Contents

1 Revision History ................................................................. 1
  1.1 Revision 3 ........................................................................... 1
  1.2 Revision 2.0 ................................................................. 1
  1.3 Revision 1 ................................................................. 1

2 RTG4 CCC Dynamic Configuration ............................................ 2
  2.1 Design Requirements .............................................................. 3
  2.2 Design Description ............................................................... 3
  2.3 Hardware Implementation ...................................................... 7
    2.3.1 CoreABC Program .......................................................... 10
    2.3.2 Configuring CCC for Dynamic Configuration ......................... 11
  2.4 Simulation Results ................................................................ 13
  2.5 Setting Up the Design ............................................................ 14
  2.6 Running the Design ............................................................. 14
  2.7 Conclusion ........................................................................ 18

3 Appendix: Design Files .............................................................. 19
**Figures**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RTG4 Fabric CCC Block Diagram</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Single CCC Use Model Design</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Dual CCC Use Model Design</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>CCCDYN and CCCAPB Macro Connection for Single CCC Dynamic Configuration</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>CCCDYN and CCCAPB Macro Connection for Dual CCC Dynamic Configuration</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>SmartDesign Top-level Single CCC Use Model</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>SmartDesign Top-level Dual CCC Use Model</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>CCCDYN and RTG4CCCAPB_IF Connection</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>CoreABC Program for Single CCC Use Model</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>CoreABC Program for Dual CCC Use Model</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>PLL Output Clock Settings</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>Single CCC Dynamic Configuration Setting</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>Single CCC Dynamic Configuration SmartDesign</td>
<td>12</td>
</tr>
<tr>
<td>14</td>
<td>Dual CCC dynamic configuration setting</td>
<td>13</td>
</tr>
<tr>
<td>15</td>
<td>Dual CCC Dynamic Configuration SmartDesign</td>
<td>13</td>
</tr>
<tr>
<td>16</td>
<td>Single CCC Dynamic Configuration Simulation Window</td>
<td>13</td>
</tr>
<tr>
<td>17</td>
<td>Dual CCC Dynamic Configuration Simulation Window</td>
<td>14</td>
</tr>
<tr>
<td>18</td>
<td>CCC Output Clock Frequency Before Dynamic Configuration</td>
<td>16</td>
</tr>
<tr>
<td>19</td>
<td>CCC Output Clock Frequency After Dynamic Configuration</td>
<td>16</td>
</tr>
<tr>
<td>20</td>
<td>GL0 Dynamic Frequency Change</td>
<td>17</td>
</tr>
<tr>
<td>21</td>
<td>GL1 Dynamic Frequency Change</td>
<td>17</td>
</tr>
</tbody>
</table>
Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Design Requirements</td>
<td>3</td>
</tr>
<tr>
<td>Table 2</td>
<td>RTG4 Development Kit Jumper Setting</td>
<td>14</td>
</tr>
<tr>
<td>Table 3</td>
<td>Probe Points</td>
<td>15</td>
</tr>
</tbody>
</table>
Revision History

1.1 Revision 3
Update the document for Libero v11.9 SP1 software release.

1.2 Revision 2.0
Updated the document for Libero v11.8 SP2 software release.

1.3 Revision 1
The first publication of this document.
This document describes how to perform dynamic configuration for both single and dual RTG4™ field programmable gate array (FPGA) clock conditioning circuit (CCC) by changing output clock frequency in a glitch-free way using general purpose divider (GPD).

RTG4 devices have 8 CCC blocks (2 in all 4 corners). Each CCC provides a complete clocking scheme for any logic implemented in the FPGA fabric and base clock for on-chip hard IP blocks like FDDR and SERDESIF. It is possible to configure the CCC parameters dynamically (via an advanced peripheral bus (APB3) interface) without reprogramming the device.

The following figure shows the block diagram of RTG4 Fabric CCC. The dotted line shows the path used when dynamically configuring GPD dividers. Each fabric CCC has four GPDs and CCC outputs can get their source from any of the GPD outputs. For instance, GPD2 divider can be used for GL0 output.

GPD is used for dividing source clock selected by general purpose multiplexer (GPMUX) by a factor of 1 to 255 to obtain necessary clock frequency on the output. In this reference design, GPD divider value is configured dynamically to different values at run time to obtain different output frequencies. Each GPD has its own GPMUX, which helps in selecting clock source to the GPD circuit from fabric CCC clock sources (external dedicated I/O, RC oscillator, fabric clock, and PLL outputs).

Any of the configuration registers can be accessed dynamically using APB3 interface. This document does not discuss about all the Fabric CCC registers that can be dynamically configured. It only shows how to dynamically change output clock frequency using GPD divider.

For more information on dynamic configuration registers, GPD and GPMUX. See UG0586: RTG4 FPGA Clocking Resources User Guide.

Figure 1  •  RTG4 Fabric CCC Block Diagram
2.1 Design Requirements

The following table lists the hardware and software requirements for this design.

Table 1 • Design Requirements

<table>
<thead>
<tr>
<th>Design Requirements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware</strong></td>
<td></td>
</tr>
<tr>
<td>RTG4 FPGA Development Kit:</td>
<td>Rev C</td>
</tr>
<tr>
<td>– USB 2.0 cable</td>
<td></td>
</tr>
<tr>
<td>– FlashPro4 programmer</td>
<td></td>
</tr>
<tr>
<td>– 12 V, 5A AC power adapter and cords</td>
<td></td>
</tr>
<tr>
<td>Host PC or Laptop</td>
<td>Any 64-bit Windows Operating System</td>
</tr>
<tr>
<td><strong>Software</strong></td>
<td></td>
</tr>
<tr>
<td>Libero® System-on-Chip (SoC)</td>
<td>v11.9 SP1</td>
</tr>
<tr>
<td>Host PC Drivers</td>
<td>–</td>
</tr>
<tr>
<td>(included with the design files)</td>
<td></td>
</tr>
</tbody>
</table>

2.2 Design Description

The reference design in this application note contains two use models to demonstrate dynamic configuration for single and dual CCC in the same corner of RTG4 device. CoreABC provides APB3 Master interface. CoreABC is a programmable soft processor used to read and write from CCC configuration registers. CoreABC program provided in the design writes in to appropriate registers to divide source clock using GPD.

Two separate designs are provided for single and dual CCC dynamic configuration use models. RTG4FCCC_0 is used for generating APB interface clock in both use models. RTG4FCCC_1 is used for single CCC dynamic configuration use model and dual_cccdyn_top_0, which contains two fabric CCCs from the same corner is used for dual CCC dynamic configuration use model. In both cases, dynamic frequency reduction in CCC output clock using GPD divider is demonstrated by probing CCC outputs in real time using oscilloscope. The upper nibble of a sequential counter in the fabric drives four on board LED's. Once the clock frequency is dynamically configured using GPD circuit, the counter slows down which in turn reduces LED blinking rate.
The following figure shows top-level block diagram of the single CCC use model.

**Figure 2 • Single CCC Use Model Design**
The following figure shows top-level block diagram of the dual CCC use model.

**Figure 3 • Dual CCC Use Model Design**
The following figure shows connectivity between CCCDYN and CCCAPB macros along with extra soft logic, which is added automatically by the configurator for single CCC dynamic configuration. For more information on how to perform dynamic configuration for single CCC, see Hardware Implementation, page 7.

**Figure 4 • CCCDYN and CCCAPB Macro Connection for Single CCC Dynamic Configuration**

The new macro CCCDYN (similar to RTG4FCCC macro with dynamic configuration enabled) is used only for dynamic CCC solution. It has an additional port APB_S_PSEL_OUT, which is a hardwired connection to CCCAPB macro. The APB macro instance is named as CCCAPB, which has extra ports CCC_0/1_APB_S_PSEL and CCC_0/1_APB_S_PRDATA to support multiple dynamic CCCs from the same corner.

The 0/1 indices refer to two CCCs in the same corner. Only two CCCDYN macros placed in the same corner can be legally connected to the APB macro instance (CCCAPB).

In the preceding figure, CCC_1_APB_S_PSEL and CCC_1_APB_S_PRDATA are connected to GND since we are using only one CCC for single CCC dynamic configuration.

The following figure shows connectivity between CCCDYN and CCCAPB macros along with extra soft logic, which is added automatically by the configurator for dual CCC dynamic configuration. The RTG4CCCAPB_IF block delimits the configurator boundaries. Connect manually two CCCDYN macros (CCC_0/CCC_1) with RTG4CCCAPB_IF module for dual CCC dynamic solution.

Address bit [8] of APB_S_PADDR is used for decoding, which of the CCC in the same corner is used by CCCAPB macro. If APB_S_PADDR[8] is equal to 1, CCC_1 is selected, if APB_S_PADDR[8] is not equal to 1, CCC_0 is selected as shown in the following figure.
2.3 Hardware Implementation

Hardware implementation involves configuring CCC for dynamic configuration along with CoreABC program to dynamically read and write in to CCC configuration registers. The reference design of both single and dual CCC use model consists of CoreABC, on-chip 50 MHz RC oscillator, Sysreset macro, AND gate, FCCC, and counter module.

In the single CCC use model, RTG4FCCC_0 generates APB interface clock at 50 MHz frequency and RTG4FCCC_1 generates global output GL0 (PLL OUT1), which is divided dynamically using GPD divider.
The following figure shows the SmartDesign top-level block diagram for single CCC dynamic configuration.

**Figure 6 • SmartDesign Top-level Single CCC Use Model**

In the dual CCC use model, RTG4FCCC_0 generates APB interface clock at 50 MHz frequency and two 100 MHz clock outputs for sourcing clock to CCC_0 and CCC_1 in dual_cccdyn_top_0 module. CCC_0/CCC_1 generates global output GL0 (PLL OUT0/PLL OUT1), which is divided dynamically using GPD divider.

The following figure shows the SmartDesign top-level block diagram for dual CCC dynamic configuration.

**Figure 7 • SmartDesign Top-level Dual CCC Use Model**
The following figure shows manual connection between CCCDYN and RTG4CCCAPB_IF macro for dual CCC dynamic solution.

_Figure 8 • CCCDYN and RTG4CCCAPB_IF Connection_
2.3.1 CoreABC Program

The CoreABC Program describes the register settings required to divide source input clock using GPD divider.

**Figure 9 • CoreABC Program for Single CCC Use Model**

```assembly
NOP
$Welcome
$restart
WAIT UNTIL INPUT0

//Selecting PLL output clock as input to GPHUX0
APBOUT DAT 0 0x38 0x0F
//Writing GPD0 divider value
APBOUT DAT 0 0x44 0x10
//GPD0 mode
APBOUT DAT 0 0x80 0x03
//Selecting GPD0 output as input to CGLMUX0
APBOUT DAT 0 0x1C 0x00

//Selecting PLL output clock as input to GPHUX1
APBOUT DAT 0 0x9C 0x07
//Writing GPD1 divider value
APBOUT DAT 0 0x4C 0x00
//GPD1 mode
APBOUT DAT 0 0x8C 0x03
//Selecting GPD1 output as input to CGLMUX1
APBOUT DAT 0 0x24 0x00

//Selecting PLL output clock as input to GPHUX2
APBOUT DAT 0 0x90 0x07
//Writing GPD2 divider value
APBOUT DAT 0 0x50 0x02
//GPD2 mode
APBOUT DAT 0 0x90 0x03
//Selecting GPD2 output as input to CGLMUX2
APBOUT DAT 0 0x2C 0x00

//Selecting PLL output clock as input to GPHUX3
APBOUT DAT 0 0x94 0x07
//Writing GPD3 divider value
APBOUT DAT 0 0x54 0x04
//GPD3 mode
APBOUT DAT 0 0x94 0x03
//Selecting GPD3 output as input to CGLMUX3
APBOUT DAT 0 0x30 0x00

| IOWRT DAT 1
Jump $restart

|halt
```

- Setting to divide PLL output with GPD divider value of 16
- Setting to divide PLL output with GPD divider value of 8
- Setting to divide PLL output with GPD divider value of 2
- Setting to divide PLL output with GPD divider value of 4
For dual CCC use model, sequence of register settings to be followed for changing output clock frequency using GPD divider is the same as single CCC use model shown in Figure 9, page 10. In order to access CCC_0 and CCC_1 separately, APB_S_PADDR[8] address bit must be used as shown in the following figure.

2.3.2 Configuring CCC for Dynamic Configuration

The following steps describes how to configure CCC for dynamic configuration in the Libero SoC software.

1. Open RTG4 CCC Configurator and then click the Basic tab.
2. In the Basic tab, enter the values and select the check boxes, as shown in the following figure.

```
//RTG4 CCC0
//Selecting PLL output clock as input to GPMUX0
APBWRAT DAT 0 0x038 0x07
//Writing GPDO divider value
APBWRAT DAT 0 0x040 0x08
//GPDO mode  APB_S_PADDR[8]=0
APBWRAT DAT 0 0x088 0x03
//Selecting GPDO output as input to CGLMUX0
APBWRAT DAT 0 0x01C 0x08

//RTG4 CCC1
//Selecting PLL output clock as input to GPMUX0
APBWRAT DAT 0 0x138 0x07
//Writing GPDO divider value
APBWRAT DAT 0 0x148 0x04
//GPDO mode  APB_S_PADDR[8]=1
APBWRAT DAT 0 0x168 0x03
//Selecting GPDO output as input to CGLMUX0
APBWRAT DAT 0 0x11C 0x08
```
3. In PLL Options tab, selecting **Enable Dynamic configuration** automatically selects **Include reconfiguration logic** for single CCC use model under **CCC Dynamic Configuration**, as shown in the following figure.

**Figure 11 • PLL Output Clock Settings**

![PLL Output Clock Settings](image)

The following figure shows smart design block of a fabric CCC with single dynamic configuration enabled.

**Figure 12 • Single CCC Dynamic Configuration Setting**

![Single CCC Dynamic Configuration Setting](image)

**Figure 13 • Single CCC Dynamic Configuration SmartDesign**

![Single CCC Dynamic Configuration SmartDesign](image)
4. In PLL Options tab, select **Enable Dynamic configuration** for dual CCC use model as shown in the following figure.

*Figure 14* • Dual CCC dynamic configuration setting

The following figure shows smart design block of a fabric CCC with dual dynamic configuration enabled.

*Figure 15* • Dual CCC Dynamic Configuration SmartDesign

5. Click **Ok** after making required setting. CCC generation completes with APB slave interface exposed for manual connection as shown in *Figure 8*, page 9.

2.4 **Simulation Results**

Simulation results for single and dual CCC dynamic configuration use model are shown in the following figures.

In single CCC dynamic configuration, GPD divider values for CCC outputs (GL0, GL1, GL2 and GL3) are written dynamically via APB3 interface as highlighted in the following figure.

*Figure 16* • Single CCC Dynamic Configuration Simulation Window
In dual CCC dynamic configuration, GPD divider values for two CCC outputs (GL0, GL1, GL2, and GL3) are written dynamically via APB3 interface as highlighted in the following figure.

Figure 17 • Dual CCC Dynamic Configuration Simulation Window

2.5 Setting Up the Design

The following steps describe how to setup hardware demo for RTG4 development kit.

1. Connect the jumpers on the RTG4 development kit as shown in the following table.

Table 2 • RTG4 Development Kit Jumper Setting

<table>
<thead>
<tr>
<th>Jumper</th>
<th>Pin From</th>
<th>Pin To</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>J11, J17, J19, J23, J26, J21, J32, and J27</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>J16</td>
<td>2</td>
<td>3</td>
<td>Default</td>
</tr>
<tr>
<td>J33</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>Default</td>
</tr>
</tbody>
</table>

Note: Ensure that power supply switch SW6 is switched OFF while connecting jumpers on RTG4 development kit.

2. Connect the host PC to the J47 connector using the USB cable.
3. Connect the power supply to J9 connector and switch ON the power supply switch, SW6.

2.6 Running the Design

Following are the steps to run the design:

1. Open the design files in the Libero SoC software
2. Program the RTG4 development kit board with the generated or provided *.stp file using FlashPro programmer.
3. Fabric counter logic which is connected to the on board LEDs is run using clock output from CCC in which dynamic configuration is enabled. Once the device is programmed, on board four LED’s (W35, W34, V30, and W33) for single CCC use model and eight LEDs (four for each CCC output) for dual CCC use model blinks.
4. Pressing on board switch SW1 initiates CoreABC program to dynamically write GPD divider values using APB3 interface. After dynamic configuration, slow down in LED’s blinking rate is observed.
Oscilloscope waveform shown in the following figure demonstrates how the CCC output (GL0 and GL1) clock frequency changes dynamically when switch SW1 on board is pressed. Output clocks are connected to bread board connector J10 for probing.

The following table describes the probe points on single and dual CCC.

<table>
<thead>
<tr>
<th>Table 3 • Probe Points</th>
<th>Output Signal</th>
<th>J10 Bread Board Pins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single CCC</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>APB_CLK</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>GL0_DYN</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>GL1_DYN</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>GL2_DYN</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>GL3_DYN</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>LOCK</td>
<td>7</td>
</tr>
<tr>
<td>Dual CCC</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>APB_CLK</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>CCC0_GL0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>CCC0_GL1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>CCC0_GL2</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>CCC0_GL3</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>CCC0_LOCK</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>CCC1_GL0</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>CCC1_GL1</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>CCC1_GL2</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>CCC1_GL3</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>CCC1_LOCK</td>
<td>26</td>
</tr>
</tbody>
</table>
**Note:** The output clock frequency using GPD clock divider is based on:

\[ \text{Clkout} = \frac{\text{Clkin}}{\text{GPD\_DIV}[7:0]}, \text{ for } \text{GPD\_DIV}[7:0] = 1 \text{ to } 255. \]

The following figures shows GL0 and GL1 dynamic frequency change with the help of trigger at the end of dynamic configuration.
**Figure 20 • GL0 Dynamic Frequency Change**

- Trigger raised at end of dynamic configuration
- GL0 frequency changing from 50MHz to 18.8MHz

**Figure 21 • GL1 Dynamic Frequency Change**

- Trigger raised at end of dynamic configuration
- GL1 frequency changing from 75MHz to 37.5MHz
2.7 Conclusion

This document describes how to perform dynamic configuration via an APB3 interface for both single and dual RTG4 FPGA CCC by changing output clock frequency in a glitch less way using GPD dividers. Results are shown using real time oscilloscope plots of the CCC output after dynamic configuration.
The design files are available for download at:
http://soc.microsemi.com/download/rsc/?f=rt4g_ac458_libero11p9sp1_df

The design files consist of a VHDL version of Libero project folder single and dual CCC use models, programming file (*.stp) for RTG4 Development Kit. See the readme.txt file included in the design files for the directory structure and description.