

文章编号 1004-924X(2011)03-0681-09

# 多目标实时跟踪可编程片上系统的软件优化

叶有时,赵保军,唐林波,蔡晓芳

(北京理工大学 电子工程系,北京 100081)

**摘要:**针对传统 DSP+FPGA 设计方案结构复杂、功耗高、体积大、抗干扰性差等问题,应用可编程片上(SOPC)技术建立了多目标实时跟踪系统并提出了基于 NIOS II 核及其数据链路特征的软件优化方法,对系统采用的迭代投影跟踪算法和通信制导控制算法进行了优化。研究了算法结构优化,链路配置优化,代码优化,自定义指令和 C2H 技术,并将这些技术应用于多目标跟踪 SOPC 系统来优化算法结构,减少代码量,去除程序相关性并进行链路层硬件转换,提升软件并行流水性能。实验结果表明,该优化方法可使软件效率提升 9~18 倍。在图像序列 100 frame/s,分辨率 256 pixel×256 pixel 的条件下,在 1.3 ms 内完成了单帧图像的多目标跟踪处理,470  $\mu$ s 内完成了单周期导航运算处理,使系统运算满足了实时处理的要求并在节约硬件成本的同时简化了结构,提高了运行稳定性。

**关键词:**软件优化;多目标跟踪;实时处理;可编程片上系统

**中图分类号:**V243.5;TP311 **文献标识码:**A **doi:**10.3788/OPE.20111903.0681

## Software optimization of multi-target real-time tracking SOPC system

YE You-shi, ZHAO Bao-jun, TANG Lin-bo, CAI Xiao-fang

(*Department of Electronic Engineering, Beijing Institute of Technology, Beijing 100081, China*)

**Abstract:** For the problems of complex structure, high power consumption, large size and poor anti-interference in the traditional DSP+FPGA design, a real-time multi-target tracking system based on a System on Programmable Chip(SOPC) was established and a software optimization method based on NIOS II core and its data link was proposed. The optimization technologies for algorithm structure, data link configuration, code optimization, custom instruction and C2H were researched. Then, these technologies were applied to the multi-target tracking SOPC system to optimize algorithm structure, reduce the size of code, remove the correlation of code and implement link-layer hardware conversion to enhance the performance of software pipelining. Experimental results indicate that these optimizing technologies can improve the efficiency of software by 9—18 times. The computing time of multi-target tracking is 1.3 ms/frame and the navigation operation time is 470  $\mu$ s/cycle for the image sequence of 256 pixel×256 pixel in 100 frame/s. The system can realize real-time processing with low hardware costs, simple structure and stable performance.

**Key words:** software optimization; multi-target tracking; real-time processing; System On Programmable Chip(SOPC)

收稿日期:2010-04-19;修订日期:2010-07-09.

基金项目:国家 863 高技术研究发展计划资助项目(No. 2008AA8012320B)

## 1 引言

随着科学技术的进步,人类在外太空的活动日益频繁,深空目标跟踪作为空间技术的研究热点在卫星导航,导弹预警等众多领域都有着广泛的应用<sup>[1-2]</sup>。太空环境复杂多变,空间跟踪系统逐渐向低功耗、轻小型方向发展,不仅功能越来越复杂,还要具有高稳定性以克服空间射线和粒子翻转等因素带来的干扰。传统的空间跟踪系统多基于 DSP+FPGA 结构<sup>[3-4]</sup>,随着任务量的增大,单芯片 DSP 已无法完成大数据量处理,而增加 DSP 芯片则会增大系统体积,使功耗增加<sup>[5-6]</sup>。鉴于此,集成微控制器、存储器、多种外围电路及实时操作系统、实时通信协议等应用软件的片上系统 SOC(System On Chip)成为空间系统设计的发展方向。

可编程片上系统(System On Programmable Chip, SOPC),可将高性能的 32 位 NIOS II 处理器、存储器、I/O 等系统设计需要的功能模块通过 IP 核调用集成到一片 FPGA 器件上,由单个芯片完成整个系统的主要逻辑功能<sup>[7]</sup>。与传统的 FPGA+DSP 构架相比,它可以很快地将硬件系统和软件设计都放在单个可编程芯片中,是一种新的软硬件综合的系统设计技术<sup>[8]</sup>。基于 SOPC 技术的目标跟踪系统以高度的集成化减小了系统复杂度,可以根据需要将多种功能集成在一块芯片上,单芯片完成传统多电路板设计的全部功能,降低了制造成本,使图像的获取和处理变得简单快捷,能设计出更灵巧的小型目标跟踪系统。充分利用 FPGA 的逻辑资源配合高速处理的 NIOS II 微处理器,可以大大简化电路设计结构,使整个系统更集成化。

一个性能优良的空间跟踪系统,不但应具有好的空间环境适应性,还要具有高速实时的处理能力。在系统硬件资源有限的情况下,庞大复杂的动态实时系统中多目标跟踪和导航算法结构复杂,运算量大,软件系统间频繁的数据交互带来了实时性瓶颈,提高软件执行效率,必须有快速高效的软件算法。多目标的检测跟踪主要涉及定点运算,导航计算精度要求较高,浮点运算使用较多。

本文结合算法各自特点为多目标跟踪系统提出一种基于 NIOS II 核及其数据链路特征的软件优化方法,采用多种基于数据链路特征的软件优化技术来提高软件效率。

## 2 多目标跟踪系统的构成及工作原理

为满足高速实时处理的需求,增强空间环境下的生存能力,本文设计了一种基于 SOPC 的深空多目标跟踪系统<sup>[9]</sup>。该设计使用 ALTERA 公司的 EP2S90 芯片,在内部构建 2 片高性能的 NIOS II 微处理器提升系统性能,具有体积小,重量轻,低功耗,运行稳定,抗粒子翻转能力强的优点,可满足空间环境对跟踪系统各方面的要求。

系统设有两片 CPU (NIOS II/f 型, 200 MHz, 1 600 MIPS<sup>[7]</sup>), 软件算法主要由双 CPU 结构实现, CPU-1 用于进行迭代投影<sup>[9]</sup>完成多目标检测跟踪; CPU-2 用于通信及制导控制。两片 CPU 独立工作, 但采用同一时钟(150 MHz 系统时钟), 各自分配 128 KB 的片上 RAM 资源作为程序与数据空间, 通过 Mailbox 进行通信。Avalon 总线是一种高效的总线结构<sup>[7]</sup>, 它定义了主端口和从端口之间传输数据所需的信号和时序, 用于连接片内 NIOS II 处理器和各种外设 IP 核, 以构成 SOPC 系统。

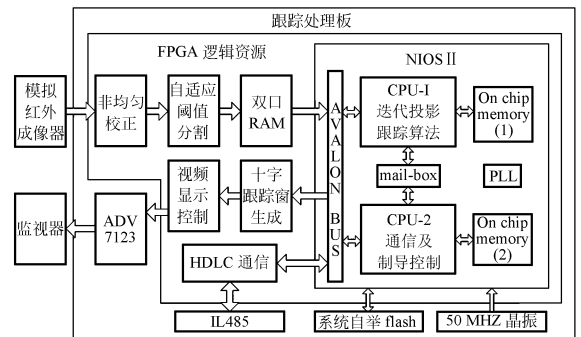


图 1 多目标跟踪系统原理图

Fig. 1 Block diagram of multi-target tracking system

系统工作流程如图 1 所示。利用 FPGA 硬件逻辑资源完成红外图像非均匀校正<sup>[10]</sup>和自适应阈值分割<sup>[11]</sup>, 并将处理后的二值图像存入双口 RAM; CPU-1 通过 Avalon 总线从双口 RAM 中读取数据进行迭代投影, 完成多目标跟踪, 加十字

跟踪窗经 DAC 转换电路由监视器输出显示,同时通过 Mailbox 将跟踪结果交给 CPU-2。负责通信及制导控制的 CPU-2 利用捷联惯导系统(IMU)采集的惯导数据和主控计算机指令进行导航运算,执行正确的控制操作,并将获取的运动目标信息及状态参量通过 HDLC 通信模块经 RS485 芯片下传给主控计算机。

### 3 算法结构优化

#### 3.1 结合系统时序及软件相关性调配算法结构

软硬件信息交互过程中的时序配合至关重要,时序太宽松会造成时隙上的浪费,降低软件执行效率;时序太苛刻会导致计算出现错误。考虑到软件函数体间的相关性,可在软件执行快于硬件执行的两个函数体中间插入非相关的数据处理函数体,将不相关的函数体放在一起执行,在满足时序要求避免发生错误的同时又能充分合理地利用每一空闲时隙,提升软件效率。本文系统中负责通信与制导控制的 CPU-2 频繁与外部 HDLC 通信模块进行软硬件信息交互,涉及大量系统时序及软件相关性,按如下原则对软件程序进行优化处理:

(1)具有相关性的两发送函数间插入非相关的数据处理函数

CPU-2 将数据通过 Avalon 总线传递给 HDLC 通信模块,按 HDLC 通信协议处理后发送出去。CPU 将数据写入硬件缓存器的速度极快(100 MHz),而硬件通信模块工作于 8 MHz 时钟下,复杂处理要耗去一定时间。如果连续向一个通信模块写入几帧数据,则后一帧数据会将前一帧未处理完的数据覆盖,造成通信故障。因此在两个发送函数之间插入了非相关的数据处理函数,以避免无谓的等待,提升软件效率。

(2)具有相关性的收发函数间插入非相关的数据处理函数

系统中 CPU-2 周期性向 IMU 发送取数据命令后,IMU 设备会有 160  $\mu$ s 的反应处理时间才能回送数据,如在此时隙查询等待,则造成时间上的浪费,因此,这里插入了与 IMU 计算不相关的

代码执行其他任务,节省处理时间。

#### 3.2 避免使用递归算法结构

递归算法<sup>[12]</sup>是一种直接或间接地调用自身的算法。在使用递归策略时,该算法将问题转化为规模缩小的同类问题,递归调用自身求解。问题规模极小时,必须直接给出解答而不再进行递归调用,即必须有一个明确的递归结束条件,否则程序不能正常结束。递归调用使算法的描述简洁且易于理解。本文系统采用迭代投影算法<sup>[9]</sup>实现多目标跟踪,采用递归结构实现,其流程如图 2 所示:

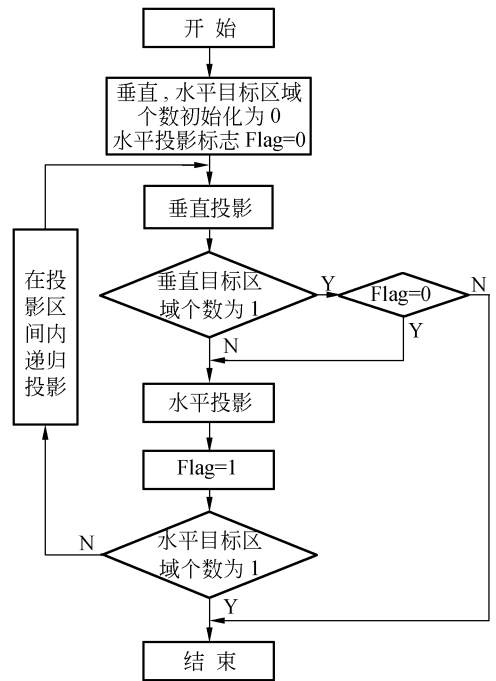


图 2 递归投影算法流程

Fig. 2 Flow chart of recursive iterative projection algorithm

虽然递归算法结构清晰、易于理解,但递归调用过程中每一层的返回点、局部量等都需要开辟栈来存储,递归次数过多容易造成栈溢出,对存储空间和硬件运算性能的要求都很高,执行效率较低。同时,NIOS II 硬件加速编译器<sup>[7]</sup>也不支持递归机制,无法对递归程序进行硬件加速。为了节省存储空间,充分利用 CPU 和 FPGA 并行流水提高算法的执行效率,采用非递归方法来实现迭代投影算法,其流程如图 3 所示。

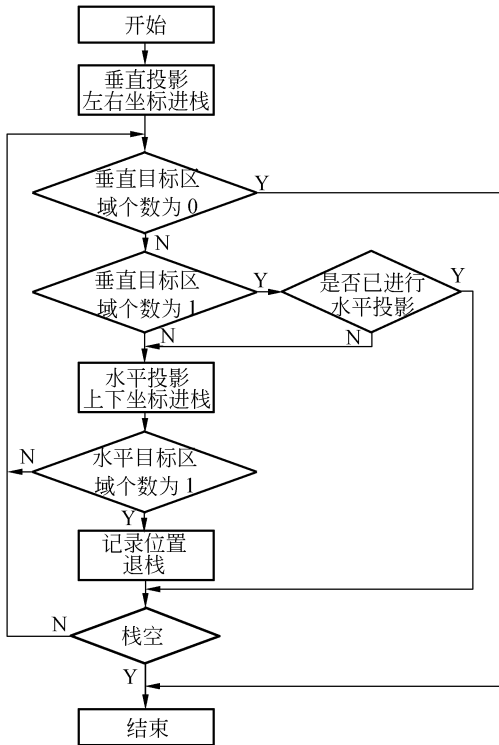


图 3 非递归投影算法流程

Fig. 3 Flow chart of non-recursive iterative projection algorithm

通过表 1 对比,可以看到将算法结构改为非递归模式后,算法执行速度提升了 1~5 倍。程序执行次数越多,任务量越大,非递归结构提升的性能就越高。

表 1 递归和非递归方法的执行时间

Tab. 1 Execution time of recursive and non-recursive approaches

程序执行次数	执行时间/ms	
	递归算法	非递归算法
100	31	16
1000	406	78
10 000	4 109	797

## 4 CPU 配置及程序优化

### 4.1 NIOS II 硬件资源优化配置

(1) 选用快速的 NIOS II/f 型处理器。该型 CPU 具有六级流水结构,带指令和数据 cache,是 ALTERA 提供的性能最强的软核 CPU<sup>[7]</sup>。增大 cache 可减小取指延迟,显著提高运算性能,本文

设计使用二级缓存,开辟数据与指令 cache 均为 64 KB。

(2) 构建哈佛结构的存储器架构,使用独立的数据与指令存储器。使用双端口紧耦合片上 RAM 作为指令空间,减少多操作数取数延迟。取指令和存取数据分别经由不同的存储空间和总线,使得各条指令可以重叠执行,克服了数据流传输瓶颈,便于软件并行流水处理。

(3) 减小代码量:①将 nios2-elf-gcc 编译器中的优化级别设为-O3,且在用户工程和系统库中都使用相同选项。②使用裁剪后的小封装驱动库 reduce device drivers。③使用简化的 C 语言库 small C library。④手动裁剪代码驱动,例如:processor 选项中加入 DAL\_T\_NO\_CFI\_FLASH 来减小代码量。⑤Active Build Configuration 选项改为 Release 模式,Debug Level 选为 None。⑥设定较小的 Max file descriptors 常数,选中 Lightweight device driver API 选项,得到简化的驱动代码。⑦完成调试后,去除 JTAG\_UART 和相应驱动。⑧选中 Program never exits 减少嵌入式系统 exit 带来的负担。

(4) 优化中断配置:①根据系统需求分配每个外设的中断优先级。②使用 NIOS II 提供的中断向量定制指令加速中断向量指派,减少中断响应时间。③选择 use a separate exception stack 选项,将异常堆栈映射到一个快速的存储空间提高异常处理速度。

(5) 提升 CPU 主频:①CPU 与存储器使用分立的时钟源。②减少不必要的存储器连接,可减少逻辑使用量并提升 CPU 主频。③使用 pipeline bridge 桥接非存储器类外设以大幅度提升 CPU 主频。

### 4.2 NIOS II 程序代码优化

(1) 尽量把使用频繁的变量定义为全局变量,避免反复开辟和释放空间造成软件执行效率下降。函数间的标志传导量,如硬件设备的状态寄存器,中断标志位,多线程应用中被几个任务共享的变量,定义为易变型全局变量(volatile)。

(2) 大量命令参数的记录和简单函数处理使用预处理器宏命令实现,使编译器产生比函数程序更优化的代码。

(3)优化中断函数。①中断中不宜有包含循环体的长执行代码,中断服务程序(ISR)中进行非关键的大数据量处理,将影响系统中中断关键任务的执行。应将其移到中断外部的应用程序中,轮询标志位处理。②合理使用中断嵌套。系统正在执行低优先级ISR时高优先级中断提出中断请求,系统会先处理高优先级ISR再返回低优先级ISR继续执行,减小高优先级ISR的中断延迟。

(4)使用执行效率较高的条件语句替代分支语句,长条件判断语句中不应插入长循环体,否则会阻断CPU软件流水使效率下降。乘除运算为多周期指令,可根据实际情况,用逻辑移位运算代替乘除运算,加快指令执行速度。

(5)正确使用循环体。循环体中不能调用一般函数;不能包含改变循环计数器的代码;不能有break和goto语句,If语句不能嵌套,条件代码尽可能简单;循环结构代码不能过长,太长的话可分解为多个循环体;循环之中不能再有循环展开<sup>[13]</sup>;使用-O3优化选项,编译器将尽可能地将循环体展开,使并行指令数增加,利用CPU的多级流水结构安排循环中的多个迭代运算并行执行,使循环体达到最佳软件流水状态。

(6)去除程序代码间的相关依赖性,代码中包含有依赖其他变量的变量时,编译后会阻断软件流水,可使用restrict关键字确保同一块内存上的指针或数组没有其它引用,去除指令相关性,让编译器更好地优化指令达到软件流水。如下例,将代码段1优化为代码段2编译后可提高软件执行的并行度。

代码段 1	代码段 2
<code>void swap( )</code>	<code>void swap( )</code>
<code>{</code>	<code>{</code>
<code>  int tmp= p[0];</code>	<code>  int * _restrict_p0=p[0];</code>
<code>  p[0]= p[1];</code>	<code>  int * _restrict_p1=p[1];</code>
<code>  p[1]= tmp;</code>	<code>  int tmp0= * p0;</code>
<code>}</code>	<code>  int tmp1= * p1;</code>
	<code>  * p0= tmp1;</code>
	<code>  * p1= tmp0;</code>
	<code>}</code>

## 5 专用加速工具的使用

专用数字信号处理芯片 DSP 具有 8 个运算

单元可同时处理 8 条指令<sup>[14]</sup>,而 NIOS II 作为通用 RISC 处理器在同一时刻只能利用一个算术逻辑单元(ALU)进行运算<sup>[7]</sup>。要想达到高速实时处理,必须利用功能强大的 FPGA 硬件逻辑资源构建功能模块,完成程序中的部分任务。传统方案将算法做功能划分,可脱离 CPU 实现的部分,如图像预处理用 FPGA 完成。但某些算法如多目标跟踪,内部各函数指令间交互调用,过于细致的功能划分会增加系统设计的复杂度,必须依赖 CPU 软件实现。这就要求软件可利用多个 ALU 并行运算。

增加并行 ALU 的方法,一种是增加 CPU 数量,即通过多核结构扩充系统的计算能力,本设计采用双核结构实现多目标跟踪和通信制导控制算法,使系统运算达到实时。另一种方法是从软件程序着手,利用 ALTERA 公司提供的开放的 ALU 和专用加速工具进行 CPU 扩展,将软件中耗时较大的函数和指令转化为独立于 ALU 的数据链路层硬件运算单元,提升运算能力。本节介绍两种方法,一种为自定义指令<sup>[7]</sup>,即在 ALU 上构建并行的硬件指令模块加速某些特定指令代码,另一种为 C2H 加速技术<sup>[15]</sup>,即将软件 C 语言函数转换为硬件功能模块。其实质都是为 CPU 构建了若干辅助运算单元。但这些运算单元不同于 ALU 可进行通用指令处理,而是专用于完成某些特定代码段功能的计算。

### 5.1 自定义浮点指令的应用

NIOS II 定制指令是与 CPU 数据链路中的 ALU 并联的用户逻辑块。使用定制指令,可以将一个复杂的标准指令序列简化为一个用硬件实现的单一指令,从而简化系统软件设计,提高运行速度。在 NIOS II CPU 配置向导<sup>[7]</sup>中导入设计文件,设置定制指令名,并分配定制指令所需的 CPU 时钟周期数目。系统生成时,将为每条用户指令产生一个在系统头文件中定义的宏,可以在应用程序中直接调用这个宏。

通信控制算法主要是浮点运算且富含大量三角函数及反三角函数,NIOS II 利用纯软件计算浮点三角函数或反三角函数,需要把这一操作分解为若干条指令进行计算,耗费大量的时钟周期(见表 2)。为满足系统要求,本文采用自行设计

的单精度浮点正/余弦,反正切和反正弦自定义指令进行加速计算。仅用几个时钟周期即完成了指令计算,使单指令执行速度提高了 200~300 倍。

表 2 浮点自定义指令的加速比

Tab.2 Accelerated ratio in floating-point custom instructions

单精度 浮点运算	CPU 时钟周期		加速比
	软件执行	自定义指令	
正/余弦	7 184	32	224.5
反正切	24 866	74	336
反正弦	26 756	75	356.7

## 5.2 C2H 硬件加速技术的应用

C2H 技术的实质就是利用 C2H 工具将软件程序中某些运算量大、耗时多的瓶颈代码通过 C2H 加速器转换为独立于 ALU 的数据链路层硬件加速模块,挂到 Avalon 总线上作为用户外设使用<sup>[15]</sup>,以硬件模式加速软件功能的同时节省了 CPU 开销。但是 C2H 编译器对要加速的 C 代码有一些限制,而且只能对定点算法进行加速。本文设计采用 C2H 技术提高多目标跟踪算法处理速度,由于算法较为复杂,有必要对相应代码段进行优化,使指令执行并行流水,生成效率更高的硬件加速模块。

(1)避免使用长运算路径。长运算路径的代码生成内含一条串行运算链路的硬件加速模块,处理延迟较长。可将其拆分为若干短运算路径的语句,生成内含多条并行运算链路的加速模块。如将代码段 3 拆分为代码段 4 的组合,优化代码后利用 C2H 加速可大大减少处理时间。

代码段 3:

```
int result=a+b+c+d+e+f+g+h;
```

代码段 4:

```
int r_abcd=a+b+c+d;
```

```
int r_efgh=e+f+g+h;
```

```
int result=r_abcd+r_efgh;
```

(2)定义功能相同的独立函数替代复用子函数。若程序中连续调用同一子函数,最好定义两个功能相同的独立函数,生成两个并行的硬件加速模块,提升系统处理性能。如将代码段 5 改写为代码段 6,可利用 C2H 工具生成 mul1 和 mul2 两个硬件加速模块,并行完成 muladd 函数的功能,相对于多次复用代码段 5 生成的单一模块,可

节省一半时间。

代码段 5

子函数

```
int mul(int x,int y)
```

```
{
```

```
    return x×y;
```

```
}
```

主函数:

```
int muladd(int a,b,c,d)
```

```
{
```

```
    int prd1=mul(a,b);
```

```
    int prd2=mul(c,d);
```

```
    return r=prd1+prd2;
```

```
}
```

代码段 6

子函数:

```
int mul1(int x,int y)
```

```
{
```

```
    return x×y;
```

```
}
```

```
int mul2(int x,int y)
```

```
{
```

```
    return x×y;
```

```
}
```

主函数:

```
int muladd(int a,b,c,d)
```

```
{
```

```
    int prd1=mul1(a,b);
```

```
    int prd2=mul2(c,d);
```

```
    return r=prd1+prd2;
```

```
}
```

(3)去除程序代码间的相关性。为避免程序内循环体由于相关依赖生成串行级联运算的硬件加速模块,软件优化中要进行去相关处理,即后一级运算尽量避免使用前一级运算的结果,以便在硬件加速模块中生成并行级联的数据链路。如将代码段 7 改写为代码段 8,可消除两个循环体间的相关性,生成并行的硬件加速模块,提升系统运算性能。

代码段 7

```
int sary(int * ra,int * rb)
```

```
{
```

```
    for(i=0;i<size;i++)
```

```
    {
```

```
        sum+=ra[i];
```

```
    }
```

```
    for(i=0;i<size;i++)
```

```
    {
```

```
        sum+=rb[i];
```

```
    }
```

```
}
```

代码段 8

```
int sary(int * ra,int * rb)
```

```
{
```

```
    for(i=0;i<size;i++)
```

```
    {
```

```
        int s_a+=ra[i];
```

```
    }
```

```
    for(i=0;i<size;i++)
```

```
    {
```

```
        int s_b+=rb[i];
```

```
    }
```

```
    sum=s_a+s_b;
```

```
}
```

## 6 实验结果及分析

利用本文所述方法分别对 CPU-1 的多目标跟踪软件和 CPU-2 的通信制导控制软件进行优

化,对 256 pixel×256 pixel 大小,帧率为 100 Hz 的图像序列进行目标跟踪实验,采用不同的优化技术得到相应的优化效果,优化效果对比如表 3~表 4 所示:

表 3 多目标跟踪软件优化效果

Tab. 3 Optimization results of multi-target tracking software

优化步骤	优化前时间	优化后时间	加速比	采用的技术
	/ms	/ms		
1	24	17	1.41	非递归结构优化
2	17	13	1.3	CPU 配置优化
3	13	12	1.08	程序代码优化
4	12	1.3	9.23	C2H 硬件加速
/	24	1.3	18.46	总效能提升

可以看到本文设计依次通过非递归的算法结构优化,CPU 配置优化,程序代码优化和 C2H 硬件加速技术,将多目标跟踪软件的效率提升了 18.46 倍。其中基于非递归结构的算法优化使加速比提升 1.41 倍,CPU 配置优化使加速比提升 1.3 倍,程序代码优化提升 1.08 倍,C2H 硬件加速技术将软件效率提升 9.23 倍。

表 4 通信制导控制软件优化效果

Tab. 4 Optimization results of communication and guidance software

优化步骤	优化前时间	优化后时间	加速比	采用的技术
	/ms	/ms		
1	4.1	3.8	1.07	时序调整结构优化
2	3.8	2.2	1.7	CPU 配置优化
3	2.2	2	1.1	程序代码优化
4	2	0.462	4.32	定制浮点指令
/	4.1	0.462	8.87	总效能提升

## 参考文献:

- [1] 漆长松,刘恩海,钟建勇.天基平台深空运动小目标检测[J].光电工程,2008,35(5):28-32.  
QI CH S, LIU EN H, ZHONG J Y. Discussion on space-based deep space small moving target detection [J]. *Opto-Electronic Engineering*, 2008, 35 (5):28-32. (in Chinese)
- [2] CLARK F O, DANIEL P H. Automatic target recognition by matching oriented edge pixels [J]. *IEEE Transactions on Image Processing*, 1997, 6

依次利用基于时序调整的算法结构优化,CPU 配置优化,程序代码优化和自定义浮点指令,将通信制导控制软件的效率提升了 8.87 倍。其中基于时序调整的算法结构优化使加速比提升 1.07 倍,CPU 配置优化使加速比提升 1.7 倍,程序代码优化提升 1.1 倍,自定义浮点三角函数指令提升 4.32 倍。

对实验结果进行分析可知,单纯针对算法结构和代码的优化很难使系统性能得到较大提升,只有结合 NIOS 处理器数据链路的特点进行相应的链路配置及代码转化,灵活扩展数据链路结构,才能有效提升软件执行效率。

## 7 结 论

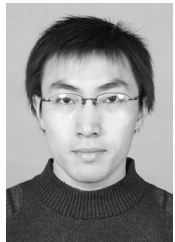
多目标跟踪系统的功能不断增加,结构越来越复杂,SOPC 设计方案以其功耗低,体积小,稳定性好,易于单片 FPGA 实现的优点得以推广应用。以往解决多目标实时跟踪问题大多将注意力集中于提升硬件性能,或追求快速的目标跟踪算法,很少对软件自身进行深层优化。而单纯针对语法的优化,系统性能也很难有大的提升,只有把软件与硬件的结构特点融为一体,才能大幅度地提高处理速度。本文提出的基于 NIOS 处理器数据链路的软件优化方法很好地解决了这一问题。实验结果表明,本文优化方法可使程序执行效率提升 9~18 倍,在 1.3 ms 内完成单帧目标跟踪处理,470  $\mu$ s 内完成单周期导航运算,使系统运算达到实时性要求。

(1):103-113.

- [3] YIN CH F, ZHAO D. The research on target recognition and image tracking system based on high-speed DSP[C]. *Proceedings of the 4th International Conference on Computer Science & Education, Nanning, P.R. China: ICCSE*, 2009:697-700.
- [4] 赵鹏,沈庭芝,单宝堂.一种基于粒子滤波的无人机电视导引系统目标跟踪算法[J].光学精密工程,2008,16(1):134-140.  
ZHAO P, SHEN T ZH, SHAN B T. An object tracking algorithm for TV guiding system of UAV

- based on particle filter [J]. *Opt. Precision Eng.*, 2008,16(1):134-140. (in Chinese)
- [5] 王广平,许廷发,倪国强,等. 多 ADSP-TS201 红外弱小目标实时检测跟踪系统的硬件设计[J]. *光学精密工程*,2007,15(6):941-944.  
WANG G P, XU T F, NI G Q, *et al.*. Hardware design of infrared dim target real time detection and tracking system based on multiple ADSP-TS201S [J]. *Opt. Precision Eng.*, 2007,15(6):941-944. (in Chinese)
- [6] 许廷发,赵思宏,周生兵,等. DSP 并行系统的并行粒子群优化目标跟踪[J]. *光学精密工程*,2007,17(9):2236-2240.  
XU T F, ZHAO S H, ZHOU S H B, *et al.*. Particle swarm optimizer tracking based on DSP parallel system [J]. *Opt. Precision Eng.*, 2007,17(9):2236-2240. (in Chinese)
- [7] ALTERA. *NIOS II processor reference handbook* [M]. Published by Altera, 2009.
- [8] XU H, WEI S H. An embedded SOPC system using automation design [C]. *Proceedings of the International Conference on Parallel Processing Workshops, Oslo, Norway; ICPP*,2005:232-239.
- [9] 叶有时,唐林波,赵保军,等. 基于 SOPC 的深空目标实时跟踪系统[J]. *系统工程与电子技术*,2009,31(12):3002-3006.  
YE Y S H, TANG L B, ZHAO B J, *et al.*. Real-time deep space target track system based on SOPC [J]. *Systems Engineering and Electronics*, 2009,31(12):3002-3006. (in Chinese)
- [10] 丁瑞,史彩成,何佩琨. 基于 FPGA+DSP 系统的红外图像非均匀校正的实现[J]. *光学技术*,2006,32(8):121-123.  
DING R, SHI C H, HE P K. The realization of two-point non-uniformity correction of infrared image based on FPGA + DSP [J]. *Optical Technique*, 2006,32(8):121-123. (in Chinese)
- [11] 王勇,赵保军. 复杂背景下低信噪比点目标的实时检测算法及实现[J]. *系统工程与电子技术*,2002,24(12):18-21.  
WANG Y, ZHAO B J. A real-time detection algorithm for low SNR point targets in a complex background [J]. *Systems Engineering and Electronics*, 2002,24(12):18-21. (in Chinese)
- [12] Bruno Pimentel. Recursion in hardware: applicability and implementation strategies[C]. *Proceedings of the Second International Conference on Advances in Circuits, Electronics and Micro-electronics. Sliema, Malta; CENICS*, 2009:94-100.
- [13] 孟占红,赵保军. 基于 DSP 的实时图像压缩软件优化技术研究[J]. *电子学报*,2006,34(9):1558-1561.  
MENG ZH H, ZHAO B J. The study on software optimizing technology of real-time image compression system based on DSP [J]. *Acta Electronica Sinica*, 2006,34(9):1558-1561. (in Chinese)
- [14] 虞响,张笑微,潘泽友. 实时目标跟踪中 TMS320DM642 的软件优化研究[J]. *兵工自动化*,2009,28(6):86-91.  
YU X, ZHANG X W, PAN Z Y. Software optimization in real-time tracking system of moving object based TMS320DM642 [J]. *Ordnance Industry Automation*, 2009,28(6):86-91. (in Chinese)
- [15] ALTERA. *Embedded Design Handbook* [M]. Published by Altera, 2010.

#### 作者简介:



叶有时(1983—),男,河北沧州人,博士研究生,主要从事高速实时信号处理、图像处理及嵌入式系统的研究。E-mail: youngtree@bit.edu.cn



赵保军(1965—),男,陕西西安人,教授,博士生导师,1998 年于北京理工大学通信与信号处理博士后流动站出站,主要从事高速实时信号处理、图像处理、嵌入式系统等方面的研究。E-mail: zbj@bit.edu.cn



唐林波(1978—),男,湖北京山人,博士,讲师,2005年于北京理工大学获得博士学位,主要从事数字图像及数字视频处理的研究。E-mail: tanglinbo@bit.edu.cn



蔡晓芳(1985—),女,海南海口人,硕士研究生,2007年于北京邮电大学获学士学位,主要从事信号处理、图像处理等方面的研究。E-mail: xiaofangcai@gmail.com

●下期预告

## 静电驱动的亚微米悬臂梁谐振器非线性特性

岳东旭,于虹,袁为民

(东南大学 电子科学与工程学院 MEMS 教育部重点实验室,江苏 南京 2100965)

当微谐振器的尺寸缩小到 nm 量级时其工作频率已达到 MHz、GHz 甚至更高,研究发现工作在高频段的谐振器具有非常明显的非线性行为。本文详细分析了亚微米尺寸的悬臂梁结构的非线性力学模型,讨论了非线性产生的物理机制,为实验分析提供理论依据。采用外部静电激励机制,使悬臂梁谐振器产生谐振,并借助 Polytec 激光多普勒测振系统检测其频率响应曲线。测试结果表明悬臂梁具有显著的非线性效应(即“弹簧变软”效应),但这种非线性几乎和交流电压无关,却随着直流电压的增大而显著增大,最大峰值偏移达到 0.5 MHz,提取出三组一阶机械弹性系数分别为 79.62, 31.75 和 14.92 N/m,实验结果符合理论规律,也进一步验证了非线性理论的正确性。对实验中的偏差也做了进一步的分析 and 讨论,考虑到过腐蚀对结果刚度和频率响应的影响,利用软件 ANSYS 做了相应的模拟,结果和实验测量的结果相吻合。