Application Report
SPNA211–July 2015

Continuous Monitor of the PLL Frequency With the DCC

Kevin Lavery

ABSTRACT

The Dual Clock Compare (DCC) may be configured to provide autonomous, real-time monitoring of the average frequency of a signal. This application report illustrates a configuration in which the DCC monitors the average PLL frequency. This comparison triggers an error when the average PLL falls out of a specified range. The accuracy window and the duration over which the frequency is averaged must be determined. These parameters are tightly linked so that they are not independent, and trade-offs must be made based upon the application requirements.

Project collateral and source code discussed in this document can be downloaded from the following URL: http://www.ti.com/lit/zip/spna211.

Contents

List of Figures

List of Tables

1 DCC Theory of Operation

The DCC accepts two clock inputs: $Clock_0$ and $Clock_1$. These clocks decrement counters in order to compare their relative frequencies. Three different time periods are created by the DCC:

- T_0 is created by decrementing $Counter_0$ with $Clock_0$
- T_{Valid} is created by decrementing Valid with $Clock_0$, referred to as the valid window
- T_1 is created by decrementing $Counter_1$ with $Clock_1$

T_0 and T_{Valid} create the reference time and window for measuring T_1. The counters are configured so that T_1 expires within the valid window; if T_1 does not expire in the Valid window, an error is generated. This behavior is the purpose of the DCC.

Figure 1 shows the behavior of the DCC.

- Initially, all counters are loaded with their user-defined preload value.
- $Counter_0$ and $Counter_1$ decrement at rates determined by the frequencies of $Clock_0$ and $Clock_1$, respectively.
- When $Counter_0$ equals 0 (expires), the Valid counter begins decrementing at a rate determined by $Clock_0$.
- If $Counter_1$ decrements to 0 in the valid window, then:
  - No error is generated
  - All counters ($Counter_0$, Valid, $Counter_1$) are reloaded when Valid decrements to 0 (1).

(1) This circuit description also shows that the DCC does not monitor $Clock_1$ in the period from expiration of $Counter_1$ to expiration of Valid.

Counter$_1$ is shown decrementing at different rates in Figure 1.

- The rates that expire outside of the Valid window generate an error and are shown in red.
- The rates that expire within the Valid window do not generate an error; these rates are shown in black.

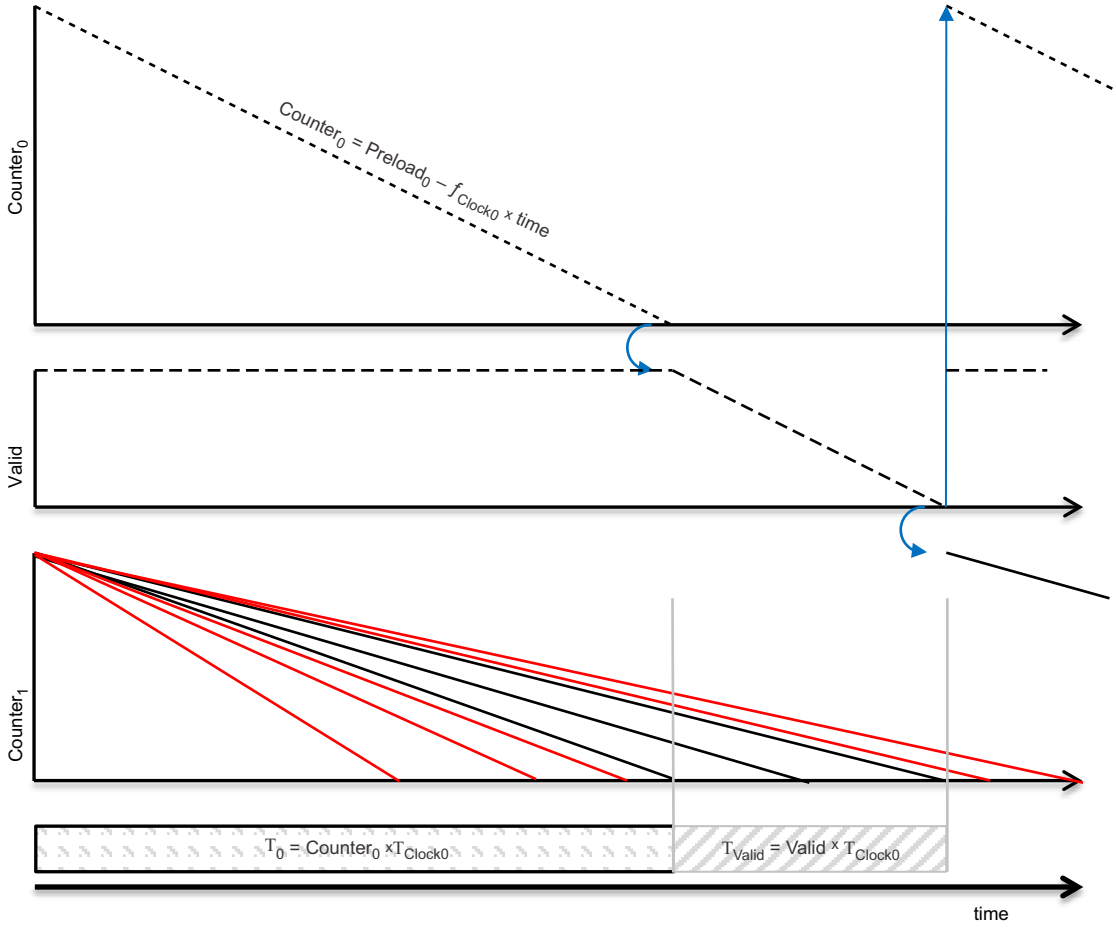

Figure 1. DCC Measurement

The expected operation of the DCC (with no errors) is shown in Figure 2. In this waveform, Counter$_1$ expires in the middle of the Valid count-down. After Valid has expired, all counters are again pre-loaded and Counter$_0$ and Counter$_1$ begins to decrement immediately.

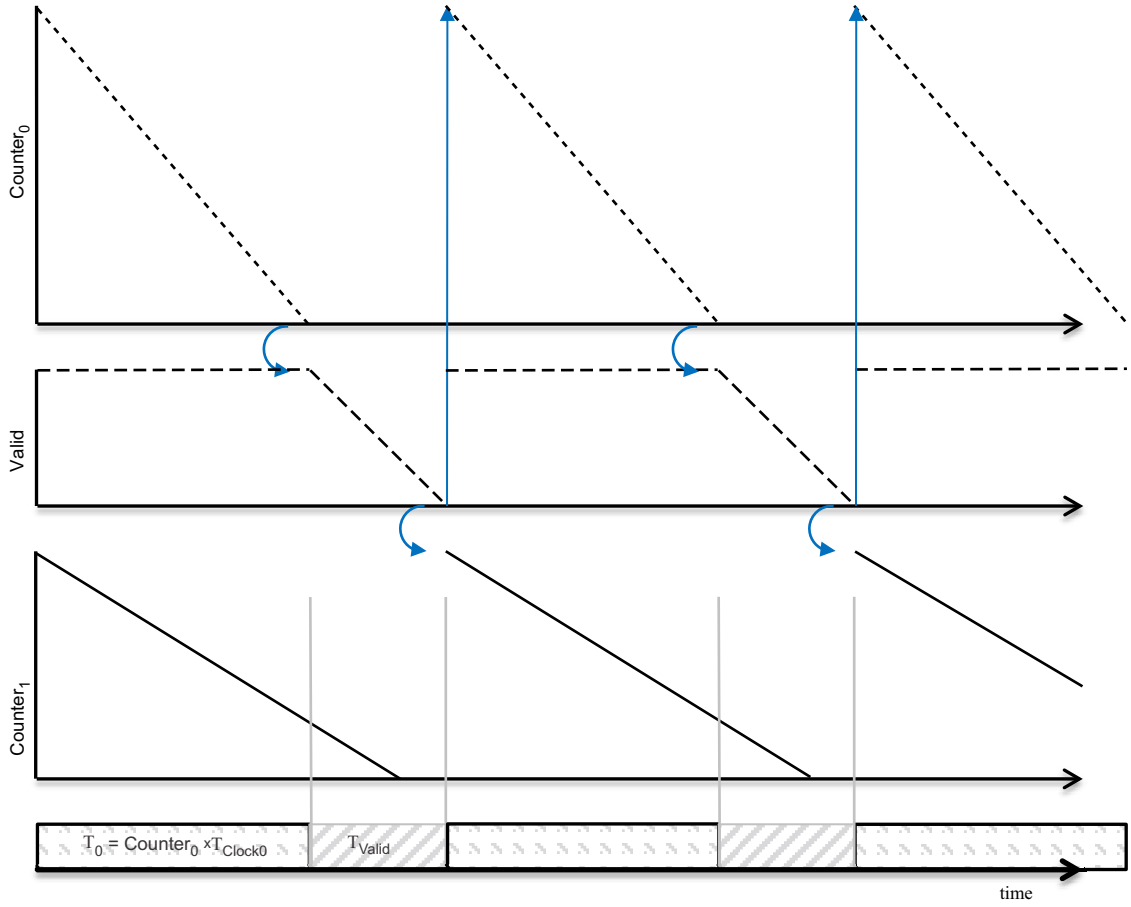

Figure 2. DCC Operating in Continuous Mode Without Error Generation

2 Calculating DCC Parameters

The counters are normally configured so that $Counter_1$ decrements to 0 in the middle of the Valid counter.

$$T_{Clock0}\left(Counter_0 + \frac{1}{2}Valid\right) = T_{Clock1} \times Counter_1 \quad (1)$$

The steps to setup this relation are:

1. Set $T_{Clock0} \times Counter_0 = T_{Clock1} \times Counter_1$
2. Compute Valid.
3. Adjust $Counter_0$ so that the new $Counter_0 = Counter_0 - \frac{1}{2}Valid$.

Given, a fixed period $Clock_0$ and $Clock_1$, the DCC Counters – $Counter_0$, Valid, and $Counter_1$ must be configured for optimal performance. The configuration of the Valid counter has some constraints, and these configuration guidelines for Valid form the subject of this application report.

3 Configuring DCC Valid

The Valid counter's configuration is determined by considering two different sources of errors:

- Errors due to the asynchronous timing of $Clock_0$ and $Clock_1$
- Digitization error

3.1 Errors Due to Asynchronous Timing of $Clock_0$ and $Clock_1$

Since $Clock_0$ and $Clock_1$ are asynchronous, the mechanism for loading the counters does not occur synchronously; this circuitry is shown in Figure 3 The diagram shows that Down Counter 0 and Down Counter 1 cannot start at the same time because Reload is synchronized to $Clock_0$ and $Clock_1$. There are two conditions to consider:

- $T_{Clock0} > T_{Clock1}$ (for example, $f_{Clock1} > f_{Clock0}$), the offset requires 2 cycle offset in Valid (2).
- $T_{Clock1} > T_{Clock0}$ (for example, $f_{Clock0} > f_{Clock1}$), the offset requires $2 \times \frac{T_{Clock1}}{T_{Clock0}}$ cycle offset in Valid (3).

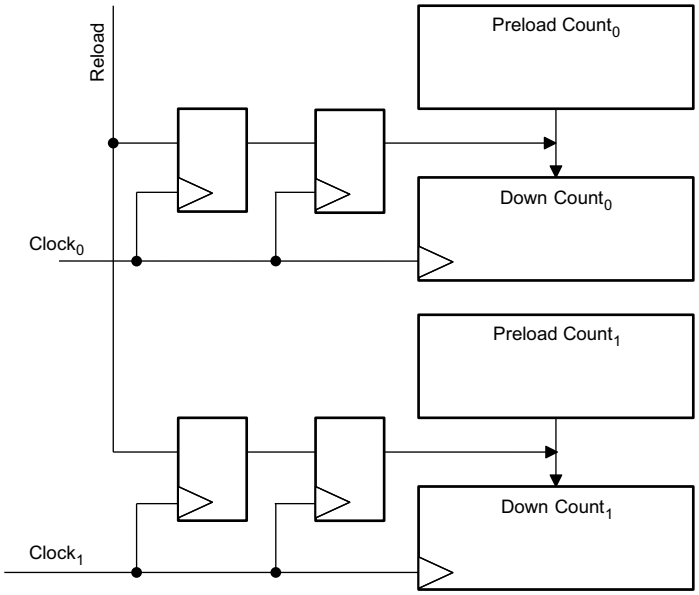

Figure 3. Asynchronous Timing Between $Clock_0$ and $Clock_1$ Starts Counters at Different Times

(2) The minimum number of Valid counts (excluding digitizing error) when $f_{Clock1} > f_{Clock0}$ is derived:

$$Valid_{Min} \times T_{Clock0} = 2T_{Clock0} - T_{Clock1}$$

$$Valid_{Min} \times T_{Clock0} = 2T_{Clock0} - \frac{T_{Clock1}}{T_{Clock0}} T_{Clock0}$$

$$Valid_{MIN} = 2 - \frac{T_{Clock1}}{T_{Clock0}} \rightarrow 2$$

(3) The minimum number of Valid counts (excluding digitizing error) when $f_{Clock0} > f_{Clock1}$ is derived:

$$Valid_{Min} \times T_{Clock0} = 2T_{Clock1} - T_{Clock0}$$

$$Valid_{Min} \times T_{Clock0} = 2\frac{T_{Clock1}}{T_{Clock0}} T_{Clock0} - T_{Clock0}$$

$$Valid_{Min} = 2\frac{T_{Clock1}}{T_{Clock0}} - 1 \rightarrow 2\frac{T_{Clock1}}{T_{Clock0}}$$

3.2 Digitization Error

Additionally, the counters have digitizing error. The code assigns 3 Clock 0 cycles for digitizing error.

3.3 Minimum Valid Count

The minimum count for Valid is expressed in terms of the synchronization error and the digitization error. The valid count is configured to be symmetric to the error sources (which gives a multiplication factor of 2).

$$Valid_{Min} = 2\left(Synchronization + Digitization\right) = \begin{cases} 2\left(2+3\right) = 10 & T_{Clock0} > T_{Clock1} \\ 2\left(2 \times \dfrac{T_{Clock1}}{T_{Clock0}} + 3\right) & T_{Clock1} > T_{Clock0} \end{cases} \quad (2)$$

4 Relationship Between Resolution and Duration

It is obvious that, independent of error terms, there is a relationship between the accuracy of the frequency measurement and the duration of the count. Since the frequency is correct if $Counter_1$ expires within the Valid window, the valid window represents an uncertainty in the timing.

$$\frac{\frac{1}{2}T_{Valid}}{T_0} = \frac{1}{Resolution} \quad (3)$$

This resolution of the measurement can be maximized (as a percentage) by extending the duration of the accumulation. The duration can be express as [(4)]:

$$Duration = \frac{Valid \times Resolution}{2f_{Clock0}} \quad (4)$$

in which the Resolution is expressed as 1 part in Resolution (1% is 1/100 and Resolution is 100; 0.2% is 1/500 and Resolution is 500).

(4) The relationship between Valid, Resolution and Duration is derived:

$$\frac{\pm\Delta t}{T} = \frac{1}{Resolution}$$

$$\frac{\pm\Delta t}{T} = \pm\frac{\frac{Valid}{2} \times T_{Clock0}}{Counter_0 \times T_{Clock0}} = \frac{1}{Resolution}$$

$$\pm\frac{Valid \times Resolution}{2f_{Clock0}} = Counter_0 \times T_{Clock0} = Duration$$

4.1 Setting $Counter_0$, Valid and $Counter_1$

There is a trade-off between Duration and Resolution (see Equation 4). As the resolution is increased, so is the duration of the sample. For a given resolution, the duration of the sample can be minimized by:

- Minimizing Valid
- Using the fastest (accurate) $Clock_0$

Since the oscillator is usually determined by other system criteria, the minimum duration (for a given resolution) is achieved by selecting the minimum Valid.

5 Example Calculations

In this example, the PLL frequency will be monitored with the main oscillator (OSCIN). The OSCIN will be selected as $Clock_0$ and a PLL will be selected as $Clock_1$.

PLL Frequency	160 Mhz
External Oscillator	16 Mhz
Desired Frequency Accuracy	0.1%

1. Calculate the minimum Valid Counter using Equation 2. Since $T_{Clock0} > T_{Clock1}$
   $Valid_{Min}$ =2×(Synchronization + Digitization)
   When $T_{Clock0} > T_{Clock1}$, $Valid_{Min}$ =2 × (2 + 3) = 10.
2. Use the frequency accuracy (see Equation 3) to find the minimum duration for the sample.
   With a frequency accuracy of 0.1%, this translates to $\frac{0.1}{100} = \frac{1}{1000} = \frac{1}{Resolution}$. Thus, the minimum duration of the sample is:

$$Duration_{Min} = \frac{Valid_{Min} \times Resolution}{2f_{Clock0}}$$
$$Duration_{Min} = \frac{10 \times 1000}{2 \times 16[MHz]}$$
$$Duration_{Min} = 312.5\ \mu s \quad (5)$$

   With the duration, it is easy to compute the $Counter_0$ and $Counter_1$ values.

$$T_{Clock1} \times Counter_1 = \frac{Counter_1}{f_{Clock1}} = Duration$$
$$\frac{Counter_1}{160[MHz]} = 312.5\ [\mu s]$$
$$Counter_1 = 5 \times 10^4 \quad (6)$$

   and $Counter_0 = 5 \times 10^3 - 5 = 4995$.

Summary of the Results:

With 16 MHz $Clock_0$ and 160 MHz $Clock_1$ and accuracy requirements of 0.1%:

- $Valid_{Min}$ = 10
- Resolution = 1000
- DurationMin = 312.5 µs
- $(Counter_0)_{Min}$ = 4995
- $(Counter_1)_{Min}$ =50000

6 Code Configuration

The configuration of the DCC is carried out in the *dccInit_demo* function. The function contains error checking that is not discussed in this application report.

Table 1. Code Configuration

The function takes five arguments:

- CLK0_src – $Clock_0$ signal; this example code is written for use on DCC1 with the LAUNCHXL2-RM46 (which uses RM46L852) or LAUNCHXL2-TMS57012 (which uses TMS570LS1224). Clock Source 0 can accept:
  - OSCIN (0)
  - HF LPO (5)
  - TCK (0xA)

  For use with other devices, see the device-specific datasheet. This clock source is programmed in line 36.
- CLK1_src – $Clock_1$ signal; DCC1 for RM46L852 or TMS570LS1224 accepts Clock Source 0 can accept:
  - N2HET1[31]
  - PLL1 (0)
  - PLL2
  - LF HPO
  - HF LPO
  - EXTCLKIN1
  - EXTCLKIN2
  - VCLK

  For use with other devices, see the device-specific data sheet. This clock source is programmed in line 34.
- CLK0_freq – frequency of $Clock_0$ in Hertz.
- CLK1_freq – frequency of Clock1 in Hertz.

The frequency of $Clock_0$ and $Clock_1$ are compared in Line 6. Valid is computed on lines 7 and 9 as in Equation 2. The frequency of $Clock_0$ and $Clock_1$ are used to compute $Counter_0$ and $Counter_1$ in Lines 20 and 21.

- Resolution – frequency accuracy of the measurement. The accuracy is passed as $\frac{1}{Resolution}$. That is, a 1/2 % accuracy has a resolution of 1 part in 200, and Resolution is passed as 200. Resolution is required to compute the duration in Line 12 following Equation 4.

$Counter_0$, Valid, and $Counter_1$ are passed to dccSetSeed() in Line 24.
The error response of the DCC is configured in Line 27.
Line 38 enables the DCC.

1	unsigned int dccInit_demo(unsigned int CLK0_src, unsigned int CLK1_src, unsigned int CLK0_freq, unsigned int CLK1_freq, unsigned int resolution)
2	{
3	unsigned int valid0seed_val;
4	double long cnt0seed_val, cnt1seed_val, duration;
5	
6	if(CLK1_freq >= CLK0_freq)
7	valid0seed_val = 10; // derived in accompanying application note
8	else
9	valid0seed_val = 2*((2*(CLK0_freq/CLK1_freq) + 1) +3);
10	
11	// duration is computed in accompanying application note.
12	duration = (double long) valid0seed_val*(double long)resolution/2;
13	
14	
15	// CLK0_freq provides the number of CLK0 cycles to count.
16	// (CLK1_freq/CLK0_freq)*duration provides the number of CLK1 cycles to count.
17	
18	// in order to center the expiration of the CLK1 counter within the CLK0 counter window,
19	// cnt0seed is programmed with the number of edges in one-half the valid counts.
20	cnt0seed_val = duration - valid0seed_val/2;
21	cnt1seed_val = (double long)CLK1_freq*duration/(double long)CLK0_freq - 1;
22	
23	// Call HalCoGen function that sets count 0, count 1, and valid
24	dccSetSeed(dccREG1, (unsigned int) cnt0seed_val, valid0seed_val, (unsigned int) cnt1seed_val);
25	
26	// Generate error if PLL count expires outside of the valid window
27	dccREG1->GCTRL = ((dccREG1->GCTRL & 0xFFFF0F0FU) | dccNOTIFICATION_ERROR | 0x5000);
28	
29	
30	// Configure clock sources
31	if(CLK1_src == 100)
32	dccREG1->CNT1CLKSRC = 0;
33	else
34	dccREG1->CNT1CLKSRC = (0xA000 | CLK1_src);
35	
36	dccREG1->CNT0CLKSRC = CLK0_src;
37	
38	// Call HalCoGen function to enable DCC1
39	dccEnable(dccREG1);
40	
41	// returns the time duration over which the frequencies are compared.
42	return duration;
43	}

7 Example Code

The key purpose of the application report and example code is to demonstrate the DCC configuration. In order to demonstrate the DCC's monitoring capabilities, the code makes small adjustments to the PLL's frequency.

7.1 Hardware

The source code is written for use on either LAUNCHXL2-RM46 or LAUNCHXL2-TMS57012. This hardware provides two user controlled buttons : User Switch A and User Switch B.

- User Switch A increases the frequency of the PLL.
  - If the PLL is at its baseline frequency, then the increased frequency generates a DCC failure based on running too fast. (In terms of Figure 1, $Counter_1$ expires before $Counter_0$ and outside of the Valid window.)
  - If the PLL is running too slow, then the increased frequency returns the PLL to its baseline frequency. At its baseline frequency, the PLL expires within the Valid window and does not generate a DCC error.
  - If the PLL is running too fast, then User Switch A does not affect the PLL frequency since it is already too fast.
- User Switch B decreases the frequency of the PLL (similar to User Switch A)
  - If the PLL is at its baseline frequency, then the decreased frequency generates a DCC failure based on running too slow. (In terms of Figure 1, $Counter_1$ does not expire until after the Valid window.)
  - If the PLL is running too fast, then the decreased frequency returns the PLL to its baseline frequency. At its baseline frequency, the PLL expires within the Valid window and does not generate a DCC error.
  - If the PLL is running too slow, then User Switch A does not affect the PLL frequency since it is already too slow.

The code is written to be executed from flash, be sure to load the correct code into the device. The same source code is compiled for either RM46 or TMS570 and the output code is not interchangeable between the hardware.

7.2 Software

The software is built from HalCoGen. Rather than using the dccInit function from Halcogen, the code develops the dccInit_demo function in order to define the relationship between $Counter_0$, Valid, and $Counter_1$. Outside of the DCC function, the software.

- Sits in an infinite loop, monitoring User Switch A and User Switch B
- Periodic interrupts are generated from the Real Time Interrupt Module (RTI) in order to toggle the LED

The PLL frequency is changed (based on User Switches A and B) by an amount that scales to the user-defined resolution. The frequency offset is computed as:

$$\Delta f = \frac{f}{\left(1 + \frac{Duration\left[inCLK0\right]}{Valid}\right)} \tag{7}$$

Table 2. Frequency Offset Computation

Resolution	Frequency Accuracy [%]	Frequency Offset [MHz]
10	10%	26.71
20	5%	14.56
50	2%	6.16
100	1%	3.14
200	0.5%	1.58
500	0.2%	0.64
1000	0.1%	0.32
2000	0.05%	0.16
5000	0.02%	0.06
10000	0.01%	0.03

NOTE: Code is written to clarify the relation to the DCC Concepts, not to optimize code execution.

While this application report has dealt with monitoring the PLL frequency with the DCC, the DCC can monitor other clock sources. If the application generates a 100KHz output on N2HET1[31], the frequency can be monitored by the DCC. In this case, the function dccInit_demo is called as dccInit_demo(0, 100, 16000000, 100000, resolution).

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

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

TI does not 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 relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license 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 significant portions 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. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

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