FASTINTDIV

USER'S GUIDE

FASTINTDIV-SW-UG-V1.02.00.00

Copyright

Texas Instruments
12203 Southwest Freeway
Houston, TX 77477
http://www.ti.com/c2000

Revision Information

This is version V1.02.00.00 of this document, last updated on March 20, 2020.

Table of Contents

1 Introduction

The Texas Instruments TMS320C28x Integer Division Unit (FASTINTDIV) is a set of specialized instructions to perform the integer division faster. It basically extends the capabilities of the C28x floating point CPU by adding instructions to support integer division operations in an optimal manner. The TI C28 Compiler support various intrinsics to enable C/C++ programmers to take full advantage of the aforementioned hardware accelerators to speed up computation time for integer division.

This document provides a description of how to utilize this fast division architecture in software. A sample example project which can be found in the **examples** directory is also provided in the package to showcase the usage of these fast division intrinsics in the code

chapter 2 provides a host of resources on the FASTINTDIV in general, as well as training material.

chapter 3 describes the directory structure of the FASTINTDIV package.

chapter 4 provides step-by-step instructions on how to enable FASTINTDIV based integer division.

chapter 5 lists the performance of each these division intrinsics.

chapter 6 provides a revision history of the package.

2 Other Resources

The user can refer to Integer Division Unit chapter of **TMS320C28x Extended Instruction Sets (SPRUHS1B)** for detailed description of FASTINTDIV

For more details on fast integer division intrinsics definitions, macros, and additional background information, please see the **TMS320C28x Optimizing C/C++ Compiler User's Guide (spru514q)** and the **TMS320C28x Assembly Language Tools User's Guide (spru513q)**.

Also check out the C2000 portfolio at: http://www.ti.com/microcontrollers/c2000-real-time-control-mcus/overview.h

And don't forget the TI community website: http://e2e.ti.com

Building the examples with FASTINTDIV requires **Codegen Tools v20.2.1.LTS or later**

3 Directory Structure

By default, the sample example code and documentation is installed into the c2000ware directory under the sub-folder

```
C:\ti\c2000\c2000ware_2_00_00_02\libraries\math\FASTINTDIV
```

Figure. 3.1 shows the directory structure while the subsequent table 3.1 provides a description for each folder.

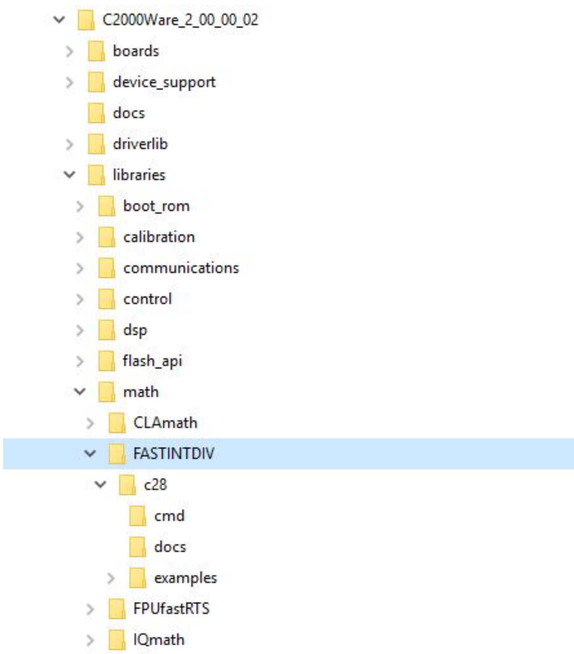

Figure 3.1: Directory Structure of the FASTINTDIV

Folder	Description
<base>	Base install directory. By default this is C:/ti/c2000/c2000ware_2_00_00_02/libraries/math/FASTINTDIV. For the rest of this document <base> will be omitted from the directory names.
<base>/c28/docs	Documentation for the current revision of the FASTINTDIV software including revision history.
<base>/c28/	Sample Example that demonstrate usage of the fast integer division intrinsics and validated for F2838x, F28003x and F28002x devices using the CCS 10 IDE.
<base>/c28/cmd	Contains linker command files for RAM and Flash mode configuration.

Table 3.1: FASTINTDIV Directory Structure Description

4 Using the FASTINTDIV Compiler Intrinsics

The sample example project with the usage of fast integer division intrinsics is provided. The user may import the example project(s) into CCSv10 and be able to build, run and validate the functionality of these intrinsics. (see Figure 4.1)

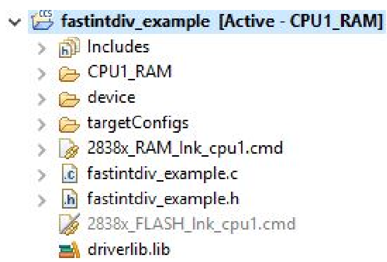

Figure 4.1: FASTINTDIV Example Project View

4.1 Enabling FASTINTDIV Hardware Support

Compiler option, –idiv_support, controls support for these fast division sequences. A value of 'none' implies no hardware support for the new instructions, and a value of 'idiv0' implies support for the current specification for the new instructions. The option is only valid when FPU32 or FPU64 is available (– float_support=fpu32 or fpu64) and when using the C2000 EABI (–abi=eabi).

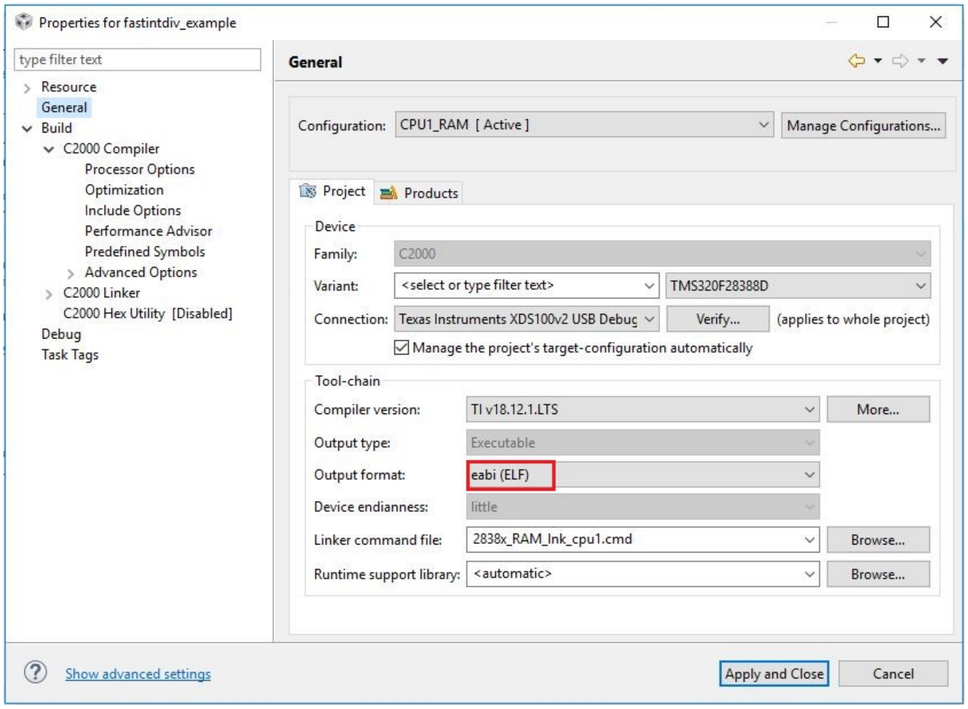

Figure 4.2: Setting EABI option

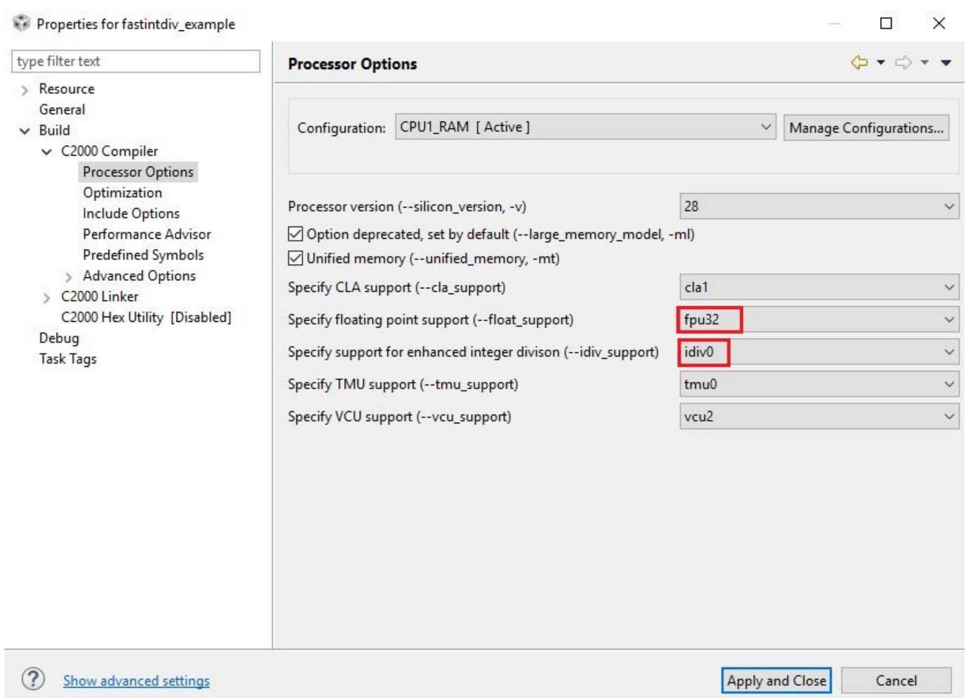

Figure 4.3: FASTINTDIV Example Build Configuration

4.2 Including Standard Library Header

All the FASTINTDIV intrinsics are declared in stdlib.h thus it is necessary to include this header in the main source file (see Figure 4.4). The intrinsics declared in stdlib.h take a numerator and denominator and return a structure containing both the remainder and quotient. The structure definition is also provided in stdlib.h.

```
//
// Included Files
//
#include "driverlib.h"
#include "device.h"
#include <stdlib.h>
#include "fastintdiv_example.h"

//
// Function Prototypes
//

// 16-bit by 16-bit
uint16_t test_traditional_div_i16byi16();
uint16_t test_euclidean_div_i16byi16();
uint16_t test_modulo_div_i16byi16();
uint16_t test_traditional_div_u16byu16();

// 32-bit by 32-bit
uint16_t test_traditional_div_i32byi32();
uint16_t test_euclidean_div_i32byi32();
uint16_t test_modulo_div_i32byi32();
uint16_t test_traditional_div_i32byu32();
uint16_t test_modulo_div_i32byu32();
uint16_t test_traditional_div_u32byu32();


// 32-bit by 16-bit
uint16_t test_traditional_div_i32byi16();
uint16_t test_euclidean_div_i32byi16();
uint16_t test_modulo_div_i32byi16();
```

Figure 4.4: Including stdlib.h in source file

4.3 Invoking FASTINTDIV Intrinsics

The FASTINTDIV hardware is capable of performing all three types of division i.e. traditional, euclidean and modulo between various data types in an optimal manner. Please refer to Integer Division Unit chapter of **TMS320C28x Extended Instruction Sets (SPRUHS1B)** to understand the fundamentals of types of division supported by FASTINTDIV. If 'idiv0' flag is enabled then division operator '/' and modulo operator '%' both will perform optimal traditional integer division by utilizing the FASTINTDIV hardware automatically and will return divisor and remainder accordingly. Also traditional integer division can be invoked by using the compiler intrinsics defined in stdlib.h as well. The euclidean / modulo division types are only supported by the defined intrinsics. The table below 4.1 describes how various types of integer division for numerator 'a' and denominator 'b' can be invoked to use FASTINTDIV hardware.

Division operation	Using C Operators	Using Intrinsics
Traditional 16-bit by 16-bit division	a/b	__traditional_div_i16byi16(a,b)
Euclidean 16-bit by 16-bit division	Not supported	__euclidean_div_i16byi16(a,b)
Modulo 16-bit by 16-bit division	Not supported	__modulo_div_i16byi16(a,b)
Traditional unsigned 16-bit by 16-bit division	a/b	__traditional_div_u16byu16(a,b)
Traditional 32-bit by 32-bit division	a/b	__traditional_div_i32byi32(a,b)
Euclidean 32-bit by 32-bit division	Not supported	__euclidean_div_i32byi32(a,b)
Modulo 32-bit by 32-bit division	Not supported	__modulo_div_i32byi32(a,b)
Traditional signed 32-bit by unsigned 32-bit division	a/(long)b	__traditional_div_i32byu32(a,b)
Modulo signed 32-bit by unsigned 32-bit division	Not supported	__modulo_div_i32byu32(a,b)
Traditional unsigned 32-bit by 32-bit division	a/b	__traditional_div_u32byu32(a,b)
Traditional 32-bit by 16-bit division	a/b	__traditional_div_i32byi16(a,b)
Euclidean 32-bit by 16-bit division	Not supported	__euclidean_div_i32byi16(a,b)
Modulo 32-bit by 16-bit division	Not supported	__modulo_div_i32byi16(a,b)
Traditional unsigned 32-bit by 16-bit division	a/b	__traditional_div_u32byu16(a,b)
Traditional 64-bit by 64-bit division	a/b	__traditional_div_i64byi64(a,b)
Euclidean 64-bit by 64-bit division	Not supported	__euclidean_div_i64byi64(a,b)
Modulo 64-bit by 64-bit division	Not supported	__modulo_div_i64byi64(a,b)
Traditional signed 64-bit by unsigned 64-bit division	a/(long long)b	__traditional_div_i64byu64(a,b)
Euclidean signed 64-bit by unsigned 64-bit division	Not supported	__euclidean_div_i64byu64(a,b)
Modulo signed 64-bit by unsigned 64-bit division	Not supported	__modulo_div_i64byu64(a,b)
Traditional unsigned 64-bit by 64-bit division	a/b	__traditional_div_u64byu64(a,b)

Table 4.1: Invoking various types of fast integer division

5 Benchmarks

The benchmarks for the various integer division intrinsics using FASTINTDIV hardware are shown in below table 5.1

Intrinsic	CPU cycles
__traditional_div_i16byi16(a,b)	16
__euclidean_div_i16byi16(a,b)	14
__modulo_div_i16byi16(a,b)	14
__traditional_div_u16byu16(a,b)	14
__traditional_div_i32byi32(a,b)	13
__euclidean_div_i32byi32(a,b)	14
__modulo_div_i32byi32(a,b)	14
__traditional_div_i32byu32(a,b)	14
__modulo_div_i32byu32(a,b)	14
__traditional_div_u32byu32(a,b)	12
__traditional_div_i32byi16(a,b)	18
__euclidean_div_i32byi16(a,b)	16
__modulo_div_i32byi16(a,b)	16
__traditional_div_u32byu16(a,b)	13
__traditional_div_i64byi64(a,b)	42
__euclidean_div_i64byi64(a,b)	42
__modulo_div_i64byi64(a,b)	42
__traditional_div_i64byu64(a,b)	42
__euclidean_div_i64byu64(a,b)	42
__modulo_div_i64byu64(a,b)	42
__traditional_div_u64byu64(a,b)	42

Table 5.1: Profiling of various types of fast integer division

6 Revision History

V1.04.00.00
Added support for F28003x devices.

V1.03.00.00
Migrated to compiler version 20.2.1.

V1.02.00.00
Added support for F28002x devices.

V1.00.00.00

- This version is the first release of the FASTINTDIV files and examples.

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 TI 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. Information of third parties may be subject to additional restrictions.

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.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

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
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless