

从 Matlab 仿真到 DSP 实现信号处理算法

侯其立, 徐科军

(合肥工业大学 电气与自动化工程学院, 安徽 合肥 230009)

摘要: 当用 Matlab 完成数字信号处理算法仿真后, 如何在 DSP 芯片上实时实现, 是电气信息类大学生需要掌握的一项重要的工程实践能力。在仿真过程中, 有算法移植、DSP 工程建立和算法实现这三个关键环节。本文介绍了当信号处理算法完成离线仿真后到 DSP 实时实现的具体步骤, 并针对此过程中的常见问题, 给出相应的解决方案。

关键词: 数字信号处理; DSP; 算法实现

中图分类号: G420

文献标识码: A

文章编号: 1008-0686(2012)02-0092-03

Signal Processing Algorithms from Simulation with Matlab to Implement with DSP

HOU Qi-li, XU Ke-jun

(School of Electrical and Automation Engineering, Hefei University of Technology, Hefei 230009 china)

Abstract: It is an important practice ability for the graduate and graduated students in the electrical and information specialty how to implement digital signal processing algorithm with DSP chips in real-time after algorithm simulated by Matlab. There are three key stages in this process, such as algorithm transplantation, DSP project design and algorithm implement. This paper presents the implement steps of signal processing algorithms with DSP chips after off-line simulation by Matlab, and provides the corresponding solutions for common problems occurred in the procedure.

Keywords: digital signal processing; DSP; algorithm implement

Matlab 是集数值处理、符号运算、图形处理于一体, 并具有信号分析工具箱, 成为数字信号处理方法研究的基本工具和首选平台。在许多应用场合, 需要用 DSP 芯片实时实现数字信号处理算法。如何在完成数字信号处理算法的 Matlab 仿真后, 去用 DSP 实时实现是数字信号处理方法应用的关键, 是电气信息类大学生和研究生必须具备的工程实践能力之一。但是, 目前相关课程和教材中很少介绍此方面的内容^[1,2]。为此, 我们根据近年来用 DSP 开

发数字信号处理系统的科研经验以及相关课程的教学体会, 介绍从 Matlab 完成数字信号处理算法的仿真后到 DSP 实时实现这一过程中的三个关键环节: 算法移植、DSP 工程建立和算法实现, 以保证用 DSP 实现算法的计算精度和处理的实时性。

1 算法移植

当信号处理算法在 Matlab 上的仿真结果满足要求后, 即可对算法程序进行移植。DSP 程序以 C

修改日期: 2011-05-17; 修回日期: 2011-07-08 基金项目: 安徽省教学研究项目: “信号检测、处理及实现” 系列课程理论和实践教学改革创新(2008JYXM207); 合肥工业大学研究生精品课程“系统建模与信号处理” 资助项目。

作者简介: 侯其立, (1987-), 男, 硕士研究生, 研究方向为检测技术与自动化装置, E-mail: hqili@163.com

徐科军, (1956-), 男, 博士, 教授, 主要从事传感器技术和信号处理等研究工作, E-mail: dsplab@hfut.edu.cn

语言为主,所以算法移植主要就是 Matlab 语言与 C 语言之间的转换。在转换过程中,可先借助 VC++ 平台,建立 VC 工程,单独考核算法程序在移植过程中可能会出现精度损失。

1) 语法转换

Matlab 语言与 C 语言在语法上存在一些差异,如 Matlab 中变量不需要定义,可以直接使用;而 C 语言中的任何变量都需要先定义后再使用。所以在数组和指针方面要避免出错^[3]。

2) 部分功能指令转换

Matlab 中自带了一些 C 语言中没有的功能函数,例如取整 floor、排序 sort 和做快速傅立叶变换 FFT 等^[3,4]。在算法移植时,需要将这些功能函数转化为 C 语言中具体的程序代码段。对于较为复杂的功能函数,如 FFT 可先在 Matlab 上用具体的程序代码实现,并与 FFT 函数的计算结果相比较,在满足计算精度要求后,再向 C 语言转换。

3) 验证正确性

在 VC 工程建立完之后,就需要对算法移植的正确性进行验证。验证的最简单方法就是同样一组输入数据,分别由 Matlab 工程和 VC 工程计算并得到结果,以 Matlab 工程的计算结果作为标准,考核 VC 工程的正确性。一个可行的步骤如下:

(1) 调用 Matlab 工程,并将 Matlab 产生的输入信号和计算结果保存至文件;

(2) 调用 VC 工程,通过文件操作将输入信号数据导入工程中^[5];

(3) VC 工程对导入的输入数据进行运算,再将计算结果导出至文件;

(4) 在 Matlab 下,将两个计算结果导入,以比较验证 VC 工程移植的正确性。

(5) VC 工程验证正确后,程序就可以进一步向 DSP 中移植。

2 DSP 工程的建立

2.1 工程建立顺序

一个 DSP 工程除了包含 Matlab 和 VC 工程中的算法外,还有整个系统配置和外设器件通信等文件,所以,必须要按照一定的顺序建立 DSP 工程。

(1) 建立 DSP 最小系统工程

系统外围器件的配置和算法的运行都是基于 DSP 系统,所以首先应该建立一个 DSP 最小系统工

程^[6]。

(2) AD 采样的加入

与仿真顺序类似,在建立完 DSP 最小系统程序后,应该将 AD 采样模块加入,为算法工程提供输入信号。若采用的是外部 AD,则 AD 采样模块会用到 DSP 与 AD 通信的接口模块。若需要用 AD 触发中断,则应配置 PIE 中断模块、外部中断模块;若用 DMA 来传输采样数据,还应配置 DMA;若片上存放 AD 采集数据的 RAM 空间不够,应增加 RAM 空间,还应配置相应外设接口 XINTF。

(3) 算法的加入

AD 采集正确之后,将算法模块加入。算法模块主要包括算法的初始化、算法调用和计算结果处理。程序在 DSP 上运行时,算法变量的初始化尤为重要。在 VC 工程中,定义一个全局变量而未赋初值,则调用该变量时,默认其为 0;而在 DSP 上,若未对其赋初值,在调用该变量时,其值为其存储空间上的值,在系统断电或上电后,RAM 空间是一个无穷大的值,如果这个值参与到算法计算,将直接导致算法的失效。算法模块加入后,要用信号发生器对算法进行测试。

(4) 输出模块及其它模块的加入

最后加入的是系统的输出以及其它模块。如将测量结果转化为脉冲和电流输出,根据测量结果合成相应波形输出控制信号等。输出模块加入后,也要对单个模块的正确性进行验证。

2.2 模块化设计

若一个 DSP 工程的功能较多,可将整个 DSP 工程按功能和外设器件等划分为各模块,如 AD 采集模块、数据传输模块、算法模块、人机接口模块和输出模块。模块化设计使得每个模块与整个软件系统的独立,这样便于每个模块向其它 DSP 工程的移植,也使得整个 DSP 工程结构框架一目了然。

3 算法具体实现

3.1 算法实现的精度问题

在不同处理器上运行相同的算式,由于处理器运算位数的差异,最终计算精度是不同的。例如,Matlab 中的算式和变量默认下具有 double 型精度,而 DSP 中,在不加修饰的情况下,浮点数仅具有 float 型精度。

算法在 DSP 上具体实现时,常遇到的问题

DSP 运算的结果与仿真结果有较大差异,考虑到 DSP 工程与仿真工程主要是在输入信号和运算平台两方面存在差异,所以可从以下两方面查找原因。

1) 输入信号

DSP 工程的输入信号由 AD 采集,当算法运算出错时,有可能是输入信号出了问题。可按如下方法检验输入信号是否存在问题:

(1) 在 DSP 工程内开辟一长数组,用来存放 AD 采样的数据结果,数组存放满后,通过 CCS 的 File-Data-Save 将数据导出。

(2) 数据导出后,先用 Matlab 对其进行初步分析,查看信号是否按程序内所设置的 AD 采样频率被采样,采样数据是否存在畸变点。

(3) 若数据被正常采样,可将数据样本导入到 VC 工程,由 VC 工程运算。如果 VC 工程运算结果同样存在较大误差,则可认为是 AD 本身采样的精度不够导致了算法的计算误差。

(4) 若 VC 运算结果正确,这时应考虑是否运算平台的不同导致了计算错误。

2) 运算平台

算法在 DSP 上运行时,计算精度会到一些限制,为考察是否计算精度的差异造成计算结果与仿真结果存在较大偏差,可采取以下方法:

(1) 选取 DSP 工程和 VC 工程作为比较对象。VC 工程应是验证正确的工程,其作为标准。

(2) DSP 工程和 VC 工程选择同一个输入信号。DSP 工程的信号可由 File-Data-Load 导入到工程内,VC 工程的信号可由文件操作导入到工程内。

(3) 运行 DSP 工程和 VC 工程,并比较运算结果。若计算结果一致,则可以排除算法移植方面的影响。若计算结果不一致,则需要两个工程中设置断点,查看程序运行到哪一函数和哪一句代码后,结果存在了差异。

3.2 算法实现的实时性要求

Matlab 和 VC 工程仅仅是对算法的仿真,而 DSP 工程则要求算法能够实时实现,这就要求简化算法,合理配置 DSP 资源。为确保算法的计算速度,要精简算式。如重复使用的算式可设定为中间变量,先计算出来后,由别的算式直接使用结果;DSP 处理除法的速度远低于乘法,因此要尽量避免除法运算;DSP 处理浮点数的速度低于定点数的速度,因此,可将浮点运算转换为定点运算。

DSP 程序在片上 RAM 中运行时,具有较快的指令速度,而程序在 Flash 中运行时,速度大大降低。例如 TMS320F28335 具有 150MHz 的 RAM 指令执行速度,而在片上 Flash 中运行时,指令执行速度仅有 90 ~ 100MHz。最终的系统完成时,算法必须烧写至 Flash 中运行。

数字信号处理方法一般要采集大量数据,为保证算法实时实现,应尽量减小 AD 采样数据传输所占用的 CPU 资源。现在 DSP 片上已带有 DMA 模块,可将 DMA 与外设接口相配合。如将 DMA 与 McBSP 相配合,McBSP 用于和外部 AD 通信,DMA 将 McBSP 接收到的数据存放到用户指定 RAM 空间而不需要 CPU 干预,大大减小数据传输对 CPU 资源的占用。

4 结语

信号处理算法在 Matlab 上仿真后,需要在 DSP 上具体实现以转换为实用技术。在实现过程中,可先借助 VC++ 平台,将 Matlab 语言转换为 C 语言,完成语法、语义和功能语句之间的转换,并使用数据导入、导出手段来验证移植的正确性。在建立 DSP 工程时,应该采用模块化的设计方法,将各模块按先后顺序加入到工程中。在 DSP 工程编写过程中,应从 DSP 运算的位数、库函数的精度等方面出发,保证算法的实现精度;通过将关键程序映射至高速 RAM 中运行,提高执行速度,合理配置 DSP 资源。我们特别要合理配置 DSP 与 AD 器件的通信接口,最大限度地减小数据传输对 CPU 的占用,以保证算法的实时实现。

参考文献

- [1] 徐科军,全书海,王建华. 信号处理技术[M]. 武汉:武汉理工大学出版社,2004,8-208
- [2] 徐科军,陈志辉,傅大丰. TMS320F2812 DSP 应用技术[M]. 北京:科学出版社,2010,252-264
- [3] 飞思科技产品研发中心. MATLAB7 辅助信号处理技术与应用[M]. 北京:电子工业出版社,2005,147-281
- [4] 薛定宇,陈阳泉. 高等应用数学问题的 MATLAB 求解[M]. 北京:清华大学出版社 2004,1-42
- [5] 郑阿奇,丁有和,郑进,等. Visual C++ 实用教程(第 2 版)[M]. 北京:电子工业出版社,2005.1-90
- [6] Texas Instruments. Code composer studio development tools v3.3: Getting started Guide,2006,October