

# 基于 Farrow 架构的任意倍重采样的 FPGA 设计与实现

袁明<sup>1</sup>, 胡志臣<sup>2</sup>, 张延顺<sup>1</sup>, 武福存<sup>2</sup>, 朱硕鹏<sup>2</sup>, 蒋明明<sup>2</sup>

(1. 北京航空航天大学 仪器科学与光电工程学院, 北京 100191;

2. 北京航天测控技术有限公司, 北京 100041)

**摘要:** 任意波形发生器通常使用直接数字波形合成技术生成任意波形, 然而使用此技术实现可变采样率功能时, 需要调整时钟的工作频率, 导致输出中存在难以滤除的镜像频率以及硬件实现难度增加; 针对上述问题, 对基于 Farrow 架构的任意倍重采样方法进行了研究, 在 FPGA 上进行了固定时钟驱动下的 32 路并行的任意倍重采样的逻辑设计与实现; 并针对逻辑实现中出现的量化误差积累问题, 提出了清除累积误差的逻辑实现方案; 经实验测试, 实现了在 187.5 MHz 固定时钟驱动下 1 ksp/s~6 Gsp/s 范围的采样率转换, 并解决了量化误差积累造成的长时间运行时波形失真的问题。

**关键词:** 采样率; Farrow 架构; 并行; 任意倍重采样; 量化误差

## Design and Implementation of Arbitrary Multiple Resampling FPGA Based on Farrow Architecture

YUAN Ming<sup>1</sup>, HU Zhichen<sup>2</sup>, ZHANG Yanshun<sup>1</sup>, WU Fucun<sup>2</sup>, ZHU Shuopeng<sup>2</sup>, JIANG Mingming<sup>2</sup>

(1. School of Instrumentation Science and Opto-Electronics Engineering, Beihang University, Beijing 100191, China;

2. Beijing Aerospace Measurement & Control Technology Co., Ltd., Beijing 100041, China)

**Abstract:** Arbitrary waveform generators usually employ direct digital waveform synthesis techniques to generate arbitrary waveform. However, when this technology is used to achieve a variable sampling rate, it needs to adjust the working frequency of the clock, resulting in image frequencies that are difficult to filter out in the output and increasing the complexity of hardware implementation. To address these issues, research is conducted on an arbitrary multiple resampling method based on the Farrow architecture. A parallel implementation of a 32-channel arbitrary resampling system is designed and realized on an FPGA under a fixed clock drive. Additionally, a logical implementation scheme for clearing error accumulation is proposed to accumulate quantization errors in logical implementation. Experimental results demonstrate that the system achieves sampling rate conversion from 1 ksp/s to 6 Gsp/s with a fixed clock frequency of 187.5 MHz and effectively solves waveform distortion caused by quantization error accumulation during long-term operation.

**Keywords:** sampling rate; Farrow architecture; parallel; arbitrary multiple resampling; quantization error

## 0 引言

电子仪器设备的持续发展, 对测试所需信号源提出了日益严格的要求。而任意波形发生器 (AWG, arbitrary waveform generator) 可以为仪器设备提供高质量、多样化的激励源<sup>[1]</sup>, 成为解决这一需求的重要手段, 并且已经在通信、雷达、国防测试等众多领域发挥重要作用<sup>[2-3]</sup>。AWG 需要对波形数据进行高速、并行处理, 而现场可编程逻辑门阵列 (FPGA, field programmable gate array) 具有高速、并行的工作特性, 可以满足 AWG 的需求<sup>[4-5]</sup>, 因此被广泛应用于 AWG 的开发中。

AWG 采用直接数字频率合成 (DDFS, direct digital

frequency synthesis) 技术实现标准波形的发生<sup>[6-7]</sup>、采用直接数字波形合成 (DDWS, direct digital waveform synthesis) 技术实现任意波形的发生<sup>[8]</sup>。但是, 采用 DDWS 技术实现任意波形输出时, 其采样率与时钟的输出频率相关, 因此在实现输出采样率的变化时, 必须相应地改变时钟的输出频率。然而, 时钟输出频率的变化会导致镜像频率位置发生改变, 使得使用截止频率不变的低通滤波器进行滤除时变得困难, 导致输出波形质量下降<sup>[9]</sup>。此外, 采用可变时钟实现采样率转换对硬件系统的要求也更高。

为了解上述问题, 本文选择对波形进行重采样, 使得重采样后的波形采样率为定值, 镜像频率在固定采样率附

收稿日期: 2024-06-24; 修回日期: 2024-07-20。

作者简介: 袁明 (1998-), 男, 硕士。

胡志臣 (1981-), 男, 硕士, 研究员。

引用格式: 袁明, 胡志臣, 张延顺, 等. 基于 Farrow 架构的任意倍重采样的 FPGA 设计与实现[J]. 计算机测量与控制, 2024, 32(10): 263-268, 276.

近, 采用截止频率不变的低通滤波器即可实现镜像频率的滤除。采样率转换可以分为模拟实现与数字实现两种方式, 数字实现方法又可以分为简单有理数倍插值抽取<sup>[9-10]</sup>与基于 Farrow 结构的任意倍重采样<sup>[11-13]</sup>, 模拟实现时会因为模拟器件的使用降低波形质量, 简单有理数倍插值抽取不能实现任意倍的插值, 而且插值抽取过程中造成了资源浪费。基于 Farrow 结构的重采样滤波器不仅可以实现任意倍的重采样, 而且占用较少的资源, 故本文采用此方法对波形序列进行采样率转换。

为了降低对硬件时钟的要求, 本文基于 Farrow 架构进行了固定时钟驱动下的任意倍重采样逻辑设计, 使用固定时钟即可实现 1 ksp/s ~ 6 Gsp/s 输出采样率的改变。然而, 在逻辑实现时需对时间参数进行量化处理, 量化误差成为了一个突出的问题, 长时间的量化误差积累会导致输出波形失真, 本文针对量化误差积累问题提出了周期性清除累计误差的逻辑实现方法。

### 1 Farrow 结构任意倍重采样算法

任意倍采样率转换时, 需要对原始数据点进行任意倍插值, 而 Farrow 结构本质上由多个 FIR 滤波器组成, 改变延时参数的值就可以实现不同倍数的插值, 且不需要对 FIR 滤波器系数值进行更改<sup>[14]</sup>。因此 Farrow 结构可以实现此功能, 并且便于硬件上实现。

采样率转换过程可以等效为原始数字信号  $x(kT_i)$  经过 DAC 转换为模拟信号, 模拟信号经过滤波器  $h(t)$  输出重构信号  $y(t)$ , 然后经过 ADC 进行重采样, 产生重采样后的信号  $y(nT_o)$ ,  $T_i$  为输入信号的采样周期,  $T_o$  为输出信号的采样周期。

$$y(t) = \sum_{k=-\infty}^{+\infty} x(kT_i)h(t - kT_i) \quad (1)$$

对重构信号以采样周期  $T_o$  进行重采样, 得到转换后的新信号。

$$y(nT_o) = \sum_{k=-\infty}^{+\infty} x(kT_i)h(nT_o - kT_i) = \sum_{k=-\infty}^{+\infty} x(kT_i)h\left[T_i\left(\frac{nT_o}{T_i} - k\right)\right] \quad (2)$$

令  $nT_o/T_i$  的整数部分为  $k_n$ , 小数部分为  $u_n$ ,  $k_n$  为距离插值点最近的原数据点,  $u_n$  为插值点与原数据点之间的距离, 也为小数延迟。为了便于硬件实现, 将滤波器长度限定为  $N$ 。

$$y(nT_o) = \sum_{k=0}^{N-1} x[(k_n - k)T_i]h[(k + u_n)T_i] \quad (3)$$

$$k_n = \text{INT}(nT_o/T_i) \quad (4)$$

$$u_n = \frac{nT_o}{T_i} - k_n \quad (5)$$

使用多项式构造滤波器  $h[(k + u_n)T_i]$ , 代入到式 (3) 中, 得到最终输出结果<sup>[14]</sup>为:

$$y(nT_o) = \sum_{k=0}^{N-1} x[(k_n - k)T_i] \sum_{m=0}^{M-1} c_{k,m}u_n^k =$$

$$\sum_{m=0}^{M-1} \sum_{k=0}^{N-1} x[(k_n - k)T_i]c_{k,m}u_n^k \quad (6)$$

其中:  $c_{k,m}$  为滤波器系数, 式 (6) 可以使用单路 Farrow 结构来实现, 实现架构如图 1 所示<sup>[15-16]</sup>。滤波器系数为固定值, 改变延时参数  $u_n$  即可改变插值位置, 不需要实时对  $c_{k,m}$  进行计算。

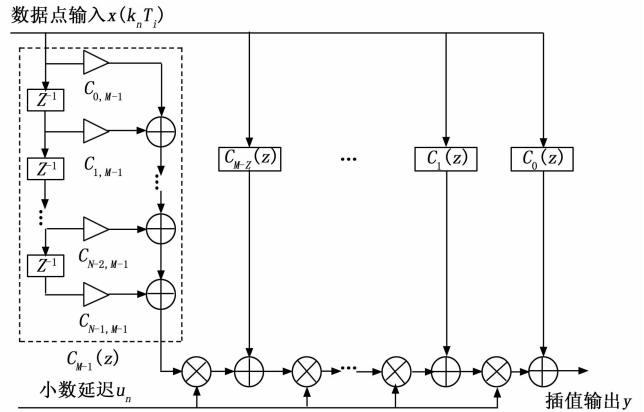


图 1 单路 Farrow 结构的采样率转换滤波器

### 2 滤波器系数与时间参数设计

式 (6) 与图 1 所示的 Farrow 架构表明, 实现任意倍插值时需要确定原始数据序列、Farrow 滤波器系数  $c_{k,m}$  和小数延时  $u_n$  等 3 部分, 其中输入的原始数据序列通过移位寄存器既可以得到, 所以只需对滤波器系数  $c_{k,m}$  与小数延时  $u_n$  进行设计。

#### 2.1 滤波器系数设计

Farrow 滤波器系数可以分为时域设计与频域设计<sup>[17]</sup>两种不同的设计方法, 时域设计方法较频域设计方法更易构建, 其中较为常用的时域求取方法包括拉格朗日插值<sup>[18-19]</sup>、B-spline 插值<sup>[20]</sup>、抛物插值<sup>[21]</sup>等。其中拉格朗日插值系数拟合数据精度高于 B-spline 插值与抛物插值, 故本文选择基于拉格朗日插值方法求取滤波器系数。

拉格朗日插值公式如式 (7) 所示:

$$f(x) = \sum_{i=0}^n \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j} y(x_i) = \sum_{i=0}^n \sum_{k=0}^n a_{ik} x^k y(x_i) \quad (7)$$

其中:  $n$  为拉格朗日插值阶数, 也为插值所需最少点数。由式 (7) 与式 (6) 对比可知, 当  $M=n$ 、 $N=n$  时,  $a_{ik}$  为 Farrow 滤波器系数  $c_{k,m}$ ,  $x$  为小数延时  $u_n$ 。现对基于立方拉格朗日插值的插值效果进行仿真, 并与同阶的抛物插值、B-spline 插值的插值效果进行对比, 对 16 点正弦波进行 4 倍插值后的结果如图 2 所示, 插值点相对于真值点之间的误差如图 3 所示, 图 2、图 3 表明立方拉格朗日的插值误差远小于抛物插值与同阶的 B-spline 插值, 具有更好的插值效果。

不同阶数的拉格朗日插值的精度不同, 阶数增加可以获得更高的插值精度, 但是也会导致消耗的资源增加。本文在硬件实现时, 综合考虑所消耗硬件资源与插值精度,  $n$

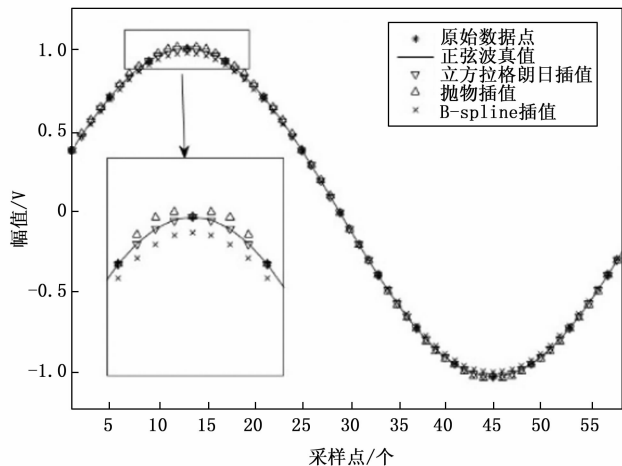


图 2 不同插值方法插值效果对比

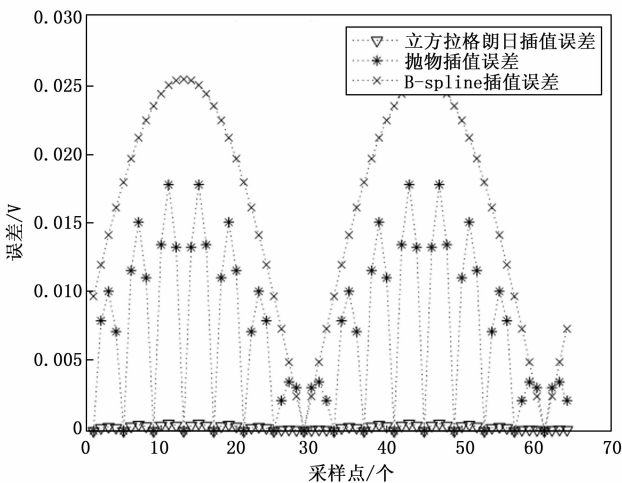


图 3 不同插值方法的插值误差

取值为 3, 即采用立方拉格朗日插值求取滤波器系数, 插值需要 4 个原始数据点, 插值点位于中间两点之间, 结合公式 (7) 可得:

$$f(u) = \mathbf{A} \times \mathbf{B} \times \mathbf{C} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}^T \begin{bmatrix} -\frac{1}{6} & \frac{1}{2} & -\frac{1}{3} & 0 \\ \frac{1}{2} & -1 & -\frac{1}{2} & 1 \\ -\frac{1}{2} & \frac{1}{2} & 1 & 0 \\ \frac{1}{6} & 0 & -\frac{1}{6} & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix} \quad (8)$$

式中,  $y_0, y_1, y_2, y_3$  为输入原始采样点, 矩阵  $\mathbf{A}$  为输入原始数据点矩阵, 矩阵  $\mathbf{B}$  为立方拉格朗日系数阵, 矩阵  $\mathbf{C}$  为小数延时矩阵,  $f(u)$  为插值后输出。式 (8) 表示确定小数延迟即可实现对对应位置处的插值, 为本文进行任意倍重采样的算法公式。

### 2.2 时间参数设计

采样率转换时插值点的位置跟随工作时钟改变, 因此

需要确定距离插值点最近且不超过其的原采样点  $k_n$ 、距离  $k_n$  的分数延迟  $u_n$ , 并转化为利于硬件实现的架构。令采样率转换因子  $r = T_o/T_i$ , 代入公式 (4) (5) 得:

$$k_n = INT(nr) \quad (9)$$

$$u_n = nr - k_n \quad (10)$$

设第  $n$  个插值点位置为  $p_n$ , 由公式 (10) 可得:

$$p_n = nr = u_n + k_n \quad (11)$$

由式 (11) 可知,  $p_n$  可由转换因子  $r$  累加得到, 其整数部分、小数部分分别为  $k_n, \mu_n$ 。令  $\Delta_p = p_n - p_{n-1}$ , 由于本文只对数据进行升采样, 故  $\Delta_p$  值为 0 或 1, 当  $\Delta_p = 0$  时, 第  $n, n-1$  个采样点对应的原始采样点相同, 当  $\Delta_p = 1$  时, 需要对原始采样点进行更新。根据前述时间参数的分析, 设计的时间参数求取流程如图 4 所示。

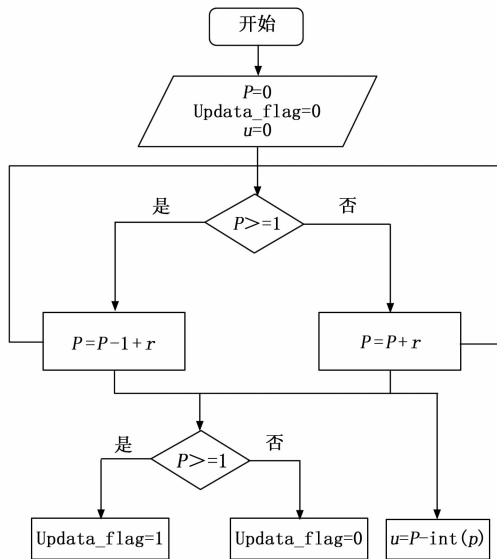


图 4 时间参数求取流程

图 4 中,  $P$  为当前插值点的插值位置,  $Updata\_flag$  为采样点数据更新标志,  $u$  为小数延迟。开始后先对插值点位置  $P$ 、更新标志  $Updata\_flag$ 、小数延迟  $u$  初始化置 0; 然后判断  $P$  值,  $P \geq 1$  时, 上一时钟周期内已经完成原始采样点的更新, 故需对其进行减 1 操作后累加采样率转换因子  $r$ ,  $P < 1$  时, 直接累加  $r$ ; 然后对小数延迟  $u$  进行赋值, 并判断  $P$  值,  $P \geq 1$  时,  $\Delta_p = 1$ , 采样点更新标志置 1, 更新原始采样点,  $P < 1$  时, 原始采样点不需更新操作。

为了便于逻辑实现, 需对采样率转换因子  $r$  进行量化操作, 转化为有限位宽的 2 进制数, 针对量化操作后的小数值可以进行截断处理、进位处理或舍入处理, 3 种处理方式都会带来量化误差, 而时间参数求取过程中需要对转换因子  $r$  进行累加, 造成量化误差积累, 长时间运行时导致波形失真, 所以在逻辑实现时需对量化误差积累问题进行处理。

### 3 FPGA 逻辑实现方案

实现 1 ksps~6 Gsps 的大范围采样率转换时, 单路实现最高需要 6 GHz 的时钟频率, 对时钟与 FPGA 处理速度

的要求过高, 实现困难, 然而 FPGA 具有并行工作的特性, 通过多路并行的方式降低对时钟、芯片处理速度的要求, 亦可以实现高速的数据处理操作。但是多路并行处理在提升数据处理速度的同时也会增加 FPGA 逻辑资源的利用, 因此需要平衡并行路数与资源的关系, 选择合适并行路数以满足 1 ksp/s~6 Gsp/s 采样率转换的需求。

本文选用 Xilinx 公司的 xc6k060-ffva1156-2-i 型号的 FPGA, 具有丰富的查找表、触发器、BRAM 以及 DSP 等众多资源。为了实现 1 ksp/s~6 Gsp/s 的采样率转换, 对 Farrow 架构进行 32 路并行处理, 将所需时钟工作频率降为 187.5 MHz, 选择的 FPGA 的板载资源、处理速度都可以满足采样率转换需求。

为了实现在 187.5 MHz 固定时钟驱动下的大范围采样率转换, 本文基于前述的滤波器系数设计与时间参数设计对 32 路并行的任意倍插值滤波器逻辑架构进行了设计, 架构如图 5 所示。图 5 中分为 6 个模块, 分别为移位寄存与滤波模块、滤波结果缓存模块、相位累加模块、读使能模块、数据选择模块、乘和累加模块, 其中相位累加模块、读使能模块、数据选择模块属于时间参数处理模块。

缓存模块对滤波结果的缓存与该模块对移位寄存与滤波模块和相位累加器模块的逻辑控制可实现 6 个模块共同在 187.5 MHz 时钟驱动下工作; 同时针对量化误差累积问题, 设定了误差清零条件, 在满足条件时对相位累加模块进行初始化操作, 清除累积的量化误差。现对 6 个模块进行详细阐述:

1) 移位寄存与滤波模块:

移位寄存与滤波模块实现对输入的 32 路数据的移位寄存并基于 Farrow 滤波器系数矩阵对 32 数据的并行滤波。该模块的工作状态受滤波结果缓存模块中存储的数据量控制, 处于分时工作状态, 当缓存数据量大于一定值时, 该模块在 187.5 MHz 时钟控制下输入 32 路数据、输出 32 路滤波结果; 当缓存数据量小于一定值时, 该模块不进行数据输入与输出。数据移位寄存如图 6 所示。

由式 (8) 可知, 基于立方拉格朗日插值的 Farrow 架构需要 4 个采样点才能实现插值过程, 所以 32 路并行处理时, 单一时钟周期内移位寄存器至少需要存放 35 个数据点。模块运行时, 时钟上升沿到来, 读入数据, 同时移位寄存器左移 32 个数据点, 并对前 32 个点进行滤波处理, 即实现公式 (8) 中矩阵 **A** 与矩阵 **B** 的相乘, 第 *i* ( $0 \leq i \leq 31$ ) 路原始采样点组成的矩阵 **A<sub>i</sub>** 如式 (12) 所示, 第 *i* 路矩阵相乘结果存入滤波结果缓存模块的 FIFO<sub>*i*</sub> 中。

$$A_i = [y_i \quad y_{i+1} \quad y_{i+2} \quad y_{i+3}] \quad (12)$$

2) 滤波结果缓存模块:

滤波模块、时间参数处理模块、乘和累加模块使用 187.5 MHz 时钟, 对原始采样点进行升采样时, 滤波模块的输出数据量大于后级模块所需的输入数据量, 因此需要缓存模块对滤波模块的输出结果进行缓存; 此外, 缓存模块有缓存量的限制, 数据溢出与读空都会造成重采样数据的丢失, 所以需要根据缓存量控制其他模块工作状态。

此模块由 32 路 FIFO 组成, 将滤波后产生的 32 路滤波

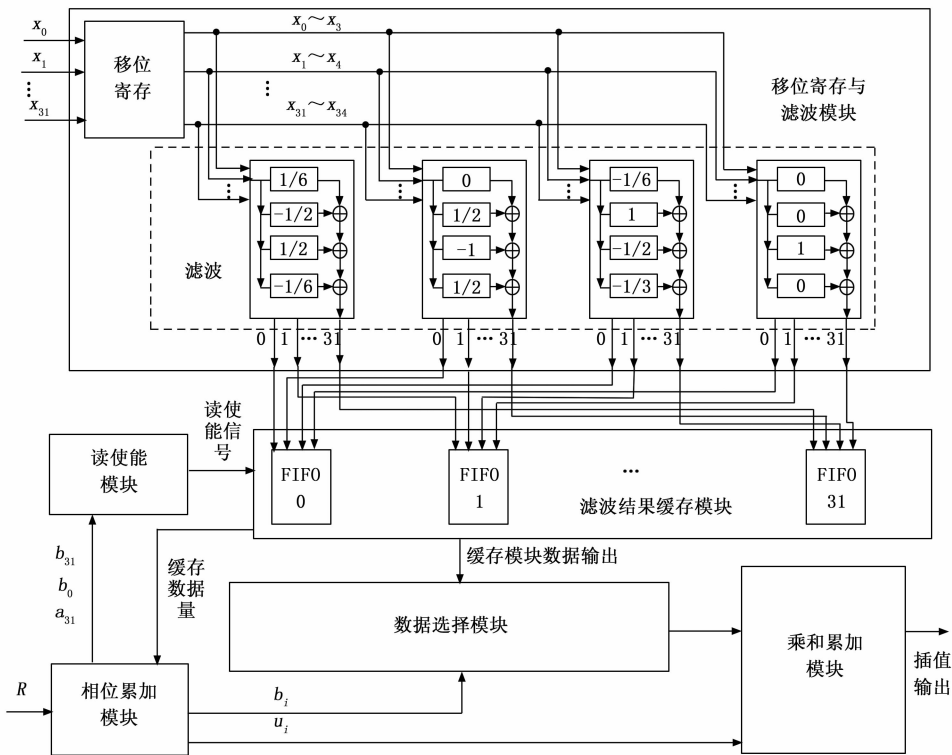


图 5 32 路并行任意倍插值滤波器逻辑架构

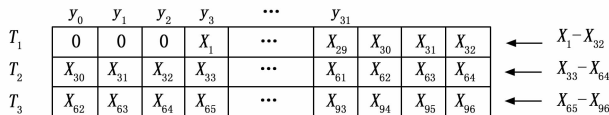


图 6 数据移位寄存

结果存入 32 路 FIFO 中, 缓存模块输出数据给入数据选择模块。读使能模块发出的读使能信号到来时, 更新输出数据。此外, 根据模块第一路 FIFO 中数据量控制移位寄存与滤波模块的工作状态, 设控制滤波模块工作状态的数据量阈值为  $count1$  与  $count2$ ,  $count1 > count2$ , 为了防止 FIFO 数据溢出, 数据量大于  $count1$  时, 移位寄存与滤波模块停止工作, 同时启动时间参数处理模块与乘和累加模块, 从缓存模块中读取数据进行重采样; 重采样数据必须保持, 故时间参数处理模块与乘和累加模块启动之后保持工作状态, 从缓存模块中连续读取数据, 为了防止 FIFO 读空时造成的采样点丢失, 数据量小于  $count2$  时, 使能移位寄存与滤波模块, 向缓存模块中输入数据。

3) 相位累加模块:

相位累加模块控制 32 路 FIFO 读使能信号的更新、缓存模块输出数据选择、采样率转换倍数, 是时间参数处理模块中的关键模块。

32 路并行时将  $n=32 * n+i$  代入式 (9) (10) 中得:

$$m_i(n) = INT(32n \cdot r + i \cdot r) \quad (13)$$

$$m_i(n) + u_i(n) = 32n \cdot r + i \cdot r \quad (14)$$

其中:  $m_i(n)$  为距离插值点最近的原始数据点,  $u_i(n)$  为插值点距离原始数据点的小数倍延迟,  $i=0, 1, 2, \dots, 31$ , 32 路相位累加的初始值分别设为  $0, r, 2r, \dots, 31r$ , 每个时钟周期每一路都累加  $32r$ , 根据式 (13) (14) 可知, 累加结果的整数部分为  $m_i(n)$ , 小数部分为  $u_i(n)$ 。在 FPGA 上实现时, 需要对相位累加结果进行量化处理, 本文采用 36 位宽进行量化,  $0 \sim 29$  位为小数部分  $u_i$ ,  $30 \sim 34$  为整数部分  $b_i$ , 最高位为  $a_i$ , 其中  $0 \leq i \leq 31$ 。量化后 bit 位含义如图 7 所示。



图 7 量化后 bit 位含义

$r$  的量化如式 (15) 所示。现根据图 4 所示时间参数求取流程进行 32 路并行设计, 状态机如图 8 所示, 32 路相位累加器为  $phase[i]$ , 初始值分别为  $0, R, 2R, \dots, 31R$ , 每个时钟判断  $phase[i]$  的最高位值,  $a_i=0$  时,  $phase[i]$  加上  $32R$ ,  $a_i=1$  时,  $phase[i]$  加上  $32R$ , 同时最高位最高位置 0, 即减去 16 进制数  $0x800000000$ 。

$$R = [r \times 2^{30}] \quad (15)$$

由式 (15) 可知, 对  $r$  进行量化, 存在向上取整的操作, 这个过程会产生量化误差, 累加操作不断进行时, 误差不断累积, 导致输出波形丢点或失真。为了降低量化误

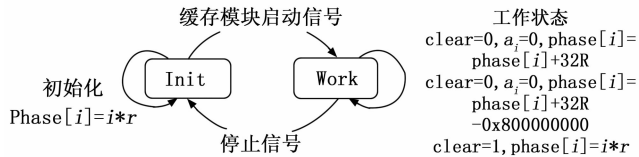


图 8 32 路并行相位累加状态机

差积累造成的影响, 需对累计误差进行清除, 此外, 在清除误差的同时也要保证数据的连续输出, 所以当满足上一时钟的  $b_{31}=31$ 、当前时刻的  $b_0=0$  时, 误差清除信号  $clear$  置 1, 将当前时刻的 32 路相位累加器的值置为  $0, R, 2R, \dots, 31R$ , 并将 32 路 FIFO 输出进行更新, 周期性对累计误差清除, 时钟上升沿到来时, 继续进行工作, 实现输出波形长时间稳定输出。

4) 读使能模块:

读使能模块输出读使能信号  $rd\_en_i(m)$ , 其中  $m$  表示当前时钟周期,  $0 \leq i \leq 31$ ,  $rd\_en_i(m)$  控制着滤波结果缓存模块中 32 路 FIFO 的读使能。读使能的控制方案如表 1 所示, 令  $\Delta'_p = a_{31}(m) - a_{31}(m-1)$ ,  $b_{31}(m-1) = 31$ 、 $b_0(m) = 0$  时, 此时对累计误差进行了清理, 为了保证数据的连续, 同时需对 32 路 FIFO 的输出进行同步更新, 即给 32 路 FIFO 读使能信号, 保证数据的连续性;  $b_{31}(m-1)$ 、 $b_0(m)$  不满足上述条件时, 若  $a_{31}(m-1) = 0$ 、 $a_{31}(m) = 0$ , 此时  $\Delta'_p = 0$ , 32 路 FIFO 不进行读操作, 输出数据不更新; 若  $a_{31}(m) = 1$ , 此时  $\Delta'_p = 1$ , 当前周期对  $0 \sim b_{31}(m)$  路 FIFO 进行更新, 下一时钟周期对  $b_{31}(m) + 1 \sim 31$  路 FIFO 进行数据更新。

表 1 32 路 FIFO 读使能控制

$b_{31}(m-1)b_0(m)$	$a_{31}(m-1)$	$a_{31}(m)$	$rd\_en_i(m)$	$rd\_en_i(m+i)$
31,0	×	×	0~31	×
其他	0	0	×	×
	×	1	0~ $b_{31}(m)$	$b_{31}(m+1 \sim 31)$

5) 数据选择模块:

数据选择模块选择某路 FIFO 的输出数据, 并将数据给到最终的乘和累加模块。在 32 路 FIFO 按照 FIFO 读使能控制信号响应完成后, 根据  $b_i$  ( $0 \leq i \leq 31$ ) 选择第  $b_i$  路 FIFO 输出, 结果给入乘和累加模块。

6) 乘和累加模块:

乘和累加模块完成插值信号的最终输出。该模块的输入为相位累加模块输出的  $u_i$  ( $0 \leq i \leq 31$ ) 和数据选择模块选入的 FIFO 输出数据, FIFO 输出数据为公式 (8) 矩阵  $A$ 、 $B$  相乘后的结果, 乘和累加模块实现该结果与公式 (8) 中矩阵  $C$  相乘的功能, 输出插值结果。乘和累加模块的实现如图 9 所示。

图 9 中,  $u_i$  为  $phase[i]$  的第 30 位,  $FIFO_{b_i}(j)$  为数据选择模块选择的第  $b_i$  路数据输出。硬件实现时  $u_i$  可与  $phase[i]$  同步产生, 但是在 FPGA 逻辑实现时, 从  $phase[i]$

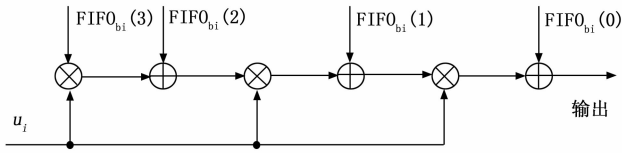


图 9 乘和累加模块计算过程

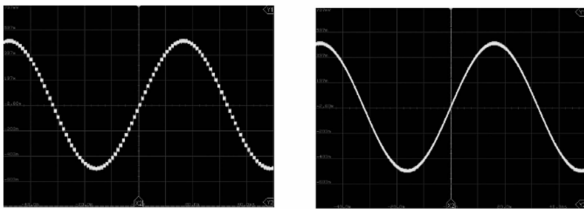
产生到根据  $phase[i]$  实现数据选择模块数据输出有固定周期的延迟, 硬件实现时需保证乘和累加模块的输入数据根据相同的  $phase[i]$  产生, 否则会导致输出波形失真。

对以上 6 个模块进行逻辑实现后, 可以实现固定时钟驱动下的任意倍重采样; 并且因为对累积误差进行了周期性清除, 可以实现长时间波形稳定输出。

#### 4 实验结果与分析

为了验证设计的重采样滤波器的正确性, 以航天测控公司研制的 AMC5425 型-12 GSPS 任意波形发生器产品作为硬件平台进行测试验证, 该平台 DAC 可以实现 6 GSPS 的真实采样率输出, 经过 DAC 内插后实现 12 GSPS 的采样率输出, 可以满足本文设计的采样率转换范围要求。在此任意波形发生器的任意波发生功能中加入任意倍重采样模块, 并通过示波器对输出波形频率、输出波形质量进行测试。

现对设计的采样率转换算法效果进行验证, 为了对比插值前后的效果, 测试波形为离散的正弦波, 波形点数为 64 点, 以 1 ksp/s 等效采样率输出波形, 采样率转换倍数为  $6 \times 10^6$ ,  $R=0 \times B3$ 。算法未使能时, 示波器输出波形如图 10 (a) 所示, 输出波形为台阶状; 算法使能后, 输出波形如图 10 (b) 所示, 输出波形质量较算法未使能时波形有明显提升, 并且验证了该重采样模块可以实现 1 ksp/s 等效采样率输出。

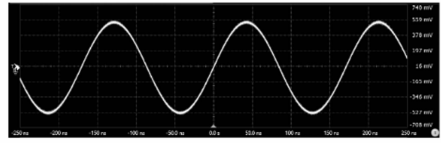


(a) 算法未使能时输出波形 (b) 算法使能后输出波形

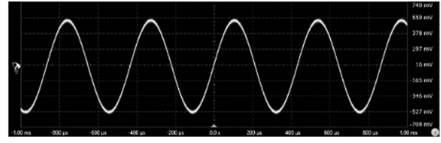
图 10 算法使能前后输出波形对比

为了验证 6 Gsp/s 采样率指标与小数倍插值功能, 测试波形为离散的正弦波, 点数为 1 024 点。直接以 6 Gsp/s 的采样率输出时,  $R=0 \times 40\ 000\ 000$ , 输出波形与波形频谱如图 11 (a) (c) 所示, 示波器显示输出频率为 5.859 8 MHz; 不改变仪器其他工作参数, 使能重采样滤波器, 并将插值倍数设为 2 531.646 倍,  $R=0 \times 678C0$ , 输出波形与波形频谱如图 11 (b) (d) 所示。

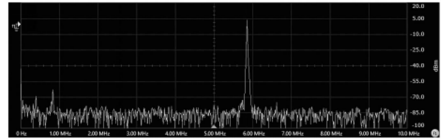
插值倍数为 2 531.646 倍时, 可推算出原始 1 024 点离散正弦波的播放时钟频率等效为 2.37 Msp/s, 输出波形频率为 2.314 4 kHz, 图 11 (d) 表明插值后波形频谱特征正常,



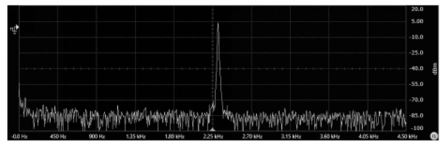
(a) 采样率为 6 Gsp/s 时波形输出



(b) 2 531.646 倍插值后波形输出



(c) 采样率为 6 Gsp/s 时波形频谱



(d) 插值后输出波形频谱

图 11 测试波形与频谱

且输出波形频率与推算结果一致。结果表明该架构可在固定时钟的驱动下, 等效实现任意采样率的采样效果, 并且可以长时间正确输出。

#### 5 结束语

本文基于 Farrow 架构对 32 路并行任意倍插值滤波器进行逻辑设计, 实现了在 187.5 MHz 固定时钟驱动下的任意倍重采样的功能; 对量化误差造成的累加过程中误差不断积累的问题提出了周期性累积误差清除方案。实验结果表明该设计能实现任意倍重采样, 具有很好的可行性与有效性, 达到了预期效果。此外, 该多路并行设计方法也可用于 8 路、16 路、64 路等其他多路并行设计中, 拓展性较好。

#### 参考文献:

- [1] 赵素梅, 张孝飞. 基于 PXIe 接口任意波形发生器幅值补偿的研究 [J]. 信息技术与信息化, 2020 (9): 154-156.
- [2] 陈 富. 基于 DDWS 的任意波形发生器研发分析 [J]. 长江信息通信, 2022, 35 (11): 123-125.
- [3] 董 浩. 雷达干扰仿真与抑制方法研究 [D]. 西安: 西安电子科技大学, 2020.
- [4] YANG Y, SHEN Z, ZHU X, et al. An FPGA-based low latency AWG for superconducting quantum computers [C] //2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2021: 1-6.
- [5] ALPERT T, WERZ M, LANG F, et al. Arbitrary waveform generator based on FPGA and high-speed DAC with real-time interface [C] //PRIME 2012; 8th Conference on Ph. D. Research in Microelectronics & Electronics, 2012: 1-4.

(下转第 276 页)