

基于 SoC 的导航接收机闭环跟踪环路设计与实现

薛志芹¹, 刘坤¹, 李立广²

(1. 航天恒星科技有限公司, 北京 100086;

2. 空装驻北京地区第七军事代表室, 北京 100086)

摘 要: 在传统基于 FPGA+DSP 架构的导航接收机中, 跟踪模块处于开环处理模式, 存在实时性和可靠性差的问题, 同时现场可编程逻辑门阵列 (FPGA) 和数字信号处理 (DSP) 之间大量的数据通信导致了 IO 资源和功耗的增加. 基于片上系统 (SoC) 架构提出了一种卫星导航接收机的闭环跟踪环路方案, 整个跟踪处理过程在 FPGA 内实现了完整的闭环处理, 有效地解决了开环跟踪存在的问题, 极大地减少了 FPGA 与 CPU 之间的数据通信量. 此外, 所有跟踪通道通过时分复用的方式共用一个跟踪环路处理模块, 有效节约了硬件资源, 降低了成本, 为小型化、低功耗卫星导航芯片的设计与发展奠定了基础.

关键词: 闭环跟踪; SoC 架构; 卫星导航接收机; 跟踪环路; 时分复用

中图分类号: P228.4; TN967.1

文献标志码: A

文章编号: 1008-9268(2021)03-0072-06

0 引 言

随着北斗卫星导航系统 (BDS) 全球组网的实现, 全球卫星导航系统 (GNSS) 目前已经发展为由 GPS、BDS、GLONASS、Galileo 四大全球系统和各地区的区域系统以及相应的天基增强卫星导航系统组成的庞大复杂的全球定位卫星网络^[1]. 随着 GNSS 的不断发展, 卫星导航在航空航天、军事领域和日常生活中的应用越来越多. 卫星导航接收机作为卫星导航系统的重要组成部分, 对其体积、功耗、性能和集成度等指标的要求日益苛刻^[2].

传统的卫星导航接收机采用 FPGA+DSP 的设计方式, 设计复杂, 整机体积和功耗较大、成本高、生产周期长^[3]. 在传统基于 FPGA+DSP 架构的卫星导航接收机中, 捕获部分和跟踪相关器部分在现场可编程逻辑门阵列 (FPGA) 中实现, 利用 FPGA 的高速并行运算能力实现多通道相关的并行处理. 由于数字信号处理 (DSP) 非常适合做过程控制和复杂的浮点运算^[4], 因此跟踪部分的通道调度、环路鉴别器和环路滤波器等均在 DSP 中实现. 这种 FPGA+DSP 的架构设计, 虽然充分发挥了 FPGA 和 DSP 各自在数据处理方面的显著优势, 但在这种架构下, FPGA 负责完成

多个通道的相关处理后, 需要将各个通道的 6 路累加量输出给 DSP, DSP 在接收到累加量后, 根据当前累加量完成载波数字振荡器 (NCO) 和码 NCO 控制字的更新, 之后再通过总线将控制字写回到 FPGA, 整个跟踪模块处于一个开环处理过程中, 因此不可避免的会存在频率控制字更新不及时、实时性差、可靠性差的问题. 此外, 这种架构下, FPGA 和 DSP 之间存在大量的数据通信, 增加了 IO 资源及功耗.

近年来, 随着集成电路制造技术的蓬勃发展, 具有速度快、集成度高、功耗低等众多优点的片上系统 (SoC) 被广泛应用^[5]. SoC 将整个嵌入式系统集成在一块芯片上, 因此基于 SoC 的卫星导航接收机在低成本、低功耗和小型化方面具有其他系统无法比拟的优势^[2, 6-7]. 本文基于 SoC 架构, 提出了一种卫星导航接收机的闭环跟踪环路设计方案. 整个跟踪过程, 包括相关器部分、通道选择、鉴相器和环路滤波器, 均在 FPGA 中实现, 实现了跟踪模块的闭环处理, 确保了跟踪的稳定性与实时性. 为有效节约硬件资源, 降低成本, 本文设计方案中所有跟踪通道通过时分复用的方式共用一个跟踪环路处理模块, 为小型化、低功耗卫星导航芯片的设计与发展奠定了基础, 最后给出了基于 modelsim 的仿真结果, 证明了该闭环跟踪

收稿日期: 2020-12-14

资助项目: 国家自然科学基金 (62073044)

通信作者: 薛志芹 E-mail: xuezhiqin_celery@163.com

环路的正确性和稳定性.

1 基于 SoC 的整体架构设计

本文系统中使用到的 SoC 是基于 ARM Cortex-A9

和 FPGA 相结合的架构, ARM 芯片具有成本低、功耗小、性能高、移植性好等优势,非常适合作为卫星导航接收机的核心处理芯片. SoC 总体架构如图 1 所示.

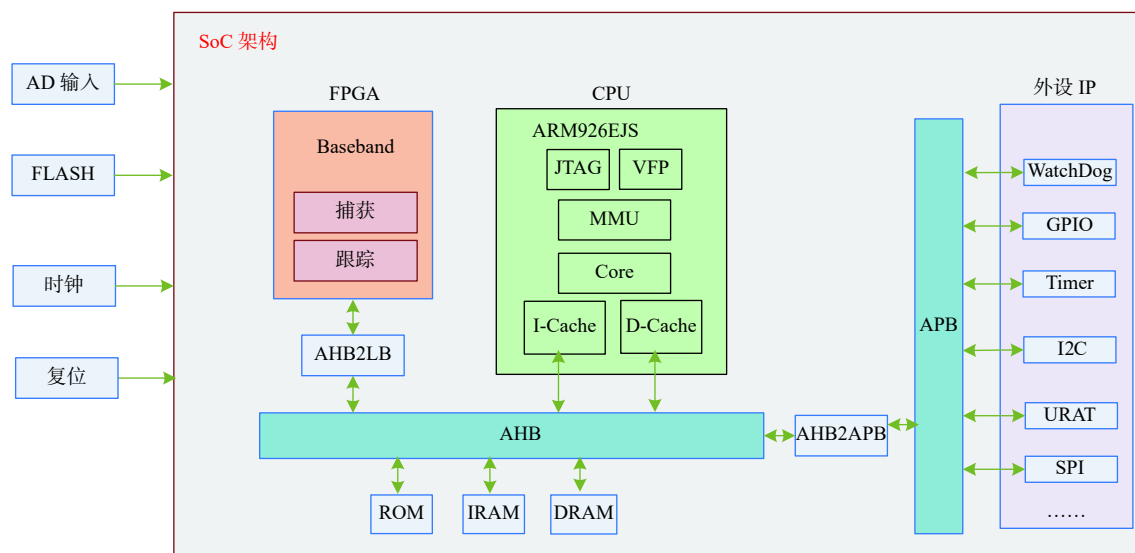


图 1 SoC 总体架构

整个 SoC 主要包括 ARM9 CPU 模块、基带 Baseband 模块、AMBA 总线系统及其他外设 IP 模块组成. CPU 模块主要用于定位解算, 以及完成系统的主控调度和对基带模块的控制. Baseband 模块为导航基带部分, 实现对卫星导航信号的捕获、跟踪 (包括相关器和环路) 等处理, 获取卫星观测量值, 并输出电文, 供解算使用. 本文采用的 CPU 是 ARM926EJS, 工作频率最大支持 200 MHz, 具有支持双精度的浮点运算的协同处理器 VFP. ARM9 具有良好的性能, 同时也因为具有较好的价格优势, 因此我们选择了该 CPU 进行了 SoC 的设计.

2 闭环跟踪环路设计方案

卫星导航信号多通道跟踪环路是载波跟踪环和码跟踪环的有机组合, 信号跟踪的目的是从码跟踪环中得到伪距测量值, 载波跟踪环剥离接收信号中的载波, 能够得到较为准确的载波多普勒频率测量值, 及时反映接收机在其与卫星连线方向上的相对运动速度, 也能够替码跟踪环消除用户动态性和接收机基准频率漂移等动态应力作用, 起到辅助码跟踪环的作用. 本文设计的卫星导航接收机闭环跟踪环路方案如图 2 所示. 卫星信号经过射频前端处理后获得中频采样数据, 中频采样数据首先进入 FPGA 中的捕获模块进行捕获处理, 捕获模块在捕获到卫星信号后, 通过

捕获转跟踪, 将捕获的卫星信息送入到 FPGA 中的跟踪模块进行处理^[8]. 捕获模块和跟踪模块同属于基带信号处理, 之后通过总线与 ARM 通信, 由 ARM 完成导航解算.

文章采用硬件环路的方式实现跟踪, 即环路计算不再在 CPU 中运算, 而是和相关器部分一样在 FPGA 中实现, 大大降低了对 CPU 的要求, 也能使 CPU 工作在较低频率下, 对于解算率要求不高的产品, 甚至可以工作在 10 MHz 的频率, 有效地降低了芯片的功耗; 另外, 跟踪环路在实现时, 环路参数可通过 CPU 进行配置、切换, 依然保证了环路的灵活性.

由图 2 可知, 整个跟踪模块采用并行通道相关器处理与串行通用环路处理相结合的方式, 完成整个跟踪过程的闭环处理. 卫星信息由捕获转入跟踪, 首先进入通道选择模块, 由通道选择模块选取对应的空闲跟踪通道对本次捕获的卫星进行跟踪. 跟踪通道相关器部分采用并行处理方式, 每个跟踪通道相关器均有各自的码发生器、载波 NCO、码 NCO、相关器等部分. 跟踪通道完成相关运算后, 会输出各自的 6 路累加量 (超前支路累加量 I_e 、 Q_e , 即时支路累加量 I_p 、 Q_p 和滞后支路累加量 I_l 、 Q_l) 进入到通道仲裁模块. N 个跟踪通道分时复用进行跟踪环路的处理, 由通道仲裁模块对 N 个跟踪通道的信息进行仲裁, 确定当前串行通用跟踪环路处理的跟踪通道号. 通用环路处理包括跟踪环路参数读取、环路控制信号产生、相干

累加、比特同步、载噪比 (CNR) 计算、鉴相、环路滤波等模块, 跟踪环路处理完成后, 获得对应通道的码

环滤波结果和载波环滤波结果, 反馈到跟踪通道部分, 形成完整的跟踪闭环处理流程。

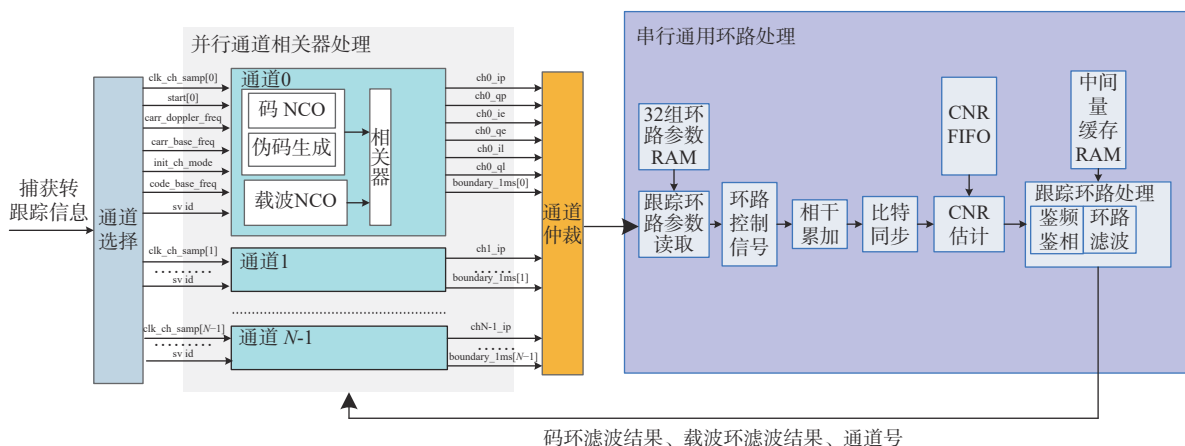


图 2 导航接收机闭环跟踪环路方案

2.1 并行通道相关器处理

卫星导航信号多通道跟踪并行相关器部分结构

如图 3 所示, 每个跟踪通道均具有相同的结构, 且各自相互独立。

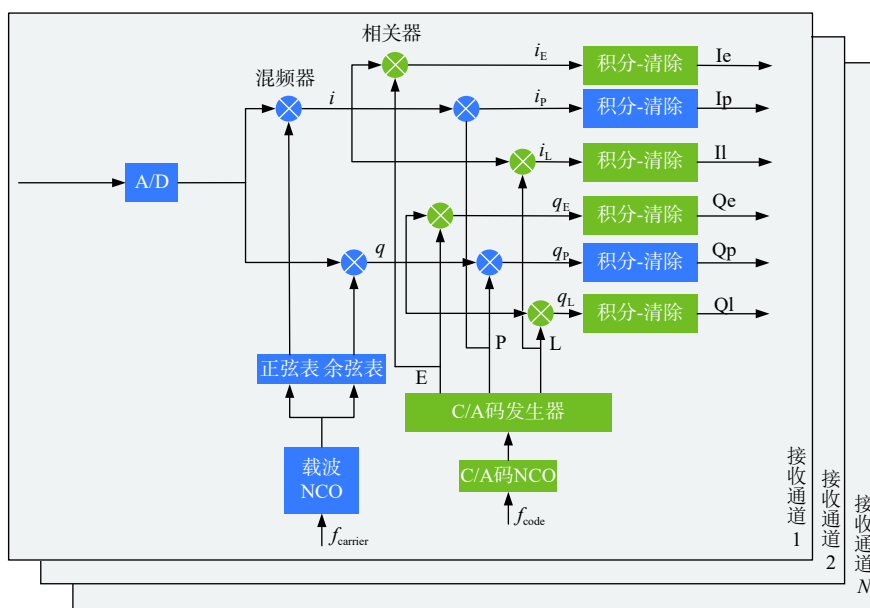


图 3 导航接收机相关器模块

输入的数字中频信号首先与载波环复制的正余弦载波混频相乘, 得到的 IQ 支路的混频信号又分别与码跟踪环复制的超前、即时、滞后三份伪码做相关运算。相关结果经积分-清除器后分别输出相干积分值。即时支路上的相干积分值输入至载波跟踪环鉴别器, 其他两条相关支路输出的相干积分值则作为码跟踪环鉴别器的输入。

2.2 跟踪通道仲裁

N 个跟踪通道在相关累加得到 1 ms 累加量后, 发出通道请求信号, 请求跟踪环路处理。通道仲裁模

块通过轮询的方式对各个通道进行环路处理, 仲裁选定通道进行后续的鉴频、鉴相、环路滤波、CNR 计算等操作。由于导航信号跟踪过程中, 相关累加时长为 1 ms, 因此从通道请求信号发出到通道处理完成必须控制在 1 ms 内, 在 1 ms 时间内完成载波频率控制字和码频率控制字的更新, 从而保证跟踪通道的稳定运行。仲裁模块需确保 1 ms 内轮询完所有的跟踪通道, 确保所有通道无遗漏。

通道仲裁模块的时序图如图 4 所示。这里假定并行跟踪通道个数为 64, 即 $N=64$ 。每个通道分配固定

的环路处理时间,假定每个通道的处理时间为 $P=320$ 个时钟周期 (后文会给出处理时间的确定方法)。64 个跟踪通道从 0 通道开始,依次轮询处理。

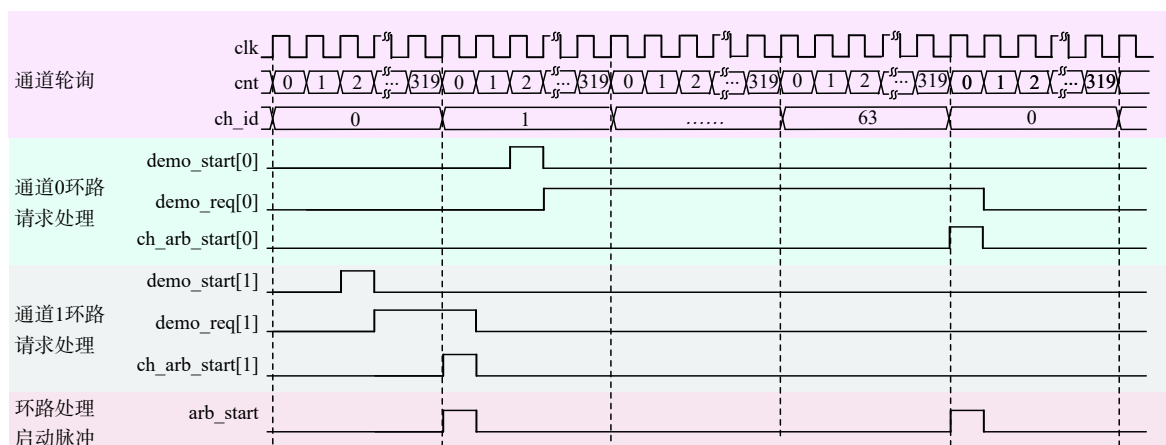


图4 通道仲裁时序图

某一时刻当通道 0 完成 1 ms 相关运算,发出跟踪环路请求脉冲 $\text{demo_start}[0]$,在该请求脉冲作用下, $\text{demo_req}[0]$ 信号拉高,等待通道 0 的处理时隙。在通道 0 处理时隙下,产生通道 0 的环路启动脉冲 $\text{ch_arb_start}[0]$ 。图 4 中同时给出了通道 1 的环路请求处理时序,这里不再赘述。所有通道的环路启动脉冲合并产生总的环路处理启动脉冲 arb_start 。在环路处理启动脉冲 arb_start 作用下,比特同步、CNR 计算、鉴频鉴相、滤波等一系列环路处理被触发启动。

接收机相关运算时间为 1 ms,载波和码频率控制字更新时间也为 1 ms,因此从通道发出环路处理请求,到跟踪环路将载波跟踪环和码跟踪环滤波结果反馈给跟踪通道的最长时间也为 1 ms,否则会导致载波和码频率控制字更新不及时,导致环路失锁。而 N 个通道通过轮询的方式依次分配环路处理时间,极限情况下是当轮询到某一通道时该通道发出了环路处理请求,此时需要等到下一次轮询到该通道时,才会对本次发出的环路处理请求进行响应,如图 5 所示。某个通道的处理时间固定为 P 个时钟周期,因此极限情况下环路处理请求的响应时间最长为 $(N+1) \cdot P$ 个时钟周期。为了保证跟踪环路的稳定性及频率控制字更新的及时性,必须保证 $(N+1) \cdot P < 1 \text{ ms}$ 。以 58 MHz 处理时钟为例,需要保证 $(N+1) \cdot P < 58\,000 \text{ ms}$,也就是说假设 FPGA 实现时,每个通道的固定处理时间 $P=320$ 个时钟周期,则该系统最多支持 $N=180$ 个并行跟踪通道。

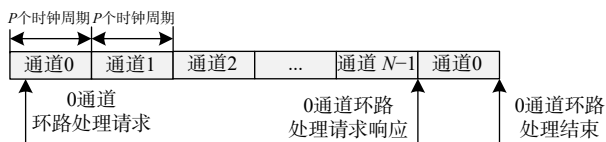


图5 极限情况下环路处理请求响应示意图

2.3 通用串行环路处理

为实现卫星信号跟踪的闭环处理,本文跟踪环路处理部分在 FPGA 中实现。跟踪环路实现时,跟踪环路采用二阶 FLL 辅助三阶 PLL 形式^[9],环路滤波所采用的环路参数由 CPU 通过总线写入到 FPGA 内部的环路参数 ARM 中,可通过 CPU 进行配置、切换,从而保证跟踪环路的灵活性。

整个串行环路处理包括跟踪环路参数读取、环路控制信号产生、相干累加、比特同步处理、CNR 估计、鉴频鉴相、环路滤波等过程,如图 2 所示。图 6 给出了通用串行环路处理状态跳转示意图以及每个状态处理所需要的时钟周期。跟踪环路未启动时,处理处于“空闲”状态,直到有通道发出环路处理请求,等待轮询到该通道处理时隙时,跟踪环路响应该通道的环路处理请求。在环路处理启动脉冲作用下,环路首先耗用 12 个时钟周期从 ARM 中读取跟踪环路参数,包括 DLL、FLL(锁频环)和 PLL(锁相环)各自的相干积分时间和环路滤波带宽,接着环路进入“控制信号产生”状态,产生后续环路处理所需的控制信号,耗时 9 个时钟周期。之后,进行相干累加操作,耗时 25 个时钟周期。若此时配置比特同步使能,则进入比特同步状态^[10],否则直接进入 CNR 计算^[11],比特同步和 CNR 计算分别耗时 88 个和 24 个时钟周期,最后环路进入鉴频鉴相和二阶 FLL 辅助三阶 PLL 滤波处理,分别耗时 129 个和 11 个时钟周期,至此整个跟踪环路处理流程结束,环路进入“空闲”状态,等待下一次跟踪环路启动。

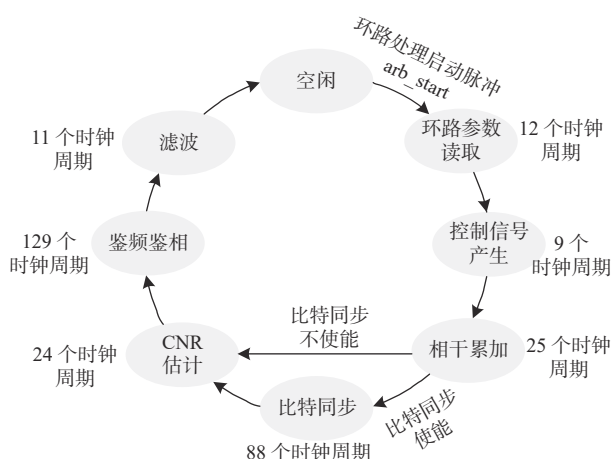


图 6 通用串行环路处理状态跳转示意图

由上面的分析可以看到, 整个跟踪过程共耗时 300 个时钟周期, 留有一定余量, 因此将每个跟踪通道的环路处理时长固定值取为 $P=320$ 个时钟周期。

本文提出的卫星导航接收机的闭环跟踪环路方案, 通过多个相关器通道分时复用跟踪环路处理的方式, 实现了跟踪环路的闭环处理。在该方案下, 每个跟踪通道的环路处理时长固定为 320 个时钟周期, 在 58 MHz 采样率下, 可支持多达 180 个并行跟踪通

道, 有效节约了硬件资源, 降低了成本。

3 FPGA 仿真结果

基于上述给出的卫星导航接收机闭环跟踪环路设计方案, 通过 FPGA 进行实现, 并进行仿真验证。modelsim 仿真下的仿真结果如图 7、图 8 所示。图 7 为 -132 dBm 信号功率下累加量的仿真结果, 其中 acc_1ms_ip 为 I_p 支路的累加量结果, acc_1ms_qp 为 Q_p 支路的累加量结果。图 8 为环路收敛过程中载波环和码环的滤波结果, 其中 $loop_carr_lpf_data$ 为载波环滤波结果, $loop_code_lpf_data$ 为码环滤波结果。由图 7、图 8 可知, 环路滤波结果逐渐趋于稳定, 环路收敛良好, 环路控制字在每 1 ms 都得到了正确的调整, 环路更新及时、稳定, 满足设计要求。图 9 给出了跟踪过程中 CNR 估计结果, 其中黑色实线、蓝色实线和红色实线分别为 100 ms、1 s 和 4 s 的 CNR 估计结果, 其方差分别为 0.233 6 dB、0.033 7 dB 和 0.011 2 dB。可以看出, 100 ms CNR 估计由于平滑时间较短, CNR 估计结果波动较大, 而 4 s 的 CNR 估计结果由于进行了足够长时间的平滑, CNR 估计结果稳定在 40 dB 左右。稳定的 CNR 估计结果, 同样证明了跟踪环路设计的稳定性和有效性。

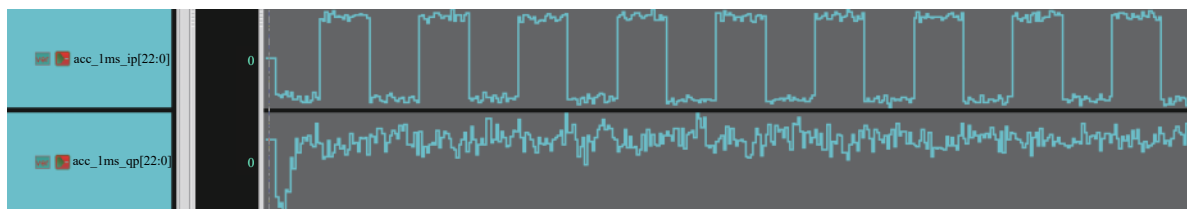


图 7 累加量仿真结果

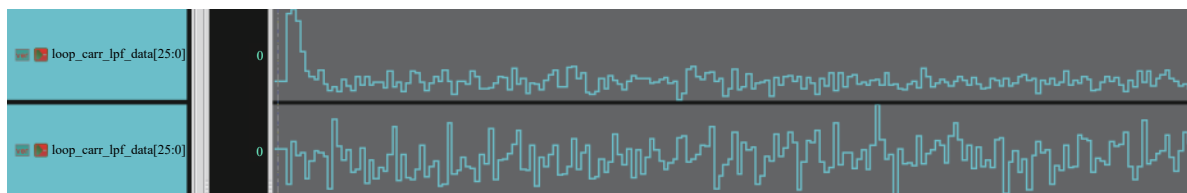


图 8 环路滤波仿真结果

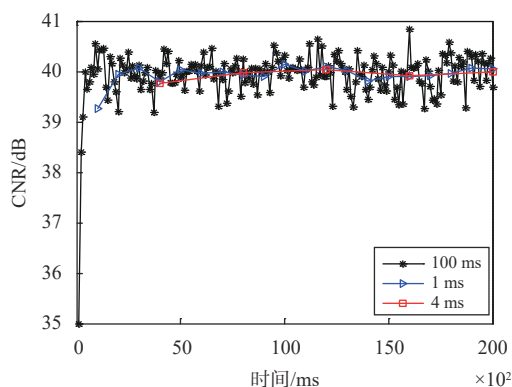


图 9 CNR 估计结果

为了验证闭环跟踪环路设计方案在动态环境下的稳定性与适应性, 图 10 和图 11 给出了动态环境下的仿真结果。仿真过程中, 设定最大加速度为 10 g、最大加速度为 20 g、最大速度为 70 g, 信号功率为 -132 dBm, 仿真时长为 12.6 s。图 10 为动态仿真环境下的累加量结果, 红色实线和蓝色实线分别为累加量 I_p 和累加量 Q_p 。图 11(a) 为码环滤波结果, 图 11(b) 为载波环滤波结果, 其中图 11(b) 中的红色实线为载波环滤波结果的理论值, 蓝色虚线载波环滤波的仿真值, 可以看出在动态环境下, 环路跟踪稳定, 滤波结

果正确,环路收敛迅速、稳定,从而证明了本文提出的环跟踪环路设计方案的有效性和实用性。

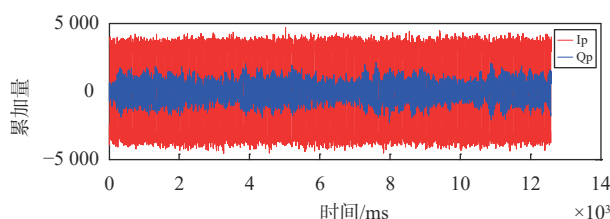


图 10 动态仿真环境下的累加量结果

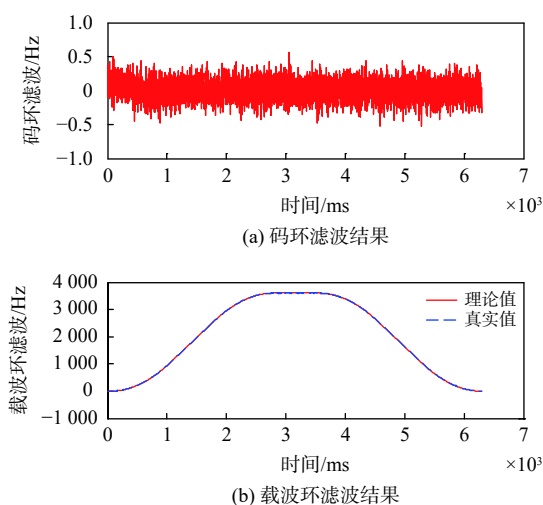


图 11 动态仿真环境下的环路滤波器结果

4 结 论

随着 GNSS 的不断发展,导航定位在日常生活、航空航天、科研等各个方面的应用需求不断提升,小型化、低功耗、高集成的卫星导航接收机成为目前重要的导航技术发展方向。本文给出了一种基于 SoC 的卫星导航接收机闭环跟踪环路设计方案,通过高度集成的 SoC 架构代替了传统的 FPGA+DSP 的架构,在 FPGA 中实现了导航信号跟踪的闭环处理,确保了跟踪的实时性和稳定性。在本文的 SoC 架构中,具有快速运算能力的 ARM9 处理器与具有强大 DSP 能力的 FPGA 相结合的同时,所有跟踪通道时分复用共享一个跟踪环路处理模块,有效地节约了硬件资源,降

低了成本,具有功耗低、体积小特点,同时具有较大的灵活性和可移植性,为卫星导航接收机在各个领域的广泛应用提供了便捷。

参考文献

- [1] 卢鋈,张弓,陈谷仓,等.卫星导航系统发展现状及前景展望[J]. 航天器工程, 2020, 29(4): 1-10.
- [2] 张荣兵,黄喆,孙树杰,等.一种卫星导航SoC芯片系统级仿真设计方法[J]. 遥测遥控, 2017, 38(6): 54-60.
- [3] 顾睿文,黄仰博,苏映雪,等.基于FPGA的导航接收机跟踪环路设计与实现[J]. 全球定位系统, 2014, 39(5): 55-58, 63.
- [4] 王家燃. 基于SoCFPGA的GPS/BDS接收机设计与实现[D]. 广州: 广东工业大学, 2017.
- [5] 廖梦新. 卫星导航SoC设计验证平台的研究与实现[D]. 上海: 上海交通大学, 2011.
- [6] 刘宁,马洪涛,王晓君. ARM+FPGA的双模导航接收机硬件平台设计[J]. 单片机与嵌入式系统应用, 2017, 17(4): 21-23.
- [7] 林广栋,马宏星,朱家兵,等. 一款BD2/GPS双模导航芯片 SoC子系统设计[J]. 单片机与嵌入式系统应用, 2018, 18(3): 59-63.
- [8] 谢钢. GPS原理与接收机设计[M]. 北京: 电子工业出版社, 2009.
- [9] 刘征岳,赵秋明. 北斗接收机载波跟踪环设计与实现[J]. 计算机工程与设计, 2014, 35(3): 846-851.
- [10] KOKKONEN M, PIETILA S. A new bit synchronization method for a GPS receiver[C]// Position Location and Navigation Symposium, IEEE, 2002. DOI:10.1109/PLANS.2002.998893
- [11] GROVES P D. GPS signal-to-noise measurement in weak signal and high-interference environments[J]. Navigation, 2005, 52(2): 83-94. DOI: 10.1002/j.2161-4296.2005.tb01734.x

作者简介

薛志芹 (1988—),女,硕士,研究方向为卫星导航基带信号处理。

刘坤 (1987—),男,硕士,研究方向为卫星导航接收机总体设计。

李立广 (1980—),男,研究方向为通信导航装备质量监督检验

(下转第 103 页)