TMS320 DSP
DESIGNER'S NOTEBOOK

Binary Search Algorithm on the TMS320C5x

APPLICATION BRIEF: SPRA238

Lawrence Wong
Digital Signal Processing Products
Semiconductor Group

Texas Instruments
May 1994

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

Abstract 7
Design Problem 8
Solution 8

Examples

Example 1. Code Listing 8

Binary Search Algorithm on the TMS320C5x

Abstract

This document shows how to implement an efficient binary search algorithm on the TMS320C5x Family that will take advantage of the ’C5x’s capability.

Design Problem

How do I implement an efficient binary search algorithm on the TMS320C5x Family that will take advantage of the ’C5x’s capability?

Solution

There are many ways to implement the classical binary search algorithm but very few would take advantage of the ’C5x’s advanced architecture and instruction set. The following is one of the many possible examples using the TMS320C5x executing the binary search algorithm.

The program takes advantage of the ’C5x’s capability of performing bit-reversed addressing to half the search after each testing and therefore freeing the accumulator for other tasks. Also, instead of using conditional branching to perform the testing, the execute conditional (XC) instruction is used, thereby saving cycles and increasing performance.

This routine performs a binary search on an ordered table. It assumes that the table is ordered from low to high, where the largest number is located in the highest memory of the array. Modifications can be made to reverse the ordering, if necessary. This program also assumes that the size of the search table is some integer power of 2 (i.e., 2^N where N=11 in the following program). As a result, the search would never pass the last entry in the array. A maximum of N iterations is required to complete the search or determine that the search failed. Modifications can be made if the size of the array is not a power of 2. In order to do this, test conditions will have to be included to determine if the last entry has been passed.

This function returns the address of the found number and it is stored in the ACCUMULATOR. A 0x0000 address in the ACCUMULATOR signifies that the search was unsuccessful.

Example 1. Code Listing

```
        .bss    NTABLE,800h    ;Sorted search table
                               ; from low to high
        .bss    LOOK,1         ;Search value
        .mmregs
        .text
        .
        .
        .
        call    bsearch
        .
        .
        .
bsearch lar     AR0,#0800h     ;AR0 size of array
```

```
        mar   *,AR0 ;
        mar   *BR0+,AR3   ;Half the size of the array
        lar   AR3,#NTABLE ;AR3 points to beginning of
                          ; array
        lacl  #11   ;RPT N Times, Size of Array is 2^N
        samm  BRCR        ;Setup Repeat Block
        ldp   #LOOK
        lacc  LOOK        ;Begin search
        sub   *           ;Compare data at AR3
        bcnd  nothere,LT  ;ERROR not found in this array
        rptb  nothere-1
        bcnd  found,EQ    ;Check if found
        xc    1,GT        ;If too low on array
        mar   *0+,AR0     ;Jump forward
        xc    1,LT        ;If too high on array
        mar   *0-,AR0     ;Jump back
        mar   *BR0+,AR3   ;Half the search space
        lacc  LOOK
        sub   *
nothere retd        ;Did not find value in the table
        zac         ;return 0x0 for failed search
        nop
found   ldp   #0
        apl   #0fffeh,PMST  ;disable block repeat bit
        retd
        lamm  AR3           ;return address of search
        nop
```