This document describes how to use Clock Tree Tool (CTT).

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTT Installation/Uninstallation</td>
<td>2</td>
</tr>
<tr>
<td>CTT Overview</td>
<td>5</td>
</tr>
<tr>
<td>CTT System Requirements</td>
<td>6</td>
</tr>
<tr>
<td>CTT Running Requirements</td>
<td>6</td>
</tr>
<tr>
<td>CTT GUI (Graphical User Interface) Description</td>
<td>6</td>
</tr>
<tr>
<td>CTT Blocks</td>
<td>16</td>
</tr>
<tr>
<td>CTT Release Notes</td>
<td>26</td>
</tr>
<tr>
<td>CTT Limitations</td>
<td>26</td>
</tr>
</tbody>
</table>
1.1.1 CTT Installation Step 1

To install the Clock Tree Tool double click (java -jar in terminal for Linux users) on the "Installer-CTT-xxxx" file. The installer will execute and display the License Agreement window shown in Figure 1. The user must accept the conditions of the license in order to install the CTT.

Figure 1. CTT Installation License Agreement Window
1.1.2 CTT Installation Step 2

When the conditions of the license agreement are accepted, the "Install" button is enabled, see Figure 2. Click on the "Install" button to proceed to the "Destination Directory Selection" window, see Figure 3. It allows the user to identify the directory for installation of the Clock Tree Tool. Once the directory is selected click the "Select" button to start the installation.

Figure 2. CTT Installation License Agreement Window 2

<table>
<thead>
<tr>
<th>INSTALLER:</th>
<th>Clock Tree Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>To install</td>
<td>Clock Tree Tools</td>
</tr>
<tr>
<td>you must accept all the conditions of this License Agreement</td>
<td></td>
</tr>
</tbody>
</table>

**IMPORTANT — PLEASE CAREFULLY READ THE FOLLOWING LICENSE AGREEMENT, WHICH IS LEGAL BINDING.** AFTER YOU READ IT, YOU WILL BE ASKED WHETHER YOU ACCEPT AND AGREE TO ITS TERMS. DO NOT CLICK "I HAVE READ AND AGREE" UNLESS: (1) YOU WILL USE THE LICENSED MATERIALS FOR YOUR OWN PERSONAL USE AND PERSONALLY ACCEPT, AGREE TO AND INTEND TO BE BOUND BY THESE TERMS; OR (2) YOU ARE AUTHORIZED TO, AND INTEND TO BE BOUND BY, THESE TERMS ON BEHALF OF YOUR COMPANY.

Important — Read carefully: In this Agreement "you" means you personally if you will exercise the rights granted for your personal benefit, but it means your company (or you on behalf of your company) if you will exercise the rights granted for the company’s benefit. This evaluation, development and demonstration software license agreement ("Agreement") is an agreement between you and Texas Instruments Incorporated ("TI"). The "Licensed Materials" subject to this Agreement include the software programs and any associated electronic documentation (in each case, in whole or in part) set forth.
1.1.3 **CTT Installation Step 3**

When the installation is complete the "Installation completed" message is displayed Figure 4. Click on "OK" button to proceed to the last window confirmation window, see Figure 5. There, click "Exit" to complete the CTT installation.

**Figure 3. CTT Destination Directory Selection Window**

**Figure 4. CTT Installation Complete Message Window**
1.2 CTT Uninstallation

The Clock Tree Tool does not leave application traces of its installation in the OS applications, features, and registry. So to uninstall it, simply navigate to the installation folder and delete it.

2 CTT Overview

The Clock Tree Tool (CTT) is a Java™ based stand-alone application. It is an interactive clock tree configuration software for the device. It allows the user to:

• visualize the device clock tree
• interact with clock tree elements and view the effect on clock-tree configuration registers
• interact with the clock-tree configuration registers and view the effect on the device clock tree
• view a trace of all the device registers affected by the user interaction with clock tree

The advantage of the tool is that the user can visualize the device clock tree state on power-on reset and then customize the configuration of the clock tree for the specific use-case and identify the device register settings associated to that configuration.

Being an interactive visual tool, the CTT gives the user a global view of the device clock tree architecture and allows determining the exact register settings to obtain the specific configuration.
3 CTT System Requirements

- Requires Java JRE 1.8 or higher (Can be downloaded from https://java.com/en/download/).
- Has been tested for Microsoft Windows ® 10.
- The ideal screen resolution is 1920x1080 (4k displays are also supported, however some renderings in the CTT view can appear small due to the high resolution).

4 CTT Running Requirements

The start-up sequence of the CTT consists of reading an entire clock tree description database files. This would normally take about 10 to 20 seconds.

Similarly, the View Refresh function that updates the main view, covers the entire clock tree of the device and takes as well about 10 to 20 seconds.

**CAUTION**

Before running CTT package, Linux user should perform the following steps:

- In the console, go to CTT install folder.
- Run the following command:
  ```
  java -cp jGraphLib/lib/*:<name_of_CTT_package>.jar:. org.ti.clockTreeTool.simulation.ClockTreeTool
  ```

5 CTT GUI (Graphical User Interface) Description

5.1 CTT Views Description

The CTT GUI is composed of 5 sub-views (see Figure 6):

- Main View
5.1.1 CTT Main View

The Main View presents a focused view of a section of the device clock tree.

The device clock tree is represented as a tree structure composed of "nodes" or "blocks" (that is, the rectangular elements) and the "links" or "signals" (that is, the arrows). The direction of the signal identifies the source and the destination blocks of the signal. A block may be a source block to multiple blocks and may in turn have multiple source blocks connected to it.

The clock tree has following types of blocks:

- Crystal
- Clock Source
- Oscillator
CTT GUI (Graphical User Interface) Description

- Pin
- Pin-Clock Source
- Clock Switch (Hardware/Manual/Automatic)
- Multiplexer
- Divider
- DPLL
- Module

(Note: Refer to the device Technical Reference Manual for the description of these blocks)

The user can mouse drag or use the scroll bars to move about the view. The view highlights the state of the blocks and the signals visually. For example, the state of a clock switch (Open/Close) is presented by a red open switch or a green close switch symbol. Similarly, the state of a clock signal (Active/ Gated) is highlighted by the signal being green or red.

5.1.2 CTT Thumbnail View

The Thumbnail View highlights a global view of the device clock tree. It also highlights the region of the clock tree visible in the Main View by a bounding rectangle. As the slide bars of the Main View are displaced the bounding rectangle in the Thumbnail View also moves accordingly.

Figure 8. CTT Thumbnail View

5.1.3 CTT Controller View

The Controller View highlights a signal or a block of the clock tree. The user selects (that is, clicks on) the signal/ block in the Main View and it is highlighted in the Controller View. If a signal is selected, its current frequency is presented, whereas, if a block is selected, depending on the block type its parameters are presented.
5.1.4 CTT Registers View

The Registers View is composed of a Register Selector list box, on the left hand side. The name of the currently selected register is highlighted in this box. There are two types of registers view - new-style format, see Figure 10, and old-style format, see Figure 11.

On the upper right hand side of the Register View is the Register Address/Value indicator. It presents the address and the current hexadecimal value of the register. In the new-style format, the user may type in a different address to select a new register, or can type in new register value to reconfigure the register bitfields.

Below these two is a Register Bits view. The register bits view lists all the bits/bitfields of the selected register (for example, 0 to 31 bits for a 32 bits register). Each bit is identified by the bit number (0 for the LSB). In the old-style format, below the bit number is the current value of the bit (1/0). In the new-style format, between the current value of the bit/bitfield is its name.

A toggle button below the bit number of the user configurable (that is read/write) bits allows the user to toggle the bit value. Pressing the button sets the bit value to 1 and in the released state the bit value is 0. There is no button associated to the RESERVED bits of the register (that is, the user cannot modify the states of these bits).

When the user selects a register in the Register Selector list box, its contents (that is, bits and value) are highlighted in the Register Bits view and the Register Value indicator.

When the user changes a parameter of a block in the Controller View, the associated bitfield is updated in the register and the Register View highlights the affected register.

When the value of a bit/bitfield of a register changes in the register view, the Trace View captures this change also.

---

**NOTE:** When the user changes a parameter of a block which affects bitfields of more than one register, the Registers View only shows the last register updated. The Trace View shows the complete list of registers affected by this change.
In the new-style format, when user positions the pointer on the name of the bit/bitfield a pop-up displays the name of the corresponding bitfield.

In the old-style format, when user positions the pointer on the number of a register bit a pop-up displays the name of the corresponding bitfield.

5.1.5 CTT Trace View

The Trace View allows the user keep track on register changes made anywhere from the GUI views.

NOTE: User can reset/clean the trace event log from Trace->Reset menu option.

5.2 CTT Zoom Control

The Zoom Control allows the user to change the zoom level of the Main View. By default the zoom level is set to x1. The user can zoom in by shifting the slider to the right hand side (towards x2) and zoom-out by shifting the slider to the left hand side (towards x0.1)

NOTE: A zoom in/out on mouse scroll and drag to move functionality is also available.
5.3 **CTT Search Bars**

The CTT search bars allows the user to navigate directly to the desired block or signal within the main view. There are 3 bars. The first two are for searching and navigating to a particular block. The third one is for searching signals.

After selecting the desired block or signal, the main view will automatically scroll and highlight the selected (from the bars) block or signal.

![Figure 14. CTT Search Bars](image)

5.4 **CTT Menu Commands Description**

The CTT menu has following commands:

![Figure 15. CTT Menu](image)

1. CTT Settings
   1. Power-on Reset
2. Trace
   1. Reset
3. View
   1. Hide Others
   2. Display All
   3. Hide Frequency
   4. Display Frequency
   5. Refresh View
   6. Print View
4. Save / Load
   1. Save Registers
   2. Save Registers (CTT only)
   3. Load Registers
   4. Save Source Clocks
   5. Load Source Clocks
5. Run (not applicable for all CTT packages)
   1. Frequency Analyzer
6. Help
   1. About Clock Tree Tool
   2. User Manual
   3. SRAS License Agreement

5.4.1 **CTT Settings**

(a) Power-on Reset - Triggers a power-on reset for all the configuration registers. All the registers are set to their reset values. As a result, the state of the clock tree is updated and reflects the state after power-on reset. (Note: When the CTT starts, the power-on reset is automatically triggered. Hence, the initial clock tree state is that of the device after power-on reset).
5.4.2 CTT Trace
   (a) Reset - Resets (clears) the Trace View table.

5.4.3 CTT View
   (a) Hide Others - When a clock signal is selected in the Main View and this command is selected from the
       menu, the CTT hides all the clocks not associated to the selected clock. A clock is considered associated
       to another clock if it is directly/indirectly a parent/child of the clock.
   (b) Display All - This command is used to redisplay the entire clock tree from a partial view (as a result of
       the Hide Others command).
   (c) Hide Frequency - This command hides the frequency value of the clock signals in the Main View.
   (d) Display Frequency - This command displays the frequency values of the clock signals in the Main View.
   (e) Refresh View - This command refreshes the Main View representation of the clock tree. It is used if the
       clock tree representation is not correct and the view needs to be refreshed.
   (f) For a particular reason a user may want to print the tree onto an image. This image could be helpful if
       one needs to have it on paper, or just look at it without the need to load the CTT. This may also help when
       a user want to create a CTT configuration and print it. Then create another one and print it. This way the 2
       or more print stamps can be compared and analyzed. When selected, the print option generated an image
       and saves it in the CTT install directory.

5.4.4 CTT Save / Load Project

5.4.4.1 CTT Save / Load Registers
   The Save / Load Registers menus allow the user to either configure the clock-tree configuration registers
   (used in the CTT) to specific settings given in a file or to write the current values of the registers of the
   CTT to a file. This way the configuration can be saved, reused, tweaked and shared between team for
   development and debug.
   There are two options for saving registers:
   • Save all registers (Clocking, Control Module, etc.), included in CTT - Save Registers
   • Save only registers used by CTT - Save Registers (CTT only)

   **CAUTION**
   When using Save Registers option, be aware that the time to load the .rd1/.rd2
   file is considerably long compared to the file generated using Save Registers
   (CTT only) option.

   The Load Registers functionality may also be used to read-in registers from actual hardware board for
   debug/reference purposes. In order to do this the user can use the GEL script in Code Composer Studio,
   CMM script in Lauterbach, or OMAPconf register print log.
   The scenario would be to have a running hardware connected and using one of the methods described
   above, print out a known CTT register dump format. A known format would be:
   DeviceName XXXXX_SRX.X
   <Register Address> <Register Value>
   Then, save the result with the above register format to a "*.rd1" file. Load the file into the CTT and the GUI
   will display the exact hardware clocking configuration at the time the registers were exported.
   The other way is also possible. User can configure a needed register settings from GUI, save registers to
   a "*.rd1" file (go to Save / Load -> Save Registers menu or to Save / Load -> Save Registers (CTT only)
   menu) and use this file to load configuration into the connected HW via the environment used.
The following sequence have to be followed for a memory dump in CCS, see Figure 16:

1. In Code Composer Studio load the appropriate target configuration and run it.
   Note: For more information on Code Composer Studio set up and usability refer the Code Composer Studio Help.

2. Wait while the connection to the target is established and the target is initialized.

3. Then from the CCS menu go to “Tools -> Gel Files”. Right click in the window that just opened and load CTT-<Device_Name>_Device_SR->REG_DUMP.gel script. A menu will appear called “Scripts”.

4. Click on Scripts -> <Device_Name>_Device_SR> Clock Tree Tool -> Memory Dump

5. During execution, the GEL script prints the needed CTT registers with values in the console.

NOTE: A set of gel files are provided in the CTT installation folder. See <CTT isstall dir>\Scripts\.

Figure 16. CCS View for Memory Dump

Creating an “.rd1” or “.rd2” files (CTT compatible files). Note that the following example is presenting creating an “.rd1” file. Creating “.rd2” file is similar.

1. Copy the output from step 5 from the sequence of a memory dump in CCS into a text file - copy everything from “DeviceName” to the end of the data that the script outputs in the console.

2. In a text file paste the data from step 5 from the sequence of a memory dump in CCS.

   • It should end up with something like:
     CortexA9_0: GEL Output: DeviceName <Device_Name>_Device_SR
     CortexA9_0: GEL Output: 0x4A004100 0x00000110
     CortexA9_0: GEL Output: 0x4A004108 0x00000000
     CortexA9_0: GEL Output: 0x4A004110 0x00000000

   • Delete any prefixes (in this example the prefix is “CortexA9_0: GEL Output: ”)
     DeviceName <Device_Name>_Device_SR
     0x4A004100 0x00000110
     0x4A004108 0x00000000
     0x4A004110 0x00000000
3. Save the text file as "<filename>.rd1".
4. Load the created ".rd1" file in the CTT (go to go to Save / Load -> Load Registers menu). The tool will now display a snapshot of the entire system clock configuration of your board at the time the gel file was executed.

5.4.4.1.2 OMAPconf Memory Dump under Linux

A console have to be opened wherever is convenient to the user. After the connection to the target is established and the target is initialized. Run the following command:

root@am57xx-evm:~# omapconf export ctt am57xx-clocks.rd1
OMAPCONF (rev v1.73 built Tue Sep 26 18:56:57 EDT 2017)

HW Platform:
Generic DRA72X (Flattened Device Tree)
DRA72X ES2.0 GP Device (STANDARD performance (1.0GHz))
TPS65917 ES2.2

SW Build Details:
Build:
    Version: _____ _____ _ _
Kernel:
    Version: 4.9.41-rt23-gc038d21a22
    Author: gtbdadm@ubuntu-16
    Toolchain: gcc version 6.2.1 20161016 (Linaro GCC 6.2-2016.11)
    Type: #2 SMP PREEMPT RT
    Date: Tue Sep 26 19:03:11 EDT 2017

Output written to file './am57xx-clocks.rd1'

The last line will print a CTT compatible file.

NOTE: A gel and a cmm script files can be found in <CTT-install-dir>/Scripts/. For OMAPconf PRCM register dump refer to OMAPconf user guide https://github.com/omapconf/omapconf/wiki.

OMAPconf currently supports TI DRA75x, DRA74x, DRA72x, TDA2x, TDA2Ex, AM572x, AM571x, AM335x, AM437x, OMAP44xx AND OMAP54xx devices.

5.4.4.2 CTT Save / Load Source Clocks

In a given scenario, the user would be using a particular hardware board with different main input source clocks. These source clocks are not directly tied to a register configuration. Therefore, when configured from CTT GUI these source clocks must be saved to be loaded later. See Section 6.4 for details on clock source configuration from GUI.

5.4.5 CTT Run

NOTE: CTT Run menu is not applicable for all CTT packages.

(a) Frequency Analyzer - compares the specific-device Data Manual (DM) Max Frequency table vs live CTT configurations and displays a friendly GUI log with proper error/warning/ok messages. The image shown in Figure 17 represent a sample frequency analyzer view.
### Figure 17. CTT Frequency Analyzer

<table>
<thead>
<tr>
<th>Instance Name</th>
<th>Status</th>
<th>PRCM Clock Name</th>
<th>Max. Clock Allowed (MHz)</th>
<th>Current Clock Running (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMC4</td>
<td>⨯</td>
<td>L4PER_32K_GFCLK</td>
<td>6.032</td>
<td>6.044262</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>L4PER_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>MMC4_GFCLK</td>
<td>48.0</td>
<td>6.75</td>
</tr>
<tr>
<td>MMC3</td>
<td>✔</td>
<td>L4PER_32K_GFCLK</td>
<td>6.032</td>
<td>6.044262</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>L4PER_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>MMC3_GFCLK</td>
<td>48.0</td>
<td>6.75</td>
</tr>
<tr>
<td>MMC2</td>
<td>⨯</td>
<td>L3INT_32K_GFCLK</td>
<td>6.032</td>
<td>6.044262</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>L3INT_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>MMC2_GFCLK</td>
<td>192.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MMC1</td>
<td>⨯</td>
<td>L3INT_32K_GFCLK</td>
<td>6.032</td>
<td>6.044262</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>L3INT_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>MMC1_GFCLK</td>
<td>192.0</td>
<td>0.0</td>
</tr>
<tr>
<td>USB2</td>
<td>✔</td>
<td>USB_OTG_SS_REF_CLK</td>
<td>38.4</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>L3INT_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>L3INT_600M_GFCLK</td>
<td>960.0</td>
<td>0.0</td>
</tr>
<tr>
<td>USB1</td>
<td>✔</td>
<td>USB_OTG_SS_REF_CLK</td>
<td>38.4</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>L3INT_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>USB_LPS_TX_GFCLK</td>
<td>34.3</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>L3INT_600M_GFCLK</td>
<td>960.0</td>
<td>0.0</td>
</tr>
<tr>
<td>COUNTER_32K</td>
<td>✔</td>
<td>FUNC_32K_CLK</td>
<td>6.032</td>
<td>6.044262</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>YKIPA04A_GICLK</td>
<td>38.4</td>
<td>0.0</td>
</tr>
<tr>
<td>RNG</td>
<td>✔</td>
<td>L4SEC_L3_GICLK</td>
<td>266.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MPU</td>
<td>⨯</td>
<td>MPU_GCLK</td>
<td>N/A</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>MA_EOCP_GICLK</td>
<td>N/A</td>
<td>6.75</td>
</tr>
<tr>
<td>ENIF1</td>
<td>✔</td>
<td>L3_EOCP_GICLK</td>
<td>N/A</td>
<td>27.6</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>EMIF_L3_GICLK</td>
<td>266.0</td>
<td>27.0</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>EMIF_PHY_GICLK</td>
<td>N/A</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>MA_EOCP_GICLK_Switch</td>
<td>N/A</td>
<td>6.75</td>
</tr>
<tr>
<td>IEEE1560_2_OCP</td>
<td>✔</td>
<td>L3INT_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td>PVMS53</td>
<td>✔</td>
<td>L4PER2_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td>PVMS52</td>
<td>✔</td>
<td>L4PER1_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td>PVMS51</td>
<td>✔</td>
<td>L4PER2_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td>ELI8</td>
<td>✔</td>
<td>L4PER_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td>DESDES5</td>
<td>✔</td>
<td>L4SEC_L3_GICLK</td>
<td>266.0</td>
<td>0.0</td>
</tr>
<tr>
<td>CTRL_MODULE_BANDGAP</td>
<td>✔</td>
<td>L3WRMSTR_LS_GICLK</td>
<td>4.8</td>
<td>1.6875</td>
</tr>
<tr>
<td>TIMER9</td>
<td>✔</td>
<td>L4PER_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>TIMER9_GFCLK</td>
<td>100.0</td>
<td>0.0</td>
</tr>
<tr>
<td>TIMER8</td>
<td>✔</td>
<td>MPU_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>TIMER8_GFCLK</td>
<td>100.0</td>
<td>27.0</td>
</tr>
<tr>
<td>TIMER7</td>
<td>✔</td>
<td>MPU_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>TIMER7_GFCLK</td>
<td>100.0</td>
<td>27.0</td>
</tr>
<tr>
<td>TIMER6</td>
<td>✔</td>
<td>MPU_L3_GICLK</td>
<td>266.0</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>TIMER6_GFCLK</td>
<td>100.0</td>
<td>27.0</td>
</tr>
</tbody>
</table>
All peripheral clock frequencies are being compared to maximum allowed frequencies documented in the Data Manual and results are being displayed in a log fashioned view. There are several types of results that can be shown:

- **The green check mark** shows that clock is running within max frequency limits
- **The red cross mark** represents an error which means clock is running faster than the maximum frequency supported by the module.
- **The yellow warning** represents a CTT internal clock that is not documented in the Data Manual. Therefore that clock cannot be verified.
- **The blue warning** represent a clock that is part/speed grade specific and user must manually check the Data Manual and determine if clock is running within frequency restrictions.

**NOTE:** Hovering the message icons will display a pop-up detailed status message.

The frequency analyzer window has two menu options:

- File
- Options.

The current results from the frequency analyzer can be saved in a .csv file using the sub-menu Save under File menu.

Rounding of the clocks is changed under Option menu. When Frequency Rounding On is selected, the clocks calculated in the CTT are rounded to the same digits after decimal point shown in the specific-device Data Manual and then the comparison is performed. When Frequency Rounding OFF is selected, the clocks calculated in the CTT are not rounded.

### 5.4.6 CTT Help

(a) About Clock Tree Tool - Shows the CTT device and model/view versioning.

(b) User Manual - This option opens the user manual document. (For Linux users, navigate to the "Docs" folder and open document with appropriate application manually.)

(c) SRAS License Agreement - This option displays the license agreement window.

### 6 CTT Blocks

This chapter highlights the different types of blocks that model the clock tree behavior, in the CTT.

**NOTE:** Any modification of the Block parameters in the Controller View affects the associated register bitfields. The Register View switches to the most recently updated register, while all the bitfield value changes are also added to the Trace View.

### 6.1 CTT Pin

The pin model presents a device pin. Figure 18 shows an example of a Pin Block.

**Figure 18. CTT Pin**

### 6.2 CTT Pin-Clock Source

The Pin-Clock Source model presents a device control bit. Figure 19 shows an example of a Pin-Clock Source Block. From functional point of view - it can be active or inactive.
6.3 **CTT Crystal**

The Crystal Block presents an Xtal. In the Control View, the user can choose its possible frequencies from a drop-down menu. Figure 20 shows an example of a CTT Crystal block.

The currently selected frequency is identified in the Trace View and also in the label next to the Crystal.

![Figure 20. CTT Crystal](image)

6.4 **CTT Clock Source**

The user defined input source clocks allows a configuration of the CTT input clocks. The input clocks are set by configuring the value (in Hz) in the text field and changing the state of the block to active as also shown in the example in Figure 21 below.
6.5 **CTT Oscillator**

Figure 22 shows an example of an oscillator block. The state of the oscillator is set in Controller View using the drop-down menu.

6.6 **CTT Clock Switch Block**

A clock switch allows the clock gating control (that is, enable/disable) within the branches of the clock tree. Essentially, three different types of switches may be defined:

1. Hardware Switch
2. Manual Switch
3. Auto Switch
6.6.1 CTT Hardware Switch

The hardware switch is controlled by hardware gating conditions:

- The derived clock is inactive.
- All modules receiving the derived clock, are inactive.
- All switches receiving the derived clock, are gated (open).

The user has no control over this switch. It is automatically closed when the hardware gating conditions are satisfied.

**NOTE:** A derived clock is the clock at the output of the switch.

6.6.2 CTT Manual Switch

The manual switch is software controlled by setting or clearing the enable bits in corresponding registers (Generally applicable to module functional clocks). The user can enable or disable the switch using the button in the controller view.

The derived clock from the switch may be connected to multiple modules and can have one or more ENABLE bits associated, to request this clock.

The switch gating condition is:

- All the associated clock ENABLE bits for this clock are cleared to 0.
6.6.3 CTT Auto Switch

The auto switch is a software/hardware controlled switch. The user can either manually (through software control) enable/disable the derived clock or set the switch to auto mode.

In the auto mode the clock is controlled by hardware gating conditions. Hence, when ever the gating conditions are satisfied the clock is automatically disabled and when any of the gating conditions is not satisfied the clock is automatically enabled by the hardware. In this case no software control of clocks is necessary.

The manual (software) control clock gating condition is:
• All the associated clock ENABLE bits for this clock are cleared to 0.

The hardware control clock gating conditions are:
• All the associated clock ENABLE bits and clock AUTO bits for this clock are set to 1.
• The derived clock is not requested by any module (that is, the module is inactive).

NOTE: Both the clock gating conditions of the auto mode must be satisfied for the derived clock to be gated automatically.

The user can use the switch in manual mode by clicking on the MANUAL check box.

In this case when the push-button on the right side is in ENABLED state, all the associated clock ENABLE bits are set and the switch is closed. Similarly if the push-button is in DISABLED state, all the clock ENABLE bits are cleared to 0 and the switch is open.

The user can set the switch to auto mode using following sequence:
1. Push the push-button to ENABLED state, to set all clock ENABLE bits to 1
2. Click on the AUTO check box to set all the clock AUTO bits to 1.

In the auto mode, the switch will automatically close when any of its gating conditions is not satisfied.

6.7 CTT Divider Block

A clock divider allows the clock frequency division. The output clock frequency is the frequency of the input clock divide by the division factor set in the divider.

The user can select the division factor by clicking on the drop-down list.

NOTE: If the divider has a fixed divide factor (that is, the software can not change the divide factor) then the drop-down list contains only one division factor.
6.8 CTT MUX Block

6.8.1 CTT Basic MUX Block

A clock mux allows selecting from multiple source clocks for the derived clock. The user can select the source clock by clicking on the check box corresponding to one of the multiple source clocks in the Trace View.

The currently selected source clock is identified in the Trace View.

6.8.2 CTT Priority MUX Block

The inputs has predefined priorities and the hardware selects the highest priority active clock. Figure 28 shows an example of a Priority MUX Block.

The currently selected source clock is identified in the Controller View.
6.9 **CTT DPLL Block**

A DPLL block receives source clocks and in turn generates the clocks for the device. Refer to the device Technical Reference Manual for details about the DPLL.

This sections present the two basic examples of the DPLL Blocks.

First example is shown on Figure 29.

The user must follow the following sequence to configure the DPLL:

1. If the DPLL is in LOCKED state, set it to one of the UNLOCKED states (for example, LOW POWER STOP state), by selecting the mode in the "Mode" drop-down list.
2. Set the M and N parameters by typing the values in the corresponding edit boxes. NOTE: After editing
the value of the parameter in the edit box, ENTER key must be pressed so that the new value is accepted by the tool.

3. Select the output divide factor M2, and so forth, by clicking on the associated drop-down list.

4. Switch the DPLL to the LOCKED mode by clicking on the "Mode" drop-down list and selecting the mode.

Once the DPLL is in LOCKED state the CLKOUT, CLKOUTX2 and the output clock frequencies (displayed after the output divide factors) will be updated.

DPLLs can also have options from the controller to select bypass clocks, 4xen mode, CLOCKOUTIF, and sd-div modes. For more information about these functionality please refer to, dependent on the device of use, the PRCM chapter or Clocking section in the device Technical Reference Manual.

Figure 30 shows the second example of a DPLL Block.

![Figure 30. CTT DPLL Block](image)

The user must follow the following sequence to configure the DPLL:

1. Select the mode in the "Mode" drop-down list, for example, PLL mode.

2. Set the multipliers and dividers parameters (for example, PLLM_LSB, PLLD) by typing the values in the corresponding edit boxes. NOTE: After editing the value of the parameter in the edit box, ENTER key must be pressed so that the new value is accepted by the tool.

3. If available, select the output divide factor PLLDIV1, and so forth, by clicking on the associated drop-down list.

Once the DPLL state is set, the state of CLKOUT, and the output clock frequencies (displayed after the output divide factors) will be updated.

Third example is shown on Figure 31.
The user must follow the following sequence to configure the DPLL:

1. Select the mode in the "Mode" drop-down list, for example, PLL mode.

2. Set the M, N and Mfrac parameters by typing the values in the corresponding edit boxes. NOTE: After editing the value of the parameter in the edit box, ENTER key must be pressed so that the new value is accepted by the tool.

3. Enable the clock by changing the state to ON of the <clock_name>_EN field.

4. Select the output divide factor M2, and so forth, by clicking on the associated drop-down list.
Once the DPLL is in LOCKED state the CLKOUT, CLKOUTX2 and the output clock frequencies (displayed after the output divide factors) will be updated.

6.10 CTT Module Block

A module block represents the destination modules, such as I2C, MCSI, McBSP, and so forth. A module receives functional and interface clocks. It may be active or inactive. It can also be in enabled, auto, or disabled module mode. Module can also have optional functional clocks associated to it.

If a module has only Active / Idle functionality, the user can switch a module to ACTIVE or IDLE state and only the MODULE STATE drop down menu will be displayed inside the controller.

If a module has MODULE MODE and MODULE STATE functionality, the user must select the mode of the module, then the module state.

If a module has Optional Functional clock functionality, the user may enable opt clocks as well.

NOTE: In the device, there are various combinations of module functionality. A given module can have one or more at the same time.

Moreover, In basics, the clocks associated to module function as follows:

1. Optional functional clock is running whenever the OptFclken bit is set to 1, and it is not concerned by the module state (idle/active).

2. Module mode associated clocks are automatically gated if ModuleMode is set to “Disabled” and this is the module reaches idle state.

3. When ModuleMode is set to “Enabled” functional clock is automatically un-gated. The interface clock is automatically gated/un-gated based on the module idle/active transition.

4. Module mode associated clocks are automatically gated/un-gated when ModuleMode is set to “Auto” based on the idle/active transition of the module. “Auto” option is available only for modules with interface idle protocol associated clock(s).

For more information about module mode, module state, and optional clocks associated to modules, please refer to, dependent on the device of use, the PRCM chapter or Clocking section in the device Technical Reference Manual.

Figure 32. CTT Module Block
6.11 **CTT Delimits**

CTT delimit blocks are highlighted areas in the GUI. They do not have a defined associated controller. Their purpose is to highlight the boundaries of a given PRCM clock domain. This way the GUI provides better visual interpretation of modules and their clock domain affiliation. Figure 33 shows an example of a delimit.

**Figure 33. CTT Delimit Block**

7 **CTT Release Notes**

Specific CTT package changes are listed in TI CTT Release Notes.

8 **CTT Limitations**

For the known limitations of the CTT go to Known Issues.
# Revision History

## Changes from F Revision (August 2018) to G Revision

<table>
<thead>
<tr>
<th>Changes</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updated CTT formatting to match the user’s guide type</td>
<td>1</td>
</tr>
<tr>
<td>Updated CTT Save / Load Registers</td>
<td>12</td>
</tr>
<tr>
<td>Added CTT Release Notes section</td>
<td>26</td>
</tr>
<tr>
<td>Added CTT Limitation section</td>
<td>26</td>
</tr>
</tbody>
</table>
IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), 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, 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 (www.ti.com/legal/termsofsale.html) 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.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2018, Texas Instruments Incorporated