UG0612
User Guide
Speed ID IQ PI Controller v4.1

Power Matters.™

Power Matters.™

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

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided “as is, where is” and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

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.

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.
- Updated Figure 3, page 4.
- Removed g_STD_IO_WIDTH configuration parameter from sections Configuration Parameter, page 6 and Resource Utilization, page 7.

1.2 Revision 2.0

Updated SAR (69696).

1.3 Revision 1.0

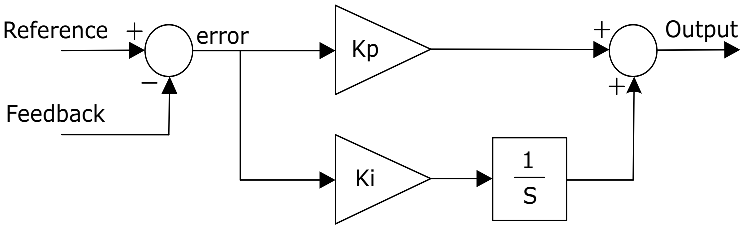
Revision 1.0 is the first publication of this document.

2 Introduction

The PI controller is a widely used closed loop controller for controlling a first order system. The basic functionality of a PI controller is to make the feedback measurement track the reference input. It performs this action by controlling its output till the error between the reference and feedback signals becomes zero.

There are two components that contribute to the output, the proportional term and the integral term, as shown in the following figure. The proportional term depends only on the instantaneous value of the error signal, whereas the integral term depends on the present and previous values of an error.

Figure 1 • **PI Controller in Continuous Domain**



PI controller in the continuous time domain is expressed as:

$$y(t) = K_P \times e(t) + Ki \times \int_0^t e(t)dt$$

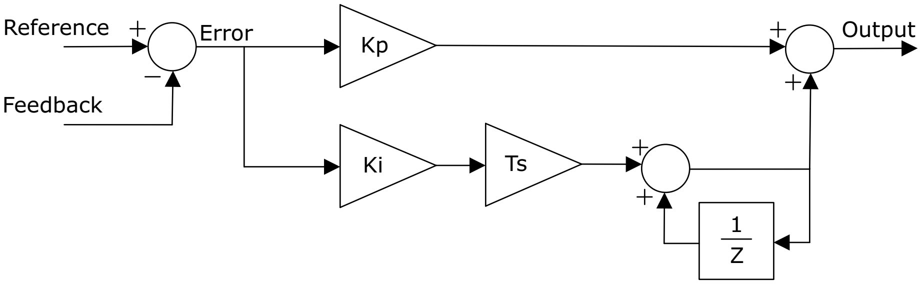
Where,

y (t) = PI controller output

e (t) = reference (t) - feedback (t) is the error between reference and feedback

To implement the PI controller in the digital domain, it has to be discretized. The discretized form of the PI controller based on zero order hold method is shown in the following figure.

Figure 2 • **PI Controller based on Zero Order Hold Method**



$$P(n) = K_p \times e(n)$$

$$I(n) = K_i \times T_s \times e(n) + I(n-1)$$

$$Y(n) = P(n) \times I(n)$$

2.1 Anti-Windup and Initialization

The PI controller has minimum and maximum limits of output, to keep the output within practical values. If a non-zero error signal persists for a long time, the integral component of the controller keeps increasing and may reach a value limited by its bit width. This phenomenon is called integrator windup and must be avoided to have a proper dynamic response. The PI controller IP has an automatic anti-windup function, which limits Integrator as soon as the PI controller reaches saturation.

In certain applications like motor control, it is important to initialize the PI controller to a proper value before enabling it. Initializing the PI to a good value avoids jerky operation. The IP block has a enable input to enable or disable the PI controller. If disabled, the output is equal to the unit input and when enabled, the output is the PI computed value.

2.2 Time Sharing of PI Controller

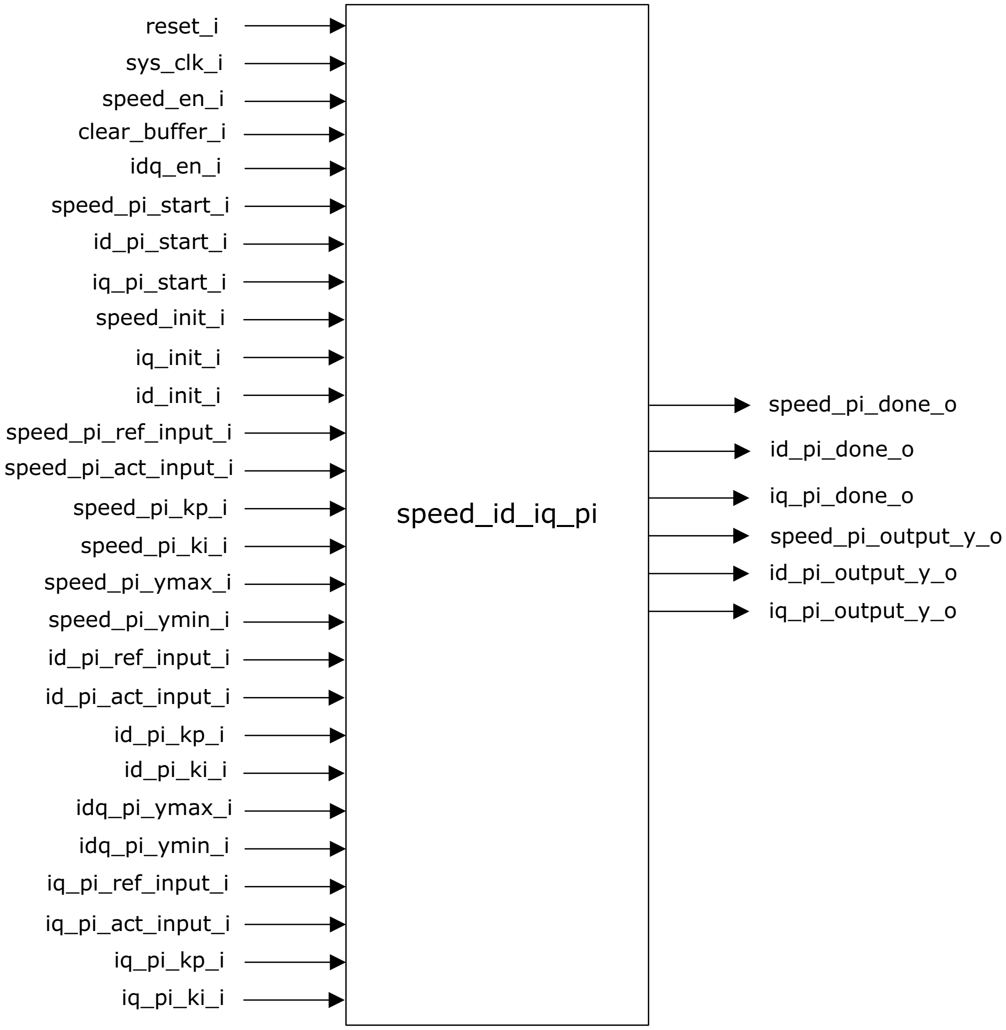
In field oriented control (FOC) algorithm, there are three PI controllers for Speed, d-axis current Id, and q-axis current Iq. The input of one PI controller depends on the output of the other PI controller and so they are executed sequentially. At any instant, there is only one instance of the PI controller in operation. Hence, instead of using three individual PI controllers, single PI controller is time shared for Speed, Id and Iq for optimum usage of resources.

The Speed_Id_Iq_PI module allows sharing of the PI controller through the start and done signals for each of Speed, Id, and Iq. The tuning parameters Kp, Ki, and minimum and maximum limits of each instance of controller can be configured independently through corresponding inputs.

3 Hardware Implementation

The following figure shows the block diagram of the Speed ID IQ PI Controller.

Figure 3 • **System-Level Block Diagram of Speed ID IQ PI Controller**



Note: The Speed ID IQ PI controller executes a PI controller algorithm for three quantities—d-axis current, q-axis current, and motor speed. The block is designed to minimize the hardware resource utilization. The block allows the PI controller algorithm to be run for one parameter at a time.

3.1 Inputs and Outputs

The following table lists the input and output ports of the Speed ID IQ PI Controller.

Table 1 • **Input and Output Ports of the Speed ID IQ PI Controller**

Signal Name	Direction	Description
reset_i	Input	Active low synchronous reset signal.
sys_clk_i	Input	System clock.
speed_en_i	Input	Enable signal for speed PI. When set to 1, normal PI controller operation occurs. When set to 0 (zero), PI controller output is fixed to the value available at the speed_init_i input.
clear_buffer_i	Input	Clears internal buffers when the motor stops.
idq_en_i	Input	Enable signal for Id and Iq PI: When set to 1, normal PI controller operation occurs When set to 0 (zero), PI controller output is fixed to the value available at the id_init_i and iq_init_i inputs respectively.
speed_pi_start_i	Input	Start signal for speed PI.
idpi_start_i	Input	Start signal for Id PI.
iqpi_start_i	Input	Start signal for Iq PI.
speed_init_i	Input	Initialization value for speed PI.
iq_init_i	Input	Initialization value for iq PI.
id_init_i	Input	Initialization value for id PI.
speed_pi_ref_input_i	Input	Speed PI reference input.
speed_pi_act_input_i	Input	Speed PI feedback measurement input.
speed_pi_kp_i	Input	Proportional gain (Kp) for Speed PI.
speed_pi_ki_i	Input	Integral gain (Ki) for Speed PI.
speed_pi_ymax_i	Input	Saturation limit (upper threshold) of speed PI controller.
speed_pi_ymin_i	Input	Saturation limit (lower threshold) of speed PI controller.
id_pi_ref_input_i	Input	Id PI Reference input.
id_pi_act_input_i	Input	Id PI feedback measurement input.
id_pi_kp_i	Input	Proportional gain (Kp) for Id PI.
id_pi_ki_i	Input	Integral gain (Ki) for Id PI.
idq_pi_ymax_i	Input	Saturation limit (upper threshold) of current PI controller.
idq_pi_ymin_i	Input	Saturation limit (lower threshold) of current PI controller.
iq_pi_ref_input_i	Input	Iq PI Reference input.
iq_pi_act_input_i	Input	Iq PI feedback measurement input.
iq_pi_kp_i	Input	Proportional gain (Kp) for Iq PI.
iq_pi_ki_i	Input	Integral gain (Ki) for Iq PI.
idq_pi_ymax_i	Input	Saturation limit (upper threshold) of current PI controller.
idq_pi_ymin_i	Input	Saturation limit (lower threshold) of current PI controller.
speed_pi_done_o	Output	Indicates Speed PI computation is complete. High for one system clock cycle.
id_pi_done_o	Output	Indicates Id PI computation is complete. High for one system clock cycle.

Table 1 • **Input and Output Ports of the Speed ID IQ PI Controller**

Signal Name	Direction	Description
iq_pi_done_o	Output	Indicates Iq PI computation is complete. High for one system clock cycle.
speed_pi_output_y_o	Output	Speed PI computation output.
id_pi_output_y_o	Output	Id PI computation output.
iq_pi_output_y_o	Output	Iq PI computation output.

3.2 Configuration Parameter

The following table shows the configuration parameter used in the hardware implementation of the Speed ID IQ PI Controller. This parameter is a generic and can be varied based on the application requirement.

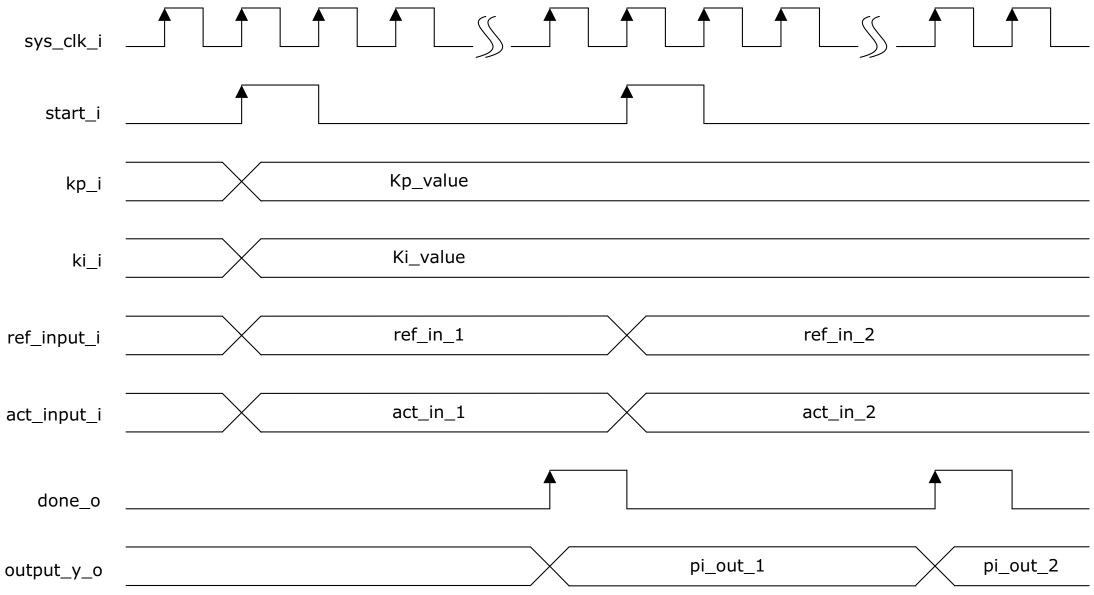
Table 2 • **Configuration Parameter of the Speed ID IQ PI Controller**

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

3.3 Timing Diagram

The following figure shows the timing diagram of the Speed ID IQ PI Controller. The following signals correspond to one set of input and output ports of the Speed ID IQ PI controller.

Figure 4 • **Timing Diagram of the PI Controller**



3.4 Resource Utilization

Speed ID IQ PI Controller is implemented on the SmartFusion®2 system-on-chip (SoC) field programmable gate array (FPGA) device.

The following table lists the resource utilization report after synthesis.

Table 3 • **Resource Utilization of the Speed ID IQ PI Controller**

Cell Usage	Description
Sequential elements	410
Combinational logic	630
MACC	1
RAM1kx18	0
RAM64x18	0