XAPP878 (v1.1) June 9, 2010

MMCM Dynamic Reconfiguration

Author: Karl Kurbjun and Carl Ribbing

Summary

This application note provides a method to dynamically change the clock output frequency, phase shift, and duty cycle of the Virtex®-6 FPGA mixed-mode clock manager (MMCM) through its dynamic reconfiguration port (DRP). An explanation of the behavior of the internal DRP control registers is accompanied by a reference design that uses a state machine to drive the DRP, which ensures the registers are controlled in the correct sequence.

While the reference design performs the operations for the user, familiarity with the functional operation of the MMCM is recommended. For more information on MMCM functionality, see UG362, *Virtex-6 FPGA Clocking Resources User Guide.*

The reference design supports two reconfiguration state addresses and can be extended to support additional states. Each state does a full reconfiguration of the MMCM so that most parameters can be changed. The design does not support outputs configured with fractional divider values or reconfiguring with fine phase shifting enabled.

Introduction

The clock management tiles (CMT) in the Virtex-6 devices each contain two MMCMs. One of the most powerful features of the MMCM is its ability to dynamically reconfigure the phase, duty cycle, and divide values of the clock outputs. This application note describes the information necessary to reconfigure the MMCM and provides a reference design that implements all of the algorithms covered.

Reconfiguration is performed through the MMCM DRP. The DRP provides access to the configuration bits that would normally only be initialized in the bitstream. This allows the user to dynamically change the MMCM clock outputs while the design is running. Frequency, phase, and duty cycle can all be changed dynamically. To properly reconfigure the MMCM, it must be initially set up with integer divider values, and fine phase shifting must not be enabled. Fractional dividers and the fine phase shifter are not supported for reconfiguration.

The MMCM Configuration Bit Groups and MMCM DRP Registers sections presents the configuration bits as five bit groups, provides an overview of their usage, and details the configuration bit locations as registers. This information is not necessary to use the DRP reference design; it is intended to give an overview of the internal MMCM attributes that must be changed along with their register locations. Specific information on how the attributes are calculated is provided through the reference design. The reference design functionality and use are explained in the Reference Design and Using the Reference Design sections.

MMCM Configuration Bit Groups

The MMCM has five user-accessible configuration bit groups that allow reconfiguration of individual clock outputs. The five groups are the divider group, the phase group, the lock group, the filter group, and the power group. These configuration bit groups are internal to the MMCM primitive and clarify the operation of the MMCM_DRP module. The user modifiable parameters for the MMCM_DRP module are described in the Reconfiguration Module Ports and Attributes section.

Divider Group

Every clock output has a divider group associated with it. The divider group is composed of the following parameters:

- High Time
- Low Time
- No Count
- Edge

The first two parameters associated with the divider group are the High and Low Time counters. These counters set the number of voltage-controlled oscillator (VCO) clock cycles through which the output clock should stay High or Low. For example, if you set both High and Low Time to 2, the effective divide value is 4 and the duty cycle is 50%.

The No Count parameter disables the High and Low Time counters. This in turn makes the divider output a clock with an effective divide value of 1.

The Edge parameter controls the High to Low transition. It forces the High Time counter to transition on a falling edge at the end of its count. This has the effect of increasing the High Time while decreasing the Low Time. Another way to think of the edge bit is that it adds half a VCO clock cycle to the High Time and subtracts half a clock cycle from the Low Time.

As an example, if a 50/50 duty cycle is desired with a divide value of 3, the Edge bit would be set. The High Time counter would be set to one and the Low Time counter would be set to 2. With the edge bit set, the net count for the High and Low times would be 1.5 clock cycles each.

Phase Group

Each clock output except the DIVCLK has a phase group associated with it. This group is composed of the following set of parameters:

- Phase Mux
- Delay Time
- MX

The Phase Mux selects a coarse phase from the VCO for a clock output with a resolution of 45° (360°/8) relative to the VCO clock period.

Delay Time is a counter that counts the number of VCO clock cycles to delay the output. This means that there is a direct correlation between the possible phase shift for the clock output and the divide value for that particular output. As the divide value increases, finer phase shift steps are available. The Delay Time counter allows for a phase offset of up to 64 VCO clock cycles.

MX must be set to `2'b00` during reconfiguration, regardless of the previous value. This parameter ensures the desired phase is output as expected.

Lock Group

This group cannot be calculated with an algorithm and is based on lookup tables created from device characterization. The appropriate lock bit settings are dependent on the feedback divider setting. This divider is set with the CLKFBOUT_MULT attribute when instantiating the MMCM_DRP module. The lock group has an effect on the MMCM's ability to detect that it is locked. The lookup table is located in the reference design within `mmcm_drp_func.h`.

Filter Group

This group cannot be calculated and is based on lookup tables created from device characterization. There are effectively two tables, one for each bandwidth setting. The feedback divider setting (CLKFBOUT_MULT) acts as the index to the chosen table. There are three bandwidth settings allowable in the tools (High, Low, and Optimized), but in effect there are only two. High and Optimized use the same table, while the Low bandwidth setting uses a separate table. The filter group has an effect on the phase skew and the jitter filtering capability of the MMCM. The lookup table is located in the reference design within `mmcm_drp_func.h`.

Power Group

This group allows the dynamic reconfiguration operations to properly function. The bits associated with this group must be all enabled when performing reconfiguration.

MMCM DRP Registers

Seventeen configuration registers share the divide and phase bit groups. Sixteen of them are identical in their bit layout and are associated with clock outputs 0 through 6 and CLKFBOUT. One output register is associated with the DIVCLK output, which is along the input path to the MMCM. The DIVCLK output does not have any phase configuration bits associated with it. Figure 1 shows the seven clock outputs, the feedback clock output, and the DIVCLK (D in the diagram).

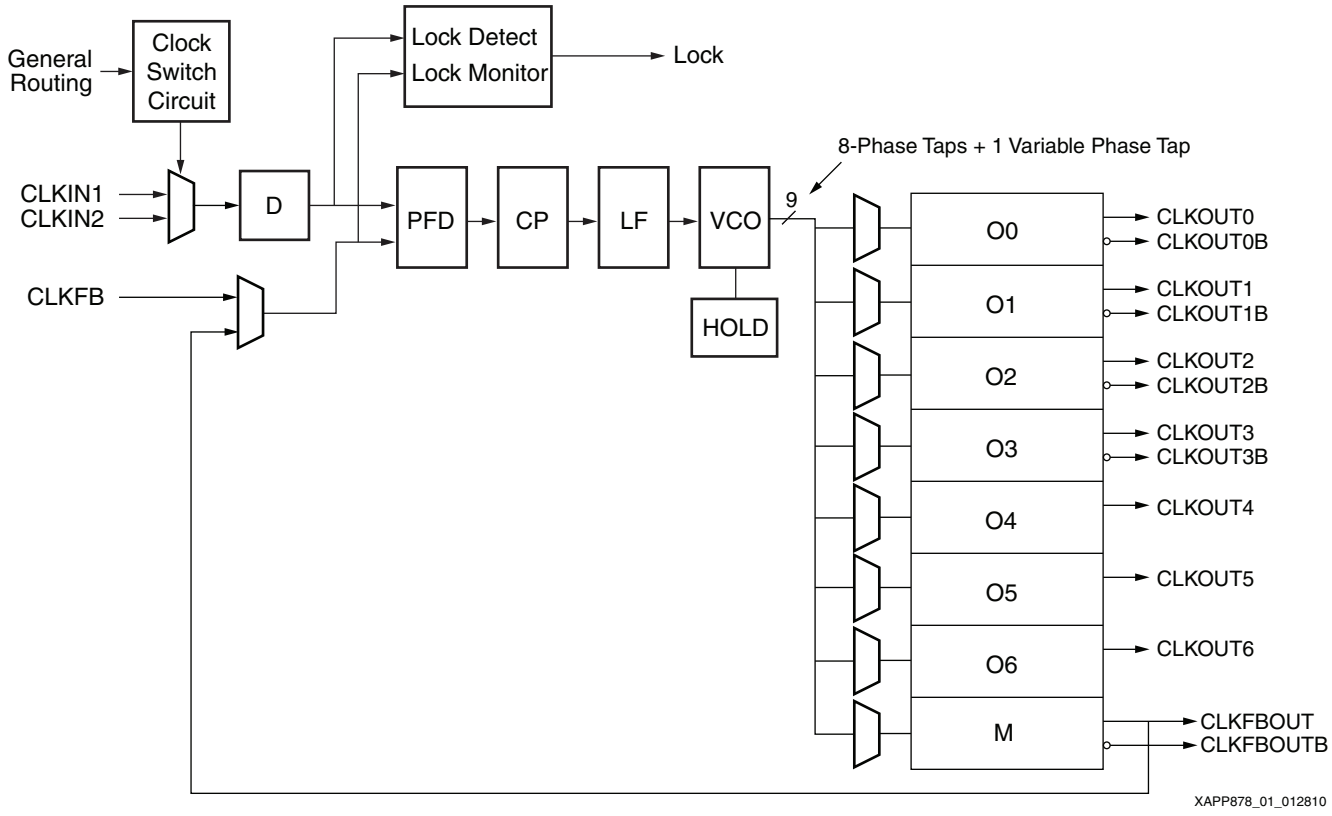

Figure 1: **MMCM Block Diagram**

Each clock output has two registers that share the phase and divide attributes. The sixteen registers that have the same layout are divided into two registers. The register layouts are shown in Table 1 through Table 6.

Table 1: **MMCM ClkReg1 Bitmap**

ClkReg1	Bit	Description
PHASE MUX	[15:13]	Chooses an initial phase offset for the clock output, the resolution is equal to 1/8 VCO period
RESERVED	[12]	Retain the previous value stored here
HIGH TIME	[11:6]	Sets the amount of time in VCO cycles that the clock output remains High
LOW TIME	[5:0]	Sets the amount of time in VCO cycles that the clock output remains Low

Table 2: **MMCM ClkReg2 Bitmap**

ClkReg2	Bit	Description
RESERVED	[15:10]	Retain the previous value stored here
MX	[9:8]	Must be set to `2'b00`
EDGE	[7]	Chooses the edge that the High Time counter transitions on
NO COUNT	[6]	Bypasses the High and Low Time counters
DELAY TIME	[5:0]	Phase offset with a resolution equal to the VCO period

Table 3: **MMCM DivReg Bitmap**

DivReg	Bit	Description
RESERVED	[15:14]	Retain the previous value stored here
EDGE	[13]	Chooses the edge that the High Time counter transitions on
NO COUNT	[12]	Bypasses the High and Low Time counters
HIGH TIME	[11:6]	Sets the amount of time in VCO cycles that the clock output remains High
LOW TIME	[5:0]	Sets the amount of time in VCO cycles that the clock output remains Low

Table 4: **MMCM LockReg1 Bitmap**

LockReg1	Bit	Description
RESERVED	[15:10]	Retain the previous value stored here
LKTABLE[29:20]	[9:0]	These bits are pulled from the lookup table provided in the reference design

Table 5: **MMCM LockReg2 Bitmap**

LockReg2	Bit	Description
RESERVED	[15]	Retain the previous value stored here
LKTABLE[34:30]	[14:10]	These bits are pulled from the lookup table provided in the reference design
LKTABLE[9:0]	[9:0]	These bits are pulled from the lookup table provided in the reference design

Table 6: **MMCM LockReg3 Bitmap**

LockReg3	Bit	Description
RESERVED	[15]	Retain the previous value stored here
LKTABLE[39:35]	[14:10]	These bits are pulled from the lookup table provided in the reference design
LKTABLE[19:10]	[9:0]	These bits are pulled from the lookup table provided in the reference design

The filter group is composed of 10 bits that are stored on two registers. The register layouts are shown in Table 7 and Table 8.

Table 7: **MMCM FiltReg1 Bitmap**

FiltReg1	Bit	Description
TABLE[9]	[15]	This bit is pulled from the lookup table provided in the reference design
RESERVED	[14:13]	Retain the previous value stored here
TABLE[8:7]	[12:11]	These bits are pulled from the lookup table provided in the reference design
RESERVED	[10:9]	Retain the previous value stored here
TABLE[6]	[8]	This bit is pulled from the lookup table provided in the reference design
RESERVED	[7:0]	Retain the previous value stored here

Table 8: **FiltReg2 Bitmap**

FiltReg2	Bit	Description
TABLE[5]	[15]	This bit is pulled from the lookup table provided in the reference design
RESERVED	[14:13]	Retain the previous value stored here
TABLE[4:3]	[12:11]	These bits are pulled from the lookup table provided in the reference design
RESERVED	[10:9]	Retain the previous value stored here
TABLE[2:1]	[8:7]	These bits are pulled from the lookup table provided in the reference design
RESERVED	[6:5]	Retain the previous value stored here
TABLE[0]	[4]	This bit is pulled from the lookup table provided in the reference design
RESERVED	[3:0]	Retain the previous value stored here

The power bits are stored in one register whose layout is shown in Table 9.

Table 9: **MMCM PowerReg Bitmap**

PowerReg	Bit	Description
POWER	[15:0]	These bits must all be set High when performing DRP

The Virtex-6 FPGA MMCM register addresses are shown in Table 10.

Table 10: **MMCM Address Map**

Address	Layout	Description
0x06	ClkReg1	CLKOUT5 Register 1
0x07	ClkReg2	CLKOUT5 Register 2
0x08	ClkReg1	CLKOUT0 Register 1
0x09	ClkReg2	CLKOUT0 Register 2
0x0A	ClkReg1	CLKOUT1 Register 1
0x0B	ClkReg2	CLKOUT1 Register 2
0x0C	ClkReg1	CLKOUT2 Register 1
0x0D	ClkReg2	CLKOUT2 Register 2
0x0E	ClkReg1	CLKOUT3 Register 1
0x0F	ClkReg2	CLKOUT3 Register 2
0x10	ClkReg1	CLKOUT4 Register 1
0x11	ClkReg2	CLKOUT4 Register 2
0x12	ClkReg1	CLKOUT6 Register 1
0x13	ClkReg2	CLKOUT6 Register 2
0x14	ClkReg1	CLKFBOUT Register 1
0x15	ClkReg2	CLKFBOUT Register 2
0x16	DivReg	DIVLCK Register
0x18	LockReg1	Lock Register 1
0x19	LockReg2	Lock Register 2
0x1A	LockReg3	Lock Register 3
0x28	PowerReg	Power Register
0x4E	FiltReg1	Filter Register 1
0x4F	FiltReg2	Filter Register 2

Reference Design

The reference design files include a Verilog MMCM reconfiguration module. This module uses only 24 total slices, comprising the reconfiguration logic and state memory.

The reference design drives the DRP port with a state machine that addresses the MMCM, reads the previous value, masks the bits that need to be changed, sets the new value, and finally writes the value to the MMCM DRP port. The addresses, masks, and new values are stored in a pre-initialized ROM that is filled during elaboration in synthesis. The ROM initialization is done with constant functions provided with the reference design.

Figure 2 is a block diagram of the reconfiguration module.

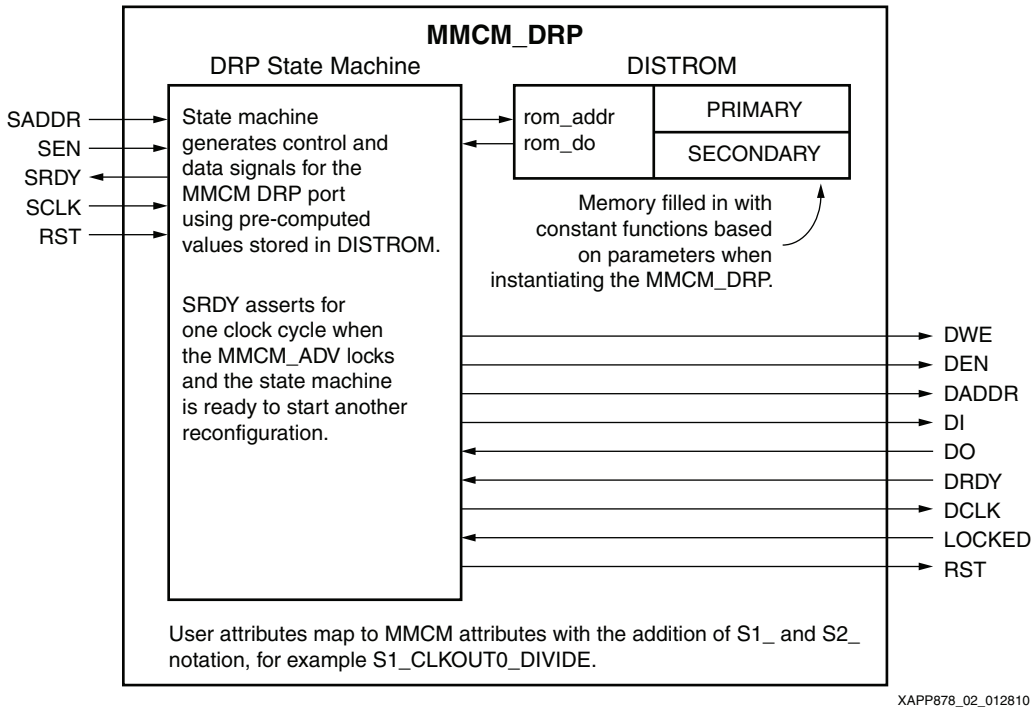

Figure 2: **MMCM_DRP Internal Block Diagram**

The `mmcm_drp.v` module contains the state machine and ROM, and calls the constant functions which are provided in `mmcm_drp_func.h`.

Figure 3 shows the block diagram of the system with the `mmcm_adv` and the `mmcm_drp` modules attached.

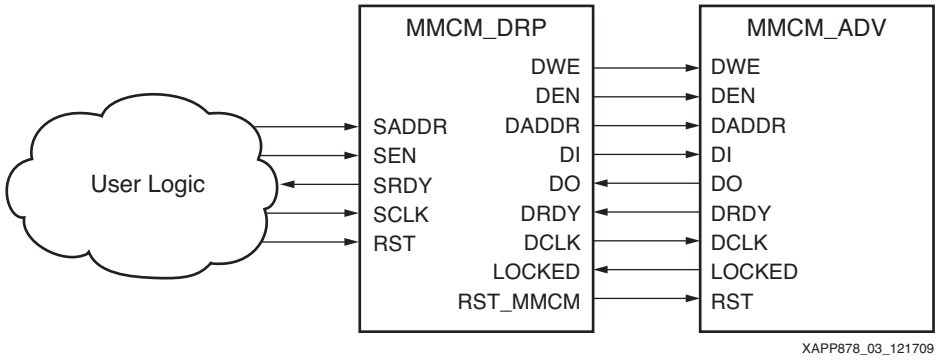

Figure 3: **Reference Design Block Diagram**

DRP State Machine

The DRP state machine is composed of the nine states shown in Table 11. It controls all of the signals from the `mmcm_drp` module.

Table 11: **DRP States**

State	Description	Next State	Transition Condition
RESTART	This state is entered whenever the SRST pin is asserted or if the current_state goes into an undefined state.	WAIT_LOCK	SRST = 0
WAIT_LOCK	This state waits for the lock signal from the MMCM to assert. When lock is asserted, SRDY = 1.	WAIT_SEN	LOCK = 1
WAIT_SEN	This state waits for SEN to be asserted and sets the appropriate ROM address according to SADDR.	ADDRESS	SEN = 1
ADDRESS	This state is entered from either WAIT_SEN or from WAIT_DRDY. The state sets DADDR according to the current value stored in the ROM and ASSERTS DEN.	WAIT_A_DRDY	<always>
WAIT_A_DRDY	This state is always entered from ADDRESS. It waits for the MMCM to assert the DRDY signal.	BITMASK	DRDY = 1
BITMASK	This state is always entered from WAIT_A_DRDY. It performs a bitwise AND on DO from the MMCM with the mask value stored in the ROM.	BITSET	<always>
BITSET	This state is always entered from BITMASK. It performs a bitwise OR with the bitset stored in the ROM and the output from the BITMASK operation.	WRITE	<always>
WRITE	This state asserts DEN, DWE, and RST_MMCM. It updates the state counter which is used to keep track of the number of register writes are needed to perform one full reconfiguration.	WAIT_DRDY	<always>
WAIT_DRDY	This state waits for DRDY to assert from the MMCM.	ADDRESS (state_count > 0) WAIT_LOCK (state_count ≤ 0)	DRDY = 1

In short, the operations that must be implemented to reconfigure one value in the MMCM are:

- Assert RST to the MMCM (do not de-assert)
- Set DADDR on the MMCM and assert DEN for one clock cycle
- Wait for the DRDY signal to assert from the MMCM
- Perform a bitwise AND between the DO port and the MASK (DI = DO and MASK)
- Perform a bitwise OR between the DI signal and the BITSET (DI = DI | BITSET)
- Assert DEN and DWE on the MMCM for one clock cycle
- Wait for the DRDY signal to assert from the MMCM
- De-assert RST to the MMCM
- Wait for MMCM to lock

Reconfiguration Module Ports and Attributes

The reconfiguration module is composed of the ports shown in Table 12.

Table 12: **MMCM Reconfiguration Ports**

Port	Direction	Description
SADDR	Input	This port chooses the state to reconfigure the MMCM to. A value of 0 corresponds to state 1; a value of 1 corresponds to state 2.
SEN	Input	This port enables the reconfiguration state machine. Reconfiguration is triggered if this port is asserted at a rising SCLK edge.
SCLK	Input	This is the clock for the reconfiguration module. It is passed through to the DCLK output.
RST	Input	This resets the state machine and the downstream MMCM.
SRDY	Output	This port asserts for one clock cycle at the end of a reconfiguration sequence. It is to be used to notify the user that a new reconfiguration can be started.
DO[15:0]	Input	This port should be directly attached to the MMCM DO port. It is used to read register values from the MMCM.
DRDY	Input	This port should be directly attached to the MMCM DRDY port. It is used to notify the reference design when the MMCM is ready to read or write a new value.
LOCKED	Input	This port should be directly attached to the MMCM LOCKED port. It is used to notify the reference design that the MMCM is locked and to then transition from the WAIT_LOCK state.
DWE	Output	This port should be directly attached to the MMCM DWE port. It is used to enable a register write.
DEN	Output	This port should be directly attached to the MMCM DEN port. It is used to initiate a register read or write.
DADDR[6:0]	Output	This port should be directly attached to the MMCM DADDR port. It is used to address a register location for reads or writes.
DI[15:0]	Output	This port should be directly attached to the MMCM DI port. It is used to output a new register value for writes.
DCLK	Output	This port should be directly attached to the MMCM DCLK port. It is used to clock the reconfiguration port on the MMCM. It is the SCLK forwarded out of the MMCM reconfiguration module.
RST_MMCM	Output	This port should be directly attached to the MMCM RST port. It is used to reset the MMCM during a reconfiguration or when the RST port is asserted.

The reconfiguration module also has the attributes shown in Table 13. The MMCM_DRP attributes correlate with the standard MMCM primitive attributes with some slight naming differences.

Table 13: **MMCM Reconfiguration Attributes**

Attribute	Description	Valid Format Values
CLKFBOUT_MULT	This attribute modifies the input clock multiplier to change the VCO output frequency of the MMCM.	5–64; Integer values only.
CLKFBOUT_PHASE	Modifies the phase of the input clock. This affects all of the MMCM outputs.	Integer values multiplied by 1,000. For example, a 45° phase shift would be 45,000

Table 13: **MMCM Reconfiguration Attributes *(Cont'd)***

Attribute	Description	Valid Format Values
BANDWIDTH	Sets the bandwidth setting of the MMCM.	OPTIMIZED, HIGH, or LOW.
DIVCLK_DIVIDE	Sets the divide value for the DIVCLK output.	1–128; Integer values only.
CLKOUT0_DIVIDE	CLKOUT0 output divide value.	1–128; Integer values only.
CLKOUT0_PHASE	CLKOUT0 output phase value.	Integer values multiplied by 1,000. For example, a 45° phase shift would be 45,000.
CLKOUT0_DUTY	Changes the CLKOUT0 Duty Cycle Low Time.	Integer values multiplied by 100,000. For example, a 0.60 duty cycle would be 60,000.
CLKOUT1_DIVIDE	CLKOUT1 output divide value.	1–128; Integer values only.
CLKOUT1_PHASE	CLKOUT1 output phase value.	Integer values multiplied by 1,000. For example, a 45° phase shift would be 45,000.
CLKOUT1_DUTY	Changes the CLKOUT1 Duty Cycle Low Time.	Integer values multiplied by 100,000. For example, a 0.60 duty cycle would be 60,000.
CLKOUT2_DIVIDE	CLKOUT2 output divide value.	1–128; Integer values only.
CLKOUT2_PHASE	CLKOUT2 output phase value.	Integer values multiplied by 1,000. For example, a 45° phase shift would be 45,000.
CLKOUT2_DUTY	Changes the CLKOUT2 Duty Cycle Low Time.	Integer values multiplied by 1,000. For example, a 60/40 duty cycle would be 60,000.
CLKOUT3_DIVIDE	CLKOUT3 output divide value	1-128; Integer values only.
CLKOUT3_PHASE	CLKOUT3 output phase value	Integer values multiplied by 1,000. For example, a 45° phase shift would be 45,000.
CLKOUT3_DUTY	Changes the CLKOUT3 Duty Cycle Low Time.	Integer values multiplied by 100,000. For example, a 0.60 duty cycle would be 60,000.
CLKOUT4_DIVIDE	CLKOUT4 output divide value.	1-128; Integer values only.
CLKOUT4_PHASE	CLKOUT4 output phase value.	Integer values multiplied by 1,000. For example, a 45° phase shift would be 45,000.
CLKOUT4_DUTY	Changes the CLKOUT4 Duty Cycle Low Time.	Integer values multiplied by 100,000. For example, a 0.60 duty cycle would be 60,000.
CLKOUT5_DIVIDE	CLKOUT5 output divide value.	1-128; Integer values only.
CLKOUT5_PHASE	CLKOUT5 output phase value.	Integer values multiplied by 1,000. For example, a 45° phase shift would be 45,000.
CLKOUT5_DUTY	Changes the CLKOUT5 Duty Cycle Low Time.	Integer values multiplied by 100,000. For example, a 0.60 duty cycle would be 60,000.
CLKOUT6_DIVIDE	CLKOUT6 output divide value.	1-128; Integer values only.

Table 13: **MMCM Reconfiguration Attributes *(Cont'd)***

Attribute	Description	Valid Format Values
CLKOUT6_PHASE	CLKOUT6 output phase value.	Integer values multiplied by 1,000. For example, a 45° phase shift would be 45,000.
CLKOUT6_DUTY	Changes the CLKOUT6 Duty Cycle Low Time.	Integer values multiplied by 100,000. For example, a 0.60 duty cycle would be 60,000.

Using the Reference Design

Design Functionality

The `mmcm_drp.v` file has been written with two available reconfigurable states. They are denoted with an S1_ or S2_ before each of the attributes in Table 13. You can modify the parameters within each state independently. Additional states can be added or register writes removed as covered in the Design Modification section.

To change between the two states, first wait for SRDY to be asserted. When SRDY has been asserted, the state machine is ready to begin reconfiguration. The SADDR port specifies which state is loaded into the MMCM using the DRP port. In an unmodified design, a 0 loads state 1 and a 1 loads state 2. Pulsing SEN for one clock cycle triggers the reconfiguration and loads all attributes set in the MMCM DRP design. Once the reconfiguration is complete, the SRDY port is asserted and the MMCM is in its newly reconfigured state.

Design Modification

The reference design is intended to be modified to suit the specific needs of a design. The process of doing these changes is left to the user, but there are a couple of common needs that warrant some general instructions on the modification process. It should be noted that the header file `mmcm_drp_func.h` should *not* be changed. The file `mmcm_drp.v` is the primary file where design-specific modifications should be done. To perform design modifications, it is expected that the user has become intimately familiar with the functionality of the reconfiguration interface in `mmcm_drp.v` by reading through the provided source.

The first common situation is to retain the previous CLKOUT# configuration for both states. For example, if it is desired to retain the previous configuration of CLKOUT4, `mmcm_drp.v` must be modified in two locations.

- The ROM initialization must be modified to remove the two CLKOUT4 registers. This requires removing the entries that modify the `0x10` and `0x11` registers on the MMCM. When the register entries have been removed, the ROM initialization must be changed so that the initialization addresses are sequential.
- Because two registers have been removed from the ROM initialization, STATE_COUNT_CONST must be updated to reflect the two fewer register writes. STATE_COUNT_CONST should be initialized to 22 with the two registers removed from each state initialization.

A second potential design modification is to add a third state to the reference design. To do this, everything that contains an S#_ (where # is a number) must be replicated to create an S3_ set of parameters, constant function calls, and ROM initializations. The SADDR port must be updated to be a vector allowing the third state to be addressed, and the WAIT_SEN state must be updated to include the ability to set the initial ROM reconfiguration address based on SADDR.

Design Verification

The reference design was verified in hardware and with simulation. This ensures that the simulation models and the hardware functionality are equivalent. The verification process chose a number of corner cases for reconfiguration along with some standard configurations to verify that the calculations worked across each scenario. The functions that calculate the various bit settings have also gone through a complete analysis to ensure they match the calculations performed by the ISE® software backend tools during implementation.

Conclusion

This application note and reference design provide a complete implementation of the MMCM DRP functionality. Due to its modular nature, the design can be used as a full solution for DRP or can be easily extended to support additional reconfiguration states. The design also uses minimal Virtex-6 FPGA resources, consuming only 24 slices.

Reference Design Additional Information

Files

The reference design files can be downloaded from:

https://secure.xilinx.com/webreg/clickthrough.do?cid=143942

Characteristics

The reference design characteristics are summarized in Table 14.

Table 14: **Reference Design Matrix**

Parameter	Description
General:	
Developer Name	Karl Kurbjun and Carl Ribbing
Target devices	Virtex-6 Family
Source code provided	Yes
Source code format	Verilog
Design uses code/IP from existing Xilinx application note/reference designs, CORE Generator™ software, or 3rd party	No
Simulation:	
Functional simulation performed	Yes
Timing simulation performed	Yes
Testbench used for functional and timing simulations	Yes
Testbench format	Verilog
Simulator software/ version used	Modelsim 6.4b
Implementation:	
Synthesis software tools/version used	XST 11.3
Implementation software tools /versions used	ISE 11.3
Static timing analysis performed	Yes
Hardware Verification:	
Hardware verified	Yes
Hardware platform used for verification	Virtex-6 FPGA Characterization Board

Device Utilization and Performance

The reference design device utilization and performance are summarized in Table 15.

Table 15: **Device Utilization and Performance for MMCM_DRP**

Parameters	Specification/Details	
Maximum Frequency (by speed grade)	-1	190 MHz
	-2	200 MHz
	-3	200 MHz
Device Utilization without Testbench	Slices	24
	GCLK Buffers	0
	Block RAM	0
HDL Language Support	Verilog	

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.
06/09/10	1.1	Updated EDGE and NO COUNT in Table 3. Updated reference design Files, page 12 on website.

Notice of Disclaimer