请注意：本文档不再进行更新。本文档可能包含旧内容和过时的商标。

请参考英文版本以获取最新更新

https://www.intel.com/content/www/us/en/programmable/documentation/lit-index.html

Please take note that this document is no longer being maintained. It may contain legacy content and trademarks which may be outdated.

Please refer to English version for latest update at

https://www.intel.com/content/www/us/en/programmable/documentation/lit-index.html

21. UART 控制器

cv_54021-1.2

硬核处理器系统（HPS）对异步串行通信提供两个 UART 控制器。UART 控制器基于工业标准 16550 UART 控制器。UART 控制器是 Synopsys® DesignWare® APB 通用异步接收器 / 发射器（DW_apb_uart）外围的实例。

UART 控制器功能

UART 控制器提供以下功能性和功能：

- 可编程的字符属性，例如每字符的数据位数，可选的奇偶校验位和结束位数
- 换行符生成和检测
- 直接存储器访问（DMA）控制器接口
- 优先的中断识别
- 可编程的 baud 速率
- 假起始位检测
- 每 16750 标准的自动流程控制模式
- 内部环回模式支持
- 128-bit 发送和接收 FIFO 缓冲器深度
  - FIFO 缓冲器状态寄存器
  - FIFO缓冲器访问模式（用于FIFO缓冲器测试）使能主器件进行的接收FIFO缓存器的写入和主器件进行的发送 FIFO 缓冲器的读取
- 映射寄存器减少软件开销并且提供可编程的复位
- 发送器保持寄存器空（THRE）中断模式

UART 控制器结构图和系统集成

图 21 – 1 显示了 HPS 系统中 UART 控制器的集成。

图 21 – 1. UART 结构图

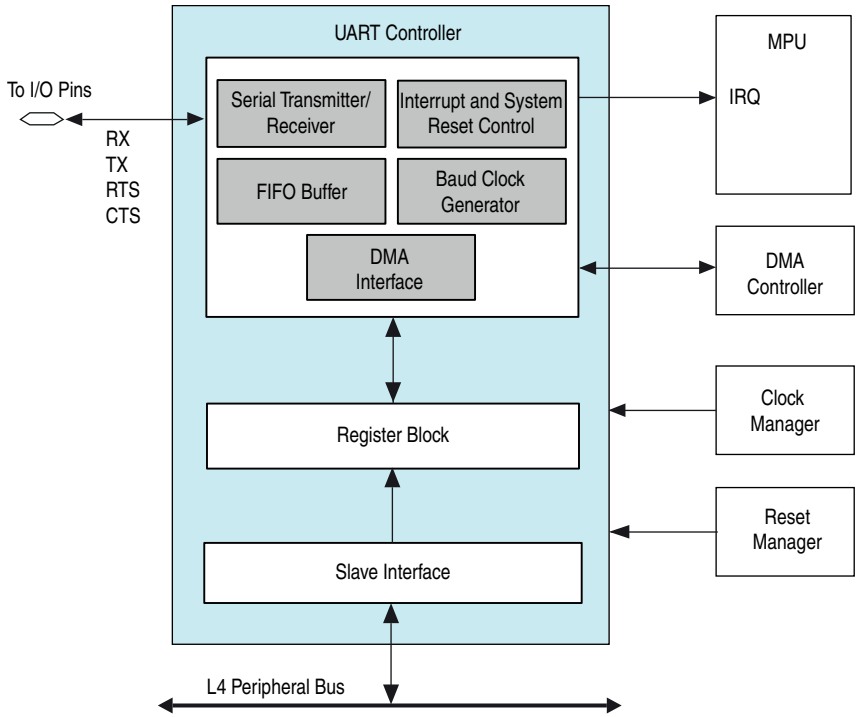


表 21 – 1 提供了 UART 控制器中主模块的总结说明。

表 21 – 1. UART 控制器模块说明

模块	说明
从接口	组件和 L4 外设总线之间的从接口。
寄存器模块	提供了主 UART 控制、状态和中断生成功能。
FIFO 缓冲器	提供了 FIFO 缓冲器控制和存储。
Baud 时钟生成器	生成了发送器和接收器 baud 时钟。通过 100 MHz 的参考时钟，UART 控制器支持 95 baud 到 6.25 Mbaud 的传输速率。这可支持所有已知 16550 器件的通信。通过编程中断使能或锁高因子 (IER_DLH) 和接收缓冲器、发送保持或锁低因子 (RBR_THR_DLL) 寄存器而控制波特率。
串行发送器	将写入到 UART 的并行数据转换成串行数据并且对发送添加控制寄存器指定的所有额外位。这个转换成的串行数据，称为一个字符，退出串行 UART 中的模块。

表 21 – 1. UART 控制器模块说明

模块	说明
串行接收器	将 UART 格式中接收到的串行数据字符（由控制寄存器指定）转换成并行形式。奇偶校验错误检测、帧错误检测和换行符检测在该模块中执行。
DMA 接口	UART 控制器包含一个 DMA 控制器接口可以指示什么时候接收到的数据可用或什么时候发送 FIFO 缓冲器需要数据。DMA 需要两个通道了，一个用于发送，另一个用于接收。UART 控制器支持单一和突发传输。可以使用 FIFO 缓冲器和非 FIFO 缓冲器模式中的 DMA。 要了解更多指定信息，请参考 *Cyclone® V 器件手册*第 3 卷中的 *DMA Controller* 章节。

UART 控制器的功能说明

HPS UART 基于工业标准 16550 UART。UART 支持外设、 modem （数据载波设备）或数据集的串行通信。主器件 (CPU) 通过从总线将数据写入 UART。UART 将数据转换成串行格式并且发送到目的器件。UART 也接收串行数据并且将其存储，以便用于主器件 (CPU)。†

UART 的寄存器控制字符长度、波特率、奇偶校验生成和检测以及中断生成。UART 的单中断输出信号由触发置位的几个优先中断类型支持。您可以使用控制寄存器单独地使能或禁用每个中断类型。

FIFO 缓冲器支持

UART 控制器包含 128-byte FIFO 缓冲器来缓冲发送和接收数据。FIFO 缓冲器访问模式支持主器件为了测试目的写入接收 FIFO 缓冲器和读取发送 FIFO 缓冲器。FIFO 缓冲器访问模式通过 FIFO 访问寄存器 (FAR) 而被使能。一旦使能，发送和接收 FIFO 缓冲器的控制部分被复位并且 FIFO 缓冲器被视为空。

当使能 FIFO 缓冲器访问模式时，可以正常将数据写入发送 FIFO 缓冲器；然而，在该模式中不出现串行传输并且无数据离开 FIFO 缓冲器。可以使用发送 FIFO 读取 (TFR) 寄存器读回写入到发送 FIFO 缓冲器的数据。TFR 寄存器在发送 FIFO 缓冲器的顶端提供当前数据。

同样的，也可以从 FIFO 缓冲器访问模式中的接收 FIFO 缓冲器读取数据。因为 UART 的正常操作在该模式中止，所以您必须将数据写入接收 FIFO 缓冲器以将其读回。接收 FIFO 写 (RFW) 寄存器将数据写入接收 FIFO 缓冲器。10-bit 寄存器的两个较高位将帧错误和奇偶错误检测信息写入到接收 FIFO 缓冲器。RFW 的 Bit 9 表示一个帧错误而 RFW 的 bit 8 表示一个奇偶校验错误。虽然不可以从接收缓冲器寄存器读回这些位，但当问题数据在接收 FIFO 缓冲器的顶部时，可以通过读取线路状态寄存器 (LSR) 和通过检查相应的位检查这些位。

自动流程控制

UART 包含 16750 兼容的 request-to-send(RTS) 和 clear-to-send(CTS) 串行数据自动流程模式。可以通过 modem 控制寄存器 (MCR.AFCE) 使能自动流程控制。

自动 RTS 模式

当以下情况出现时，自动 RTS 模式变为有效：

- RTS（MCR.RTS bit 和 MCR.AFCE bit 都被设置）
- FIFO 缓冲器被使能（IIR_FCR.FIFOE 位被设置）

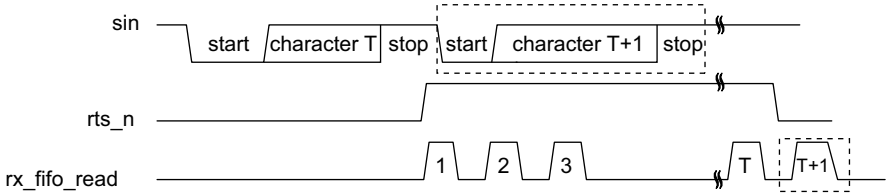
当使能自动 RTS 时，当接收 FIFO ***缓冲器程度***达到 RCVR 触发（IIR_FCR.RT）设置的阀值时，rts_n 输出管脚被强制无效（高电平）。当 rts_n 连接到另一个 UART 器件的 cts_n 输入管脚时，另一个 UART 停止发送串行数据直到接收 FIFO 缓冲器具有可用空间（直到它完全为空）。

可选的接收 FIFO 缓冲器阀值是未满的 1、¼、½ 和 2。因为一个额外的字符可能在 rts_n 无效之后被发送到 UART（由于数据已经进入其它 UART 中的发送器模块），所以将阀值设置为未满的 2 支持最大限度使用一个字符裕量的 FIFO 缓冲器。

一旦接收 FIFO 缓冲器通过读取接收器缓冲器寄存器（RBR_THR_DLL）完全为空时，rts_n 又会变为有效（低电平），***发出信号提示另一个 UART 继续发送数据***。

即使当您设置正确的 MCR 位，如果 FIFO 缓冲器通过 FCR.FIFOE 被禁用，那么自动流程控制也被禁用。当自动 RTS 不被实现或禁用时，rts_n 由 MCR.RTS 单独控制。图 21 – 2 说明了自动 RTS 操作。在图 21 – 2 中，字符 T 被接收到是因为在下一个字符进入发送 UART 发送器之前没有检测到 rts_n。

图 21 – 2. 自动 RTS 时序



自动 CTS 模式

当下列情况发生时，自动 CTS 模式变为无效：

- AFCE（MCR.AFCE bit 被设置）
- FIFO 缓冲器被使能（通过 FIFO 缓冲器控制寄存器 IIR_FCR.FIFOE bit）

当自动 CTS 被使能时（有效），任何时候 cts_n 输入 b 变为无效时（高电平），UART 发送器都会被禁用。这会防止接收 UART 的 FIFO 缓冲器的上溢。

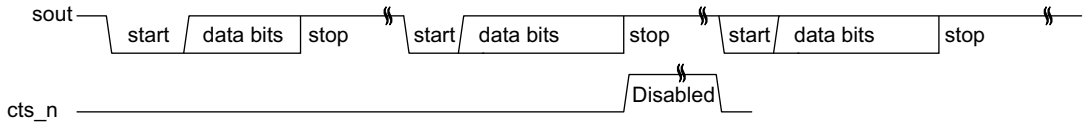
如果在最后停止位中间之前没有使 cts_n 输入失效，那么另一个字符在禁用发送器之前被发送。当发送器被禁用时，可以继续写入，甚至上溢到发送 FIFO 缓冲器。

自动 CTS 模式要求以下序列：

1. UART 状态寄存器被读取以验证发送 FIFO 缓冲器为满（UART 状态寄存器 USR.TFNF 设置为 0）。
2. 当前 FIFO 缓冲程度通过发送 FIFO 程度（TFL）寄存器读取。
3. 可编程 THRE 中断模式必须被使能以便从 LSR 访问 FIFO 缓冲器满状态。

当使用 FIFO 缓冲器满状态时，软件可以在每次写入发送 FIFO 缓冲器之前对此轮询。当 cts_n 输入再次变为有效时（低电平），传输恢复。如果 FIFO 缓冲器通过 IIR_FCR.FIFOE bit 被禁用，那么不管任何其它设置如何自动流程控制也被禁用。当自动 CTS 没有被实现或禁用时，发送器不受 cts_n 影响。图 21 – 3 时序图显示了自动 CTS 操作。

图 21 – 3. 自动 CTS 时序



时钟

UART 控制器被连接到 l4_sp_clk 时钟。时钟输入由时钟管理器驱动。

要了解关于时钟管理器的更多信息，请参考 *Cyclone V 器件手册的*第 3 卷的 *Clock Manager* 章节。

复位

UART 控制器被连接到 uart_rst_n 复位信号。复位管理器驱动冷或暖复位时的信号。

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

中断

当以下其中一个中断类型被使能或激活时，UART 中断输出信号的置位发生：

表 21 – 2. 中断类型和优先权

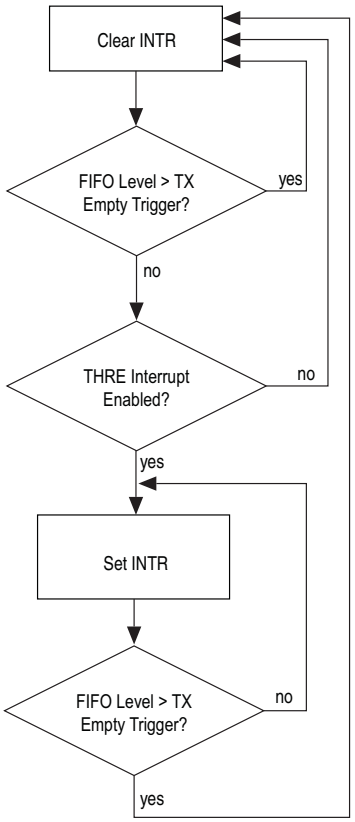
中断类型	优先权
接收器线路状态	最高
接收到的可用数据	第二
字符超时指示	第二
发送保持寄存器空	第三

您可以通过中断使能寄存器 (IER_DLH) 使能中断类型。

可编程的 THRE 中断

UART 具有一个可编程的 THRE 中断模式以提高系统性能。可以通过中断使能寄存器 (IER_DLH.PTIME) 使能可编程的 THRE 中断模式。当使能 THRE 模式时，THRE 中断和 dma_tx_req 信号在编程的发送 FIFO 缓冲器空阀值水平及其以下有效，如图 21 – 4 中的流程图所示。

图 21 – 4. 可编程的 THRE 中断模式

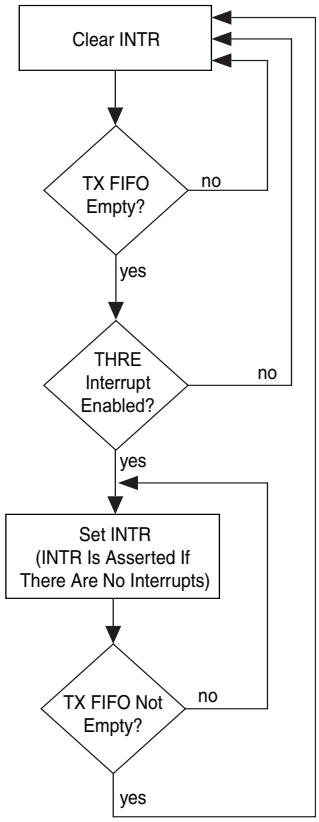


阀值电平被编程进 IIR_FCR.TET。可用空阀值为空、2、¼ 和 ½。最优阀值根据系统的能力以及时的方式开始一个新传输序列。***然而，这些阀值的其中一个应该通过防止发送 FIFO 缓冲器运行为空来增加系统性能而证明为最佳阀值。***

除了中断更改以外，线路状态寄存器 (LSR.THRE) 也从表示发送 FIFO 缓冲器为空切换到表示 FIFO 缓冲器为满。这一更改支持软件通过写入另一个字符之前轮询 LSR.THRE 来填充每个发送序列的 FIFO 缓冲器。这样可以指示 UART 在中断出现和发送数据的任何时候填充发送 FIFO 缓冲器，而不是等待直到 FIFO 缓冲器完全为空。只要系统太忙而不能立即响应时，等待直到 FIFO 缓冲器为空会降低系统性能。当该模式与自动流程控制一起使能时，可以提高系统效率。

当不被选择或禁用时，THRE 中断和 LSR.THRE 正常运行，会显示一个空 THR 或 FIFO 缓冲器。图 21 – 5 显示了不在可编程的 THRE 中断模式中的 THRE 中断生成。

图 21 – 5. 不在可编程的 THRE 中断模式中的中断生成



UART 控制器编程模型

该部分介绍 UART 控制器的编程模型。

DMA 控制器操作

UART 控制器包含一个 DMA 控制器接口，可以表明什么时候接收 FIFO 缓冲器数据可用或什么时候发送 FIFO 缓冲器需要数据。DMA 需要两个通道，一个用于发送，另一个用于接收。UART 控制器支持单一和突发传输。

要了解关于DMA控制器的详细信息，请参考*Cyclone V器件手册*第3卷的*DMA Controller*章节。

UART 控制器中的 RX 和 TX 缓冲器的 FIFO 缓冲器深度 (FIFO_DEPTH) 是 128 个入口。

发送 FIFO 下溢

UART 串行传输期间，无论发送 FIFO 中的入口数少于还是等于 FIFO 控制寄存器 (IIR_FCR) 的发送空触发 (TET) 域中的解码层 (也称为水印层)，都会对 DMA 控制器进行发送 FIFO 请求。通过将一个突发的数据写入发送 FIFO 缓冲器 (长度被指定为 DMA 突发长度)，DMA 控制器做出响应。

要了解关于 DMA 突发长度微代码设置的详细信息，请参考 *Cyclone V 器件手册*第 3 卷的 *DMA Controller* 章节。

数据应该从 DMA 经常获取以便发送 FIFO 连续地执行串行传输，也就是，当 FIFO 开始为空时，另一个 DMA 请求应该被触发。否则 , FIFO 将会用完数据 (下溢)，导致一个 STOP 被插入到 UART 总线。要防止这种情况发生，您必须正确地设置水印层。

发送水印层

请考虑下列进行的假设实例：

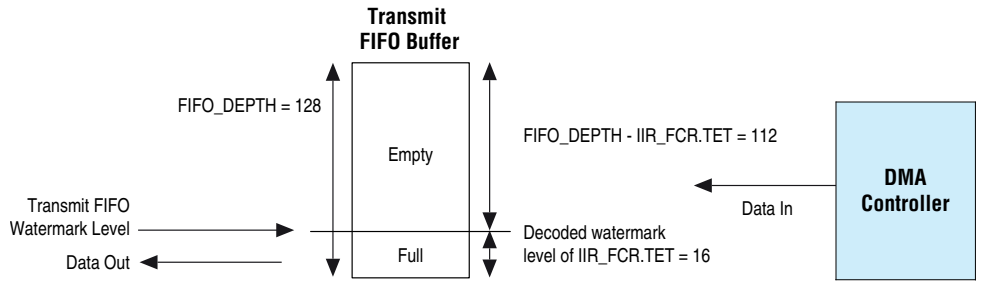
DMA 突发长度 = FIFO_DEPTH - IIR_FCR.TET 的解码水印层

这里 DMA 突发中传输的数据项数等于发送 FIFO 中的空区域。请考虑以下两个不同的水印层设置：

- 用例 1：IIR_FCR.TET = 1，它解码为 16 的水印层：
  - 发送 FIFO 水印层 = IIR_FCR.TET 的解码水印层 = 16
  - DMA 突发长度 = FIFO_DEPTH - IIR_FCR.TET 的解码水印层 = 112
  - UART 发送 FIFO_DEPTH = 128
  - 模块传输容量 = 448

图 21 – 6 显示了水印层等于 16 时的发送 FIFO。

图 21 – 6. 发送 FIFO 水印层 = 16



所需的突发传输数等于模块容量除以每突发的数据项数：

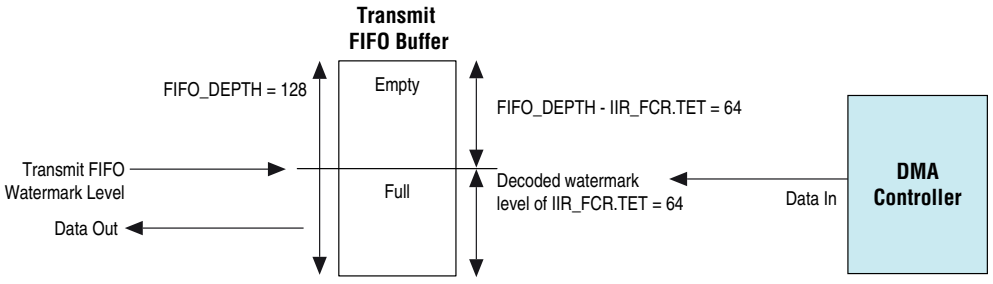
模块传输容量 /DMA 突发长度 = 448/112 = 4

DMA 模块传输中的突发传输数是 4。但是水印层，IIR_FCR.TET 的解码层，很低。因此，UART 串行发送线需要发送数据的发送下溢的可能性很高，但是在发送 FIFO 中不会留下数据。这种情况发生是因为 DMA 没有时间在 FIFO 变空之前执行 DMA 请求。

- 用例 2：IIR_FCR.TET = 3，它解码为 64 的水印层：
  - 发送 FIFO 水印层 = IIR_FCR.TET 的解码水印层 = 64
  - DMA 突发长度 = FIFO_DEPTH – IIR_FCR.TET 的解码水印层 = 64
  - UART 发送 FIFO_DEPTH = 128
  - 模块传输容量 = 448

图 21 – 7 显示了水印层等于 64 时的发送 FIFO。

图 21 – 7. 发送 FIFO 水印层 = 64



模块中的突发传输数：

模块传输容量 /DMA 突发长度 = 448/64 = 7

对于该模块传输，在 DMA 模块传输中具有 15 个目的突发传输。但是水印层，IIR_FCR.TET 的解码层很高。因此 ,UART 发送下溢的可能性很低，因为 DMA 控制器有足够时间在 UART 发送 FIFO 变空之前执行目的突发传输请求。

因此，通过每模块的较多突发传输，第 2 个用例的下溢具有较低可能性。跟第 1 个用例相比，这提供了一个潜在地每模块的较高量突发和较差总线利用率。

因此，选择水印层的目标是最小化每模块的传输数，同时保持下溢情况的可能性到可接受的程度。***在实践中，这是一个速率比函数，在这个速率比上，UART 将数据发送为 DMA 可以响应目的突发请求的速率***。

发送 FIFO 上溢

当发送 FIFO 中没有足够的空间执行目的突发请求时，设置 DMA 突发长度为一个大于水印层 (触发 DMA 请求) 的值会导致上溢。因此，必须遵循以下的公式来避免上溢：

DMA 突发长度 <= FIFO_DEPTH – IIR_FCR.TET 的解码水印层

用例 2 中：IIR_FCR.TET 的解码水印层 = 64，进行了突发请求时的发送 FIFO 中的空间量等于 DMA 突发长度。因此，突发传输完成时，发送 FIFO 可能为满，但是不上溢。

因此，要实现最佳操作，DMA 突发长度应该设置在触发一个发送 DMA 请求的 FIFO 水平；也就是：

DMA 突发长度 = FIFO_DEPTH – IIR_FCR.TET 解码水印层

遵循这一公式会减少模块传输所需的 DMA 突发数，反而这会提高总线利用率。

如果传输期间，UART 控制器在 UART 串行发送线上已经成功地发送了一个或多个数据项，那么发送 FIFO 在 DMA 突发传输结束时不会满。

接收 FIFO 上溢

UART 串行传输期间，***无论接收 FIFO 中的入口数是在 FIFO 控制寄存器 (IIR_FCR) 的接收触发 (RT) 域的解码层还是在其之上***，对 DMA 都会进行接收 FIFO 请求。这被称为水印层。 DMA 通过从接收 FIFO 获取一个突发的数据做出响应。

数据应该由 DMA 经常获取，以便接收 FIFO 连续地接受串行传输，也就是，当 FIFO 开始填充时，另一个 DMA 传输被请求。否则，FIFO 将会被填充数据 (上溢)。要防止这一情况，用户必须正确地设置水印层。

接收水印层

与之前描述的选择发送水印层相似，接收水印层，IIR_FCR.RT 的解码水印层，应该被设置以最小化上溢的可能性，如图 21 – 8 所示。***它是每模块所需的 DMA 突发传输数和上溢发生的可能性之间的权衡***。

接收 FIFO 下溢

将源传输突发长度设置为大于水印层会导致下溢，其中没有足够数据执行源突发请求。因此，必须遵循以下公式来避免下溢：

DMA 突发长度 = IIR_FCR.RT + 1 的解码水印层

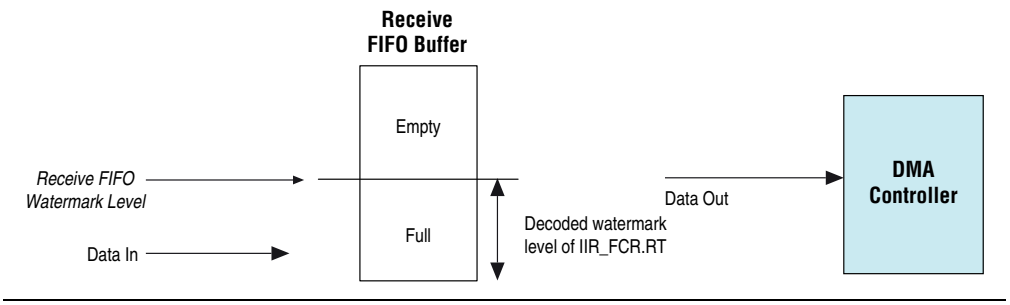
进行了突发请求时，如果接收 FIFO 中的数据项数等于源突发长度，那么突发传输完成时，接收 FIFO 可能为空，但是不下溢。为了实现最佳操作，DMA 突发长度应该被设置在水印层，IIR_FCR.RT 的解码水印层。

遵循该公式会减少模块传输中 DMA 突发数，从而可以避免下溢和提高总线利用率。

如果突发期间，UART 控制器在 UART 串行接收线上已经成功地接收了一个或多个数据项，那么源突发传输结束时接收 FIFO 不会为空。

图 21 – 8 显示了接收 FIFO 缓冲器。

图 21 – 8. 接收 FIFO 缓冲器



UART 控制器地址映射和寄存器定义

地址映射和寄存器定义位于该手册卷附带的 hps.html 文件。点击链接以打开文件。

要查看模块说明和基地址，找到并且点击以下其中一个模块实例的链接：

- uart0
- uart1

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

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

文档修订历史

表 21 – 3 显示了该文档的修订历史。

表 21 – 3. 文档修订历史

日期	版本	修订内容
2012 年 11 月	1.2	少量文本编辑。
2012 年 5 月	1.1	添加了编程模型、地址映射、寄存器定义和复位部分。
2012 年 1 月	1.0	首次发布。