

小波变换的电路集成

张永梅 徐葭生

(清华大学微电子学研究所 北京 100084)

摘要 小波变换是信号分析和处理中的重要方法. 本文设计了一种实现小波变换的集成电路. 它充分利用了算法本身的特点, 采用电路复用的方法, 节省了电路单元的需要量, 并且, 采用并行和流水线的电路结构, 使得工作速度较快. 平均每个时钟周期内输出一个小波变换结果. 时钟的频率由加法器延时决定. 电路得到了 VHDL (硬件描述语言) 的模拟和验证.

EEACC: 2570, 1265B, 1270F

1 引言

小波变换方法自从 1984 年被提出以来^[1], 在数学、物理和电子工程等领域受到广泛的重视和应用. 它的出现被视为傅立叶发展史上的突破性进展. 虽然, 小波变换与傅立叶变换都是在正交基函数上的展开, 但小波基函数更具有时间和频率的局域性. 因此它在信号分析和处理方面显示出独特的优势.

小波变换方法用集成芯片实现, 可以大幅度地提高工作速度, 这在实际应用中具有重要的意义. 对此方面研究国外已有报道^[2,3], 但涉及具体技术方案的很少. 本文提出一种小波变换电路结构, 并且阐述了它的工作原理. 文中还给出了它的 VHDL 模拟和实验结果.

2 小波变换原理

数字信号处理中使用的小波变换采取离散形式, 其一维变换式为:

$$\begin{cases} yh(k) = \sum_n h(n-2k) * x(n), \\ yg(k) = \sum_n g(n-2k) * x(n) \end{cases} \quad n, k \in z \quad (1)$$

其中 $x(n)$ 为输入信号序列; $yh(k)$ 和 $yg(k)$ 是一对变换结果; $h(n)$ 和 $g(n)$ 为小波基函数所构成的低通和高通滤波器, 两者的关系式为:

$$g(n) = (-1)^n h(1-n) \quad n \in z \quad (2)$$

小波变换后再经过反变换可以完全恢复原信号值,公式如下:

$$x(n) = \sum_k h(n-2k) * y_h(k) + \sum_n g(n-2k) * y_g(k) \quad (3)$$

常用的 Daubechies 类型^[4]的小波滤波器系数,见表 1 所示.

表 1 高通和低通小波滤波器系数

n	0	1	2	3
$h(n)$	0.482962913145	0.836516303738	0.224143868042	-0.129409522551
$g(n)$	0.129409522551	0.224143868042	-0.836516303738	0.482962913145

其中低通滤波器 $h(n)$ 的总和值为 $\sqrt{2}$, 高通滤波器 $g(n)$ 的总和值为零.

3 电路结构

从公式(1)可看出,小波变换的计算相当于输入信号的滤波过程.根据小波滤波器的特点,我们对电路结构进行了特殊设计.

首先,为了避免计算结果的溢出,把 $h(n)$ 序列归一化; $h'(n) = h(n) / \sqrt{2} = \{0.34150635, 0.59150635, 0.15849365, -0.09150635\}$. $g(n)$ 也做相同的处理; $g'(n) = g(n) / \sqrt{2}$. 这样,输入信号经过 $h'(n)$ 和 $g'(n)$ 的滤波后必定不会越过原值.因此可用相同的二进制位数表示输入和输出信号值的整数部分.

小波滤波系数的小数值转化成二进制数时,既要保证较高的精度,又要不占用过多的二进制位数.经过反复计算实验,我们选用 9 位表示小数.见表 2 所示.

在表 2 和下文中,我们均用 $h(n)$ 表示 $h'(n)$ 的绝对值.表 2 中的二进制与十进制数完全等价.它们与原 $h'(n)$ 序列的均方误差为 8.4×10^{-8} .

表 2 滤波器 $h(n)$ 的二进制转化

$h(n)$	二 进 制	十 进 制
$h(0)$	0.010101111	0.341796875
$h(1)$	0.100101111	0.591796875
$h(2)$	0.001010001	0.158203125
$h(3)$	0.000101111	0.091796875

依据公式(1)和表 2, y_h 的表达式可展开为:

$$\begin{aligned} y_h(0) &= x(0) * h(0) + x(1) * h(1) + x(2) * h(2) - x(3) * h(3) \\ &= x(0) * (2^{-2} + 2^{-4} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9}) + x(1) * (2^{-1} + 2^{-4} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9}) + x(2) * (2^{-3} + 2^{-5} + 2^{-9}) - x(3) * (2^{-4} + 2^{-6} + 2^{-7} + 2^{-8} 2^{-9}) \end{aligned}$$

$$y_h(1) = x(2) * h(0) + x(3) * h(1) + x(4) * h(2) - x(5) * h(3)$$

⋮

y_g 的表达式为:

$$\begin{aligned} y_g(0) &= x(0) * g(0) + x(1) * g(1) - x(2) * g(2) + x(3) * g(3) \\ &= x(0) * h(3) + x(1) * h(2) - x(2) * h(1) + x(3) * h(0) \end{aligned}$$

$$yg(1) = x(2) * h(3) + x(3) * h(2) - x(4) * h(1) + x(5) * h(0)$$

·
·
·

因为 x 乘 2^{-1} 可以用电路中的右移 1 位完成, 所以 yh 和 yg 的计算可不用乘法而只用移位和累加实现. 注意到 $h(0)$ 和 $h(3)$ 的运算结构很相似, 因此采用电路复用的方法, 由同一组电路单元来实现, 以节省单元门数, 减小芯片面积. 为加快运算速度, 电路采用流水线方式工作, 即: 把移位和累加运算分解为多级运算, 输入信号在时钟的控制下依次进行各级的运算. 例如, 当输入 $x(i)$ 完成移位后就“流入”随后的加法运算级, 此时, 下一个输入 $x(i+1)$ 开始移位而不必等待 $x(i)$ 完成全部的移位和累加运算. 这种工作方式不仅可以提高工作速度, 而且使电路的利用率很高, 即: 在每个时钟节拍内, 各级的电路单元都在工作.

我们把 $x(i)$ 与 $h(1)$ 和 $h(2)$ 相乘和累加运算用另一组移位器和加法器电路实现. 它与计算 $h(0)$ 和 $h(3)$ 的电路是并行工作的. 见图 1 所示.

假定输入的数字信号范围在 0—255 之间, 因此用 8 位无符号二进制数表示. 因为电路中具有 9 位小数的运算, 并且减法运算会造成负数结果, 所以输入信号 x 经过输入缓冲器被扩展为 18 位. 其中最高位为符号位, 其次的高 8 位表示整数, 低 9 位表示小数. 扩展的位补零.

电路中输入控制单元的作用是使输入信号按照正确的顺序分配给到两组运算电路的输入端 XI 和 XJ.

设输入信号为 $x(n) = \{x_0, x_1, x_2, x_3, \dots\}$, 在时钟下降沿的作用下被依次输入. 第一个时钟周期时, 信号 x_0 输入. 如图 2 所示, 在第 3 个时钟周期的下降沿后, x_0 到 XI 端. 此时, 信号 x_1 也经过两个触发器的延时刚好到达 XJ 端. x_0 和 x_1 分别经过缓冲器后, 在第 4 个时钟周期的上升沿时, 同时开始移位操作. 电路图 1 的移位寄存器中, 数字表示右移位数. 在时钟的作用下, 移位的结果依次进行下面的各级加法运算.

当第 4 个时钟下降沿到来后, 输入信号 x_3 通过第一个 D 触发器, 并且在二选一电路的控制下输出到 XI 端. 与此同时, x_2 信号到达 XJ 端. 当第 5 个时钟下降沿时, 二选一电路又把第三个 D 触发器的输出给到 XI 端. 此时, XI 和 XJ 端上分别得到 x_2 和 x_3 信号. 依此类推, 在每个时钟的下降沿到来时, XI 端和 XJ 端都出现一对新的输入信号: (x_0, x_1) 、 (x_3, x_2) 、 (x_2, x_3) 、 (x_5, x_4) 、 $\dots$.

第一对输入信号 (x_0, x_1) 在第 5 个时钟的下降沿后, 到达信号线①和②(见图 1 所示), 并得到计算结果 $x_0 * (2^{-4} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9})$, 即 $x_0 * g(0)$, $(= x_0 * h(3))$ 和 $x_1 * (2^{-3} + 2^{-5} + 2^{-9})$, 即 $x_1 * g(1)$, $(= x_1 * h(2))$.

在第 6 个时钟上升沿后, 信号线③和④上分别得到 $x_0 * h(0)$ 和 $x_1 * h(1)$ 的运算结果. 与此同时, 信号线⑤上得到 $(x_0 * g(0) + x_1 * g(1))$, 此结果通过多路分配器连到 YG 输出加法器前的寄存器输入端. 当第 6 个时钟下降沿到来后, 信号线⑥上得到 $(x_0 * h(0) + x_1 * h(1))$, 并通过多路分配器连到 YH 输出加法器前的寄存器输入端上. 在流水线方式工作下, 此时第二对输入信号 x_3 和 x_2 已经在信号线①和②上分别得到 $x_3 * h(3)$ 和 $x_2 * h(2)$ 的运算结果.

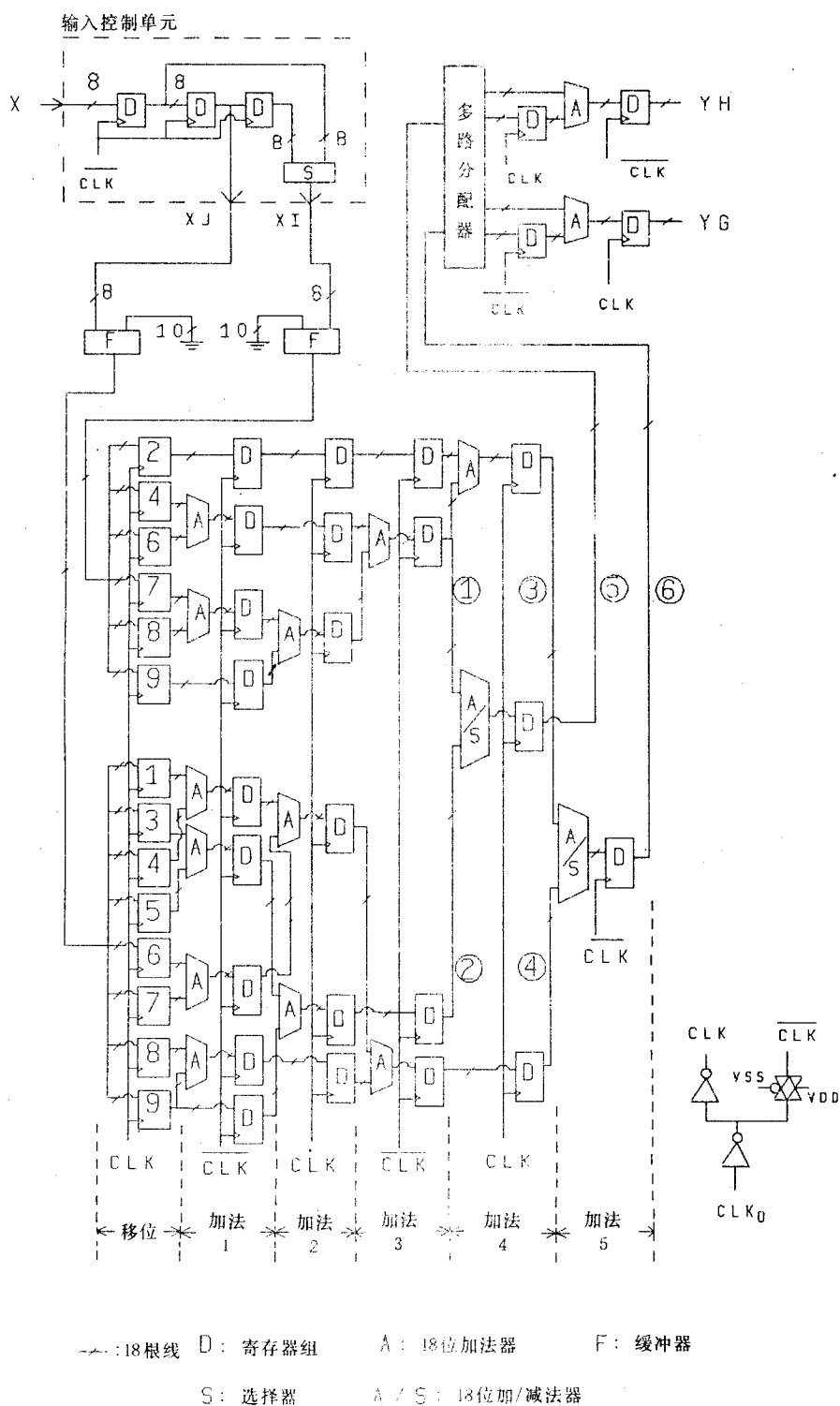


图 1 小波变换电路结构图

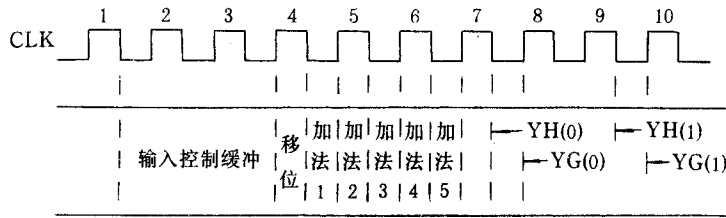


图2 小波变换电路时序示意图

当第7个时钟上升沿到来时,信号线⑤上得到 $(x_2 * h(2) - x_3 * h(3))$,此结果通过多路分配器给到 YH 加法器的输入端.此刻,YH 加法器前的寄存器也把输出 $(x_0 * h(0) + x_1 * h(1))$ 给到加法器的另一输入端,并开始加法运算.当第7个时钟下降后,输出端 YH 得到第一个计算结果 $YH(o)$,即 $x_0 * h(0) + x_1 * h(1) + x_2 * h(2) - x_3 * h(3)$.与此同时,信号线⑥上得到 $(x_3 * g(3) - x_2 * g(2))$,并给到 YG 电路加法器的一个输入端上,寄存器也把 $(x_0 * g(0) + x_1 * g(1))$ 给到加法器的另一输入端,开始加法运算.

当第8个时钟上升沿时,输出端 YG 得到第一个计算结果 $YG(o)$,即 $x_0 * g(0) + x_1 * g(1) - x_2 * g(2) + x_3 * g(3)$.可见,第一个输出 $YH(o)$ 和 $YG(o)$ 比输入分别延迟了七个半和八个时钟周期.当流水线被充满时,即输出 $YH(o)$ 和 $YG(o)$ 之后,每隔两个时钟周期可输出一对计算结果 yh 和 yg .平均每个小波变换结果只花费一个时钟周期.时钟的频率由加法器的延迟时间决定,因此具有较快的小波变换运算速度.

电路中共使用了14个十八位加法器,2个十八位加/减法器,27个十八位寄存器,3个八位寄存器和14个移位寄存器.电路规模约6400门.电路的工作速度取决于十八位加法器的延迟时间.VLSI Technology 公司提供的标准单元库中十八位加法器延迟时间为10ns.因此电路可工作在约50MHz.

4 模拟和验证

图1所示电路的设计是用VHDL语言完成的.其逻辑功能正确性通过View Logic公司的VHDL模拟器来模拟和验证.

输入的信号 $x(n)$ 是一组随机整数值,范围在0—255,因此可用8位无符号二进制数表示.模拟结果见图3所示.其中的数据均用十六进制数表示.“xxxx”为不定态.

输入信号的等价十进制数为 $x(n) = \{0, 2, 4, 136, 144, 162, 68, 8, 16, 4, 2, 164, 34, 161, 74, \dots\}$.时钟信号的周期取20ns.输出端 YH 和 YG 的信号分别延迟到第7个时钟周期的下降沿和第8个时钟周期的上升沿时得到,随后每间隔两个时钟周期输出一对新结果.这与前文理论分析的波形结果(见图2)完全一致.

为验证模拟输出的 YH 和 YG 数值正确性,我们用C语言编写了小波变换运算程序.小波滤波系数采用表2中的十进制数值.输入信号 $x(n)$ 与VHDL模拟采用的数值相同.软件运行的结果见表3所示.

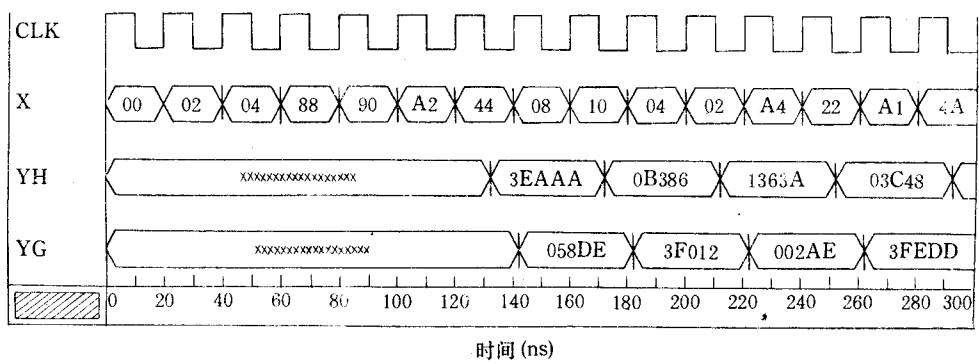


图 3 小波变换电路的模拟结果

表 3 小波变换的软件计算结果

序 号	YH		YG	
	十进制	二进制	十进制	二进制
0	-10.667969	3EAAA	44.433594	058DE
1	89.761719	0B386	-7.964844	3F012
2	155.113281	1363A	1.339844	002AE
3	30.140625	03C48	-0.593750	3FED0

表 3 中的十进制数为计算得到的直接结果,十六进制数为十进制数转化为等价二进制数并把小数点右移 9 位后只取整数部分得到的.显然,这与图 3 中 VHDL 模拟结果完全相同.

采用表 2 中的滤波系数,把小波变换结果 y_h 和 y_g 进行反变换得到的 $x'(n)$,可以看出小波变换电路的均方误差为:

$$r = \frac{\sum_{n=1}^N [x'(n) - x(n)]^2}{N} = 0.000030, N = 512$$

式中 N 为信号个数.

为进一步验证图 1 所示电路的 VHDL 描述的准确性,我们用 Compass 软件做了 VHDL 综合工作,即:写入 VHDL 语言,软件自动综合输出电路图.综合结果表明与图 1 所示电路结构完全一致.

5 总结

本文提出的小波变换电路结构,充分利用了其算法特点,使两个小波滤波运算用同一组电路单元完成.因此节省了单元总数,并且使电路具有很高的利用率,即在每个时钟周期内,没有“闲置”的电路单元.乘法和累加的运算采用移位和加法的流水线和并行的电路结构,因此运算速度较快.当流水线充满后,平均每个时钟周期得到一个小波变换结果.时钟的工作

频率取决于加法器的延时. VHDL 模拟和综合的结果验证了电路功能和正确性.

参 考 文 献

- [1] A. Grossman and J. Morlet, SIAM J. Math. Anal., 1981, **15**:723.
- [2] A. S. Lewis and G. Knowles, Electronics Letters., 1991, **27**(2):171.
- [3] H. S. Malvar, Electronics Letters, 1992, **28**(15):1393.
- [4] I. Daubechies, Communications on Pure and Applied Math., 1988, **XLI**:909.

IC Architecture for Wavelet Transform

Zhang Yongmei and Xu Jiasheng

(Institute of Microelectronics, Tsinghua University, Beijing 100084)

Received 17 March 1994, revised manuscript received 25 May 1994)

Abstract Wavelet transform (WT) is a highly effective tool for signal analysis and signal processing. A specific IC architecture for WT is designed, which makes good use of the characteristics of the WT. The circuit uses multiplex structure to reduce the number of gates, as well as the parallel and the pipeline structure to make the circuit work at a high speed. On the average, one output of the WT can be got per clock cycle. The frequency of the clock is limited by the delay of the adder used. The VHDL simulation which verifies our IC architecture is presented.

EEACC: 2570, 1265B, 1270F