FUNCTION
Field Orientation Control (FOC)

VHDL File
motorfoc.vhd

Applicable Devices
Spartan3ADSP, Spartan6, 7-Family, UltraScale+

Xilinx primitives used
DSP48A/A1/E1
RAMB16_S18_S18

Sub modules used
encdig3w.vhd
resolver.vhd
hallsnsp.vhd
manrot.vhd
clarke.vhd
rectopol.vhd
park.vhd
pi_control.vhd
pwmmod.vhd
rpfm3phxlm.vhd

Execution time
177 to 206 clocks

Introduction
This IP Core implements the main field orientation control (FOC) module. Basically it is a container for other specialized IP Cores.
The module uses the following main functions:

- encoder signal interface
- hall sensor interface
- resolver interface
- Clarke transformation
- Park and Inverse Park transformation
- Proportional Integral (PI) control for current loop
- Cartesian to polar transformation
- BEMF feed forward compensation
- Pulse Width Modulation (PWM)
- Regenerative Pulse Frequency Modulation (RPFM) 2/3 level
- Very fast speed loop and position loop regulators.

Detailed Description

Figure 1 shows the main scheme of the FOC implemented in the IP Core.
This FOC implements a **current control** accepting as reference inputs setval_x (direct current component) and setval_y (quadrature current component). Two stator currents ia and ib are supplied to this FOC and converted from a rotating 3-phase system into a rotating two-phase coordinate system described by the variables curr_x and curr_y via the Clark transform. The two currents curr_x and curr_y are then forwarded to a Park's transform that using the rotor's angle rot_angle maps them into a fixed frame curr_rot_x and curr_rot_y. In steady state conditions curr_rot_x and curr_rot_y are constant. The setval_x reference controls rotor magnetizing flux; the setval_y reference controls the output torque of the motor. The difference between curr_rot_y and setval_y defines the torque error. The difference between curr_rot_x and setval_x defines the rotor magnetizing flux error. The errors are fed into a PI (proportional integral) controller that transform the current error into a voltage error vsmod_rot_x and vsmod_rot_y. An inverse Park transform is applied to vsmod_rot_x and vsmod_rot_y mapping them from a fixed frame into a rotating frame vsmod_x and vsmod_y. Rectangular to polar conversion is then applied to vsmod_x and vsmod_y to obtain vsangval and vamodval, representing the module and angle of the stator voltage. At this level the BEMF feed forward compensation is applied by adding a value to Vs modulo. This function is driven by Vs speed evaluation modude. The module and angle are fed into the power modulation unit, PWM or RPFM that provide sinusoidal or space vector modulation using pulse width modulation (PWM) or pulse frequency / pulse density modulation (RPFM) to the motor.
The FOC uses an incremental rotary encoder to capture the rotors angle position.

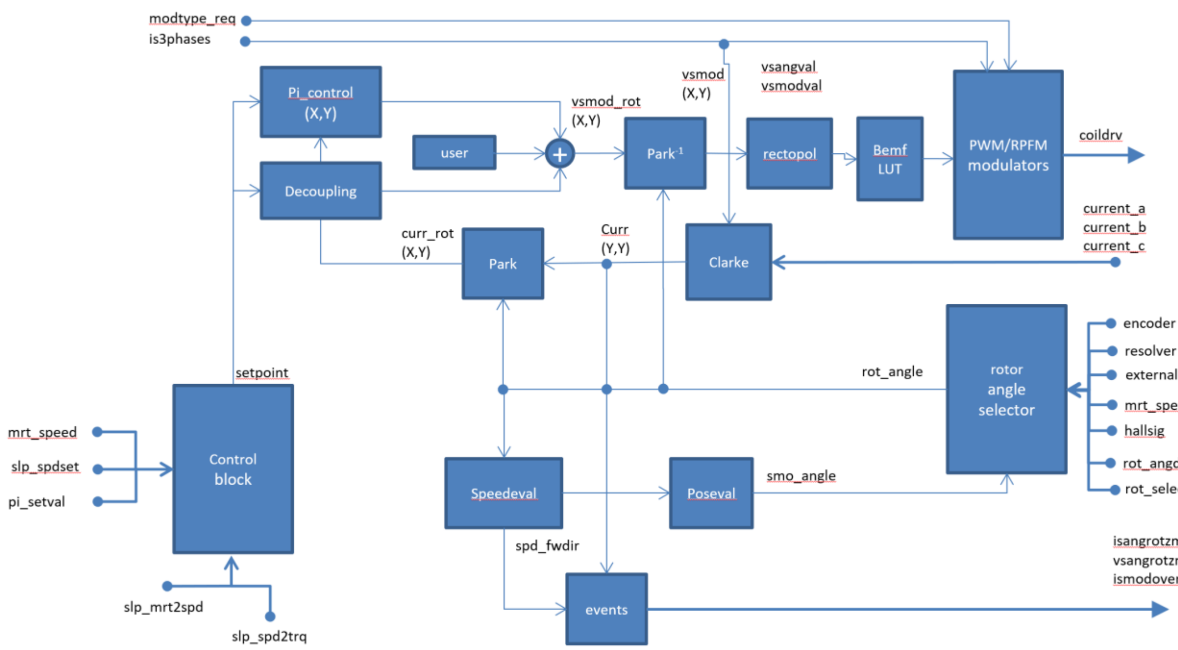

Figure 1

Sub modules description

tsmeasure_proc

This process measures the FOC cycle time and holds the result is “tsclocks” signals.

speed_ctrl_inst

the speed control loop is based on simple PI controller with antiwindup function for integrator. A the output current is limited by signal.
The “X”,”Y” current components are defined by dedicated signals.
The module is detailed defined in TDS_119.

smo_eval_inst

This module is an instance of *posslmeval* for the sliding mode observer (SMO) position estimation. The module analyzes the supplied voltage vector and the measured current vector to evaluate the BEMF vector and then the rotor electrical angle.
The module is detailed defined in TDS_117.

speed_eval_inst

This module is an instance of *speedeval*. The module evaluates the rotor electrical angle speed. The output is required by the position estimator.
The module is detailed defined in TDS_118.

rsv_inst

This module is an instance of *resolver* for resolver sensor. The module interfaces a set of 3 analog inputs: sine/cosine/exciter.
The module is detailed defined in TDS_124.

hls_inst

This module is an instance of *hallsnsp* for hall sensor interface. The module interface a double set of 3 hall sensors to decode up to 12 angle positions. A s/w programmable DPR is used to set the angle for each hall sensor position. Independent forward/backward direction angle compensation register is implemented. A s/w interpolator is used to evaluate intermediate angle between two consecutive hall sensor positions.
The module is detailed defined in TDS_122.

enc_inst

This module is an instance of *encoder3r* and executes two basic functions:

- It decodes the signals from an incremental encoder. The phase channel CH-A and CH-B are mandatory. The index channel to indicate complete revolution is optional.
- It calculates the electrical angle of the rotor. To calculate the electrical angle the module uses the counting of the encoder phases, a parameter that is the delta of angle for each phase and the timing signal **curr_sync** to interpolate the electrical angle between two phase change.

The outputs of the module are an input for the manrot instance.
The module is detailed defined in TDS_101.

mrot_inst

This module is an instance of *manrot* and it is used to emulate the encoder function. In this module an integrator, synchronized with FOC current event, increments a parameter that represents the electrical angle. The module can be used in place of other rotor position evaluation (encoder, hall sensor, sensorless etc..). The basic implementation consists of speed integrator. The full implementation includes a speed ramp limitation and a double LPF1 filter.
The module is detailed defined in TDS_102.

clarke_inst

This module is an instance of *clarke* and it transforms a three-phase currents systems into a two phase orthogonal system.
The module is detailed defined in TDS_103.

rectopol_inst

This instance of *rectopol* converts the rectangular coordinates pairs (X, Y) to the polar coordinates (modulo, angle). This instance is used to evaluate Is vector (module, angle) from Is currents (X, Y) and also to evaluate Vs vector (modulo, angle) for modulator. The instance is shared to save FPGA resources.
The module is detailed defined in TDS_112.

park_inst

This is an instance of *park*. The park module implements both park_direct and park_inverse transforming. The park_direct is used to convert Is vector <u>from stator</u> reference <u>to rotor</u> reference. The park_inverse is used to convert the Vs vector <u>from rotor</u> reference <u>to stator</u> reference. The shared implementation saves FPGA resources.
The module is detailed defined in TDS_104.

ctrl_X_inst, ctrl_Y_inst

These are two instances of *pi_control*. Each module implements a PI (Proportional Integral) control and takes as input the set point of the current (setval_x, setval_y) and the calculated current of the motor referred to the rotor (curr_rot_X_i, curr_rot_Y_i). The set point has the same dimension and scale of the curr_rot_X_i (or curr_rot_Y_i).
The output has the dimension of a voltage; the conversion is made by the gain of the PI. The output value is scaled to maximize the dynamic of calculus; the final scaling to get the real voltage to apply to the motor is executed further in the control chain.
The module is highly configurable by the user and it is provided with anti-windup.
The module is detailed defined in TDS_111.

pwm_mod_inst

This is an instance of *pwmmod*. It implements the PWM (Pulse Width Modulation) module.
It gets as inputs:

- Supply voltage value (dc_link) of the motor as provided by the power supply acquisition A/D depending on the hardware implementation.
- Polar coordinates of the voltage vector desired. It is [vsangval_i, vsmodval_i].
- Scaling parameters to calculate the correct modulation factor.
- PWM frequency desired.
- Selector for 3-phase motors and bipolar stepper motors.

The module uses the intersecting method between two waveforms to generate the PWM. The reference waveform is user programmable (in the IP core there are two default waveforms), while the other waveform is a triangular wave. The module has 4 tables to store the user defined waveforms. The default functions are a cosine for the sinusoidal modulation and a cosine with overlapped a harmonic to get a zero-sequence-insertion modulation (only for three-phase motors).
It is possible to drive both three-phase motors and stepper motors.
The module is detailed defined in TDS_106.

rpfm3ph_mod_inst

This module, an instance of *rpfm3phxlm*, realizes the PFM (Pulse Frequency Modulation).
It gets as inputs:

- Supply voltage value (dc_link) of the motor as provided by the power supply acquisition A/D depending on the hardware implementation.
- Polar coordinates of the voltage vector desired. It is [vsangval_i, vsmodval_i].
- High speed modulator prescaler.

The module is detailed defined in TDS_123.

mod_type_sel_proc

This is a process that selects the type of modulation to output (PWM or PFM). It takes in input both the modulation generated by the modules **pwm_mod_inst** and **rpfm3ph_mod_inst** and selects one of this as output. The selection is decided by the user with the signal *modtype_req* and *is3phases*.
In Table 1 is reported the meaning of the selection signals.

Signal	Value	Meaning
modtype_req	MOD_TYPE_PWM	Select PWM.
	MOD_TYPE_RPFM	Select PFM.
is3phases	0	Stepper motor selected. Only PWM is available.
	1	Three-phase motor selected.

Table 1: modulation selection table

The signal *modtype_ack* report to the user the modulation selected. In the table below there are the possible values.

input		output
modtype_req	is3phases	modtype_ack
MOD_TYPE_PWM	0	MOD_TYPE_PWM
MOD_TYPE_PWM	1	MOD_TYPE_PWM
MOD_TYPE_RPFM	0	MOD_TYPE_PWM
MOD_TYPE_RPFM	1	MOD_TYPE_RPFM

Table 2: modtype_ack values

The output of this module, *coildrv_i*, is mapped on the output *coildrv* of *motorfoc* and it is the input for the motor drivers.

isover_proc

This process checks the module of the I_s current. The value is compared with a reference value set by the host system. If the current is higher than the reference value for C_ISOVERFLOW_CLIM consecutive times, the signal ismodover is set (value '1').

Control block

The control block consists of a set of sub modules and some controlled switches to feed the pi_control regulators.
The involved modules are:

- Posandspd : position loop and speed loop pi regulator
- Manrot : manual rotor angle generator and speed filtering for scalar control mode
- Pi_control : both regulators for X and Y current components.

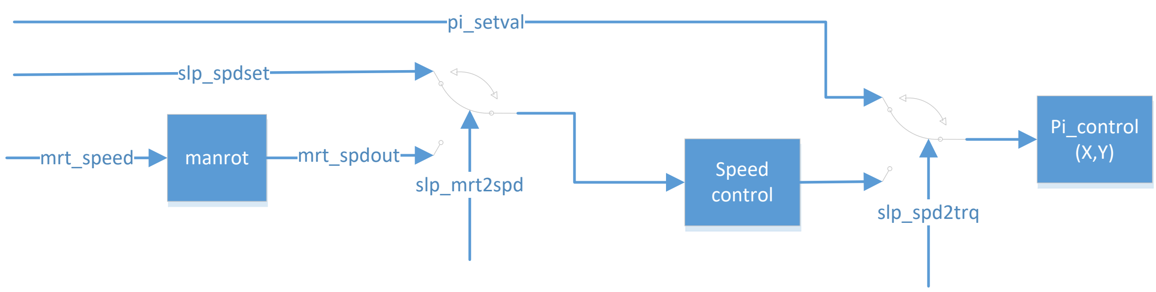

Figure 2 - control block diagram

Spd2trq	*Mrt2spd*	*Control mode*
0	x	Torque/current
1	0	Speed direct
1	1	Speed with MRT

Figure 3 - control matrix

Rotor angle position

In FOC (field oriented control) algorithm the knowledge of rotor angle value is the main task.
In ***motorfoc*** there are several functional block that can be used to establish the rotor angle value.
A software controlled switch is used as selector.

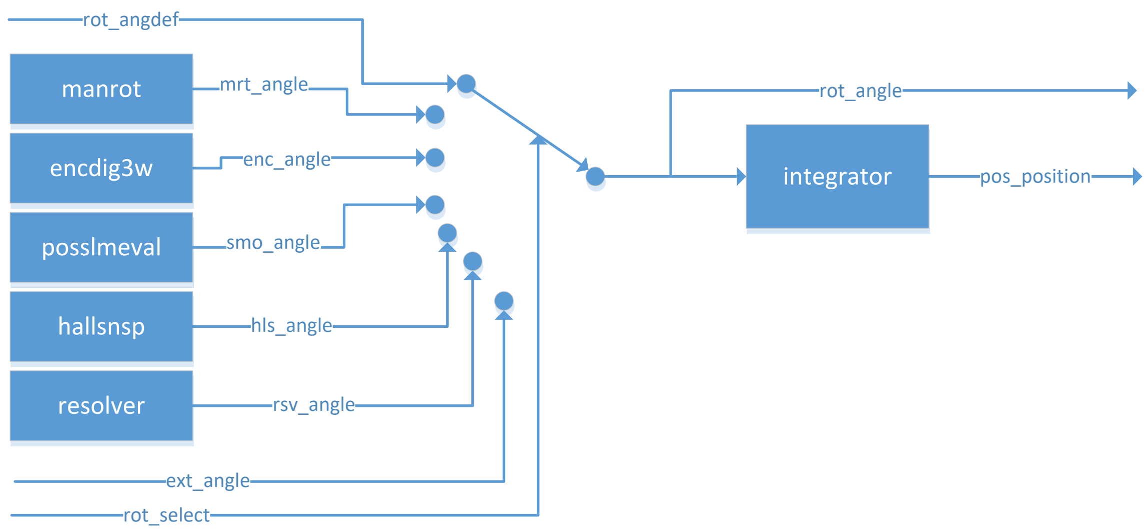

Figure 4 - rotor angle selector

PARAMETERS

Parameter	Type	Values	Default	Description
C_FAMILY	string	spartan3adsp spartan6 artix7 kintex7 virtex7 zynq	zynq	Xilinx FPGA Family name
Inverter analog inputs				
C_INV_IN_MAP[11:0]	std_logic_vector	0x000..0xFFF	0x083	Bit map enabled input channels: 0=IPHS_A 1=IPHS_B 2=IPHS_C 3=IBUS_X 4=VPHS_A 5=VPHS_B 6=VPHS_C 7=VBUS_X 8=VPHS_N
C_INV_IN_NOT[11:0]	Std_logic_vector	0x000..0xFFF	0x000	Bit map inverted channels. Bit values as per C_INV_IN_MAP
C_INV_OFSV_MODE	Integer	0..3	1	Offset set mode functions bit definition: 0=Self-Zero 1=S/W registers
C_INV_OVER_IPHS	integer	0..1	1	Overcurrent detection motor phases
C_INV_OVER_IBUS	integer	0..1	1	Overcurrent detection dc_link
C_INV_FILTER	Integer	0..1	1	2nd order LPF inputs
C_INV_EVAL_VPHS	Integer	0..1	1	Evaluate V-PHS vector
C_INV_R2P_VPHS	Integer	0..1	1	Rec2Pol on V-PHS
C_CLARKE_NPHS	Integer	0,2,3	3	Clarke transform input phases. 0=transparent (2-phases only for bipolar stepper motor) 2=2-phases used in 3-phases motor 3=3-phases used in 3-phases motor
C_DCPL_KJFF	Integer	0..1	1	Enable Decoupling module
C_VSVEC_AJFF	Integer	0..1	0	Add user Vs vector
C_BEMF_KLIN	Integer	0..1	1	BEMF Linearization LUT

C_PWM_MODULATOR	integer	0..2	1	Include PWM modulator IP
C_RPFM_MODULATOR	integer	0..2	1	Include RPFM modulator IP
C_RPFM_3_LEVEL	integer	0..1	1	RPFM 3-level extension
C_RPFM_TPNC	integer	0..1	1	RPFM 3-level T-PNC variant
C_SMO_EVAL	integer	0..1	1	Include SMO Position Estimator IP
C_SPD_EVAL	integer	0..1	1	Include Speed Measurement IP
C_SPEED_CTRL	Integer	0..1	1	Include speed loop
C_EXTANGLE	Integer	0..2	1	Include external angle
C_RESOLVER	integer	0..2	1	Include resolver sensor IP
C_RSV_EXCMIN	integer	1..65536	50000	Resolver Minimum Exciter Values
C_RSV_SOCMIN	integer	1..65536	50000	Resolver Minimum Sin/Cos Values
C_RSV_EXCTMO	integer	1..65536	100	Resolver Exciter timeout
C_RSV_EXCPOL	integer	0..2	0	Resolver Exciter polarity 0=both, 1=positive, 2=negative
C_RSV_ANGLUT	integer	0..3	0	Resolver angle LUT 0=off, 1=1K, 2=2K, 3=4K
C_RSV_ANGFLT	integer	0..1	0	Resolver angle filter enable
C_RSV_ALRELK	integer	0..18,255	5	Resolver angle LUT mode 255=abosulute, 0..18=relative
C_HALLSENSOR	integer	0..1	1	Include hall sensor IP
C_ENCODER	Integer	0..1	2	Include enhanced 2..3 wire encoder IP
C_MANROT	Integer	0..2	2	Scalar mode rotor angle IP: 0=no, 1=angle update, 2=enhanced with ramp and 2nd order filter
C_MRT_ACCMAX_DLN2	Integer	8..16	16	Base 2 logarithm of MRT acceleration limiter

C_DCK_SP_DLN2	Integer	-	9	Decoupling Speed Ln2(divider)
C_DCK_KE_DLN2	Integer	-	24	Decoupling Bemf KE Ln2(divider)
C_DCK_KL_DLN2	Integer	-	41	Decoupling Inductor Ln2(divider)
C_SMO_ZS_DLN2	integer	22..26	24	Base 2 logarithm of proportional error divisor
C_SMO_F2_DLN2	integer	26..30	28	Base 2 logarithm divider use to eval K of 2nd LPF
C_SLP_PRO_DLN2	Integer	0..n	1	Base 2 logarithm of speed loop proportion error regulator
C_SLP_INT_DLN2	Integer	0.n	5	Base 2 logarithm of speed loop integrative error regulator
C_SLP_INDWP_DLN2	Integer	0.n	1	Base 2 logarithm of speed loop integrative error regulator for anti windup
C_SLP_INDWP_KDIV	Integer	0.n	1	Base 2 logarithm of speed loop integrative error regulator for anti windup
C_ISOVERFLOW_CMAX	integer	1..15	1	Current modulo overflow counter limit
C_PI_ERPRO_DLN2	integer	0..n	12	Base 2 logarithm of current loop PI proportional error divisor
C_PI_ERINT_DLN2	integer	0..n	18	Base 2 logarithm of current loop PI integrative error divisor

Important:
The exclusion of components can be used to save FPGA resources. Some components are internally interconnected so the exclusion can inhibit the connected IP components.
Refer to the block diagram for details.

SIGNALS

Signal	I/O	Description
clock	IN	Clock (rising edge).
reset	IN	Reset. Active high.
Hall Sensors LUT host access		
hls_mem_we	IN	write enable.
hls_mem_addr[3:0]	IN	memory address.
hls_mem_din[17:0]	IN	memory data input.
hls_mem_dout[17:0]	OUT	memory data output.
PWM modulator LUT host access		
pwm_wvfm_en	IN	memory enable
pwm_wvfm_we	IN	memory write enable.
pwm_wvfm_addr[9:0]	IN	memory address.
pwm_wvfm_din[17:0]	IN	memory data input.
pwm_wvfm_dout[17:0]	OUT	memory data output.
BEMF linearization LUT host access		
bemf_llin_en	IN	memory enable
bemf_llin_we	IN	memory write enable.
bemf_llin_addr[9:0]	IN	memory address.
bemf_llin_din[17:0]	IN	memory data input.
bemf_llin_dout[17:0]	OUT	memory data output.
Resolver #0 linearization LUT host access		
rsv_0_mem_en	IN	memory enable
rsv_0_mem_we	IN	memory write enable.
rsv_0_mem_addr[31:0]	IN	memory address.
rsv_0_mem_din[17:0]	IN	memory data input.
rsv_0_mem_dout[17:0]	OUT	memory data output.
Resolver #1 linearization LUT host access		
rsv_1_mem_en	IN	memory enable
rsv_1_mem_we	IN	memory write enable.
rsv_1_mem_addr[31:0]	IN	memory address.
rsv_1_mem_din[17:0]	IN	memory data input.
rsv_1_mem_dout[17:0]	OUT	memory data output.
tsclocks[11:0]	OUT	System clocks per FOC cycle
zer_iphs_a	IN	Auto Zero for IPHS_A
zer_iphs_b	IN	Auto Zero for IPHS_B
zer_iphs_c	IN	Auto Zero for IPHS_C
zer_ibus_x	IN	Auto Zero for IBUS_X
zer_vphs_a	IN	Auto Zero for VPHS_A
zer_vphs_b	IN	Auto Zero for VPHS_B
zer_vphs_c	IN	Auto Zero for VPHS_C
zer_vbus_x	IN	Auto Zero for VBUS_X

zer_vphs_n	IN	Auto Zero for VPHS_N
xbus_fk1[16:0]	IN	UNSIGNED17. K value for 1st LPF1 DC_LINK filter
xbus_fk2[16:0]	IN	UNSIGNED17. K value for 2nd LPF1 DC_LINK filter
xphs_fk1[16:0]	IN	UNSIGNED17. K value for 1st LPF1 Motor Phases filter
xphs_fk2[16:0]	IN	UNSIGNED17. K value for 2nd LPF1 Motor Phases filter
ibus_limit[16:0]	IN	UNSIGNED17. Dc_link current limit.
iphs_limit[16:0]	IN	UNSIGNED17. Motor phases current limit.
ovi_clear	IN	Remove overcurrent error
SIGNED18. RAW values from A/D channel		
acq_iphs_a[17:0]	IN	IPHS_A
acq_iphs_b[17:0]	IN	IPHS_B
acq_iphs_c[17:0]	IN	IPHS_C
acq_ibus_x[17:0]	IN	IBUS_X
acq_vphs_a[17:0]	IN	VPHS_A
acq_vphs_b[17:0]	IN	VPHS_B
acq_vphs_c[17:0]	IN	VPHS_C
acq_vbus_x[17:0]	IN	VBUS_X
acq_vphs_n[17:0]	IN	VPHS_N
acq_sync	IN	Data input sync
SIGNED18. Input offset values		
iof_iphs_a[17:0]	IN	IPHS_A
iof_iphs_b[17:0]	IN	IPHS_B
iof_iphs_c[17:0]	IN	IPHS_C
iof_ibus_x[17:0]	IN	IBUS_X
iof_vphs_a[17:0]	IN	VPHS_A
iof_vphs_b[17:0]	IN	VPHS_B
iof_vphs_c[17:0]	IN	VPHS_C
iof_vbus_x[17:0]	IN	VBUS_X
Iof_vphs_n[17:0]	IN	VPHS_N
SIGNED18. Write Input offset command		
wof_iphs_a	IN	IPHS_A
wof_iphs_b	IN	IPHS_B

wof_iphs_c	IN	IPHS_C
wof_ibus_x	IN	IBUS_X
wof_vphs_a	IN	VPHS_A
wof_vphs_b	IN	VPHS_B
wof_vphs_c	IN	VPHS_C
wof_vbus_x	IN	VBUS_X
wof_vphs_n	IN	VPHS_N
SIGNED18. output offset values		
oof_iphs_a[17:0]	OUT	IPHS_A
oof_iphs_b[17:0]	OUT	IPHS_B
oof_iphs_c[17:0]	OUT	IPHS_C
oof_ibus_x[17:0]	OUT	IBUS_X
oof_vphs_a[17:0]	OUT	VPHS_A
oof_vphs_b[17:0]	OUT	VPHS_B
oof_vphs_c[17:0]	OUT	VPHS_C
oof_vbus_x[17:0]	OUT	VBUS_X
oof_vphs_n[17:0]	OUT	VPHS_N
SIGNED18. Input multiplier		
mul_iphs_a[17:0]	IN	IPHS_A
mul_iphs_b[17:0]	IN	IPHS_B
mul_iphs_c[17:0]	IN	IPHS_C
mul_ibus_x[17:0]	IN	IBUS_X
mul_vphs_a[17:0]	IN	VPHS_A
mul_vphs_b[17:0]	IN	VPHS_B
mul_vphs_c[17:0]	IN	VPHS_C
mul_vbus_x[17:0]	IN	VBUS_X
mul_vphs_n[17:0]	IN	VPHS_N
SIGNED18. Normalized values		
nrm_iphs_a[17:0]	OUT	IPHS_A
nrm_iphs_b[17:0]	OUT	IPHS_B
nrm_iphs_c[17:0]	OUT	IPHS_C
nrm_ibus_x[17:0]	OUT	IBUS_X
nrm_vphs_a[17:0]	OUT	VPHS_A
nrm_vphs_b[17:0]	OUT	VPHS_B
nrm_vphs_c[17:0]	OUT	VPHS_C
nrm_vbus_x[17:0]	OUT	VBUS_X
nrm_vphs_n[17:0]	OUT	VPHS_N
nrm_sync	OUT	Data sync
SIGNED18. Filtered values		

flt_iphs_a[17:0]	OUT	IPHS_A
flt_iphs_b[17:0]	OUT	IPHS_B
flt_iphs_c[17:0]	OUT	IPHS_C
flt_ibus_x[17:0]	OUT	IBUS_X
flt_vphs_a[17:0]	OUT	VPHS_A
flt_vphs_b[17:0]	OUT	VPHS_B
flt_vphs_c[17:0]	OUT	VPHS_C
flt_vbus_x[17:0]	OUT	VBUS_X
flt_vphs_n[17:0]	OUT	VPHS_N
flt_sync	OUT	Data sync
Overcurrent flags		
ovi_iphs_a	OUT	IPHS_A
ovi_iphs_b	OUT	IPHS_B
ovi_iphs_c	OUT	IPHS_C
ovi_ibus_x	OUT	IBUS_X
Speed evaluation (see QD_TDS_118 for more details)		
spd_fktau1[16:0]	IN	UNSIGNED17. K parameter of the first low pass filter for speed eval.
spd_fktau2[16:0]	IN	UNSIGNED17. K parameter of the second low pass filter for speed eval.
spd_speed[31:0]	OUT	SIGNED32. Speed evaluated.
spd_fwdir	OUT	Evaluated Rotor direction 1=forward, 0=reverse
spd_moving	OUT	Evaluated Rotor status 1=moving, 0=still.
rot_select[3:0]	IN	Rotor position selector.
rot_angdef[31:0]	OUT	Rotor position initial angle (preset value).
rot_angle[31:0]	OUT	UNSIGNED32 rotor angle from mux.
fbk_angdef	IN	Initial angle selector. 1=rot_angle, 0=rot_angdef
Position evaluation (see QD_TDS_117 for more details)		
smo_rstang	IN	SMO reset command. 1=reset, 0=operating.
smo_usevsact	IN	SMO Vs selector 0=target, 1=measured
smo_smontmr[15:0]	IN	SMO On timer
smo_vs_mult[31:0]	IN	UNSIGNED32 Vs multiplier for slide mode function.
Smo_is_mult[31:0]	IN	UNSIGNED32 Is multiplier for slide mode function.

Smo_zs_max[16:0]	IN	UNSIGNED17 error limit for sliding mode function
smo_es1_kflt[16:0]	IN	UNSIGNED17. K parameter of the LPFT1 pos eval BEMF1.
Smo_es2_kflo[16:0]	IN	UNSIGNED17. K parameter of the LPFT1 pos eval BEMF2 base value
smo_es2_kfmx[16:0]	IN	UNSIGNED17. K parameter of the LPFT1 pos eval BEMF2 speed multipler value
pos_es2_kflt[16:0]	OUT	UNSIGNED17. K current value of the LPFT1 pos eval BEMF2.
Smo_angofs[15:0]	IN	SIGNED16. Angle offset compensation.
Smo_bemf_x[17:0]	OUT	SIGNED18. Bemf X.
smo_bemf_y[17:0]	OUT	SIGNED18. Bemf Y.
smo_bemf_p[17:0]	OUT	UNSIGNED31. Bemf vector angle.
Smo_bemf_m[16:0]	OUT	UNSIGNED17. Bemf vector modulo.
Smo_angle[31:0]	OUT	UNSIGNED31. Rotor position angle.
Position		
pos_position[31:0]	OUT	Low resolution position (1/65536)
Speed control		
slp_spdset[31:0]	IN	Speed set point
slp_kmpro[31:0]	IN	Proportional K gain IEEE-754 Float 32bits
slp_kmint[31:0]	IN	Integrative K gain IEEE-754 Float 32bits
slp_outlim[16:0]	IN	Current limit
slp_kmultx[17:0]	IN	Current – X multiplier (div/65536)
slp_kmulty[17:0]	IN	Current – Y multiplier (div/65536)
Speed mux control		
slp_spd2trq	IN	Speed loop to Torque loop
slp_mrt2spd	IN	MRT to speed loop
paipol[7:0]	IN	Number of pair poles (1..n)
rsv_m2rppk[7:0]	IN	Resolver motor pair poles / resolver pair poles
rsv_spdfkt[16:0]	IN	Resolver LPF speed filter K-value
rsv_angfkt[16:0]	IN	Resolver LPF angle filter K-value
rsv_0_angofs[15:0]	IN	Resolver #0 angle offset for alignment
rsv_1_angofs[15:0]	IN	Resolver #1 angle offset for alignment

rsv_0_angle[31:0]	OUT	Resolver #0 angle output
rsv_1_angle[31:0]	OUT	Resolver #1 angle output
rsv_0_mangle[31:0]	OUT	Resolver #0 mechanical angle output
rsv_1_mangle[31:0]	OUT	Resolver #1 mechanical angle output
rsv_0_sync	OUT	Resolver #0 electric angle sync
rsv_1_sync	OUT	Resolver #1 electric angle sync
rsv_0_msync	OUT	Resolver #0 mechanical angle sync
rsv_1_msync	OUT	Resolver #1 mechanical angle sync
hls_ctolim[19:0]	IN	Hall Sensor Coasting mode or interpolator timeout
hls_angle[31:0]	OUT	UNSIGNED32. Phase electrical counter.
enc_rstang	IN	Encoder reset angle counter.
enc_inten	IN	Encode interpolator enable
enc_xmcha	IN	Encoder index match level for CH-A encoder signal
enc_xmchb	IN	Encoder index match level for CH-B encoder signal
enc_xmchi	IN	Encoder index match level for CH-I encoder sig-nal
enc_cyprnd[11:0]	IN	Encoder cycles per round
enc_angphs[31:0]	IN	UNSIGNED32. Set the electric angle equivalent to a step of the encoder data phase.
enc_fwdir	OUT	Encoder forward direction.
enc_phsev	OUT	Encoder phase event.
enc_idxev	OUT	Encoder index event.
enc_errev	OUT	Encoder error.
enc_phcnt[15:0]	OUT	SIGNED16.Encoder phase per round counter.
enc_phase[31:0]	OUT	SIGNED32.Encoder phase counter.
enc_phcpt[31:0]	OUT	SIGNED32.Encoder phase counter at the index event.
enc_angle[31:0]	OUT	UNSIGNED32. Phase electrical counter.
enc_index[31:0]	OUT	UNSIGNED32.Encoder index counter.
mrt_rstang	IN	Manrot reset angle
mrt_speed[31:0]	IN	Manrot speed set value
mrt_accmax[31:0]	IN	Maximum acceleration (ramp limit)
mrt_fktau1[16:0]	INT	UNSIGNED17. K parameter of the first LPFT1 speed
mrt_fktau2[16:0]	INT	UNSIGNED17. K parameter of the second LPFT1 speed

mrt_spdout[31:0]	OUT	Manrot speed output value
mrt_angle[31:0]	OUT	UNSIGNED32. Manrot rotor electrical angle.
ismodmax[16:0]	IN	UNSIGNED17. Module current limit.
ismodover	OUT	Current limit exceeded flag.
ismodval[16:0]	OUT	UNSIGNED17. Current module from the acquisition chain.
isangval[31:0]	OUT	UNSIGNED32. Current angular value from the acquisition chain.
vamodval[16:0]	OUT	UNSIGNED17. Voltage module from the acquisition chain.
vaangval[31:0]	OUT	UNSIGNED32. Voltage angular value from the acquisition chain.
Current control loop PI		
pi_deafmd	IN	Ignore the feedback signal. Active high. If this signal is asserted the PI regulator works in open loop.
pi_himdovf	IN	Stop integrator
pi_kmprox[16:0]	IN	UNSIGNED17. Proportional gain X-frame
pi_kmproy[16:0]	IN	UNSIGNED17. Proportional gain Y-frame
pi_kmintx[16:0]	IN	UNSIGNED17. Integral gain X-frame
pi_kminty[16:0]	IN	UNSIGNED17. Integral gain Y-frame
pi_setvalx[17:0]	IN	SIGNED18. Set point X-frame
pi_setvaly[17:0]	IN	SIGNED18. Set point Y-frame
pix_setval[17:0]	OUT	SIGNED18. Probe target X-frame
piy_setval[17:0]	OUT	SIGNED18. Probe target Y-frame
pix_fbkval[17:0]	OUT	SIGNED18. Probe feedback X-frame
piy_fbkval[17:0]	OUT	SIGNED18. Probe feedback Y-frame
pix_outpro[17:0]	OUT	SIGNED18. Probe X-proportional
piy_outpro[17:0]	OUT	SIGNED18. Probe Y-proportional
pix_outint[17:0]	OUT	SIGNED18. Probe X-integrative
piy_outint[17:0]	OUT	SIGNED18. Probe Y-integrative
pix_outval[17:0]	OUT	SIGNED18. Probe X-output
piy_outval[17:0]	OUT	SIGNED18. Probe Y-output
Decoupling		

dck_er_out_x[17:0]	OUT	SIGNED18. Probe BEMF X-frame
dck_er_out_y[17:0]	OUT	SIGNED18. Probe BEMF Y-frame
dck_vs_out_x[17:0]	OUT	SIGNED18. Probe Output X-frame
dck_vs_out_y[17:0]	OUT	SIGNED18. Probe Output Y-frame
vs_rotor_x[17:0]	OUT	SIGNED18. Probe Vs rotor X-frame
vs_rotor_y[17:0]	OUT	SIGNED18. Probe Vs rotor Y-frame
vsmodval[16:0]	OUT	UNSIGNED17. Module coordinate of the control voltage in a polar system.
vsangval[31:0]	OUT	UNSIGNED32. Angular coordinate of the control voltage in a polar system.
Decoupling configuration		
dck_useisset	IN	1=Use Is target, 0=use Is feedback
dck_ldkmul[16:0]	IN	UNSIGNED17. Ld flux multiplier
dck_lqkmul[16:0]	IN	UNSIGNED17. Lq flux multiplier
dck_kekmul[16:0]	IN	UNSIGNED17. Ke flux multiplier
User Vs feed forward		
vs_ajff_x[17:0]	IN	SIGNED18. Vs user feed forward X-frame
vs_ajff_y[17:0]	IN	SIGNED18. Vs user feed forward Y-frame
bemfangofs[15:0]	IN	SIGNED16. Bemf LUT angle offset
Modulator common		
mod2angskw [31:0]	IN	UNSIGNED32. 2nd modulator angle skew.
modtype_req[0:0]	IN	Modulator type request 0=PWM, 1=RPFM
pwm_angofs[15:0]	IN	SIGNED16. Feed forward angle.
pwm_kmod[31:0]	IN	UNSIGNED32. Parameter used in the PWM duty cycle evaluation.
pwm_presc[16:0]	IN	UNSIGNED17. PWM prescaler setting.
pwm_mdmax[16:0]	IN	UNSIGNED17. Max output modulation.
pwm_mdval[16:0]	OUT	UNSIGNED17. PWM modulation value. Range is 0 to pwm_mdmax (prescaler=100%).
pwm_minpw[16:0]	IN	UNSIGNED17. Minimum pulse width 0=disabled.
pwm_table[1:0]	IN	Select one of the 4 waveform stored in the DPRAM.
pwm_cmmofs	IN	Common mode offset
pwm_mdovf	OUT	Modulator overflow flag. Active high.

rpfm_angofs[15:0]	IN	SIGNED16. Feed forward angle.
rpfm_mdkd[7:0]	IN	UNSIGNED8. RPFM system clock prescaler for high-speed modulator.
rpfm_mdpr[3:0]	IN	UNSIGNED4, RPFM high speed modulator pre-scaler.
rpfm_v07n	IN	PFM vector 0/7 nice transition.
rpfm_v7h	IN	PFM vector 7 hold.
rpfm_modz[1:0]	OUT	PFM modulation zone.
coilgear	IN	Coil gear
coilenab	IN	Coil enable
ext_0_angle[31:0]	IN	External #0 sensor for angle position
ext_1_angle[31:0]	IN	External #1 sensor for angle position
rsv_0_priexc[17:0]	IN	Resolver #0 primary exciter
rsv_0_secsin[17:0]	IN	Resolver #0 secondary sine
rsv_0_seccos[17:0]	IN	Resolver #0 secondary cosine
rsv_1_priexc[17:0]	IN	Resolver #1 primary exciter
rsv_1_secsin[17:0]	IN	Resolver #1 secondary sine
rsv_1_seccos[17:0]	IN	Resolver #1 secondary cosine
hls_hallsig [5:0]	IN	Hall Sensor interface
enc_cha	IN	Encoder channel A.
enc_chb	IN	Encoder channel B.
enc_chi	IN	Encoder channel index.
generator	OUT	0=drive, 1=generator
modtype[0:0]	OUT	Modulation type acknowledge.
modlevels[0:0]	IN	0=2-levels, 1=3-levels
is3phases	IN	Three-phase motor selection flag
pwm2p_l2c1w1[1:0]	OUT	Pwm 2-phases, 2-levels, coil-1, winding-1
pwm2p_l2c2w1[1:0]	OUT	Pwm 2-phases, 2-levels, coil-2, winding-1
pwm2p_l2c1w2[1:0]	OUT	Pwm 2-phases, 2-levels, coil-1, winding-2
pwm2p_l2c2w2[1:0]	OUT	Pwm 2-phases, 2-levels, coil-2, winding-2
pwm2p_l2sync	OUT	Pwm 2-phases, 2-levels sync
pwm3p_l2cxw1[2:0]	OUT	Pwm 3-phases, 2-levels, coil-321, winding-1
pwm3p_l2cxw2[2:0]	OUT	Pwm 3-phases, 2-levels, coil-321, winding-2
pwm3p_l2sync	OUT	Pwm 3-phases, 2-levels, sync

rpfm3p_l2cxw1[2:0]	OUT	Rpfm 3-phases, 2-levels, coil-321, winding-1
rpfm3p_l2cxw2[2:0]	OUT	Rpfm 3-phases, 2-levels, coil-321, winding-2
rpfm3p_l2sync	OUT	Rpfm 3-phases, 2-levels, sync
rpfm3p_l3c1w1[1:0]	OUT	Rpfm 3-phases, 3-levels, coil-1, winding-1
rpfm3p_l3c2w1[1:0]	OUT	Rpfm 3-phases, 3-levels, coil-2, winding-1
rpfm3p_l3c3w1[1:0]	OUT	Rpfm 3-phases, 3-levels, coil-3, winding-1
rpfm3p_l3c1w2[1:0]	OUT	Rpfm 3-phases, 3-levels, coil-1, winding-2
rpfm3p_l3c2w2[1:0]	OUT	Rpfm 3-phases, 3-levels, coil-2, winding-2
rpfm3p_l3c3w2[1:0]	OUT	Rpfm 3-phases, 3-levels, coil-3, winding-2
rpfm3p_l3sync	OUT	Rpfm 3-phases, 3-levels, sync

Execution time

Start event	End of execution	clock cycles[1]	Time @ 50 MHz	Time @ 100 MHz
Curr_sync	Pwm(2-phases)	188	3.76 µS	1.88 µS
Curr_sync	Pwm(3-phases)	201	4.04 µS	2.01 µS
Curr_sync	RPFM(3-phases)	172	3.44 µS	1.72 µS

The FOC execution time is measured from ***flt_sync*** (filtered current values signal) to end of modulator internal execution.

Due to parallel implementation and dataflow signals propagation, the current sync data rate can be higher than execution time.

The absolute minimum ***acq_sync*** rate in clocks is 156. Do not drive the ***acq_sync*** at higher rate.
The currents data rate can be 3.12 µS @ 50 MHz and 1.56 µS @ 100 MHz .

[1] Unless otherwise noted.

Reference Documents

1. QD_TDS_101_01 [encoder3r specification, QDeSys 2012]
2. QD_TDS_102_01 [manrot specification, QdeSys 2012]
3. QD_TDS_103_01 [clarke transform specification, QdeSys 2012]
4. QD_TDS_104_01 [park transform specification, QdeSys 2012]
5. QD_TDS_105_01 [low pass filter specification, QdeSys 2012]
6. QD_TDS_106_01 [PWM module specification, QdeSys 2012]
7. QD_TDS_111_01 [PI control specification, QdeSys 2012]
8. QD_TDS_112_01 [rectopol specification, QdeSys 2012]
9. QD_TDS_116_01 [rpfm specification, QdeSys 2012]
10. QD_TDS_117_01 [position estimator, QdeSys 2012]
11. QD_TDS_118_01 [speed evaluation, QdeSys 2012]
12. ~~QD_TSD_119_01 []~~
13. ~~QD_TDS_120_01 []~~
14. QD_TDS_121_01 [incremental encoder 2/3 channels, QdeSys 2013]
15. QD_TDS_122_01 [hall sensor, QdeSys 2014]
16. QD_TDS_123_01 [rpfm 3-levels modulator, QdeSys 2015]]
17. QD_TDS_124_01 [resolver, QdeSys 2015]

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
26/08/2011	1.0	QdeSys first draft
09/09/2011	1.1	First release
12/09/2011	1.2	Modified footnote 2
11/10/2011	1.3	Added global diagram, updated reference documents
18/11/2011	1.4	Position estimator, optimization end general review
20/11/2011	1.5	Modified chain execution time and devices utilization table
23/12/2011	1.6	Corrected execution time table. Update for new PI_control IP
23/03/2012	1.7	Update speed evaluation, sliding mode SFOC, removed LPF1 on input currents
06/04/2012	1.8	Removed LPF1 from block diagrams
12/05/2012	1.9	Added FOC description; added

10/07/2013	1.10	BEMF feed forward compensation. Position & speed loop. New 3-wire encoder IP. Scalar loop for sensorless.
13/05/2014	1.11	Hall sensor IP, double 3-phase modulator, enhanced feature for MRT IP.
14/02/2015	1.12	RPFM 3-levels modulator, Resolver sensor IP, extra trigger in current acquisition, direct access to dc_link and currents.
20/05/2016	1.13	Inclusion A/D post processing, 3-phases Clarke transform
6-Jan-17	1.14	Update speed loop control, current loop control. Remove BEMF compensation. Update SMO.
March 21, 2017	1.15	Removed acquisition, simplified pi-control
August 6, 2017	1.16	Added PI controller probes, update resolver
20-Dec-17	1.17	Update module signal interface
June 1, 2018	1.18	Update interface for resolver and MRT acceleration limiter
June 27, 2019	1.19	Update for SMO Vs vector selector
April 12, 2022	1.20	Added decoupling, and BEMF linerization

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