

一种 SPARC V8 结构 ALU 的并行错误检测结构设计

张 洵 颖

(西安微电子技术研究所, 陕西 西安 710065)

摘 要: 针对嵌入式处理器在线错误检测的需求, 文中基于编码预测的并行错误检测策略, 选择 Berger 编码作为编码预测错误检测编码, 设计了 SPARC V8 结构的加/减、逻辑和移位运算的 B0 编码预测结构. 该结构可以直接集成成为带有并行错误检测结构的 SPARC V8 结构 ALU, 相比于电路复制这样的结构实现了硬件资源的节省. 与对偶复执这样两倍的执行时间并且附带结果比较的策略相比, 具有明显的性能优势.

关键词: 错误检测; Berger 编码; SPARC V8; ALU

中图分类号: TP302.8

文献标识码: A

文章编号: 1000-7180(2011)01-0180-05

A Parallel Error Detection Structure Design of SPARC V8 Architecture Compatible ALU

ZHANG Xun-ying

(Xi'an Microelectronics Technology Institute, Xi'an 710065, China)

Abstract: According to the on-line error detection requirement of embedded processors, this paper presents a parallel error detection structure of SPARC V8 architecture compatible ALU, which takes the B0 encoding of Berger code prediction as the parallel on-line error detection strategy. This structure can be integrated into the SPARC V8 architecture compatible ALU directly, and forms a parallel error detection ALU. Comparing with the two-rail logic method, this structure gets the hardware cost decrease. Comparing with pseudoduplication method, in which the same circuit successively processes data twice but along different data path, this structure has an obviously performance advantage.

Key words: error detection; Berger code; SPARC V8; ALU

1 引言

对于嵌入式系统而言, 从设计规范制定、设计、综合, 再到制造和使用, 可能出现不同种类的故障, 以至于错误、失效^[1]. 作为嵌入式系统中核心器件的嵌入式处理器, 其故障行为所导致的恶性后果, 往往是不能承受的. 为了避免出现灾难性的后果, 可靠的嵌入式微处理器需要加入早期的故障检测机制来进行故障检测.

在实现策略上, 在线故障检测技术可以分为信息冗余和硬件冗余两种冗余策略, 但是大多数情况下都是这二者的组合应用.

最简单的硬件冗余策略是在线并发检测, 通过

复制相关的电路模块, 然后通过比较信号或者数据来判断故障. 这种为检测错误而进行比较的容错策略, 是通过相应电路部分的多模实现来进行的, 这需要付出比较大的实现开销, 一般来讲都超过 100% 的硬件开销. 这种方法的一个优点是能够检测运行中的任意错误. 另外一种硬件冗余错误检测方法是双路逻辑, 该方法实现了原始设计和带有取反输出的原始设计的复制对偶^[2], 在该方法中, 并非所有的开销都是 100%. 作为一种逻辑级的冗余方法, 双路逻辑电路带来了可测性难度的上升. 通过对于电路复制的修改来减少实现开销, 比如伪复制方法在同样电路的不同数据路径上顺序两次处理数据, 这样是用处理时间的上升换来了实现开销的降低.

在错误检测和纠错中应用比较广泛的还有错误检测编码,比如奇偶校验编码、海明码等^[3-4]. 为了能够进行检测,编码策略附带的还有相应的硬件逻辑开销. 可以进一步的将这些编码技术划分成数据冗余和硬件冗余. 数据冗余通过向数据向量中加入附加位或者提供信源/信道编码来实现. 硬件冗余是通过提供数据编码、码预测、以及结果比较来实现. 在应用编码技术进行容错设计时,要在设计复杂性的基础上,综合考虑实现的功能、性能、开销等综合因素^[5].

基于嵌入式处理器对于在线故障检测机制的需求,出现了以寄存器三模冗余、存储器接口 EDAC 等为代表的纠检错机制,但是对于数据路径上出现错误的检测技术则相对研究较少. 文中首先抽象出基于编码预测的并行错误检测结构的形式化描述和组织结构图;其次,对 Berger 编码的错误检测能力和编码开销在内的错误检测特性进行了分析;然后,根据 SPARC V8 结构定义的加/减、逻辑操作和移位操作,给出了其基于 B0 编码的编码预测计算模型,并设计了相应的硬件实现结构;最后,对该并行错误检测结构的硬件实现开销和性能进行了分析.

2 基于编码预测的并行错误检测结构

对于 RISC 微处理器的数据路径而言,对应的行为可以形式化为:

$$S = X \text{ op } Y \quad (1)$$

其中, X 和 Y 对应于操作数, op 对应于操作类型. 为了能够进行基于编码预测的并行错误检测,首先,需要在可用于错误检测的编码策略中进行选择,适用的编码策略是不改变原始信息,而将编码结果作为额外的附加位,在对原始信息进行对应操作时,能够根据作为编码结果的附加位进行操作结果对应编码的预测. 对于式(1)所形式化的数据路径操作而言,以 X_c 和 Y_c 指代 X 和 Y 的对应编码结果,则对于结果 S 的编码预测可以形式化为:

$$S_c = F(X, Y, X_c, Y_c) \quad (2)$$

通过对最终 S 的编码和编码预测结果进行比较来判断在数据路径上是否有错误发生. 其中对于 S_c 的编码预测结构可以和 S 的计算过程并行的执行,而且通过选择不同的编码策略,可以控制编码预测结构的硬件开销. 基于该策略的并行错误检测结构的组织框图如图 1 所示.

按照以上的形式化描述和结构组织图,对于不同的操作而言,为了能够实现基于编码预测的并行错误检测机制,必须建立该操作对应式(1)和式(2)

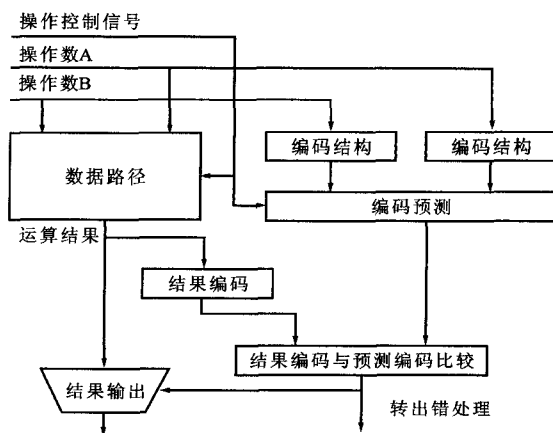


图1 基于编码预测的并行错误检测结构

的具体计算模型,并根据图 1 的组织结构进行相应的集成硬件结构设计.

3 Berger 编码的错误检测特性

Berger 编码是一种以发明人 J. M. Berger 姓名命名的能够检测所有单向错误的检测编码. 所谓单向,指的是错误影响的位变化都在同一个方向,也就是从 0 到 1,或者全是从 1 到 0^[6]. Berger 编码部分可以作为独立的检测位附加在信息的后面,这样不需要额外的译码器来提取信息的内容. Berger 编码是可以和信息分离的,即可以按照编码预测策略来进行相应的并行错误检测结构设计.

3.1 Berger 编码的编码开销和错误检测能力

信息的 Berger 编码可以有两种方式生成^[6]:

① B_0 : 信息中 0 的个数的二进制表示;

② B_1 : 信息中 1 的个数的二进制表示进行逐位取反作为编码.

文中仅就 B_0 模式的编码进行讨论, B_1 模式编码的内容是可以很直观的由 B_0 进行类推的.

Berger 编码的开销可以按照如下来计算: 对于 d 位数据而言,其中 0 最多有 d 个,最少有 0 个,也就是说会有 $d+1$ 种 Berger 编码的可能,为了能够进行错误的检测,也就是说能区分这些校验编码,最多需要 $r = \lceil \log_2(d+1) \rceil$ 来编码,这样,每一数据位的开销如下:

$$\frac{\lceil \log_2(d+1) \rceil}{d} \quad (3)$$

基于此, Berger 编码是在所有分离编码策略中需要位数最少的编码,并因此成为众所周知的最佳最大长度编码^[7].

由于 Berger 编码是一种单向检测编码,能够检测任何单位翻转错误,然而无法检测所有的任意位

任意翻转错误. 如果不考虑是否能够指示出错的位数, 而只是需要检测出是否有错误的发生, 那么对于 d 位的信息而言, 其能够检测的错误在所有的错误中所占比例可以定义如下:

(1) 对于 d 位数据而言, 出现 i 位翻转错误的情况为 C_d^i , 对于所有可能的 i , 所有的错误为:

$$\sum_{i=1}^d C_d^i = 2^d - 1$$

(2) 当 d 位中出现偶数位翻转, 并且从 0 到 1 的翻转数目和从 1 到 0 的翻转数目相同的情况下, 应用 Berger 编码进行比较来进行的检测无法发现错误, 对应于这种情况下错误的数目为:

$$\sum_{j=1}^{\lfloor \frac{d}{2} \rfloor} C_{2j}^j$$

(3) 这样, 使用 Berger 编码所能够检测的位翻

$$\text{转比例为: } \left[1 - \frac{\sum_{j=1}^{\lfloor \frac{d}{2} \rfloor} C_{2j}^j}{\sum_{i=1}^d C_d^i} \right] \times 100\%$$

基于以上分析, 表 1 给出了典型位数数据的编码位数和开销, 以及能够检测的错误的百分比.

表 1 Berger 编码的开销和错误检测率

信息位 (d)	编码位 (r)	编码的 开销	能够有效检测的 错误百分比
8	4	0.5000	61.57%
16	5	0.3125	73.18%
32	6	0.1875	81.13%
64	7	0.1094	86.68%

可以看出, 随着数据位数的增加, Berger 编码对应的编码开销在持续下降, 而能够检测的错误在整个错误空间中的比例在上升. 尽管, Berger 编码无法全部覆盖 d 位数据中任意位翻转错误, 但是相比于只能检测固定个数错误的其他检测编码来讲, 其检错能力的优势还是显而易见的.

3.2 Berger 编码的编码结构设计

为了得到数据的 B_0 编码需要对数据中“0”的个数进行计数, 考虑到设计的简单性因素, 对数据进行按位取反, 然后作为检“1”计数器的输入, 检得“1”的个数就是对应数据的 B_0 编码. 该检“1”计数器就成了基于 Berger 编码和预测的错误检测机制中的必要电路模块.

对于这样的检“1”计数器的设计来讲, 全静态的

实现策略包括通过使用全加器网络来实现、使用 ROM 阵列来实现. 全加器网络的实现是基于全加器本身就是一个 (3, 2) 计数器这一现实, ROM 阵列则是通过分类对应的查表模式. 相对于这两种方式的实现, 在速度和面积开销上更加优越的是基于阈值逻辑电路的并行实现^[8].

阈值逻辑函数定义了包含一组重要的多位二进制输入变量的布尔函数子集, 阈值门是指的这样的一组 m 位输入的逻辑门, 其输出由下面的函数决定:

$$f(x) = 0, \sum_{i=1}^m w_i \cdot x_i < T$$

$$f(x) = 1, \sum_{i=1}^m w_i \cdot x_i \geq T \quad (4)$$

其中, x_i 对应 m 位输入向量 x 的第 i 位, w_i 对应该位的输入权值, $f(x)$ 对应阈值门实现的逻辑函数, T 代表相应的门阈值, 阈值门的输出代表的布尔函数称为阈值函数, 可以应用该结构来方便的实现各种布尔函数. 文献[8]对于应用阈值门结构来实现各种规模的检“1”计数电路, 进行了详细的介绍, 可以方便的应用其结果来实现不同规模的 Berger 编码电路, 对于该电路的细节这里不再赘述.

4 编码预测结构

文献[9-10]针对应用 Berger 编码进行 ALU 的设计进行了研究, 建立了加/减、逻辑操作的数学基础, 该研究工作为研究基于 Berger 编码和预测的在线错误检测微处理器数据路径设计提供了很好的基础. 本节的计算模型都是基于文献[9-10]所给出的数学推导而来的.

4.1 加法/减法运算的编码预测结构

在 SPARC V8 结构中, 有符号整数使用的是二进制补码模式, 而无符号数使用原码模式, 基于文献[9-10]的数学推导, 对于 SPARC V8 结构中的加法操作:

$$S = X + Y$$

可以得到加法的 B_0 编码预测计算公式:

$$S_b = X_b + Y_b + c_n - C_b - c_0 \quad (5)$$

其中, 下标 b 用于指代对应值的 B_0 编码, C 为整个加法器的进位向量, c_i 对应第 i 位加法器进位, n 对应于操作数的位宽.

同样的, 可以得到减法的 B_0 编码预测计算公式:

$$S_b = X_b - Y_b + N(C) - \bar{c}_0 + c_n \quad (6)$$

其中, $N(C)$ 表示 C 中 1 的数目。

由于在 ALU 设计中, 加法和减法操作是通过加法器来实现的, 所以要通过式(5)、(6)的合并设计来实现加/减法结构的编码预测电路, 对应的设计结构如图 2 所示。其中, 使用“加/减选择信号”来进行对应加(对应“0”)或者减(对应“1”)操作的指示。

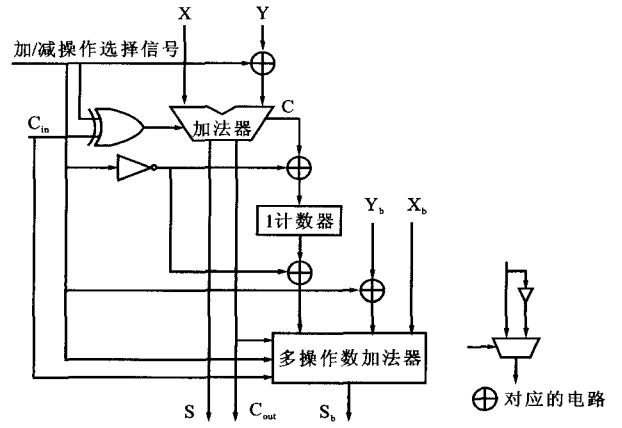


图 2 基于 Berger 编码和预测的加法器结构

由于加、减操作中对进位的 Berger 编码的操作不同, 使用加/减操作选择信号来控制是否对其进行求补操作; 相应的, 针对操作初始的进位输入和进位输出也需要进行处理以实现加、减操作编码预测的统一, 对应的结构如图 3 所示。表 2 给出了在图 2 结构下, 对应于加 / 减操作选择信号 O_s 、进位输入 C_{in} 和进位输出 C_{out} 作用下对应的情况。

表 2 加 / 减操作的对应控制信号

O_s	C_{in}	C_{out}	计算结果
0	0	0	$S_b = X_b + Y_b + c_n - C_b - c_0$
0	0	1	
0	1	0	
0	1	1	
1	0	0	$S_b = X_b - Y_b + N(C) - \bar{c}_0 + c_n$
1	0	1	
1	1	0	
1	1	1	

在这些处理的基础上, 经过所示的多操作数进位保留加法器进行编码计算后的输出就是对应操作结果的 B_0 编码, 通过其和计算结果的 B_0 编码的比较来进行错误检测。

4.2 逻辑操作的编码预测结构

在 SPARC V8 体系结构中定义的逻辑操作包括: AND、ANDN、OR、ORN、XOR 和 XNOR 六种,

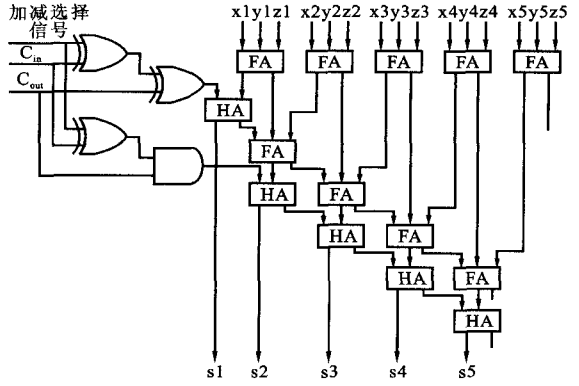


图 3 5 位多操作数加法器

基于文献[9-10] 的数学分析, 按照 SPARC V8 体系结构中定义的逻辑操作的内容, 可以得到以上的逻辑结构的编码预测计算公式如下:

$$\begin{aligned} X \wedge Y: S_b &= X_b + Y_b - (X \vee Y)_b \\ X \wedge \bar{Y}: S_b &= X_b - Y_b + (\bar{X} \vee Y)_b \\ X \vee Y: S_b &= X_b + Y_b - (X \wedge Y)_b \\ X \vee \bar{Y}: S_b &= X_b - Y_b + (X \wedge \bar{Y})_b \\ X \oplus Y: S_b &= X_b + Y_b - 2(X \wedge \bar{Y})_b + n \\ X \otimes Y: S_b &= -X_b - Y_b + 2(\bar{X} \wedge Y)_b \end{aligned} \quad (7)$$

按照前面对于逻辑操作的结构分析, 统一考虑所有的逻辑操作结构得到对应的逻辑操作的 B_0 编码预测结构如图 4 所示。“ $\times 2$ ”电路部件通过高位或低位补零来实现是否乘以 2; “ $+n$ ”电路通过从对应位加 1 来实现, 对应操作的控制信号在表 3 给出。

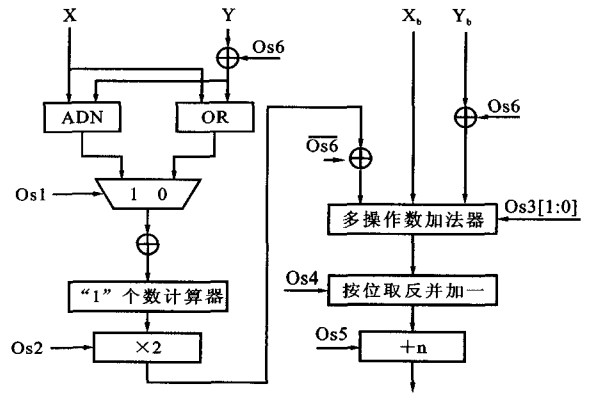


图 4 逻辑操作的 Berger 编码预测结构

表 3 逻辑操作对应的控制信号

操作	Os1	Os2	Os3	Os4	Os5	Os6
AND	0	0	01	0	0	1
ANDN	0	0	01	0	0	0
OR	1	0	01	0	0	1
ORN	1	0	01	0	0	0
XOR	1	1	10	0	1	1
XNOR	1	1	10	1	0	1

4.3 移位操作的编码预测结构

在 SPARC V8 体系结构中定义的移位操作包括:逻辑右移 SRL、逻辑左移 SLL 和算术右移 SRA. 对于这些移位操作而言, 根据 B_0 编码的编码预测结构主要是应用被移位数据的 B_0 编码减去移出数据中“0”的个数再加上移入数据中“0”的个数来得到. 这样, 对于移位操作, 以 X 代表被移位操作数, S_N 为五位移位码, S_C 为由 S_N 译码而来的 32 位移位控制码, 对其 B_0 编码的编码预测进行具体分析.

逻辑右移操作结果的 B_0 编码预测, 需要计算移出的 k 位数据中“0”的个数, 通过移位输入数据减去移出 k 位中“0”的个数加上移位数目 k 来得到. 为了计算移出的 k 位的数据中的“0”的个数, 对移位输入 X 进行如下处理:

$$X^{k^r} = \bar{S}_C \wedge \bar{X} \quad (8)$$

对得到的进行 X^{k^r} “1”的个数计数, 就能得到移出的 k 位中“0”的个数, 记为 $X_b^{k^r}$, 那么, 对应的预测编码为:

$$S_b = X_b - X_b^{k^r} + S_N \quad (9)$$

同样, 对于逻辑左移可以得到:

$$X^{k^l} = \bar{S}_C^l \wedge \bar{X} \quad (10)$$

$$S_b = X_b - X_b^{k^l} + S_N$$

其中, S_b^l 为 S_C 的倒序.

对于算术右移操作而言, 填充位为数据的最高有效位, 那么与逻辑右移的区别仅在于填充位“0”的个数计算. 对应的填充位中“0”的个数可以通过 $X[31]$ 的反的五位并置与 S_N 中的对应位进行与操作来计算:

$$S_N^a = X[31] \cdots X[31] \wedge S_N \quad (11)$$

则算术右移的编码预测计算公式为:

$$S_b = X_b - X_b^{k^r} + S_N^a$$

基于以上的分析, 移位操作的 Berger 编码和预测电路结构如图 5 所示. 其中 MCSA 指代图 4 所示结构的多操作数加法器.

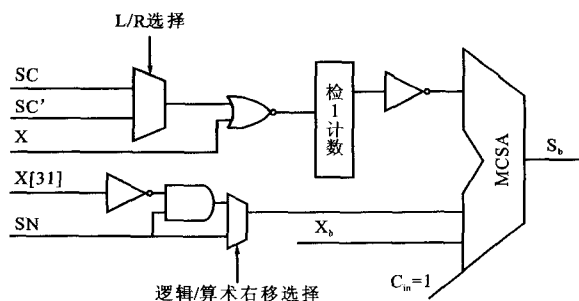


图 5 移位操作的 Berger 编码预测结构

考虑到设计的简单性以及性能, 原始的 ALU 设计采用加法器、逻辑部件和移位部件相互独立的设计, 这里不再给出. 将原始的 ALU 设计、 B_0 编码电路以及加法器、逻辑部件和移位部件对应的编码预测结构按照图 2 所示的组织结构进行集成, 即得到并行错误检测结构的 SPARC V8 结构的 ALU 设计, 该结构可以实现与数据路径并行计算的编码预测. 相比于原始的 797 个等效门的 ALU 设计, 加入并行错误检测结构后的 ALU 为 1489 个等效门, 错误检测策略的开销占原始设计的 86.83%, 但是相比于逻辑复制这样的双路逻辑策略甚至大于 100% 的实现开销, 实现了硬件资源的节省. 编码预测结构能够与数据路径并行执行, 整个结构的逻辑延时由数据路径和错误检测结构的最大逻辑延时决定, 而相比于 32 位加法器的延时来讲, 错误检测结构的延时较小, 所以该并行错误检测结构的 ALU 对于运算性能的影响只是增加结果比较及选择输出的延时, 与对偶复执这样两倍的执行时间并且附带结果比较的策略相比, 具有明显的性能优势.

5 结束语

文中分析了嵌入式处理器的在线错误检测需求, 提出了一种基于编码预测的并行错误检测策略的实现结构. 对 Berger 编码的错误检测特性进行了分析, 设计了 SPARC V8 结构的加/减、逻辑和移位运算的 B_0 编码预测结构. 基于此, 可以直观的集成为带有并行错误检测结构的 SPARC V8 结构 ALU.

参考文献:

- [1] Elena Dubrova. Fault tolerant design: an introduction [M]. Boston: Kluwer Press, 2007.
- [2] Gossel M, Graf S. Error detection circuits [M]. London: McGraw-Hill, 1993.
- [3] Michael Purser. Introduction to error-correcting codes [M]. Norwood: Artech House, 1995.
- [4] Cary Huffman W, Vera Pless. Fundamentals of error-correcting codes [M]. Cambridge: Cambridge University Press, 2003.
- [5] Clark J A, Pradhan D K. Fault injection: a method for validating computer - system dependability [J]. Computer, 1995, 28(6): 47-56.
- [6] Berger J M. A note on an error detection code for asymmetric channels [J]. Information Control, 1961, 4(1): 68-73.

(下转第 188 页)

可以在 16ms 时间内处理 1024×1024 分辨率的光照不均图像,相比 CPU 加速比可达 49 倍.随着图像尺寸的增大,并行处理的性能提升倍数也随之增大,但并非线性增长,而是随着输入图像尺寸的增大,增幅趋于平缓,性能提升稳定在 42 倍左右^[8].

5 结束语

根据同态滤波算法的高并行性,充分利用 GPU 的硬件并行架构,及高度优化的 CUFFT 库,在 GPU 上实现了并行同态滤波算法.实验结果表明,相比 CPU,算法的 GPU 实现有很大的效率提升,在本平台上,对 1024×1024 图像进行处理时,速度可达 16ms,较大程度地提高了图像增强速度,可以实时增强高分辨率光照不均图像.这种并行化加速的思想对其他具有并行性的图像处理算法也有很好的借鉴.

参考文献:

- [1] Nvidia. NVIDIA official site [EB/OL]. [2009-10-09]. <http://www.nvidia.com>
 - [2] 陈春宁,王延杰.在频域中利用同态滤波增强图像对比度[J].微计算机信息,2007,23(2/3):264-266.
 - [3] Nvidia. NVIDIA CUDA programming guide Version 1.1 [EB/OL]. [2009-10-10]. www.nvidia.com/object/cuda_home.html.
 - [4] 多相复杂系统国家重点实验室多尺度离散模拟项目组.基于 GPU 的多尺度离散模拟并行计算[M].北京:科学出版社,2009.
 - [5] 阮军,韩定定.基于 CUDA 的快速 DCT 变换实现方法[J].微电子学与计算机,2009,26(8):201-205.
 - [6] Nvidia. CUFFT_Library_1.1 [EB/OL]. [2009-10-06]. http://www.nvidia.com/object/cuda_home.html
 - [7] Frigo M, Johnson S. FFTW: an adaptive software architecture for the FFT[C]// ICASSP Conference Proceedings. USA: Washington, 1998:1381-1384.
 - [8] 伍尤富.基于离散平稳小波变换的图像同态增强[J].微电子学与计算机,2008,25(2):151-153.
- 作者简介:**
张楠 女,(1984-),博士研究生.研究方向为基于 GPU 的图像处理.
王建立 男,(1971-),研究员,博士生导师.研究方向为光电跟踪技术.
王德俊 男,(1983-),硕士研究生.研究方向为图像处理.
-
- (上接第 184 页)
- [7] Christian Martyn Jones, Satnam Singh Dlay, Raouf Gorgui Naguib. Berger check prediction for concurrent error detection in the braun array multiplier[C]// The Third IEEE International Conference on Electronics, Circuits, and Systems. Italy: Bari, 1996: 81-84.
 - [8] Leblebici Y, Ozdemir H, Kepkep A, et al. A compact high-speed (31,5) parallel counter circuit based on capacitive threshold-logic gates[J]. Journal of Solid-State Circuits, 1996, 31(8):1177-1183.
 - [9] Jien-Chung Lo, Suchai Thanawastien, Rao T R N. Concurrent error detection in arithmetic and logical operations using berger codes[C]// The 9th Symposium on Computer Arithmetic. Orlando: Florida, 1989: 233-240.
 - [10] Lo J C, Thanawastien S, Rao T R N, et al. An SFS berger check prediction ALU and its application to self-checking processor designs[J]. IEEE Journal of Computer-Aided Design of Integrated Circuits and Systems, 1992, 11(4):525-540.
- 作者简介:**
张洵颖 男,(1981-),博士.研究方向为高性能容错微处理器设计、多核同构/异构 SoC 设计.