

文章编号: 1672-2892(2010)04-0455-04

一种基于 FPGA 节省资源实现 FIR 滤波器的设计方法

李海军, 王玉萍, 黄耀群

(黑龙江科技学院 电气与信息工程学院, 黑龙江 哈尔滨 150027)

摘要: 有限长脉冲响应(FIR)滤波器的结构决定了用现场可编程门阵列(FPGA)设计非常消耗触发器资源或存储器资源。以 n 阶滤波器为例, 提出一种节省触发器和双口 RAM 的读写操作方法。通信系统应用仿真表明, 与常规 FIR 滤波相比, 本文方法不仅滤波效果良好, 而且大量减少乘法器和加法器数量, 有效节省触发器和双口 RAM。滤波器阶数越高, 越节省资源。

关键词: 滤波器; 现场可编程门阵列; 有限长脉冲响应; 节省资源

中图分类号: TN713⁺.7

文献标识码: A

A method of resources saving to realize FIR filter based on FPGA

LI Hai-jun, WANG Yu-ping, HUANG Yao-qun

(Institute of Electrical and Information Engineering, Heilongjiang Institute of Science and Technology, Harbin Heilongjiang 150027, China)

Abstract: The structure of FIR filter determines that it will consume a great deal of the resources of flip-flop or memory to design the FIR(Finite Impulse Response) filter by using the FPGA(Field Programmable Gate Array). Taking the n -order filters for example, a method of read and write operations to save the resources of flip-flop and dual-port RAM was presented. Comparing with a conventional FIR filter, the application simulation in a communication system showed that the method was not only good for filtering, but also in reducing a number of adders and multipliers, saving flip-flop and dual-port RAM effectively. The higher orders of filter, the more resources were saved.

Key words: filter; FPGA; FIR; resources saving

FIR 滤波器^[1-2]是由大量的延时单元和乘法器组成。用 FPGA 实现时, 延时单元消耗的是触发器或存储器这种时序逻辑资源, 乘法器则消耗的是组合逻辑资源或者专用乘法器资源。不管是延时单元还是乘法器, 由于数量很多, 对 FPGA 而言是一个沉重的资源负担。尤其在 FPGA 工作时钟是 FIR 滤波器输入数据速率 N 倍($N \geq 2$)的情况下, 如果仍采用触发器实现延时单元, 消耗的触发器资源会非常庞大; 即使采用存储器资源来实现延时, 对存储器资源非常少的 FPGA 系列(如 Altera 的 Cyclone 系列, Xilinx 的 Spartan 系列)来说, 也是非常奢侈的做法。因此, 提出一种既能节省触发器资源, 又能节省存储器资源的方法是一件很有意义的事情。

本文提出的方法, 主要是针对高速工作时钟、低速输入数据的情况; 如果工作时钟与输入数据同频, 则没有节省资源的效果。

1 方法描述

1.1 前提

假设 FIR 滤波器阶数为 n , 则滤波器系数有 n 个, 但由于 FIR 滤波器总是对称的($C_1=C_n, C_2=C_{n-1}, \dots$), 因此真正不同的系数只有一半, 即 $n/2$ 个(n 为偶数时)或者 $(n+1)/2$ 个(n 为奇数时)。假设 FPGA 工作时钟频率为 f_c , FIR 滤波器的输入数据速率为 f_d , 且 $f_c=Nf_d$, 其中, $N \geq 2$ 。 N 即是下文经常提到的复用倍数。

定义自然数 k , 当 n 为偶数, 则 $k=\text{ceiling}(n/2/N)$; 当 n 为奇数, 则 $k=\text{ceiling}((n+1)/2/N)$ 。ceiling 运算就是: 除法有余数时, 则运算结果为商加 1; 否则运算结果等于商。

收稿日期: 2010-01-11; 修回日期: 2010-03-15

基金项目: 黑龙江省教育厅面上项目(11541314)

1.2 结构

如图 1 所示。data_in: FIR 滤波器输入数据; data_out: FIR 滤波器输出数据; DPRAM_m: 用于延时的双口 RAM, $m=1,2,\dots,2k$; waddr: 各双口 RAM 的写地址, 各写地址相同; raddr_m: 各双口 RAM 的读地址, $m=1,2,\dots,2k$, 各读地址不同; $C(m)$: FIR 滤波器系数, $m=1,2,\dots,k$ 。

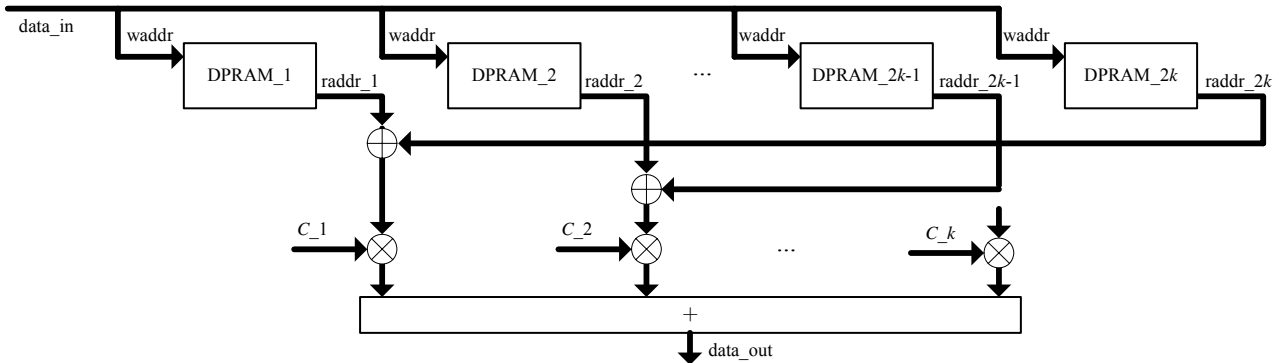


Fig.1 Diagram of FIR filter
图 1 FIR 滤波器结构框图

1.3 实现过程

同其他 FIR 滤波器的设计方法一样, 本方法的设计也分为以下几个步骤: 1) 完成每一阶抽头的延时处理; 2) 合并系数相同的抽头; 3) 合并后抽头数据乘以系数; 4) 合并所有的乘积。

对于后面 3 个步骤, 不同的设计方法基本都一样。通常做法是, 用高速工作时钟将延时后的各抽头数据时分复用起来, 同样将滤波器系数也时分复用; 其次用复用后数据进行同系数合并, 并与复用后的系数相乘; 然后对得到的乘积做积分, 积分长度为复用倍数; 最后将并行的多个积分结果加起来就得到了滤波器输出结果。这样, 所需的乘法器和加法器只是复用前的 $1/N$, N 为复用倍数^[3]。

对于不同的设计方法, 实际区别在于第 1 个步骤。第 1 个步骤的不同实现方法, 决定了该方法所消耗的资源是多还是少。在本文方法中, 第 1 个步骤的实现可以分双口 RAM 的写操作和读操作两个方面来描述。

写双口 RAM 时, 输入数据的速率是工作时钟的 $1/N$, 因此要用写使能来控制。这样需要设置一个复用计数器 counter_mux, counter_mux 的值为 $0 \sim N-1$, 且 0 和 $N-1$ 对应了数据周期的开始和结束。利用复用计数器产生写使能, 只要保证写使能周期与输入数据周期相同即可。同样利用复用计数器产生写地址, 写地址的边界最好与数据边界对齐。如图 2 所示。

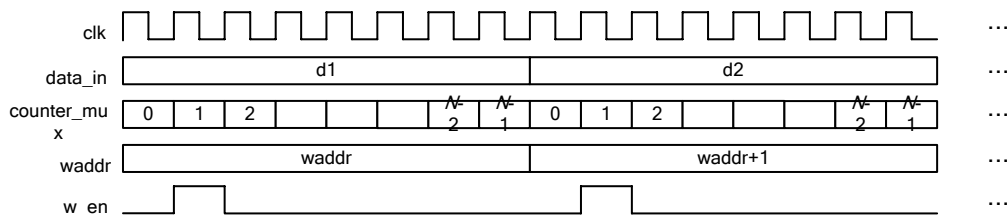


Fig.2 Process of writing dual-port RAM
图 2 写双口 RAM 过程

读地址产生是本方法中最重要的。读操作在本方法中所起的作用有 2 个: 一个是完成数据延时, 另一个是完成数据复用。因此读操作是没有读使能的, 而是一直在读, 也就是说读地址是以工作时钟的频率在变化的。

由于读地址的产生要达到延时数据的目的, 因此读地址要按一定规律滞后于写地址, 即比写地址小一个特定值。又为了达到复用数据的目的, 每个时钟沿到来时, 读地址要加 1, 直到加到 $N-1$ 为止。因此, 读地址可以分为 2 个部分: 即基地址(base)和偏移地址(offset), 读地址是两者之和。基地址是用来完成数据延时的, 偏移地址是用来完成数据复用的。每一个输入数据到来时, 基地址加 1, 而在输入数据持续时间内, 偏移地址由 0 加到 $N-1$ 。对于第一个双口 RAM, 其读地址比写地址滞后一个复用周期, 即地址小 N ; 而之后的每相邻的 2 个双口 RAM, 后一个双口 RAM 的读地址比前一个双口 RAM 的读地址也滞后一个复用周期。

如果滤波器的一半抽头数 $n/2$ 对复用倍数 N 求模等于 M 且 $M \neq 0$ 时, 则滤波器前半段最后一个双口 RAM 只复用 M 个抽头, 而不是前面的双口 RAM 所复用的 N 。

由于滤波器的对称结构, 在滤波器的后半部分考虑读地址时, 要让系数相等的两个抽头同时读出来, 这就要求后半段读地址的读顺序与前半段读地址顺序相反, 且相应的双口 RAM 的复用抽头数相同。如果前半段每个双

口 RAM 的复用抽头数为 N, N, \dots, N, M ，则后半段每个双口 RAM 的复用抽头数为 M, N, \dots, N, N 。

需要注意的是，后半段读地址的基地址产生与前半段有细微差别。如果前半段每个双口 RAM 读地址的基地址比前一块双口 RAM 滞后 N, N, \dots, N, M ，如图 3 所示，则后半段每个双口 RAM 读地址的基地址为：当滤波器阶数 n 为偶数时，后半段每个双口 RAM 读地址的基地址比前一块双口 RAM 滞后 M, N, \dots, N, N ，如图 4 所示；当滤波器阶数 n 为奇数时，后半段每个双口 RAM 读地址的基地址比前一块双口 RAM 滞后 $M-1, N, \dots, N, N$ ，如图 5。

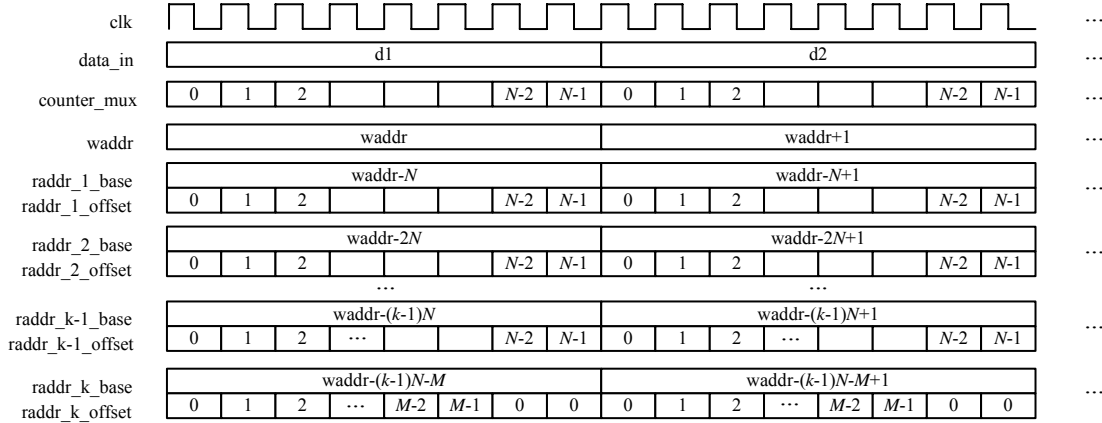
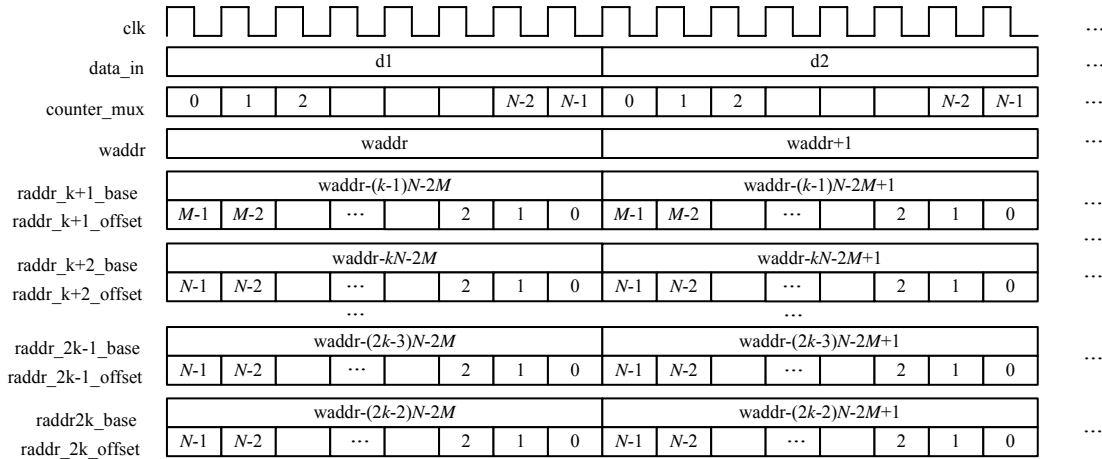


Fig.3 Dual-port RAM read address generation timing of the first half filter

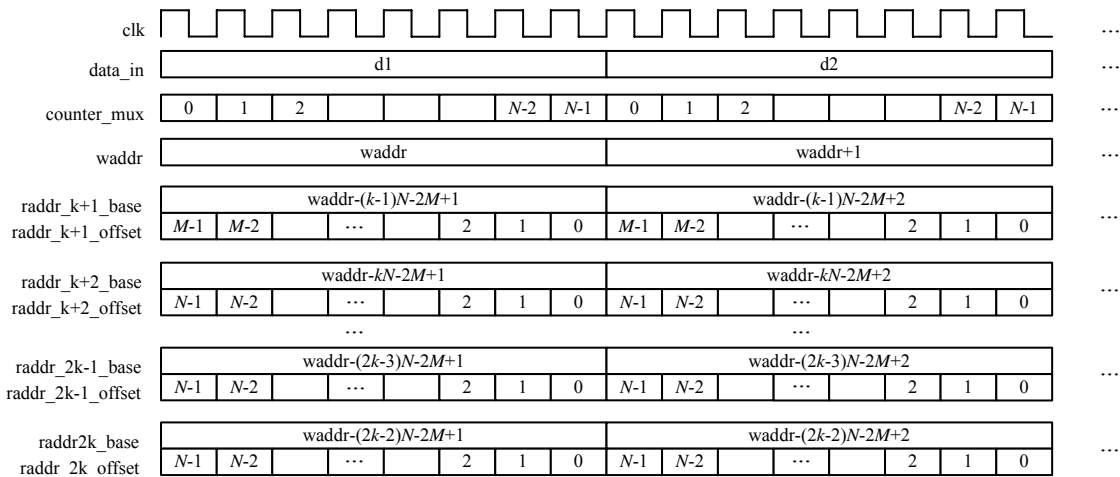
图 3 滤波器前半段各双口 RAM 的读地址产生时序



Note: filter order is an even number

Fig.4 Dual-port RAM read address generation timing of the second half filter

图 4 滤波器后半段各双口 RAM 的读地址产生时序



Note: filter order is an odd number

Fig.5 Dual-port RAM read address generation timing of the second half filter

图 5 滤波器后半段各双口 RAM 的读地址产生时序

根据上述方法产生各 RAM 的读地址, 读出数据, 就可进行同系数抽头合并。但注意在滤波器阶数为奇数时, 最中间的系数对应的抽头数据会多加 1 次; 针对这种情况, 只需单独做一个判断, 然后, 强制只加一次即可。

最后, 需要注意的是, 上述方法读出的数据(合并后)规律是: 先读出的数据延时大, 后读出的数据延时小。正好与经典结构中的顺序相反, 因此相应的系数复用也要将顺序反过来。

2 仿真与实验结果

基于文中描述的设计方法, 实现了波纹 0.2 dB, 带外抑制 75 dB 的 43 阶 FIR 滤波器, 其频率响应如图 6 所示。在通信系统中应用该滤波器, 和采用常规方式实现的 FIR 滤波结果相比, 两者均很好地实现了滤波功能, 滤波后信号频谱和 64QAM 解调结果如图 7 所示。

3 结论

通常在设计数字上变频通道时, 数据总是先插值再滤波, 而插值方法一般为插 0, 这样在滤波时, 对于所有 0 值是不用做乘法和加法的, 因此, 利用这个特点, 可以节省一半以上的乘法器和加法器以及存储单元(触发器或 RAM)^[4]。如果是 2 倍插值, 可以节省一半; 如果是 3 倍插值, 则可节省 2/3。且插值为 3 倍或 3 倍以上时, 本方法在节省乘法器和加法器数量的同时, 可节省大量的触发器和双口 RAM。

参考文献:

- [1] 翟海涛, 杨军, 朱江. 一种基于 FPGA 的高速 FIR 滤波器的设计[J]. 信息化研究, 2009, 35(4): 26-29.
- [2] 薛严冰, 韩雪, 邵远. 基于分布式算法的 FIR 数字滤波器的 FPGA 实现[J]. 大连交通大学学报, 2009, 30(4): 84-88.
- [3] 李伟. FIR 数字滤波器的 FPGA 实现[J]. 计算机与数字工程, 2007, 35(1): 53-55.
- [4] 黄小娟, 王波. 基于 FPGA 的高速数字 FIR 滤波器设计[J]. 火控雷达技术, 2009, 38(1): 87-90.

作者简介:



李海军(1971-), 男, 吉林省扶余市人, 硕士, 副教授, 主要研究方向为电子信息工程. email: lhjlh4@yahoo.com.cn.

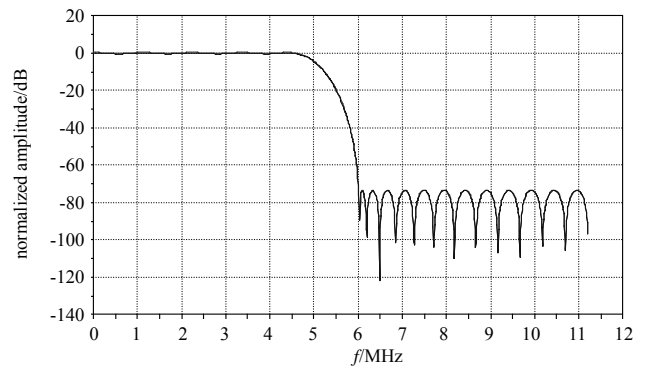


Fig.6 Frequency response of 43-order FIR filter
图 6 43 阶 FIR 滤波器的频率响应

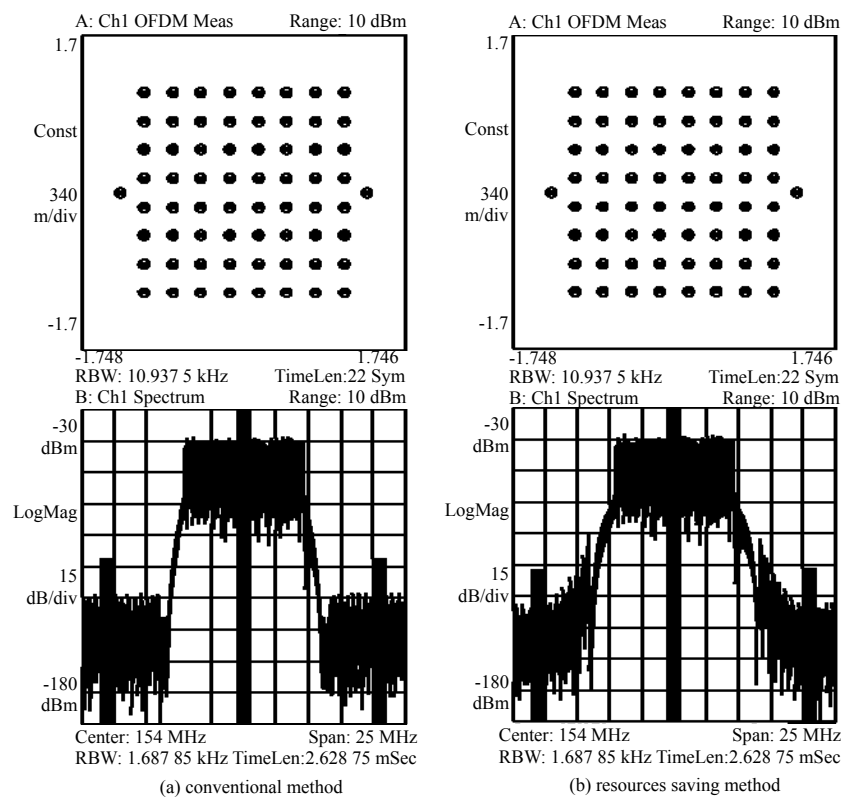


Fig.7 Comparison of the filters
图 7 滤波结果比较

王玉萍(1979-), 女, 黑龙江省鸡西市人, 硕士, 讲师, 主要研究方向为电子信息工程。

黄耀群(1981-), 男, 辽宁省凤城市人, 硕士, 讲师, 主要研究方向为电子信息工程。