

快速开方算法在微控制器上的实现

石一辉, 易 攀, 张承学

(武汉大学 电气工程学院, 湖北 武汉 430072)

摘 要:介绍了两种微控制器快速开方算法:改进牛顿-拉夫逊算法和模拟手算开方算法。前者是以牛顿-拉夫逊算法为基础的一种改进算法;后者是模拟手算开方过程实现开方的微控制器算法,这两种算法都具有较高的开方速度和计算精度。笔者以 32 位数开方为例,详细介绍了这两种算法用汇编语言实现的过程,并给出算法实现的流程图,最后根据两种算法的特点和实际运算时间,总结了两种算法的优缺点。

关键词:开方;改进牛顿-拉夫逊算法;模拟手算开方算法;汇编语言

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2007)04-0080-03

Rapid Squiring Algorithm Realized on Micro-Controller

SHI Yi-hui, YI Pan, ZHANG Cheng-xue

(School of Electrical Engineering of Wuhan University, Wuhan 430072, China)

Abstract: Introduces two kinds of algorithms on how to get square roots quickly on micro-controller: advanced Newton-Rough algorithm and the simulation-calculation squiring algorithm. The former is an improved algorithm based on the traditional Newton-Rough algorithm, while the latter is a micro-controller squiring method by simulating the manual squiring algorithm. These two algorithms all have improved the calculating speed and precision. In this article, taking a 32-bit data as an example, introduces the realizing process by assembly language in details, and shows the flow chart about these two algorithms. Finally, summarizes the strong and weak points in accordance with the characteristics and the actual operation time of these two algorithms.

Key words: squiring; advanced Newton-Rough algorithm; simulation-calculation squiring algorithm; assembly language

0 引言

在工程的实际应用过程中^[1]常常会碰到用单片机开方的问题。目前,对于具有强大计算能力的 DSP 等微处理器来说,实现快速开方已不是难题,然而对于在微控制领域占据主流的 8 位单片机而言却是一个不可忽视的问题。如果应用传统的牛顿-拉夫逊算法实现数值开方将会消耗掉大量的时间,这对于实时性要求较高的系统将会是致命的缺陷。笔者介绍的这两种方法能够极大地缩短计算时间,很好地解决这一难题。

改进牛顿-拉夫逊算法是常规牛顿-拉夫逊算法的改进形式,常规牛顿-拉夫逊算法的计算时间主要消耗在斜率计算和多次迭代的过程中,其中,斜率计算时又以除法运算消耗的时间最多。笔者针对这几个难点提出了初值选择算法和精简除法算法两种改进措施。牛顿-拉夫逊算法的迭代过程中,迭代初值越接

近真值,迭代算法的收敛速度越快,选择合适的迭代初始值能够有效地减低计算的迭代次数;精简除法算法主要是对除法计算程序本身作了一定的简化,提高了除法计算的效率。

模拟手算开方算法是手算开方方法的计算机实现。它完全独立于牛顿-拉夫逊算法,没有复杂的除法程序,只是用加、减法和简单的移位操作,实现开方算法的全过程。对于 32 位长度的二进制数,模拟手算开方算法固定循环 16 次。

这些算法均已用汇编语言在 C8051F020 的单片机上实现,根据实际的实验效果,开方算法的计算时间大幅度缩短,有效地提高了单片机的计算效率。

1 改进牛顿-拉夫逊算法

1.1 基本算法^[2]

开方算法^[3]即为求解方程 $X = \sqrt{Y}$ 的过程,问题等价于求解: $Y = X^2$ 。

根据牛顿-拉夫逊算法的切线方程:

$$Y = Y'(X_{k+1} - X_k) + X_k^2 =$$

收稿日期:2006-07-21

作者简介:石一辉(1982-),男,湖北武汉人,硕士研究生,研究方向为电力系统自动化;张承学,教授,博士生导师,研究方向为电力系统自动化。

$$2X_k(X_{k+1} - X_k) + X_k^2 \quad (1)$$

$$\text{所以: } X_{k+1} = (X_k + Y/X_k)/2 \quad (2)$$

设被开方数 $Y = C$ 为迭代初值,则:

$$X_{k+1} = (X_k + C/X_k)/2 \quad (3)$$

式中 X_k 为第 k 次近似根, C 为被开平方数。

1.2 改进算法

1.2.1 算法分析

在公式(3)中,如果不论 C 值的大小,以 $X_0 = C$ (或 $C/2$) 为首次根,当 C 值很大(如 $X > 10^{10}$) 或很小($X < 10^{-10}$) 时要迭代 20 次以上才能达到根 9 位有效十进制数据。若 $X = 10^2$ (或 $X = 10^{-2}$),也要迭代 7 次才能达到以上精度。迭代过程的特点是近似根与精确根相差较大时,逼近速度很慢,迭代时间主要浪费在这一阶段上,一旦接近精确根,逼近速度大大加快。所以为牛顿迭代法选择一个合适的初始值将会大大地提高其收敛的速度。改进牛顿-拉夫逊算法正是针对初值选择这一问题进行了研究。

1.2.2 初值选择

设被开方数 $C = 2^m \cdot T$, m 为阶数, T 为尾数。这时 $0.5 < T < 1$ 。

则有:

$$\sqrt{C} = 2^{m/2} * \sqrt{T} \quad (m \text{ 为偶数时}) \quad (4)$$

式中: $0.7071 < \sqrt{T} < 1$ 。

此时选取 $X_0 = 2^{m/2} - 1$ 作为迭代初值,比较接近平方根值。

$$\sqrt{C} = 2^{(m+1)/2} * \sqrt{\frac{T}{2}} \quad (m \text{ 为奇数时}) \quad (5)$$

式中: $0.5 < \sqrt{\frac{T}{2}} < 0.7071$

此时选取 $X_0 = 2^{(m+1)/2} - 1$ 作为迭代初值,比较接近平方根值。

1.2.3 精简除法算法

目前,许多微控制器^[4]都没有多字节除法指令,为了实现多字节除法运算,并保证开方计算的速度,这里提出一种除法运算的简化算法——精简除法算法。

精简除法算法是对传统除法运算程序作了一定的简化,由于在开方运算过程中涉及到的除法运算都只是 4 字节除以 2 字节,且商不超过 2 字节的情况,所以在除法运算过程中可以精简掉部分判断和移位操作,但精简算法本身也只能适用于 4 字节除以 2 字节,且商不超过 2 字节的情况。精简除法算法的实现过程如下:

其中 R2R3R4R5 为被除数、R6R7 为除数、R4R5 为商、R2R3 为余数, F0 是软件标志位。

(1)被除数左移一位(以 0 补最低位,最高位进入 F0)。

(2)R2R3 减去 R6R7。

(3)若 F0 为 0,则向下执行,否则 R5 + 1, R2R3 为减后的差,且跳转到(6)。

(4)若相减有借位,则向下执行,否则 R5 + 1, R2R3 为减后的差,且跳转到(6)。

(5)恢复 R2R3 为减前的数。

(6)若移位 16 次,则向下执行,否则跳转到(1)。

(7)输出结果:商 R4R5、余数 R2R3。

1.2.4 流程图

改进牛顿-拉夫逊算法流程图如图 1 所示。

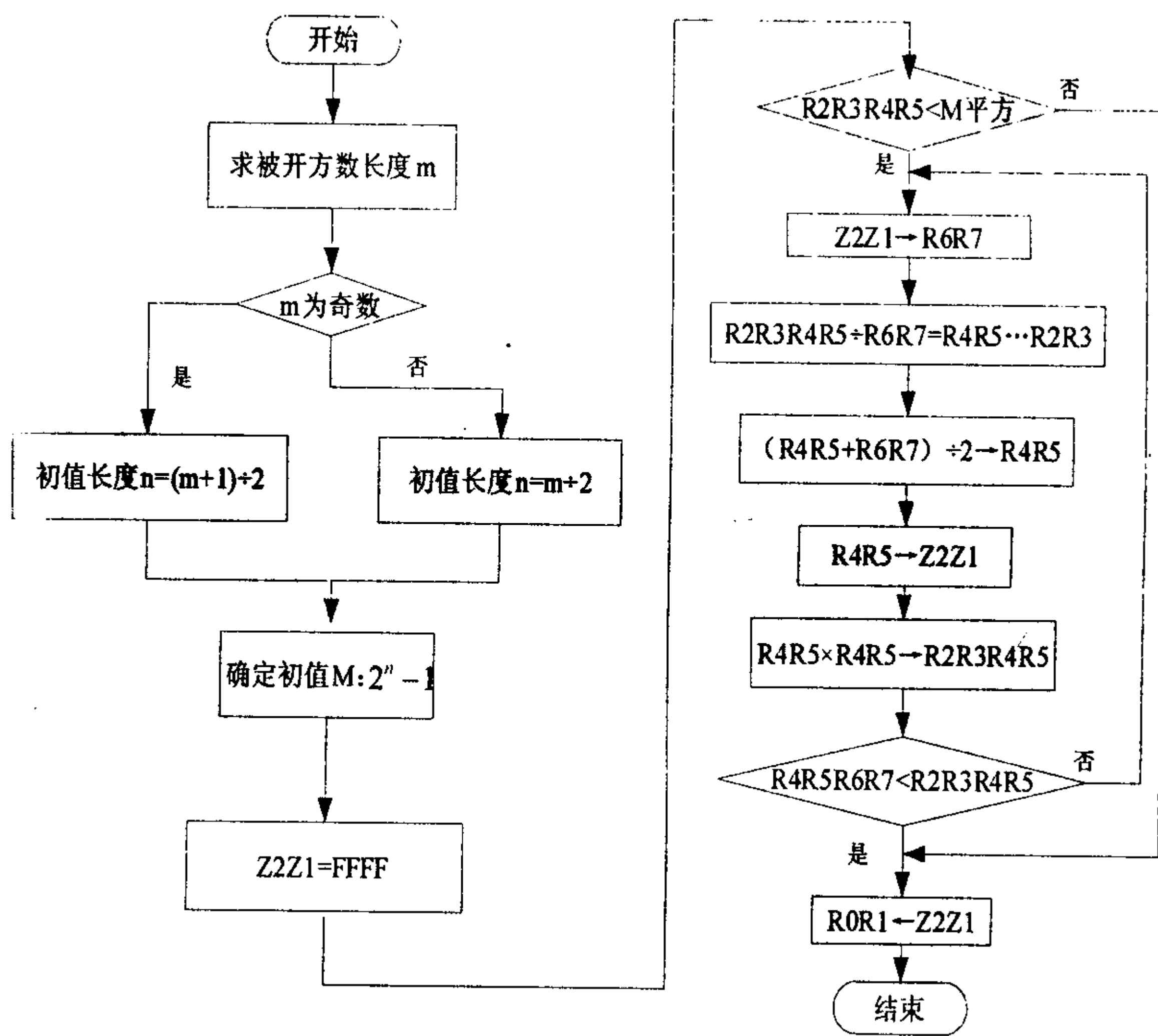


图 1 改进牛顿-拉夫逊算法流程图

2 模拟手算开方算法

2.1 基本原理

模拟手算开方算法的基本原理实际上是一个逐步试探、逐次逼近的过程,用到的主要算法为移位和减法。由二进制平方计算的规律可知 16 位的二进制数平方后位数不超过 32 位。若设被开方数 $A_0A_1 \cdots A_{31}$ (A_0 为最高位, A_{15} 为最低位) 为一个 32 位的二进制数,则开方后,根不会超过 16 位,故可记根为 $G_0G_1 \cdots G_{15}$ (G_0 为最高位, G_{15} 为最低位),并且被开方数每变动两位,根才会改变一位,所以在试探、逼近的过程中被开方数是两位两位进行处理的。

不破坏整体的和弦和谐效果,因为 G 后也迅速解决到了 E 音。整体来看图 3 这个片段,它的首尾并没有出现 A 或 B 音,在音的跨度上也没有超过规则的限定,因此这个小节的片段还是很好地满足了创作规则的要求。同样经过分析图 4,5 所示的片段也可发现相同的效果,所以这几个片段都具有较高的适应度值。



图 3 动机片段一



图 4 动机片段二



图 5 动机片段三

4 结束语

通过对遗传算法创作动机的研究可以发现,遗传算法对动机创作的应用是可行的。应用遗传算法实现乐曲的效果是取决于创作中音乐知识的表示的,也就是规则的设立以及适应度函数的选取。如果规则制定的越精细,那么得到的乐曲也就会越精确。但是在用

遗传算法创作动机时仍然存在一定的局限性:

- 1)遗传算法的编码存在不规范以及不准确性,由于音乐信息的复杂性,导致在对乐曲进行编码时,很可能会出现编码不确切的现象。
 - 2)尽管算法是根据设定的音乐知识进行创作,但由于动机创作的变化性,有时并不完全拘泥于规则,故完全按照设定的规则进行创作的结果有时会略显单调呆板。
 - 3)由于遗传算法是一个启发式的搜索算法,因此它不一定能够找到最优解。
- 尽管利用遗传算法创作动机片段仍然存在上述不足,但它仍是对机器作曲进行的有益的尝试,而且还存在很大的发展空间,特别是怎样将情感信息加入到算法创作动机的过程中,仍然需要进行进一步的探索。

参考文献:

[1] 陈 明. 基于遗传算法的优化计算[J]. 软件学报,1998,9 (11):876-879.

[2] 周昌乐. 心脑计算举要[M]. 北京:清华大学出版社,2003.

[3] 柏西·该丘斯. 应用对位法(上卷)创意曲[M]. 北京:人民音乐出版社,1986.

[4] 冯 寅,周昌乐. 算法作曲的研究进展[J]. 软件学报,2005,16(6):209-215.

[5] 张英俐,刘 弘,李少辉. 遗传算法在作曲中的应用[J]. 计算机应用研究,2005(11):143-145.

[6] 李重光. 音乐理论基础[M]. 北京:人民音乐出版社,1962.

(上接第 82 页)

表 1 开方结果表

被开方数	1H	3H	212H	2487H	267E FFH	FDF7 EFFH	FFFFF FFFH
改进 N—L 算法	1H	1H	17H	60H	634H	3FBEBH	FFFFFH
手算开方算法	1H	1H	17H	60H	634H	3FBEBH	FFFFFH

表 2 开方耗时表

(单位:以定时器计数值为单位)

被开方数	1H	3H	212H	2487 H	267E FFH	FDF7 EFFH	FFFFF FFFH
改进 N—L 算法	3C9H	3FDH	81H	B5H	B0H	43H	3FH
手算开方算法	97H	97H	97H	97H	97H	97H	97FH

4 结束语

改进牛顿-拉夫逊算法和模拟手算开方算法各有优缺点。共同的优点是:相对传统的牛顿-拉夫逊算法,这两种算法在保证计算精度不变的前提下,提高了计算的速度。不同的是:改进牛顿-拉夫逊算法本身

的计算速度与被开方数有关,不易控制开方计算的时间,但该算法用到的内存资源较少,而模拟手算开方算法用到的内存资源较多,但是算法的计算时间固定,便于控制。根据不同的应用场合,有针对地选择合适的算法可以大幅度提高开方计算的速度。

参考文献:

[1] 王明芳,汪 梅,淮文军,等. 基于 AT89S52 的便携式智能室内环保监控仪[J]. 计算机技术与发展,2006,16(6):57-59.

[2] 涂时亮,姚志石. 单片微机 MCS-96/98 实用子程序[M]. 上海:复旦大学出版社,1991.

[3] 彦庆津. 数值分析[M]. 北京:北京航空航天大学出版社,1999.

[4] 丁元杰. 单片微机原理及应用[M]. 北京:机械工业出版社,1999.

[5] 潘琢金,施国君. C8051Fxxx 高速 SOC 单片机单片机原理及应用[M]. 北京:北京航空航天大学出版社,2002.