XAPP640 (v1.1) January 16, 2003

Timing Constraints for Virtex-II Pro Designs

Summary

This application note discusses the usage of timing constraints in a Virtex-II Pro™ design with the PowerPC™ 405 (PPC405) processor. The interaction of the timing constraints with the PPC405, Processor Local Bus (PLB), On-Chip Peripheral Bus (OPB), and RocketIO™ transceiver are described. The interactions are specified by the clock ratio between the busses and the designs processor block. The clock ratios between the PPC405 and PLB, and the PLB to the OPB are also discussed. A reference design is used to show the exact syntax of the timing constraints and Timing Analyzer results. The reference design includes the PPC405, a RocketIO transceiver component, several PLB components, and several OPB components. User's knowledge of basic timing and constraints, and the Virtex-II Pro architecture is assumed.

Introduction

The trend in embedded system design is to integrate all the major functions into a single device. The Virtex-II Pro platform FPGA family is a complete programmable system solution allowing more flexible, complex system-on-chip (SoC) development while providing a secure means to rapidly deliver embedded designs to production.

The high-level view of a Virtex-II Pro embedded system with the PowerPC 405 processor and CoreConnect bus architecture is shown in Figure 1. The sub-systems of an embedded application include:

1. PowerPC 405 Processor
2. Processor Local Bus (PLB) Peripherals
3. On-Chip Peripheral Bus (OPB) Peripherals
4. Device Control Register (DCR) Peripherals
5. On-Chip Memory (OCM) Interface and On-Chip Block RAM (BRAM) Module

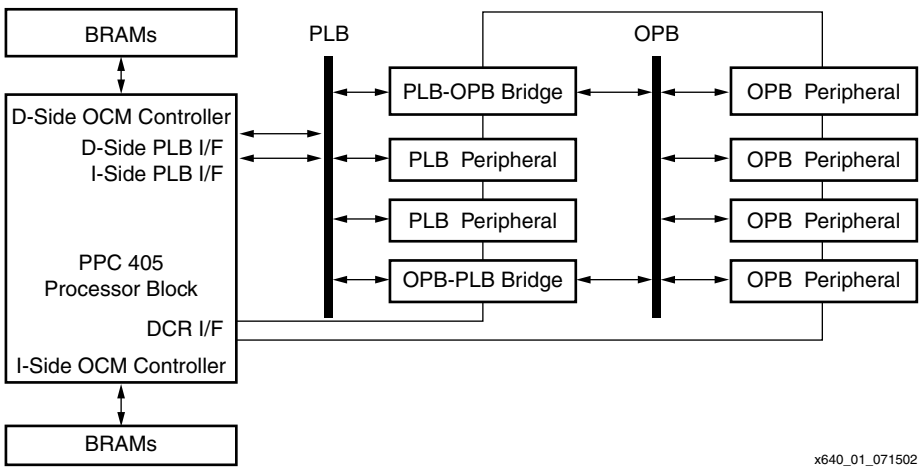

Figure 1: **Example Embedded Application**

The PowerPC 405 processor runs specific application software. The software is either stored in on-chip BRAMs or in the off-chip memories.

The PLB is the primary I/O bus protocol and generally supports higher bandwidth peripherals. The OPB is the secondary I/O bus protocol and is intended for less complex, lower performance peripherals. The processor core communicates to the OPB through a PLB-to-OPB bridge or an OPB-to-PLB bridge. The DCR is used for status and control registers in the PLB and OPB peripherals, and the processor block.

The critical clocks that interface with the processor block clocks are listed in Table 1. These clocks communicate to the processor block the specific clock ratio between the processor block clock and the other system clocks in the design. The DCRCLK and OPBCLK are clock ratios, not processor block inputs.

Table 1: **Processor Block Clock Inputs**

Clock Signal	Description
CPMC405CLOCK	Main Processor Block Clock
PLBCLK	Primary I/O Bus Clock
BRAMISOCMCLK	Reference Clock for the I-Side OCM Controller
BRAMDSOCMCLK	Reference Clock for the D-Side OCM Controller

The PowerPC405 processor block supports multiple clock domains. Using several DCM and BUFG components are recommended to create and drive the clock domains. The primary clock domains include the PLB and OPB clocks. The secondary clock domains include the DCR, INTC, and OCM clocks. The clock structure for the RocketIO transceiver components are discussed in the **RocketIO Transceiver** section.

PLB

The PLB is used as an interface between the processor block and the higher performance peripherals. The processor block has some internal logic to both align the positive edges of the PLBCLK and the CPMC405CLOCK, and to generate the appropriate enabling signals for controlling the PLB. The PLB clock must be phased aligned to the processor block. All communication between the processor block and the PLB are based upon the rising edge of the CPMC405CLOCK. The PLB is synchronous with the processor block. Table 2 shows the supported integer clock ratios between the processor block and the PLB. As an example, the processor block can be run at 300 MHz while the PLB bus is run at 100 MHz.

Table 2: **Integer Clock Ratios**

Processor Block : PLB Bus
1 : 1
2 : 1
3 : 1
4 : 1
.
13 : 1
14 : 1
15 : 1
16 : 1

For further information, please refer to the Virtex-II Pro Developers Kit Documentation.

OPB

The OPB is an interface between the lower performance peripherals and the PLB bus. The OPB requires a PLB-to-OPB bridge and OPB-to-PLB bridge to communicate to the processor block. The OPB bus is also synchronous with the PLB bus. Table 3 shows the supported integer clock ratios between the PLB bus and the OPB bus. As an example, the PLB bus can be run at 100 MHz, when the OPB bus is running at 50 MHz.

Table 3: **Integer Clock Ratios**

PLB Bus : OPB Bus
1 : 1
2 : 1
3 : 1
4 : 1

The PLB-to-OPB transfer interface logic includes the PLB-OPB and OPB-PLB bridges. It is implemented as an asynchronous FIFO interface between the PLB and OPB interfaces. The PLB-to-OPB transfer also synchronizes and passes handshaking flags between the two sides communicating that the FIFO is not empty, or to signaling the termination of a PLB read burst by the PLB master.

Since it is dual-port and supports independent clocks on each side, a BRAM is used as a memory element to store the data. The BRAM can be replaced by a number of LUT-based dual-port memories to yield higher clock frequencies but at the expense of LUT utilization. In addition, a simplified, synchronous version of this module can reduce logic utilization while increasing speed. The synchronous version requires dividing a OPB clock down from the PLB clock by an integer value and phase aligning the rising edges.

For further information, please refer to the Virtex-II Pro Developers Kit Documentation.

DCR

The Device Control Register (DCR) clock domain is a secondary clock domain. The processor block clock and the DCR clock must come from the same source and be in phase with each other. The DCR clock covers both of the processor block DCR and the memory-mapped DCR. The clock ratio between the DCR clock domain and the processor block can run at any clock ratio as long as the bus transaction completes in 64 processor block cycles. If the bus transaction does not complete in 64 processor block clock cycles, the processor block will time out and move on to the next instruction. Beyond a clock ratio of 1:8 the results are really slow.

The interface between the processor block and the DCR modules can be set to either of two modes. Mode 0 allows the DCR clock domain and the processor clock domain to be at different frequencies. The DCR clock domain must be in phase with the processor clock domain. In Mode 1 the DCR clock domain is at the same frequency as the processor clock domain. The most common frequency of the DCR clock is at half or a quarter of the PLB clock.

For further information, please refer to the Virtex-II Pro Developers Kit Documentation.

OCM

The On-Chip Memory (OCM) clock domain is another secondary clock domain. For high-speed access, the OCM clock domain covers the interface between the processor block and the block RAM surrounding the processor block. There are two independent clocks for the OCM controllers in the processor block: BRAMDSOCMCLK (data side controller) and BRAMISOCMCLK (instruction side controllers). The data side controller and the instruction side controllers can run at different frequencies, based upon the access time of the BRAM.

When the processor block, OCM controller, and BRAMs run at the same clock frequency, the processor is in single-cycle mode. Multi-cycle mode occurs when the processor is running at a

higher frequency than the BRAMs. In the single-cycle mode and multi-cycle mode, the BRAMISOCMCLK and BRAMDSOCMCLK signals are provided to the OCM controller as inputs.

The OCM controller has internal logic to align the rising edges of the BRAMs clocks and the processor block clock. Through timing analysis, the clock ratio between the processor block clock and the BRAMs clocks is determined by the worst case access time between the OCM controller interface and the BRAMs interface. Based upon the timing analysis, most designs use multi-cycle mode. The processor block clock and the BRAMDSOCMCLK must be integer multiples. The same is true for the BRAMISOCMCLK with respect to the processor block clock. They do not have to be the same integer values, nor do they have to be the same integer clock ratio to the PLB clock.

Since the clock ratio between the processor block and the OCM clocks is unknown, the processor block has control registers in the OCM controllers. The control registers, are called ISCNTL[0:7] and DSCNTL[0:7], for the instruction side and data side, respectfully. To tell the processor block what the clock ratio is, bits five through seven are called the ISOCMMCM bits and DSOCMMCM bits. The value of the bits is based upon the intended clock ratio, shown in Table 4 and follows the equation 2n–1, where n is the number of processor block clocks in one OCM clock cycle. The most common clock frequency of the OCM clock is half of the processor block clock. These bits are assigned through the ISCNTLVALUE input and the DSCNTLVALUE input for the instruction side and data side. Using an example of the DSCNTLVALUE input value of `83h`, the clock ratio between the processor block and the data side of the OCM is a 2:1 ratio, and the DSOCM address decoder is enabled.

Table 4: **Processor Block Clock: OCM Clock Ratio**

ISOCMMCM[0:2] / DSOCMMCM[0:2]	Processor Block: OCM Clock Ratio
000b	Not Supported
001b	1:1
010b	Not Supported
011b	2:1
100b	Not Supported
101b	3:1
110b	Not Supported
111b	4:1

The larger the amount of BRAM used for OCM, the greater the performance penalty. If the amount of BRAM for the DSOCM/ISOCM is to large, based on timing analysis, then use the multi-cycle mode and/or reduce the processor block frequency. Timing analysis will determine the clock ratio.

For further information, please refer to the Virtex-II Pro Developers Kit Documentation.

INTC

The third clock domain is the Interrupt Controller clock domain. This clock domain runs at the same frequency and clock ratio as the DCR. The key interrupt signals going into the processor block are EICC405EXTINPUTIRQ for the off-core non-critical interrupts and EICC405CRITINPUTIRQ for the off-core critical interrupts.

For further information, please refer to the Virtex-II Pro Developers Kit Documentation.

RocketIO Transceiver

The flexible programmable features of the RocketIO transceiver allows a multi-gigabit serial transceiver to be easily integrated into any Virtex-II Pro design. The RocketIO transceiver has

its own clocking scheme and usage. The main clock (REFCLK) for the RocketIO transceiver must be a low-jitter source and not driven by a DCM. The other user clocks (TXUSRCLK, RXUSRCLK, TXUSRCLK2, and RXUSRCLK2) of the RocketIO transceiver have no required phase relationship with the REFCLK and are usually driven by a DCM and BUFG. The USRCLKs are frequency locked to the REFCLK. The frequency range for REFCLK and USRCLKs are from 50 MHz to 156.25 MHz. The frequency range for the USERCLK2s is from 25 MHz to 312.5 MHz, which is from the clock ratios changing with the data widths.

The most common REFCLK_RIO topology is shown in Figure 2. The user clocks are driven off of the CLK0 pin of the DCM and also drives the feedback pin of the DCM. The path from REFCLK_RIO to the feedback pin is required for proper clock skew compensation between the REFCLK_RIO at the external pad and the user clocks through out the FPGA. The REFCLK_RIO may also be connected to the RocketIO component via another BUFG.

For further information, please refer to the Virtex-II Pro Developers Kit Documentation.

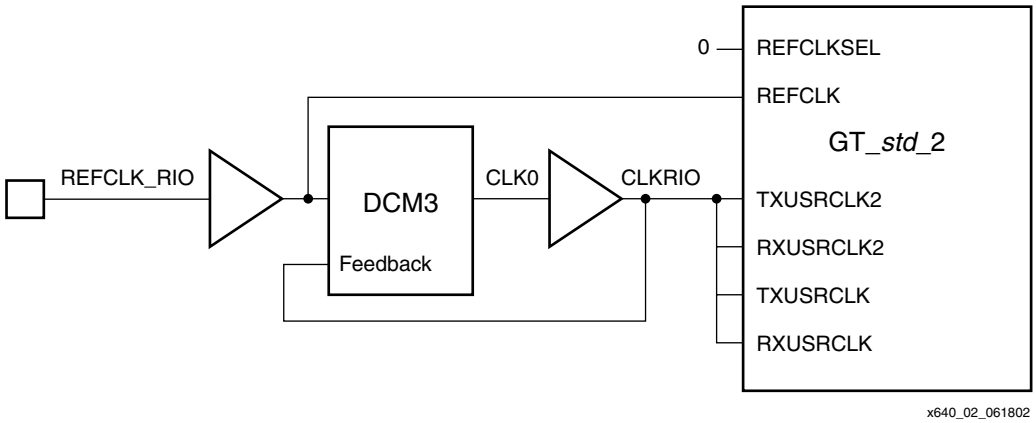

Figure 2: **RocketIO DCM Configuration**

Design Background

The reference design (**xapp640.zip**) is based upon the Embedded PPC405 Reference System from the Virtex-II Pro Developers Kit. It is available on the Xilinx FTP site at **ftp://ftp.xilinx.com/pub/applications/xapp/xapp640.zip**. Several of the modules have been removed to simplify the overall design and analysis. For further information on the original Embedded PPC405 Reference System design, please refer to the Virtex-II Pro Developers Kit Documentation.

The design consists of several of the most commonly used modules; the processor block (PPC405), RocketIO transceiver, PLB-Arbiter, PLB2OPB Bridge, OPB2PLB Bridge, OPB-Arbiter, and several OPB and PLB peripherals. The peripherals include SRAM and BRAM MEMC on the PLB bus and LCD Controller and DCR Bridge on the OPB bus. Figure 3 shows the block diagram of this design. The files associated with this design are listed in Table 5.

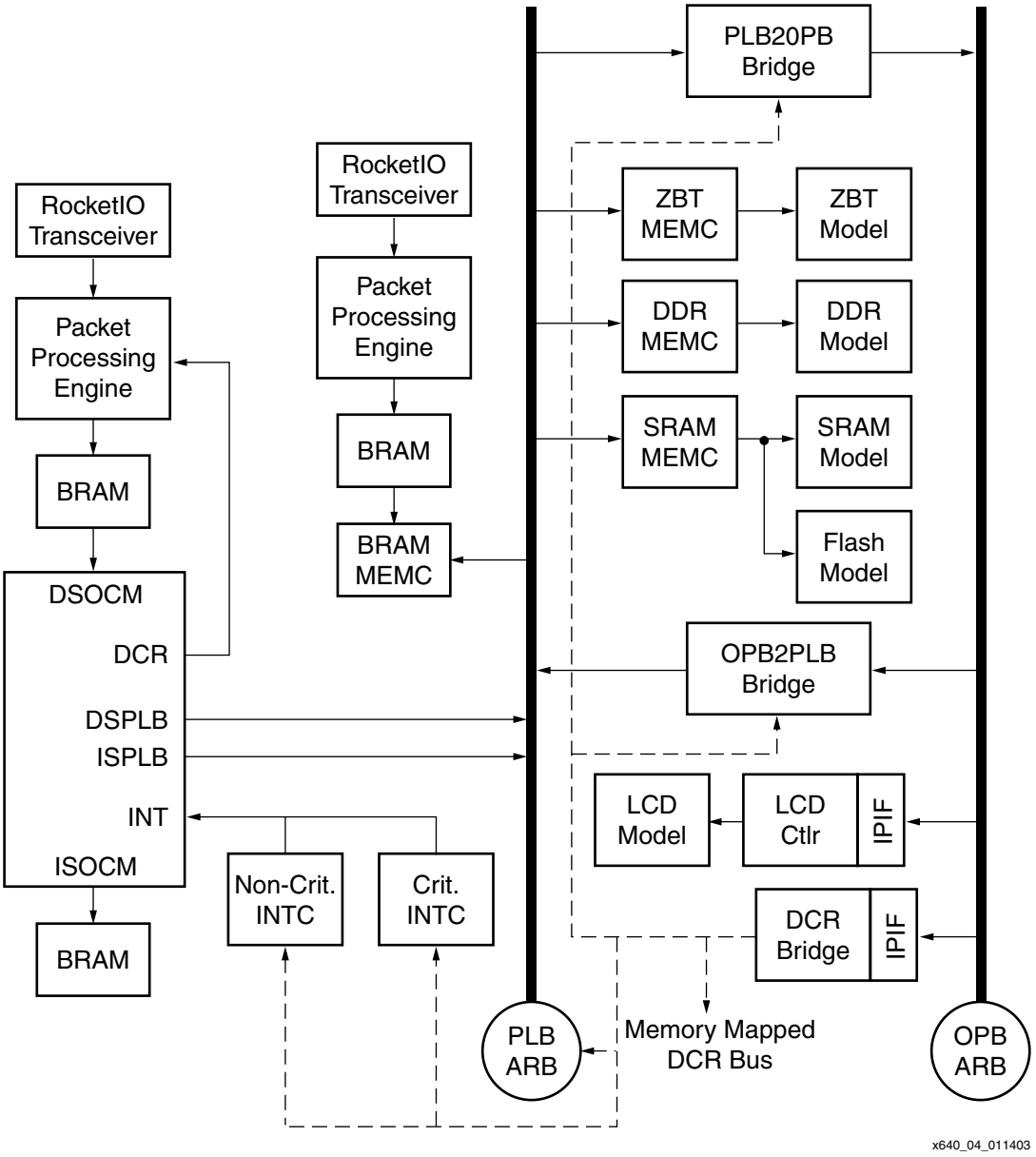

Figure 3: **Overall Block Diagram of Reference Design**

Table 5: **Design Files**

Design File	Description
block_ram.v	Instantiates the dual-port block RAMs connected to both the PLB BRAM controller and Packet Processor B. It contains 16 BRAMs for a total of 32K bytes of memory
clk_rst_startup.v	Generates the global clocks and reset signals for the FPGA as well as a startup circuit to hold reset until the DCMs are locked onto the reference clock.
global_params.v	Defines constants for system memory map and configuration options
ip_wrapper.v	Instantiates the PLB arbiter, PLB devices, OPB arbiter, and OPB devices. It also connects up the DCR to access various status registers.
opb_bus_logic.v	Instantiates the AND-OR and OR logic needed by the OPB bus. This bus logic assumes there are up to 13 slaves on the bus. The modifications to increase the number of supported slaves should be evident by looking at the code.
plb_bus_logic.v	Instantiates the OR logic needed by slaves on the PLB bus. This bus logic assumes there are up to nine slaves on the bus. The modifications to increase the number of supported slaves should be evident by looking at the code.
src.lst	List of files for the makefile
top.v	This module is in the top level of a Virtex-II Pro based system. It instantiates the CPU and IP wrapper. The IP wrapper contains all the user IP designs. Global clock and power-on reset logic are also instantiated in this module.
top_cpu.v	This module instantiates the CPU, RocketIO transceivers, and Packet Processor module.
top_ip.v	This module instantiates the PPC405 IP.

The clocking structure of the design uses two main reference clocks to drive four DCMs. These four DCMs drive the remaining clocks. The reference clock (REFCLK) comes into the device at 100 MHz. The first DCM drives PLBCLK, OPBCLK, and CPUCLK from the CLK0 pin, CLKDV pin, and CLKFX pin, respectfully. The CLKDV pin has an attribute of CLKDV_DIVIDE = 2 and the CLKFX pin has an attribute of CLKFX_MULTIPLY = 3. The frequency of the PLBCLK is 100 MHz, the OPBCLK is 50 MHz, and the CPUCLK is 300 MHz.

The second DCM drives DCRCLK and OCMCLK with the CLK0 pin and CLKFX pin, respectfully. The CLKFX pin has an attribute of CLKFX_DIVIDE = 2 and CLKFX_MULTIPLY = 3. The frequency of the DCRCLK is 100 MHz and the OCMCLK is 150 MHz.

The third DCM drives DDR_CLK with the CLK0 pin and CLK2X with the CLK2X pin. The frequency of the DDR_CLK is 100 MHz and the CLK2X is 200 MHz.

The fourth DCM drives the USRCLKs for the RocketIO transceiver. The input of the fourth DCM also drives the REFCLK of the RocketIO transceiver.

All of the clock structure is in the `clk_rst_startup.v` file, shown in Figure 4. In this reference design each major clock is on a BUFG from the DCM. This is only done for analysis purposes. If both the DCRCLK and the PLBCLK run at the same frequency, they can be combined to use one BUFG. This increases device utilization in a design.

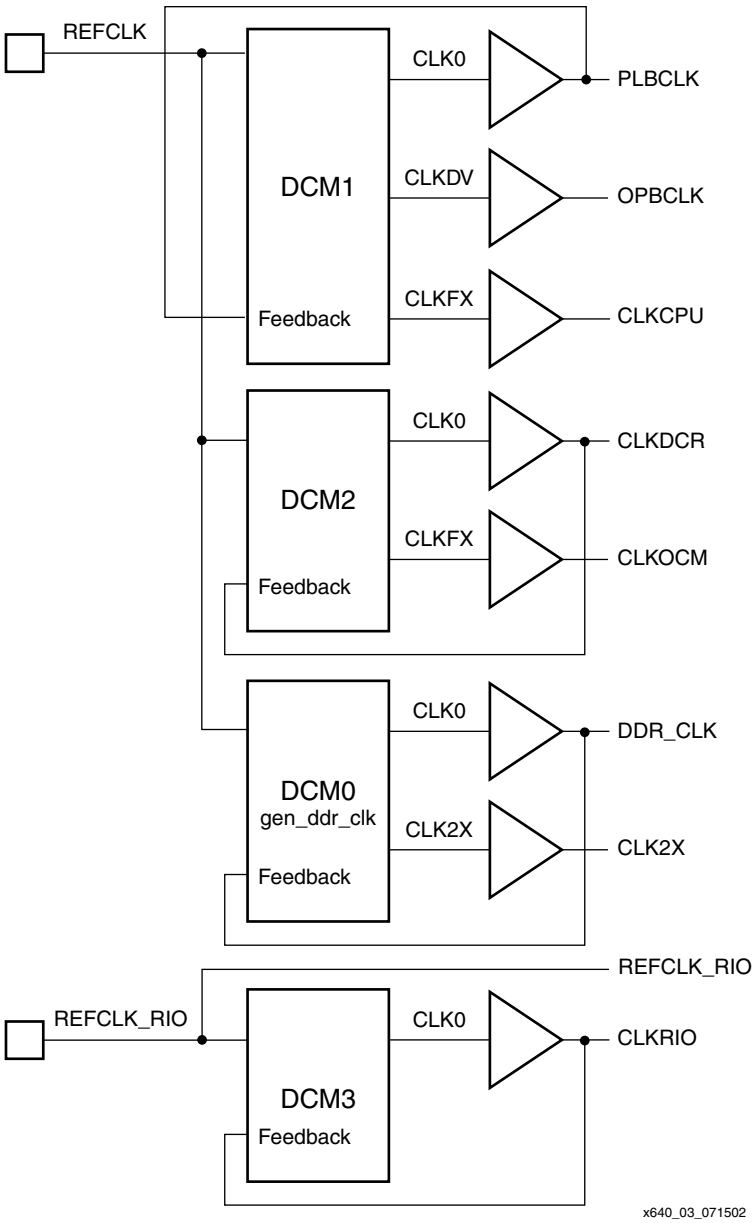

Figure 4: **Clocking Structure with Four DCMs**

The timing constraints for each of the clocks used in this design are listed below. The main timing constraint used is the PERIOD constraint, the clocks are related back to the processor block clock. For more information about the timing constraints used in this reference design, please review the Constraints Guide: http://toolbox.xilinx.com/docsan/xilinx4/data/docs/cgd/cgd.html. The timing constraints are based upon the primary and secondary clock domains in Figure 4.

Processor Block

Processor block clock PERIOD constraint at 300 MHz:

```
NET top_ip/clk_rst_startup/clkcpu_i TNM_NET = CLKCPU;
TIMESPEC TS_CLKCPU = PERIOD CLKCPU 3.334;
```

PLB

PLB clock PERIOD constraint at 100 MHz:

```
NET top_ip/clk_rst_startup/plbclk_i TNM_NET = PLBCLK;
TIMESPEC TS_PLBCLK = PERIOD PLBCLK TS_CLKCPU * 3 ;
```

OPB

OPB clock PERIOD constraint at 50 MHz:

```
NET top_ip/clk_rst_startup/opbclk_i TNM_NET = OPBCLK;
TIMESPEC TS_OPBCLK = PERIOD OPBCLK TS_PLBCLK * 2 ;
```

DCR

DCR clock PERIOD constraint at 100 MHz:

```
NET top_ip/clk_rst_startup/clkdcr_i TNM_NET = CLKDCR;
TIMESPEC TS_CLKDCR = PERIOD CLKDCR TS_CLKCPU * 3;
```

In addition to the PERIOD constraint, other timing constraints are added for specific paths. The interface between the DCR and the processor block runs at the frequency of the DCR clock. The signals between the DCR and the processor block are shown in Figure 5. These signals do not need to run at the same frequency as the processor block since several key nets going between the DCR and the processor block were identified and used as through points for the timing constraints. The following timing constraint covered these paths.

```
NET "top_cpu/C405DCRABUS*" TPTHRU = "DCR_DATA_GRP";
NET "top_cpu/C405DCRDBUSOUT*" TPTHRU = "DCR_DATA_GRP";
NET "top_cpu/DCRC405DBUSIN*"  TPTHRU = "DCR_DATA_GRP";
TIMESPEC "TS_DCR1" = FROM FFS  THRU DCR_DATA_GRP TO CPUS TS_CLKDCR ;
TIMESPEC "TS_DCR2" = FROM CPUS THRU DCR_DATA_GRP TO CPUS TS_CLKDCR ;
TIMESPEC "TS_DCR3" = FROM CPUS THRU DCR_DATA_GRP TO FFS  TS_CLKDCR ;

INST "top_ip/ip_wrapper/opb2dcr_brg/opb_ipif_slv_sram/*Sl_DBus*"   TNM =
"OPB_DCRI_GRP";
TIMESPEC "TS_OPB_DCRI" = FROM FFS TO "OPB_DCRI_GRP" TS_CLKDCR;

INST "top_ip/ip_wrapper/opb2dcr_brg/dcr_brg_ipif/*OPB_dcrDBusOut*" TNM =
"OPB_DCRO_GRP";
INST "top_ip/ip_wrapper/opb2dcr_brg/dcr_brg_ipif/*OPB_dcrABus*"    TNM =
"OPB_DCRO_GRP";
TIMESPEC "TS_OPB_DCRO" = FROM "OPB_DCRO_GRP" TO FFS TS_CLKDCR;
```

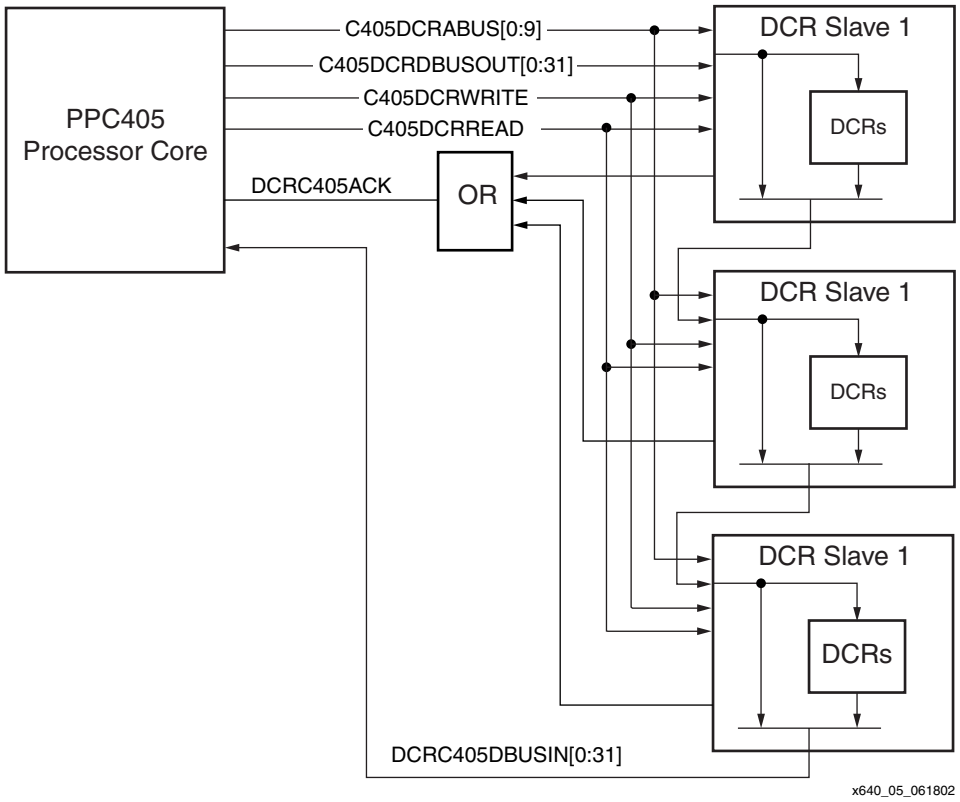

Figure 5: **Example CPU DCR Bus**

OCM

OCM clock PERIOD constraint at 150 MHz:

```
NET top_ip/clk_rst_startup/clkocm_i TNM_NET = CLKOCM;
TIMESPEC TS_CLKOCM = PERIOD CLKOCM TS_CLKCPU * 2 ;
```

In addition to the PERIOD constraint, specific attributes are set on the processor block. The relationship between the OCM and the processor block is set in the ISOCMMCM bits and DSOCMMCM bits. The ISOCMMCM bits and DSOCMMCM bits are assigned in the ISCNTLVALUE and DSCNTLVALUE inputs to the processor block in the `top_cpu.v` file. The values assigned to these inputs are listed in the `global_params.v` file. In this design, the value was set to `83h` for the instruction side and `C3h` for the data side.

INTC

The interface between the Interrupt Controllers (INTC) and the processor block were constrained in a similar fashion as the DCR. The signals between the INTC and the processor block, as shown in Figure 3, only needed to run at the DCR clock frequency. Some key nets are identified as through points for the constraints. The following timing constraints covered these paths.

```
NET EICC405CRITINPUTIRQ TPTHRU = INTC_DATA_GRP;
NET EICC405EXTINPUTIRQ TPTHRU = INTC_DATA_GRP;
TIMESPEC "TS_INTC1" = FROM FFS  THRU INTC_DATA_GRP TO CPUS TS_CLKDCR ;
TIMESPEC "TS_INTC2" = FROM CPUS THRU INTC_DATA_GRP TO CPUS TS_CLKDCR ;
TIMESPEC "TS_INTC3" = FROM CPUS THRU INTC_DATA_GRP TO FFS  TS_CLKDCR ;
```

RocketIO Clock Period

RocketIO clock PERIOD constraint at 150 MHz:

```
NET top_ip/clk_rst_startup/clkrio_i TNM_NET = CLKRIO;
TIMESPEC TS_CLKRIO = PERIOD CLKRIO 6.67;
```

External Peripheral Interface

In addition to the previous timing constraints, the interface to the downstream and upstream devices is included in the reference design. This interface uses an OFFSET constraint.

```
#------------------------------------------------------------------------
# OFFSET Constraints for the IO
#------------------------------------------------------------------------

# ZBT Memory
NET ZBT_ad<*> TNM = zbt_grp;
NET ZBT_be_n<*> TNM = zbt_grp;
NET ZBT_ce* TNM = zbt_grp;
NET ZBT_lbo_n TNM = zbt_grp;
NET ZBT_oe_n TNM = zbt_grp;
NET ZBT_rw_n TNM = zbt_grp;
NET ZBT_dq<*> TNM = zbt_grp;
TIMEGRP zbt_grp OFFSET = OUT 7 AFTER REFCLK;
TIMEGRP zbt_grp OFFSET = IN 5 BEFORE REFCLK;

# DDR Memory
NET  DDR_ba<*> TNM = ddr_grp;
NET  DDR_casb TNM = ddr_grp;
NET  DDR_csb<*> TNM = ddr_grp;
NET  DDR_dm<*> TNM = ddr_grp;
NET  DDR_dqs<*> TNM = ddr_grp;
NET  DDR_rasb TNM = ddr_grp;
NET  DDR_web TNM = ddr_grp;
NET  DDR_dq<*> TNM = ddr_grp;
TIMEGRP ddr_grp OFFSET = OUT 7 AFTER REFCLK;
TIMEGRP ddr_grp OFFSET = IN 5 BEFORE REFCLK;

#SRAM Memory
NET  SRAM_ad<*> TNM = sram_grp;
NET  SRAM_ben<*> TNM = sram_grp;
NET  SRAM_wen<*> TNM = sram_grp;
NET  SRAM_dq<*> TNM = sram_grp;
TIMEGRP sram_grp OFFSET = OUT 7 AFTER REFCLK;
TIMEGRP sram_grp OFFSET = IN 5 BEFORE REFCLK;
```

Timing Analysis

The processor block is treated by the timing tools as a synchronous element. The PERIOD constraint covers from synchronous elements to other synchronous elements, including the processor block. By default, the PERIOD constraint will not cover data paths between clock domains. For the PERIOD constraint to cover these paths, the PERIOD constraints need to be related. This is normally done in the User Constraints File (UCF). The destination clock PERIOD constraint covers the data path crossing between clock domains. The FROM:TOs cover the paths where there is a slow exception to the PERIOD constraints. Please refer to the Constraints Guide, **http://toolbox.xilinx.com/docsan/xilinx4/data/docs/cgd/cgd.html**, for more information on these constraints.

The following examples are based upon a specific speeds file (1.65 2002-06-19) and version of the implementation tools (5.1i, F.23) and synthesis tools (Synplicity 7.02). Specific design results can vary from the following examples. The examples are shown by the primary and secondary clock domains. If the same or similar results are not found, please go through the Timing Improvement Wizard Problem Solver at: **http://service.xilinx.com/support/cgibin-/webcgi.exe?New,KB=Timing_Improve_PS**.

Processor Block

```
Timing constraint: TS_CLKCPU = PERIOD TIMEGRP "CLKCPU"  3.334 nS   HIGH
50.000000 % ;
 1 item analyzed, 0 timing errors detected.
 Minimum period is   3.064ns.
--------------------------------------------------------------------------------
Slack:                  0.270ns (requirement - (data path - clock skew))
  Source:               top_cpu/DCRC405ACK (FF)
  Destination:          top_cpu/PPC405 (CPU)
  Requirement:          3.334ns
  Data Path Delay:      3.064ns (Levels of Logic = 0)
  Clock Skew:           0.000ns
  Source Clock:         CLKDCR rising at 0.000ns
  Destination Clock:    CLKCPU rising at 3.334ns
  Data Path: top_cpu/DCRC405ACK to top_cpu/PPC405
```

PLB

```
Timing constraint: TS_PLBCLK = PERIOD TIMEGRP "PLBCLK" TS_CLKCPU * 3.000000
HIGH 50.000 % ;
 16839 items analyzed, 0 timing errors detected.
 Minimum period is   9.835ns.
--------------------------------------------------------------------------------
Slack:                  0.167ns (requirement - (data path - clock skew))
  Source:               top_cpu/PPC405 (CPU)
  Destination:          top_ip/ip_wrapper/ZBT_S4/burst_count_reg[2] (FF)
  Requirement:          10.002ns
  Data Path Delay:      9.321ns (Levels of Logic = 3)
  Clock Skew:           -0.514ns
  Source Clock:         CLKPLB rising at 0.000ns
  Destination Clock:    CLKPLB rising at 10.002ns
  Data Path: top_cpu/PPC405 to top_ip/ip_wrapper/ZBT_S4/burst_count_reg[2]
```

OPB

```
Timing constraint: TS_OPBCLK = PERIOD TIMEGRP "OPBCLK" TS_PLBCLK * 2.000000
HIGH 50.000 % ;
 30144 items analyzed, 0 timing errors detected.
 Minimum period is  18.086ns.
--------------------------------------------------------------------------------
Slack:                  0.959ns (requirement - (data path - clock skew))
  Source:
top_ip/ip_wrapper/plb_arb/plb_arb_top/DCR/reg_PLB_CPU_exeMxDcr/Q_fast[0]
(FF)
  Destination:          top_ip/ip_wrapper/intc_crit/intc_IERreg/ier25 (FF)
  Requirement:          10.002ns
  Data Path Delay:      9.022ns (Levels of Logic = 7)
  Clock Skew:           -0.021ns
  Source Clock:         CLKPLB rising at 10.002ns
  Destination Clock:    top_ip/CLKOPB rising at 20.004ns
  Data Path:
top_ip/ip_wrapper/plb_arb/plb_arb_top/DCR/reg_PLB_CPU_exeMxDcr/Q_fast[0] to
top_ip/ip_wrapper/intc_crit/intc_IERreg/ier25
```

DCR

```
Timing constraint: TS_CLKDCR = PERIOD TIMEGRP "CLKDCR" TS_CLKCPU * 3.000000
HIGH 50.000 % ;
 380 items analyzed, 0 timing errors detected.
 Minimum period is   4.578ns.
--------------------------------------------------------------------------------
Slack:                  5.424ns (requirement - (data path - clock skew))
  Source:               top_cpu/dcr_write_d1 (FF)
  Destination:          top_cpu/PPDCR_A/gbio_control[8] (FF)
```

```
  Requirement:          10.002ns
  Data Path Delay:      4.578ns (Levels of Logic = 2)
  Clock Skew:           0.000ns
  Source Clock:         CLKDCR rising at 0.000ns
  Destination Clock:    CLKDCR rising at 10.002ns
  Data Path: top_cpu/dcr_write_d1 to top_cpu/PPDCR_A/gbio_control[8]
Timing constraint: TS_DCR1 = MAXDELAY FROM TIMEGRP "FFS" THRU TIMEGRP
"DCR_DATA_GRP" TO TIMEGRP
"CPUS" TS_CLKDCR * 1.000 ;
 226 items analyzed, 0 timing errors detected.
 Maximum delay is   5.981ns.
--------------------------------------------------------------------------------
Slack:                  4.021ns (requirement - (data path - clock skew))
  Source:               top_cpu/PPDCR_A/ReadAndDecodeReg (FF)
  Destination:          top_cpu/PPC405 (CPU)
  Requirement:          10.002ns
  Data Path Delay:      5.981ns (Levels of Logic = 2)
  Clock Skew:           0.000ns
  Source Clock:         CLKDCR rising at 0.000ns
  Destination Clock:    CLKCPU rising at 3.334ns
  Data Path: top_cpu/PPDCR_A/ReadAndDecodeReg to top_cpu/PPC405
Timing constraint: TS_DCR2 = MAXDELAY FROM TIMEGRP "CPUS" THRU TIMEGRP
"DCR_DATA_GRP" TO TIMEGRP
"CPUS" TS_CLKDCR * 1.000 ;
 337 items analyzed, 0 timing errors detected.
 Maximum delay is   9.801ns.
--------------------------------------------------------------------------------
Slack:                  0.201ns (requirement - (data path - clock skew))
  Source:               top_cpu/PPC405 (CPU)
  Destination:          top_cpu/PPC405 (CPU)
  Requirement:          10.002ns
  Data Path Delay:      9.801ns (Levels of Logic = 4)
  Clock Skew:           0.000ns
  Source Clock:         CLKCPU rising at 0.000ns
  Destination Clock:    CLKCPU rising at 3.334ns
  Data Path: top_cpu/PPC405 to top_cpu/PPC405
Timing constraint: TS_DCR3 = MAXDELAY FROM TIMEGRP "CPUS" THRU TIMEGRP
"DCR_DATA_GRP" TO TIMEGRP
"FFS" TS_CLKDCR * 1.000 ;
 1816 items analyzed, 0 timing errors detected.
 Maximum delay is   9.285ns.
--------------------------------------------------------------------------------
Slack:                  0.717ns (requirement - (data path - clock skew))
  Source:               top_cpu/PPC405 (CPU)
  Destination:          top_cpu/PPDCR_A/gbio_control[8] (FF)
  Requirement:          10.002ns
  Data Path Delay:      8.824ns (Levels of Logic = 4)
  Clock Skew:           -0.461ns
  Source Clock:         CLKCPU rising at 6.668ns
  Destination Clock:    CLKDCR rising at 10.002ns
  Data Path: top_cpu/PPC405 to top_cpu/PPDCR_A/gbio_control[8]
Timing constraint: TS_OPB_DCRI = MAXDELAY FROM TIMEGRP "FFS" TO TIMEGRP
"OPB_DCRI_GRP" TS_CLKDCR *
1.000 ;
 786 items analyzed, 0 timing errors detected.
 Maximum delay is   9.022ns.
--------------------------------------------------------------------------------
Slack:                  0.980ns (requirement - (data path - clock skew))
  Source:
top_ip/ip_wrapper/plb_arb/plb_arb_top/DCR/reg_PLB_CPU_exeMxDcr/Q_fast[0]
(FF)
```

```
  Destination:
top_ip/ip_wrapper/opb2dcr_brg/opb_ipif_slv_sram/Sl_DBus[25] (FF)
  Requirement:          10.002ns
  Data Path Delay:      8.986ns (Levels of Logic = 10)
  Clock Skew:           -0.036ns
  Source Clock:         CLKPLB rising at 10.002ns
  Destination Clock:    top_ip/CLKOPB rising at 20.004ns
  Data Path:
top_ip/ip_wrapper/plb_arb/plb_arb_top/DCR/reg_PLB_CPU_exeMxDcr/Q_fast[0] to
top_ip/ip_wrapper/opb2dcr_brg/opb_ipif_slv_sram/Sl_DBus[25]
Timing constraint: TS_OPB_DCRO = MAXDELAY FROM TIMEGRP "OPB_DCRO_GRP" TO
TIMEGRP "FFS" TS_CLKDCR *
1.000 ;
 21096 items analyzed, 0 timing errors detected.
 Maximum delay is   9.060ns.
--------------------------------------------------------------------------------
Slack:                  0.942ns (requirement - (data path - clock skew))
  Source:
top_ip/ip_wrapper/opb2dcr_brg/dcr_brg_ipif/OPB_dcrABus_19[8] (FF)
  Destination:          top_ip/ip_wrapper/intc_crit/intc_IERreg/ier25 (FF)
  Requirement:          10.002ns
  Data Path Delay:      9.041ns (Levels of Logic = 7)
  Clock Skew:           -0.019ns
  Source Clock:         top_ip/CLKOPB rising at 0.000ns
  Destination Clock:    top_ip/CLKOPB rising at 20.004ns
 Data Path: top_ip/ip_wrapper/opb2dcr_brg/dcr_brg_ipif/OPB_dcrABus_19[8] to
top_ip/ip_wrapper/intc_crit/intc_IERreg/ier25
```

OCM

```
Timing constraint: TS_CLKOCM = PERIOD TIMEGRP "CLKOCM" TS_CLKCPU * 2.000000
HIGH 50.000 % ;
 350 items analyzed, 0 timing errors detected.
 Minimum period is   6.154ns.
--------------------------------------------------------------------------------
Slack:                  0.514ns (requirement - (data path - clock skew))
  Source:               top_cpu/PPC405 (CPU)
  Destination:          top_cpu/DSBRAM/u1.A (RAM)
  Requirement:          6.668ns
  Data Path Delay:      5.711ns (Levels of Logic = 0)
  Clock Skew:           -0.443ns
  Source Clock:         CLKOCM rising at 0.000ns
  Destination Clock:    CLKOCM rising at 6.668ns
  Data Path: top_cpu/PPC405 to top_cpu/DSBRAM/u1.A
```

INTC

```
Timing constraint: TS_INTC1 = MAXDELAY FROM TIMEGRP "FFS" THRU TIMEGRP
"INTC_DATA_GRP" TO TIMEGRP
"CPUS" TS_CLKDCR * 1.000 ;
 2 items analyzed, 0 timing errors detected.
 Maximum delay is   3.713ns.
--------------------------------------------------------------------------------
Slack:                  6.289ns (requirement - (data path - clock skew))
  Source:               top_ip/ip_wrapper/EICC405EXTINPUTIRQ (FF)
  Destination:          top_cpu/PPC405 (CPU)
  Requirement:          10.002ns
  Data Path Delay:      3.713ns (Levels of Logic = 0)
  Clock Skew:           0.000ns
  Source Clock:         CLKPLB rising at 0.000ns
  Destination Clock:    CLKCPU rising at 3.334ns
  Data Path: top_ip/ip_wrapper/EICC405EXTINPUTIRQ to top_cpu/PPC405
```

External Peripheral Interface

```
Timing constraint: TIMEGRP "zbt_grp" OFFSET = OUT 7 nS  AFTER COMP "REFCLK" ;
 158 items analyzed, 0 timing errors detected.
 Minimum allowable offset is   4.926ns.
--------------------------------------------------------------------------------
Slack:                 2.074ns (requirement - (clock arrival + clock path +
data path))
  Source:               REFCLK (PAD)
  Destination:          ZBT_dq<14> (PAD)
  Source Clock:         CLKPLB rising at 0.000ns
  Requirement:          7.000ns
  Data Path Delay:      5.261ns (Levels of Logic = 1)
  Clock Path Delay:     -0.335ns (Levels of Logic = 3)
  Clock Path: REFCLK to top_ip/ip_wrapper/ZBT_S4/wr_data_en_reg_rep14_i
Timing constraint: TIMEGRP "zbt_grp" OFFSET = IN 5 nS  BEFORE COMP "REFCLK" ;
 64 items analyzed, 0 timing errors detected.
 Minimum allowable offset is   1.007ns.
--------------------------------------------------------------------------------
Slack:                3.993ns (requirement - (data path - clock path - clock
arrival))
  Source:               ZBT_dq<1> (PAD)
  Destination:          top_ip/ip_wrapper/ZBT_S4/Sl_rdDBus[62] (FF)
  Destination Clock:    CLKPLB rising at 0.000ns
  Requirement:          5.000ns
  Data Path Delay:      0.660ns (Levels of Logic = 1)
  Clock Path Delay:     -0.347ns (Levels of Logic = 3)
  Data Path: ZBT_dq<1> to top_ip/ip_wrapper/ZBT_S4/Sl_rdDBus[62]
Timing constraint: TIMEGRP "ddr_grp" OFFSET = OUT 8 nS  AFTER COMP "REFCLK" ;
```

```
 5 items analyzed, 0 timing errors detected.
 Minimum allowable offset is   7.827ns.
--------------------------------------------------------------------------------
Slack:                  0.173ns (requirement - (clock arrival + clock path +
data path))
  Source:                REFCLK (PAD)
  Destination:           DDR_web (PAD)
  Source Clock:          CLKPLB falling at 5.001ns
  Requirement:           8.000ns
  Data Path Delay:       3.177ns (Levels of Logic = 0)
  Clock Path Delay:      -0.351ns (Levels of Logic = 3)
  Clock Path: REFCLK to top_ip/ip_wrapper/DDR_S1/I_ddr_controller/ip_DDR_web
Timing constraint: TIMEGRP "sram_grp" OFFSET = OUT 11 nS  AFTER COMP "REFCLK"
;
 143 items analyzed, 0 timing errors detected.
 Minimum allowable offset is   8.922ns.
--------------------------------------------------------------------------------
Slack:                  2.078ns (requirement - (clock arrival + clock path +
data path))
  Source:                REFCLK (PAD)
  Destination:           SRAM_wen<2> (PAD)
  Source Clock:          CLKPLB falling at 5.001ns
  Requirement:           11.000ns
  Data Path Delay:       4.267ns (Levels of Logic = 0)
  Clock Path Delay:      -0.346ns (Levels of Logic = 3)
  Clock Path: REFCLK to top_ip/ip_wrapper/SRAM_FLASH_S2/SRAM_wen[2]
```

Summary

From the timing analysis outlined in this application note, all of the timing constraints were met. The next step is to perform timing simulation on this design. When running the reference design through the implementation tools, there is a detailed timing report that can have different timing delays.

The clock structure for the primary and secondary clock domains, and the RocketIO clock domain are discussed in this application note. The timing constraints for each clock domain were included under each clock domain. The timing analysis shows that the processor block in the reference design ran faster than 300 MHz.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
08/01/02	1.0	Initial Xilinx release.
01/16/03	1.1	Clarified integer clock ratios in the **PLB** and **OPB** sections.