Park and Inverse Park Transformations Hardware Implementation

User Guide

Microsemi

Table of Contents

Park and Inverse Park Transforms Theory

The behavior of three-phase machines is usually described by their voltage and current equations. The coefficients of the differential equations that describe their behavior are time varying (except when the rotor is stationary). The mathematical modeling of such a system tends to be complex since the flux linkages, induced voltages, and currents change continuously as the electric circuit is in relative motion. For such a complex electrical machine analysis, mathematical transformations are often used to decouple variables and to solve equations involving time varying quantities by referring all variables to a common frame of reference.

Park Transformation

Park transformation transforms the orthogonal stationary reference frame (α-β reference frame) quantities, obtained from the Clarke transformation applied on three-phase quantities, into rotating reference frame (d-q reference frame) as shown in Figure 1.

The Park transformation is expressed by the following equations:

$$I_d = I_\alpha * \cos(\theta) + I_\beta * \sin(\theta)$$

EQ1

$$I_q = I_\beta * \cos(\theta) - I_\alpha * \sin(\theta)$$

EQ2

where,

I_d and I_q are rotating reference frame quantities

I_α and I_β are orthogonal stationary reference frame quantities

θ is the rotation angle

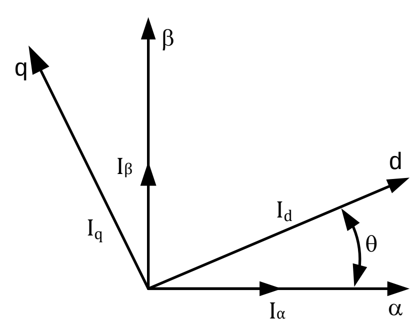

Figure 1 · Park Transformation

Inverse Park Transformation

The quantities in rotating reference frame are transformed to two-axis orthogonal stationary reference frame using Inverse Park transformation as shown in Figure 2.

The Inverse Park transformation is expressed by the following equations:

$$V_\alpha = V_d * \cos(\theta) - V_q * \sin(\theta) \qquad \textit{EQ3}$$

$$V_\beta = V_q * \cos(\theta) + V_d * \sin(\theta) \qquad \textit{EQ4}$$

where,

V_α and V_β are orthogonal stationary reference frame quantities

V_d and V_q are rotating reference frame quantities

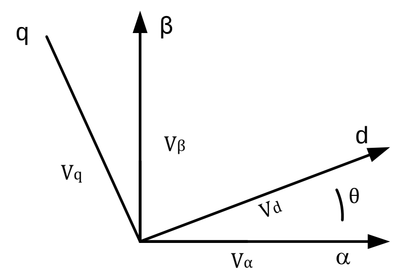

Figure 2 · Inverse Park Transformation

Park Transform Hardware Implementation

This section describes the hardware implementation and the internal configuration of the Park Transform implemented on SmartFusion2.

Park Transformation Implementation

The system level block diagram of the Park transformation implemented is shown in Figure 3.

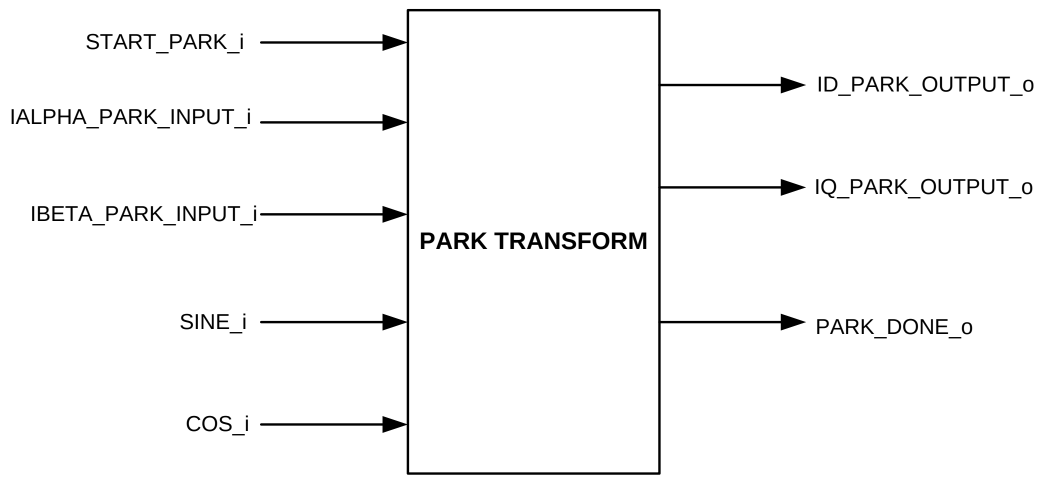

Figure 3 · System Level Block Diagram of Park Transformation

The above block implements the following equations:

$$ID\_o = I\_ALPHA\_i * COS\_i + I\_BETA\_i * SINE\_i \tag{EQ5}$$

$$IQ\_o = I\_BETA\_i * COS\_i - I\_ALPHA\_i * SINE\_i \tag{EQ6}$$

where,

I_ALPHA_i and I_BETA_i are orthogonal stationary reference frame current components COS_i and Sine_i are cos(θ) and sin(θ) values, respectively

ID_o and IQ_o are rotating reference frame current components (I_d and I_q current components, respectively)

Figure 4 shows the implementation of Park transformation. The Park transformation block uses MAS block, which performs basic operations like multiplication, addition, and subtraction, for the computation of EQ5 and EQ6.

The START_PARK_i signal must undergo a LOW to HIGH transition to accept new inputs and compute the corresponding output. The PARK_DONE_o output signal goes HIGH when the computations are completed and output is obtained. Once a set of inputs are given and the transformation process has already started, no new input will be accepted before the PARK_DONE_o output signal goes HIGH, even if the START signal undergoes LOW to HIGH transition.

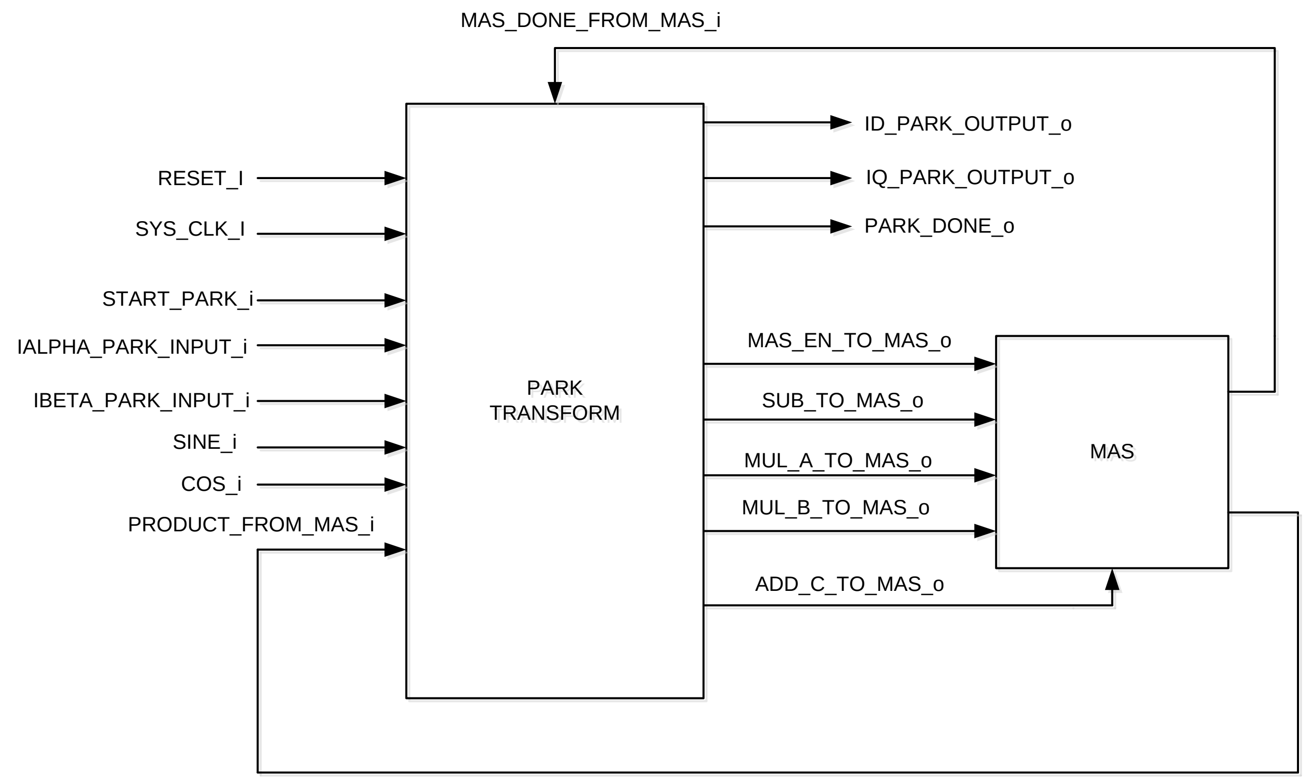

Figure 4 · Park Transformation Implementation

The SINE_i and COS_i inputs are sin and cos values respectively obtained from a RAM block available on board. The inputs, IALPHA_PARK_INPUT_i and IBETA_PARK_INPUT_i, are obtained from the Clarke transformation block.

Inputs and Outputs of Park Transformation Block

The description of input and output ports of the Park transformation block is listed in Table 1.

Table 1 · Input and Output Ports of Park Transformation

Signal Name	Direction	Description
RESET_I	Input	Asynchronous reset signal to design. Active state is defined by RESET_STATE (configuration parameter)
SYS_CLK_I	Input	System clock
IALPHA_PARK_INPUT_i	Input	Current component in stationary orthogonal reference frame on alpha axis
IBETA_PARK_INPUT_i	Input	Current component in stationary orthogonal reference frame on beta axis
COS_i	Input	Cosine component of electrical angle is held in this register
SINE_i	Input	Sine component of electrical angle is held in this register
START_PARK_i	Input	Start signal for the park function
MAS_DONE_FROM_MAS_i	Input	Done signal from MAS block indicating that computations by the MAS block are complete
PRODUCT_FROM_MAS_i	Input	Product from the MAS block
ID_PARK_OUTPUT_o	Output	Direct axis current component in rotor reference frame (Id)
IQ_PARK_OUTPUT_o	Output	Quadrature axis current component in rotor reference frame (Iq)
PARK_DONE_o	Output	Signal indicating the Park transformation is completed
MAS_EN_TO_MAS_o	Output	Enable signal to the MAS block
SUB_TO_MAS_o	Output	Signal when goes HIGH indicates MAS block to perform subtraction.
MUL_A_TO_MAS_o	Output	Operand for multiplication by the MAS block
MUL_B_TO_MAS_o	Output	Operand for multiplication by the MAS block
ADD_C_TO_MAS_o	Output	Carry input to the MAS block

Configuration Parameters of Park Transformation Block

Table 2 lists and describes the configuration parameters used in the hardware implementation of Park transformation block. These are generic parameters and can be varied as per the requirement of the application.

Table 2 · Configuration Parameters of Park Transformation Block

Name	Description
g_RESET_STATE	When 0, supports active LOW reset When 1, supports active HIGH reset
g_SINE_COS_WIDTH	Defines the bit length of the SINE_i, COS_i registers
g_I_ALPHA_BETA_WIDTH	Defines the bit length of the IALPHA_PARK_INPUT_i and IBETA_PARK_INPUT_i registers
MUL_A_WIDTH	Defines the bit length of one of the operands to the MAS block for multiplication
MUL_B_WIDTH	Defines the bit length of one of the operands to the MAS block for multiplication
ADD_C_WIDTH	Defines the bit length of carry input to the MAS block

Park Transformation Block FSM Implementation

The finite state machine (FSM) of the Park transformation block is as shown in Figure 5.

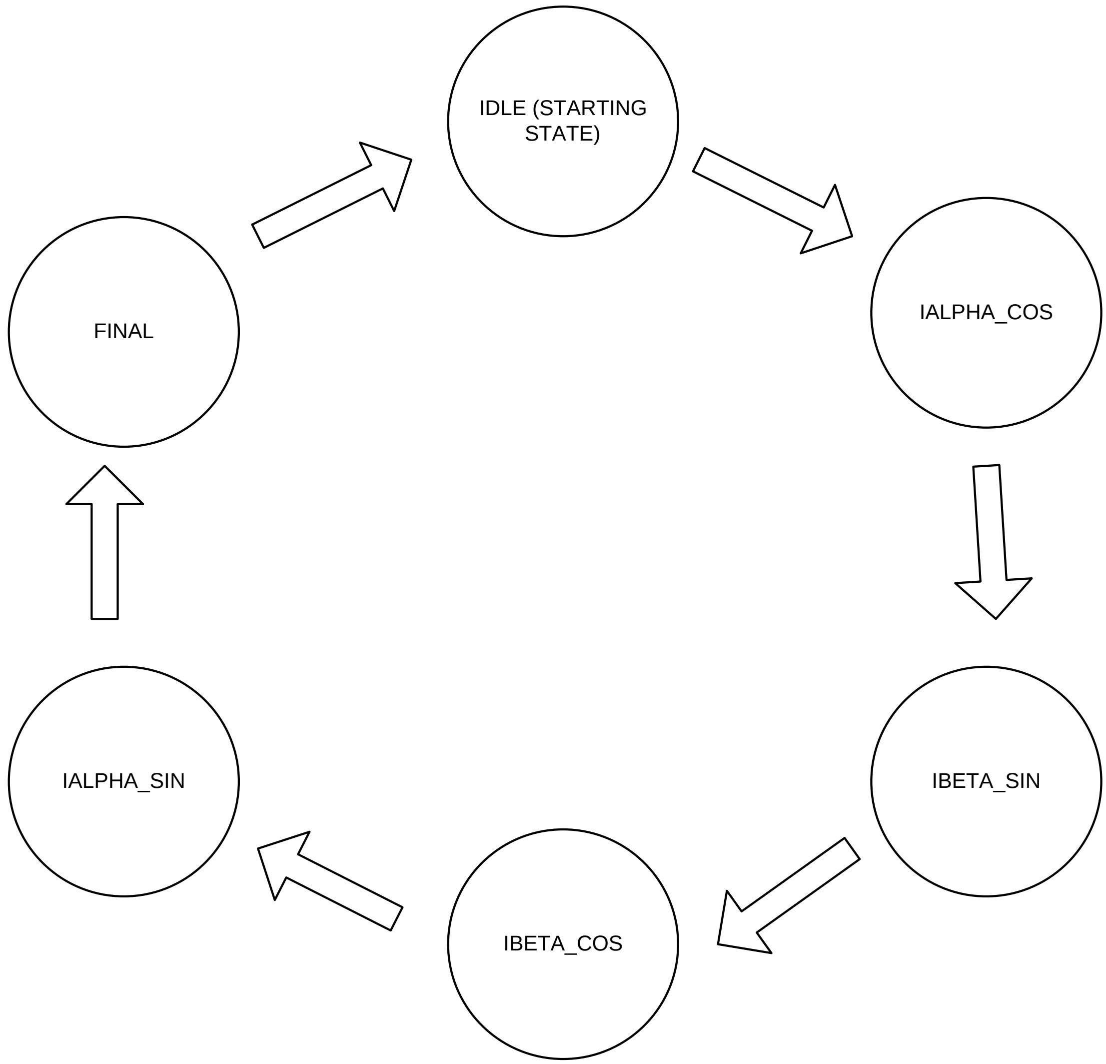

Figure 5 · Park Transformation Finite State Machine

Following are the six states in the FSM of the Park transformation block:

- IDLE
- IALPHA_COS
- IBETA_SIN
- IBETA_COS
- IALPHA_SIN
- FINAL

The FSM is synchronized to the rising-edge of the clock.

IDLE state: This is the initial state of the FSM. The FSM moves to this state when a reset signal is given to the system or when the computations corresponding to the given inputs are completed and output is obtained. The FSM moves to IALPHA_COS state in the next clock cycle, when a rising-edge on the START_PARK_i input signal is detected.

IALPHA_COS state: In this state, the MAS block is enabled and the IALPHA_PARK_INPUT_i and COS_i (inputs) are given to the MAS block for multiplication. The FSM moves to IBETA_SIN state in the next clock cycle.

IBETA_SIN state: The FSM remains in this state until the done signal (MAS_DONE_FROM_MAS_i) of the MAS block goes HIGH, indicating that the computation of previous state is completed. After the done signal from the MAS block goes HIGH, IBETA_PARK_INPUT_i and SINE_i (inputs) are given to the MAS block for multiplication. The FSM moves to IBETA_COS state in the next clock cycle.

IBETA_COS state: The FSM remains in this state until the done signal of the MAS block goes HIGH, indicating that the computation of previous state is completed. After the done signal from the MAS block goes HIGH, IBETA_PARK_INPUT_i and COS_i are given to the MAS block for multiplication. The FSM moves to IALPHA_SIN state in the next clock cycle.

IALPHA_SIN state: The FSM remains in this state until the done signal of the MAS block goes HIGH, indicating that the computation of previous state is completed. After the done signal from the MAS block goes HIGH, IALPHA_PARK_INPUT_i and SINE_i are given to the MAS block for multiplication. The FSM moves to the FINAL state in the next clock cycle.

FINAL state: The FSM remains in this state until the done signal of the MAS block goes HIGH, indicating that the computation of previous state is completed. After the done signal from the MAS block goes HIGH, the PARK_DONE_o signal is changed to high (reflected in the next clock cycle) indicating that the Park transformation is completed as shown in Figure 6.

Timing Diagram of Park Transformation Block

The timing waveform of the Park transformation block is shown in Figure 6.

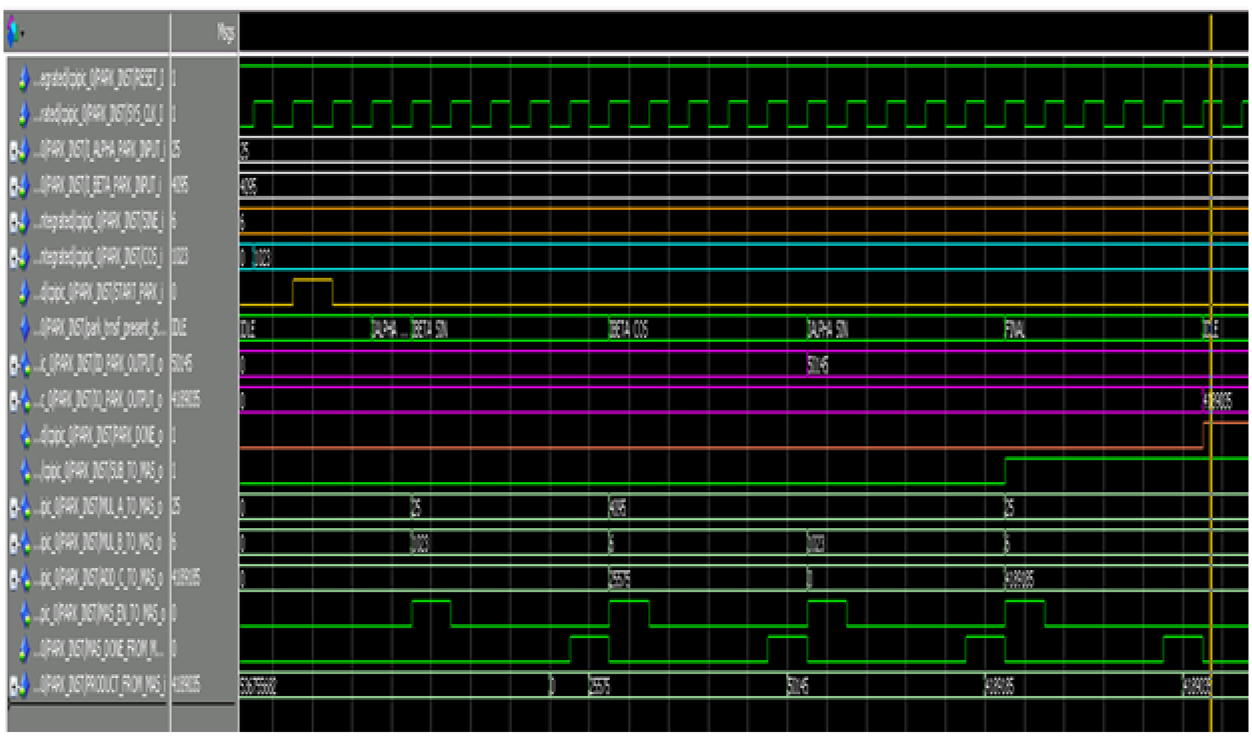

Figure 6 · Park Transformation Timing Diagram

Colour code:

- Yellow: START_PARK_i
- WHITE: I_ALPHA_PARK_INPUT_i, I_BETA_PARK_INPUT_i (inputs)
- Gold: SINE_i
- Cyan: COS_i
- Purple: ID _PARK_OUTPUT_o, IQ_PARK_OUTPUT_o (outputs)
- Brown: PARK_DONE_o

Resource Utilization of Park Transformation Block

The resource utilization of Park transformation implemented on SmartFusion2 device is listed in Table 3.

Table 3 · Resource Utilization of Park Transformation Block

Resource Usage Report for Park	
Cell Usage	**Description**
CLKINT	2 uses
CFG2	6 uses
CFG3	3 uses
CFG4	58 uses
Carry primitives used for arithmetic functions	
ARI1	19 uses
Sequential Cells	
SLE	198 uses
Latch bits not including I/Os	198 (0%)
DSP Blocks	1
MACC	1 MultAdd
I/O ports	148
I/O primitives	148
INBUF	75 uses
OUTBUF	73 uses
Global Clock Buffers	2
Total LUTs	67

Inverse Park Transform Hardware Implementation

This section describes the hardware implementation and the internal configuration of the Inverse Park transform implemented on the SmartFusion2 device.

Inverse Park Transformation Implementation

The system level block diagram of the Inverse Park transformation implemented is shown in Figure 7.

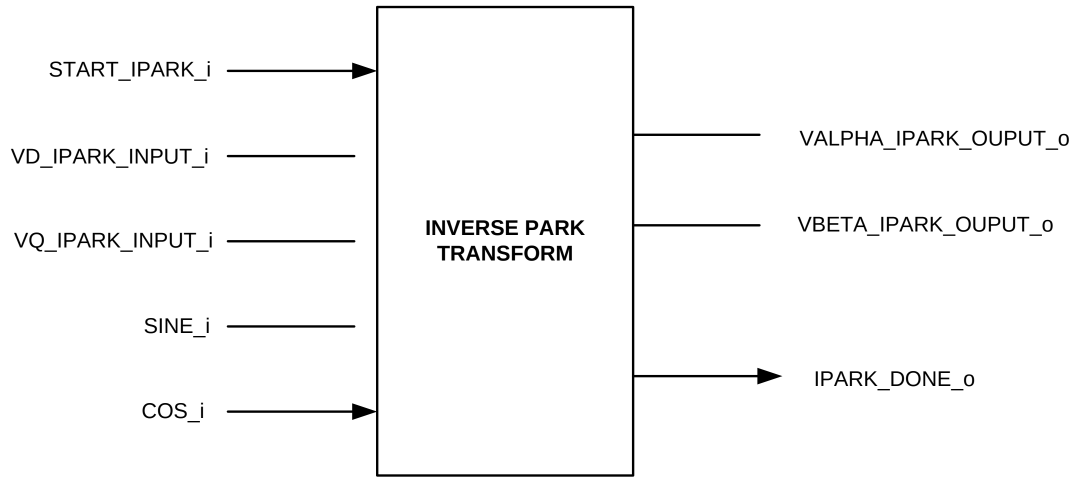

Figure 7 · System Level Block Diagram of Inverse Park Transformation

The above block implements the following equations:

$$V\_ALPHA\_o = VD\_i * COS\_i - VQ\_i * SINE\_i$$

EQ7

$$V\_BETA\_o = VD\_i * COS\_i - VQ\_i * SINE\_i$$

EQ8

where,

VD_i and VQ_i are rotating reference frame voltage components

COS_i and Sine_i are cos(θ) and sin(θ) values respectively

V_ALPHA_o and V_BETA_o are orthogonal stationary reference frame voltage components

The implementation of the Inverse Park transformation equations is done as shown in Figure 8. The Inverse Park transformation block uses the MAS block, which performs basic operations like multiplication, addition, and subtraction, for the computation of EQ7 and EQ8.

The START signal must undergo a LOW to HIGH transition to accept new inputs and compute the output. The IPARK_DONE_o goes HIGH when the computations are completed and output is obtained. Once a set of inputs are given and the transformation process begins, no new input will be accepted before the IPARK_DONE_o output signal goes HIGH, even if the START signal changes state from LOW to HIGH.

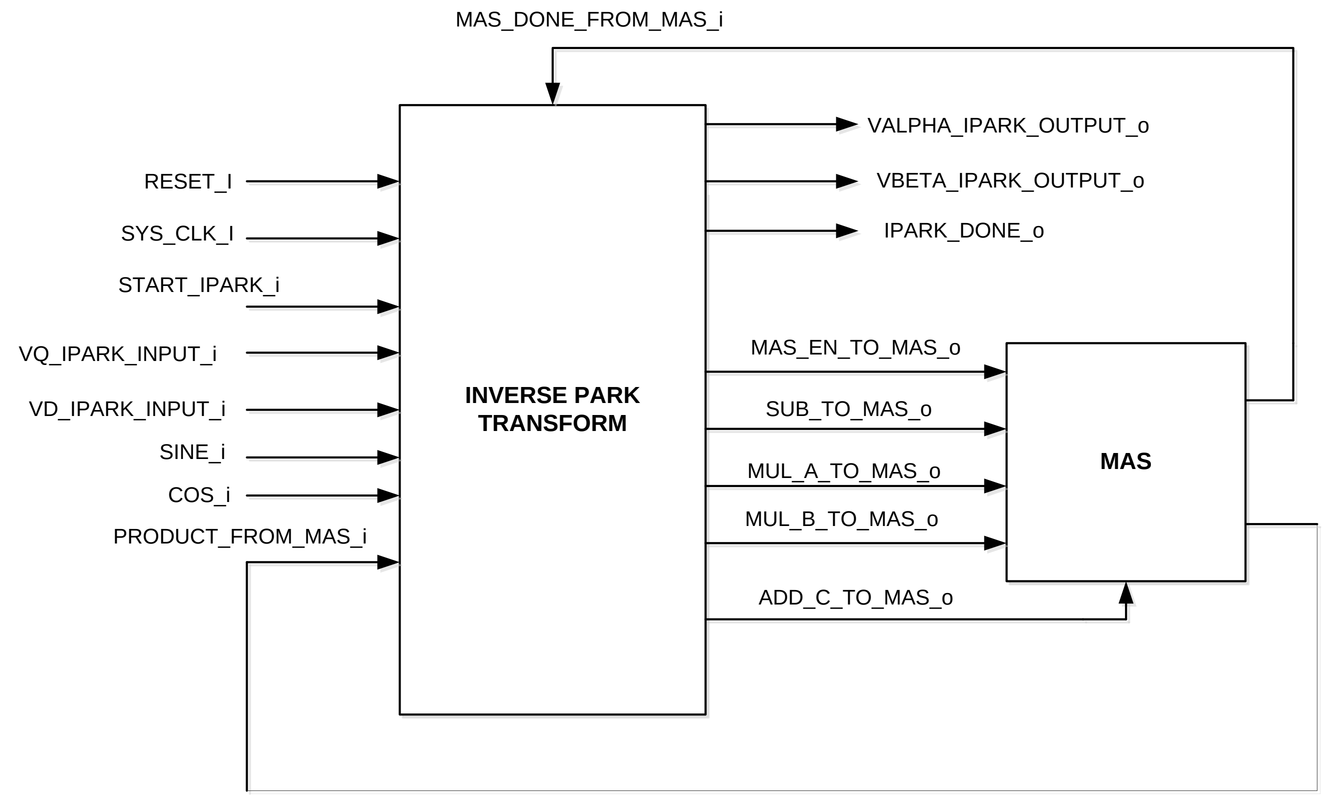

Figure 8 · Inverse Park Transformation Implementation

Inputs and Outputs of Inverse Park Transformation Block

The description of input and output ports of Inverse Park transformation block is listed in Table 4.

Table 4 · Input and Output Ports of Inverse Park Transformation

Signal Name	Direction	Description
RESET_I	Input	Asynchronous reset signal to design. Active state is defined by RESET_STATE.
SYS_CLK_I	Input	System clock
VD_IPARK_INPUT_i	Input	Direct axis voltage component in rotor reference frame (Vd)
VQ_IPARK_INPUT_i	Input	Quadrature axis voltage component in rotor reference frame (Vq)
COS_i	Input	Cosine component of electrical angle is held in this register
SINE_i	Input	Sine component of electrical angle is held in this register
START_IPARK_i	Input	Start signal for the Inverse park function
MAS_DONE_FROM_MAS_i	Input	Done signal from the MAS block indicating that computations by the MAS block are complete
PRODUCT_FROM_MAS_i	Input	Product from the MAS block
VALPHA_IPARK_OUTPUT_o	Output	Voltage component in stationary orthogonal reference frame

Signal Name	Direction	Description
		(Valpha)
VBETA_IPARK_OUTPUT_o	Output	Voltage component in stationary orthogonal reference frame (Vbeta)
IPARK_DONE_o	Output	Signal indicating that the Inverse Park transformation is completed
MAS_EN_TO_MAS_o	Output	Enable signal to the MAS block
SUB_TO_MAS_o	Output	When this signal when goes high, it indicates that the MAS block is to perform subtraction.
MUL_A_TO_MAS_o	Output	Operand for multiplication by the MAS block
MUL_B_TO_MAS_o	Output	Operand for multiplication by the MAS block
ADD_C_TO_MAS_o	Output	Carry input to the MAS block

Configuration Parameters of Inverse Park Transformation Block

Table 5 lists and describes the configuration parameters used in the hardware implementation of Inverse Park transformation block. These are generic parameters and can be varied as per the requirement of the application.

Table 5 · Configuration Parameters of Inverse Park Transformation Block

Name	Description
RESET_STATE	When 0, supports active LOW reset When 1, supports active HIGH reset
g_SINE_COS_WIDTH	Defines the bit length of the SINE_i and COS_i registers
g_I_VD_VQ_WIDTH	Defines the bit length of the VD_IPARK_INPUT_i and VQ_IPARK_INPUT_i registers
MUL_A_WIDTH	Defines the bit length of one of the operands to the MAS block for multiplication
MUL_B_WIDTH	Defines the bit length of one of the operands to the MAS block for multiplication
ADD_C_WIDTH	Defines the bit length of carry input to the MAS block

Inverse Park Transformation Block FSM Implementation

The FSM of the Inverse Park transformation block is as shown in Figure 9.

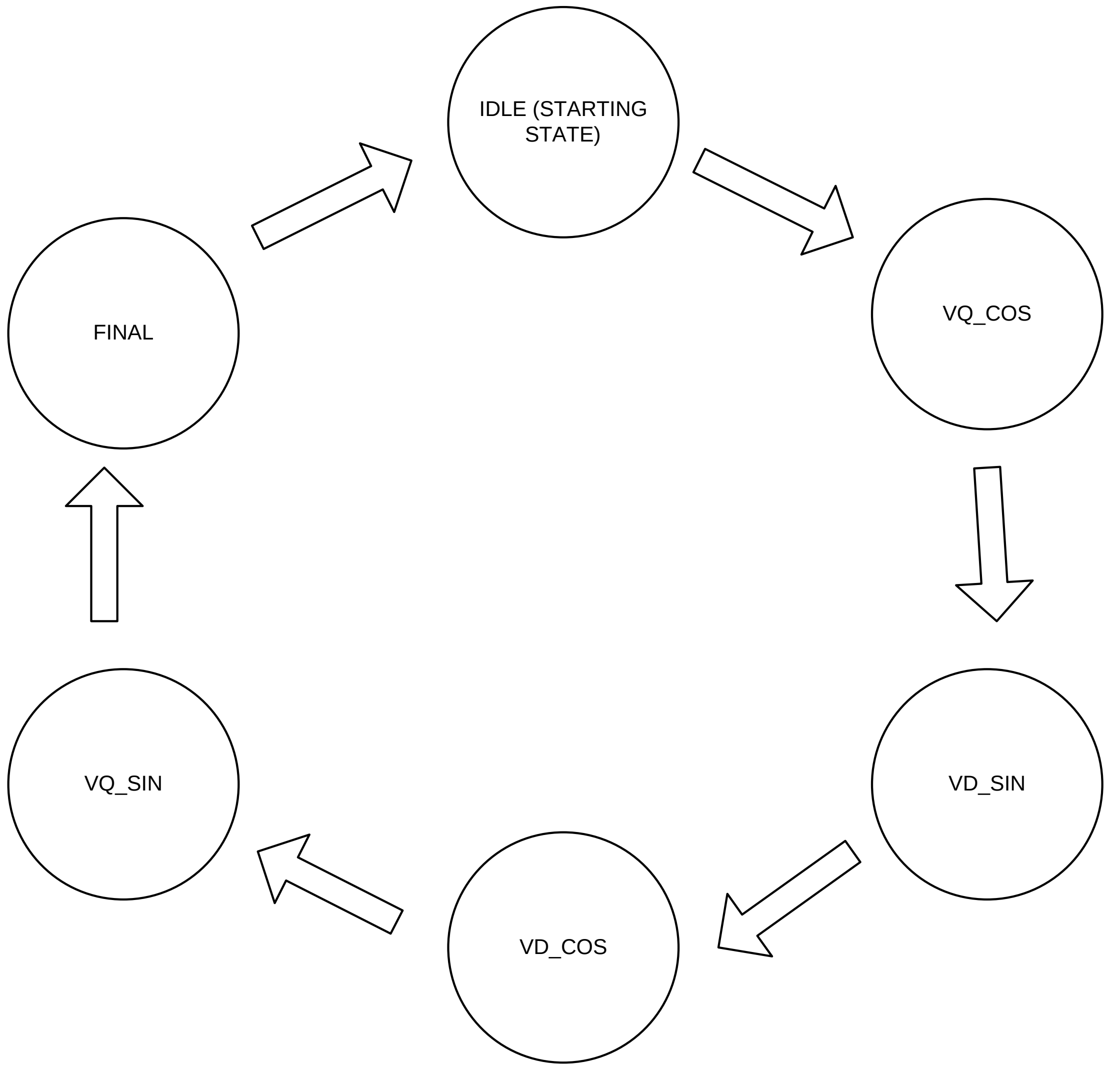

Figure 9 · Inverse Park Transformation FSM

Following are the six states in the FSM of the Park transformation block:

- IDLE
- VQ_COS
- VD_SIN
- VD_COS
- VQ_SIN
- FINAL

The FSM is synchronized to the rising-edge of the clock.

IDLE state: This is the initial state of the FSM. The FSM moves to this state when a reset signal is given to the system or when the computations corresponding to the given inputs are completed and output is obtained. The FSM moves to VQ_COS state in the next clock cycle, when a rising-edge on the START_IPARK_i input signal is detected.

VQ_COS state: In this state the MAS block is enabled, and VQ_IPARK_INPUT_i and COS_i (inputs) are given to the MAS block for multiplication. The FSM moves to VD_SIN state in the next clock cycle.

VD_SIN state: The FSM remains in this state until the done signal of the MAS block goes HIGH, indicating that the computation of previous state is completed. After the done signal (MAS_DONE_FROM_MAS_i) from the MAS block goes HIGH, VD_IPARK_INPUT_i and SINE_i (inputs) are given to the MAS block for multiplication. The FSM moves to VD_COS state in the next clock cycle.

VD_COS state: The FSM remains in this state until the done signal of the MAS block goes HIGH, indicating that the computation of previous state is completed. After the done signal from the MAS block goes HIGH, VD_IPARK_INPUT_i and COS_i are given to the MAS block for multiplication. The FSM moves to VQ_SIN state in the next clock cycle.

VQ_SIN state: The FSM remains in this state until the done signal of the MAS block goes HIGH, indicating that the computation of previous state is completed. After the done signal from the MAS block is goes HIGH, VQ_IPARK_INPUT_i and SINE_i are given to the MAS block for multiplication. The FSM moves to FINAL state in the next clock cycle.

FINAL state: The FSM remains in this state until the done signal of the MAS block goes HIGH, indicating that the computation of previous state is completed. After the done signal from the MAS block is goes HIGH, the IPARK_DONE_o signal is changed to high (reflected in the next clock cycle) indicating that the Park transformation is completed as shown in Figure 10.

Timing Diagram of Inverse Park Transformation Block

The Inverse Park transformation block takes 9 clock cycles to compute the complete output as shown in Figure 10.

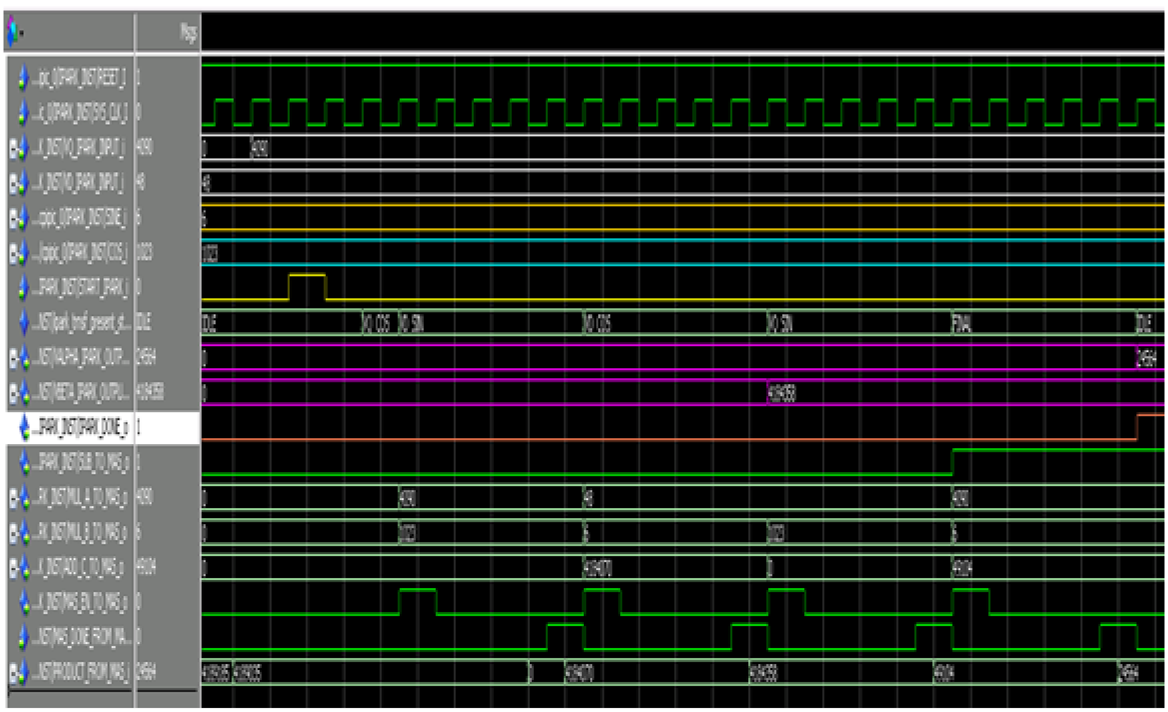

Figure 10 · Inverse Park Transformation Timing Diagram

Colour code:

- Yellow: START_IPARK_i
- WHITE: ID_IPARK_INPUT_i, IQ_IPARK_INPUT_i (inputs)
- Gold: SINE_i
- Cyan: COS_i
- Purple: VALPHA_IPARK_OUTPUT_o, VBETA_IPARK_OUTPUT_o (outputs)
- Brown: IPARK_DONE_o

Resource Utilization of Inverse Park Transformation Block

The resource utilization of Inverse Park transformation implemented on the SmartFusion2 device is listed in Table 6.

Table 6 · Resource Utilization of Inverse Park Transformation Block

Resource Usage Report for Inverse_Park	
Cell Usage	**Description**
CLKINT	2 uses
CFG2	6 uses
CFG3	3 uses
CFG4	58 uses
Carry primitives used for arithmetic functions	
ARI1	19 uses
Sequential Cells	
SLE	198 uses
Latch bits not including I/Os	198 (0%)
DSP Blocks	1
MACC	1 MultAdd
I/O ports	148
I/O primitives	148
INBUF	75 uses
OUTBUF	73 uses
Global Clock Buffers	2
Total LUTs	67

Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call **800.262.1060**
From the rest of the world, call **650.318.4460**
Fax, from anywhere in the world **408.643.6913**

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Microsemi SoC Products Group Customer Support website for more information and support (http://www.microsemi.com/soc/support/search/default.aspx). Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on website.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page, at http://www.microsemi.com/soc/.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

50200474-0/11.13