

文章编号: 1001-2486 (2001) 03-0060-04

在 TMS320C6201 上实现时延神经网络目标识别算法*

付 强, 马君国, 肖怀铁

(国防科技大学 ATR 实验室, 湖南 长沙 410073)

摘 要: 本文介绍了一种基于 TMS320C62x 的软件设计方法, 在 TMS320C62x EVM 板上实现了宽带毫米波雷达目标时延神经网络识别算法, 通过程序验证, 取得了比较好的处理效果。

关键词: 实时处理; 目标识别; TMS320C6201; DSP; 软件设计

中图分类号: TP13 **文献标识码:** A

Implement Target Recognition Algorithm Based on Time - delay Neural Network on TMS320C6201

FU Qiang, MA Jun-guo, XIAO Huai-tie

(ATR Lab., National Univ. of Defense Technology, Changsha 410073, China)

Abstract: This paper demonstrates a kind of software design method based on TMS320C62x, implements the target recognition algorithm based on wide - band milliwave radar time - delay neural network on TMS320C62x EVM board, and achieves better effect through programming testing.

Key words: real - time processing; target recognition; TMS320C6201; DSP; software design

数字信号处理 (DSP) 技术近年来取得了高速发展, 目前 DSP 芯片已经广泛地应用于通信、图像处理、语音处理、雷达等领域。TMS320C6000 是 TMS320 系列产品中的新一代高性能 DSPs 芯片, 其中定点系列为 TMS320C62x, 浮点系列为 TMS320C67x。本文着重分析如何高效地运用 TMS320C6201 的软件和硬件资源实现宽带毫米波雷达目标时延神经网络识别算法。

1 TMS320C6201 的系统结构

TMS320C6201 是新一代高性能数字信号处理 (DSP) 芯片, 处理能力高达 1600 MIPs, 支持 TMS320C6201 芯片的高效率的 C 语言编译器, 使得用户可以快速地开发出应用程序。TMS320C6201 芯片的系统结构如图 1 所示。

CPU 内核中的 8 个功能单元可以完全并行运行, 功能单元执行逻辑、位移、乘法、加法和数据寻址等操作。TMS320C6000 系列芯片的体系结构采用甚长指令字 (VLIW) 方式, 单指令字长为 32 位, 每条 32 位指令占用一个功能单元, 取指令、指令分配和指令译码单元每周期可以从程序存储器到功能单元传递 8 条指令, 这 8 条指令组成一个指令包, 总字长为 $8 \times 32 = 256$ 位, 芯片内部设置了专门的指令分配模块, 可以将每个 256 位的指令分配到 8 个功能单元中, 并由 8 个功能单元并行运行。TMS320C6201 芯片的最高时钟频率可以达到 200 MHz, 当 8 个功能单元同时运行时, 该芯片的处理能力高达 1600MIPs。

TMS320C6201 芯片的片内存储器总容量为 1M 位, 其中 $2k \times 256$ 位用于程序内存和程序 cache, 宽度为 256 位; 64 k 字节用于数据内存和数据 cache, 用户可以访问 8 位、16 位和 32 位的数据; DMA 控制器可以控制完成在存储器空间的不同区域间转移数据; 外部存储器接口 EMIF 可以访问的片外存储器最大容量为 64 MB, 数据总线宽度为 32 位, 同时也提供对 8 位和 16 位存储器的读写支持; 16 位宽

* 收稿日期: 2001-01-08
基金项目: 国家部委基金资助项目
作者简介: 付强 (1963-), 男, 副教授。

的主机口 HPI 可以访问 TMS320C6201 的所有存储空间和设备；多种外设模块使得 TMS320C6201 芯片的功能十分强大。

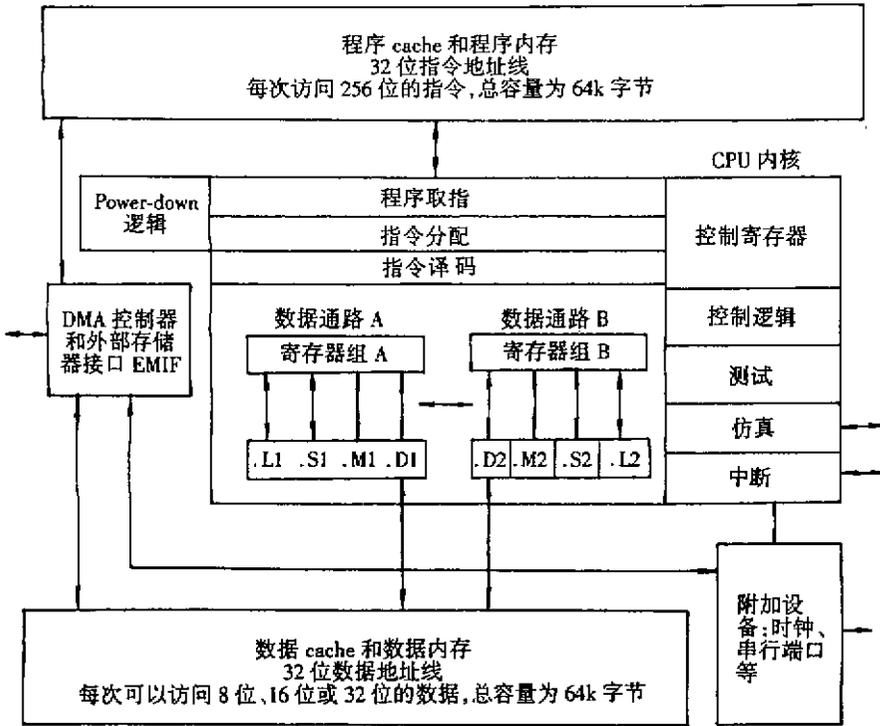


图 1 TMS320C6201 芯片的系统结构

Fig.1 System structure of TMS320C6201 chip

2 TMS320C62x 的软件设计方法

TMS320C62x 的软件设计包括三个阶段：第一阶段是开发 C 代码；第二阶段是优化 C 代码；第三阶段是编写线性汇编代码。以上的三个阶段不是必须经过的，如果在某一个阶段已经实现了应用软件的功能和性能要求，那么就不必进入下一个阶段。

2.1 开发 C 代码

开发 C 语言代码需要考虑的要点表述如下：

① 数据结构

TMS320C62x 编译器对每种数据结构定义一个尺寸，在编写 C 代码时应当遵循的规则是：避免在代码中将 int 和 long 型作为同样尺寸处理；对于定点乘法，应当尽可能使用 short 型数据；对循环计数器使用 int 或者无符号 int 类型。

② 分析 C 代码性能

应用调试器的 Profile 工具可以得到一个关于 C 代码中各特定代码段执行情况的统计表，也可以得到特定代码段执行所用的 CPU 时钟周期数，因此可以找出影响软件程序总体性能的 C 语言代码段来加以改进，通常是循环代码段影响软件程序总体性能。

③ 使用查找表

在 C 语言代码中通过直接计算得到结果的语句或函数可以用查找表或常数数值代替，所以可以提高指令执行速度。

④ 用整型数 (int) 来表示浮点数

由于 TMS320C6201 是定点芯片，在程序的编写过程中应当尽量采用定点的数据结构来提高应用程序的处理速度，对于 C 语言，应当尽量采用整型数 (int) 来表示浮点数。

2.2 优化 C 代码

这一阶段是对开发的 C 代码进行优化,优化 C 代码的方法包括使用编译器选项、使用内联函数、使用字访问短整型数据和使用软件流水等。

① 向编译器指明不相关的指令

不相关的指令可以并行地执行,用户可以通过如下的方法向编译器指明不相关的指令:

- 使用关键字 `const` 指定一个目标来提高代码的性能和适应性。
- 一起使用 `-pm` 选项和 `-o3` 选项确定程序优先级,从而使编译器更有效地消除相关性。
- 使用 `-mt` 选项向编译器说明在代码中不存在存储器相关性。

② 使用内联函数 (intrinsic)

TMS320C62x 编译器提供的内联函数是直接映射为内联的 C6000 指令的特殊函数,用户可以用内联函数来快速优化 C 代码。

③ 使用字来访问短整型数据

内联函数中有些指令是对存储在 32 位寄存器的高 16 位和低 16 位字段进行操作。当有大量短整型数据进行操作时,可以使用字 (整型数) 一次访问两个短整型数据,然后使用内联函数对这些数据进行操作,从而减少对内存的访问。

④ 使用软件流水

软件流水是用来安排循环指令,使这个循环的多次迭代并行执行的一种技术,在编译时使用 `-o2` 和 `-o3` 选项,编译器可对循环代码实现软件流水。用户可以向编译器传递循环次数信息;可以使用投机执行 (`mh` 选项) 来消除软件流水循环的排空,从而减少代码尺寸;对于执行周期很少的内循环作循环展开,外循环进行软件流水,这样可以改进 C 代码的性能。使用软件流水时还应当注意的问题有:软件流水循环不能包含函数调用;在循环中不可以有条件终止指令;在循环体中不可以修改循环控制变量,等等。

2.3 编写线性汇编代码

编写线性汇编代码是代码开发流程的第三个阶段。为了提高代码性能,对于影响应用程序速度的关键 C 代码,可以用线性汇编重新编写,线性汇编文件是汇编优化器的输入文件。优化线性汇编代码的方法包括:为线性汇编指令指定功能单元,使得最后的汇编指令并行执行;使用字访问短整型数据;使用软件流水对循环进行优化。编写线性汇编代码的工作量非常大,需要很长的开发周期,而且开发后的汇编代码不能像 C 代码那样移植在其它的 DSP 平台上,因此建议用户尽量采用第一阶段和第二阶段来进行软件设计。

3 时延神经网络识别算法分析

目标识别算法对于实时处理的需求是比较苛刻的,例如某种宽带导引头目标识别处理时间要求小于 1.5 ms,即要求目标识别算法在 1.5 ms 内完成对一组数据的处理,正确地把目标识别出来。人工神经网络 (ANN) 具有并行处理运算能力和网络的信息存储能力,能满足宽带毫米波雷达目标识别系统并行计算和较小数据存储空间的要求,ANN 技术应用于雷达目标识别极具潜力。在多层前馈感知器神经网络模型中引入时延单元可以使神经网络增加记忆功能,由此导出的神经网络模型适用于处理序列数据。宽带毫米波雷达目标时延神经网络识别算法的基本原理:对一维距离像进行非相干平均和自适应门限、等距离间隔峰值下采样预处理,获得较稳定的低维样本后,作为特征矢量提供给时延神经网络分类器进行自动分类识别。

根据误差梯度下降法可以推导出时延神经网络的权值的递推公式,然后采用大量的训练样本集对时延神经网络进行训练,得到时延神经网络的权值。网络训练结束后,用它来对测试样本集进行分类。对某一序列的分类方法如下:

(1) 设序列长度为 Z ($Z \geq F$), 令 $k = 0$;

(2) $k = k + 1$;

- (3) 输入第 s 个序列的第 k 组特征矢量 $X_i(t - T_1 + k)$;
 (4) 计算隐层输出矢量 $x_j^2(t)$, $y_m^3(t)$, $x_m^3(t)$;
 (5) 若 $k = Z - T_1 - 1$, 则将结果分类, 否则转 (2)。

4 算法的程序实现

时延神经网络为三层网络, 网络输入层节点数为 17, 网络隐层节点数为 10, 网络输出层节点数为 3。基于以上的时延神经网络识别算法分析, 采用大量的训练样本集对时延神经网络进行训练, 得到时延神经网络的权值。然后在 TMS320C6201 芯片上实现了基于时延神经网络的目标自动分类算法, 分类结果采用投票的综合判决方式。程序流程图如图 2 所示。

在程序实现过程中, 对于经常用到的两个函数值 $1.0 / (1.0 + \exp(-x))$ 和 $\tanh(x)$ 采用查找表的形式来提高程序执行速度。根据对程序的测试, 使用整型数 (int) 来表示浮点数, 选取整型数的低 13 位来表示小数。支持 TMS320C6201 EVM 板的开发软件 Code Composer Studio (CCS) 是一个集编译、连接、实时调试、跟踪和分析应用程序于一体的软件包, CCS 能够加快用户的开发进度, 增强用户的应用程序的性能, 使用户可以创建和测试实时的、嵌入的数字信号处理应用程序。在开发软件 CCS 的集成开发环境中, 综合应用 TMS320C62x 的软件设计方法来编写和优化时延神经网络识别算法的 C 代码。

经过实际测试, 在 TMS320C6201 上实现宽带毫米波雷达目标时延神经网络识别算法的程序执行时间为 0.850ms, 满足了目标识别算法的实时性需求, 取得了比较好的处理效果。

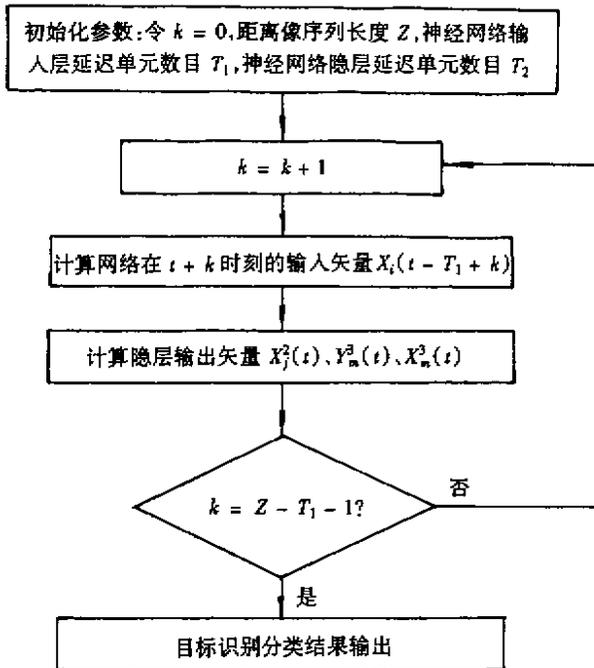


图 2 目标识别算法的程序流程

Fig.2 The program diagram of the target recognition algorithm

参考文献

- [1] 肖怀铁, 宽带极化毫米波雷达目标特征信号测量与识别算法研究 [D], 国防科技大学, 2000.
 [2] 任丽香 马淑芬 李方慧, TMS320C6000 系列 DSPs 的原理与应用 [M], 北京: 电子工业出版社, 2000.
 [3] TMS320C6x 用户手册 [M], TI 公司, 1999.

