LogiCORE IP Clock Generator (v4.02a)

DS614 June 22, 2011 **Product Specification**

Introduction

The Clock Generator core takes in common clock requirement through its parameters and generates the architecture-specific clocking circuitry. The circuitry is implemented in a VHDL source. When the Clock Generator cannot generate circuitry for the given requirement, it provides failure analysis.

The generation algorithm is implemented in C++ programming language and currently it is only integrated with the Hardware Platform Generator (PlatGen) and Simulation Generator (SimGen) EDK implementation tools.

Features

- Take common clock requirements and generate architecture specific clocking circuitry.
- Support up to 16 different clock requirement.
- Generate synthesizable structural VHDL.
- Indicates generation result through parameter value, messaging in interactive environment and log file.

LogiCORE IP Facts Table		
Core Specifics		
Supported Device Family (1)	Kintex-7(5), Artix-7, Virtex-7, Zynq™-7000, Virtex-6(3)/6CX, Spartan-6(4), Spartan-3A/3A DSP, Spartan-3, Spartan-3E, Automotive Spartan 3/3E/3A/3A DSP, Virtex-5/5FX, Virtex-4/4Q/4QV	
Resources Used		
	Min	**Max**
LUTs	N/A	N/A
FFs	N/A	N/A
Block RAMs	N/A	N/A
DCMs	0	4
PLLs	0	2
PLLEs	0	1
MMCMs	0	4
Provided with Core		
Documentation	Product Specification	
Design Files	VHDL	
Example Design	Not Provided	
Test Bench	Not Provided	
Constraints File	EDK TCL Generated	
Simulation Model	Not Provided	
Tested Design Tools (2)		
Design Entry Tools	ISE®	
Simulation	Mentor Graphics ModelSim	
Synthesis Tools	XST	
Support		
Provided by Xilinx, Inc.		

Notes:

1. For a complete listing of supported devices, see the release notes for this core.
2. For a listing of the supported tool versions, see the ISE Design Suite 13: Release Note Guide.
3. For more information, see the DS150 Virtex-6 Family Overview Product Specification.
4. For more information, see DS160 Spartan-6 Family Overview Product Specification.
5. For more information, see DS180 7 Series FPGAs Overview.

Conventions for this Document

Because the Clock Generator has 16 output clocks named CLKOUTi in which "I" is 0 to 15, CLKOUTi is used to refer to any of the 16 clock output ports named in this document.

Functional Description

The Clock Generator design framework is shown in Figure 1 and described in the following sections.

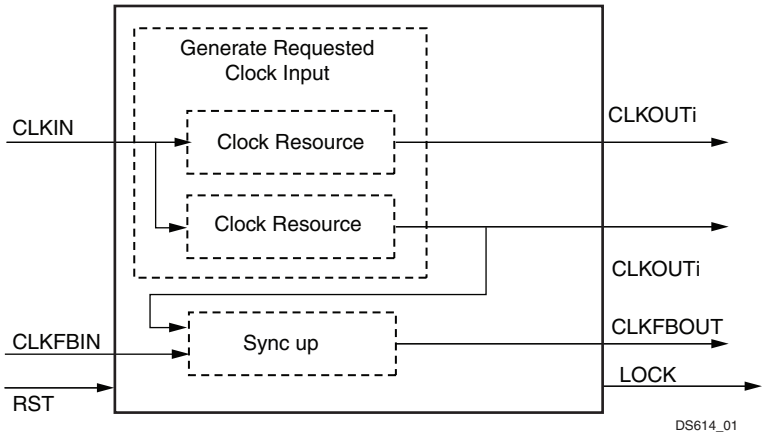

Figure 1: **Clock Generator Modules Block Diagram**

Clock input

The Clock Generator has one input clock port, CLKIN. It is the clock source for the overall clocking circuitry in the Clock Generator. The driving clock for the clock input can be from the off-chip or in-chip source. The system designer decides whether a clocking buffer should be used for the driving clock. The Clock Generator does not insert a buffer for the CLKIN.

Clock circuitry to generate the requested clock outputs

Clock circuitry is dynamically generated based on clock requirement and target FPGA architecture. There could be up to 4 DCM, 2 PLL or 4 MMCM used in the circuitry, depending on the architecture.

The generation algorithm is called by EDK PlatGen and Simgen when the EDK user translates the EDK design to netlist design. The generated VHDL source file is at <project directory>/hdl/elaborate/<clock_generator instance name>_<clock_generator version>/hdl/vhdl.

The clock circuitry is described in the subsequent sections.

Reset input

The Clock Generator connects the reset input port, RST, to the reset input ports of the clock resource in the generated circuitry, for example, the reset port of DCM, reset port of PLL and reset port of MMCM.

When there are cascaded clock resources, the reset port of the clock resource at the downstream is driven by the lock output from upstream clock resource.

Lock output

When there is one clock resource only, its LOCK port is directly connected to the Clock Generator LOCK port.

When there are multiple clock resources cascaded, their LOCK outputs are connected to AND logic and the output of the AND logic connects to the LOCK port of the Clock Generator.

Figure 2 shows a combination of cascading and parallel clock resources.

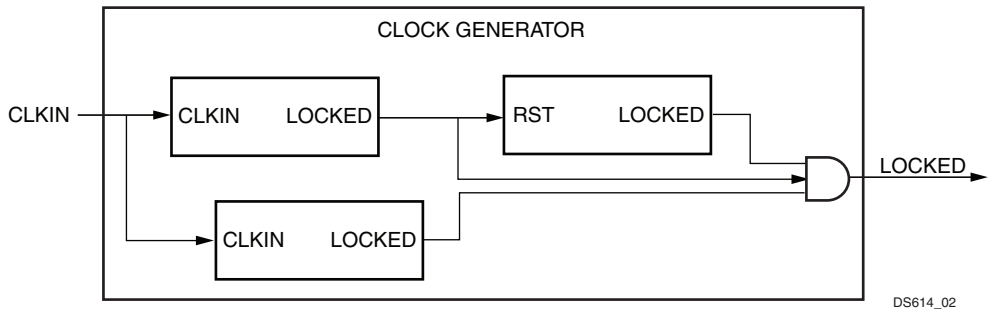

Figure 2: **Reset Connection Example**

If clock requirements cannot be met, the LOCKED output signal remains inactive and the output clocks are undetermined.

With the Spartan®-6 FPGA, the Clock Generator needs to have only two (2) PLL cascaded. The LOCK of the second PLL directly connects to LOCK of the Clock Generator.

Clock sync-up

The Clock Generator provides clock sync-up function for an input clock and for one of the required clocks. The input clock is through clock port, CLKFBIN and its frequency is defined in parameter C_CLKFBIN_FREQ.

With non-Virtex®-6 and non-7 Series FPGAs, the Clock Generator uses the algorithms listed below to generate the sync-up circuitry:

1. Generate a DCM dedicated for this sync-up function.
2. Connect CLKFBIN to CLKIN of the DCM.
3. Go through the required clocks in the sequence: CLKOUT0, CLKOUT1, CLKOUT2, and so forth.
4. Select the first one with same frequency as CLKFBIN as defined by C_CLKFBIN_FREQ.
5. Connect the selected clock to CLKFB of the DCM.
6. Connect CLK0 of the DCM to CLKFBOUT.

On Virtex-6 and 7 Series FPGAs, the Clock Generator uses the algorithms listed below to generate the sync-up circuitry:

1. Generate a MMCM dedicated for this sync-up function.
2. Connect CLKFBIN to CLKIN of the MMCM.
3. If parameter C_CLKFBIN_DESKEW is set to one of 16 CLKOUTi, select that parameter and proceed to step 6; if not, proceed to step 4
4. Go through the required clocks in the sequence: CLKOUT0, CLKOUT1, CLKOUT2, and so forth.
5. Select the first one with same frequency as CLKFBIN as defined by C_CLKFBIN_FREQ.
6. Connect the selected clock to CLKFB of the MMCM.
7. Connect CLKFBOUT of the MMCM to CLKFBOUT.

Design parameters

The parameters defined for the Clock Generator module are listed and described in Table 1.

Table 1: **Clock Generator Parameters**

Parameter Name	Feature Description	Allowable Values	Default	VHDL Type
C_CLKIN_FREQ	Frequency (Hz) of CLKIN	natural	0	integer
C_CLKFBIN_FREQ	Frequency (Hz) of CLKFBIN	natural	0	integer
C_CLKFBIN_DESKEW	Applicable to Virtex-6 and 7 Series devices only. Set the clock output port to be used to deskew with CLKFBIN.	NONE — Let the algorithm select from CLKOUTi; i is 0 to 15. Deskew this clock output with CLKFBIN.	NONE	string
C_CLKFBOUT_FREQ	This must be equal to C_CLKBIN_FREQ.	Frequency (HZ) of CLKFBOUT port.	0	integer
C_CLKFBOUT_PHASE	This must be 0. Applicable to Virtex-6 and 7 Series devices only.	Phase of CLKFBOUT port.	0	integer
C_CLKFBOUT_GROUP (1)	Applicable to Virtex-6 and 7 Series devices only. Group name of CLKFBOUTi. The MMCM used for clock deskew is named as this parameter value.	NONE, MMCM0, MMCM1, MMCM2, MMCM3	NONE	string
C_CLKFBOUT_BUF	Insert BUFG for CLKFBOUT.	TRUE: BUFG is inserted. FALSE: BUFG is not inserted.	TRUE	boolean
C_PSDONE_GROUP	Applicable to Virtex-6 and 7 Series devices only. Group name of PSDONE to specify the MMDM with variable phase. (2)	NONE, MMCM0, MMCM0_FB, MMCM1, MMCM1_FB, MMCM2, MMCM2_FB, MMCM3, MMCM3_FB	NONE	string
C_EXT_RESET_HIGH	Reset polarity of RST port.	0: active Low 1: active High	1	integer
C_CLK_PRIMITIVE_ FEEDBACK_BUF	Applicable to Virtex-6 and 7 Series devices only. Insert BUFG into the self feedback path of the clock resource MMCM.	TRUE: insert BUFG FALSE: does not insert BUFG	FALSE	boolean

Table 1: **Clock Generator Parameters *(Cont'd)***

Parameter Name	Feature Description	Allowable Values	Default	VHDL Type
C_CLK_GEN	Set the value to UPDATE to generate for generation algorithm to generate circuitry. The generation algorithm sets the value to PASSED or FAILED in the generated VHDL source indicating the result.	UPDATE, PASSED, FAILED	UPDATE	string
C_FAMILY	The EDK tool sets the parameter. User assignment is ignored.	Check the Clock generator MPDs ARCH_SUPPORT_MAP for complete FPGA family support.	virtex6	string
C_DEVICE	Specify the target FPGA device.	Valid FPGA device.	NOT_SET	string
C_PACKAGE	Specify the target FPGA package.	Valid FPGA package.	NOT_SET	string
C_SPEEDGRADE	Specify the target FPGA speed grade.	Valid FPGA speed grade	NOT_SET	string

Notes:

1. See detailed descriptions in the subsequent sections.
2. See detailed descriptions and examples in the subsequent sections and in Table 2.

I/O Signals

The interface signals for the Clock Generator module are listed and described in Table 2.

Table 2: **Clock Generator Signal Descriptions**

Signal Name	I/O	Initial State	Description
CLKIN	I		Connect to CLKIN of DCM, PLL, or MMCM.
CLKOUTi: i=0 -15	O	Low	Connect to the clock output port of a DCM, PLL, or MMCM optionally through a BUFG or an inverter.
CLKFBIN	I		Connect to CLKFB of DCM or the CLKFBIN port of an MMCM.
CLKFBOUT	O	Low	Connect to the CLK0 port of a DCM or the CLKBOUT port of an MMCM.
LOCKED	O	Low	LOCKED = High indicates that all required clocks are stable.
PSCLK	I		Connect to PSCLK of MMCM. Applicable for Virtex-6 and 7 Seriees devices only.
PSEN	I		Connect to PSEN of MMCM. Applicable for Virtex-6 and 7 Series devices only.
PSINCDEC	I		Connect to PSINDEC of MMCM. Applicable for Virtex-6 and 7 Series devices only.
PSDONE	O		Connect to PSDONE of MMCM. Applicable for Virtex-6 and 7 Series devices only.
RST	I		If C_EXT_RESET_HIGH = 0, an inverter is inserted; otherwise, this signal is connected to the reset port of the DCM, PLL, PLLE or MMCM.

Parameter - Port Dependencies

Table 3 shows the effect of setting various parameters.

Table 3: **Clock Generator Parameter-Port Dependencies**

Parameter	Port	Description
C_CLKIN_FREQ	CLKIN and all output ports	If C_CLKIN_FREQ is 0, clock_generator has no function.
C_CLKOUT*i*_FREQ	CLKOUT*i* (*i*=0,..,15)	If C_CLKOUTi_FREQ is 0, CLKOUTI is not used and the corresponding C_CLKOUTi_BUF, C_CLKOUTi_GROUP, C_CLKOUTi_VARIABLE_PHASE, and C_CLKOUTi_PHASE are ignored.
C_CLKFBIN_FREQ	CLKFBIN, CLKFBOUT	If CLKFBIN_FREQ is 0, all the ports and corresponding parameters of the sync up function module are ignored. Refer to the clock sync up section for details.

Clock Circuitry Generation Algorithm

On Spartan-3 and Virtex-4 FPGAs, DCM is used for clock generation and clock sync-up; on Virtex-5 and Spartan-6 FPGAs, PLL and DCM is used for clock generation and DCM for clock sync-up; on Virtex-6 FPGAs, MMCM is used for clock generation and clock sync-up. On 7 Series FPGAs, MMCM and PLLE2 are used for clock generation and MMCM is used for clock sync-up. The clock sync-up function always uses one dedicated resource, either DCM or MMCM. The clock generation algorithm adopts the following guidelines:

- Use as few clock resource of DCM, PLL or MMCM as possible
- When multiple resources are used, the preference is to put them in parallel rather than cascading them
- If cascading, do one level at the most.

Static Phase Shift

Phase shift is set directly to the clock resource, as is set on the Clock Generator.

When there are two (2) CLKOUTi signals with the same frequency, but with opposite phase alignment, the algorithm uses inverter logic to generate one clock from the other.

De-skew Among Clock Outputs

When a multiple CLKOUTi signal needs to be phase aligned, set the corresponding C_CLKOUTi_GROUP parameters to the same value. For example, given the following parameters:

Parameter C_CLKOUT0_GROUP = pll0

Parameter C_CLKOUT1_GROUP = pll0

The generation algorithm will ensure that CLKOUT0 and CLKOUT1 are from same PLL and that the instance name of that PLL is "pll0".

When phase alignment is not required, set C_CLKOUTi_GROUP to "NONE".

For a design with the EDK PowerPC® 440 core, when the C_CLKOUTi_GROUP parameter of a CLKOUTi port is set to PLL0_ADJUST or PLL1_ADJUST, the algorithm sets the C_COMPENSATION parameter of the PLLs to "SYSTEM_SYNCHRONOUS".

Dynamic Phase Shift

Dynamic phase shift is supported only on Virtex-6 and 7 Series FPGAs. Below is an example how to use dynamic phase shift.

```
PARAMETER C_CLKOUT3_VARIABLE_PHASE = TRUE
PARAMETER C_PSDONE_GROUP = MMCM1_FB
PARAMETER C_PSDONE_GROUP = MMCM1_FB
```

In this example, CLKOUT3 of the Clock Generator is the dynamic phase shift port, PSDONE of one MMCM that connects to PSDONE of the Clock Generator. That MMCM is named "MMCM1" and its feedback is dynamically phase shift. Another is shown below.

```
PARAMETER C_CLKOUT3_VARIABLE_PHASE = TRUE
PARAMETER C_PSDONE_GROUP = MMCM1
```

In the second example, CLKOUT3 of the Clock Generator is the dynamic phase shift port, PSDONE, of one MMCM that connects to PSDONE of the Clock Generator. That MMCM is named "MMCM1" and its feedback is *not* dynamically phase shift.

Restrictions

Because the proposed target function of the Clock Generator is ease-of-use rather than functional completeness, not all the clock functions provided by DCM, PLL, MMCM, and clocking buffer are available in the Clock Generator. For example, it does not utilize CLKIN2 of MMCM_ADV on the Virtex-6 device.

In addition, the Clock Generator does not explore all possible scenarios of clock circuitry, so it is likely that a system designer can devise circuitry manually. In that case, the Clock Generator will not be able to identify any working circuitry and will report a failure.

Differences Between Clock Generator v3.02a and v4.01a

Because low level parameters were removed from v4.00a of the Clock Generator, the user is not able to directly manipulate the final clock circuitry in the MHS file and must rely on the core algorithm to generate the circuitry.

The Clock Generator v4.01a has all the high level parameters of v3.02a. To migrate the design with the Clock Generator v3.02a to Clock Generator v4.01a, change the core version from 3.02a to 4.01a. Please note that the delineated procedure does not work if the low level parameters of v3.02a are used (PARAMETER C_CLK_GEN is defined in the design and its value is not "UPDATE").

Differences Between Clock Generator v4.00a and v4.01a

Because low level parameters were removed from v4.00a of the Clock Generator, the user is not able to directly manipulate the final clock circuitry in MHS file and must rely on the core algorithm to generate the circuitry.

The Clock Generator v4.01.a has all the high level parameters of v3.02a. To migrate the design with Clock Generator v3.02.a toClock Generator v4.01.a, change the core version from 3.02a to 4.01a. Please note that the delineated procedure does not work if low level parameters of v3.02a are used (PARAMETER C_CLK_GEN is defined in the design and its value is not "UPDATE").

Differences Between Clock Generator v4.00a and v4.01a

Because parameter C_CLK_PRIMITIVE_FEEDBACK_BUF has been added to v4.01a, setting this parameter to TRUE causes BUFG to be inserted into the self feedback paths of all clock resources. This setting brings better phase alignment between CLKIN and CLKOUTi. The default value is FALSE.

To migrate the design with the Clock Generator v4.00a to the Clock Generator v4.01a, change the core version from 4.00a to 4.01a only.

Differences Between Clock Generator v4.01a and v4.02a

The Clock Generator v4.02a has all the high level parameters of the v4.01a. To migrate the design with Clock Generator v4.01a to Clock Generator v4.02a, change the core version from 4.01a to 4.02a.

New parameters, C_CLKOUT0_DUTY_CYCLE to C_CLKOUT15_DUTY_CYCLE, have been added to v4.02a. The default value of these parameters is "0.5". The user is not allowed to change these parameters from SAV by clicking on clock_generator v4.02a. The PLLE0 group has been added to the list of all C_CLKOUT*_GROUP.

Design Implementation

Target Technology

The target technology is an FPGA listed in the Supported Device Family field of the LogiCORE IP Facts Table.

Device Utilization and Performance Benchmarks

The device utilization depends on the number of output clocks used and the value of the parameters of each output clock. Up to 4 DCM modules, 2 PLL modules and 4 MMCM modules may be instanced with BUFGs, clock inverters, and reset logics. See respective FPGA family user guide for details on DCM, PLL, MMCM, and BUFG primitive performance and available resources.

In one Clock Generator v4.02a module:

- All 7 Series FPGAs will use up to 4 MMCMs and 1 PLLE2
- Virtex-6 family FPGAs will use up to 4 MMCMs (no DCM or PLL)
- Virtex-5 and Spartan-6 family FPGAs will use up to 2 PLLs and 4 DCMs (no MMCM)
- All other FPGA families will use up to 4 DCMs (no PLL or MMCM)

Support

Xilinx provides technical support for this LogiCORE product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY*.

Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx ISE Design Suite Embedded Edition software under the terms of the Xilinx End User License. The core is generated using the Xilinx ISE Embedded Edition software (EDK)

For more information, please visit the Clock Generator product web page.

Information about this and other Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page. For information on pricing and availability of other Xilinx LogiCORE modules and software, please contact your local Xilinx sales representative.

Revision History

Date	Version	Description of Revisions
5/15/07	1.0	Initial Xilinx release.
1/16/08	1.1	Released v2.00a; added PLL support.
4/22/08	1.2	Released v2.01a; added Automotive SP3E, SP3A, SP3, and SP3A DSP support.
7/25/08	1.3	Added QPro Virtex-4 Hi-Rel, QPro Virtex-4 Rad Tolerant, and SP-3AN support.
3/31/09	1.4	Release v3.00a, changed C_CLK_GEN parameter, removed obsolete parameters.
6/24/09	1.5	Released v3.01a; added MMCM support.
12/2/09	1.6	Released v3.02a for EDK_L 11.4.
4/19/10	1.7	Released v4.00a for EDK 12.1; removed low-level internal view parameters.
12/14/10	1.8	Released v4.01a for EDK 12.4.
6/22/2011	1.9	Released v4.02a for EDK 13.2; added 7 Series support; incorporated CR592667.

Notice of Disclaimer