

LFM 脉冲压缩的 FPGA 时域实现

陆 聪, 黄敬华, 杨维明, 曾张帆

(湖北大学 计算机与信息工程学院, 武汉 430062)

摘要: 为解决频域法实现线性调频 (LFM) 脉冲压缩时硬件开销较大的问题, 采用时域法实现; 针对间距为 20 m 的两目标 LFM 信号, 设计了一款 64 阶分布式 FIR 时域匹配滤波器; 采用全流水线并行处理结构实现, 并利用 FPGA 的 ROM 宏模块构建查找表代替乘法运算, 既提高了运算速度又减小了硬件开销; 基于 FPGA 器件 EP2C35F672C8 完成了 LFM 信号时域脉冲压缩的逻辑设计与集成; 仿真结果显示, 系统占用 2268 个逻辑单元、1573 个寄存器、27K 字节存储器资源。

关键词: LFM 信号; 分布式滤波算法; 时域; 脉冲压缩; FPGA

Time Domain Implementation of LFM Pulse Compression by FPGA

Lu Cong, Huang Jinghua, Yang Weiming, Zeng Zhangfan

(School of Computer & Information Engineering, Hubei University, Wuhan 430062, China)

Abstract: To solve the problem that the hardware overhead of the linear frequency modulated (LFM) pulse compression is large when realized in frequency domain, the time domain implementation method was used in this paper. A 64th-order distributed matched filter is designed for the LFM signal of two targets 20 meters away. The filter was implemented in pipeline-liked parallel competition structure, and the lookup table was built to replace the multiplication by using ROM macro-module of FPGA, which improved the operating speed while reduced the hardware overhead. The logic circuit of the filter was designed and integrated in FPGA device EP2C35F672C8. The simulated results indicate that the filter consumes 2268 logic cells, 1573 registers and 27K bytes memory resources.

Keywords: LFM signal; distributed filter algorithm; time domain; pulse compression; FPGA

0 引言

现代雷达对速度分辨力和距离分辨力性能均提出了较高的要求, 为了兼顾这两项指标, 常采用脉冲压缩技术, 即在发射端发射大时宽的线性调频 (LFM) 脉冲, 在接收端进行匹配滤波实现脉冲压缩, 以获得大带宽的回波信号, 这样可同时获得较高的速度分辨力和距离分辨力^[4]。脉冲压缩可采用时域法和频域法两种方式实现。频域法实现时速度较快, 但需多次用到 FFT 和 IFFT, 硬件开销较大; 采用时域法实现时电路结构简单, 但运算速度较慢。

本文采用分布式算法设计 FIR 结构的匹配滤波器, 并对通过对匹配滤波器窗函数进行改进, 改善了脉冲压缩的效果, 提高了目标距离分辨能力。基于 FPGA 完成了对 LFM 信号脉冲压缩的时域实现, 具有节省硬件开销和运算速度快的双重优势。

1 LFM 脉冲压缩原理分析及窗函数加权

1.1 LFM 脉冲压缩原理

LFM (线性调频) 信号表示为:

$$s(t) = \text{rect}(t/T) \exp(j\pi k t^2) \quad (1)$$

$\text{rect}(t/T)$ 表示信号的幅度。用 s_{i_n} 表示第 i 个点目标的回波信号:

$$s_{i_n} = A_i \cdot \text{rect}\left(\frac{t - \Delta t_i}{T}\right) \exp[j\pi k (t - \Delta t_i)^2] \quad (2)$$

总的回波信号: $s_{\Sigma}(t) = \sum_{i=1}^N s_{i_n}(t) \Delta t_i = 2R_i/c$, R_i 为第 i 个

目标的距离。

雷达发射端发射宽脉冲的 LFM 信号, 在雷达的接收端对回波信号进行处理获得窄脉冲信号, 这种由宽脉冲信号到窄脉冲信号的处理过程被称为脉冲压缩, 其核心就是匹配滤波器。

匹配滤波器的原理: 回波信号与滤波器传输系数进行时域卷积, 并不断调整传输函数 $h(t)$, 使匹配滤波器的脉压效果达到最佳, 同时输出信号的信噪比达到最大值。设 $s(t)$ 、 $h(t)$ 分别是回波信号和滤波器的传输系数, 得到匹配滤波器的输出为:

$$y(t) = s(t) * h(t) = \int_{-\infty}^{\infty} s(\tau) h(t - \tau) d\tau \quad (3)$$

令 $h(t) = s(-t)^*$, 则滤波器输出为:

$$y(t) = \int_{-\infty}^{\infty} s(t - \tau) h(\tau) d\tau = KR(t_0 - t) \quad (4)$$

K 代表幅度归一化常数。匹配滤波器的核心: 对回波信号 $s(t)$ 与滤波器的传输函数 $h(t)$ 进行线性卷积, 滤波器的长度由发射脉冲的带宽 B 、时宽 T , 以及采样倍数 n 决定的。脉冲压缩后回波信号的主瓣宽度: $\tau = 1/B$, 脉冲宽度比: $D = BT = T/\tau$, D 也称为发射脉冲的时宽带宽积。由式 (4) 可知, 当 $h(t)$ 为回波信号 $s(t)$ 的镜像函数时, 匹配效果最佳, 由于回波信号是动态的、不确定的, 观察图 (a) 发现, 回波信号类似于带有波纹的 LFM 信号, 所以可采用发射脉冲信号的复共轭近似代替匹配滤波器的传输函数。

发射脉冲的设计参数: $T = 3 \mu\text{s}$, $B = 10 \text{ MHz}$ 时, 距离分辨力 $\delta = c/2B = 15 \text{ m}$, c 为光速, 目标间距设计为 20 m。图 1 为双目标 LFM 回波信号脉冲压缩前后的仿真波形: 其中 (a) 是脉冲压缩前的回波信号, 该回波信号的波形近似于带波纹的发射脉冲, 没有明显的目标信息, 需要对该回波信号进行脉压处理; (b) 为回波信号经脉冲压缩后的仿真结果, 观察发现,

收稿日期: 2017-09-08; 修回日期: 2018-03-13。

基金项目: 国家自然科学基金资助项目 (61601175)。

作者简介: 陆 聪 (1993-), 男, 主要从事雷达信号处理方向的研究。

通讯作者: 黄敬华 (1968-), 女, 硕士, 副教授, 主要从事电工与电子技术方向的研究。

脉冲压缩后信号的主瓣与旁瓣区分明显, 过渡带更加陡峭, 很容易提取目标的有效信息, 通过对主瓣时间间距进行计算, 可以获得较为准确的目标间距。

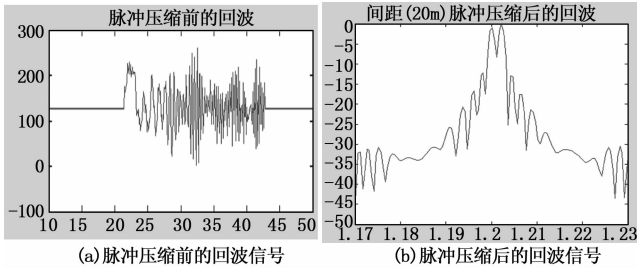


图 1 两目标 LFM 回波信号脉压前后的波形

1.2 对匹配滤波器进行窗函数加权

对回波信号直接进行脉压得到的窄脉冲信号旁瓣比较大, 信号能量不集中, 造成能量的损失, 要接收到信噪比较大的信号就需要发射端发射更大功率的信号, 故需要一种方法使信号的能量尽可能集中在主瓣上。常见的抑制旁瓣的方式就是窗函数加权, 不仅使信号能量集中在主瓣, 也使目标信息的提取变得更加容易。常见的窗函数有矩形窗、三角窗、海明窗以及布莱克曼窗, 且不同窗函数的频谱特性不同。采用窗函数加权的最终目的是抑制旁瓣、减小主瓣宽度, 但是这两个目标相互矛盾, 不能同时实现, 所以窗函数的选择要根据实际需求。以经典 Hamming 窗为例先对匹配滤波器加权处理, 窗函数表达式为:

$$w(t) = 0.54 - 0.46\cos\left(\frac{2\pi t}{T}\right) 0 \leq t \leq T \quad (5)$$

其中: T 为发射脉冲的时宽, Hamming 窗加权后信号的旁瓣抑制较好, 但是过渡带变得平缓。在实际数字滤波器中, 受采样频率、数据位宽以及滤波器长度的限制, 当目标距离较近时, 目标识别的难度加大。虽然旁瓣受到了有效抑制, 但是平滑的过渡带降低了目标的可识别性。为改善这种情况, 对 Hamming 窗进行改进。

LFM 信号的能量主要集中在低频, 而回波信号与发射脉冲的波形相似, 所以在匹配滤波器进行线性卷积时, 主瓣的能量也主要来自低频部分, 而且发射脉冲采用的是上变频 LFM 信号, 故得到改进后的加窗函数为:

$$w_h(t) = 0.54 - 0.46\cos\left(\frac{2\pi t}{T}\right) T/2 \leq t \leq T \quad (6)$$

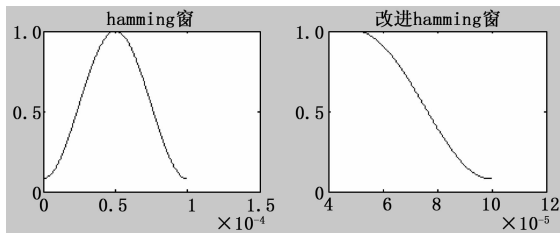


图 2 窗函数时域仿真波形

对比改进前后的窗函数可以看出, 改进窗只有 Hamming 函数的一半, 也近似于 LFM 信号的第一个周期的一半。对滤波器进行加权就是对滤波器的传输系数进行加权, 得到新的传输系数为: $h_h(t) = h(t) \cdot w(t)$, 带入式 (4) 得到加权后匹配滤波器的输出为:

$$y(t) = \int_{-\infty}^{\infty} s(t-\tau)h(\tau)W(\tau)d\tau = KR(t_0-t)W(t_0-t) \quad (7)$$

其中: $W(t)$ 为 $w(t)$ 或 $w_h(t)$, 回波信号经过窗函数加权后的匹配滤波器进行脉冲压缩 (以间距 $d = 20$ m 为例), 对比 Hamming 窗改进前后的仿真结果, 如图 3 所示。对比图 1、图 3 看出, 进行窗函数加权后的回波信号能量更加集中, 降低了能量的损失, 然而波峰间的过渡带变得平缓, 目标识别难度加大。与传统的 Hamming 窗相比, 使用改进窗函数进行加权可以获得较为陡峭的过渡带, 提高目标的距离分辨能力, 而且旁瓣进一步被抑制。

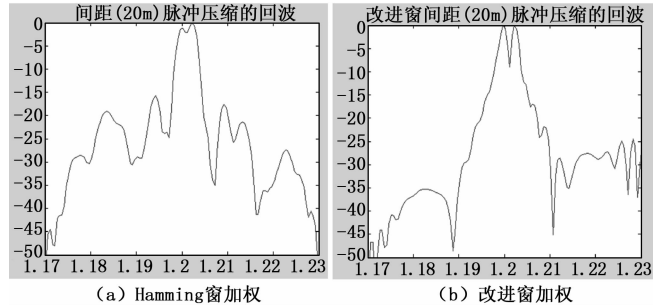


图 3 窗函数改进前后的脉压波形

2 分布式算法原理和 FPGA 实现

2.1 分布式算法原理分析

时域数字卷积是通过 N (N 为采样长度) 阶 FIR 结构的滤波器实现的^[2], 由上节可知滤波器的输入数据和匹配系数都是复数形式。实现一次长度为 N 的卷积运算需要 $4N$ 次乘法器和 $4N-1$ 次加法器^[5]。当 N 取值 64 时, 完成一次实时卷积运算就需要 256 个乘法器同时运行, 然而一般的 FPGA 芯片没有如此多的嵌入 IP 核, 所以在考虑实际成本和不降低运算速度的情况下, 采用分布式算法, 设计多条流水线并行执行的匹配滤波器, 可以很好解决高阶滤波器设计受限于乘法器资源不足的情况。

分布式算法就是利用嵌入在 FPGA 芯片的存储模块和丰富的逻辑资源, 采用数据存储、格式转换、地址查询的方式代替卷积运算中的乘法器。分布式滤波器设计是先将卷积运算结果的所有可能情况存储在 RAM 模块, 接着将输入数据转换成存储模块的寻址数据, 并进行查表输出, 然后对存储器的输出进行移位求和就得到实际卷积运算的结果, 该算法实现乘法到存储器、寄存器的转换, 利用 FPGA 丰富的存储和逻辑资源代替稀有的 IP 核, 使普通 FPGA 芯片设计高阶滤波器成为可能, 节省了硬件成本。分布式算法具体如下:

首先对回波信号进行采样, 得到滤波器的输入信号 $x(n)$, 二进制表示形式为:

$$x(n) = \sum_{k=0}^{b-1} x_k(n)2^k \quad (8)$$

$x_k(n)$ 表示 $x(n)$ 的第 k 位, b 是采样数据的位长, 则匹配滤波器输出:

$$y(n) = \sum_{n=0}^{N-1} x(n)h(N-1-n) = \sum_{n=0}^{N-1} h(N-1-n) \sum_{k=0}^{b-1} x_k(n)2^k = \sum_{k=0}^{b-1} 2^k \sum_{n=0}^{N-1} h(N-1-n)x_k(n) \quad (9)$$

由式 (9) 看出, 首先输入信号第 k 位的值 (1 或 0) 与滤波器系数进行与运算并求和, 然后将上一步的累加和左移 k 位 (2^k 相当于左移 k 位), 再次求和最终得到 $y(n)$, 此方法就是将卷积运算转变为逻辑运算和移位求和的过程。通过此方法可以

看出, 只要知道第一步的累加值, 再进行移位求和就可以得到时域卷积的值。在实际设计中首先将第一步累计和的所有可能值存储在芯片嵌入的 RAM 块中, 然后对输入数据进行变换, 作为存储器的寻址数据, 对存储器输出的数据进行移位求和, 这就是分布式算法设计滤波器的大致流程。

为了改善脉冲压缩后信号的信噪比, 需要对设计的滤波器进行窗函数加权, 设 $w(n)$ 为窗函数经采样后的数据, 采样长度为为滤波器的设计阶数, 得到滤波器的传输系数变为: $h(0)w(0), h(1)w(1), \dots, h(63)w(63)$ 。

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \quad n = 0, 1, \dots, N-1 \quad (10)$$

为了设计方便和节省资源, 先将加权后的系数值带入式 (9), 即先对滤波器进行加权, 此时滤波器的输出即为窗函数加权后的滤波结果。

2.2 分布式算法的 FPGA 实现

由式 (9) 可知, 当采用滤波器设计长度为 64 阶, 式 $\sum_{n=0}^{N-1} h(N-1-n)x_i(n)$ 的可能乘积项有 2^{64} 种, 直接采用 ROM 表进行存储, 则需要 2^{64} 个存储单元, 对于现有的 FPGA 芯片是不可能实现的。所以针对 FIR 结构较长的情况, 采用分布式算法设计匹配滤波器时, 需要设计多条流水线并行执行的结构, 对总表进行切割就可以减少存储资源的使用量。

匹配滤波器的设计如图 4, 设计 8 条流水线并行处理的结构, 此时每条流水线都为 8 阶 FIR 结构的滤波器, 每条流水线的查找表大小为 2^8 单位, 流水线设计将卷积运算的 RAM 使用量降到 8×2^8 单位, 这使得一般的 FPGA 芯片都可以满足。甚至可以进一步采用 16 条流水线的进行电路设计, 将查找表的使用量降为 16×2^4 单位, 不过模块衔接复杂, 硬件电路设计难度加大, 需要注意更多的时序问题。本次设计采用 8 条流水线并行执行的 FIR 结构, 匹配滤波器的具体设计框图如图 5。

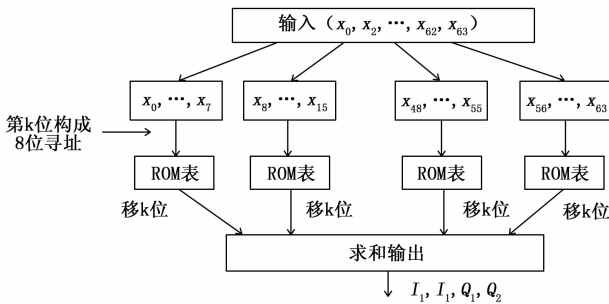


图 4 匹配滤波器的结构

图 4 中 k 表示输入数据的第 k 位, 每个 ROM 表存储 8 阶 FIR 结构采用分布式算法的所有可能乘积项, 需要个存储单元, 对输入数据进行转换并寻址存储器, 接着将 ROM 表的输出数据进行移位 (左移 k 位) 求和, 输出的值为图 5 中的一个值。因为滤波器的输入数据是复数, 由复数乘法可知, 故需要 4 条图 4 的流水线设计。匹配滤波器整体设计结构如下图:

由图 5 可知, 首先将回波信号进行分解, 然后对回波信号的实部和虚部进行采样、存储及数据转换, 滤波器最终输出的是回波信号脉冲压缩后的模值。然而传统的求模方式依旧要用到乘法器和开方运算, 实现开方运算非常复杂而且延时比较高, 所以需要找到一种简单的模值估算方法求取信号的模值,

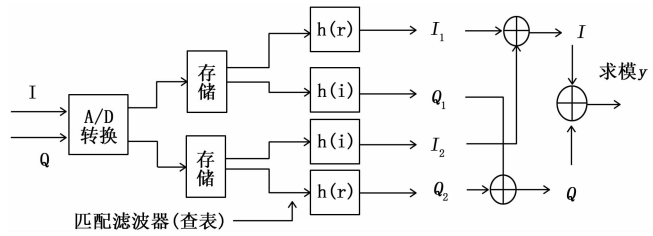


图 5 匹配滤波的总体结构

且延时较低。设信号的模值为 Y , 估算算法^[3]为:

$$Y = \text{MAX}\{\text{MAX}(|I|, |Q|), 7/8\text{MAX}(|I|, |Q|) + 1/2\text{MIN}(|I|, |Q|)\} \quad (11)$$

据统计, 采用该复数求模公式对信号的损失不超过 0.13 db, 其中 $7/8\text{MAX}(|I|, |Q|)$ 可以采用移位寄存器与加法器的结合来实现。至此匹配滤波器的整体结构完成, 整个设计完全使用寄存器和加法器资源, 理论上只要 FPGA 的 ROM 和加法器资源足够, 可以设计任意长度的滤波器。

3 匹配滤波器的 FPGA 设计与测试

3.1 基于 FPGA 设计匹配滤波器

采用 ALTERA 公司的 FPGA 器件 EP2K35F672C8 进行匹配滤波器的设计集成。图 6 是利用 FPGA 设计实现脉冲压缩功能的匹配滤波器的整体结构。ROM_real、ROM_imag 模块存储的是回波信号经分解、采样、量化处理后的实部和虚部数据, 在时钟 CLK 控制下, 通过 counter 计数器计数作为存储模块的输入地址产生滤波器的输入数据; address 模块对输入数据的第 k 进行转换作为分布式算法查找表模块的输入地址, 通过该地址提取模块 ROM_R0toR7、ROM_I0toI7 模块中的数据 (完成乘积项的提取), 接着对输出的数据移位、求和得到图 5 中 I_1, I_2, Q_1, Q_2 的值; 此时得到回波信号经过脉冲压缩后的实部、虚部, 且都为有符号数据, 需要进一步设计求模电路方便数据分析。

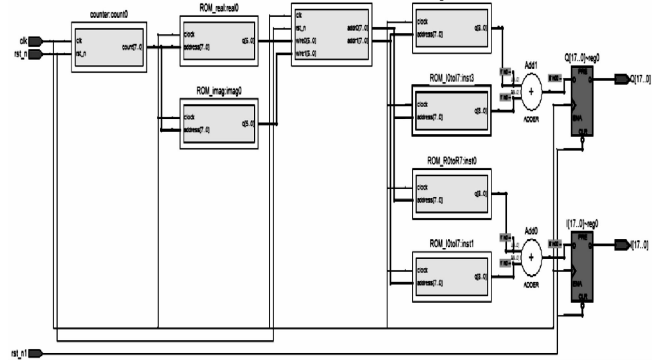


图 6 匹配滤波器设计结构

图 7 是由式 (11) 设计的求模电路, 采用 xor2 模块求取回波信号实部和虚部的绝对值 (输入数据与其最高位逐位异或), 通过比较器和加法器对输出信号求模, 输出 data4 [17:0] 为回波信号脉压后的近似模值。至此基于 FPGA 的匹配滤波器的整体硬件电路设计已经完成, 进一步对设计好的电路添加时序约束并进行时序仿真分析, 检测整个电路是否满足要求。首先编写 Test Bench 驱动文件, 设计时钟精度并初始化电路, 接着加载 ROM 模块存储的数据进入滤波器进行处理,

最终将滤波器输出的是数据通过波形显示器显示，观察回波信号脉冲压缩后的波形。

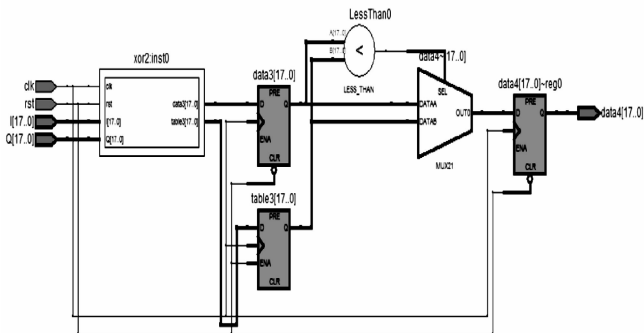


图 7 求模电路结构

3.2 滤波器测试结果

LFM 信号设计参数：带宽，时宽，采样频率，时钟频率，距离分辨率 $\delta=15\text{ m}$ 。通过 Modelsim 加载滤波器的输出数据，Wave 设置为 Analog 模式，观察脉冲压缩后信号的波形。图 8 是两目标回波信号脉压前实部和虚部的测试波形，与 MATLAB 仿真结果一致，直接对回波信号进行分析，是难以提取到目标信号的有用信息，需要进一步对回波信号进行匹配滤波。

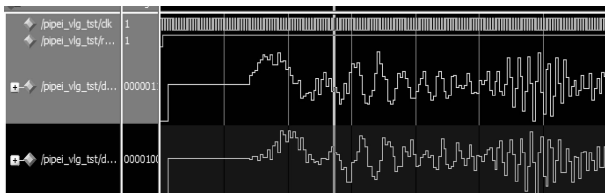
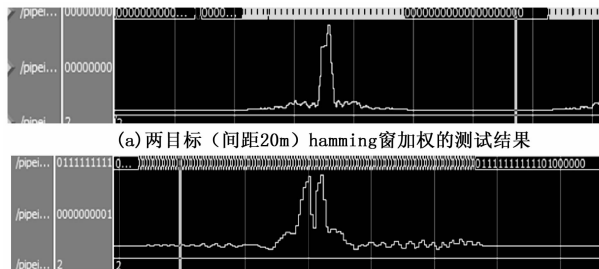


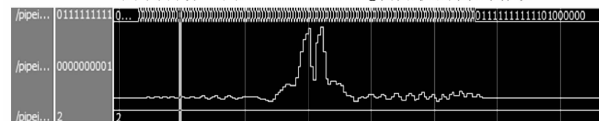
图 8 两目标脉冲压缩前的回波信号

图 9 分别为两目标间距 20 m 的回波信号。可以看出，与图 3 的 Hamming 窗加权仿真结果相比，经 FPGA 设计数字滤波器处理后，目标的分辨性能下降，这主要是由于采样频率与采样精度受限的影响。考虑 FPGA 硬件资源有限和成本的控制，过多地提高采样频率和采样精度的方法并不可取，故需要提高算法的有效性。对窗函数进行改进并加权，观察滤波器的输出波形如下所示，此时目标信号区分明显，虽然主瓣的宽度却有所增加，但是信号能量依旧集中在主瓣。对图 9 的测试波形进行分析，测得两主瓣间距 $\Delta t=16\text{ ns}$ ，经计算测量间距 $\Delta d=f_{dk} \cdot c\Delta t/2f_p=20\text{ m}$ ，与设计的目标间距一致，这说明改进窗函数能够改善因采样频率、采样精度受限造成信息损失的影响。

仿真报告显示，该数字滤波器设计占用 2 268 个逻辑单



(a) 两目标（间距20m）hamming窗加权的测试结果



(b) 间距20m改进hamming窗加权的测试波形

图 9 两目标脉冲压缩后的测试结果

元、1 573 个寄存器、28 032 字节存储器。可以看出使用寄存器和存储器代替卷积运算中的乘法器时，逻辑资源、存储资源的使用量都比较大。当采用级联结构设计匹配滤波器时，资源的使用量会成倍减小，但运算速度也会成倍降低，二者不能同时兼顾，具体选择要根据实际要求。

4 结束语

本次设计根据 FPGA 并行执行机制，采用分布式算法设计 FIR 结构的匹配滤波器。以寄存器和加法器代替乘法运算，不仅保证了数据处理的速率，而且打破了 FPGA 的乘法器资源的限制，可以设计高阶滤波器。通过对仿真结果进行分析，与直接对回波信号进行压缩相比，Hamming 窗函数加权后，可以降低旁瓣，但目标距离分辨性能并不理想，使用改进窗函数加权后，目标的距离分辨能力明显改善。

本次设计采用 64 阶的 FIR 结构以及全流水线同步算法，虽然完全替代了乘法器的使用，但是资源的使用量依旧较大。若采用串行处理的方式，每削减一半的流水线，硬件资源的使用量就会减少 1/2，所以如何考虑硬件成本和处理速度需要结合实际情况。

参考文献：

[1] 杜 勇. 数字滤波器的 MATLAB 与 FPGA 实现 [M]. 北京：电子工业出版社，2015.

[2] Baese U M. 数字信号处理的 FPGA 实现 [M]. 刘 凌译. 北京：清华大学出版社，2011.

[3] 杨维明. 一种基于 EPLD 技术的信号取模方法 [J]. 湖北大学学报，1999，21 (2)：1-4.

[4] 丁智泉. 线性调频信号的脉冲压缩系统的设计与 FPGA 实现 [D]. 成都：电子科技大学，2006.

[5] 李文刚. 基于 FPGA 的高阶高速 FIR 滤波器设计 [D]. 成都：电子科技大学，2005.

[6] Li WZ, Wu LY, Guo L. Reduced-Order Active Control for Structural System with Nonlinear Uncertainty Based on Genetic Algorithm [J]. IFAC Proceedings Volumes, 2008, 41 (2)：13426-13431.

[7] 李亮亮, 何 勇, 叶海翔. 永磁同步电动机神经网络逆系统简化模型的研究 [J]. 微特电机, 2010, 38 (7)：57-59.

[8] 赵 君, 刘卫国, 骆光照, 等. 永磁同步电动机神经网络逆解耦控制研究 [J]. 电机与控制学报, 2012, 16 (3)：90-95.

[9] 王 新, 戴先中. 同步旋转坐标系下感应电动机神经网络逆控制 [J]. 电气传动, 2008, 38 (5)：52-57.

[10] 丁文双, 胡育文, 鲁文其, 等. 永磁同步电机模糊 PID 参数自整定 [J]. 微特电机, 2011, 39 (5)：17-21.

[11] 屈 毅, 宁 铎, 刘飞航, 等. 模糊 PID 控制器的设计及其仿真 [J]. 计算机仿真, 2009, 26 (12)：130-132.

[12] 朱志莹, 李祖明, 周云红, 等. 磁悬浮开关磁阻电机线性二次型最优逆解耦控制 [J]. 微电机, 2015, 48 (12)：45-50.

[13] 朱茂飞, 陈无畏, 夏 光. 基于神经网络逆系统方法的汽车底盘解耦控制 [J]. 农业机械学报, 2011, 42 (12)：13-17.

[14] 徐前勇, 张运陶. Paūta 准则在醚化温度预测建模异常数据剔除中的应用 [J]. 西华师范大学学报 (自然科学版), 2011, 32 (4)：348-352.

(上接第 76 页)