基于不同排序方法的快速霍夫曼编码硬件实现

李宜珂 王 旃
(浙江大学电气工程学院 杭州 310007)

摘 要 针对软件霍夫曼静态编码计算量大,而动态霍夫曼编码使得解码器同样复杂的缺点,提出了一种准动态霍夫曼硬件编码器。该编码器每次对一组数据序列进行静态编码,然后将编码并行输出,从而使得编码器具有较高的编码速度,而其延迟时间仅为一次编码过程的总时间。首先,为了充分利用硬件并行特性,分别使用动态排序和静态排序两种排序网络,以适应不同场合的编码需要。然后,使用数据流驱动的硬件二叉树构建和解析结构得到信源符号对应的霍夫曼编码。最后,将储存在FIFO中的输入数据查表并输出。设计结果表明,当使用Nexys4 DDR平台时,该编码器可以工作于100MHz以上的频率,同时具有吞吐高、延迟低、编码效率高和译码器简单的特性。

关键词 霍夫曼编码,硬件排序,硬件二叉树,现场可编程门阵列,先入先出
中图法分类号 TP302 **文献标识码** A

Hardware Implementation of Fast Huffman Coding Based on Different Sorting Methods

LI Yi-ke WANG Zhan
(College of Electrical Engineering, Zhejiang University, Hangzhou 310007, China)

Abstract Aiming at the drawback of static software Huffman coding which has large computational complexity and the problem of dynamic Huffman coding that makes the decoder more complicated, a quasi-dynamic hardware Huffman encoder was designed. This encoder encodes an array of data statically, then outputs the result parallelly which guarantees that it has a high encoding speed, and its dalay time is only the total time of an encoding process. First, for taking advantage of parallelism of hardware, static and dynamic sorting network are both implemented to meet different encoding demands. Then, a hardware binary tree building unit and a resolving unit driven by data flow are implemented to get the final Huffman codes. Finally, the Huffman codes are outputted based on data stored in FIFO. The design results illustrate that, this quasi-dynamic based on Nexys4 platform shows advantages of high working frequency (over 100MHz), high throughput, low latency and can simplify the design of decoder.

Keywords Huffman coding, Hardware sorting, Hardware binary tree, Field-programmable gate array(FPGA), First in first out(FIFO)

1 引言

霍夫曼编码是由David. A. Huffman在1952年发明的无损数据压缩编码。得益于其高效性,霍夫曼编码已经在计算机、数据加密和通信等领域中广泛采用。霍夫曼编码的实现一般是通过查找静态表[1]或动态编码的方式[2],前者的灵活性较差,而后者由于数据解码时同样是动态的,使得接收端的软硬件构成同样复杂[3]。霍夫曼编码的核心过程是排序,而排序对于通用CPU来说又是一件相对复杂且频繁的运算。对于快速和实时性比较高的霍夫曼编码,软件排序过程已经难以满足。相比之下,由硬件实现的排序电路在保证运算速度的同时,使得功耗相对降低,且使得通用CPU能够解放出来去处理更复杂的问题,不失为一种较好的策略[4]。Nexys4 DDR是Digilent公司出品的数字电路开发平台,搭载了Xilinx公司生产的Artix7大容量FPGA芯片xc7a100t,能够满足各种从组合逻辑到嵌入式CPU的设计需求。

2 霍夫曼编码原理

2.1 唯一可译变长码与即时码

对于信源符号集 $S=\{s_1,s_2,\cdots,s_n\}$,有对应的编码 $C=\{c_1,c_2,\cdots,c_n\}$,其中 $c_i(c_i\in C)$ 由 $X=\{x_1,x_2,\cdots,x_n\}$ 组成,则称 C 是信源符号 S 的 p 元编码。同时若满足 C 内各元素码长不全相同,且由 C 中元素组成的任意长度的序列只能被唯一地译码为对应的信源符号序列,则称 C 是 S 的唯一可译变长码。对于一组唯一可译变长码 C,若对于其中任意的 $c_i=\{x_{i1},x_{i2},\cdots,x_{iq}\}$ 都不是其他元素的前缀时,则称 C 是 S 的即时码。本文中讨论的霍夫曼编码则是一种变长即时码,且其平均码长也是所有唯一可译码中最短的,属于最佳码的一种。

2.2 静态霍夫曼编码

对于静态霍夫曼编码,一种简单的办法是构造霍夫曼树。由于数字电路的特性,本文中讨论的编码均为二元。下面简要介绍编码过程。

将上述信源符号集 S 中的信源符号按照概率分布排序,使得 $p_1\geqslant p_2\geqslant\cdots\geqslant p_n$,将概率最小的两个信源符号分别编码为0和1,并将这两个信源符号合并为一个新的信源符号,新信源符号的概率为合并前信源符号的概率之和。此时信源符号集 S 中只剩下 $n-1$ 个信源符号,将其重新按照概率分布排序,并重复对概率最小的两个信源符号的编码与合并过程,直到当 S 中只剩下两个符号时,分别用0和1对其进行编码即

李宜珂(1996-),男,主要研究方向为片上系统与嵌入式CPU,E-mail:yikeli@zju.edu.cn。

可。最后由各级合并过程中得到的编码向前反推，即可得到原始信源符号集 S 中所有信源符号的二元霍夫曼编码。

3　功能描述与硬件结构

3.1　功能描述

本文针对静态查表法和动态霍夫曼编码的主要不足，对于拥有 10 个码元符号的 BCD 编码(BCD)的霍夫曼编码做了如下改进：采用准动态的编码过程，即每次编码读入一定的数据量，使用 FIFO 记录数据序列，针对当前批次的数据进行编码并输出，并在输出的过程中同时接收并编码下一批次的数据，提高了硬件利用率。对于具体的编码过程，分别采用了动态排序和静态排序两种排序方法[5]。前者在数据输入时利用动态排序网络进行排序，后者则在数据排序完成后使用组合逻辑电路实现的改进 odd-even 排序算法进行排序。在得到当前批次数据的概率分布后，利用组合逻辑进行霍夫曼树的构建以及编码，最后利用 FIFO 输出的数据直接映射到对应的霍夫曼编码并输出。

3.2　硬件结构

3.2.1　总体结构

针对上述编码过程，系统的硬件结构如图 1 所示。

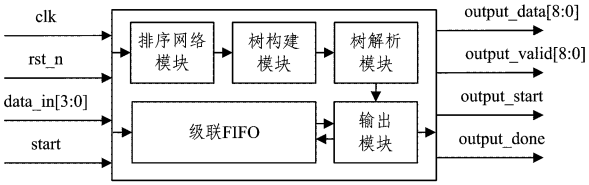

图 1　编码器顶层结构

排序网络分为动态排序和静态排序，其功能为统计各 BCD 的频数，并按降序排序，两种方案各有所长。树构建模块每次取最小的两个频数，求和并插入序列，记录每次的操作数。编码模块确定节点之间的父子关系，从根节点开始对子节点递归编码。FIFO 用于记录输入数据序列，以提供电路输入编码到输出的 pipeline 工作流程。输出模块将 FIFO 的输出译码为对应的哈夫曼编码，以并行方式输出。

3.2.2　排序网络

(1)动态排序

动态排序每读入一个新的 BCD，调整一次顺序[6-7]，回避了传统插入排序的移位和插入过程，以巧妙的元素交换方式提高了电路工作频率，同时通过硬件复用减小资源消耗。该方案用 10 个桶(bucket)存储各个 BCD 和对应的频数。初始化后，每个 BCD 的频数为 0。将每个桶按频数排序，使其满足上方的桶大于或等于下方。将当前输入的 BCD 对应的桶定义为目标桶。分析输入过程可得到如下推论。

推论 1　不必考虑在目标桶下方的桶，即目标桶频数增加，只有可能使目标桶与上方的桶互换位置，对其下方的桶的顺序没有影响。

推论 2　不必考虑比目标桶频数大的桶，即目标桶频数增加后，至多与这些桶频数相同，无需调整它们的顺序。

图 2 给出了一次数据输入中不需交换数据顺序的过程，图 2(b)为一次需要交换多组数据顺序的过程。对于图 2，由比较器得知，当前输入 BCD 所在桶的位置后，取出其频数并利用比较器进行比较，根据推论 1 可直接增加其对应频数。对于图 3，当前输入为 9，目标桶为 5 号桶。5 号桶上方的 3，4 号桶的当前频数与 5 号相同。处理步骤如下：通过比较模块，选择 3，4，5 号中最上方的桶，即 3 号桶；将 3 号桶(而不是 5 号桶)的频数增加 1；将被增加的桶的频数赋值给实际的目标桶，即 5 号桶；将当前输入的 BCD 赋值给被增加的桶，即 3 号桶。

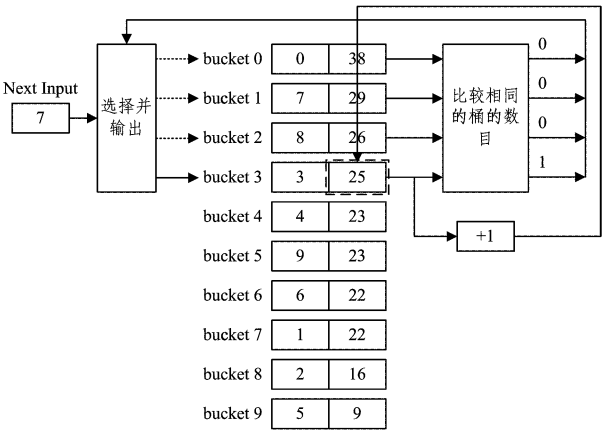

图 2　无交换数据通路

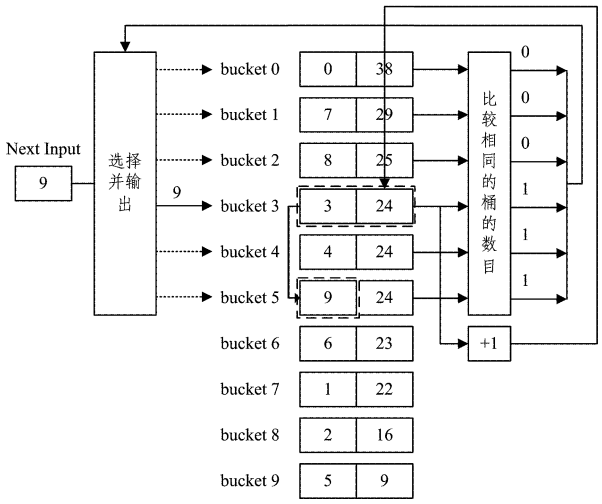

图 3　有交换数据通路

可以看到，只要当前状态满足排序关系，读入新的输入后依然可以保证排序关系。那么，只要最开始满足排序关系，最终的结果就满足排序关系。而在初始状态，所有桶的频数都为 0，满足上方的桶大于或等于下方的桶。

(2)改进的 odd-even 静态排序

常用的 odd-even 排序只能处理 2^n 种输入的情况[8]，针对 BCD 只有 10 种输入的情况则需要做出改进。首先定义一个如图 4 所示的交换单元，其输入为两个无符号数，经比较后按大小顺序输出。

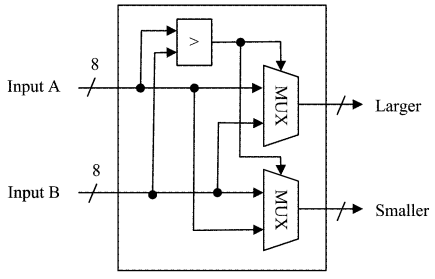

图 4　交换单元

整个静态排序网络的结构如图 5 所示。为便于表示，水

平长线代表一条数据通路，垂直箭头表示交换单元，且箭头所指的方向为 Larger 端口。其核心过程是先通过 4 级比较得到 10 个数的最大值和最小值(Min Max Phase)，之后中间的 8 个数通过一般的 odd-even 网络进行排序即可(Odd Even Phase)。可以看出，每一条数据通路至多经过 5 个交换，组合逻辑延时比普通的插入排序等方法大大减少。

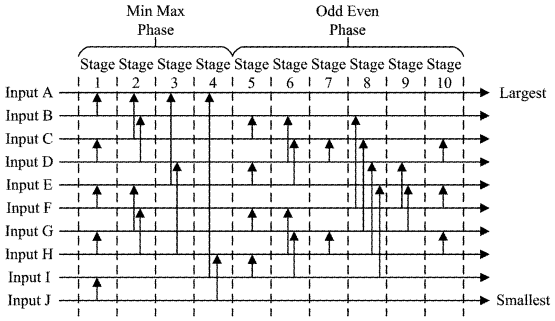

图 5 静态排序网络结构

Min Max Phase 的 4 个 Stage 分别可以得到如下结果。

Stage 1：

$A_1 \geqslant B_1, C_1 \geqslant D_1$

Stage 2：

$A_2 = \max\{A,B,C,D\}, D_2 = \min\{A,B,C,D\}$

$E_2 = \max\{E,F,G,H\}, H_2 = \min\{E,F,G,H\}$

Stage 3：

$A_3 = \max\{A,B,C,D,E,F,G,H\}$

$H_3 = \max\{A,B,C,D,E,F,G,H\}$

Stage 4：

$A_4 = \max\{A,B,C,D,E,F,G,H,I,J\}$

$J_4 = \max\{A,B,C,D,E,F,G,H,I,J\}$

分析上述两种排序网络可知，其均由组合逻辑构成，具有数据流驱动的特点。其中动态排序网络可以方便地增加排序元素，静态排序网络在数据量较小时可获得更小的排序延时。

3.2.3 树构建

通过改进原始的动态排序网络可以将其用于霍夫曼树的构建，其中共有 9 次合并过程。改进的主要内容在于为每个桶增加一个 4 位的数据位宽 child，目的是记录霍夫曼编码过程中概率合并的父子节点关系，从而方便编码模块解析树构建输出从而得到最终的编码。图 6 给出了一次典型的树构建过程。

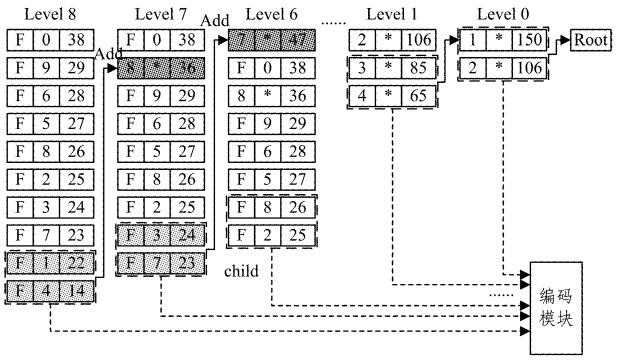

图 6 树构建典型过程

原始输入的 child 域赋值为 0xf。取最小的两个桶求和后插入上方的合适位置，使其满足排序关系，并将 child 中的数据传递给下一级。依次递归，直至剩下一个桶，该桶为哈夫曼树的根节点。最后将每一级最下方的两个桶的数据作为编码模块的输入即可。其中为了满足编码方差最小的特性，若存在与待插入数相同的桶，则必须优先插入在所有相同的桶的最上方。其硬件结构中的某两级如图 7 所示，其中的比较器单元如图 8 所示。

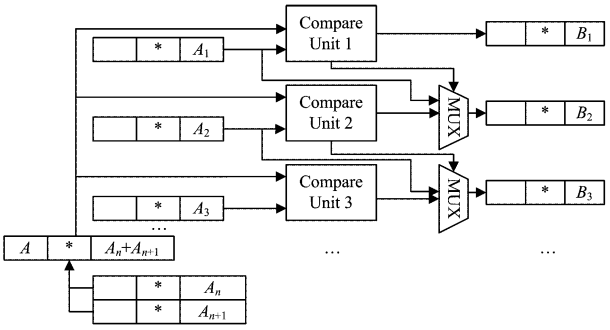

图 7 树构建硬件结构

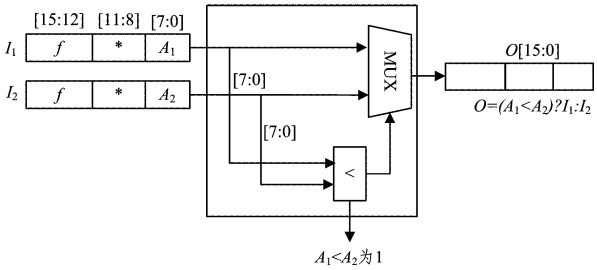

图 8 比较器单元

3.2.4 编码

编码部分通过一个简单的组合逻辑映射实现。由于对于 n 码元符号进行编码的霍夫曼树的最大高度为 $n-1$，因此该模块仅需 9 位数据位即可表示所有编码。同时，为了输出方便和降低译码电路的复杂度，对于每个位霍夫曼编码使用一位数据位记录其有效性，从而使得输出部分的电路可在串行和并行输出之间灵活选择。图 9 给出了一次典型编码的具体过程，其输入为树构建过程每一级最下方的两个桶的输出。从根节点开始，根节点的编码为空，其两个子节点的较大的编码为 1，较小的编码为 0，即“150”节点的编码为“1”，“106”节点的编码为“0”。递归对子节点的编码。例如，“150”的子节点为“85”节点和“65”节点，因此“85”和“65”节点将继承“150”节点的编码，然后“85”增补 1，“65”增补 0。依次类推，通过不断地“继承”、“增补”操作，可以得到所有叶子节点的哈夫曼编码。

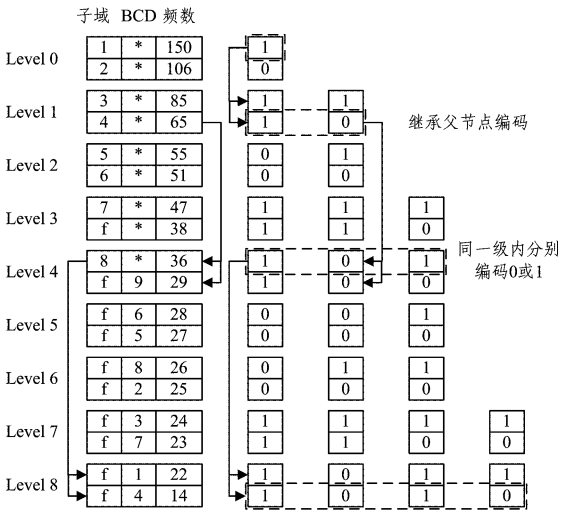

图 9 编码典型过程

3.2.5　FIFO 及输出

为了尽可能地提高编码速度并充分利用 Artix7 FPGA 芯片的资源，编码器中使用了 FPGA 芯片内建 DRAM 实例化的 FIFO。其中 FIFO 在进行当前批次数据编码的同时输出上次的输入数据，并结合上次编码结果查表输出，从而使得整个编码器以流水线的方式工作，提高了工作效率。另外，在编码器的内建 FIFO 前级串联了一个由 D 触发器构成的 10 位 FIFO，其作用是在每次输出序列编码之前顺次输出 BCD 的霍夫曼编码，从而使简化译码端结构得到相应简化，其结构如图 10 所示。

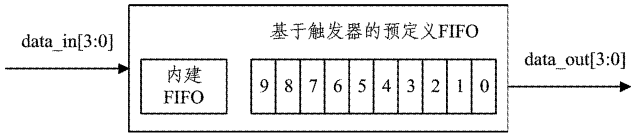

图 10　级联 FIFO

4　资源消耗及性能

在完成整个系统的设计后，使用 Vivado 2016.2 选择器件 Artix7 xc7a100tcsg324-1 分别对基于两种排序策略的编码器进行 Synthesis 后的资源消耗，分别如表 1 和表 2 所列。

表 1　动态排序网络编码器的资源消耗

Resource	Utilization	Available	Utilization/%
LUT	2641	63400	4.17
LUTRAM	3	19000	0.02
FF	204	126800	0.16
BRAM	0.50	135	0.37
IO	27	210	12.86
BUFG	1	32	3.13

表 2　改进的 odd-even 静态排序编码器的资源消耗

Resource	Utilization	Available	Utilization/%
LUT	2841	63400	4.44
LUTRAM	3	19000	0.02
FF	164	126800	0.13
BRAM	0.50	135	0.37
IO	27	210	12.86
BUFG	1	32	3.13

在完成 Implementation 后进行 Post-Implementation Timing Simulation，采用随机数据输入，分别得到基于两种排序策略的编码器的波形，如图 11 和图 12 所示。其中基于动态排序网络和改进的 odd-even 静态排序网络的编码器的最高工作频率为 100MHz 和 161MHz，编码速度分别为 6 个和 9 个时钟周期。在此基础上继续分析系统，发现 FPGA 内置的由 DRAM 构成的 FIFO 是系统速度的瓶颈，通过改用 D 触发器的方式后系统的性能得到了一定程度的提升。以静态排序网络为例，其工作频率能够达到 240MHz 左右。

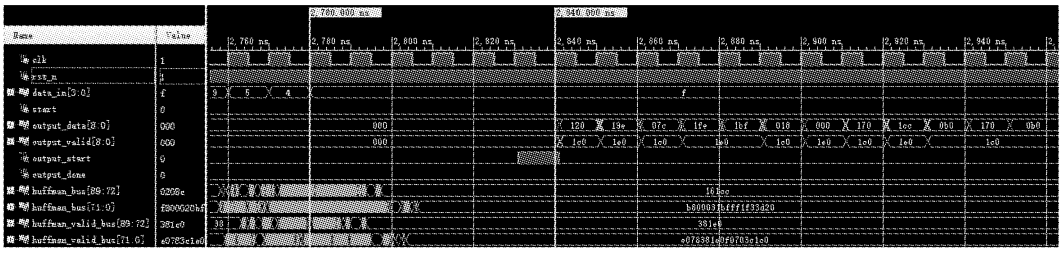

图 11　动态排序网络编码器的仿真结果

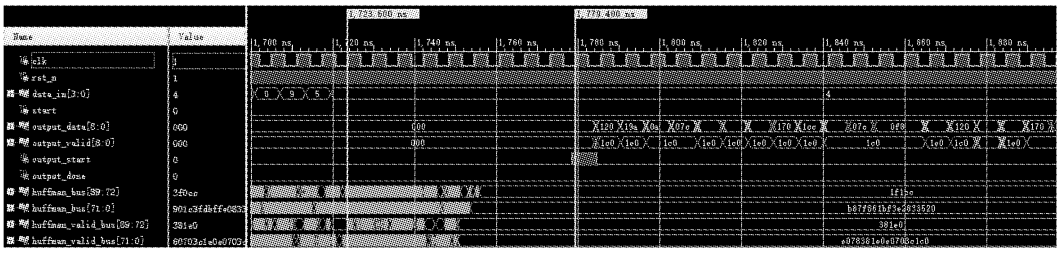

图 12　改进的 odd-even 静态排序编码器的仿真结果

从上述仿真结果可以看出，基于两种排序网络的霍夫曼编码器均能够实现高效编码，但是采用改进的 odd-even 排序网络的编码器的工作频率高于采用动态排序网络的编码器。考虑到信源符号集的可变性，改进的 odd-even 排序网络需要通过扩展的方式来实现排序，这无疑会大大增加硬件的资源消耗。相比而言，动态排序网络虽然速度略慢，但由于其扩展性较强，在信源符号集较大时可以使用。

另外，本文实现的霍夫曼编码器虽然采用了并行输出，提高了编码器工作频率，但输出线宽利用率不高，可以采用流水线归并输出的方式来提高输出线宽利用率[9]。

结束语　本文在静态霍夫曼编码器的基础上，提出了准动态霍夫曼编码器的思路。并针对编码数量的不同，分别使用两种硬件排序网络以及相应的硬件二叉树实现了该编码器。设计结果表明该编码器具有高速、高吞吐、低延迟和使得译码器简单的优点，具有一定的实际意义。但是，由于编码器的核心部分都是由数据流驱动的组合逻辑电路实现的，使得编码器的资源消耗偏大，有待进一步改进。另外考虑到数据的分布情况，针对分布较为均匀的数据，本文中实现的编码器并没有对其进行优化[10]，还需要进一步的改进。

参 考 文 献

[1] WANG G Y, ZHANG H S, LU M Y. Transformed HCT for parallel Huffman decoding [J]. International Journal of Circuit Theory and Application, 2015, 43(11): 1759-1774.

[2] LAWRENCE L, DANIEL H. A Fast Algorithm for Optimal Length-Limited Huffman Codes[J]. Journal of the ACM (JACM), 1990, 37(3): 464-473.

（下转第 509 页）

[4] 程莹，刘明波. 求解离散无功优化的非线性原—对偶内点算法[J]. 电力系统自动化，2001，25(9)：23-27.

[5] COELLO C A C. Evolutionary multi-objective optimization：a historical view of the field[J]. Computational Intelligence Magazine，IEEE，2006，1(1)：28-36.

[6] COELLO C A C，VAN VELDHUIZEN D A，LAMONT G B. Evolutionary algorithms for solving multi-objective problems[M]// New York：Kluwer Academic. 2002.

[7] IBA K. Reactive power optimization by genetic algorithm m[J]. IEEE Transactions on Power Systems，1994，9(2)：685-692.

[8] 张勇军，任震，钟红梅，等. 实时无功优化调度中的邻域搜索改进遗传算法[J]. 电网技术，2003，27(1)：22-25.

[9] 张勇军，任震，钟红梅，等. 基于灾变遗传算法的无功规划优化[J]. 电力系统自动化，2002，26(23)：29-32.

[10] 赵登福，周文华，张伏生，等. 遗传算法在无功优化应用中的改进[J]. 电网技术，1998，22(10)：34-36.

[11] 王志华，尹项根，李光熹. 伪并行遗传算法在无功优化中的应用[J]. 电网技术，2003，27(8)：33-35.

[12] 周晖，周任军，谈顺涛，等. 用于无功电压综合控制的改进粒子群优化算法[J]. 电网技术，2004，28(13)：45-49.

[13] 吴秀华，朴在林，徐静，等. 基于改进粒子群优化算法的电力系统无功电压综合控制[J]. 电力系统保护与控制，2007，35(21)：28-33.

[14] WU Q H，MA J T. Power system optimal reactive power dispatch using evolutionary programming[J]. IEEE Transactions on Power Systems，1995，10(3)：1243-1249.

[15] GOMES J R，SAAVEDRA O R. Optimal reactive power dispatch using evolutionary computation：extended algorithms[J]. IET Proceedings-Generation Transmission and Distribution，1999，146(6)：586-592.

[16] PRADHAN P M，PANDA G. Solving multi-objective problems using cat swarm optimization [J]. Expert Systems with Applications，2012，39(3)：2956-2964.

[17] 顾丹珍，徐瑞德. 一种地区电网多目标无功优化的新方法—改进模拟退火算法[J]. 电网技术，1998，22(1)：71-74.

[18] 王克文，张东岳. 电力系统无功优化算法综述[J]. 电测与仪表，2016，53(10)：73-76.

[19] WOLPERT D H，MACREADY W G. No free lunch theorems for optimization[J]. IEEE Transactions on Evolutionary Computation，1997，1(1)：67-82.

[20] 张蓓，郑宇军. 水波优化算法收敛性分析[J]. 计算机科学，2016，43(4)：41-44.

[21] 李章维，周晓根，张贵军. 一种动态自适应差分进化算法[J]. 计算机科学，2015，42(6A)：52-56，74

[22] MIRJALILI S. Moth-flame optimization algorithm：A novel nature-inspired heuristic paradigm[J]. Knowledge-Based Systems，2015，89：228-249.

[23] COELLO C A C，PULIDO G T，LECHUGA M S. Handling multiple objectives with particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation，2004，8(3)：256-279.

[24] ZHANG Q，ZHOU A，ZHAO S，et al. Multi-objective optimization test instances for the CEC 2009 special session and competition[OL]. http：//dces. essex. ac. uk/staff/qzhang/MOEAcompetition/cec09testproblem0904. pdf. pdf.

[25] SIERRA M R，COELLO C A C. Improving PSO-based multiobjective optimization using crowding，mutation and∈－dominance[M]// Evolutionary multi-criterion optimization. Springer Berlin Heidelberg，2005：505-519.

[26] VAN VELDHUIZEN D A，LAMONT G B. On measuring multiobjective evolutionary algorithm performance[C]// Proceedings of the 2000 Congress on Evolutionary Computation. IEEE，2000，1：204-211.

[27] ZITZLER E，DEB K，THIELE L. Comparison of multiobjective evolutionary algorithms：Empirical results [J]. Evolutionary computation，2000，8(2)：173-195.

[28] KACHORE P，PALANDURKAR M V. TTC and CBM Calculation of IEEE-30 Bus System[C]// International Conference on Emerging Trends in Engineering & Technology，Icetet 2009. Nagpur，Maharashtra，India，2009：539-542.

[29] EISSA M M，ABDEL-HAMEEDT S，GABBAR H. A novel approach for optimum number and location of FACTS devices on IEEE-30 bus grid using meta-heuristic basd Harmony Search[C]// IEEE International Conference on Smart Energy Grid Engineering. 2013：1-10.

[30] 陈前宇，陈维荣，戴朝华. 电力系统无功优化多目标处理与算法改进[J]. 电力系统保护与控制，2014，42(5)：129-135.

（上接第 479 页）

[3] LIU L Y，WANG J F，LEE，et al. Design and hardware architectures for dynamic Huffman coding [J]. IEE Proceedings：Computers and Digital Techniques，1995，142(6)：411-418.

[4] 张新超. 基于 FPGA 的 Huffman 编码并行实现及高速存储系统设计[D]. 西安：长安大学，2015.

[5] 郭诚欣，陈红，孙辉，等. 基于现代硬件的并行内存排序方法综述[J]. 计算机学报，2017，40(9)：2070-2092.

[6] BATCHER K E. Sorting networks and their applications [C]// AFIPS '68. 1968：307-314.

[7] 高劲松，孟令奎. 硬件排序器研究进展[J]. 电子计算机与外部设备，1997(5)：38-40.

[8] KOCH D，TORRESEN J. FPGASort：A high performance sorting architecture exploiting run-time reconfiguration on FPGAs for large problem sorting [C]// Proceedings of the 19th ACM/SIGDA International Symposium on Field Programmable Gate Arrays. 2011：45-54.

[9] 倪泽峰，王振华，谭毅华，等. 并行哈夫曼编码器的硬件设计与实现[J]. 微电子学与计算机，2002，19(10)：66-68.

[10] HE T，XLI T，LI Y J，et al. The Analysis and Improvement of Huffman Algorithm[C]// Proceedings of 2011 4th IEEE International Conference on Computer Science and Information Technology(ICCSIT 2011). 2011：529-532.

[11] TAEYEON L，JAEHONG P. Design and implementation for static Huffman encoding hardware with parallel shifting algorithm[C]// IEEE Nuclear Science Symposium Conference Record. 2003：1315-1318.