Application Note

DCC Computation Tool for AM263x MCU

ABSTRACT

This application note describes the operation of the dual clock comparator (DCC) computation tool, made to be used with the Sitara™ AM263x MCU. This tool provides the pre-computed register values to be programmed, in order to check if any required system/peripheral clock frequency is within a specific accuracy % by comparing it against a reference clock, enabling easy integration into your application code.

The spreadsheet mentioned in this document can be downloaded from the following URL: https://www.ti.com/lit/zip/sprad69.

Table of Contents

Trademarks

1 Introduction

Dual clock comparator (DCC) is primarily intended to determine the clock accuracy during the application execution. The exact accuracy is programmable and should be calculated for each application. It can also be used to determine the frequency of any selectable clock, using another independent input clock as a reference. It uses two independent clock sources to detect when one is out of spec.

Clock frequency drift is inevitable, which can occur due to multiple reasons such as environmental conditions like humidity, pressure and temperature, aging, and so forth. Impact of aging can be noticed when the device has been on the field for a few years, and the clocks start drifting causing catastrophic implications if not monitored. Hence, DCC is particularly useful in safety critical automotive/ industrial timing applications.

DCC contains three counters – counter0 (20-bit), valid0 (16-bit) and counter1 (20-bit). Initially, all counters are loaded with their user-defined, pre-load value. Counter0 and counter1 start decrementing once the DCC is enabled at rates determined by the frequencies of clock0 and clock1, respectively. When counter0 equals 0 (expires), the valid0 counter decrements at a rate determined by clock0. If counter1 decrements to 0 in the valid window, then no error is generated and clock1 is considered to be good within allowable tolerance as configured by the user. An error is generated when the frequency is not within the allowable tolerance, and counter1 doesn't decrement within the valid window (either when clock0 /clock1 is absent or clock1 expired before the counter0 reaches 0 or clock1 expired after both counter0 and valid0 reach 0).

Counter0 and counter1 are configured based on the ratio between the frequencies of clock0 and clock1 (Fclk1×Counter0 = Fclk0×Counter1). The valid0 counter provides tolerance and is configured based on the allowed frequency error and the inherent DCC error (asynchronous and digitization error). Since clock0 and clock1 are asynchronous, the start and stop of the counters do not occur synchronously. Hence, while configuring the counters, two different sources of errors must be accounted for, which are:

- DCC Error due to the asynchronous timing of clock0 and clock1 - This depends on the frequency of clock0 and clock1:
  - If Fclk1 > Fclk0, then Async. Error (In Clock0 cycles) = 2 + 2×(Fsysclk/Fclk0)
  - If Fclk1 < Fclk0, then Async. Error (In Clock0 cycles) = 2×(Fclk0/Fclk1) + 2×(Fsysclk/Fclk0)
  - If Fclk1 is unknown, then Async. Error (In Clock0 cycles) = 2 + 2×(Fsysclk/Fclk0) Note: Fsysclk is 200MHz
- Digitization error - 8 Clock0 cycles

The following formulas are used by the tool to compute the counter0, valid0 and counter1 seed values:

- Counter0 Seed = Window - Total Error
- Valid0 Seed = 2 × Total Error
- Counter1 Seed = Window × (Fclk1/Fclk0)

Where,

- DCC Error (in Clock0 Cycles) = Async. Error + 8 Clock0 Cycles (Digitization error)
- Minimum accuracy possible (%) = (100 × DCC Error × (Fclk1/Fclk0)) / 1048575
- Window (in Clock0 Cycles) = (DCC Error) / (0.01×Minimum accuracy possible (in %))
- Frequency Error Allowed (In Clock0 Cycles) = Window × (Minimum accuracy possible (in %) / 100)
- Total Error (in Clock0 Cycles) = DCC Error + Frequency Error Allowed

The DCC computation tool provides the register values to be programmed in-order to compare any given system or peripheral clock against a configurable reference clock to determine if the frequency of the said clock is within the expected accuracy. The accuracy required can be provided as an input, and the tool calculates the counter seed values based on that. There is a separate calculator present for each of the four instances: DCC0, DCC1, DCC2 and DCC3.

2 Procedure to use the Tool

Identify the DCC instance that can be used to monitor the clock source that you require, from the tool or the device technical reference manual.

1. In the (1) Input Section:
   a. Input the frequency of clock being provided as external reference clock in the field EXT_REFCLK frequency input in the DCC instance section being used.
   b. Input the clock source to be used as reference clock for comparison, DCCCLKSSRC0 (Reference Clock) using the provided drop-down menu. The options present are EXT_REFCLK, RCCLK10M (10 MHz), RCCLK32K (32 KHz) and XTALCLK (25 MHz).
   c. Input the clock source to be used for verification, DCCCLKSSRC1 (Clock to be verified) using the provided drop-down menu. The options present are EXT_REFCLK, R5SS0_CLK (400 MHz), R5SS1_CLK (400 MHz), RCCLK32K (32 KHz), SYSCLK (200 MHz) and XTALCLK (25 MHz).
   d. Input the clock accuracy % required for your application. If no input is provided, the tool uses the default value of 0.2 or the minimum possible accuracy (whichever is higher).
   e. Based on the clock sources provided as input, Async Error, DCC Error, Minimum accuracy possible, Window, Frequency Error Allowed, Total error (formula defined in previous section) are computed by the tool.

   Note

   Minimum accuracy possible cannot be greater than 48 (beyond which the counter seed value goes to zero), in such clock source combinations, an “ERROR” is displayed.

2. From the (2) Register Configuration Section:
   a. Program the DCCCNTSEED0 (@0x00000008) with the hexadecimal of the Counter0 seed value computed in previous step.
   b. Program the DCCCNTSEED1 (@0x00000010) with the hexadecimal of the Counter1 seed value computed in previous step.
   c. Program the DCCVALIDSEED0 (@0x0000000C) with the hexadecimal of the Valid0 seed value computed in previous step.
   d. Program the DCCCLKSRC0 (@0x00000028) with the corresponding (A000 + hexadecimal index of the clock source 0 that is chosen).
   e. Program the DCCCLKSRC1 (@0x00000024) with the corresponding (A000 + hexadecimal index of the clock source 1 that is chosen).
   f. After all the above configuration is done, set the DCCGCTRL (@0x00000000) with 0x0000AAAA to set the comparison in SINGLE_SHOT mode, enable done signal, enable error signal and start the comparison.

   Note

   For continuous mode, configure bits 11:8 with 0101.

   g. At the end of the DCC completion phase, the DCC generates a DCC_done interrupt to R5F if the clock is within the configured tolerance. The Done bit (bit 1) in the register DCCSTAT(@0x00000014) will be set to 1 to indicate the completion.

      If the DCC detects an error in the frequency measurement, then it generates DCCx_error to ESM (Error signaling module) instead of the DCC_done interrupt to R5F. The Error status (bit 0) in the register DCCSTAT(@0x00000014) will be set to 1 to indicate Error condition. In this case, the ESM can be configured to generate an interrupt to R5F or Device Error pin for further action. See the device technical reference manual for details on configuring and setting up the interrupts.

      If interrupt mode is not used, after configuring the SINGLE_SHOT mode, R5F can be configured to continuously poll Error status (bit 0) and Done bit (bit 1) in the register DCCSTAT(@0x00000014).

3. In case an error is obtained, the counter0, counter1 and valid0 values can be read out (from the address offsets 0x00000018, 0x00000020 and 0x0000001C, respectively) in-order to determine the type of error (clock0/clock1 is absent or clock1 expired before the counter0 reaches 0 or clock1 expired after both counter0 and valid0 reach 0).

The above example is for DCC0. Follow similar procedure to use any of the other DCC instances DCC1, DCC2 and DCC3.

3 References

- Texas Instruments: *AM263x Sitara™ Microcontroller Data Sheet*
- Texas Instruments: *AM263x Sitara™ Microcontroller Technical Reference Manual*
- Texas Instruments: *AM263x Sitara Processors Technical Reference Manual Register Addendum*
- AM263x MCU Plus SDK Read Me First Guide: https://www.ti.com/tool/MCU-PLUS-SDK-AM263X

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

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