

矩阵求逆运算的 VLSI 实现

郭春煊, 毛志刚, 谢 憬

(上海交通大学 微电子学院, 上海 200240)

摘 要: 矩阵运算广泛应用于各类电路计算中, 矩阵运算的硬件实现能够充分发挥硬件的速度和并行性。其中矩阵求逆是矩阵运算中重要的运算。为加速矩阵求逆, 研究构造出一种非常易于实现的基于心动阵列的矩阵求逆的电路结构。通过硬件描述语言 Verilog 建模, VCS 仿真及用 Design Compile 进行综合, 结果表明这种并行结构能快速实现矩阵求逆, 且运算精度较高。

关键词: 心动阵列; 矩阵求逆; LU 分解; VLSI

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2008)05-0219-05

VLSI Realization of Matrix Inversion

GUO Chun-xuan, MAO Zhi-gang, XIE Jing

(School of Microelectronics, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: Matrix operation is widely used in different kinds of circuit calculation. Hardware implementation of matrix operation can fully realize the speed and parallel of the hardware. Matrix inversion is a kind of very important matrix operation. A circuit structure with systolic array is introduced in this paper in order to accelerate the speed of matrix inversion, which is quite prone to implement. The system is described in Verilog, which is verified in VCS and compiled by Design Compile. The result shows this parallel structure can be used for fast inverse matrix calculation and get high precision.

Key words: systolic array; matrix inversion; LU decompose; VLSI

0 引 言

矩阵运算是科学与工程计算中的基本问题, 科学计算中有很大一部分可以归结为矩阵运算, 而矩阵运算的难点在于矩阵求逆。传统的矩阵求逆算法多用处理器串行计算来实现, 严重制约着计算速度的提高。为加快矩阵求逆的运算速度, 可以采用硬件实现的方法。在这一方面, 已经有很多人进行了深入研究, 特别是使用基于心动阵列的方法来构造电路结构, 使用十分广泛^[1-3]。这些论文研究的重点是通过对应算法的研究改进, 最大限度地提高资源的利用效率及时空复杂度, 而忽略电路结构本身的硬件实现难度。例如文献[2]中使用 QR 分解的方法来实现矩阵求逆, 提出的电路结构非常紧凑, 复用率非常高, 但是基本处理单元需要完成的运算非常复杂, 不但涉及比较复杂的乘除运算, 还需要进行开平方操作, 这不仅增加了硬件开销, 更主要的是大大增加了硬件实现的难度。事实上, 很多时候工程上涉及的矩阵求逆对硬件开销及速度并

没有非常高的要求。所以, 设计一种性能上略微低一点, 但是易于硬件实现的电路结构就尤为重要了。基于这种考虑, 构造了基于心动阵列的矩阵求逆电路, 虽然在复用率及时空复杂度上较上述文献中的结构略微不足, 但其基本处理单元只是涉及十分简单的加减乘除, 非常易于硬件实现。实现了关键的浮点乘除法和浮点加减法运算, 通过 EDA 工具对模型进行仿真及综合, 验证了其正确性。结果表明, 这种心动阵列能快速实现可逆矩阵的求逆运算, 且有较高的精度。

1 矩阵求逆的算法及其硬件映射

由矩阵运算中的定理知: 若 n 阶矩阵 A 非奇异, 则矩阵 A 可以分解成一个下三角矩阵 L 和上三角矩阵 U , 使得

$$A = L * U$$

这样就把一般的可逆矩阵求逆转化成了三角矩阵求逆。同时考虑到对于三角矩阵 U 、 V , 假设 $U = V^T$, 则 $V^{-1} = ((V^T)^T)^{-1} = (U^T)^{-1} = (U^{-1})^T$, 可见上三角和下三角矩阵求逆是一回事。综上所述, 只要实现 LU 分解及上三角矩阵求逆, 就可以实现一般可逆矩

收稿日期: 2007-09-23

作者简介: 郭春煊(1982-), 男, 浙江慈溪人, 硕士研究生, 研究方向为电路与系统; 毛志刚, 教授, 研究方向为电路与系统。

阵的求逆。下面将主要针对这两部分的 VLSI 实现展开讨论。

1.1 LU 分解

文中采用 Gauss 消去法对矩阵进行 LU 分解^[4]。Gauss 消去法的实质是对方程组的系数矩阵进行三角化,其一般计算过程可用下面递推式来表示:

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - m_{ik}a_{kj}^{(k-1)} \quad (1)$$

$$(k = 1, \dots, n; i = k + 1, \dots, n; j = k + 1, \dots, n + 1)$$

$$m_{ik} = a_{ik}^{(k-1)} / a_{kk}^{(k-1)} \quad (2)$$

式中 m_{ik} 称为乘数, $a_{ij}^{(k)}$ 为经过 k 次 Gauss 消去后的第 i 个方程的第 j 个系数。

上述矩阵的 LU 分解的算法可以映射为图 1 所示电路结构。在这个阵列中,有两种基本的处理单元 PE(Processing Element):方形的 PE 用于内积步的计算;圆形 PE 用于实现除法运算。矩阵 A 的元素从阵列顶部按列输入,其中 $a_{i(j+1)}$ 比 a_{ij} 晚一个时钟周期输入。计算后,所得到的上三角矩阵 U 的元素存于各个 PE 之中。当 A 的第一行元素经过到第一行 PE 时,它们被存入第一行相应 PE 之中,之后的各行元素经过第一行 PE 时,只进行相应的变换而不再存入 PE 之中。 A 的第二行元素经过第一行 PE 变换后存入第二行相应的 PE 当中,之后各行元素经过第二行 PE 时也只进行相应变换后流向下行 PE。以此类推,直至 A 矩阵第 n 行元素(经过了 $n-1$ 次变换)存入第 n 行 PE 为止。第 i 行左端的圆形 PE 依次产生消去第 i 列下三角元素的各个乘数 $m_{ij}(i = j + 1, j + 2, \dots, j = n - 1, j = n)$,然后将它们沿水平方向向右传送,参加内积步计算,最后由阵列右端输出。至整个计算结束时, PE_{ij} 中所存数据是 U 的元素 u_{ij} ,而阵列的第 i 行输出是矩阵 L 的第 $n - i$ 列元素。在 L 矩阵元素输出之后,通过控制信号的设定,紧接着输出 U 的元素,阵列的第 i 行输出的是矩阵 U 的第 i 行元素。

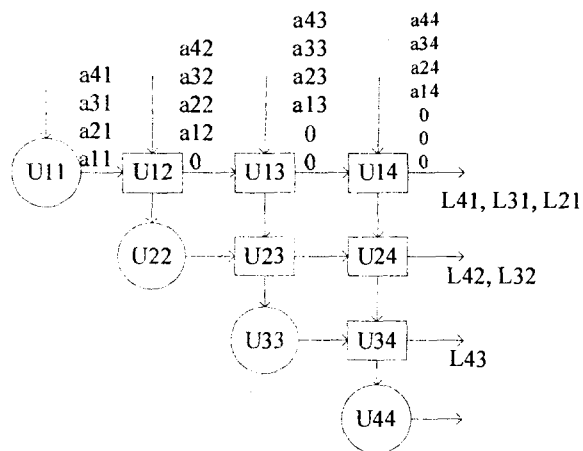


图 1 矩阵 LU 分解的心动阵列(U 元素存入 PE)

1.2 上三角矩阵求逆

上三角矩阵求逆的电路结构参考了文献[5],其结构如图 2 所示。整个阵列呈三角形,需要 $n(n+1)/2$ 个 PE(图中 $n=4$)。在这个阵列中有两类 PE:一种是六角形内积器,实现内积步运算;另一种是右上侧边上的圆形 PE,它实现除法运算,除了产生输出外,还将 v_{ij} 的值垂直向下传回阵列。如图 2 中所示,三角矩阵 A 的元素沿阵列的左上向右下方向流动,单元矩阵 E 的元素则沿阵列的左下至右上方向流动;反馈输入的 V 的元素沿阵列垂直方向向下流动,与 A 、 E 的元素在 PE 中汇合,执行内积步计算。分析可知,由右上方向输出的数据,正好是上三角矩阵 A 的逆矩阵 V 的元素。

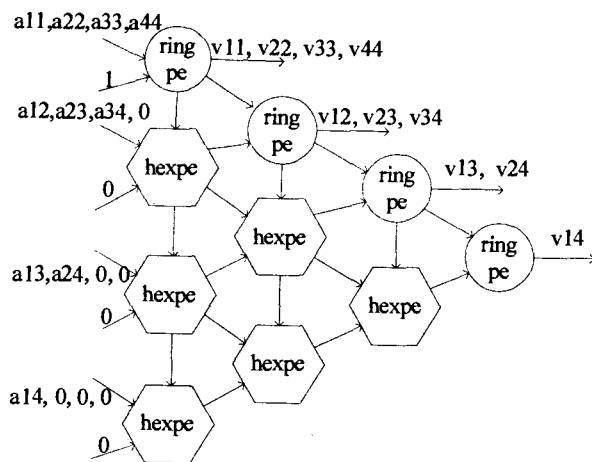


图 2 三角矩阵求逆的心动阵列结构

2 硬件单元电路的设计与实现

在硬件实现基于心动阵列的矩阵求逆设计中主要设计如下几个关键技术。

2.1 浮点数运算的实现

现在科学与工程计算中需要处理的数据多是浮点数,所以矩阵求逆是在实数范围内进行的,矩阵元素需用实数表示,因此要设计用于浮点运算的电路结构。文中,浮点数的表示采用 IEEE 754 标准^[6],其编码格式为 1 位符号位、8 位阶码和 23 位尾数来表示一个浮点数。尾数的表示范围为 $[1, 2)$, 1 是默认存在的,小数点的位置在第 22 位之前,比如 1.5 的 23 位尾数用 16 进制表示为 400000,而 1 的 23 位尾数全是 0。阶码的表示范围为 $-127 \sim 128$,值得一提的是阶码部分没有符号位,而将真正的阶码值加 127 进行编码,如 6.0 的实际阶码值应该是 2,但是编码时其阶码部分的值为 129。符号位为 1 表示该实数为负,为 0 则表示是正数。另外,如果阶码和尾数都为 0,则表示该实数值为 0。由矩阵求逆的算法可知,设计中需设计浮点乘除电

路和浮点加减法电路。

2.2 处理单元 PE 的实现

处理单元是心动阵列的基本单元,也是心动阵列设计的关键所在,下面介绍一下阵列中 PE 的实现。

LU 分解和上三角矩阵求逆的四种 PE 可见图 1 和图 2。其中方形 PE 和六角形 PE 实现的功能基本类似,其核心是实现 $\text{result} = \text{opc} - \text{opa} * \text{opb}$ 的浮点运算;而两种圆形 PE 实现的功能也类似,其核心是实现 $\text{result} = \text{opa}/\text{opb}$ 的浮点运算。上述 opa, opb, opc 及 result 都是采用 IEEE754 标准编码的 32 位浮点数。

从上面的分析不难看出,设计浮点乘除电路和浮点加减法电路是实现 PE 功能的关键。

2.2.1 浮点乘法电路实现

图 3 所示是浮点乘法的电路结构。浮点乘法的符号位和阶码求解都比较简单,为了求解其尾数,需要从 DesignWare 调用了个 24 位乘法器,根据其结果来判定尾数值。如果乘积最高位为 1,说明在编码规则下其值大于 2,需要右移 1 位,相当于截取 46~24 位作为尾数的前 22 位,相应的阶码加 1;否则,其值满足尾数区间,截取乘积 45~23 位作为尾数的前 22 位。另外这里采用冯·诺依曼舍入法,截取的尾数末位总为 1,这样可以有效避免误差的累计。

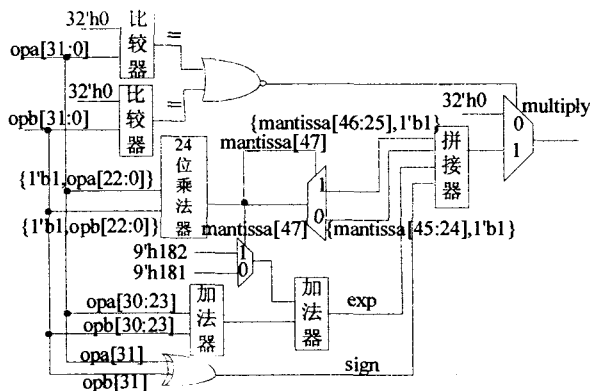


图 3 浮点乘法电路结构

2.2.2 浮点除法电路实现

图 4 所示为浮点除法电路的结构。和乘法类似,除法的难点也在于尾数的计算,为了提高计算精度,这里把被除数左移 24 位,调用 48 位除法器,根据原被除数和除数尾数的大小,截取所得商的 22~0 位或 23~1 位即可得到相应的尾数值。

2.2.3 浮点加减法电路实现

浮点加减法电路由于比较复杂,这里暂不列出电路结构图,整个电路实现了对阶、尾数运算、规格化及舍入等操作,最终得到结果。其中电路设计中值得一提的有:

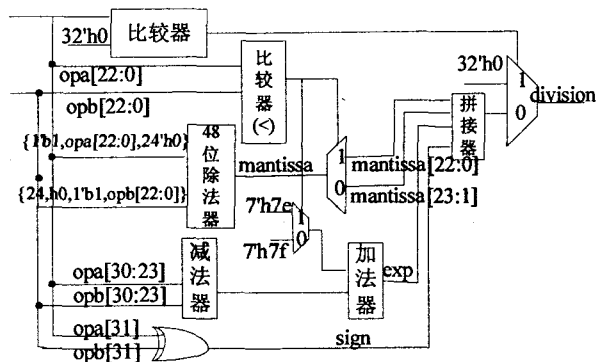


图 4 浮点除法电路结构

1) 通过设定一个标志位 mode, 电路可以实现浮点加法和减法的运算, 当 mode 为 1 时, 电路结构实现的是减法运算, 当 mode 为 0 时, 其实现的是加法运算。实现的方法是把 mode 和 opb(第二个操作数) 的符号位相异或得到实际运算中 opb 的符号位, 然后对 opa(第一个操作数) 和 opb 进行加法运算。

2) 对于两个操作数中有一个为零或者都为零的情况, 事先进行判断, 通过一个 MUX 电路把运算结果直接传递给输出, 提高效率。

3) 结构中有一个“确定左移位数的判断逻辑”的模块, 它实现尾数规格化的功能, 通过判断操作数尾数相加所得值的第一个“1”(从高位到低位) 的位置, 来确定需要左规格化的次数。另外采取了四位一组的方法, 有效地减小了面积。

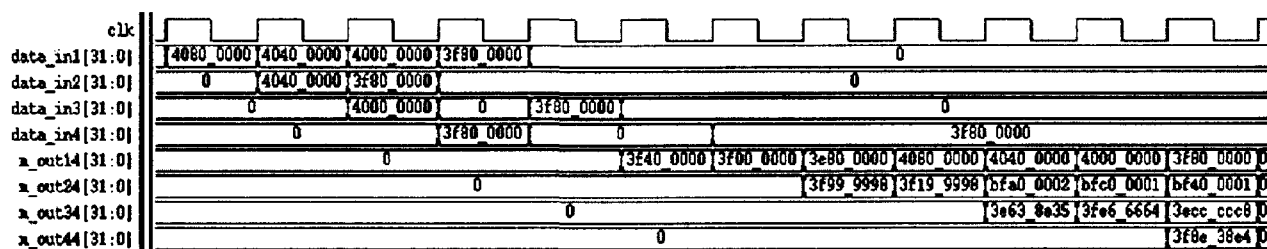
3 仿真及综合结果

文中用一个 4 阶可逆矩阵来验证求解其逆矩阵。由于整个求逆过程需要的周期比较长, 故把整个波形图分割成四个子图来说明, 如图 5 所示。输入的原始矩阵 A 为:

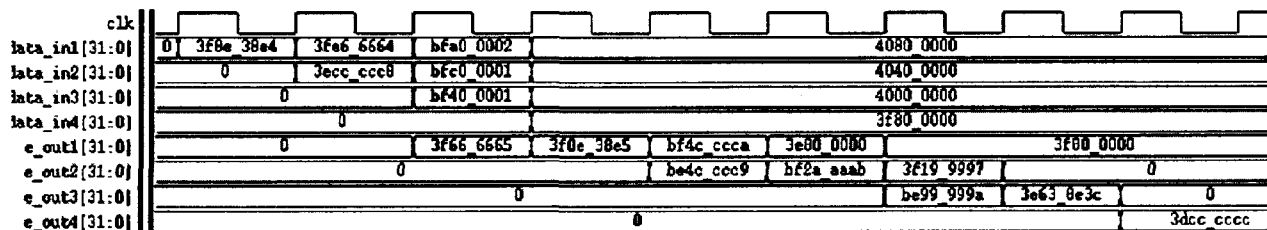
$$A = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} 40800000 & 40400000 & 40000000 & 3f800000 \\ 40400000 & 3f800000 & 0 & 0 \\ 40000000 & 0 & 3f800000 & 0 \\ 3f800000 & 0 & 0 & 3f800000 \end{bmatrix}$$

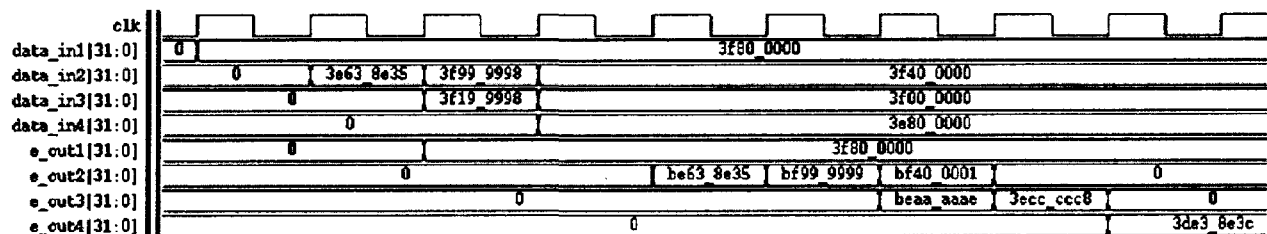
LU 分解的仿真波形如图 5(a) 所示。端口 data_in1~4 输入 A 矩阵第 1~4 列元素, 每列之间延迟一个时间单元。端口 m_out14 依次输出 L 矩阵的第一列元素及 U 矩阵的第一行元素, m_out24 端口输出 L 矩阵的第二列元素及 U 矩阵的第二行元素, m_out34 端口输出 L 矩阵的第三列元素及 U 矩阵的第三行元



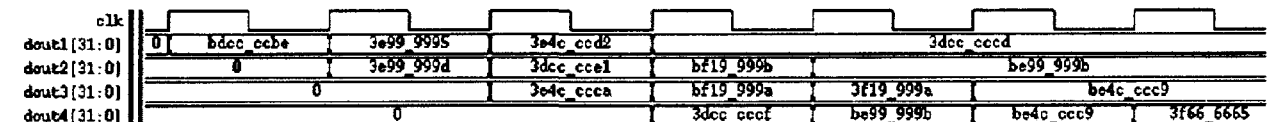
(a)



(b)



(c)



(d)

图 5 矩阵求逆的仿真波形

素, m-out44 端口输出 U 矩阵的第四行元素。值得注意的是, L 、 U 都是三角函数, 且 L 矩阵的对角线元素为 1, 对于这些已知的 0 和 1 这里没有输出。由此可得 L 、 U 矩阵为:

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 3f400000 & 1 & 0 & 0 \\ 3f000000 & 3f999998 & 1 & 0 \\ 3e800000 & 3f199998 & 3e638e35 & 1 \end{bmatrix} \cong$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 3/4 & 1 & 0 & 0 \\ 1/2 & 6/5 & 1 & 0 \\ 1/4 & 3/5 & 2/9 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} 40800000 & 40400000 & 40000000 & 3f800000 \\ 0 & bfa00002 & bfc00001 & bf400001 \\ 0 & 0 & 3fe66664 & 3ecccc8 \\ 0 & 0 & 0 & 3f8e38e4 \end{bmatrix} \cong$$

$$\begin{bmatrix} 4 & 3 & 2 & 1 \\ 0 & -5/4 & -3/2 & -3/4 \\ 0 & 0 & 9/5 & 2/5 \\ 0 & 0 & 0 & 10/9 \end{bmatrix}$$

LU 分解后分别对 U 矩阵及 L 的转置矩阵求逆, 其仿真波形分别如图 5(b)和(c)所示。

图 5(b)中, 端口 data_in1~4 输入 U 矩阵的元素, 端口 e_out1~4 依次输出第一行开始的对角线方向的逆矩阵元素, 其中 e_out1 输出对角线元素。由此可得 U 的逆矩阵为:

$$U^{-1} = \begin{bmatrix} 3e800000 & 3f199997 & 3e638e3c & 3dcccccc \\ 0 & bf4ccccca & bf2aaaab & be99999a \\ 0 & 0 & 3f0e38e5 & be4ccc9 \\ 0 & 0 & 0 & 3f666665 \end{bmatrix} \cong$$

$$\begin{bmatrix} 1/4 & 3/5 & 2/9 & 1/10 \\ 0 & -4/5 & -2/3 & -3/10 \\ 0 & 0 & 5/9 & -1/5 \\ 0 & 0 & 0 & 9/10 \end{bmatrix}$$

图 5(c)的仿真波形和图 5(b)类似,唯一有区别的是图 5(c)得到的输出是 L 的转置矩阵的逆矩阵元素,这点在之后做矩阵乘法时需要注意。

可以从图中看出,得到矩阵如下:

$$(L^T)^{-1} = \begin{bmatrix} 3f800000 & 0 & 0 & 0 \\ bf400001 & 3f800000 & 0 & 0 \\ 3ecccc8 & bf999999 & 3f800000 & 0 \\ 3de38e3c & beaaaaaa & be638e35 & 3f800000 \end{bmatrix} \cong \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3/4 & 1 & 0 & 0 \\ 2/5 & -6/5 & 1 & 0 \\ 1/9 & -1/3 & -2/9 & 1 \end{bmatrix}$$

把 L 、 U 矩阵相乘即可得到 A 的逆矩阵 C ,如图 5(d)所示。端口 dout1~4 分别输出 C 矩阵的 1~4 行,每行之间延迟一个时间单元输出,即 A 的逆矩阵为:

$$C = \begin{bmatrix} bdccccbe & 3e999995 & 3e4cccd2 & 3dcccccd \\ 3e999995 & 3dcccccl & bf19999b & be99999b \\ 3e4cccca & bf19999a & 3f19999a & be4cccc9 \\ 3dcccccf & be99999b & be4cccc9 & 3f666665 \end{bmatrix} \cong \begin{bmatrix} -0.0999999 & 0.2999999 & 0.2000000 & 0.1000000 \\ 0.2999999 & 0.1000000 & -0.6000000 & -0.3000000 \\ 0.1999999 & -0.6000000 & 0.6000000 & -0.1999999 \\ 0.1000000 & -0.3000000 & -0.1999999 & 0.8999999 \end{bmatrix}$$

整个矩阵求逆过程从输入第一个 A 矩阵元素到输出所有 A 的逆矩阵元素共花费了 38 个时钟周期。

从上面的仿真波形中可以看出,除去由浮点数精度所限带来的截断误差(仅为 10^{-7} 数量级),4 阶矩阵求逆的结果和理论所得的值完全吻合。

用 smic 的 $0.18\mu\text{m}$ 工艺对 Verilog 代码进行综合,可得整个电路的面积约为 $2.29\text{mm} \times 2.29\text{mm}$,最长路径约为 25ns ,时钟频率约为 40MHz 。由于整个求逆电路主要由 LU 分解、上三角矩阵求逆、矩阵相乘三部分构成,三者可以实现流水线操作,因此,考虑到工程上通常是大量数据需要处理,其实际的工作效率可以提

高 3 倍左右。

文中的最长路径是由 DesignWare 中调用的除法器决定的,不考虑流水线,时钟频率可以达到 40MHz ,对于一般的工程问题完全可以满足要求。文献[2]中提出的结构虽然只需要 $5n = 20$ 个时钟周期就可以完成整个求逆过程,但是由于它基本处理单元中运算操作相当复杂,必然会对其时钟频率产生一定的影响而导致整个求逆运算实际的运算时间变长。综上所述,从运算时间的角度来讲,文中的结构相比于文献[2]来说损失有限,但从硬件开销和硬件实现的难易程度来讲,文中的结构优势明显。

4 结束语

运用心动阵列硬件实现了矩阵的 LU 分解以及上三角矩阵的求逆算法,从而实现了非奇异方阵的快速求逆。结构设计中采用了心动阵列,模块内部结构规则,基本处理单元和传输控制信号简单,非常易于硬件实现,且运算精度较高,适用科学研究和工程应用中矩阵求逆的实时求解。

参考文献:

- [1] EL-Amawy A. A systolic architecture for fast dense matrix inversion[J]. IEEE Trans. Comput., 1989, C-38(3):449-455.
- [2] EL-Amawy A, Dharmarajan K R. Parallel VLSI algorithm for stable inversion of dense matrices[J]. IEEE Proceedings, 1989, 136:575-580.
- [3] Chern M, Murata T. A fast algorithm for LU decomposition and matrix inversion[C]//Proc. Int. Conf. on Parallel Processing. Columbus, OH:[s.n.], 1983:79-86.
- [4] 李庆扬,王能超,易大义. 数值分析[M]. 武汉:华中科技大学出版社,2004.
- [5] 陈国良. 并行算法的设计与分析[M]. 北京:高等教育出版社,1994.
- [6] Kahan W. IEEE Standard 754 for Binary Floating-Point Arithmetic[D]. Berkeley, CA:Elect. Eng. & Computer Science University of California, 1996.

(上接第 218 页)

参考文献:

- [1] 王 岚,乐毓俊. 计算机自动推理与智能教学[M]. 北京:北京邮电大学出版社,2005:306-308.
- [2] 王 陆,王美华. ITS 系统中基于关系模型的知识表示[J]. 北京大学学报:自然科学版,2000(9):659-664.
- [3] 刘以安,管捍权. 学生模型库的设计思想和实现方法[J]. 华东船舶工业学院学报,1999,13(5):38-40.
- [4] 陈 颖,何钦铭,王申康. 基于超文本结构的智能计算机辅助教学系统[J]. 计算机研究与发展,1998(5):442-446.
- [5] 张 跃,邹寿平,宿 芬. 模糊数学方法及其应用[M]. 北京:煤炭工业出版社,1992:146-212.
- [6] 杨 卉,王 陆. 智能教学系统中两层动态学生模型的研究[J]. 电化教育研究,2005(1):72-75.
- [7] 中科永联高级技术培训中心. MVC[EB/OL]. 2006. <http://www.itisedu.com/phrase/200603071552065.html>.