

IIR 数字滤波器的 FPGA 仿真与实现

许金龙, 苏武浔

(国立华侨大学 信息科学与工程学院, 福建 泉州 362021)

摘要: 采用自顶向下的模块化设计思想, 介绍了一种采用级联结构在 FPGA 上实现 IIR 数字滤波器的设计方案。设计 IIR 数字滤波器的二阶节, 将二阶节 IIR 数字滤波器级联实现高阶 IIR 数字滤波器, 从而实现通过修改外围参数来改变滤波器的频率响应, 根据不同的要求在不同规模的 FPGA 上加以实现。

关键词: IIR 数字滤波器; FPGA; QuartusII; 级联结构

中图分类号: TN911.5

文献标识码: A

Implementation and simulation of IIR digital filter using FPGA

XU Jin Long, SU Wu Xun

(College of Information Science and Engineering, Huaqiao University, Quanzhou 362021, China)

Abstract: In this paper, we used the thought of blocking method named "top-down" and introduced a method of implementing the high-step IIR digital filter on FPGA by using the cascade structure. Designing the two-step IIR digital filter is the first step. Then two-step filters are cascaded to implement high-step IIR digital filter. When in actual usage the filter can change its frequency response by modifying its peripheral parameters and can be implemented on different scale of FPGA according to different request.

Key words: IIR digital filter; FPGA; QuartusII; cascade structure

数字滤波器是实现数字滤波的核心器件。常用的数字滤波器有 FIR 数字滤波器和 IIR 数字滤波器。文章介绍了一种基于 EDA 技术中的模块化设计思想, 采用 VHDL 硬件描述语言对 IIR 数字滤波器中的一些关键电路模块进行设计, 最终在 FPGA 上实现 IIR 数字滤波器的方法^[1]。

1 IIR 数字滤波器硬件设计原理

1.1 数字滤波器的原理

数字滤波器的功能是完成信号滤波处理, 是有有限精度算法实现离散时间线性非时变系统^[2]。一个数字滤波器的系统函数 $H(z)$ 可以表示为:

$$H(z) = \frac{\sum_{i=0}^M a_i z^{-i}}{1 - \sum_{j=1}^N b_{j-1} z^{-j}} \quad (1)$$

直接由 $H(z)$ 得出表示输入输出关系的常系数线性差分方程为:

$$y(n) = \sum_{i=0}^M a_i x(n-i) + \sum_{j=1}^N b_{j-1} y(n-j) \quad (2)$$

式中 a_i 、 b_{j-1} 为滤波系数, 当 b_{j-1} 均为零时, 该滤波器为 FIR 数字滤波器, 当 b_{j-1} 不均为零时, 则为 IIR 数字滤波器。

1.2 IIR 数字滤波器的硬件实现方案比较

滤波器的实现主要包括两大类: 采用 TMS320 系列 DSP 芯片实现和采用 PLD 器件实现(主要包括 FPGA 和 CPLD)。利用 PLD 器件实现时, 可以采用硬件乘加模块, 从而使其运算速度比采用 DSP 器件快很多。因此基于 FPGA 的自编程实现方式成了滤波器实现的首选^[3]。以下简要介绍 IIR 数字滤波器的设计方案和基于 FPGA 的实现方法^[4]。

方案一: 直接相乘累加式

对于二阶的 IIR 数字滤波器, 其差分方程为:

$$y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} + b_0 y_{n-1} + b_1 y_{n-2}$$

因此可用 5 个硬件乘法器和 4 个硬件加法器来实现, 采用这种方法对于高阶的 IIR 数字滤波器的 FPGA 的设计来说是比较耗费资源的。

硬件纵横 Hardware Technique

方案二：基于ROM查找表的VHDL结构化设计^[5]

滤波器实现的主要任务是完成乘累加运算，采用ROM查找表方法可以避免使用硬件乘法器。由二阶IIR滤波器的差分方程可以看出： y_n 仅取决于变量 x_n 、 x_{n-1} 、 x_{n-2} 、 y_{n-1} 和 y_{n-2} ，因此可将 a_0 、 a_1 、 a_2 、 b_0 、 b_1 的所有部分积存储在ROM中，而以变量 x_n 、 x_{n-1} 、 x_{n-2} 、 y_{n-1} 和 y_{n-2} 作为ROM的地址，从中选出对应的值，从而得到 y_n 。

此方法避免了占主要运算量的乘法运算，节省了FPGA硬件资源，但使用不够灵活。特别是当阶数比较大时，更改ROM内的数据十分不便。

方案三：改进型设计

结合直接相乘累加式和ROM查表法的优点，用1个5路8位×1位的乘法器在8个时钟周期内实现。其实现结构如图1所示。

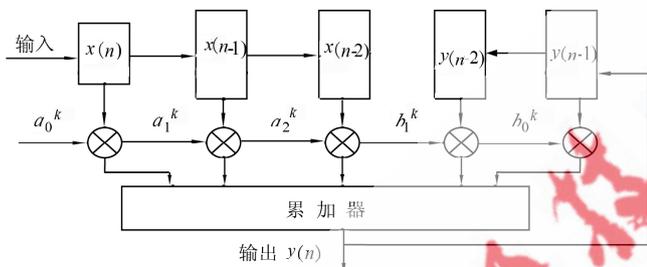


图1 改进型实现框图

图1中的 $X(n)$ 与各自的系数的最高位相乘后，送入累加器相加后将和左移一位，以实现乘2运算。下一个时钟，寄存器内数据与系数的次高位相乘，再送入累加器，然后再左移一位。接下来的6个时钟进行类似的操作。第8个时钟后，将累加器所得数据输出即得到 $y(n)$ ，对累加器清零，接着再进行下一次运算。

比较3种方案，方案三实现较为方便简洁，在节省了FPGA硬件资源的同时，使得设计灵活，设计周期大为缩短，本设计即采用方案三来实现IIR数字滤波器。

2 IIR 数字滤波器的具体实现和仿真

二阶级联法用于实现IIR数字滤波器硬件电路的应用是极为广泛的，为此本文首先实现数字滤波器的二阶节，然后用二阶节的级联实现更高阶的滤波器。

2.1 IIR 数字滤波器二阶节的FPGA实现

IIR数字滤波器的二阶节电路采用模块化设计，把二阶节分为延时模块和补码乘加模块，其中补码乘加模块是设计的核心。

2.1.1 延迟模块

延迟模块功能是在clk时钟作用下将差分方程的各x、y值延迟一个时钟以实现一次延时运算，即当延时模块输入为 $x(n)$ 和 $y(n)$ 时，经过延时后输出 $x(n)$ 、 $x(n-1)$ 、 $x(n-2)$ 、 $y(n-1)$ 和 $y(n-2)$ ，然后将输出送到补码乘加模块运算。

2.1.2 补码乘加模块

补码乘加模块主要是用方案三的方法实现输入序列 $\{x(n)\}$ 、 $\{y(n)\}$ 与系数 a_i 、 b_i 分别相乘后再相加的过程，即实现 $y_n = a_0x_n + a_1x_{n-1} + a_2x_{n-2} + b_0y_{n-1} + b_1y_{n-2}$ 。为了避免过多地使用乘法器，本设计中乘加单元(MAC)的乘法器采用阵列乘法器，以提高运算速度。

2.1.3 二阶节的顶层模块设计

顶层模块设计采用原理图输入方法，将延迟模块和补码乘加模块加入到系统中，并加入相应的输入输出引脚构成IIR滤波器的二阶节。

2.1.4 二阶节的系统仿真与结果分析

本次设计的二阶节的输入端 $x(n)$ 宽为8位，可接8位AD转换输出的数据，输出 $y(n)$ 为16位数据。将实现的二阶节在QuartusII中进行编译综合后，在波形图中分别给定不同的滤波器系数和输入不同的序列时，进行了波形仿真，这里给出其中的一个仿真波形如图2所示(图2是实现系统函数为： $H(z)=(5+5z^{-1}+5z^{-2})/(1-z^{-1}-z^{-2})$ 的二阶节的波形图，其中 a_0 、 a_1 、 a_2 、 b_0 、 b_1 为滤波器系数， x_n 为输入序列， y_n 为输出序列)。

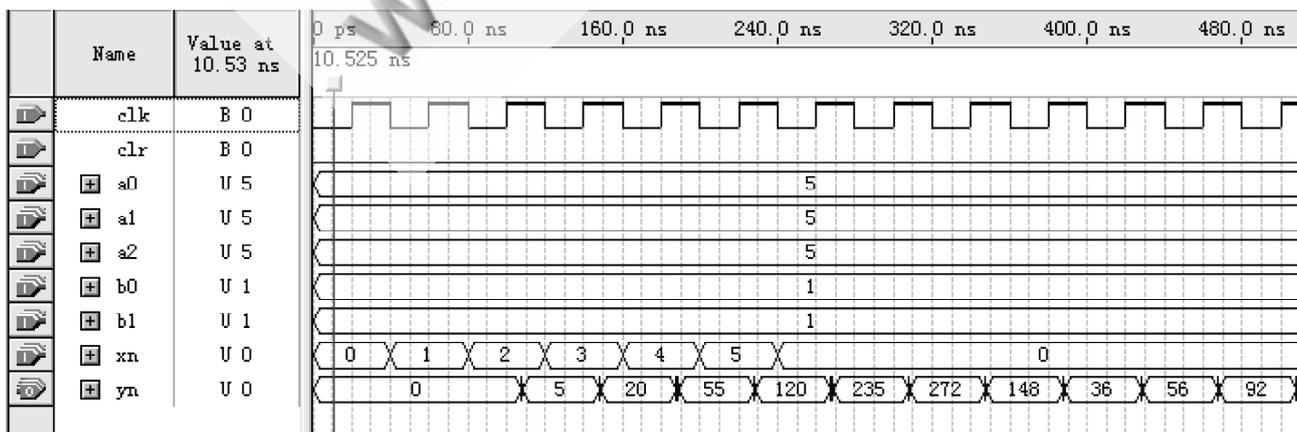


图2 二阶节时序仿真图

硬件纵横 Hardware Technique

下面将对用 FPGA 实现的二阶节时序波形仿真值和用 Matlab 对二阶节的仿真结果相比较，分别对在输入序列相同、但滤波器系数不同和滤波器系数相同、但输入序列不同等几种情况分别进行了比较，比较结果如表 1 所示。

表1 比较结果

输入数据	$x_n = \{1, 2, 3, 4, 5\}; a_0=a_1=a_2=b_0=b_1=1$					
Matlab 仿真值	1	4	11	24	47	80
FPGA 仿真值	1	4	11	24	47	132
输入数据	$x_n = \{1, 2, 3, 4, 5\}; a_0=a_1=a_2=5, b_0=b_1=1$					
Matlab 仿真值	5	20	55	120	235	400
FPGA 仿真值	5	20	55	120	272	148
输入数据	$x_n = \{2, 1, 0, 1, 2\}; a_0=a_1=a_2=5, b_0=b_1=1$					
Matlab 仿真值	10	25	50	85	150	250
FPGA 仿真值	10	25	50	85	122	154

由表 1 可以看出所设计的二阶节在不同系数和不同输入情况下的 FPGA 仿真值和由 Matlab 计算值基本一致，说明采用方案三设计的二阶节达到了设计要求。此外，表中仿真值与计算值有所偏差，而且随着值的增大偏差越来越大，这是由于输入位宽仅为 8 位的有限精度运算所引起的截断误差所致，要解决这一问题可以通过增加二进制位数来提高系统运算精度。

2.2 高阶 IIR 数字滤波器的 FPGA 实现

要实现一个高阶 IIR 数字滤波器，如果采用直接型结构实现，需要的乘法器和延迟单元相对较多，而且分子和分母的系数相差很大，因此需要较多的二进制位数才能实现相应的精度要求。而采用二阶节级联实现，

一方面各基本节的零极点可以很方便地单独进行调整，另一方面可以降低对二进制位数的要求，因此选用级联结构。下面以四阶 IIR 数字滤波器为例来说明高阶滤波器的实现。

2.2.1 四阶 IIR 滤波器的实现

四阶 IIR 数字滤波器需要采用 2 个二阶节级联构成。其系统构成原理图如图 3 所示。图中 busconv 模块为总线转换模块，作用是将 16 位数据转换为 8 位数据输出。

2.2.2 四阶 IIR 滤波器的仿真波形和结果分析

四阶滤波器输入端 $x(n)$ 宽为 8 位，输出 $y(n)$ 为 16 位数据。与二阶节一样将四阶滤波器在 QuartusII 中进行编译综合并进行了时序波形仿真。在仿真中，设定的输入序列 x_n 为 {1,2,3,4,5}，输出为 YN。第 1 个二阶节 IIR2(1) 的系数为： $A00=1, A01=2, A02=1, B00=1, B01=1$ ，第 2 个二阶节 IIR2(2) 系数为： $A10=1, A11=1, A12=2, B10=1, B11=2$ ，即实现的四阶滤波器的系统函数为：

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 - z^{-1} - z^{-2}} \cdot \frac{1 + z^{-1} + 2z^{-2}}{1 - z^{-1} - 2z^{-2}}$$

同样将用 FPGA 实现的四阶 IIR 滤波器时序仿真值与用 Matlab 仿真值进行了比较，如表 2 所示。因为采用二阶节级联，这里仅给出一组比较，其他情况与二阶节相同。

表2 比较结果

输入数据	$x_n = \{1, 2, 3, 4, 5\}$			IIR2(1): $a_{00}=1, a_{01}=2, a_{02}=1, b_{00}=1, b_{01}=1$			
				IIR2(2): $a_{10}=1, a_{11}=1, a_{12}=2, b_{10}=1, b_{11}=2$			
Matlab 仿真值	1	7	30	99	279	450	552
FPGA 仿真值 (YN)	1	7	30	99	279	450	552

由表 2 可见仿真结果与计算结果一致，且在不同的滤波器系数和不同输入序列时所得出的结果与 Matlab 仿真值都一致，说明采用 2 个二阶节级联实现四阶 IIR 数字滤波器达到了设计要求。同样由于二阶节存在有

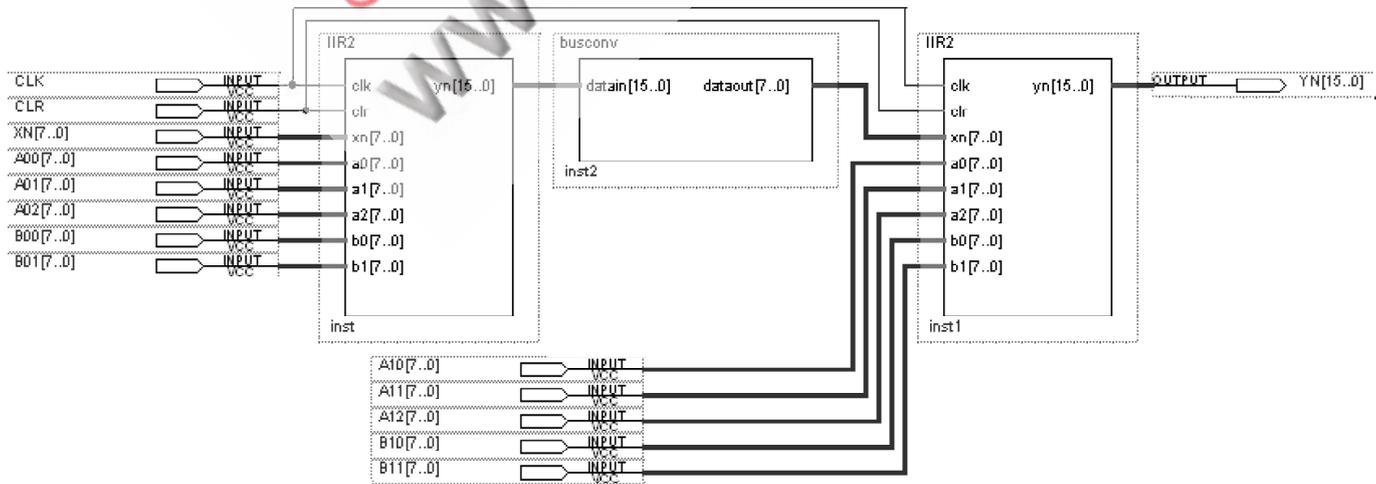


图3 四阶 IIR 数字滤波器原理图

硬件纵横 Hardware Technique

限精度引起的截断误差，四阶滤波器也存在截断误差。

本文采用改进型方法在 FPGA 上成功实现了 IIR 数字滤波器，这种方法优点是不但减少了硬件乘法器的数量，节省了硬件资源，而且避免了系数发生变化时 ROM 中数据更改不方便的问题。缺点是在时间上付出了一定代价。在实际应用中可根据不同的阶数、精度等要求，通过修改 IIR 数字滤波器外围参数改变滤波器的频率响应以实现任意阶数滤波器，并且可以根据不同要求选择在不同规模的 FPGA 上加以实现。

参考文献

[1] UWE M B.数字信号处理的FPGA实现[M]. 刘凌,胡永生译. 北

京:清华大学出版社,2003.

[2] 丁玉美,高西全.数字信号处理[M]. 西安:西安电子科技大学出版社,2004.

[3] 潘松,黄继业. EDA技术实用教程[M]. 北京:科学出版社,2000.

[4] 潘松,王国栋. VHDL实用教程[M]. 成都:电子科技大学出版社,2001.

[5] 李香萍.基于FPGA的IIR低通数字滤波器的实现[J]. 天津师范学院学报,2005,9(03):45-49.

(收稿日期:2009-04-29)