时延估计算法的FPGA实现

袁慧琴，尚俊娜，赵知劲
（杭州电子科技大学 通信学院，浙江杭州 310018）

摘　要： 时延估计是雷达、声纳等领域经常遇到的一个问题，提出了利用相关计算法实现时延估计，并通过互谱插值提高估计精度。结合FPGA器件特性，运用VHDL语言编程，实现了整个相关算法。利用Quartus Ⅱ和Matlab软件进行了联合仿真，结果证明了设计的正确性。

关键词： 时延估计　内插　FPGA

时延估计是国际信号处理领域十分活跃的研究课题，其技术随着目标定位需求的提高而得到不断的发展。时延估计的基本方法是相关计算法，其主要思想是：若一路信号是另一路信号的移位，则利用互相关技术比较两路信号的相似性，相关函数的最大值对应相似性最大，其最大的位置就是时延估计值。数学描述如下：设信号为 $s(t)$，两接收机输入噪声分别为 n_1 和 n_2，它们互不相关，则两接收机输出为：

$$x_1(t)=A_1 s(t)+n_1(t)$$
$$x_2(t)=A_2 s(t-\Delta t)+n_2(t) \quad (1)$$

式中，A_1 和 A_2 分别为信号源至天线接收机的传输增益。计算两接收机输出信号的相关函数，有：

$$r_{12}(\tau)=E[x_1(t)x_2(t+\tau)]=Cr_s(\tau-\Delta t) \quad (2)$$

式中，C 为常数。由相关函数的性质可知，当 $\tau=\Delta t$ 时，相关函数的模达到最大值，因此按 $r_{12}(\tau)$ 达到最大时的值估计时差 Δt。

考虑离散系统，对于采用长度为 N 点的序列 $x_1(n)$ 和 $x_2(n)$，相关函数 $r(n)$ 的主峰对应的时间就是 $x_1(n)$ 和 $x_2(n)$ 之间的时延值。传统的算法以FFT为基础计算互谱，用IFFT计算相关函数。但是当信号的采样频率较低时，计算出的相关函数的分辨率也就较低。为了提高相关峰的分辨率，通常有两种方法：提高采样频率和相关峰插值。采样频率越高，估计误差越小，但是采样频率越高对芯片的性能要求也就越高。例如，要求1ns的估计误差时，时钟频率就需要提高到1GHz，常用的数字信号处理芯片很难达到这么高的处理速率，同时由于电磁干扰问题也会给电路板的布线、材料选择、加工等带来更高的要求。对相关峰进行插值计算，如采用最小二乘法或者三次样条插值，不仅增加了计算量，而且会引入新的误差。

本文提出一种在低采样率情况下，通过频域插值来提高精度的互谱插值时延估计方法，并结合VHDL语言在FPGA上加以实现。

1 互谱插值时延估计算法原理

根据采样定理，对时域连续、频谱受限的相关函数 $r(t)$ 以采样间隔 T_s 进行采样，抽样序列 $r(n)$ 的频谱 $R_s(f)$ 应该为 $r(t)$ 频谱 $R(f)$ 的周期重复，其重复间隔等于采样率 f_s，只要满足 $f_s \geq 2f_{max}$，就可以从 $R_s(f)$ 中恢复出 $r(t)$。所以，把 $R_s(f)$ 在频域拉开，相当于扩大频谱的重复间隔 f_s，逆变换得到的 $r(n)$ 波形不会改变，更不会带来新的误差，但是 $r(n)$ 的采样率得以提高。

根据互谱插值算法的核心思想，即频域补零可以提高相关函数时域波形的分辨率，对采样长度为 N 点的

信号 $x_1(n)$和 $x_2(n)$分别做 $2N$ 点补零 FFT 运算，得到信号的频谱 $X_1(k)$与 $X_2(k)$，根据相关定理，互谱为 $R(k)=X_1(k)\cdot X_2^*(k)$。将 $R(k)$在互谱间隔处进行补零扩展，即将互谱扩大，取 $N_1\geq 2N$，在互谱序列中补 N_1-2N 个零，拉长了频谱，构成 $R_1(k)$序列：

$$R_1(k)=\begin{cases}R_1(k), & k=0,1,\cdots,N-1\\ 0, & k=N,N+1,\cdots,N_1-N-1\\ R(2N-N_1+k), & k=N_1-N,N_1-N+1,\cdots,N_1-1\end{cases}\quad(3)$$

计算其逆变换，得到相关函数采样序列 $r_1(n)$，相当于把相关函数 $r(n)$的采样率提高到 $\frac{N_1}{2N}\cdot f_s$[1]。

为了提高 $r(t)$的采样率以获得高精度的时延估计值，就需要把 N_1 取得足够大，但这直接导致了 IFFT 点数的加大，增加了很多运算量，使整体时延估计的计算时间很长。因此，在 FPGA 实现时应综合考虑时延估计的精度和计算量，合理选取。

2 FPGA 实现

系统整体框图如图 1 所示。FPGA 实现过程：首先要计算的是采样长度为 N 点的信号 $x_1(n)$和 $x_2(n)$的 $2N$ 点补零 FFT 运算，信号输入单元需要完成对输入数据的补零；FFT 运算器计算两路信号的频谱；互谱插值单元通过复数乘法器计算两路信号的互谱，借助 FPGA 芯片内部 RAM 对数据缓存以实现插值运算；IFFT 运算器对插值后的频谱进行逆变换得到相关函数的采样序列；时延估计单元通过搜索相关函数的最大值，得到时延估计值。

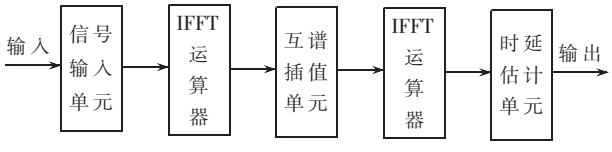

图 1 系统整体框图

信号输入单元和时延估计单元可用选择器、计数器、比较器、寄存器等电路实现，相对简单，在此不作详述。下面重点介绍 FFT/IFFT 运算器、互谱插值单元中复数乘法器的实现以及流水线控制问题。

2.1 FFT/IFFT 运算器设计

本系统输入信号为两路实信号，因此可以用一次 DFT 来计算两路实信号的频谱[2]，从而有效地减少计算工作量，提高计算速度，节省系统资源。该方法可归纳为如下三个步骤：

(1)$x_1(n)$和 $x_2(n)$是两个实函数，$n=0,1,\cdots,N-1$，将其中的一个作为实部而另一个作为虚部，构成复函数 $z(n)$，$z(n)=x_1(n)+jx_2(n),n=0,1,\cdots,N-1$。

(2)计算 $z(n)$的 N 点 DFT 得：$Z(k)=\sum_{n=0}^{N-1}z(n)W^{nk}=Z_r(k)+jZ_i(k),k=0,1,\cdots,N-1$，其中，$Z_r(k)$和 $Z_i(k)$分别是 $Z(k)$的实部和虚部。

(3)从 $Z(k)$中分离出 $X_1(k)$和 $X_2(k)$：

$$X_1(k)=\left(\frac{Z_r(k)+Z_r(N-k)}{2}\right)+j\left(\frac{Z_i(k)-Z_i(N-k)}{2}\right)$$

$$X_2(k)=\left(\frac{Z_i(k)+Z_i(N-k)}{2}\right)-j\left(\frac{Z_r(k)-Z_r(N-k)}{2}\right)$$

$X_1(k)$和 $X_2(k)$分别是 $x_1(n)$和 $x_2(n)$的 DFT。

FFT 运算采用 Altera 公司的 FFT MegaCore 实现[3]，该内核是一款高性能、高度参数化的 FFT 处理器，采用四输出引擎(Quad-output FFT engine)结构和连续(Streaming)I/O 数据流方式。FFT 模块接口如图 2 所示。具体的工作流程：系统复位后，等待采样数据输入；start 信号置位，表示输入数据块的起始，下一个时钟 start 被清零，输入数据按照自然顺序输入；输入数据达到预先设定的点数时，系统自然启动 FFT 运算；当 FFT 转换结束时，运算结果按照自然顺序被输出，在输出过程中，sop 信号被置位，表示输出数据块的起始。

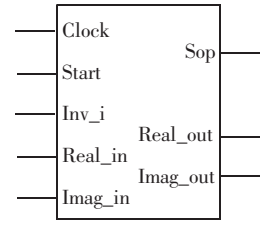

图 2 FFT 模块接口

IFFT 运算器设计时，根据 DFT 和 IDFT 的公式可得：$x(n)=\frac{1}{N}\{DFT[X^*(k)]\}^*$，只要先将 $X(k)$取共轭，就可以直接利用 FFT 子程序，最后再将运算结果取一次共轭，并乘以 $1/N$，即得到 $x(n)$。因此，IFFT 运算和 FFT 运算可以共用一个模块，通过选择信号 inv_i 决定执行 FFT 运算还是 IFFT 运算，当 inv_i 输入 0 时执行 FFT 运算，输入 1 时执行 IFFT 运算。需要注意的是 inv_i 信号必须和 start 信号严格同步。

2.2 复数乘法器设计

复数乘法器完成两路信号的互谱计算，其设计的好坏直接影响整个系统的性能。复数乘法可表示为：$(a+jb)\times(c+jd)=(ac-bd)+j(ad+bc)$，1 次复数乘法需要 4 次实数乘法和 2 次实数加/减法来实现。采用组合逻辑电路构成的乘法器可以使系统具有较高的工作速率，但这样的乘法器需占用大量的硬件资源，很难实现宽位乘法器功能。FPGA 中的 DSP 块是为实现多种最大性能和最小逻辑资源利用率的 DSP 功能而优化的，每个 DSP 块提供了乘法器、加法器、减法器、累加器及求和单元。在一个 DSP 块中即可实现复数乘法[4]，如图 3 所示。当输入为信号频谱 $X_1(k)$和 $X_2(k)$的实部和虚部时，输出即为互谱的实部和虚部。

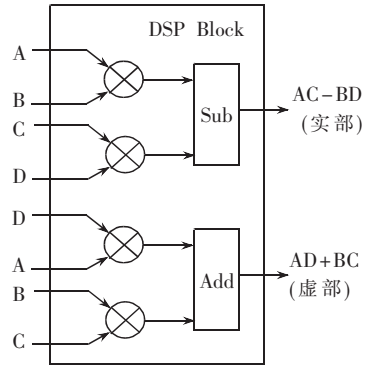

图 3 DSP 块实现复数乘法器

2.3 流水线操作设计

流水线处理是高速设计中的一种常用设计手段。由图 1 可见，每一次时延估计都要经过信

号输入、FFT 计算、互谱插值、IFFT 计算、时延估计单元等模块,整个数据处理是“单流向”的,即没有反馈或者迭代运算,前一个模块的输出是下一个模块的输入,因此可以考虑采用流水线设计方法来提高系统的工作速率。流水线设计的结构示意图如图 4 所示。在运行过程中这几个步骤是重叠的,即在一次估计时,还依次完成了后面几个步骤,因此将平均时延估计时间降低到最小值,用面积换取了速度。

3 仿真结果

仿真工具采用 Alteria 公司的 Quartus Ⅱ4.2 和 MathWork 公司的 MATLAB 6.5 软件。Quartus Ⅱ仿真文件可以有两种格式:vwf 格式和 vec 格式。前者是在 Quartus Ⅱ中通过波形编辑器生成的,当仿真输入数据复杂时,用波形输入的方法将难以胜任。后者是有一定格式要求的文本文件输入方式,可以利用任何文本编辑器产生。本文选用 MATLAB 软件对 vec 文件进行编辑是因为其强大的计算能力。Quartus 的仿真结果如图 5 所示,由于是数字形式,不是很直观,所以将仿真结果保存为 tbl 文件,调入 MATLAB 观察相关函数波形,结果如图 6(a)所示。图 6(b)是用 MATLAB 程序实现互谱插值时延估计算法的理论结果。比较实际仿真结果和理论算法结果,可以看出两者基本一致,从而验证了 FPGA 设计的正确性。

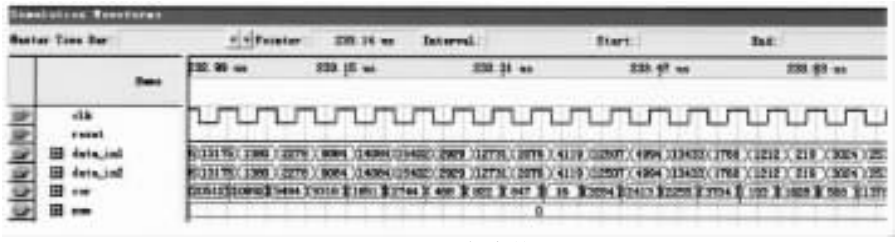

图 5 Quartus 仿真结果

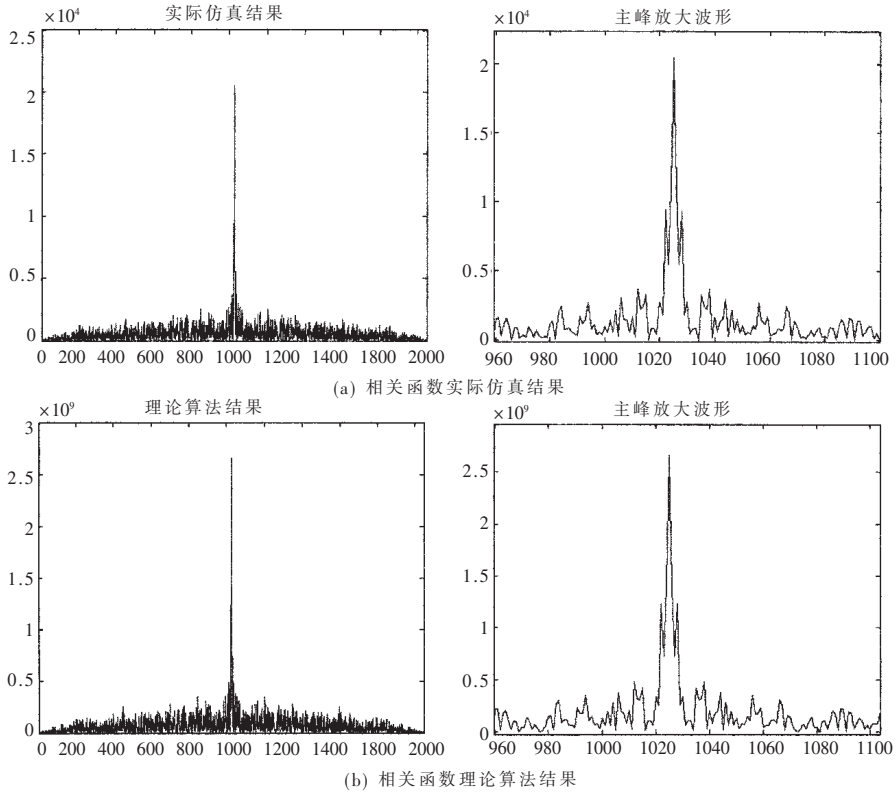

图 6 仿真结果比较

本文用 FPGA 实现了一种互谱插值高精度时延估计算法。该方法在计算相关函数的过程中,直接计算精细的相关峰值,使得峰值附近样本点间隔变小,减小了时延估计误差,同时不需要计算插值,减少了误差源。结合 FPGA 芯片特性,用 VHDL 语言编程实现了整个算法,并通过流水线处理技术提高了整个系统的工作效率,因而特别适合于实时的时延估计。

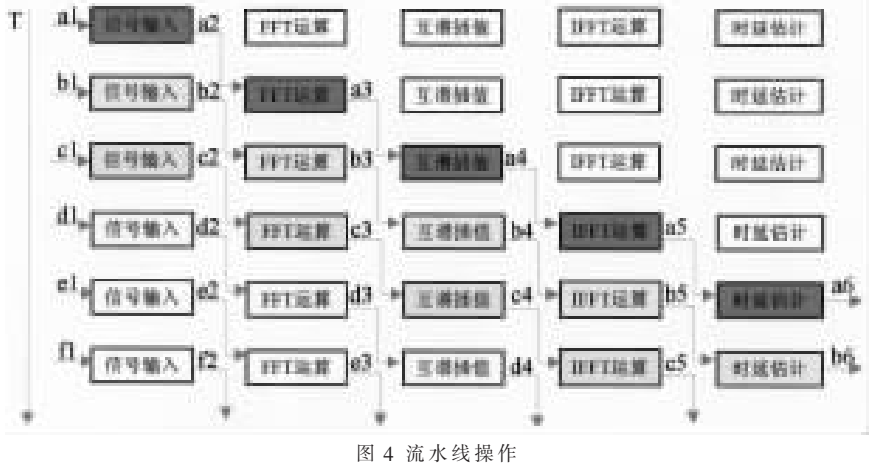

图 4 流水线操作

参考文献

[1] 周 莉. 时延估计系统的研究与设计[D].杭州:杭州电子科技大学出版社,2004.

[2] 程佩清.数字信号处理教程[M].北京:清华大学出版社,2001.

[3] 陈德望,汤淑明,宫晓燕,等. 城市交叉口交通信号控制研究的发展与展望[J].自动化博览,2002,(1):48-50.

[4] Altera Corporation.Stratix Device Handbook. http://www. altera.com.2005.

(收稿日期:2006-09-26)