

A Low Complex and High Precision Frame Synchronization Method for OFDM System Based on PSS Sequence and Its FPGA Implementation

Zhichao Huang, Kang Wang

School of Information Science and Engineering, Southeast University, Nanjing Jiangsu
Email: huangzhichao@seu.edu.cn

Received: Mar. 29th, 2018; accepted: Apr. 17th, 2018; published: Apr. 24th, 2018

Abstract

Wireless frame synchronization is one of the key technologies in OFDM system. Only through frame synchronization between the base station and the user and the starting point of the wireless frame is found, communication signal can be successfully processed. Synchronization precision directly affects the performance of signal processing in the system. At the same time, how to reduce the complexity of the algorithm and reduce the consumption of hardware resources is also a key problem that must be considered in the algorithm design. In view of the above problems, a low complex and high precision frame synchronization algorithm is proposed in this paper. On the basis of traditional wireless frame synchronization, the algorithm only adds a plural multiplier, two accumulators and related logic modules. It can implement low complex and high precision wireless frame synchronization based on PSS sequence. Meanwhile, this paper implements the algorithm in hardware using LabVIEW FPGA language and the algorithm is deployed in the actual MIMO-OFDM system, operating stably and effectively.

Keywords

OFDM, Frame Synchronization, Low Complex, PSS, LabVIEW FPGA

一种基于PSS序列的OFDM系统低复杂高精度帧同步方法及其FPGA实现

黄志超, 王 康

东南大学信息科学与工程学院, 江苏 南京

Email: huangzhichao@seu.edu.cn

收稿日期: 2018年3月29日; 录用日期: 2018年4月17日; 发布日期: 2018年4月24日

摘要

无线帧同步是OFDM系统中的关键技术之一, 只有在基站与用户之间通过帧定时同步, 确定无线帧的起始点才能顺利进行后续的通信信号处理, 同步精度直接影响到系统的信号处理的性能, 同时如何降低算法的复杂度, 减少硬件资源的消耗, 也是算法设计所必须考虑的重点问题。针对上述问题, 本文提出一种低复杂高精度帧同步算法, 该算法在传统的无线帧同步基础上, 仅添加一个复数乘法器、两个累加器以及相关逻辑模块, 即可实现基于PSS序列的低复杂高精度无线帧同步, 同时本文使用LabVIEW FPGA语言在硬件上实现了该算法, 并且在实际的MIMO-OFDM系统中部署, 算法运行效果稳定有效。

关键词

OFDM, 帧同步, 低复杂度, PSS, LabVIEW FPGA

Copyright © 2018 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

在无线通信系统中, 尤其是以 OFDM (Orthogonal Frequency Division Multiplexing) 即正交频分复用技术为代表的具有固定帧结构的通信系统, 基站和用户必须在空口上进行无线帧的时间同步, 即用户(或者基站)确定基站(或用户者)所发送的无线帧的时域起点位置, 从而接收端按照固有的帧结构可以区分出整个无线帧各个不同传输符号的位置, 为后续的信号处理做准备。无线帧时域同步主要有基于 CP 和基于同步序列的两种方法[1], 前者易受噪声的干扰, 后者的同步精度要高, 且性能稳定。在基于同步序列的帧同步方法中, 总体做法如下: 把接收到的基带数据与本地存储的同步序列进行互相关, 找到互相关器输出值的峰值位置, 即对应着同步序列所对应的位置, 再根据同步序列在整个无线帧中的位置, 从而推出无线帧的起点[1]。

为了与无线帧的结构相匹配, 同步序列所在 OFDM 符号在长度一般较大, LTE 中规定的 OFDM 符号长度为 2048 点(不包括 CP, Cyclic Prefix, 即循环前缀), 同时对于传输速率要求高的系统, 采样率较高, 基带数据吞吐量大。为了减少互相关算法的计算次数, 降低复杂度, 易于在硬件上实现, 目前的帧同步算法多会对基带数据进行降采样, 再进行互相关。这样降低了互相关器的数据输入速率, 减小了互相关窗的大小, 降低了复杂度, 提高无线帧同步速度, 能够在硬件上对同步序列进行实时检测, 然而该同步算法损失了帧同步的精度, 给系统造成了一定的误差, 随着降采样的倍数越高, 误差越大, 帧同步效果越差。

为了解决上述降低采样率后做的帧同步(以下简称为粗同步)带来的同步位置不够精确的问题, 同时避免冗余的计算带来极高的复杂度, 本文提出了一种基于同步序列的无线通信低复杂高精度无线帧同步算法, 相比于粗同步仅加入复杂度极低的模块即可在粗同步基础上, 获得精确的帧同步位置。

本文算法基本按照 LTE 相关标准实现, LTE 是非常成熟的 OFDM 系统, 为人们所熟知, 该系统的采样率为 30.72 MHz, 同步序列为 PSS (Primary Synchronization Signal, 主同步序列) 序列, OFDM 符号的大小为 2048 (不包括 CP 的长度), 每个无线帧为 10 ms, 包含 10 个子帧, 每个子帧包含两个时隙, 每个时隙中包含 7 个 OFDM 符号, 每个无线帧包含一个同步符号, 位于无线帧的起始位置[2] [3] [4]。

图 1 为无线帧格式, 在 LTE 标准的帧格式中做了细节上的修改, 在子帧 0 中两个时隙共 14 个 OFDM 符号, 仅在第一个符号上添加了一个 PSS 符号, 其余符号全部做补零操作。另外在其他子帧中, 放置了上行导频、上行数据、补零、下行导频以及下行数据。

2. 低复杂高精度帧同步算法

本节介绍低复杂高精度帧同步算法的原理以及运行步骤, 首先介绍 PSS 序列的生成过程, 再介绍粗同步的原理, 在此基础上提出低复杂高精度帧同步算法。

2.1. LTE 中同步序列 PSS

PSS 序列是由长度为 63 点的 ZC [3] 序列(Zadoff-Chu)组成, 本身为频域信号, 具有自相关性极高互相关性极低的特点, 通过 IFFT (Inverse Fast Fourier Transformation) 转换到时域, 所得时域信号也具有同样的特点, 适用于加入到无线帧的头部, 可用于无线帧同步使用。

$$a_q = \exp \left[-j2\pi q \frac{n(n+1)/2 + l * n}{N_{zc}} \right] \tag{1}$$

公式(1)是 ZC 序列的生成函数, 其中 N_{zc} 为奇数, $q \in \{1, 2, \dots, N_{zc} - 1\}$ 是 ZC 序列的根指数, $n = 0, 1, \dots, N_{zc} - 1$, l 可以是任何整数, 一般为了简化, $l = 0$ 。

ZC 序列有以下特性:

1) ZC 序列具有恒定振幅, 经过 IFFT 也是恒定振幅的。这限制了峰均值比, 降低了对数据信号的干扰。

2) 任何长度的 ZC 序列具有极好的循环自相关性。

实际中被选中作为 PSS 的 ZC 序列的根指数 q 为 29、34、25, 具体公式如下:

$$ZC_q^{63}(n) = \exp \left[-j \frac{\pi q n(n+1)}{63} \right] \tag{2}$$

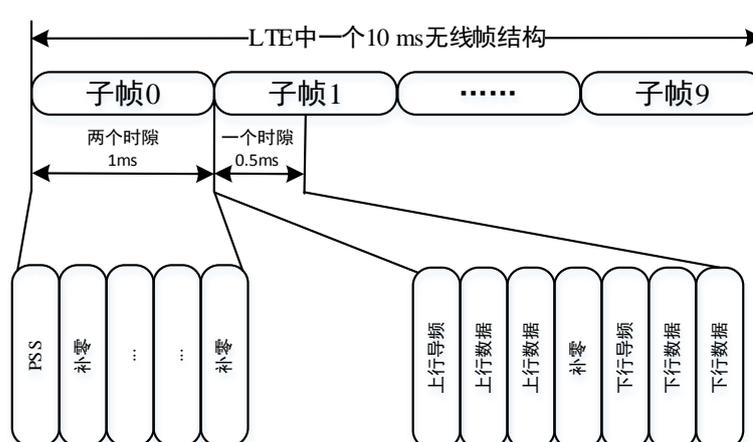


Figure 1. Wireless frame format
图 1. 无线帧格式

该 63 点频域复数信号对称地摆放在 OFDM 符号的中间, 其中 31 号点舍弃, 为 OFDM 信号的直流子载波。

图 2 为 PSS 符号的频域幅度图, 横轴共 2048 个点, 代表 2048 个子载波, 图中只显示了部分点, 纵坐标为 PSS 信号模值, 为恒幅信号。图中可以清楚看到中间点的幅度为 0, 该点为直流子载波, 舍弃不用, PSS 频域信号只有 62 点模值为 1, 其余点全部为 0。

图 3 取根指数为 29 的 PSS 序列经过 2048 点 OFDM 调制得到的时域波形的自相关图, 图中可以看到在中间位置产生了一个极高的峰值, 并且该峰值约为次高峰幅度的 3 倍, 具有极好的自相关性质, 完全可用作无线帧帧的搜索与定位。

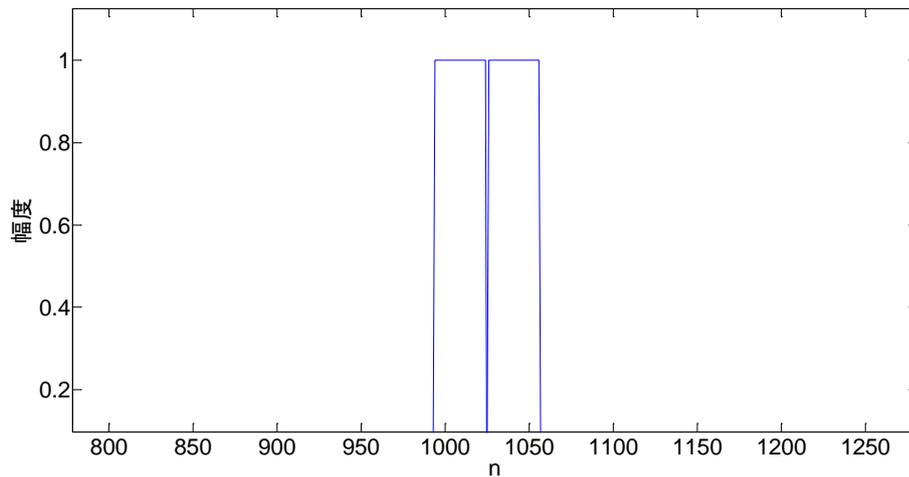


Figure 2. Frequency domain signal of PSS

图 2. PSS 的频域信号

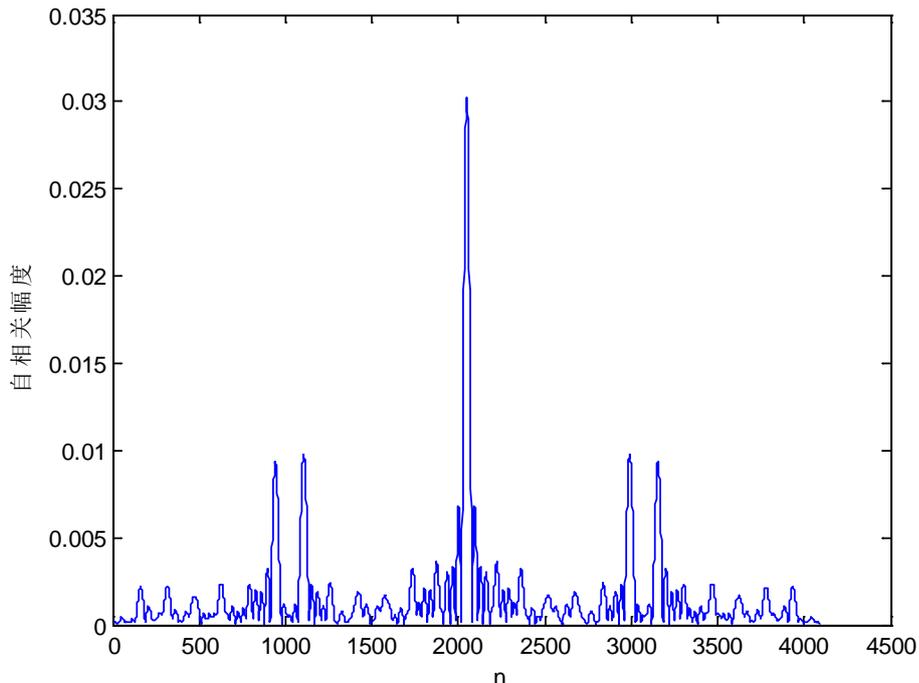


Figure 3. PSS sequence autocorrelation

图 3. PSS 序列自相关

2.2. 粗同步算法

本 OFDM 系统粗同步算法为基于 PSS 序列的帧定时算法, 其中误差范围是(-8, +8)。该帧同步算法部署在 FPGA 上, 为了降低复杂度, 实际中 2048 点相关器实现过于复杂, 而 128 点相关器可采用 Fir 滤波器(Finite Impulse Response, 有限长单位冲激响应滤波器)较为方便地实现, 对基带信号进行了 16 倍降采样, 在同步精度上会产生一定的误差。算法流程见图 4。

步骤一: 由 ADC 采用获取速率为 30.72 MHz 的基带时域信号, 通过低通滤波器, 滤波器带宽为 PSS 信号带宽, 从而获取纯净的 PSS 信号, 有利于提高后续互相关器精度, 对输出的基带信号进行 1/16 抽取, 降低采样率;

步骤二: 对降速后的基带信号输入 Fir 互相关器, 与本地存储的固定 PSS 信号(已做过 1/16 抽取)互相关, 并检测互相关器输出的峰值, 确定峰值的位置, 该位置代表同步位置;

步骤三: 判断所获取的同步位置的有效性, 即确定峰值的幅度是否足够大, 峰值的位置是否与前一个无线帧的位置靠近, 若通过检测即可得到粗同步位置;

以上是粗同步的算法流程, 该算法对 PSS 信号的捕捉能力极高, 同步速度快, 采用 128 点相关器, 极大降低了算法复杂度, 减少硬件资源消耗, 由于加了两个有效性判定的步骤, 使该同步算法能达到极高的同步准确率。该算法的不足是同步位置有(-8, +8)的误差, 会给系统带来一定的 STO。

2.3. 高精度帧同步算法

本 OFDM 系统高精度帧同步算法基于 2.2 中的粗同步算法结果, 再设计小范围内的 2048 点 PSS 序列相关器, 找出(-8, +8)误差范围内无线帧的实际位置, 即找出实际的无线帧同步位置与粗同步算法位置的偏差, 该偏差加上粗同步位置即为实际无线帧同步位置。

图 5 为 OFDM 系统高精度帧同步算法的流程图, 具体流程如下:

步骤一: 对采样后的数据进行低通滤波, 进行粗同步, 得到粗同步位置;

步骤二: 根据粗同步位置, 存储 PSS 序列可能所在位置的全部基带数据, 由于误差为(-N, +N), 那么需要在粗同步位置向前 N 个点, 粗同步确定的 PSS 序列结尾后 N 个点, 以上两点范围内的数据全部存储进存储器, 设 PSS 序列长度为 L, 则存储 L + 2N + 1 个点, 如图 6 所示, 这些点为基带数据经过低通滤波之后的数据点, 但为进行降采样;

步骤三: 用存储的序列与本地固定的 PSS 序列进行 2N + 1 次 L 点互相关, L 取 2048, 并搜索峰值, 得到峰值位置, 相关器的输出序号是 1~2N + 1, 对应偏置为-N~N;

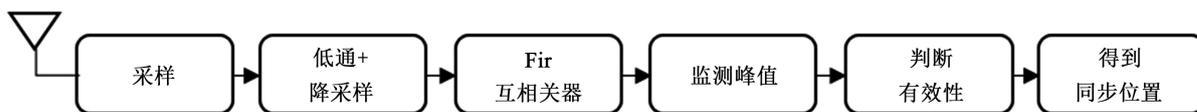


Figure 4. Block diagram of rough synchronization algorithm for OFDM system

图 4. OFDM 系统粗同步算法框图

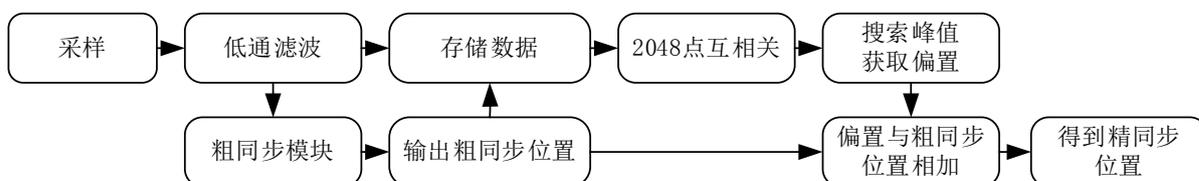


Figure 5. The flow of high precision frame synchronization algorithm

图 5. 高精度帧同步算法的流程

步骤四: 把偏置与粗同步位置相加, 即可得到精同步位置。

在以上精同步算法流程中, 与粗同步相比, 关键是引入了对部分数据进行 2048 点相关运算, 该数据是根据粗同步得到的同步位置, 在时域基带信号上截取的可能存在的部分数据, 其内部包含了整个的 PSS 信号, 仅仅是 PSS 的具体位置不确定, 需要进行 2048 点互相关进行更加精确的搜索, 考虑到一个无线帧 307,200 个采样点中仅仅需要进行 $2N + 1 = 17$ 次 2048 点相关, 计算量较小, 因而可以采用一个复数乘法器、两个累积器进行两层循环来实现, 大大降低了硬件资源的消耗, 同时能保证应有的精度。在理想环境下, 本同步算法找到的同步位置即为实际的同步位置, 不存在误差。

3. 高精度无线帧同步的 LabVIEW FPGA 实现

LabVIEW FPGA 是美国国家仪器公司推出的一款图形化界面 FPGA 开发语言[5] [6], 具有开发简单、可读性强等特点, 被广泛应用于该公司的 FPGA 系列芯片的开发中, 本论文采用该语言进行帧同步算法的实现。粗同步模块是高精度无线帧同步算法的必须步骤, 首先简要介绍粗同步模块的实现[5] [6] [7]。

图 7 为帧同步的低通降采样滤波器、128 点互相关器、2048 点相关器、数据存储流程, 这里使用的低通降采样滤波器为 Xilinx FIR IP 核, 低通滤波器的带宽为 PSS 信号的带宽, 为 0.5 MHz, 目的是滤除

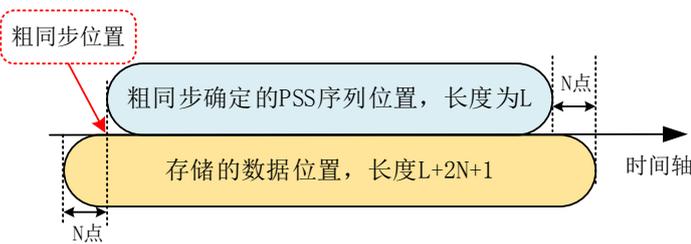


Figure 6. 2048 point correlator input data range
图 6. 2048 点相关器输入数据范围

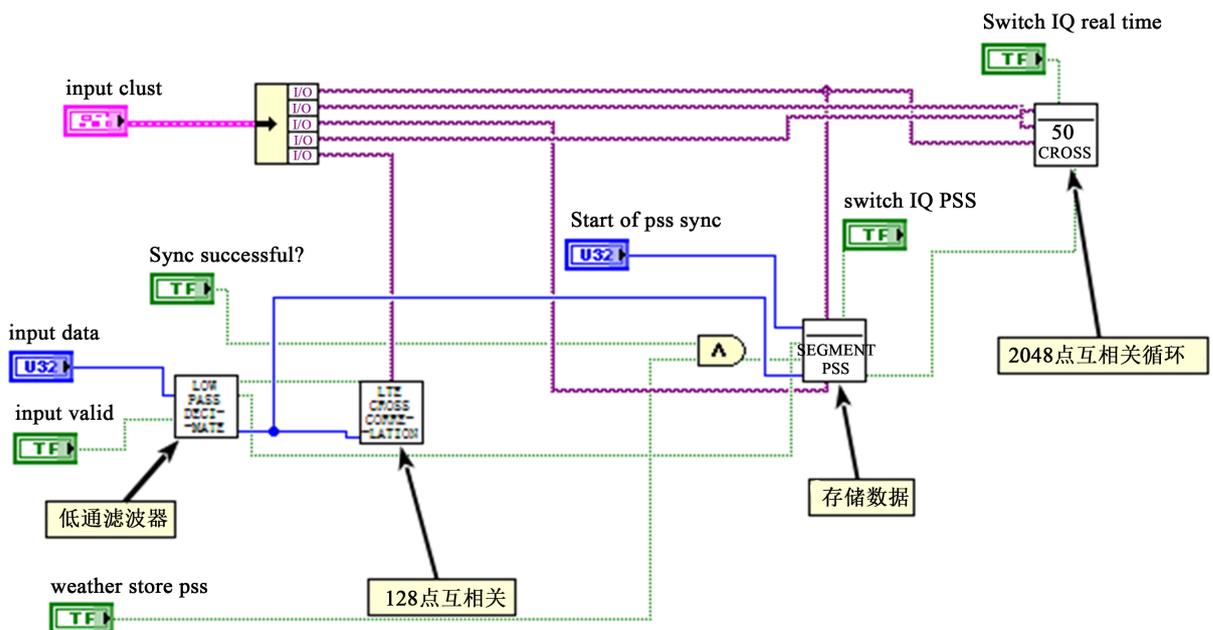


Figure 7. Low pass sampling, cross correlation and other processes
图 7. 低通降采样、互相关等流程

带外信号, 降低干扰, 降采样的实现为抽取器, 每 16 个采样点, 抽取一个点, 输出到下一链路, 同时该 IP 核可以实现滑动相乘相加运算, 可以完美实现 128 点互相关器, 实际发现若相关窗为 512 时, 编译速度慢, 且不能实现 2048 点互相关器, 因为相关窗极大, FPGA 资源本身已被耗尽。因而本论文同步算法不能直接使用 2048 点 FIR 互相关器实现, 需要寻找复杂度低的实现方式。

图 8 主要采用 Xilinx FIR IP 核为实现基础的低通滤波器模块, 其中输入数据为复数的基带时域信号, 实部虚部分别为 16 bits, 打包成一个 U32, 把该数据输入 Xilinx FIR 低通滤波器, 该滤波器的带宽为 1 MHz, Xilinx FIR IP 核的具体实现过程在文献[8]中有详细的步骤, 滤除 PSS 信号之外的相对于同步过程无用的信号, 由于滤波过程中会产生数据范围的增大, 实部、虚部都变成了 40 bits, 因而需要对输出的数据进行数据类型转换, 防止数据位数过大, 在图 8 中的虚线框中, 把输出的 40 bits 实部虚部分别取前 16 bits, 组成一个新的 U32, 输入到本地 FIFO (First In First Out, 先入先出缓冲器)中。之后从本地 FIFO 中读取滤波之后的数据, 进行后续互相关等过程。这里采用本地 FIFO 起到缓冲数据的作用。

图 9 为 Xilinx FIR IP 核实现的 128 点互相关器。图中的输入信号为 data_in, 该信号是图 8 中滤波器的输出 1/16 降采样信号(由于 1/16 抽取器较为简单, 虽然包含在流程中但是省略之), 即为时域基带信号。图中位于上方的互相关器内部系数为 1/16 倍抽取后的本地 PSS 时域信号的实部, 而对应的下方互相关器内部系数为 1/16 倍抽取后的本地 PSS 时域信号的虚部, 二者同时对输入基带信号进行相关计算, 这里两个实数相关器分别对输入信号的实部和虚部进行相关, 合并这 4 个输出, 得到一个复数相关器, 如图中红色虚线框部分, 具体 128 点互相关原理如下公式(3)所示:

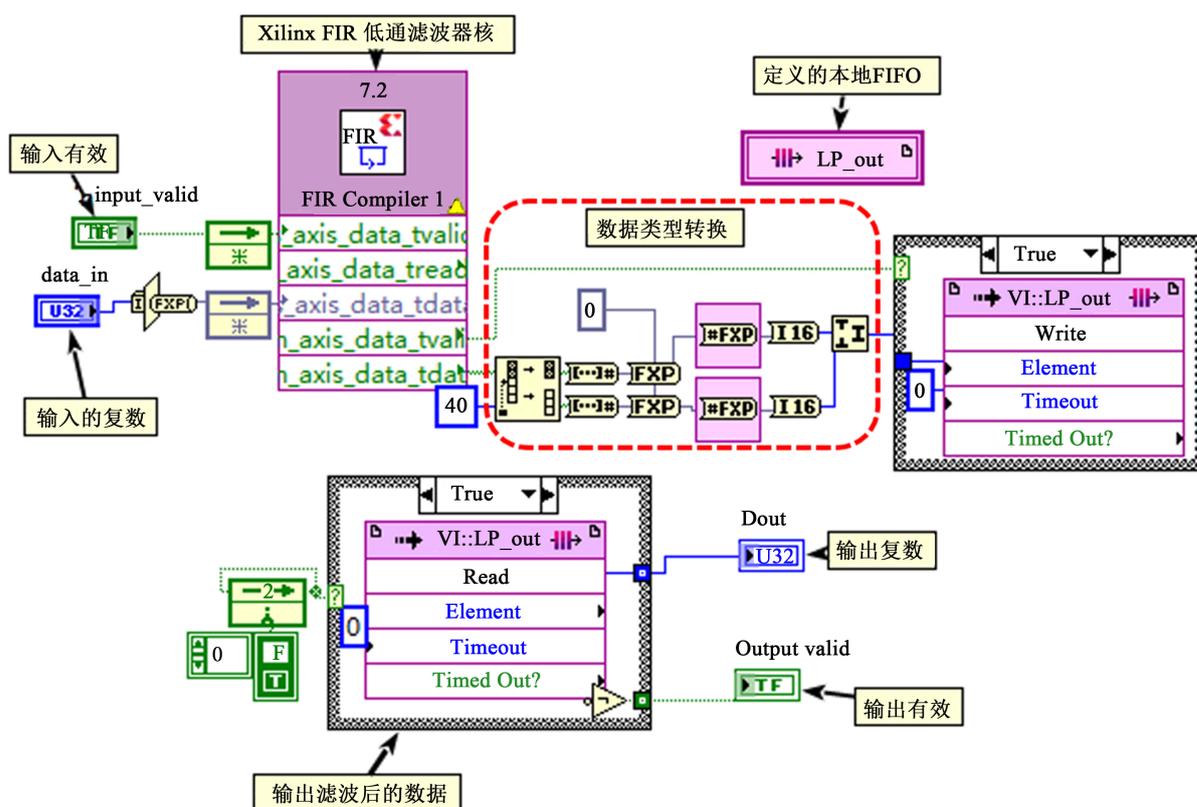


Figure 8. Low pass filter module
图 8. 低通滤波器模块

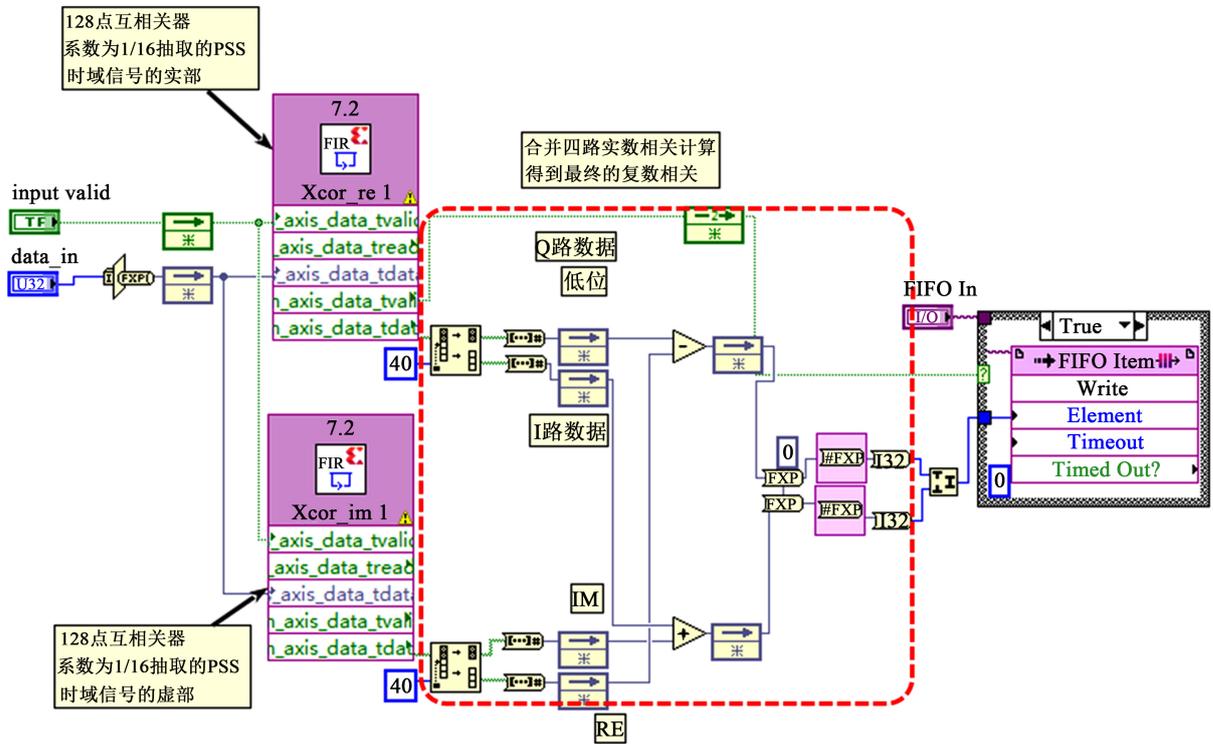


Figure 9. 128 point cross-correlation
图 9. 128 点互相关器

$$\begin{aligned}
 C_n &= \sum_{i=1}^{128} (re_{local,i} + i * im_{local,i}) (re_{time,n+i} + i * im_{time,n+i})^* \\
 &= \sum_{i=1}^{128} [(re_{local,i} * re_{time,n+i} + im_{local,i} * im_{time,n+i}) + i(-re_{local,i} * im_{time,n+i} + im_{local,i} * re_{time,n+i})] \\
 &= \sum_{i=1}^{128} re_{local,i} * re_{time,n+i} - i * \sum_{i=1}^{128} re_{local,i} * im_{time,n+i} + i * \sum_{i=1}^{128} im_{local,i} * re_{time,n+i} + \sum_{i=1}^{128} im_{local,i} * im_{time,n+i}
 \end{aligned} \tag{3}$$

公式(3)中 C_n 为互相关计算的最终相关数值, $re_{local,i}$ 表示本地 PSS 序列第 i 点的实部, $im_{local,i}$ 表示本地 PSS 序列第 i 点的虚部, $re_{time,n+i}$ 表示第 $n+i$ 点的时域 PSS 信号的实部, $im_{time,n+i}$ 表示第 $n+i$ 点的时域 PSS 信号的虚部, 这些信号都是经过 1/16 倍降采样的, 公式(3)中字体稍大的 i 为虚数单位, 式中第三行有 4 个实数相关计算, 需要采用 4 个实数相关器来实现, 注意到前两个相关是用 $re_{local,i}$, 分别与基带时域信号进行的相关计算, 因而可以把 Xilinx FIR 相关器配置成双路输入, 把基带信号的实部和虚部同时输入到该相关器中, 输出的数据只需要把虚部添加一个负号即可。同理, 对于式中第三行后两个相关计算也可以集成到一个双路输入的实数相关器中, 这样降低了硬件资源的开销。

图 10 为 2048 点互相关器的实现, 互相关器的数据来源是两块 memory, 一块存储的数据是固定的 2048 点 PSS 时域信号, 另一块存储的是实时输入的低通滤波之后的基带数据, 来自于图 7 中的数据存储过程。在程序的开始, 第一层循环的循环次数为 50, 含义为 2048 点互相关器的滑动窗共计移动 50 次(实际中不需要一定取上文中提到的 17 次, 50 次可以在更大范围内寻找 PSS, 健壮性更佳), 第二层循环的循环次数为 2048, 代表滑动窗的大小为 2048, 即每一次互相关共有 2048 对复数进行相乘相加, 同时, 这两次循环中的计数器可用来生成两块 memory 的地址。可以看到, 这里进行了 2 次循环, 每一个数据点都需要读取多次, 需要采用 memory 做存储器, 可以重复读取数据, 而不方便采用 FIFO 结构。从两块 memory 中取出的数据进行复数相乘, 这里只需要使用一个复数乘法器即可完成, 在乘法器之后连接两个

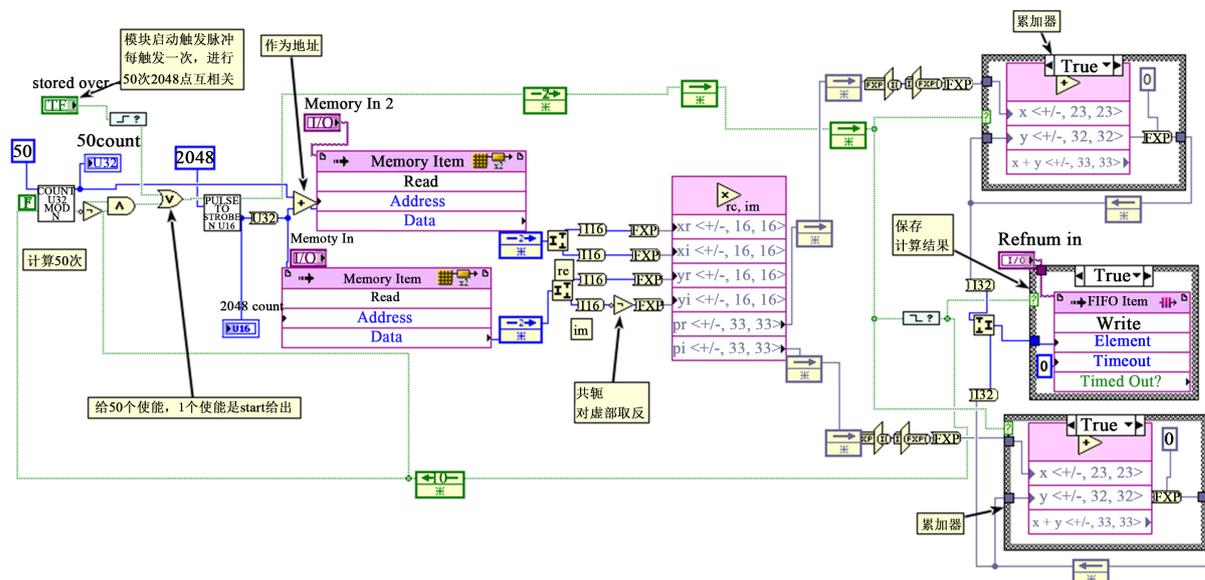


Figure 10. The realization of 2048 cross-correlation

图 10. 2048 点互相关器的实现

累加器, 分别对乘法器输出复数的实部和虚部进行累加, 每做完一次 2048 点累加需要把累加和存入 FIFO 中, 后续步骤为峰值搜索。

现在计算 2048 点互相关的复杂度。一个 10 ms 的无线帧共计进行一次图 10 的过程, 共计 $50 \times 2048 = 102,400$ 个时钟周期, 在 FPGA 实现中, 该循环使用 200 MHz 时钟, 因而 10 ms 有 2,000,000 个时钟周期, 所以工作周期占时钟总周期的 5.12%, 图 10 过程完全可以在一个无线帧的时间内执行完成, 可以做到实时计算, 因而精同步过程是实时进行的。

图 11 为对 2048 点互相关器的输出信号进行峰值检测过程。首先从 FIFO 中读取互相关后的数据, 并对该数值进行求模运算, 图中的求模计算模块是 LabVIEW FPGA 中集成的模块, 输入复数的实部、虚部以及控制信号(标志输入信号是否有效), 经过运算, 延迟一定的周期, 即可得到复数的幅值、相位以及控制信号(标志输出信号是否有效)。得到模值之后, 对模值进行峰值搜索, 搜索的过程为: 比较当前输入的模值与前一次模值的大小, 若当前输入值大, 则保存当前值及其 index (输入值的位置, 范围是 0 到 49, 下同), 若前一次模值大, 则不做操作, 此时寄存器中依然存储的是上一次模值和对应的 index, 该过程每执行 50 次为一次循环, 输出峰值的 index 和控制信号(表示输出是否有效)。同时每进行完 50 次比较之后, 对寄存器初始化: 记录输入模值位置的 index 置 0、保存模值的寄存器置 0、保存模值 index 的寄存器置 0, 从而可以继续下一次循环。

本模块输出峰值的 index, 范围是 0 到 49, 表示图 10 中的 50 次 2048 点互相关器输出值的峰值的具体位置, 转换偏置为 -25 到 25, 与粗同步位置相加即可得到精同步位置。

图 12 为峰值搜索与有效性判定模块, 其中粗同步模块的峰值搜索与图 11 所示峰值搜索模块类似, 有效性判定即判定粗同步结果的有效性, 首先比较峰值的幅度是否足够大, 再比较峰值的位置与前一个无线帧的位置是否接近, 因为每一个无线帧的同步位置基本不会变化的(若基站与用户的时钟有偏差, 则同步位置会缓慢变化)。流程首先寻找 128 点互相关后的峰值, 并把得到的峰值和触发脉冲发送给有效性判定模块, 启动对粗同步结果的有效性判定, 若通过有效性判定模块, 则对 Sync Successful? 信号置 1, 表明粗同步成功, 同时输出粗同步位置。另一方面, 对 2048 点互相关输出结果进行峰值搜索, 得到 PSS 信号与粗同步位置精确的偏置, 并把该偏置与粗同步位置相加, 得到最终的精同步位置 Start of sync accurate 信号。至此,

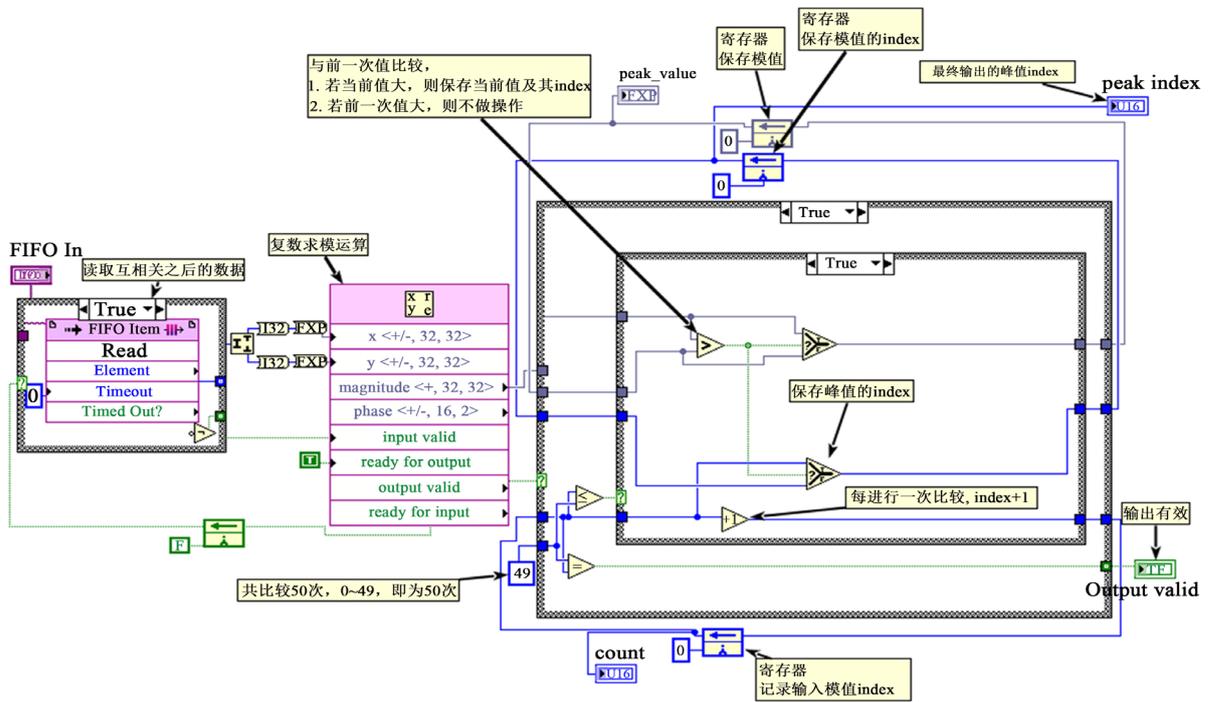


Figure 11. 50 times peak search module
图 11. 50 次峰值搜索模块

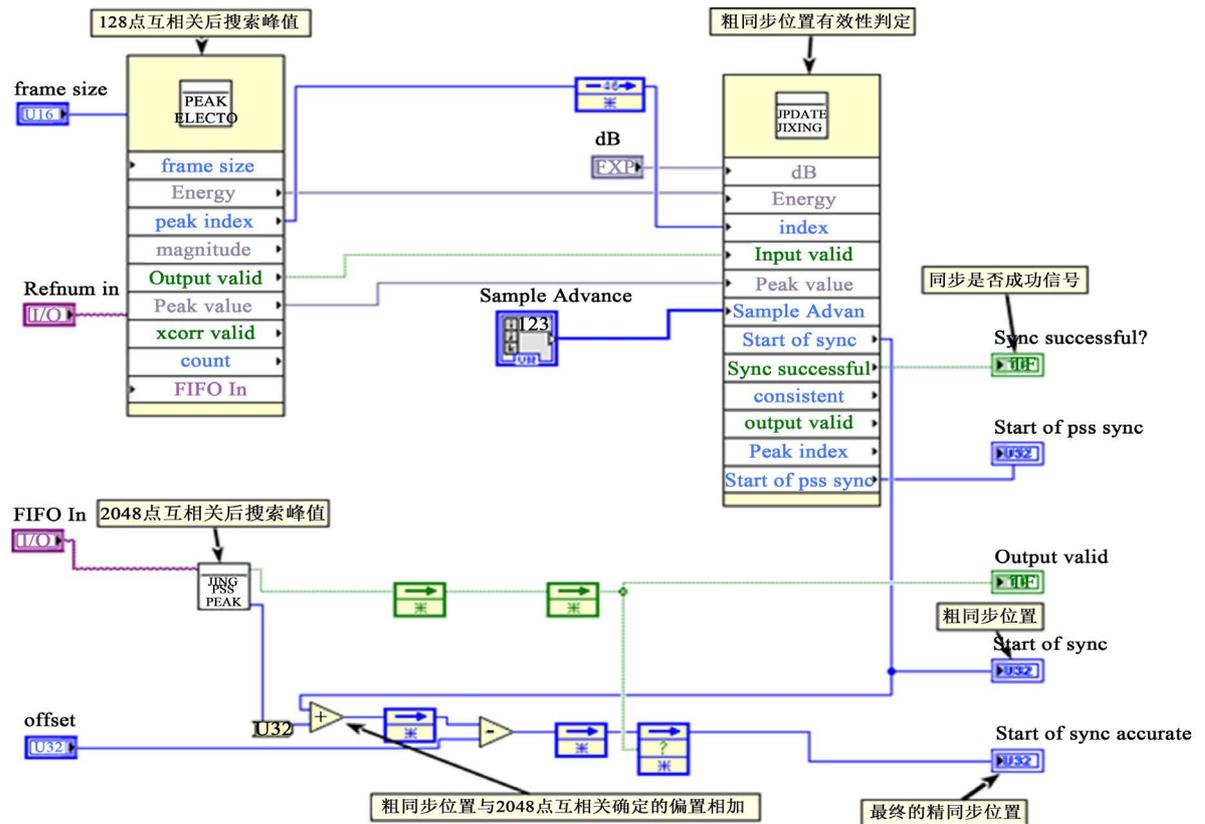


Figure 12. Peak search and validity determination of synchronization results
图 12. 峰值搜索与同步结果有效性判定

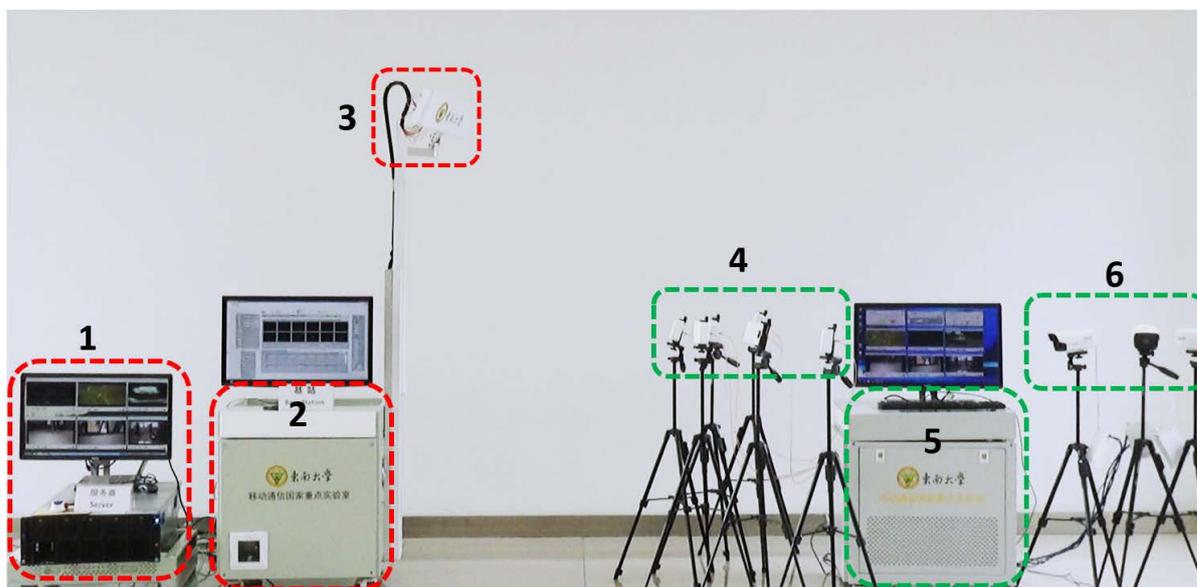


Figure 13. The successful operation of this algorithm in the MIMO-OFDM system

图 13. 本算法在 MIMO-OFDM 系统中的成功运行

同步过程完成, 输出 Sync Successful? 信号和 Start of sync accurate 信号。

4. 应用案例

本文所述低复杂高精度无线帧同步算法已在 MIMO-OFDM 系统[9]中成功应用, 并展现出了优良的效果, 系统同步稳定, 精度高, 为 OFDM 系统后续的信号处理提供了必要条件。

图 13 为本算法在 MIMO-OFDM 系统中的成功运行效果图, 在图中用虚线框标注了各个模块, 含义如下: 模块 1, 为基站端的信号处理服务器, 负责信道估计和信道均衡; 模块 2, 为软件无线电 USRP RIO 模块组, 负责基带数据的采样, 本低复杂高精度算法即在该模块上运行, 用于捕获天线接收信号中的无线帧同步信号, 确定其精确位置; 模块 3, 为基站端的天线阵列, 包括 16 根接收天线; 模块 4, 为用户端的发射天线, 共计 6 根发射天线, 代表 6 个单用户; 模块 5, 为用户端的软件无线电 USRP RIO [10] 模块组, 负责产生用户基带信号, 并发射到空口; 模块 6, 为摄像头模块, 用户获取视频信号, 并打包发送到软件无线电模块, 发送到基站端。

图 13 为 6×16 MIMO-OFDM 系统, 用户端发射信号符合 LTE 帧格式(实际做了细节修改, 参见图 1), 6 个用户同步地发送包含 PSS 信号的 OFDM 信号, 基站端每根天线都接收到基带数据, 并进行本论文所述低复杂高精度帧同步算法的处理, 找到精确的同步位置, 从而基站端可以进行后续信号处理流程。从模块 1 中可以看到显示器中播放着视频, 此视频即为用户端发送的视频, 表明该系统能够稳定正确运行。

5. 结论

本文提出了一种低复杂度的高精度 OFDM 系统无线帧同步算法, 并使用 LabVIEW FPGA 语言实现该算法, 该算法在粗同步的基础上仅添加一个复数乘法器和两个累加器等逻辑模块, 复杂度极低, 精确度高, 并在实际的 MIMO-OFDM 系统中部署了该算法, 运行效果良好, 可以扩展到更多的 OFDM 系统中。

参考文献

- [1] Yong Soo Cho, Jaekwon Kim, 等. MIMO-OFDM 无线通信技术及 MATLAB 实现[M]. 北京: 电子工业出版社,

- 2014: 136-143.
- [2] 赵邵刚, 李岳梦. LTE-Advanced 宽带移动通信系统[M]. 北京: 人民邮电出版社, 2012.
 - [3] Stefania Sesia, Issam Toufik, Matthew Baker. LTE/LTE-Advanced-UMTS 长期演进理论与实践[M]. 北京: 人民邮电出版社, 2012: 115-118.
 - [4] Dahlman, E., Parkvall, S., Skold, J., *et al.* (2010) 3G Evolution: HSPA and LTE for Mobile Broadband. Academic Press, Cambridge.
 - [5] 阮奇楨. 我和 LabVIEW: 一个 NI 工程师的十年编程经验[M]. 北京: 北京航空航天大学出版社, 2012.
 - [6] 严雨. LabVIEW 入门与实战开发 100 例(第 3 版) [M]. 北京: 电子工业出版社, 2017.
 - [7] Erik, L. (2014) 5G Massive MIMO Testbed: From Theory to Reality. <http://www.ni.com/white-paper/52382/zhs/>
 - [8] 刘东华. Xilinx 系列 FPGA 芯片 IP 核详解[M]. 北京: 电子工业出版社, 2013: 127-156.
 - [9] Yang, X., Huang, Z.C., Han, B., Zhang, S., Wen, C.K., Gao, F. and Jin, S. (2017) RaPro: A Novel 5G Rapid Prototyping System Architecture. *IEEE Wireless Communications Letters*, **6**, 362-365. <https://doi.org/10.1109/LWC.2017.2692780>
 - [10] National Instruments: NI USRP RIO. <http://sine.ni.com/nips/cds/view/p/lang/zhs/nid/212991>

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2163-3983, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: hjwc@hanspub.org