

# 基于FPGA的并行全比较排序算法

师廷伟 金长江

(中国电子科技集团公司第二十七研究所 河南郑州 450047)

**摘要:**基于FPGA硬件技术,以空间换时间的思路,提出了一种并行全比较的排序算法。该算法通过对数据的并行全比较,计算出每个数据在排序中的位置实现数据排序。该算法可在4个时钟周期内实现数字序列的排序,通过实验证明,实时性好,通用性强。

**关键词:**排序 并行 比较 FPGA

**中图分类号:**TP311.13

**文献标识码:**A

**文章编号:**1007-9416(2013)10-0126-02

**Abstract:**Based on the hardware technology FPGA, via the approach of time-for-space, the comparison sort algorithm in parallel is proposed in this paper. The data is compared in parallel via the algorithm and each data in the sort of location is calculated. The algorithm can be implemented in 4 clock cycles the sort sequence of the numbers. Through the experiments proved that the algorithm has good performance of real-time and versatility.

**Key Words:**sorting in parallel comparison FPGA

## 1 引言

排序是一种重要的数据运算,传统的排序方法主要靠软件串行方式实现,包括冒泡法、选择法、计数法等,这些算法大多采用循环比较,运算费时,实时性差。不能满足工程上越来越高的实时性要求。实时性排序在工程计算中的要求越来越迫切。

本文基于FPGA的硬件特点,提出了一种全新的并行全比较排序算法,又可被称为“以空间换时间”并行排序算法,可大幅提高数据处理的实时性。

## 2 并行全比较排序算法原理

各种传统排序算法,大多都是以两两之间顺序比较为基础。本文所提出的并行全比较实时排序算法是基于序列中任意两个数并行比较实现。由于全部数字同时进行比较处理,将会占用大量的处理空间,因此此算法也可称为“以空间换时间”排序算法。

一组数据,先进行两两之间的比较,每两个数比较都会得到一个比较结果。可以根据两数的大小定义输出排序结果0或1。对这些比较结果进行累加计算,即可得到该数在序列中的排序值。由于所有数的两两之间的比较都在硬件内同时进行,只需一个时钟的时间即可得到比较结果,再加上比较结果的和加等计算时间,几个时钟周期,就实现了数字序列的排序。

为了说明这种排序算法的排序过程,我们先看以下实例:

假设有一数组{20, 80, 40, 40, 60, 70}, 定为A<sub>0</sub>=20、A<sub>1</sub>=80、A<sub>2</sub>=40、A<sub>3</sub>=40、A<sub>4</sub>=60、A<sub>5</sub>=70,要求对该数组按从大到小的顺序排列。排序按以下过程进行:

### 2.1 输入比较器的选取

要进行数据之间的两两比较,首先要确定比较器。考虑到数组中有相同数据,并且后续还要进行和加计算。数据相同的排序按照原数据谁在前谁优先原则,要实现这种原则下的排序,在每个数据

与其它数据比较时,根据两个数字原来在数组中的顺序,比较器的类型要发生变化。比如A<sub>m</sub>和其它数比较时,在与数A<sub>n</sub>比较时:

如果m>n,则选用“≥”比较器;

如果m<n,则选用“>”比较器。

采用这种方式,既为后续的两两比较结果计算提供比较基础,又避免了数组中两个数字相同时的排序问题。

### 2.2 进行各数之间的比较

为了后续的计算,先作统一规定,如果数A<sub>m</sub>和数A<sub>n</sub>相比较,如果A<sub>m</sub>≥A<sub>n</sub>(或A<sub>m</sub>>A<sub>n</sub>),则比较结果Z=1;否则,则Z=0。

各数之间比较结果如下表:

### 2.3 比较结果累加计算

各数字两两比较后,对与一个数相比较的所有值的结果相加(COMP(X, Y)为比较器),以A<sub>0</sub>为例,其对应的比较累加和为:

由表可知计算结果,CHA<sub>0</sub>=0,在序列中从大到小的顺序为0(数列的最大值排序为第5位),也即是在序列中为最小值。

同样,A<sub>1</sub>、A<sub>2</sub>、A<sub>3</sub>、A<sub>4</sub>、A<sub>5</sub> 的比较值累加和分别为CH A<sub>1</sub>=5, CH A<sub>2</sub>=1, CH A<sub>3</sub>=2, CH A<sub>4</sub>=3, CH A<sub>5</sub>=4。那么,各数字在数组中的排序位置已经确定,各值从大到小排列为:A<sub>1</sub>、A<sub>5</sub>、A<sub>4</sub>、A<sub>3</sub>、A<sub>2</sub>、A<sub>0</sub>。

### 2.4 排序结果输出

各数字的排序位置确定后,要把排序结果输出。

## 3 算法基于verilog语言描述

排序算法在FPGA内进行,实现过程主要有以下几步,采用verilog语言来描述:

(1)第一个时钟周期,数据全比较程序,6个数据排序,输入数据为in<sub>0</sub>~in<sub>5</sub>:

//in<sub>0</sub>和其他数据进行比较

表 1 数据比较结果表

X	Y	A <sub>0</sub> (20)	A <sub>1</sub> (80)	A <sub>2</sub> (40)	A <sub>3</sub> (40)	A <sub>4</sub> (60)	A <sub>5</sub> (70)	累加值
A <sub>0</sub> (20)			0	0	0	0	0	0
A <sub>1</sub> (80)		1		1	1	1	1	5
A <sub>2</sub> (40)		1	0		0	0	0	1
A <sub>3</sub> (40)		1	0	1		0	0	2
A <sub>4</sub> (60)		1	0	1	1		0	3
A <sub>5</sub> (70)		1	0	1	1	1		4

$$\begin{aligned} CHA_0 = & COMP(A_0, A_1) + COMP(A_0, A_2) + COMP(A_0, A_3) + COMP(A_0, A_4) \\ & + COMP(A_0, A_5) \end{aligned}$$

图 1

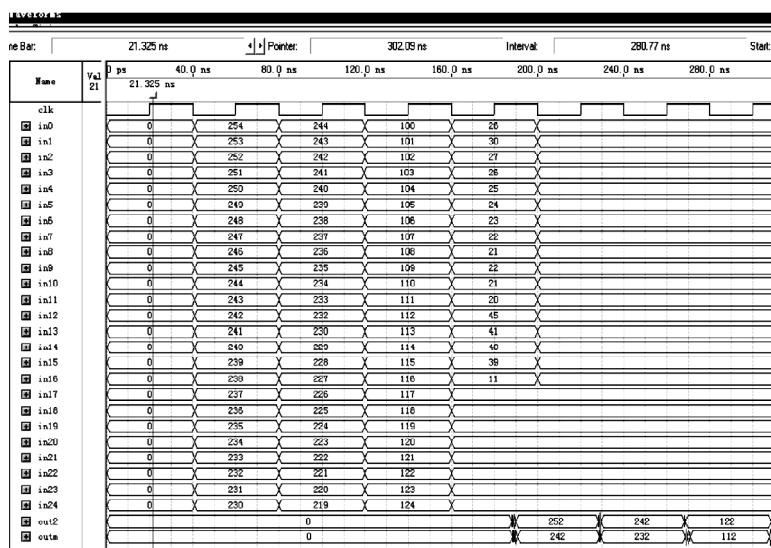


图1 排序设计仿真效果图

```
if(in0>=in1) a0<=0;   else  a0<=1;
if(in0>=in2) a1<=0;   else  a1<=1;
if(in0>=in3) a2<=0;   else  a2<=1;
.....
```

//in1和其他数据进行比较

```
if(in1>in0)  b0<=0;   else  b0<=1;
if(in1>=in2) b1<=0;   else  b1<=1;
if(in1>=in3) b2<=0;   else  b2<=1;
.....
```

//in3和其他数据进行比较

```
if(in2>in1) c0<=0;   else  c0<=1;
if(in2>in0) c1<=0;   else  c1<=1;
if(in2>=in3) c2<=0;   else  c2<=1;
.....
```

(2)第二个时钟周期,比较值累加,mid0、mid1、mid2等为比较累加值

```
mid0<=a0+a1+a2+a3+a4;
mid1<=b0+b1+b2+b3+b4;
mid2<=c0+c1+c2+c3+c4;
.....
```

(3)第三个时钟周期,把输入值赋给其对应的排序空间

```
chn[mid0]=in0;
chn[mid1]=in1;
chn[mid2]=in2;
.....
```

(4)第四个时钟周期,把排序结果输出

```
out 0<=chn[0];   //最小值
out 1<= chn[1];   //次小值
out 2<= chn[2];
.....
```

经过以上几个步骤,算法完成,整个算法共需要4个时钟周期。

#### 4 算法仿真与验证

有25个数,要求快速得到数据的中值和第三大值。这里可以采用并行全比较排序算法。设计采用ALTERA公司CycloneIII系列FPGA,时钟周期为10ns。在QUARTUSII平台环境下,按照设计原理

完成工程设计。

工程编译完成后,对工程文件进行仿真,观察仿真结果,与实际值对照,确认算法的正确性。图1为设计仿真效果图。图中,in0~in24为待比较的25个数据,约4个时钟周期的流水线运算时间,输出排序结果结果,outm,out2分别为数据中的中值和第三大值。

#### 5 并行全比较排序性能分析

##### 5.1 空间复杂性

由于并行全比较排序算法需要全并行处理,因此占用了大量的处理空间,又可称之为“空间换时间”排序算法。经计算统计,如果需要排序的总数字个数为n,则并行排序算法所需要的比较器为 $(n-1)2$ 个,每个比较器在FPGA中设计占用逻辑单元为5个左右,那么排序需要的逻辑单元为 $5 \times (n-1)2$ 个。

如果要处理100个数排序,需要近5万逻辑单元。对于ALTERA公司的CycloneIII系列FPGA来说,EP3C120型芯片有12万逻辑单元,需要的处理空间可以满足。

##### 5.2 时间复杂性

在FPGA内实现并行全比较算法,只需4个时钟周期就可实现排序计算。时间复杂度为定值。如果FPGA的时钟周期为10ns,那么整个排序算法时间只有40ns。

而传统的排序算法比如冒泡法,时间复杂度为 $n(n-1)/2$ ,按照时钟周期10ns计算,100个数的冒泡排序时间可49.5  $\mu$ s,远远大于并行排序算法时间,这其中还不包括相关循环计算时间。

#### 6 结语

从上述论述可知,全比较排序算法基于FPGA技术,实现数据排序并行处理,排序运算只需要几个时钟的运算时间,并且运算时间不随排序数据量变化,达到了实时性排序的效果。该算法具有通用性,可以应用到各种数据快速排序运算领域。

#### 参考文献

- [1]陈树平,梁咏梅.排序算法时间复杂度研究.商丘师范学院学报,2004.4:74~77.
- [2]马占欣,凌凤彩.一种基于统计的排序算法.小型微型计算机技术,2002.1:1403~1405.

# 基于FPGA的并行全比较排序算法

作者: [师廷伟](#), [金长江](#)  
作者单位: [中国电子科技集团公司第二十七研究所 河南郑州 450047](#)  
刊名: [数字技术与应用](#)  
英文刊名: [Digital Technology and Application](#)  
年, 卷(期): 2013(10)

## 参考文献(2条)

1. [陈树平;梁咏梅](#) 排序算法时间复杂度研究[期刊论文]-[商丘师范学院学报](#) 2004(5)
2. [马占欣;凌凤彩](#) 一种基于统计的排序算法[期刊论文]-[小型微型计算机系统](#) 2002(11)

引用本文格式: [师廷伟](#). [金长江](#) [基于FPGA的并行全比较排序算法](#)[期刊论文]-[数字技术与应用](#) 2013(10)