UG0607
User Guide
FOC Transformations v4.1

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

50200607. 3.0 10/16

Contents

Figures

Tables

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 3.0

The following is a summary of the changes in revision 3.0 of this document.

- Added the IP version to the document title.
- Removed g_STD_IO_WIDTH configuration parameter from sections Configuration Parameter, page 7 and Resource Utilization, page 8.

1.2 Revision 2.0

Updated SAR 71203.

1.3 Revision 1.0

Revision 1.0 was the first publication of this document.

2 Introduction

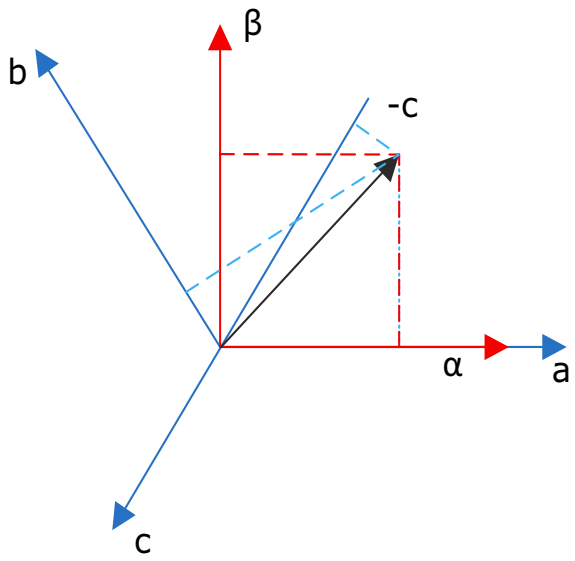
In a three-phase motor, the sum of three phase voltages or currents at any instant is equal to zero. Hence, it is sufficient to have only two currents or voltages to represent the motor behavior. However, the actual three-phase quantities are displaced by 120° and therefore, they are not completely decoupled. In order to decouple the two-phases that are used to represent the motor dynamics, Clarke transformation is applied to a, b, c phases to transform them to alpha–beta vectors. Clarke transform preserves the magnitude vectors while transforming them from three-phase a, b, c to two-phase alpha–beta. Similarly, inverse Clarke transformation is used to convert alpha–beta to a, b, c components because the motor needs actual three-phase voltages to be applied to its stator terminals.

When the motor is rotating, alpha–beta quantities will appear as sinusoidal quantities whose frequency depends on the speed of the motor and the number of motor poles. It is easier to control DC quantities than time varying sinusoidal quantities, because of which, Park transformation is used to transform time varying alpha–beta vectors to constant d–q vectors. This is called transforming from stator reference frame to rotor reference frame. After the d–q vectors are controlled to get required motor dynamics, they have to be transformed back to stator reference frame using Inverse Park transformation.

2.1 Clarke Transformation

The three-phase quantities are translated from the three-phase reference frame to the two-axis orthogonal stationary reference frame using Clarke transformation, as shown in the following figure.

Figure 1 • **Clarke and Inverse Clarke Computation**



The general equation for Clarke transformation is:

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \frac{2}{3}\begin{bmatrix}1 & -\frac{1}{2} & -\frac{1}{2}\\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2}\end{bmatrix}\begin{bmatrix}a\\ b\\ c\end{bmatrix}$$

$$\alpha = \frac{2}{3}a - \frac{1}{3}b - \frac{1}{3}c$$

$$\beta = \frac{1}{\sqrt{3}}b - \frac{1}{\sqrt{3}}c$$

Where,

a, b, and c are three-phase quantities and α and ß are two-phase orthogonal quantities.

Using a + b + c = 0, the equations can be simplified:

$$\alpha = a$$

$$\beta = \frac{a + 2b}{\sqrt{3}}$$

2.2 Inverse Clarke Transformation

The transformation from a two-axis orthogonal stationary reference frame to a three-phase stationary reference frame is accomplished using Inverse Clarke transformation, as shown in Figure 1, page 2. The Inverse Clarke transformation is expressed by the following equations:

$$\begin{bmatrix}a\\ b\\ c\end{bmatrix} = \frac{2}{3}\begin{bmatrix}1 & 0\\ -\frac{1}{2} & \frac{\sqrt{3}}{2}\\ -\frac{1}{2} & -\frac{\sqrt{3}}{2}\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$$

$$a = \alpha$$

$$b = -\frac{1}{2}\alpha + \frac{\sqrt{3}}{2}\beta$$

$$c = -\frac{1}{2}\alpha - \frac{\sqrt{3}}{2}\beta$$

2.3 Park Transformation

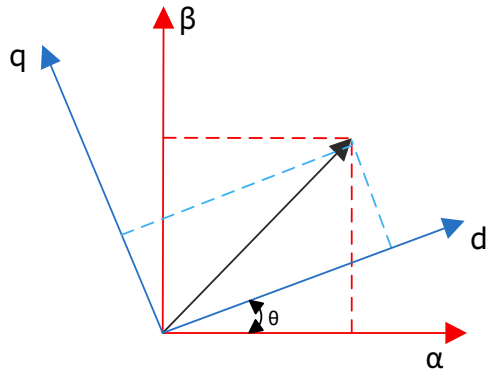
Park transformation, transforms quantities from stationary reference frame to rotating reference frame. As the rotor rotates, the rotor flux rotates in space and it is the position of the flux vector that is taken as reference for transformation. The equations that define Park transformation are:

$$d = \alpha \times \cos\theta + \beta \times \sin\theta$$

$$q = -\alpha \times \sin\theta + \beta \times \cos\theta$$

Where e is rotor flux angular position, as shown in the following figure.

Figure 2 • **Park and Inverse Park Computation**



2.4 Inverse Park Transformation

Inverse Park transformation, transforms vectors from rotating reference frame to stationary reference frame. The same angle that is used for Park transformation is generally used for inverse Park transformation. The equations for inverse park transform are expressed as:

$$\alpha = d \times \cos\theta - q \times \sin\theta$$

$$\beta = d \times \sin\theta + q \times \cos\theta$$

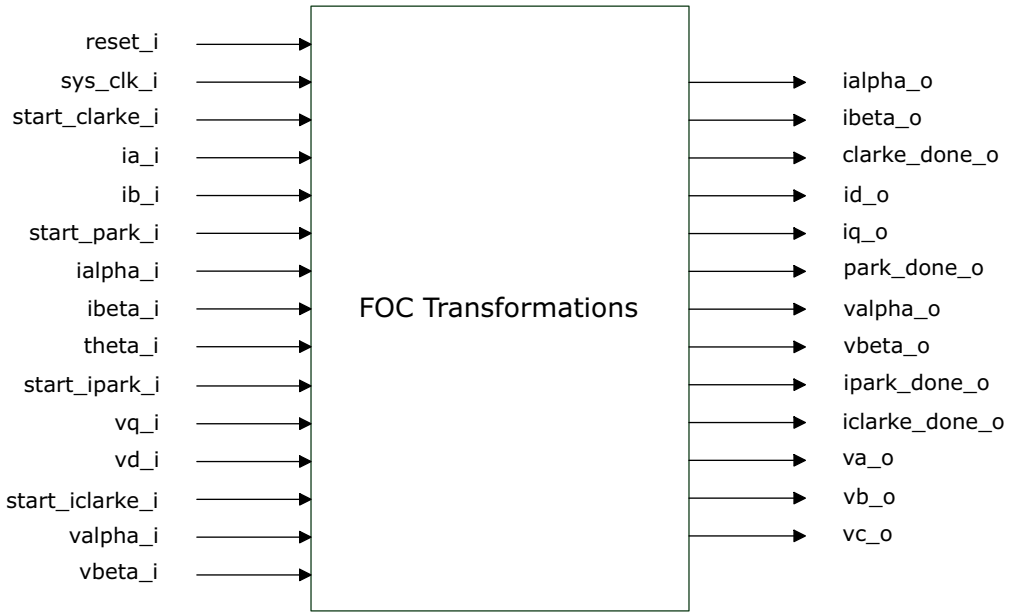
3 Hardware Implementation

This section describes the implementation details of the field oriented control (FOC) transformations block implemented in the SmartFusion®2 system-on-chip (SoC) field programmable gate array (FPGA) device.

3.1 FOC Transformations Implementation

The following figure shows the system-level block diagram of the FOC Transformations block.

Figure 3 • **FOC Transformations Block Diagram**



The FOC transformations block consists of the four transformation blocks (Clarke, Inverse Clarke, Park, and Inverse Park), which share a common math block, which can perform addition, subtraction, and multiplication. The entire system is synchronized with a system clock, which is available at sys_clk_i.

Each computation is triggered by a rising edge at the start input signal (start_clarke_i, start_park_i, start_ipark_i, and start_iclarke_i) and a rising edge on the corresponding done signal (clarke_done_o, park_done_o, iclarke_done_o, and ipark_done_o) indicates the completion of the computation.

The Clarke computation is triggered by a rising edge at the start_clarke_i input, and the inputs ia_i and ib_i are sampled at this time. At the end of the computation, the outputs ialpha_o and ibeta_o are valid when the clarke_done_o signal goes high.

The Park computation is triggered by a rising edge at the start_park_i input, and the inputs i_alpha_i and i_beta_i are sampled at this time. At the end of the computation, the outputs id_o and iq_o are valid when the park_done_o signal goes high.

The Inverse Park computation is triggered by a rising edge at the start_ipark_i input, and the inputs vq_i and vd_i are sampled at this time. At the end of the computation, the outputs valpha_o and vbeta_o are valid when the ipark_done_o signal goes high.

The Inverse Clarke computation is triggered by a rising edge at the start_iclarke_i input, and the inputs valpha_i and vbeta_i are sampled at this time. At the end of the computation, the outputs va_o, vb_o and vc_o are valid when the iclarke_done_o signal goes high.

Note: The math block is shared by each computation block, and so, only one computation should be triggered at a time.

3.2 Inputs and Outputs

The following table lists the input and output ports of the FOC transformations block.

Table 1 • **Inputs and Outputs of the FOC Transformations Block**

Port Name	Direction	Description
reset_i	Input	Active low asynchronous reset signal.
sys_clk_i	Input	System clock.
start_clarke_i	Input	Start signal for Clarke computation should be high for one system clock cycle.
ia_i	Input	Phase A current input for Clarke computation.
ib_i	Input	Phase B current input for Clarke computation.
start_park_i	Input	Start signal for Park computation should be high for one system clock cycle.
alpha_i	Input	Alpha-axis current input for Park computation
beta_i	Input	Beta-axis current input for Park computation.
theta_i	Input	Angle value for park and inverse park computation.
start_ipark_i	Input	Start signal for Inverse Park computation should be high for one system clock cycle.
vq_i	Input	Q-axis voltage input for inverse Park computation.
vd_i	Input	D-axis voltage input for inverse Park computation.
start_iclarke_i	Input	Start signal for inverse Clarke computation must be high for one system clock cycle.
valpha_i	Input	Alpha-axis voltage input for inverse Clarke computation.
vbeta_i	Input	Beta-axis voltage input for inverse Clarke computation.
ialpha_o	Output	Alpha-axis current output from Clarke computation.
ibeta_o	Output	Beta-axis current output from Clarke computation.
clarke_done_o	Output	Indicates Clarke computation is complete and the computation outputs are available. High for one system clock cycle.
id_o	Output	D-axis current output from Park computation
iq_o	Output	Q-axis current output from Park computation
park_done_o	Output	Indicates Park computation is complete and the computation outputs are available. High for one system clock cycle.
va_o	Output	A-phase voltage output from inverse Clarke computation.
vb_o	Output	B-phase voltage output from inverse Clarke computation.
vc_o	Output	C-phase voltage output from inverse Clarke computation.
iclarke_done_o	Output	Indicates inverse Clarke computation is complete and the computation outputs are available. High for one system clock cycle.
valpha_o	Output	Alpha-axis voltage output from inverse Park computation.
vbeta_o	Output	Beta-axis voltage output from inverse Park computation.
ipark_done_o	Output	Indicates inverse Park computation is complete and the computation outputs are available. High for one system clock cycle.

3.3 Configuration Parameter

The following table shows the configuration parameter used in the hardware implementation of the FOC transformations block. This parameter is generic and can be varied based on the application requirement.

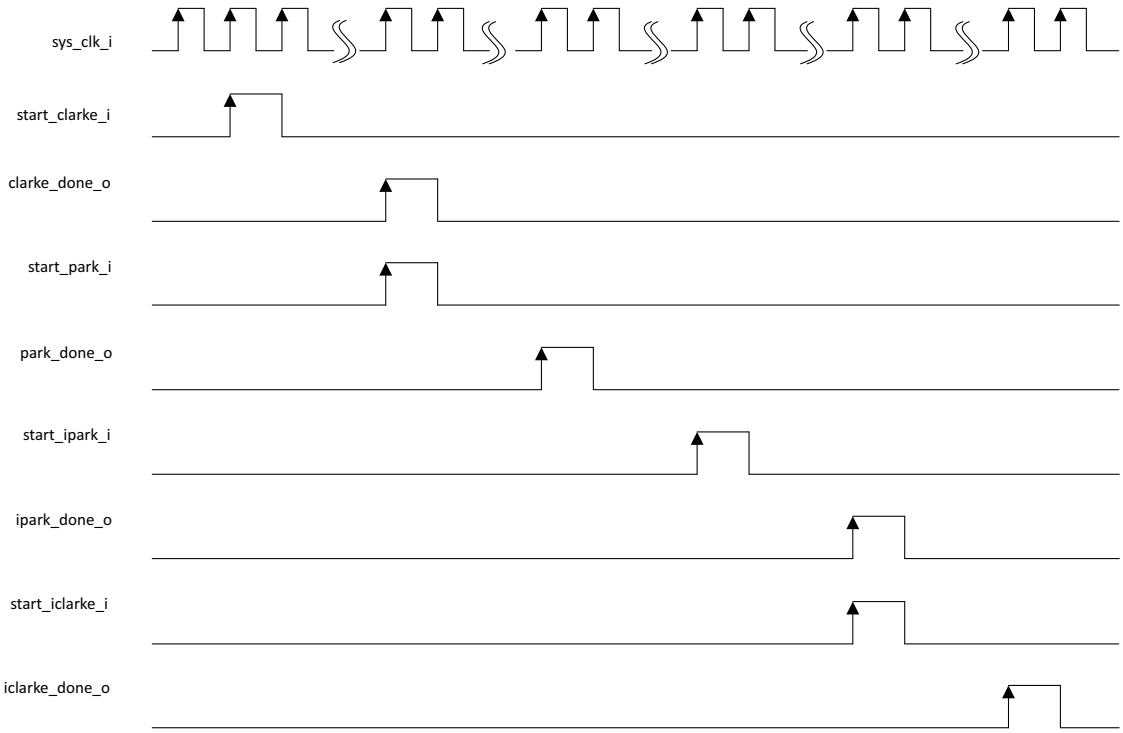
Table 2 • **Configuration Parameter of the FOC Transformations Block**

Name	Description
g_NO_MCYCLE_PATH	The number of clock delays required before the multiplication product ready signal is asserted.

3.4 Timing Diagram

The following figure shows the timing diagram of the FOC transformations block.

Figure 4 • **Timing Diagram of the FOC Transformations Block**



3.5 Resource Utilization

FOC transformations is implemented on the SmartFusion2 SoC FPGA and IGLOO®2 devices.

The following table lists the resource utilization report after synthesis.

Table 3 • **Resource Utilization of the FOC Transformations Block**

Resource	Usage
Sequential elements	370
Combinational logic	610
MACC	1
RAM1kx18	0
RAM64x18	0