西北工业大学

硕士学位论文

基于FPGA的FIR数字滤波器的设计和实现

姓名：刘朋全

申请学位级别：硕士

专业：电路与系统

指导教师：徐建城

20060301

摘 要

在现代电子系统中，FIR数字滤波器以其良好的线性特性被广泛使用，属于数字信号处理的基本模块之一。在工程实践中，往往要求对信号处理要有实时性和灵活性，而已有的一些软件和硬件实现方式则难以同时达到这两方面的要求。随着可编程逻辑器件和EDA技术的发展，使用FPGA来实现FIR滤波器，既具有实时性，又兼顾了一定的灵活性，越来越多的电子工程师采用FPGA器件来实现FIR滤波器。

本文对基于FPGA的FIR数字滤波器实现进行了研究。本论文所做的主要工作如下：

1. 以FIR数字滤波器的基本理论为依据，使用分布式算法为滤波器的硬件实现算法，并对其进行了详细的讨论。针对分布式算法中查找表规模过大的缺点，对其进行了优化方面的讨论，采用多块查找表以及线性FIR滤波器的对称性特点使得硬件规模极大的减小。

2. 针对基于FPGA硬件实现的特点，本论文分别采用了并行和串行的设计方案，分别实现了级联方式实现16阶线性FIR低通滤波器和8阶FIR低通滤波器，并对两种方式进行了性能的比较。在设计中都采用了层次化、模块化的设计思想，将整个滤波器划分为多个功能模块，利用VHDL语言设计方式进行了各个功能模块的设计，最终完成了FIR数字滤波器的系统设计。

3. 设计中采用Xilinx Virtex-II系列器件，通过ISE6.1i软件对两种设计方案进行了综合仿真。为了更好的验证仿真结果的正确性，文中应用了MATLAB和VHDL联合仿真方法对设计的电路进行仿真测试，结果达到设计指标。并用MATLAB对仿真结果进行了分析，证明了所设计的FIR数字滤波器功能正确。

仿真结果表明，本论文设计的滤波器硬件规模较小，是对DA设计方法的一种发展改进，改进后系统最高时钟频率达到了100MHz以上。同时只要将查找表进行相应的改动，就能分别实现低通、高通、带通FIR滤波器，体现了设计的灵活性。

关键词：FIR滤波器 可编程逻辑器件 流水线 分布式算法 查找表

Abstract

In the modern electrical system, the FIR digital filter is used for many practical applications for its good linear phase character, and it provide an important function in digital signal processing design. In engineering practice, there are often real-time and flexible requirements for signal processing .However, software and hardware techniques available for implementation are difficult to meet the demand for the two aspects in the same time. Along with the development of PLD device and EDA technology, more and more electrical engineers use FPGA to implement FIR filter, as it not only meet the real-time requirement, but also has some flexibility.

In this paper, a method to implement the FIR filter using FPGA is proposed. The work is mainly as follows:

1. According to the basic theory of FIR filters, a scheme of hardware implementation is worked out using distributed arithmetic algorithm. As the scale of the LUT in the distributed arithmetic algorithm is so large, the thesis reduces it with the use of multiple coefficient memory banks and the symmetry characteristic of linear FIR filter.

2. According to the characteristics of hardware realization based on FPGA,the parallel and serial scheme are implemented separately in the paper , using the examples of sixteen-tapped lowpass FIR filter with method of eight-tapped cascade and eight-tapped lowpass FIR filter for each, and also the performances of each scheme are compared through the concrete design. From the clew of implementing a stratified, modular design, the thesis describes the hardware design of all functional modules and the FIR system with the VHDL design methods.

3.The design is adopted to the Xilinx Virtex-II serials FPGA, synthesized and simulated with the ISE6.1i. In order to make the verification more available, the complex simulation with Matlab and VHDL is used to testify the design whether fulfills the requirement. And also the result of the simulation is analyzed with the use of MATLAB, and it proved that the function of the design is correct.

The simulation results indicate that the scale of the design is small, which proves that the improvement on DA arithmetic is effective. And the highest system clock can reach 100MHz after the improvement. The design indicates the flexibility at realizing the low-pass, high-pass and band-pass filter easily just through some modifications on FIR filters LUT respectively.

Keywords: FIR filter PLD Pipeline Distributed Arithmetic Look Up Table

第一章 绪 论

1.1 本课题的研究意义

许多工程技术领域都涉及到信号，这些信号包括电的、磁的、机械的、热的、声的、光的及生物体的等等。如何在较强的背景噪声和干扰信号下提取出真正的信号并将其用于实际工程，这正是信号处理要研究解决的问题。20 世纪 60 年代，数字信号处理理论得到迅猛发展，理论体系和框架逐渐趋于成熟，到现在它已经成长为一门独立的数字信号处理学科。数字滤波器在数字信号处理中占有很重要的地位，它涉及的领域很广，如通信系统、系统控制、生物医学工程、机械振动、遥感遥测、地质勘探、航空航天、电力系统、故障检测、自动化仪器等。

数字滤波器的好坏对相关的众多工程技术领域影响很大，一个好的数字滤波器会有效的推动众多工程技术领域的技术改造和学科发展。所以对数字滤波器的工作原理、硬件结构和实现方法进行研究具有一定的意义。

1.2 国内外研究现状

在国内外的研究中，设计 FIR 滤波器所涉及的乘法运算方式有:并行乘法、位串行乘法和采用分布式算法的乘法。

并行乘法虽然速度快，同时占用的硬件资源极大。如果滤波器的阶数增加，乘法器位数也将变大，硬件规模将变得十分庞大。

位串行乘法器的实现方法主要是通过对乘法运算进行分解，用加法器来完成乘法的功能，也即无乘法操作的乘法器。但由于一个 8*8 位的乘法器输出为 16 位，为了得到正确的 16 位结果，串行输入的二进制补码数要进行符号位扩展，即将串行输入的 8 位二进制补码数前补 8 个 0(对正数)或 8 个 1(对负数)后才输入乘法器。如果每一位的运算需要一个时钟周期的话，这个乘法器需要 16 个时钟周期才能计算出正确结果，这就意味着此类乘法器要完全计算出结果的延迟必将会很大。所以位串行乘法器虽然使得乘法器的硬件规模达到了最省，但是由于是

串行运算，使得它的运算周期过长，速度与规模折衷考虑时不是最优的。

分布式算法(distributed arithmetic, DA)的主要特点是巧妙的利用 ROM 查找表将固定系数的乘累加(Multiply-accumulator, MAC)运算转化为查表操作，它与传统算法实现乘累加运算的不同在于执行部分积运算的先后顺序不同。分布式算法在完成乘累加功能时是通过将各输入数据每一对应位产生的部分积预先进行相加形成相应的部分积，然后再对各个部分积累加形成最终结果，而传统算法是等到所有乘积已经产生之后再来相加来完成乘累加运算的。就小位宽来说，DA 算法设计的 FIR 滤波器的速度可以显著的超过基于 MAC 的设计。

相对于前两种方法，DA 算法既可以全并行实现，又可以全串行实现，还可以串并行结合实现，可以在硬件规模和滤波器速度之间作适当的折中，是现在被研究的主要方法。

FIR 数字滤波器的实现，大体可以分为软件实现和硬件实现方法两种。

软件实现方法即是在通用的微型计算机上用软件实现。利用计算机的存储器、运算器和控制器把滤波所要完成的运算编成程序通过计算机来执行，软件可由使用者自己编写，也可以使用现成的。国内外的研究机构、公司已经推出了不同语言的信号滤波处理软件包。但是这种方法速度慢，难以对信号进行实时处理；虽然可以用快速傅立叶变换算法来加快计算速度，但要达到实时处理要付出很高的代价，因而多用于教学与科研。

硬件实现即是设计专门的数字滤波硬件，采用硬件实现的方法一般都比采用软件实现方法要困难得多，目前主要采用的方法有以下几种:

(1)采用 DSP(Digital Signal Processing)处理器来实现

DSP 处理器是专为数字信号处理而设计的，如 TI 公司的 TMS320CX 系列，AD 公司的 ADSP2IX, ADSP210X 系列等。它主要数字运算单元是一个乘累加器(Multiply-accumulator MAC)，能够在一个机器周期内完成一次乘累加运算，配有适合于信号处理的指令，具备独特的循环寻址和倒序寻址能力。这些特点都非常适合数字信号处理中的滤波器设计的有效实现，并且它速度快，成本低，在过去的 20 多年的时间里，软件可编程的 DSP 器件几乎统治了商用数字信号处理硬件的市场。

(2)采用固定功能的专用信号处理器

Circuits)来实现，适用于过程固定而又追求高速的信号处理任务，是以指定的算法来确定它的结构，使用各种随机逻辑器件组成的信号处理器。它们体积小、保密性好，具有极高的性能，然而灵活性差。

二者相比，固定功能的 DSP 专用器件可以提供很好的实时性能，但其灵活性差，研发周期长，难度也比较大:DSP 处理器的成本低且速度较快，灵活性好，但由于软件算法在执行时的顺序性，限制了它在高速和实时系统中的应用。在一些高速应用中，系统性能的要求不断增长，而 DSP 性能的提高却落后于需求的增长。

现在，大规模可编程逻辑器件为数字信号处理提供了一种新的实现方案。分布式算法可以很好地在 FPGA(Field Programmable Gate Array)中实现，然而却不能有效的在 DSP 处理器中实现，所以采用 FPGA 使用分布式算法实现 FIR 数字滤波器有着很好的发展前景。

采用现场可编程门阵列 FPGA 来实现 FIR 数字滤波器，既兼顾 ASIC 器件(固定功能 DSP 专用芯片)的实时性，又具有 DSP 处理器的灵活性。FPGA 和 DSP 技术的结合能够更进一步提高集成度、加快速度和扩展系统功能。用 FPGA 设计的产品还具有体积小、速度快、重量轻、功耗低、可靠性高、仿制困难、上批量成本低等优点。但是，DA 算法中的查找表的规模随着 FIR 数字滤波器阶数的增加呈指数增长，而且随着滤波器系数的位数的增加，查找表的规模也会增加，这将极大的增加设计的硬件规模。所以如何减小查找表的规模成为尚待解决的问题。

1.3 研究思路

1. 要研究基于 FPGA 实现的 FIR 数字滤波器，首先要选定 FPGA 器件。Xilinx 公司的 Virtex-II 系列器件芯片密度大，使用频率高，是目前大规模数字逻辑设计的发展趋势，是用户专用数字滤波器设计的理想载体，并且它的设计软件方便易用，有现成的硬件开发板，所以选用它进行设计。

2. 对 FIR 数字滤波器的结构和设计方法要有一定的了解，会使用 MATLAB 仿真软件设计各种 FIR 滤波器，以便对设计结果进行仿真和比较。

3. 设计数字系统，有多种方法，可以采用传统的数字系统设计方法，也可以

采用使用硬件描述语言的数字系统设计方法。传统的设计方法不适合大规模系统的设计，所以采用使用硬件描述语言的数字系统设计方法。这就要求学会自顶向下的系统设计方法、硬件描述语言 VHDL(Very High Speed Integrated Circuit HardwareDescription Language)，综合工具、仿真工具等。

4. 采用分布式算法实现 FIR 数字滤波器，对分布式算法要有深刻的理解，得出用它来实现 FIR 滤波器的硬件结构,对其实现方式进行研究，分别采用合适的方法来设计，最后利用 FPGA 器件实现 FIR 数字滤波器的硬件电路，并用 Matlab 对实现的结果进行仿真分析。

1.4 本论文所做的主要工作

本论文讨论了利用 FPGA 器件实现 FIR 滤波器的研究过程。所做的主要工作如下:

(1) 学习和研究了采用分布式算法实现 FIR 数字滤波器的设计方法，分析了其不足之处，针对以速度快为最终目标和以硬件规模小为最终目标这两种情况，对各种方案进行了比较，得出速度与规模折衷考虑时最优化的方案，并对其进行了进一步的改进，减小了查找表的规模。

(2) 采用了应用硬件描述语言和自顶向下的数字系统设计方法，对传统的数字系统设计方法和以使用硬件描述语言的数字系统设计方法以及自底向上和自顶向下的数字系统设计方法进行了比较，指出采用硬件描述语言和自顶向下方法设计数字系统的优势，并给出了利用 FPGA 器件进行数字系统设计的设计流程。

(3) 利用硬件描述语言采用自顶向下的数字系统设计方法，给出有限脉冲响应(FIR)数字滤波器的各模块结构，对其进行可综合的寄存器传输级描述。作者将 Virtex-II 系列的结构、功能和特点进行了介绍，并利用 Xilinx 公司的 Virtex-II 系列器件和 ISE6.1i 软件对设计进了综合和仿真，并将实验结果和用 MATLAB 软件计算的结果进行了比较。

(4) 对此次设计作了小结，并提出了进一步对设计进行改进的方法。

随着大规模可编程逻辑器件在数字硬件系统设计中的应用，用 VHDL 语言设计数字系统，逻辑综合和仿真等 EDA 技术将成为对电子工程师的基本要求。

第二章 FIR 数字滤波器及其设计方法

数字滤波器通常都是应用于修正或改变时域或频域中信号的属性。最为普通的数字滤波器就是线性时间不变量（linear time-invariant, LTI）滤波器。LTI 与其输入信号之间相互作用，经过一个称为线性卷积的过程。表示为 $y = f * x$，其中 f 是滤波器的脉冲响应，x 是输入信号，而 y 是卷积输出。线性卷积过程的正式定义如下：

$$y[n] = x[n] * f[n] = \sum_k x[n] f[n-k] = \sum_k x[n-k] f[k]$$

LTI 数字滤波器通常分成有限脉冲响应（finite impulse response, 也就是 FIR）和无限脉冲响应（infinite impulse response, 也就是 IIR）两大类。顾名思义，FIR 滤波器由有限个采样值组成，将上述卷积的数量降低到在每个采用时刻为有限个。而 IIR 滤波器需要执行无限数量次卷积。研究数字滤波器的动机就在于它们正日益成为一种主要的 DSP 操作。数字滤波器正在迅速的代替传统的模拟滤波器，后者是利用 RLC 元器件和运算放大器实现的。模拟原型设计只能应用在 IIR 设计之中，而 FIR 通常采用直接的计算机规范和算法进行分析的。

2.1 FIR 数字滤波器基础

数字滤波器(DF)是个离散系统，它所处理的对象是用序列表示离散信号或数字信号。DF的因果离散系统函数可表示成：

$$H(z) = \frac{\sum_{r=0}^{M} b_r z^{-r}}{1 - \sum_{k=1}^{N} a_k z^{-k}} = \frac{Y(z)}{X(z)} \tag{2-1}$$

其常系数线性差分方程为：

$$y(n) = \sum_{k=1}^{N} a_k y(n-k) + \sum_{r=0}^{M} b_r x(n-r) \tag{2-2}$$

可以看出，数字滤波器是把输入和之前输出的序列经过一定的运算变换成输出的

序列。大多数普通的数字滤波器都是 LTI 滤波器，对于 FIR 系统，其系统函数仅有零点（除 Z=0 的极点外），因此 FIR 系统的差分方程可以表示为：

$$y(n)=\sum_{r=0}^{M}b_r x(n-r) \tag{2-3}$$

转移函数为：

$$H(z)=\sum_{r=0}^{M}b_r z^{-r} \tag{2-4}$$

由(2-3)式可知，系统的脉冲响应是因果序列，因为其输出仅与即时输入以及过去的输入数据有关，而与过去的输出数据没有直接的关系，所以 FIR 滤波器是因果的，是物理可实现的系统，因而它在实际中往往采用非递归(无反馈作用)形式的结构来实现。人们把用非递归形式实现的 FIR 滤波器叫做非递归型滤波器。而且，由(2-3)式还可以知道，此系统的脉冲响应是绝对可加的，所以 FIR 滤波器总是稳定的。

FIR滤波器相对于IIR滤波器有很多独特的优越性，在保证满足滤波器幅频响应的同时，还可以获得严格的线性相位特性。对于非线性FIR滤波器一般可以用IIR滤波器来替代。由于在数据通信、语音信号处理、图像处理以及自适应等领域往往要求信号在传输过程中不允许出现明显得相位失真，而IIR存在明显得频率色散的问题，所以FIR滤波器得到了更广泛的应用。

2.1.1 FIR 数字滤波器的基本结构

FIR 滤波器的构成形式主要有直接型、级联型、线性相位 FIR 滤波器的结构等，下面分别加以讨论。

1. 直接型结构

图 2-1 给出了 N 阶 LTI 型 FIR 滤波器的图解。可以看出 FIR 滤波器是有一个“抽头延迟线”加法器和乘法器的集合构成的。传给每个乘法器的操作数就是一个 FIR 系数，显然也可以称作“抽头权重”。因此该结构也称为“横向滤波器”。

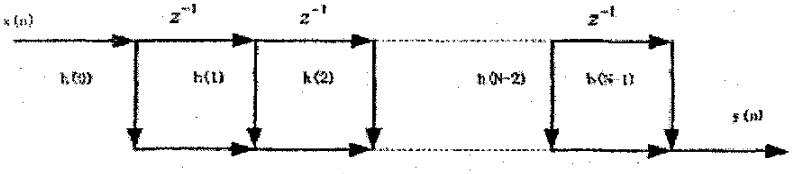


图 2-1　直接型结构的 FIR 滤波器

直接 FIR 模型的一个变种称为转置式 FIR 滤波器，它是根据转置定理定义如果将网络中所有支路的方向倒转，并将输入 x(n)和输出 y(n)互换，则其系统传递函数 H(z)不变。其转置结构见图 2-2.

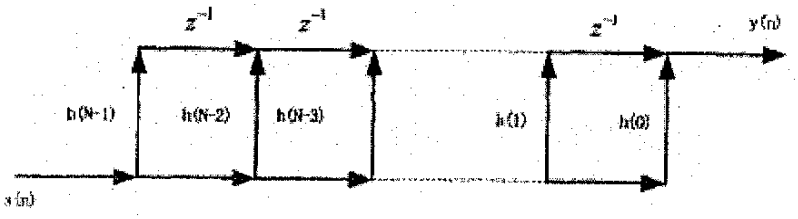


图2-2 转置结构的FIR滤波器

转置式滤波器通常是指 FIR 滤波器的实现。该滤波器的优点在于我们不再需要给 x(n)提供额外的移位寄存器，而且也不必要为达到高吞吐量给乘积的加法器(树)添加额外的流水线级。

2. 级联型

如将(2-4)式分解为二阶实系数因子形式：

$$H(z)=\sum_{n=0}^{N-1}h(n)z^{-n}=\prod_{i=1}^{M}(\beta_{0i}+\beta_{1i}z^{-1}+\beta_{2i}z^{-2})$$

便可得二阶级联结构。这种结构每一节控制一对零点，因而在需要控制传输零点时可以采用。但相应的滤波系数增加，乘法运算次数增加，因此需要较多的存储器，运算时间也比直接型增加。

3. 线性相位 FIR 系统的结构

在许多应用领域，例如通信和图像处理中，在一定频率范围内维持相位的完整性是一个期望的系统属性。因此，设计能够建立线性相位一频率功能的滤波器是必须遵循的规范。系统相位线性度的标准尺度就是“组延迟”，其定义为：

$$\tau(w)=\frac{d\phi(w)}{dw} \qquad (2\text{-}5)$$

完全理想的线性相位滤波器对于一定频率范围的组延迟是一个常数。可以看到如果滤波器是对称或者反对称的，就可以实现线性相位。

线性相位(相移)表示一个系统的相频特性与频率成正比，由于不同频率传输速度都一样，所以信号通过它产生的时间延迟等于常数 k 所以不出现相位失真，对一个数字系统来说，即

$$\varphi(w) = -kw$$

假设一个离散时间系统的幅频特性等于 1，则当信号 $x(n)$ 通过该系统后，其输出 $y(n)$ 的频率特性

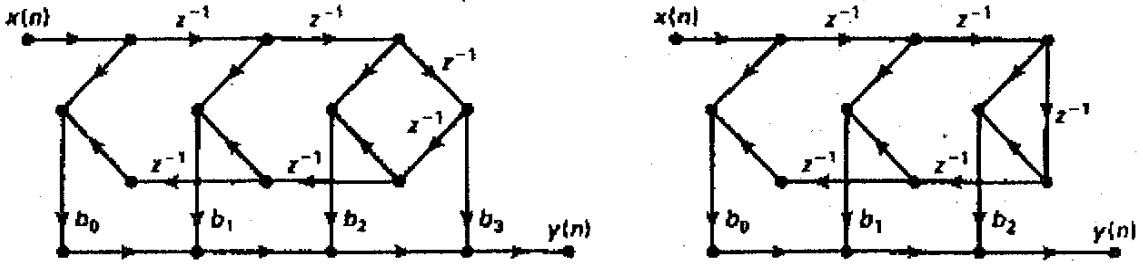
$$Y(e^{jw}) = H(e^{jw})X(e^{jw}) = e^{-jkw}\left|X(e^{jw})\right|e^{j\arg\left[X(e^{jw})\right]} = \left|X(e^{jw})\right|e^{j\arg[X(e^{jw})-kw]}$$

所以 $y(n) = x(n-k)$，这样输出 y(n) 等于输入在时间上的唯一，达到了无失真输出的目的。

可以证明，线性相位条件为：

$$\begin{cases} h(n) = h(N-1-n) \text{偶对称} \\ h(n) = -h(N-1-n) \text{奇对称} \end{cases}$$

即如果单位脉冲响应 h(n) 为实数，且具有偶对称或奇对称性，则 FIR 数字滤波器具有严格的线性相位特性。其对称中心在 $n = \dfrac{N-1}{2}$ 处。当 N 分别为奇数和偶数时，其网络结构可以分别用图 2-3(a)，(b) 的信号流图来实现。由该信号流图可以看出，线性相位结构比图 2-1 的直接实现形式少用 $\dfrac{N-1}{2}$ 个乘法器（或乘法运算）。



(a)　　(b)

图 2-3 线性相位 FIR 滤波器结构

(a)N=7；(b)N=6

2.1.2 FIR 数字滤波器的设计

FIR 滤波器设计方法是以直接逼近所需离散时间系统的频率响应为基础。设计方法包括窗函数法和最优化方法（等同纹波法），其中窗函数方法是设计 FIR 数字滤波器是最常用的方法之一。

1. 窗函数法

任何数字滤波器的频率响应 $H(e^{jw})$ 都是 w 的周期函数，它的傅立叶级数展开式为：

$$H_d(e^{jw}) = \sum_{-\infty}^{+\infty} h_d(n)e^{-jwn} \tag{2-6}$$

其中 $h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{jw})e^{jwn}dw$ (2-7)

傅立叶系数 h(n) 实际上就是数字滤波器的冲击响应，由于 h(n) 可能是无限长序列且为非因果响应，是物理不可实现的。为此要寻找一个因果的 h(n)，在相应的误差准则下最近逼近 $h_d(n)$。窗函数法设计的初衷是使设计的滤波器频率特性 $H(e^{jw})$ 在频域均方误差最小意义下进行逼近，即

$$\varepsilon^2 = \frac{1}{2}\int_{-\pi}^{\pi}\left|H_d(e^{jw}) - H(e^{jw})\right|^2 d\omega = \min$$

窗函数法就是用被称为窗函数的有限加权序列 $G_N(n)$ 来修正式(2-7)，则所需 h(n) 表示为：$h(n) = h_d(n)G_N(n)$ (2-8)

$G_N(n)$ 是有限长序列，当 $n > N-1$ 及 $n < 0$ 时，$G_N(n)$=0，这里我们仅以冲激响应对称即 $h(n) = h(N-1-n)(n = 0,1,2,\ldots N-1)$ 时低通滤波器为例进行说明。低通滤波器的频率响应函数 $H(e^{jw})$ 如下式所示：

$$H(e^{jw}) = e^{-jw(N-1)/2}, 0 \le |w| \le w_c \tag{2-9}$$

在$\omega < |\omega| < \pi$时为 0，其中ω为对抽样频率归一化的角频率，ω_c为归一化截止角频率。利用反傅立叶变换公式求出式(2-9)对应的冲激响应 h(n)为：

$$h_d(n) = \frac{\sin\left[\omega_c\left(n-\frac{N-1}{2}\right)\right]}{\pi\left[n-\left(\frac{N-1}{2}\right)\right]} \qquad (2\text{-}10)$$

几种窗函数及其窗函数选择原则：

设计 FIR 滤波器常用的窗函数有：矩形窗函数、三角窗（Bartlett）函数、汉宁（Hanning）窗函数、海明(Hamming)窗函数、布莱克曼（Blackman）窗函数和凯塞（Kaiser）窗函数，具体性能指标可参看表 2－1。

窗的类型	最大旁瓣幅度（相对值）	过度带宽度	最大逼近误差 $20\log_{10}\delta(dB)$	等效 Kaiser 窗 β
矩形	-13	$4\pi/N$	-21	0
Bartlrtt	-25	$8\pi/N$	-25	1.33
Hanning	-31	$8\pi/N$	-44	3.86
Hamming	-41	$8\pi/N$	-53	4.86
Blackman	-57	$12\pi/N$	-74	7.04

表 2-1　窗函数性能指标比较

窗函数的选择原则是：

（1）　具有较低的旁瓣幅度，尤其是第一旁瓣幅度。

（2）　旁瓣幅度下降速度要快，以利于增加阻带衰减。

（3）　主瓣宽度要窄，以获得较陡的过渡带。

通常上述几点很难同时满足，当选用主瓣宽度较窄时，虽然得到较陡的过渡带，但通带和阻带的波动明显增加；当选用最小的旁瓣幅度时，虽然能得到匀滑的幅度响应和较小的阻带波动，但是过渡带加宽。因此，实际选用的窗函数往往事它们的折中。在保证主瓣宽度达到一定要求的情况下，适当地牺牲主瓣的宽度来换取旁瓣波动的减少。

2. 等同纹波设计方法

窗函数存在某些缺陷。首先，在设计中不能将边缘频率 ω_p 和 ω_s 精确的给定；也就是意味着在设计完成之后无论得到何值都必须接受。其次，不能够同时标定纹波因子 δ_1 和 δ_2；在窗函数设计法上只能设定 $\delta_1 = \delta_2$。最后，近似误差在频带区间上不是均匀分布的，在靠近频带边缘误差愈大，远离频带边缘误差愈小。

一种非常有效的解决这种问题的 FIR 滤波器就是等同纹波 FIR 滤波器。对于线性相位 FIR 滤波器来说，有可能导得一组条件，对这组条件能够证明，在最大近似误差最小化的意义下，这个设计时最优的。具有这种性质的滤波器就称为等同纹波滤波器，因为近似误差在通带和阻带上都是均匀分布的。

等同纹波法通常都是采用 Park-McClellan 迭代方法来实现的，与直接频率法相比，等同纹波设计法的优点在于通频带和抑制代偏差可以分别指定，且实现相同指标的滤波器时所用的滤波器阶数较小。

3. 借助 Matlab 设计 FIR 滤波器

在 Matlab 的 SIGNAL PROCESSING TOOLBOX 中有一个专门的数字滤波器设计软件模块 FDA（Filter Design & Analysis Tool），其功能强大，可以设计多种滤波器，而且可以采用多种方法设计 FIR 滤波器，包括窗函数法和等同纹波法，它使用起来非常直观有效，在输入设计要求和设计方法选择后，计算出各阶系数，并以图形的直观方式显示幅频、相频、冲击响应和零极点图。它还可以把各阶系数以二进制补码的形式导出到文本文件中去，方便了系数的转换。

2.2 FIR 数字滤波器的硬件实现方法及其改进

2.2.1 分布式算法基础

分布式算法(distributed arithmetic, DA)是一项重要的 FPGA 技术，广泛地应用在计算乘积和

$$y=\langle c,x\rangle=\sum_{n=0}^{N-1}c[n]x[n] \tag{2-11}$$

之中。本论文就是采用分布式算法来设计 FIR 滤波器，并对其进行了进一步的改进。

一个线性时不变网络的输出可以用式(2-11)表示，进一步假设系数 c[n]是已知常数，x[n]是变量，无符号 DA 系统假设变量 x[n]的表达式如下：

$$x[n]=\sum_{b=0}^{B-1}x_b[n]*2^b,\quad x_b[n]\in[0,1] \tag{2-12}$$

其中 $x_b[n]$ 表示 x[n]的第 b 位，而 x[n]也就是 x 的第 n 次采样，联立式(2-11)和(2-12)得内积 y 可以表示为

$$\begin{aligned}
y&=\sum_{n=0}^{N-1}c[n]*\sum_{b=0}^{B-1}x_b[n]\cdot 2^b=c[0](x_{B-1}[0]2^{B-1}+x_{B-2}[0]2^{B-2}+\ldots+x_0[0]2^0)+\\
&c[1](x_{B-1}[1]2^{B-1}+x_{B-2}[1]2^{B-2}+\ldots+x_0[1]2^0)+\ldots\\
&+c[N-1](x_{B-1}[N-1]2^{B-1}+x_{B-2}[N-1]2^{B-2}+\ldots+x_0[N-1]2^0)\\
&=(c[0]x_{B-1}[0]+c[1]x_{B-1}[1]+\ldots+c[N-1]x_{B-1}[N-1])2^{B-1}+\\
&(c[0]x_{B-2}[0]+c[1]x_{B-2}[1]+\ldots+c[N-1]x_{B-2}[N-1])2^{B-2}+\\
&\ldots+(c[0]x_0[0]+c[1]x_0[1]+\ldots+c[N-1]x_0[N-1])2^0\\
&=\sum_{b=0}^{B-1}2^b\cdot\sum_{n=0}^{N-1}c[n]\cdot x_b[n]=\sum_{b=0}^{B-1}2^b\cdot\sum_{n=0}^{N-1}f(c[n],x_b[n])
\end{aligned} \tag{2-13}$$

由于我们在实际应用当中，滤波器系数有正有负，处理的是有符号数的补码的形式，对于有符号的 DA 系统，我们采用下面(B+1)位表达式：

$$x[n]=-2^B\cdot x_B(n)+\sum_{b=0}^{B-1}x_b[n]\cdot 2^b \tag{2-14}$$

与（2-13）联立得到输出 y 如下：

$$y=-2^B\cdot f(c[n],x_B[n])+\sum_{b=0}^{B-1}2^b\cdot\sum_{n=0}^{N-1}f(c[n],x_b[n]) \tag{2-15}$$

从上式可以看出分布式算法是一种以实现乘累加运算为目的的运算方法。它与传统算法实现乘累加运算的不同在于执行部分积运算的先后顺序不同。分布式算法在完成乘累加功能时是通过将各输入数据每一对应位产生的部分积预先进行相加形成相应的部分积，然后再对各个部分积累加形成最终结果，而传统算法是等到所有乘积已经产生之后再来相加来完成乘累加运算的。

2.2.2 硬件实现改进的 DA 解决方案

本人针对基本的 DA 概念进行了两项有效的改进，在通过 FPGA 硬件实现的过程中，我们最关心的无非是运行速度和电路的规模这两项因素。下面就针对减小规模和提高速度这两点来进行讨论。

1) 并行方式的分布式算法

该结构改进是以增加额外的 LUT、寄存器和加法器为代价提高了速度。若将式(2-13)中每个括号之间的加法并行执行，即将每个 DA 查找表的输出采用并行的加法，就得到了全并行结构。现将(2-13)式中的每个括号内容改写如下，并缩写为 sum:

$$sum[0]=c[0]x_0[0]+c[1]x_0[1]+\ldots+c[N-1]x_0[N-1] \tag{2-16}$$

同理，则

$$sum[B-1]=c[0]x_{B-1}[0]+c[1]x_{B-1}[1]+\ldots+c[N-1]x_{B-1}[N-1] \tag{2-17}$$

则式 2-13 可以改写为

$$y=sum[0]\cdot 2^0+sum[1]\cdot 2^1+\ldots+sum[B-1]\cdot 2^{B-1} \tag{2-18}$$

对于有符号数的 DA 算法，根据式(2－15)，写成式(2-18)的形式为：

$$y=sum[0]\cdot 2^0+sum[1]\cdot 2^1+\ldots+sum[B-1]\cdot 2^{B-1}-sum[B]\cdot 2^B \tag{2-19}$$

利用式(2-19)可得一种直观的加法器树。一个 N 阶系数，采样值为 4 位的 FIR 滤波器的全并行实现如图 2-4 所示，虚线为流水线寄存器。

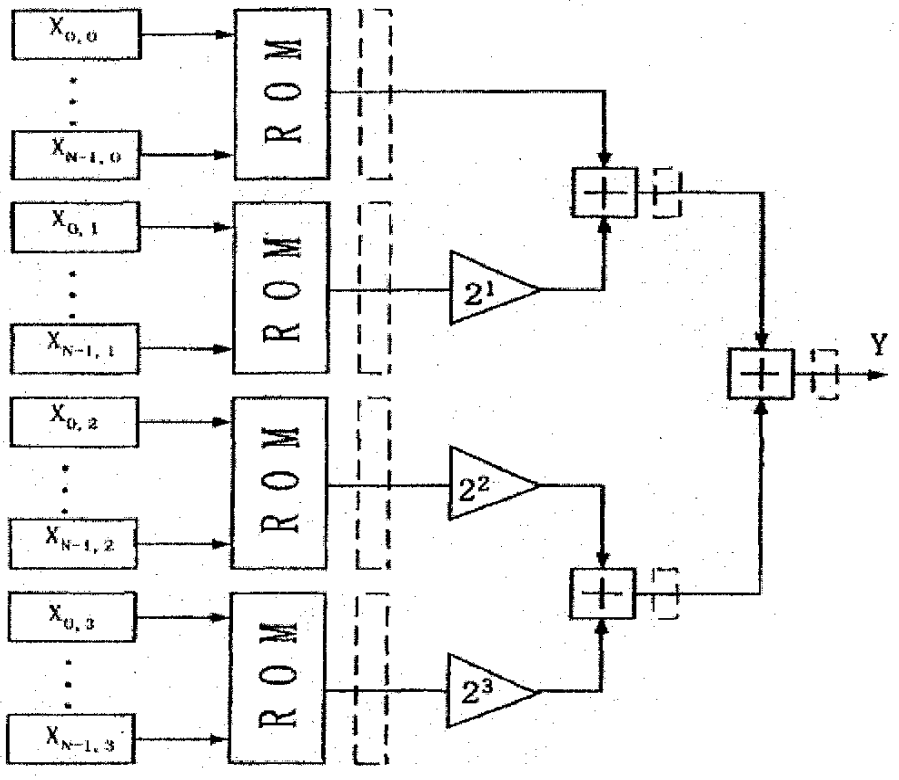


图 2-4 全并行 DA 结构

2) 全串行形式的分布式算法

当对系统速度的要求不太高时，可以采用全串行的设计方法，即一个 DA 查找表，一个并行运算的可控加减法器，以及简单少量的寄存器就可达到目的。这样可以节省大量的 FPGA 资源。串行方法，顾名思义就是输入数据是以串行的方式输入。先从最低位开始，用所有 N 个输入量的最低位对 DA 查找表进行寻址，得到了一个部分积，将其右移一位即将其乘以 2^{-1} 后，放到寄存器当中，同时，N 个输入量的次低位已经开始对 DA 查找表寻址得到另一个部分积，与右移一位后的上一个部分积相加，再重复上一步，直到所有的位数都已经寻址一遍。特别要注意，在最高位寻址得到的值不是与上一个右移一位后的部分积相加，而是相减。这样最后得到的值就是我们需要的结果，由此可以得到全串行 DA 模式。由上可知，完成一次运算需要 B+1 个时钟周期。它的实现框图如下(虚线为流水线寄存器)：

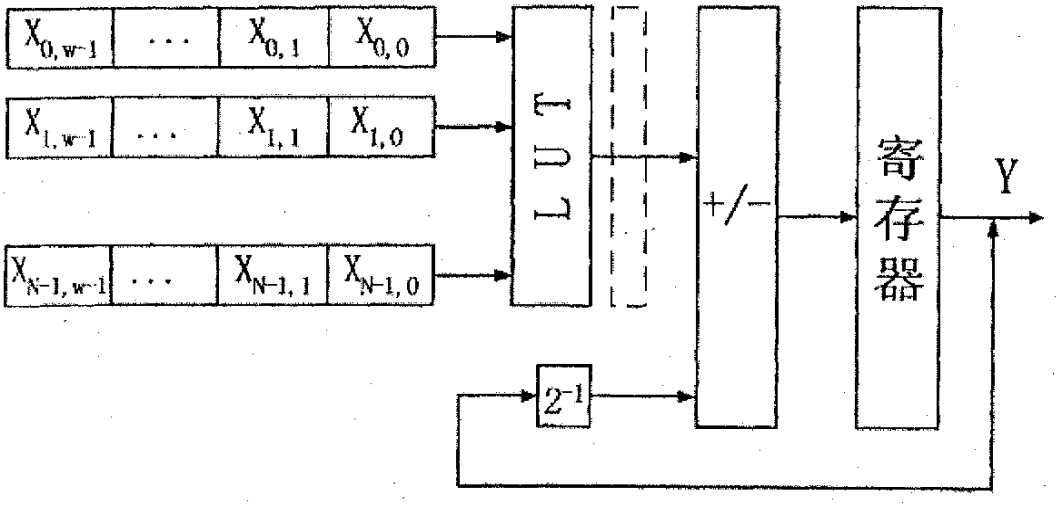


图 2-5 全串行实现方式

3) 表分割简化规模 DA

如果系数 N 过多，用单个 LUT 不能执行全字（输入 LUT 位宽＝系数的数量），就可以利用部分表并将结果相加。如果我们在加上流水线寄存器，这一改进并没有降低速度，却可以极大地减小设计规模，因为 LUT 的规模随着地址空间，也就是输入系数 N 的增加而指数增加。假定长度为 LN 的内积：

$$y=\langle c,x\rangle=\sum_{n=0}^{LN-1}c[n]x[n] \tag{2-20}$$

可以用一个 DA 结构实现。将和分配到 L 个独立的 N 阶并行 DA 的 LUT 中，结果如下：

$$y=\langle c,x\rangle=\sum_{l=0}^{L-1}\sum_{n=0}^{N-1}c[Ll+n]x[Ll+n] \tag{2-21}$$

如图 2-6 所示，实现一个 4N 地 DA 设计需要三个辅助加法器。表格的规模从一个 $4N\times 2^{B}$ 的 LUT 降低到 4 个 $N\times 2^{B}$ 表。

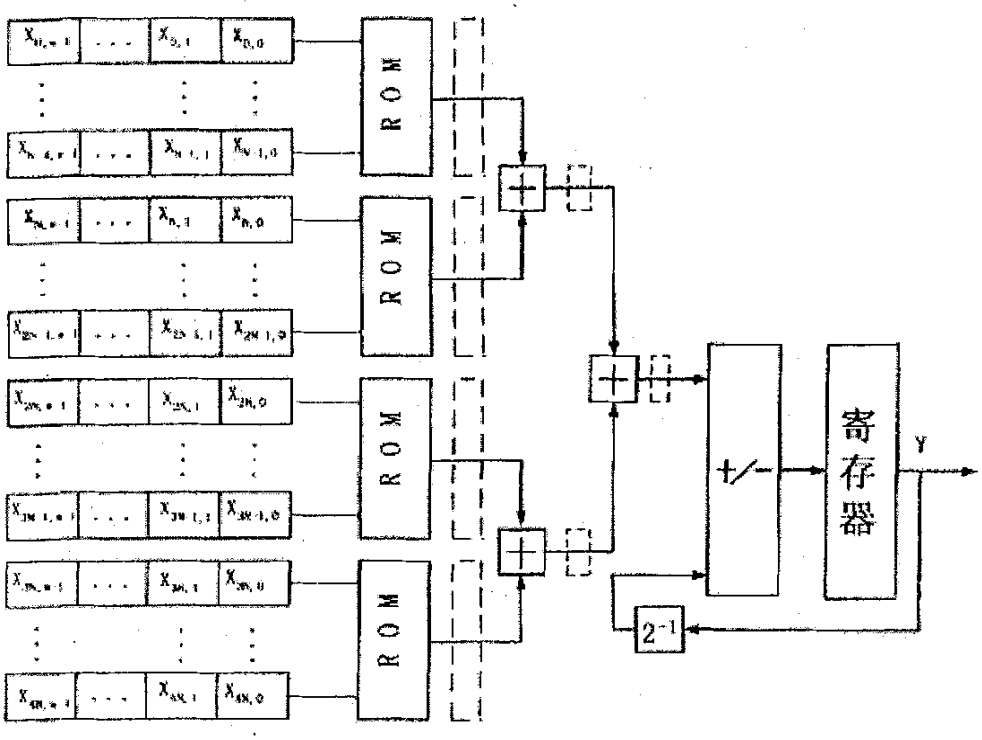


图 2-6 将表分割以产生简化规模的分布式算法

第三章 EDA 技术和可编程逻辑器件

3.1 电子设计自动化 EDA 技术

EDA 是 Electronic Design Automation 的缩写，意为电子设计自动化，即利用计算机自动完成电子系统的设计。EDA 技术是以计算机和微电子技术为先导，汇集了计算机图形学、拓扑、逻辑学、微电子工艺与结构学和计算数学等多种计算机应用学科最新成果的先进技术。

它与电子技术、微电子技术的发展密切相关，吸收了计算机领域的大多数最新研究成果，以高性能的计算机作为工作工具，在 EDA 软件平台上，根据硬件描述语言 HDL 完成的设计文件，自动地完成逻辑编译、化简、分割、综合及优化、布线、仿真，直至对于特定目标芯片的适配编译、逻辑映射和编程下载等工作。

回顾近 30 年的电子设计技术的发展历程，可将 EDA 技术分为三个阶段:

七十年代为 CAD(Computer Aide Design)阶段。这个阶段主要分别研制了一个个单独的软件工具，主要有电路模拟、逻辑模拟、版图编辑、PCB 布局布线等，通过计算机的使用，从而可以把设计人员从大量繁琐、重复的计算和绘图工作中解脱出来。其核心是电路 CAD 技术，产生了计算机辅助设计概念。但这样的设计过程存在两个方面的问题:第一，由于各个工具软件是由不同的公司和专家开发的，只解决一个领域的问题，若将一个工具软件的输出作为另一个工具软件的输入，就需要人工处理，过程很繁琐，影响了设计速度;第二，对于复杂电子系统的设计，当时的 EDA 工具由于缺乏系统级的设计考虑，不能提供系统级的仿真与综合，设计错误如果在开发后期才被发现，将给修改工作带来极大的不便。

八十年代为 CAE 阶段。这个阶段在集成电路与电子系统方法学，以及设计工具集成方面取得了众多成果，与 CAD 相比，除了纯粹的图形绘制功能外，又增加了电路功能设计和结构设计，并且通过电气连接网络表将两者结合在一起，实现了工程设计。由于采用了统一数据管理技术，因而能够将各个工具集成为一个 CAE(Computer Aided Engineering)系统。CAE 的主要功能是:原理图输入，逻辑仿真，电路分析，自动布局布线，PCB 后分析等。这个阶段主要采用基于单

元库的半定制设计方法，采用门阵列和标准单元设计的各种 ASIC 得到了极大的发展。将集成电路工业推入了 ASIC 时代。

九十年代为 EDA 阶段，尽管 CAD/CAE 技术取得了巨大的成功，但并没有把人从繁重的设计工作中彻底解放出来。在整个设计过程中，自动化和智能化程度还不高，各种 EDA 软件界面千差万别，学习实用困难，并且互不兼容，直接影响到设计环节间的衔接。基于以上不足，人们开始追求贯穿整个设计过程的自动化，即电子系统设计自动化。此阶段出现了以高级语言描述、系统仿真和综合技术为特征的第三代 EDA 技术，不仅大大提高了系统的设计效率，而且使设计人员摆脱了大量的辅助性及基础性工作，将精力集中于创造性的方案与概念的构思上。

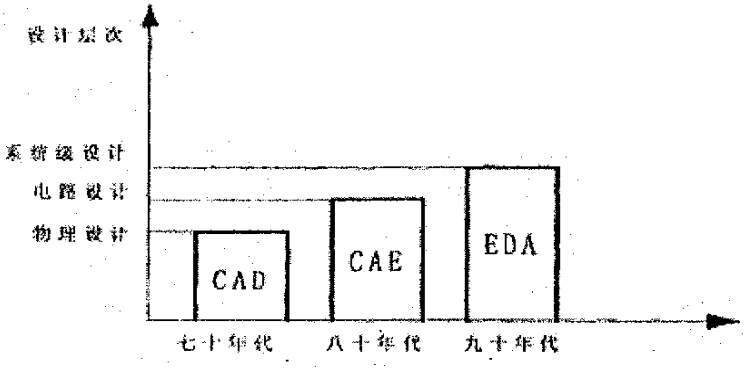


图 3-1 EDA 技术发展层次的变化

3.2 可编程逻辑器件

可编程逻辑器 PLD(Programmable Logic Devices)是 ASIC(Application SpecificIntegrated Circuits)的一个重要分支。ASIC 按制造方法又可分为全定制(Full Custom)产品、半定制(Semi-custom)产品和可编程逻辑器件(PLD)。前两种 ASIC 的设计和制造都离不开器件生产厂家，用户主动性较差。随着微电子技术的发展，设计师们更愿意自己设计专用集成电路芯片，并尽可能缩短设计周期，最好是在实验室里就能设计出合适的 ASIC 芯片，并且立即投入实际应用之中，在使用中也能比较方便的对设计进行修改。可编程逻辑器件就是为满足用户的这一需求应运而生的。

3.2.1 可编程逻辑器件简介

PLD 从 20 世纪 70 年代发展到现在，已形成了许多类型的产品，其结构、工艺、集成度、速度和性能都在不断的改进和提高。PLD 又可分为简单低密度 PLD 和复杂高密度 PLD。

可编程阵列逻辑器件 PAL 以（Programmable Array Logic)和通用阵列逻辑器件 GAL(Generic Array Logic)都属于简单 PLD，结构简单，设计灵活，对开发软件的要求低，但规模小，难以实现复杂的逻辑功能。随着技术的发展，简单 PLD 在集成度和性能方面的局限性也暴露出来。其寄存器、I/O 引脚、时钟资源的数目有限，没有内部互连，因此包括复杂可编程逻辑器件 CPLD(Complex PLD)和现场可编程门阵列器件 FPGA(Field Programmable Gate Array)在内的复杂 PLD 迅速发展起来，并向着高密度、高速度、低功耗以及结构体系更灵活、适用范围更广阔的方向发展。

FPGA 具备阵列型 PLD 的特点，结构又类似掩膜可编程门阵列，因而具有更高的集成度和更强大的逻辑实现功能，使设计变得更加灵活和易实现。相对于 CPLD,它还可以将配置数据存储在片外的 EPROM 或者计算机上，设计人员可以控制加载过程，在现场修改器件的逻辑功能，即所谓的现场可编程。所以 FPGA 得到了更普遍的应用。

3.2.2 使用 FPGA 器件进行开发的优点

使用 FPGA 器件设计数字电路，不仅可以简化设计过程，而且可以降低整个系统的体积和成本，增加系统的可靠性。它们无需花费传统意义下制造集成电路所需大量时间和精力，避免了投资风险，成为电子器件行业中发展最快的一族。使用 FPGA 器件设计数字系统电路的主要优点如下:

1.设计灵活

使用 FPGA 器件，可不受标准系列器件在逻辑功能上的限制。而且修改逻辑可在系统设计和使用过程的任一阶段中进行，并且只须通过对所用的 FPGA 器件进行重新编程即可完成，给系统设计提供了很大的灵活性。

2.增大功能密集度

功能密集度是指在给定的空间能集成的逻辑功能数量。可编程逻辑芯片内的组件门数高，一片 FPGA 可代替几片、几十片乃至上百片中小规模的数字集成电路芯片。用 FPGA 器件实现数字系统时用的芯片数量少，从而减少芯片的使用数目，减少印刷线路板面积和印刷线路板数目，最终导致系统规模的全面缩减。

3.提高可靠性

减少芯片和印刷板数目，不仅能缩小系统规模，而且它还极大的提高了系统的可靠性。具有较高集成度的系统比用许多低集成度的标准组件设计的相同系统具有高得多的可靠性。使用 FPGA 器件减少了实现系统所需要的芯片数目，在印刷线路板上的引线以及焊点数量也随之减少，所以系统的可靠性得以提高。

4.缩短设计周期

由于 FPGA 器件的可编程性和灵活性，用它来设计一个系统所需时间比传统方法大为缩短。FPGA 器件集成度高，使用时印刷线路板电路布局布线简单。同时，在样机设计成功后，由于开发工具先进，自动化程度高，对其进行逻辑修改也十分简便迅速。因此，使用 FPGA 器件可大大缩短系统的设计周期，加快产品投放市场的速度，提高产品的竞争能力。

5.工作速度快

FPGA/CPLD 器件的工作速度快，一般可以达到几百兆赫兹，远远大于 DSP 器件。同时，使用 FPGA 器件后实现系统所需要的电路级数又少，因而整个系统的工作速度会得到提高。

6.增加系统的保密性能

很多 FPGA 器件都具有加密功能，在系统中广泛的使用 FPGA 器件可以有效防止产品被他人非法仿制。

7.降低成本

使用 FPGA 器件实现数字系统设计时，如果仅从器件本身的价格考虑，有时还看不出来它的优势，但是影响系统成本的因素是多方面的，综合考虑，使用 FPGA 的成本优越性是很明显的。首先，使用 FPGA 器件修改设计方便，设计周期缩短，使系统的研制开发费用降低;其次，FPGA 器件可使印刷线路板面积和需要的插件减少，从而使系统的制造费用降低;再次，使用 FPGA 器件能使系统的可靠性提高，维修工作量减少，进而使系统的维修服务费用降低。总之，使用

FPGA 器件进行系统设计能节约成本。

3.2.3 FPGA 设计的开发流程

FPGA 设计的开发流程如图 3-2 所示。

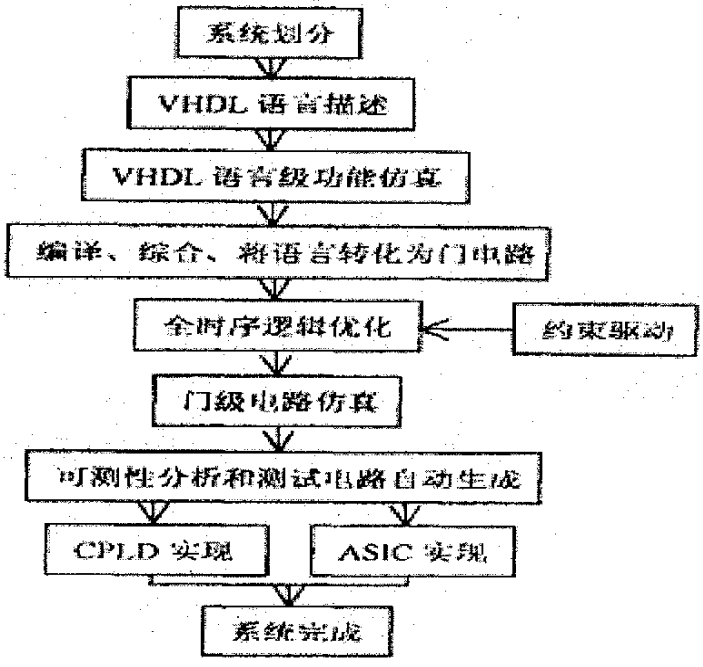


图 3-2 FPGA 开发流程图

设计开始需利用 EDA 工具的文本或图形编辑器将设计者的设计意图用文本方式(如 VHDL 程序)或图形方式(原理图、状态图等)表达出来。完成设计描述后即可通过编译器进行排错编译，变成特定的文本格式，为下一步的综合做准备。在此，对于多数的 EDA 软件来说，最初的设计究竟采用哪一种输入形式是可选的，也可混合使用。

编译形成标准 VHDL 文件后，在综合前即可以对所描述的内容进行功能仿真，又可称为前仿真。即将源程序直接送到 VHDL 仿真器中仿真。功能仿真仅对设计描述的逻辑功能进行测试模拟，以了解其实现的功能是否满足原设计的要求，由于此时的仿真只是根据 VHDL 的语义进行的，与具体电路没有关系，仿真过程不涉及具体器件的硬件特性，如延迟特性。

设计的第三步是综合，将软件设计与硬件的可实现性挂钩，这是软件化为硬件电路的关键步骤。综合后，可生成 VHDL 网表文件，利用网表文件进行综合

后仿真。综合后仿真虽然比功能仿真精确一些，但是只能估计门延时，而不能估计线延时，仿真结果与布线后的实际情况还有一定的差距，并不十分准确。这种仿真的主要目的在于检查综合器的综合结果是否与设计输入一致。

综合通过后必须利用 FPGA 布局/布线适配器将综合后的网表文件针对某一具体的目标器件进行逻辑映射操作，这个过程叫做实现过程。

布局布线后应进行时序仿真。时序仿真中应将布局布线后的时延文件反标到设计中，使仿真既包含门时延，又包含线时延的信息。由于不同器件的内部延时不一样，不同的布局布线方案也给时延造成不同的影响，因此在设计处理完以后，对系统各个模块进行时序仿真，分析其时序关系，估计设计的性能，以及检查和消除竞争冒险是非常有必要的。与前面各种仿真相比，这种仿真包含的时延信息最为全面、准确，能较好地反映芯片的实际工作情况。

如果以上的所有过程，包括编译、综合、布线/适配和功能仿真、综合后仿真、时序仿真都没有发现问题，即满足原设计要求，就可以将适配器产生的配置/下载文件通过编程器或下载电缆载入目标新片中。

3.3 硬件描述语言 VHDL 及数字系统设计方法

可编程逻辑器件和EDA技术给今天的硬件系统设计者提供了强有力的工具，使得数字系统的设计方法发生了质的变化。传统的采用原理图的设计方法正逐步的退出历史舞台，而基于硬件描述语言的设计方法正在成为数字系统设计的主流。同时数字系统的设计方法也由过去的那种由集成电路厂家提供通用芯片，整机系统用户采用这些芯片组成电子系统的“Bottom-up”(自底向上)设计方法改变为一种新的"Top-down"(自顶向下)设计方法。

3.3.1 硬件描述语言 VHDL 简介

硬件描述语言 VHDL(Very High Speed Integrated Circuit Hardware Description Language)是一种用于设计硬件电子系统的计算机语言，它用软件编程的方式来描述电子系统的逻辑功能、电路结构和连接形式。与传统的门级描述方式相比，它更适合于大规模集成电路系统的设计。

VHDL 一种全方位的硬件描述语言，包括系统行为级、寄存器传输级和逻辑门级多个设计层次，支持结构、数据流、行为三种描述形式的混合描述，因此 VHDL 几乎覆盖了以往各种硬件描述语言的功能。通常整个自顶向下或自底向上的电路设计过程都可以用 VHDL 来完成。

VHDL 主要用于描述数字系统的结构、行为、功能和接口，非常适用于可编程逻辑芯片的应用设计。与其它的 HDL 相比，VHDL 具有更强大的行为描述能力，从而决定了它称为系统设计领域最佳的硬件描述语言。强大的行为描述能力是避开具体的器件结构，从逻辑行为上描述和设计大规模电子系统的重要保证。

VHDL 语言在硬件设计领域的作用将与 C 和 C++在软件设计领域的作用一样，在大规模数字系统的设计中，它将逐步取代如逻辑状态表和逻辑电路图等级别较低的繁琐的硬件描述方法，而成为主要的硬件描述工具，它将成为数字系统设计领域中所有技术人员必须掌握的一种语言。VHDL 和可编程逻辑器件的结合作为一种强有力的设计方式，将为设计者的产品上市带来创纪录的速度。

3.3.2 利用硬件描述语言 VHDL 设计数字系统

利用 VHDL 语言设计数字系统硬件电路，与传统的数字系统硬件设计方法相比，具有以下优点：

1. 采用自顶向下(TOP-DOWN)的设计方法

自顶向下是指从系统总体要求出发，在顶层进行功能方框图的划分和结构设计。在方框图一级进行仿真、纠错，并用硬件描述语言对高层次的系统行为进行描述，在系统一级进行验证。然后利用综合优化工具生成具体门电路的网表，其对应的物理实现级可以是 FPGA 电路或专用集成电路。由于设计的主要仿真和调试过程是在高层次上完成的，这一方面有利于早期发现结构设计上的失误，避免设计工作的浪费，同时减少了逻辑功能仿真的工作量，提高了设计的一次成功率。

2.电路设计更趋合理

硬件设计人员在设计硬件电路时使用 PLD 器件，就可以自行设计所需的专用功能模块，而无需受通用元器件的限制，从而使电路设计更趋合理，其体积和功耗也可大为减小。

3.降低了硬件电路的设计难度

在使用 VHDL 语言设计硬件电路时，可以免除编写逻辑表达式或真值表的过程，使得设计难度大大下降，从而也缩短了设计周期。

4.主要设计文件是用 VHDL 语言编写的源程序

在传统的硬件电路设计中，最后形成的主要文件是电路原理图，而采用 VHDL 语言设计系统硬件电路时主要的设计文件是 VHDL 语言编写的源程序。如果需要也可以转化成电路原理图输出。

5.VHDL 语言可以与工艺无关编程

在用 VHDL 语言设计系统硬件时，没有嵌入与工艺有关的信息，其综合生成的是一种标准的电子设计互换格式文件，它独立于采用的实现工艺。有关工艺参数的描述可通过 VHDL 语言提供的属性包括进去，然后利用不同厂家的布局布线工具，将设计映射成不同工艺，在不同的芯片上实现。这使得工程师在功能设计、编辑、验证阶段，可以不必过多地考虑工艺实现的具体细节。

6.方便 ASIC 移植

VHDL 语言的效率之一，就是如果你的设计是被综合到一个 FPGA 或 CPLD 的话，则可以使你设计的产品以最快的速度上市。当产品的产量达到相当的数量时，采用 VHDL 进行的设计很容易转换成专用集成电路来实现，仅仅需要更换不同的库重新进行综合。由于 VHDL 是一个成熟的硬件描述语言，可以确保 ASIC 厂商交付优良品质的器件产品。此外，由于工艺技术的进步，需要采用更先进的工艺时，仍可以采用原来的 VHDL 代码。

基于 VHDL 的众多优点以及是今后设计的趋势，作者采用了利用 VHDL 语言设计数字系统的方法。

第四章 基于 FPGA 的 FIR 滤波器硬件实现

4.1　器件介绍和系统开发环境

4.1.1 Virtex-II 系列器件结构及特点

1.Virtex-II 系列 FPGA 概述

Virtex-II FPGA 是第一个基于 Platform 的 FPGA 具有 IP-Immersion 结构，设计人员可以更轻易的整合软件和硬件 IP 核:具有 4 万到 800 万系统逻辑门内部时钟可以高达 420MHz;I/O 带宽可以高达 840Mb/s。高性能的时钟管理电路，每个 Virtex-II 器件都有十六个预先设计好的低相偏时钟网路（low-skew clock networks）省去了高性能设计中复杂的时钟树分析。此外，Virtex-II 器件还包含多达 12 个时钟管理器(Digital Clock Manager DCM)，产生允许范围内任何频率的时钟信号，并提高时钟边缘配置（clock edge placement）的准确率，使误差降到百分之一。此外，Virtex-II 还支持片上与片外的时钟的同步化，保持精确的 50/50 工作周期，适合 Rapid IO, LDT 及 SPI-4 等的 DDR 应用系统。具有数控阻抗匹配(Digitally Controlled Impedance, DCI)DCI 技术可避免制造差异所造成的不同驱动强度，而且在温度、电压发生波动时，仍然能保持稳定的阻抗。此外，Xilinx 的控制阻抗技术(XCITE)使用两个外接参考电阻器保持数百个 I/O 管脚的输入及输出阻抗匹配，不但可减少电路板上的电阻器数量，大幅降低系统成本，而且还可降低电路板重新绕线(re-spins)的机率，简化电路板布局，并增加系统的稳定性。具备加密功能，全面保障设计的安全性。应用安全的三重数据加密标准(DES)演算法编码加密，加密演算所使用的密钥是通过 IEEE 1149.1(JTAG)来提供，使用电池或其他恒定电源将密钥存储芯片中。这项功能全面提升设计的安全性，避免设计遭人窃用。具有内嵌的 SelectRAM 存储结构。每个 SelectRAM 块为 18Kbit，可配置成高达 3Mbit 的双口 RAM。具有高性能的外部存储器接口。支持 SDR/DDR RAM,FCRAM,QDR RAM,CAM 等存储方式。具有专用的算术逻辑单元。有高达 168 个专用的 18bit×18bit 乘法器块和快速先行进位逻辑链。具有灵活的逻辑资源。带时钟使能的内部寄存器/锁存器高达 93184 个；查找表 LUT（look up table）

或可级连的移位寄存器高达 93184 个；具有多路选择器，以支持多输入功能；支持水平级连链和积之和(Sum of Products)表达式。具有内部三态总线。具有 SelectIO Ultra 技术。支持多种 IO 标准。可以支持 19 种单端口标准和 6 种差分标准。支持多种编程模式。支持串行、并行和 JTAG 编程，具有回读功能(read-back)。灵活的开发环境。支持 Synplify/Synplify Pro, LeonardoSpectrum, XST, ModelSim 等多种综合工具和仿真工具。其 ISE 开发上具支持 VHDL,VerilogHDL 等硬件描述语言，具有网络团队设计上具(Internet Team Design ITD)。

Virtex-II FPGA 系列芯片资源如表 4-1。

器件	系统门数	CLBs 1CLB=4slices=Max128bit			乘法器数量	SelectRAM Blocks		DCM
		Anay Row×Col	Slices	Maximum Distributed RAM (Kbit)		18Kbit 块数量	Max RAM (Kbit)	
XC2V40	40K	8×8	256	8	4	4	72	4
XC2V80	80K	16×8	512	16	8	8	144	4
XC2V250	250K	24×16	1536	48	24	24	432	8
XC2V500	500K	32×24	3072	96	32	32	576	8
XC2V1000	1M	40×32	5120	160	40	40	720	8
XC2V1500	1.5M	48×40	7680	240	48	48	864	8
XC2V2000	2M	56×48	10752	336	56	56	1008	8
XC2V3000	3M	64×56	14336	448	96	96	1728	12
XC2V4000	4M	80×72	23040	720	120	120	2160	12
XC2V6000	6M	96×88	33792	1056	144	144	2592	12
XC2V8000	8M	112×104	46592	1456	168	168	3024	12

表 4-1 Virtex II FPGA 系列芯片资源

2.Virtex-II FPGA 的结构

Virtex II 系列 FPGA 由多种可编程单元构成，主要用于高密度和高性能的逻辑设计，如图 4-1 由可配置逻辑块(Configurable Logic Blocks, CLBs)、数字时钟管理器(Digital Clock Manager, DCM)、SelectRAM 块、乘法器、全局时钟多路缓冲器以及可编程 IOB 组成。Virtex-II FPGA 的可配置逻辑块 CLB 每个可配置逻辑块包括 4 个 Slice, 2 个三态缓冲器。每个 Slice 包含有 2 个函数发生器(F&G)、2 个存储单元、多个算术逻辑门、快速超前进位链、水平级连链(即 OR 门)。函数发生器(F&G)可以编程为 4 输入的查找（Clook-up tables,LUTs)，或 16 位的移位寄存器，或 16 位 Distributed SelectRAM 存储器。2 个存储单元可以编程为边沿触发的 D 触发器或电平触发的锁存器。此外，每个 CLB 内部都有快速互连资源。

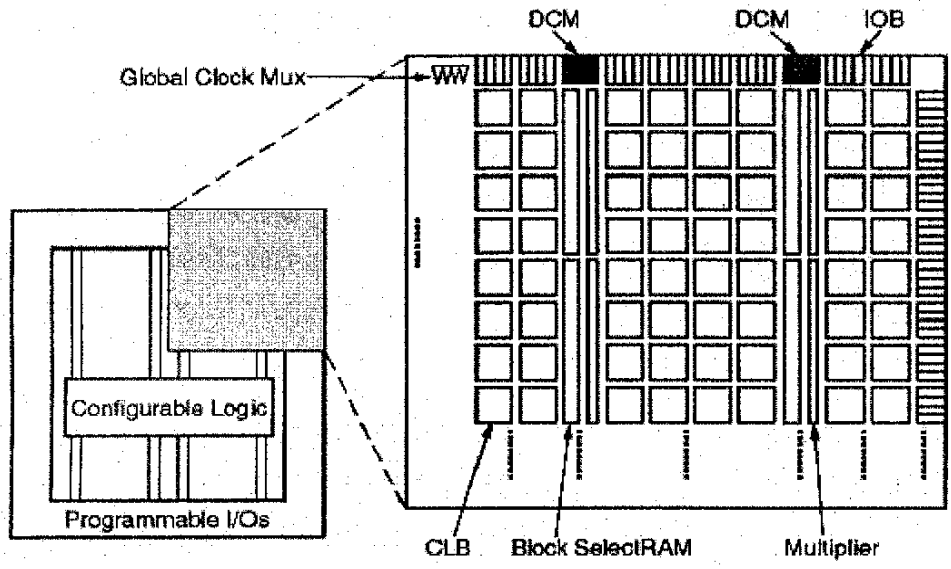


图 4-1 Virtex-II FPGA 结构图

4.1.2 开发工具简介

ISE 6.1i 开发系统是 XILINX 公司自主开发的用于可编程逻辑器件的开发系统，目前应用的较为广泛。这个开发系统提供了友好的用户界面，具有强大的开发和设计功能。在这个系统的开发环境下，设计人员可以完成从设计输入到设计仿真、从设计编译到器件编程的全部过程，而且这些操作可以在一个图形界面下完成。一般来说，XILINX ISE6.1i 开发系统的功能十分强大，它具有以下几个显著特点：

- 具有友好的用户界面，操作起来十分简单方便；
- 与其他许多 EDA 工具一样，它支持多种设计输入方式；
- 具有完全集成化的开发环境，用户能够方便、快捷地完成开发任务；
- 具有非常丰富的元件库，大大地提高了用户的开发效率；

通常，一个典型的 ISE6.1i 开发系统必须包含有两个特殊的软件包，它们分别是综合器和适配器。这两个软件包对于设计人员的项目开发起着至关重要的作用，因此这里有必要了解一下它们的具体功能。

综合器的功能就是将设计人员在 EDA 开发平台上完成的系统项目的 HDL、原理图和状态图形进行描述，针对给定的硬件系统组件进行编译、优化、转换和综合，最终获得实现项目功能的描述文件。通常，综合器首先必须给定所要实现

的硬件结构参数，硬件结构参数的功能将软件描述与给定的硬件结构以一定的方式联系起来。也就是说，综合器是软件与硬件实现的桥梁。综合过程就是将电路的高级硬件语言描述转换成低级的、可与目标器件 CPLD/FPGA 相映射的网表文件。

适配器的功能是有综合器产生的网表文件配置到指定的目标器件中，最终产生需要的下载文件，如 JED 文件。适配所选定的目标器件（FPGA/CPLD）必须属于综合器中已指定的目标器件系列。

ISE6.1i 很好的支持第三方仿真软件，本人就是结合使用 Modelsim SE 6.0 来进行仿真实现的。下面对 Modelsim 进行一个简单的介绍：

ModelSim 是业界最主要的硬件描述语言仿真工具，设计人员有两个版本可供选择：支持业界主流 UNIX 和 Windows NT 平台的特别版本 ModelSim SE（Special Edition），或是支持 Windows 个人计算机的个人版本 ModelSim PE（Personal Edition）。这两种版本都可以模拟 VHDL、Verilog 或是混合硬件描述语言设计，让工程师享有最大的作业平台和设计语言弹性。

ModelSim 是全世界应用最广的 VHDL 和 VHDL/Verilog 混合语言仿真器，也是成长速度最快的 Verilog 仿真器，不但深受客户欢迎，也证明 Model Technology 努力提供最好的模拟技术、工作效能、技术支持和价格。ModelSim 产品架构采用多项先进技术，例如最佳化直接编译（Optimized Direct Compile），单核心模拟（Single Kernel Simulation）和 Tcl/Tk；只有 ModelSim 可结合这些创新技术，带来领先业界的编译器 / 仿真器效能、不受限制的 VHDL 和 Verilog 混合设计、以及最强大的仿真器客制设定能力。此外，无论设计人员采用那种 ModelSim 模拟工具，他们都享有 Model Technology 驰名业界的操作简单性、除错支持、强健可靠的仿真质量和技术支持。

4.2　并行 FIR 数字滤波器的硬件实现

在第二章中已经大体介绍了FIR数字滤波器的几种实现方法。在此通过本人在设计中的具体实现给予详细的说明。这里的并行FIR滤波器指的是输入数据是并行的，然后并行的对这些数据进行处理，以便能够达到一个时钟完成对一个输入数据的处理。

入数据的处理。

4.2.1 并行结构的改进

传统的线性FIR滤波器的实现结构如图4-2所示。

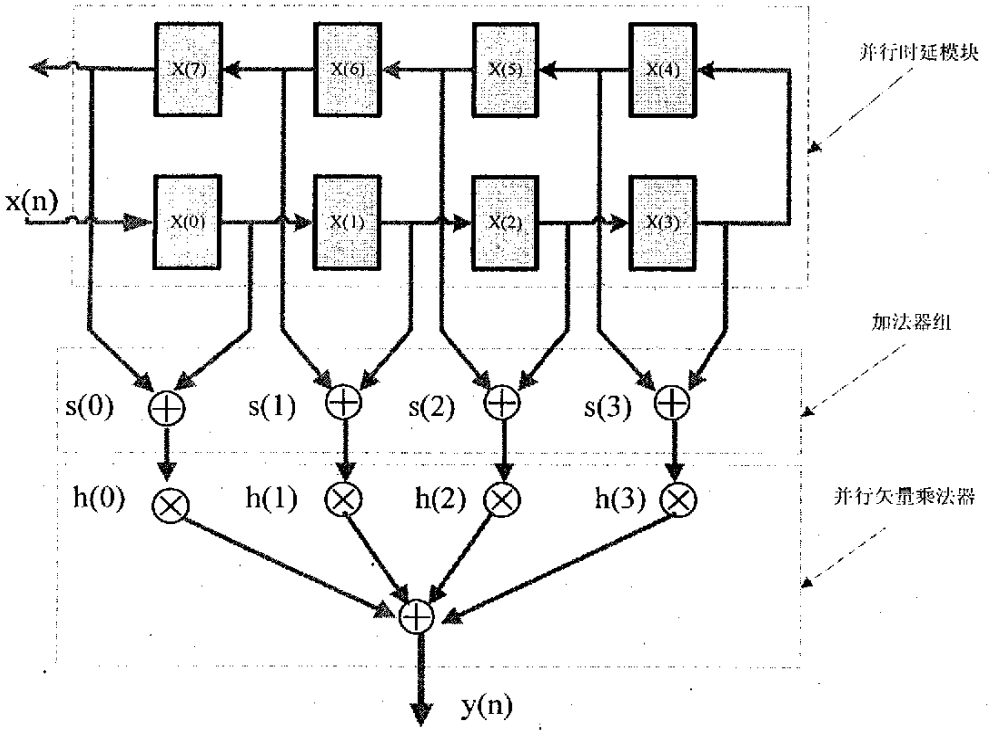


图4-2 传统线性FIR滤波器实现结构（n=8）

对于改进型的结构主要在于并行矢量乘法器模块，由于并行矢量乘法器是这个设计的重要部分，由于它占用这个系统的大量的资源，它的设计是否优化决定了这个系统设计的好坏，甚至决定是否可行。这部分的设计方法有许多种，例如直接像图4-2中给出的一样，通过乘法器把输入数据与冲击响应相乘，然后再通过加法器把它们相加。这样设计的缺点在于乘法器的实现非常困难，占用资源多，系统冗余大，需要的时钟延迟长。因此这种方法不适用于在Virtex系列器件上实现。像上面所介绍，由于Virtex-II系列器件具有大量的逻辑单元(LE)和可配置逻辑模块（CLB），这些单元都可以快速生成查找表，因此本设计采用了查找表的方法来设计并行矢量加法器。

4.2.2 模块的划分

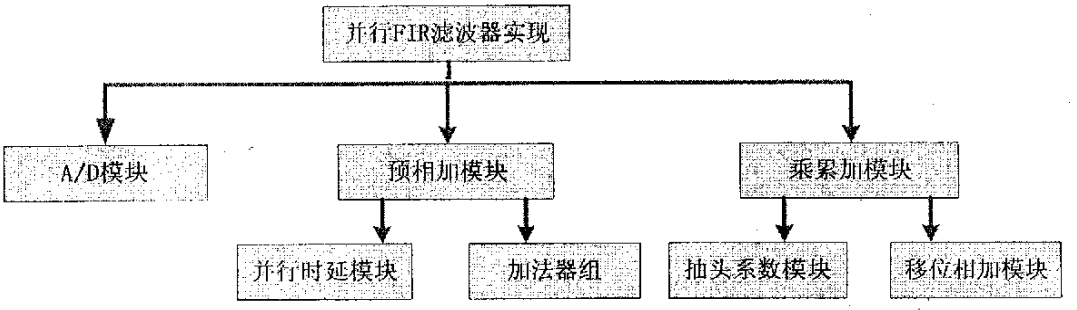


图4-3 并行结构设计模块框图

1. A/D模块

数字滤波器处理的是数字信号，首先必须将模拟信号经过A/D转换成离散的数字信号，由于条件的限制，此过程是通过Matlab来实现的。

2. 并行时延模块

A/D模块的采样数据送到并行时延模块，在每个时钟周期，把一组数据向下移动一位，而且前面介绍的Virtex-II器件中的逻辑单元具有同步使能的可编程触发器。所以移位寄存器组中每个移位寄存器都设计成一组D触发器并联，这里的D触发器数目为输入数据的有效位数。这些D触发器共用一个时钟脉冲，所以每个时钟就可以输出一组数据，从使而每个移位寄存器可以实现一个时钟延迟。

3. 加法器组模块

由于线性FIR滤波器抽头系数具有对称性，因此可以通过加法器组将对称的x(n)进行预相加，这样可以减少$\frac{N-1}{2}$个乘法器,从而降低了硬件的规模。

4. 抽头系数模块

抽头系数模块是将抽头系数的所以可能组合固化在FPGA的ROM中。加法器模块输出数据的相应位的组合，根据查找表输出相应的值。

5. 移位相加模块

移位相加模块是将LUT输出的值进行移位相加，从而实现乘法的功能，经过一组移位相加运算，通过寄存器输出y(n).

4.2.3 各模块具体功能的实现

为使设计结果更加的直观，下面是16阶线性FIR低通滤波器的具体实现：

假设输入为 $x=\sin(40\pi t)+\cos(400\pi t)$ ，如图4-4所示。

其中 $x_s=\cos(400\pi t)$ 是叠加在 $x=\sin(40\pi t)$ 的干扰信号。叠加后的输入信号波形如图4-5所示。

设计指标：$\omega_p=0.1\pi$　$\omega_s=0.15\pi$　$f_s=1000hz$

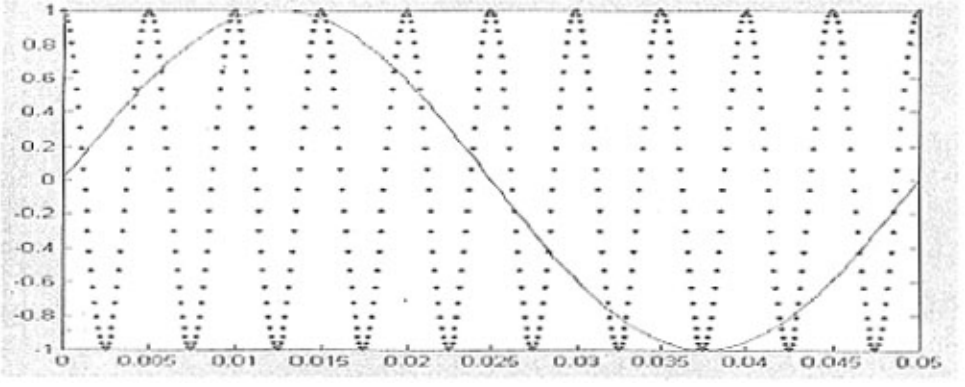


图 4-4 源信号和干扰信号波形

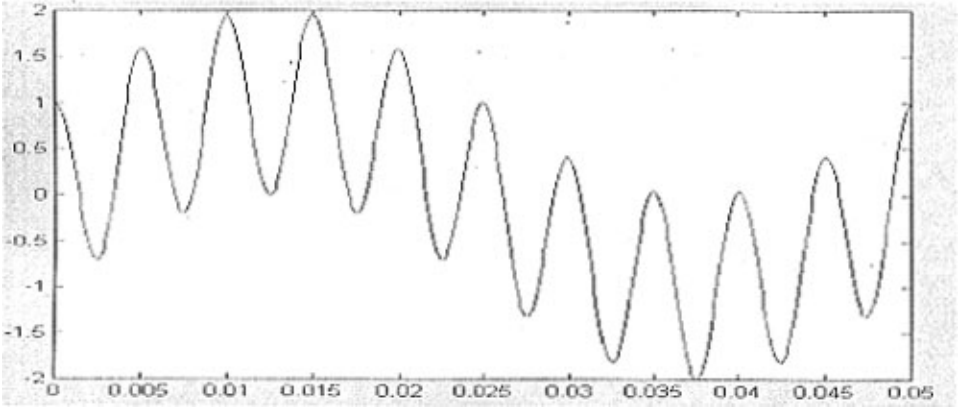


图4-5 叠加后输入信号波形

1. 输入信号的抽样和量化

此步骤用于并行方式中的A/D模块，其目的就是将连续信号进行抽样，并量化成二进制补码的形式，即x(n)。对于Virtex-II系列FPGA只支持定点运算，这样可以实现更高的速度和更低廉的成本，因此必须对采样后的信号进行相应的处理转换，由于采样的信号数值较小，必须先进行放大，然后在通过编写一个M程序将采样的数值转换成7位二进制补码的形式。本过程是通Matlab来实现的，具

体采样量化后的数值，请参见表4-1.

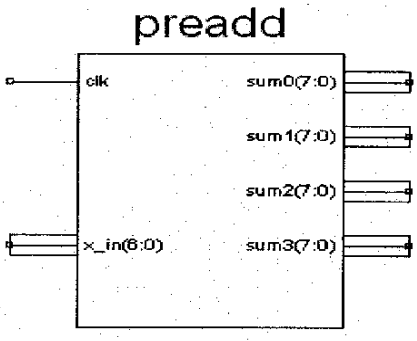
n	$x(n)_{10}$	$x(n)_{2c}$	量化后 x(n)	n	$x(n)_{10}$	$x(n)_{2c}$	量化后 x(n)
1	1	20	1	51	1	20	1
2	0.43435	0E	0.4375	50	0.18368	06	0.1875
3	-0.56033	6E	-0.5625	49	-1.0577	5E	-1.0625
4	-0.44089	72	-0.4375	48	-1.1771	5A	-1.1875
5	0.79077	19	0.78125	47	-0.17274	7A	-0.1875
6	1.5878	33	1.5938	46	0.41221	0D	0.40625
7	0.99356	20	1	45	-0.37553	74	-0.375
8	-0.038504	7F	-0.03125	44	-1.5795	4D	-1.5938
9	0.035311	01	0.03125	43	-1.6533	4B	-1.6563
10	1.2138	27	1.2188	42	-0.59581	6D	-0.59375
11	1.9511	3E	1.9375	41	0.048943	02	0.0625
12	1.2913	29	1.2813	40	-0.67327	6A	-0.6875
13	0.18901	06	0.1875	39	-1.807	46	-1.8125
...	...	...	...	...	...	...	...

表4－1 输入x的采样值x(n)

上表中，$x(n)_{10}$ 是十进制表示的采样数值。$x(n)_{2c}$ 是十六进制表示的7位二进制补码。x(n)为量化后的输入数值。

2. 预相加模块

该模块包括并行时延模块和加法器组模块。主要实现将输入采样数据x(n)，在每个时钟周期进行延时，并将对称抽头系数的x(n)进行预相加，结果输出到抽头系数模块。在该模块中，x_in(6:0)为采样输入信号，clk为系统时钟，sum0, sum1, sum2, sum3为八阶滤波器预相加的结果并行输出信号。其模块符号图如下：



具体程序如下：

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity preadd is
    port( x_in :   in std_logic_vector(6   downto 0);
          clk  :   in std_logic;
sum0, sum1, sum2, sum3:out std_logic_vector(7   downto 0));
end preadd;
architecture Behavioral of preadd is
signal x0, x1, x2, x3, x4, x5, x6, x7 : std_logic_vector(6 downto
    0) :="0000000";
signal x0_temp, x1_temp, x2_temp, x3_temp, x4_temp, x5_temp,
     x6_temp, x7_temp:  std_logic_vector(7   downto 0);
component  reg
port( d   :   in std_logic_vector ( 6 downto 0);
    clk :   in std_logic;
    q   :   out std_logic_vector ( 6 downto 0)
   );
end component;
begin
```

```
    p1:reg port map (x_in,clk,x0);
    p2:reg port map (x0,clk,x1);
    p3:reg port map (x1,clk,x2);
    p4:reg port map (x2,clk,x3);
    p5:reg port map (x3,clk,x4);
    p6:reg port map (x4,clk,x5);
    p7:reg port map (x5,clk,x6);
    p8:reg port map (x6,clk,x7);
    x0_temp <= x0(6) & x0;
    x1_temp <= x1(6) & x1;
    x2_temp <= x2(6) & x2;
    x3_temp <= x3(6) & x3;
    x4_temp <= x4(6) & x4;
    x5_temp <= x5(6) & x5;
    x6_temp <= x6(6) & x6;
    x7_temp <= x7(6) & x7;
    sum0 <= x0_temp + x7_temp;
    sum1 <= x1_temp + x6_temp;
    sum2 <= x2_temp + x5_temp;
    sum3 <= x3_temp + x4_temp;
end Behavioral;
```

3. 抽头系数h(n)的确定

根据设计指标，该滤波器的截止频率为50HZ，对于$\omega_p = 0.1\pi$，通过Matlab的强大功能通过调用相应的窗函数，根据截止频率和设计滤波器的阶数，调用fir1函数，可以求出脉冲响应系数h(n).对应于不同的窗函数，所求得的h(n)也不相同通过表2－1的比较，折中的选用汉明窗来进行设计，得到的结果见表4－2，在此采用8位字长来表示h(n)。

脉冲响应	数值	脉冲响应	16进制补码表示

h(0)	0.0034089	h(15)	00
h(1)	0.0074203	h(14)	01
h(2)	0.018846	h(13)	02
h(3)	0.039467	h(12)	05
h(4)	0.067664	h(11)	09
h(5)	0.098433	h(10)	0D
h(6)	0.12477	h(9)	10
h(7)	0.13998	h(8)	0D

表4-2 采用汉明窗得到的h(n)

4. 乘累加模块的实现

该模块又包括抽头系数和移位相加两个子模块。前面已经介绍Virtex-II器件拥有大量的可配置逻辑模块（CLB），每个模块有两个独立的4输入1输出的LUT和快速进位，另外一个3输入1输出的LUT将两个独立的LUT和两个触发器连接起来。因此，为了充分的发挥FPGA的硬件资源的独特性，采样基于查找表结构的DA算法的改进结构来替代硬件的乘法器。由于是四输入的LUT，因此为了节省资源简化结构，在此将16阶分成两个八阶FIR数字滤波器，采用级联的形式，这样子一方面有利于扩展成更高阶的滤波器，同时由于有限字长效应，高阶滤波器与低阶相比，采用级联结构实现的零点位置敏感度比直接型实现的要低，该部分在后面有详细的说明。对于八阶线性FIR滤波器，由于h(n)具有对称结构，因此可以简化成四输入的形式，将h(n)的各种组合预先存在查找表中，参见表4－3。

$s_b[n]$	$P_i(n)$	ROM中的值
0000	0	000000000
0001	h(0)	000000000
0010	h(1)	000000001
0011	h(0)+ h(1)	000000001
0100	h(2)	000000010
0101	h(0)+ h(2)	000000010
0110	h(1)+ h(2)	000000011

0111	h(0)+ h(1)+h(2)	000000011
1000	h(3)	000000101
1001	h(0)+ h(3)	000000101
1010	h(1)+ h(3)	000000110
1011	h(0)+ h(1)+h(3)	000000110
1100	h(2)+ h(3)	000000111
1101	h(0)+ h(2)+ h(3)	000000111
1110	h(1)+ h(2)+ h(3)	000001000
1111	h(0)+ h(1)+ h(2)+ h(3)	000001000

表4-3 16阶FIR滤波器LUT表

在设计过程中，为了设计的通用普遍性，所以的数据使用的是二进制补码的数据表示形式，考虑到中间计算过程中可能有溢出，如两个同符合的数相加，可能产生溢出，从而导致计算结果的错误。因此在两数相加的过程中，本人采用增加符号扩展位的方法来防止溢出，方法是将要相加的两个数最高位的符号位向前扩展一位，这样子就可以保证中间计算结果的正确性。本设计中LUT的输出位宽是9位其中最高位为符号扩展位，由于输入数据的位宽为8位，在并行结构中，$s_b[n]$（b=0, 1, …, 7）对应于8个查找表，前七位对应的查找表都一样，对于最高位符号位对应的查找表，根据式（2－15）为了全部用加法器提高速度，该查找表中写入的值是表4－3的补码的形式，详见表4－4，对于另一个级联的8阶滤波器设计方法完全相同，不同的是写入ROM中的数值为h(4), h(5), h(6), h(7)的所有组合形式。

$s_7[n]$	$P_7(n)$	ROM中的值
0000	0	000000000
0001	h(0)	000000000
0010	h(1)	111111111
0011	h(0)+ h(1)	111111111
0100	h(2)	111111110
0101	h(0)+ h(2)	111111110

0110	h(1)+ h(2)	111111101
0111	h(0)+ h(1)+h(2)	111111101
1000	h(3)	111111011
1001	h(0)+ h(3)	111111011
1010	h(1)+ h(3)	111111010
1011	h(0)+ h(1)+h(3)	111111010
1100	h(2)+ h(3)	111111001
1101	h(0)+ h(2)+ h(3)	111111001
1110	h(1)+ h(2)+ h(3)	111111000
1111	h(0)+ h(1)+ h(2)+ h(3)	111111000

表4-4 最高位LUT中的数据

前面介绍的LUT是在抽头系数模块来实现的，实现方式可以通过Xilinx ISE中的Core generator将以上数据固化在ROM中，本人是通过VHDL语言形式固化在ROM中的，其优点就是更具有通用性和可移植性。并行移位相加模块是将sum0, sum1, sum2, sum3相同的有效位组成地址信号通过LUT输出值table_out[0:8]得到相对应的部分积，通过移位寄存器和加法器将部分积相加，从而得到最终的乘积y(n)。该模块硬件实现结构见框图4－6。

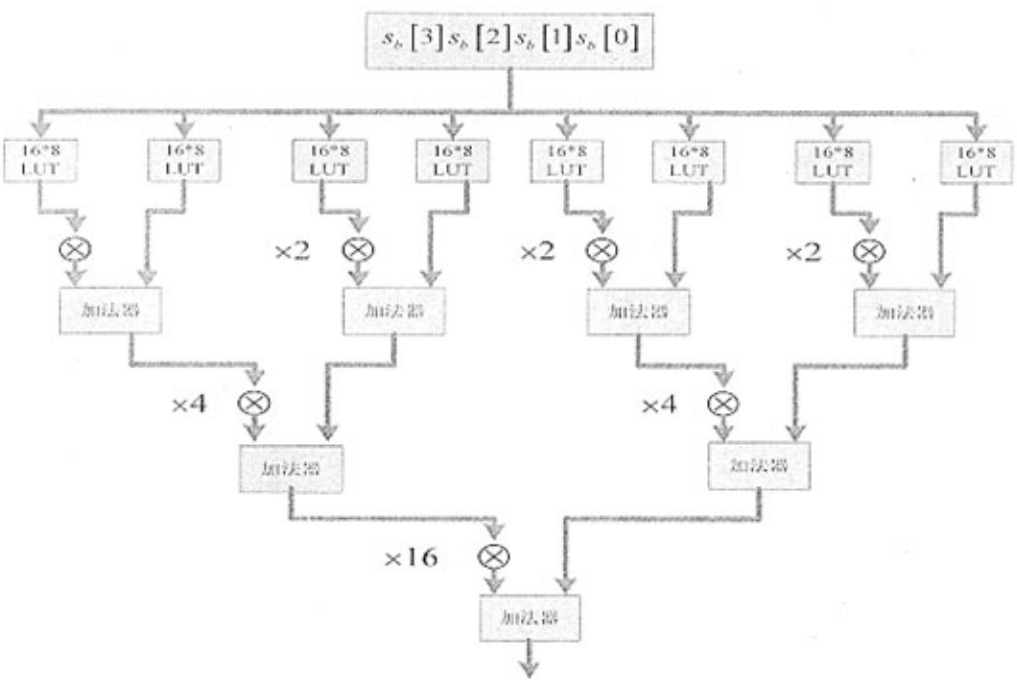
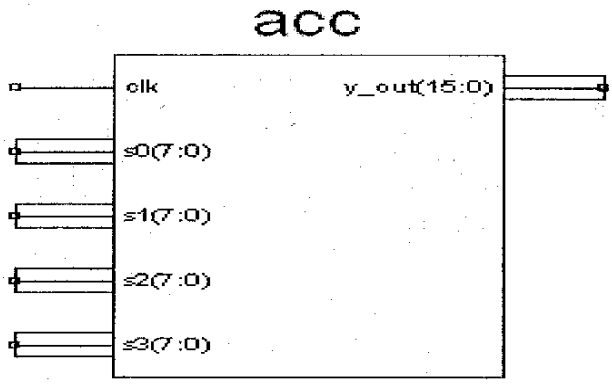


图4-6 四乘8位乘累加模块结构示意图

其模块符号图如下：



4.2.4 并行 FIR 滤波器的流水线设计

流水线技术在数字电路设计中是为了提高系统的工作时钟频率而采用的一种特殊的设计方法。全并行实现方法可以利用流水线技术，将复杂的数字逻辑电路分级实现。这样，每一级的电路结构得到简化，从而减少输入到输出间的电路延时，在较小的时钟周期内就能够完成这一级的电路功能。在下一个时钟周期到来的时候，将前一级的结果锁存为该级电路的输入，这样逐级锁存，由最后一级完成最终结果的输出。在这个过程当中，数据就好像流过了一根数据管道，流水线技术由此得名。也就是说流水线技术是将待处理的任务分解为相互有关而又相互独立的、可以顺序执行的子任务来逐步实现。

在流水线技术中，由于算法分解后，数据逐级锁存，输出不是实时的，电路中有几级流水线，输出相对于输入就会延迟相应的时钟周期。另外，在每一级都要用寄存器将上一级的结果寄存，所以，当电路中位数增多时，电路的规模就会迅速增加。这是流水线技术为了得到较高工作速率而增加的额外开销。虽然如此，但是如果我们把系数的个数限制在 4 个或是 8 个的时候，再加上流水线寄存器，这个代价还是值得的。

FIR滤波器的整个运算过程包括移位、相加运算和乘法运算。在并行FIR滤波器中这些运算应在一个时钟内完成，但由于运算逻辑相当复杂，信号延迟较长，从而限制了时钟频率的提高。当整个运算采用流水线操作结构时，把在一个时钟

内欲完成的运算化成若干子运算，各个子运算(加减运算、查表和各级移位相加运算)采用寄存器输出模式，这样既缩短了延时路径，可提高时钟频率，又可使各个子运算同时进行，提高数据吞吐率。同时由于Virtex-II器件中含有大量的D触发器也为实现流水线结构提供了方便。

4.2.5 并行FIR滤波器的扩展应用

以上设计的滤波器可以作为一个部件使用在数字信号处理系统中，滤波器的输入和输出均采用并行方式，且数据采用二进制的补码表示。若采用Virtex-II器件，滤波器的工作频率可达到100MHz以上。采用流水线结构在提高系统处理速度的同时也造成了输出滞后，从数据输入到数据输出要经过9个时钟的延时，每个时钟周期内可以产生一个有效的输出。为了实现阶数更高的滤波器，可以将多个滤波器模块级联而成，其公式表达为：

$$y(n)=\sum_{i=0}^{N-1}h(i)s(n-i)=\sum_{i=0}^{3}h(i)s(n-i)+\sum_{i=4}^{N-1}h(i)s(n-i) \qquad (4\text{-}1)$$

依据公式(4－1)，级联形式如图4－7所示。

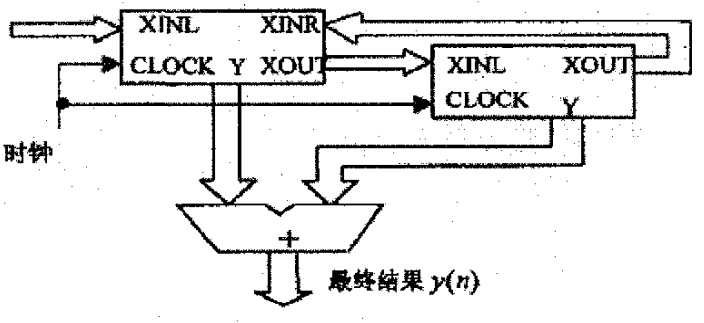


图4-7 并行滤波器的级联结构

本设计就是采用两个8阶线性FIR滤波器来级联实现16阶FIR滤波器的，通过级联的方式，原则上我们可以实现任意高阶的FIR滤波器，只要我们所用的FPGA硬件资源足够的充分。但是限于资源的限制，以及设计要求的需要，往往不能够实现太高阶的滤波器。图4－8是本设计通过Xilinx ISE6.0 所生成的顶层结构连接图。

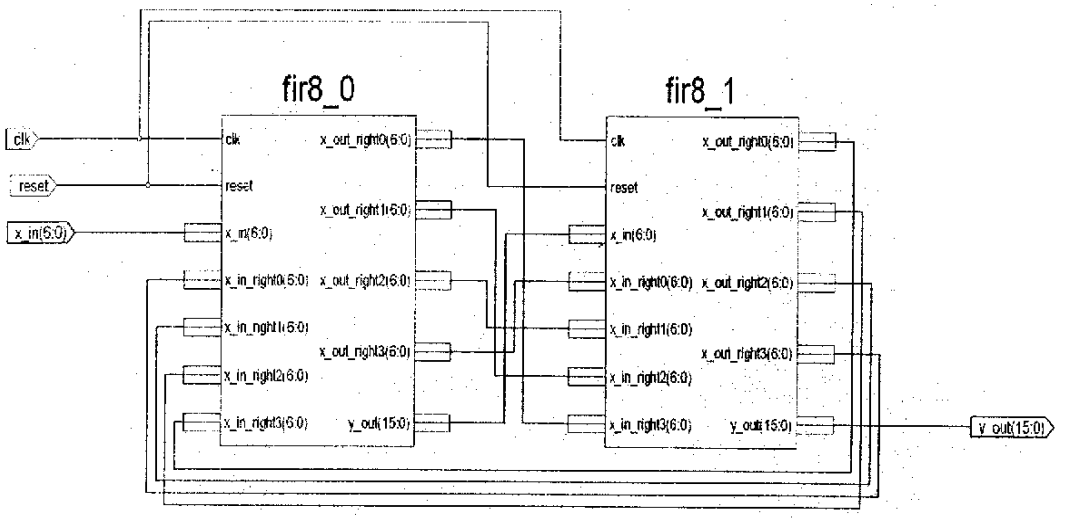


图4-8 16阶FIR滤波器顶层连接原理图

4.2.6 小结

在并行FIR滤波器的设计中，由于使用了大量的查找表和采用了流水线技术，所以并行FIR滤波器的处理数据的速度很快，每隔一个时钟可以输出一个数据，可以运行在很高的时钟下，缺点是比较耗费FPGA的硬件资源。下面介绍另一种实现方案——串行实现方式。

4.3 串行 FIR 数字滤波器的硬件实现

4.3.1 设计思想与实现

对于串行FIR数字滤波器的硬件实现的设计思想请参考2.3.2章节，在此就不赘述了。下面主要讨论一下其具体硬件实现的方式，根据串行FIR滤波器的特点，本人采用TOP—DOWN的设计方式，先进行模块的划分，具体划分为以下模块，见图示4-9。

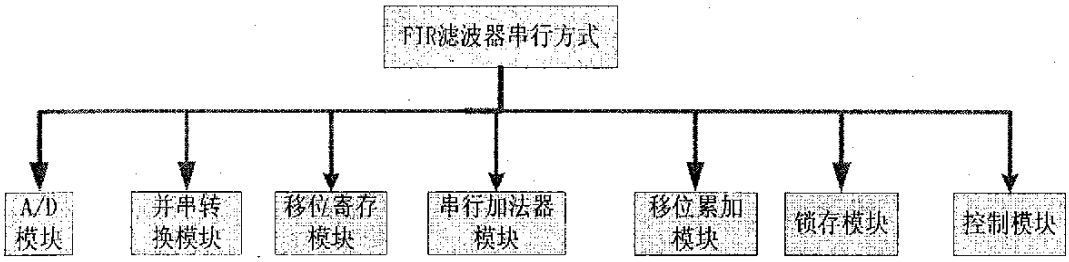


图4-9 线性FIR滤波器串行方式模块框图

4.3.2 各模块具体功能的设计实现

1. A/D转换模块

该模块的功能介绍和实现和前面的FIR滤波器的并行实现方式基本是一样的，在此就不赘述了。

2. 并串转换模块

由于整个电路以位串行方式工作，而输入数据是并行的，因此，必须首先将并行输入的数据转成串行数据，以作为下一级模块的输入。

din[6..0]为并行输入数据;clk为同步时钟信号;p2s_load为并行输入同步加载信号，且高电平有效;sout为串行数据输出。具体程序如下:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity PSC is
port(din : in  std_logic_vector(6 downto 0);
     clk : in  std_logic;
     p2s_load: in std_logic;
     sout: out std_logic
     );
end PSC;
architecture Behavioral of PSC is
```

```
signal din_temp : std_logic_vector(7downto 0):=x"00";
begin
process(clk)
    begin
    if (clk'event and clk ='1') then
        if (p2s_load ='1') then
            din_temp <= din(6)&din;
        else
            din_temp <=din_temp(7)&din_temp(7 downto 1);
        end if;
    end if;
end process;
sout <= din_temp(0);
end Behavioral;
```

3. 移位寄存器模块

由于输入数据有效位数为8位，因此通过8个D触发器的串联组成8×1移位寄存器，每个时钟沿下移移位。实际上是完成时延的功能。由于是8阶滤波器，一共需要8个8×1移位寄存器。Din为串行输入数据，dout[7:0]为8个8×1移位寄存器最低位输出的组合。

4. 串行加法器模块

串行加法器模块主要功能是利用滤波器系数的对称性，将与相同滤波器系数相乘的两个采样值预先相加，以减小硬件规模，这样能使滤波器的乘法次数减半。

串行加法器模块由串行加法器(Serial Adder)来构成。串行加法器在位串行电路中被广泛采用。串行加法器由一个全加器与D触发器组成。加法器的两个输入数据均为补码形式并且是时钟同步的，全加器的进位端Cin由D触发器经过一个时钟周期的延时后反馈至输入端，相当于下一组加法运算的进位位。清零端CLEAR在每一个字周期的开始将进位位清零，也即将D触发器同步清零。这里介绍的为偶对称的FIR滤波器，当FIR滤波器为奇对称时，则将加法器改为减法器，其余不变。该模块的顶层代码如下：

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity preadd is
   port(A,B: in std_logic_vector (3 downto 0);
        clk :   in std_logic;
        clear: in std_logic;
        S:   out std_logic_vector( 3 downto 0)
       );
end preadd;
architecture Behavioral of preadd is
    component full_adder
       port(a,b :   in std_logic;
            clr: in std_logic;
            clk: in std_logic;
            s: out std_logic
       );
    end component;
begin
    u0: for i in 0 to 3 generate
add: full_adder port map (a=>A(i),b=>B(i),clr =>clear,
                          clk =>clk,s=>S(i))
    end generate ;
end Behavioral ;
```

5. 移位累加模块

1) 查找表单元

前面并行实现方式对此有了比较详细的介绍，同样是将预先计算的所有h(n)的组合保存到开辟的ROM中。其代码如下：

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity rom is
port( table_in    :  in  std_logic_vector( 3 downto 0);
      table_out   :  out std_logic_vector( 7 downto 0)
      );
end rom;
architecture Behavioral of rom is
begin
process(table_in)
begin
    case table_in is
        when "0000" =>   table_out <= "00000000" ;
        when "0001" =>   table_out <= "00101000" ;
        when "0010" =>   table_out <= "00100000" ;
        when "0011" =>   table_out <= "01001000" ;
        when "0100" =>   table_out <= "11001000" ;
        when "0101" =>   table_out <= "11110000" ;
        when "0110" =>   table_out <= "11101000" ;
        when "0111" =>   table_out <= "00010000" ;
        when "1000" =>   table_out <= "11010000" ;
        when "1001" =>   table_out <= "11111000" ;
        when "1010" =>   table_out <= "11110000" ;
        when "1011" =>   table_out <= "00011000" ;
        when "1100" =>   table_out <= "10011000" ;
        when "1101" =>   table_out <= "11000000" ;
        when "1110" =>   table_out <= "10111000" ;
```

```
            when "1111" =>    table_out <= "11100000" ;
            when others =>    table_out <= "00000000";
        end case;
    end process;
    end Behavioral;
```

2) 移位累加器单元

移位累加器结构如图4-10所示，它是由一个加减法控制器和一个算术移位寄存器组成的。在每个时钟到来时，查找表的输出送到加减法控制器，其运算结果输出经过算术移位寄存器作为下一个时钟周期加减法控制器的输入。当add_sub等于0时，使移位寄存器输出的数据加上从data端输入的数；否则将减去data端输入的数。移位累加器的功能为把出s(n)的每一位从查找表中取得的数相加，并且与出它的符号扩展位所从查找表中取得的数相减。下面介绍一下移位累加器的工作过程：

首先控制模块把acc_clr置1，算术移位寄存器输出0(相当于把移位累加器置零)，然后与输入的数据相加，这时把acc_clr置0，计算的结果在下个时钟输入到算术移位寄存器，经过算术右移一位输入到加减法控制器中与下一个数相加。当计算到第7个数的时候，控制模块把add_sub置1，使得用前7个数的移位相加后的和减去第8个数(符号扩展位从查找表中查出的值)，同时又把acc_clr置1，把移位累加器清零，准备计算下一个y(n)。

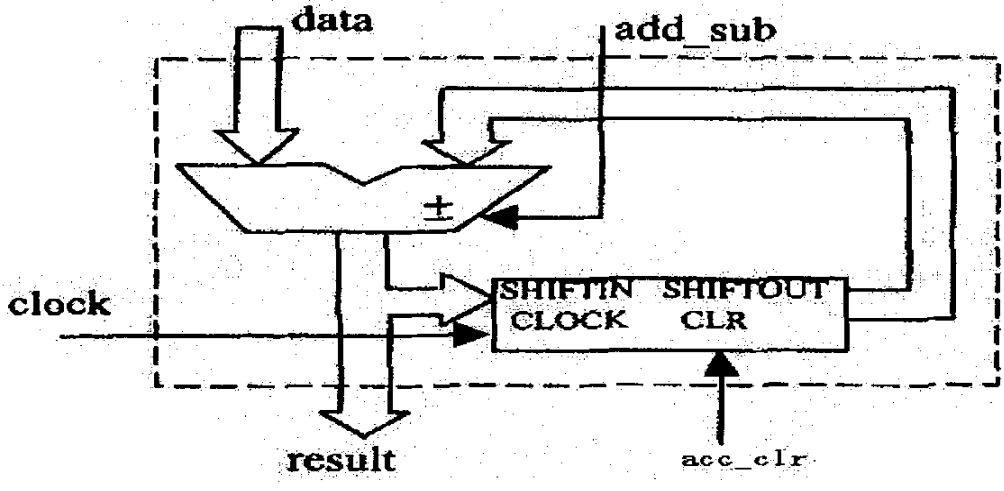


图4-10 移位累加器结构

6. 锁存模块

锁存模块的主要功能是对输出的最后结果进行锁存。这是由于加法器的输出很不稳定，影响了最后结果的输出，所以要将结果进行锁存。在y(n)计算完成后，锁存控制信号latch有效，从而将结果锁存.

7. 控制模块

控制模块主要是产生p2s_load 、carry_clr、acc_clr等控制信号。本人是通过状态机来设计完成的。由于每一个控制信号具有不同的时序，只有保证时序的正确，各模块才能互相协调工作，从而得出正确的结果。

其MDS流程图见图4-11:

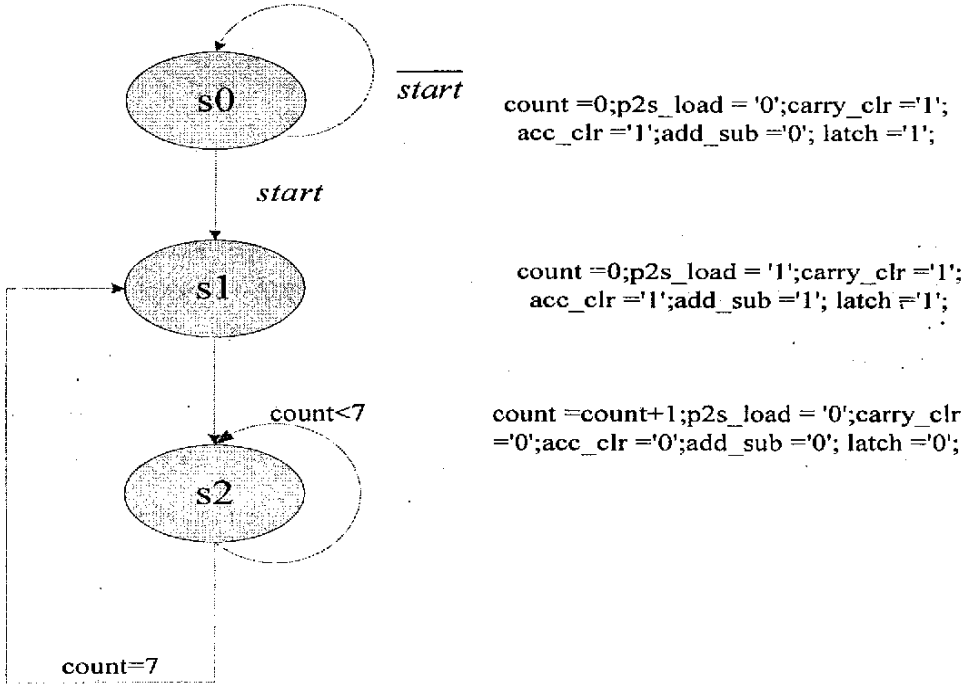


图4-11 MDS流程图

其VHDL代码如下：

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity controler is
    port(start :  in  std_logic;
```

```
        clk :    in std_logic;
        carry_clr: out std_logic;
        p2s_load: out std_logic;
        latch: out std_logic;
        acc_clr: out std_logic;
        add_sub: out std_logic
       );
end controler;
architecture Behavioral of controler is
type state_type is (s0,s1,s2);
signal state : state_type;
begin
   process(clk)
   variable count : integer range 0 to 7 :=0;
   begin
      if (clk'event and clk ='1') then
      case state is
        when s0 => count :=0;p2s_load <='0';carry_clr<='1';
                   acc_clr<='1';add_sub<='0';latch<='1';
         if(start ='1') then
        state <=s1;
         else
        state <=s0;
         end if;
        when s1 => count :=0;p2s_load <= '1';carry_clr <='1';
         acc_clr <='1';add_sub <='1'; latch <='1';state <=s2;
        when s2 => p2s_load <='0';carry_clr <='0';acc_clr <='0';
                   latch<='0';add_sub <='0';count:=count+1;
             if (count=7) then
```

```
                state <=s1;
            end if;
        end case;
    end if;
  end process;
end Behavioral;
```

4.3.3 FIR 滤波器串行方式实现的系统分析

根据前面的各模块的设计实现，通过ISE6.0生成的FIR串行方式的顶层原理图见图4-12：

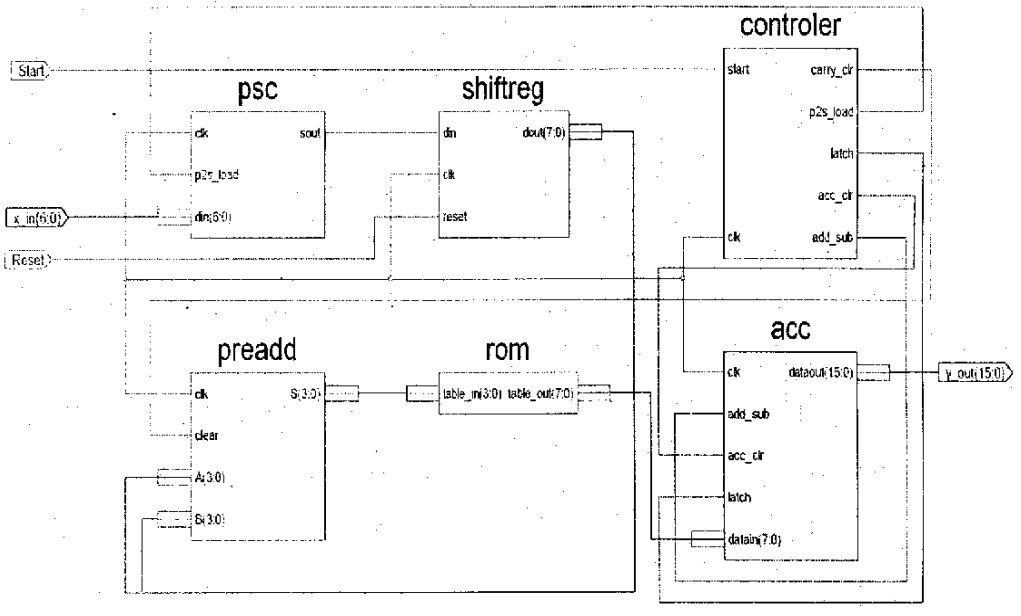


图4-12 顶层原理图

4.3.4 串行 FIR 滤波器的扩展应用

串行滤波器的输入采用并行，而输出采用串行方式，而且数据也采用2进制补码表示。选用VIRTEX-II系列器件，滤波器的工作频率可达到100MHz以上。由于流水线造成的输出滞后，使数据从输入到输出要经过22个时钟的延时，每8个时钟周期产生一个有效的输出。串行滤波器同并行滤波器一样都可以扩展，并且

它们的扩展方法基本相同。在串行滤波器的级联结构中，由于使用了相同的控制模块和移位累加器，所以两个模块共用一个控制模块和移位累加器。而在串行滤波器的并联结构中，由于结构的要求只能共用一个控制模块。

4.3.5 小结

串行方式滤波器的输入是并行的，输出采用的是串行方式，数据也是采用二进制补码的形式表示。串行数字滤波器的硬件实现采用的理论与并行数字滤波器的硬件实现是基本相同的，只是在实施上串行方式对输入数据采用的是按位处理的方法，而并行方式对输入数据的所有位同时处理的。为了提高系统的性能，串行滤波器在实现时也采用了流水线技术，由于流水线造成的输出滞后，数据从输入到有效输出要经过 18 个时钟周期的延时。同时，由于串行方式进行的是串行位操作，因此每 8 个时钟周期产生一个有效的输出；而并行方式是一个周期产生一个有效输出。尽管如此，串行方式比并行方式在所需硬件规模上大大的减少，比较适合于硬件资源有限的器件设计当中。

4.4 综合与仿真验证

4.4.1 线性 FIR 滤波器并行方式的综合与仿真

1. 16阶线性FIR滤波器的综合

前面介绍采用流水线的方式可以大大的提高系统工作的频率，提高数据吞吐率，流水线级数的不同，系统稳定工作的最高时钟频率也不一样。采用流水线的级数越多，系统稳定工作的频率越高，缺点是占有的系统硬件资源越多，同时输出延迟的时钟周期越多。本人分别采用了四级和五级流水线的设计方式，通过ISE6.1i对16阶线性FIR滤波器进行了综合后的对比，得出一般性的结论。对比如下：

采用五级流水线设计结构的综合部分报告1.

```
Device utilization summary:
---------------------------
```

```
Selected Device : 2v4000ff1152-6
 Number of Slices:                      173  out of  23040     0%
 Number of Slice Flip Flops:            309  out of  46080     0%
 Number of 4 input LUTs:                220  out of  46080     0%
 Number of bonded IOBs:                  24  out of    824     2%
 Number of GCLKs:                         1  out of     16     6%
=========================================================================
====
TIMING REPORT
Clock Information:
------------------
-----------------------------------+------------------------+-------+
Clock Signal                       | Clock buffer(FF name)  | Load  |
-----------------------------------+------------------------+-------+
clk                                | BUFGP                  | 309   |
-----------------------------------+------------------------+-------+
Timing Summary:
---------------
Speed Grade: -6
   Minimum period: 4.058ns (Maximum Frequency: 246.457MHz)
   Minimum input arrival time before clock: 1.329ns
   Maximum output required time after clock: 4.575ns
   Maximum combinational path delay: No path found
```

采用四级流水线设计结构的综合部分报告2.

```
Device utilization summary:
---------------------------
Selected Device : 2v4000ff1152-6
 Number of Slices:                      180  out of  23040     0%
 Number of Slice Flip Flops:            253  out of  46080     0%
```

```
Number of 4 input LUTs:                242  out of  46080     0%
 Number of bonded IOBs:                 24  out of    824     2%
 Number of GCLKs:                        1  out of     16     6%
=========================================================================
====

Clock Information:
------------------
-----------------------------------+------------------------+-------+
Clock Signal                       | Clock buffer(FF name)  | Load  |
-----------------------------------+------------------------+-------+
clk                                | BUFGP                  | 253   |
-----------------------------------+------------------------+-------+

Timing Summary:
---------------

Speed Grade: -6
   Minimum period: 6.100ns (Maximum Frequency: 163.934MHz)
   Minimum input arrival time before clock: 1.329ns
   Maximum output required time after clock: 4.575ns
   Maximum combinational path delay: No path found
```

根据以上综合报告可以看出，在增加一级流水线后，系统的最高时钟频率大大的提高了，但同时系统占用的硬件资源多了。在此我们可以得出，在使用的FPGA资源比较丰富的情况下，尤其在大规模的设计中，逻辑比较复杂，要求能够工作在比较高的时钟频率，那么就可以采用多级流水线的方式。总而言之，我们在设计一个系统过程中，要根据需求，兼顾资源和系统稳定性，这样才能设计出比较合理的系统。

2. 16阶线性FIR滤波器并行方式的仿真验证

五级流水线并行方式FIR滤波器的仿真结果见图4-13。

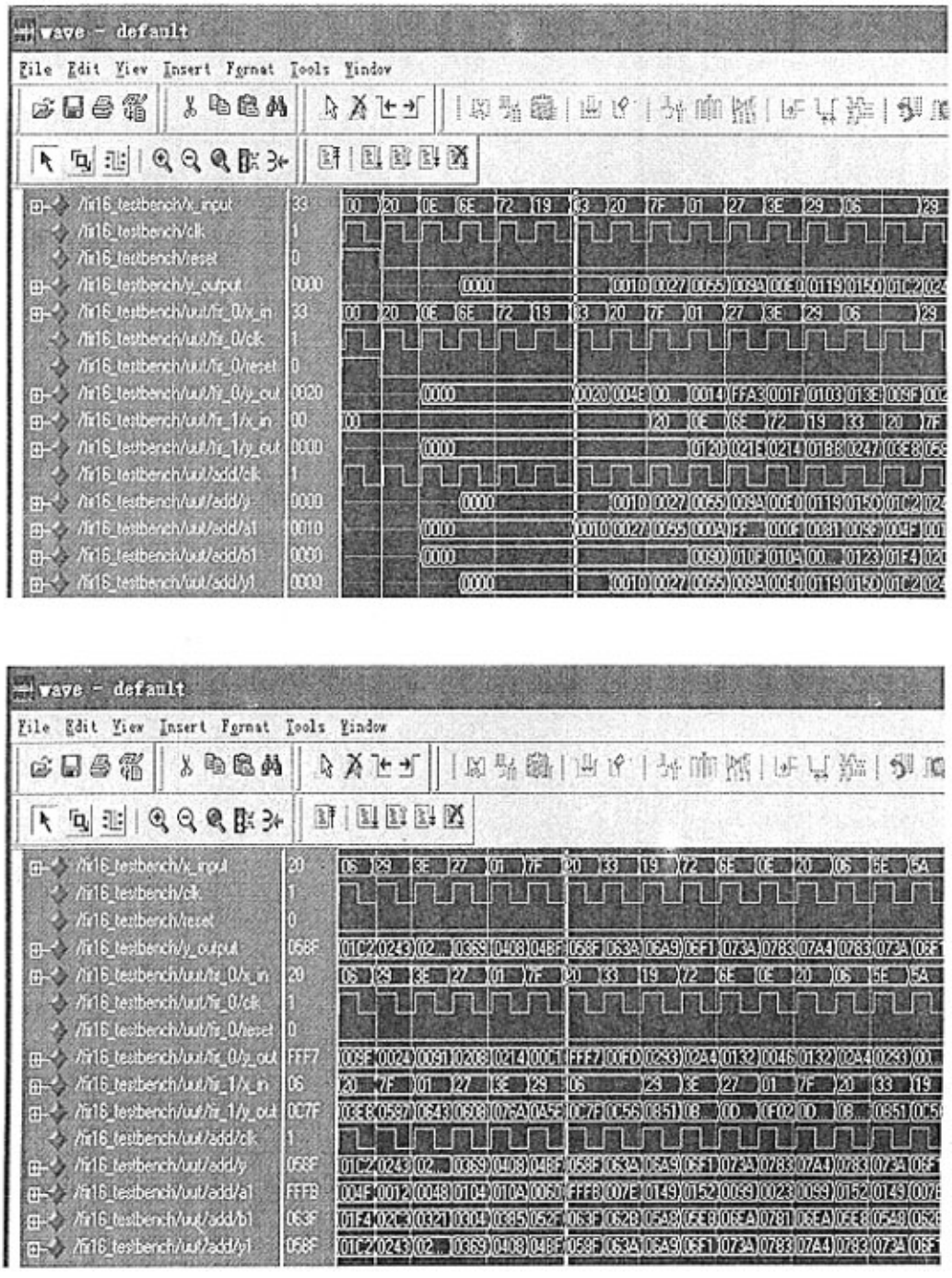


图4-13 五级流水线FIR滤波器全仿真结果

由上图可以看出，仿真结果与设计思想完全符合，输出延迟输入信号5个时钟周期，每个时钟周期输出最终结果，将仿真结果整理成表的形式，并与理论值相比较，详见表4-5。

n	理论输出y(n)	实际输出y(n)	
	十进制	十六进制	十进制
1	0.0034089	0000	0
2	0.0089009	0010	0.0078125
3	0.020159	0027	0.019043
4	0.041992	0055	0.041504
5	0.073671	009A	0.075195
6	0.10868	00E0	0.10962
7	0.14229	0119	0.13745
8	0.17757	015D	0.17065
…	…	…	…
32	0.17887	0173	0.18115
33	0.054141	0062	0.048096
34	-0.064417	FF6C	-0.072021
35	-0.17494	FE95	-0.177
36	-0.28539	FDB8	-0.28467
37	-0.40001	FCC2	-0.40479
38	-0.511	FBCF	-0.52368
39	-0.60692	FB0B	-0.61963
40	-0.68624	FA77	-0.69165
…	…	…	…

表4-5 实际值与理论值对比

利用Matlab工具将以上输出的数据导入到Matlab中，利用绘图工具得到输出一个周期的波形，见图4-14，可见该滤波器成功的实现所要求的功能。

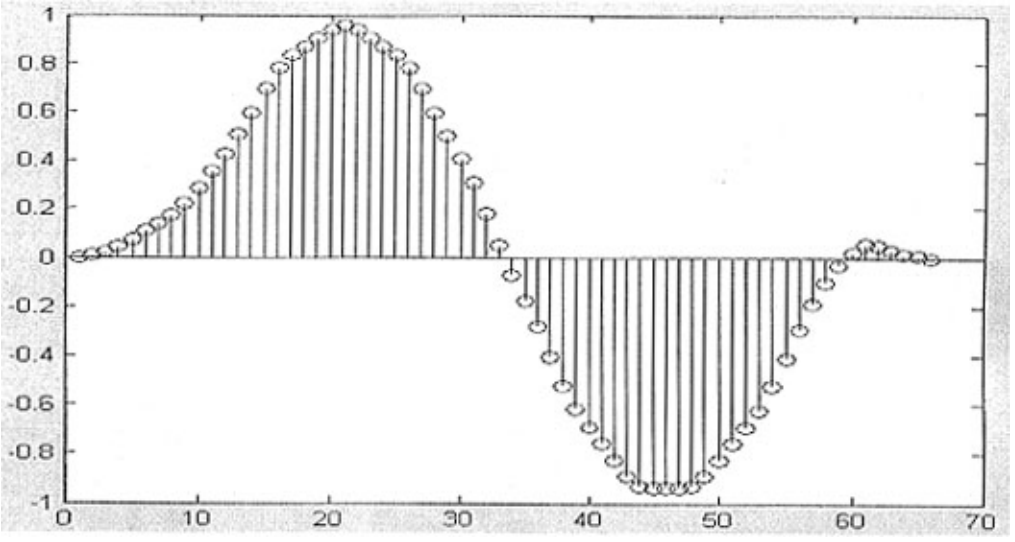


图4-14 输出结果在Matlab中一个周期波形

3. 16阶FIR数字滤波器的幅频相频特性

16阶线性相位FIR数字滤波器幅频响应特性曲线见图4-15，由图可见截止角频率$\omega_p = 0.1\pi$　$\omega_s = 0.15\pi$，阻带最大衰减为30dB（见4.2.3），结合图4-14可知该16阶级联并行方式线性相位FIR数字滤波器满足设计要求。其相频响应特性曲线见图4-16，由图可见符合线形相位的形式。

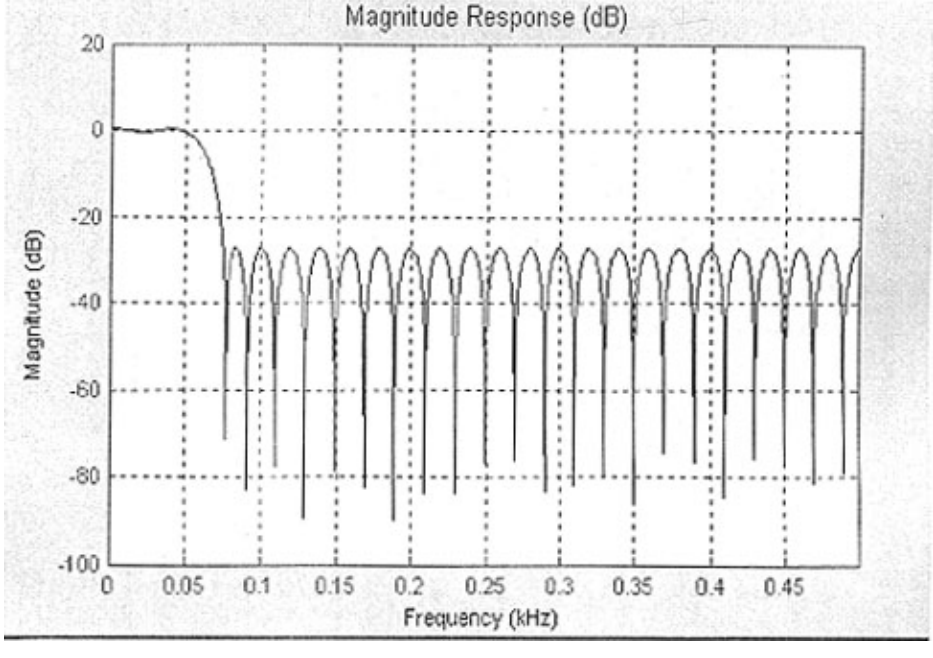


图4-15 幅频响应曲线

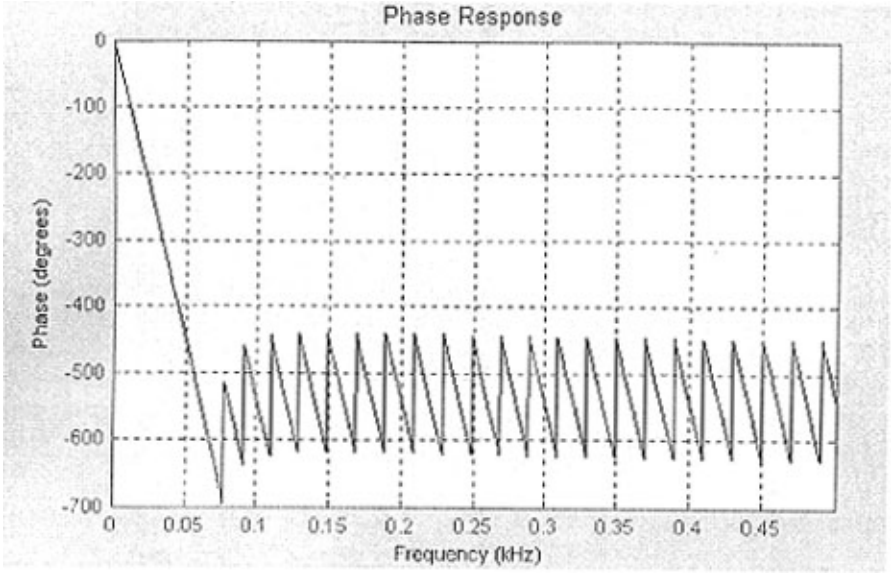


图4-16 相频响应曲线

4.4.2 线性FIR滤波器串行方式的综合与仿真

1. 8阶线性FIR滤波器的综合

8阶线性FIR滤波器串行方式的综合报告结果如下：

```
Device utilization summary:
---------------------------

Selected Device : 2v4000ff1152-6

 Number of Slices:                       65  out of  23040     0%
 Number of Slice Flip Flops:            117  out of  46080     0%
 Number of 4 input LUTs:                 39  out of  46080     0%
 Number of bonded IOBs:                  24  out of    824     2%
 Number of GCLKs:                         1  out of     16     6%
=========================================================================

Timing Summary:
---------------

Speed Grade: -6

   Minimum period: 4.354ns (Maximum Frequency: 229.648MHz)
   Minimum input arrival time before clock: 2.162ns
```

Maximum output required time after clock: 4.575ns

Maximum combinational path delay: No path found

根据以上综合报告可以看出，串行方式大大的节省了硬件资源，尤其是LUT的数目大大的减少。

2. 线性FIR滤波器串行方式的仿真验证

a) 控制模块的时序仿真

控制模块的时序仿真见图4-17.

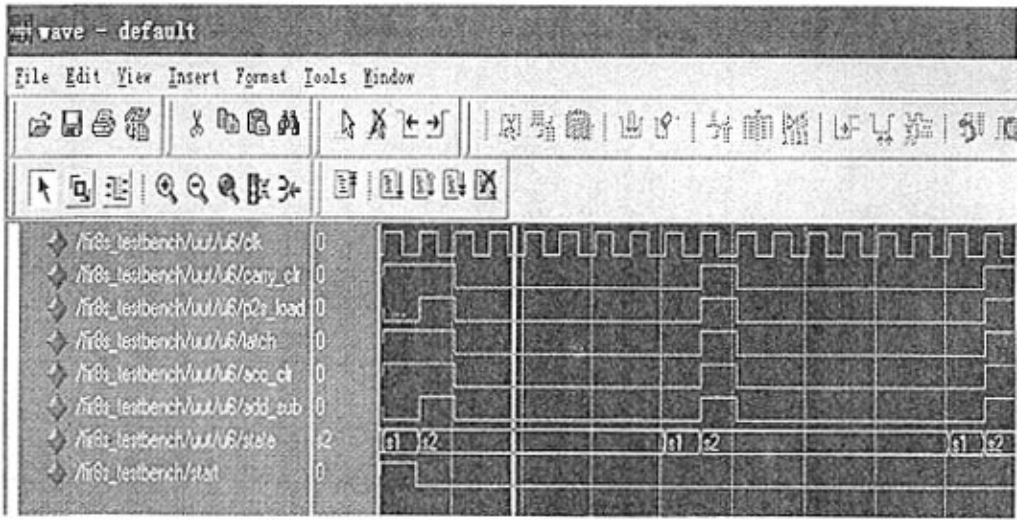


图4-17 控制模块时序仿真图

图4-17中start为开始信号;clk为时钟信号;state为控制器状态;其它的为控制输出信号。

从图4-17中可知，当start信号来到后，控制模块开始工作，p2s_load在下一个上升沿有效为高电平，将输入信号数据锁存，在PSC模块转换完毕第一个锁存数据后，p2s_load又高有效，将第二个输入数据锁存。其规律是每8个时钟周期高有效。对于carry_clr, acc_clr, add_sub, latch信号都是每8个时钟周期有效一次。符合系统各模块实现的时序要求。

b) 并串转换模块仿真

并串转换模块仿真结果见图4-18。

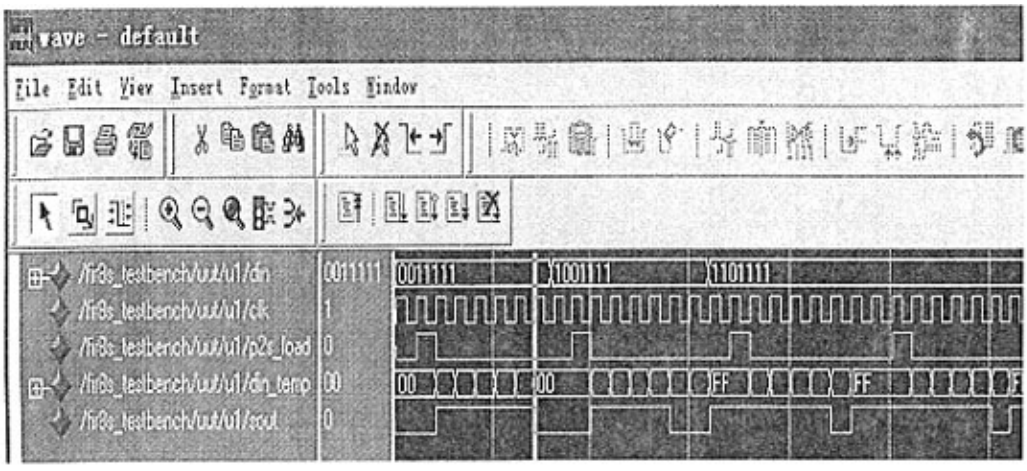


图4-18 并串转换模块仿真图

c) FIR滤波器系统仿真

由于前面在并行方式方案中，已经详细介绍了通过级联方式来实现高阶线性FIR滤波器，一般是将多个八阶滤波器模块级联来实现的。限于篇幅，在此只验证八阶串行方式FIR滤波器模块，其串行高阶实现方式与并行类似。

在此假设抽头系数为：

h(0)=h(7)=0.625, h(1)=h(6)=0.5, h(2)=h(5)=-0.875, h(3)=h(4)=-0.75. 输入数据的字长为7位，输出数据字长为16位。设输入为：

$x_0 = 31 = (1F)_H$，$x_1 = -49 = (4F)_H$，$x_2 = -17 = (6F)_H$

八阶FIR滤波器的顶层仿真结果见图4-19。

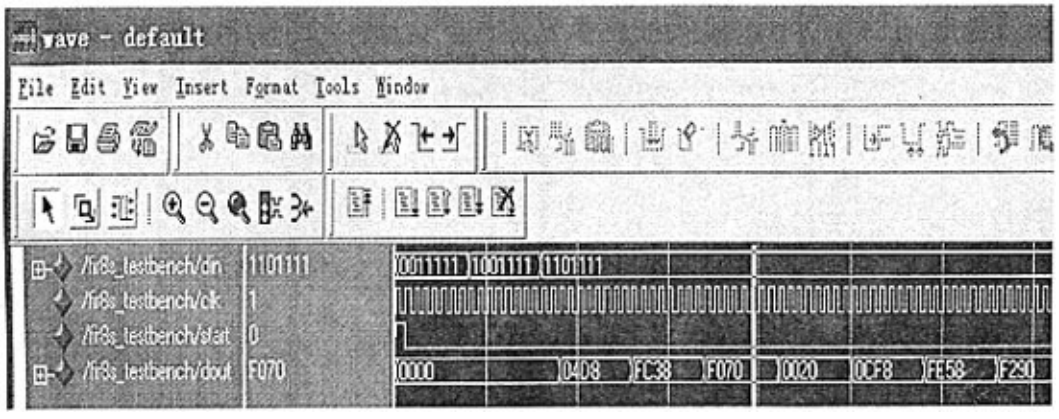


图4-19 八阶FIR滤波器顶层仿真结果

由于设计采用的是整型运算，在此将抽头系数放大64倍后，h(0)=h(7)=40 h(1)=h(6)=32, h(2)=h(5)=-56, h(3)=h(4)=-48, 因此通过Matlab可以求得

$y(1)=(1240)_{10}=(04D8)_{H补}$，$y(2)=(-968)_{10}=(FC38)_{H补}$，$y(3)=(-3984)_{10}=(F070)_{H补}$等等

对比硬件仿真结果完全一致，从而证明该8阶串行方式FIR滤波器设计满足要求。从仿真结果中，我们可以看到输出延迟输入信号18个时钟周期，每隔八个周期输出一个数据，与设计预期相符。

4.5 FIR 滤波器的数据分析

不论用并行方式还是串行方式实现FIR滤波器，对输入信号都要进行处理抽样和量化。数字滤波器作为一个数字系统，其中采用的滤波系数、输入与输出的序列值，以及运算过程中的结果，都是用有限字长的二进制数来表示的。通常采用的字长有8位、12位、16位和32位。所选字长愈短，运算误差愈大，其后果轻则使滤波器特性偏离原来的要求的特性，重则引起滤波器的不稳定而无法工作。但所选字长过长，会对硬件的要求太高使成本过高、价格昂贵，同时也使运算时间加长。有限字长效应对数字滤波器的影响主要表现在几个方面：

1）A/D转换的量化效应

如果信号用b+1位二进制数表示(量化)，其中一位表示符号，b位表示小数部分，则量化阶为2^{-b}。对于超过b位的部分要进行尾数处理，尾数处理有舍入法和截尾法两种。本论文中的数据采用的是截尾法，即将尾数第b+1位以及以后的数码略去。如果信号x(n)值量化后用$x_Q(n)$表示，则量化误差$e(n)=x_Q(n)-x(n)$。A/D变换器(由MATLAB软件完成)功能原理如图4-20所示，图中$x_Q(n)$是量化编码后的输出，如果未量化的二进制编码用$\hat{x(n)}$表示，那么量化误差$e(n)=x_Q(n)-\hat{x(n)}$，因此A/D变换器的输出$x_Q(n)=\hat{x(n)}+e(n)$。那么考虑A/D变换器的量化效应，其方块图如图4-21所示，这样，由于e(n)的存在而降低了输出端的信噪比。增加A/D变换器的位数，会增加输出端的信噪比，但同时也使后续硬件成本增加。因此，应根据实际需要，合理选择A/D换器的位数。

图4-20 A/D变换器功能原理图

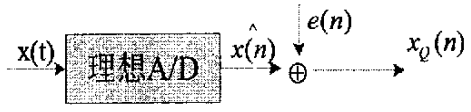


图4-21 考虑量化效应原理图

2) 数字运算过程中的有效字长效应

在网络运算中，其中间结果和最后结果的位数，如果超出了规定的有限位二进制数长度，则需要进行尾数处理，这样便引起了运算量化误差;运算中还可能出现溢出。例如，在定点制网络系统中，补码二进制0110加0011，结果为1001，其真值为--7，实际值为9。这样，由于加法进位，产生了溢出，形成了很大的误差。这种情况在编制LUT表中会出现。因此，在编制查找表时，把原来的h(n)的二进制补码数据的符号位进行了扩展，以保证得到正确的结果。

此外，滤波器系数量化也会对系统产生影响，在此不做讨论了。由上面的分析，不难看出量化中的有效字长效应是使结果数据偏离理论值的主要原因。要客观的分析结果数据，以判断设计的正确性。

第五章 总结与展望

在现代电子系统中，FIR数字滤波器以其良好的线性特性被广泛使用，属于数字信号处理的基本模块之一。本论文就基于FPGA器件实现FIR数字滤波器完成了以下研究：

首先以FIR数字滤波器的基本理论为依据，使用分布式算法为滤波器的硬件实现算法，并对其进行了详细的讨论。针对分布式算法中查找表规模过大的缺点，对其进行了优化方面的讨论，采用多块查找表以及线性FIR滤波器的对称性特点使得硬件规模极大的减小。

其次，针对基于FPGA硬件实现的特点，分别采用了并行和串行的设计方案，分别实现了8阶级联方式实现16阶线性FIR低通滤波器和8阶FIR低通滤波器，两种方案中都采用了流水线技术，通过对两种方式性能的比较，我们可以看出流水线在硬件设计中的重要性的同时，还可以得出并行设计运算速度快，但资源占用多；而串行方式资源占用少，但延迟长。并行方式每个时钟周期就可以完成整个运算，而串行方式，对于7位有效输入数据来说，完成整个运算需要8个时钟周期。因此，可以看出，在具体的设计当中要根据系统资源和具体设计要求两方面来具体的考虑。

最后，设计采用Xilinx Virtex-II系列器件，通过ISE6.1i软件对两种设计方案进行了综合仿真。为了更好的验证仿真结果的正确性，文中应用了MATLAB和VHDL联合仿真方法对设计的电路进行仿真测试，结果达到设计指标。通过MATLAB对仿真结果进行了分析，证明了所设计的FIR数字滤波器功能正确。

本文作为对硬件的方式设计FIR数字滤波器给出了比较通用的设计方法，通过修改LUT，我们可以很容易的实现高通，带通等FIR数字滤波器；对于设计中碰到的问题，如如何对数据进行量化处理，如何防止中间计算数据的溢出等问题，本文也给出了相应的解决方法。但是，设计中还有很多不足之处，在实际应用中，FIR数字滤波器做为一个模块在于外部器件通信时，还要添加一些状态信号，如RDY（数据输出准备）和RFY(数据输入准备)等信号。

作者认真进行了课题的研究并完成了本论文，由于作者水平有限，论文中可能仍有错误和不足之处，敬请大家批评指正。

参 考 文 献

[1]姜立东等编著.VHDL 语言程序设计及应用.北京：北京邮电大学出版社，2003.6

[2]褚振勇，翁木云编著.FPGA 设计及应用.西安：西安电子科技大学出版社，2002.7

[3]余成波，杨如民，周登义编著.数字信号处理及 Matlab 实现.北京：清华大学出版社，2005

[4]Xilinx，“Virtex-II Platform FPGA Handbook” January 31,2002

[5]靳希，杨尔滨，赵玲编著.信号处理原理与应用..北京：清华大学出版社，2004

[6]张志涌，徐彦琴等编著.Matlab 教程-基于 6.x 版本.北京：北京航空航天大学出版社，2001.4

[7](美)奥本海姆等编著，刘树堂，黄建国译.离散时间信号处理.西安：西安交通大学出版社，2001.9

[8](美)贝斯著，刘凌，胡永生泽.数字信号处理的 FPGA 实现.北京：清华大学出版社，2003.1

[9](美)恩格尔，普洛克斯编著，刘树棠译.数字信号处理:使用 MATLAB 西安：西安交通大学出版社，2002.6

[10]郭继昌，李香萍，滕建辅，基于位串行分布式算法和 FPGA 实现 FIR 电路的研究，电子测量与仪器学报，2001.6

[11]蒋亚坚，张庆雷.分布式运算单元的原理及其实现方法，电子技术应用，2000.2

[12]侯伯亨，顾新编著.VHDL 硬件描述语言与数字逻辑电路设计.西安:西安电子科技大学出版社，2003

[13]徐志军，徐光晖编著.CPLD/FPGA 的开发与应用.北京:电子工业出版社，2002.1

[14]Atmel Corporation,USA, Application note, "FPGA-based FIR Filter Using Bit-Serial Digital Signal Processing", by Lee Ferguson,1995

[15]Stanley A. White, "Application of Distributed Arithmetic to Digital Signal Processing: A tutorial Review", IEEE ASSP Magazine, July 1989,p4-19

[16]The Programmable Logic DATA BOOK. Xilinx Incorporation San Jose USA, 1999

[17]G.R.Goslin, ``A guide to Using Field Programmable Gate Arrays for Application-Specific Digital Signal Processing Performance", www.xinlinx.com.dsp

[18]A. Peled, B. Liu, "A New Hardware Realization of Digital Filters", IEEE Trans. on ASSP, Vol.22, December 1974, pp456-462

[19]Javier Valls, Marcos M. Peiro, Trini Sansaloni, Eduardo Boemo, "A Study About FPGA-Based Digital Filters", IEEE SIPS, pp191-201,October 1998

[20]K.K.Parhi, "A systematic approach for design of digit-serial processing architectures," IEEE Trans, Circuits and System, Vol. 38, pp358-375, April 1991

[21]Allred, D.J.; Heejong Yoo; Krishnan, V.etc; A novel highperformance distributed arithmetic adaptive filter implementation on an FPGA. IEEE International Conference on，Vol: 5，17-21 May 2004

[22]Shousheng He, Mats Torkelson, "FPGA Implementation of FIR Filters Using Pipelined Bit-Serial Canonical Signed Digit Multipliers", IEEE 1994 Custom Integrated Circuits Conference.

[23]Xilnx, "modelsim vhdl simulation tutorial", UG102 (v1.1) July 21, 2000

[24]Altera, "FIR Filters", A-FS-OI-O1, www.altera.com

[25]Huang, W., Krishnan, V., Allred, D.,ect. Design analysis of a distributed arithmetic adaptive FIR filter on an FPGA. Signals, Systems&Computers, 2003 Vo1.1,9-12 Nov.:926—930

硕士期间参与的科研工作和发表的论文

1. 参与西安桑瑞微电子有限公司 DVD 刻录控制芯片设计项目

2. 刘朋全，徐建城. 基于 FPGA 实现 PWM 模块的方法研究，已被《微处理机》录用，拟在 2006 年第 4 期刊发。

致 谢

作者由衷地感谢导师徐建城教授。导师以他渊博的知识、严谨的治学态度以及开明、自由创新的良好学术作风关心并指导着作者的工作，使论文得以顺利的完成。徐建城教授的高深理论造诣以及在科学研究中远见卓识给作者留下了深刻的印象，所有这些都将使我终生受益。在此，作者向导师徐建城教授表达深深的敬意和谢意。

在整个研究生期间，本教研室的于海勋教授以及各位师兄在我的学习和生活的各个方面给予了作者很多的关心和帮助。谨在此对电路与系统教研室全体老师和同学表示我衷心的感谢。

最后，深深感谢辛勤养育我的父母，感谢他们这么多年来对我物质上的支持和精神上的鼓励，是他们的付出和支持，才使我顺利完成了今天的学业。

西北工业大学

学位论文知识产权声明书

本人完全了解学校有关保护知识产权的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属于西北工业大学。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版。本人允许论文被查阅和借阅。学校可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律注明作者单位为西北工业大学。

保密论文待解密后适用本声明。

学位论文作者签名：刘朋全　　指导教师签名：[illegible]

2006年 3 月 30日　　2006年 3 月 30日

西北工业大学

学位论文原创性声明

秉承学校严谨的学风和优良的科学道德，本人郑重声明：所呈交的学位论文，是本人在导师的指导下进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容和致谢的地方外，本论文不包含任何其他个人或集体已经公开发表或撰写过的研究成果，不包含本人或其他已申请学位或其他用途使用过的成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式表明。

本人学位论文与资料若有不实，愿意承担一切相关的法律责任。

学位论文作者签名：刘朋全

2006年 3月30 日

基于FPGA的FIR数字滤波器的设计和实现

作者：刘朋全
学位授予单位：西北工业大学

参考文献(25条)

1. 姜立东 VHDL语言程序设计及应用 2003

2. 褚振勇.翁木云 FPGA设计及应用 2002

3. 余成波.杨如民.周登义 数字信号处理及Matlab实现 2005

4. Xilinx Virtex-Ⅱ Platform FPGA Handbook 2002

5. 靳希.杨尔滨.赵玲 信号处理原理与应用 2004

6. 张志涌.徐彦琴 Matlab教程-基于6.x版本 2001

7. 奥本海姆.刘树堂.黄建国 离散时间信号处理 2001

8. 贝斯.刘凌.胡永生 数字信号处理的FPGA实现 2003

9. 恩格尔.普洛克斯.刘树棠 数字信号处理:使用MATLAB 2002

10. 郭继昌.李香萍.滕建辅 基于位串行分布式算法和FPGA实现FIR电路的研究[期刊论文]-电子测量与仪器学报 2001(2)

11. 蒋亚坚.张庆雷 分布式运算单元的原理及其实现方法[期刊论文]-电子技术应用 2000(2)

12. 侯伯亨.顾新 VHDL硬件描述语言与数字逻辑电路设计 2003

13. 徐志军.徐光晖 CPLD/FPGA的开发与应用 2002

14. Lee Ferguson Application note,"FPGA-based FIR Filter Using Bit-Serial Digital Signal Processing" 1995

15. Stanley A White Application of Distributed Arithmetic to Digital Signal Processing:A tutorial Review 1989

16. The Programmable Logic DATA BOOK 1999

17. G R Goslin A guide to Using Field Programmable Gate Arrays for Application-Specific Digital Signal Processing Performance

18. A Peled.B Liu A New Hardware Realization of Digital Filters 1974

19. Javier Valls.Marcos M Peiro.Trini Sansaloni.Eduardo Boemo A Study About FPGA-Based Digital Filters 1998

20. K K Parhi A systematic approach for design of digit-serial processing architectures 1991

21. Allred D J.Heejong Yoo.Krishnan V A novel highperformance distributed arithmetic adaptive filter implementation on an FPGA 2004

22. Shousheng He.Mats Torkelson FPGA Implementation of FIR Filters Using Pipelined Bit-Serial Canonical Signed Digit Multipliers

23. Xilnx "modelsim vhdl simulation tutorial"UG102 (v1.1) 2000

24. Altera "FIR Filters",A-FS-01-01

25. Huang W.Krishnan V.Allred D Design analysis of a distributed arithmetic adaptive FIR filter on an FPGA 2003(9-12)

相似文献(10条)

1. 学位论文 刘敏 高密度可编程逻辑器件实现参数化FIR滤波器的研究 2000

该文从FIR滤波器的重要性和实用性出发,研究了其专用硬件实现这一实际技术问题;从FIR滤波器的基本理论出发,根据其系数相对输入数据是固定的这一特点,首先在传统结构基础上得出减少乘、加次数的优化结构,并利用密度可编程逻辑器件CPLD中的查找表LUT结构实现向量乘法器,进一步解决实现乘法的速度问题并达到节省器件资源简化系统设计的目的.最后采用自顶向下(top_down)的模块化设计方法,围绕优化结构设计出便于硬件实现的并、串行FIR滤波器层次结构图.该课题选用ALTERA公司功能强大的可编程逻辑器件芯片开发软件MAX+PLUSⅡ,对组成并、串层次结构图的各个功能模块进行了大量的逻辑设计与功能仿真,直至正确;将并、串FIR滤波器适配到FLEX 10K系列芯片中,达到系统的单片集成.系统仿真测试结果表明:系统实现了满足设计要求的并、串参数化FIR滤波器,即设计者可以通过修改参数达到方便地更改滤波器性能;实现的FR滤波器与以往的FIR滤波器相比,速度得以大大提高,尤以并行方式为最,可达到105MSPS(Million Samples Per Second).

2. 期刊论文 刘志新. 吴淑泉. LIU Zhi-xin. WU Shu-quan FIR 滤波器的 CPLD 参数化模块设计 -华北工学院学报 2000,21(4)

目的介绍采用可编程逻辑器件 CPLD 对 FIR 滤波器参数化模块设计的原理和技术. 方法根据 FIR 滤波器的卷积表示式, 用 CPLD 的查找表来实现 FIR 滤波器. 结果采用此方法可以用 CPLD 方便地设计不同阶 FIR 滤波器. 结论模块参数化设计能够快速设计 FIR 滤波器.

3. 会议论文 康长武. 李景华 FIR滤波器在可编程逻辑器件上的实现 2002

主要讨论如何在可编程逻辑器件上更有效地实现FIR滤波器,同时也讨论如何在FIR滤波器的设计中采用流水线技术,并以8阶FIR滤波器为例说明了它的结构、实现以及结果仿真.

4. 学位论文 郭晓宇 基于FPGA实现FIR数字滤波器的研究 2004

在现代电子系统中,FIR数字滤波器以其良好的线性特性被广泛使用,属于数字信号处理的基本模块之一.在工程实践中,往往要求对信号处理要有实时性和灵活性,而已有的一些软件和硬件实现方式则难以同时达到这两方面的要求.随着可编程逻辑器件和EDA技术的发展,使用FPGA来实现FIR滤波器,既具有实时性,又兼顾了一定的灵活性,越来越多的电子工程师采用FPGA器件来实现FIR滤波器.该文对基于FPGA的FIR数字滤波器实现进行了研究.该论文所做的主要工作如下:1.以FIR数字滤波器的基本理论为依据,使用分布式算法为滤波器的硬件实现算法,并对其进行了详细的讨论.针对分布式算法中查找表规模过大的缺点,采用多块查找表和OBC编码方式使得硬件规模极大的减小.2.在设计中采用了层次化、模块化的设计思想,将整个滤波器划分为多个功能模块,利用VHDL语言和原理图输入两种设计技术进行了各个功能模块的设计,最终完成了FIR数字滤波器的系统设计.3.最后给出了采用FLEX10K系列器件实现一个16阶的FIR低通滤波器的设计实例,用MAX+PLUSII软件进行了仿真,并用MATLAB对仿真结果进行了分析,证明所设计的FIR数字滤波器功能正确.仿真结果表明,该论文设计的滤波器硬件规模较小,采样率达到了8.8MHz.同时只要将查找表进行相应的改动,就能分别实现低通、高通、带通FIR滤波器,体现了设计的灵活性.

5. 期刊论文 韩德红. 石新智 基于CPLD器件的FIR滤波器的设计 -武汉大学学报(理学版)2004,50(1)

给出了一种适合于用CPLD器件实现有限冲击响应(FIR)滤波器的补码算法. 用Lattice公司的ispLSI8840器件设计了8阶、11位的线性相位FIR滤波器,并提出了用多片CPLD器件进行扩展设计的方法,实现了更高阶的线性相位FIR滤波器.通过小型乘法查找表和具有超前进位的流水线加法器实现FIR滤波器的设计,提高了工作速度,节约了器件资源, 其最高工作频率可达60 MHz.在计算机上进行了硬件仿真分析,并将仿真结果与理论计算结果进行了比较.表明该滤波器工作可靠,精度高,具有较好的实用价值.

6. 学位论文 胡光荣 用VHDL语言设计基于FPGA器件的高采样率FIR滤波器 2002

该文围绕硬件描述语言在数字硬件系统设计中的应用来展开的.首先从传统的数字硬件系统设计方法与采用硬件描述语言的数字硬件系统设计方法的特点出发,介绍了EDA发展的过程、VHDL语言特点及ALTERA的FLEX10K的结构特点.重点介绍了在数字算法设计和实现中基于ALTERA公司的FPGA器件四输入查找表结构的FIR滤波器流水线设计技术,和结合先进的EDA软件进行高效设计的方法和途径,给出了设计的仿真结果.该设计能满足高采样率的要求,设计效率高,对FPGA硬件资源的利用高效合理.而且文中提到的基于流水线技术的算法分解方法可推广应用到其它需要高速数字算法实现的领域中,从而充分挖掘和利用FPGA的高速特性.

7. 期刊论文 唐治德. 刘敏. 刘晓明 用高密度可编程逻辑器件实现参数化FIR滤波器 -重庆大学学报(自然科学版) 2002,25(3)

从提高FIR滤波器的处理速度出发,在传统结构的基础上导出减少乘、加次数的优化结构,并利用FLEX器件系列中的查找表LUT结构构成向量乘法器,快速完成乘、加运算,提高滤波器的工作速度并节省器件资源.最后,利用Altera公司的MAX+PLUSII软件进行了并、串FIR滤波器的逻辑设计,达到了硬件实现参数化FIR滤波器的目的.

8. 期刊论文 王秀敏. 汪毓铎. 张洋. 杨世华. WANG Xiu-min. WANG Yu-duo. ZHANG Yang. YANG Shi-hua 通信系统中FIR数字滤波器的设计研究 -通信技术2009,42(9)

文中基于FPGA技术设计了一个16阶FIR数字低通滤波器.采用分布式算法作为滤波器的硬件实现算法.通过将FIR滤波器的乘加运算转化为查找表,极大提高了FIR滤波器的速度.在程序设计中采用了层次化、模块化的设计思想,将整个滤波器划分为多个功能模块.仿真结果表明:文中设计的滤波器硬件规模较小.同时只要将查找表进行相应的改动,就能分别实现低通、高通、带通FIR滤波器,体现了设计的灵活性.

9. 期刊论文 冯祥 基于ISP可编程逻辑器件的线性相位FIR滤波器 -半导体技术2000,25(4)

基于ISPExpert系统和ISP可编程逻辑器件,实现了线性相位FIR滤波器,该滤波器性能稳定、抗干扰能力强、能快速实现功能重构,可广泛应用于雷达信号处理等领域.

10. 学位论文 吴国庆 软件无线电中高效FIR滤波器的研究与设计 2005

本文结合软件无线电多抽样率信号处理的基本理论设计窄带FIR滤波器，并对其进行了MATLAB仿真。 在软件无线电高采样率的条件下实现窄带FIR滤波器的设计，主要是应用多抽样率信号处理的理论对信号进行抽取、内插、等效变换和多级抽样率转换来实现的。通过抽取器将抽样率降低，在低抽样率条件下，设计出符合性能要求的FIR滤波器。然后，通过内插器将抽样率升高到一个需要的数值上。在设计中，当抽样率转换的转换因子(抽取因子D或内插因子I)过大时，一次完成抽样率转换工作是不现实的，这对硬件的速度和存储量都提出了较高的要求。因此，往往经过两次或两次以上转换实现多级抽样率转换。 本设计中抽取器和内插器的多级实现是基于积分梳状滤波器(CIC)和半带滤波器(HBF)的，主要原因在于积分梳状滤波器不需要乘法器，半带滤波器有近一半系数为0。这样做的好处在于计算效率高，并且为可编程逻辑器件的实现奠定了良好的基础。 在分析窄带FIR滤波器的多级实现原理和基本特性的基础上，给出窄带FIR滤波器的详细结构。通过实例分析了所设计FIR滤波器的频率特性、抗混叠特性、去镜象特性，说明了采用多抽样率结构设计窄带的FIR滤波器能够有效地从通带宽度、通带波纹、矩形系数、阻带衰减等方面对窄带FIR滤波器的性能进行控制。

引证文献(2条)

1. 郑传家. 屈德新. 邱晓军. 周铁 数字下变频的FPGA实现[期刊论文]-电子产品世界 2008(12)

2. 覃岭. 邓小炜. 何子述. 段军棋 宽带数字下变频的高效实现方法研究[期刊论文]-电子科技大学学报 2008(5)