TMS320 DSP
DESIGNER'S NOTEBOOK

Bit-reversed Addressing without Data Alignment on the 'C3x

APPLICATION BRIEF: SPRA199

Tim Grady
Digital Signal Processing Products
Semiconductor Group

Texas Instruments
December 1992

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

Bit-reversed Addressing without Data Alignment on the ’C3x

Abstract

Bit-reversed addressing mode normally requires that the n-element array be aligned on an n-word boundary. When n is large, this may result in a large “hole” in the memory map. To enable more efficient use of memory, a technique to use bit-reversed addressing *without* data alignment is presented.

Design Problem

Bit-reversed addressing mode requires that the n-element array be aligned on an n-word boundary. When n is large, this may result in a large “hole” in the memory map. To use memory more efficiently, a technique to use bit-reversed addressing without data alignment is required.

Solution

Figure 1 shows a block diagram of one solution to this problem. AR2 points to the data. AR1 is initialized to 0 and becomes an offset into the array. Bit-reversed addressing mode is used to modify AR1. Figure 2 shows an assembly language version. Figure 3 shows a C version that uses in-line assembly to permit bit-reversed addressing.

Figure 1. Solution diagram

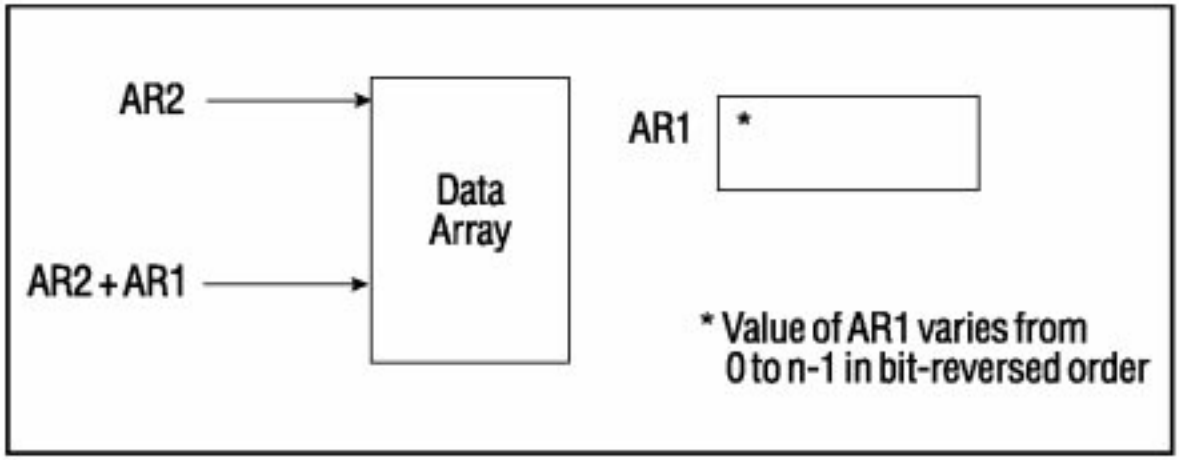

Figure 2. Assembly Code Implementation

```
                .data
table           .word   8,9,10,11,12,13,14,15
taddr           .word   table
                .text
                .global _main
_main           ldp     taddr
                ldi @taddr,ar2   ; pointer to array
                ldi 4,ir0  ; 1/2 array size for bit-rev addressing
                ldi 0,ar1     ; first address in bit-rev list
                ldi 7,rc
                rptb endloop
                ldi ar1,ir1   ; put new offset into index register
                      ; This instruction may also be put in parallel
                      ; ifthe right application comes along.
endloop         ldi *+ar2(ir1),r0 ;r0 holds array elements one at a
                                  ;time so that results can be observed
```

```
              || ldi *ar1++(ir0)B,r7 ;calculate next address in
                   ;parallel r7 is a dummy variable to allow paral ops
                 rets
```

Figure 3. C Code Implementation

```
int x[15]= {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
int *y=(int *)&x;
int m;
main()
{
       int i;
       y += 7; /* start with non-aligned array element */
       asm(“ ldi @_y,ar0");     /* ar0 points to array */
       asm(“ ldi 0,ar2");       /* index for bit-rev */
       asm(“ ldi 4,ir0");       /* set up for bit-rev */

       for(i=0;i<8;i++)
       {
              asm(“ ldi ar2,ir1"); /* load index of array */
              asm(“ ldi *+ar0(ir1),r7"); /* traverse */
              asm(“ || ldi *ar2++(ir0)b,r6"); /* array with */
              asm(“ sti r7,@_m”); /* bit-rev offset */
       }
}
```