FUNCTION

Park and inverse Park transformation

File

park.vhd

Applicable Devices

Spartan 3A DSP, Spartan 6, Kintex 7, Zynq

Xilinx primitive used

BRAM, DSP48A

Sub modules used

sincos.vhd

Execution time

18 clocks for I_{sd} (outX[17:0]) result.
32 clocks for I_{sd} (outX[17:0]) an I_{sq} (outY[17:0]) results.
The same execution time is applicable for the inverse Park transformation.

Introduction

This IP implements a Park and the inverse Park transformation. Park transformation is used to rotate two currents referred to two fixed axes (α,β) into a frame attached to the rotor flux of a motor (d,q axes).

The Park transformation is:

$$I_{sd} = I_{\alpha}\cos\theta + I_{\beta}\sin\theta$$
$$I_{sq} = -I_{\alpha}\sin\theta + I_{\beta}\cos\theta$$

The inverse Park transformation is:

$$I_{\alpha} = I_{sd}\cos\theta - I_{sq}\sin\theta$$
$$I_{\beta} = -I_{sd}\sin\theta + I_{sq}\cos\theta$$

Where:
I_{α},I_{β} are the currents referred to (α,β) axes.
I_{sd},I_{sq} are the currents referred to (d,q) axes.
θ is the rotation angle.

Detailed Description

The Park transform is used to convert the α,β vectors coordinates from stator rotating frame to rotor fixed frame. Inverse Park converts rotor fixed frame to stator rotating frame.
This module can be used to perform the transformations on any physical component thus X and Y are the labels used in the IP for representing the components as pairs. Respectively the ***α*** or ***d*** maps in **X** and ***β*** or ***q*** maps in **Y.**
The ***direct*** selector is used to enable (direct=1) direct Park transform (stator to rotor) or to enable (direct=0) inverse Park transform (rotor to stator).
The vector coordinates (both inputs arguments and output results) are defined as SIGNED18 values. The valid range is -131072..131071.
The ***angle*** argument is defined as UNSIGNED27 value where: degrees = angle / 2^{27} * 360.
The function process starts with the ***start*** command. The end of process is notified to the consumer by the ***finish*** flag.

Some applications may require only the X component thus the ***doneX*** flag is on when ***outX*** result is ready.

Below the transforms are depicted. The Park transform makes a coordinate transformation to remap a vector from a rotating frame into a fixed frame.
The inverse Part makes a coordinate transformation to remap from a fixed frame to a rotating frame.
Both use an angle θ (theta) representing the amount of rotation.

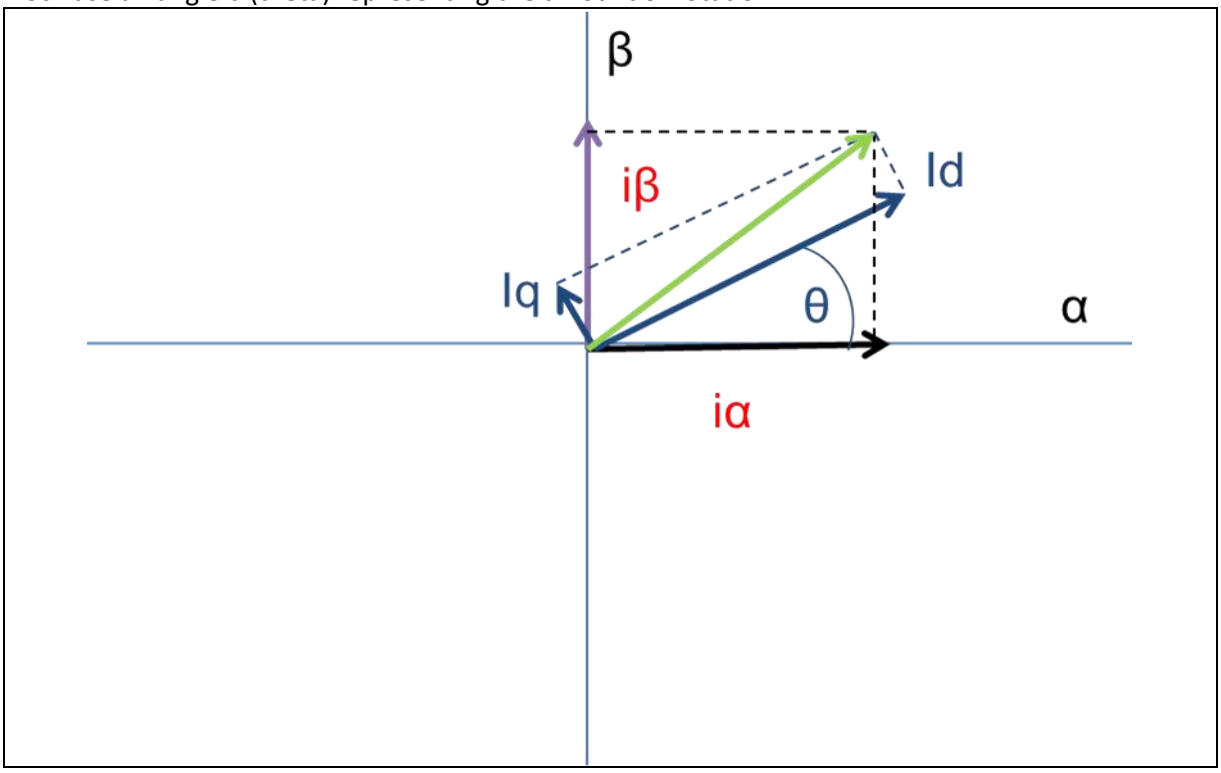

Figure 1: Park coordinates transformation

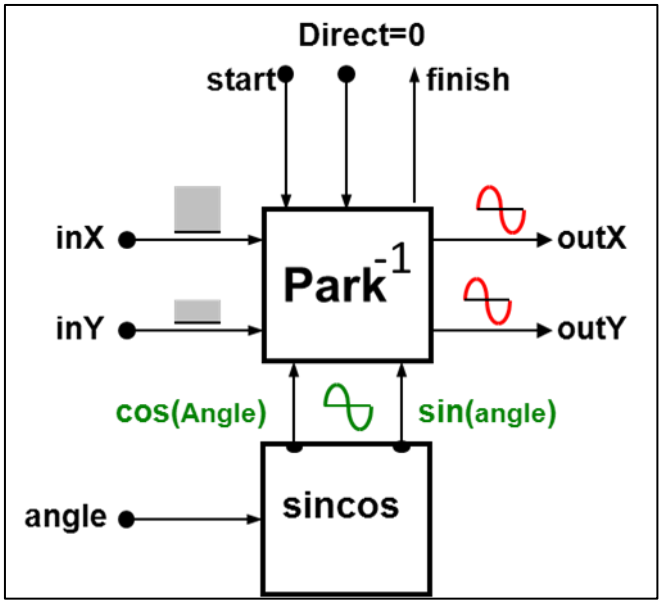

Figure 2: inverse park transformation

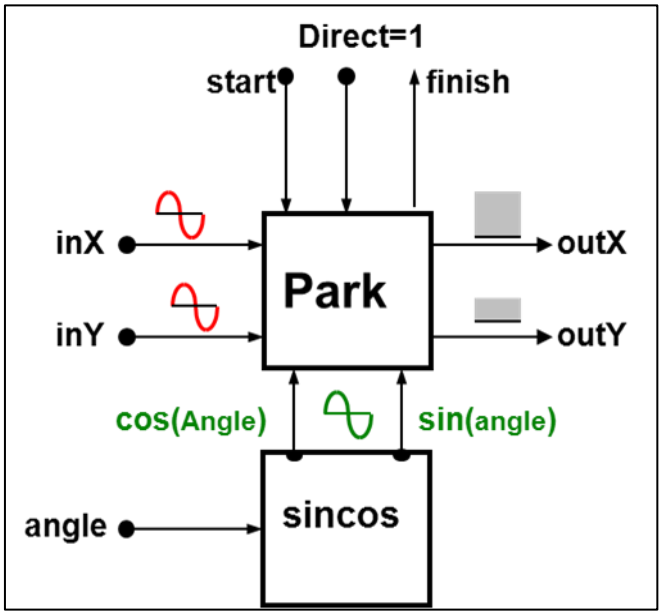

Figure 3: park transformation

PARAMETERS

Parameter	Type	Values	Default	Description
C_FAMILY	string	spartan3adsp spartan6 kintex7	spartan3adsp	Xilinx FPGA Family name

SIGNALS

Signal	I/O	Description
clock	IN	Clock (rising edge).
reset	IN	Reset. Active high.
start	IN	Start the elaboration. Active high. The pulse width must be of 1 clock.
direct	IN	Park transform selector. High=direct transform, low=inverse transform.
inX[17:0]	IN	I_α (direct) or I_{sd} (inverse) input current.
inY[17:0]	IN	I_β (direct) or I_{sq} (inverse) input current.
angle[26:0]	IN	θ rotation angle.
outX[17:0]	OUT	I_{sd} (direct) or I_α (inverse) output current.
outY[17:0]	OUT	I_{sq} (direct) or I_β (inverse) output current.
doneX	OUT	outX[17:0] data valid. Active high. The pulse is 1 clock width.
finish	OUT	End of process. OutX[17:0] and OutY[17:0] are valid. Active high. The pulse width is of 1 clock.

TIMING PERFORMANCE AND RESOURCE USAGE

This section provides data on the timing performance and resource utilization of the core. Performance has been obtained on one representative device from the Spartan-3 Generation and Spartan 6 families of FPGAs. The following tables lists the devices used for characterization.

Device Utilization

Device Utilization Summary (estimated values)	
Logic Utilization	**Spartan3A DSP**
Number of Slices	107
Number of Slice Flip Flops	147
Number of 4 input LUTs	148
Number of BRAMs	1
Number of DSP48s	2

Device Utilization Summary (estimated values)	
Logic Utilization	**Spartan 6**
Number of Slice Registers	132
Number of Slice LUTs	76
Number of fully used LUT-FF pairs	64
Number of Block RAM/FIFO	1
Number of DSP48A1s	2

Device Utilization Summary (estimated values)	
Logic Utilization	**Kintex 7**
Number of Slice Registers	132
Number of Slice LUTs	116
Number of fully used LUT-FF pairs	68
Number of Block RAM/FIFO	1
Number of DSP48E1s	2

Execution time

output	input	clock cycle[1]
doneX	start	18
finish	start	32

[1] Unless otherwise noted.

Timing[2]

In the figure below there are the timings relative to a whole start/finish cycle of calculation. Note that when the process is running the signals **direct**, **inX**, **inY** and **angle** must stay stable.

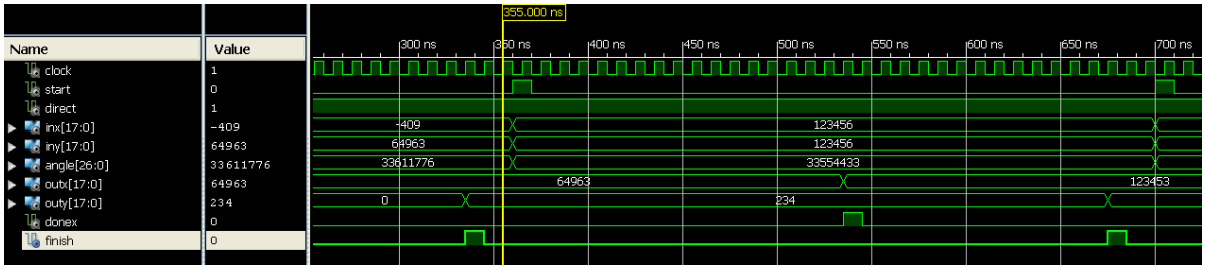

Figure 4: timings of a whole start/finish cycle

The process begins on the rising edge of the clock when the **start** signal is high; the **direct**, **inX**, **inY** and **angle** must be stable when **start** is set high and they must stay stable until the **finish** signal is set high.

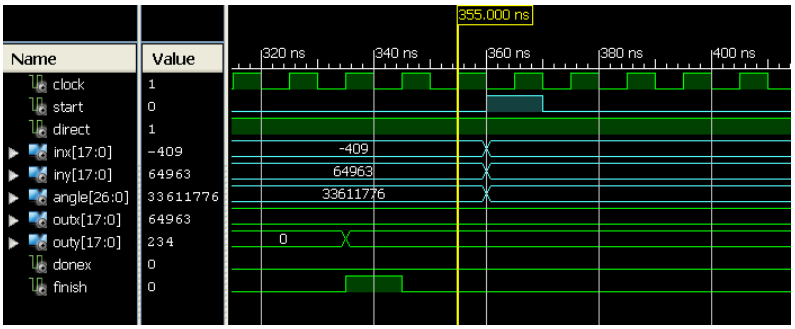

*Figure 5: timings for **start**, **inX**, **inY** and **angle** signal*

The **doneX** (see Figure 6) signal is set high for 1 clock cycle when the **outX** data is ready. In this way an application that needs only **outX** data, can read the data without wait for the end of the process.

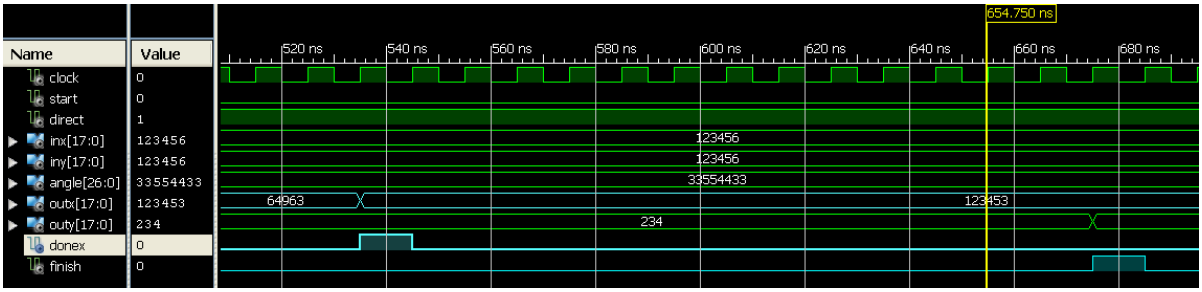

*Figure 6: **doneX** and **outX** signals*

[2] The clock period is only chosen with the purpose to draw the waveforms.

The **finish** signal is set high when the process terminates and both **outX** and **outY** are ready. It stays high for 1 clock cycle and it can be used by the consumer of the **outX** and **outY** data to synchronize with the process. The **outX** and **outY** signals are valid when the **finish** signal becomes high, and they stay valid until a new **start** signal is detected by the process.

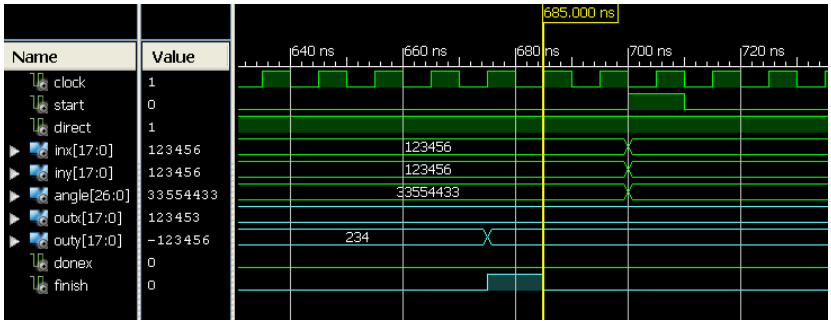

*Figure 7: timings for **finish**, **outX** and **outY** signals*

The **reset** signal resets the process and sets all the output signals to 0 until **reset** is set high.

Reference Documents

1. Xilinx LogiCORE IP DSP48 Macro V2.1 [DS754 March 1, 2011]

Support

QDESYS provides technical support for this LogiCORE product when used as described in the product documentation.
QDESYS cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled DO NOT MODIFY.

Ordering Information

For information on pricing and availability of QDESYS modules and software, please contact info@qdesys.com

Revision History

Date	Version	Description
07/07/2011	1.0	Initial QDeSys release.
18/11/2011	1.1	Modified Devices Utilization table.
22/12/2011	1.2	Added parameters for FPGA family
12/05/2012	1.3	Added Kintex 7 and Zynq support

Disclaimer

In disclosing the information contained in this document QDeSys assumes no obligation to correct any errors herein contained, or to advise you of any corrections or updates. QDeSys expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the information.
THE DOCUMENTATION IS DISCLOSED TO YOU "AS-IS" WITH NO WARRANTY OF ANY KIND. QDESYS MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT WILL QDESYS BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION