

用于RF收发器的简单基带处理器

Rejeesh Kutty
ADI公司

简介

如今，无线系统无处不在，无线设备和服务的数量持续增长。设计完整的RF系统是一项跨学科设计挑战，模拟RF前端是其中最关键的部分。然而，AD9361等集成RF收发器的推出显著减少了此类设计的RF挑战。这些收发器可为模拟RF信号链提供数字接口，允许轻松集成到ASIC或FPGA，进行基带处理。基带处理器(BBP)允许在终端应用和收发器设备之间的数字域中处理用户数据。此外，使用Simulink等系统建模工具可以轻松完成基带处理器设计。然而，新手用户可能会发现难以理解和解决这个通信系统难题。本文尝试为无线传输通信系统设计和实施简单的RF基带处理器。设计使用AD9361 FPGA参考设计框架，在AD-FMCOMMS2-EBZ和Xilinx® ZC706平台上实施。

本文第一部分详细描述该基带处理器的一般设计原则。该部分主要是BBP的理论介绍。在第二部分，使用ADI公司的AD9361 FPGA参考设计讨论BBP的实际硬件实施。值得注意的是，主要设计目标是使设计尽可能简单，并在实验室环境中演示快速无线数据传输。在使用和干扰RF频谱时，须考虑到法规及其他影响。

基本设计

典型的RF系统如图1所示，直接RF系统除外。该图仅显示了单个数据路径，反方向是该数据路径的镜像图像。本文中提出的相关基带处理器允许对数据进行处理，以使其在两个RF系统之间进行无线传输。下文讨论了基本设计要求。

在两个正交信号I&Q上重复数据

注意，载波相互独立且彼此不同步。因此，发射和接收载波之间存在相位和频率偏移。这将对接收器的解调产生不利影响。一个重要问题是信号反转，正交信号可能会反转其作用，因为偏移会定期合并和漂离。克服这种不确定性的简单方法是在两个正交信号上重复相同数据。

以串行形式发送和接收数据(按位)

大多数情况下，与BBP连接的RF前端接口是DAC和ADC。这些是模拟信号的数字接口。因此，不能简单地将数据发送到DAC输入，并预计在ADC输出端获得相同数据。数据以串行形式发射，将单个位数据映射到DAC的全分辨率。同样，数据以串行形式接收，从ADC的全分辨率解映射。这提供了充足的冗余。如果这些是16位转换器，则接收器将从可能的65536数据集中决定1或0。仅这一点，便可以显著简化解码。

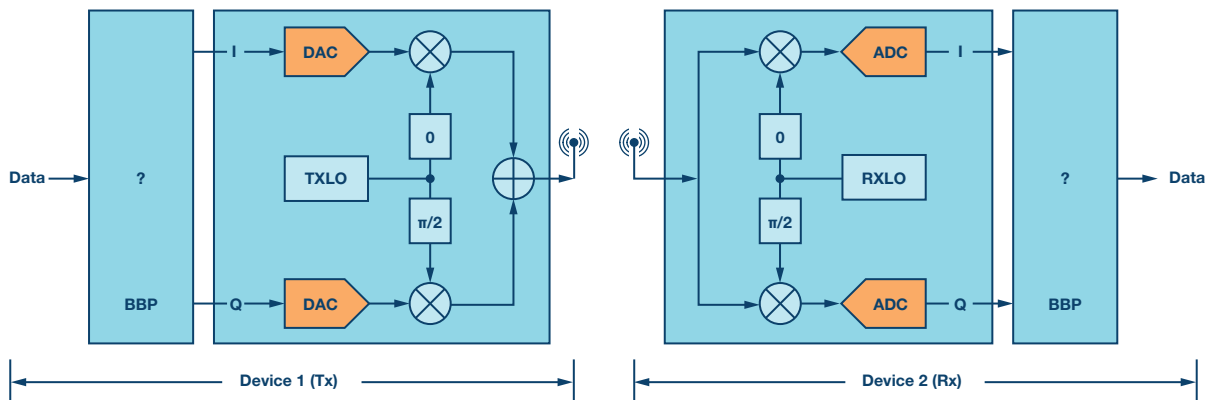


图1. RF系统框图

I&Q信号相互正交

RF前端设备(如AD9361)是I/Q收发器。如果输入是正交信号, 这些设备最有效。这些设备通常沿两个数据路径进行内部I/Q匹配和校正, 以抵消二者之间的任何差异。规则是, 实部(I)信号是余弦函数, 虚部(Q)信号是正弦函数。

调制方案是BPSK

可以部署信号幅度、频率或相位调制的所有常见方法。检测相位差异相对来说更加简单。由于数据以串行形式传输, 因此必然会选择二进制相移键控(BPSK)。

位间隔是8个样本

数据需要时序信息, 位间隔。可能的最大位间隔是采样周期。为了使接收器保持简单, 需要足够的时间来解码信号并做出决定。最简单的时序恢复方法是零交越和峰值检测。在这种情况下, 峰值将不一致。因此, 选择零交越进行位间隔检测和跟踪。两种系统之间也存在载波差异。在某些情况下, 在用户数据的任意端, 样本可能模糊不清。为每半个正弦信号留出4个样本, 位间隔设置为8个样本。因此, 有效的传输速率是采样频率除以8。

数据没有直流成分

时序和相对相位恢复以信号的零交越为基础。因此, 单个信号需要不含任何直流成分。此外, 要求信号每隔一个位间隔允许至少一个零交越。正弦信号兼具两者的属性, 并且非常符合上述BPSK调制方案要求。

数据已加扰

用户数据是任意的很可能是一长串1或0。数据需要加扰, 以便在接收器端恢复时序和相位, 从而更高效地跟踪信号。

数据以数据包的形式传输

由于系统彼此不同步, 因此接收器的信号会存在幅度、频率和相位误差。解调信号是发射信号相对于本地载波发生相位变化的信号。载波可能会跟踪一段时间, 选取数据, 然后再跟踪。因此, 设计需要做好部分数据丢失的准备。为此, 数据以数据包的形式传输。可重复传输多个数据包, 而非整个数据。

使用CRC验证数据包

数据包携带循环冗余校验(CRC)码, 因此如果存在不匹配, 则允许接收器丢包, 并请求再次发送。

在每个前同步码期间完成时序和相位校正

数据包表头携带前同步码, 用于将其从接收到的数据流中划分出来。此外, 接收器使用该前同步码复位信号的时序和相位信息, 以解调数据包数据。

内置性能指标

接收器也支持统计计数器, 如接收到的、丢弃的或校正的数据包数量。这些计数器用于衡量和监控性能指标, 包括误码率和有效数据速率。

总而言之, 数据作为数据包以串行形式发送和接收。数据包携带前同步码和CRC。数据在收发器设备前的中间正交信号上经

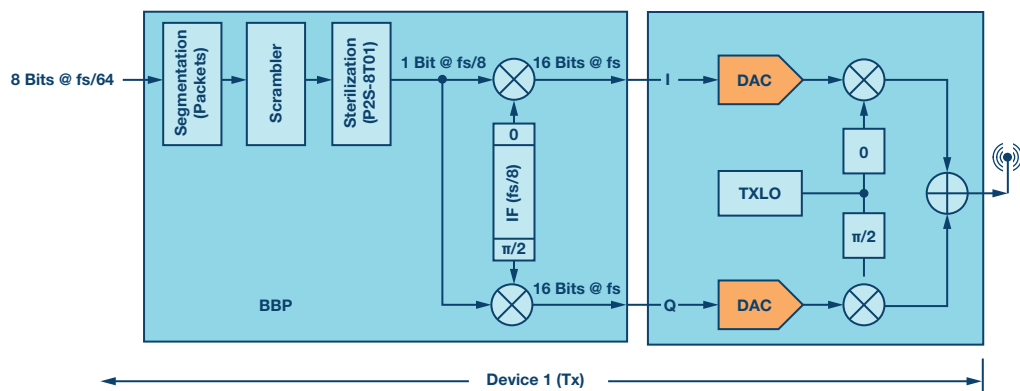


图2. BBP发送功能框图

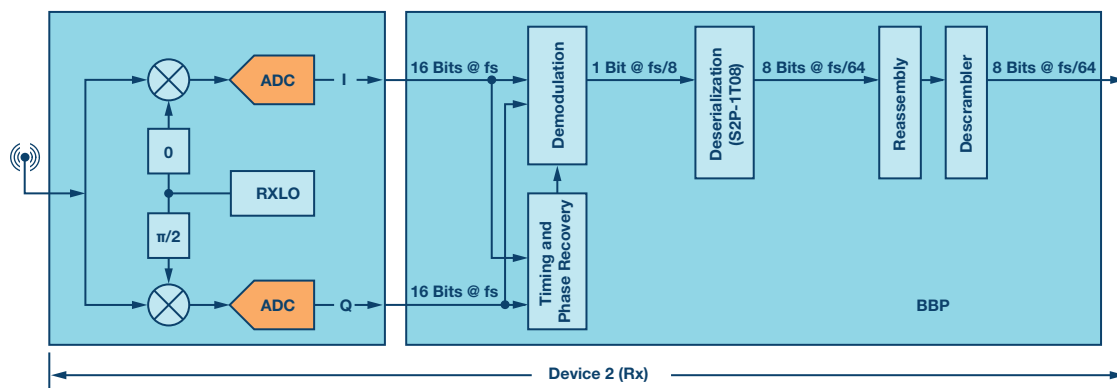


图3. BBP接收功能框图

过BPSK调制和解调。因此，中间信号频率和数据的位速率是采样速率的八分之一。基带处理器模块及上述设计细节如图2和3所示。

发送器读取数据字节(字符宽度)，并将其转换为带有表头或前同步码的数据包。将CRC添加到数据包末端。然后，对数据包数据进行加扰和串行处理。在连接到收发器之前，单个位数据相位调制余弦(I)和正弦(Q)函数。

在接收方向，离线模块恢复并跟踪时序间隔和调制信号的相对相位。该信息用于从输入的ADC样本中恢复串行数据。然后组装到数据包，并进行解扰。在数据包结束时，比较CRC，如果不匹配，则丢弃数据包。如果CRC匹配，数据传递给终端用户。

实现

BBP设计在硬件中实施和测试。硬件是两个评估板的组合：具有Zynq FPGA设备的Xilinx ZC706评估板，以及具有AD9361收发器的AD-FMCOMMS3-EBZ评估板。ADI提供支持该硬件的完整参考设计。该开源设计在主要工具版本中免费提供，可获得完全支持和更新。硬件详细信息参见下列URL：

- [ZC706](#)
- [AD-FMCOMMS3-EBZ](#)
- [Zynq SOC](#)
- [AD9361](#)
- [ADI GitHub库](#)
- [ZC706和AD-FMCOMMS3-EBZ HDL参考设计](#)
- [AXI AD9361 IP](#)

ADI参考设计是支持Linux®框架的嵌入式系统。包含围绕ARM®处理器的各种外设。AD9361设备连接到axi_AD9361 IP外设。它在RF设备和系统存储器之间传输原始采样数据。外设和设备通过Linux内核驱动程序进行初始化和控制。BBP则作为连接到axi_AD9361的另一个IP外设。出于历史原因，BBP IP命名为axi_xcomm2ip。Linux中的用户空间应用程序用于在系统之间控制、发送和接收数据。

在ADI参考设计中，在发送方向，axi_AD9361 IP连接到解包模块(util_upack)，在接收方向，连接到打包模块(util_cpack)。在发送方向，BBP数据插入解包模块和AD9361内核之间。为了使其不影响默认数据路径，BBP支持可选的数据路径多路复用器，以选择解包数据源或BBP数据源。BBP允许参考设计数据路径作为默认路径，并仅在启用时选择BBP数据源。在接收方向，BBP仅连接到AD9361内核。参考设计数据路径不受影响。这允许框架不受妨碍地引导和设置系统。在系统设置后，启用BBP，可通过覆盖默认数据路径来进行数据传输。以ADI参考设计实施的BBP的框图如图4所示。

本文中讨论的设计、初始化和数据传输使用一对这种硬件。设置仅需一对HDMI®监视器、键盘和鼠标及天线。系统彼此完全不同步，但需要相同设置。在每个方向，数据在不同载波上传输。设备1的发射载波频率和设备2的接收载波频率相同，但在另一个方向上不同。然而，如果回送中使用单个设备，发射和接收载波必须具有相同的频率。BBP的HDL设计采用ADI库模块。

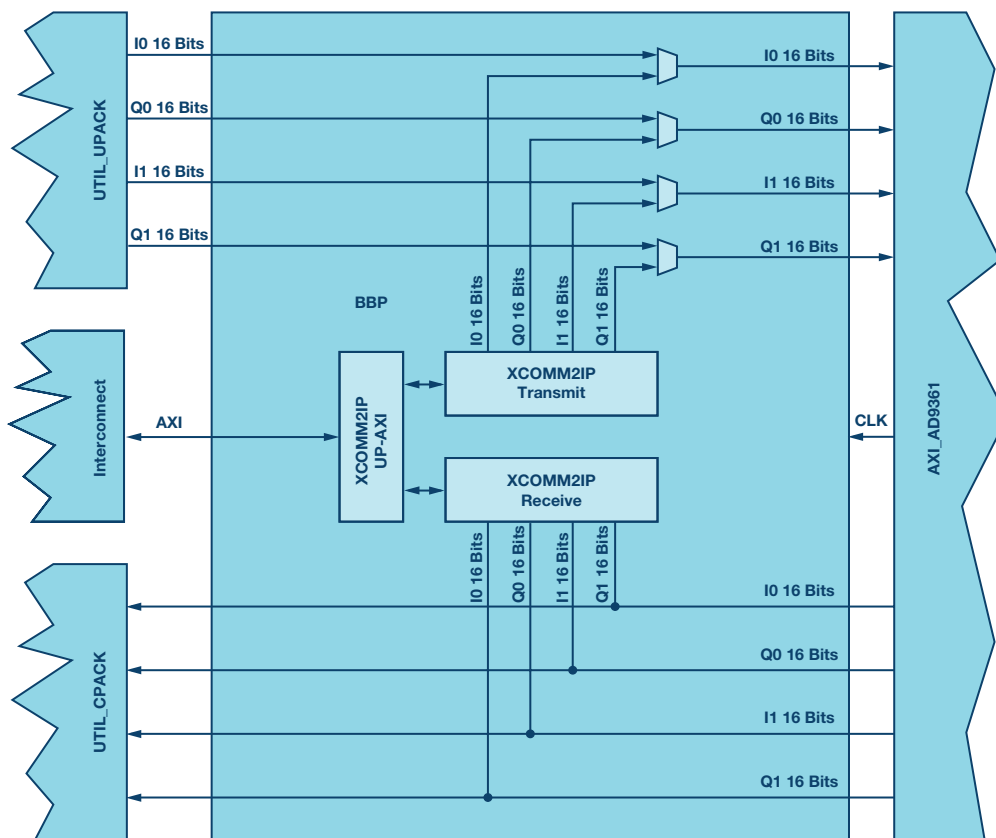


图4. BBP IP框图

表1. BBP寄存器映射

地址	寄存器名称			类型
	类型	名称	说明	默认值
0x000	XCOMM2IP_VERSION			RO
	31:0	VERSION	IP版本。	0x40063
0x008	XCOMM2IP_SCRATCH			RW
	31:0	SCRATCH	暂存寄存器。	0x0
0x800	XCOMM2IP_TX_RESET			RW
	0	TX_RESET	如果设置为0x1, 复位发送。正常工作时, 必须将此位设置为0x0。	0x1
0x804	XCOMM2IP_TX_ENABLE			RW
	0	TX_ENABLE	如果设置为0x0, 数据路径不受影响, 解包数据发送到DAC。如果设置为0x1, BBP数据发送到DAC。	0x0
0x808	XCOMM2IP_TX_REQ			RW
	0	TX_REQ	如果由软件设置, 在发送方向启动数据包传输。当传输完毕时, 该位由硬件自清零。	0x0
0x80C-0x87C	XCOMM2IP_TX_PKT_DATA_3- XCOMM2IP_TX_PKT_DATA_31			W
	7:0	TX_PKT_DATA_3- TX_PKT_DATA_31	数据包数据字节3至31。硬件将前三个字节用于表头, 最后一个字节用于CRC。	W
0xC00	XCOMM2IP_RX_RESET			RW
	0	RX_RESET	如果设置为0x1, 复位接收。正常工作时, 必须将此位设置为0x0。	0x1
0xC08	XCOMM2IP_RX_REQ			RW
	0	RX_REQ	如果由硬件设置, 这表示已接收数据包, 需要由软件读取。在读取数据包数据后, 软件必须清零该位。所有后续的数据包丢弃, 直到该位被软件清零。	0x0
0xC0C-0xC7C	XCOMM2IP_RX_PKT_DATA_3- XCOMM2IP_RX_PKT_DATA_31			W
	7:0	RX_PKT_DATA_3- RX_PKT_DATA_31	数据包数据字节3至31。硬件将前三个字节用于表头, 最后一个字节用于CRC。	W

控制(微处理器)接口

AXI-Lite接口用于通过处理器控制和监控BBP。使用ADI公共库(hdl/library/common/up_axi.v)中的up_axi模块, 可以轻松推断该接口模块。该模块将AXI-Lite接口转换为简单内存, 如读取和写入总线。和任何其他ADI IP一样, 添加内部寄存器和内存。寄存器映射如表1所示。

up_axi模块端口及其端口映射如下所述。

up_rstn: AXI接口复位(异步低电平有效), 连接到s_axi_aresetn。

up_clk: AXI接口时钟, 连接到s_axi_aclk。

up_axi_*: AXI接口信号, 连接到等效s_axi_*端口。

up_wreq, up_waddr, up_wdata, up_wack: 内部写入接口, up_wreq信号与地址和数据一同置位, 以指示写入请求。请求需要通过up_wack端口应答。

如下所示, 实施简单的寄存器写入。

```
always @(negedge up_rstn or posedge up_clk)
begin
if (up_rstn == 0) begin
up_wack <= 'd0;
up_reg0 <= UP_REG0_RESET_VALUE;
end else begin
up_wack <= up_wreq_s;
if ((up_wreq_s == 1'b1) && (up_waddr == UP_REG0_ADDRESS)) begin
up_reg0 <= up_wdata[UP_REG0_WIDTH-1:0];
end
end
end
```

模块在二者之间执行地址转换。AXI接口使用字节地址, 但内部总线使用DWORD地址。结果是, up_axi模块丢弃AXI地址的两个最低有效位, 以生成内部DWORD地址。

up_rreq, up_raddr, up_rdata, up_rack: 内部读取接口, up_rreq信号与地址一同置位, 以指示读取请求。请求需要与读取数据一同通过up_rack端口应答。

如下所示, 实施与上述相同的寄存器用于读取。

```
always @(negedge up_rstn or posedge up_clk)
begin
if (up_rstn == 0) begin
up_rack <= `d0;
up_rdata <= `d0;
end else begin
up_rack <= up_rreq_s;
if ((up_rreq_s == 1'b1) && (up_raddr == UP_
REG0_ADDRESS)) begin
up_rdata <= up_reg0;
end else begin
up_rdata <= 32`d0;
end
end
end
end
```

相同地址转换也适用于读取。读取数据仅在请求时驱动, 否则设置为零。这是因为up_axi模块将单个读取数据从各个地址组传递到OR门。因此, 未选择的地址组需要驱动读取数据零。

如上方寄存器映射表中所列, BBP有三个地址空间。常见寄存器空间映射至0x000、发送(DAC)映射至0x800 (0x200), 接收(ADC)映射至0xC00 (0x300)。软件(Linux用户空间应用程序)应当将发送数据包数据写入缓冲器, 并从另一个缓冲器中读取接收到的数据包数据。数据包大小选择为32字节, 带有3字节前同步码和1字节CRC。

数据接口

对于接收和发送方向的两个通道, AD9361接口内核包含两对16位I/Q数据。内核按照与AD9361数字接口相同的时钟运行。在2R2T模式下, 这是采样速率的4倍。在1R1T模式下, 这是采样速率的2倍。有效数据速率由有效信号控制。因此在2R2T模式下, 每4个时钟置位一次有效。在1R1T模式下, 每2个时钟置位一次有效。BBP旨在支持2R2T和1R1T模式。它使用单个发送和接收通道。内部逻辑在2R2T和1R1T模式下以采样速率运行。然后, BBP在其时钟频率下, 通过接口内核传输数据。这样是为了在BBP内演示时钟转换。在许多情况下, 用户可能希望无论收发器的接口速率如何, 都能在采样速率下运行BBP逻辑。

使用Xilinx基元BUFR和BUFG, 生成采样频率内部时钟。BUFR是分压器, BUFG是高扇出时钟缓冲器。为此, 也可以使用MMCM。如下所示, 生成内部时钟。

```
parameter XCOMM2IP_1T1R_OR_2T2R_N = 0;
localparam XCOMM2IP_SCLK_DIVIDE =
(XCOMM2IP_1T1R_OR_2T2R_N == 1) ? "2" : "4";

BUFR #(.BUFR_DIVIDE(XCOMM2IP_SCLK_DIVIDE))
i_bufnr (
.CLR (1'b0),
.CE (1'b1),
.I (clk),
.O (s_clk_s));

BUFG i_bufgr (
.I (s_clk_s),
.O (s_clk));
```

使用BUFR和BUFG可确保时钟频率锁定, 但会影响相位确定性。最大相位不确定性是单个接口时钟周期。通过带有同步信号的四级寄存器阵列, 可以轻松补偿该不确定性。然而, 设计采用了双端口RAM模块来实现数据传输。这也是为了展示常见信号处理要求的应用实例。使用ADI库内存模块(ad_mem)可以推断出双端口RAM元件。

发送接口

在发送方向, 处理器将数据包数据写入缓冲器(参见上方寄存器映射表)。然后, 请求硬件发送该数据包。BBP将数据包连续发送给设备。在数据包开始时, 检查是否有任何请求。如果没有待处理的请求, 则发送空闲数据包。如果有请求等待处理, 读取并发送数据包缓冲器。

发送逻辑使用自由运行位计数器, 按照位宽运行。当位计数器为0x0时, 更新缓冲器读取地址。由于在数据包传输期间可能会出现处理器请求, 因此在数据包传输开始时就会立即捕获并清零。在数据包传输开始时, 如果请求等待处理, 则应答回至处理器接口。利用请求在缓冲器数据或空闲数据之间进行选择。

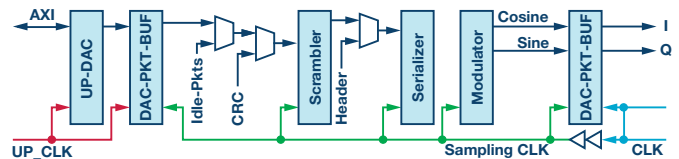


图5. 发送数据路径

数据包数据的前两个字节设置为0xfff0。第三个字节用于表示空闲(0xc5)或数据(0xa6)数据包。CRC字节作为数据包的最后一个字节插入。CRC多项式为 $x^8 + x^2 + x + 1$ 。除表头外的所有字节已加扰。加扰多项式与SONET/SDH ($x^7 + x^6 + 1$)相同。

余弦和正弦查找表用于生成调制载波。在8个样本中, 位间隔等于完整的信号周期(0至 2π)。位数据用于反转信号。然后, 数据写入小缓冲器, 并使用接口时钟, 根据AD9361接口内核中的有效信号读取。

接收接口

在接收方向, 针对表头模式0xfff0, 监控I/Q数据。这种独特的模式在数据包传输中只出现一次。可以发送数据包数据, 以使加扰器输出重复该模式。软件会限制并阻止该做法。12个连续位间隔的该系列同相数据序列用于通过时序恢复模块复位并跟踪接收器时序和相位。因此, 将复位其时序计数器并将其相位值设置为0x1。该序列后的第一个反转被视为0x0。在此之后, 时序恢复模块在整个数据包传输过程中保持其状态不变。

数据恢复模块计算信号的平均值, 并决定信号的当前相位。然后, 与时序恢复模块跟踪的相对相位进行比较。如果发生冲突, 根据过去的信号变化做决定。这是因为冲突通常是由相位切换引起的。

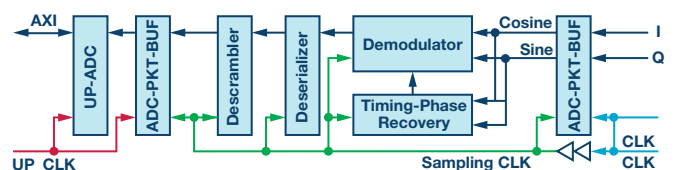


图6. 接收数据路径

然后，解调的位数据组装到字节，并进行解扰。如果传输缓冲器为空，则数据写入传输缓冲器。在数据包结束时验证CRC。如果匹配且传输缓冲器已写入，则通知处理器接口。软件应当监控该请求，如果已设置，通过读取其内容来清空缓冲器。然后，必须清除请求，以便继续进行数据包传输。

构建说明和下载

本文介绍简单RF基带处理器的理论和实施详情。讨论了在ZC706和AD-FMCOMMS3-EBZ硬件上实现该设计的实际方案。快速演示和构建说明的完整设计文件参见<https://wiki.analog.com/resources/fpga/docs/hdl/xcomm2ip>。此页面也详细介绍了HDL设计、软件、RF设置、性能和分析。

作者简介

Rejeesh Kutty于2011年加入ADI公司，任客户解决方案支持部门的HDL工程师。他获得印度科学理工学院的电子工程硕士学位。

在线支持社区



EngineerZone™

访问ADI在线支持社区，
与ADI技术专家互动。提

中文技术论坛

出您的棘手设计问题、浏览常见问题解答，或参与讨论。

请访问 ezchina.analog.com

全球总部
One Technology Way
P.O. Box 9106, Norwood, MA
02062-9106 U.S.A.
Tel: (1 781) 329 4700
Fax: (1 781) 461 3113

大中华区总部
上海市浦东新区张江高科技园区
祖冲之路2290号展想广场5楼
邮编: 201203
电话: (86 21) 2320 8000
传真: (86 21) 2320 8222

深圳分公司
深圳市福田中心区
益田路与福华三路交汇处
深圳国际商会中心
4205-4210室
邮编: 518048
电话: (86 755) 8202 3200
传真: (86 755) 8202 3222

北京分公司
北京市海淀区
西小口路66号
中关村东升科技园
B-6号楼A座一层
邮编: 100085
电话: (86 10) 5987 1000
传真: (86 10) 6298 3574

武汉分公司
湖北省武汉市东湖高新区
珞瑜路889号光谷国际广场
写字楼B座2403-2405室
邮编: 430073
电话: (86 27) 8715 9968
传真: (86 27) 8715 9931

©2016 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners. Ahead of What's Possible is a trademark of Analog Devices. TA14701sc-0-9/16

analog.com/cn

 **ANALOG
DEVICES**
超越一切可能™