储控制器用于 Arria 10 器件所支持的所有存储器标准。

存储器标准	定序器		硬核定序器
	硬核	软核	
DDR4 SDRAM ⁽²¹⁾	Yes	—	Yes
DDR3 SDRAM ⁽²²⁾	Yes	—	Yes
DDR3L SDRAM ⁽²²⁾	Yes	—	Yes
LPDDR3 SDRAM ⁽²³⁾	Yes	—	Yes
继续...			

⁽²¹⁾ x4/x8 DQ 组，POD12 I/O 标准，以及突发长度 BL8。

⁽²²⁾ x4/x8 DQ 组和突发长度 BL8。

⁽²³⁾ Arria 10 器件支持单组件 x32 数据使用 x8 DQ 组。

存储器标准	定序器		硬核定序器
	硬核	软核	
RLDRAM 3 ⁽²⁴⁾	—	Yes	Yes
QDR IV SRAM	—	Yes	Yes
QDR II/II+/II+ Xtreme SRAM	—	Yes	Yes

相关链接

[Arria 10 器件支持的存储器标准](#) (第 178 页)

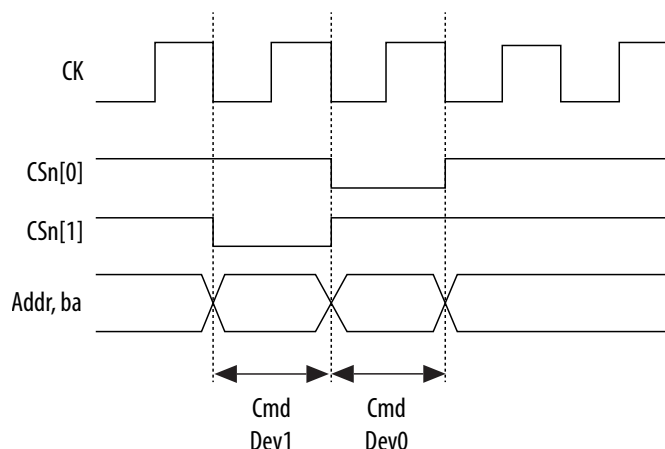
列出了 Arria 10 器件所支持的所有存储器标准。

6.6.1. Ping Pong PHY IP

Ping Pong PHY 允许两个存储器接口以共享使用时序复用的 address/command 总线。与两个独立接口相比，Ping Pong PHY 为您提供使用更少管脚的优势，并且不对吞吐量造成任何影响。

图 126. Ping Pong PHY 1T 时序

通过 Ping Pong PHY，将其中一个控制器输出延迟一个全速率时钟周期，两个独立控制器中的 address 和 command 信号被多路复用用到共享总线。导致 1T 时序，在每个全速率时钟周期发布一个新的命令。



相关链接

- [Arria 10 器件支持的存储器标准](#) (第 178 页)
- [硬核存储控制器功能](#) (第 200 页)

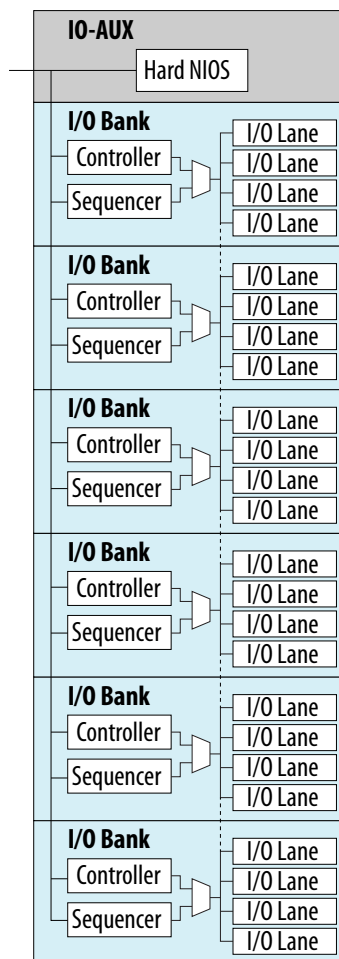
6.7. Arria 10 器件的外部存储器接口体系结构

Arria 10 外部存储器接口解决方案用于提供高性能、速率以及外部存储器接口可靠的实现。而不像上一代 Arria 器件的外设 I/O，Arria 10 器件采用 I/O 列。

⁽²⁴⁾ Arria 10 器件通过使用带软核存储控制器的硬核 PHY 来支持此外部存储器接口。

图 127. I/O 列体系结构

I/O 列由 I/O bank 和一个 I/O-AUX 模块组成。



相关链接

- [Arria 10 外部存储器接口解决方案的关键特性](#) (第 178 页)
- [Arria 10 器件的外部存储器接口 I/O 管脚](#) (第 180 页)

6.7.1. I/O Bank

硬核 IP 被组织成垂直 I/O bank。这些模块化 I/O bank 可能连接在一起形成更大的接口。

每个 I/O bank 包含下列模块：

- 嵌入式硬核控制器
- 硬核定序器
- 专用 DLL
- 整数 PLL

- OCT 校准模块
- PHY 时钟网络
- 四个 I/O 通道

6.7.1.1. 硬存储控制器

The Arria 10 hard memory controller is designed for high speed, high performance, high flexibility, and area efficiency. The hard memory controller supports all the popular and emerging memory standards including DDR4, DDR3, and LPDDR3.

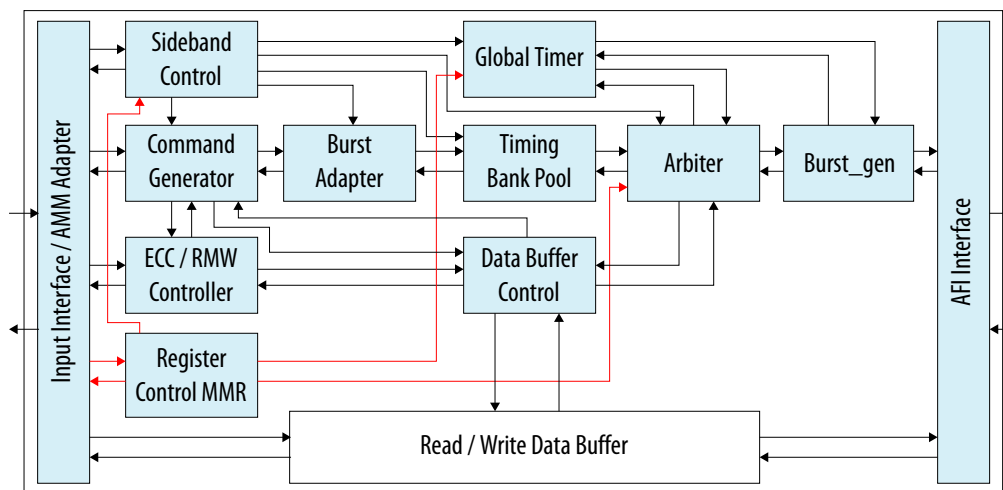
通过实现高级动态命令和数据重排序算法来实现高性能。此外，本设计还采用高效流水线技术，以提高存储器带宽的使用以及在保持高速的同时降低延迟。硬核解决方案提供最佳的可用性和较短的上市时间。控制器内部以及从控制器到 PHY 的时序因简化时序收敛已被 Intel 预先关闭。

该控制器体系结构是一种模块化设计，适用于单一 I/O bank。这种结构为硬核解决方案提供最佳灵活性。

- 可将每个 I/O bank 配置成以下路径之一：
 - 控制路径，以驱动所有用于存储器接口的 address/command 管脚
 - 数据路径，以驱动用于 DDR 型接口的最高 32 位数据管脚。
- 存储控制器可放置在任意位置。
- 可将多个 bank 打包在一起，形成不同宽度（最高 144 位）的存储器接口。

为获得更大灵活性，可旁路硬存储控制器，并根据需要使用定制 IP。

图 128. 硬存储控制器体系结构



硬存储控制器由以下逻辑块组成：

- Core 和 PHY 接口
- 主控制路径
- 数据缓冲控制器
- 读写数据缓冲器

内核接口支持 Avalon® Memory-Mapped (Avalon-MM) 接口协议。该接口与 PHY 的通信遵循 Altera PHY 接口 (AFI) 协议。整个控制路径被分成主控制路径和数据缓冲控制器。

6.7.1.1.1. 硬核存储控制器功能

表 89. Arria 10 硬核存储控制器的功能

功能	说明
存储器件支持	支持下面的存储器件： <ul style="list-style-type: none"> • DDR4 SDRAM • DDR3 SDRAM • 低功耗的 LPDDR3
存储控制器支持	<ul style="list-style-type: none"> • 定制控制器支持 — 可编程旁路模式支持旁路硬核存储控制器和使用定制控制器。 • Ping Pong 控制器 — 支持两个硬核存储控制器实例分时同套 address/command 管脚。
接口协议支持	<ul style="list-style-type: none"> • 支持 Avalon-MM 和 Avalon-ST 接口。 • PHY 接口遵循 AFI 协议。
速率支持	您可以将控制器配置成运行在 1/2 速率或 1/4 速率上。
可编程存储器接口宽度	支持宽度为 8 至 144 位，以 8 位递增。
多个 rank 支持	支持高达 4 个 rank。
突发适配器	在控制器的本地接口上能够接受 255 最大突发长度之内的任何大小的突发，并映射这些突发到有效的存储器命令中。
效率优化功能	<ul style="list-style-type: none"> • Open-page 原则 — 默认情况下，数据流量在每个访问中是关闭页(closed-page)。不过，控制器基于输入的流量，智能地保持打开一行，这提高了控制器的效率，特别是随机传输(random traffic)的效率。 • 优先购买的 bank 管脚 — 控制器能够提早发出 bank 管脚命令，当出现读写操作时，就会打开需要的行。 • 数据重排序 — 控制器重排序读/写命令。 • 附加延迟 — 该控制器在 ACTIVATE 命令后可以发出一个 READ/WRITE 命令到存储器 bank (到 t_{RCD} 之前)，这提高了命令效率。
用户请求优先级	您可以分配命令的优先级。该功能使您能够指定较早发出以减少延迟的优先级更高的命令。
Starvation 计数器	确保所有的请求都在预定义超时周期后服务，这确保在重新排序数据的效率时，低优先级访问的数据不会落在后面。
address/command 总线的时序	要最大化命令带宽，您可以在一个控制器时序周期内加倍存储器命令的数量。 <ul style="list-style-type: none"> • Quasi-1T 地址 1/2 速率 address/command 总线。 • Quasi-2T 地址 1/4 速率 address/command 总线。
Bank 交错	能够发出读写命令来'随机'选择地址。必须正确地循环 bank 地址。
片上匹配	控制器控制存储器中的片上匹配信号。该功能改善了信号的完整性，并简化了电路板设计。
刷新(Refresh)功能	<ul style="list-style-type: none"> • 用户可控的刷新时序 — 您可以选择性地控制何时发生刷新，这使您能够预防重要的读写操作在刷新的锁定时间内出现冲突。 • 每个 rank 刷新 — 支持对每个单独的 rank 进行刷新。 • 控制器控制的刷新。
ECC 支持	<ul style="list-style-type: none"> • 8 位 ECC 代码；单纠错，双错误检测(SECDED)。 • 用户 ECC 支持通过用户 ECC 位作为数据位的一部分。
继续...	



功能	说明
功耗节省功能	<ul style="list-style-type: none"> 低功耗模式(断电和自动刷新) — 您可以选择性地要求控制器将存储器放在两个低功耗状态中的其中一个。 自动断电 — 控制器处于空闲状态时, 将存储器件设置在断电模式。您可以配置空闲等待时间。 存储器时钟门控(clock gating)
模式寄存器集	访问存储器模式寄存器。
DDR4 功能	<ul style="list-style-type: none"> Bank 组支持 — 支持 bank 组之间不同的时序参数。 数据总线 CRC — 数据总线编码和解码。 Command/Address 校验 — 命令和地址总线上的奇偶校验。 警报报告 — 响应错误警报标志。 多用途寄存器访问 — 支持在串行读出模式的多用途寄存器访问。 精细粒度刷新 — 支持 1x、2x 和 4x 固定的刷新率。 温度控制刷新 — 根据温度范围调整刷新率。 低功耗自动自刷新 — 运行温度触发自动调整来自刷新率。 最大化功耗节省。
LPDDR3 功能	<ul style="list-style-type: none"> 深度断电(Deep power down)模式 — 通过消除存储器阵列的功率以实现最大化功耗节省。当器件进入深度断电模式时, 数据将无法保留。 部分阵列自刷新。 每个 bank 刷新。
ZQ 校准命令	支持 DDR3 或 DDR4 的长或短 ZQ 校准命令。

相关链接

[Ping Pong PHY IP \(第 197 页\)](#)
提供了 Ping Pong PHY 的简介。

6.7.1.1.2. 主控制路径

主控制路径执行下面功能:

- 包含命令处理流水线。
- 监控所有的时序参数。
- 跟踪存储器访问命令的相依性。
- 预防存储器访问的危害。

表 90. 主控制路径组件

组件	说明
输入接口	<ul style="list-style-type: none"> 接受来自内核逻辑(运行在 1/2 速率或 1/4 速率上)的存储器访问命令。 使用 Avalon-MM 或 Avalon-ST 协议。默认的协议是 Avalon-ST。可通过配置寄存器使能硬核适配器, 以兼容输入接口 Avalon-MM。 硬核存储控制器具有一个本地的 Avalon-ST 接口。可以例化一个标准的软核适配器, 桥接 Avalon-ST 接口到 AMBA AXI。 要支持所有的旁路模式, 并保持最小的端口数量, 超集中的所有端口列表被用作物理宽度。端口在旁路模式之间共享。
命令生成器和突发适配器	<ul style="list-style-type: none"> 从输入接口中漏极命令, 并将它们驱动到 timing bank pool。 如果需要 read-modify-write, 那么插入必要的 read-modify-write 和写命令到数据流中。 突发适配器突变任意突发长度至存储器类型指定的数量。
继续...	

组件	说明
Timing Bank Pool	<ul style="list-style-type: none"> 存储控制器中的关键组件。 设置平行队列以跟踪命令的相依性。 每个准备状态命令的信号在最终配送时被跟踪至仲裁。 大记分板结构。条目的数量目前是 8，其中它同时监控多达 8 个命令。 处理存储器访问的危害(RAW、WAR 和 WAW)，而部分时序约束被跟踪。 具有高度的责任来协助仲裁实现重新排序： <ul style="list-style-type: none"> 行指令重新排序(激活和预充电)。 列命令重新排序(读和写)。 当 pool 满时，控制信号将被送回上游来拖延流量。
仲裁	<ul style="list-style-type: none"> 强制执行仲裁规则。 执行最终仲裁从所有准备命令中选择命令，并将所选择的命令发送到存储器。 支持 1/2 速率的 quasi-1T 模式和 1/4 速率的 quasi-2T 模式。 对于 quasi 模式，行命令必须与列命令配对。
全局定时器	跟踪全局时序约束包括： <ul style="list-style-type: none"> t_{FAW} — 仅允许 4 个激活命令的指定时间周期的 Four Activates Window 参数。 t_{RRD} — 到不同 bank 的背靠背激活命令之间的延迟。 一些总线周转时间的参数。
MMR/IOCSR	<ul style="list-style-type: none"> 所有配置寄存器的主机。 使用 Avalon-MM 总线与内核进行交谈。 内核逻辑可以读写所有的配置位。 调试总线通过该模块布线到内核。
边带	执行刷新和断电功能。
DMA 控制器	虽然在软件逻辑中执行 ECC 编码和解码 ⁽²⁵⁾ ，但是 ECC 控制器在硬核解决方案中保持 read-modify-write 状态机。
AFI 接口	存储控制器通过该接口与 PHY 进行通信。

6.7.1.1.3. 数据缓冲控制器

数据缓冲控制器具有以下主要职责：

- 管理数据缓冲的读写访问：
 - 当读数据被接受或者读返回数据到达时，提供到缓冲的数据存储指示器。
 - 当写数据被分到存储器或者从缓冲器中读取读数据，并送回到用户时，提供漏极指定器。
- 满足所需的写延迟。
- 如果使能了 ECC 支持，协助主控制路径执行 read-modify-write 操作。

数据重排序使用数据缓冲控制器及数据缓冲来执行。

数据缓冲通道的每个 I/O bank 包含两个数据缓冲控制器模块，该数据缓冲通道在每个 bank 内被分开。要提高时序，可布局数据缓冲控制器在靠近 I/O 的通道。

⁽²⁵⁾ 在软件逻辑中执行 ECC 编码和解码，免除从布线数据位到中央 ECC 计算位置的硬核连接。将数据布线到中央位置，移除了模块化设计的优势，并减少了灵活性。

6.7.1.2. 延迟锁相环

延迟锁相环(DLL)查找 9 位延迟链的延迟设置，这样，该延迟链等于 1 个时钟周期。

每个 I/O bank 含有 1 个延迟锁相环(DLL)，位于中央，它支持 800 MHz 至 1.3 GHz 的频率范围。

DLL 的参考时钟来自同一 I/O bank 的 PLL 输出。该 DLL 将参考时钟划分为八，并创建两个时钟脉冲 — launch 和 measure。launch 和 measure 之间的相位差是 1 个参考时钟周期。时钟脉冲 launch 通过延迟链控制的延迟设置来布线。延迟后的 launch 之后与 measure 进行比较。

DLL 延迟链的设置来自 9 位的计数器，它向上或向下移动来改变延迟时间，直到延迟后的 launch 和 measure 对齐到同一相位。一旦 DLL 被锁定，通过延迟链的延迟等于 1 个参考时钟周期，而延迟设置被发送到 DQS 延迟模块。

6.7.1.3. 定序器

定序器通过校准接口补偿建立中的各种变化以及保持由传送延迟引起的要求，从而实现高频存储器接口操作。

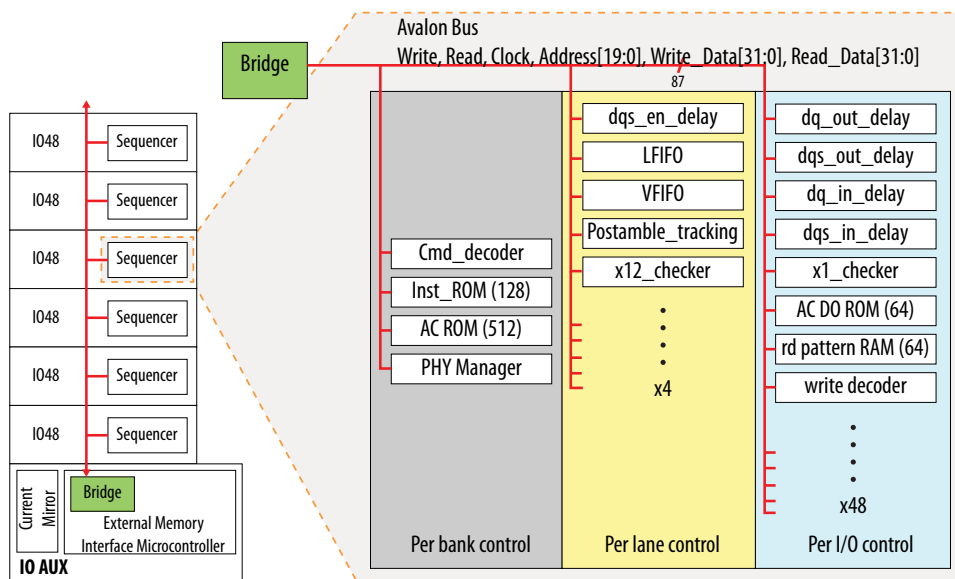
定序器实现的校准算法可确定延迟和相位设置的结合中是否需要保持数据和时钟信号的中央对齐（即使存在明显的延迟变化的情况）。随后 FPGA I/O 中的可编程延迟链实现所计算的延迟从而确保数据保持居中。

定序器嵌入于每个 I/O bank 中。该定序器由如下组件组成：

- read-write 管理器。
- address/command 集或指令 ROM。
- 辅助模块，（例如，PHY 管理器、数据管理器和追踪管理器。）
- 基于每管脚的数据码型和数据输出缓冲由 read-write 管脚器进行管理。

定序器所有主要的组件都与 Avalon 总线连接，并对 Nios II 子系统提供可控性、可见性和灵活性。

图 129. 定序器



6.7.1.4. 时钟树

Arria 10 外部存储器接口 PHY 时钟网络被设计用于支持 1.2 GHz DDR4 存储器标准。

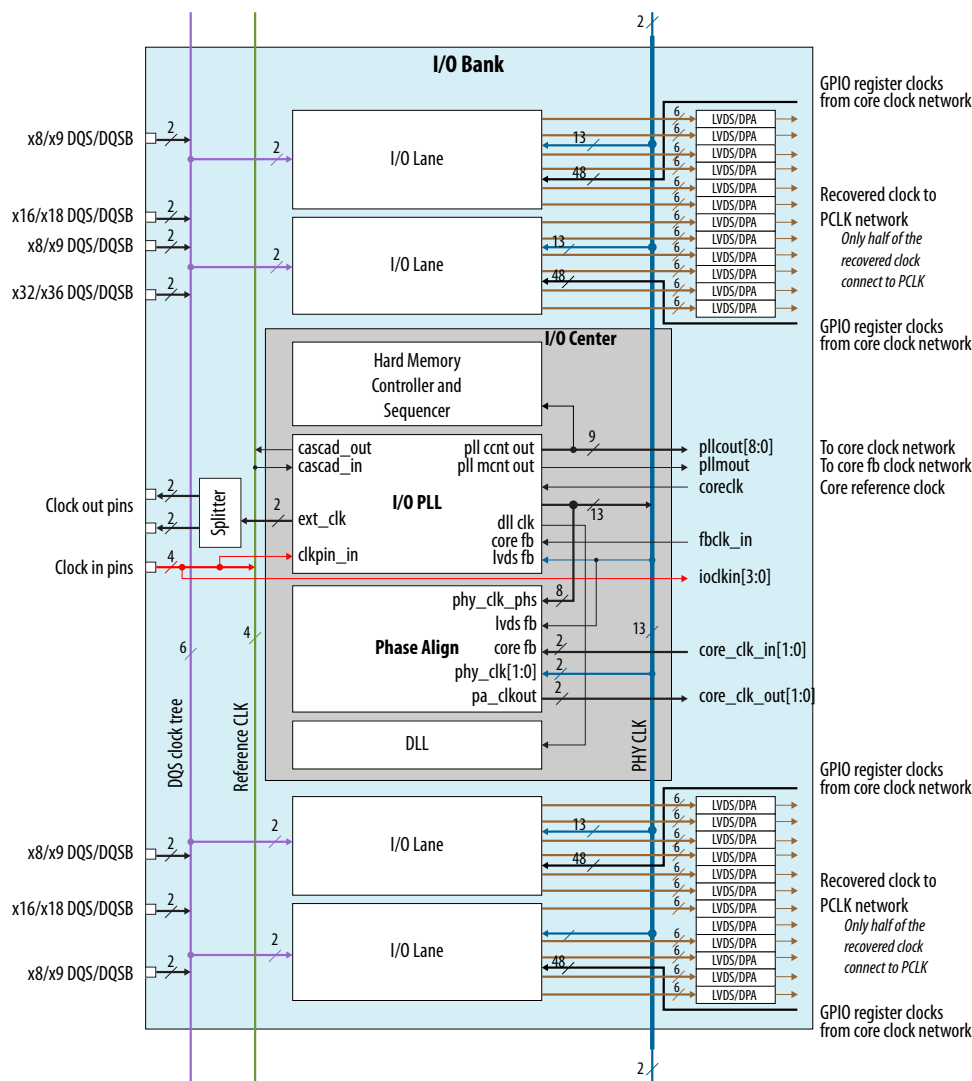
相比上一代器件，PHY 时钟网络具有较短的时钟树，生成更小的抖动和占空比失真。

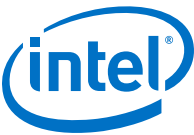
PHY 时钟网络包括下面这些时钟树：

- 参考时钟树
- PHY 时钟树
- DQS 时钟树

图 130. 时钟网络结构图

参考时钟树采用了模块化设计以方便简单集成。

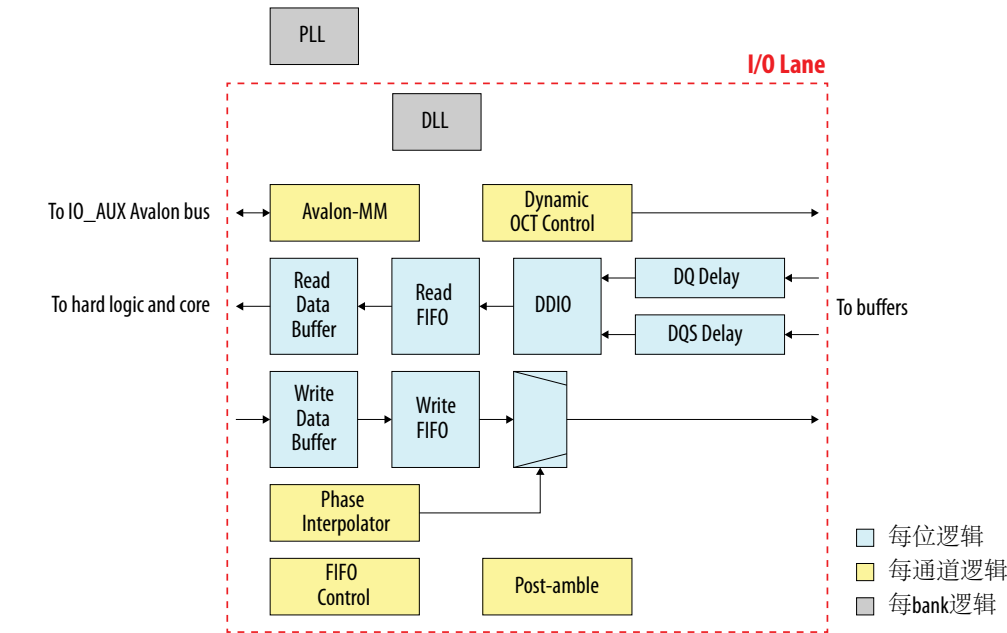




6.7.1.5. I/O 通道

每个 I/O bank 有 4 个 I/O 通道。每个 I/O 通道包含 12 个 I/O 管脚，具有相同的读写数据路径和缓冲。

图 131. I/O 通道体系架构



数据路径组件	说明
输入路径	包含采集寄存器和读 FIFO。
输出或输出使能(oe)路径	包括： <ul style="list-style-type: none">写 FIFO时钟 mux相位内插器 — 基于频率支持 5 至 10 ps 分辨率双倍数据速率控制
输入延迟链	支持延迟范围 0 至 625 ps 的 5 ps 分辨率。
读/写缓冲	读写缓冲含有嵌入式选项可以从内核或硬核存储控制器中获取数据。

相关链接

Arria 10 EMIF IP 的通用管脚输出(Pin-Out)指南

6.7.1.5.1. DQS 逻辑模块

DQS 逻辑模块包含：

- 后同步(Post-amble)寄存器
- DQS 延迟链
- FIFO 控制
- Multi-rank 切换控制模块

DQS 延迟链

DQS 延迟链对 DQS 信号提供可变延迟，使您能够在校准期间调整 DQS 信号时序，最大化用于 DQ 采集的 t_{setup} 和 t_{hold} 。

为保持延迟值不变，DQS 延迟链还包含：

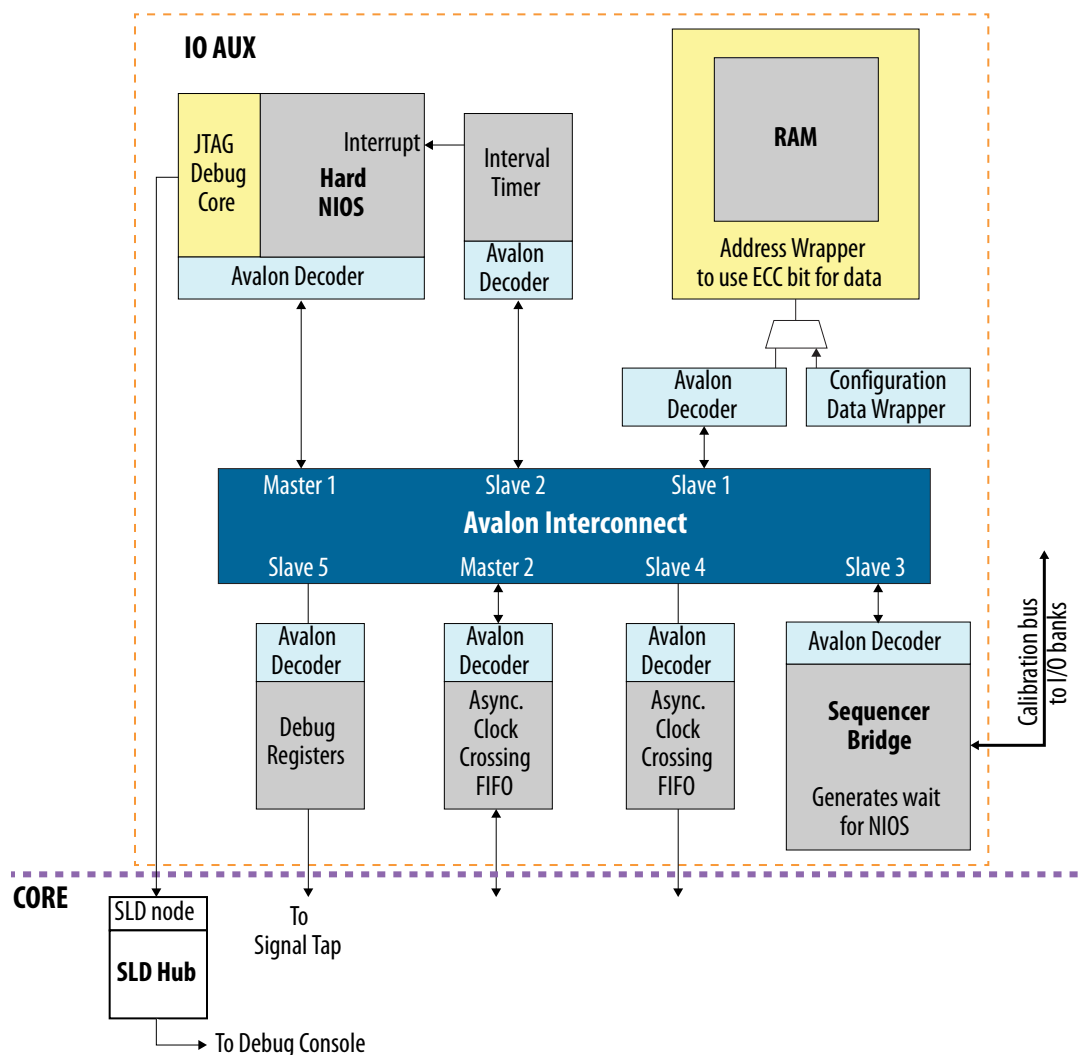
- 跟踪温度和低频电压类的逻辑
- 保持 multi-rank 接口已校准延迟设置的 Shadow 寄存器，将 DQS 延迟链设置切换为最多 4 种不同设置之一。

6.7.2. I/O AUX

每 I/O 列中有一个 I/O AUX 模块：

- 包含一个硬 Nios II 处理器，并且支持嵌入式存储器模块
- 处理整个 I/O 列的校准算法
- 通过专用 Avalon 接口与每个 I/O bank 中的定序器进行通信

图 132. IO AUX 结构图



硬 Nios II 处理器执行如下操作：

- 配置并开始定序器上的校准任务
- 收集和处理数据
- 使用最终结果来配置 I/O

Nios II 代码和定序器的组合，该算法的实现可支持下列存储器接口标准的校准：

- DDR2、DDR3 和 DDR4 SDRAM
- QDR II 和 QDR IV SRAM
- RLDRAM 3
- LPDDR2 和 LPDDR3

注意：

Intel 建议您将 Nios 子系统用于存储器接口校准。



6.8. 文档修订历史

日期	版本	修正内容
2017 年 3 月	2017.03.15	<ul style="list-style-type: none">移除了硬核存储控制器对 Avalon Streaming (Avalon ST)接口协议的支持。重命名为 Intel。
2016 年 10 月	2016.10.31	移除了 Arria 10 GX 器件系列产品中的 F36 封装。
2016 年 5 月	2016.05.02	<ul style="list-style-type: none">更新了 QDR II、QDR II+和 QDR II+ Xtreme SRAM 的最大频率。更新了 DDR4 SDRAM 最大支持的频率。移除了 Arria 10 GT 器件的 NF40 和 UF45 封装支持。在 Arria 10 器件章节的外部存储器接口 I/O 管脚中添加了指南：外部存储器接口的 I/O Bank 2A 的使用。移除了 HPS 硬核存储控制器中的 LPDDR3 支持。在 Arria 10 章节中添加了 HPS 外部存储器接口连接，解释在同一器件中使用 non-HPS EMIF 的 HPS EMIF 的限制。更新了 F36 和 KF40 封装(GX 570 和 GX 660 器件)中含有 ECC 的 DDR4 x40 所支持的接口数。移除了有关使用 3 V I/O bank 来支持含有 ECC 接口的 DDR4 x40 的注释和脚注。添加了表格以显示在同一器件中使用 HPS EMIF 实例时，Arria 10 SX 器件封装所支持的存储器接口。移除了主控制路径组件表中 DDR3 和 DDR4 的突发突变功能。移除了硬核存储控制器特性表中的 DDR4 减速模式功能。移除了硬核存储控制器特性表中硬核存储控制器的 DQS 跟踪功能。
2015 年 11 月	2015.11.02	<ul style="list-style-type: none">在每种存储器标准支持的 Altera IP 类型的表格中移除了对 DDR4、DDR3 和 DDR3L SDRAM 的 BC4 和 On-the-fly 的支持。在每种存储器标准支持的 Altera IP 类型的表格中将 DDR4、DDR3 和 DDR3L SDRAM 支持的 DQ 组修改成 x4/x8。在硬核存储控制器中添加了 LPDDR3 SDRAM 和 IP 支持。添加链接到 Arria 10 器件数据表 - 硬核存储控制器支持的存储器标准以及 Arria 10 器件数据表 - 软核存储控制器支持的存储器标准。添加了 HPS 含有 ECC 的 DDR3 x32、HPS 的 Single 和 Dual-Rank 的 DDR3 x 72、HPS 含有 ECC 的 DDR4 x32 以及 Single-Rank 的 DDR3 x72 表在 Arria 10 中的封装支持。将 Quartus II 实例更改成 Quartus Prime。
2015 年 6 月	2015.06.15	在显示硬核存储控制器体系结构的图上移除了 DFI 标签。Arria 10 器件不支持 DFI。
2015 年 5 月	2015.05.15	在列出由 Arria 10 硬核存储控制器支持的存储器标准的表格中更正了 DDR3 1/2 速率和 1/4 速率的最大频率。
2015 年 5 月	2015.05.04	更新了列出 Arria 10 器件中硬核存储控制器所支持的存储器标准的表格。
2015 年 1 月	2015.01.23	<ul style="list-style-type: none">更新了列出 Arria 10 器件所支持的存储器标准的表格。移除了 LPDDR3 SDRAM 的硬核存储控制器和 IP 支持。移除了 RLDram 2 的支持。更新了 QDR II+/II+ Xtreme SRAM 的支持，使其也包括 QDR II SRAM。添加了 QDR IV 软核存储控制器的支持。添加了脚注阐明 Arria 10 SX 480 器件的 F34 封装支持的 DDR4 x32 接口数，包括使用 I/O bank 2K。如果在 FPGA 的 DDR4 x32 接口中使用 I/O bank 2K，那么 HPS 将没有访问 DDR4 x32 接口的权限。添加了阐明具有 ECC 的 DDR3 和 DDR4 x32 接口，包括 32 比特数据和 8 比特 ECC 的信息。移除了有关子系统的硬核和软核部分的信息。Arria 10 的硬核存储控制器 IP 仅通过硬核 Nios II 处理器来校准外部存储器接口。
继续...		



日期	版本	修正内容
2014 年 8 月	2014.08.18	<ul style="list-style-type: none"> 移除了 DDR4 SDRAM 的硬核存储控制器 1/2 速率的支持。 移除了 DDR3U SDRAM 的硬核存储控制器和 IP 支持。 添加了 QDR II+ SRAM 和 QDR II+ Xtreme SRAM 的软核存储控制器全速率的支持。 更新了 HPS 支持的外部存储器标准列表。 更新了支持 U19 封装的 DDR3 x72 (单列) 存储器接口的数量。 移除了将 3 V I/O bank 用于 HPS 的注释。对于 HPS, 3 V I/O bank 不用于外部存储器接口。 更新了支持 Arria 10 SX 器件的 DDR3 x72 (双列) 存储器接口的数量。 更新了支持 Arria 10 GT 1150 器件 NF45 的 DDR4 x32 (含有 ECC) 存储器接口的数量。 添加了 QDR II + SRAM 软核存储控制器 IP 的支持。 添加了阐明 RLD RAM3 支持通过软核存储控制来使用硬核 PHY 的信息。 更新了列出硬核存储控制器功能来提高精确度的表格, 并添加了缺失的信息。 在主题列出外部存储器接口封装支持之前添加了一个注释, 以阐明并非所有的 I/O bank 可用于外部存储器接口。 将外部存储器接口管脚指南以及 DDR4 的外部存储器接口实现的实例移到外部存储器接口手册。
2013 年 12 月	2013.12.10	更新了 LPDDR2 到 LPDDR3 的 HPS 存储器标准的支持。
2013 年 12 月	2013.12.02	首次发布。

7. Arria 10 器件中的配置、设计安全和远程系统更新

本章介绍 Arria 10 器件所支持的配置方案、设计安全和远程系统更新。

相关链接

- [Arria 10 器件手册:已知问题](#)
列出了 Arria 10 器件手册章节已计划的更新。
- [Arria 10 器件数据手册](#)
为所有支持的配置方案提供更多关于已评估的未压缩 .rbf 文件大小, FPP DCLK-to-DATA[] 比率和时序参数的信息。
- [Arria 10 收发器 PHY 用户指南的 PLLs 和时钟网络章节](#)
Arria 10 器件上电到正常工作时, 未使用收发器通道配置需求的更多详情。

7.1. 增强配置和通过协议配置(Configuration via Protocol)

表 91. Arria 10 器件的配置方案和特性

Arria 10 器件支持 1.8 V 编程电压和多个配置方案。

方案	数据位宽	最大时钟速率 (MHz)	最大数据速率 (Mbps) (26)	解压缩	设计安全性(27)	部分重配置 (28)	远程系统更新
JTAG	1 bit	33	33	—	—	Yes (29)	—
通过 EPCQ-L 配置器件的主动串行 (AS)	1 bit, 4 bits	100	400	Yes	Yes	Yes(29)	Yes
通过 CPLD 或外部微控制器的被动串行 (PS)	1 bit	100	100	Yes	Yes	Yes(29)	并行闪存加载器 (PFL) IP 内核
通过 CPLD 或外部微控制器的快速被动并行(FPP)	8 bits	100	3200	Yes	Yes	Yes (30)	PFL IP 内核
	16 bits			Yes	Yes		
	32 bits			Yes	Yes		
继续...							

(26) 使能压缩或者设计安全功能都会影响最大数据速率。请参考 Arria 10 器件数据表获得更多信息。

(27) 不能同时进行加密和压缩。

(28) 部分重配置是器件系列的高级功能。如果您有兴趣使用部分重配置, 请联络 Intel 获得支持。

(29) 仅作为内部主机被配置时才能进行部分配置。

(30) 支持 100 MHz 最大时钟速率。

Intel Corporation. All rights reserved. Agilx, Altera, Arria, Cyclone, Enpirion, Intel, the Intel logo, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.



方案	数据位宽	最大时钟速率 (MHz)	最大数据速率 (Mbps) (26)	解压缩	设计安全性 (27)	部分重配置 (28)	远程系统更新
通过 HPS 配置	16 bits	100	3200	Yes	Yes	Yes ⁽³⁰⁾	—
	32 bits			Yes	Yes		
Configuration via Protocol [CvP (PCIe*)]	x1、x2、x4 和 x8 通道	—	8000	Yes	Yes	Yes ⁽²⁹⁾	—

可使用 Configuration via Protocol (CvP) 通过 PCIe 配置 Arria 10 器件。Arria 10 CvP 实现符合 PCIe 100 ms 的上电到有效(power-up-to-active)时间要求。

相关链接

[Intel FPGA 中配置通过协议 \(CvP\) 实现用户指南](#) 提供了关于 CvP 配置方案的更多信息。

7.2. 配置方案

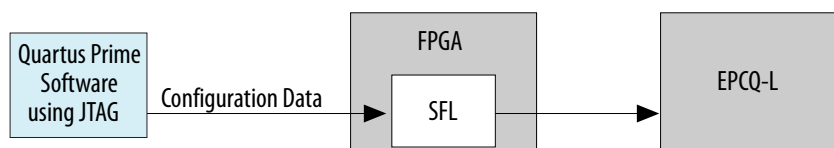
这一部分介绍 AS、PS、FPP 和 JTAG 配置方案。

相关链接

- [Intel FPGA 中配置通过协议 \(CvP\) 实现用户指南](#) 提供了关于 CvP 配置方案的更多信息。
- [部分重配置的设计规划](#) 提供了关于部分重配置的更多信息。

7.2.1. 主动串行配置

图 133. AS 配置方案的高级 EPCQ-L 编程的概述



AS 配置方案中，配置数据储存在 EPCQ-L 配置器件中。通过带有 Serial Flash Loader (SFL) IP 内核的 JTAG 接口编程 EPCQ-L 器件在线系统。在 JTAG 接口和 EPCQ-L 器件间的 FPGA 中，SFL 作为桥接运行。Arria 10 器件中的 AS 存储器接口块控制配置过程。

AS 配置方案支持 AS x1 (1 位数据宽度)和 AS x4 (4 位数据宽度)模式。AS x4 模式的配置时间比 AS x1 模式快 4 倍。在 AS 配置方案中，Arria 10 器件控制配置接口。

(26) 使能压缩或者设计安全功能都会影响最大数据速率。请参考 Arria 10 器件数据表获得更多信息。

(27) 不能同时进行加密和压缩。

(28) 部分重配置是器件系列的高级功能。如果您有兴趣使用部分重配置，请联络 Intel 获得支持。

注意: 对于使用 SFL 的有效串行编程，MSEL 管脚必须设定为 Active Serial 设置从而允许 programmer 读取 EPCQ-L ID。

相关链接

- [Arria 10 器件数据表](#)
提供了关于 AS 配置时序的更多信息。
- [AN 370: 通过 Quartus Prime 软件使用串行闪存加载器](#)
- [Nios II 闪存编程器用户指南](#)
- [EPCQ-L 串行配置器件数据表](#)
- [EPCQ-L 器件封装信息](#)
提供了关于 EPCQ-L 封装规范，热变电阻和尺寸的详细信息。

7.2.1.1. DATA 时钟(DCLK)

Arria 10 器件生成串行时钟 DCLK，为串行接口提供时序。在 AS 配置方案中，Arria 10 器件在 DCLK 下降沿驱动控制信号并在该时钟的下降沿锁存配置数据。

AS 配置方案支持的最大 DCLK 频率是 100 MHz。您可以使用 CLKUSR 或内部振荡器获得 DCLK 源。如果使用内部振荡器，可在 Quartus Prime 的 **Configuration** 页面，**Device and Pin Options** 对话框下，选择 12.5，25，50 或 100 MHz 时钟。

上电后，DCLK 由默认的 12.5 MHz 内部振荡器驱动。Arria 10 器件通过读取编程文件的选项位来决定使用的时钟源和频率。

相关链接

[Arria 10 器件数据表](#)

提供了关于 AS 配置方案中的 DCLK 频率规范的更多信息。

7.2.1.2. 主动串行单器件配置

要配置 Arria 10 器件，请按下图所示，将该器件连接到一个四路串行配置(EPCQ-L)器件。

图 134. 单一器件 AS x1 模式配置

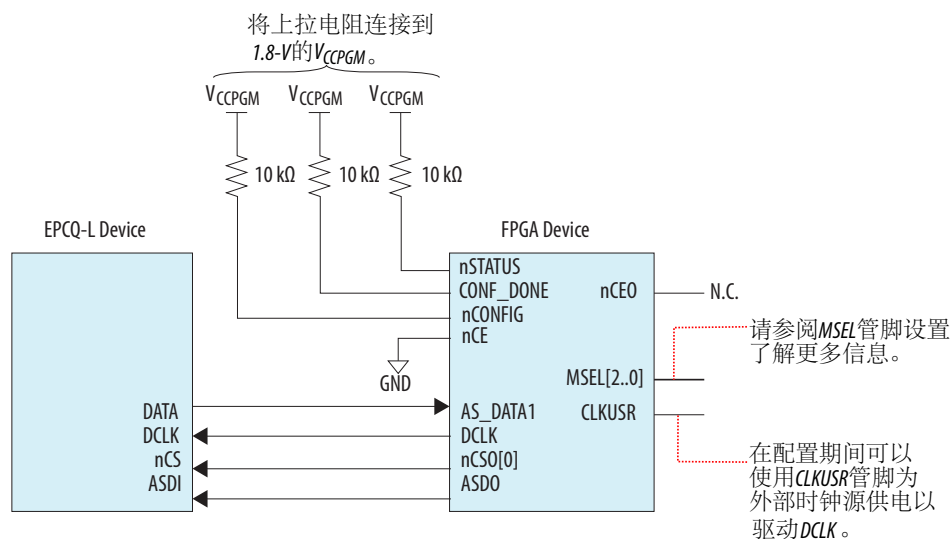
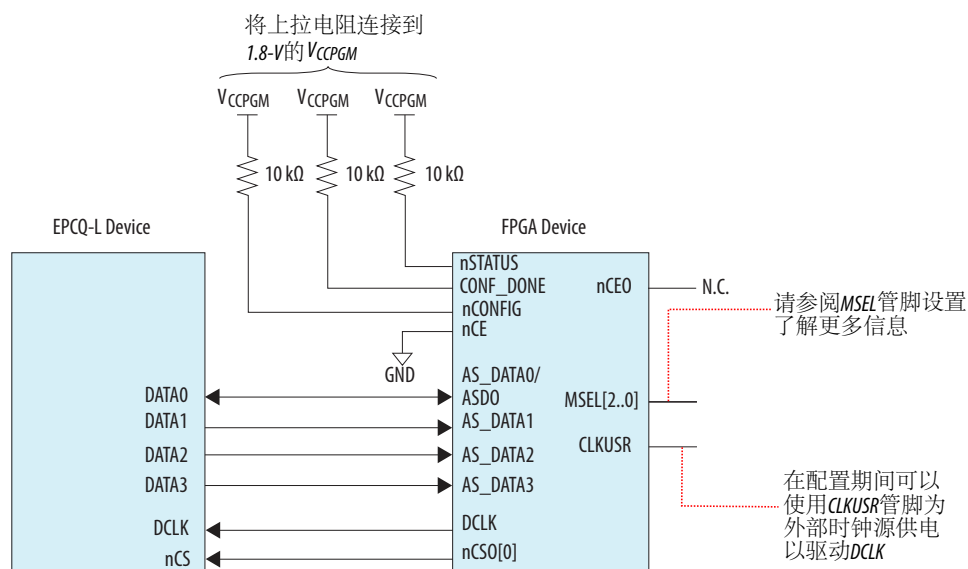


图 135. 单一器件 AS x4 模式配置



7.2.1.3. 主动串行多器件配置

可配置被连接到同条链中的多个器件。仅 AS x1 模式支持多器件配置。

链中的第一个器件是配置主器件。链中接下来的器件是配置从器件。

7.2.1.3.1. 管脚连接和指南

对于该配置设置，请遵循以下管脚连接和指南：

- 硬接线链中首个器件中的 MSEL 管脚，以选择 AS 配置方案。对于链中后续器件，硬接线其 MSEL 管脚以选择 PS 配置方案。支持 PS 配置的任何其他 Intel FPGA 也可成为链的部分作为配置从器件。
- 在链中将所有器件的以下管脚连接在一起：
 - nCONFIG
 - nSTATUS
 - DCLK
 - DATA[]
 - CONF_DONE

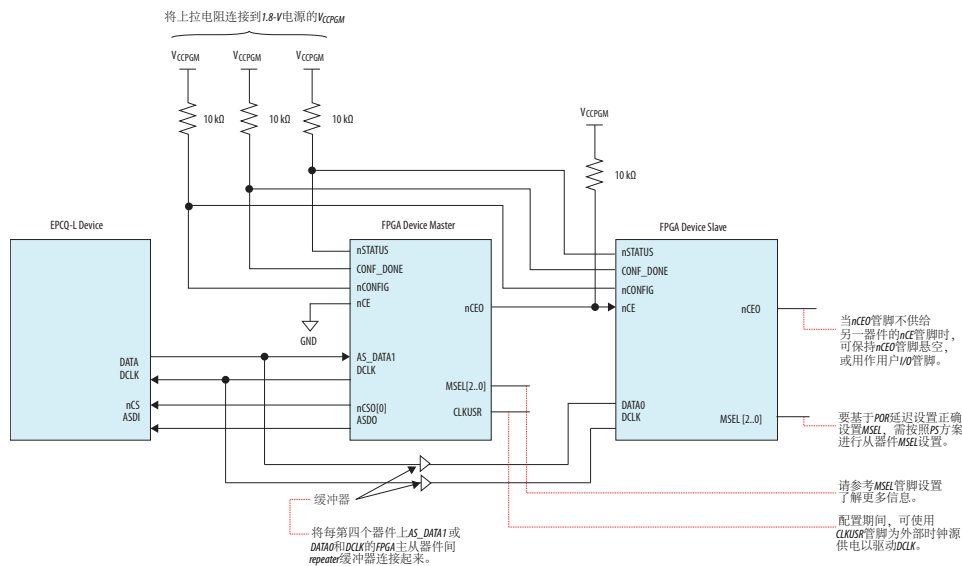
通过将 CONF_DONE、nSTATUS 和 nCONFIG 管脚连接在一起，器件同时进行初始化并进入用户模式。如果链中的任何器件检测到一个错误，那么整个链的配置停止并且您必须重新配置所有器件。例如，如果链中的第一个器件在 nSTATUS 管脚上标记一个错误，那么它通过拉低 nSTATUS 管脚而复位链。

- 确保 DCLK 和 DATA[] 在每四个器件进行缓冲，以避免信号完整性和时钟偏移问题。

7.2.1.3.2. 使用多个配置数据

为使用多个配置数据配置多个 Arria 10 器件，则如下图所示，将各器件连接到一个 EPCQ-L 器件。

图 136. 链中两个器件接收到不同配置数据集时的多器件 AS 配置





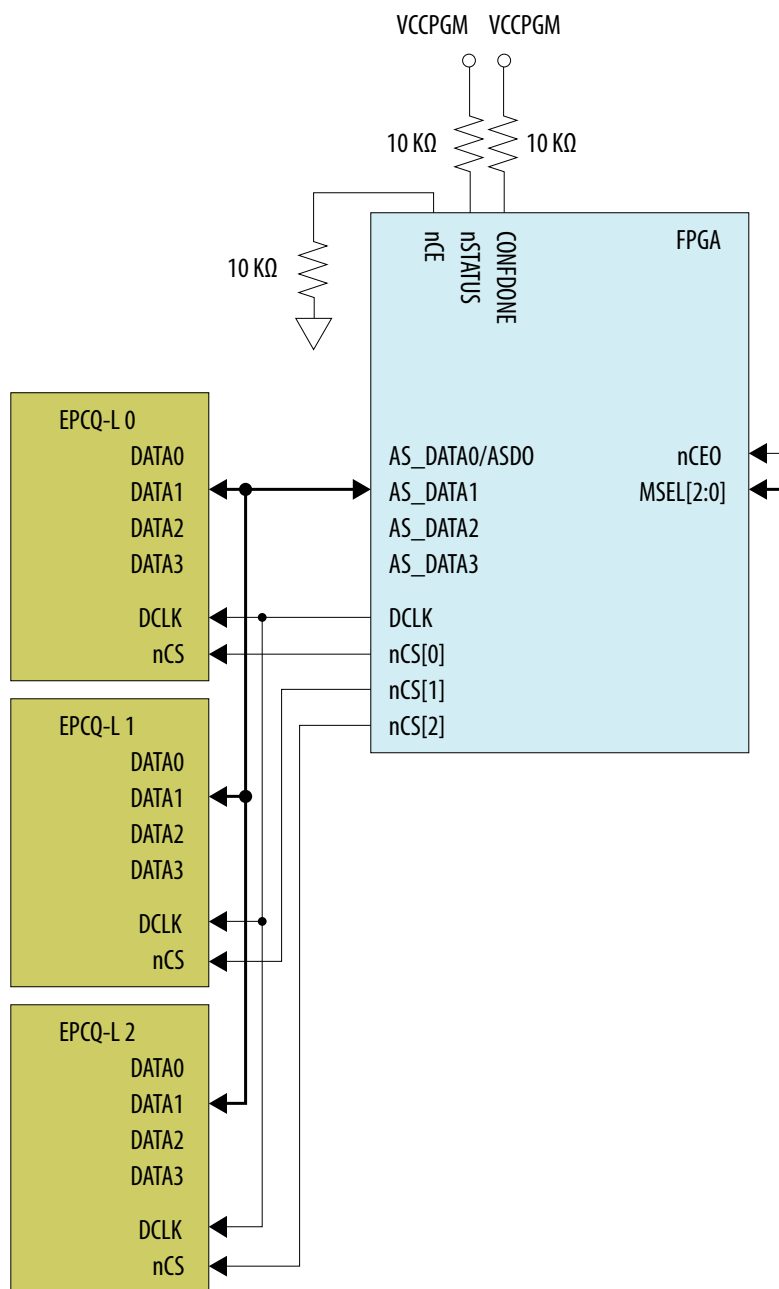
当器件完成配置时，它的 **nCEO** 管脚被释放为低电平来激活链中下一个器件的 **nCE** 管脚。一个时钟周期后，第二个器件的配置自动开始。

7.2.1.4. 多个 EPCQ-L 器件的主动串行配置

Arria 10 器件支持多达 3 个 EPCQ-L 器件用于实现配置和远程系统更新。

每个 Arria 10 器件可以使用多达 3 个 EPCQ-L 器件。每个 EPCQ-L 器件具有一个专用的 **nCSO** 管脚，但共享其他管脚，如下图所示。

图 137. 多个 EPCQ-L 器件的 AS 配置



可使用 Quartus Prime 软件选择 EPCQ-L 器件的数目。

7.2.1.5. 使用 EPCQ-L 器件

EPCQ-L 器件支持 AS x1 和 AS x4 模式。

注意: Arria 10 器件仅支持 EPCQ-L 器件。



每个 Arria 10 器件具有三个 nCS0 管脚—nCS0[2..0]。这使得 Arria 10 器件最多可以连接到多达 3 个 EPCQ-L 器件。

连接多达 3 个 EPCQ-L 器件的优势：

- 能够存储多个设计文件以实现远程系统更新。
- 增加超出可用的最大单一 EPCQ 器件的存储。

相关链接

- [EPCQ-L 串行配置器件数据表](#)
- [EPCQ-L 器件封装信息](#)
提供了关于 EPCQ-L 封装规范，热变电阻和尺寸的详细信息。

7.2.1.5.1. 控制 EPCQ-L 器件

配置过程中，Arria 10 器件通过驱动其 nCS0 输出管脚(连接到 EPCQ-L 器件的片选 (nCS)管脚)为低电平而使能 EPCQ-L 器件。Arria 10 器件使用 DCLK 和 ASDO 管脚来将操作命令和读地址信号发送到 EPCQ-L 器件。EPCQ-L 器件提供数据到其串行数据输出(DATA[])管脚(连接到 Arria 10 器件的 AS_DATA[] 输入)。

注意: 如果您想要控制 EPCQ-L 管脚，那么保持 nCONFIG 管脚低电平并上拉 nCE 管脚。这会导致器件复位并三态 AS 配置管脚。

7.2.1.5.2. 走线长度指南

最大走线长度适用于下表中列出的单器件和多器件 AS 配置设置。走线长度为从 Arria 10 器件到 EPCQ-L 器件的长度。

注意: 板级 DCLK 与 AS_DATA [3..0] 走线之间的最大偏移不应超过 400 ps。

表 92. Arria 10 器件的 AS x1 和 x4 配置的最大走线长度

Arria 10 器件 AS 管脚	最大电路板走线长度(英寸)	
	12.5/ 25/ 50 MHz	100 MHz
DCLK	10	6
AS_DATA[3..0]	10	6
nCS0[2..0]	10	6

相关链接

[Arria 10 器件数据表中的 AS 时序参数](#)
提供了关于数据建立和保持时间要求的更多信息。

7.2.1.5.3. 编程 EPCQ-L 器件

可使用 Intel FPGA 下载电缆在系统编程 EPCQ-L 器件。或者，采用具有 SRunner 软件驱动程序的微处理器编程 EPCQ-L。

在系统编程 (ISP) 提供了通过 AS 编程接口或 JTAG 接口编程 EPCQ-L 的两个选项。选择 AS 编程接口时，配置数据由 Quartus Prime 软件或任何支持的第三方软件编程到 EPCQ-L 中。选择 JTAG 接口时，必须将名为 SFL IP 内核的 Intel FPGA IP 下载到 Arria 10 器件中以构成 JTAG 接口和 EPCQ-L 间的桥接。从而直接通过 JTAG 接口编程 EPCQ-L。

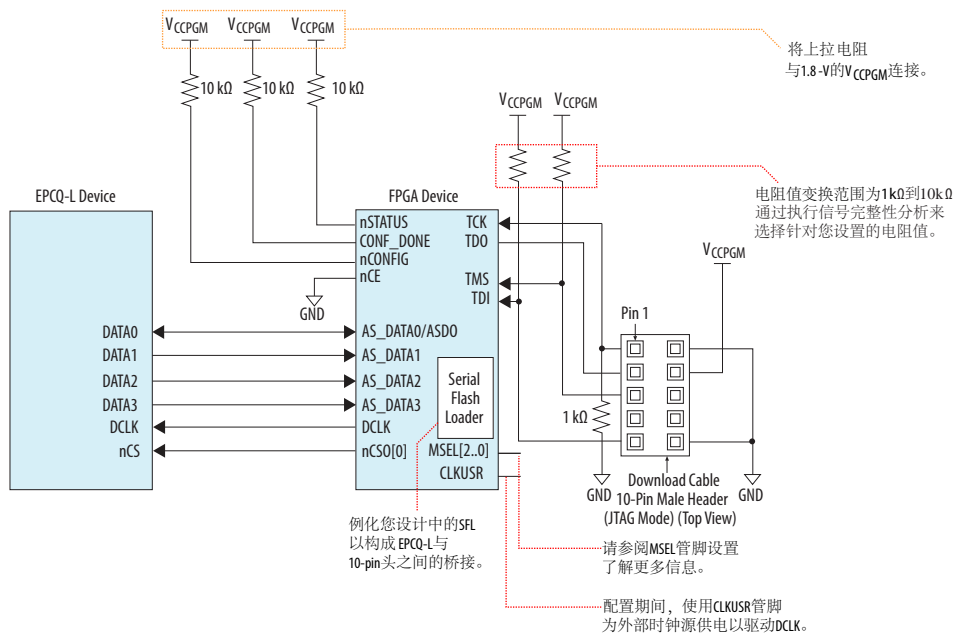
相关链接

- AN 370: 通过 Quartus Prime 软件使用串行闪存加载器
- AN 418: SRunner: 串行配置器件编程的嵌入式解决方案
- Nios II 闪存编程器用户指南

使用 JTAG 接口编程 EPCQ-L

要使用 JTAG 接口编程 EPCQ-L 器件，请按下图所示连接器件。

图 138. 使用 JTAG 接口编程 EPCQ-L 的连接设置

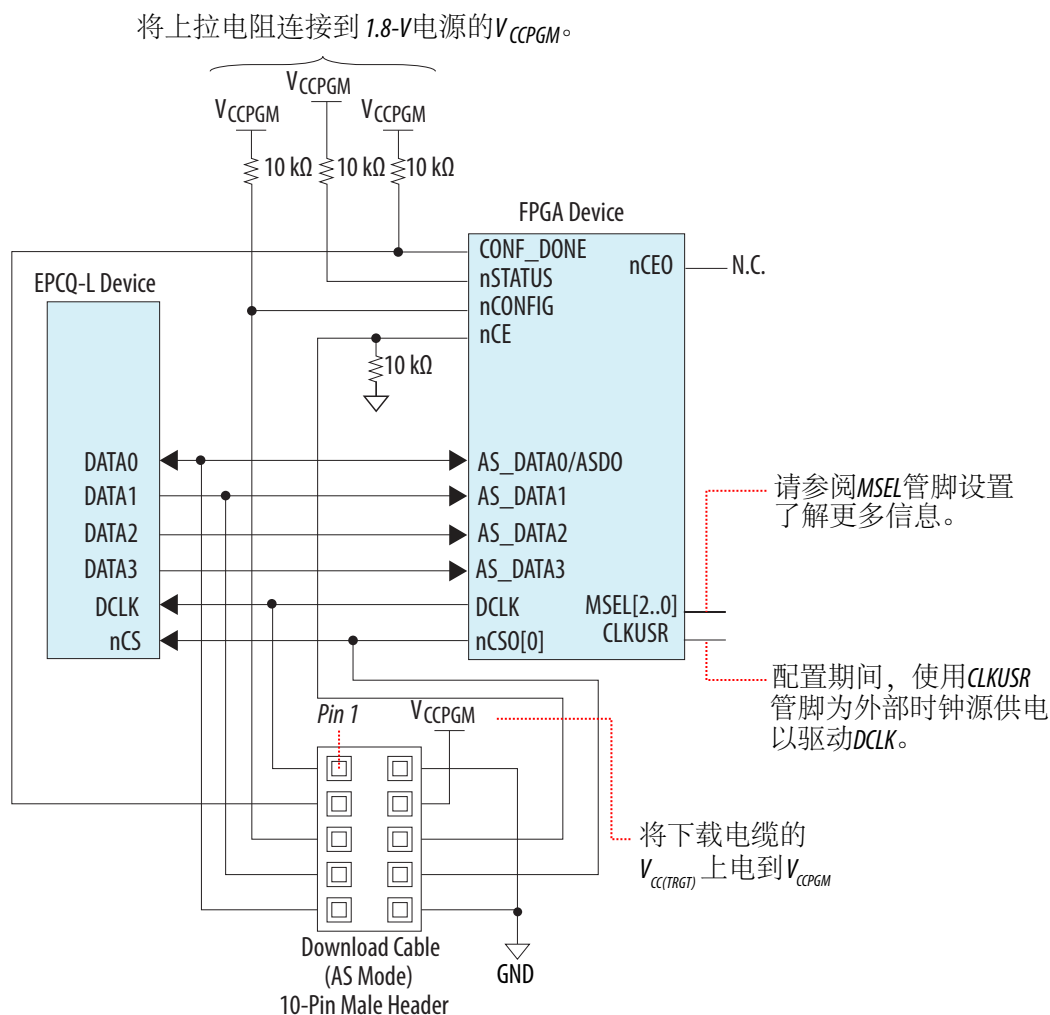


使用主动串行接口编程 EPCQ-L

要使用 AS 接口编程 EPCQ-L 器件，请按下图所示连接器件。

图 139. 使用 AS 接口编程 EPCQ-L 的连接设置

使用 AS 接头时，programmer 将操作命令和配置位串行地发送到 DATA0 的 EPCQ-L。

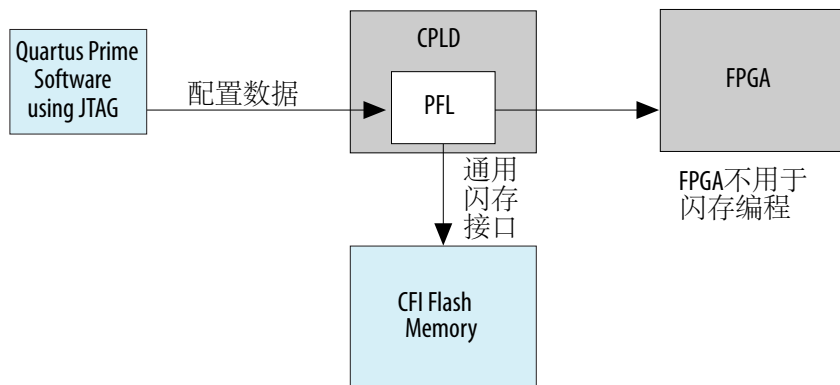


编程 EPCQ-L 器件时，下载电缆将 nCE 管脚驱动为高电平以禁用对 AS 接口的访问。nCONFIG 线也被拉低以保持 Arria 10 器件处于复位阶段。编程完成后，下载电缆释放 nCE 和 nCONFIG，以使得下拉和上拉电阻能够分别驱动 GND 和 V_{CCPGM} 的管脚。

使用下载电缆进行 EPCQ-L 编程期间，DATA0 将编程数据、操作指令和地址信息从下载电缆传输到 EPCQ-L。使用下载电缆进行 EPCQ-L 验证期间，DATA1 将编程数据返回到下载电缆。

7.2.2. 被动串行配置

图 140. PS 配置方案中闪存编程总括



PS 配置方案使用外部主机。您可以使用一个微处理器、MAX II 器件、MAX V 器件或一台主机 PC 作为外部主机。

您可以使用一个外部主机控制配置数据从外部存储(例如闪存)传输到 FPGA。控制配置过程的设计位于外部主机。

您可以将配置数据存储在程序目标文件 (.pof)、.rbf、.hex 或 .ttf 中。如果您使用 .rbf、.hex 或 .ttf 中的配置数据，请首先发送每个数据字节的 LSB。例如，如果 .rbf 包含字节排序 02 1B EE 01 FA，那么发送到器件的串行数据一定是 0100-0000 1101-1000 0111-0111 1000-0000 0101-1111。

您可以使用 PFL IP core 及 MAX II/MAX V 器件来读取闪存器件的配置数据并且配置 Arria 10 器件。

对于 PC 主机，使用 Intel FPGA 下载电缆将 PC 连接到器件。

配置数据被串行地移入器件的 DATA0 管脚。

如果您在使用 Quartus Prime 编程器并且 CLKUSR 管脚被使能，那么不需要为管脚提供时钟源来初始化器件。

相关链接

- [Arria 10 硬核处理器系统技术参考手册](#)
提供了关于通过 HPS 进行配置的详细信息。
- [并行闪存加载器□□内核用户指南](#)

7.2.2.1. 使用外部主机的被动串行单器件配置

要配置 Arria 10 器件，就将器件连接到外部主机，如下图所示。

将电阻连接到为FPGA器件提供可接受输入信号的电源。
 V_{CCPGM} 必须足够高，以符合器件和外部主机上I/O的 V_{IH} 规范。
 Intel建议通过 V_{CCPGM} 上电配置系统所有I/O。

当nCEO管脚不供给另一器件的nCE管脚时，可保持nCEO管脚悬空，或用作用户I/O管脚。

请参阅MSEL管脚设置，了解更多信息。

要配置 Arria 10 器件，就将器件连接到下载电缆，如下图所示。

如果电路板上仅使用下载电缆这个配置方案，则只需DATA0和DCLK的上拉电阻。可确保配置后DATA0和DCLK不会保持悬空。如果还使用MAX10器件，MAXV器件或微处理器则无需DATA0和DCLK上的上拉电阻。

将上拉电阻连接到与下载电缆相同的电源(VCCIO)。

请参阅MSEL管脚设置了解更多信息

您可以配置连接到链中的多个 Arria 10 器件。

7.2.2.3.1. 管脚连接和指南

对于该配置设置，请遵循以下管脚连接和指南：

- 在链中将所有器件的以下管脚连接在一起：
 - nCONFIG
 - nSTATUS
 - DCLK
 - DATA0
 - CONF_DONE

通过将 CONF_DONE 以及 nSTATUS 管脚连接在一起，器件初始化的同时进入用户模式。如果链中的任何器件检测到一个错误，那么整个链的配置停止并且您必须重新配置所有器件。例如，如果链中的第一个器件在 nSTATUS 管脚上标记一个错误，那么它通过拉低 nSTATUS 管脚而把链复位。

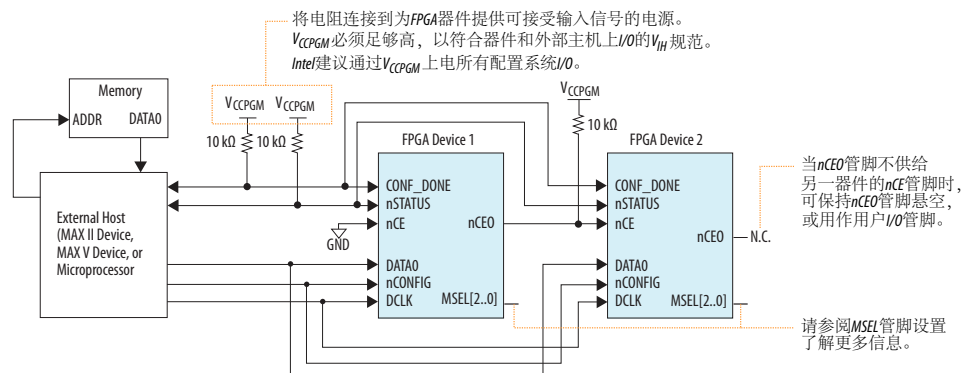
- 如果使用相同的配置数据配置链中的器件，那么器件必须具有相同的封装和密度。

7.2.2.3.2. 使用多个配置数据

要使用多个配置数据配置一条链中的多个 Arria 10 器件，需要按下图所示将所有器件连接到外部主机。

注意：默认情况下，nCEO 在 Quartus Prime 软件中是禁用的。对于多器件配置链，您必须在 Quartus Prime 中使能 nCEO 管脚。否则，器件配置可能会失败。

图 143. 两个器件接收到不同配置数据集时的多器件 PS 配置



当器件完成配置时，它的 nCEO 管脚被释放为低电平来激活链中下一个器件的 nCE 管脚。一个时钟周期后，第二个器件的配置自动开始。

7.2.2.3.3. 使用一个配置数据

要使用一个配置数据配置一条链中的多个 Arria 10 器件，就将所有器件连接到一个外部主机，如下图所示。

注意：默认情况下，nCEO 管脚在 Quartus Prime 软件中是禁用的。对于多器件配置链，您必须在 Quartus Prime 中使能 nCEO 管脚。否则，器件配置可能会失败。

[illegible]

7.2.2.3.4. 使用 PC 主机和下载电缆

注意: 默认情况下，nCEO 管脚在 Quartus Prime 软件中是禁用的。对于多器件配置链，您必须使能 Quartus Prime 中的 nCEO 管脚。否则，器件配置可能会失败。

如果电路板上仅使用下载电缆这个配置方案，则只需DATA0和DCLK上的上拉电阻。可确保配置后DATA0和DCLK不会保持悬空。如果还使用其他配置器件，则无需DATA0和DCLK上的上拉电阻。

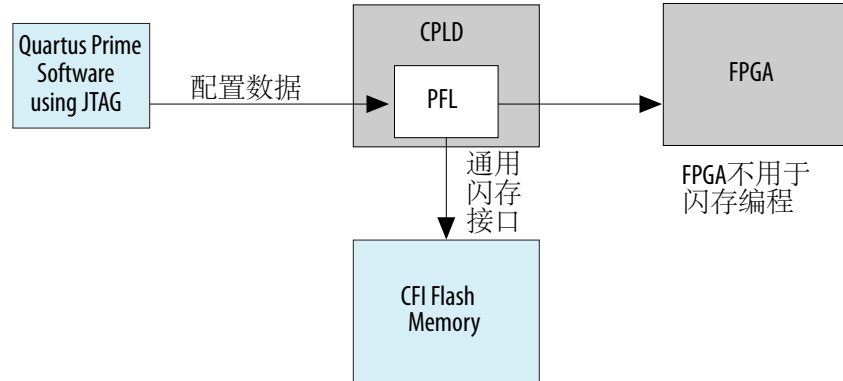
请参阅MSEL管脚设置了解更多信息。

将上拉电阻连接到与下载电缆相同的电源(V_{CCPGM})。

 反馈

7.2.3. 快速被动并行配置

图 146. FPP 配置方案中的闪存编程总括



FPP 配置方案使用一个外部主机，例如微处理器、MAX® II 器件或 MAX V 器件。这个方案是配置 Arria 10 器件的最快方法。FPP 配置方案支持 8 位、16 位和 32 位数据宽度。

您可以使用一个外部主机控制配置数据从外部存储(例如闪存)传输到 FPGA。控制配置过程的设计位于外部主机。您可以使用原始二进制文件(.rbf)、十六进制 (Intel-Format)文件 (.hex)或表格文本文件(.tff)格式存储配置数据。

使用 PFL IP 内核通过 MAX II 或 MAX V 器件从闪存器件读取配置数据并配置 Arria 10 器件。

注意: 在 FPP 配置中，在 CONF_DONE 信号变高后需要两个 DCLK 时钟的下降沿以开始用于压缩和未压缩配置数据的器件初始化。

相关链接

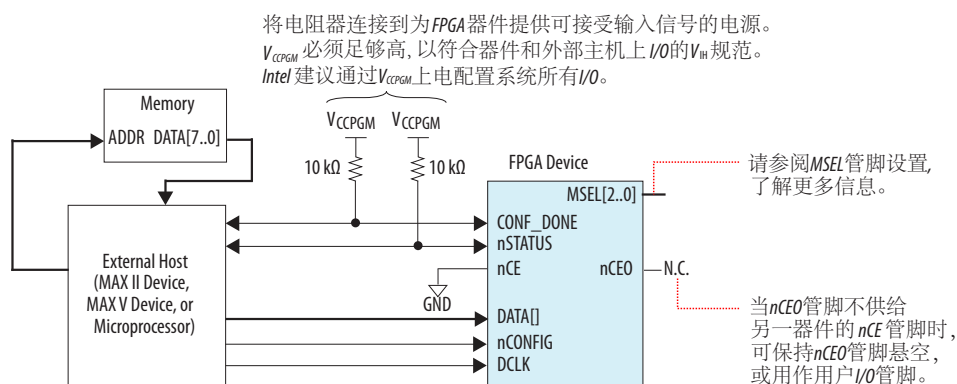
- [Altera 并行闪存加载器 IP 内核用户指南](#)
- [Arria 10 器件数据表](#)
提供了关于 FPP 配置时序的更多信息。

7.2.3.1. 快速被动并行单器件配置

要配置 Arria 10 器件，如下图所示，将器件连接到外部主机。

注意: 如果使用 FPP x8 配置模式，那么使用 DATA[7..0]管脚。如果使用 FPP x16 配置模式，那么使用 DATA[15..0]管脚。如果使用 FPP x32 配置模式，那么使用 DATA[31..0]管脚。

图 147. 使用外部主机的单一器件 FPP 配置



7.2.3.2. 快速被动并行多器件配置

您可以配置连接到链中的多个 Arria 10 器件。

7.2.3.2.1. 管脚连接和指南

对于该配置设置，请遵循以下管脚连接和指南：

- 在链中将各个器件的以下管脚分别连接在一起：
 - nCONFIG
 - nSTATUS
 - DCLK
 - DATA[]
 - CONF_DONE

通过将 CONF_DONE 和 nSTATUS 管脚连接在一起，器件初始化的同时进入用户模式。如果链中的任何器件检测到一个错误，那么整个链的配置停止并且您必须重新配置所有器件。例如，如果链中的第一个器件在 nSTATUS 管脚上标记一个错误，那么它通过拉低 nSTATUS 管脚而把链复位。

- 确保 DCLK 和 DATA[] 在每四个器件进行缓冲，以防止信号完整性和时钟偏移问题。
- 链中所有器件都必须使用相同的数据宽度。
- 如果使用相同的配置数据配置链中的器件，那么器件必须具有相同的封装和密度。

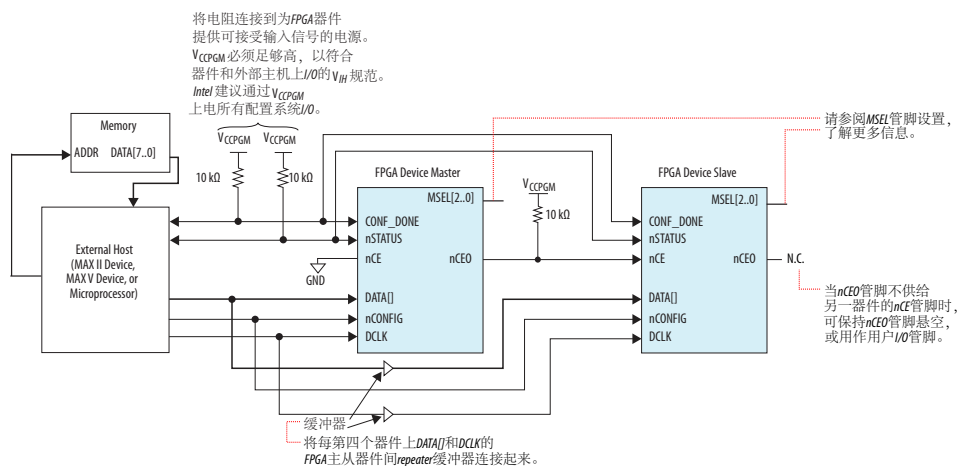
7.2.3.2.2. 使用多个配置数据

要使用多个配置数据配置一条链中的多个 Arria 10 器件，需要按下图所示将所有器件连接到一个外部主机。

注意： 如果使用 FPP x8 配置模式，那么使用 DATA[7..0] 管脚。如果使用 FPP x16 配置模式，那么使用 DATA[15..0] 管脚。如果使用 FPP x32 配置模式，那么使用 DATA[31..0] 管脚。

注意： 默认情况下，nCEO 管脚在 Quartus Prime 中是禁用的。对于多器件配置链，您必须在 Quartus Prime 中使能 nCEO 管脚。否则，器件配置可能会失败。

图 148. 两个器件接收到不同配置数据集时，使用外部主机的多器件 FPP 配置



当器件完成配置时，它的 **nCEO** 管脚被释放为低电平来激活链中下一个器件的 **nCE** 管脚。一个时钟周期后，第二个器件的配置自动开始。

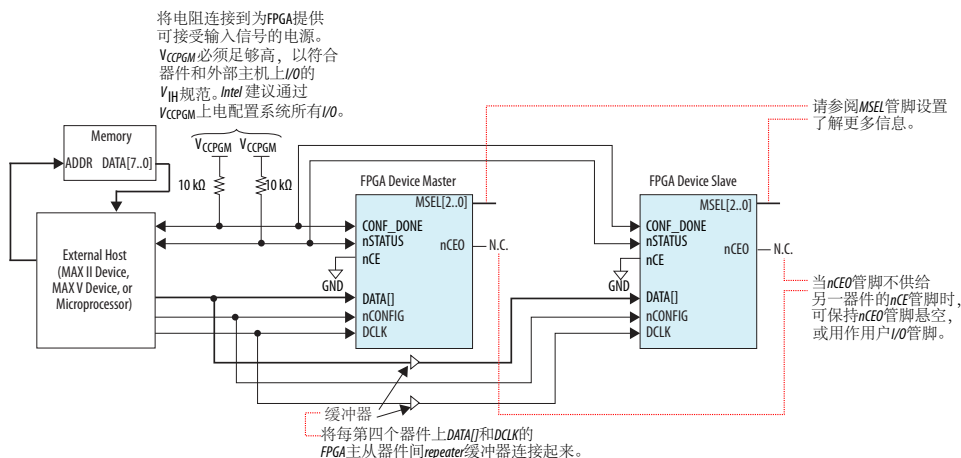
7.2.3.2.3. 使用一个配置数据

要使用一个配置数据配置一条链中的多个 **Arria 10** 器件，需要按下图所示将所有器件连接到一个外部主机。

注意: 如果使用 FPP x8 配置模式, 那么使用 DATA[7..0]管脚。如果使用 FPP x16 配置模式, 那么使用 DATA[15..0]管脚。如果使用 FPP x32 配置模式, 那么使用 DATA[31..0]管脚。

注意: 默认情况下，nCEO 管脚在 Quartus Prime 中被禁用。对于多器件配置链，您必须在 Quartus Prime 中使能 nCEO 管脚。否则，器件配置可能会失败。

图 149. 当两个器件接收相同的数据时，使用外部主机的多器件 **FPP** 配置



链中器件的 nCE 管脚被连接到 GND，从而使得这些器件的配置在同时开始和结束。

7.2.4. JTAG 配置

在 Arria 10 器件中，JTAG 指令优先于其它的配置方案。

Quartus Prime 通过编程器中的下载电缆生成一个用于 JTAG 配置的 SRAM（目标文件）（**.sof**），您可以通过一条下载电缆使用该文件在 Quartus Prime 软件编程器中进行 JTAG 配置。或者，您可以将 JRunner 软件和**.rbf**一起使用或将 JAM™ 标准测试和编程语言(STAPL)格式文件(**.jam**)或 JAM 字节代码文件(**.jbc**)和其它第三方编程器工具一起使用。

注意: 如果您使用基于 JTAG 的配置对 Arria 10 器件进行配置，那么就不能使用 Arria 10 解压缩或设计安全功能。

Arria 10 器件的芯片全复位(DEV_CLRn)和芯片全输出使能 (DEV_OE)管脚不影响 JTAG 边界扫描或编程操作。

Intel FPGA 下载电缆支持 1.5 V 或 1.8 V 的 V_{CCPGM} 电源；但不支持 1.2 V 的目标电源电压。

相关链接

- [器件配置管脚](#) (第 240 页)
提供了 JTAG 配置管脚的更多信息。
- [JTAG 安全模式](#) (第 252 页)
- [Arria 10 器件数据表](#)
提供了关于 JTAG 配置时序的更多信息。
- [Jam STAPL 语言的编程支持](#)
- [Intel FPGA USB 下载电缆用户指南](#)
- [ByteBlaster II 下载电缆用户指南](#)
- [EthernetBlaster 通信电缆用户指南](#)
- [EthernetBlaster II 通信电缆用户指南](#)

7.2.4.1. JTAG 单器件配置

要在 JTAG 链中配置单个器件，编程软件将其他器件设置为旁路模式。旁路模式下的器件通过单个旁路寄存器将编程数据从 TDI 管脚传输至 TDO 管脚。一个时钟周期后，编程数据出现在 TDO 管脚上。

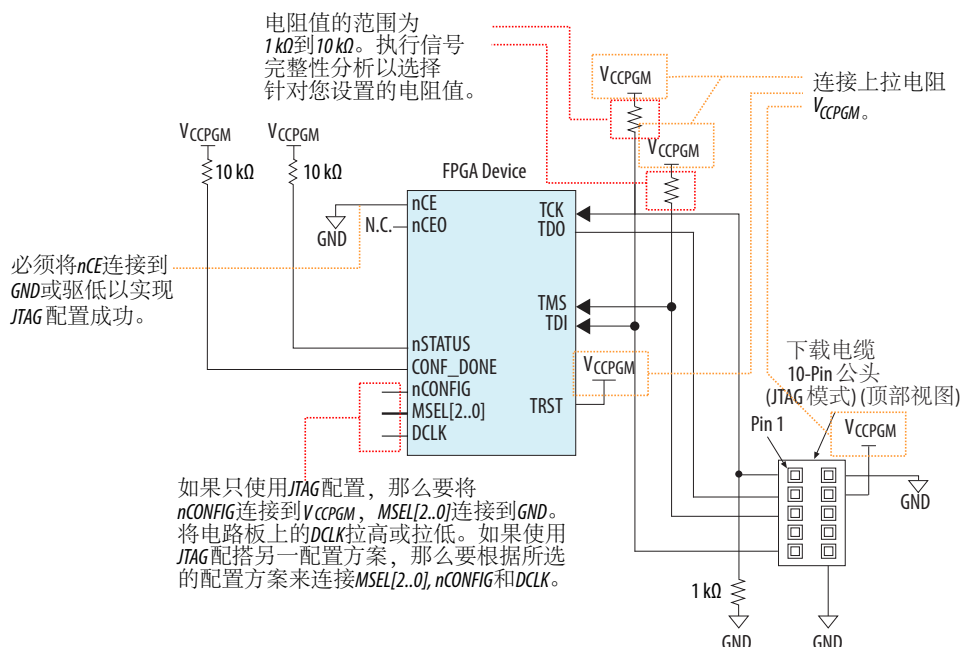
Quartus Prime 软件可使用 CONF_DONE 管脚来验证使用 JTAG 端口的配置过程是否完成：

- CONF_DONE 管脚为低电平—表明配置已经失败。
- CONF_DONE 管脚为高电平，表示配置成功。

在使用 JTAGTDI 端口串行地发送配置数据后，TCK 端口被提供额外的 1,222 周期以执行器件初始化。

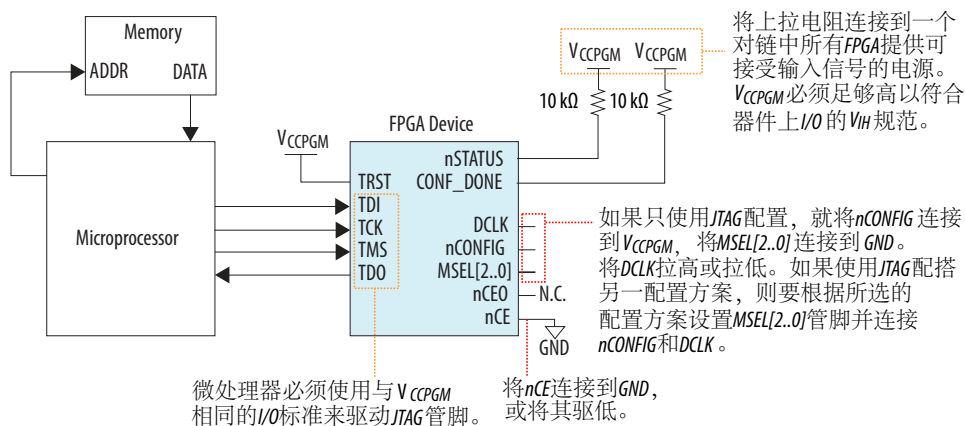
要使用下载电缆配置 Arria 10 器件，请按下图所示连接器件。

图 150. 使用一个下载电缆时的单器件的 JTAG 配置



要使用微处理器配置 Arria 10 器件, 请按下图所示连接器件。您可以将 JRunner 用作软件驱动器。

图 151. 使用一台微处理器的单器件的 JTAG 配置



相关链接

AN 414: JRunner 软件驱动程序: PLD JTAG 配置的嵌入式解决方案

7.2.4.2. JTAG 多器件配置

你可以配置 JTAG 链中的多个器件。

7.2.4.2.1. 管脚连接和指南

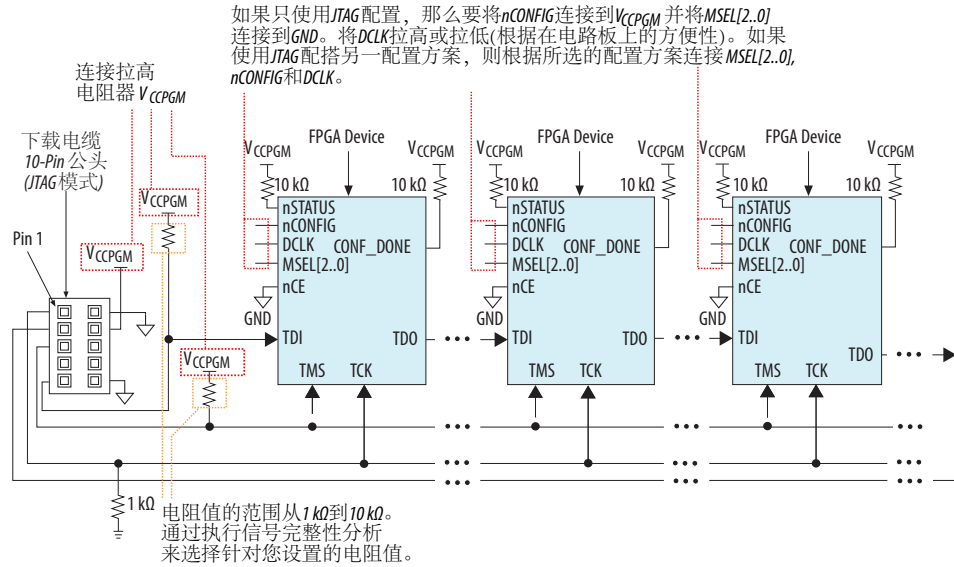
对于该配置设置，请遵循以下管脚连接和指南：

- 分隔 CONF_DONE 和 nSTATUS 管脚，以允许各个器件单独地进入用户模式。
- 兼容 JTAG 的接头被连接到 JTAG 链中的多个器件。链中的器件数仅受下载电缆的驱动能力的限制。
- 如果 JTAG 链中有 4 个或更多器件，通过板上缓冲器来缓冲 TCK，TDI，和 TMS 管脚。也可将 JTAG 支持的其它 Intel FPGA 连接到此链。
- 当系统包含多个器件或当使用边界扫描测试 (BST) 电路测试您的系统时，JTAG 链器件编程是理想的。

7.2.4.2.2. 使用一个下载电缆

下图显示了多器件 JTAG 配置。

图 152. 使用下载电缆时的多器件的 JTAG 配置



相关链接

AN 656: 结合多种配置方案

提供了关于将 JTAG 配置和别的配置方案相结合的更多信息。

7.3. 配置详细信息

这一部分介绍 $MSEL$ 管脚设置、配置序列、器件配置管脚、配置管脚选项和配置数据压缩。

7.3.1. $MSEL$ 管脚设置

将 $MSEL$ 管脚直接连接到 V_{CCPGM} 或 GND ，不需使用任何的上拉或下拉电阻，即可选择出所需的配置方案。

注意:

表 93. Arria 10 器件每种配置方案中 MSEL 管脚设置

- 不要通过微处理器或其他器件驱动 MSEL 管脚。
- 对“通过 HPS 的配置”（Configuration via HPS）使用 PS 或 FPP MSEL 管脚设置。

配置方案	V _{CCPGM} (V)	上电复位(POR)延迟	有效 MSEL[2..0]
基于 JTAG 的配置	—	—	使用以下任何有效的 MSEL 管脚设置
AS (x1 和 x4)	1.8	快速	010
		标准	011
PS 和 FPP (x8, x16 和 x32)	1.2/1.5/1.8	快速	000
		标准	001

注意: 您也可以在 Quartus Prime 的 **Device and Pin Options** 对话框的 **Configuration** 页面中选择配置方案。根据您的选择, 编程文件中的选项位被相应地设置。

相关链接

- [Arria 10 硬核处理器系统技术参考手册](#)
提供了关于通过 HPS 进行配置的详细信息。
- [Arria 10 GX、GT 和 SX 器件系列管脚连接指南](#)
提供了关于 JTAG 管脚电平连接的更多信息。

7.3.2. CLKUSR

CLKUSR 管脚可用作 Arria 10 器件配置和初始化的时钟源。CLKUSR 管脚也可同时用于配置和收发器校准。

对于收发器校准, 根据下表中列出的器件配置方案, 上电时 CLKUSR 必须是一个在 100 MHz 到 125 MHz 之间的自由运行时钟。收发器校准在器件配置期间开始使用 CLKUSR, 即便在器件进入用户模式时也可能继续使用 CLKUSR。

表 94. Arria 10 器件的可用配置时钟源和收发器校准 CLKUSR 频率

Configuration Scheme	用于器件配置的所支持时钟源	用于器件初始化的所支持时钟源	用于收发器校准的所支持 CLKUSR 频率
AS	内部振荡器, CLKUSR	内部振荡器, CLKUSR	100 MHz
PS	DCLK only	内部振荡器, CLKUSR, DCLK	100 到 125 MHz
FPP (x8, x16, x32)			

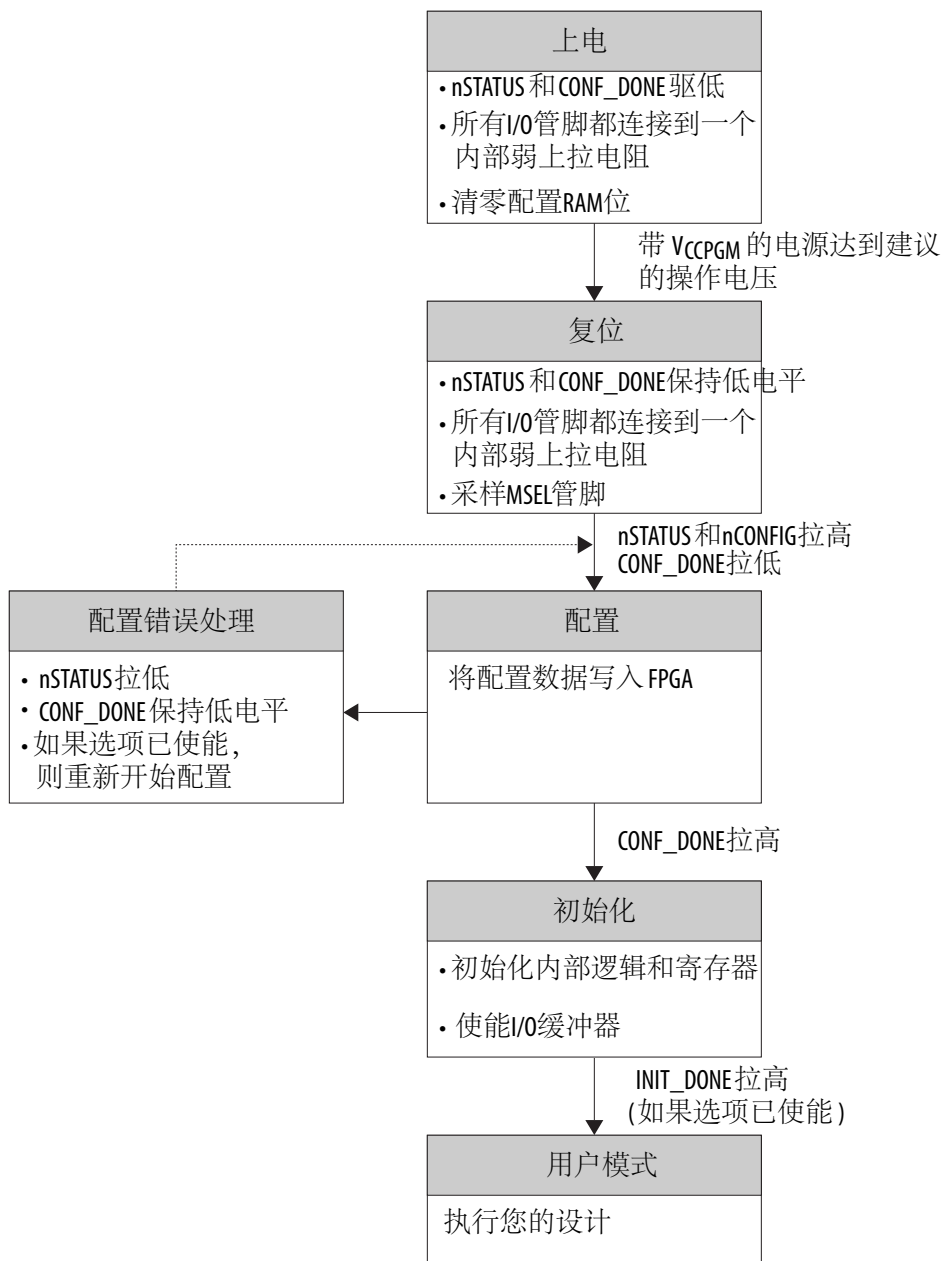
相关链接

- [Arria 10 器件系列管脚连接指南](#)
提供了关于 CLKUSR 管脚的更多信息。

7.3.3. 配置序列

介绍配置序列和每个配置阶段。

图 153. Arria 10 器件的配置序列



通过至少拉低 nCONFIG 管脚到最低 t_{CFG} 低脉冲宽度(除了使用部分重配置操作的配置以外), 您可以启动重配置。当该管脚被拉低时, nSTATUS 和 CONF_DONE 管脚被拉低并且所有 I/O 管脚被连接到一个内部弱上拉电阻。

7.3.3.1. 上电

上电由 POR 电路监控的所有电源。所有的电源,包括 V_{CCPGM} , 必须从 0 V 上电到上电时间规范中所建议的操作电压电平。否则, 保持 nCONFIG 管脚低电平直到所有的电源达到所建议的电压电平。

V_{CCPGM} 管脚

在 Arria 10 器件中, 配置输入缓冲器不必与普通 I/O 缓冲器共享电源线。将 V_{CCPGM} 连接到 1.8 V。

配置期间, 配置输入管脚的操作电压与 I/O bank 电源, V_{CCIO} 无关。因此, Arria 10 器件不需要约束 V_{CCIO} 上的配置电压。

Intel 建议将用于 FPP x8, x16, 和 x32 复用配置管脚的 I/O bank 组电源, V_{CCIO} 连接到 V_{CCPGM} 。

相关链接

- [Arria 10 器件数据表](#)
提供了关于斜升时间规范的更多信息。
- [Arria 10 GX、GT 和 SX 器件系列管脚连接指南](#)
提供了关于配置管脚连接的更多信息。
- [器件配置管脚 \(第 240 页\)](#)
提供了关于配置管脚的更多信息。

7.3.3.2. 复位

POR 延迟是 POR 电路监控的所有电源达到所推荐的操作电压和 nSTATUS 释放为高电平并且 Arria 10 器件开始配置之间的时间延迟。

使用 MSEL 管脚设置 POR 延迟。

用户 I/O 管脚被连接到一个内部弱上拉电阻直到器件被配置。

相关链接

- [MSEL 管脚设置 \(第 231 页\)](#)
- [Arria 10 器件数据表](#)
提供了关于 POR 延迟规范的更多信息。

7.3.3.3. 配置

要了解关于每个配置方案的 DATA[]管脚的更多信息, 请参考相关配置方案。

7.3.3.3.1. 配置错误检测

当 Quartus Prime 生成配置比特流时, 此软件也计算每个 CRAM 帧的 32 位 CRC 值。配置比特流包含每个数据帧的 CRC 值。数据帧的长度根据器件的不同而不同。

配置期间, 一个数据帧加载到 FPGA 时, 预先算出的 CRC 值被移进 CRC 电路。同时, FPGA 中的 CRC 引擎计算数据帧的 CRC 值并且与预先算出的 CRC 值比较。如果两个 CRC 值不匹配, 那么 nSTATUS 管脚被设置为低电平以表示配置错误。



7.3.3.4. 配置错误处理

要自动重启配置, 在 Quartus Prime 的 **Device and Pin Options** 对话框的 **General** 页面上打开 **Auto-restart configuration after error** 选项。

如果您不打开该选项, 那么可以监控 nSTATUS 管脚来检测错误。要重启配置, 将 nCONFIG 管脚拉低至少 t_{CFG} 的持续时间。

相关链接

[Arria 10 器件数据表](#)

提供了关于 t_{STATUS} 和 t_{CFG} 时序参数的更多信息。

7.3.3.5. 初始化

初始化时钟源来自内部振荡器, CLKUSR 管脚或 DCLK 管脚。默认情况下, 内部振荡器为初始化时钟源。如果使用内部振荡器, 那么 器件将有足够的时钟周期进行适当的初始化。

注意: 如果在器件初始化期间使用可选的 CLKUSR 管脚作为初始化时钟源并且拉低 nCONFIG 管脚来重启配置, 那么确保 CLKUSR 或 DCLK 管脚继续翻转直到 nSTATUS 管脚变低然后再变高。

CLKUSR 管脚使您可以灵活地同步多个器件的初始化或延迟初始化。初始化期间, 对 CLKUSR 管脚提供一个时钟不影响配置。在 CONF_DONE 管脚变高后, CLKUSR 或 DCLK 管脚在 t_{CD2CU} 指定的时间后被使能。经过这段时间后, Arria 10 器件需要 T_{init} 指定的最小时钟周期数来正确进行初始化以及进入 t_{CD2UMC} 参数指定的用户模式。

相关链接

[Arria 10 器件数据手册](#)

提供了关于 t_{CD2CU} 、 t_{init} 、 t_{CD2UMC} 时序参数和初始化时钟源的更多信息。

7.3.3.6. 用户模式

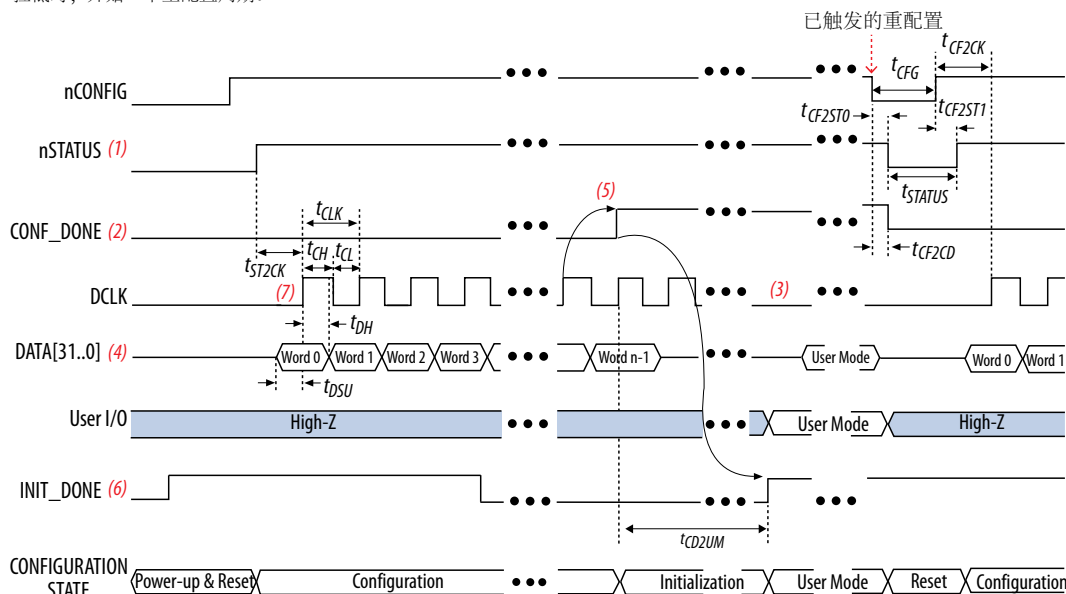
您可以使能可选的 INIT_DONE 管脚来监控初始化阶段。INIT_DONE 管脚被拉高后, 初始化完成并且设计开始执行。用户 I/O 管脚将会按照您所设计运行。

7.3.4. 配置时序波形

7.3.4.1. FPP 配置时序

图 154. 当 DCLK-to-DATA[] Ratio 为 1 时的 FPP 配置时序波形

此波形的开始显示了用户模式下的器件。在用户模式下，nCONFIG、nSTATUS 和 CONF_DONE 在逻辑高电平。当 nCONFIG 拉低时，开始一个重配置周期。

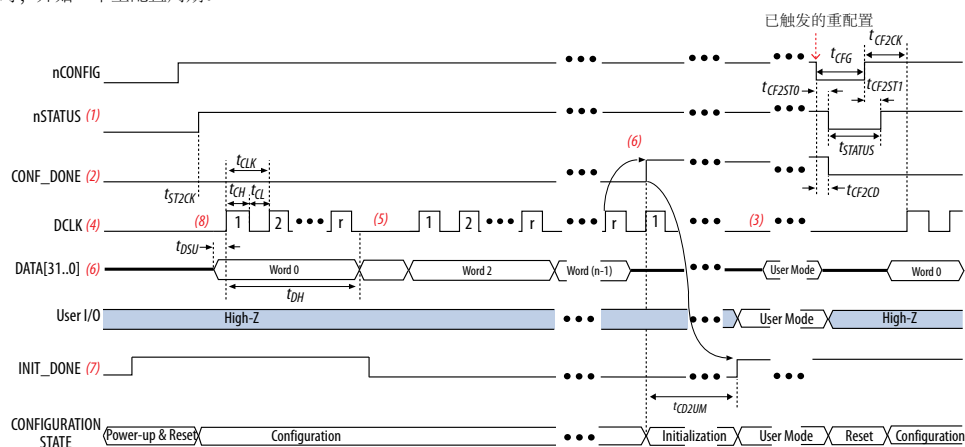


- (1) 上电后，器件在POR延迟时间内保持nSTATUS为低电平。
- (2) 上电后，配置前和配置期间，CONF_DONE为低电平。
- (3) 配置后不要使DCLK悬空。配置完成后忽略DCLK。它可根据需要翻转为高或低电平。
- (4) 对于FPP×16，使用DATA[15..0]。对于FPP×8，使用DATA[7..0]。配置后DATA[31..0]可用作用户I/O管脚。此管脚的状态取决于复用管脚设置。
- (5) 要确保配置成功，需要将完整配置数据发送到器件。在器件成功接收到所有配置数据后，CONF_DONE被释放为高电平。CONF_DONE变高后，发送DCLK上的两个额外的下降沿以开始初始化并进入用户模式。
- (6) 将使能INIT_DONE管脚的选项位配置到器件中后，INIT_DONE变低。
- (7) 拉高nSTATUS前，不要将DCLK翻转为高电平。



图 155. 当 DCLK-to-DATA[] Ratio>1 时的 FPP 配置时序波形

此波形的开始显示了用户模式下的器件。在用户模式下，nCONFIG、nSTATUS 和 CONF_DONE 在逻辑高电平。当 nCONFIG 拉低时，开始一个重配置周期。



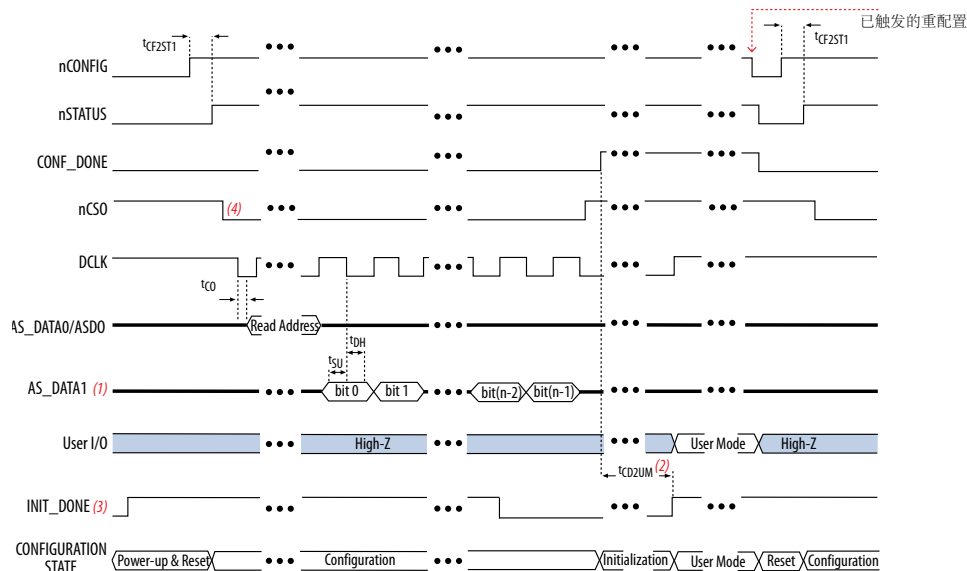
- (1) 上电后，器件在POR延迟时间内保持 nSTATUS 为低电平。
- (2) 上电后，配置前和配置期间，CONF_DONE 为低电平。
- (3) 配置后不要使 DCLK 悬空。可依方便性将其驱高或驱低。
- (4) "r" 代表 DCLK-to-DATA[] 率。而 DCLK-to-DATA[] 率基于解压缩和设计安全功能使能设置。
- (5) 如果需要，通过保持 DCLK 为低电平来暂停 DCLK。当 DCLK 重新开始时，外部主机必须先提供 DATA[31..0] 管脚上的数据，然后发送第一个 DCLK 上升沿。
- (6) 要确保配置成功，需要发送整个配置数据到器件。在器件成功地接收到所有配置数据后，CONF_DONE 被释放为高电平。CONF_DONE 变高后，发送 DCLK 上的两个额外的下降沿以开始初始化并进入用户模式。
- (7) 将使能 INIT_DONE 管脚的选项位配置到器件中之后，INIT_DONE 变低。
- (8) 拉高 nSTATUS 前，不要将 DCLK 翻转为高电平。

相关链接

FPP 配置的 DCLK-to-DATA[] 比率 (r)

7.3.4.2. AS 配置时序

图 156. AS 配置时序波形

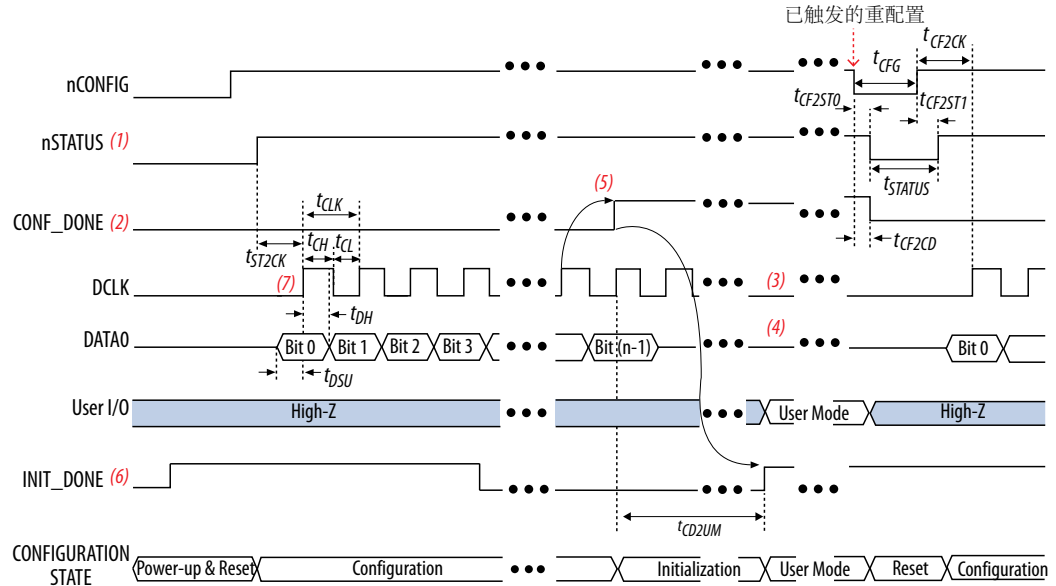


- (1) 如果使用的是 AS×4 模式，此信号表示 AS_DATA[3..0] 且 EPCQ-L 为每个 DCLK 周期发送 4-bit 数据。
- (2) 初始化时钟可来自于内部振荡器或 CLKUSR 管脚。
- (3) 在使能 INIT_DONE 管脚的选项位被配置到器件中以后，INIT_DONE 变低。
- (4) nCS0 下降沿与 DCLK 第一个翻转的时间多于 15ns。

7.3.4.3. PS 配置时序

图 157. PS 配置时序波形

此波形的开始显示了用户模式下的器件。在用户模式下，nCONFIG、nSTATUS 和 CONF_DONE 在逻辑高电平。当 nCONFIG 拉低时，开始一个重配置周期。



- (1) 上电后，器件在POR延迟时间内保持nSTATUS为低电平。
- (2) 上电后，配置前和配置期间，CONF_DONE为低电平。
- (3) 配置后不要使DCLK悬空。可依方便性将其驱高或驱低。
- (4) 配置后DATA0可用作用户I/O管脚。此管脚的状态取决于Device and Pins Option中的复用管脚设置。
- (5) 要确保配置成功，需要发送完整配置数据到器件。在器件成功地接收到所有配置数据后，CONF_DONE被释放为高电平。CONF_DONE变高后，发送DCLK上两个额外下降沿以开始初始化并进入用户模式。
- (6) 将使能INIT_DONE管脚的选项位配置到器件中后，INIT_DONE变低。
- (7) 拉高nSTATUS前，不要将DCLK翻转为高电平。

7.3.5. 估算配置时间

配置时间通常是从 CFI 闪存或 EPCQ-L 器件传输配置数据到 Arria 10 器件所消耗的时间。

使用下面的公式来评估配置时间：

AS 配置

默认情况下，使用 AS x1 模式。Arria 10 器件通过读取编程文件中的选项位来决定 AS 模式。

- AS x1 模式
估算最短配置时间 = .rbf 大小 x (最短 DCLK 周期 / 1 位每 DCLK 循环)
- AS x4 模式
估算最短配置时间 = .rbf 大小 x (最短 DCLK 周期 / 4 位每 DCLK 循环)

PS 配置

估算最短配置时间 = .rbf 大小 x (最短 DCLK 周期 / 1 位每 DCLK 循环)

FPP 配置

估算最短配置时间 = $\text{.rbf 大小} / \text{FPP 数据宽度} \times r \times \text{最短 DCLK 周期}$

r 是 DCLK-to-DATA[] 比率。

注意: 压缩配置数据减少配置时间。基于配置方法和相应的 DCLK 比率，增加的时间会有所不同。

相关链接

FPP 配置的 DCLK-to-DATA[] 比率 (r)

7.3.6. 器件配置管脚

配置管脚汇总

下表列出了 Arria 10 配置管脚和它们的电源。

注意: TDI、TMS、TCK、TDO 和 TRST 管脚由 V_{CCPGM} 供电。

注意: CLKUSR、DEV_OE、DEV_CLRn、DATA[31..1] 和 DATA0 管脚在配置期间由 V_{CCPGM} 供电，并且如果它们被用作用户 I/O 管脚，那么将会由管脚所位于的 bank 的 V_{CCIO} 供电。

表 95. Arria 10 器件的配置管脚总结

配置管脚	配置 方案	输入/输出	用户模式	供电支持
TDI	JTAG	输入	—	V_{CCPGM}
TMS	JTAG	输入	—	V_{CCPGM}
TCK	JTAG	输入	—	V_{CCPGM}
TDO	JTAG	输出	—	V_{CCPGM}
TRST	JTAG	输入	—	V_{CCPGM}
CLKUSR	所有方案	输入	I/O	$V_{CCPGM} / V_{CCIO}^{(32)}$
CRC_ERROR	可选的，所有方案	输出	I/O	$V_{CCPGM} / \text{上拉}$
CONF_DONE	所有方案	双向	—	$V_{CCPGM} / \text{上拉}$
DCLK	FPP 和 PS	输入	—	V_{CCPGM}
	AS	输出	—	V_{CCPGM}
DEV_OE	可选的，所有方案	输入	I/O	$V_{CCPGM} / V_{CCIO}^{(32)}$
DEV_CLRn	可选的，所有方案	输入	I/O	$V_{CCPGM} / V_{CCIO}^{(32)}$
INIT_DONE	可选的，所有方案	输出	I/O	上拉
MSEL[2..0]	所有方案	输入	—	V_{CCPGM}
nSTATUS	所有方案	双向	—	$V_{CCPGM} / \text{上拉}$
nCE	所有方案	输入	—	V_{CCPGM}
nCEO	所有方案	输出	I/O	上拉
nCONFIG	所有方案	输入	—	V_{CCPGM}

继续...



配置管脚	配置 方案	输入/输出	用户模式	供电支持
DATA[31..1]	FPP	输入	I/O	V _{CCPGM} / V _{CCIO} ⁽³²⁾
DATA0	FPP 和 PS	输入	I/O	V _{CCPGM} / V _{CCIO} ⁽³²⁾
nCSO[2..0]	AS	输出	—	V _{CCPGM}
nIO_PULLUP ⁽³¹⁾	所有方案	输入	—	V _{CC}
AS_DATA[3..1]	AS	双向	—	V _{CCPGM}
AS_DATA0 / ASD0	AS	双向	—	V _{CCPGM}
PR_REQUEST	部分重配置	输入	I/O	V _{CCPGM} / V _{CCIO} ⁽³²⁾
PR_READY	部分重配置	输出	I/O	V _{CCPGM} / V _{CCIO} ⁽³²⁾
PR_ERROR	部分重配置	输出	I/O	V _{CCPGM} / V _{CCIO} ⁽³²⁾
PR_DONE	部分重配置	输出	I/O	V _{CCPGM} / V _{CCIO} ⁽³²⁾

相关链接

Arria 10 GX、GT 和 SX 器件系列管脚连接指南
提供了关于每个配置管脚的更多信息。

7.3.6.1. 用于配置管脚的 I/O 标准和驱动强度

用于 Arria 10 器件的标准 I/O 电压是 1.8 V。专用配置 I/O 的驱动强度设置是硬线连接的。配置期间双功能配置 I/O 管脚的默认驱动强度是 1.8V@50Ω。当使能配置管脚，Quartus Prime 软件设置 CVP_CONF_DONE 管脚的驱动强度为 1.8VCMOS 4 mA，且 INIT_DONE 和 CRC_ERROR 管脚的驱动强度为 1.8 VCMOS8mA。

表 96. 用于配置管脚的 I/O 标准和驱动强度

配置管脚	输入/输出	驱动强度
nSTATUS	专用	1.8 V CMOS 4 mA
CONF_DONE	专用	1.8 V CMOS 4 mA
TDO	专用	1.8 V CMOS 12 mA
DCLK	专用	1.8 V CMOS 24 mA
nCSO[2..0]	专用	1.8 V CMOS 12 mA
AS_DATA0 / ASD0	专用	1.8 V CMOS 24 mA
AS_DATA1	专用	1.8 V CMOS 24 mA
AS_DATA2	专用	1.8 V CMOS 24 mA
AS_DATA3	专用	1.8 V CMOS 24 mA
继续...		

(31) 如果将 nIO_PULLUP 管脚连接到 VCC，请确保配置前或配置器件所有用户 I/O 管脚和复用 I/O 管脚都处于逻辑 0，从而避免从 I/O 管脚处抽取额外电流。

(32) 该管脚在配置之前和期间由 V_{CCPGM} 供电，并且如果它被用作用户 I/O 管脚，那么在用户模式期间由 V_{CCIO} 供电

配置管脚	输入/输出	驱动强度
INIT_DONE	双功能	1.8 V CMOS 8 mA
CRC_ERROR	双功能	1.8 V CMOS 8 mA
CvP_CONF_DONE	双功能	1.8 V CMOS 4 mA

7.3.6.2. Quartus Prime 中的配置管脚选项

下表列出了 Quartus Prime 的 **Device and Pin Options** 对话框中的复用配置管脚。

表 97. 配置管脚选项

配置管脚	分类	选项
CLKUSR	General (通用)	Enable user-supplied start-up clock (CLKUSR) (使能“用户提供的开始时钟”)
DEV_CLRN	General (通用)	Enable device-wide reset (DEV_CLRN) (使能“器件范围复位”)
DEV_OE	General (通用)	Enable device-wide output enable (DEV_OE) (使能“器件范围输出使能”)
INIT_DONE	General (通用)	Enable INIT_DONE output (使能“INIT_DONE”输出)
nCEO	General (通用)	Enable nCEO pin (使能“nCEO 管脚”)
CRC_ERROR	Error Detection CRC (错误测试 CRC)	Enable Error Detection (使能“错误测试”) CRC_ERROR pin
		Enable open drain on CRC_ERROR pin (使能“CRC_ERROR 管脚上的开渠”)
		Enable internal scrubbing (使能“内部擦拭”)
PR_REQUEST	General (通用)	Enable PR pin (使能“PR 管脚”)
PR_READY		
PR_ERROR		
PR_DONE		

相关链接

通过 [Quartus Prime 软件浏览印刷电路板结构图](#)

提供了关于器件和管脚选项对话框设置的更多信息。

7.3.7. 配置数据压缩

Arria 10 器件在配置过程中可以接收压缩的配置比特流并且实时解压缩数据。初步数据表明，取决于设计，压缩通常可以减少 30% 到 55% 的配置文件大小。

除了 JTAG 配置方案以外，其他所有配置方案都支持解压缩。

您可以在设计编译之前或之后使能压缩。

注意： 对于所有配置方案，不能同时使能加密和压缩功能。

7.3.7.1. 设计编译前的使能压缩

要在设计编译之前使能压缩，请按照下列步骤操作：

1. 在 Assignment 菜单上，点击 **Device**。
2. 选择您的 Arria 10 器件，然后点击 **Device and Pin Options**。
3. 在 **Device and Pin Options** 窗口中的 **Category** 列表下选择 **Configuration** 并打开 **Generate compressed bitstreams**。

7.3.7.2. 设计编译后的使能压缩

要在设计编译后使能压缩，请按照下列步骤操作：

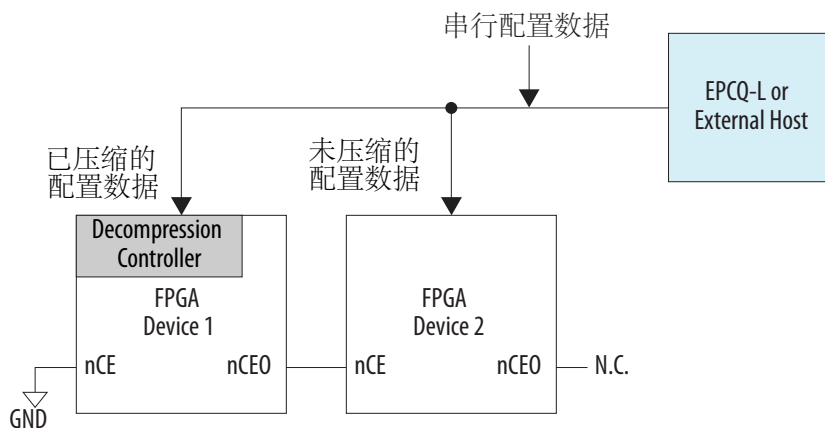
1. 在 File 菜单中，点击 **Convert Programming Files**。
2. 选择编程文件类型(.pof、.sof、.hex、.hexout、.rbf 或 .ttf)。对于 POF 输出文件，选择一个配置器件。
3. 在 **Input files to convert** 列表中，选择 **SOF Data**。
4. 点击 **Add File**，然后选择 Arria 10 器件 .sof。
5. 选择添加到 **SOF Data** 的文件名，然后点击 **Properties**。
6. 打开 **Compression** 复选框。

7.3.7.3. 在多器件配置中使用压缩

下图显示了一个包括两个 Arria 10 器件的链。只有第一个器件的压缩被使能。

这个设置仅由 AS 或 PS 多器件配置支持。

图 158. 相同配置文件中的压缩和未压缩的串行配置数据

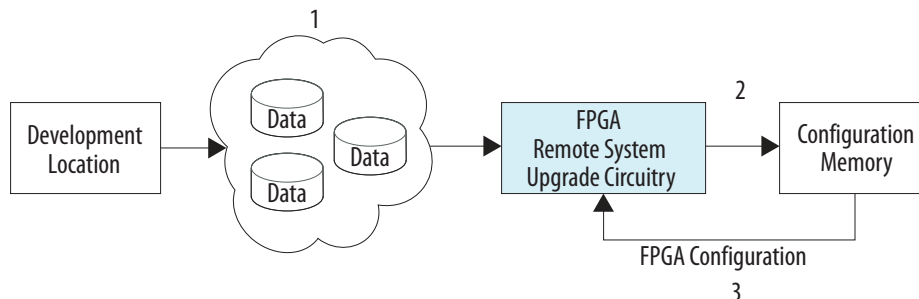


对于 FPP 配置方案，由于 DCLK 到 DATA[] 比率的不同，相同多器件配置链中的压缩和未压缩的配置组合不被支持。

7.4. 使用主动串行方案升级远程系统

Arria 10 器件包含专用远程更新电路。可使用此特性从远程位置升级您的系统。

图 159. Arria 10 远程系统升级框图



您可以设计系统来管理配置器件中的应用配置映像的远程更新。以下是远程系统更新的步骤：

1. Arria 10 器件中的逻辑（嵌入式处理器或用户逻辑）接收来自远程位置的配置映像。可使用通信协议，如 TCP/IP，PCI，用户数据报协议(UDP)，UART，或专属接口将器件连接到远程源。
2. 逻辑存储在非易失性配置存储器的配置映像中。
3. 逻辑开始使用新收到的配置映像重新配置周期。

出现错误时，该电路检测错误且恢复到一个安全配置映像，并在设计中显示错误状态。

7.4.1. 配置映像

Arria 10 器件呈现了一个新的远程系统更新功能，可提供 **direct-to-application** 以及 **application-to-application** 更新。当 Arria 10 器件在远程更新编程模式中上电时，Arria 10 器件加载由 EPCQ-L 器件的 32'd0 地址的起始地址指针指示的出厂或应用配置映像。

系统中的每个 Arria 10 器件都需要一个出厂映像。出厂映像为用户定义的配置映像，包含执行下列操作的逻辑：

- 基于专用远程系统升级电路提供的状态处理错误。
- 与远程主机通信，接收新应用映像并将映像存储在本地非易失性存储器件中。
- 决定要加载到 Arria 10 器件的应用映像。
- 使能或禁用用户看门狗定时器（watchdog timer）并加载其超时值。
- 指示专用远程系统更新电路来开始一个重配置周期。

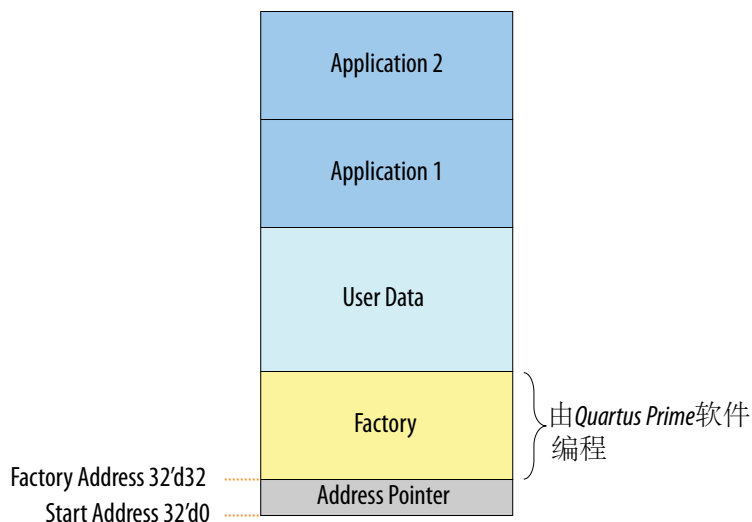
您也可以为器件创建一个或多个应用映像。应用映像包含要在目标器件中实现的所选功能。

将映像存储在 EPCQ-L 器件中的以下位置：

- 出厂配置映像—EPCQ-L 器件上的 PGM[31..0] = 32'h00000020 起始地址。
- 应用配置映像—任意扇区边界。Intel 建议一个扇区边界仅存储一个映像。
- 起始地址(0x00 到 0x1F)—存储 32 位地址指针以在上电后加载应用配置映像。

图 160. 起始地址和出厂地址位置

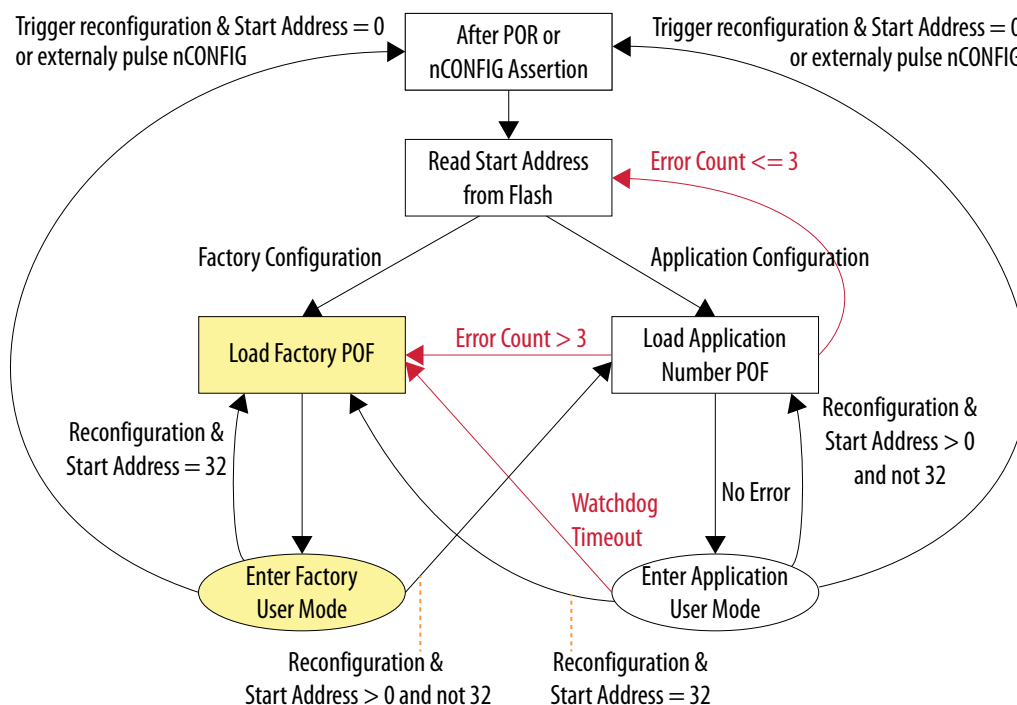
以下结构图显示了出厂、用户数据、应用 1 和应用 2 部分。每一部分从一个新的扇区边界开始。



注意: Intel 建议您设置一个固定的起始地址并不要用户在用户模式下更新该地址。当有新的应用映像时, 您应该仅覆盖已存在的应用配置映像。以此避免在每次更新起始地址时不经意地删除出厂配置映像。

7.4.2. 远程更新模式中的配置序列

图 161. 远程更新模式中的出厂和应用配置之间的跳变



上电后或使用 nCONFIG 触发重配置时，AS 控制器从 EPCQ-L 器件读取起始地址并加载初始配置映像(出厂或应用配置映像)。如果初始镜像是一个应用配置映像并且出现错误，那么控制器在加载出厂配置映像之前将尝试加载相同的初始应用配置映像三次。如果初始应用配置映像遇到一个用户看门狗 (watchdog) 超时错误，那么控制器加载出厂配置映像。您可以在出厂用户模式或应用用户模式期间加载一个新的应用配置映像。如果遇到错误，那么控制器加载出厂配置映像。

注意: 出现错误时，AS 控制器将在恢复出厂映像之前把相同的应用配置映像加载三次。那时，总耗时超出 100ms 并违反了 CvP 使用过程中的 PCIe 启动时间。如果您的设计对 PCIe 启动要求敏感，Intel 建议不要使用 direct-to-application 功能。

相关链接

[远程系统更新状态机](#) (第 249 页)

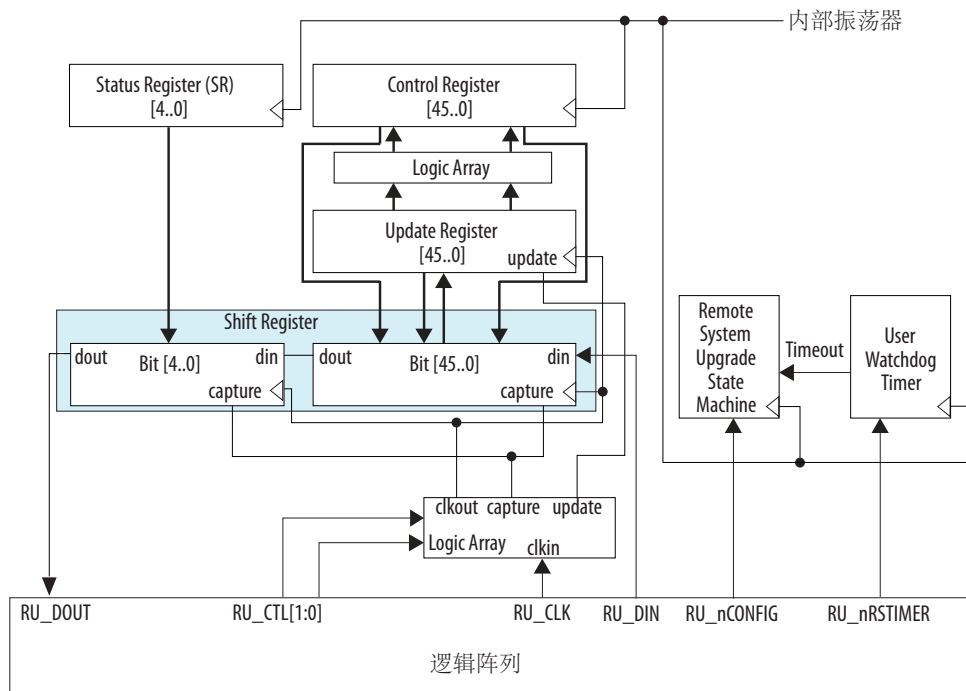
远程更新模式中配置序列的详细说明。

7.4.3. 远程系统更新电路

远程系统更新电路包含远程系统更新寄存器、看门狗定时器（watchdog timer）、和控制这些组件的状态机。

注意: 如果您使用 **Altera Remote Update IP** 内核，那么 IP 内核从内部控制 RU_DOUT，RU_CTL[1:0]，RU_CLK，RU_DIN，RU_nCONFIG 和 RU_nRSTIMER 信号，以执行所有相关的远程系统更新操作。

图 162. 远程系统更新电路



相关链接

Arria 10 器件数据表

提供了关于远程系统更新电路时序规范的更多信息。

7.4.4. 使能远程系统更新电路

要使能远程系统更新功能，需要在 Quartus Prime 的 **Device and Pin Options** 对话框，**Configuration** 页面，配置方案列表中选择 **Active Serial** 或 **Configuration Device**。

Intel 提供的 **Altera Remote Update IP core** 为远程系统更新电路提供一个类存储器接口，并处理 Arria 10 器件逻辑中移位寄存器读和写协议。

相关链接

Altera 远程更新 IP 内核用户指南

7.4.5. 远程系统更新寄存器

表 98. 远程系统更新寄存器

寄存器	说明
移位	该寄存器通过内核逻辑存取，也可以通过用户逻辑对更新寄存器、状态寄存器，和控制寄存器进行写入和采样。
控制	该寄存器包含当前的页面地址、用户看门狗定时器 (watchdog timer) 设置和一个指定当前的配置映像是否出厂配置映像还是应用配置映像的位。该寄存器用于 AS 控制器以在远程系统更新过程中加载 EPCQ-L 器件的配置映像。
更新	该寄存器包含与控制寄存器相似的数据，出厂配置映像及应用配置映像通过把数据移位到移位寄存器后，更新寄存器被更新。远程系统更新的软 IP 内核使用下一个重配置周期过程中用于控制寄存器的值更新该寄存器。
状态	每个重配置周期过程中，该寄存器由远程更新模块写入以记录重配置的触发。远程系统更新的软 IP 内核通过这一信息决定重配置周期之后的相应操作。

相关链接

- [控制寄存器](#) (第 248 页)
- [状态寄存器](#) (第 249 页)

7.4.5.1. 控制寄存器

表 99. 控制寄存器位

位	名称	复位值 ⁽³³⁾	说明
0	AnF	1'b0	非出厂应用位。表示目前加载到器件中的配置映像类型； 0 表示出厂映像， 1 表示应用映像。当该位为 1 时，控制寄存器的访问被限于只读并且看门狗定时器 (watchdog timer) 被使能。使用应用配置映像触发重配置之前，出厂配置设计必须将该位设置为 1 。
1..32	PGM[0..31]	32'h00000000	AS 配置起始地址。
33	Wd_en	1'b0	用户看门狗定时器 (watchdog timer) 使能位。将该位设置为 1 以使能监视计时器。
34..45	Wd_timer[11..0]	12'h000	用户看门狗 (watchdog) 超时值。

⁽³³⁾ 这是器件退出 POR 之后以及重配置恢复到出厂配置镜像期间的默认值。



7.4.5.2. 状态寄存器

表 100. 状态寄存器位

位	名称	复位值 ⁽³⁴⁾	说明
0	CRC	1'b0	当设置为 1 时，表明应用配置期间的 CRC 错误。
1	nSTATUS	1'b0	当设置为 1 时，表明由于一个错误，nSTATUS 由外部器件置位。
2	Core_nCONFIG	1'b0	当设置为 1 时，表明重配置已经由器件的逻辑阵列触发。
3	nCONFIG	1'b0	当设置为 1 时，表明 nCONFIG 被置位。
4	Wd	1'b0	当设置为 1 ，表明用户看门狗 (watchdog) 超时。

7.4.6. 远程系统更新状态机

远程系统更新状态机的操作如下所示：

1. 上电后，远程系统更新寄存器被复位为 **0** 并且出厂或应用配置映像基于存储在 EPCQ-L 器件的 0x00 到 0x1F 的起始地址被加载。
2. 在出厂配置映像中，用户逻辑设置 ANF 位为 **1** 以及加载应用映像的起始地址。用户逻辑也写入看门狗定时器 (watchdog timer) 设置。
3. 当配置复位(RU_CONFIG)走低时，状态机使用更新寄存器的内容更新控制寄存器，并且使用应用配置映像触发重配置。
4. 如果发生错误，状态机返回到出厂映像。控制和更新寄存器被复位到 **0**，并且状态寄存器由该错误信息更新。
5. 成功的进行重配置之后，系统保持在应用配置。

7.4.7. 用户看门狗定时器 (User Watchdog Timer)

用户看门狗定时器防止由于意外停止器件而出现的错误的应用配置。在应用配置被成功的加载到器件后，您可以使用定时器检测功能错误。定时器在出厂配置中被自动禁用；在应用配置中被自动使能。

注意：如果在应用配置中不需要该功能，那么您需要在出厂配置用户模式操作期间，通过设置更新寄存器的 Wd_en 位为 **1'b0** 而关闭该功能。您不可以在应用配置中禁用该功能。

计数器为 29 位宽，具有 2^{29} 的最大计数值。当指定用户看门狗定时器值时，仅指定最重要的 12 位。定时器设置的粒度是 2^{17} 个周期。周期时间基于用户看门狗定时器内部振荡器的频率。

定时器在应用配置进入用户模式后开始计数。定时器超时后，远程系统更新电路生成超时信号，更新状态寄存器，并触发出厂配置映像的加载。要复位时间，请置位 RU_nRSTIMER。

相关链接

Arria 10 器件数据表

提供了关于用户看门狗 (watchdog) 内部振荡器频率的操作范围的更多信息。

⁽³⁴⁾ 器件退出 POR 和上电后，状态寄存器内容是 5'b00000。

7.5. 设计安全

Arria 10 设计安全特性支持以下功能：

- 增强内置的高级加密标准(AES)解密模块支持 256 位密钥 工业标准 设计安全算法(FIPS-197 认证)
- 易失性和非易失性密钥编程支持
- 防篡改模式下的易失性和非易失性密钥安全操作模式
- JTAG 安全模式上电期间，可访问 JTAG 指令受限
- 支持 POF 验证以及防止旁路攻击
- 通过熔丝位或选项位提供 JTAG 访问控制和安全密钥控制
- 禁止上电时的所有 JTAG 指令直到器件被初始化
- 支持板级测试
- 支持非易失性密钥的板外密钥编程
- 独立的 Qcrypt 工具结合其他安全设置进行加密和解密从而配置比特流。
- 用于除了 JTAG 以外的所有配置方案
- 支持远程系统更新功能

表 101. Arria 10 FPGA 的设计方法

设计安全性组件	说明
Non-Volatile key	非易失性密钥安全地存储于器件内熔丝中。专有的安全特性使密钥判定变得困难。
Volatile Key	易失性密钥安全地存储于器件内电池供电的 RAM 中。专有的安全特性使得该密钥很难被确认。
Key Generation	用户提供的 256-bit 密钥先经单向函数处理然后才被编程到器件中
Key Choice	器件中可同时存在易失性和非易失性密钥。用户通过设置 Convert Programming File 工具或 Qcrypt 工具中已加密配置文件的选项位来选择使用哪种密钥。
Tamper Protection Mode	防篡改模式可防止未加密配置文件载入 FPGA。使能此模式时，仅可将以您的密钥加密的配置载入 FPGA。未加密的配置或经错误密钥加密的配置都将导致配置失败。可设置器件内部熔丝使能此模式。
Configuration Readback	这些器件不支持配置读回功能。出于安全性角度，使未加密配置数据无法读回。
Security Key Control	通过使用各种 JTAG 指令和 Qcrypt 工具中的安全选项，可灵活选择永久或暂时禁用易失或非易失性密钥。您还可选择锁定易失性密钥来避免器被覆盖或重新编程。
JTAG Access Control	使用 Qcrypt 工具通过设置配置文件中的 OTP 熔丝或选项位使能各级别 JTAG 访问控制： 1. 强制完整配置或部分配置仅通过 HPS 完成。 2. 旁路外部 JTAG 管脚或 HPS JTAG。该功能禁用外部 JTAG 或 HPS JTAG 访问，但可通过内核访问解锁该功能。 3. 禁用与所有 AES 密钥相关的来自外部 JTAG 管脚的 JTAG 指令。 4. 仅允许通过外部 JTAG 访问限定组的强制性 JTAG 指令，与 JTAG Secure 模式类似。

- 注意：**
- 所有配置方案都不可同时使能加密和压缩。
 - 当您设计安全性用于 FPP 配置方案中的 Arria 10 器件时，需要不同的 DCLK-to-DATA [] 比率。



相关链接

[AN556: 使用 Intel FPGA 中的设计安全特性](#)

提供更多关于在 Arria 10 器件中应用安全特性的信息。

7.5.1. 安全密钥类型

Arria 10 器件提供两种密钥类型—易失性和非易失性。下表列出了易失性和非易失性密钥之间的区别。

表 102. 安全密钥类型

密钥类型	密钥编程性	密钥 存储的电源	编程方法
易失性	<ul style="list-style-type: none">可重新编程的可擦除的	需要外部电池 V_{CCBAT} 。(35)	电路板上
非易失性	一次性编程	不需要外接电池	板上和 in-socket 编程 (36)

非易失性和易失性密钥编程使设计免于逆向工程和非法复制。如果设置为防篡改保护模式，则设计也会被保护免于篡改。

相关链接

- [AN556: 使用 Intel FPGA 中的设计安全特性](#)
提供更多关于将易失性和非易失性密钥编程到 FPGA 的信息。
- [Arria 10 GX、GT 和 SX 器件系列管脚连接指南](#)
提供了关于 V_{CCBAT} 管脚连接建议的更多信息。
- [Arria 10 器件数据表](#)
提供了关于电池规范的更多信息。
- [所支持的 JTAG 指令](#) (第 271 页)

7.5.2. 安全模式

表 103. Arria 10 器件中可用的安全模式

注意: 更多关于这些指令或如何熔断熔丝进入各模式的详情，请联络您的 Intel 技术支持。或者，可使用 Qcrypt 工具使能全部设计安全模式。相对于燃烧熔丝的一次性编程限制，Qcrypt 工具提供一个非永久性解决方案。

安全模式	JTAG 指令	安全特性
JTAG Secure ⁽³⁷⁾	EXT_JTAG_SECURE	仅允许强制性 IEEE Std. 1149.1 BST JTAG 指令。请参看表 104 (第 252 页)。
Tamper Protection	OTP_VOLKEY_SECURE	仅允许经正确密钥加密的配置文件载入 Arria 10 器件。未加密或经错误密钥加密将导致配置失败。
继续...		

(35) V_{CCBAT} 是易失性密钥存储的专用电源。不论片上电源条件如何， V_{CCBAT} 都会连续地对易失性寄存器供电。

(36) 第三方供应商提供 in-socket 编程。

(37) 使能 JTAG Secure 或 Test Disable 模式禁用 Arria 10 器件中的测试模式并禁用通过 JTAG 接口编程。此过程不可逆但会阻止 Intel 进行故障分析。

安全模式	JTAG 指令	安全特性
JTAG Bypass	EXTERNAL_JTAG_BYPASS	禁用所有来自外部 JTAG 管脚或 HPS JTAG 的直接控制。相对于 JTAG Secure 模式，JTAG Bypass 模式中的器件可通过内部 JTAG 核访问外部 JTAG 管脚或 HPS JTAG 接口
Key Related Instruction Disable	KEY_EXT_JTAG_DISABLE	禁用外部 JTAG 管脚发布的所有与 AES 密钥有关的 JTAG 指令。
HPS Configuration Only	FORCE_HPS_CONFIG	从配置或部分重配置器件禁用外部 JTAG 管脚。仅 HPS 控制配置管脚且 MSEL 管脚将处于被动模式。
HPS JTAG Bypass	EXTERNAL_JTAG_BYPASS	旁路 HPS JTAG 控制器并禁用 HPS 内部主控制器。
PR and Scrubbing Disable	PR_SCRUBBING_DISABLE	禁用部分重配置以及来自外部管脚和 HPS 的外部擦拭。仅 FPGA 核可执行部分重配置。
Volatile Key Lock	VOLKEY_LOCK	锁定已归零或被重配置的易失性密钥。然而，也可使用 KEY_CLR_VREG 指令擦除易失性密钥。仅易失性密钥被编程到器件中之后才可发布 VOLKEY_LOCK 指令。
Volatile Key Disable	VOLKEY_DISABLE	禁用易失性密钥的进一步编程。如果有易失性密钥已被编程到器件中，则该器件将不用于解密配置文件。
Non-Volatile Key Disable	OTP_DISABLE	禁用非易失性密码进一步编程。如果有非易失性密钥已被编程到器件中，则该器件将不用于解密配置文件。
Test Disable Mode	TEST_DISABLE	禁用所有测试模式以及所有测试相关的 JTAG 指令。此过程不可逆且会阻止 Intel 进行故障分析。

相关链接

Arria 10 硬核处理器系统技术参考手册中的 SoC 安全

提供更多关于仅 HPS 配置和 HPS JTAG Bypass 安全模式的信息。

7.5.2.1. JTAG 安全模式

Arria 10 器件处于 JTAG Secure 模式时，除强制性 IEEE Standard JTAG 1149.1 BST JTAG 指令外，所有 JTAG 指令为禁用

表 104. 强制性和非强制性 IEEE Standard 1149.1 BST JTAG 指令

强制性 IEEE Standard 1149.1 BST JTAG 指令	非强制性 IEEE Standard 1149.1 BST JTAG 指令
<ul style="list-style-type: none"> BYPASS EXTEST IDCODE LOCK UNLOCK SAMPLE/PRELOAD SHIFT_EDERROR_REG 	<ul style="list-style-type: none"> CONFIG_IO CLAMP EXTEST_PULSE ⁽³⁸⁾ EXTEST_TRAIN ⁽³⁸⁾ HIGHZ KEY_CLR_VREG KEY_VERIFY ⁽³⁸⁾ PULSE_NCONFIG USERCODE

注意: 发布 EXT_JTAG_SECURE 指令后，无法解锁 Arria 10 器件。

相关链接

所支持的 JTAG 指令 (第 271 页)

⁽³⁸⁾ JTAG Secure 模式期间可执行这些 JTAG 指令。

7.5.3. Arria 10 Qcrypt 安全工具

Qcrypt 工具是独立的加密工具用于 Arria 10 FPGA 配置比特流文件的加密和解密。Qcrypt 工具还可通过脚本用于 HPS 引导映像加密。从 Quartus Prime 图形界面无法访问的各种安全性设置现在都可通过 Qcrypt 工具实现。

Qcrypt 工具仅加密和解密原始二进制文件（.rbf）而不是其他配置文件，如 .sof 和 .pof 文件。整个加密流程中，Qcrypt 工具在加密 .rbf 文件时会生成一个认证标签。此认证标签可防止对配置比特流文件的任何修改或篡改。除了加密和解密之外，Qcrypt 工具允许使能和设定各种安全性和设置。通过将安全性和设置并入 .rbf 文件，可在 Arria 10 器件上灵活地使用各种安全性和功能而不必永久性地燃断安全熔丝。要生成 .ekp 文件或加密的配置文件而非 .rbf，您必须使用 Quartus Prime Convert Programming File 工具。

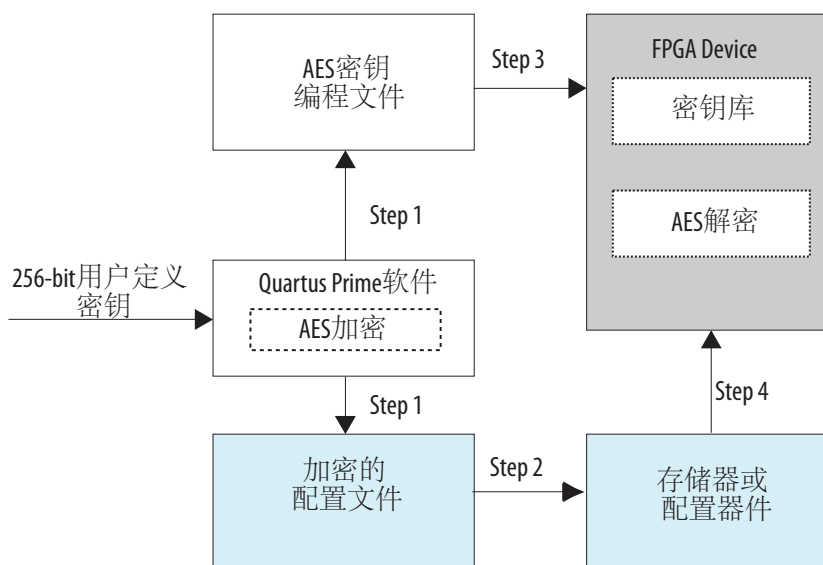
注意: Qcrypt 工具无许可证保护因而所有 Quartus Prime 软件用户都可使用。

相关链接

- [AN556 的 Qcrypt 工具项：使用 Intel FPGA 中的设计安全特性](#)
提供更多关于 Qcrypt 工具特性的信息。
- [AN 759: Arria 10 SoC 安全引导用户指南](#)
提供更多关于加密 HPS 引导映像的信息。
- [AN556：使用 Intel FPGA 中的设计安全特性](#)
提供更多关于在 Arria 10 器件中应用安全特性的信息。

7.5.4. 设计安全实现步骤

图 163. 设计安全实现步骤





要进行安全配置，请遵循以下步骤：

1. Quartus Prime 生成设计安全性密钥编程文件，并且使用用户自定义 256 位安全密钥加密配置数据。
2. 将加密的配置文件存储在外部存储器中。
3. 通过 JTAG 接口将 AES 密钥编程文件编程到 Arria 10 器件。
4. 配置 Arria 10 器件。系统上电时，外部存储器器件将加密的配置文件发送到 Arria 10 器件。

相关链接

[AN556: 使用 Intel FPGA 中的设计安全特性](#)

提供更多关于在 Arria 10 器件中应用安全特性的信息。

7.6. 文档修订历史

日期	版本	修订内容
2017 年 3 月	2017.03.15	<ul style="list-style-type: none">• 更名为 Intel。• 在 Arria 10 器件配置管脚摘要列表中为 nIO_PULLUP 管脚添加了注释。• 两个器件接收到不同配置数据集时，使用外部主机的多器件 FPP 配置图示中，在 nCEO 和 nCE 间添加上了拉电阻。• 链中两个器件接收到不同配置数据集时的多器件 AS 配置图示中，在 nCEO 和 nCE 间添加上了拉电阻。• 两个器件接收到不同配置数据集时的多器件 PS 配置图示中，在 nCEO 和 nCE 间添加上了拉电阻。
2016 年 10 月	2016.10.31	<ul style="list-style-type: none">• 更新了配置管脚的驱动强度：<ul style="list-style-type: none">— DCLK—从 1.8 V CMOS 12 mA 到 1.8 V CMOS 24 mA。— NCSO[2..0]—从 1.8 V CMOS 8 mA 到 1.8 V CMOS 12 mA。— AS_DATA0/ASD0, AS_DATA1, AS_DATA2 和 AS_DATA3—从 1.8 V CMOS 8 mA 到 1.8 V CMOS 24 mA。
2016 年 6 月	2016.05.13	<ul style="list-style-type: none">• 更新了设计安全特性和方法。• 把 EX_JTAG_SECURE 更新为 EXT_JTAG_SECURE 的实例• 添加了强制性和非强制性 IEEE Standard 1149.1 JTAG BST 指令的列表。• 更新了 Arria 10 器件中可用的安全模式及相应使能指令。• 添加了 Qrypt 安全工具信息。
2016 年 5 月	2016.05.02	<ul style="list-style-type: none">• 添加了 FPP 和 PS 配置时间估算到 <i>Estimating Configuration Time</i> 并删除了 <i>Configuration Details</i> 部分的小节。• 添加了关于使用 direct-to-application 时可能的 PCIe 时序违规的注释。• 添加了建议用户设置一个固定的配置映像起始地址的注释。• 添加了配置管脚的 I/O 标准和驱动强度部分。• 更新了 AS 配置时序波形以包含 nCSO。• 更新了 AS 配置时序波形中的 T_{SU} 和 T_{DH}。
2015 年 12 月	2015.12.14	<ul style="list-style-type: none">• 更新了 CLKUSR 信息。• 将 CLKUSR 部分从主动串行配置移到配置详情。
2015 年 11 月	2015.11.02	<ul style="list-style-type: none">• 将配置模式更新成配置方案，以符合一致性。• 添加了 MSEL 管脚设置的链接到 Arria 10 Hard Processor System Technical Reference Manual。• 合并了 MSEL 管脚设置表中的 PS 和 FPP 行。这两种方案具有相同的 MSEL 管脚设置。• 对 MSEL 管脚设置表添加了描述，通过 HPS 的配置使用 PS 或 FPP MSEL 管脚设置。
继续...		



日期	版本	修订内容
		<ul style="list-style-type: none"> 更新了配置模式和特性表，包括"Yes"在 JTAG, AS 和 PS 配置模式中的部分重配置，以及包括一个脚注，提及了只有部分重配置配置为内部主机时。 更新了注释："对于所有配置方案，不能同时使用压缩和加密。 更新了时序波形 FPP, AS 和 PS，包括了预上电状态。 删除了步骤：在 Quartus II 软件的 Device and Pin Options 对话框的 Configuration 页面中的配置模式列表中选择 Remote。 增添了注释：将 MSEL 管脚设置为主动串行，以防止 SFL 编程期间 EPCQ-L ID 读取失败。 将 <i>Quartus II</i> 更改成 <i>Quartus Prime</i>。
2015 年 5 月	2015.05.04	<ul style="list-style-type: none"> 增添了 FPP, AS 和 PS 配置的时序波形。 将 'Trace Length and Loading' 更新成 'Trace Length Guideline'，并删除了加载内容。 添加了 Arria 10 Device Datasheet 的链接，以及获得加载信息。 更新了 'Configuration Modes and Features of Arria 10 Devices' 中 FPP 支持 8 和 32 比特。 添加了 'Design Security' 和 'Configuration Data Compression' 注释，不能同时使用压缩和加密。
2015 年 1 月	2015.01.23	<ul style="list-style-type: none"> 更新了 100 MHz 上的 AS 配置期间的 CLKUSR 管脚使用 更新 PS, FPP x8, FPP x16 和 Configuration via HPS 的最大时钟速率从 125 MHz 到 100 MHz。 更新了 Remote System Upgrade Circuitry 图，替换了 RU_SHIFTnLD and RU_CAPTnUPDT to RU_CTL[1:0]。 将 ALTREMOTE_UPDATE megafunction 更新成 Altera remote Update IP Core。 将用户看门狗超时值从 34..46 更新成 34..45。 更新了 nIO_PULLUP 由 V_{CC} 供电。 添加了 Arria 10 器件表中的 Max Data Rate in Configuration Modes and Features 的注释。
2014 年 8 月	2014.0818	<ul style="list-style-type: none"> 添加了具有多个 EPCQ-L 器件的主动串行配置部分。 删除了独特的芯片 ID 部分。 更新了 JTAG 配置部分，将 USB-Blaster 下载电缆支持的详细信息包含在内。 更新了上电部分。 更新了配置镜像部分，将起始地址包含在内。 更新了远程更新模式中的配置序列部分。 更新了“远程系统更新状态机”部分。 更新了图 7-18：使用微处理器进行单一器件的 JTAG 配置中的 JTAG 管脚的电源参考。 更新了图 7-20：Arria 10 器件的配置序列。 更新了图 7-22：Arria 10 远程系统更新结构图。 更新了表 7-1：Arria 10 器件的配置模式和功能，其中更新了部分重配置的所支持的时钟速率。 更新了表 7-3：Arria 10 器件的每个配置方案的 MSEL 管脚设置，以将所支持的 V_{CCPGM} 电压添加到 FPP 和 PS 配置方案中。 更新表 7-6：远程系统更新寄存器，其中更新了对移位、控制、更新和状态寄存器的说明。 更新了表 7-7：控制寄存器位。 删除了独特的芯片 ID 部分。
2013 年 12 月	2013.1202	首次发布。



8. Arria 10 器件的 SEU 缓解

8.1. SEU 缓解概述

单粒子翻转(SEU)指器件或系统中存储元件的状态发生了变化,通常发生在 SRAM 中,这是由于宇宙辐射影响所造成的。这个状态是一个软错误,将存储元件的状态改回其原始值,就可以修复这个错误,而且不会对器件本身造成永久性损坏。由于非预期的存储状态,该器件可能会错误地进行操作,直到修复了这个翻转。

软错误率(SER)表达为及时故障(FIT)单元,定义为每运行一百万小时出现一次软错误。由于发生的几率很低,所以通常不需要考虑 SEU 缓解。然而,对于高度复杂的系统,例如包含多个高密度器件,错误率可能成为一个主要的系统设计因素。如果您的系统包括多个 FPGA,并且要求非常高的可靠性和可用性,那么您应该考虑到软错误带来影响,利用现有的技术来检测这些类型的错误,并且从这些错误中恢复。

相关链接

- [介绍单粒子翻转](#)
- [了解 FPGA 设计中的单粒子功能性中断](#)
- [Arria 10 器件手册: 已知问题](#)
列出关于 *Arria 10 器件手册* 各章节的计划更新。
- [AN 737: Arria 10 器件中的 SEU 检测和恢复](#)
通过参考设计说明 *Arria 10 SEU 检测和恢复* 的实现。

8.1.1. SEU 缓解应用

Arria 10 SEU 缓解特性能够帮助确保系统一直保持正常运行,避免由 SEU 事件导致的功能异常,或是解决关键的系统级 SEU 事件。通常需要使用 SEU 缓解功能的系统如下:

- 军事或航天一飞行系统
- 汽车或工业一安全应用
- 电信,数据中心或云计算一高系统运行时间

8.1.2. 配置 RAM

FPGA 在用户逻辑(大容量存储器和寄存器)以及配置 RAM (CRAM)中存储数据。CRAM 是存储用户设计的存储器。CRAM 配置器件中的所有逻辑和布线。如果 SEU 发生在一个 CRAM 比特位中,但此 CRAM 比特没有被使用,那么这次 SEU 将是无害的。然而,如果它影响了关键逻辑内部信号布线(例如查找表比特位),那么可能会出现功能性错误。



8.1.3. 嵌入式存储器

Arria 10 器件包括两种类型的存储器模块：

- 20 Kb M20K 模块—专用存储器资源的模块。M20K 模块是大型存储器阵列的理想选择，同时也提供大量的独立端口。
- 640 比特存储器逻辑阵列模块(MLAB)—从复用逻辑阵列模块(LAB)配置的增强型存储器模块。MLAB 适用于宽而浅的存储器阵列。MLAB 被优化用于实现数字信号处理 (DSP)的移位寄存器，宽而浅的 FIFO 缓冲器和滤波器延迟线。每个 MLAB 都是由 10 个自适应逻辑模块(ALM)组成。在 Arria 10 器件中，您可以将这些 ALM 配置成 10 个 32 x 2 模块，这样每个 MLAB 中有一个 32 x 20 简单双端口 SRAM 模块。

嵌入式存储器会受到 SEU 影响，Intel 实现了交错和特殊布技术以最小化 FIT 率，并增加纠错码 (Error Correction Code, ECC) 功能以降低 SEU FIT 率至接近于零。

相关链接

[Arria 10 器件中的嵌入式存储模块](#)

8.2. Arria 10 缓解技术

Arria 10 器件提供各种单粒子翻转(SEU)缓解方法，以适用于不同的应用领域。

表 105. Arria 10 器件的 SEU 缓解领域和方法

领域	SEU 缓解方法
硅芯片设计： CRAM/SRAM/触发器	Intel 使用各种设计技术来降低翻转和/或限制可纠正的双比特错误。
错误检测循环冗余检验(EDCRC) / Scrubbing	您可以使能 EDCRC 功能来检测 CRAM SEU 事件和 CRAM 内容的自动纠正。
M20K SRAM 模块	Intel 实现了交互的，特殊布局技术和纠错码 (Error Correction Code, ECC) 以 SEU FIT 率降低至近乎于零。
敏感度处理(Sensitivity processing)	您可以使用敏感度处理特性来识别 CRAM 比特中的 SEU 是否是一个使用过的或未使用的比特。
故障注入(Fault injection)	您可以使用故障注入特性通过改变 CRAM 状态来触发一个错误来验证对 SEU 事件的系统响应。
分级标示(Hierarchical tagging)	敏感度处理和故障注入特性的一个补充性能，用于报告 SEU 和限制对设计逻辑具体部分的注入。
三模冗余(Triple Modular Redundancy (TMR))	您可以在诸如状态机的关键逻辑上实现 TMR 技术。

8.2.1. 存储器模块纠错码支持

ECC 使您能够检测并纠正存储器输出上的数据错误。ECC 能够执行 32-bit 字的单一错误纠正，双邻错误纠正和三邻错误检测。然而，ECC 无法检测四个或更多错误。

当 M20K 模块处于×32 宽简单双端口模式中时，它具有对 ECC 的内置支持：

- 当使用 ECC 时，M20K 要比非 ECC 简单双端口模式运行的慢。然而，您可以在输出解码器之前使能可选的 ECC 流水线寄存器，实现比非流水线模式更高的性能，但要以一个周期的延迟为代价。
- M20K ECC 状态与两个 ECC 状态标志信号（e（错误）和 ue（不可纠正的错误））进行通信。状态标志是存储器模块的常规输出的一部分。当使用 ECC 时，您不能访问两个奇偶校验位，因为 ECC 状态标志将它们替换了。

相关链接

[存储模块纠错编码支持](#)

8.2.2. CRAM 的错误检测和纠正

8.2.2.1. 错误检测循环冗余检查

用户模式中，已配置的配置 RAM (CRAM)位的内容会受到软错误影响。这些由电离粒子引起的软错误在 IntelFPGA 器件中并不常见。然而，高可靠性应用要求无误的器件操作，于是需要在设计中考考虑到这些错误。

固定片上 EDCRC 电路允许您执行以下操作且不影响器件的布线或性能：

- 配置期间循环冗余校验(CRC)错误的自动检测。
- 用户模式中可选的软错误（SEU 和多位翻转）检测和识别。
- 快速检测软件错误。错误检测速度已提高。
- check-bits 的两种类型：
 - 基于帧的校验位—存储在 CRAM 中并用于验证帧的完整性。
 - 基于列的校验位—存储在寄存器中并用于保护所有帧的完整性。

用户模式下的错误检测期间，一些 EDCRC 引擎因 Arria 10 器件并列运行。错误检测 CRC 引擎的数目取决于帧的长度——帧中的比特总数。

每个基于列的错误检测 CRC 引擎从每帧中读取 128 个位，并在四个周期内处理。要检测到错误，错误检测 CRC 引擎需要读回所有的帧。

图 164. 用户模式中错误检测的结构图

下图显示了用户模式中的寄存器和数据流程。

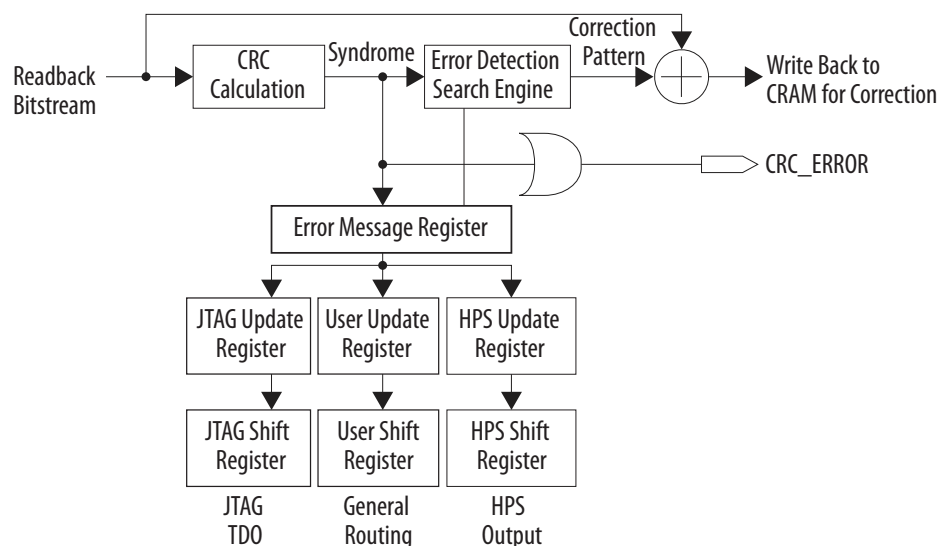


表 106. 错误检测寄存器

名称	说明
错误信息寄存器(EMR)	包含单比特和双相邻错误的详细信息。每当错误检测电路检测到一个错误时就会更新此寄存器。
用户更新寄存器	该寄存器的内容被验证后的一个时钟周期，该寄存器被自动地更新为 EMR 的内容。用户更新寄存器包含一个时钟使能，它必须在其内容被写入到用户移位寄存器之前被置位。这个要求确保用户更新寄存器不被重写(当其内容在被 用户移位寄存器读取时)。
用户移位寄存器	此寄存器允许用户逻辑通过内核接口访问用户更新的寄存器内容。可以使用 Altera Error Message Register Unloader IP 内核通过用户移位寄存器移出 EMR 信息。请参阅相关信息了解详情。
JTAG 更新寄存器	该寄存器的内容被验证后的一个时钟周期，该寄存器被自动地更新为 EMR 的内容。JTAG 更新寄存器包含一个时钟使能，它必须在其内容被写入到 JTAG 移位寄存器之前被置位。这个要求确保 JTAG 更新寄存器不被重写 (当其内容在被 JTAG 移位寄存器读取时)。
JTAG 移位寄存器	该寄存器使您能够使用 SHIFT_EDERROR_REG JTAG 指令通过 JTAG 接口访问 JTAG 更新寄存器的内容。
硬核处理器系统 (HPS) 更新寄存器	此寄存器内容被验证后就自动更新为 EMR 一个时钟周期的内容。(HPS) 更新寄存器包含一个已使能的时钟，且必须在其内容被写入 HPS 移位寄存器前将它置位。此要求确保了 HPS 更新寄存器不会在 HPS 移位寄存器读取其内容时被重写。
HPS 移位寄存器	该寄存器使您能够通过 HPS 接口访问 HPS 更新寄存器的内容。

相关链接

- [Altera 错误消息寄存器卸载器 IP 内核用户指南](#)
提供关于使用用户移位寄存器移出 EMR 的更多信息。
- [Arria 10 硬核处理器系统技术性参考手册中关于 FPGA 管理器地址映射和寄存器的定义。](#)
提供关于使用硬核处理器系统读取错误检测寄存器的更多信息。

8.2.2.1.1. 基于列和基于帧的校验位

图 165. 基于列和基于帧的校验位

128-Bits Data	128-Bits Data	32-Bits Frame-Based Check-Bits	Frame 0
128-Bits Data	128-Bits Data	32-Bits Frame-Based Check-Bits	Frame 1
⋮	⋮	32-Bits Frame-Based Check-Bits	Frame 2
⋮	⋮	⋮	
⋮	⋮	⋮	
128-Bits Data	32-Bits Frame-Based Check-Bits	Last Frame
32-Bits Column-Based Check-Bits	32-Bits Column-Based Check-Bits	
Column 0	Column 1				Last Column	

EDCRC 校验位更新

基于帧的校验位于配置期间由片上计算。基于列的校验位在配置后被更新。

EDCRC 功能被使能时，在器件进入用户模式后，EDCRC 功能开始读取 CRAM 帧。基于帧的校验位验证从读回帧收集的数据。

基于基于帧的初始检验完成后，将根据相应列 CRAM 计算基于列的校验位。EDCRC 硬核模块将在以下情况中重计算基于列的校验位：

- FPGA 重配置
- 部分重配置(PR)成功完成后
- 通过协议进行配置(CvP)完成后

8.2.2.1.2. 错误消息寄存器

EMR 包含有关错误类型，错误位置 and 实际特征群 (syndrome) 信息。Arria 10 器件中，该寄存器为 78 位宽。EMR 不识别不可纠正错误位的位置。各错误的位置信息中包含帧代码，双字位置以及各位所在帧和列中的位置。

通过下面途径可以移出寄存器内容：

- EMR Unloader IP 内核—内核接口
- SHIFT_EDERROR_REG JTAG 指示—JTAG 接口
- HPS 移位寄存器—HPS 接口

图 166. 错误信息寄存器映射

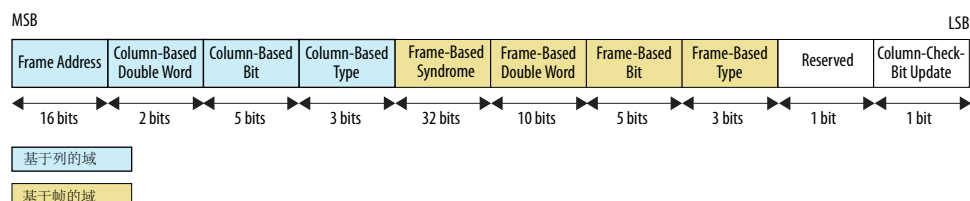


表 107. 错误消息寄存器宽度及说明

名称	宽度(Bit)	说明
Frame Address	16	错误位置的帧数
Column-Based Double Word	2	一列中的每帧有 4 个双字，表示错误的双字位置。
Column-Based Bits	5	32-bit 双字中的错误位置
Column-Based Type	3	表 108 (第 261 页)中出现的错误类型
Frame-Based syndrome register	32	包含计算出的当前帧的 32-bit CRC 特征码。如果 CRC 的值为 0，那么 CRC_ERROR 管脚被驱低以表明无错误。否则，此管脚被拉高。
Frame-Based Double Word	10	CRAM 帧中的双字位置。
Frame-Based Bit	5	32-bit 双字中的错误位置
Frame-Based Type	3	表 108 (第 261 页)中出现的错误类型
Reserved	1	保留位
Column-Based Check-Bits Update	1	逻辑高电平，如果在列校验位更新阶段出现错误。CRC_ERROR 管脚将被置位并保持在高电平直到重配置 FPGA。

检索错误信息

可使用 SHIFT_EDERROR_REG JTAG 指令通过内核接口或 JTAG 接口来检索 EMR 内容。Intel 提供可通过内核接口卸载 EMR 内容的 Error Message Register Unloader IP Core，并允许多个设计组件对其共享。

相关链接

[Altera 错误消息寄存器卸载器 IP 内核用户指南](#)

提供关于使用用户移位寄存器移出 EMR 的更多信息。

EMR 中的错误类型

表 108. EMR 中的错误类型

下表列出了 EMR 中错误类型域所报告的可能的错误类型。

错误类型	Bit 2	Bit 1	Bit 0	说明
基于帧	0	0	0	没有错误
	0	0	1	单比特错误
	0	1	X	双邻错误
	1	1	1	不可纠正错误
基于列	0	0	0	没有错误

继续...

错误类型	Bit 2	Bit 1	Bit 0	说明
	0	0	1	单比特错误
	0	1	X	同一帧中的双邻错误
	1	0	X	不同帧的双邻错误
	1	1	0	不同帧的双邻错误
	1	1	1	不可纠正错误

8.2.2.1.3. CRC_ERROR 管脚行为

Arria 10 快速 EDCRC 功能运行全部并列的基于列的校验位引擎。当检测到 SEU 时，基于列的校验位置位 CRC_ERROR，然后把已检测到的帧位置传递到基于帧的校验位从而进一步定位受影响的位置。此过程导致 CRC_ERROR 管脚两次置位。基于列的校验位置位第一个 CRC_ERROR 脉冲，随后由基于帧的校验位置位第二个脉冲。

在 Arria 10 中，一旦检测到 SEU，则 CRC_ERROR 将被置高且保持为高，直到 EMR 被准备读取。您可在 CRC_ERROR 管脚电平走低时尽快卸载 EMR 数据。一旦 EMR 数据被卸载，就可以决定错误类型及受影响的位置。通过这些信息可以决定您的系统如何应答具体的 SEU 事件。

图 167. 快速 EDCRC 处理流程图

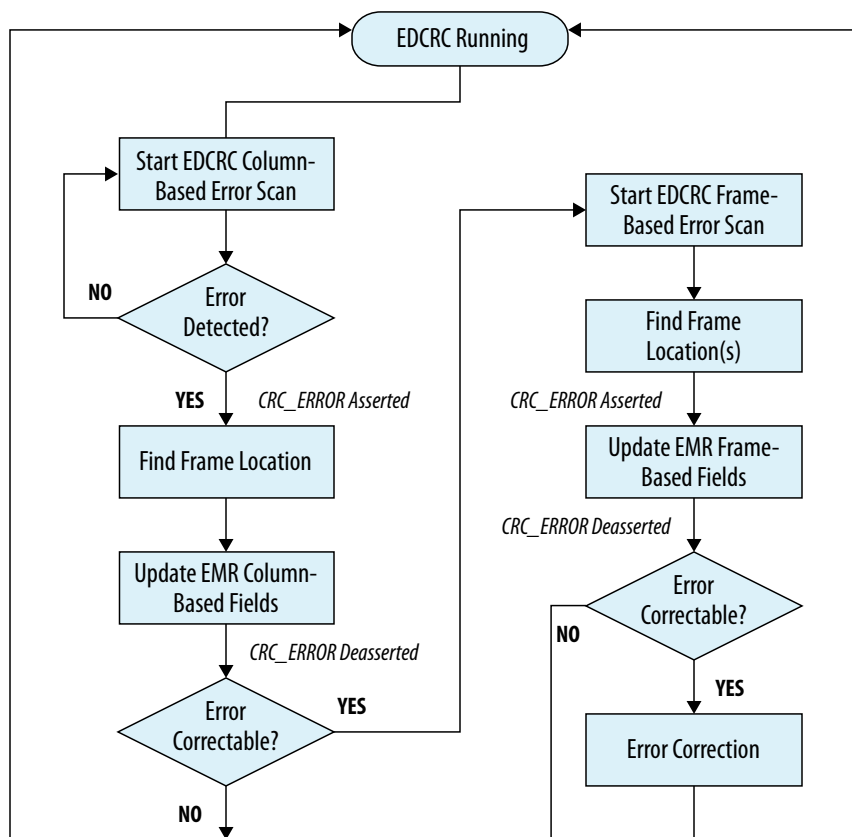
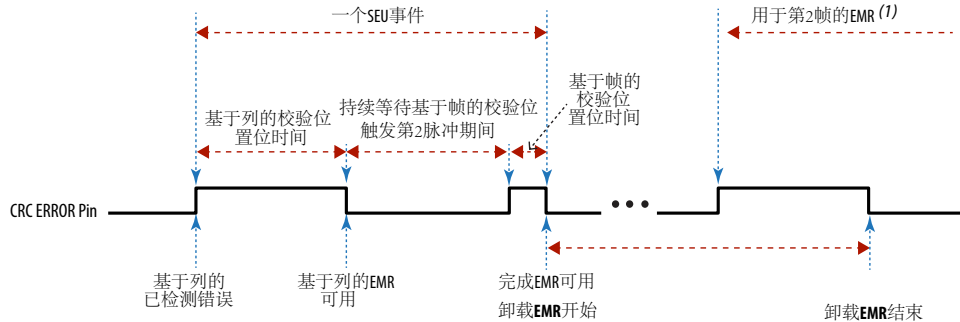


图 168. 基于列的校验位的时序图

如果是可纠正错误，大多数情况下，单个 SEU 时间中存在一个第二脉冲。同时存在不可纠正错误的情况下，CRC_ERROR 管脚置位两个脉冲，请参阅[可纠正和不可纠正错误](#)以获得关于可纠正和不可纠正错误的完整情况。完整的 EMR 将仅在第二脉冲的下降沿可用。

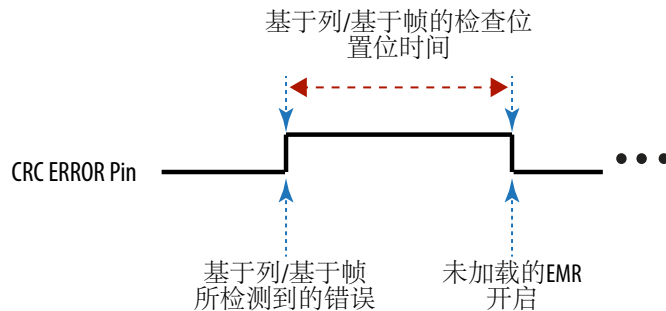


(1)在可校正的双邻错误位于不同帧的极少情况下。

在极少数不可纠正和未定位错误的情况下，CRC_ERROR 信号仅被置位一次。由于无法定位不可纠正错误的位置，将不会有基于帧的校验位的第二脉冲置位。对于典型环境条件下的器件，出现不可纠正的多位 SEU 可能性极低，一万年难有一个。

图 169. 基于列或基于帧的校验位时序图

一个 SEU 事件中由单个脉冲观测的基于列/基于帧的校验位的 CRC_ERROR 管脚行为实例。



相关链接

Arria 10 器件系列管脚连接指南

提供关于 CRC_ERROR 连接指导的更多信息。

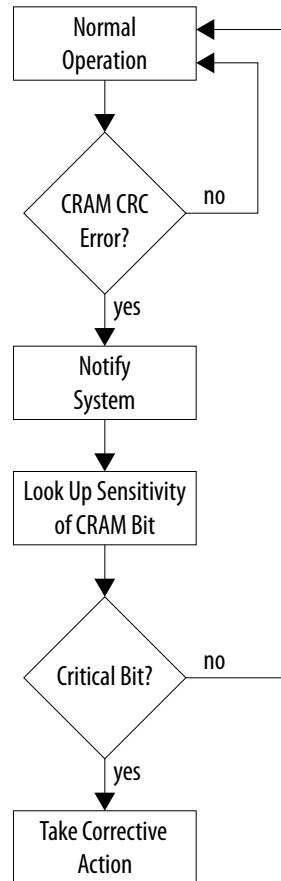
8.2.2.2. SEU 敏感度

重配置一个正在运行的 FPGA 会对使用 FPGA 的系统有显著的影响。当规划 SEU 恢复时，要考虑到将 FPGA 恢复到与当前的系统状态相一致的状态所需的时间。例如，如果内部状态机处于非法状态，那么它可能需要复位。此外，周围逻辑可能需要考虑这一意外操作。

通常情况下，SEU 会影响到那些没有被设计使用的 CRAM 比特位。很多配置比特位没有被使用，这是因为它们控制的逻辑和布线在设计中没有被使用。根据不同的实现，甚至在占用最高的器件中也只能使用全部 CRAM 的 40%。这意味着只有 40% 的 SEU 事件需要干预，而其余 60% 的 SEU 可以忽略。此时我们可以称被使用的比特位为关键比特位，而未被使用的为非关键比特位。

您可以决定已实现设计的部分在 FPGA 功能中未被使用。比如已实现但对设备操作不重要的测试电路，或者其他可以被记录但无需重新编程或复位的非关键功能。

图 170. 敏感度处理流程



8.2.2.3. 层次标记

层次标记(hierarchy tagging)是对设计部分的敏感度进行分类的过程。

通过使用 Quartus Prime 软件创建设计分区，然后将 Advanced SEU Detection (ASD) Region 参数分配给此分区来执行层次标记。此参数的值可设为 0 到 15，因此对设计部分的系统响应的分类总共有 16 种。

设计层次的敏感度处理取决于 Sensitivity Map Header (.smh)文件的内容。该文件决定 FPGA 逻辑设计的哪一部分对 CRAM 比特位翻转敏感。可使用 .smh 文件中的敏感度信息决定正确(最少损坏)的恢复顺序。

要生成功能有效的 .smh，必须从功能性逻辑的角度，通过层次标记过程来指定设计敏感度。

相关链接

[Altera 高级 SEU 检测 IP 用户指南](#)

提供使用 Altera 高级 SEU 检测 IP 进行层次标记的更多信息。



8.2.2.4. 评估您的系统对功能翻转的响应

SEU 冲击和功能中断的比率就是单粒子功能性中断 (Single Event Functional Interrupt, SEFI) 比率。最小化该比率以提高 SEU 缓解。由于 SEU 可随机冲击任意存储器单元, 为确保全面的恢复响应, 系统测试至关重要。

使用故障注入有助于 SEU 恢复响应。故障注入功能允许在系统中操作 FPGA 并注入随机 CRAM 位翻转以测试 FPGA 和系统从 SEU 检测和完全恢复的能力。且应该能够观察到 FPGA 和系统从这些仿真 SEU 冲击中恢复。然后, 通过对这些冲击的观察, 进而改进 FPGA 和系统的恢复顺序。请使用故障注入功能确定 SEFI 率。

相关链接

- [Altera 故障注入 IP Core 用户指南](#)
提供使用 Altera 故障注入 IP Core 将软件错误注入仿真 SEU 的更多信息。
- [使用故障注入调试器调试单粒子翻转 \(SEU\)](#)
提供关于使用故障注入调试器的更多信息。

8.2.2.5. 从 CRC 错误中恢复

Arria 10 器件支持内部擦拭能力。内部擦拭功能可以在检测到翻转时纠正可纠正的 CRAM 自动翻转。然而, 内部擦拭不能把 FPGA 修复到已知的良好状态。发现错误与完成擦拭可能需要数十毫秒的时间。该持续时间表示数千的时钟周期, 在此期间数据被合法的写入存储器或状态寄存器。这是一个关于始终遵循支持软复位的 SEU 事件并将 FPGA 操作带入已知良好状态的不错练习。

如果软复位无法把 FPGA 带入已知的良好状态, 那么您可以重配置器件来重写 CRAM 并再次初始化设计寄存器。承载 Arria 10 器件的系统必须控制器件的重配置。当重配置成功完成时, Arria 10 器件按计划运行。

相关链接

[Arria 10 器件的配置, 设计安全, 和远程系统更新](#)
提供关于配置序列的更多信息。

8.2.2.5.1. 使能错误纠正 (内部擦拭)

Arria 10 支持内部擦拭功能并自动擦除由 SEU 引起的翻转位。请按以下步骤使能内部擦拭功能:

1. 在 **Assignments** 菜单上, 点击 **Device**。
2. 点击 **Device and Pin Options** 并选择 **Error Detection CRC** 标签。
3. 开启 **Enable internal scrubbing**。
4. 点击 **OK**。

8.3. 规范

该部分列出了用户模式中错误检测的错误检测频率和 CRC 计算时间。

8.3.1. 错误检测频率

当您无法在指定的 EMR 更新间隔期间卸载 EMR 时, 可以降低错误检测频率。您可以在 Quartus Prime 软件中设置时钟频率的分频因子从而控制错误检测过程的速率。

注意: 降低错误检测频率后并没有显著的电源受益。

每个数据帧的错误检测进程的速度由以下公式决定：

图 171. 错误检测频率公式

$$\text{Error Detection Frequency} = \frac{\text{Internal Oscillator Frequency}}{N}$$

表 109. Arria 10 器件的错误检测频率范围

下表列出了每个速度等级的 F_{MIN} 和 F_{MAX} 。

注意：当 $N = 1$ 时，频率如下所示。当 $N = 2$ 或 4 时，对应地除以以下所显示的频率。

速度等级	错误检测频率	
	f_{MIN}	f_{MAX}
1	49	77
2	45	77
3	42	77

8.3.2. 错误检测时间

检测 SEU 错误所用的时间相对于实际的 SEU 事件，由所用器件和错误检测时钟的频率决定。

表 110. Arria 10 器件中的 CRC 计算时间

$$\text{Error detection time}_{\text{Maximum}} = \text{Error detection time} \times \left(\frac{\text{Error Detection Frequency } f_{\text{MAX}}}{\text{Error Detection Frequency } f_{\text{MIN}}} \right) \times N$$

$$\text{Error detection time}_{\text{Minimum}} = \text{Error detection time} \times N$$

- 速度等级 1: $N=1, 2$ 或 4 。
- 速度等级 2 和 3: $N=2$ 或 4 。

系列	密度	错误检测时间(ms)
GX/SX	160 / 220	14.29
	270 / 320	14.29
	480	21.13
	570/ 660	27.84
GX/ GT	900 / 1150	27.84

8.3.3. EMR 更新间隔

您必须在最短的 EMR 更新间隔内卸载 EMR 数据，以避免当前的 EMR 数据被下一个错误信息覆盖。然而，Altera EMR Unloader IP Core 能够通过确保在卸载期间无 EMR 数据丢失来处理这一情况。IP 内核将通过标记 `emr_error` 信号来检测丢失的 EMR 信息。

错误信息寄存器的每次更新的时间间隔取决于器件和错误检测时钟的频率。



表 111. Arria 10 器件中的估计 EMR 更新间隔

$$\text{EMR update interval}_{\text{Maximum}} = \text{EMR update interval} \times \left(\frac{\text{Error Detection Frequency } f_{\text{MAX}}}{\text{Error Detection Frequency } f_{\text{MIN}}} \right) \times N$$

$$\text{EMR update interval}_{\text{Minimum}} = \text{EMR update interval} \times N$$

- 速度等级 1: N=1, 2 或 4。
- 速度等级 2 和 3: N=2 或 4。

系列	密度	EMR 更新间隔(ms)
GX/SX	160 / 220	0.28
	270 / 320	0.28
	480	0.41
	570 / 660	0.54
GX/ GT	900 / 1150	0.55

8.3.4. 纠错时间

Arria 10 提供最快的纠错功能，下表列出了每种器件系列的纠错时间。

表 112. 纠错时间

$$\text{Correction time}_{\text{Maximum}} = \text{Correction time} \times \left(\frac{\text{Error Detection Frequency } f_{\text{MAX}}}{\text{Error Detection Frequency } f_{\text{MIN}}} \right) \times N$$

$$\text{Correction time}_{\text{Minimum}} = \text{Correction time} \times N$$

- 注意:
- 速度等级 1: N=1, 2 或 4。
 - 速度等级 2 和 3: N=2 或 4。

系列	密度	纠错时间 (µs)
GX/SX	160/220	19.73
	270/320	27.62
	480	27.21
	570/660	27.21
GX/GT	900/1150	39.68

8.4. 文档修订历史

日期	版本	修订内容
2017 年 3 月	2017.03.15	更名为 Intel。
2016 年 5 月	2016.05.02	<ul style="list-style-type: none"> 编辑了 错误检测循环冗余校验。 更新了 CRC 校验位实例到基于列的校验位和基于帧的校验位。 更新了 错误消息寄存器映射图示。 添加了 快速 EDCRC 处理流程图 图示。 添加了注释以注明减少错误检测频率后没有显著的电源受益。
2015 年 12 月	2015.12.14	<ul style="list-style-type: none"> 更新了章节结构。 增添了 错误纠正时间规范。 对嵌入式存储器，存储器模块纠错代码支持，SEU 敏感度，层次标记和评估对 SEU 的系统响应增添了扼要说明和外部相关信息链接。 更新了错误检测频率中的分频值。 更新了显示 fMAX 和 fMIN 的错误检测频率范围表。 更新了评估 EMR 更新间隔和 CRC 计算时间表。 增添了 EMR 更新间隔和 CRC 计算时间的公式。
2015 年 11 月	2015.11.02	将 Quartus II 更改成 Quartus Prime。
2015 年 6 月	2015.06.15	为 Altera EMR Unloader IP Core User Guide, Altera Fault Injection IP Core User Guide 和 Altera Advance SEU Detection IP Core User Guide 增添了链接。
2015 年 5 月	2015.05.04	<ul style="list-style-type: none"> 为 Altera EMR Unloader IP Core User Guide, Altera Fault Injection IP Core User Guide and Altera Advance SEU Detection IP Core User Guide 增添了链接。。 更新了 CRC_ERROR 管脚行为，包括基于列和基于帧的 CRC 错误检测和仅基于帧的 CRC 错误检测。 更新了 Bit 0 的 ‘Error Type in EMR’ 中基于列的类型。
继续...		



日期	版本	修订内容
		<ul style="list-style-type: none"> 编辑修改。 更新了错误检测频率的分频值和范围。 更新了 CRC 计算时间，包括速度等级并相应地重新排列。 更新了 EMR 更新间隔。 更新了 Error Message Register Map 和 Error Detection in User Mode 结构图中的寄存器。
2015 年 1 月	2015.01.23	<ul style="list-style-type: none"> 添加了 EMR 时序间隔。 添加了 CRC 计算时间。 添加了时序约束。
2014 年 8 月	2014.0818	<ul style="list-style-type: none"> 更新了"XAUI Supported Features"部分。 更新了配置错误检测部分，修改了 CRC 值。 更新了用户模式错误检测部分，对错误检测 CRC 添加了校验位计算。 更新了 CRC_ERROR 管脚行为部分。 更新了检索错误信息部分。 更新了 CRC_ERROR 管脚部分，其中更新了管脚说明。 更新了表 8-4，其中更新了基于帧的综合寄存器、用户更新寄存器和用户移位寄存器的说明。 更新表 8-5，其中更新了基于帧和基于列的类型的错误类型名称。
2013 年 12 月	2013.1202	首次发布。

9. Arria 10 器件中的 JTAG 边界扫描测试

本章节介绍了 Arria 10 器件中的边界扫描测试(BST)功能。

相关链接

[Arria 10 器件手册：已知问题](#)

列出了对 *Arria 10 器件手册* 章节规划的更新。

9.1. BST 操作控制

Arria 10 GX、Arria 10 GT 和 Arria 10 SX 器件支持 IEEE Std. 1149.1 BST 和 IEEE Std. 1149.6 BST。您可以在配置之前，配置期间，配置之后对 Arria 10 器件执行 BST。

9.1.1. IDCODE

每个 Arria 10 器件对应唯一的 IDCODE。使用此代码识别 JTAG 链中的器件。

表 113. Arria 10 器件的 IDCODE 信息

器件类型	产品系列	IDCODE (32 位)			
		版本(4 位)	部件编号(16 位)	制造商识别编号(11 位)	LSB (1 位)
Arria 10GX	GX 160	0000	0010 1110 1110 0010	000 0110 1110	1
	GX 220	0000	0010 1110 0010 0010	000 0110 1110	1
	GX 270	0000	0010 1110 1110 0011	000 0110 1110	1
	GX 320	0000	0010 1110 0010 0011	000 0110 1110	1
	GX 480	0000	0010 1110 0010 0100	000 0110 1110	1
	GX 570	0000	0010 1110 1110 0101	000 0110 1110	1
	GX 660	0000	0010 1110 0010 0101	000 0110 1110	1
	GX 900	0000	0010 1110 1110 0110	000 0110 1110	1
	GX 1150	0000	0010 1110 0110 0110	000 0110 1110	1
Arria 10GT	GT 900	0000	0010 1110 0010 0110	000 0110 1110	1
	GT 1150	0000	0010 1110 0000 0110	000 0110 1110	1
Arria 10SX	SX 160	0000	0010 1110 0110 0010	000 0110 1110	1
	SX 220	0000	0010 1110 0000 0010	000 0110 1110	1
	SX 270	0000	0010 1110 0110 0011	000 0110 1110	1
	SX 320	0000	0010 1110 0000 0011	000 0110 1110	1

继续...

Intel Corporation. All rights reserved. Agilix, Altera, Arria, Cyclone, Enpirion, Intel, the Intel logo, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered



器件类型	产品系列	IDCODE (32 位)			
		版本(4 位)	部件编号(16 位)	制造商识别编号(11 位)	LSB (1 位)
	SX 480	0000	0010 1110 0000 0100	000 0110 1110	1
	SX 570	0000	0010 1110 0110 0101	000 0110 1110	1
	SX 660	0000	0010 1110 0000 0101	000 0110 1110	1

9.1.2. 所支持的 JTAG 指令

表 114. Arria 10 器件支持的 JTAG 指令

JTAG 指令	指令编码	说明
SAMPLE ⁽³⁹⁾ / PRELOAD	00 0000 0101	<ul style="list-style-type: none"> 器件正常操作期间, 使您能够在器件管脚采集和检查信号的抓拍并且允许初始数据码型为一个输出。 加载 EXTEST 指令之前, 使用该指令预加载测试码型到更新寄存器。
EXTEST	00 0000 1111	<ul style="list-style-type: none"> 通过强制输出管脚的测试码型使您能够测试外部电路和板级互联, 并且在输入管脚采集测试结果。在输出管脚上强制已知逻辑高电平或低电平使您能够检测到扫描链中所有器件的管脚的开路和短路。 EXTEST 的高阻抗状态被总线保持和弱上拉电阻功能覆盖。
BYPASS	11 1111 1111	<ul style="list-style-type: none"> 在 TDI 和 TDO 管脚之间放置 1bit 旁路寄存器。在器件正常操作期间, 1bit 旁路寄存器使得 BST 数据从所选器件同步地传递到相邻器件。 从旁路寄存器输出数据中您会读取 '0'。
USERCODE	00 0000 0111	选择 32bit USERCODE 寄存器并且将其放置在 TDI 和 TDO 管脚之间以便支持 USERCODE 从 TDO 的串行移位。
IDCODE	00 0000 0110	<ul style="list-style-type: none"> 识别 JTAG 链中的器件。如果您选择 IDCODE, 那么器件识别寄存器 会被加载 32bit 供应商定义的识别代码。 选择 IDCODE 寄存器并且将其放置在 TDI 和 TDO 管脚之间以便支持 IDCODE 从 TDO 的串行移位。 IDCODE 在上电时是默认指令并且处于 TAP RESET 状态。没有加载任何指令的情况下, 可以进入 SHIFT_DR 状态并且移出 JTAG 器件 ID。
继续...		

(39) 高速串行接口(HSSI)管脚不支持 SAMPLE JTAG 指令。

JTAG 指令	指令编码	说明
HIGHZ	00 0000 1011	<ul style="list-style-type: none"> 将所有的用户 I/O 管脚设置为无效驱动状态。 将 1bit 旁路寄存器放置在 TDI 和 TDO 管脚之间。正常操作期间, 1bit 旁路寄存器支持 BST 数据从所选的器件同步地传递到相邻的器件同时三态所有的 I/O 管脚直到新的 JTAG 指令被执行。 如果测试配置后的器件, 那么可编程的弱上拉电阻或总线保持功能会覆盖管脚的 HIGHZ 值。
CLAMP	00 0000 1010	<ul style="list-style-type: none"> 将 1bit 旁路寄存器放置在 TDI 和 TDO 管脚之间。正常操作期间, 1bit 旁路寄存器支持 BST 数据从所选的器件同步地传递到相邻的器件, 同时保持 I/O 管脚为边界扫描寄存器中数据定义的状态。 如果测试配置后的器件, 那么可编程的弱上拉电阻或总线保持功能会覆盖管脚的 CLAMP 值。CLAMP 值存储在边界扫描单元(BSC)的更新寄存器中。
PULSE_NCONFIG	00 0000 0001	仿真脉冲 nCONFIG 管脚为低电平以触发重配置(即使物理管脚没有受到影响)。
EXTEST_PULSE	00 1000 1111	<p>使能发送器和接收器之间的板级连接检查, 这些发送器和接收器是通过生成 3 个输出跳变的 AC 耦合:</p> <ul style="list-style-type: none"> 在 UPDATE_IR/DR 状态中, 驱动器在 TCK 的下降沿上驱动数据。 进入 RUN_TEST/IDLE 状态后, 驱动器在 TCK 的下降沿上驱动反向数据。 离开 RUN_TEST/IDLE 状态后, 驱动器在 TCK 的下降沿上驱动数据。
EXTEST_TRAIN	00 0100 1111	只要 TAP 控制器处于 RUN_TEST/IDLE 状态, 作用和 EXTEST_PULSE 指令的作用相同, (除了输出继续在 TCK 下降沿上翻转的情况以外)。
SHIFT_EDERROR_REG	00 0001 0111	JTAG 指令将 EMR 连接到 TDI 与 TDO 管脚之间的错误检测模块中的 JTAGpin。

注意: 如果器件处于复位状态并且 nCONFIG 或 nSTATUS 信号为低电平, 那么器件 IDCODE 可能不会被正确地读取。为了正确地读取器件 IDCODE, 必须只有在 nCONFIG 和 nSTATUS 信号处于高电平时, 发出 IDCODEJTAG 指令。

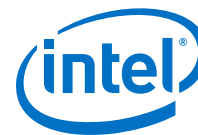
9.1.3. JTAG 安全模式

在 JTAG 安全模式中, JTAG 管脚仅支持 BYPASS, SAMPLE/PRELOAD, EXTEST, 及 IDCODEJTAG 指令。

相关链接

[JTAG Secure Mode in AN 556](#)

Provides more information about JTAG Secure Mode



9.1.4. JTAG 专用指令

警告: 一定不要调用以下指令代码。这些指令可以毁坏并且导致器件不可用:

- 1100010000
- 1100010011
- 0111100000
- 0101011110
- 0000101010
- 0011100000
- 0000101010
- 0101000001
- 1110000001
- 0001010101
- 1010100001

9.2. JTAG 操作的 I/O 电压

在 IEEE Std. 1149.1 和 IEEE Std. 1149.6 模式中操作的 Arria 10 器件使用 4 个所需的 JTAG 管脚—TDI、TDO、TMS、TCK 和一个可选管脚 TRST。

TCK 管脚具有一个内部弱下拉电阻，而 TDI、TMS 和 TRST 管脚具有内部弱上拉电阻。1.8-、1.5-或 1.2-V V_{CCPGM} 电源对 TDI、TDO、TMS、TCK 和 TRST 管脚供电。所有用户 I/O 管脚在 JTAG 配置期间都处于三态。

JTAG 管脚支持 1.8 V、1.5V 和 1.2V TTL/CMOS I/O 标准。对于任何高于 1.8 V 的电压，您必须使用电平位移器。用于 JTAG 管脚的电平位移器的输出电压必须与对 V_{CCPGM} 电源设置的电压相同。

注意: 不要使用高于 1.8-、1.5-和 1.2-V V_{CCPGM} 电源的电压驱动 TDI、TMS、TCK 和 TRST 管脚的信号。TDI、TMS、TCK 和 TRST 输入管脚的电源电压必须和对 V_{CCPGM} 电源设置的电压相同。

表 115. TDO 输出缓冲器

TDO 输出缓冲器	电压 (V)		
	1.8	1.5	1.2
V_{CCPGM}			
$V_{OH} (MIN)$	1.7	1.4	1.1

9.3. 执行 BST

您可以在配置前，配置后或配置期间发出 BYPASS、IDCODE 和 SAMPLE JTAG 指令，而无需中断配置。

要发出其它的 JTAG 指令,请遵循以下这些指南:

- 要在配置之前执行测试,需保持 nCONFIG 管脚为低电平。
- 要在配置期间执行 BST,需发出 CONFIG_IO JTAG 指令来中断配置。当中断配置时,您可以发出其它的 JTAG 指令来执行 BST。BST 完成后,发出 PULSE_NCONFIG JTAG 指令或脉冲 nCONFIG 为低电平以重配置器件。

Arria 10 器件上的芯片全复位 (DEV_CLRn) 和芯片全输出使能 (DEV_OE) 管脚不影响 JTAG 边界扫描或配置操作。 翻转这些管脚不会扰乱 BST 操作(除了所预期的 BST 行为以外)。

如果对 Arria 10 器件的 JTAG 配置设计一个电路板,那么需要考虑专用配置管脚的连接。

注意: 对于 SoC 器件, FPGA 模块中的 JTAG 连接和 HPS 模块中的 JTAG 连接被链接到 Arria 10 器件中。FPGA 中的 JTAG 连接的优先权高于 HPS 模块中的 JTAG 连接。

注意: 如果您在配置之前或期间执行 HIGHZ JTAG 指令,那么您需要拉高 nIO_PULLUP 管脚来禁用 I/O 单元中的内部弱上拉电阻。如果您在用户模式中执行该 JTAG 指令,那么您可以拉高或拉低 nIO_PULLUP 管脚。

注意: 如果您在用户模式中执行 BST,那么当这些管脚不被用作用户 I/O 管脚时,您不能够对 PR_ENABLE、CRC_ERROR 和 CVP_CONFDONE 管脚采集正确的值。

注意: 上电后,仅当 nCONFIG 以及 nSTATUS 都变高时,您才可以执行 JTAG BST。

相关链接

- [Arria 10 GX、GT 和 SX 器件系列管脚连接指南](#)
提供关于管脚连接的更多信息。
- [JTAG 配置](#)
提供关于 JTAG 配置时序的更多信息。
- [JTAG Configuration](#)
提供关于 JTAG 配置时序的更多信息。
- [JTAG 配置](#) (第 228 页)

9.4. 使能和禁用 IEEE Std. 1149.1 BST 电路

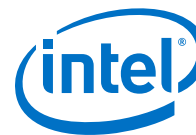
IEEE Std. 1149.1 BST 电路在 Arria 10 器件上电后被使能。然而对于 Arria 10 SoC FPGA,则必须上电 HPS 和 FPGA 来执行 BST。

为确保在不需要 IEEE Std. 1149.1 电路的情况下不会意外地使能此电路,需要通过下表列出的管脚连接来永久性地禁用电路。

表 117. Arria 10 器件的永久地禁用 IEEE Std. 1149.1 电路的管脚连接

JTAG 管脚 ⁽⁴¹⁾	禁用连接
TMS	V _{CCPGM}
TCK	GND
继续...	

⁽⁴¹⁾ JTAG 管脚是专用的。在 Arria 10 器件中,软件选项不可用于禁用 JTAG。



JTAG 管脚 ⁽⁴¹⁾	禁用连接
TDI	V _{CCPGM}
TDO	不连接
TRST	GND

9.5. IEEE Std. 1149.1 边界扫描测试指南

当使用 IEEE Std. 1149.1 器件执行 BST 时，请考虑以下指南：

- 如果在 SHIFT_IR 状态的第一个时钟周期期间，“10...” 码型没有通过 TDO 管脚移出指令寄存器，那么 TAP 控制器不会达到合适状态。要解决这个问题，需要尝试以下其中一个方法：
 - 验证 TAP 控制器已经正确地达到 SHIFT_IR 状态。要促进 TAP 控制器达到 SHIFT_IR 状态，需要返回到 RESET 状态并且发送 01100 代码到 TMS 管脚。
 - 检查 VCC、GND 和 JTAG 的连接并且检查器件上的专用配置管脚。
- 在第一个 EXTEST 测试周期之前，执行一个 SAMPLE/PRELOAD 测试周期，以确保进入 EXTEST 模式时，已知数据出现在器件管脚。如果 OEJ 更新寄存器包含 0，那么 OUTJ 更新寄存器中的数据被驱动。状态必须是已知和正确的以便避免在系统中与其它器件的冲突。
- 不要在电路中的重配置期间执行 EXTEST 测试，因为在电路中的重配置期间不支持 EXTEST。要执行测试，就要等待配置完成或发出 CONFIG_IO 指令来中断配置。
- 配置完成后，不可以测试差分管脚对中的任何管脚。要在配置后执行 BST，那么就要编辑和重新定义与这些差分管脚对相一致的 BSC 组为一个内部单元。

相关链接

IEEE 1149.1 BSDL Files

提供了关于 BSC 组定义的更多信息。

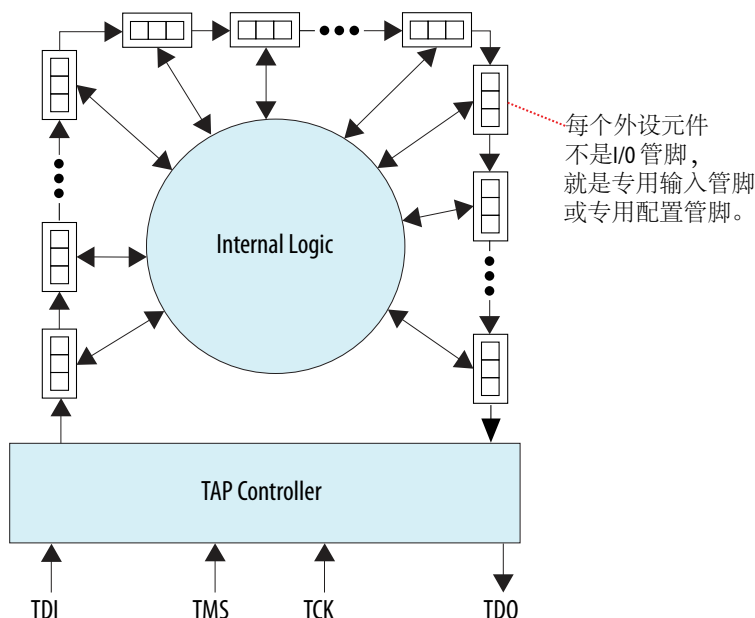
9.6. IEEE Std. 1149.1 边界扫描寄存器

边界扫描寄存器是一个大型串行移位寄存器，它使用 TDI 管脚作为输入以及 TDO 管脚作为输出。边界扫描寄存器包含与 Arria 10 I/O 管脚相关的 3 位外围单元。可以使用边界扫描寄存器测试外部管脚连接或采集内部数据。

⁽⁴¹⁾ JTAG 管脚是专用的。在 Arria 10 器件中，软件选项不可用于禁用 JTAG。

图 172. 边界扫描寄存器

该图显示了测试数据如何在 IEEE Std. 1149.1 器件的外围串行地移位。



9.6.1. Arria 10 器件 I/O 管脚的边界扫描单元

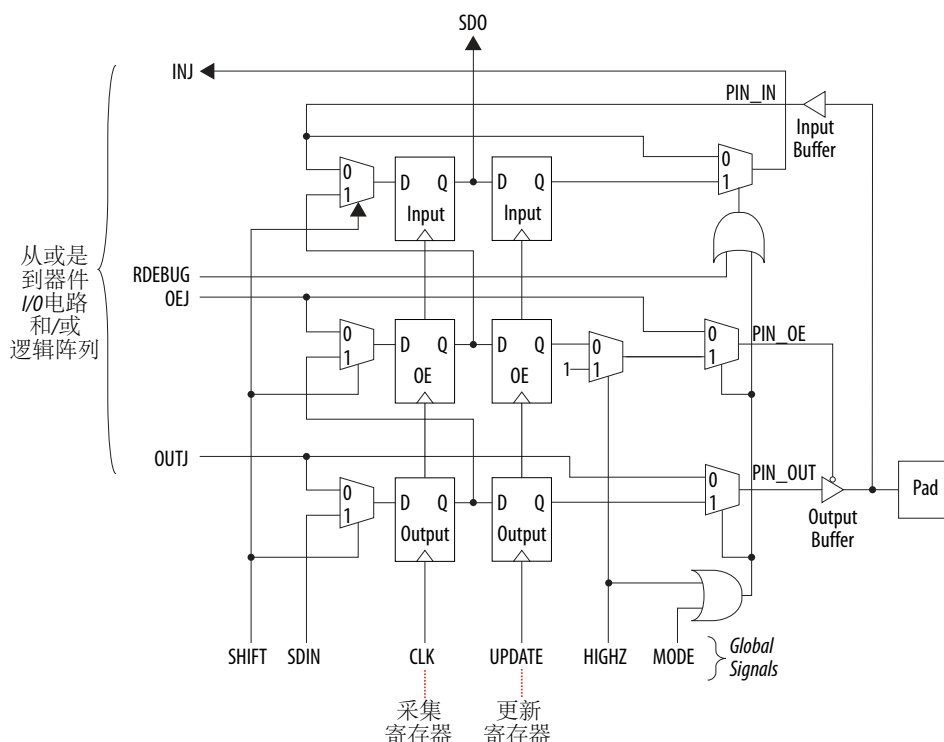
Arria 10 器件 3-bit BSC 包含下面寄存器：

- 采集寄存器—通过 OUTJ、OEJ 和 PIN_IN 信号连接到内部器件数据。
- 更新寄存器—通过 PIN_OUT 和 PIN_OE 信号连接到外部数据。

TAP 控制器从内部对 IEEE Std. 1149.1 寄存器((移位、时钟和更新)生成全局控制信号。指令寄存器的解码生成 MODE 信号。

边界扫描寄存器的数据信号路径从 serial data in (SDI)信号到 serial data out (SDO)信号运行。扫描寄存器开始于器件的 TDI 管脚并且结束于器件的 TDO 管脚。

图 173. Arria 10 器件中包含 IEEE Std. 1149.1 BST 电路的用户 I/O BSC



注意: TDI、TDO、TMS、TCK、TRST、VCC、GND、VREF、VSIGP、VSIGN、TEMPDIODE 和 RREF 管脚没有 BSC。

表 118. Arria 10 器件的边界扫描单元描述

下表列出了 Arria 10 器件中的所有 BSC 的采集寄存器和更新寄存器性能。

管脚类型	采集			驱动			注释
	输出采集 寄存器	OE 采集 寄存器	输入 采集 寄存器	输出 更新 寄存器	OE 更新 寄存器	输入 更新 寄存器	
用户 I/O 管脚	OUTJ	OEJ	PIN_IN	PIN_OUT	PIN_OE	INJ	—
专用时钟输入	无连接(N.C.)	N.C.	PIN_IN	N.C.	N.C.	N.C.	PIN_IN 驱动到时钟网络或逻辑阵列
专用输入 ⁽⁴²⁾ ⁽⁴³⁾	N.C.	N.C.	PIN_IN	N.C.	N.C.	N.C.	PIN_IN 驱动到控制逻辑
继续...							

(42) 这包括 nCONFIG、MSEL0、MSEL1、MSEL2、MSEL3、MSEL4 和 nCE 管脚。

(43) 包括 PLL_ENA、VCCSEL、PORSEL、nIO_PULLUP、nCONFIG、MSEL0、MSEL1、MSEL2、MSEL3、MSEL4 和 nCE 管脚。

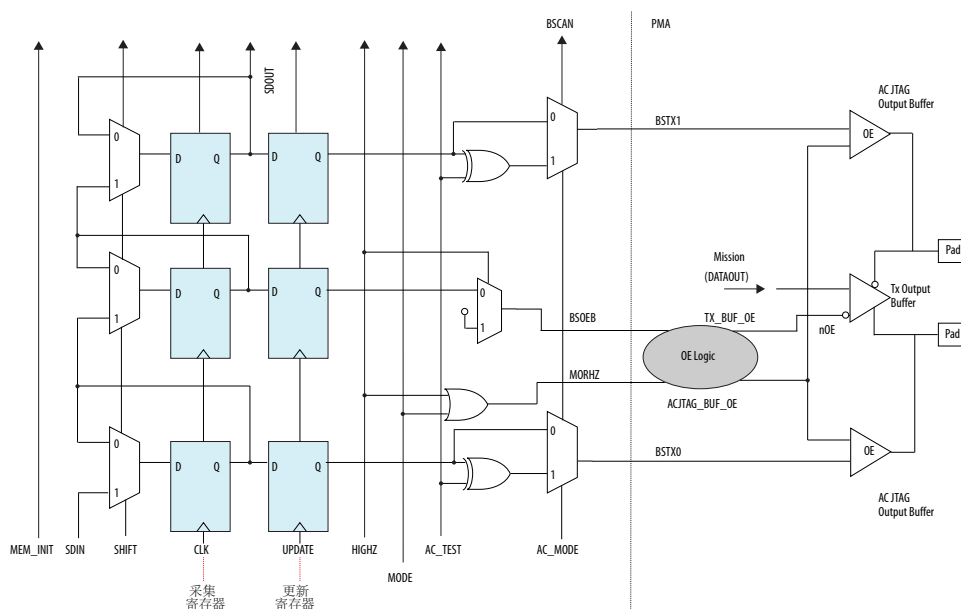
管脚类型	采集			驱动			注释
	输出采集 寄存器	OE 采集 寄存器	输入 采集 寄存器	输出 更新 寄存器	OE 更新 寄存器	输入 更新 寄存器	
专用双向(开漏) ⁽⁴⁴⁾	0	OEJ	PIN_IN	N.C.	N.C.	N.C.	PIN_IN 驱动到配置控制
专用双向 ⁽⁴⁵⁾	OUTJ	OEJ	PIN_IN	N.C.	N.C.	N.C.	PIN_IN 驱动到配置控制并且 OUTJ 驱动到输出缓冲器
专用输出 ⁽⁴⁶⁾	OUTJ	0	0	N.C.	N.C.	N.C.	OUTJ 驱动到输出缓冲器

9.6.2. IEEE Std. 1149.6 边界扫描寄存器

Arria 10 GZ 器件中的 HSSI 发送器的 BSC (GXB_TX[p,n])和接收器/输入时钟缓冲器 (GXB_RX[p,n])/(REFCLK[p,n])与 I/O 管脚的 BSC 不同。

注意: 您必须将 EXTEST_PULSE JTAG 指令用于 HSSI 收发器的 AC 耦合。不要将 EXTEST JTAG 指令用于 HSSI 收发器的 AC 耦合。您可以在配置之前，之后和期间对 Arria 10 器件执行 AC JTAG。

图 174. Arria 10 器件中具有 IEEE Std. 1149.6BST 电路的 HSSI 发送器 BSC

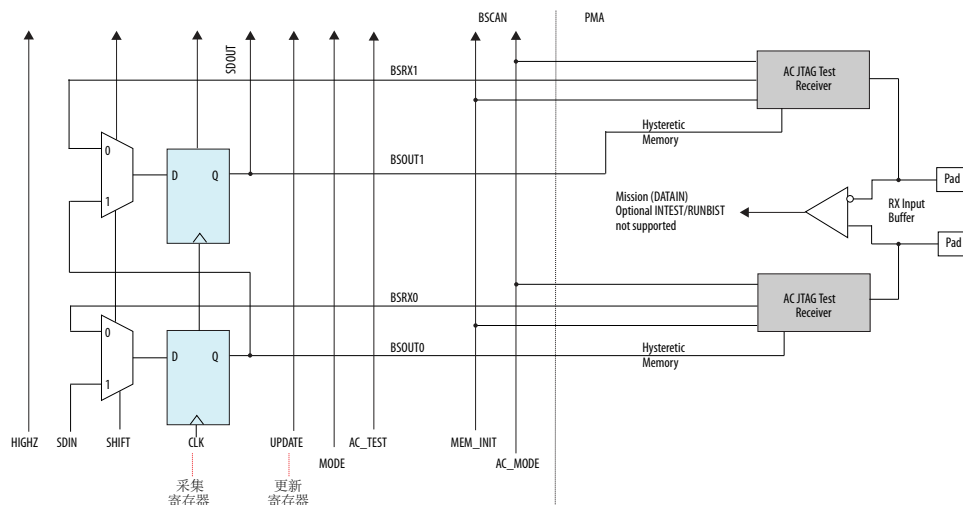


(44) 包括 CONF_DONE 和 nSTATUS 管脚。

(45) 包括 DCLK 管脚。

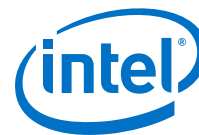
(46) 包括 nCEO 管脚。

图 175. Arria 10 器件的具有 IEEE Std. 1149.6BST 电路的 HSSI 接收器/输入时钟缓冲器



9.7. 文档修订历史

日期	版本	修订内容
2017 年 3 月	2017.03.15	更名为 Intel。
2016 年 5 月	2016.05.02	<ul style="list-style-type: none"> 更新了 IDCODE。 添加了关于 HSSI 管脚不支持 SAMPLE 指令的注释。
12 月	2015.12.14	<ul style="list-style-type: none"> 更新了 Arria 10 器件中具有 IEEE Std. 1149.1 BST 电路的用户 I/O BSC 图。 在“支持的 JTAG 指令”表中增添了 SHIFT_EDERROR_REG。
2015 年 11 月	2015.11.02	添加了注释：nSTATUS 和 nCONFIG 为高电平后才能执行 JTAG BST。
2014 年 8 月	2014.08.18	<ul style="list-style-type: none"> 更新了 JTAG 专用指令部分，添加了一个新的指令代码。 更新了 JTAG 操作部分的 I/O 电压，其中更新了 TDO 输出缓冲器的详细信息。 更新了执行 BST 部分，添加了一个关于在用户模式中执行 BST 的注释。 更新了 Arria 10 器件 I/O 管脚部分的边界扫描单元
2013 年 12 月	2013.1202	首次发布。



10. Arria 10 器件中的电源管理

本章介绍功耗，功耗降低技术，功率检测线特性，片上电压传感器，内部和外部温度感测二极管（TSD），上电复位（POR）要求，上电和断电排序要求，电源设计。

相关链接

- [Arria 10 器件手册 : 已知问题](#)
列出了 *Arria 10 器件手册* 章节的计划的更新。
- [Quartus Prime 手册卷 3 中的 PowerPlay 功耗分析章节](#)
提供了关于 Quartus Prime PowerPlay Power Analyzer 工具的详细信息。
- [建议的操作条件](#)
提供关于每个电源的所建议的操作条件的更多信息。
- [Arria 10 GX、GT 和 SX 器件系列管脚连接指南](#)
提供关于电源管脚连接指南和电源稳压器共享的详细信息。
- [电路板设计资源中心](#)
提供关于电源设计要求的详细信息。
- [PowerPlay 早期功耗评估器\(EPE\)和功耗分析器](#)
提供关于每个电源轨的电源和电流要求的更多信息。
- [Intel 电源管理 PowerSoC 解决方案](#)
提供更多关于针对上电 FPGA 而设计的 Intel 电源管理 IC 和 PowerSoC 解决方案的信息。

10.1. 功耗

an Arria 10 器件的总功耗包含以下组成部分：

- 静态功耗—上电时配置器件所消耗的功率，且无运行的用户时钟。
- 动态功耗—信号活动或翻转而导致的器件额外功耗。

10.1.1. 动态功耗公式

图 176. 动态功耗

下列公式表示如何计算动态功耗，其中 P 代表功耗，C 为负载电容，而 V 是供电电压电平。频率是指时钟频率且每时钟周期数据翻转一次。

$$P = \frac{1}{2} CV^2 \times frequency$$

从公式可知功耗取决于设计并由设计的操作频率决定。通过使用高级工艺流程优化，Arria 10 器件可最小化静态和动态功耗。这些优化支持 Arria 10 设计通过尽可能低的功耗满足指定的性能要求。



10.2. 功耗降低技术

Arria 10 器件采用先进的 20nm 工艺技术，加强内核体系结构，以及总功耗降低优化。以下为 Arria 10 Power Play Early Power Estimator (EPE) 提供并支持的可选择功耗降低技术，可在 Arria 10 设计中通过使能各个选项来评估功耗降低的效果。

- SmartVID
- 可编程电源技术
- 低静态功耗器件等级

10.2.1. SmartVID

SmartVID 功能支持电源稳压器在为 Arria 10 器件提供较低 V_{CC} 和 V_{CCP} 电压电平的同时保持器件指定速度等级的性能。Arria 10 器件在低于额定 V_{CC} 和 V_{CCP} 电压水平下运行时可降低总功耗。

Arria 10 器件所要求的最低电压电平已在其制造过程中被编程到一个熔丝块中。Intel 提供了一个读取该值并将其传达给外部电源稳压器或系统电源控制器的 IP core。只有具备 -V power 选项的 -2 和 -3 速度级器件支持此功能。

使用 SmartVID 功能时，Arria 10 器件需要在额定电压电平上电。在配置和部分重配置模式中，Arria 10 器件继续在额定电压电平上运行。进入用户模式后，Arria 10 器件就可在熔丝块中指定的较低电压下运行。即便使用 SmartVID 功能，在 -2 速度等级器件中仍支持使用错误检测的循环冗余检查(EDCRC)功能。而对于其它速度等级，当执行 EDCRC 功能时，Arria 10 器件需要在额定电压上运行。只有当器件在额定电压上运行时，才支持擦除和部分重配置功能。

相关链接

- [Arria 10 器件中的功耗降低特性](#)
- [SmartVID 控制器 IP 内核用户指南](#)

10.2.2. 可编程电源技术

Arria 10 器件为高速或低速模式操作提供按内核各部分(又称为块，或磁贴)，进行配置的能力。该配置由 Quartus Prime 软件自动执行且无需用户干预。通过片上电路来完成将某块(tile)设置为高速或者低功耗模式的操作，且无需额外电源。在设计编译中，Quartus Prime 软件根据设计的时序约束来决定一个块应该处于高速还是低功耗模式。

Arria 10 tile 包含下列部分：

- 储存逻辑阵列模块(MLAB)/通过布线连接成对的逻辑阵列模块(LAB)对
- 通过布线连接成对和连接相邻数字信号处理(DSP)的 MLAB/LAB 对/储存模块布线
- TriMatrix 储存模块
- DSP 模块

与块相关的所有模块和布线共享相同的高速或低功耗模式设置。默认情况下，包括 DSP 模块或存储器模块的块被设定为高速模式，以实现最佳性能。未使用的 DSP 模块和存储器模块被设置为低功耗模式，以实现最小化静态功耗。通过禁用 V_{CCERAM} ，未使用的 M20K 模块被设置为睡眠模式，以便降低静态功耗。时钟网络不支持可编程的功耗技术。

通过可编程功耗技术，较快速度等级的 FPGA 比没有可编程功耗技术的 FPGA 可能需要较少静态功耗。对于带有可编程功耗技术的器件，关键路径是设计的一小部分。因此，也具有更少的高速 MLAB 和高速模式的 LAB 对。对于不带有可编程功耗技术的器件，整个 FPGA 必须进行过度设计以满足关键路径上的时序。

Quartus Prime 将设置中未使用的器件资源设置为低功耗模式，以降低静态功耗。它还将设计中不使用的下列资源设置为低功耗模式：

- LAB 和 MLAB
- TriMatrix 储存模块
- DSP 模块

如果 phase-locked loop (PLL) 在设计中被例化，那么将 areset 管脚置位为高电平以保持 PLL 处于低功耗模式。

表 119. Arria 10 器件的可编程功耗性能

该表列出了可用的 Arria 10 可编程功耗性能。速度等级的考量可加入到置换中，以提供系统设计的灵活性。

功能	可编程功耗技术
LAB	Yes
布线	Yes
储存模块	固定设置 ⁽⁴⁷⁾
DSP 模块	固定设置 ⁽⁴⁷⁾
时钟网络	No

相关链接

[Arria 10 GX、GT 和 SX 器件系列管脚连接指南](#)

提供了关于每个电源轨的所需电压电平的更多信息。

10.2.3. 低静态功耗器件等级

Intel 提供比标准功耗器件消耗更低静态功耗且同时保持运行性能的 Arria 10 器件。仅带有功耗选项“L”的特选器件提供低静态功耗器件等级功能。

相关链接

[Arria 10 器件系列和封装](#)

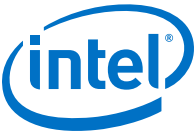
提供了关于订购码的更多信息。

10.2.4. SmartVID 功能实现

SmartVID 功能的实现包含在器件生产过程中被编程到熔丝模块中的 7-bit VID。

7-bit VID 表示 0.85V 到 0.9V 的电压电平。每个器件都有其指定的 7-bit VID。可以使用 SmartVID 控制器 IP 内核读取 7-bit VID。您也可以选择使能和禁用 VID bit 读取。

⁽⁴⁷⁾ 用于设计的 DSP 模块和储存模块的块通常被设置为高速模式。在默认模式下，未使用的 DSP 模块和储存模块被设置为低功耗模式。



从熔丝块中读取 7 位 VID，并通过 Intel 支持的接口发送到外部稳压器或系统电源控制器。接收到 7 位 VID 值后，可调节的稳压器将 V_{CC} 和 V_{CCP} 电压电平调整到 7 位 VID 指定的较低电压。Arria 10 器件支持多种接口方法以将 VID 值传达到外部稳压器或系统电源控制器。首选方法是 7 位并行接口。

Intel 提供外部稳压器和系统电源控制器以支持 SmartVID 功能并可兼容 Arria 10 器件所使用的多接口方法。

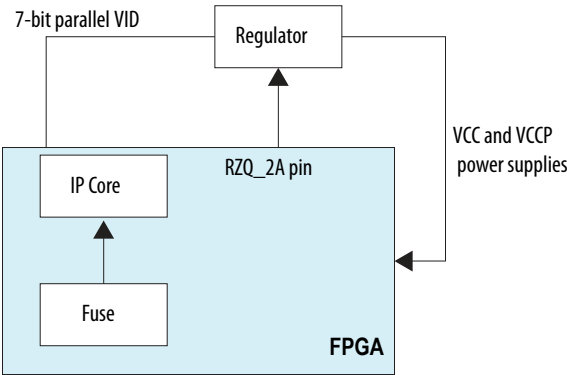
7-Bit 并行接口解决方案

7-bit 并行解决方案是 Intel 支持的一个并行 VID 位接口。该接口要求七个并行 VID 位的 7 个 I/O 管脚，以及 VID_EN 的一个管脚与外部稳压器通信。

Intel 建议将 RZQ_2A 管脚用于 VID_EN 功能。如果 bank 2A 用于 DDR 接口，那么 RZQ_2A 管脚就必须用于校准，则可将其它可用的通用 I/O 管脚用于 VID_EN 管脚功能。在置位 VID_EN 管脚前，需要确保 VID_EN 管脚所属的 I/O bank 及 VID 管脚都已上电。将 VID_EN 管脚连接到 1-k Ω 下拉电阻。

上电期间以及 VID_EN 管脚被置前，VID 管脚需处于三态。Intel 建议使用电平位移器来隔离 VID 信号和稳压控制器。因为一些 VID 位的设置可能会超出最大 V_{CC} 和 V_{CCP} 值。

图 177. 7-Bit 并行解决方案的外部接口连接



下表列出了满足 Intel SmartVID 解决方案的稳压器要求。

表 120. Intel SmartVID 解决方案的稳压器要求

规范	值
电压范围	0.82V – 0.93V ⁽⁴⁸⁾
电压阶跃	10 mV step
V_{CC} 电源	10 W – 100 W
VID 输入	7-Bit VID
继续...	

(48) 此为稳压器向 Arria 10 器件的输出电压范围，包括容限。

规范	值
额定电压	0.85V – 0.9 V ⁽⁴⁹⁾
斜坡时间	0.5 mV/us
VID_EN 管脚	1 pin

相关链接

SmartVID 控制器 IP 内核用户指南

10.3. 电源感应线 (Power Sense Line)

Arria 10 器件支持电源感应线功能。VCCLSENSE 和 GNDSENSE 管脚是差分远程传感管脚，用于监控 V_{CC} 电源。

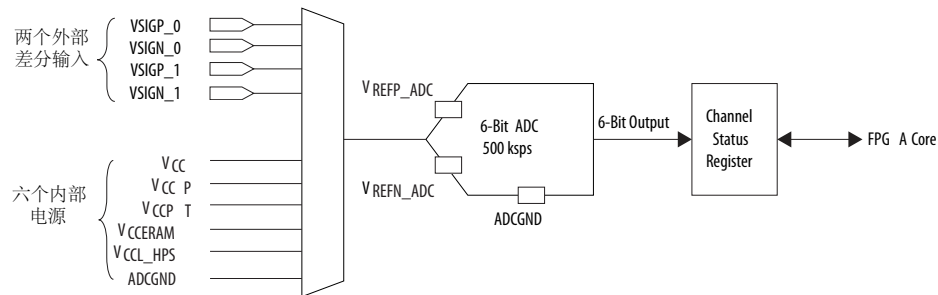
Intel 建议将支持电源感应线功能的稳压器的 VCCLSENSE 和 GNDSENSE 管脚连接起来。将 VCCLSENSE 和 GNDSENSE 线连接到稳压器远程传感输入所需要的条件如下：

- V_{CC} 或 V_{CCP} 电流 > 30A。
- 已使用 SmartVID 功能。

10.4. 电压传感器

Arria 10 支持片上电压传感器。电压传感器通过 6-bit 数字表示已观察到的模拟信号。电压传感器监测 2 个外部差分输入和 6 个内部电源，如下图所示。要获得 ADC 输入，就将 V_{CCPT} 电压值除以 2。要获得实际的 V_{CCPT} 电压值，将 ADC 输出乘以 2。

图 178. 电压传感器



ADC 的转换速度是 500 ksps 累加。当使用多个通道时，每个通道的速度也相应降低。

注意: VREFP_ADC 管脚消耗很少的电流（少于 10 μ A），而大部分电流消耗都是漏电流。对于 VREFN_ADC 管脚，电流少于 0.1 mA。

为获得更好的 ADC 性能，将 VREFP_ADC 和 VREFN_ADC 管脚连接到一个外部 1.25 V 精确参考源 ($\pm 0.2\%$)。通过将 VREFP_ADC 管脚连接到 GND 来激活片上参考源 ($\pm 10\%$)。将 VREFN_ADC 作为模拟信号并通过与 VREFP_ADC 信号一起提供差分 1.25V 电压。

如果未提供外部参考，那么将 VREFP_ADC 和 VREFN_ADC 管脚连接到 GND。

(49) 器件的额定上电电压是 0.9 V。



相关链接

[Altera 电压传感器 IP 内核用户指南](#)

10.4.1. 外部模拟信号的输入信号范围

可配置 ADC 来测量单极性模拟外部输入信号。

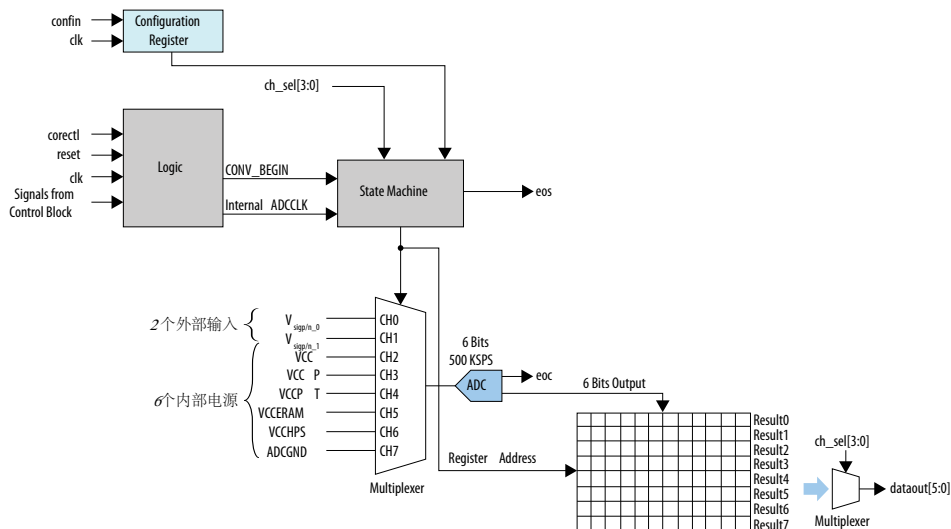
10.4.1.1. 单极性输入模式

在单极性输入模式中，相对于在 VSIGN 管脚上测得的电压，VSIGP 管脚上的电压必须一直为正。VSIGP 输入必须始终由外部模拟信号驱动。VSIGN 管脚被连接到一个局部接地或共模信号。

10.4.2. 在 Arria 10 器件中使用电压传感器

您可以使用电压传感器功能来监控关键性片上电源和外部模拟电压。Arria 10 器件的电压传感器模块支持从 FPGA 内核接入。以下小节介绍在 Arria 10 器件中使用电压传感器的流程。

图 179. 电压传感器组件



10.4.2.1. 使用 FPGA 内核接入对电压传感器进行访问

用户模式期间，可实现一个软 IP 来访问电压传感器模块。要从内核架构访问电压传感器模块，需要在 Quartus Prime 工程中包含以下 WYSIWYG atom：

实例-1: 通过 WYSIWYG Atom 访问电压传感器模块 Voltage Sensor Block

```
twentynm_vsbblock<name>
(
    .clk (<input>, clock signal from core),
    .reset(<input>, reset signal from core),
    .corectl(<input>, core enable signal from core),
    .coreconfig(<input>, config signal from core),
    .confin(<input>, config data signal from core),
    .chsel(<input>, 4 bits channel selection signal from core),
    .eoc(<output>, end of conversion signal from vsblock),
    .eos(<output>, end of sequence signal from vsblock),
    .dataout(<output>, 12 bits data out of vsblock)
);
```

表 121. 电压传感器模块 WYSIWYG 的说明

端口名称	类型	说明
clk	输入	来自内核的时钟信号。电压传感器最多支持 11-MHz 时钟。
reset	输入	有效低电平复位信号。复位信号必须异步地由低跳变到高以使电压传感器开始转换。当复位信号为高电平时，所有寄存器被清零，并且内部电压传感器时钟被关断。
corectl	输入	有效高电平信号。“1”表示电压传感器对于内核访问而被使能。“0”表示电压传感器对于内核访问而被禁用。
coreconfig	输入	串行配置信号。有效高电平。
confin	输入	对配置寄存器进行配置的内核的串行输入数据。内核访问模式的配置寄存器是 8-bit 宽。
chsel[3:0]	输入	4-bit 通道地址。指定要转换的通道。
eoc	输出	表示转换结束。每个通道数据的转换结束后，这个信号被置位。
eos	输出	表示序列结束。在所选序列的一个周期内完成转换后，该信号被置位。
dataout[11:0]	输出	<ul style="list-style-type: none"> dataout[11:6]—6-bit 输出数据。 dataout[5:0]—保留。

10.4.2.1.1. 内核访问模式的配置寄存器

内核访问配置寄存器是 8 位寄存器。

图 180. 内核访问配置寄存器

D7	D6	D5	D4	D3	D2	D1	D0
NA	CAL	NA	NA	BU1	BU0	MD1	MD0

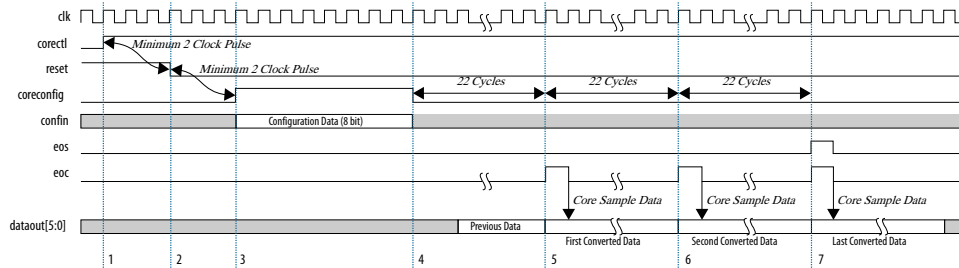
表 122. 为内核访问配置寄存器的说明

位编号	位名称	说明
D0	MD0	通道定序器的模式选择： <ul style="list-style-type: none"> MD[1:0]=2'b00—通道 2 到通道 7 的通道定序器周期 MD[1:0]=2'b01—通道 0 到通道 7 的通道定序器周期 MD[1:0]=2'b10—通道 0 到通道 1 的通道定序器周期 MD[1:0] = 2'b11—由 IP 内核控制。指定 chsel[3:0] 的转换的通道。
D1	MD1	
D2	BU0	
D3	BU1	
D4	NA	已保留。设置为“0”。
D5	NA	已保留。设置为“0”。
D6	CAL	校准使能位。“0”表示校准关闭，“1”表示校准打开。当校准关闭时，校准结果不包括在最后 12 位转换的数据中。
D7	NA	已保留。设置为“0”。

10.4.2.1.2. 当 MD[1:0]不等于 2'b11 时，在内核访问模式中访问电压传感器

下面的时序结构图显示了当 MD [1:0]不等于 2'b11 时，IP 内核在内核访问模式中访问电压传感器的要求。

图 181. MD [1:0]不等于 2'b11 时的时序结构图

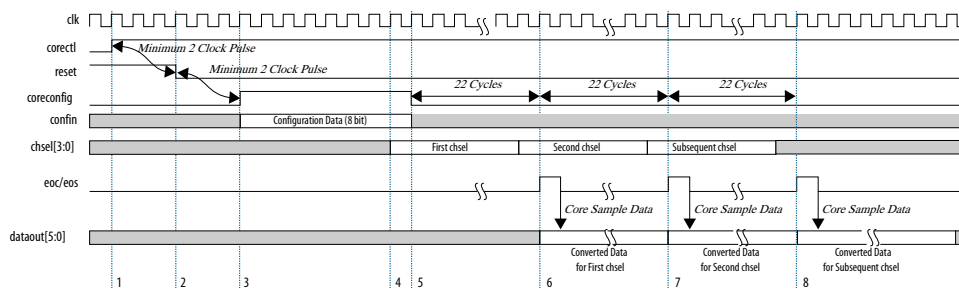


1. `corectl` 信号的低到高跳变使能内核访问模式。
至少等待两个时钟脉冲，然后继续步骤 2。
2. 置低 `reset` 信号将电压传感器从复位状态释放。
至少等待两个时钟脉冲，然后继续步骤 3。
3. 通过写入配置寄存器以及置位 `coreconfig` 信号 8 个时钟周期而配置电压传感器。内核访问模式的配置寄存器是 8 位宽，并且配置数据被串行地移入配置寄存器。
4. `coreconfig` 信号变低表示根据配置寄存器中定义的配置的转换的开始。
5. 轮询 `eoc` 和 `eos` 状态信号来检查 MD[1:0]定义的第一个通道的转换是否完成。在 `eoc` 信号的下降沿锁存 `dataout[5:0]` 信号的输出数据。
6. 轮询 `eoc` 和 `eos` 状态信号来检查 MD[1:0]定义的接下来的通道的转换是否完成。在 `eoc` 信号的下降沿锁存 `dataout[5:0]` 信号的输出数据。
7. 重复步骤 6 直到 `eos` 信号被置位，表明 MD[1:0]指定的通道的一个周期的转换的完成。
 - a. 当电压传感器完成最后通道的转换时，`eoc` 和 `eos` 信号在同一个时钟周期被置位。
 - b. 通过写入配置寄存器而中断电压传感器的操作只能在 `eos` 信号的一个周期结束后执行。
8. 步骤完成时，如果 `corectl` 和 `reset` 信号保持不变，那么转换将再次重复同样的步骤，直到 `corectl` 为 0，并且 `reset` 为 1。如果要测量其他步骤，那么重复步骤 2 到步骤 7。

10.4.2.1.3. 当 MD[1:0]等于 2'b11 时，在内核访问模式中访问电压传感器

下面的结构图显示了当 MD[1:0]等于 2'b11 时，在内核访问模式中访问电压传感器的 IP 内核的要求。

图 182. 当 MD [1:0]等于 2'b11 时的结构图



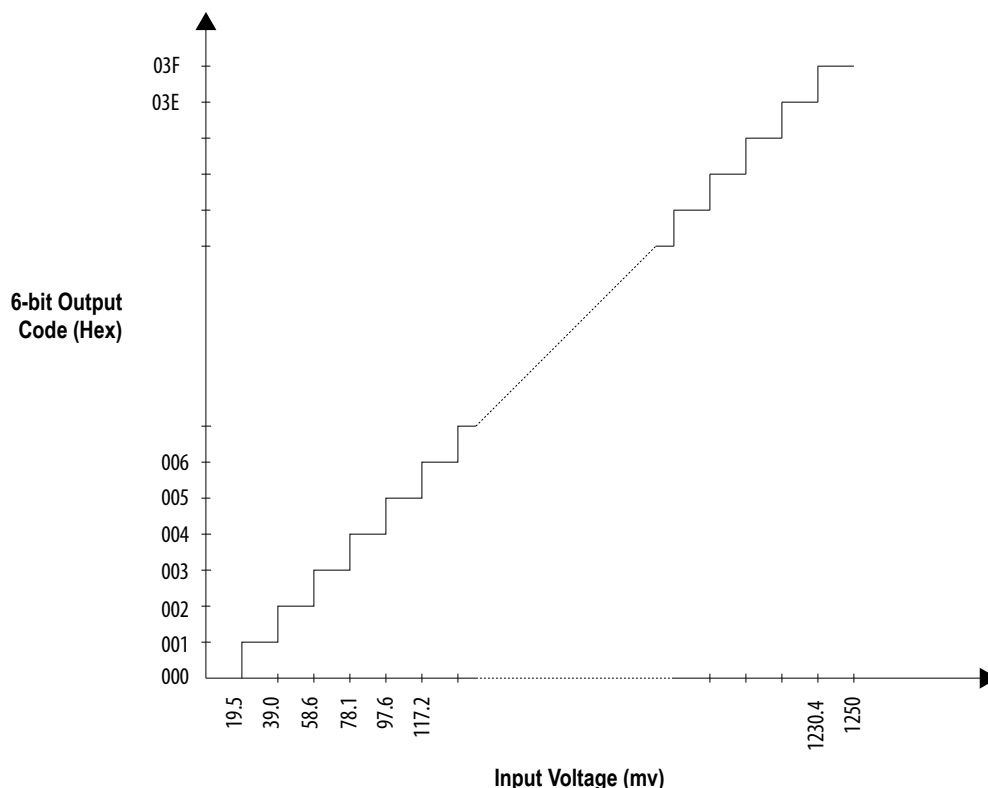
1. `corectl` 信号的低到高跳变使能内核访问模式。

- 至少等待两个时钟脉冲，然后继续步骤 2。
2. 置低 `reset` 信号将电压传感器从复位状态释放。
至少等待两个时钟脉冲，然后继续步骤 3。
3. 通过写入配置寄存器以及置位 `coreconfig` 信号 8 个时钟周期而配置电压传感器。内核访问模式的配置寄存器是 8 位宽，并且配置数据被串行地移入配置寄存器。
4. 指定 `chsel[3:0]` 信号的通道的转换。`chsel[3:0]` 信号的数据需要在 `coreconfig` 信号被置低之前准备就绪。
5. `coreconfig` 信号变低表示根据配置寄存器和 `chsel[3:0]` 信号中定义的配置的转换的开始。
6. 指定 `chsel[3:0]` 信号的通道的转换。`eoc` 信号置位之前的一个周期，`chsel[3:0]` 信号的数据需要准备就绪。轮询 `eoc` 和 `eos` 状态信号以检查步骤 4 中 `chsel[3:0]` 信号定义的第一个通道的转换是否完成。在 `eoc` 信号的下降沿，锁存 `dataout[5:0]` 信号的输出数据。
7. 对于所有接下来的通道重复步骤 6。

10.4.2.2. 电压传感器传输函数

下图表示单极性模式的电压传感器传输函数。

图 183. 单极性模式的电压传感器传输函数



10.5. 温度感应二极管

Arria 10 温度感应二极管(TSD)使用 PN 结点二极管的特性来确定芯片温度。了解结点温度对于散热管理很关键。结温是通过环境或者外壳温度，结点到环境(ja)或者结点到外壳(jc)热阻系数和器件功耗来计算的。Arria 10 器件通过使用内部的 TSD 以及内建 模拟到数字转换器 (ADC)电路或者外部的 TSD 以及一个外部温度感应器监控它的芯片温度。这使您可以控制到器件的气流。

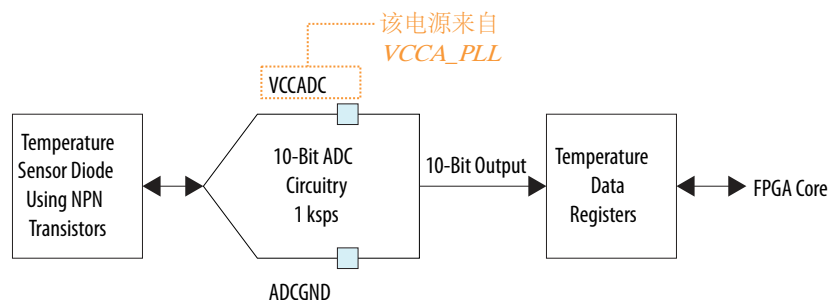
相关链接

Altera 温度传感器 IP 内核用户指南

10.5.1. 内部温度感应二极管

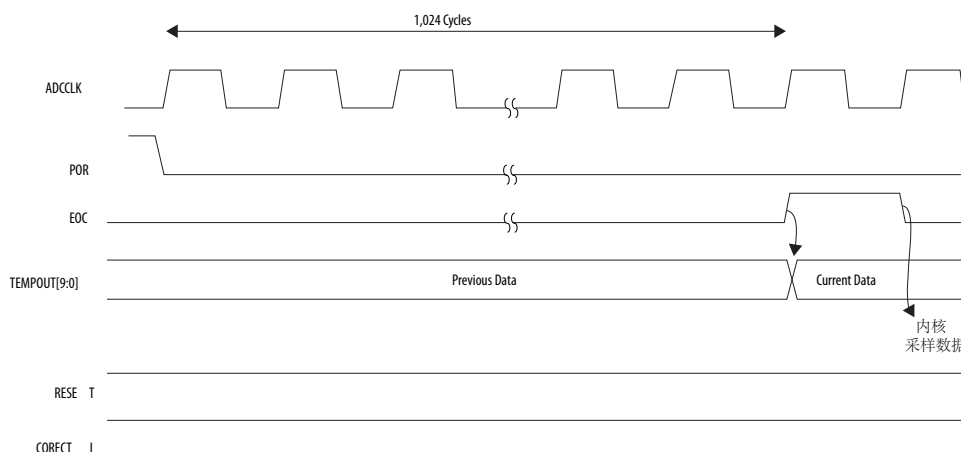
Arria 10 器件支持内部 TSD 以及一个内置 10-bit ADC 电路来监控芯片温度。Arria 10 器件使用一组 NPN 晶体管，以感应温度并生成用于转换的自身的参考电压。内部 TSD 的转换速度大约是 1 ksp/s。

图 184. 内部 TSD 结构图



要在用户模式期间读取芯片温度，需要置位 CORECTL 信号从低到高。有效高电平 RESET 信号用于随时复位寄存器。ADC 电路使用 1,024 个时钟周期来完成一个转换。EOC 信号变高一个时钟周期表明转换的完成。FPGA 内核在 EOC 信号的下降沿读取 TEMPOUT[9:0] 信号的数据。

图 185. 内部 TSD 结构图



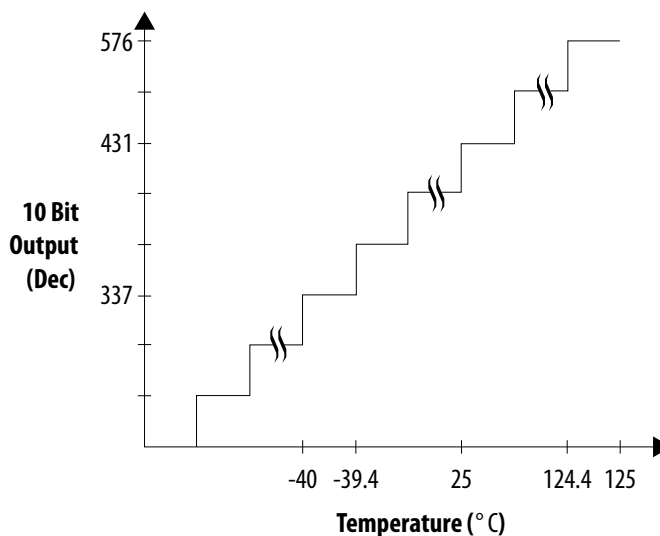
相关链接

- [内部温度传感二极管规范](#)
提供了关于 Arria 10 内部 TSD 规范的更多信息。
- [Altera 温度传感器 IP 内核用户指南](#)

10.5.1.1. 内部 TSD 的传输函数

下图显示了内部 TSD 的传递函数。

图 186. ADC 传递函数



使用下面公式，从 tempout[9:0] 计算出温度值：

$$\text{Temperature} = \{ (\text{AxC}) \div 1024 \} - B$$

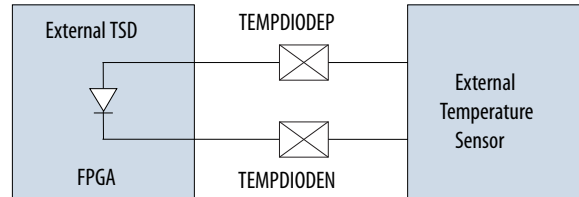
其中：

- A = 693
- B = 265
- C = tempout[9..0] 的十进制值

10.5.2. 外部温度感应二极管

Arria 10 外部 TSD 需要两个管脚用于电压参考。下图显示了如何将外部 TSD 与外部温度传感器件连接以支持 Arria 10 芯片温度的外部感应。

图 187. TSD 外部管脚连接



TSD 是非常敏感的电路，根据器件的使用方法，它可能受到板上其它走线或器件自身封装内耦合噪声的影响。从 Arria 10 器件到外部温度传感器的接口信号取决于 millivolts (mV) 的差异，与在外部 TSD 管脚处所见相同。切换 TSD 管脚旁的 I/O 会影响温度的读取。Intel 建议在器件休止期间进行温度读取，或者使用带有内置 ADC 电路的内部 TSD。

以下是 TSD 外部管脚连接的板级连接指南：

- TEMPDIODE_P /TEMPDIODE_N 走线的最大走线长度 必须少于 8 英寸。
- 并行的布置两个走线，将它们彼此靠近的放置，并且在两边放置接地的保护路径。
- Intel 建议两条走线的线宽和与其它走线的线间距为 10-mils。
- 通过最小数量的过孔和穿接布线两个走线，以便使热电偶的影响最小化。
- 确保在两个走线上的过孔的数量相同。
- 确保两个走线的长度大致相同。
- 通过将 GND 平面放置在二极管走线和高频信号之间，避免与触发信号进行耦合(例如时钟和 I/O)
- 为了进行高频的噪音过滤，在 TEMPDIODE_P /TEMPDIODE_N 走线之间，放置一个外部的电容(靠近外部芯片)。对于 Maxim 器件，使用一个 2200 pF 到 3300 pF 之间的外部电容。
- 放置一个 0.1 uF 旁路电容靠近外部器件。
- 您可以同时使用外部 TSD 和内建 ADC 电路的内部 TSD。
- 如果您仅使用内部的 ADC 电路，那么外部的 TSD 管脚(TEMPDIODE_P /TEMPDIODE_N)可以连接到 GND，因为外部的 TSD 管脚不被使用。

要了解关于器件规范和连接指南的详细信息，请参考器件生产商的外部温度传感器数据表。

相关链接

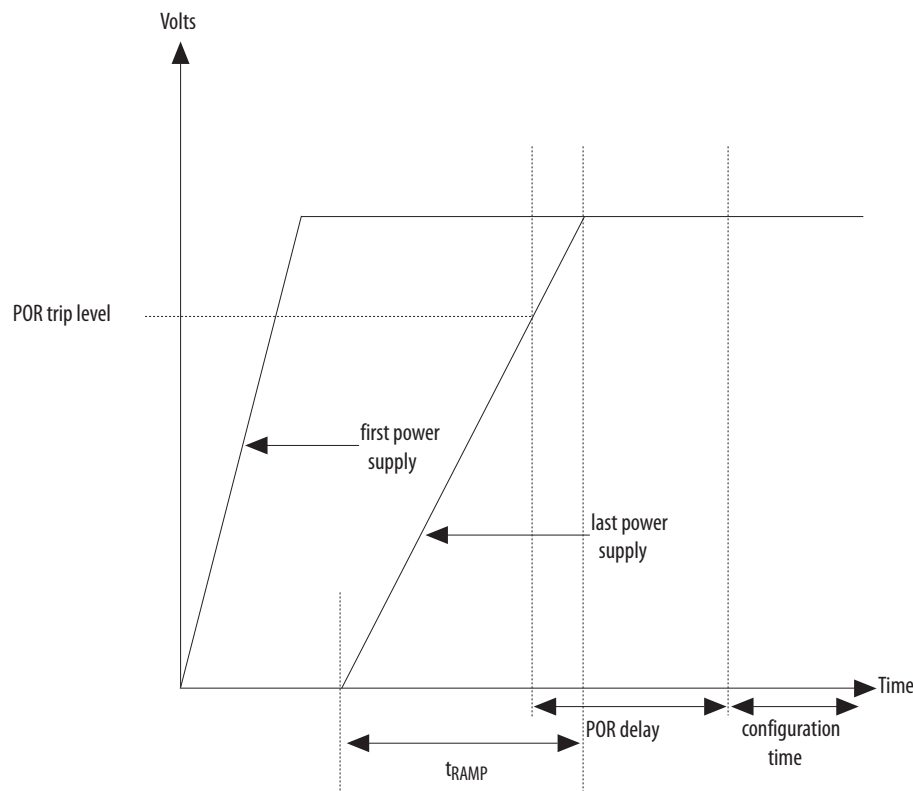
- [外部温度传感二极管规范](#)
提供了关于外部 TSD 规范的详细信息。
- [Arria 10 GX、GT 和 SX 器件系列管脚连接指南](#)
提供了关于不使用外部 TSD 时，TEMPDIODE_P/TEMPDIODE_N 管脚连接的更多信息。

10.6. 上电复位电路

POR 电路把 Arria 10 器件保持在复位状态中，直到电源供应输出达到建议的操作范围。

上电 Arria 10 器件直到全部供电达到最大供电斜坡时间 (t_{RAMP}) 中的推荐操作范围时，就出现 POR 事件。如果未满足 t_{RAMP} ，Arria 10 器件 I/O 管脚和编程寄存器保持三态，在此期间的器件配置可能会失败。

图 188. t_{RAMP} 和 POR 延迟之间的关系



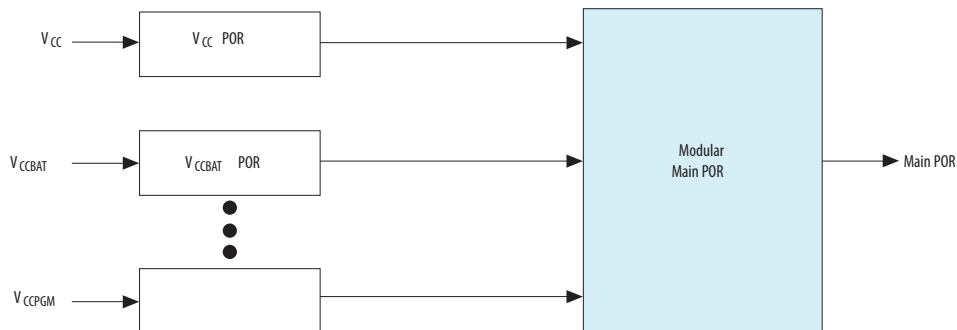
Arria 10 POR 电路使用单独的检测电路独立地监测每个与配置相关的电源。主要的 POR 电路由所有单个监测器的输出选通 (gated)。当电源开始上升, 主要 POR 信号被置位。最后的上升电源达到 POR 断路电平 (伴随一个 POR 延迟) 后此信号被置位。您可以设置 MSEL 管脚以选择快速的或标准的 POR 延迟时间。

关于配置通过协议 (CvP) 配置方案, 从第一个电源斜坡上升到最后一个电源斜坡的总斜坡时间必须少于 10 毫秒。必须选择第一个 POR 以便有足够的时间进行 PCIe*链路初始化和配置。

用户模式中, 当任意监控的电源低于它的 POR 跳变电平时, 主 POR 信号被置位。置位 POR 信号强制器件进入复位状态。

在上电模式期间, POR 电路检查由 V_{CCPT} 和 V_{CCPGM} 电源上电的 I/O 电平位移器的功能性。主 POR 电路等待所有的单个 POR 电路释放 POR 信号, 然后支持控制模块开始对器件进行编程。

图 189. Arria 10 器件 POR 简化图



相关链接

- [POR 规范](#)
提供了关于 POR 延迟规范的详细信息。
- [MSEL 管脚设置](#)
提供了关于每个 POR 延迟的 MSEL 管脚设置的更多信息。
- [建议的操作条件](#)
提供了关于电源斜坡时间的详细信息。

10.6.1. POR 电路监控和未监控电源

表 123. Arria 10 POR 电路监控和未监控电源

监控电源	未监控电源
<ul style="list-style-type: none"> • VCCBAT • VCC • VCCIO ⁽⁵⁰⁾ • VCCERAM • VCCP • VCCPT • VCCPGM • VCLL_HPS ⁽⁵²⁾ ⁽⁵¹⁾ 	<ul style="list-style-type: none"> • VCC_GXB • VCCR_GXB • VCC_GXB • VCCA_PLL • VCCIO_HPS ⁽⁵²⁾ • VCCPLL_HPS ⁽⁵²⁾

注意: 要退出 POR 的器件, 即便未使用易失性密钥, 也必须上电 VCCBAT 电源。

10.7. 上电和断电序列

Arria 10 器件要求上电和断电排序。电源序列被分成 3 个电源组。

⁽⁵⁰⁾ Only for VCCIO of bank 2A.

⁽⁵¹⁾ VCLL_HPS 是被监控的电源, 仅用于 HPS 模块, 不门控主 POR。如果不使用 HPS 模块, 那么将 VCLL_HPS 连到 GND。

⁽⁵²⁾ 仅片上系统(SoC) FPGA 支持这些电源。



注意: 上电和断电期间不可外部驱动 I/O 管脚以避免 I/O 管脚上电流过载:

- I/O 管脚电流过载会影响器件使用寿命及可靠性。
- 3 V I/O 管脚上的过载电流会损毁 Arria 10 器件。

表 124. 电源组上电排序

电源组	斜坡顺序 (Order to Ramp)		条件
	上电	下电	
第 1 组	首先	最后	<ul style="list-style-type: none"> • 上电时, 第 1 组中的所有电源轨必须提升到额定电压的 90% 后, 第 2 组中的电源轨才能开始提升。如果 V_{CC} 和 V_{CCP} 电压水平与 V_{CCT_GXB}, V_{CCR_GXB}, 和/或 V_{CCERAM} 不同, 则首先提升 V_{CC} 和 V_{CCP} 到额定电压的 90%。然后, 任意顺序提升 V_{CCT_GXB}, V_{CCR_GXB}, 和 V_{CCERAM}。 • 断电期间, 必须最后斜降第一组中的 V_{CC}。 • V_{CC}, V_{CCP}, 和 V_{CCERAM} 必须连接到同一稳压器, 除非您正使用 SmartVID 功能, 则 V_{CCERAM} 必须由单独的稳压器供电。
第 2 组	第二	第二	<ul style="list-style-type: none"> • 上电时, 第 2 组中的所有电源轨必须提升到额定电压的 90% 后, 第 3 组中的电源轨才能开始提升。 • 断电期间, 第 2 组中的所有电源轨降低到额定电压的 10% 后, 第 1 组中的电源轨才能开始降低。 • 第 2 组中的电源轨可按任何顺序提升/降低。 • 只要第 3 组中的 V_{CCIO}, V_{CCPGM}, V_{CCIO_HPS} 为 1.8 V 并与第 2 组共享相同的稳压器, 就可与第 2 组中的电源轨一起提升/下降。
第 3 组	第三	第一	<ul style="list-style-type: none"> • 断电期间, 第 3 组中的所有电源轨降低到额定电压的 10% 后, 第 2 组中的电源轨才能开始降低。 • 第 3 组中的电源轨可按任意顺序提升/降低。

若无法遵循完整的断电序列, 就必须在断电时满足下列条件, 从而将通过 FPGA 看见的不必要行为最少化:

- 第 1 组在最后断电
- 避免所有电源轨中的板级电源波动和毛刺

断电序列是上电序列的逆序。当遵循正确的电源序列时, I/O 管脚在上电或断电期间处于三态。

对于断电, 请确保所有电源轨在断电序列开始后的 100 ms 以内断电。

图 190. Arria 10 器件的上电排序要求

只要第 3 组中的 VCCIO, VCCPGM, VCCIO_HPS 为 1.8 V 并与第 2 组共享相同的稳压器, 就可与第 2 组中的电源轨一起提升/下降。

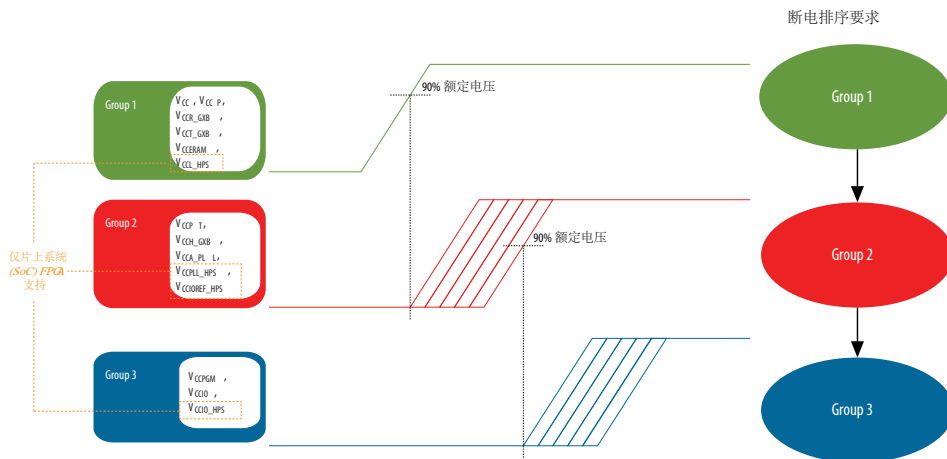


图 191. Arria 10 器件的上电排序要求

只要第 3 组中的 VCCIO, VCCPGM, VCCIO_HPS 为 1.8 V 并与第 2 组共享相同的稳压器, 就可与第 2 组中的电源轨一起提升/下降。

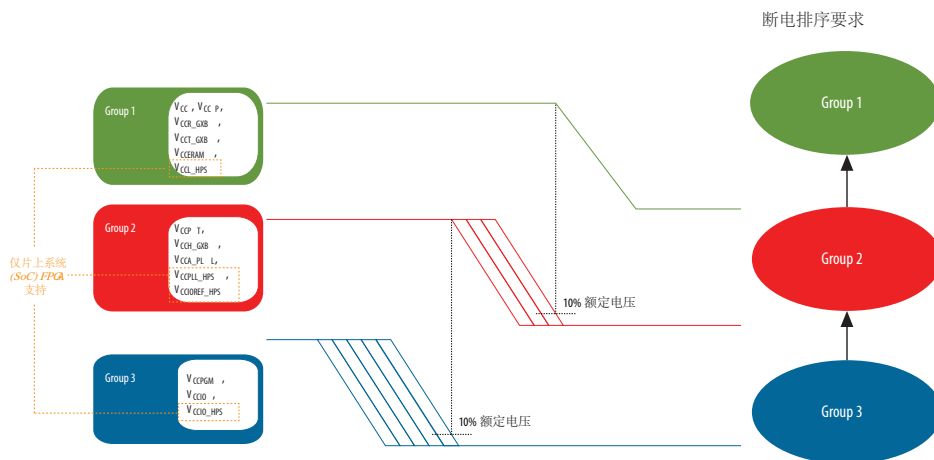
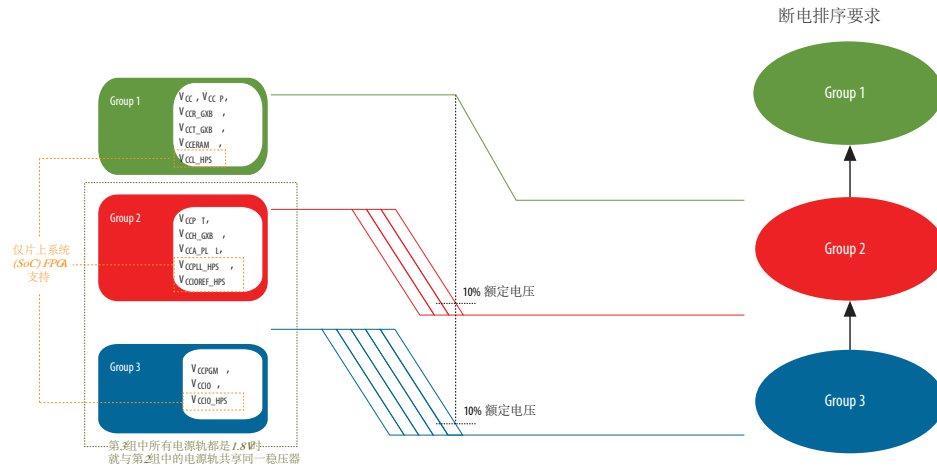


图 192. Arria 10 器件的上电排序要求，只要第 3 组中的所有电源轨均为 1.8 V 且与第 2 组共享相同稳压器。



注意: 上电或断电序列期间， V_{CCBAT} 可按任意顺序上电或断电。

所有电源轨必须单调地上电。在器件数据手册中指定的 t_{RAMP} 时间内，将所有电源轨上电到名义电压电平。上电排序必须满足标准或快速 POR 延迟时间。

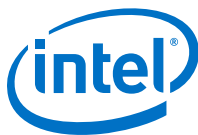
10.8. 电源设计

Arria 10 器件的电源要求会根据每个特定使用情况下的静态和动态电源而不同。要在断电前把 Arria 10 器件的动态电源量降低到可忽略不计，就保持 $nCONFIG$ 管脚低电平以强制 Arria 10 器件进入复位状态。Intel 关于电源管理解决方案的 **Enpirion®** 系列，结合了完善的设计工具，成为 Arria 10 器件最佳电源设计。Enpirion 系列包含的电源管理解决方案兼容 Arria 10 器件使用的多种接口方式，并旨在支持 Arria 10 功耗降低功能，如 SmartVID 功能。

Arria 10 器件需要稳定的电源运行其多个输入电压轨。可依据系统因素（如电压需求，噪声敏感度和排序）分组多个输入轨需求。Arria 10 GX, GT, 和 SX 器件系列管脚连接指南提供了关于输入轨分组的详细建议。用于 Arria 10 器件的 PowerPlay Early Power Estimators (EPE) 工具还可根据每个特定 Arria 10 使用情况无缝且自动提供输入电源轨需求和特定器件建议。关于个别输入轨电压和电流需求的总结在“Report”选项卡中，而输入轨分组和特定电源建议分别位于“Main”和“Enpirion”选项卡中。

相关链接

- [Arria 10 GX、GT 和 SX 器件系列管脚连接指南](#)
提供了关于电源管脚连接指南和电源稳压器共享的详细信息。
- [PowerPlay 早期功耗评估器\(EPE\)和功耗分析器](#)
提供了关于每个电源轨的电源和电流要求的更多信息。
- [Intel FPGA 电源管理 PowerSoC 解决方案](#)
提供更多关于针对上电 FPGA 而设计的 Intel 电源管理 IC 和 PowerSoC 解决方案的信息。
- [Arria 10 和 MAX 10 器件的电源配送网络\(PDN\)](#)



10.9. 文档修订历史

日期	版本	修订内容
2017 年 3 月	2017.03.15	<ul style="list-style-type: none">更名为 Intel。更新了上电和断电序列部分。
2016 年 10 月	2016.10.31	<ul style="list-style-type: none">删除了 AN692 的链接。更新了关于电压传感器模块 WYSIWYG 列表的说明。更新了关于上电和断电排序的话题以强调 I/O 管脚电流过载会影响器件可靠性甚至损毁器件。
2016 年 6 月	2016.06.13	<ul style="list-style-type: none">更新了 Altera SmartVID 解决方案列表稳压器要求中 VID_EN 管脚的值。更新了上电和断电排序部分以包含更多关于断电排序的信息。为单极性模式图示添加了电压传感器传输函数。
2016 年 5 月	2016.05.02	<ul style="list-style-type: none">更新了 WYSIWYG Atom 到 Access the Voltage Sensor Block 的实例。更新了 Altera SmartVID 解决方案表格中稳压器要求的电压范围和额定电压范围。更新了电压感应器模块 WYSIWYG 表格的说明。更新了内核访问配置寄存器表格。更新了电源组上升序列表格中第一组上电序列的条件。更新了通电复位电路部分中 CvP 配置方案的要求。删除了 V_{CC} PowerManager 功能的支持。
2015 年 12 月	2015.12.14	<ul style="list-style-type: none">对“Arria 10 POR 电路监控和没有监控的电源”表中的 V_{CCIO} 和 V_{CCL_HPS} 电源轨增添了注释。更新了“内部 TSD 时序”图中的 RESET 和 CORECTL 信号。更新了 ADC 传输功能的公式。更新了 SmartVID 功能所支持的速度等级器件。更新了“电源组上电顺序”表中的第 1 组的条件。更新了“上电和断电顺序”部分。更新了“电压传感器”部分。更新了“外部温度传感二极管”部分。将两极输入模式支持从电压传感器特性中删除。将 JTAG 访问模式支持从电压传感器特性中删除。删除了“电压传感器传输功能”部分。
2015 年 11 月	2015.11.02	<ul style="list-style-type: none">更新了“ADC 传输功能”图。将 Quartus II 更改成 Quartus Prime。
2015 年 6 月	2015.06.15	<ul style="list-style-type: none">对“电压传感器”部分中的 VREFP_ADC 和 VREFN_ADC 管脚增添了注释以描述当前电流。更新了“ADC 传输功能”图。
2015 年 5 月	2015.05.04	<ul style="list-style-type: none">更新了上电和断电序列，添加了每组电源轨的断电序列要求。更新了表 10-4 中 config 端口的描述。更新了“内部 TSD 的传输功能”部分，添加了从 tempout[9:0] 值计算温度的公式。将“SmartVID 和 V_{CC} PowerManager 功能实现”部分中所支持的并行 VID 比特接口更新为 7 比特。更新了 SmartVID 和 V_{CC} PowerManager 的电压范围的注释，包括了公差。将片上参考源更新为 ±10%。
2015 年 1 月	2015.01.23	<ul style="list-style-type: none">更新了单极性输入部分。更新了电压传感器部分的 VREFP_ADC 管脚的片上参考源。更新了使用 JTAG 访问电压传感器部分中的步骤。
继续...		



日期	版本	修订内容
		<ul style="list-style-type: none"> 更新了 Description for the Voltage Sensor Block WYSIWYG 表中的 <code>reset</code> 和 <code>corectl</code> 端口的描述。 更新了“内部温度传感二极管”部分，如何在用户模式下读取晶片的温度。 更新了“当 MD[1:0]不等于 2'b11 时的时序”图。 更新了“当 MD[1:0]等于 2'b11 时的时序”图。 更新了内部 TSD 时序图。
2014 年 8 月	2014.08.18	<ul style="list-style-type: none"> 增添了“SmartVID 和 V_{CC} PowerManager 功能实现”部分。 在 Arria 10 器件部分添加了使用电压传感器。 对内部 TSD 部分添加了传输功能。 添加了电源设计部分。 更新了动态功耗公式部分。 更新了功耗降低技术部分。 更新了 SmartVID 部分。 更新了可编程的功耗技术部分。 更新了电压传感器部分。 更新了上电和关电排序部分
2013 年 12 月	2013.12.02	首次发布。