

基于宏指令集的专用汇编器的研究与实现

许娜, 张晓彤, 王沁, 梁静

(北京科技大学信息工程学院, 北京 100083)

摘要: 针对将复杂算法转化为硬件电路难度很大的问题, 设计一种专用汇编器, 可将由宏指令组成的汇编语言自动转化为二进制代码表示的机器语言, 并根据宏指令集计算机理论的系统结构, 将二进制代码组织成 VHDL 语言描述的状态机输出, 便于进一步的硬件优化设计。性能测试结果证明, 该汇编器支持从汇编语言到硬件电路设计语言的自动转化, 可较大地提高用硬件电路实现复杂算法的研发效率。

关键词: 宏指令; 汇编语言; 汇编器; 宏指令集计算机; 硬件电路

Research and Implementation of Specific Assembler Based on Macroinstruction Set

XU Na, ZHANG Xiao-tong, WANG Qin, LIANG Jing

(School of Information Engineering, University of Science and Technology Beijing, Beijing 100083)

【Abstract】 Aiming at the problem that it is difficult to implement complex algorithm to hardware circuit, this paper designs a specific assembler based on macroinstruction set model. It can automatically translate assembly language composed of macroinstructions into binary code. According to structure of Macro instruction Set Computer(MISC) system, the binary code can be organized into the state machine output in the form of VHDL. Performance test proves that the assembler supports auto-translation from assembly language to hardware circuit design language, and it can improve the developing efficiency by using hardware circuit to implement complex algorithm.

【Key words】 macroinstruction; assemble language; assembler; Macroinstruction Set Computer(MISC); hardware circuit

本文通过对宏指令集计算机 (Macroinstruction Set Computer, MISC) 系统结构的研究, 设计了一种可将宏指令组成的汇编语言自动转化成高级硬件设计语言的专用汇编器系统。

1 宏指令专用汇编器的应用环境

混合激励线性预测 (Mixed Excitation Linear Prediction, MELP)^[1] 是一种能在 2.4 Kb/s 的低码率上获得较好合成语音质量的语音编码算法。软件实现语音压缩技术相对容易, 但压缩速度慢^[2]。硬件电路实现语音压缩算法速度快, 且具有并行性, 可以大幅提高语音压缩的速度。但优化后的 MELP 算法仍然具有复杂度高、运算量大的缺点^[3], 从算法转化成硬件电路也存在一定难度^[4], 因此, 实现一个运行速度快且规模较小的专用数字信号处理器是系统设计的目标。

本文的硬件体系结构设计采用 MISC 模型。先将语音算法函数按照功能划分为若干个宏指令, 然后进行理想模型六要素分析 (操作分析、数据分析、资源分析、时效分析、连接分析、控制分析), 操作分析采用最大并行性分析; 将分析结

果叠加, 确定最佳硬件资源组成、连接关系和控制节点组成; 最后进行电路设计。系统的硬件体系结构如图 1 所示。

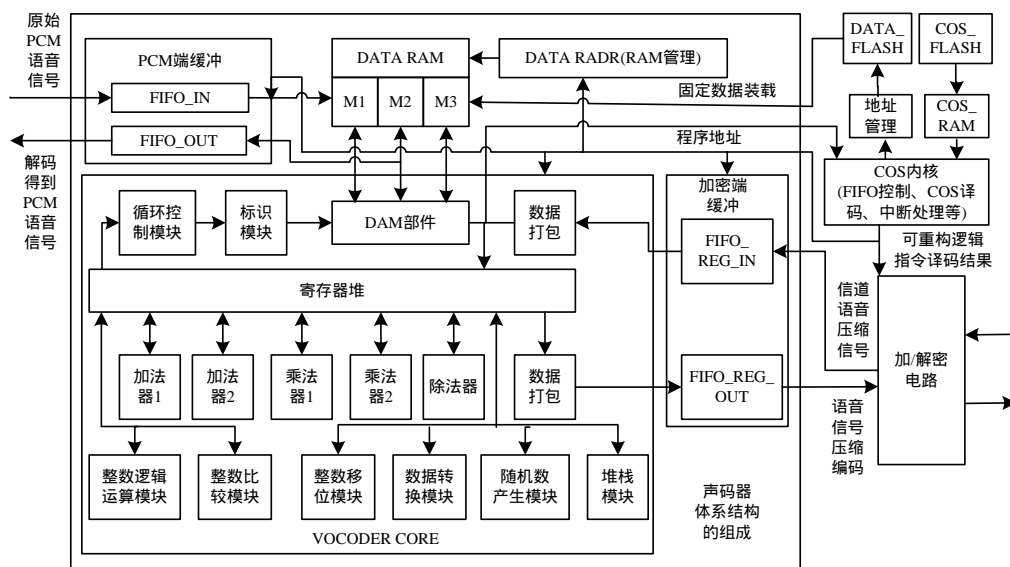


图 1 声码器的硬件体系结构

基金项目: 北京市科技基金资助重大项目“交互式数字电视信道传输核心技术开发”(京科技发[2002]188号)

作者简介: 许娜(1981-), 女, 博士研究生, 主研方向: 嵌入式系统, 无线传感器网络; 张晓彤、王沁, 教授、博士; 梁静, 博士研究生

收稿日期: 2009-05-12 **E-mail:** xunause@sohu.com

2 宏指令的概念及应用

宏指令以算子为指令成分构成高层语义，支持复杂指令功能的并行操作。以声码器的设计为例，宏指令的运用反映了一种硬件允许的并行操作，即在数据、硬件资源不冲突且指令宽度允许的情况下，进行算子操作并行性设计，其目的是尽量提高声码器硬件资源的使用效率并增强指令功能，从而提高声码器的运行速度。本身可以嵌套子宏指令的宏指令被称为复合宏指令。为了降低指令设计的复杂度，调用子宏指令时不再并行其他算子操作或宏指令。

3 宏指令专用汇编器的设计实现

3.1 汇编器的设计要求

宏指令有相对固定的语法格式和约束条件，只有完全按照这些语法设计的宏指令才能正确转化成二进制机器代码^[5]。将 C 语言编写的语音压缩算法转化成状态机要充分利用宏指令的可并行性。

在基于宏指令的专用汇编器出现前，开发人员需手工把 C 语言函数转化成宏指令，翻译成二进制机器代码，然后用 Summit Design 公司的 Summit Elite 软件^[6]设计状态机。手工转化效率低下，易出错，人员还要专门培训。专用汇编器系统的开发就是为了解决此问题，其在整个转换过程中的位置和作用如图 2 所示。其中，左图为没有汇编器时的开发流程；右图为包含汇编器的流程。汇编器系统使开发过程在 3 个语言层次间实现了自动转化。

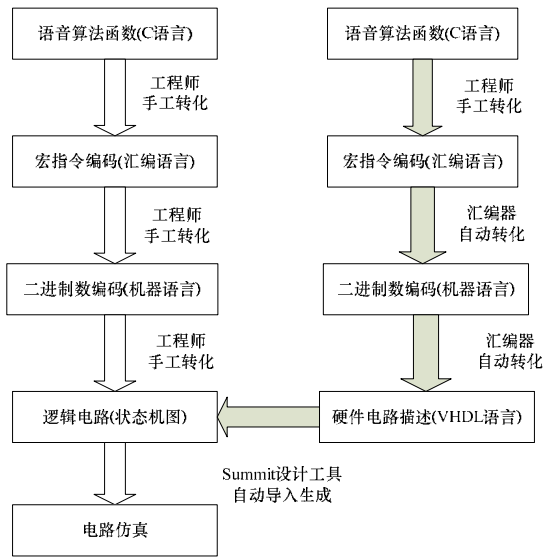


图 2 汇编器在转化过程中的位置和作用

3.2 汇编器的设计要点

汇编器的核心模块是输入宏指令文件，输出二进制代码。执行主要包括 3 个阶段：(1)词法分析：从左到右扫描宏指令字符流，获取相对独立的关键字，这些关键字都是有确定含义的字符序列。(2)语法分析：把单词按语法结构有层次地进行分组。(3)语义分析：进行语义规则检查，保证各部分结构有意义地结合在一起。

汇编器内部对宏指令的一般处理流程的状态跳转如图 3 所示，其中，各状态的含义和功能依次是：S0：初始态；S1：字符流输入，提取单词，构成操作符、[第一操作数]和[第二操作数]([]里的内容表示可选)；S2：根据操作符的不同，选择不同的二进制代码查找表；S3：执行二进制代码的查找和拼装，当遇到查找表中不存在的单词时，显示出错信息；S4：

检查并记录各个 IP 器件的使用情况，添加器件使能信号，构造用 VHDL 语言格式描述的语句；S5：终态。

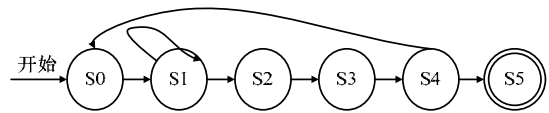


图 3 宏指令处理状态图

3.3 汇编器的功能实现

汇编器的功能模块主要有：解析输入字符串流和提取关键字模块，基于关键字的查找分类模块，六大类宏指令转换模块，机器编码拼装模块，VHDL 代码构造模块，状态跳转模块和检错纠错模块。其中，六大类宏指令对应的 IP 器件共有 14 个，各个 IP 器件又分别对应不同的功能算子(宏指令操作符)，功能算子的总数达 63 个。正因为有如此多的功能算子，才使复杂的语音压缩算法能够用高效的硬件电路实现。六大转换模块与 IP 器件、功能算子的对应关系如表 1 所示。

表 1 转换模块、IP 器件和功能算子的关系

转换模块	IP 器件	功能算子
	(1)DAU	WADD,DADD 等 7 个
	(2)DLOG	WAND,DAND 等 8 个
	(3)MULAS	MUL,WMUL 等 7 个
	(4)DIV	WDIV,DDIV
m_dau	(5)DSHIFT	WSHIFTR,WSHIFTL 等 6 个
	(6)DTRAN	WTRANSE,WTRANCZ 等 7 个
	(7)CMP	WCMP,DCMP
	(8)TMS	TMS
	(9)RSTACK	PUSH,POP,EXP,LOG,SC
m_reg	REGi	MOVA,MOVB,MOVC
m_jump	无	JMP,JUMP,END,CALL
m_adr	ADR	ADRMVA,ADRMVB 等 6 个
m_for	FOR	FORI,FORJ,FORK
m_acc	LS	LOAD,STORE

以循环程序流控制单元为例，其机器代码编码格式为

```
FOR<11:0> OPI<0>
RSFORI<2:0> OPJ<0>
RSFORJ<2:0> OPK<0>
RSFORK<2:0>
```

对应的宏指令形式可以是

```
FORI GR2.L
```

查询系统设计说明文档中的相关内容，找到位于操作数位置的寄存器所对应的机器代码。表 2 是 IP 器件 FOR 对应的操作数编码表。生成的 VHDL 代码为

```
FORL<="110000000000";
```

表 2 IP 器件 FOR 对应的操作数编码表

RSFORI<2:0>	数据来源	RSFORI<2:0>	数据来源
000	GR0.L	100	GR2.L
001	GR0.H	101	DAU1DDL
010	GR1.L	110	IMMD.L
011	GR1.H	111	FORIDEC

4 性能测试

汇编器系统可以发现多种宏指令错误，如拼写错误、语法错误、内存访问冲突、器件复用、立即数冲突，还可指出具体的错误类型，自动纠正。

根据以往数据,使用汇编器可以使工作效率提高 5 倍~6 倍。由它生成的 VHDL 代码文件 90% 以上可以无差错地生成状态机。图 4 显示了随汇编文件个数的增多,汇编器性能提高的趋势。

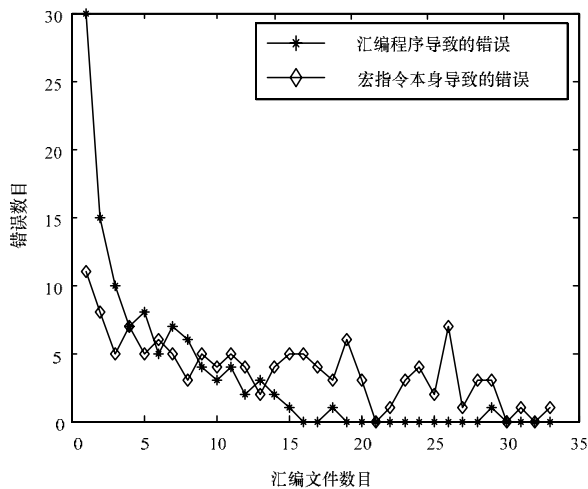


图 4 汇编器运行可靠性统计

只要载入汇编器输出的 VHDL 代码, Summit Elite 工具软件即可自动生成逻辑电路的状态机图。设计人员可以用鼠标拖动线条的端点进行模块之间的连接,也可双击某个状态圈,进入状态内部,进行子状态级的嵌套设计。

5 结束语

基于宏指令集模型的专用汇编器系统的实现有效提高了声码器的研发效率。实践证明,依照自顶向下方法开发的专用汇编器软件具有稳定性高、性能优异、容错纠错性强等特点。虽然是针对宏指令集模型而设计的一种专用汇编器,但本系统的设计思路,尤其是将机器代码进一步转化为 VHDL 语言描述,进而形成硬件电路状态机的设计,对其他体系结构下的硬件电路设计和汇编器开发具有一定的启迪意义。

参考文献

- [1] 王炳锡. 语音编码[M]. 西安: 西安电子科技大学出版社, 2002.
- [2] 刘实, 张雄伟. MELP: 新的 2400bit/s 美国联邦声码器标准[J]. 数字通信, 1999, 25(2): 40-42.
- [3] Jackie R, Gary J K. An Evolutionary Approach to Group Decision Making[J]. INFORMS Journal of Computing, 2002, 14(3): 278-292.
- [4] Falavigna D, Gretter R, Riccardi G. Acoustic and Word Lattice Based Algorithms for Confidence Scores[C]//Proceedings of the International Conference on Spoken Language Processing. Denver, USA: [s. n.], 2002: 1621-1624.
- [5] 刘宇红, 刘桥, 任强. 基于模糊聚类神经网络的语音识别方法[J]. 计算机学报, 2006, 29(10): 1895-1900.
- [6] Rueber B. Obtaining Confidence Measures from Sentence Probabilities[C]//Proceedings of the 5th European Conference on Speech Communication and Technology. Rhodes, Greece: [s. n.], 2001.

编辑 张帆

(上接第 246 页)

5 结束语

高斯投影的跨带计算会带来很大的跨带误差,采用分带拆分计算,然后线性叠加计算结果的方法,能有效地减少误差,但在高带差情况下的非线性误差仍然存在,利用带差作为控制变量,然后利用最小二乘拟合曲线法,对误差曲线进行拟合,最后把拟合参数作为直接修正参数运用到算法中,从而找到了一种快速、准确、简单的跨带计算方法。

参考文献

- [1] 欧龙, 陈性义, 欧阳平. 高斯投影正反算与换带计算 True

BASIC 程序[J]. 铁道勘察, 2007, 32(5): 12-15.

- [2] 成英燕, 李夕银. 适用于不同椭球的高斯平面坐标正反算的实用算法[J]. 测绘科学, 2007, 29(4): 26-28.
- [3] 王晏民. 多源 GIS 高斯投影快速换带算法研究[J]. 测绘工程, 2007, 11(1): 8-13.
- [4] 李学民. 数值计算方法在高斯投影反算中的应用[J]. 城市勘测, 2008, (3): 75-77.
- [5] 李凯, 张娄. Matlab 技术的研究[J]. 鞍山师范大学学报, 2008, (2): 38-41.

编辑 陆燕菲

(上接第 248 页)

从图 3 可以看出,本文所述文件备份系统相对于传统的 Rsync 系统,大大降低了备份操作时所需要传输的网络流量。从图 4 可以看出,采用 Rsync 进行日常备份时所产生的网络流量随着文件大小的增加其增长速度为线性,而本文所述的文件备份方法进行日常备份时所产生的网络流量随着文件大小的增加其增长速度趋于线性。

3 结束语

本文的文件备份方法能够为分散的个人用户提供廉价的文件级数据备份服务。通过把差异计算工作完全放在客户端完成,与 Rsync 算法相比,本文的文件备份方法能够利用用户计算机的计算资源来减轻备份中心的计算负荷,并大量减少网络流量。实验结果表明,与 Rsync 相比,本方法产生的网络流量较小。

参考文献

- [1] 李涛. 网络安全概论[M]. 北京: 电子工业出版社, 2004.

- [2] Tridgell A, Mackerras P. The Rsync Algorithm[D]. Canberra, Australia: Australian National University, 1996.
- [3] Qian Cunhua, Syouji N, Toshio N. Optimal Backup Policies for a Database System with Incremental Backup[J]. Fundamental Electronic Science, 2002, 85(4): 1-9.
- [4] Source A. SANs Sharing Storage to Infinity and Beyond[J]. Electronic Design, 2004, 52(9): 50-59.
- [5] Adler-32-Wikipedia[Z]. [2009-01-11]. <http://en.wikipedia.org/wiki/Adler-32>.
- [6] Lo Chi-chun. A Novel Approach of Backup Path Reservation for Survivable High-speed Networks[J]. IEEE Communications Magazine, 2003, 41(3): 146-152.
- [7] Hayes P E, Hammons A. Disaster Recovery Project Management[C]//Proc. of Conference Record IAS Annual Meeting. San Antonio, USA: [S. n.], 2000: 2814-2821.

编辑 陆燕菲