Delta39K™ PLL and Clock Tree

Introduction

The purpose of this application note is to provide information and instruction in utilizing the functionality of the Delta39K™ Phase-Locked Loop (PLL) and associated clock tree. Delta39K is a family of high-density Complex Programmable Logic Devices (CPLDs) containing on-chip components such as Single-Port RAM, advanced Dual-Port RAM, and a PLL. The Delta39K PLL can be used in any system requiring clock frequency or clock phase manipulation.

For Delta39K, programming is defined as the loading of a user's design into the on-chip FLASH device internal to the Delta39K package. Configuration, on the other hand, is the loading of a user's design into the volatile Delta39K die.

Overview of PLL & Clock Tree

Within each 3.3V/2.5V Delta39K device, a single on-chip PLL resides as part of a larger clocking scheme. Four local dedicated clock input pins, referred to here as GCLK[3:0], provide direct inputs to this clock tree. GCLK[0] and GCLK[1] may also be used as an input port and external feedback port, respectively to the PLL. Within the clock tree are four global clocks, referred to here as INTCLK[3:0], which are accessible to any macrocell, I/O cell, or memory block. GCLK[3:0] and the outputs of the PLL feed a set of multiplexors which source INTCLK[3:0]. *Figure 1* contains a block diagram of the clock tree and PLL.

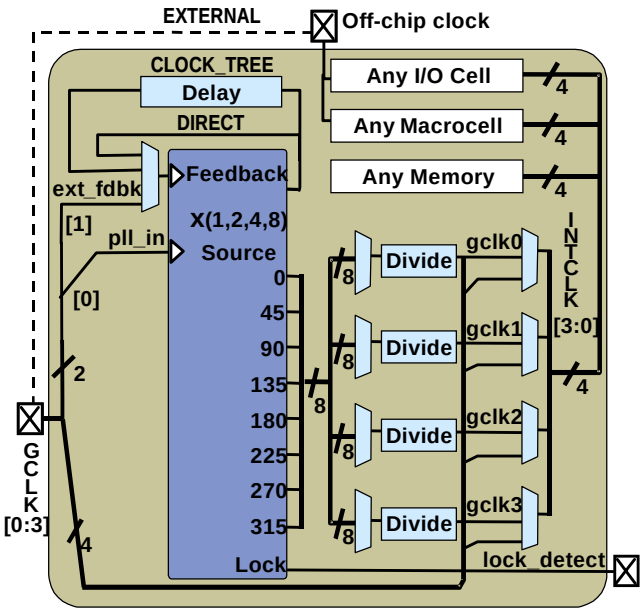


Figure 1. Delta39K Phase Locked Loop and Clock Tree.

The Delta39K PLL and the global clock tree provide designers with functionality that can be configured to meet various design requirements. This functionality includes clock phase adjustment, clock multiplication and division, Spread Aware™ feature, lock detection, off-chip clocking and buffering, and JTAG support.

Clock Phase Adjustment

Designers may use the PLL and clock tree to re-position the edges of the PLL-generated clock in order to shift performance toward either improved set-up time or improved clock-to-out time. The clock's phase, or the position of its edges relative to the PLL input, may be adjusted in either of two ways: Skewing the clock moves its phase backward on the time axis, while de-skewing the clock moves its phase foreword on the time axis.

There are eight options for skewing the incoming clock. The clock can be skewed so that the phase is delayed 0°, 45°, 90°, 135°, 180°, 225°, 270°, or 315°. A 45° phase shift increment is equal to a delay of 1/8th of the clock's period length. Note that adding delay to the incoming clock has the effect of increasing effective clock-to-out time while decreasing the effective setup time requirements. Refer to *Figure 2* for an example.

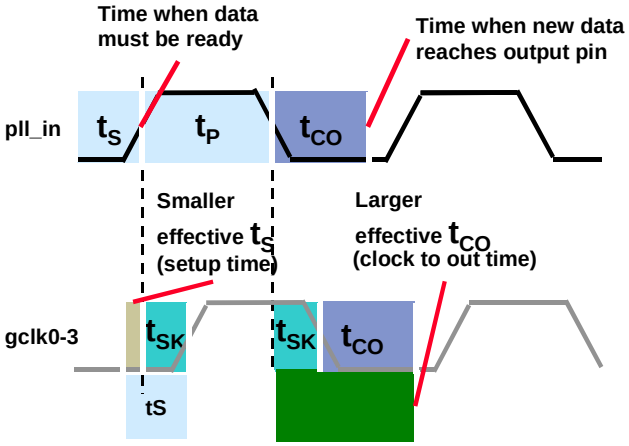


Figure 2. Skewed Clock and Adjusted t_S and t_{CO}.

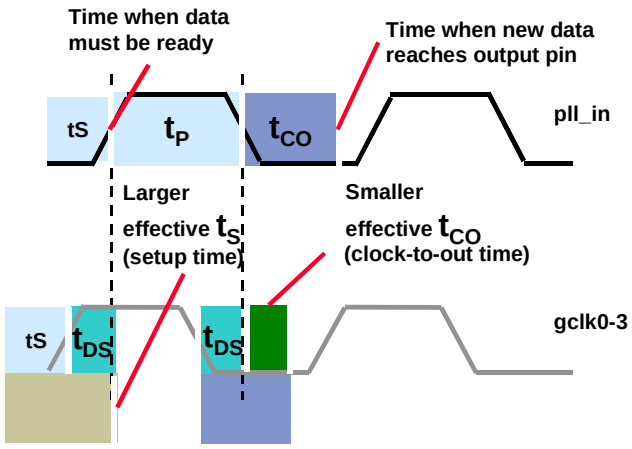


Figure 3. De-skewed Clock and Adjusted t_S and t_{CO}.

There are two options for de-skewing a clock using the PLL. Both of these involve the insertion of a delay into the PLL feedback path, which decreases the effective clock-to-out time by the delay value but increases the effective set-up time requirement by the same amount. Please see *Figure 3* for a description of this.

The first option is to insert a delay equal to the delay of the clock tree in the 39K by selecting the feedback path labeled “CLOCK_TREE” in *Figure 1*.

The second option is to feedback a PLL-generated off-chip clock through the GCLK[1] pin to ext_fdbk. This means that whatever delay that is associated with the connection between the off-chip clock and GCLK[1] will become included in the PLL’s feedback path. This delay will directly determine to what extent gclk0-3 are de-skewed. This path is indicated as a dashed line and is labeled “EXTERNAL” in *Figure 1*.

Using the DIRECT feedback path means that an internal trace directly connects the PLL output to the PLL feedback input. This configuration should be used when no de-skew is desired. This path is labeled “DIRECT” in *Figure 1*.

When gclk0-3 have been derived from any of the 7 phase-delayed or skewed PLL outputs, selecting CLOCK_TREE or EXTERNAL as the method of feedback in order to implement de-skew will cancel part or all of this skew. Therefore, it is not possible to improve set-up and clock-to-out times simultaneously on a single clock.

Selecting a Phase Adjustment Option

Only one of the three feedback option values may be inserted into the LPM instantiation. Selecting the EXTERNAL option will require the use of an off-chip clock, and will constrain the number of valid values that may be selected for the Multiplication/Division and Phase Adjustment options. To view these constraints, please see *Tables 1*, *2 and 4*.

Table 1. Valid Skew Option Values

Skew Option	
No EXTERNAL Feedback	**EXTERNAL Feedback**
0, 45, 90, 135, 180, 225, 270, 315	0

Table 2. Valid De-skew Option Values

De-skew Option[1]		
No De-skew Desired	**Clock Tree De-skew Desired**	**Board De-skew Desired**
DIRECT	CLOCK_TREE	EXTERNAL

Note:
1. Add a ‘prefix to each of the options when instantiating in Verilog.

If EXTERNAL is selected as the method of feedback, only Phase 0 may be selected as the phase adjustment option value in the LPM instantiation for the on-chip global clock that drives the off-chip clock used as the external feedback. If CLOCK_TREE or DIRECT is selected, any of the eight phases may also be selected. The four global clocks, INTCLK[3:0], may be inverted locally at the macrocell register or I/O register used in order to create additional phases. The resulting phase delay would be 180° for clocks that have been generated by the PLL and have passed through a divider with a binary divide value (1, 2, 4, 8, or 16). The phase delay is the same for clocks that have been derived directly from one of the GCLK[3:0] input pins without using the PLL. However, if any of these global clocks have been generated by the PLL and pass through a divider with a non-binary divide value (3, 5, or 6), the amount of phase delay will vary based on their non-50% duty cycles.

Clock Multiplication and Division

Designers may use the PLL and Clock Tree to multiply or divide the incoming clock to generate multiple frequencies, to increase or decrease incoming clock speeds, or to correct the incoming clock’s duty cycle.

The clock can be multiplied by a factor of 1, 2, 4 or 8. Each multiply option selects an element, which is effectively a counter or divider set to the appropriate value, for use in the feedback path. The result is a PLL-generated clock that is either the same speed as or is multiplied up to 8 times faster than the incoming clock.

The PLL-generated clock may be divided by one of eight factors after the incoming clock has been multiplied. Division selections are 1, 2, 3, 4, 5, 6, 8, and 16.

Selecting a Multiplication/Division Option

The Voltage Controlled Oscillator (VCO), the core of the Delta39K PLL is designed to operate within the frequency range of 100 MHz to 266 MHz. When using the DIRECT or CLOCK_TREE feedback configurations the pll_in frequency (GCLK0) and multiply option must be selected to ensure the VCO is operating within this range. Valid pll_in frequencies for each multiply option are detailed in *Table 3*. Note the maximum input frequency to the PLL is limited to 133MHz. In these

configurations any one of the eight division settings may be individually selected for gclk[0-3].

When the EXTERNAL feedback option is selected only multiplication by a factor of one and division by a factor of one may be selected for the off-chip clock that is being used in the feedback path. The other three PLL generated clocks may use any of the eight division settings. Valid pll_in frequencies, multiply and divide options for this configuration are detailed in *table 4*.

Note that if an on-chip clock is sent off-chip, it will be further divided by two on top of any divide settings chosen

Multiplying an input frequency allows one to use slower frequencies throughout the board (externally) and use faster frequencies internally. (Assuming they are not needed other places so as to make this less advantageous). This allows for less power consumption, less electromagnetic interference (EMI), and less undesirable high speed signal integrity issues.

Also, designs requiring a given throughput or bandwidth running at a lower frequency and wider bus-width, can be increased to a higher frequency, while the bus-width can be reduced to a narrower width. This way, less logic resources (real estate) can be used on the Cypress CPLD, while maintaining the same throughput.

Table 3. Valid Multiply & Divide Options: DIRECT or CLOCK_TREE Selected

Valid Multiply Option			Valid Divide Option		
Value	**pll_in Freq (MHz)**	**VCO Freq (MHz)**	**Value**	**gclk0-3 Freq (MHz)**	**Off-chip Freq[2] (MHz)**
N/A	DC-12.5	N/A	N/A	DC-12.5	DC-6.25
8	12.5- 33	100-266	1-6,8,16	6.25-266	3.125-133
4	25-66	100-266	1-6,8,16	6.25-266	3.125-133
2	50-133	100-266	1-6,8,16	6.25-266	3.125-133
1	100-133	100-133	1-6,8,16	6.25-133	3.125-66
N/A	133-f_{MAX}	N/A	N/A	133-f_{MAX}	66-f_{MAX}

Table 4. Valid Multiply & Divide Options: EXTERNAL Selected

pll_in Freq (MHz)	Valid Multiply Option		Valid Divide Option		
	Value	**VCO Freq[3] (MHz)**	**Value**	**gclk0-3 Freq (MHz)**	**Off-chip Freq (MHz)**
50-133	1	100-266	**For Feedback Path**		
			1	100-266	50-133
			For Non-Feedback Path		
			1-6,8,16	6.25-266	3.125-133

Spread Aware™ Feature

The PLLs incorporated in all Delta39K CPLDs are Spread Aware. This feature refers to the ability of the PLL to track a spread-spectrum input clock such that its spread is seen on the output clock. Spread-spectrum is a method of 'spreading' or modulating a fundamental or original frequency in a controlled, oscillatory manner, such that the electromagnetic energy broadcast at any given component in the frequency spectrum is below the maximum value imposed by FCC regulations. Spread Aware does not mean that the PLL is capable of generating a spread-spectrum output from a non spread-spectrum input.

When configured with a x1, x2 or x4 multiply option the Delta39K PLL is Spread Aware whereas the x8 multiply option does not support the Spread Aware feature.

Designers may choose to use the spread aware Delta39K PLL with a spread spectrum input clock. Down spread, up spread, or some form of middle spread such as center spread may be used on the incoming clock. However, the total amount of spread in either or both directions should be limited to 0.6% of the fundamental frequency. The modulation frequency of the spread should be 50 kHz or less in order to ensure that the PLL will meet the performance specified in the data sheet and maintain its intended 1 MHz loop bandwidth.

Lock Detection

A finite amount of time is required from the moment that the incoming clock signal is placed on the GCLK[0] pin, and thus drives the source of the PLL, to the time that the PLL actually achieves steady-state operation, or lock. Designs that contain logic that will not operate properly until a valid clock signal is present at the output of the PLL, are dependent upon the PLL first achieving lock. Designs such as these may require a signal to indicate when the PLL has reached lock. The PLL can be configured to generate a lock detect signal at a dedicated I/O pin. This pin is fed by a multiplexor that can be configured to select either a general-purpose I/O function or a lock detection indicator function for this pin.

Notes:

2. An off-chip clock is the output of a toggle flip-flop, which acts as a /2.
3. When an off-chip clock is fed back to ext_fdbk, the toggle flip-flop is effectively placed into the feedback path, resulting in an implicit x2 on top of the x1 multiplication setting.

JTAG Support

The Delta39K PLL supports the JTAG instruction INTEST. When the Delta39K device is in JTAG mode and is executing the instruction INTEST, the clock driving the TCK pin is multiplexed onto INTCLK[3:0]. This allows the internal logic to be controlled by the JTAG clock, TCK, instead of the system clock, and enables the user to verify proper operation of his or her design in a given device. For more information about JTAG programming, please refer to the application note titled 'Using IEEE 1149.1 Boundary Scan (JTAG) With Cypress Ultra37000 CPLDs,' which can be downloaded from http://www.cypress.com/pld/pldappnotes.html.

Off-Chip Clocking and Buffering

The Delta39K global clocks, INTCLK[3:0], may be driven off-chip by clocking a macrocell or I/O register configured as a toggle flip-flop (TFF). This way, upon every rising edge of the INTCLK clock used, the register output level will toggle from its current state (HIGH or LOW) to the opposite state (LOW or HIGH). Since, for every two rising edges of the INTCLK clock used a single rising edge at the register output is generated, a periodic signal that is half the frequency of the register's clock will result. In order to illustrate how to implement off-chip clocks in source code, sample code in both VHDL and Verilog is listed below:

```
library ieee;
use ieee.std_logic_1164.all;

entity off_chip_example is
  port (
        clkin:        in   std_logic;
offchipclk: buffer   std_logic
 );
end off_chip_example;

architecture off_chip_arch of off_chip_example is
signal    shifted_clock:  std_logic;
begin

plloutclock<=clkin;

p1:  process (plloutclock)
       begin
         if (plloutclock'event and plloutclock= '1') then
   offchipclock <= not(offchipclock);
 end if;
     end process;

end off_chip_arch;
```

Listing 1a: VHDL Example of Off-chip Clock

```
module off_chip_example(clkin, offchipclock);
input clkin;
output offchipclock;

wire plloutclock;
reg offchipclock;

assign plloutclock  = clkin;

always@ (posedge plloutclock)
begin
if (plloutclock)
  offchipclock <= ~( offchipclock);
end
endmodule
```

Listing 1b: Verilog Example of Off-chip Clock

Multiple identical output clocks can be buffered from a single global clock by driving multiple outputs or 'copies' to the same output described, above.

Overview of Software Support

The Delta39K PLL can be configured in either VHDL or Verilog code using *Warp* Enterprise™, *Warp* Professional™, or *Warp*® Release 6.0 or later software. The methodology for implementing this configuration is via a component, module, or LPM instantiation. Since Delta39K contains a single PLL, each design project should contain only one instantiation of the PLL.

Selecting a Target Device

When creating a new project or editing a current project to target a device, only 2.5V/3.3V devices in the Delta39K family should be selected if PLL functionality is desired. These devices are indicated by a 'V' in the package name, while low voltage device selections include a “Z” in the package name and do not offer PLL functionality. An example of such a device selection is “CY39100V676-125MBC”.

Component / Module or LPM Instantiation

The PLL configuration may be implemented by instantiating the PLL component “cy_c39kpll.” For *Warp* to recognize these instantiations, the user must place the appropriate VHDL 'library' and 'use' statements or Verilog ‘include’ statements in the source code file where the PLL instantiation resides. For an example of the PLL instantiation in source code, please see Listing 2a and 2b. In these samples, 'clkin' is multiplied by 4, while clock signals 'm1p0clock', 'm2p45clock', 'm4p0clock', and 'm4p90clock' are internal clocks accessible to every register on the selected Delta39K device. Each of the four internal clocks is configured with different divide and phase shift selections. 'm1p0clock' is also configured to toggle 'off chip clock' as an off-chip clock. The method of feedback selected is 'DIRECT'.

```
library ieee;
use ieee.std_logic_1164.all;
```

```
library cypress;
use cypress.lpmpkg.all;
use cypress.rtlpkg.all;

entity pll_example is
  port (
        clkin:       instd_logic;
offchipclock:    bufferstd_logic
  );
end pll_example;

architecture pll_arch of pll_example is
signal    m1p0clock:   std_logic;
begin

U0: cy_c39kpll
generic map(
feedback      => DIRECT,-- optional
multiply       => 4,-- optional
gclk0_phase    => 0,-- optional
gclk0_divide   => 4,-- optional
gclk1_phase    => 0,-- optional
gclk1_divide   => 1,-- optional
gclk2_phase    => 0,-- optional
gclk2_divide   => 1,-- optional
gclk3_phase    => 0,-- optional
gclk3_divide   => 1 -- optional
)
port map(
pll_in          => clkin,
ext_fdbk       => zero,-- optional
lock_detect   => open,-- optional
gclk0           => m1p0clock,-- optional
gclk1          => open,-- optional
gclk2          => open,  -- optional
gclk3          => open -- optional
);

p0:   process (m1p0clock)
        begin
          if (m1p0clock'event and m1p0clock = '1') then
    offchipclock <= not(offchipclock);
 end if;
      end process;
end pll_arch;
```

Listing 2a: VHDL Sample of PLL Instantiation

```
`include "lpm.v"
`include "rtl.v"
`define DIRECT 0

module pll_example(clkin, offchipclock);

input clkin;
output offchipclock;

reg m1p0clock;
reg offchipclock;

defparam U0.feedback  = `DIRECT; // optional
defparam U0.multiply        = 4;// optional
defparam U0.gclk0_phase   = 0;// optional
defparam U0.gclk0_divide  = 4;// optional
defparam U0.gclk1_phase   = 0;// optional
defparam U0.gclk1_divide  = 1;// optional
defparam U0.gclk2_phase   = 0;// optional
defparam U0.gclk2_divide  = 1;// optional
defparam U0.gclk3_phase   = 0;// optional
defparam U0.gclk3_divide  = 1;// optional
cy_c39kpll  U0(
.pll_in( clkin ),
//.ext_fdbk(0),// optional
//.lock_detect(),// optional
.gclk0    (m1p0clock),  // optional
.gclk1    (),// optional
.gclk2    (),// optional
.gclk3    ());            // optional

always @(posedge m1p0clock)
begin
  offchipclock = ~(offchipclock);
end
endmodule
```

Listing 2b: Verilog Sample of PLL Instantiation

The PLL instantiation can be pasted into the source code by selecting “CY 39kPLL” from the “Template” pull-down on the toolbar. The text of the PLL instantiation will be automatically inserted directly at the last cursor position in the source code displayed in the editor window. There will be default values listed for some of the instantiation items, but those that are either blank or require modification must be typed in manually

by the user. For a brief description of each port and LPM property, please see *Figure 5* and *Table 6*.

Report File Information

When the 'Detailed Report File' option has been selected from the 'Messaging' tab in 'the 'Compiler Options' dialogue box, all PLL and clock tree configuration information is printed into the Warp report file after compilation. Specifically, this information can be found in the 'DESIGN EQUATIONS' section of the report file under 'Clock/PLL listing'. Please see *Figure 4* for a snapshot of this text. To select the appropriate compiler setting, select 'Compiler Options' from the 'Project' dropdown menu in *Warp*.

```
--------------------------------------------------------------
DESIGN EQUATIONS
--------------------------------------------------------------

    --------------------------------------------------------------
    Clock/PLL listing
    --------------------------------------------------------------

    Clock/PLL Cell: Name=ul, Mode=(Multiplier=4, FeedBack=Direct)
        GCLKD => mlpDclock, Phase=0, Divider=4
        GCLK1 => --Not used--
        GCLK2 => --Not used--
        GCLK3 => --Not used--
        LockDetect => --Not used--
        clkin @ GCLKD, IoType=<LVTTL,3.3,8A>
```

Architecture Explorer

The PLL along with the clock tree is shown as a green rectangular box in the upper right hand corner of Delta39K devices in the Architecture Explorer. Right-clicking over this box will open a window from which “Properties” can be selected. Doing so will display the same information that is contained in the report file screen shot, mentioned above. Please see Figure 5 for a screenshot of the PLL / Clock Tree in the Architecture Explorer.

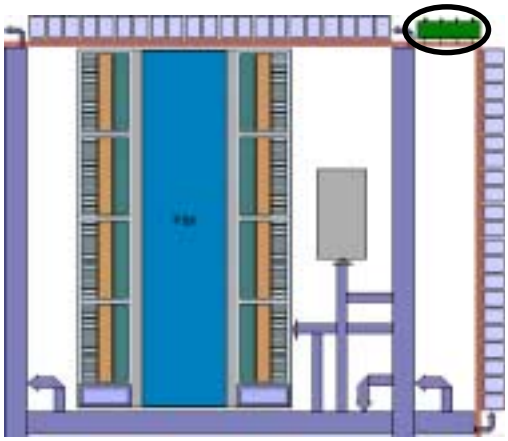


Figure 4. Screenshot of PLL in Architecture Explorer.

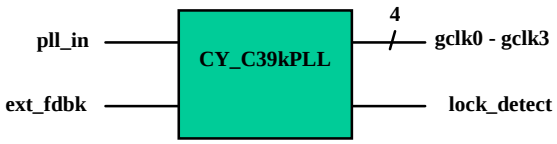
Using the zoom feature on this box will reveal the four clock input connections, which are GCLK[3:0] and the four clock output connections, which are INTCLK[3:0]. Zooming in more closely will reveal text labels at each of these connections identifying the input and output clock signal names assigned in each case.

Valid Frequency Ranges

In order to ensure that the PLL and Clock Tree operate correctly and perform within the data sheet's specifications, certain ranges of operating frequency have been determined for the inputs, outputs, and Voltage Controlled Oscillator (VCO) contained within the PLL and Clock Tree. Operating outside of these ranges may result in poor performance, non-operability, or damage to the board or related devices.

Table 5. Valid Operating Ranges for Clock Tree and PLL

Item		Operating Range (MHz)
Clock Tree	GCLK[3:0]	DC-f_{MAX}
	INTCLK[3:0]	DC-f_{MAX}
	Off-chip clock	DC-f_{MAX}/2
PLL	pll_in	12.5-133
	ext_fdbk	50-133
	gclk0-gclk3	6.25-266
	VCO	100-266



LPM PROPERTIES

feedback	gclk0_divide
multiply	gclk1_divide
gclk0_phase	gclk2_divide
gclk1_phase	gclk3_divide
gclk2_phase	
gclk3_phase	

Figure 5. CY_C39kPLL LPM Module Ports and Properties.

Table 6. Description of PLL Instantiation Port and Properties

	Name	Type	Usage	Description
PORT	pll_in	In	Required	Source reference input to PLL
	ext_fdbk	In		External feedback input to PLL
	lock_detect	Out		Output to dedicated pin indicating lock detection
	gclk0	Out	at least one of these is Required	PLL output clock driving Intclk[0]
	gclk1	Out		PLL output clock driving Intclk[1]
	gclk2	Out		PLL output clock driving Intclk[2]
	gclk3	Out		PLL output clock driving Intclk[3]
PROPERTY	feedback	In	Optional	Property to select de-skewing method
	multiply	In	Optional	Property to select multiplication factor
	gclk0_phase	In	Optional	Property to select incremental delay, based on VCO frequency for gclk0
	gclk1_phase	In	Optional	Property to select incremental delay, based on VCO frequency for gclk1
	gclk2_phase	In	Optional	Property to select incremental delay, based on VCO frequency for gclk2
	gclk3_phase	In	Optional	Property to select incremental delay, based on VCO frequency for gclk3
	gclk0_divide	In	Optional	Property to select division factor for gclk0
	gclk1_divide	In	Optional	Property to select division factor for gclk1
	gclk2_divide	In	Optional	Property to select division factor for gclk2
	gclk3_divide	In	Optional	Property to select division factor for gclk3