

*TMS320 DSP  
DESIGNER'S NOTEBOOK*

# ***Optimizing Control Algorithms on the 'C5x***

---

---

---

*APPLICATION BRIEF: SPRA200*

*Alex Tessarolo  
Digital Signal Processing Products  
Semiconductor Group*

*Texas Instruments  
December 1992*



## **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

## Figures

Figure 1. Register Values.....	8
Figure 2. Fast Algorithm Design.....	9

# Optimizing Control Algorithms on the 'C5x

---

---

---

## Abstract

In many control algorithms, a value that resides in the 32-bit accumulator must either be stored to a 16-bit memory location or to a peripheral device that may be less than or equal to 16 bits in resolution, i.e., 8-bit A/D converter. Prior to storage, a range check must be performed on the sign bit (S) and the guard bits (G) in the accumulator. For positive numbers within range of the desired value,  $S = G = 0$  and for negative numbers,  $S = G = 1$ . If this is not the case, then overflow has occurred and the value stored must be saturated. Standard published code to perform this saturation can take up to 15 cycles. This overhead can be minimized using the technique shown.

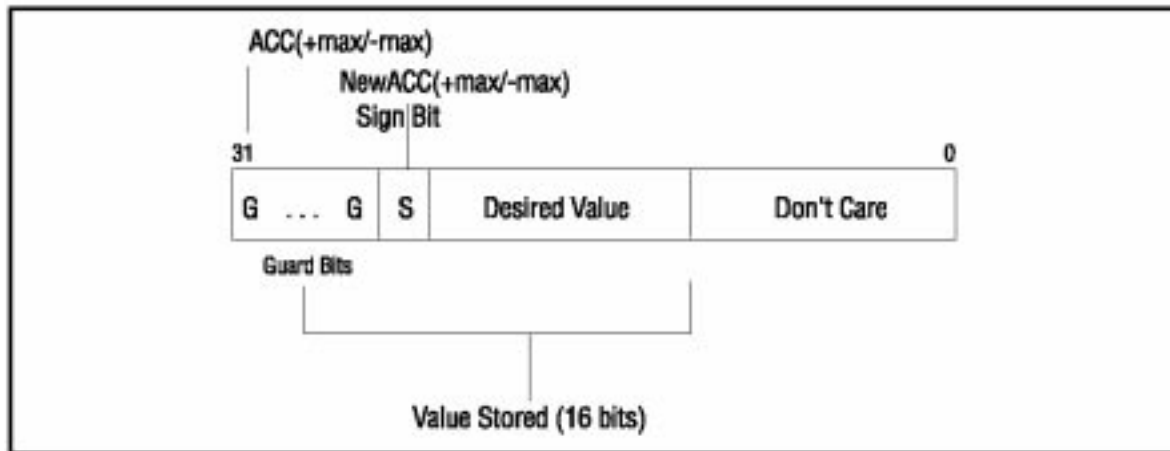


## Design Problem

In many control algorithms, a value resides in the 32-bit accumulator that must be either stored to a 16-bit memory location or to a peripheral device which may be less than or equal to 16 bits in resolution, i.e., 8-bit A/D converter. Prior to storage, a range check must be performed on the sign bit (S) and the guard bits (G) in the accumulator. For positive numbers within range of the desired value,  $S = G = 0$  and for negative numbers,  $S = G = 1$ . If this is not the case, then overflow has occurred and the value stored must be saturated. See Figure 1, below.

Standard published code to perform this saturation can take up to 15 cycles. How do you minimize this overhead?

Figure 1. Register Values



## Solution

A technique for doing this operation that requires a minimum number of cycles is described below:

- 1) Calculate the difference (Diff) between the ACC positive maximum value [ACC(+max)] and the desired positive maximum value [NewACC(+max)]:

$$\text{Diff} = \text{ACC}(+\text{max}) - \text{NewACC}(+\text{max})$$

- 2) Make sure saturation mode is on (SOVM).

- 3) Execute the following instructions:

```
ADDH Diff    ;Step 1
SUBH Diff    ;Step 2
SUBH Diff    ;Step 3
ADDH Diff    ;Step 4, Value to be stored is either
              ;saturated or unchanged if within range
```

The above operation is shown in Figure 2, below.

Figure 2. Fast Algorithm Design

