Clock Tree Tool

Texas Instruments Family of Products

User's Guide

Literature Number: SPRUIA4
March 2017

Contents

List of Figures

Read This First

Community Resources

The following link connects to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

TI Embedded Processors Wiki —Texas Instruments Embedded Processors Wiki

Established to assist developers using the many Embedded Processors from Texas Instruments to get started, help each other innovate, and foster the growth of general knowledge about the hardware and software surrounding these devices.

About This Manual

FCC Warning

This equipment is intended for use in a laboratory test environment only. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to subpart J of part 15 of FCC rules, which are designed to provide reasonable protection against radio frequency interference. Operation of this equipment in other environments may cause interference with radio communications, in which case the user at his own expense will be required to take whatever measures may be required to correct this interference.

Information About Cautions and Warnings

This book may contain cautions and warnings.

CAUTION

This is an example of a caution statement.

A caution statement describes a situation that could potentially damage your software or equipment.

WARNING

This is an example of a warning statement.

A warning statement describes a situation that could potentially cause harm to you.

The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

Register, Field, and Bit Calls

The naming convention applied for a call consists of:

- For a register call: *<Module name>.<Register name>*; for example: UART.UASR
- For a bit field call:
  - *<Module name>.<Register name>[End:Start] <Field name> field*; for example, UART.UASR[4:0] SPEED bit field
  - *<Field name> field <Module name>.<Register name>[End:Start]*; for example, SPEED bit field UART.UASR[4:0]
- For a bit call:
  - *<Module name>.<Register name>[pos] <Bit name> bit*; for example, UART.UASR[5] BIT_BY_CHAR bit
  - *<Bit name> bit <Module name>.<Register name>[pos]*; for example, BIT_BY_CHAR bit UART.UASR[5]

To help the reader navigate the document, each register call is hyperlinked to its register description in the register manual section. After each register description, a table summarizes all hyperlinked register calls.

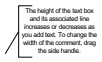
Coding Rules

The programming models or code listings follow the rules:

Type	Definition	Example
File	Starts with the module name	PRCM_test1.c MCBSP1_init.h
Variable	Global variables are prefixed by "g_" Pointers are prefixed by "p" Global pointers are prefixed by "g_p"	g_SDMA_LogicalChan pAddrCounter g_pSDMA_LogicalChan
Function	Starts with the module name	PRCM_SetupClocks() ArmIntC_MaskInterrupts()
Typedef	Ends with "_t"	PRCM_Struct_t
Definition	Starts with the module name and is followed by the register name	#define SMS_ERR_TYPE *((volatileUint32*)0x680080F4) #define MCBSP2_RCR1_REG *((volatile Uint32*)0x4807401C)
Enumeration	Starts with the module name	Typedef enum DMA_Mode_Label { INPUT_MODE OUTPUT_MODE } DMA_Mode_t;

Flow Chart Rules

Flow charts follow the following rules:

Shape	Name	Definition
	Process	Any computational steps or processing function of a program; defined operation(s) causing change in value, form, or location of information
	Decision	A decision or switching-type operation that determines which of a number of alternate paths is followed
	Predefined process or sub-process	One or more named operations or program steps specified in a subroutine or another set of flow charts
	Data or I/O	General I/O function; information available for processing (input) or recording of processed information (output)
	Terminator	Terminal point in a flow chart: start, stop, halt, delay, or interrupt; may show exit from a closed subroutine
	Annotation	Additional descriptive clarification, comment
	On page connector (reference)	Exit to, or entry from, another part of chart in the same page
	Off page connector (reference)	The flow continues on a different page.
	Summing Junction	Logical AND
	Or	Logical OR
	Parallel mode (ISO)	Beginning or end of two or more simultaneous operations
	Flow Line	Lines indicate the sequence of steps and the direction of flow.

Trademarks

JAVA is a trademark of Oracle.

Windows is a registered trademark of Microsoft.

All other trademarks are the property of their respective owners.

CTT Installation

CAUTION

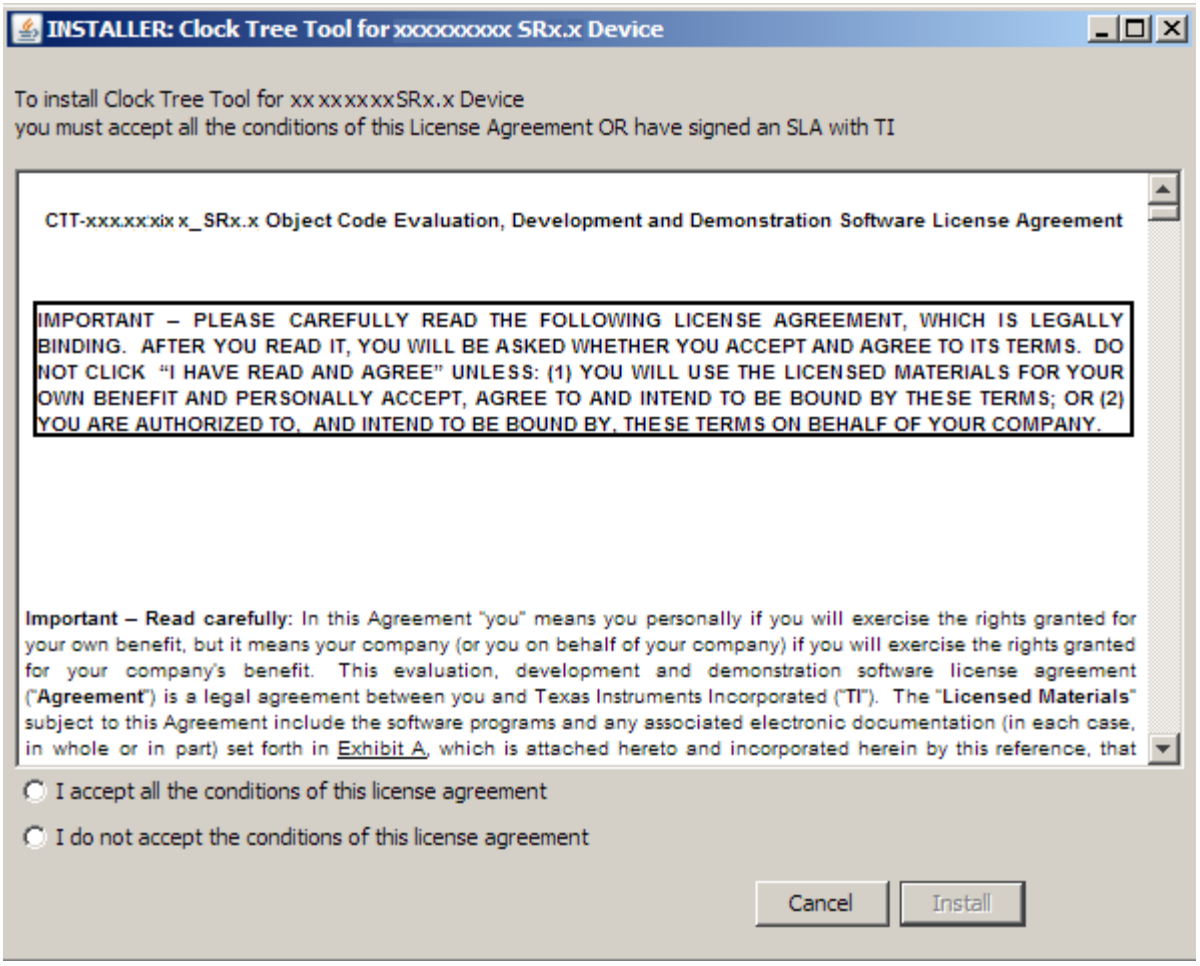
Java™ Runtime Environment, Standard Edition (v 1.7 or higher) must be installed before Clock Tree Tool is installed.

The Clock Tree Tool (CTT) installation procedure is composed of following steps.

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 1-1. The user must accept the conditions of the license in order to install the CTT.

Figure 1-1. CTT Installation License Agreement Window



1.2 CTT Installation Step 2

When the conditions of the license agreement are accepted, the "Install" button is enabled, see Figure 1-2. Click on the "Install" button to proceed to the "Destination Directory Selection" window, see Figure 1-3. 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 1-2. CTT Installation License Agreement Window 2

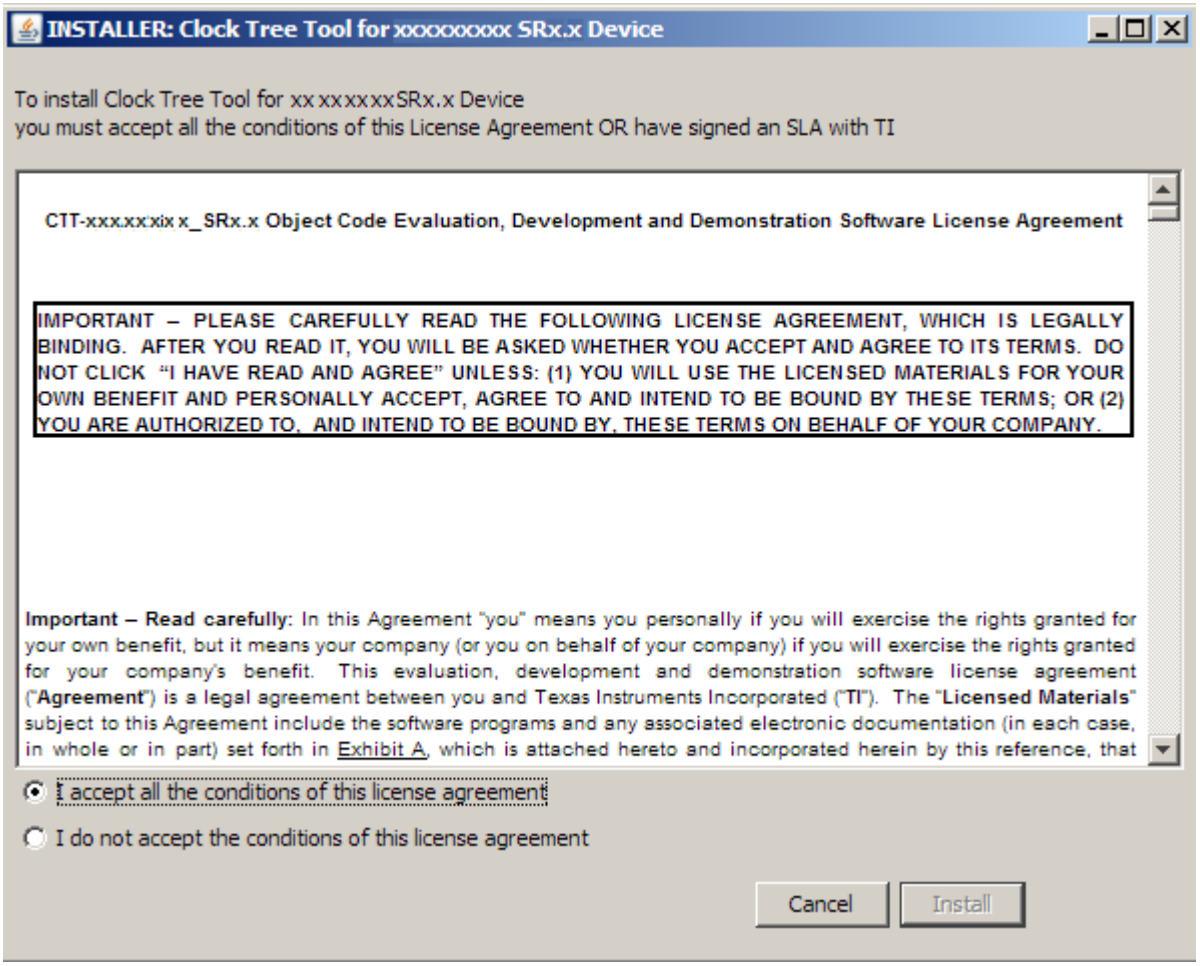
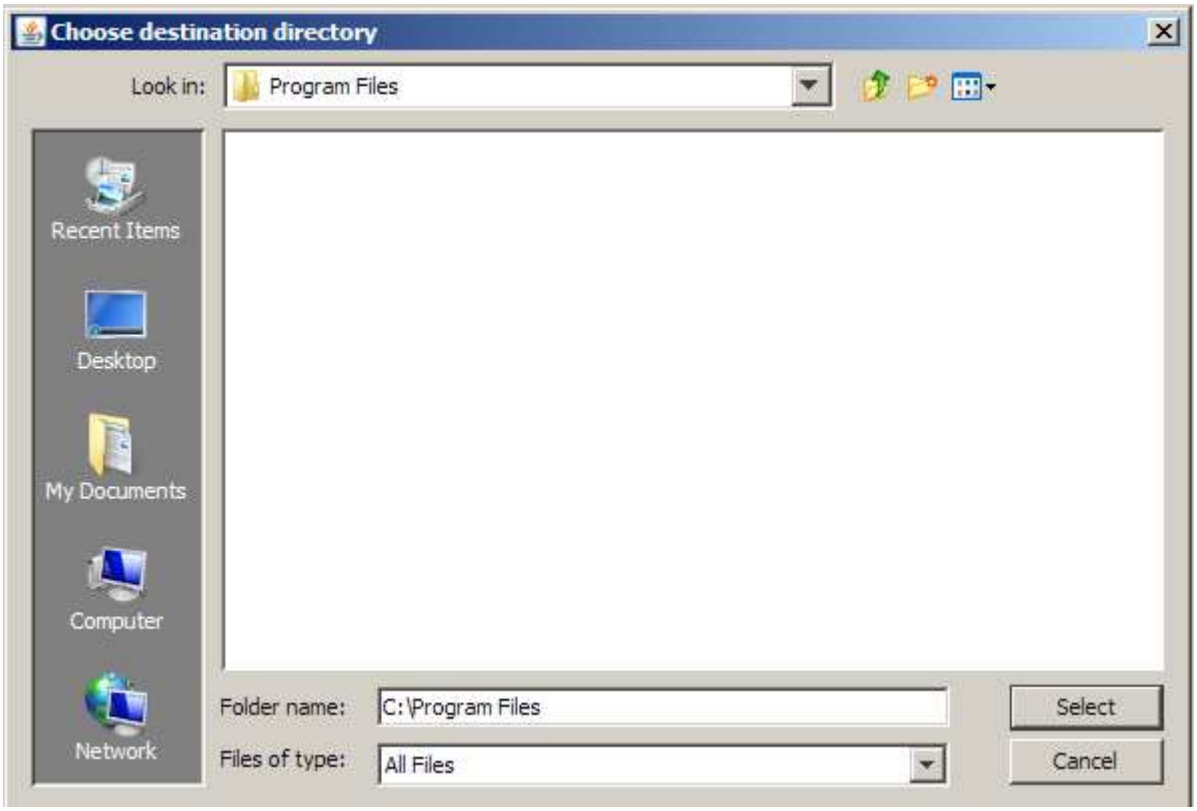


Figure 1-3. CTT Destination Directory Selection Window

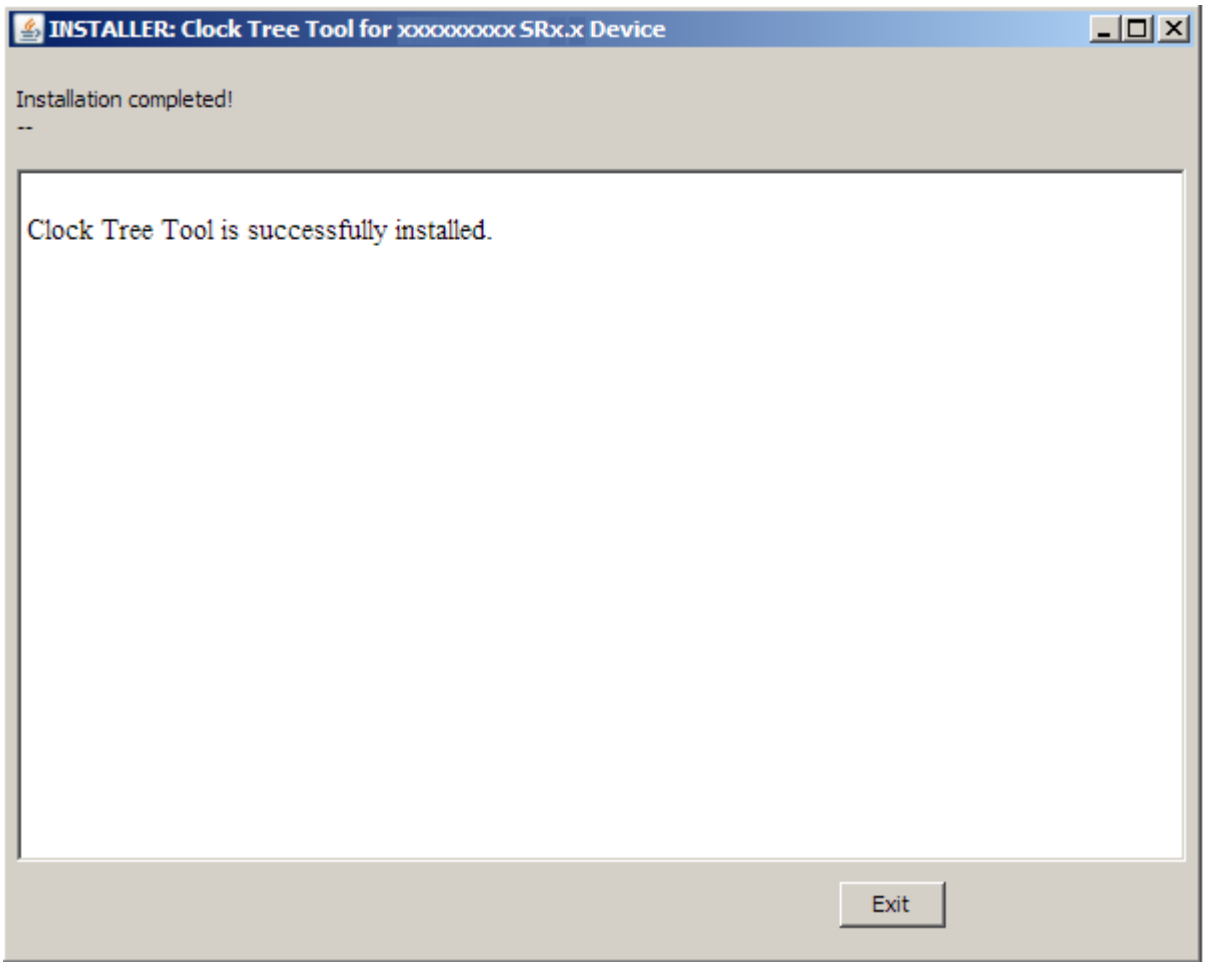


1.3 CTT Installation Step 3

When the installation is complete the "Installation completed" message is displayed Figure 1-4. Click on "OK" button to proceed to the last window confirmation window, see Figure 1-5. There, click "Exit" to complete the CTT installation.

Figure 1-4. CTT Installation Complete Message Window

Figure 1-5. CTT Installation Exit Window



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 PRCM registers
- interact with the PRCM 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.

CTT System Requirements

- Requires Java™ JRE 1.7 or higher (Can be downloaded from https://www.java.com/en/download/).
- Has been tested for Microsoft Windows 7 ®.
- The ideal screen resolution is 1920x1080.

CTT Running Requirements

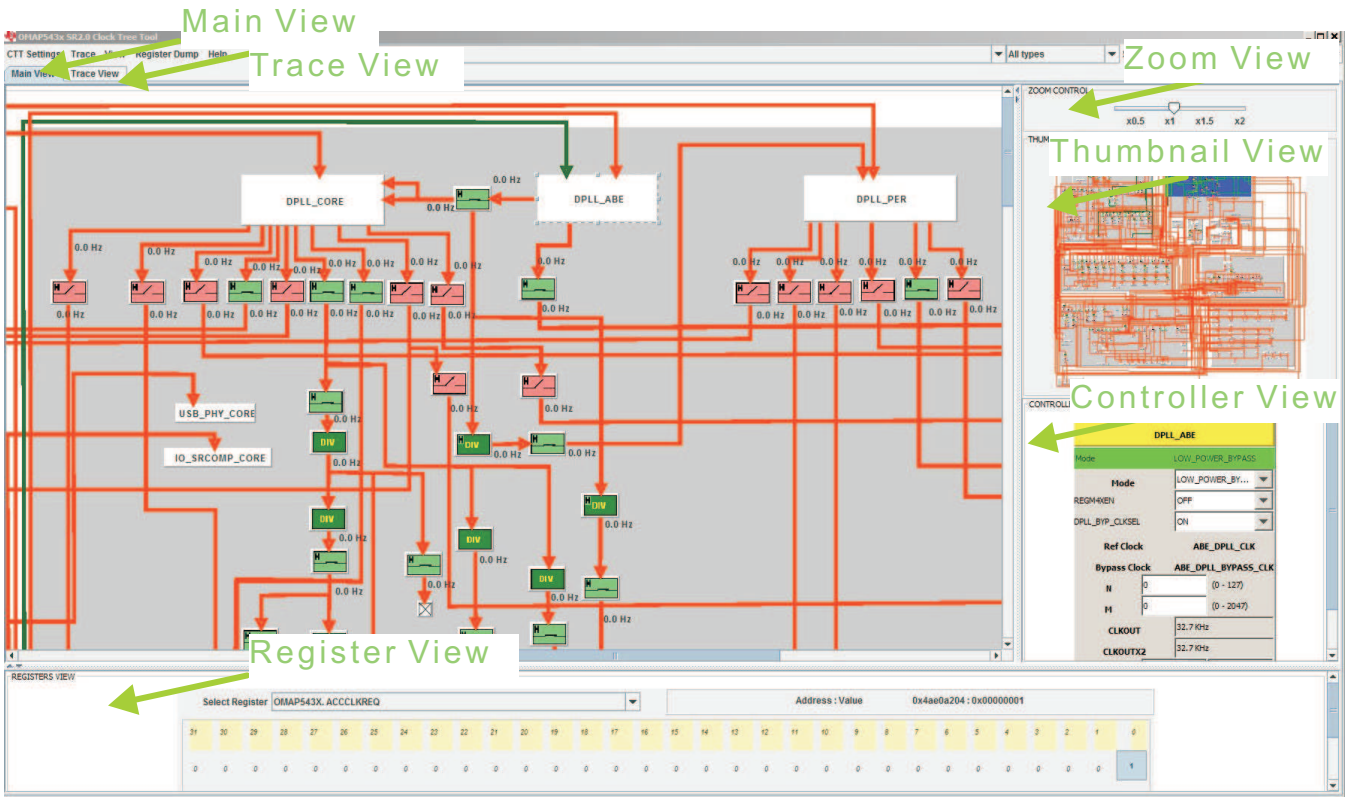
The start-up sequence of the CTT consists of reading an entire clock tree description database files. This would normally take about 10 to 20 seconds.

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.

CTT GUI (Graphical User Interface) Description

5.1 CTT Views Description

Figure 5-1. CTT Views



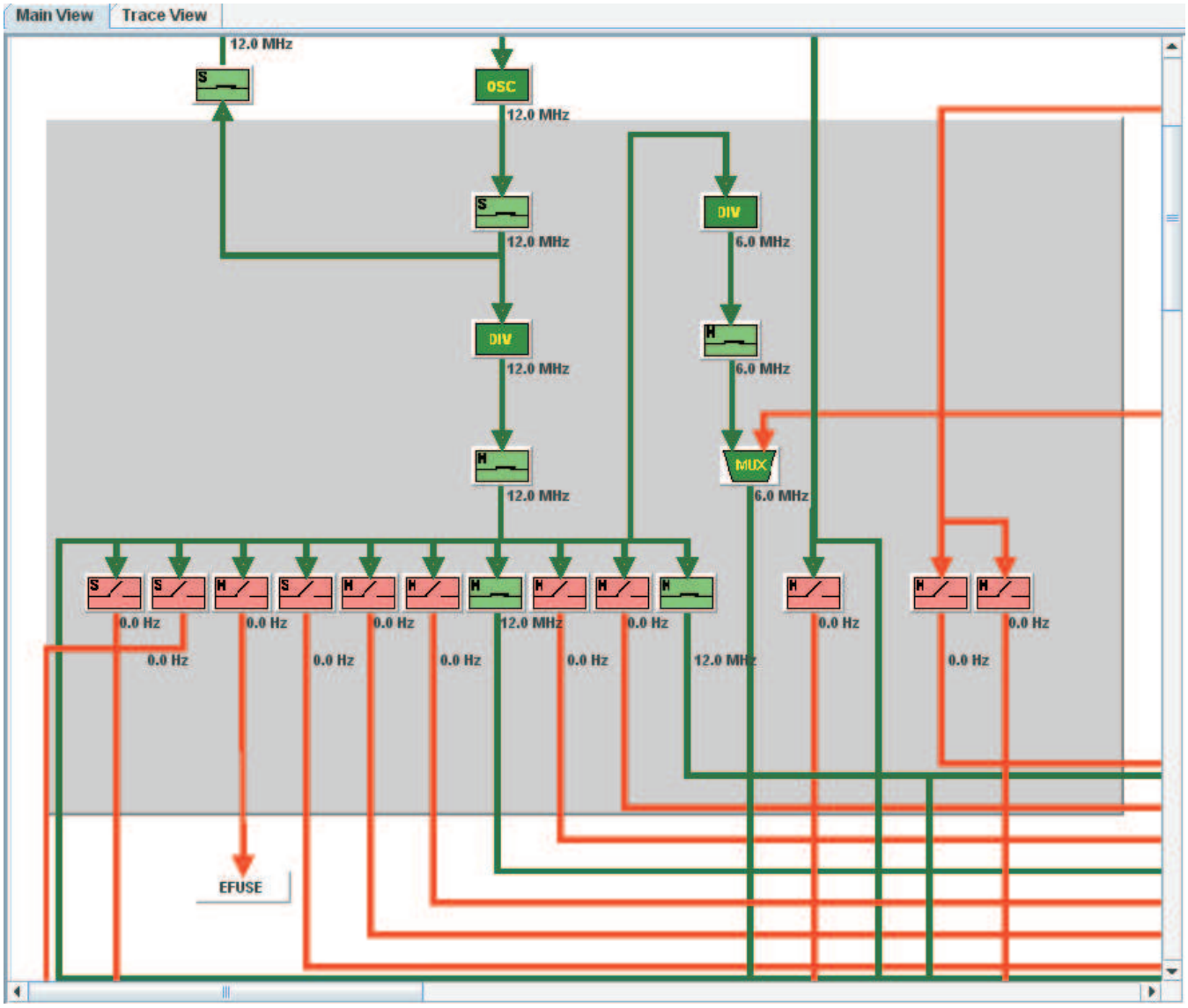
The CTT GUI is composed of 5 sub-views (see Figure 5-1):

- Main View
- Thumbnail View
- Controller View
- Registers View
- Trace View

5.1.1 CTT Main View

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

Figure 5-2. CTT Main View



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

- Delimit

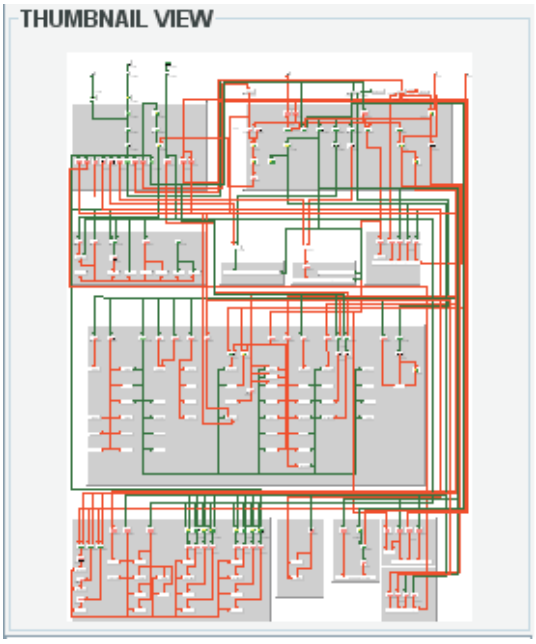
(Refer to the device Technical Reference Manual for the description of these blocks)

The user can use the slide bars on the right side and the bottom of the view to move up/down and left/right in the Main View, respectively. 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.

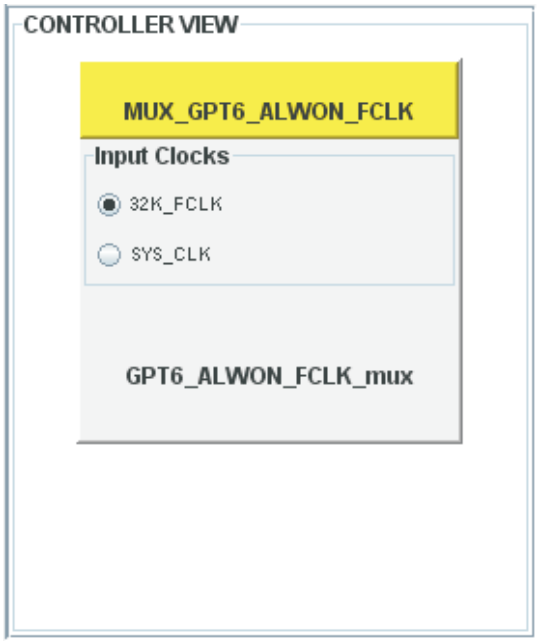
Figure 5-3. CTT Thumbnail View



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.

Figure 5-4. CTT Controller View

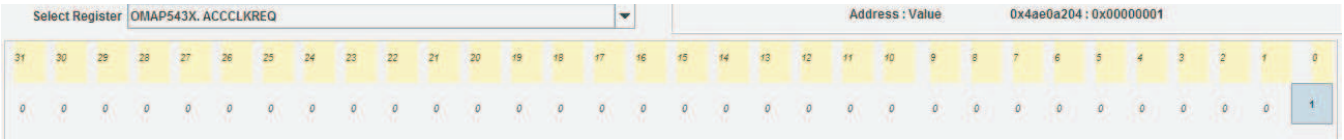


5.1.4 CTT Registers View

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.
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.
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 of PRCM). Each bit is identified by the bit number (0 for the LSB). Below the bit number is the current value of the bit (1/0).
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.

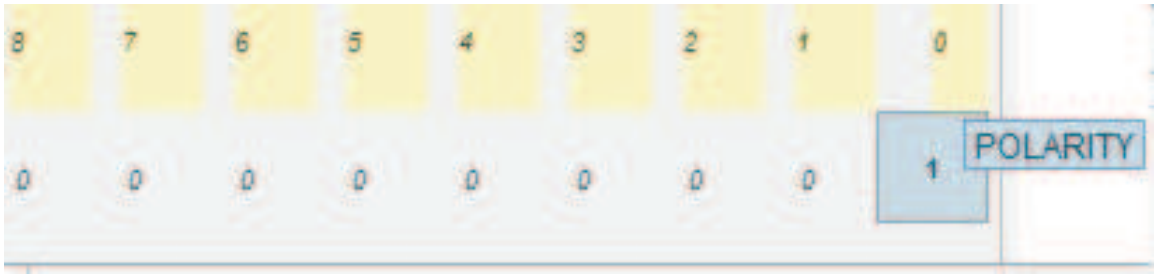
NOTE: When the user changes a parameter of a block which affects bitfields of more than one register, the Registers View only shows the last register updated. The Trace View shows the complete list of registers affected by this change.

Figure 5-5. CTT Register View



When user positions the pointer on the number of a register bit a pop-up displays the description of the corresponding bitfield, see Figure 5-6.

Figure 5-6. CTT Register View

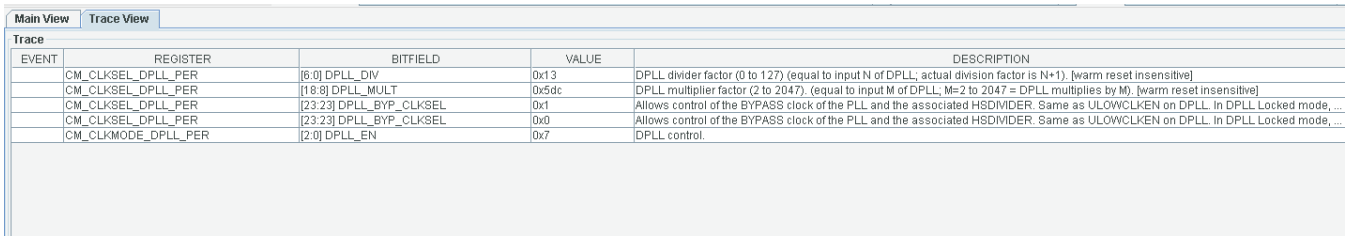


5.1.5 CTT Trace View

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.

Figure 5-7. CTT Trace View

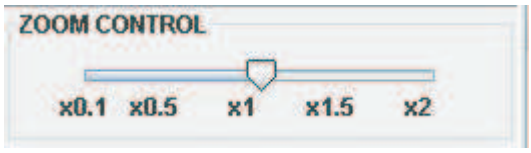


5.2 CTT Zoom Control

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.

Figure 5-8. CTT Zoom Control

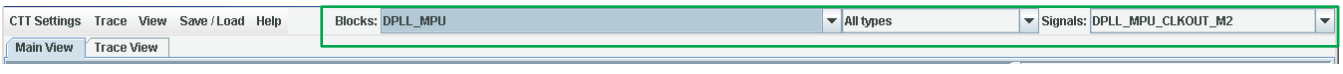


5.3 CTT Search Bars

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 5-9. CTT Search Bars



5.4 CTT Menu Commands Description

The CTT menu has following commands:

Figure 5-10. CTT Menu

1. CTT Settings
   (a) Power-on Reset
2. Trace
   (a) Reset
3. View
   (a) Hide Others
   (b) Display All
   (c) Hide Frequency
   (d) Display Frequency
   (e) Refresh View
   (f) Print View
4. Save / Load
   (a) Save Registers
   (b) Save Registers (CTT only)
   (c) Load Registers
   (d) Save Source Clocks
   (e) Load Source Clocks
5. Help
   (a) About Clock Tree Tool
   (b) User Manual
   (c) Licence Agreement
   (d) Export Control Notice

5.4.1 CTT Settings

a. Power-on Reset - Triggers a power-on reset for all the registers of the PRCM. 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 registers of the PRCM (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 (PRCM, 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.

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.

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>

or similar to the following example

```
DeviceName DRA75x_DRA74x_SR1.1
 0x4a005000 0x00000000
0x4a005040 0x00030001
...
0x4ae07f30 0x00000000
```

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 PRCM 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 and use this file to load configuration into the connected HW via the environment used.

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 souce clocks must be saved to be loaded later. See Section 6.3 for details on clock source configuration from GUI.

5.4.5 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. Licence Agreement - This option displays the licence agreement window.

d. Export Control Notice - This option displays the export control notice window.

CTT Blocks

This section 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 6-1 shows an example of a Pin Block.

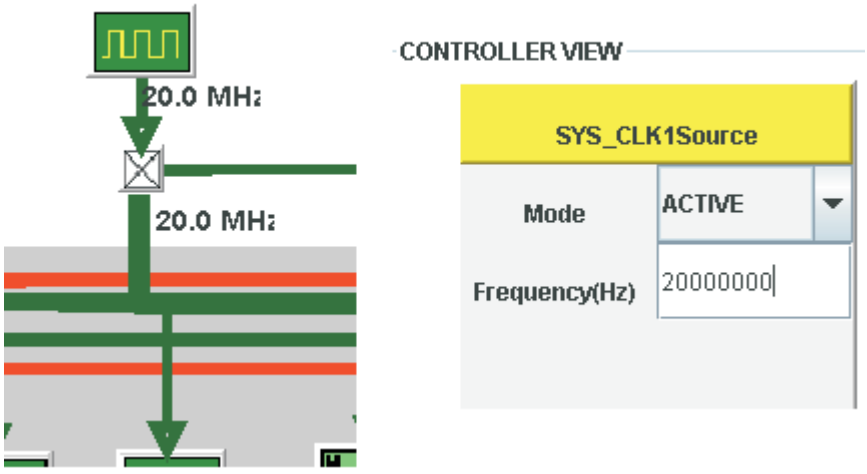
Figure 6-1. CTT Pin

6.2 CTT Crystal

The Crystal Block presents an Xtal. In the Control View, the user can choose its possible frequencies from a drop-down menue. Figure 6-2 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.

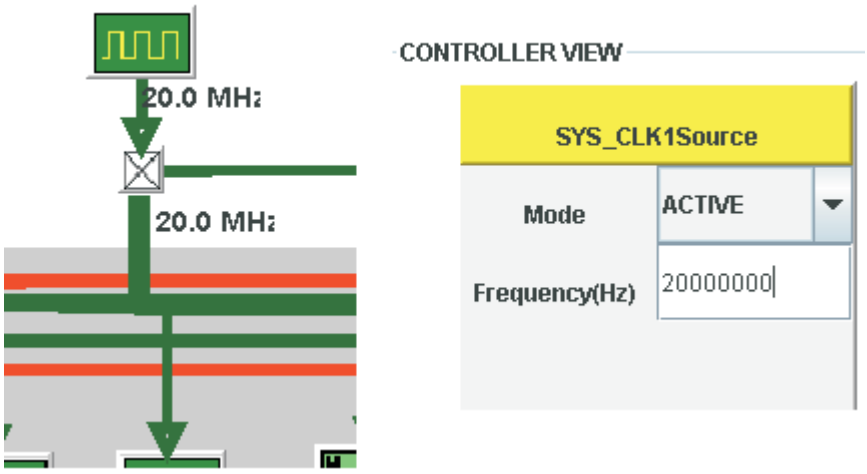
Figure 6-2. CTT Crystal



6.3 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 6-3 below.

Figure 6-3. CTT Clock Source

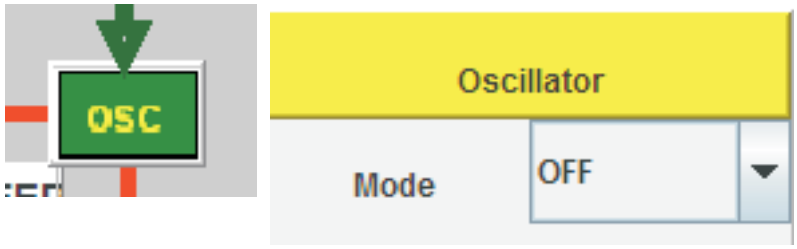


See also Section 5.4.4.2 for saving / loading source clock configuration.

6.4 CTT Oscillator

Figure 6-4 shows an example of an oscillator block. The state of the oscillator is set in Controller View using the drop-down menue.

Figure 6-4. CTT Oscillator



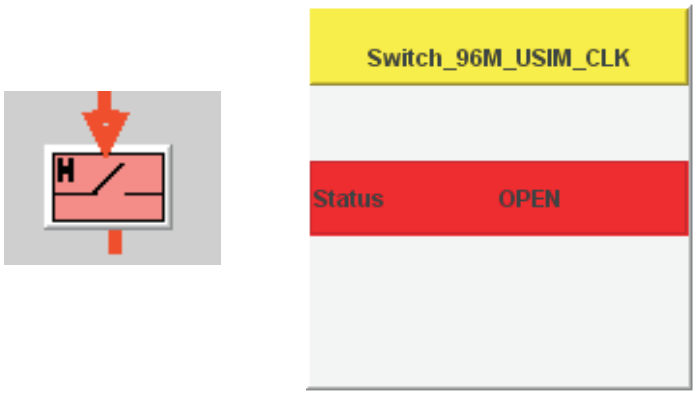
6.5 CTT Clock Switch Block

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

6.5.1 CTT Hardware Switch

Figure 6-5. CTT Hardware Switch



The hardware switch is controlled by hardware gating conditions:

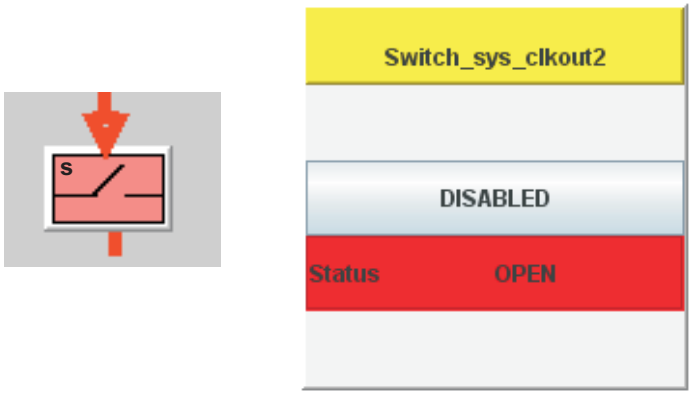
- The derived clock is inactive.
- All modules receiving the derived clock, are inactive.
- All switches receiving the derived clock, are gated (open).

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.

6.5.2 CTT Manual Switch

Figure 6-6. CTT Manual Switch

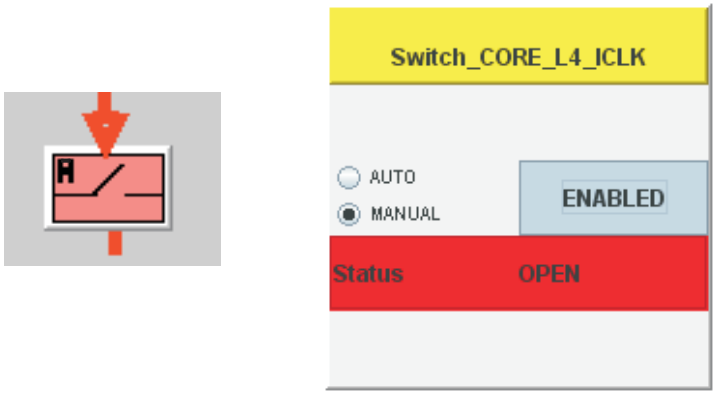


The manual switch is software controlled by setting or clearing the enable bits in corresponding registers (Generally applicable to module functional clocks). The user can enable or disable the switch using the button in the controller view.
The derived clock from the switch may be connected to multiple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

6.5.3 CTT Auto Switch

Figure 6-7. 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.

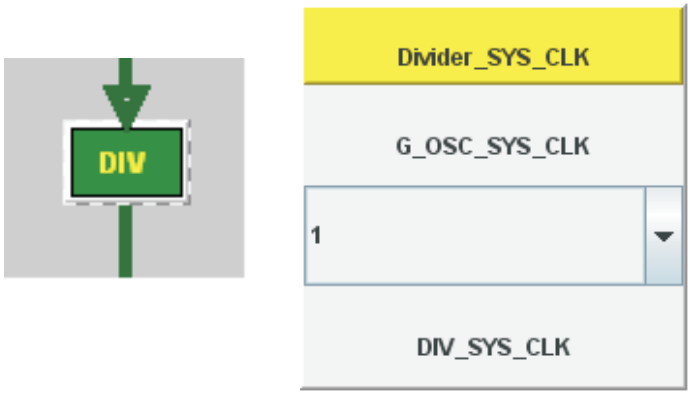
6.6 CTT Divider Block

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.

Figure 6-8. CTT Divider Block



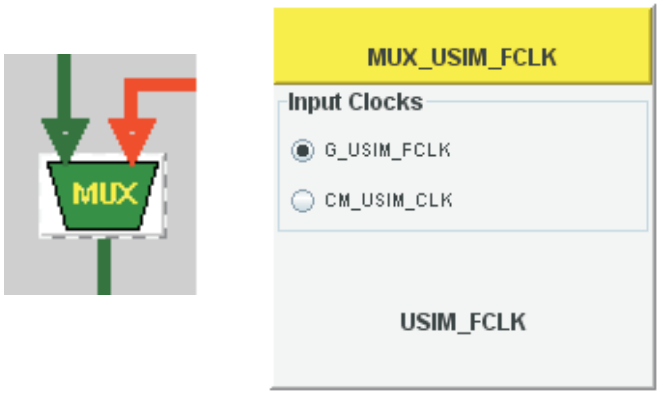
6.7 CTT MUX Block

6.7.1 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.

Figure 6-9. CTT MUX Block

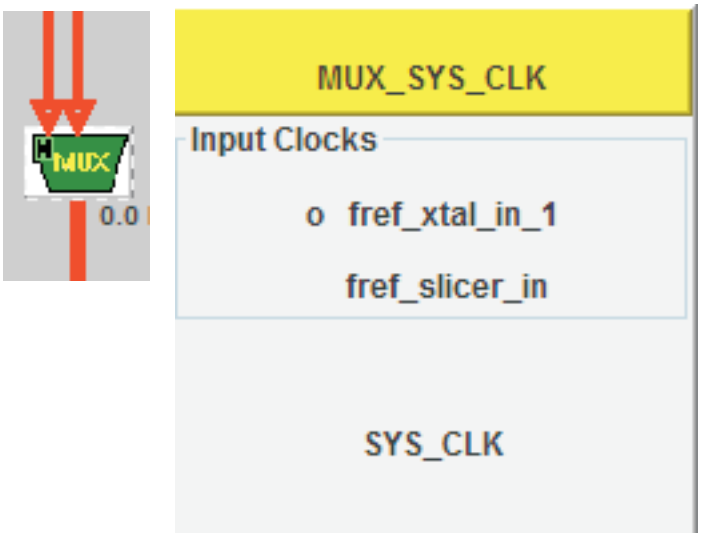


6.7.2 CTT Priority MUX Block

The inputs has predefined priorities and the hardware selects the highest priority active clock. Figure 6-10 shows an example of a Priority MUX Block.

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

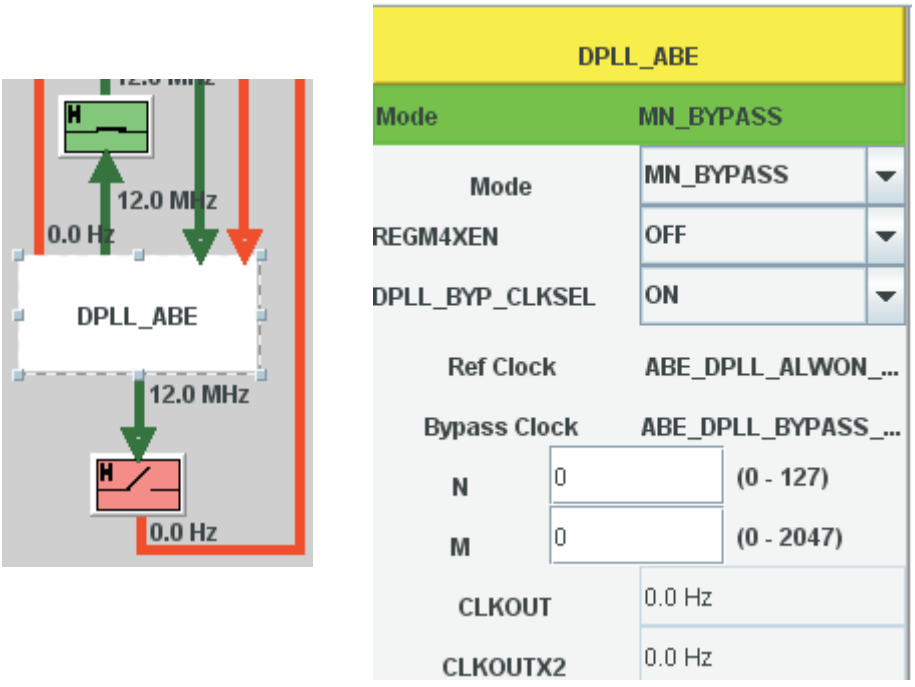
Figure 6-10. CTT Priority MUX Block



6.8 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.

Figure 6-11. CTT DPLL Block



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 the PRCM chapter in the device Technical Reference Manual.

6.9 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.

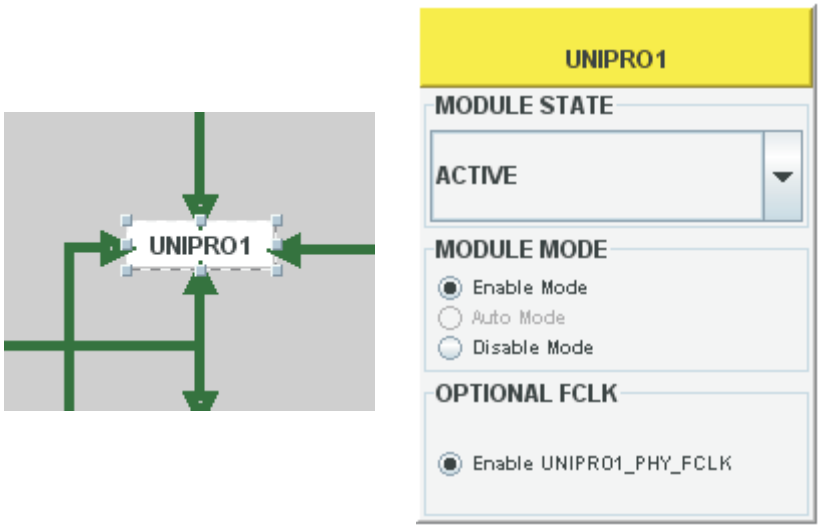
NOTE: In the device, there are various combinations of module functionality. A given module can have one or more at the same time.

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 PRCM chapter in the device Technical Reference Manual.

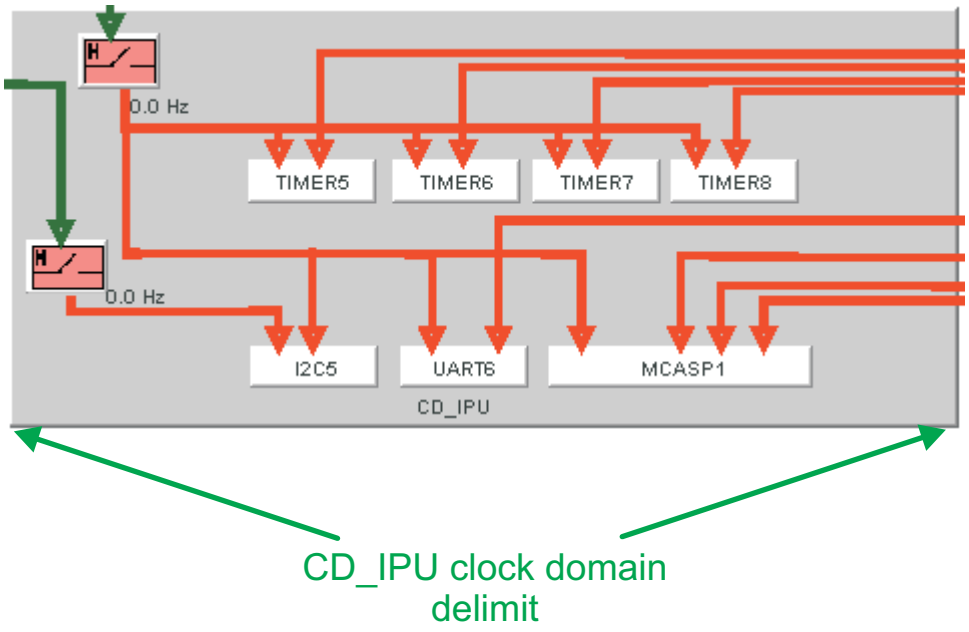
Figure 6-12. CTT Module Block



6.10 CTT Delimits

CTT delimit blocks are highlighted areas in the GUI. They do not have a defined associated controller. Their purpose is the highlight the boundaries of a give PRCM clock domain. This way the GUI provide better visual interpretation of modules and their clock domain affiliation. Figure 6-13 shows an exmple of a delimit.

Figure 6-13. CTT Delimit Block



IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

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