

文章编号: 2095-2163(2024)01-0124-06

中图分类号: TN919.5

文献标志码: A

融合压缩感知和 LZW 编码的电力数据压缩算法

谢宇杰, 陈志聪, 吴丽君

(福州大学 物理与信息工程学院微纳器件与太阳能电池研究所, 福州 350108)

摘要: 为了减少电力信号传输时的数据量, 本文提出了一种融合压缩感知和 LZW 编码的电力数据压缩算法, 在保证重构精度不变的条件下对电力数据进行进一步的压缩, 提高了整体的压缩率。首先, 对压缩感知的各种观测矩阵进行仿真分析; 其次, 选择使用稀疏随机矩阵作为本文的观测矩阵, 提出了一种能够快速完成压缩感知计算的硬件实现方法, 并完成了硬件的设计和验证。实验表明, 本设计在 FPGA 器件上的工作频率最高可达 200 MHz; 整个数据压缩过程的总延时约为 16.11 μ s; 在重构误差约为 4.83% 时, 数据压缩率约为 36.83%, 比仅使用压缩感知提升了约 13.17%。

关键词: 电力信号; 压缩感知; LZW 编码

Power data compression algorithm combining compressed sensing and LZW coding

XIE Yujie, CHEN Zhicong, WU Lijun

(Institute of Micro-Nano Devices and Solar Cells, College of Physics and Information Engineering, Fuzhou University, Fuzhou 350108, China)

Abstract: In order to reduce the amount of data during power signal transmission, this paper proposes a power data compression algorithm that integrates compressive sensing and LZW encoding. It further compresses the power data while ensuring the same reconstruction accuracy, thereby improving the overall compression rate. Firstly, simulate and analyze various observation matrices of compressed sensing; Secondly, we chose to use sparse random matrices as the observation matrix in this paper, proposed a hardware implementation method that can quickly complete compressive sensing calculations, and completed the hardware design and verification. Experiments show that the maximum operating frequency of FPGA can reach 200 MHz, and the total delay of the entire data compression process is about 16.11 μ s. When the reconstruction error is about 4.83%, the data compression rate is about 36.83%, which is about 13.17% higher than that of using compressed sensing only.

Key words: power signal; compressed sensing; LZW coding

0 引言

随着 5G 时代的到来, 人们对电能信息的需求量也大幅攀升, 带来的是电力系统中设备数量的增加, 使得电能质量问题愈加严重。无间断的数据采集和监测使得电力系统的负载剧增, 电力故障所带来的后果严重, 故必须对电力系统中产生的数据进行实时的监测。而直接对大量的电力数据进行采集和传输是不实际的, 必须对其进行压缩处理。目前很多相关研究提出了各种电力数据的压缩方法, 主要有小波变换 (Wavelet Transform, WT) 及其改进算

法、贪婪算法和图像算法等。这些数据压缩算法的共同点在于对信号进行采样时都受 Nyquist 采样定理的限制, 即首先都需要用较高的采样率对原始的信号进行采样, 再对采样后的信息使用压缩算法进行处理, 在压缩处理的过程中通过丢弃掉大部分采样过程中产生的冗余数据, 从而达到压缩的目的。

压缩感知理论 (Compressed Sensing, CS) 的诞生突破了采样定理的限制。若待压缩的原始信号本身或其某变化域下是稀疏的, 则可以利用某些观测矩阵将该信号投影到其低维空间上, 从而在对信

作者简介: 谢宇杰 (1998-), 男, 硕士研究生, 主要研究方向: 电力信号数据压缩; 吴丽君 (1984-), 女, 博士, 副教授, 硕士生导师, 主要研究方向: 计算机视觉。

通讯作者: 陈志聪 (1983-), 男, 博士, 副教授, 硕士生导师, 主要研究方向: 故障诊断、机器学习、智能信号处理等。Email: zhicong.chen@fzu.edu.cn

收稿日期: 2023-01-03

号进行采样的同时完成数据压缩^[1]。压缩感知不仅摆脱了采样定理的约束,且相比于传统算法,更易于计算和硬件实现。电力信号在频域上是稀疏的,故可使用压缩感知算法对其进行数据的压缩采样。

由于压缩感知属于有损压缩算法,必然会受重构误差的限制,即在到达一定重构误差时,压缩率将难以得到进一步提升^[2]。而 LZW (Lempel - Ziv - Welch, LZW) 算法基于字典编码,属于无损压缩算法,压缩率高,易于软硬件实现,尤适于对重复出现的数据进行压缩。对压缩感知采样后的数据进行 LZW 编码,能在不损失重构精度的条件下提升压缩率,适用于电力信号的数据压缩。因此,本文提出了压缩感知和 LZW 结合的电力信号数据压缩算法。

1 电力信号压缩算法

1.1 电力信号压缩算法流程

本文的电力信号压缩算法流程如图 1 所示。整个压缩算法分为 CS 压缩感知模块和 LZW 编码模块。待压缩的电力数据 x 输入到 CS 模块中,首先经过压缩感知运算,得到压缩采样数据 $f(x)$,再经过 LZW 模块的运算,得到压缩后数据 y 。

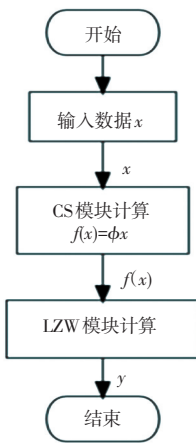


图 1 压缩算法流程图

Fig. 1 Flow chart of compression algorithm

1.2 压缩感知算法模型

对于信号的获取与传输,传统方法需要在满足采样定理的条件下,在进行数据压缩之前至少以原始信号两倍的采样频率对其进行采样,再对这些数据进行压缩,其次才进行传输。压缩感知理论的出现解决了传统方法采样后压缩的繁琐,信号的获取过程就是利用观测矩阵进行的压缩过程^[3]。使用观测矩阵使得信号的采样与压缩两个过程合并,如图 2 所示。具体过程为:输入数据 x 为一组数据量

为 256 大小的电力信号数据,将其作为待压缩计算的一个列向量。使用 128×256 大小的观测矩阵与输入数据进行矩阵乘,得到一组 128 维的列向量作为压缩感知计算后的压缩结果集 $f(x)$,完成压缩采样,压缩率为 50%。通过调整观测矩阵的行数,可以实现对压缩率的改变和控制。

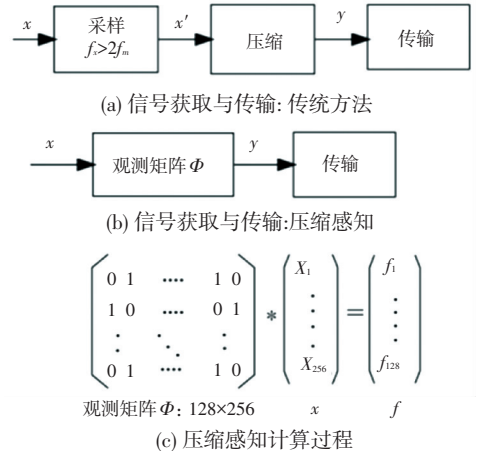


图 2 信号的获取与传输方法及压缩感知计算过程

Fig. 2 Signal acquisition and transmission method and compressed sensing calculation process

1.3 LZW 模块算法流程

CS 模块的输出数据输入到 LZW 模块进行 LZW 编码。整个 LZW 算法压缩过程由 LZW 模块中的状态机子模块控制,算法流程如图 3 所示。

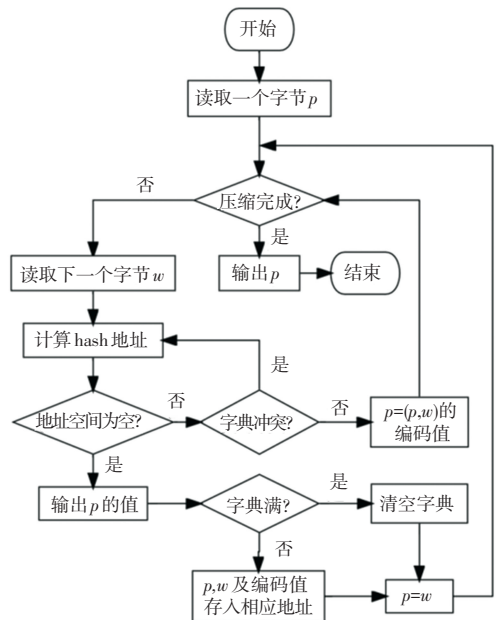


图 3 LZW 算法流程图

Fig. 3 Flow chart of LZW encoding

(1) 开始压缩,读取第一个字节 p ;

(2)判断压缩是否完成:若压缩未完成,则读取下一个字节 w ;否则进入步骤 8;

(3)根据读入的两个字节计算哈希地址,本文中的哈希函数为 $index = (w \ll 1) \wedge p$ 。若发生过冲突,则 $index = index + 1$;

(4)根据哈希地址读取字典,判断字典中该地址中的值是否为空:是则输出 p 值,进入(6),否则进入(5);

(5)判断字典是否发生冲突:是则进入(3),否则更新 p 的值为 (p, w) 组合数的编码值,并进入(2);

(6)判断字典是否已经存满:是则清空字典,否则将 p, w 及相应的编码值存入字典中;

(7)将 w 的值赋给 p ,并进入(2);

(8)压缩完成,输出结束信号。

2 压缩算法仿真分析

2.1 数据压缩效果的衡量指标

采用衡量数据压缩效果的指标对压缩算法的效果做出客观评价。

(1)压缩率(Compression Rate, CR):压缩率的大小决定了压缩效果。 CR 的值越小,说明压缩后的数据量相比于压缩前的比例越小,压缩算法的效果也越好,式(1):

$$CR(\%) = \frac{l'}{l} \times 100\% \quad (1)$$

其中, l' 表示压缩后输出数据长度, l 为其原始数据长度。

(2)均方根误差(Root Mean Squared Error, $RMSE$):反映了压缩数据在解压缩后与原始信号之间的误差。 $RMSE$ 越小,说明重构信号的失真越小,压缩算法的性能也越好,式(2):

$$RMSE(\%) = \frac{\sqrt{\sum_0^{N-1} (y(x) - f(x))^2}}{\sqrt{\sum_0^{N-1} (y(x))^2}} \times 100\% \quad (2)$$

其中, $f(x)$ 为原始信号, $y(x)$ 为重构信号。

2.2 观测矩阵的选择

观测矩阵的选择,不仅影响着整个系统的压缩效果,也决定了后续硬件实现的难易程度。压缩率设为 50%,选择稀疏变换和重构算法分别为离散余弦变换(Discrete Cosine Transform, DCT)和广义正交匹配追踪(generalized Orthogonal Matching Pursuit, gOMP)的条件下,4种观测矩阵对数据集中

的每条电压数据的平均 $RMSE$ 见表 1。软件仿真实验采用的是聚丙烯和全膜电力电容器的实验室提供的电流、电压和有功功率测量文件的数据库中的电压数据。数据集中的所有的电压波形均按照与谐波限制相关的 IEEE 标准进行配置。

表 1 不同观测矩阵的平均 $RMSE$

Table 1 Mean $RMSE$ of different measurement matrices

观测矩阵	平均 $RMSE/\%$
高斯随机矩阵	4.86
部分哈达玛矩阵	4.62
随机伯努利矩阵	4.86
稀疏随机矩阵	4.83

可见在所选择的 4 种观测矩阵中,压缩感知算法的重构性能没有太大差异。而从硬件设计和实现的角度,需要进一步考虑计算量、存储量、功耗、硬件实现难易等因素。为此,对于观测矩阵的生成采用的是提前在 ROM 中存储一个确定的观测矩阵,而非每次压缩都重新生成一个矩阵,这样在硬件实现的面积和功耗上都能有大幅节约,且易于实现。

对比各种观测矩阵的特点,以判断其硬件实现难易程度。高斯随机矩阵和随机伯努利矩阵计算过程复杂,且对存储资源消耗较大;部分哈达玛矩阵的维度必须为 2 的整数倍,应用场景有限;而稀疏随机矩阵仅由 0 和 1 两种元素组成,并且绝大部分元素都为 0,不仅需要的存储空间较小,在计算时也只需进行加法计算,计算复杂度低,在硬件实现上具有明显优势。但如果直接使用 ROM 进行稀疏矩阵的存储,对于一个 128×256 大小的观测矩阵,仍需花费大量的存储资源。为此,本文提出一种压缩感知的硬件实现方法:利用稀疏矩阵中大部分元素都为 0 的特点,在实际存储时,只需要存储每一行中的元素“1”在这行中的位置即可,然后再用一个“0”表示换行,在计算时利用状态机控制元素的读取、相加以及换行操作完成压缩感知运算。假设该 128×256 大小的稀疏矩阵是由一个 128×128 的单位矩阵和一个 128×128 的零矩阵拼接而成(该矩阵只用于描述算法,而非实际中使用),则 ROM 中存储的元素依次是 1, 0, 2, 0, ..., 128, 0, 仅需要 256 的深度。利用状态机依次读出 ROM 中的数据,每个非零数据表示了相应的存放输入数据的 RAM 地址,遇到连续的非零数时则将取出的数据相加,遇到零则将相加的数输出,随后置零并进行下一轮操作,算法流程如图 4 所示。这种对观测矩阵存储方式的优化,不仅节约了大量的存储空间,还加快了压缩采样的计算过程。

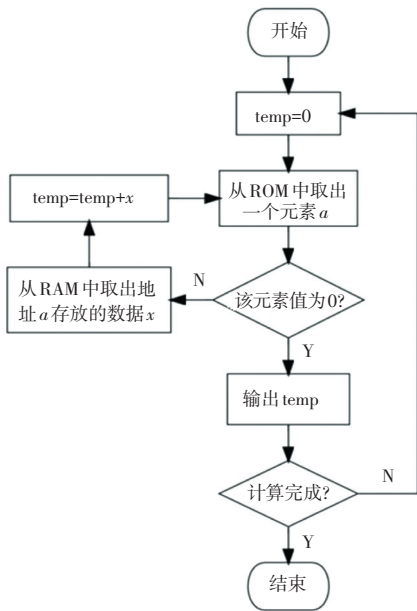


图 4 压缩感知算法计算流程

Fig. 4 Computing flow of compressed sensing

2.3 压缩仿真结果

在数据集中随机选取 6 条电压数据进行算法对比, 分别在仅使用压缩感知算法、仅使用 LZW 编码和使用压缩感知结合 LZW 编码的条件下, 比较输

入数据的压缩率, 以及重构算法为 gOMP 算法时的 RMSE, 结果见表 2。

表 2 算法性能对比

Table 2 Algorithm performance comparison

数据集	压缩率/%		RMSE/%	压缩率/%		RMSE/%
	CS	CS+LZW		LZW	RMSE/%	
1		36.4	5.8	70.1		
2		36.8	3.2	70.0		
3		36.4	5.2	69.6		
4	50.0	37.2	4.7	69.6		0
5		37.3	4.4	68.6		
6		36.9	5.7	69.7		

从表 2 可以看出, 本文提出的数据压缩算法可以达到的平均压缩率约为 36.83%, 平均 RMSE 约为 4.83%; LZW 无损压缩算法的加入不仅能使重构精度保持不变, 且压缩率也能得到提升, 比仅使用压缩感知或 LZW 编码分别提高了约 13.17% 和 32.77% 的压缩率。

数据集中的电压、电流和功率信号重构后的波形与压缩前波形的对比如图 5 所示。可以看出, 重构后的信号很好地还原了原始信号, 其 RMSE 均在 5% 以下。

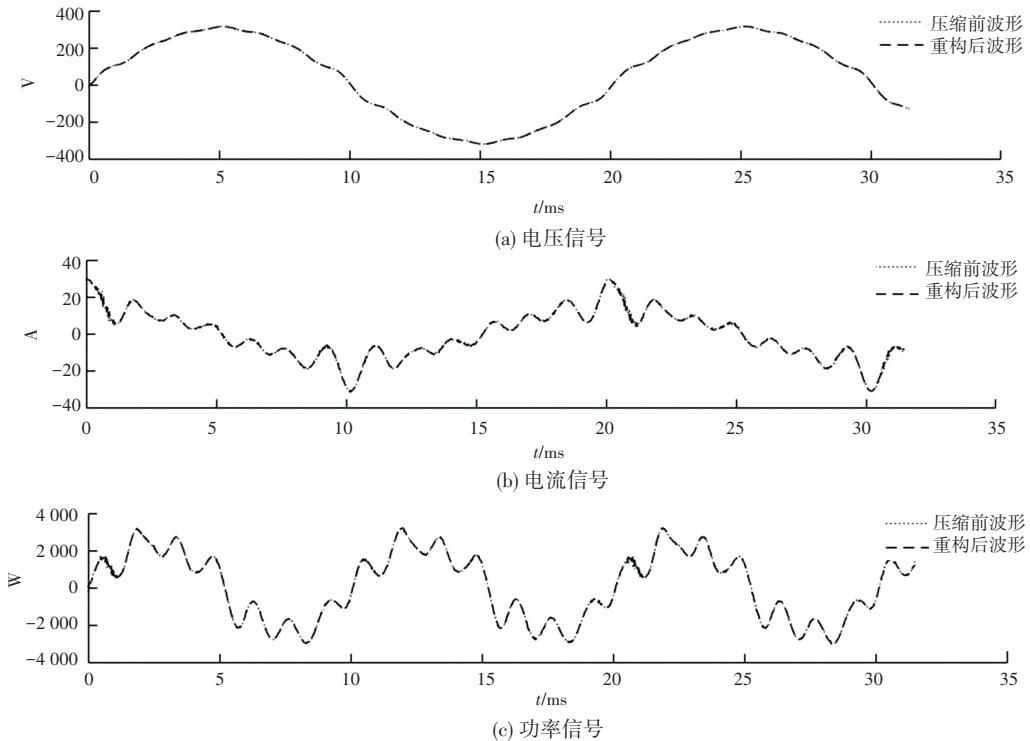


图 5 重构信号与原始信号波形对比

Fig. 5 Comparison between reconstructed signal and original signal waveform

3 硬件实现及验证

3.1 硬件设计框图

本文设计的硬件框架图如图 6 所示。在顶层模

块中, 共有两个子模块: 压缩感知模块和 LZW 模块, 分别用于实现压缩感知算法和 LZW 字典编码。

压缩感知模块主要由状态机模块、数据输入及存放模块和稀疏矩阵模块组成。状态机模块控制待

压缩数据的读写及计算,实现压缩感知的前端压缩采样。压缩感知模块输出的数据进入 LZW 模块中的缓冲模块,随后进行字典编码。LZW 模块主要由 8 个子模块组成,缓冲模块是一个先入先出(First In First Out, FIFO),用于压缩感知后的数据缓冲,并提供状态信号给状态机;状态机模块是整个 LZW 算法的核心控制部分,控制各个子模块的协调运行,决定

了压缩算法是否能够正常进行;字典前缀值更新模块负责更新用于存放前缀值的寄存器的值;字典索引模块负责根据输入生成唯一的哈希地址;字典模块负责将数据存入哈希地址;编码模块负责更新字典项编码的值;比较模块负责对输入值和字典读出的值进行比较;输出选择模块负责对输出的数据进行选择,产生输出使能信号。

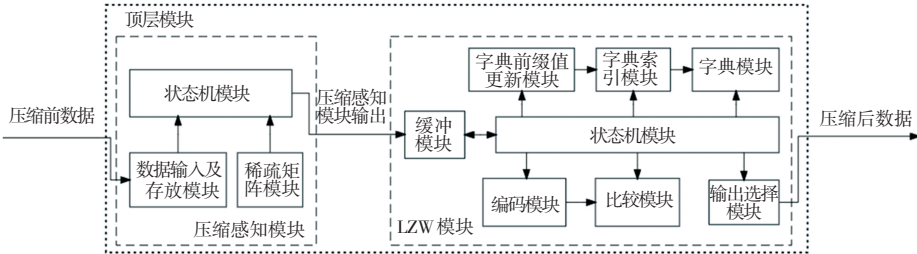


图 6 硬件设计框架图

Fig. 6 Block diagram of hardware design

3.2 功能验证

数字电路的功能验证基于统一验证方法(Universal Verification Methodology, UVM)搭建的验证平台对设计进行功能验证。所搭建的验证平台框架如图 7 所示。

设计(Design Under Test, DUT)产生的输出和参考模型

的输出,判断其是否相同。代理(agent):与环境类似,是一个容器类,用于将定序器、监视器和驱动器封装在一起,且在被动模式下(从代理)只能例化监视器,而在主动模式下(主代理)则需要同时例化这 3 个组件。

定序器(sequencer):产生并发送连续的激励事务(transaction)至驱动器的一侧。

驱动器(driver):用于通过 TLM(Transaction Level Modeling)端口从与其相连的定序器中获取事务,并对事务经过转化后通过接口(interface)发送至待测设计,以对待测设计进行时序激励。

在搭建的验证平台中,验证环境(软件)与待测设计通过接口连接。不同的组件之间通过 TLM 端口通信,组件之间的传输对象为事务,其派生自 uvm_sequence_item,由序列(sequence)产生并通过定序器发送至驱动器。

在利用各种组件类搭建好验证平台进行功能验证,验证结果如图 8 所示,其输入数据为数据集中随机选取的电压数据。在最后几个数据比较通过后,记分板输出数据比较完成的结果,表示电路正确完成了数据压缩算法,本次设计的功能达到了预期要求。

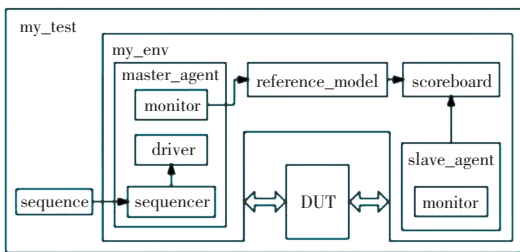


图 7 基于 UVM 的验证框架图

Fig. 7 Verification framework diagram based on UVM

验证环境组件继承自 UVM 组件类(uvm_component),各个组件的功能如下:

环境(environment):是一个容器类且是验证环境的顶层,用于实例化验证平台的各个组件。

参考模型(reference_model):是根据需求设计的能够得到正确结果的算法模型。

监视器(monitor):监测接口数据,并将其转化为事务。

记分板(scoreboard):比较从监视器采集的待测

```

# UVM_INFO top_bus_scoreboard.sv(108) @ 52610000: uvm_test_top_m_top_bus_env_m_top_bus_scoreboard [scb drv s mon data compared SUCCESS] data=0e0
# UVM_INFO top_bus_scoreboard.sv(108) @ 52630000: uvm_test_top_m_top_bus_env_m_top_bus_scoreboard [scb drv s mon data compared SUCCESS] data=12e
# UVM_INFO top_bus_scoreboard.sv(108) @ 52650000: uvm_test_top_m_top_bus_env_m_top_bus_scoreboard [scb drv s mon data compared SUCCESS] data=1af
# UVM_INFO top_bus_scoreboard.sv(108) @ 52670000: uvm_test_top_m_top_bus_env_m_top_bus_scoreboard [scb drv s mon data compared SUCCESS] data=0c6
# UVM_INFO top_bus_scoreboard.sv(108) @ 52690000: uvm_test_top_m_top_bus_env_m_top_bus_scoreboard [scb drv s mon data compared SUCCESS] data=0c1
# UVM_INFO top_bus_scoreboard.sv(108) @ 52710000: uvm_test_top_m_top_bus_env_m_top_bus_scoreboard [scb drv s mon data compared SUCCESS] data=id3
# UVM_INFO top_bus_scoreboard.sv(122) @ 229110000: uvm_test_top_m_top_bus_env_m_top_bus_scoreboard [SCB] ALL DATA WERE COMPARED AND SUCCEEDED

```

图 8 UVM 验证平台测试结果

Fig. 8 Test results of UVM verification platform

3.3 电路时序与逻辑综合

设计的硬件电路正常工作时的时序图如图 9 所示。当指示压缩开始的 start 信号有效时,待压缩数据 din 及其有效信号 din_vld 开始输入到压缩电路中,经过压缩感知模块的计算,输出压缩数据 cs_out

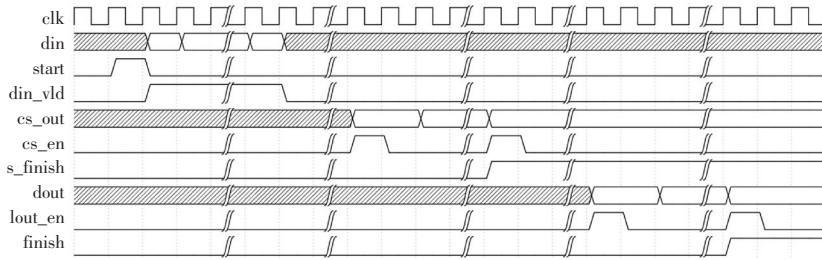


图 9 硬件电路的时序图

Fig. 9 Waveform diagram of hardware circuit

使用 FPGA 器件所实现的硬件电路的最高工作频率能够达到 200 MHz,该频率下完成整个数据压缩计算过程的总延时约为 16.11 μs ,共花费 3 221 个时钟周期。其中,数据输入共花费 256 个时钟周期,CS 模块的计算共花费 2 051 个时钟周期,LZW 模块的计算共花费 914 个时钟周期,每个时钟周期为 5.0 ns。

4 结束语

本文提出了电力信号数据压缩的算法。首先,探讨了压缩感知算法的优越性及硬件实现的可能性,并结合 LZW 字典编码,对实际的电力信号进行了数据压缩;其次,从硬件实现的角度讨论了各种观测矩阵的差异,在此基础上选择了稀疏随机矩阵作为观测矩阵,并结合状态机思想实现算法的优化,提

及其使能信号 cs_en。压缩感知模块计算完成后拉高 cs_finish 信号,同时 LZW 模块开始工作,其输出的压缩数据 dout 即为最终的压缩结果。当 LZW 模块完成计算后,输出 finish 信号,表示压缩电路工作完成。

高了压缩速率;再者,对压缩采样后的数据进行 LZW 编码,在重构精度不变的情况下提高了压缩率。实验表明,本文的算法在 FPGA 器件上能够达到与软件压缩一样的结果,时钟频率可达 200 MHz,整个数据压缩过程的总延时约为 16.11 μs ,重构误差约为 4.83%,平均压缩率可达 36.83%,相比只使用压缩感知提高了约 13.17%的压缩率。

参考文献

- [1] WANG G, NIU M Y, FU F W. Deterministic constructions of compressed sensing matrices based on codes[J]. *Cryptography and Communications*, 2019, 11(4): 759-775.
- [2] 叶彦斌. 基于压缩感知的脑电数据压缩及 FPGA 实现[D]. 广州:华南理工大学, 2019.
- [3] 王强, 张培林, 王怀光, 等. 压缩感知中测量矩阵构造综述[J]. *计算机应用*, 2017, 37(1): 188-196.
- [4] 周钦坤, 岳建平, 杨恒, 等. 机载 LiDAR 数据中电力线的自动提取与重建[J]. *测绘通报*, 2020, 523(10): 26-30, 37.
- [5] 王柯, 易琳, 钱金菊, 等. 一种高效的机载激光雷达点云电力线塔全自动分割方法[J]. *地理信息世界*, 2020, 27(4): 115-118, 122.
- [6] 梁安祺, 马洪超, 蔡湛. 基于 SVM 的机载 LiDAR 数据电力线提取方法[J]. *激光杂志*, 2019, 40(2): 29-34.
- [7] 吴建军, 陈灵, 李磊, 等. 机载 LiDAR 点云中电力线的提取和重建研究[J]. *激光技术*, 2019, 43(4): 64-69.
- [8] 温玉维, 邓长勇, 曾德培, 等. 基于相对密度聚类的激光点云电力线提取方法[J]. *测绘与空间地理信息*, 2022, 45(8): 262-264.
- [9] 程宇航, 白征东, 辛浩浩, 等. 机载激光点云中电力线的自动提取与重建[J]. *测绘工程*, 2021, 30(5): 58-63.
- [10] 朱依民, 田林亚, 毕继鑫, 等. 基于无人机机载 LiDAR 的电力线点云提取与重建[J]. *激光技术*, 2021, 45(5): 554-560.
- [11] 蔡岐彬, 许小龙. 随机森林算法在机载 LiDAR 电力线自动提取中的应用[J]. *电力信息与通信技术*, 2018, 16(7): 16-20.
- [12] 韩东升, 徐茂林, 金远航. 多源异构点云配准数据的滤波及精度分析[J]. *测绘科学技术学报*, 2020, 37(5): 503-508.
- [13] ZHANG W, QI J, WAN P, et al. An easy-to-use airborne LiDAR data filtering method based on cloth simulation[J]. *Remote Sensing*, 2016, 8(6): 501.
- [14] 武永彩. 基于 LiDAR 点云的电力线自适应密度聚类提取[J]. *工程勘察*, 2023, 51(5): 52-56.
- [15] CAO Y, WANG Y, XUE Y, et al. FEC: Fast euclidean clustering for point cloud segmentation[J]. *Drones*, 2022, 6(11): 325.
- [16] 宋晨洋, 王强, 高德涵, 等. 顾及地物邻域特征的 LiDAR 点云布料模拟滤波优化[J]. *激光与红外*, 2023, 53(1): 51-56.
- [17] 韩宇. 一种基于无人机载 LiDAR 点云的电力线安全距离判定算法[J]. *北京测绘*, 2018, 32(9): 1037-1040.
- [18] 沈小军, 秦川, 杜勇, 等. 复杂地形电力线机载激光雷达点云自动提取方法[J]. *同济大学学报(自然科学版)*, 2018, 46(7): 982-987.
- [19] 宋向荣. 基于机载 LiDAR 点云的电力线提取与三维重建[J]. *北京测绘*, 2023, 37(2): 254-259.
- [20] 麻卫峰, 王金亮, 王成, 等. 一种基于机载 LiDAR 点云电力线自动提取方法[J]. *测绘科学技术学报*, 2019, 36(1): 39.

(上接第 123 页)