DOI:10.3969/j.issn.1672-2337.2024.02.014

基于16相快速滤波实现采样率4~8 GS/s 中频信号预处理

王利华,赵微微

(中国航空工业集团公司雷华电子技术研究所, 江苏无锡 214063)

摘 要: 在机载雷达或电子战接收系统对大带宽数字中频信号的预处理过程中,针对传统并行多相滤波方式存在FPGA乘法器资源消耗过多的缺陷,提出对并行多相分解系数进行快速滤波算法构建,实现高速ADC采样率在4~8 GS/s之间的数字下变频处理。即先将高速中频采样信号解析为并行32支路,再通过数字混频及2倍抽取将基带复信号的并行度降至16,最后基于短卷积算法构建的16相快速滤波架构,实现对高采样率、大带宽信号的数字下变频预处理。通过基于并行16相快速滤波算法的宽带数字下变频设计与应用,将FPGA乘法器资源降至传统并行多相滤波方式的32%左右,大幅节省资源并提升单片FPGA对多通道、高采样率中频信号的预处理能力。

关键词:采样率;16相快速滤波;数字下变频;FPGA

中图分类号:TN957.5 文献标志码:A 文章编号:1672-2337(2024)02-0231-06

引用格式:王利华,赵微微.基于16相快速滤波实现采样率4~8 GS/s中频信号预处理[J]. 雷达科学与技术, 2024,22(2):231-236.

WANG Lihua, ZHAO Weiwei. Implementation of 4~8 GS/s Intermediate Frequency Sampling Signal Preprocessing with 16-Phase Fast Filter[J]. Radar Science and Technology, 2024, 22(2):231-236.

Implementation of 4~8 GS/s Intermediate Frequency Sampling Signal Preprocessing with 16-Phase Fast Filter

WANG Lihua, ZHAO Weiwei

(AVIC Leihua Electronic Technology Research Institute, Wuxi 214063, China)

Abstract: In the preprocessing of large bandwidth digital intermediate frequency (IF) signals in the airborne radar or electronic warfare receiving system, the traditional parallel polyphase filtering has the drawback of consuming excessive FPGA's multiplier resource. A method with construction of fast filtering algorithm by parallel polyphase decomposition coefficients is proposed to realize digital down conversion (DDC) processing of high-speed ADC sampling rate between 4 GS/s and 8 GS/s. First, the IF sampling signal is decomposed into 32 parallel branches. Then, the parallelism of the baseband complex signal is reduced to 16 through digital mixing and double decimation. Finally, to realize DDC processing of high sampling rate and large bandwidth signal, the 16-phase fast filtering architecture based on the short convolution algorithm is constructed. Through the design and application of wideband DDC based on 16-phase fast filtering, the FPGA multiplier resource is reduced to about 32% of the traditional parallel polyphase filtering, which greatly saves FPGA's resource and improves the preprocessing ability of single FPGA for multi-channel and large bandwidth signals.

Key words: sampling rate; 16-phase fast filter; digital down conversion; FPGA

0 引 言

针对未来机载雷达宽带与雷达内部电子战侦 收一体化综合软硬件设计的发展趋势,以及雷达 系统更高精度成像要求信号带宽能够达到1.8~3.6 GHz级甚至更高;电子战侦收系统^[1]小型化与集成 化设计要求单模块能够实现更大频带信号侦测, 进而需要单个通道的处理带宽能够达到2GHz 级及以上,从而改变多个1 GHz级侦收带宽拼接带 来的射频接收与数字采样硬件规模大的缺点。如 此大带宽信号的数字中频采样,要求 ADC采样率 要达到4~8 GS/s,而对数字中频信号的预处理要首 先进行宽带数字下变频。工程应用上对于高速 ADC采样率大于 500 MS/s 的数字接收信号预处 理,通常基于并行多相滤波架构实现宽带数字下 变频,即采用扩展数字中频解析信号并行度来换

收稿日期: 2023-09-24; 修回日期: 2023-10-24

 (Λ)

取FPGA对数字信号合适的处理速度,但是代价是 增加FPGA乘法器的资源消耗。当数字中频信号 解析和多相滤波的并行度随着ADC采样率的不断 增加而扩大时,采用传统并行多相滤波算法的宽 带数字下变频预处理,必将导致FPGA乘法器资源 消耗成倍增加,从而限制FPGA对大带宽数字中频 信号的预处理能力。

对高采样率、大带宽信号的数字下变频预处 理,部分研究文献^[23]提供了一些解决措施:通过多 级 FIR 和 HB 滤波器级联逐级降低处理速率、对传 统并行多相滤波进行结构优化以降低资源,以及 根据基带信号抽取倍数改进滤波架构等,这些方 法能够缓解 ADC 采样率不高时的 FPGA 预处理压 力,但很难灵活适用于并行度较高时,需要 FPGA 大量乘法器进行多相滤波运算的应用需求。

本文采用并行滤波的快速运算形式,在传统 并行2相滤波基础上首先构建出其快速运算形式, 然后再依据2相快速滤波的构建过程得到快速滤 波的4相结构,依次类推最终构建出并行16相快 速滤波运算架构。基于并行16相快速滤波算法实 现高速中频采样信号的数字下变频预处理,降低 多相滤波的子滤波器并行度,进而大幅降低FPGA 乘法器资源消耗,为机载雷达宽带接收或电子战 大带宽侦收系统的采样信号预处理提供一种有效 的工程实现解决方法。

1 并行16相快速滤波算法

1.1 快速滤波构建

设 FIR 数字低通滤波器系数为*H*(*z*),输入信 号为*X*(*z*),输出信号为*Y*(*z*),那么数字下变频滤波 过程可表示为

$$Y(z) = X(z)H(z) \tag{1}$$

将式(1)按传统并行多相分解方式进行2相分 解,得到

$$\begin{cases} Y_0 = X_0 H_0 + Z^{-1} X_1 H_1 \\ Y_1 = X_0 H_1 + X_1 H_0 \end{cases}$$
(2)

根据 Winograd 短卷积算法,将式(2)构建为并行2相滤波的快速运算形式^[4]:

$$\begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & Z^{-1} \\ -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} H_0 & 0 & 0 \\ 0 & H_0 + H_1 & 0 \\ 0 & 0 & H_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \end{bmatrix}$$
(3)

$$(Y_0 = X_0 H_0 + Z^{-1} X_1 H_1)$$

$$Y_{1} = -X_{0}H_{0} - X_{1}H_{1} + (X_{0} + X_{1})(H_{0} + H_{1})$$
⁽⁴⁾

比较式(2)和式(4)的并行2相滤波的传统方 式和快速算法架构,其关键的滤波运算过程并行 度由4减少至3,即快速算法方式仅需要3个并行 支路滤波器运算,相比4个并行支路的传统方式能 够节省FPGA乘法器资源25%。

并行4相快速滤波可通过由3组并行2相快速 滤波构建而成,其实现架构如图1所示^[5]。由于每 个2相快速滤波包含3个并行支路滤波器,那么并 行4相快速滤波算法的子滤波器并行度即为3× 3=9。相比传统并行4相滤波的子滤波器并行度 为16,快速滤波算法的FPGA乘法器资源仅为传统 方式的9/16。



依此类推,由3组并行4相快速滤波构建并行 8相快速滤波,其子滤波器运算并行度为3×9= 27;进而并行16相快速滤波通过3组并行8相快速 滤波构建,其子滤波器运算并行度为3×27=81。 而传统并行16相滤波的子滤波器运算并行度达到 了16×16=256,由此得出采用快速滤波算法架 构的FPGA乘法器资源仅为原来的81/256。

1.2 16相快速滤波

设*X_i*,*H_i*,*Y_i*(*i* = 0,1,2,…,9,*a*,*b*,…,*f*)分别表示并 行16路输入信号、数字滤波器并行16相分解系数 以及并行16路滤波输出信号。同时,令*Y*/表示81 个子滤波器,且定义如下:

$$\begin{cases} Y_{i}' = X_{i}H_{i} \\ Y_{ij}' = (X_{i} + X_{j})(H_{i} + H_{j}) \\ Y_{mnpq}' = (X_{m} + X_{n} + X_{p} + X_{q})(H_{m} + H_{n} + H_{p} + H_{q}) (5) \\ \dots \\ (i,j,m,n,p,q = 0,1,2,\dots,9,a,b,\dots,f) \end{cases}$$

根据快速滤波的构建方法,得到并行16相快速滤波算法表达式: $Y_{0} = Y_{0}' + Z^{-16} \begin{pmatrix} -Y_{1}' + Y_{2}' - Y_{3}' - Y_{4}' - Y_{5}' + Y_{6}' - Y_{7}' + Y_{8}' - Y_{9}' + Y_{a}' - Y_{b}' - Y_{c}' - Y_{d}' + Y_{e}' - Y_{f}' + Y_{1}' + Y_{13}' + Y_{15}' + Y_{19}' - Y_{26}' - Y_{2a}' + Y_{37}' + Y_{3b}' + Y_{4c}' + Y_{5d}' - Y_{6e}' + Y_{7f}' + Y_{9b}' + Y_{9d}' - Y_{ae}' - Y_{bf}' + Y_{df}' - Y_{1357}' - Y_{139b}' - Y_{159d}' + Y_{26ae}' - Y_{37bf}' - Y_{57df}' - Y_{9bdf}' + Y_{13579bdf}'$ $Y_{1} = \begin{pmatrix} -Y'_{0} - \\ Y'_{1} + Y'_{01} \end{pmatrix} + Z^{-16} \begin{pmatrix} -Y'_{2} - Y'_{3} + Y'_{4} + Y'_{5} - Y'_{6} - Y'_{7} - Y'_{8} - Y'_{9} - Y'_{a} - Y'_{b} + Y'_{c} + Y'_{d} - Y'_{e} - Y'_{f} + Y'_{23} + Y'_{24} + Y'_{24} + Y'_{37} + Y'_{3b} - Y'_{45} - Y'_{4c} - Y'_{5d} + Y'_{67} + Y'_{6e} + Y'_{7f} + Y'_{89} + Y'_{ab} + Y'_{ae} + Y'_{bf} - Y'_{2d} + Y'_{ef} - Y'_{2367} - Y'_{23ab} - Y'_{26ae} - Y'_{37bf} + Y'_{45cd} - Y'_{67ef} - Y'_{abef} + Y'_{2367abef}$ $Y_{2d} + Y_{ef} - Y_{2367} - Y_{2367} - Y_{26ae} - Y_{37bf} + Y_{45cd} - Y_{6f} - Y_{abef} + Y_{2367abef}$ $Y_{2} = \begin{pmatrix} -Y_{0}' + Y_{1}' - \\ Y_{2}' + Y_{02}' \end{pmatrix} + Z^{-16} \begin{pmatrix} -Y_{3}' + Y_{4}' - Y_{5}' + Y_{6}' + Y_{7}' - Y_{8}' + Y_{9}' - Y_{a}' + Y_{b}' + Y_{c}' - Y_{d}' + Y_{e}' + Y_{f}' - \\ Y_{3}' - Y_{3b}' - Y_{3b}' - Y_{4c}' - Y_{4c}' + Y_{5d}' - Y_{6e}' - Y_{7f}' + Y_{8a}' - Y_{bf}' + Y_{ce}' + Y_{37bf}' + Y_{46ce}' \end{pmatrix}$ $Y_{3} = \begin{pmatrix} +Y_{0}' + Y_{1}' + Y_{2}' + Y_{3}' - Y_{01}' - \\ Y_{02}' - Y_{13}' - Y_{23}' + Y_{0123}' \end{pmatrix} + Z^{-16} \begin{pmatrix} -Y_{4}' - Y_{5}' - Y_{6}' - Y_{7}' + Y_{8}' + Y_{9}' + Y_{a}' + Y_{b}' - Y_{c}' - Y_{d}' - Y_{e}' - Y_{f}' + Y_{45}' + Y_{46}' + \\ Y_{4c}' + Y_{57}' + Y_{5d}' + Y_{67}' + Y_{6e}' + Y_{7f}' - Y_{89}' - Y_{8a}' - Y_{9b}' - Y_{ab}' + Y_{cd}' + Y_{ce}' + \\ Y_{df}' + Y_{ef}' - Y_{4567}' - Y_{4567}' - Y_{45cd}' - Y_{45ce}' - Y_{57df}' - Y_{6e}' + Y_{40}' - Y_{6e}' + Y_{40}' + Y_{60}' + Y_{40}' + Y_{13}' \end{pmatrix} + Z^{-16} \begin{pmatrix} +Y_{5}' - Y_{6}' + Y_{7}' - Y_{8}' - Y_{9}' + Y_{a}' - Y_{b}' - Y_{6e}' + Y_{10}' - Y_{10}' - Y_{10}' + Y_$ $Y_{5} = \begin{pmatrix} Y_{0}' + Y_{1}' - Y_{2}' - Y_{3}' + Y_{4}' + Y_{5}' - \\ Y_{01}' - Y_{04}' - Y_{15}' + Y_{23}' - Y_{45}' \end{pmatrix} + Z^{-16} \begin{pmatrix} +Y_{6}' + Y_{7}' + Y_{8}' + Y_{9}' - Y_{a}' - Y_{b}' + Y_{c}' + Y_{d}' + Y_{e}' + Y_{f}' - Y_{67}' - \\ Y_{6e}' - Y_{7f}' - Y_{89}' - Y_{8e}' - Y_{9d}' + Y_{ab}' - Y_{cd}' - Y_{ef}' + Y_{67ef}' + Y_{89ed}' \end{pmatrix}$ $Y_{6} = \begin{pmatrix} Y_{0}' - Y_{1}' + Y_{2}' + Y_{3}' + Y_{4}' - Y_{5}' + Y_{6}' - \\ Y_{02}' - Y_{04}' + Y_{15}' - Y_{26}' - Y_{46}' + Y_{0246}' \end{pmatrix} + Z^{-16} \begin{pmatrix} -Y_{7}' + Y_{8}' - Y_{9}' + Y_{a}' + Y_{b}' + Y_{c}' - Y_{d}' + Y_{e}' - Y_{f}' + \\ Y_{7f}' - Y_{8a}' - Y_{8a}' - Y_{8a}' - Y_{9d}' - Y_{ae}' - Y_{ce}' + Y_{8aee}' \end{pmatrix}$ $Y_{7} = \begin{pmatrix} -Y_{0}' - Y_{1}' - Y_{2}' - Y_{3}' - Y_{4}' - Y_{5}' - Y_{6}' - Y_{7}' + Y_{01}' + Y_{02}' + Y_{13}' + Y_{13}' + Y_{15}' + Y_{23}' + Y_{26}' + Y_{37}' + Y_{45}' + Y_{46}' + Y_{57}' - Y_{13}' + Y_{15}' + Y_{26}' + Y_{1357}' - Y_{2367}' - Y_{4567}' + Y_{101234567}' + Y_{1012}' + Y_{1012}' + Y_{101}' + Y_{101}$ $Y_{9} = \begin{pmatrix} Y_{0}' + Y_{1}' + Y_{2}' + Y_{3}' - Y_{4}' - Y_{5}' + Y_{6}' + Y_{7}' + Y_{8}' + Y_{9}' - Y_{01}' - \\ Y_{08}' - Y_{19}' - Y_{23}' - Y_{26}' - Y_{37}' + Y_{45}' - Y_{67}' - Y_{89}' + Y_{0189}' - Y_{2367}' \end{pmatrix} - Z^{-16} \begin{pmatrix} -Y_{a}' - Y_{b}' + Y_{c}' + Y_{d}' - Y_{e}' - Y_{f}' + \\ Y_{ab}' + Y_{ae}' + Y_{bf}' - Y_{cd}' + Y_{ef}' - Y_{abef}' \end{pmatrix}$ $Y_{a} = \begin{pmatrix} Y_{0}' - Y_{1}' + Y_{2}' - Y_{3}' - Y_{4}' + Y_{5}' - Y_{6}' - Y_{7}' + Y_{8}' - Y_{9}' + \\ Y_{a}' - Y_{02}' - Y_{08}' + Y_{19}' + Y_{37}' + Y_{46}' - Y_{2a}' + Y_{028a}' \end{pmatrix} - Z^{-16} \begin{pmatrix} +Y_{b}' + Y_{c}' - Y_{d}' + Y_{e}' + \\ Y_{f}' - Y_{bf}' - Y_{ce}' \end{pmatrix}$ $\begin{pmatrix} Y_{a} - I_{02} - I_{08} + I_{19} + I_{37} + I_{40} - I_{60} & I_{40} & I_{60} &$ $Y'_{9b} + Y'_{ab} - Y'_{0123} - Y'_{0189} - Y'_{028a} - Y'_{139b} - Y'_{23ab} + Y'_{4567} - Y'_{89ab} + Y'_{012389ab}$ $Y_{c} = \begin{pmatrix} Y_{0}' + Y_{1}' - Y_{2}' + Y_{3}' + Y_{4}' - Y_{5}' + Y_{6}' - Y_{7}' + Y_{8}' + Y_{9}' - Y_{a}' + Y_{b}' + Y_{c}' - \\ Y_{04}' - Y_{08}' - Y_{13}' - Y_{19}' + Y_{2a}' - Y_{3b}' - Y_{4c}' + Y_{57}' - Y_{8c}' - Y_{9b}' + Y_{048c}' + Y_{139b}' \end{pmatrix} + Z^{-16} \begin{pmatrix} -Y_{d}' + Y_{e}' - Y_{d}' + Y_{e}' - Y_{10}' + Y_$ $Y_{d} = \begin{pmatrix} -Y_{0}' - Y_{1}' + Y_{2}' + Y_{3}' - Y_{4}' - Y_{5}' - Y_{6}' - Y_{7}' - Y_{8}' - Y_{9}' + Y_{a}' + Y_{b}' - Y_{c}' - Y_{d}' + Y_{a}' + Y_{04}' + Y_{04}' + Y_{08}' + Y_{15}' + Y_{19}' - Y_{23}' - Y_{2a}' - Y_{3b}' + Y_{45}' + Y_{4c}' + Y_{5d}' + Y_{67}' + Y_{89}' + \end{pmatrix} + Z^{-16} \begin{pmatrix} -Y_{e}' - Y_{a}' + Y_{a}'$ $(Y'_{sc} + Y'_{9d} - Y'_{0145} - Y'_{0189} - Y'_{048c} - Y'_{159d} + Y'_{23ab} - Y'_{45cd} - Y'_{89cd} + Y'_{014589cd}$ $Y_{e} = \begin{pmatrix} -Y'_{0} + Y'_{1} - Y'_{2} + Y'_{3} - Y'_{4} + Y'_{5} - Y'_{6} + Y'_{7} - Y'_{8} + Y'_{9} - Y'_{a} - Y'_{b} - Y'_{c} + Y'_{d} - Y'_{e} + Y'_{2} + Y'_{2} + Y'_{2} + Y'_{2} + Y'_{2} + Y'_{3} + Y'_{4} + Y'_{4} - Y'_{5} + Y'_{6} + Y'_{6} + Y'_{6} + Y'_{6} - Y'_{6} + Y'_{6} +$ $(Y'_{9d} + Y'_{ae} + Y'_{ce} - Y'_{0246} - Y'_{028a} - Y'_{048c} - Y'_{159d} - Y'_{26ae} - Y'_{46ce} - Y'_{8ace} + Y'_{02468ace}$ $Y'_{0} + Y'_{1} + Y'_{2} + Y'_{3} + Y'_{4} + Y'_{5} + Y'_{6} + Y'_{7} + Y'_{8} + Y'_{9} + Y'_{4} + Y'_{6} + Y'_{4} + Y'_{4} + Y'_{6} + Y'_{1} - Y'_{01} - Y'_{02} - Y'_{04} - Y'_{08} - Y'_{08} - Y'_{10} + Y'_{10} +$ $Y_{13}' - Y_{15}' - Y_{19}' - Y_{23}' - Y_{26}' - Y_{2a}' - Y_{37}' - Y_{3b}' - Y_{45}' - Y_{46}' - Y_{4c}' - Y_{57}' - Y_{5d}' - Y_{67}' - Y_{6e}' - Y_{7f}' - Y_{89}' - Y_{8a}' - Y_{8c}' - Y_{8$

 $Y_{f} = \begin{bmatrix} Y_{9b}^{15} - Y_{9d}^{15} - Y_{ab}^{15} - Y_{ab}^{25} - Y_{cd}^{25} - Y_{cd}^{26} - Y_{ce}^{15} - Y_{ef}^{15} - Y_{ef}^{15} + Y_{0123}^{15} + Y_{0145}^{15} + Y_{0145}^{15} + Y_{0189}^{15} + Y_{0186}^{15} + Y_{1357}^{15} + Y_{139b}^{15} + Y_{1$ $Y'_{159d} + Y'_{2367} + Y'_{23ab} + Y'_{26ae} + Y'_{37bf} + Y'_{45c7} + Y'_{45cd} + Y'_{45ce} + Y'_{57df} + Y'_{67ef} + Y'_{89ab} + Y'_{89cd} + Y'_{8ace} + Y'_{9bdf} + Y'_{abef} + Y'_{$ $Y_{cdef}' - Y_{01234567}' - Y_{012389ab}' - Y_{014589cd}' - Y_{02468ace}' - Y_{13579bdf}' - Y_{2367abef}' - Y_{4567cdef}' - Y_{89abcdef}' + Y_{0123456789abcdef}' - Y_{13579bdf}' - Y_{13579$

(6)

根据上述16相快速滤波计算公式,宽带数字 下变频的并行16支路低通滤波输出是对81个子 滤波器运算结果的再次组合,即其核心运算并行 度只有81。相比传统并行滤波架构的256个子滤 波器及组合,快速滤波的运算复杂度与FPGA资源 消耗都明显降低,这是工程上能够基于FPGA实现 对高采样率信号预处理的重要前提。

2 信号与滤波系数预处理

2.1 信号解析与混频

对于采样率位于4~8 GS/s的高速采样中频预 处理,由于其支路信号并行度和数据率的双重增 加,在进行数字中频信号解析和数字下变频过程 中,FPGA的处理速度和资源消耗同时面临较大压 力。为尽可能降低FPGA处理复杂度,高速ADC采 样率 f_s 与大带宽信号的载频 f_c 通常要符合带通采 样定理,即 $f_c = (2n + 1) f_s/4 (n = 0, 1, 2, \cdots)$ 。符合 这种特定关系的数字下变频的数字本振只有0和 ±1三个常数,数字混频只需对信号进行加减运算, 简化运算的同时也节省 FPGA乘法器资源开销。

此外,由于数字本振中的常数0,使并行解析 信号在数字混频的过程中也同时完成了2倍抽取。 以将数字中频信号x解析为并行度M的多支路信 号为例,即 $x = x_0, x_1, x_2, \dots, x_{M-1},$ 数字本振实部为 $n_i = 1,0, -1,0,1,0, \dots, 虚部为n_q = 0,1,0, -1,0,1,\dots,$ 数字混频后基带复信号可表示为

$$\begin{cases} x_i = x_0, 0, -x_2, 0, x_4, 0, x_6, \cdots, x_{M-2} \\ x_a = 0, x_1, 0, -x_3, 0, x_5, 0, x_7, \cdots, x_{M-1} \end{cases}$$
(7)

将基带复信号实部与虚部值为0的支路分别 剔除掉,那么数字混频后的基带信号可进一步简 化为

$$\begin{cases} x_i = x_0, -x_2, x_4, -x_6, \cdots, x_{M-2} \\ x_q = x_1, -x_3, x_5, -x_7, \cdots, x_{M-1} \end{cases}$$
(8)

简化后的基带复信号并行度是数字中频解析 信号的1/2,也相当于对数据实现了2倍抽取,这使 得后续并行多相滤波的并行度也减半,降低数字 下变频多相滤波运算的复杂度。

如果信号载频 f_e 与高速ADC采样率 f_s 无法满 足带通采样定理中的特定关系,则可以将 f_e 拆解为 两级: $f_e = (2n + 1) f_s/4 + f_0 (n = 0, 1, 2, ...)$ 。第一级 载频按照本文数字混频及2倍抽取方法进行处理, 然后在数字下变频后通过对基带IQ信号进行频谱 搬移的方式实现去除第二级载频f₀,而不必增加多 相滤波的并行度。

2.2 滤波器系数分解

数字下变频数字低通滤波实际上是时域信号 与滤波器系数的线性卷积过程,设输入信号和输 出信号分别为x(n)和y(n),(N-1)阶 FIR 滤波器 为h(n),则数字滤波计算表达式为

$$y(n) = x(n) * h(k) = \sum_{k=0}^{N-1} x(n) h(n-k)$$
(9)

由于数字中频信号解析并行度*M*,采用基于 并行多相滤波架构的数字下变频算法也需将滤波 器系数分解为并行*M*相^[6]。多相分解后的第*m*相 滤波器系数可表示为

 $h_m(k) = h(m + Mk)$

 $(m = 0, 1, 2, \dots, M - 1; k = 0, 1, 2, \dots, N/M - 1)(10)$

根据式(8)、式(9)和式(10)的中频信号预处 理及滤波线性卷积实现过程,数字混频预处理后 并行 *M*/2路I信号滤波器系数为多相分解的 *M*/2 个偶数相分支,而并行 *M*/2路Q信号滤波器系数则 为*M*/2个奇数相分支,即

$$\begin{cases} h_{mi}(k) = h_m(k)(m = 1, 3, 5, \dots, M - 1) \\ h_{mi}(k) = h_m(k)(m = 1, 3, 5, \dots, M - 1) \end{cases}$$
(11)

 $h_{mq}(k) = h_m(k)(m = 0, 2, 4, \dots, M - 2)$

数字中频信号与滤波器系数预处理后,将并 行 M 路解析信号数字混频后简化为 M/2 路基带复 信号,滤波器系数配合这种简化方式将系数拆解 为并行 M/2 相奇偶分支,进而可以基于并行 M/2 相 快速滤波架构实现数字下变频低通滤波处理。

3 高采样信号预处理实现

3.1 数字下变频架构

在机载雷达或电子战系统基于FPGA实现高速采样信号预处理,通常要确保FPGA速率与资源的均衡。FPGA的处理时钟不宜过高,适当速率的处理时钟有利于FPGA在高低温环境下的时序稳定性;资源消耗也需预留足够的余量,保证FPGA工程编译可实现。FPGA在实现多通道ADC采样信号预处理功能,处理时钟一般不超过250 MHz,

而当时钟达到300 MHz量级时,数字下变频算法等 逻辑时序调试就较为困难,尤其是在-55~70℃的 全温度工作范围内。基于此,可以将高速ADC采样 率区间划分为500 MS/s~1 GS/s、1~2 GS/s、2~4 GS/s、 4~8 GS/s等,中频信号数据解析并行度分别对应4、 8、16和32,这样解析后的并行支路速率基本上在 250 MHz内。不过在区间临界点则可根据工程实 现需要选择适合的并行度,采样区间的划分与解 析并行度并不是绝对不变的。

本文针对采样率在 4~8 CS/s 区间的数字接收 预处理,高速 ADC采样中频信号在 FPGA 中解析为 并行 32 支路,其单个支路的数据率在 125~250 MS/s 之间。由于中频信号解析预处理能够将并行 32 路 信号混频及抽取至并行 16 路基带复信号,进而与 并行 16 相快速滤波算法架构结合实现数字下变频 低通滤波,最终实现高速采样信号的预处理。以 传统并行 4路 LVDS 接口高速 ADC 芯片为例,应用 Xilinx FPGA 基于并行 16 相快速滤波对高采样率 信号进行数字下变频的实现架构如图 2 所示。



3.2 硬件实现与测试

由于高速ADC采样率位于4~8 GS/s的数字中

频信号分解为32相时,单支路的处理时钟都不大于250 MHz,位于此采样率范围内的数字中频信号预处理均可以采用本文的16相快速滤波架构,只是不同采样率下的FPGA处理时钟速率不同。TI公司的一款并行 LVDS 接口的高速 ADC 芯片ADC12DL3200,其模拟接收带宽7.8 GHz、单通道模式最大采样率为6.4 GS/s,适用于机载雷达大带宽信号或者电子战侦收系统的高速中频信号采样。本文选取某块搭载 ADC12DL3200采样芯片的数字中频接收模块、且硬件布线设计能够支持的5.6 GS/s采样率(4~8 GS/s之间的处理架构完全相同)为例,通过FPGA 仿真与硬件测试,验证基于16 相快速滤波架构的高采样率中频信号预处理实现过程。

采样率为5.6 GS/s 的数字中频信号如果解析 为并行16支路,则 FPGA处理时钟为350 MHz,可 能会导致FPGA 对信号的后续处理在时序上很难 实现;而按照本文方法解析为并行32支路、处理时 钟降至175 MHz,尽管会导致多相滤波的并行度加 倍、进而增加 FPGA乘法器资源数量,但资源与速 率的合理均衡更符合 FPGA 工程设计与应用规律。 采样率f_s设置为5.6 GS/s,接收处理的信号带宽最 大可达2.5 GHz,模拟信号载频设计为f_e=3f_s/4= 4 200 MHz符合带通采样定理。选择频率为4 209 MHz 和5 450 MHz的点频信号进行测试,中频信号解析 后的频谱分析结果如图3所示。



数字下变频数字低通滤波器以63阶为例,按 照式(11)系数分解方法和图2数字下变频实现架 构,即*N* = 64,*M* = 32,则I路(或Q路)信号滤波器

组的子滤波器系数个数为L = N/M = 2。根据前文 分析的并行16相传统滤波和快速滤波的滤波器组 并行度分别为256和81,对I路(或Q路)信号分别 采用一组滤波器组进行滤波,FPGA乘法器资源比 较如表1所示。

表1 并行16相传统滤波与快速滤波资源比较

滤波方式	FPGA乘法器(DSP48E1)数量
传统滤波	$256 \cdot L \cdot 2 = 1024$
快速滤波	$81 \cdot L \cdot 2 = 324$

应用16相快速滤波算法对频率为4209 MHz 和5450 MHz点频信号进行数字下变频预处理,基 带信号频谱如图4所示。



3.3 不同滤波架构对比

对高速采样信号的数字下变频多相滤波处 理,不同实现方式^[78]的优缺点比较如表2所示。

滤波架构	优点	缺 点
传统并 行多相	逻辑实现架构简单	子滤波器并行度成倍增加, FPGA乘法器资源消耗大
多级滤 波级联	逐级降低处理速率	滤波器级联相对于一次滤 波,乘法器资源消耗更多, 且滤波处理延时更大
本文快 速滤波	子滤波器并行度与 FPGA资源消耗均大 幅减少,达到资源和 处理速率的均衡	前期快速滤波构建复杂,但 是构建完成后不必再更改

表2 多相滤波不同架构比较

不过部分高速ADC芯片有内置DDC功能,并可以进行低通滤波后输出基带信号,进而将ADC

原始采样中频信号的并行度降低一倍,达到本文 对中频信号数字混频过程中的2倍抽取相同的效 果,同时滤波在ADC芯片中实现能够节省FPGA处 理资源。但是,并不是所有的高速ADC芯片都支 持DDC功能,当模块硬件设计阶段对芯片选型无 法确保能够采用此类ADC芯片时,本文基于FPGA 实现大带宽信号的数字下变频则提供了一种解决 方案,并具有相应的应用价值。

4 结束语

通过构建并行多相滤波的快速运算形式,以 及对高速采样信号解析与滤波器系数分解的优化设 计,实现了基于16相快速滤波算法架构对4~8 GS/s 高采样率中频信号数字下变频预处理。快速滤波 运算架构的应用,极大地降低了多相滤波运算对 FPGA乘法器资源消耗,提升FPGA对大带宽信号 的预处理能力。

参考文献:

- [1] 曹兰英,郭明明,罗美方.雷达与电子战的认知博弈[J]. 雷达科学与技术,2021,19(5):552-557.
- [2] 吴晓晔, 胡志伟, 白晓远. 基于高速采样的实时 DDC架 构技术[J]. 计算机测量与控制, 2020, 28(1):255-260.
- [3] 戴曜泽,张棋,徐小淇.基于FPGA的高速高效率数字下 变频[J].航天电子对抗,2021,37(2):47-51.
- [4] 李凯勇. 基于 FPGA 的高效 FIR 滤波器设计[J]. 青海大 学学报, 2017, 35(6):56-60.
- [5] 孙田雨.并行 FIR 滤波器系数设计及结构优化[D].杭州:浙江大学,2017.
- [6] 赵飞,杨建.多相分解的高速并行 FIR 滤波器的 FPGA 实现[J].电子信息对抗技术,2022,37(5):105-109.
- [7] 田波,程云鹏,罗庆驰.短波宽带数字下变频设计与仿 真[J].通信技术,2022,55(8):977-983.
- [8] 李来轩.多通道采集与处理的若干关键技术研究[D]. 杭州:浙江大学,2022.

作者简介:

王利华 男,硕士,高级工程师,主要研究方向为数字 中频信号预处理。

赵微微 男,硕士研究生,主要研究方向为中频信号 采样接收与预处理。