XAPP1065 (v1.0) March 22, 2010

Spread-Spectrum Clock Generation in Spartan-6 FPGAs

Author: Jim Tatsukawa

Summary

Consumer display applications commonly use high-speed LVDS interfaces to transfer video data. Spread-spectrum clocking can be used to address electromagnetic compatibility (EMC) issues within these consumer devices. This application note uses Spartan®-6 FPGAs to generate spread-spectrum clocks using the DCM_CLKGEN primitive.

Introduction

Using Spread-Spectrum Clocking to Reduce EMI

Manufacturers of electronic devices must ensure that their products do not interfere electrically with nearby devices. For example, the clarity of a phone call should not degrade when it is next to a video display. EMC regulates the noise that causes these disturbances, including electromagnetic interference (EMI). EMC regulations can vary depending on where the product is used, but the typical solutions involve adding expensive shielding, ferrite beads, or chokes. These solutions could further impact the overall cost of the final product by requiring more complicated PCB routing and lengthening the product development cycle.

EMC regulations test a completed end-product to measure the field strength, which is the amount of EMI noise a product emits (Table 1). An antenna measures the field strength across a range of frequencies and distances. To pass, the end-product's emissions are measured across multiple frequencies and varying distances depending on the standards body regulating the emissions.

Table 1: **Acceptable EMC Levels**

Frequency (MHz)	FCC Field Strength (dbµV/m) at 10 m	CISPR Field Strength (dbµV/m) at 10 m
88	29.5	30
216	33	30
230	35.6	30
960	35.6	37

Notes:

1. FCC: Federal Communications Commission.
2. CISPR: International Special Committee on Radio Interference.

EMC testing on a single FPGA is not useful information to a system designer since every end-product (system) has different characteristics based on all the components used, how the PCB is designed, and the mechanical enclosures. However, measuring how a clock's energy is spread across multiple frequencies has a direct impact on the emissions of an entire system.

Using a spectrum analyzer, a typical clock signal is measured as a reference point as shown in Figure 1 (1R). After switching the clock signal to a spread-spectrum clock, the energy is reduced by 13 dB. In this example, the frequencies are evenly distributed around the input frequency with a 3.0% center-spread modulation, sometimes referred to as ±1.5% spread.

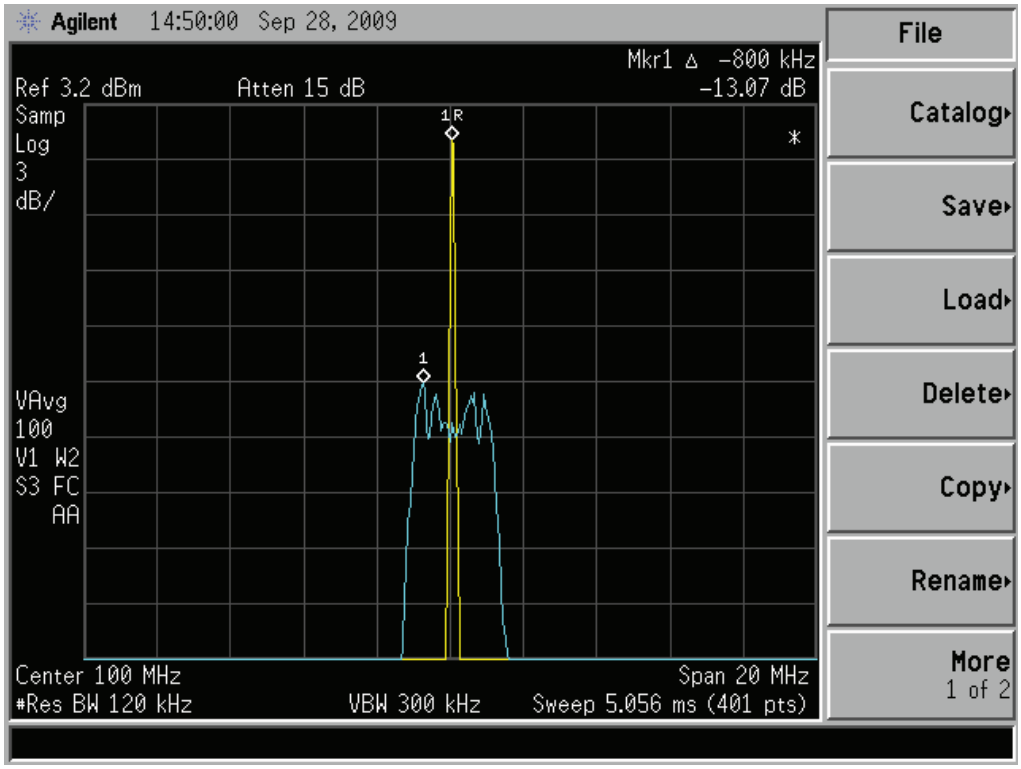

Figure 1: **Center-Spread Modulation as Viewed With a Spectrum Analyzer**

When using center-spread modulation, the clock frequency can be higher than the input frequency. Because these higher clock frequencies are introduced, the system's maximum clock frequency should be adjusted. In the example in Figure 1, the center-spread modulation is 1.5%. Spread spectrum should not be confused with jitter or clock uncertainty. Because spread spectrum uses a relatively low modulation frequency, the only effect of using spread spectrum is the higher frequencies created by the center-spread modulation.

Note: The average frequency of the modulated clock can be different than the input frequency.

Down-spread modulation (similar to the example shown in Figure 1) is used in systems that cannot tolerate frequencies higher than the input frequency. In down-spread modulation, the maximum frequency matches the input frequency. Fully synchronous designs are not affected because the maximum frequency is unchanged.

How the clock frequencies change, or modulate, can also impact how the system is affected by the introduction of the spread-spectrum clock. As long as the clock frequency is slowly changed over a large number of clock cycles, the effect of the spread-spectrum clock does not impact the system. If the clock frequencies are changed too slowly (20–30 KHz), then the positive benefits of spread-spectrum clocking are diminished. When clock frequencies change too quickly (> 120 KHz), the spread-spectrum clock can appear as jitter or not properly lock the PLL to the spread-spectrum clock.

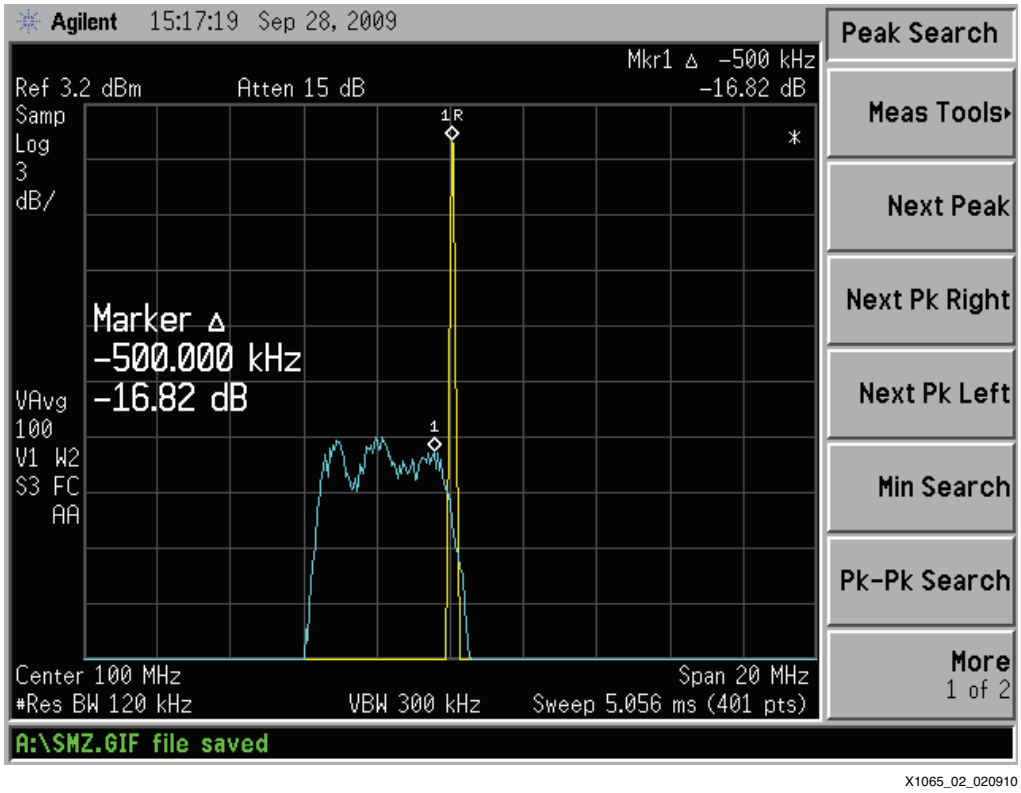

Figure 2: **Down-Spread Modulation as Viewed With a Spectrum Analyzer**

Clock frequencies can be tracked over time to measure the modulation (Figure 3). The modulation frequency is the frequency at which a modulated spread-spectrum clock sweeps through all of the frequencies.

Consider both frequency deviation and modulation frequency when using spread-spectrum clocking in Spartan-6 FPGA designs.

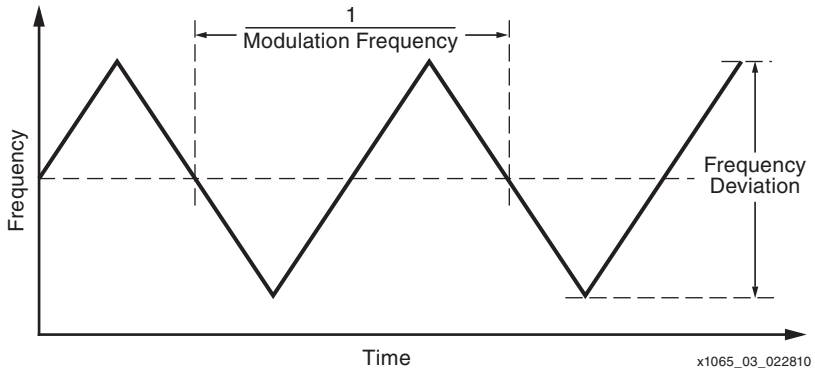

Figure 3: **Triangular Spread-Spectrum Clocking**

Spread-Spectrum Generation

Spartan-6 FPGAs can generate a spread-spectrum clock source from a standard fixed-frequency oscillator. A Spartan-6 FPGA spread-spectrum clock source is generated by using the DCM_CLKGEN primitive. The DCM_CLKGEN primitive can either use a fixed spread-spectrum solution, providing the simplest implementation, or a soft spread-spectrum solution that adds flexibility but requires additional control logic to generate the spread-spectrum clock.

As detailed in Table 2, the fixed spread-spectrum solution is for typical spread-spectrum clock requirements. It only requires setting the SPREAD_SPECTRUM attribute. The soft spread-spectrum solution provides additional flexibility, but requires an additional state machine to control the DCM_CLKGEN primitive and is focused on video applications (M = 7, D = 2).The attributes used in conjunction with the soft spread-spectrum solution are VIDEO_LINK_M0, VIDEO_LINK_M1, or VIDEO_LINK_M2.

Table 2: **Summary of DCM_CLKGEN Spread-Spectrum Modes**

	Fixed Spread-Spectrum Clock	Soft Spread-Spectrum Clock
SPREAD_SPECTRUM Values	CENTER_LOW_SPREAD	VIDEO_LINK_M0
	CENTER_HIGH_SPREAD	VIDEO_LINK_M1
		VIDEO_LINK_M2
Additional Logic	None	Use `sstop.v`
Modulation Profile	Triangular	Triangular
Spread Direction	Center	Down
F_{MOD} (Modulation Frequency)	F_{IN}/1024	See Figure 8
Spread of CLKFX Clock Periods (Frequency Deviation)	CENTER_LOW_SPREAD: 100 ps/CLKFX_DIVIDE CENTER_HIGH_SPREAD: 240 ps/CLKFX_DIVIDE	See Figure 11
CLKFX_MULTIPLY	2–32	7
CLKFX_DIVIDE	1–4	2, 4
DCM_CLKGEN Programming Ports	N/A	PROGCLK, PROGEN, PROGDATA, PROGDONE

Fixed Spread Spectrum

The simplest way to implement spread-spectrum clocking is to use one of the fixed modes of the DCM_CLKGEN primitive. When SPREAD_SPECTRUM is set to either CENTER_LOW_SPREAD or CENTER_HIGH_SPREAD, DCM_CLKGEN automatically creates a spread-spectrum clock. Using the fixed mode, the DCM_CLKGEN primitive internally contains all of the circuitry to create a triangular center-spread modulation. Figure 4 shows a typical video implementation using the fixed mode. An additional PLL is required to multiply the clock frequency to the full SDR clock rates required for OSERDES2.

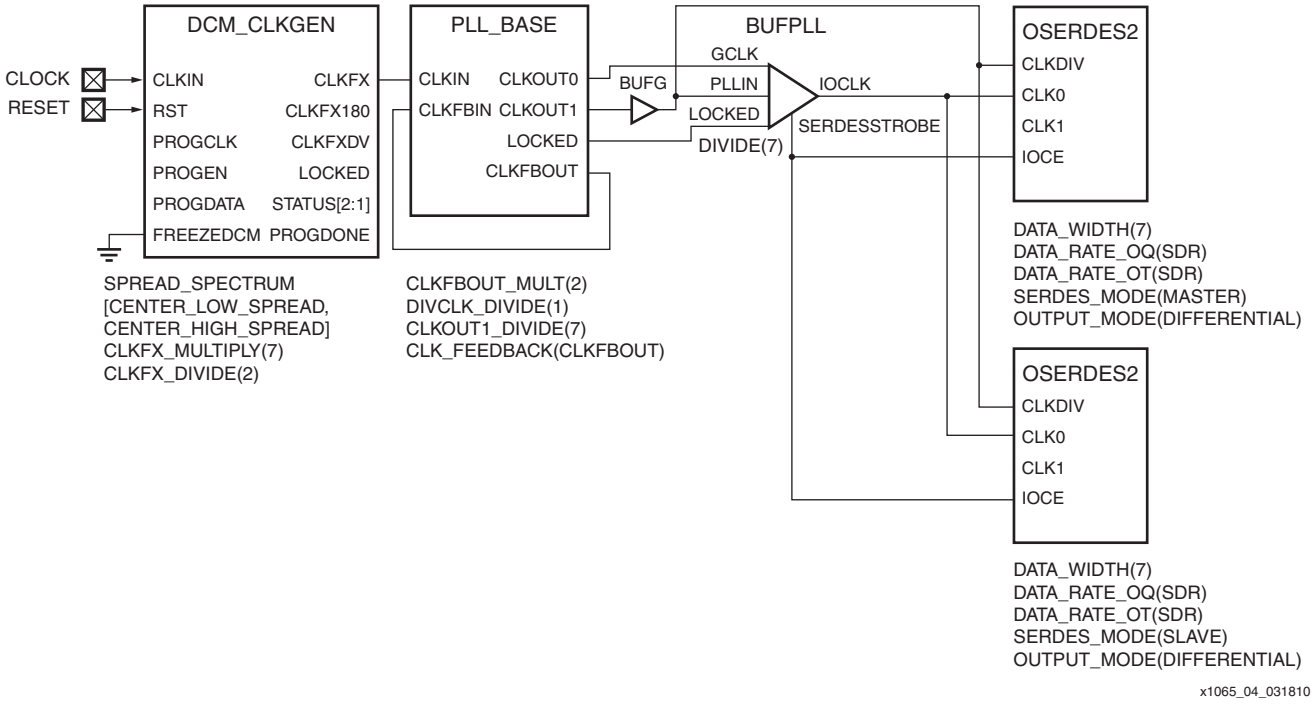

Figure 4: **Example of Center-Spread Modulation using Fixed DCM_CLKGEN Spread-Spectrum Clocking**

The amount of spread is dependent on the final configuration of the design. Figure 5 shows the spread for a typical video application where the DCM_CLKGEN multiplies the clock frequency by 3.5x (CLKFX_MULTIPLY = 7, CLKFX_DIVIDE = 2). Depending on the amount of spread required for the application, the designer can select either CENTER_LOW_SPREAD or CENTER_HIGH_SPREAD.

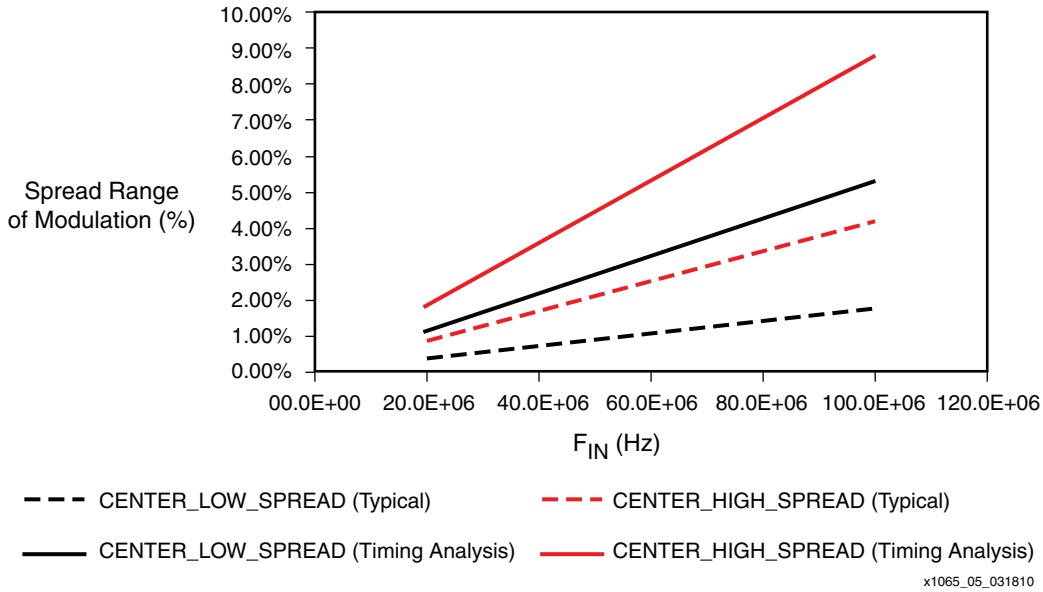

Figure 5: **Example Spread for Fixed Spread-Spectrum Clocking in Video Applications**

When using the fixed modes, the modulation frequency is the same for both CENTER_LOW_SPREAD and CENTER_HIGH_SPREAD as shown in Figure 6.

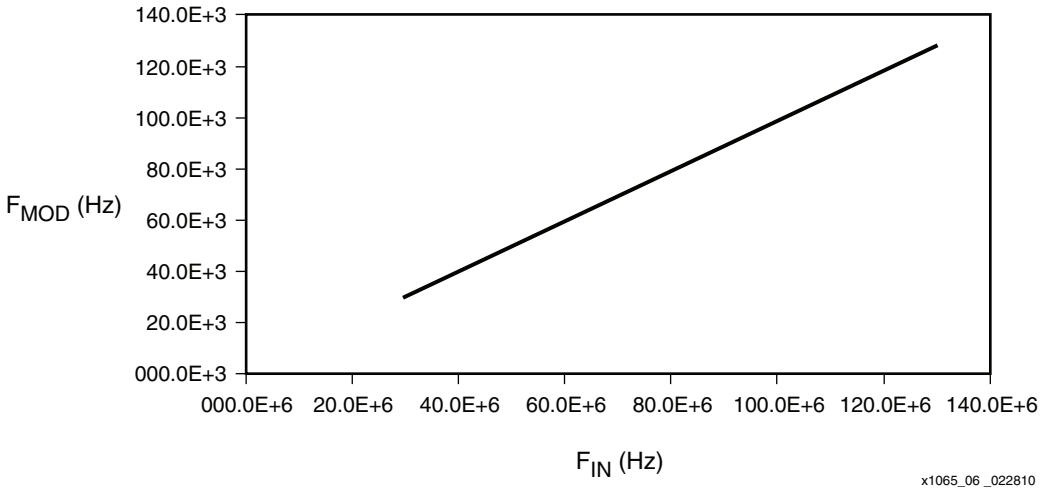

Figure 6: **Modulation Frequency For Fixed Spread Spectrum**

As shown in Equation 1, the modulation frequency depends on the input frequency.

$$F_{MOD} = \frac{F_{IN}}{1024} \qquad \textit{Equation 1}$$

Because center-spread modulation increases the highest frequency of the clock, adjust the timing constraints to match. As an example, to constrain a design using a 100 MHz input clock with a 4% center-spread modulation (±2%) by setting the input frequency to 102 MHz.

Using the DCM_CLKGEN primitive, the designer can specify the maximum amount of spread using CLKFX_MD_MAX. CLKFX_MD_MAX takes an integer value.

$$CLKFX_{MAX\_SPREAD\_SPECTRUM} = \left(1 + \frac{SPREAD}{2}\right) \times \frac{M}{D} \qquad \text{Equation 2}$$

Where:

M = CLKFX_MULTIPLY

D = CLKFX_DIVIDE

SPREAD = Spread using CENTER_LOW_SPREAD/ CENTER_HIGH_SPREAD [%]

Using the same example of a 4% center-spread (±2% spread) and Equation 2, and assuming that CLKFX_MULTIPLY = 7 and CLKFX_DIVIDE = 2, then:

CLKFX_MD_MAX = (1.02 x 7) / 2 = 3.57

Soft Spread Spectrum

Soft spread-spectrum clocking uses a different mode of the DCM_CLKGEN primitive. Soft spread-spectrum clocking generates a triangular down-spread modulation by a continual process of reprogramming DCM_CLKGEN. The DCM_CLKGEN primitive must be set to:

```
SPREAD_SPECTRUM = <VIDEO_LINK_M0, VIDEO_LINK_M1, or VIDEO_LINK_M2>
```

Soft spread-spectrum clocking only works when using the programming interface. As a result, not using PROGCLK and PROGDATA will generate an error.

Note: In Spartan-6 devices, only the top eight BUFGMUX clock buffers can drive PROGCLK.

By focusing on down-spread modulation, the soft spread-spectrum solution does not affect overall timing analysis for the system. Because down-spread modulation slows down the average frequency, the design must ensure data is not lost when switching between the input clock domain and the spread-spectrum clock domain.

Reference Design For Soft Spread-Spectrum Clocking

The Reference Design file contains the following files:

`xapp_ss.v`
Reference design combining PLL to generate OSERDES outputs targeting M/D = 7/2

- `dcm_clkgen_softspread.v`
  Wrapper to be used in place of DCM_CLKGEN instance targeting M/D = 7/2
  - `dcm_clkgen`
    DCM_CLKGEN instantiation
  - `sstop.v`
    State machine used to control spread
    - `sscontrol`
      Programming state machine

`xapp_ss_m7d4.v`
Reference design combining PLL to generate OSERDES outputs targeting M/D = 7/4

- `dcm_clkgen_softspread_m7d4.v`
  Wrapper to be used in place of DCM_CLKGEN instance M/D = 7/4

The reference design also contains a typical video application using soft spread-spectrum clocking to generate the clock frequencies for OSERDES2. As shown in Figure 7, DCM_CLKGEN_SOFTSPREAD replaces the DCM_CLKGEN primitive. Within DCM_CLKGEN_SOFTSPREAD, DCM_CLKGEN is connected to the state machine used for controlling the generation of spread spectrum (SSTOP).

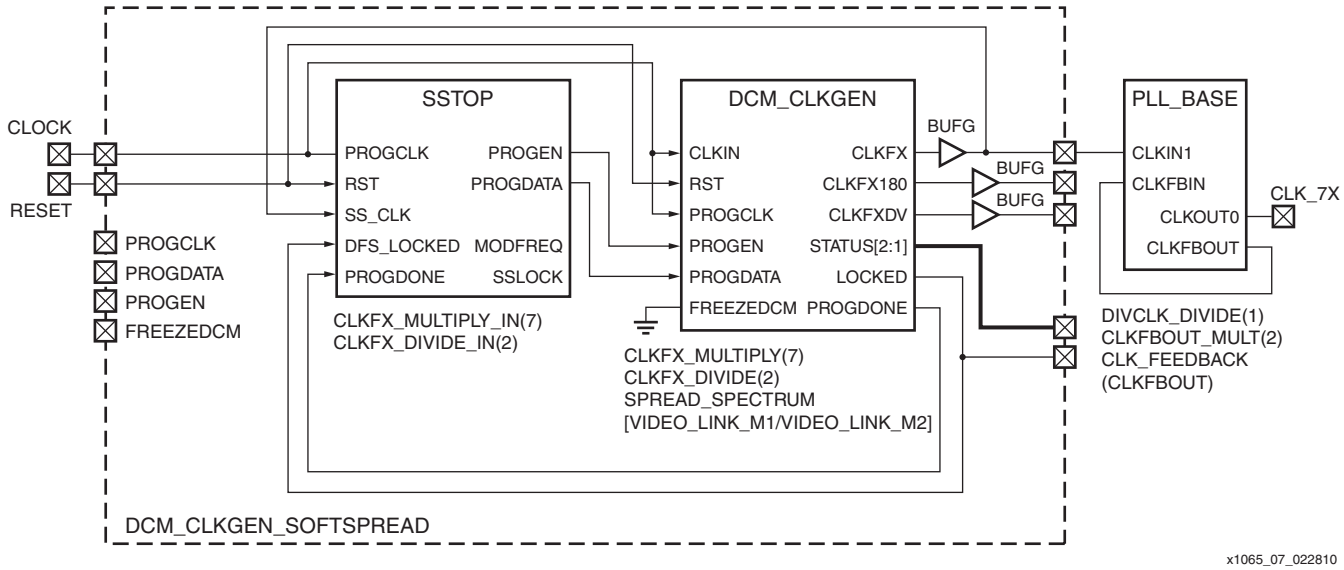

Figure 7: **Reference Design Clock Structure**

To create the modulation, the reference design programs different CLKFX frequencies using a small state machine contained within SSTOP. When the modulation is at the lowest frequency within the expected modulation, CLKFX is increased. Similarly, when the modulation is at the peak of the modulation range, CLKFX is decreased.

As an example, if the input frequency is 85 MHz with the DCM_CLKGEN primitive programmed using CLKFX_MULTIPLY = 7 and CLKFX_DIVIDE = 2, then CLKFX is set to 297.5 MHz. To speed up, CLKFX is reprogrammed to CLKFX_MULTIPLY = 7 and CLKFX_DIVIDE = 1. An 85 MHz input clock implies a CLKFX frequency of 595 MHz. However, this frequency can never actually be reached. Because the frequencies are slowly changed, SSTOP is able to monitor the frequencies. As the modulation reaches the fastest frequency within the expected range, SSTOP sets the direction of the modulation by programming a slower CLKFX frequency. To slow down the modulated frequency, the DCM_CLKGEN primitive is reprogrammed to CLKFX_MULTIPLY = 7 and CLKFX_DIVIDE = 3. Again, SSTOP reprograms CLKFX when the frequency has slowed down to the frequency defined by SSTOP. Because the reference design is targeting video applications, both M and D must be taken into consideration. The reference design uses M = 7 and D = 2.

To control how quickly the DCM_CLKGEN primitive frequencies change after being programmed, place the SPREAD_SPECTRUM attribute into one of three modes for soft spread-spectrum clocking: VIDEO_LINK_M0, VIDEO_LINK_M1, or VIDEO_LINK_M2. When the SPREAD_SPECTRUM attribute is set to VIDEO_LINK_M0, CLKFX changes the frequencies at the fastest rate. Modulation frequencies are the fastest when using VIDEO_LINK_M0 as shown in Figure 8.

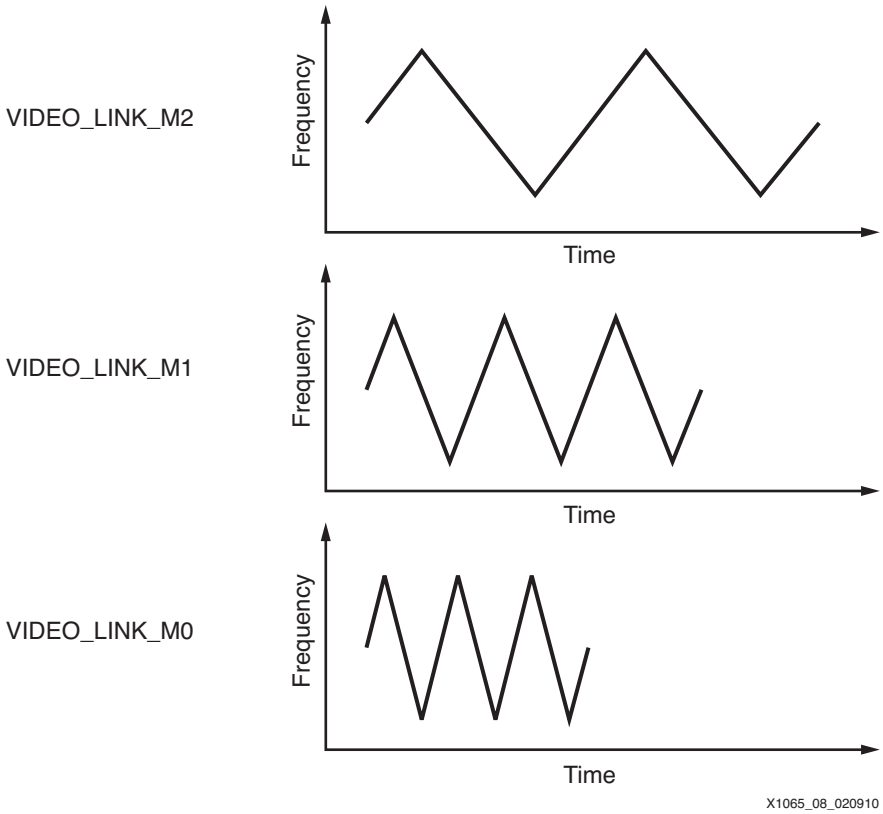

Figure 8: **SPREAD_SPECTRUM Effect on Modulation Frequency**

Example: LVDS for Video Clock Frequencies of 30 To 105 MHz

Focusing on a typical LVDS implementation commonly found in video designs, the SOFT_SS state machine is contained in the reference design. The reference design supports typical video applications requiring 7:1 serialization using the DCM_CLKGEN primitive with a cascaded PLL (Figure 9). The DCM_CLKGEN primitive generates the spread-spectrum modulation. The cascaded PLL provides the high-speed clock required for the BUFPLL as well as providing additional filtering to reduce the cycle-to-cycle jitter.

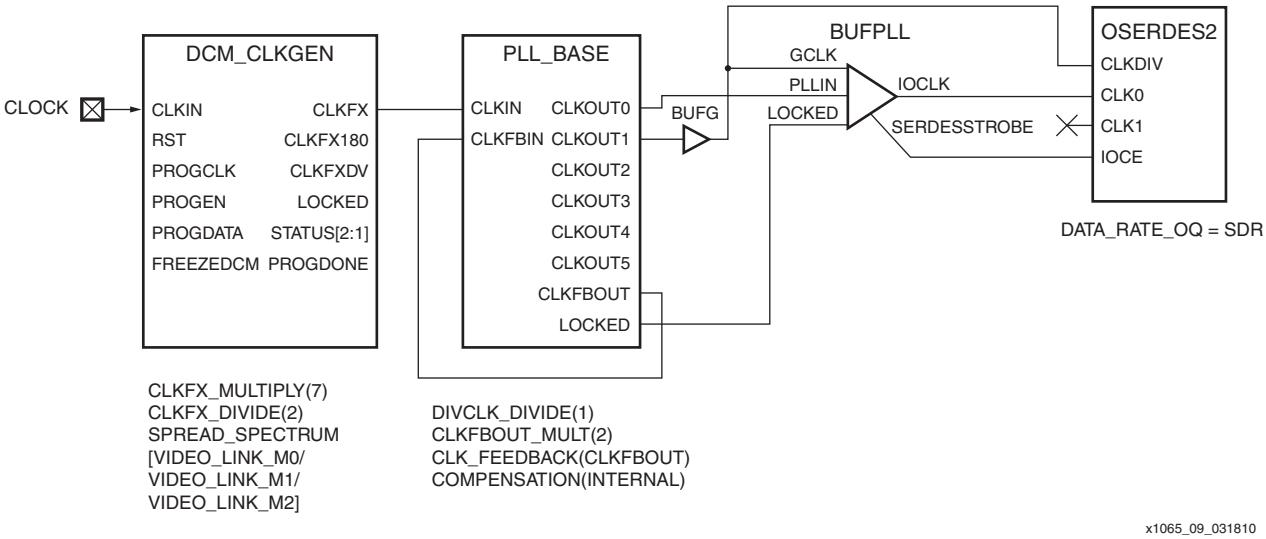

Figure 9: **Soft Spread-Spectrum Clocking Video Implementation (30 < F_{IN} < 105 MHz) Support of OSERDES2**

Example: LVDS for Video Clock Frequencies Beyond 105 MHz

As long as the pixel clock frequencies can be supported, the solution shown in Figure 9 works. However, it is limited by the maximum frequency for CLKFX and CLKOUT_FREQ_FX. As the pixel clock frequencies increase beyond 105 MHz, a higher divide setting must be used (shown in Figure 10). To compensate for the higher divide settings, the following changes are made in `xapp_ss_m7d4.v` and `dcm_clkgen_softspread_m7d4.v`.

- Set **.CLKFX_DIVIDE(4)** for DCM_CLKGEN in `dcm_clkgen_softspread_m7d4.v`
- Set **.SPREADSPECTRUM VIDEO_LINK_M1** for DCM_CLKGEN in `dcm_clkgen_softspread_m7d4.v`
- Set **.CLKFX_DIVIDE_IN(4)** for SSTOP in `dcm_clkgen_softspread_m7d4.v`
- Set **.CLKFBOUT_MULT(4)** for PLL_BASE in `xapp_ss_m7d4.v`

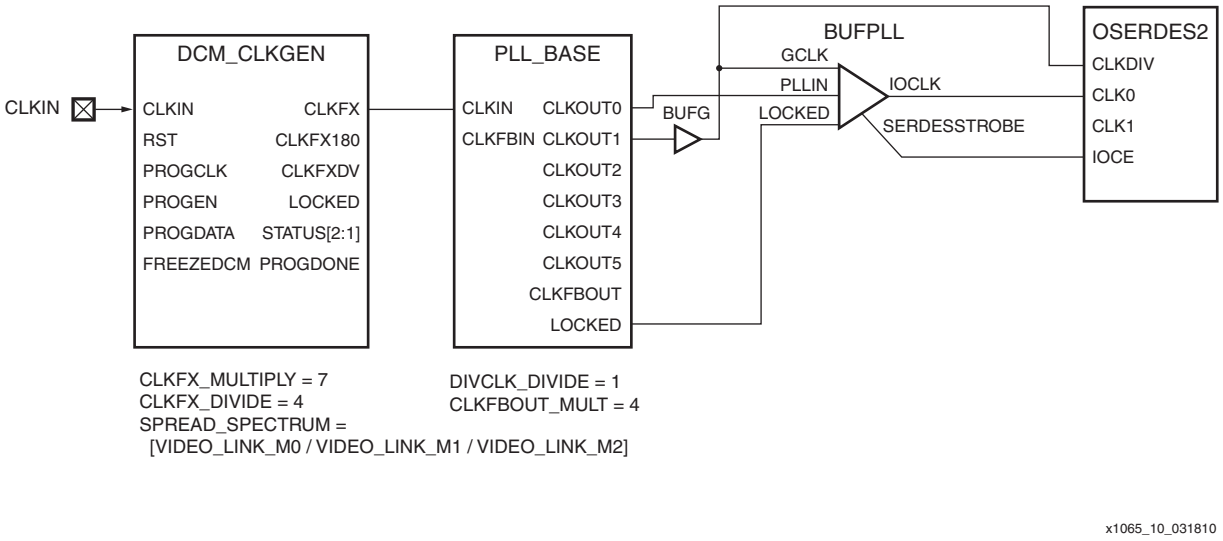

Figure 10: **Soft Spread-Spectrum Clocking Video Implementation (F_{IN} > 105 MHz) Support of OSERDES2**

Within `dcm_clkgen_softspread_m7d4.v`, after changing CLKFX_DIVIDE to four for DCM_CLKGEN, the modulation rates slow down. To compensate, the SPREAD_SPECTRUM attribute must be set to VIDEO_LINK_M1 to speed up the modulation rate back into an acceptable range.

Another change to `dcm_clkgen_softspread_m7d4.v` requires a change to the state machine. It can be adjusted by changing the parameter CLKFX_DIVIDE_IN = 4 within the instantiation for SSTOP. By setting CLKFX_DIVIDE_IN in the instantiation, the state machines within `sstop.v` and `sscontrol.v` adjust to target the new target frequencies. The reference design uses CLKFX_MULTIPLY = 7 and CLKFX_DIVIDE = 5 when speeding up the modulation, and CLKFX_MULTIPLY = 7 and CLKFX_DIVIDE = 3 when slowing it down.

In `xapp_ss_m7d4.v`, because DCM_CLKGEN is sending a 183.75 MHz clock frequency, the original PLL_BASE settings for DIVCLK_DIVIDE and CLKFBOUT_MULT must be increased to meet the minimum acceptable VCO range, F_{VCOMIN}. Set CLKFBOUT_MULT to 4 while DIVCLK_DIVIDE remains unchanged.

Spread for Soft Spread-Spectrum Clocking

The soft spread-spectrum state machine controls the range to be used based on a set of registers. The reference design is setup to create a down-spread spread-spectrum modulation. `sstop.v` controls the size of the modulation through a series of counters. The size of the modulation is controlled by the variables, SLOW and FAST. The reference design sets FAST = 7 and SLOW = 5 to create the down-spread modulation. To further increase the spread, SLOW can be decreased (down to 0). As the spread increases, the modulation frequency reduces. Check the spread and modulation frequency for values other than FAST = 7 and SLOW = 5.

Figure 11 shows the reference design's spread as the input frequencies are varied. However, by ignoring modulation frequencies, Figure 9 shows how spread varies across pixel clock frequencies.

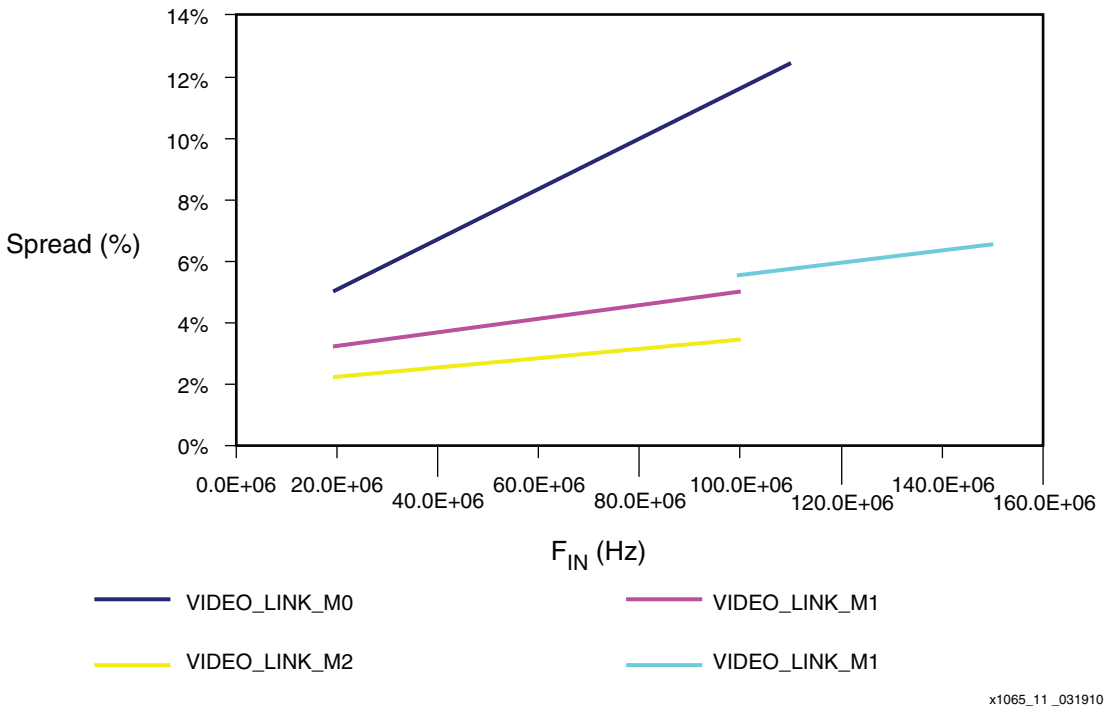

Figure 11: **Reference Design Spread Across Video Frequencies**

Modulation Frequencies for the SOFT_SS State Machine

When working with a spread-spectrum clock source, the modulation frequency also influences how the system responds to the modulation of the spread-spectrum clock.

To effectively reduce the emitted noise using a spread-spectrum solution, keep the modulation frequency higher than 20 KHz. If the modulation frequency is too low, the spread-spectrum modulation is too slow to efficiently reduce EMI.

When the modulation is too high and with a large spread, spread spectrum modulation could appear as system noise or could start affecting the ability of other PLL components to follow the modulation. The modulation frequency should be less than 120 kHz.

When using the SOFT_SS state machine for video applications, a number of factors affect the modulation frequency. The SPREAD_SPECTRUM modulation is not proportional to the input frequencies and requires more analysis to estimate the modulation frequencies. When using the SOFT_SS state machine, modulation frequency becomes dependent on input frequency, CLKFX_MULTIPLY, CLKFX_DIVIDE, and the SPREAD_SPECTRUM settings (VIDEO_LINK_M0, VIDEO_LINK_M1, VIDEO_LINK_M2). To simplify the potential variables,

use Equation 3 to approximate the modulation frequency (F_{MOD}) when using the reference designs.

$$F_{MOD} = F_{IN2} \times CONST \quad \text{Equation 3}$$

The modulation settings are defined using the equations from Table 3 and bounding the modulation frequencies between 20 to 120 KHz. For low frequencies, the VIDEO_LINK_M0 speeds up the modulation to keep the modulation frequency above the audio range frequencies. As the input frequency increases, VIDEO_LINK_M1 and VIDEO_LINK_M2 are used to slow down the modulation rates down below 120 KHz.

Table 3: **Modulation Frequency Calculation**

	CLKFX_MULTIPLY	CLKFX_DIVIDE	CONST
VIDEO_LINK_M0	7	2	40.0×10^{-12}
VIDEO_LINK_M1	7	2	15.0×10^{-12}
VIDEO_LINK_M2	7	2	5.0×10^{-12}
VIDEO_LINK_M3	7	4	2.5×10^{-12}

As the input frequency continues to increase to 105 MHz, DCM_CLKGEN can no longer support M = 7 and D = 2. As described in the Example: LVDS for Video Clock Frequencies Beyond 105 MHz section, the reference design can be adjusted to support CLKFX_MULTIPLY(7) and CLKFX_DIVIDE(4). The increase in CLKFX_DIVIDE causes the modulation frequency to slow down. To compensate, use VIDEO_LINK_M1. The result is the $M1\_74F_{MOD}$ in Figure 12.

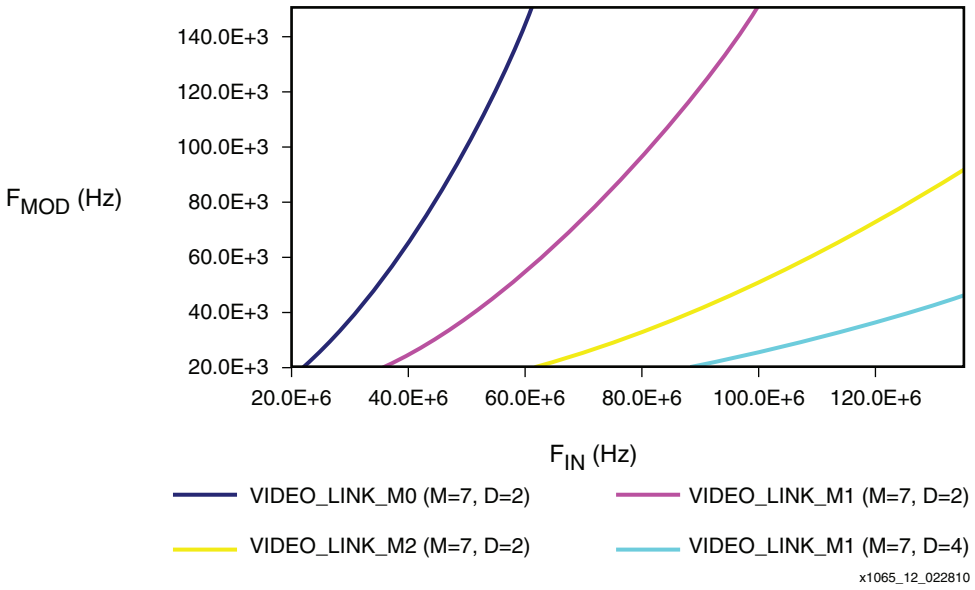

Figure 12: **Modulation Frequency vs F_{IN}**

Spread Spectrum Receiving

The PLL_BASE or DCM_SP primitives can be used for designs where a spread-spectrum clock is externally created.

Because the DCM_CLKGEN primitive is designed to filter out input clock noise and will distort received spread spectrum clocks, do not use the DCM_CLKGEN primitive with SPREAD_SPECTRUM = NONE for spread-spectrum clock receiving.

When using the DCM_SP primitive to receive spread-spectrum clocks, it is important to understand how DCM_SP will follow the modulation. To maintain phase and frequency

alignment, the DCM must follow the spread-spectrum clock source as shown in Figure 3. To stop distortion on the original spread-spectrum source, DCM_SP must be able to update the deskewing and phase shifting often.

A rough approximation can be made to estimate the limits of the modulation frequency for a given frequency deviation. Beyond these limits, the DCM_SP phase and frequency alignment deteriorates, reducing the receiver's skew margin.

Video display receiver designs require both the DCM's delay-locked loop (DLL) and digital frequency synthesizer (DFS) to be used together. The DLL provides a digital deskew circuit. Equation 4 estimates the maximum modulation frequency (F_{MDLL} expressed in Hz) for a spread-spectrum clock source at any given speed:

$$F_{MDLL} < \frac{DCM\_DELAY\_STEP}{24 \times S \times T_{IN}^{2}} \quad \text{Equation 4}$$

Where:

F_{MDLL} = Maximum modulation frequency in Hz that the DLL can follow

DCM_DELAY_STEP = Finest delay resolution for the DCM(s)

S = Maximum spread or frequency deviation

T_{IN} = Highest effective input clock period(s)

24 = Constant related to the update rate of the DLL frequency and phase adjustments

The DFS provides a flexible range of output frequencies based on the ratio of two user-defined integers, a multiplier (CLKFX_MULTIPLY) and a divisor (CLKFX_DIVIDE). Equation 5 estimates the maximum modulation frequency (F_{MDFS}) for the DFS portion:

$$F_{MDFS} < \frac{DCM\_DELAY\_STEP}{2 \times M \times D \times S \times T_{IN}^{2}} \quad \text{Equation 5}$$

Where:

F_{MDFS} = Maximum Modulation Frequency (Hz) DFS can follow

DCM_DELAY_STEP = Finest delay resolution for the DCM(s)

M = CLKFX_MULTIPLY

D = CLKFX_DIVIDE

S = Maximum spread or frequency deviation

T_{IN} = Highest effective input clock period (s)

2 = Constant related to the update rate of DFS frequency and phase adjustments

In video applications that use both DFS and DLL, the maximum modulation frequency is determined by Equation 5.

The DCM settings in this section were used during reference design testing. The parameters best represent the 7:1 LVDS designs commonly found in displays: CLKFX_MULTIPLY = 7 and CLKFX_DIVIDE = 2. The DCM multiplies the clock to 3.5 times the original frequency. Using the DDR registers provides the full 7x data rate multiplication.

When using spread spectrum clocks, use the fixed phase-shift mode. The variable phase-shift mode disables the internal phase shift controls that update the phase shift with the frequency changes. As a result, variable phase shifting should not be used with spread-spectrum input clocks.

Applying the DCM settings from the 7:1 LVDS applications and using Equation 4 and Equation 5, limits for a 75 MHz spread-spectrum clock can be estimated with a 5% frequency deviation.

$$F_{MDLL} < \frac{23 \times 10^{-12}}{24 \times 0.05 \times (13.3 \times 10^{-9})^2} = 108 \text{ KHz}$$

$$F_{MDFS} = \frac{23 \times 10^{-12}}{2 \times 7 \times 2 \times 0.05 \times (13.3 \times 10^{-9})^2} = 92.9 \text{ KHz}$$

Similarly, other display functions such as PPDS (10:1) or MINI-LVDS (8:1) use other serialization. The ability of the DCM to track the spread-spectrum signal when multiplying is a function of the multiply and divide values shown in Equation 5. For other display functions, such as PPDS (M = 5, D = 1) or MINI-LVDS (M = 4, D = 1), the modulation frequency threshold is higher than this 7x example.

Other Methods of Reducing EMI

In addition to supporting spread-spectrum clocks, Spartan-6 devices further reduce EMI by controlling the I/O type by selecting a specific SelectIO™ interface.

Spartan-6 devices use LVCMOS and LVTTL I/Os with separate slew rate and drive strength attributes. The drive strength can be reduced by lowering DRIVE. The slew rate can be reduced by changed SLEW to SLOW or QUIETIO, which further reduces the ringing.

LCD modules are replacing the noisy LVTTL interfaces with differential interfaces like reduced-swing differential signaling (RSDS), MINI-LVDS, and even point-to-point differential signaling (PPDS). For the lowest EMI, Spartan-6 devices directly drive these differential interfaces.

Reference Design

General information about the reference design is shown in Table 4. The device utilization is shown in Table 5 for both the fixed spread-spectrum and soft spread-spectrum designs.

The reference design files can be downloaded at:

https://secure.xilinx.com/webreg/clickthrough.do?cid=143697

Table 4: **Reference Design Checklist**

Developer Name	Xilinx
Target devices	Spartan-6 FPGAs
Source code provided	Yes
Source code format	Verilog
Design uses code/IP from an existing reference design, application note, 3rd party, Core Generator	No
Simulation	
Functional simulation performed	Requires ISE software v12.1 or later
Timing simulation performed	Requires ISE software v12.1 or later
Testbench used for functional and timing simulations provided	Not applicable
Testbench format	Not applicable
Simulator software used	MXE
SPICE/IBIS simulations	No
Implementation	

Table 4: **Reference Design Checklist *(Cont'd)***

Synthesis software tools used	XST 11.4
Implementation software tools used	ISE v11.4 software
Static timing analysis performed	Yes
Hardware verified	Yes
Hardware platform used for verification	SP601

Table 5: **Device Utilization**

	Fixed Spread-Spectrum Design	Soft Spread-Spectrum Design
DCM_CLKGENs used	1	1
BUFGs used	0	2
Slice LUT Flip-Flop pairs used	0	99

Conclusion

This application note gives examples of a typical spread-spectrum clock for video applications using the Spartan-6 FPGA DCM_CLKGEN primitive. DCM_CLKGEN can be used for fixed spread-spectrum generation without any logic or in a soft spread-spectrum solution using a state machine. While the focus is specifically on LVDS display applications, applications with similar DCM usage can use similar spread-spectrum clocks.

Reference

This document references the following Xilinx documentation:

1. DS162, *Spartan-6 FPGA Data Sheet*
2. UG382, *Spartan-6 FPGA Clocking Resources User Guide*
3. XAPP469, *Spread-Spectrum Clocking Reception for Displays*
   This application note only applies to Spartan-3E and Extended Spartan-3A family devices.

Additional Resources

The following resources provided additional information useful for working with this application note and reference design:

4. http://www.fcc.gov, Federal Communications Commission
5. http://www.ansi.org, American National Standards Institute
6. http://www.ntia.doc.gov/osmhome/international/cispr.html, International Special Committee on Radio Interference, National Telecommunications and Information Administration
7. http://www.national.com/appinfo/fpd/, Flat Panel Displays, National Semiconductor
8. http://www.cclab.com/engnotes/eng290.htm, Engineering Note 290: Comparison of FCC Limits with CISPR Limits, Communication Certification Laboratory

Revision History

The following table shows the revision history for this document.

Date	Version	Description of Revisions
03/22/10	1.0	Initial Xilinx release.

Notice of Disclaimer

Xilinx is disclosing this Application Note to you “AS-IS” with no warranty of any kind. This Application Note is one possible implementation of this feature, application, or standard, and is subject to change without further notice from Xilinx. You are responsible for obtaining any rights you may require in connection with your use or implementation of this Application Note. XILINX MAKES NO REPRESENTATIONS OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL XILINX BE LIABLE FOR ANY LOSS OF DATA, LOST PROFITS, OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES ARISING FROM YOUR USE OF THIS APPLICATION NOTE.

AUTOMOTIVE APPLICATIONS DISCLAIMER

XILINX PRODUCTS ARE NOT DESIGNED OR INTENDED TO BE FAIL-SAFE, OR FOR USE IN ANY APPLICATION REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS APPLICATIONS RELATED TO: (I) THE DEPLOYMENT OF AIRBAGS, (II) CONTROL OF A VEHICLE, UNLESS THERE IS A FAIL-SAFE OR REDUNDANCY FEATURE (WHICH DOES NOT INCLUDE USE OF SOFTWARE IN THE XILINX DEVICE TO IMPLEMENT THE REDUNDANCY) AND A WARNING SIGNAL UPON FAILURE TO THE OPERATOR, OR (III) USES THAT COULD LEAD TO DEATH OR PERSONAL INJURY. CUSTOMER ASSUMES THE SOLE RISK AND LIABILITY OF ANY USE OF XILINX PRODUCTS IN SUCH APPLICATIONS.