

基于 FPGA 的战情信息压缩算法^①

苏伟朋¹, 郝永生¹, 邱国龙², 李跟臣², 李冬鹏²

¹(军械工程学院, 石家庄 050003)

²(中石油管道分公司, 廊坊 065000)

摘要: 战场信息中包含大量固定相关术语且快速有效传输, 对信息压缩有利于快速传输, 但现有的一些压缩算法针对战情报文压缩效果不是和理想. 文中介绍了 LZW 算法原理, 在 LZW 算法的基础上增加了战情动态词典、重复编码器进行算法改进. 在多项改进的基础上结合 FPGA 的优点用硬件实现, 获得了很好的效果.

关键词: 战情; LZW 算法; 动态词典; 重复编码; FPGA

Battlefield Intelligence Information Compression Algorithm Based on FPGA

SU Wei-Peng¹, HAO Yong-Sheng¹, QIU Guo-Long², LI Gen-Chen², LI Dong-Peng²

¹(Ordnance Engineering College, Shijiazhuang 050003, China)

²(PetroChina Pipeline Company, Langfang 065000, China)

Abstract: Fixed battlefield information contains a large number of related terms with fast and efficient transmission, compression is conducive to rapid transmission of information, but some of the existing compression algorithm for war intelligence and compression effect is not ideal. This paper introduces the principle of LZW algorithm, based on the LZW algorithm increases the situation dynamic dictionary, repetitive encoder algorithm is improved. In a number of improvement on the basis of combining the advantage of FPGA in hardware implementation, obtained the good effect.

Key words: pattern; LZW algorithm; dynamic dictionary; repeat codes; FPGA

战场信息中的内容有战场环境、气象、地理环境、还包括双方的后勤保障措施与能力、作战、策略与指挥条令、通信与情报能力. 所以战情信息并不仅仅为单纯的汉字或英文, 而是将中英文、数字、各种预定的军用符号联合使用, 而某些词出现的频率会非常高^[1]. 现有的压缩算法中, 对文本进行压缩采用的技术主要是 Huffman 编码、LZ 算法和 LZW 算法以及基于这些算法改进算法^[2]. 但是这些算法是针对西文编码或者通用字符设计的, 对战情报文压缩效果还并不是很理想.

数据量过大一方面会带来数据存储和传输的困难, 某些情况下又出于通信带宽和通信传输速度的限制, 会使得数据不经过压缩处理而直接传输所付出的代价大. 目前存储器容量的级数一直在增长, 虽然在一定程度上可以缓解数据存储的需求, 但现阶段解决问题的根本途径还是要研究更为有效的压缩算法. 相对来说

基于字典的 LZ 系列编码, 通过维护动态字典来实现对变长符号串编码, 适合压缩某些词出现频率很高的算法. LZW 算法在实时性、复杂度、压缩效果、存储面积和适用场合方面都有不错的表现. 文献[3]中结合以词汇为单位的处理模式的设计思想, 提出一种基于动态词典的英文文本压缩算法, 在 LZW 算法中加入一个常用单词词典来提高压缩效率. 文献[4]中以对已压缩过的数据进行历史分析为手段, 采用三个编码器对源数据序列分类编码. 若直接用 LZW 算法对信息进行压缩的话会造成用时长且压缩效果不甚理想. 基于此, 根据战情信息的特殊情况, 将以上二者优点结合, 设计一种针对战情的 LZW 改进压缩算法, 改进后的算法对战情信息压缩会有较强的适应性. FPGA 具有高集成度、低功耗、灵活性及并行运算的优点, 所以利用 FPGA 硬件实现 LZW 算法, 可以提高系统的实时压缩能力.

① 收稿时间:2013-07-24;收到修改稿时间:2013-09-03

1 LZW算法介绍

LZW 算法首先将字母表中的所有字符初始化到字典中, 一般如果是使用 8 位字符, 因而在输入任何数据之前先占用了字典的前 256 项, 即 0~255. 因为字典已经经过初始化, 所以下一个输入字符总能在字典中查到^[5].

LZW 的编码原理简述如下^[6]: 编码器逐个输入字符, 并累积形成一个字符串 C, 每输入 1 个字符就被串接在 C 后面, 然后再字典中查找 C; 只要在字典中找到 C, 该过程就继续, 直到在某一点, 添加下一个字符 x 使搜索不成功, 即字符串 C 在字典中, 而 Cx 不再字典中, 此时编码器就输出表示字符串 C 的字典的标识, 并在下一个可用的字典词条中存储字符串 Cx, 且将字符串 C 预置为 x. 这意味着虽然只加入了一个有效的字符 x 但存入了过多的字符这样即浪费了内存空间又增加了处理时间.

2 LZW改进算法

根据战情信息中某些词出现频率较高的特点, 初始化时在字典前 256 项先载入初始 ASCII 码, 再载入一个事先生成好了的常用战情信息单词词典. 有了这个常用战情词典, 编码时某个常用的单词就算是第一次遇到, 压缩算法也可以直接从战情词典中搜索到这个单词的索引值, 然后直接输出. 加入新生成的战情词典会导致压缩编码输出位宽的增加, 在压缩过程中信息内容如果过短, 可能会使得不如原算法的压缩效果. 所以常用战情词典必须把长度控制在一定范围之内, 在常用的战情信息长度上找到一个合适点, 尽可能提高压缩效率.

传统的 LZW 算法的编码器在读入其输入短语时, 每次只读取一个字符, 并用这个字符去做相应的词典匹配搜索, 由于编码算法没有很强的分析能力, 使它本身不知道什么字符序列将来出现的概率比较大, 会具有一定的盲目特性. 这样导致有的字符串会重复出现很多次, 如果 LZW 编码表要想完全识别一个长度为 n 的字符序列, 至少需要该序列部分或全部重复出现 n 次才行, 这样会浪费时间且影响压缩效率. 当一个字符串重复出现两次的时候, 我们可以很容易把它识别出来^[7]. 基于此, 在缓冲区增加一个重复编码器 RC(Repeat Corde)来直接处理重复的内容. RC 编码器先对输入流中重复的数据进行编码, 剩下的则由 LZW

编码器进行编码. 最后 LZW 编码器的编码表将 RC 编码器以及 LZW 编码器的编码按次序统一整合起来.

如图 1 所示的历史表是一个固定长度的队列缓冲区, 它存储最近的一段已编码的输入数据. 数据流先输入到一个超前缓冲区, 超前缓冲区是一个固定长度的输入数据缓冲区, 它存储的是即将进行编码的一小段数据. RC 编码器在串表中对超前缓冲区的最大匹配串进行搜索, 若匹配串的长度满足 RC 编码器编码条件, 那么输出匹配串编码, 同时把刚编码的数据加入到历史表中, 然后超前缓冲区重新读入数据.

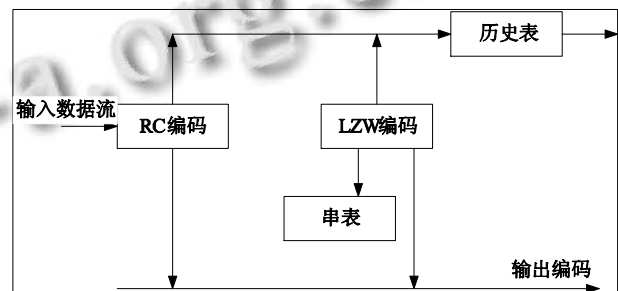


图 1 改进算法结构

RC 编码是在缓冲区增加超前缓冲备份的过程, 历史表是一个先入先出且长度固定的数据队列. 它记录的是输入数据流最新的一段局部历史, 其随着输入流的变化自动更新, 根据上文可以自我慢慢适应下文, 直至全文的过程. 而在初始化时根据经验事先增加常用战情词典, 增加固定的词典选项, 当压缩时第一次遇到词典中某个词的时候, 可以直接输出索引值. RC 编码则是根据历史表缓存的内容, 第二次遇到某个词时可直接处理重复内容. 它们在压缩时有着异曲同工之妙. 所以将这两处改进进行结合, 会对数据压缩率有着极大改善.

所以改进后的算法与原 LZW 算法相比增加了战情词典和 RC 编码, 原 LZW 算法是根据通用文本设计的, 根据具体问题具体分析的原则, 设计的 LZW 改进算法对战情信息有良好的压缩效果.

3 改进算法的FPGA硬件设计

FPGA 有速度快, 功耗低, 通用性好, 适应性强等特点. 使用 FPGA 可以非常大的减小硬件规模, 降低设计成本, 缩短设计周期, 提高系统的可靠性、灵活性和保密性. 同时 FPGA 逻辑功能强大、运行速度快、片内 RAM 资源丰富, 也非常有利于实现 LZW 压

缩算法^[8]。根据以上算法描述, FPGA 实现实时压缩内部模块连接图如图 2 所示。其中最核心的是图 3 所示的比较字典内容元件以及数据实时压缩总模块。根据需要将 FPGA 初始化: 设置一个大小为 1024×18 的字符串表, 串表的高 10 位存放字符串的前缀(C), 低 8 位存放当前输入的码字(x), 串表的前 256 项初始化为 0~255, 然后再存放已定义的常用战情短语 64 项, 然后在后 704 项初始化为 0。字典存储器需要存放字典项的三部分内容: 字典项编码、前缀码、当前码。当字典存满时, 重新初始化字典会占用较长时间, 这样会影响整体压缩速率, 所以设计采用两个同样的字典存储器交替进行工作, 存储器 1 存满时自动转入存储器 2, 然后存储器 1 进行初始化。系统在工作速率上有所提高。

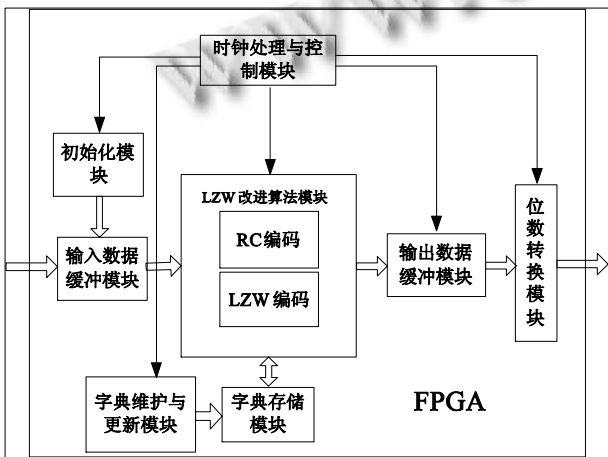


图 2 FPGA 内部模块连接关系图

3.1 主要模块工作原理

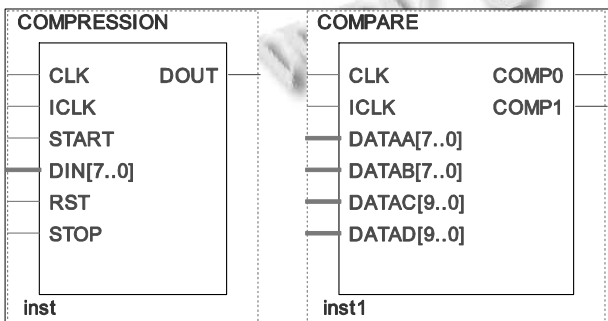


图 3 比较字典内容元件以及数据实时压缩模块原理图

其中:

DATAA[7..0]和 DATAB[7..0]是当前码 8 位数据输

入端;

DATAC[9..0]和 DATAD[9..0]是前缀码 10 位数据输入端;

CLK: 是主频输入;

ICLK: 输入数据频率;

START: 控制压缩开始信号;

DIN: 数据输入总线;

RST: 是复位信号;

STOP: 控制压缩结束信号;

DOUT: 数据输出

3.1.1 RC 编码实现

先经由 RC 编码器在输入短语时, 读入短语中各字符 X, C 和 X 生成一个索引, 与字典中的词条数进行匹配。当匹配成功, 即 COMP0=1 时, 表示符合 RC 编码器编码, 此时 COMP1 无效。输出编码格式: RC 标志+字符+重复长度, 然后直接输出编码。如果匹配不成功, 即 COMP0=0, COMP1 有效, 即转入 LZW 编码。

3.1.2 LZW 编码实现

LZW 编码的核心是 COMPRESSION 模块的实现, 实现过程可分为初始化状态、读第一个字符状态、读下一个字符状态、生成字符串状态、字符串比较状态和压缩数据输出状态。在初始化状态中首先将字符串表和程序中定义的寄存器初始化, 然后转入读第一字符状态。如果在读第一字符状态接收到的空信号有效时, 则表示此时没有数据读入, 转到初始化状态, 当读入空信号无效时, 此时读入第一个字符串, 并将该字符赋值给前缀码 C, 然后转到读下一个字符状态; 当读下一个字符状态的时候, 若空信号有效, 那么就在此状态循环一直循环, 直到空信号有效。如果空信号有效, 则读入一个字符到当前码 X, 然后再转到生成字符串状态; 在生成字符串状态将前缀码 C 左移 8 位, 再与当前码 X 相加, 就可以得到需要比较的当前字符串, 然后进入字符串比较状态。在字符串比较状态将 C+X 与串表中已经定义好的字符串比较, 就可以得到比较结果, 然后转到压缩数据输出状态。在数据输出状态, 根据比较结果(COMP)的值来决定怎么进行状态转移。当 COMP1=0 时, 表示 C+X 不在串表中, 则将 C+X 加入串表, 输出数据 C, X 的值赋给 C, 然后转到读下一字符状态。当 COMP1=1 时, 表示 C+X 在串表中, 将 C+X 的地址赋给 C, 然后再转到读下一字符状态进行工作。

3.2 数据压缩时序仿真及其分析

图 4 经过时序性能分析,系统工作的最高频率为 52.9MHz. 本仿真设为 20MHz 的主频, FPGA 对输入数据块进行压缩, 输入数据频率设为 1MHz, 这样压缩 1k 的数据块约耗 0.5ms, 如果对输入数据进行连续压缩, 输入数据频率可达 20kHz. 压缩速度比较理想. 同

时可以看出, 在压缩的开始时刻, 数据相关性不强的情况下, 压缩效果会不明显; 当输入数据相关性高时, 比如存在大量相同或相似的数据时, 输入多个数据, 输出一个码字, 数据压缩的效果会比较好. 数据的关联程度越大, 压缩效果越理想.

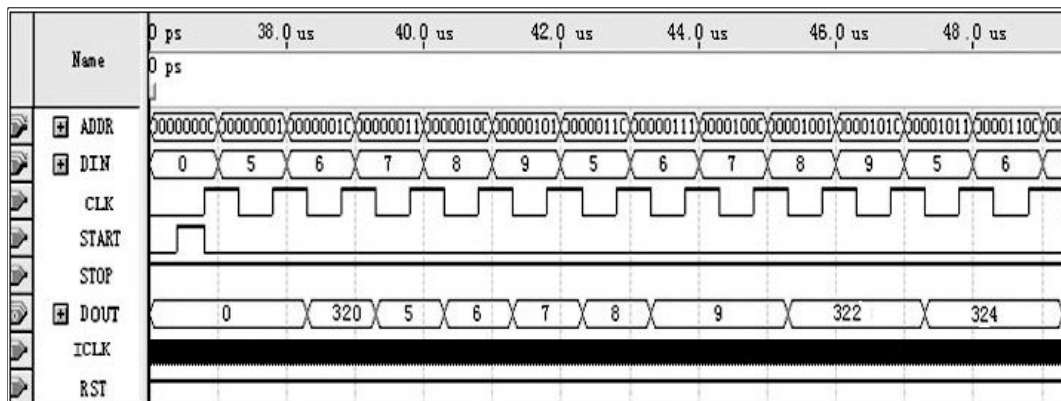


图 4 数据压缩时序仿真图

4 结语

战情信息传送在实时性以及有效性上有着极高的要求, 针对战情信息的特点, 参照以往优秀的压缩算法, 对 LZW 算法进行的改进, 相对于别的算法更适合战场信息的压缩. 结合 FPGA 的优良特性, 在硬件上采用 FPGA 实现了 LZW 算法. 经过仿真验证, 压缩效果好, 并且压缩速度满足某些系统的实时要求, 系统所占逻辑资源较少、可移植性强、功能扩展容易, 在数据的存储和传输效率上得到有效的提高了, 在大容量数据实时采集及传输系统中有广阔的应用前景.

参考文献

- 1 汤志荔, 张安. 战场威胁估计理论与方法研究. 火力指挥与控制, 2011, 36(9): 1-4.
- 2 黄建华, 常守锋, 董晶晶. 面向北斗短报文的中文分词及压缩编码算法. 第三届中国卫星导航学术年会电子文集, 2012.
- 3 江力, 孙建伶. 一种基于动态词典的英文文本压缩算法. 江南大学学报(自然科学版), 2007, 6(4): 442-445.
- 4 朱巧明, 赵英英, 钱培德. 基于中文词编码的压缩算法 ZHCP 的实现. 小型微型计算机系统, 2003, 24(2): 306-308.
- 5 卿大杰. 基于动态词典的文本压缩研究[学位论文]. 北京: 北京邮电大学, 2006.
- 6 吴家安. 数据压缩技术及应用. 北京: 科学出版社, 2009.
- 7 Zhang SW, Liu CL, Li QZ. A research study on the spallation strength of LY12 aluminum under the precompression condition. Science China, March 2010, 55(3): 505-513.
- 8 华清远见培训中心. FPGA 应用开发入门与典型实例. 北京: 人民邮电出版社, 2008.
- 9 (上接第 43 页)
- 10 龚仁伟, 尹超, 鄢萍. 基于 MES 的车间制造过程动态质量管理体系研究. 现代制造工程, 2008, (6): 26-30.
- 11 陈洁, 孔庆华, 吴晶晶. 面向服务的纺织机械制造执行系统研究. 现代制造工程, 2010, (7): 13-17.
- 12 张洋洋, 陈进. 基于 RFID 的离散制造车间实时数据采集系统的设计与实现. 江南大学学报, 2013, 65(1): 54-58.
- 13 杨桂霞. 存储过程及触发器在 SQL Server 数据库开发中的应用. 工程技术, 2012, (1): 66-71.