

银河 TS-1 向量处理及其在量子计算中的应用

宋 辉, 陆洪毅, 鲁建壮, 戴 葵, 王志英

(国防科技大学计算机学院, 湖南长沙 410073)

摘 要: 量子计算是近年来新出现的计算技术, 具有非常好的发展前景, 目前量子计算的研究大多通过在经典计算机上模拟实现. 向量运算是量子计算模拟的基础, 增强向量运算能力, 可以提高量子计算模拟的性能和效率. 本文以银河 TS-1 高性能嵌入式微处理器的中的向量处理技术为基础, 提出了在向量微处理器上实现量子计算的基本方法, 并对其进行了性能模拟, 分析了微处理器向量部件设计对量子计算模拟的影响以及改进方法.

关键词: 量子计算; 微处理器; 向量处理器

中图分类号: TP302 **文献标识码:** A **文章编号:** 0372-2112 (2002) 12-1828-04

The Vector Processing of YH TS-1 and Its Application on Quantum Computing

SONG Hui, LU Hong-yi, LU Jian-zhuang, DAI Kui, WANG Zhi-ying

(National University of Defense Technology, School of Computer, Changsha, Hunan 410073, China)

Abstract: Quantum computing is a new technology developed in recent years. It has great computing power, but the research is by simulation on classic computer. The base of quantum computing simulation is vector operation. To enhance the power of vector operation can improve the performance and efficiency of quantum computing simulation. YH TS-1 is a 32-bit embedded microprocessor designed by School of Computer in National University of Defense Technology. This microprocessor has been integrated with classic vector operation technology. This paper introduces the vector operation architecture in YH TS-1 microprocessor, giving a basic implementation of quantum computing on the YH TS-1 microprocessor. We test the performance enhancement by resolving the Deutsch-Jozsa problem on scalar processor and YH TS-1 with vector component. At the end of this paper, how the design of vector component impacts on the performance of quantum computing simulation and its improvement methodologies are discussed.

Key words: quantum computing; micro processor; vector computer

1 引言

量子计算机(Quantum Computer)^[1]是近来研究和发展的—种前景非常好的新型计算机技术. 量子计算机和当前计算机相比, 具有极大的并行性和惊人的加速比. 可以用于空气动力学、有限元分析、天气预报、石油勘探等大型问题的计算和数据处理. 目前人们已经可以使用核磁共振(NMR)技术成功地建立 7 个量子位的量子系统^[1]. 但这些量子计算系统还仅仅处于实验阶段, 无法投入实际应用, 所以许多量子计算算法的研究还是依赖于在经典计算机上的模拟实现完成.

研究量子计算特点不难发现, 在量子计算理论中, 量子寄存器(Quantum Register)和量子门(Quantum Gate)的操作都是基于矩阵和向量的运算. 因此研究向量计算与量子计算的结合, 显然会提高量子计算模拟的性能.

银河 TS-1 微处理器^[8]是国防科学技术大学计算机学院设计的 32 位 RISC 高性能嵌入式微处理器, 该嵌入式微处理器采用在现有的银河 TS-1 流水线上进行向量化扩充的方法,

充分利用已有的功能部件以及控制逻辑, 在较少的硬件开销下, 实现了比较完整的向量处理功能. 在该微处理器上进行量子计算模拟, 已取得了许多有意义的结果.

本文首先介绍了银河 TS-1 微处理器中的向量处理技术, 以及在该处理器上进行量子计算的模拟方法, 最后并对其模拟实现的性能进行了测试和分析, 并对模拟性能和进一步提高的方法进行了深入探讨.

2 银河 TS-1 嵌入式微处理器向量处理技术

在银河 TS-1 中, 有四个向量寄存器(V0、V1、V2、V3)以及一个向量长度寄存器(VLR). 每个向量寄存器包含 64 个向量分量, 每个向量分量为 32 位长. 向量长度寄存器用于决定后续向量操作中有效向量分量的个数.

银河 TS-1 支持如下三种类型的向量操作:

向量访存指令:用于在向量寄存器和存储器之间传送数据. 向量访存指令是通过 Load/Store 多个寄存器相类似的控制实现的, 即在译码器中设置有限状态机, 控制多个周期的

收稿日期: 2001-11-22; 修回日期: 2002-03-20

基金项目: 国家自然科学基金(No. 60173040); 国家“八六三”高技术研究发展计划(No. 863-SOC-Y-3-2-1)

向量访存指令的执行,所操作的向量分量的个数由向量长度寄存器 VLR 决定。

向量运算指令:用于在一个、两个、三个向量或者一个向量与一个标量之间进行运算,并将结果存入目标向量寄存器中。目前银河 TS-1 支持的向量运算操作包括:向量加、向量减、向量分量乘、向量标量加、向量标量乘、向量乘加等。所操作的向量元素的个数由向量长度寄存器 VLR 决定。

向量长度寄存器操作:在向量长度寄存器 VLR 与一个通用标量寄存器之间互相传送数据,用于决定后续向量操作的有效向量元素的个数。也可以将一个立即数传递到 VLR 中。

同时,银河 TS-1 的流水线支持标量的乘法和加法的链接操作(MAC),可以很方便的支持三个向量之间的乘加链接运算。

为了提高银河 TS-1 微处理器向量处理的效率,在相应的向量化编译器设计中,已经实现了一种基于汇编代码的向量化技术,可以有效地对汇编代码中的循环体进行向量化,并将循环体转换为向量指令和标量指令混合执行的指令序列。

根据嵌入式微处理器的特点,结构简单、功耗低等。在向量处理设计中确立了以下几个设计目标:

- * 尽量不引入过多的信号。
- * 尽量采用银河 TS-1 中的现有部件,从而减少了面积的开销。
- * 尽量不改动银河 TS-1 的流水线结构,使得向量的实现能够更好的融合于整个流水线结构中,使之同时支持标量和向量处理。
- * 采用与标量相似的微操作及其控制方式。

为了达到上述目标,在设计中增加了流水线中一些与向量处理相关的流水段的功能以及一个向量寄存器文件。共用执行段的 ALU 部件和乘法部件。向量处理的流水结构如图 1 所示。

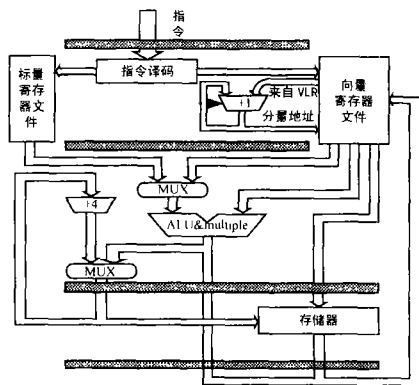


图 1 向量处理的流水结构

银河 TS-1 中的向量访存指令、向量运算指令都是多周期指令,它们都是采用在译码器段设计相应的有限状态机来实现的,由于多重状态机的嵌套比较容易产生毛刺,使得信号不够稳定,并且控制比较复杂,因此使用了一个全局的状态机。向量处理的状态转换如图 2 所示。

图 2 中的开始状态是图 2 中不同类型的指令处理的出发

点。向量处理部分的访存操作和标量乘法处理是多周期的,其余的每个状态都是单周期的,并且每向前推进一个状态,相当于一个 CPU 周期。最终的下一条指令处理和开始状态的意义相同。

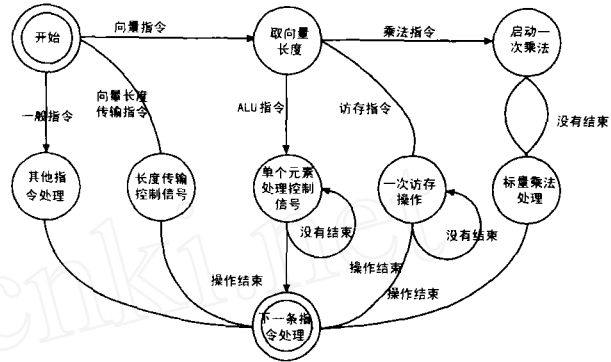


图 2 译码段向量处理状态转换图

向量访存指令中访存的向量元素的个数由一个计数器(VLR)来控制。向量在存储器中是按照地址递增的方式连续存储的,而在向量寄存器(V0、V1、V2、V3)中,则是从第一个分量开始存储的。由于使用单一的寻址模式,因此控制逻辑要比数据块访存指令的控制逻辑简单。由于操作的向量元素个数是可变的(1 个到 64 个),所以在每一次发出访存请求之前,都需要检查计数器的值,是否达到向量长度寄存器的值,如果是,则在发出最后一次访存请求之后,设置微控制信号,表明向量访存指令结束;否则,发出一次访存请求,计数器加一。

向量运算指令的设计,采用重复发出向量运算操作的方法实现。在译码器段重复的发出相类似的控制信号,完成向量运算指令的执行。向量运算指令操作的元素个数,由向量长度寄存器 VLR 的值决定。

3 量子计算

一个量子位(Qubit)就是一个微观粒子构成的两基态(0, 1)系统。在信息表示上与经典位相同。不同之处在于它可以处在两个基态的叠加态^[2]。根据量子力学规定使用|0>表示基态 0, |1>表示基态 1,那么量子位的叠加态就表示为

$$|\psi\rangle = \omega_0|0\rangle + \omega_1|1\rangle$$

其中 $\omega_i (i=0,1)$ 表示基态 i 的概率幅。 $|\omega_i|^2$ 表示系统处于基态 i 的概率,并且 $\sum |\omega_i|^2 = 1$ 。

量子寄存器的状态描述量子计算机的状态。一个 n 位的量子寄存器能够同时保存 2^n 个状态信息。在经典计算机上使用向量描述量子位和量子寄存器,如下表 1 和表 2 所示。不仅需要一组向量表示所有可能的状态,还要有一组对应的向量

表 1 状态为 1 的向量表示

取值	向量表示	概率幅	概率
0	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	0	0
1	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	1	1

表示状态的概率幅, 概率幅的平方就是该状态出现的概率^[2].

表 2 叠加态的向量表示

取 值	向量表示	概 率 幅	概 率
0	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{2}$
1	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{2}$

量子计算就是将一系列的量子门作用到量子寄存器上^[3], 修改量子寄存器的状态, 达到所需结果, 也即:

$$|\psi\rangle = U_m \otimes U_{m-1} \otimes \dots \otimes U_1 \otimes |0 \dots 0\rangle$$

其中 $|\psi\rangle$ 为系统状态, U_i 为量子门, $i=0, 1, \dots, m$.

量子门 U_i 对量子寄存器进行操作, 实现量子态的转换. 由于在复向量空间中, 任何线性变换都可以用一个矩阵来描述. 所以在量子计算中, 使用么正矩阵来描述相应的量子门. 量子门阵列由基本量子门构成, 基本量子门有受控非门、恒等变换门、非门、相位移动门等^[4]. 对长度为 n 的量子寄存器进

行变换的量子门用一个 $2^n \times 2^n$ 的矩阵描述.

在计算完成后, 要对系统状态进行测量, 提取出结果. 所谓测量, 就是将系统的状态投影到某个基态上, 提取出相应的概率幅. 量子算法中必须保证最后所需结果的概率最大. 也即:

$$|\psi_i\rangle = U_p |\psi\rangle$$

其中 U_p 为投影算子, $|\psi\rangle = \sum_j \omega_j |\psi_j\rangle$, $|\psi_i\rangle$ 为最后测量得到的状态, $\omega_i = \max\{\omega_j\}$.

4 用银河 TS-1 进行量子计算模拟

这里通过对量子计算中的典型算法 Deutsch-Jozsa 问题^[5-7]的量子算法进行模拟来评价银河 TS-1 微处理器向量处理对量子计算模拟的支持, 并对影响量子计算性能的因素和改进方法进行分析.

在银河 TS-1 上进行量子算法模拟, 绝大部分的时间花费在量子门的运算上. 在 Deutsch-Jozsa 算法中, 量子门 $H^{(n)}$ 对向量寄存器的操作时间占到了 85% 以上, 并且随着量子位的增加, 该量子门对算法运行时间的影响越大 (如图 3 所示).

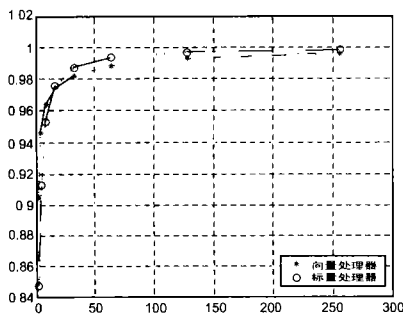


图 3 量子门占整个运算时间的比例

这是因为在量子计算机上量子门对数据处理具有并行性, 一次就可以对量子寄存器中所有状态计算. 但是在经典处理器上, 由于不具备这样的物理机制, 因此在标量处理器上是通过双重循环完成; 在银河 TS-1 处理器上, 由于有向量部件的支持, 所以其算法实现如图 4 所示.

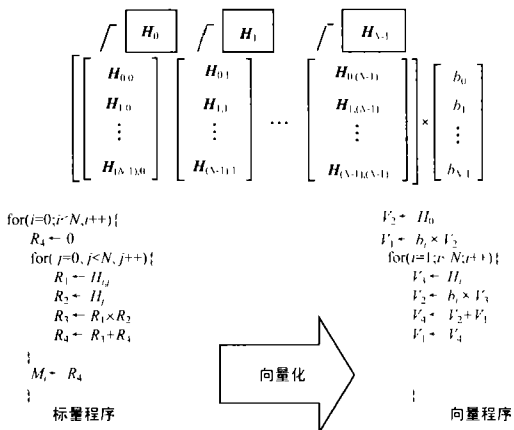


图 4 量子门 $H^{(n)}$ 操作的矩阵说明、标量程序和向量程序

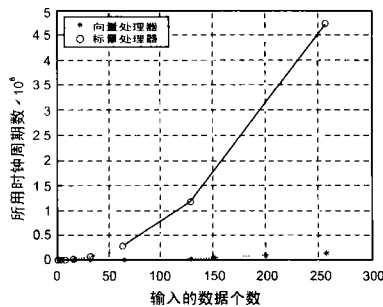


图 5 不同处理器运行量子算法的时间

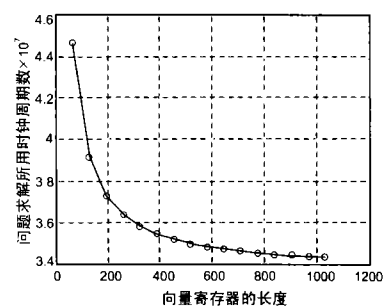


图 6 向量寄存器长度对性能的影响

5 性能评测

图 5 是 Deutsch-Jozsa 量子算法在银河 TS-1 微处理器上用标量处理和向量处理做模拟时所需的时间 (以时钟周期数为单位). 银河 TS-1 处理器标量处理功能部件时钟周期数与银河 TS-1 微处理器对应的功能部件时钟周期数相同. 银河 TS-1 微处理器的向量处理采用了流水和链接技术.

从图 5 中可以看出: 当模拟的量子位超过 4 位时, 标量处理的求解问题的时间明显超过了向量处理的求解时间, 并且随着模拟量子位的增加, 求解的时间出现了巨大差异. 使用银河 TS-1 处理器求解 Deutsch-Jozsa 算法, 其加速比将随着量子位的增加而增加, 其极限为 36. 可以看出, 银河 TS-1 处理器采用了向量处理技术后, 在求解量子计算这类问题上表现出了明显的优势.

通过模拟结果可以看出, 在模拟和评价多量子位系统的量子算法时, 采用具有向量支持的处理器是一条现实的解决方向.

为了进一步改进性能, 研究了向量寄存器长度对量子计算模拟的影响, 并作了模拟测试.

由于工艺和价格因素, 一般来说, 向量处理器中向量寄存

器的长度只能是一定的长度.当向量长度超过了向量寄存器的长度后,就必须使用向量开采的技术将其分段,然后以段为单位从存储器中调入并进行处理.这样就会引入很多的控制开销,从而影响整个计算的运行时间和性能.图 6 是模拟银河 TS-1 处理器中向量寄存器长度变化对 Deutsch-Jozsa 算法性能的影响.算法中量子寄存器长度为 10.

从图 6 中可以看出向量寄存器的长度从 64 增长到 256 的时候,程序的运行时间急剧下降.此后的运行时间变化趋向平稳.所以在量子计算模拟中,必须根据应用的特点、计算规模选择合适的向量处理器.向量寄存器长度过小,会严重影响处理器的性能.但一味增加向量寄存器的长度,也会大大增加整个芯片的硬件开销.

6 结束语

量子计算具有重要的理论研究意义,但在目前,只能通过模拟的方法进行量子计算的算法、应用和系统结构的研究.而向量计算机或者具有向量支持的微处理器的结构对量子计算模拟具有极大的优势,在此基础上进行研究,一方面有助于量子计算及其模拟实现的研究,另一方面又可以为今后体系结构的研究提供新思路.

参考文献:

- [1] IBM Research News Release. IBM's test-tube quantum computer makes history[DB/OL]. December 19, 2001, <http://www.research.ibm.com/resources/news/20011219-quantum.shtml>.
- [2] R Feynman, R Leighton, M Sands. The Feynman Lectures on Physics III [M]. Boston: Addison-Wesley Publishing Company. 1965.
- [3] R Feynman. Quantum mechanical computers [J]. Foundations of Physics. 1986, 16(6): 507 - 531.
- [4] A Barenco, C Bennett, R Cleve, et al. Elementary gates for quantum computation[J]. Physical Review A, March 1995.

- [5] D COLLINS, W C HOLTON, K W KIM. Deutsch-Jozsa algorithm as a test of quantum computation[J]. Phys Rev A, 1998, 58(3): R1633 - 6.
- [6] D DEUTSCH. Quantum theory, the church-turing principle and the universal quantum computer[A]. Proc Roy Soc[C]. London: Ser. A, 1985, 400: 97 - 117.
- [7] D DEUTSCH, R JOZSA. Rapid solution of problems by quantum computer[A]. Proc Roy Soc[C]. London: Ser. A, 1992, 439: 553 - 8.

作者简介:



宋 辉 男, 1974 年生于河南许昌市. 1999 年进入国防科大计算机学院攻读博士学位, 主要研究方向: 新型计算机系统结构, 量子计算.



陆洪毅 男, 1974 年生于广西柳州. 博士研究生. 主要研究方向: 计算机系统结构、高性能微处理器设计、向量微处理器设计.

鲁建壮 男, 1978 年, 博士研究生. 主要研究方向: 计算机系统结构、高性能微处理器设计、向量微处理器设计.

戴 葵 男, 1968 年生, 博士, 副教授, 主要从事先进计算机体系结构、微处理器设计技术研究.

王志英 男, 1956 年生于山西长治, 博士生导师, 教授, 主要从事先进计算机体系结构、微处理器设计技术研究.