

原创性声明

郑重声明：此篇题为《基于 FPGA 的高性能离散小波变换设计》的论文是作者在导师的指导下，于武汉大学攻读硕士学位期间，进行研究工作所取得的成果。根据作者所知，论文中除了参考文献列举的地方外，不包含其他人已经发表或撰写过的研究成果。

作者签名：王亚娟 倪奎 旷捷 王安文

导师签名：黄启俊 常胜

撰写日期：二零零九年八月十日

基于 FPGA 的高性能离散小波变换设计

作者：王亚娟 旷捷 倪奎 王安文

导师：黄启俊 常胜

(武汉大学物理科学与技术学院电子科技系，武汉，430072)

摘要：在数字信号处理领域，小波变换无论在理论研究还是工程应用方面都具有广泛的价值，因此高性能离散小波变换的 FPGA 实现架构的研究就显得尤为重要。本文针对 db8 (Daubechies 8) 小波设计了一个 16 阶 16 位的正、反变换系统，用 DE2 开发板进行了系统验证，在 FPGA 的逻辑单元资源消耗 12% 的情况下，正、反变换的最高时钟频率分别达到了 217.72MHz、217.58MHz。对比同类文献，本设计在最高处理速度方面具有明显的优势。在此基础上，考虑到通用性的要求，本文还设计了一种小波种类可选、小波阶数可调的通用小波变换 FPGA 架构，该通用小波正、反变换系统的最高时钟频率分别为 114.10 MHz、152.09 MHz。此结构具有通用性强的特点，可高性能实现多种小波变换。设计采用 DA(Distributed Arithmetic) 算法和 LUT(Look-Up Table) 技术来实现小波变换中的滤波器，并使用流水线结构以及调用 Altera 提供的 IP 核完成了设计的优化，用 MATLAB 验证了设计的功能。

关键词：DWT; IDWT; FPGA; Mallat 算法; DA 算法

Design of High-performance Discrete Wavelet Transform Based on FPGA

Author: Wang Yajuan, Kuang Jie, Ni Kui, Wang Anwen

Supervisors: Huang Qijun, Chang Sheng

Abstract: In the field of digital signal processing, the wavelet transform has great value of research in both theoretical study and engineering applications. Therefore, it would be significant to study the FPGA implementation architecture of high-performance discrete wavelet transform. The present paper firstly proposes a high-speed FPGA architecture of discrete wavelet transform (DWT) as well as inverse DWT (IDWT), and using Daubechies 8 wavelet implements a system whose filter order is 16 and the data width

operated is 16 bits. The system is verified on the DE2 development board, and after optimized, when the resource consumption is 12%, the maximum clock frequency of DWT and IDWT can respectively achieve 217.72MHz and 217.58MHz. Compared with similar work in other literatures, the design has an obvious advantage of the maximum processing speed. On this basis, considering the request of versatility, a universal FPGA architecture with optional wavelet and adjustable wavelet filter order is also designed in this paper. The maximum clock frequency of DWT and IDWT in this general architecture can respectively achieve 114.10 MHz and 152.09 MHz. What's more, the proposed architecture is characteristic of versatility and can efficiently implement varieties of DWT and IDWT. In design scheme, DA(Distributed Arithmetic) and LUT(Look-Up Table) technology are used to implement the filter in wavelet transform, pipeline architecture and Quartus II built-in IP core are used to optimize the design. Finally compared with the MATLAB calculation results, the correctness of the system results is verified.

Keywords: DWT; IDWT; FPGA; Mallat Arithmetic; Distributed Arithmetic

0 引 言

随着现代数字信号处理技术的高速发展,小波变换起到了越来越重要的作用。但小波变换数据依赖关系复杂,运算量大,靠软件实现不能达到实时处理的要求。因此其硬件实现就被提到日程上来,但现阶段的研究仍处于初步阶段,且存在较大的局限性。FPGA 可以通过并行和流水线设计,实现高速的信号处理,且具有可重构能力。因此,研究离散小波变换的高速、通用 FPGA 实现技术,将对小波变换的硬件实现和应用具有很好的借鉴意义。

目前离散小波变换硬件实现的文献比较多,大部分针对具体小波函数设计:

龚敏元基于 Mallat 算法进行了设计,采用 APEX20KE 系列 FPGA 芯片对 db2(Daubechies 2)小波综合,正、反变换的最高时钟频率分别为 96.35MHz 和 110.44MHz^[1]。王胜坤采用提升算法进行了设计,实现的基于流水线并行阵列结构的 9/7 提升小波电路,采用 Xilinx Spartan-III 综合后的资源占用为 43%,最高时钟频率为 151.722MHz^[2]。Patrick Longa 采用 DA(Distributed Arithmetic)算法实现了全并行查找表结构的小波滤波器组。采用此方法实现 db8 小波,Altera Stratix II FPGA 综合结果表明需要 614 个逻辑门单元,达到的最高时钟频率为 149.3MHz^[3]。S.V.Silva 通过提升算法实现了 5 种方案的 9/7 小波 DWT。从这五种方案综合结果的比较中可以看出,即使是优先考虑时钟频率的设计,采用 APEX 20KE FPGA 器件综合所达到的最高时钟频率只有 157MHz,对于一些高速应用场合,不能满足实时需求^[4]。

涉及到通用小波变换的 FPGA 实现的文献较少,典型工作如下:

顾文军提出了小波变换硬件实现的一种架构,该架构可编程滤波器的阶数、变换级数、变换数据段长度 N(即帧长)、变换种类(小波变换及反变换)。采用 APEX 系列 FPGA 综合:此系统占用了 1835 个逻辑单元,10172bits 存储单元,但最高工作频率只有 29MHz^[5]。S.Masud 等设计了一个离散小波变换的 IP 核,该系统可以选择小波种类、数据宽度和分解级数等。对 9/7 小波采用

Xilinx 4052XL 综合,当分解级数为 1,输入数据宽度为 16bits,系数宽度为 9bits 时,总共占用 1097 个 CLBs,最大时钟频率为 78.85MHz^[6]。

综上,目前离散小波变换的 FPGA 实现存在以下缺陷:

1. 大部分文献是针对具体的小波函数来实现 DWT 和 IDWT,通用性不好。而且就所查的文献来看,目前的小波变换 FPGA 实现系统的速度没有做到很高。
2. 少数文献涉及到了离散小波变换的通用实现。文献[5]、[6]涉及到了离散小波变换通用系统的实现,但是系统最大时钟频率分别限制在 29MHz 和 78.85MHz,系统的时钟频率有待提高。
3. 国外现有的小波变换芯片^[7],也只是针对具体的小波函数,且实用性不强。

基于上述问题,本文提出如下设计目标:

- 高速实现具体小波的 FPGA 正、反变换系统,要求该系统具有较高的时钟频率,可以满足大部分实时系统的需求:系统最高时钟频率 $\geq 200\text{MHz}$ 。
- 考虑到目前的小波变换系统过分依赖于具体的小波函数和分解级数,本文尝试实现具有通用性的 FPGA 小波正、反变换系统,要求该系统具有通用性、可编程性和实时性,系统最高时钟频率 $\geq 100\text{MHz}$,且应能进行参数化设计:
 - (1) 在不同场合下使用不同的小波函数。
 - (2) 按照要求不同,实现不同阶数的小波变换。

1 算法原理及实现思路

1.1 Mallat 算法

Mallat 算法^[8]是由法国科学家 Stephane Mallat 于 1988 年提出的一种离散小波变换 (DWT) 的快速算法。该算法的地位相当于快速傅里叶变换在经典傅里叶分析中的地位,成为各种硬件实现的直接理论依据。Mallat 算法的基本思路是迭代运算,即先计算出第一级小波变换,然后在此基础上计算下一级小波变换,如此重复下去。

假设有一组离散信号 $x(n)$,通过一系列的低通和高通滤波器,分别可以得到近似 (approximation) 信号和细节 (detail) 信号。由于经过一组滤波器后频率谱范围会减少一半,根据 Nyquist 采样定律,可以减少一半的采样率,所以滤波器的输出为:

$$\begin{cases} y_{low}[n] = \sum_{k=0}^{N-1} x[k]H_0[2n-k] \\ y_{high}[n] = \sum_{k=0}^{N-1} x[k]H_1[2n-k] \end{cases} \quad (1)$$

在实际应用中,低通和高通滤波器是相互关联的,它们之间的关系式为 $H_0(n) = (-1)^n H_1(2N-1-n)$ 。用滤波器组来实现 Mallat 算法,可以使小波变换的计算量明显减少。如图 1 所示,为离散信号的 Mallat 算法分解过程:

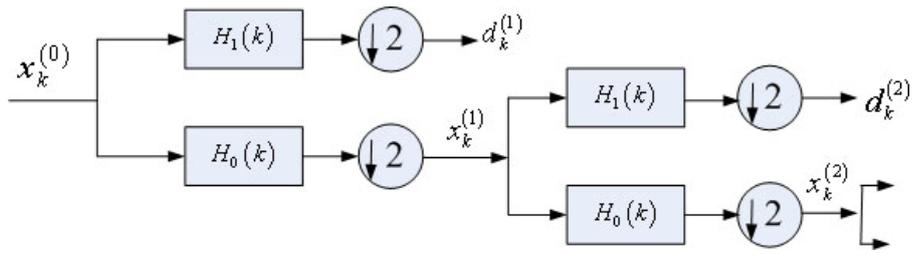


图 1 离散信号的 Mallat 算法分解

信号重建是信号分解的逆过程，但需注意以下两点：一是“二抽取”换成“二插值”，然后滤波。二是综合滤波器 G_0, G_1 与分析滤波器 H_0, H_1 未必一样。其中，各滤波器间关系如下：

$$\begin{cases} G_0(n) = -(-1)^n H_1(n) \\ G_1(n) = (-1)^n H_0(n) \end{cases} \quad (2)$$

则离散信号小波重构的 Mallat 算法如图 2 所示。

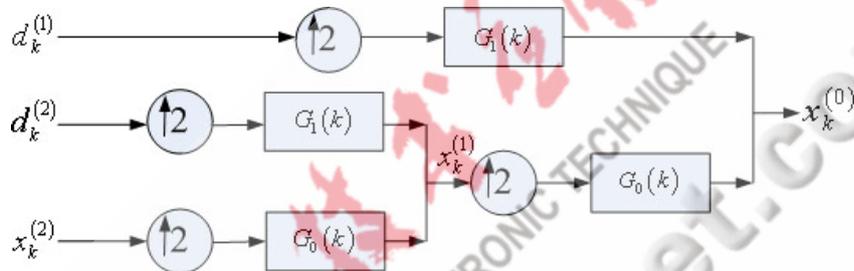


图 2 离散信号的 Mallat 算法重构

本文只针对一维、一级离散小波变换作处理，因为多级小波变换的处理方式具有重复性，所以只要将一级小波变换作为基本模块，通过级联的方式就可以完成多级小波变换。另一方面，实现了一维离散小波变换，二维小波变换先完成行方向的滤波，再完成列方向的滤波即可实现。

1.2 DA 算法及其实现

上述 Mallat 算法中的核心滤波模块，采用 DA 算法，即分布式算法实现。分布式算法作为一项数字信号处理算法，广泛地应用在计算乘积和之中。与传统的乘积和结构相比，具有并行处理的高效性特点。

DA 算法的原理推导如下：

一个 N 阶的 FIR 滤波器的输出可由线性卷积求出：

$$y[n] = \sum_{i=0}^{N-1} h[i]x[n-i] \quad (3)$$

对于传统的 MAC，每次采样， $y[n]$ 都要经过 N 次乘法和 $N-1$ 次加法操作，在硬件实现中，既占用资源，又影响速度。

由于 FIR 滤波器的系数具有对称性，下面看乘积和公式：

$$y[n] = \sum_{n=0}^{N-1} h[n]x[n] \quad (4)$$

假设 FIR 系数 $h[n]$ 已知。有符号的 $x[n]$ 可以用 B 位的二进制补码表示为：

$$x[n] = x_{B-1}[n] \times (-2^{B-1}) + \sum_{b=0}^{B-2} x_b[n] \times 2^b \quad (5)$$

其中，最高位为符号位。将公式 (5) 代入公式 (4)，可得：

$$\begin{aligned} y[n] &= -2^{B-1} \sum_{n=0}^{N-1} h[n]x_{B-1}[n] + \sum_{n=0}^{N-1} h[n] \sum_{b=0}^{B-2} x_b[n] 2^b \\ &= [h[0]x_{B-1}[0] + h[1]x_{B-1}[1] + \dots + h[N-1]x_{B-1}[N-1]] \cdot (-2^{B-1}) \\ &\quad + [h[0]x_{B-2}[0] + h[1]x_{B-2}[1] + \dots + h[N-1]x_{B-2}[N-1]] \cdot 2^{B-2} \\ &\quad + \dots \\ &\quad + [h[0]x_0[0] + h[1]x_0[1] + \dots + h[N-1]x_0[N-1]] \cdot 2^0 \end{aligned}$$

上式称为分布式算法^[9]。

由上式可知，分布式算法的实质就是：将输入变量的一个数据位和所有滤波器抽头系数 $h(0) \sim h(N-1)$ 的每一位进行“与”运算并求和，而指数部分则说明了求和结果的位权，整数乘以 2^b 就是左移 b 位，对此可以通过硬件连线实现，不占用逻辑资源。这样就可以通过建立查找表来实现上述运算，查找表可用所有输入变量的同一位进行寻址。

因此，采用分布式算法实现的 FIR 滤波器在 FPGA 中的工作速度只与输入数据的宽度 B 有关，与滤波器的阶数 N 无关，阶数只影响 FPGA 资源的使用量。

如果考虑具体的 FPGA 实现，该算法需要做如下优化^[9]：

(1) 串行级联查找表结构——优化资源

如果滤波器抽头数（即滤波器阶数）过多，用单个查找表就不能执行全字（因为查找表位宽=滤波器抽头数）。因为抽头数大于 5 的查找表并不节省硬件资源^[9]。在这种情况下，必须采用串行级联方式：将表的地址输入位数（即滤波器抽头数 N）进行降低，可以将 N 输入的查找表分割成 L 个 N/L 输入查找表的级联，再将各个查找表输出的值求和。这种通过查找表的级联实现高阶滤波器的方法指数倍地节省了硬件资源。

如果再加上流水线寄存器，这一改进并没有降低速度，但是却可以极大地减少设计规模。所以通过对高阶查找表的地址进行分割，会使系统性能提升很多。

(2) 多位并行查找表结构——优化时钟

当输入数据宽度 B 增大时，滤波器的工作速度将降低。因此，对位宽较大的输入需采用并行方式：使输入数据相同的位对应于同一个查找表，不同的位对应于不同的查找表；再将几个查找表的输出结果求和。这样，每次处理多个位，速度会大大提高。当然，这种改进是以增加资源的使用量为代价的。

综上，串行结构在节约资源的同时并不能提高滤波器的速度，而并行结构正好弥补了串行结构在这方面的不足，采用多位同时处理的方式极大地提高了效率，使计算速度可以从根本上翻倍，进而适应数字信号实时处理的要求。如果二者联合使用，就可以很好的改善系统性能。

由上面讨论可知，改进后的 DA 算法的 FPGA 实现结构如图 3 所示：

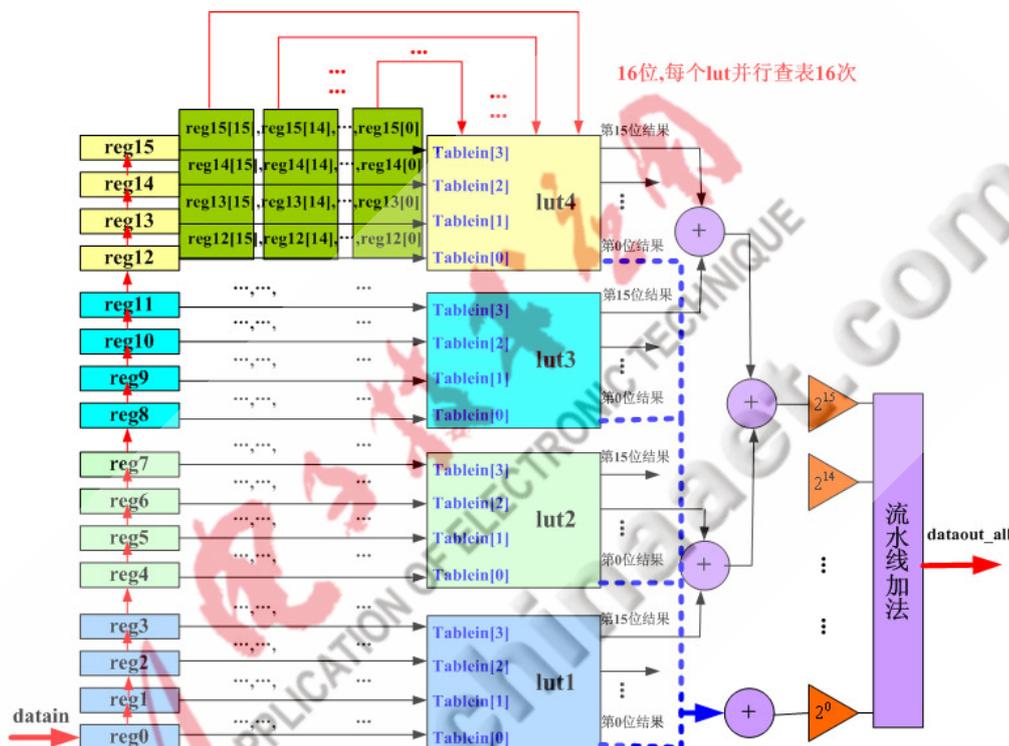


图 3 DA 算法的 FPGA 实现结构

1.3 系统FPGA实现思路

● 精度设置问题

信号处理系统若用浮点运算实现的话，虽有精度保证，但硬件复杂度和成本都太高。而小波变换目前更多的应用于图像、声音等数字信号处理领域，而这些信号本身就多为整数（图像一般用位数较低的整数表示），因此用定点实现是比较合适的。因此，下面分析考虑有限字长带来的精度问题。

对于小波变换的计算过程而言，最终数据结果的精度取决于输入数据、滤波器系数以及中间变换结果的精度。因此，本系统具体的精度设置如下：

1. 输入数据

考虑到实际背景，将小波变换用于数字信号处理领域时一般字长不会超过 16bits，因此输入数据位数选取为 16bits(其中包括符号位)。

2. 滤波器系数

实验表明：当滤波器系数的位数取 16bits 时，其精度已经足够^[5]。因此，本系统设置滤波器系数的位宽为 16bits(其中包括符号位)。

另外，考虑到常用的一些小波函数对应的变换滤波器系数的绝对值均偏小，不适合采用 FPGA 设计，故在设计之前先对滤波器系数放大，考虑到实际小波滤波器系数的大小，将各系数分别左移 13 位（乘以 $2^{13}=8192$ ），然后四舍五入，转换为整数参与 FPGA 运算。

3. 中间变换结果的精度

由于输入数据位宽为 16bits，而如上也对小波滤波器系数左移了 13bits，在系统的最后必定要右移 13 位，以得到正确的结果输出，因此，设置中间变换结果的精度为 $16+13=29$ bits,以防止溢出现象的发生。

4. 输出数据

考虑到实际情况，本设计输出数据位宽为 16bits 已足够，不会发生溢出^[5]。因此，输出数据位宽设置为 16bits。

● 边界延拓问题

在利用 Mallat 算法进行快速离散小波变换时，要注意卷积的边界效应问题。解决边界效应的办法就是进行边界延拓。而不同的延拓方式，对于重构信号所造成的误差不同，因此，需要选择合适的边界延拓方式进行系统设计。通常采用的延拓方法有零延拓、周期延拓、对称周期延拓等。

考虑到零延拓法误差较大，对称周期延拓法对使用的小波滤波器组提出了对称性的要求，而周期延拓法对滤波器没有特殊的要求，并且误差也处于可以接受的范围之内。因此，本文采用周期延拓法。

2 一维 DWT 系统的高速 FPGA 实现

在前述 FPGA 设计思路下，下面针对 db8 小波进行正变换和反变换系统的设计，要求该系统具有很好的实时性。另外，由于小波反变换的设计是正变换的逆过程。因此，下面重点说明正变换的设计，反变换不再详述。

2.1 系统设计

基于前述 Mallat 算法，本文提出小波正变换的系统架构，如图 4 所示：

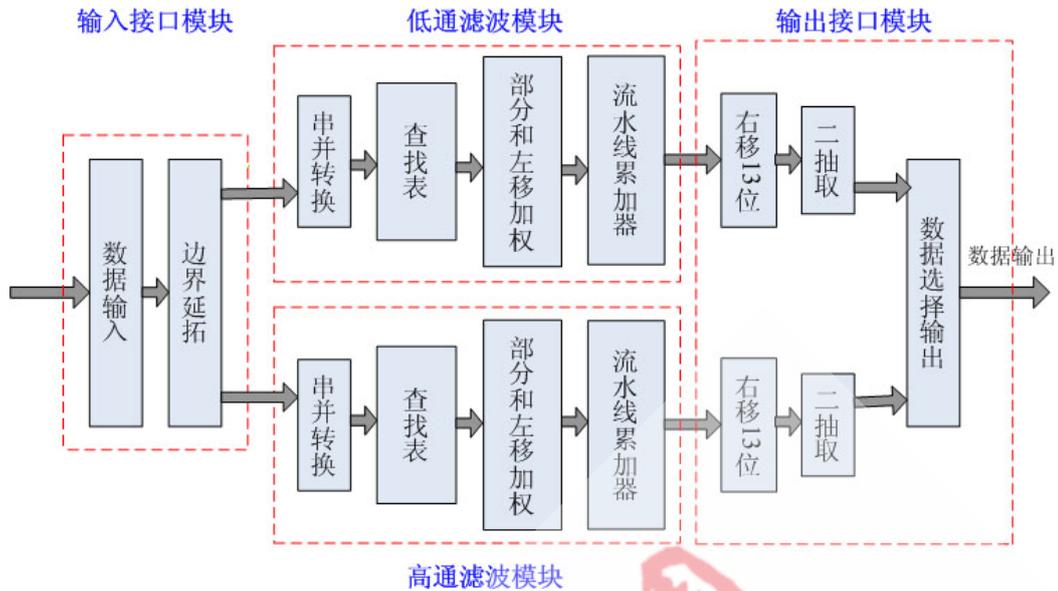


图 4 小波正变换系统构架

主要模块设计思想：

1. **边界延拓：** 由于对输入数据进行边界延拓，必然使得系统的算法效率下降。考虑到系统高性能的要求，以及输入测试数据的长度不会很长。因此，本系统不单独做模块进行边界延拓，而是对输入信号进行延拓之后作为系统输入。这样可以既保证系统精确度的要求，又达到较高的系统效率。
2. **滤波模块：** 滤波模块采用前述的经改进后的 DA 算法实现，其 FPGA 实现结构如图 3 所示。
3. **串并转换：** 串并转换单元采用移位寄存器实现，而滤波模块采用 DA 算法。因此，串并转换之后的输出，可直接进行组合作为 DA 查找表的输入。
4. **查找表和右移 13 位：** 由于查找表中滤波器系数很小，所以在系统设计之前，就对各个系数左移 13 位，即扩大了 8192 倍。因此，在滤波结束后，应添加右移 13 位模块，以得到正确的结果。
5. **二抽取：** 二抽取通过调用 Altera 的 IP 核—FIFO 实现。调用 IP 核，使得该模块具有很高的效率，并且该部分结构和功能均已得到了很好的优化。二抽取的 FPGA 实现结构如图 5 所示：

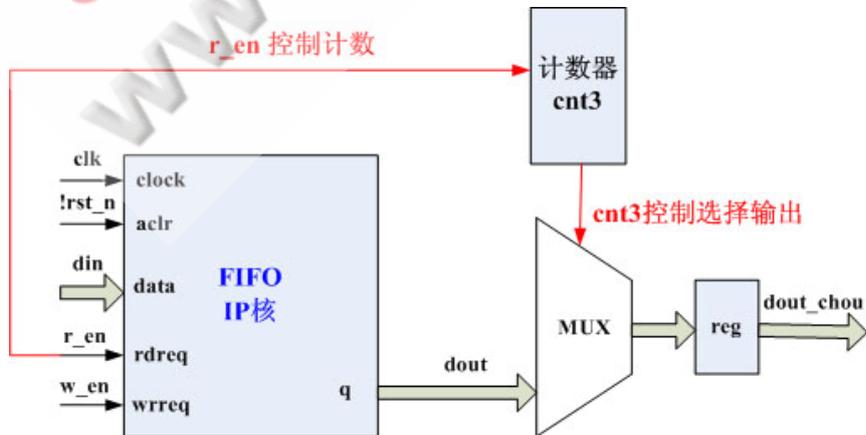


图 5 二抽取的 FPGA 实现结构

6. 关于数据选择输出：由高、低通输出使能信号，决定是否输出结果序列。

具体各个模块的设计不再详述。

2.2 系统优化

本系统的性能还可做大量优化。由于本设计有较长的加法器链，所以系统时钟总是被限制在一个很低的水平上。为了更好的满足系统实时性的设计目标，在第 1 节所述的改进后的 DA 算法的 FPGA 实现基础上，提出流水线式的加法器链结构，大大的提高了系统的速度。

具体做法是：将一大串数据的直接相加进行分解，分解成很多个 2 输入的加法器结构。与此同时，增加一些中间寄存器，寄存中间结果。这样，在时钟边沿的触发下，每个 2 输入加法器并行执行，大大提高了算法的执行速率，从而提高系统速度。

通过对系统作以上优化之后，系统性能得到了很好的改进。尤其是采用了流水线方式之后，系统时钟有了很大的改进，具体性能比较如表 2 所示：

表 2 系统优化前后性能比较

	小波正变换		小波反变换	
	优化前	优化后	优化前	优化后
小波阶数	16 阶	16 阶	16 阶	16 阶
小波种类	db8 小波	db8 小波	db8 小波	db8 小波
处理数据位宽	16 bits	16 bits	16 bits	16 bits
数据类型	有符号数	有符号数	有符号数	有符号数
Device(DE2)	EP2C35F672C6			
LE(33,216)	3,342 (10%)	3,888 (12%)	3,355 (10%)	3,895 (12%)
Registers	874	2788	1112	3020
Memory(483,840)	1,024 (<1%)	1,024 (<1%)	1,024 (<1%)	1,024 (<1%)
Pins(475)	52(11%)	52(11%)	51(11%)	51(11%)
最高时钟频率	93.26MHz	217.72MHz	89.34MHz	217.58MHz

表 2 是采用流水线加法器优化前后结果的对比。由比较可知，经过流水线加法器的优化之后，小波正变换系统的最大时钟频率从原来的 93.26MHz，优化到了 217.72MHz。反变换类似。由此可见，系统的性能得到了很好的改善，能够满足大部分实时处理系统对于速度的要求。

2.3 系统实现

在系统整体构架下，经过各种优化之后，进行模块划分和各个子模块的设计及验证，最后进行进行顶层模块的代码编写，完成整体仿真。系统顶层模块的仿真结果如图 6、图 7 所示：

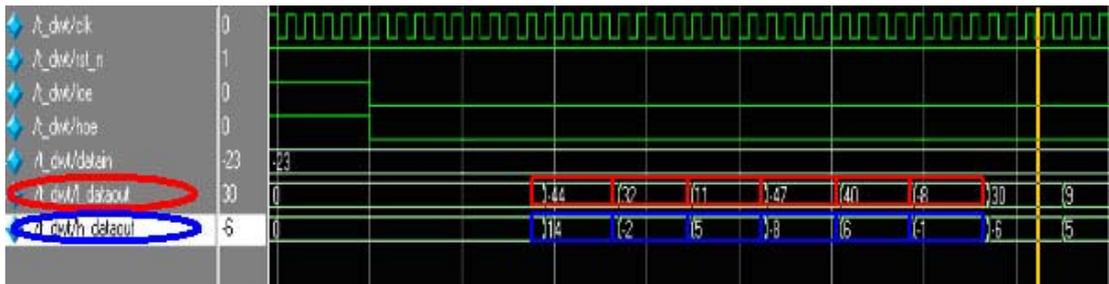


图 6 整体仿真结果截图（一）

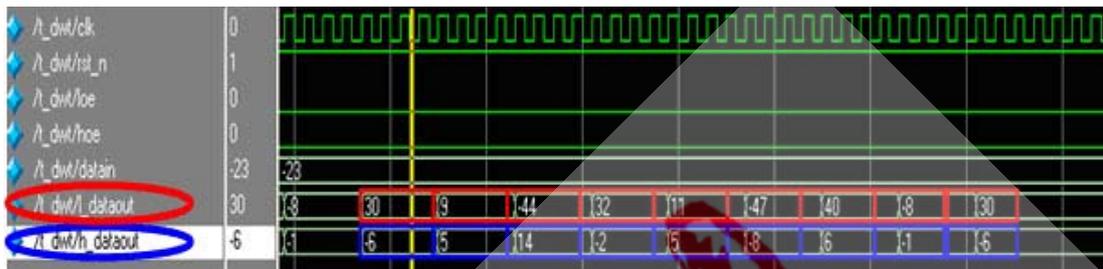


图 7 整体仿真结果截图（二）

在图 6、图 7 两部分截图中，红色表示的是小波变换的低通输出结果；蓝色表示的是高通输出结果。图中的黄线标示着（一）、（二）两部分截图的交界处。

由上可知，小波正变换的高速 FPGA 实现系统功能正确。为了测试该系统，在 QUARTUS II 上综合，选用 Altera Cyclone II 系列 FPGA EP2C35F672C6 为目标器件；设计总共占用 LEs 资源 3888 个，占所用 FPGA 资源的 12%；寄存器资源用到了 2788 个；存储器资源用了 2048bits。根据系统设计的复杂度来分析，资源的使用量合理。

本设计的最高时钟频率可以达到 217.72MHz，如图 8，可以满足大部分数字信号处理实时性的要求。

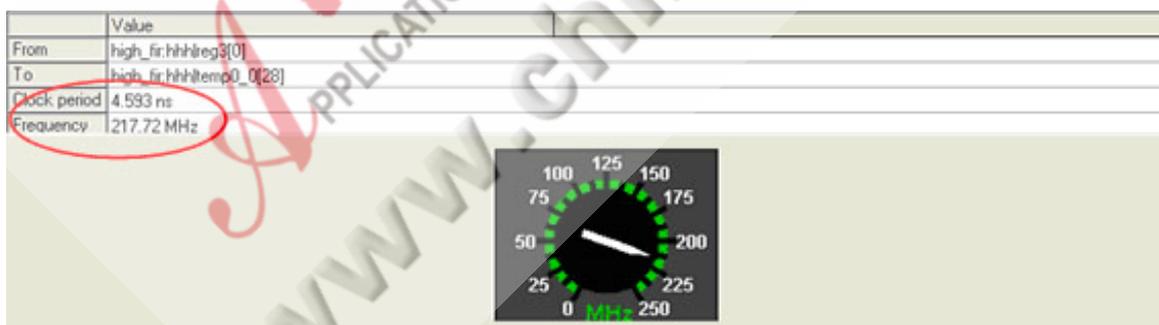


图 8 小波正变换系统最高时钟频率

2.4 系统验证

采用 MATLAB 软件实现此系统功能，然后将 Modelsim 仿真结果与 MATLAB 结果作比较，验证系统功能。

```

>> t=0:pi/3:5*pi;
>> s=32*sin(t+pi/4);
>> [cA,cD] = dwt(s,'db8','mode','ppd')

cA =

Columns 1 through 10
-42.9377  32.3702  11.4735 -46.0791  39.6284  -7.3563  30.2942  10.3196 -42.9377  32.3702

Columns 11 through 15
11.4735 -46.0791  39.6284  -7.3563  30.2942

cD =

Columns 1 through 10
14.6141  -1.8691  5.3032  -7.9180  5.9878  -0.1265  -5.1480  5.1566  14.6141  -1.8691

Columns 11 through 15
5.3032  -7.9180  5.9878  -0.1265  -5.1480

>>

```

图 9 小波正变换的 MATLAB 实现结果

如图 9，cA 表示小波变换的低频输出，与图 6、图 7 中红线标示出来的结果一一对应；cD 表示小波变换的高频输出，与图 6、图 7 中蓝线标示出来的结果对应。对比可知，两者结果基本一致。小波反变换是类似的逆过程，不再详述。综上，本系统实现了小波正、反变换的高速 FPGA 实现，功能正确，并且相比同类文献，具有很好的系统性能。

3 一维 DWT 系统的通用 FPGA 实现

在前述 db8 小波高速实现的基础上，设计一个小波种类可选、小波阶数可调的变换系统，从而满足系统设计目标的通用性要求。由于是在第 2 节基础上做的改进，系统大的框架并未做改变。因此，对 16 阶及其以下阶数的任意小波，均可以采用该构架实现小波正、反变换，并且具有较好的系统性能。

为了测试系统的可编程性和通用性，系统对应用广泛的 Db8 小波、Db4 小波和 Sym6 小波进行了测试，并且将测试结果与 MATLAB 实现结果进行了误差分析。

3.1 系统设计

基于前述思想，提出通用小波变换系统的架构，如图 10 所示：

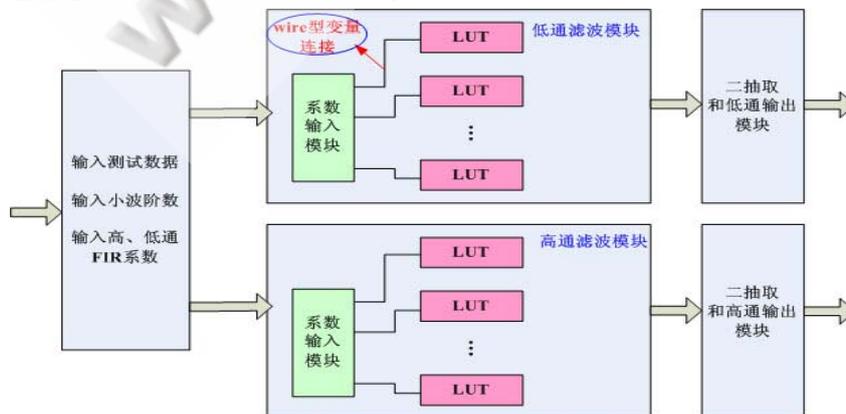


图 10 通用小波正变换系统架构

与上一节的特定小波变换系统相比，通用小波变换系统在此基础上增加了小波阶数和滤波器系数的输入模块。同时，查找表不再是已知系数的固定查找表了，而是多种加法器的组合。因此，算法效率相比特定小波系统的固定滤波系数的查找表结构下降很多。

滤波器系数的输入模块，将输入的系数分别赋给 16 个不同的系数变量；查找表中也有 16 个系数变量；将二者在滤波模块中同时例化。使得系数输入模块中的系数与查找表中的 16 个系数变量呈线型连接，这样并不耗费系统资源。这是本设计的特色，很好地节省了系统资源。

3.2 系统优化

本部分主要是针对资源进行了优化，因为在没有优化前，所做的系统的资源占用达到 80%左右，相对于该系统所实现的功能而言，这是不合理的。因此，对系统进行了进一步的优化，主要是查找表单元的优化。

在第 2 节所介绍的特定系统中，由于小波特定，则它相应的查找表结构就是一个简单的 RAM，通过地址选择输出正确的结果即可；在本节介绍的系统中，由于具有通用性，即小波的种类和阶数均是不确定的。则相对于原来的查找表单元来说，它的输出结果不再是一个确定的值，而是要在一系列加法器中选择一个作为结果输出；这样对于系统资源的耗费是巨大的。因此，本章对于系统做了如下的优化，使得系统资源减少了近一半，系统时钟也得到了提升。

1. 由于对于第 2 节的查找表单元来说， h_0, h_1, \dots, h_{15} 都是查找表单元的内部参数，有确定的值；而对于本节的查找表单元来说，它的端口信号如下图 11 所示：

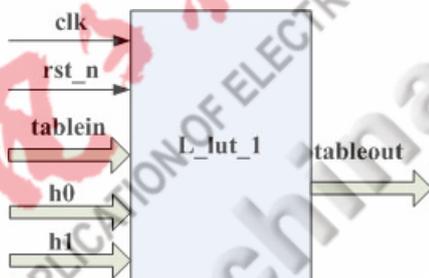


图 11 通用系统的查找表结构端口信号

也就是说， h_0, h_1, \dots, h_{15} 是不确定的，它们是该查找表单元的输入。则该查找表模块的功能为：通过 $tablein$ 进行控制选择输出的形式，可能的输出形式就是 h_0, h_1, \dots, h_{15} 的各种不同组合。由于 h_0, h_1, \dots, h_{15} 均是未知的系统输入，则该模块其实就是：通过一个选择信号，选择输出哪一种加法器结构的结果。因此，当然输入变量的种类越少，得到的加法器结构就越少，就越节省系统资源。

因此，本节将前一节的 4 个四输入查找表结构优化成 8 个两输入加法器选择结构。经优化，系统的资源减少了一半有余。

2. 另外，第 2 节的查找表结构是采用组合结构实现的。本节的查找表结构不再是一个简单的组合逻辑问题，而是一系列的加法器选择结构，因此改用时序结构实现。这项改进使得系统时钟得到了较大的改进。

综合以上两种优化策略，本节系统性能得到了很大的改进。具体见表 4 所示：

表 4 系统优化前后性能比较

	小波正变换		小波反变换	
	优化前	优化后	优化前	优化后
小波阶数	可调 (≤ 16 阶)			
小波种类	可选	可选	可选	可选
处理数据位宽	16 bits	16 bits	16 bits	16 bits
数据类型	有符号数	有符号数	有符号数	有符号数
Device(DE2)	EP2C35F672C6			
LE(33,216)	27,521 (83%)	10,719 (32%)	27,718 (83%)	10,921 (33%)
Registers	3623	9907	3855	10139
Memory(483,840)	2,048 (<1%)	2,048 (<1%)	1,024 (<1%)	1,024 (<1%)
Pins(475)	89(19%)	89(19%)	88(19%)	88(19%)
最高时钟频率	83.37MHz	114.10MHz	83.77MHz	152.09MHz

由表 4 可知，系统的优化策略很好的改善了系统的资源使用情况以及系统时钟。资源减少了一半有余，时钟也提高了很多，使得本系统的最大时钟频率可以满足数字信号实时处理的需要，这也很好的达到了系统设计所要求的实时性目标。相比系统优化前的大资源、低时钟频率，本设计优化后具有很好的性能指标。

3.3 系统实现

由于本系统的设计目标定位在通用性、可编程性及实时性。因此，下面给出三种不同种类、不同阶数的小波应用于本系统的仿真情况，以测试本设计是否具有通用性和可编程性。

1. Db8 小波（16 阶）

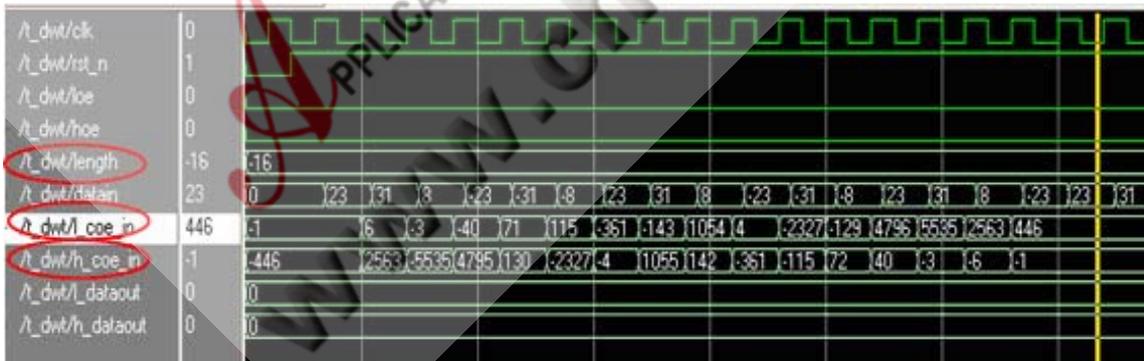


图 12 高、低通滤波器阶数、系数输入部分截图

如图 12 所示，length 表示输入小波的阶数，也就是滤波器的长度；datain 为输入的系统测试数据；l_coe_in, h_coe_in 分别是低通和高通滤波器的 FIR 系数，它们是将通过 MATLAB 中的 wfilters() 函数得到的数据放大 8192 倍，并且四舍五入取整得到的。

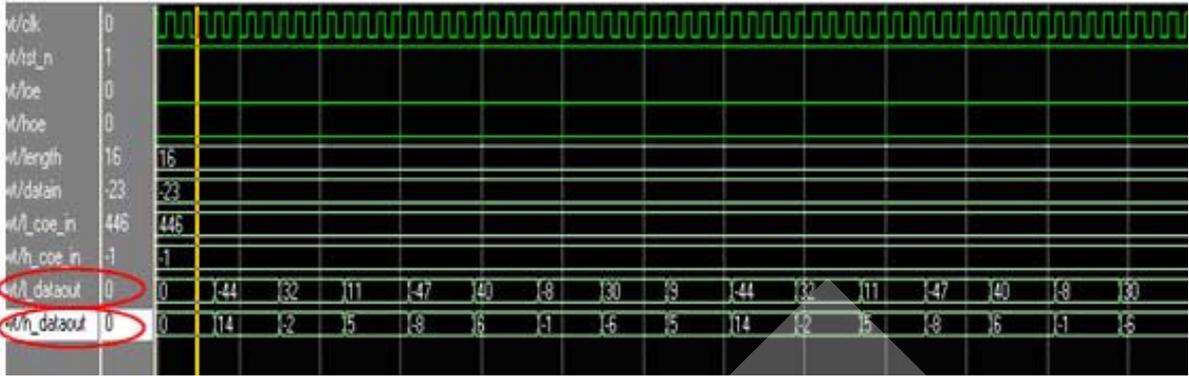


图 13 Db8 小波正变换

在如图 12 输入情况下，仿真结果如图 13。通过仿真，通用小波正变换系统的高、低通滤波结果输出分别如上图中的 h_dataout 和 l_dataout。

2. Db4 小波（8 阶）

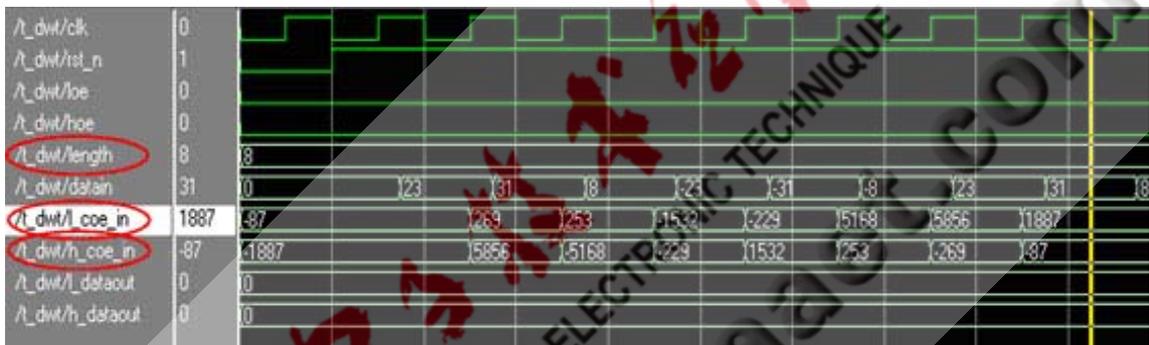


图 14 高低通滤波器阶数、系数输入部分截图

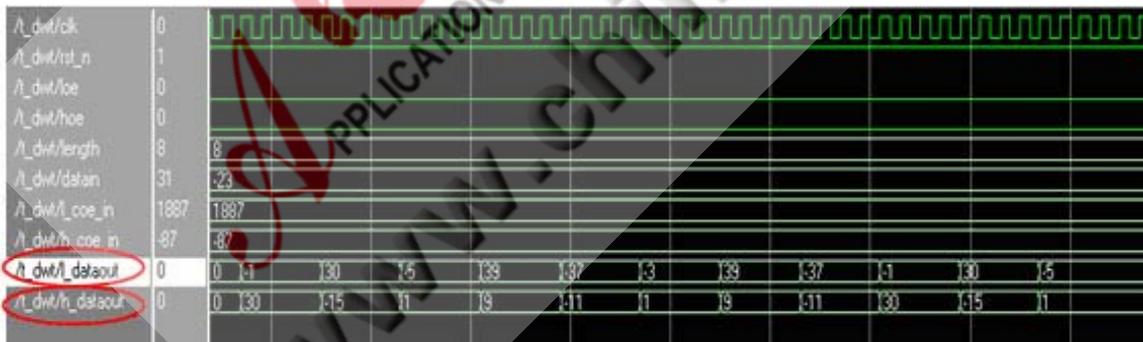


图 15 Db4 小波正变换仿真结果

在如图 14 输入情况下，仿真结果如图 15：Db4 小波正变换系统的高、低通滤波结果输出分别如图 15 中的 h_dataout 和 l_dataout。与 MATLAB 实现结果基本一致。

3. Sym6 小波（12 阶）

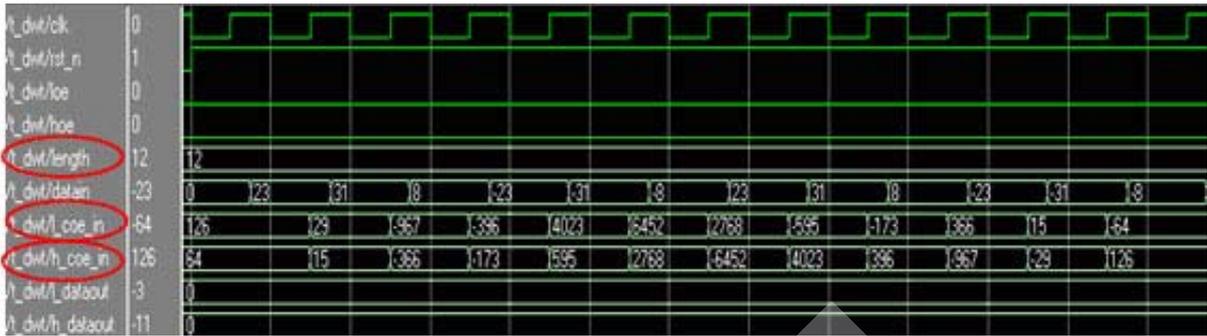


图 16 高低通滤波器阶数、系数输入部分截图

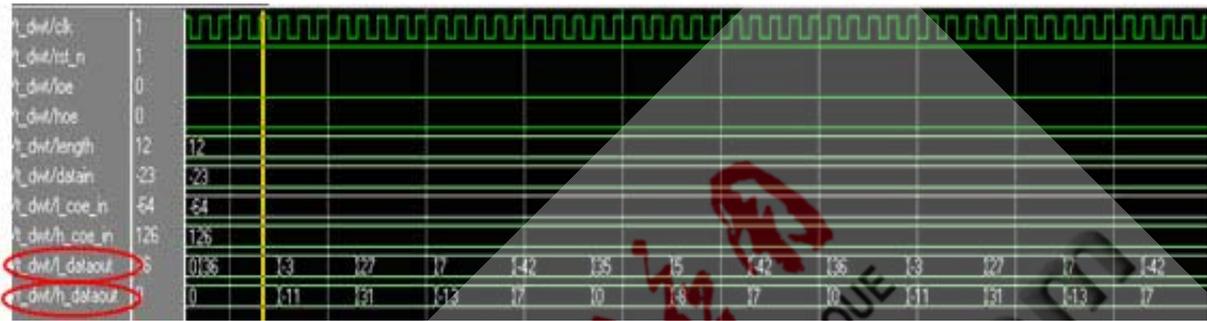


图 17 Sym6 小波正变换仿真结果

在如图 16 输入下，仿真结果如图 17：Sym6 小波正变换系统的高、低通滤波结果输出分别如上图中的 h_dataout 和 l_dataout。与 MATLAB 实现结果基本一致。

综上，本设计很好地实现了 16 阶以下各种小波的正变换，功能正确，且具有较好的通用性。下面对该系统进行综合：通用小波正变换系统 LEs 占用 DE2 资源的 32%，寄存器资源占用了 9907 个，存储器资源占用不到 1%。相对于本系统的设计复杂度及其实现的功能而言，本系统的资源占用情况比较合理。

在实现如上较为复杂的功能之后，该系统的最高时钟频率可以达到 114.10MHz，如图 18。反变换的最大时钟频率也达到了 152.09MHz，能够满足一般系统的实时性要求。考虑到本系统的功能复杂性，最大时钟频率仍保持在 100MHz 以上，可以满足系统设计的实时性要求。

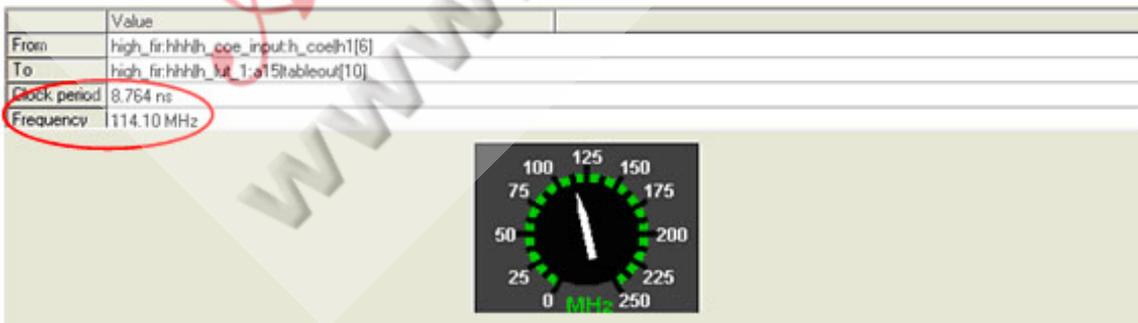


图 18 通用小波正变换系统最高时钟频率

3.4 系统验证

采用 MATLAB 软件实现此系统功能，然后将 Modelsim 仿真结果与 MATLAB 结果作比较分析。

1. Db8 小波已在第 2 节做了介绍，不再说明。

2. Db4 小波的 MATLAB 实现结果如下:

```
>> t=0:pi/3:5*pi;
>> s=32*sin(t+pi/4);
>> [cA,cD] = dwt(s,'db4','mode','ppd')

cA =

Columns 1 through 10
-1.4616  30.4001  -4.0324  39.1090  -36.3023  -2.8067  39.1090  -36.3023  -1.4616  30.4001
Column 11
-4.0324

cD =

Columns 1 through 10
30.1766  -14.2330  0.9912  9.9114  -10.8462  0.9348  9.9114  -10.8462  30.1766  -14.2330
Column 11
0.9912

>> |
```

图 19 Db4 小波的 MATLAB 实现结果

图 19 中，cA 表示 Db4 小波变换的低频输出，与图 15 中 l_dataout 的输出结果一一对应；cD 表示 Db4 小波变换的高频输出，与图 15 中 h_dataout 的输出结果对应。对比可知，两者结果基本一致。因此，本系统实现了 Db4 小波正变换，功能正确。

3. Sym6 小波的 MATLAB 实现结果如下:

```
>> t=0:pi/3:5*pi;
>> s=32*sin(t+pi/4);
>> [cA,cD] = dwt(s,'sym6','mode','ppd')

cA =

Columns 1 through 10
35.9969  -2.1587  27.3743  7.9645  -41.4642  35.2063  5.9268  -41.1331  35.9969  -2.1587
Columns 11 through 13
27.3743  7.9645  -41.4642

cD =

Columns 1 through 10
0.5257  -10.6465  30.8756  -12.6064  7.8517  0.1253  -7.3232  7.1979  0.5257  -10.6465
Columns 11 through 13
30.8756  -12.6064  7.8517

>>
```

图 20 Sym6 小波的 MATLAB 实现结果

如图 20, cA 表示 Sym6 小波变换的低频输出, 与图 17 中 l_dataout 的输出结果一一对应; cD 表示 Sym6 小波变换的高频输出, 与图 17 中 h_dataout 的输出结果对应。对比可知, 两者结果基本一致。因此, 本系统实现了 Sym6 小波正变换, 功能正确。

4 设计结果分析

4.1 设计平台

本设计采用 Altera 公司的 DE2 板上的 Cyclone II FPGA 芯片^[10], 型号为 EP2C35F672C6。在 Altera Quartus II 中完成代码设计和系统综合, 采用 Modelsim 进行系统功能仿真, 最后采用 MATLAB 计算结果, 验证系统的正确性。

4.2 设计总结及误差分析

本文主要设计了两个系统, 一个是一维 DWT 系统的高速 FPGA 实现, 主要针对应用广泛的 16 阶 db8 小波设计了一个高速的小波正、反变换系统, 该系统具有很好的实时性; 一个是一维 DWT 系统的通用 FPGA 实现, 主要设计了一个小波种类可选、小波阶数可调 (16 阶及其以下各阶) 的通用小波正、反变换系统, 该系统具有通用性、可编程性和实时性。

同时, 本设计做了大量的优化。通过优化, 使得本设计性能优于其他同等设计, 尤其是本设计的系统时钟, 相比目前所查文献, 具有明显的优势。表 5 对本设计进行总结如下:

表 5 本文设计内容总结

	特定小波的高速实现		阶数可调小波的实现	
	小波正变换	小波反变换	小波正变换	小波反变换
小波阶数	16 阶	16 阶	可调 (≤ 16 阶)	可调 (≤ 16 阶)
小波种类	db8 小波	db8 小波	可选	可选
处理数据位宽	16 bits	16 bits	16 bits	16 bits
数据类型	有符号数	有符号数	有符号数	有符号数
Device(DE2)	EP2C35F672C6			
LE(33,216)	3,888 (12%)	3,895 (12%)	10,719 (32%)	10,921 (33%)
Registers	2788	3020	9907	10139
Memory(483,840)	1,024 (<1%)	1,024 (<1%)	2,048 (<1%)	1,024 (<1%)
Pins(475)	52(11%)	51(11%)	89(19%)	88(19%)
最高时钟频率	217.72MHz	217.58MHz	114.10MHz	152.09MHz

根据上表性能分析可知: 高速小波正、反变换系统的最高时钟频率分别为 217.72MHz、217.58 MHz, 均大于 200 MHz, 达到了系统实时性的设计目标; 通用小波正、反变换系统的分别为 114.10 MHz、152.09 MHz, 均大于 100 MHz, 达到了实时性的目标, 而且该系统可编程, 具有一定的通用性。

由于浮点数的运算利用 FPGA 很难实现，所以本系统将所有浮点数经四舍五入，化为整数参与 FPGA 运算。因此，本系统必然引入误差。下面以特定小波反变换系统为例，分析系统误差。如表 6 所示：

表 6 特定小波反变换系统误差比较

MATLAB 实现结果	Modelsim 仿真结果	相对误差
113.1371	112	1.0050638%
154.5481	153	1.0016946%
41.411	42	1.4223274%
-113.1371	-114	0.7627030%
-154.5481	-156	0.9394486%
-41.411	-43	3.8371447%
113.1371	112	1.0050638%
154.5481	154	0.3546469%
41.411	40	3.4073072%
-113.1371	-114	0.7627030%
-154.5481	-155	0.2924009%
-41.411	-42	1.4223274%
113.1371	112	1.0050638%
154.5481	153	1.0016946%
41.411	40	3.4073072%
-113.1371	-114	0.7627030%

如表 6 所示，小波反变换系统的最大相对误差为 3.8371447%，在误差允许的条件下，可以满足一般应用系统要求。其中，误差的主要来源有以下三个方面：1. 由于输入数据的截断误差；2. 由于滤波器系数的截断误差；3. 由于采用右移代替除法导致的误差。

4.3 性能对比

为了更好的测试本设计性能，选取两篇与本设计近似的典型工作进行了对比分析。

与文献[1]性能相比较的结果如表 7 所示。本系统设计为 16 阶，为与参考文献具有可比性，选择与文献相同的器件对第 2 节所设计的系统进行综合。由比较可知，本设计的待处理数据位宽是文献的 2 倍，阶数是参考文献的 4 倍。而这两个因素对于系统性能的影响至关重要。尽管如此，本系统的时钟频率比参考文献仍然高出很多。由此可知，本设计得到了很好的优化，相比同类文献具有明显优势。

另外，由于本设计相比文献[1]复杂得多，实现小波反变换需要 4383 个 LEs,而该器件总共只有 4160 个 LEs。因此，反变换综合不能完成。但仅从正变换已可以看出本设计的优势。小波反变换是一个逆过程，结构类似，因此理论上它的性能也类似于正变换的综合结果。

表 7 与文献[1]性能比较

	文献[1] (中文)		本文实现	
	小波正变换	小波反变换	小波正变换	小波反变换
小波阶数	4 阶	4 阶	16 阶	16 阶
小波种类	db2 小波	db2 小波	db8 小波	db8 小波
处理数据位宽	8bits	8bits	16bits	16bits
数据类型	有符号数	有符号数	有符号数	有符号数
device	EP20K100EBC356 1X			
LEs(4160)	1037	861	4152(100%)	超出总数
Memory(53,248)	1024	0	2048(4%)	
Pins(246)	100	52	52(21%)	
最高时钟频率	96.35MHz	110.44MHz	121.04MHz	

与文献[3]性能相比较的结果如表 8 所示。文献[3]是来自 2007 年 IEEE 上的一篇 ICASSP 会议文章。由比较可知，两种设计的阶数相同，均为 db8 小波。处理数据的位宽不同，本设计的处理位宽为 16bits，文献为 8bits。采用同一型号的 FPGA（Stratix II 系列）进行综合时，文献实现的小波正变换的最高时钟频率为 149.3MHz，而本设计的最高时钟频率达到了 389.71MHz，相比具有较好的实时性。当然，本系统的资源消耗比该文献要多，这是以面积换速度的必然代价。考虑到本设计最终的资源消耗只占到该 FPGA 器件的 21%，相对于系统功能的复杂度来说，这个资源消耗是合理的。而考虑小波变换作为一种数字信号处理的工具，其实时性是应该优先考虑的，因此本设计的目标定位在实时系统上。由此可知，本设计达到了较好的性能，完成了系统实时性的设计目标。

表 8 与文献[3]性能比较

	文献[3] (英文)	本设计
主要作者	Patrick Longa, University of Ottawa	
文献来源	IEEE, ICASSP 2007	
小波种类	db8	db8
小波阶数	16 阶	16 阶
处理数据位宽	8bits	16bits
device	Stratix II, EP2S15F484C3	
LEs	614	2619 (21%)
Registers	未知	2806
Memory	未知	2048
Pins	未知	52
最高时钟频率	149.3MHz	389.71MHz

综上,通过与国内外近几年的两篇典型文献的性能比较可知,本设计在最高时钟频率方面具有很好的优越性,同时资源消耗也比较合理,完成了系统的设计目标。

5 结束语

本文考虑到离散小波变换在数字信号处理领域实时性和通用性的要求,分别设计了一维 DWT 和 IDWT 的高速和通用 FPGA 实现系统。它们分别具有以下特点:

1. 由于采用经改进后的 DA 算法以及流水线、IP 核等技术对设计进行了大量优化,针对 db8 小波设计的高速 FPGA 正、反变换系统具有较好的系统性能,正、反变换系统的最高时钟频率分别为 217.72MHz、217.58 MHz (CycloneII EP2C35F672C6),相比同类文献具有明显优势。能够应用到实时性要求严格的系统,可进行图像或信号的压缩、去噪、检测突变信号等等。
2. 考虑到目前的小波变换系统过分依赖于具体的小波函数,本文设计了一个具有通用性的 FPGA 实现系统,正、反变换的最高时钟频率分别为 114.10 MHz、152.09 MHz (CycloneII EP2C35F672C6),均大于 100 MHz,达到了实时性的目标,而且该系统可编程,具有一定的通用性。本设计可以扩展到多种小波的正、反变换,做成一个通用的 IP 核,根据输入小波的不同,对应输出相应的小波正、反变换结果。
3. 在系统验证方面,将 MATLAB 小波变换工具箱中的相关函数进行小波正、反变换的结果与实际仿真结果比较,二者误差可以满足一般应用系统要求,并对误差来源作了详细分析。
4. 由于本设计具有模块化的特征,可以很方便地扩展到多级、多维的小波变换系统。

参考文献

- [1] 龚敏元.小波变换研究及其 FPGA 实现[D].广东:华南理工大学, 2004.
- [2] 王胜坤.JPEG 2000 中小波变换的 FPGA 实现[D].山西:太原理工大学, 2007.
- [3] Patrick Longa, Ali Miri, Miodrag Bolic. A Flexible Design of Filterbank Architectures for Discrete Wavelet Transforms [J].IEEE,2007(1):1441-1444.
- [4] S.V.Silva, S.Bampi.Area and Throughput Trade-Offs in the Design of Pipelined Discrete Wavelet Transform Architectures[J].in Proc.Design,Automation and Test in Europe Conference Exhibition(DATA'05),2005.
- [5] 顾文军.小波变换的硬件架构研究及其芯片实现[D].上海: 上海大学, 2004.
- [6] S.Masud, J.McCanny. Reusable Silicon IP Cores for Discrete Wavelet Transform Applications[J].IEEE Transactions on Circuits and Systems I:Fundamental Theory and Applications.2004,51(6):1114-1112.
- [7] 张雷, 陈向东, 李辉.小波变换及其芯片研究[J]. 电路与应用, 2006 (4) : 21-22.
- [8] Mallat S.G., A Theory for Multiresolution Signal Decomposition: The Wavelet Representation[J].IEEE Trans on PAMI,1989,11(7):674-693.
- [9] 于大卫, 董辉.基于 FPGA 的高速 FIR 数字滤波器设计[J].空间电子技术, 2008 (2) : 47-51.
- [10] Altera Corporation. Cyclone II Device Handbook. www.altera.com.