TMS320 DSP
DESIGNER'S NOTEBOOK

Efficient Coding on the TMS320C5x

APPLICATION BRIEF: SPRA211

Mansoor Chishtie
Digital Signal Processing Products
Semiconductor Group

Texas Instruments
February 1993

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain application using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

TRADEMARKS

TI is a trademark of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.

CONTACT INFORMATION

US TMS320 HOTLINE	(281) 274-2320
US TMS320 FAX	(281) 274-2324
US TMS320 BBS	(281) 274-2323
US TMS320 email	dsph@ti.com

Contents

Figures

Examples

Efficient Coding on the TMS320C5x

Abstract

Algorithms based on dynamic programming techniques often make use of looped code, conditional execution, min-max searches, and pointer addressing. The TMS320C5x core CPU allows zero-overhead looping, multiple-condition branches, delayed jumps and calls, min-max instructions, and post-modify indirect addressing to implement efficient search algorithms. This document discusses how to determine when these techniques are best for the application, and how to implement them.

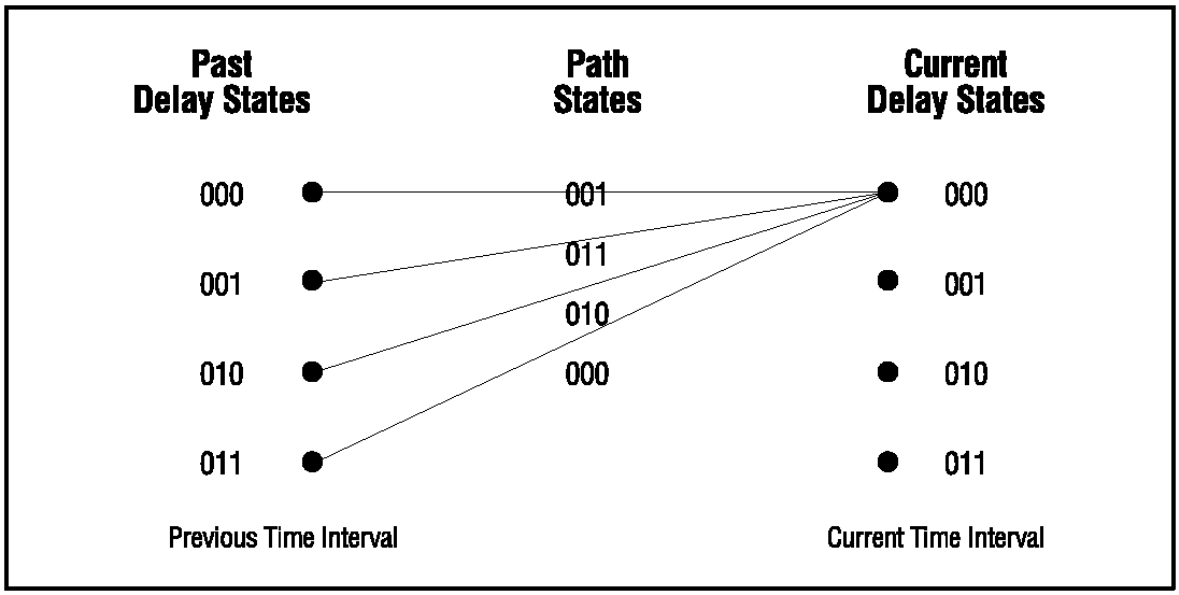
Design Problem

What are some examples of software that take advantage of the ’C5x architecture?

Solution

Algorithms based on dynamic programming techniques often make use of looped code, conditional execution, min-max search, and pointer addressing. The TMS320C5x core CPU allows zero-overhead looping, multiple-condition branches, delayed jumps and calls, min-max instructions, and post-modify indirect addressing to implement efficient search algorithms.

Figure 1. A popular Viterbi subroutine selects the “minimum cost” path



The Viterbi decoding algorithm is quite popular in data communications applications. One subroutine used by this algorithm is presented here to demonstrate efficient ’C5x code. The function of the subroutine is to select the “minimum cost” path to current delay state out of four possible paths (see Figure 1). Each path has its associated “cost value” and each past delay state has an accumulated cost. The new accumulated cost is computed by adding the path cost to the accumulated cost of the past delay state. The path with the minimum accumulated cost is selected and the rest are discarded.

When a path link is selected, the path state value identifying the link and the past delay state are stored in appropriate tables (PAST_PTH, PAST_DLY). The accumulated distance and current distance tables (ACC_DIST, DIST) are accessed by indirect circular addressing mode. The four path states are not in binary ascending or descending order, but a four-word circular buffer can be set up that steps through each path state in the correct sequence. It also resets the pointer to the first state automatically after exiting the loop. Note that all the instructions inside the block-repeat loop are single-cycle instructions.

The complete ’C5x Viterbi implementation is available on the BBS.

- AR1 - ACC_DIST (set up as 4-word circular buffer)
- AR2 - DIST (set up as 4-word circular buffer)
- AR3 - MIN_DIST (minimum accumulated distance table)
- AR5 - PAST_DLY (past delay state table)
- AR6 - PAST_PTH (past path state table)

Example 1. Code Listing

```
BEGIN
  MAR     *,AR1     ;ARP = AR1
  SPLK   #3,BRCR    ;set up count
  LACC   #07FFFH    ;max value
  SACB              ;ACCB=07fffh
  RPTB   LOOP-1     ;repeat 4 times
  ACC    *,0,AR2    ;Get old acc distance
  ADD    *,0,AR3    ;Add current distance
CRLT               ;if (ACC ACCB) then ACCB = ACC
  SACL   *,0,AR5    ;Save new min value
  XC     2,C        ;Update PAST_PTH and PAST_DLY
  SAR    AR1,*,AR6  ;pointer to ACCDIST - PAST_DLY
  SAR    AR2,*,AR1  ;pointer to DIST - PAST_PTH
  MAR    *,AR1      ;ARP = AR1
  MAR    *+,AR2     ;AR1++ (circular addressing)
  MAR    *+,AR1     ;AR2++ (circular addressing)
LOOP
```