

C implementation of the TMS320C64x Intrinsic Operators

Gayathri Krishnamurthy

Digital Signal Processing Solutions

ABSTRACT

The first optimization step that can be performed on C source code for the TMS320C6000™ DSP platform is to use intrinsic operators. These are built-in functions in the compiler that can be directly translated to assembly code, for faster execution. The C6000 DSP family fixed-point processors are organized into two main families: the TMS320C62x™ DSP and the TMS320C64x™ DSP. The C62x is the base member of the family and the C64x came later and has extra architectural features not supported on the C62x. To support these features and to enable users to take advantage of them, the C64x has some additional assembly instructions and related set of intrinsics, which are a superset of the C62x intrinsics. The C64x compiler is backwards compatible with the C62x compiler. It would be advantageous to have the reverse capability, of having the C64x code run on the C62x, special intrinsics and all. The code proposed within this application report attempts to do just that, by implementing a library of C functions that would be called instead of the intrinsics, when C64x code is compiled on the C62x.

This application report contains project code that can be downloaded from this link:
<http://www-s.ti.com/sc/psheets/spraa75/spraa75.zip>

Contents

1	Design Problem	1
2	Solution	2

1 Design Problem

The C6000 compilers support special functions called intrinsic operators, that will enable translation of some common user operations into efficient assembly code. These operators directly map to assembly code and by using them, the user can get faster execution of their code.

The C6000 family fixed-point processors are organized into two main families: the C62x and the C64x. The C62x is the base member of the family. Code with C62x intrinsics can be compiled/run on the C64x - but currently we do not have the reverse capability, as the C64x has some additional intrinsics. The code proposed within this application report attempts to provide this capability, by implementing a library of C functions that would be called instead of the intrinsics, when C64x code is compiled on the C62x.

A related problem with a wider scope that is not fully addressed in this document is making the C64x intrinsic C code ISO-C compatible. A previous document, *TMS320C6000 C Compiler: C Implementation of Intrinsics* (SPRA616), dealt with making C62x Intrinsics ISO-C compatible. The routines in this document are ISO-C compatible for the most part, except for those that manipulate 64-bit long long integers. These are not part of the ISO standard. You can use this document as a starting point from which you can create a set of functions which are compatible for simulating the C64x intrinsics on other compilers. You would first have to define the appropriate typedef's for that compiler in C64intrinsic.h. Also, to return a 64-bit type on the C64x before 'long long' was supported, some of these functions/intrinsics were defined to return type 'double'. For other non-C6x compilers, these functions would have to be updated to return a 64-bit fixed-point type supported on this other compiler.

2 Solution

The code proposed within this document allows the user to write C code using C64x intrinsic operators and compile the code on any C6000 target.

The code is organized in two files: `C64intrins.h` and `C64intrins.c`. The code in `C64intrins.c` implements a library of C functions which perform functions equivalent to the C64x intrinsics. The user must include the header file `C64intrins.h` in their C files which contain intrinsics, and compile and link with `C64intrins.c`.

The `C64intrins.h` file contains wrappers for the intrinsics that function as follows: if the user code is compiled on the C64, then the intrinsics will be interpreted as such. If the same code is compiled on any other C6000 target, the equivalent functions in `C64intrins.c` will be called instead.

All the C64 intrinsics, except for the memory-related intrinsics (`lddw`, `ldndw`, `stdw`, `stndw`) are implemented in `C64intrins.c`.

The files `C64intrins.c` and `C64intrins.h` have been compiled and tested on the C6000 compilers. They can be accessed from the same location where you download this application report.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265