PPC405 (Wrapper)

DS422 (v2.5.1) January 15, 2004

Product Overview

Introduction

This document describes the specifications for the PPC405 top level wrapper.

Features

- Instantiate PowerPC405 Processor Block primitive.
- Parameter controlled TIE ports
- Parameter controlled DCR interface resynchronization

Implementation

The signal interface is the same as PowerPC 405 Processor Block with the following exceptions:

- TIEDSOCMDCRADDR input is driven by the parameter C_DSOCM_DCR_BASEADDR
- TIEISOCMDCRADDR input is driven by the parameter C_ISOCM_DCR_BASEADDR
- TIEC405DETERMINISTICMULT input is driven by the parameter C_DETERMINISTIC_MULT
- TIEC405DISOPERANDFWD input is driven by the parameter C_DISABLE_OPERAND_FORWARDING
- TIEC405MMUEN input is driven by the parameter C_MMU_ENABLE
- The input DCRCLK is used inside the wrapper but does not connect to the PowerPC 405 Processor Block

The PPC405 top level wrapper conditionally instantiates resynchronization registers on the DCR bus interface based on the user setting of the parameter C_DCR_RESYNC. These registers are clocked with the wrapper input signal DCRCLK. The resynchronization registers are used to simplify the timing convergence in the place and route tool.

For more information about PowerPC 405 Processor, see the *PowerPC 405 Processor Block Reference Guide*.

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Virtex-II Pro™	
Version of Core	ppc405	v2.00.c
Resources Used		
	Min	Max
Slices	N/A	N/A
LUTs	0	0
FFs	0	77
Block RAMs	0	0
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	None	
Design Tool Requirements		
Xilinx Implementation Tools	6.2i or later	
Verification	N/A	
Simulation	ModelSim SE/EE 5.6e or later	
Synthesis	XST	
Support		
Provided by Xilinx, Inc.		

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

PPC405 Parameters

Table 1: **Parameters**

Parameter	Description	Type
C_MMU_ENABLE	This parameter will be used to drive the input pin TIEC405MMUEN of processor block. The default value in MPD is 1.	integer
C_DETERMINISTIC_MULT	This parameter will be used to drive the input pin TIEC405DETERMINISTICMULT of processor block. The default value in MPD is 0. For Virtex-II Pro devices, it should always be set to logic 0, due to errata. Please refer to Answer Database 14052 for more details.	integer
C_DISABLE_OPERAND_FORWARDING	This parameter will be used to drive the input pin TIEC405DISOPERANDFWD of processor block. The default value in MPD is 1.	integer
C_DSOCM_DCR_BASEADDR	The first 8 bit of this parameter will be used to drive the input pin TIEISOCMDCRADDR[0:7] of processor block. Bit [8:9] are not used. The default value in MPD is 0b0000100000	std_logic_vector (0 to 9)
C_ISOCM_DCR_BASEADDR	The first 8 bit of this parameter will be used to drive the input pin TIEISOCMDCRADDR[0:7]of processor block. Bit [8:9] are not used The default value in MPD is 0b0001000000	std_logic_vector (0 to 9)
C_DCR_RESYNC	This parameter has three settings: 0, 1 and 2. With the 0 setting no registers will be instantiated in the wrapper and the DCRCLK input signal is unused. The 1 setting only resynchronizes the control signals for read, write and acknowledge; in all 3 flip-flops. The 2 setting will instantiate resynchronisation registers on all DCR bus signals; all in all 74 flip-flops. The default value in MPD is 0, i.e. no resynchronization registers will be instantiated.	integer

DCR Resynchronization

When clocking the PowerPC 405 at 150MHz or higher, the designer can experience problems getting timing convergence in the place and route tool, due to setup violations in the DCR bus interface. The DCR bus slaves are normally clocked at 100MHz or less, however the PPC405 clocks its DCR interface registers at it's internal clock speed. Thus a PPC405 DCR master running at 300MHz that interfaces with DCR slaves running at 100MHz must achieve timing of <3.3 ns between master and slaves.

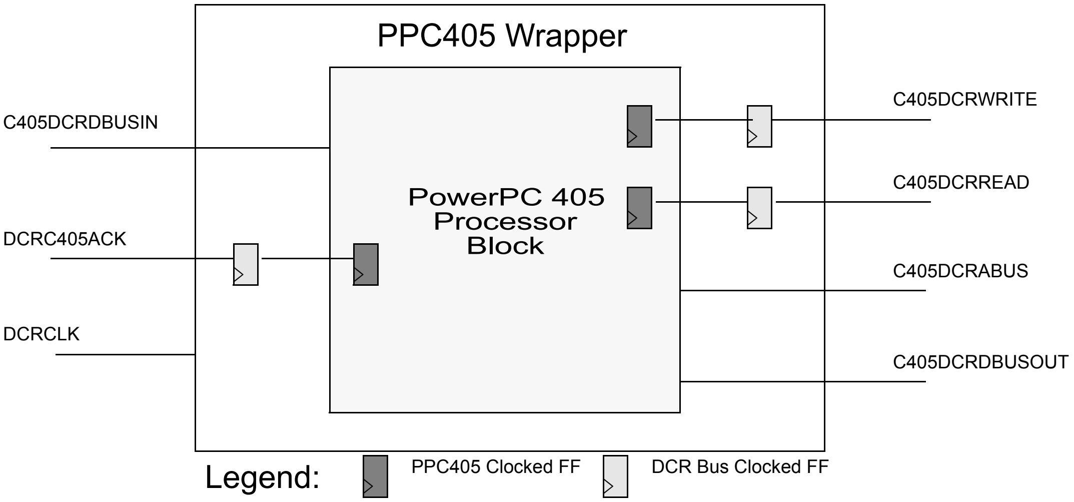
Fully Resynchronized DCR Interface

The resynchronisation registers in the PPC405 wrapper are used to solve this problem. If the C_DCR_RESYNC=2 or higher setting is used the wrapper will instantiate flip-flops on all DCR bus signals. The registers are clocked by the wrapper input DCRCLK, which typically should be the same clock signal used to clock the DCR slaves. By doing this the timing between the wrapper and slaves is constrained by this slower clock. The harder constraint based on the internal PPC405 frequency are only applied inside the wrapper. Since these registers can be placed in close proximity to the PPC405 Processor Block, it will be easier to meet timing without any additional user action. However the complete DCR interface requires 74 flip-flops placed close to the processor. This is a very high resource overhead as well as additional latency and may not be acceptable in a design.

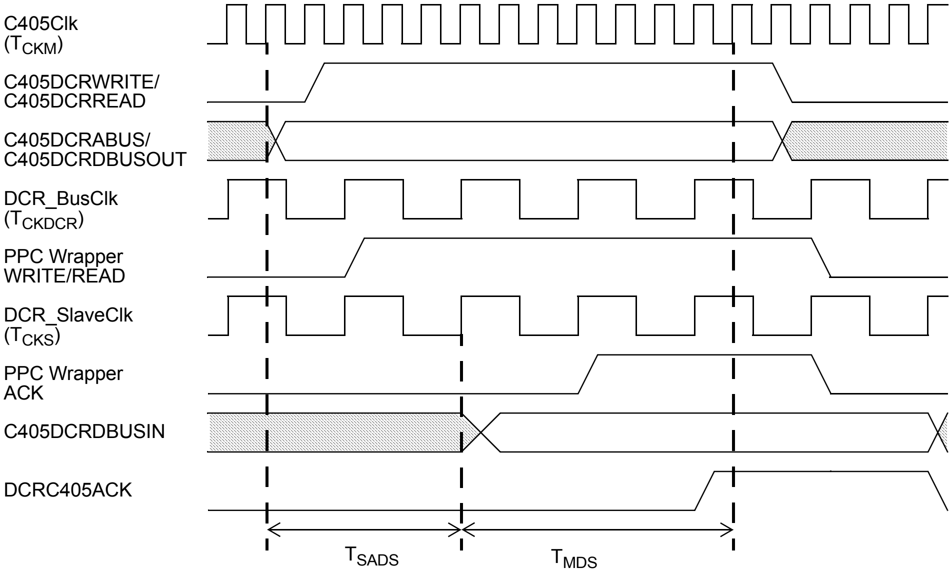
Partially Resynchronized DCR Interface

In the C_DCR_RESYNC=1 setting only the control signals: read, write and acknowledge are resynchronized. This will basically achieve the same function as with a fully registered DCR interface, but it can no longer be automatically analyzed by the static timing analyzer in the place and route tool. Thus it is up to the designer to provide the additional information in the form of UCF constraints to the tool. These constraints define all the non-registered signals in the interface as multi cycle paths. With this solution the resource overhead is much only 3 registers, but the latency increase remains the same.An example of what the UCF constraints would look like in this case is presented below:

Schematic



Timing Diagram



Time	Min	Max	Description
T_{SADS}	$2*T_{CKM}+T_{CKS}$	$T_{CKM}+T_{CKDCR}+T_{CKS}$	DCR slave address and data setup
T_{MDS}	$T_{CKS}+T_{CKDCR}+T_{CKM}$		DCR master data setup time

Example UCF Constraints

```
# Be sure to replace ‘ppc405_0’ in the constraints below with the PPC instance name in your HDL.
# Use KEEP to make sure names don’t disappear
# Make DataOut and Address 2 cycle paths
# Eliminate false path from DataOut to DataIn
# TS_PPC_Clk is assumed as the PowerPC clock period constraint
# The constant M = min($T_{SADS}$)/$T_{CKM}$
# The constant N = $T_{MDS}$/$T_{CKM}$
TIMEGRP SeqGrp = RAMS FFS LATCHES;
NET “ppc405_0/C405DCRABUS*”       KEEP | TPTHRU = MultiPPCaddr;
NET “ppc405_0/C405DCRDBUSOUT*”    KEEP | TPTHRU = MultiPPCdout;
NET “ppc405_0/DCRC405DBUSIN*”     KEEP | TPTHRU = MultiPPCdin;
TIMESPEC TS_MultiPPCaddr          = FROM CPUS THRU MultiPPCaddr TO FFS TS_PPC_Clk*M;
TIMESPEC TS_MultiPPCdout          = FROM CPUS THRU MultiPPCdout TO FFS TS_PPC_Clk*M;
TIMESPEC TS_MultiPPCdin           = FROM SeqGrp THRU MultiPPCdin TO CPUS TS_PPC_Clk*N;
TIMESPEC TS_FalsePPCloop          = FROM CPUS THRU MultiPPCdout THRU MultiPPCdin TO CPUS TIG;
```

Non Resynchronized DCR Interface

FInally if C_DCR_RESYNC is set to 0 , no registers are added in the wrapper. If timing can be met on the DCR bus using this setting (or the DCR bus is not used) then it is the preferred solution with no resource or latency overhead. For a detailed description of the timing constraints that can be used in this case, please refer to: Answer Record # 18146.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
12/17/03	1.0	Initial Xilinx release. EDK 6.2
1/15/04	1.1	Updated copyright and trademarks.