32-bit RISC Microcontroller

TXZ/TXZ+ Family

Reference Manual
Advanced Vector Engine Plus
(A-VE+-B)

Revision 3.1

2021-02

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

Contents

List of Figures

List of Tables

Preface

Related document

Document name
Advanced Programmable Motor Control Circuit
12-bit Analog to Digital Converter
Exception
Clock Control and Operation Mode
Product Information

Conventions

- Numeric formats follow the rules as shown below:
  Hexadecimal: 0xABC
  Decimal: 123 or 0d123 – Only when it needs to be explicitly shown that they are decimal numbers.
  Binary: 0b111 – It is possible to omit the "0b" when the number of bit can be distinctly understood from a sentence.
- "_N" is added to the end of signal names to indicate low active signals.
- It is called "assert" that a signal moves to its active level, "deassert" to its inactive level.
- When two or more signal names are referred, they are described like as [m: n].
  Example: S[3: 0] shows four signal names S3, S2, S1 and S0 together.
- The characters surrounded by ***[ ]*** defines the register.
  Example: ***[ABCD]***
- "n" substitutes suffix number of two or more same kind of registers, fields, and bit names.
  Example: ***[XYZ1]***, ***[XYZ2]***, ***[XYZ3]*** → ***[XYZn]***
- "x" substitutes suffix number or character of units and channels in the Register List.
  In case of unit, "x" means A, B, and C ...
  Example: ***[ADACR0]***, ***[ADBCR0]***, ***[ADCCR0]*** → ***[ADxCR0]***
  In case of channel, "x" means 0, 1, and 2 ...
  Example: ***[T32A0RUNA]***, ***[T32A1RUNA]***, ***[T32A2RUNA]*** → ***[T32AxRUNA]***
- The bit range of a register is written like as [m: n].
  Example: Bit[3: 0] expresses the range of bit 3 to 0.
- The configuration value of a register is expressed by either the hexadecimal number or the binary number.
  Example: ***[ABCD]***<EFG> = 0x01 (hexadecimal), ***[XYZn]***<VW> = 1 (binary)
- Word and Byte represent the following bit length.
  Byte: 8 bits
  Half word: 16 bits
  Word: 32 bits
  Double word: 64 bits
- Properties of each bit in a register are expressed as follows:
  R: Read only
  W: Write only
  R/W: Read and Write are possible
- Unless otherwise specified, register access supports only word access.
- The register defined as reserved must not be rewritten. Moreover, do not use the read value.
- The value read from the bit having default value of "—" is unknown.
- When a register containing both of writable bits and read-only bits is written, read-only bits should be written with their default value, in the cases that default is "—", follow the definition of each register.
- Reserved bits of the Write-only register should be written with their default value.
  In the cases that default is "—", follow the definition of each register.
- Do not use read-modified-write processing to the register of a definition which is different by writing and read out.

Terms and Abbreviation

Some of abbreviations used in this document are as follows:

ADC	Analog to Digital Converter
A-PMD	Advanced Programmable Motor Control Circuit
A-VE+	Advanced Vector Engine Plus
PMD	Programmable Motor Control Circuit
VE	Vector Engine

1. Outlines

The advanced vector engine plus (hereafter referred to as VE) can support vector control of 1 channel motor in 1 unit. The following is a list of functions.

Function Classification	Function	Operation
Calculation function	Basic function	Calculate with fixed point number. Computation task for vector control Interface task for PMD(Note1) and ADC(Note2)
	Current control task	d axis PI control, q axis PI control. - Non-interference control - Output limit by voltage scalar value
	SIN/COS calculation task	Calculate sine and cosine values at phase θ - Phase interpolation and phase clipping possible
	Output voltage transformation task	- Coordinate transformation(Inverse park transformation) - Phase transformation: 2 types(Space vector transformation, Inverse Clarke transformation)
	Output control task	Convert 3-phase voltage to PWM output setting of PMD(Note1) (2 types) - Output limit possibility - Compensation control of dead time
	Trigger generation task	Calculate AD conversion sampling timing set value of PMD from 3-phase duty.
	Input process task	Read the ADC(Note2) conversion results and convert it to fixed point number (2 types). - Current polarity determination (hysteresis/reverse hysteresis)
	Input current transformation task	- Phase transformation (Clarke transformation) - Axis transformation (park transformation)
	Individual function	- Arctangent (ATAN) calculation task - Square root calculation task
Schedule manager	Scheduling control	Execute tasks sequentially - 15 types of schedule
	Start control	- Command start - Repeat start - Start input schedule by end of AD conversion. Start from the standby state after the output schedule ends from the input process task by the ADC conversion end interrupt
Interrupt control	Schedule end interrupt	Interrupt that occurs at the completion of schedule
	Error interrupt	Interrupt that occurs when PWM interrupt is input from PMD(Note1) during output schedule execution
	Task end interrupt	Interrupt that occurs when the specified task completion.
Other function	Output for debugging	Outputs a signal at task transition timing during schedule operation. It can be monitored by PMD's(Note1) debug output function.

Note1: For details of the PMD, refer to the reference manual "Advanced Programmable Motor Control Circuit".

Note2: For details of the ADC, refer to the reference manual "12-bit Analog to Digital Converter".

2. Configuration

2.1. Configuration of the VE

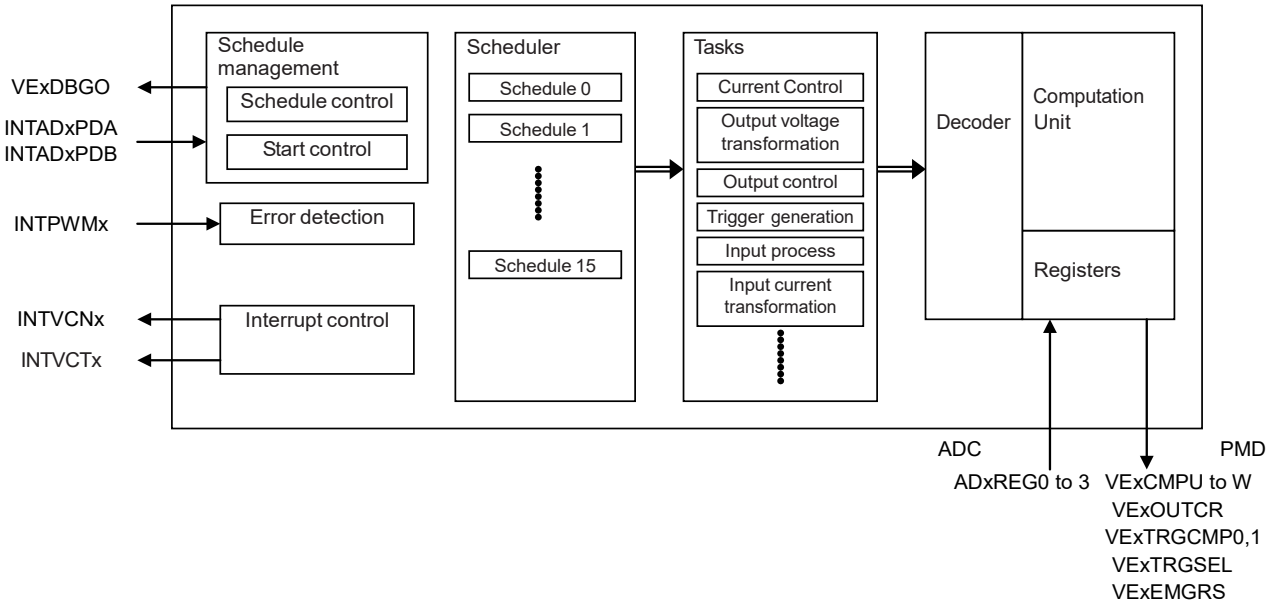


Figure 2.1 Block diagram of VE

Table 2.1 List of signals

No	Signal name	Signal name	I/O	Related Reference manual
1	VExCMPU	U-phase PWM duty (***[VExCMPU]*** register output to PMD)	Output	Product Information
2	VExCMPV	V-phase PWM duty (***[VExCMPV]*** register output to PMD)	Output	Product Information
3	VExCMPW	W-phase PWM duty (***[VExCMPW]*** register output to PMD)	Output	Product Information
4	VExTRGCMP0	Trigger compare 0 (***[VExTRGCMP0]*** register output to PMD)	Output	Product Information
5	VExTRGCMP1	Trigger compare 1 (***[VExTRGCMP1]*** register output to PMD)	Output	Product Information
6	VExTRGSEL	Synchronous trigger output selection (***[VExTRGSEL]*** register output to PMD)	Output	Product Information
7	VExOUTCR	Conduction control / output control (***[VExOUTCR]*** register output to PMD)	Output	Product Information
8	VExEMGRS	EMG release (***[VExEMGRS]*** register output to PMD)	Output	Product Information
9	INTPWMx	PWM interrupt (from PMD)	Input	Product Information
10	ADxREG0	ADC conversion result 0 (Current 1 data from ADC)	Input	Product Information
11	ADxREG1	ADC conversion result 1 (Current 2 data from ADC)	Input	Product Information
12	ADxREG2	ADC conversion result 2 (Current 3 data from ADC)	Input	Product Information
13	ADxREG3	ADC conversion result 3 (DC voltage data from ADC)	Input	Product Information
14	INTADxPDA	ADC conversion end interrupt A (Trigger from ADC)	Input	Product Information
15	INTADxPDB	ADC conversion end interrupt B (Trigger from ADC)	Input	Product Information
16	INTVCNx	Schedule end interrupt	Output	Exception, Product Information
17	INTVCTx	Task end interrupt	Output	Exception, Product Information
18	VExDBGO	Task transition signal (debug output)	Output	Product Information

2.2. Interaction between the VE, PMD, and ADC

As shown in Figure 2.2, the VE allows to swap data between the PMD and ADC directly.

When the ***[PMDxCMPU]***, ***[PMDxCMPV]***, ***[PMDxCMPW]***, ***[PMDxMDOUT]***, ***[PMDxTRGCMP0]***, ***[PMDxTRGCMP1]*** and ***[PMDxTRGSEL]*** registers of the PMD is set to the VE mode with the ***[PMDxMODESEL]*** register, these registers are switched to ***[VExCMPU]***, ***[VExCMPV]***, ***[VExCMPW]***, ***[VExOUTCR]***, ***[VExTRGCMP0]***, ***[VExTRGCMP1]*** and ***[VExTRGSEL]*** of the VE registers(Note). In this case, these VE registers cannot be controlled via the corresponding registers of the PMD by the CPU. These registers can be written by the VE. There are no read/write restrictions for other PMD registers.

The VE can read the value of the conversion result storage registers (***[ADxREG0]***, ***[ADxREG1]***, ***[ADxREG2]***, ***[ADxREG3]***) of the ADC in the input process task. When reading the conversion result, phase information set in the conversion program for each synchronous trigger from the PMD is also read from the ADC.

Note: Even if PMD register is switched to VE register, double buffer / triple buffer function of PMD is valid. Refer to "Advanced Programmable Motor Control Circuit" in the reference manual for details of buffer function.
When executing double buffers / triple buffers with PMD, while executing the task of operating these registers with VE (output control 1 task/output control 2 task/trigger generation task, refer to "3.3 Description of Tasks".) Even if the double buffer / triple buffer execution stage update timing is reached, the execution stage is not updated.

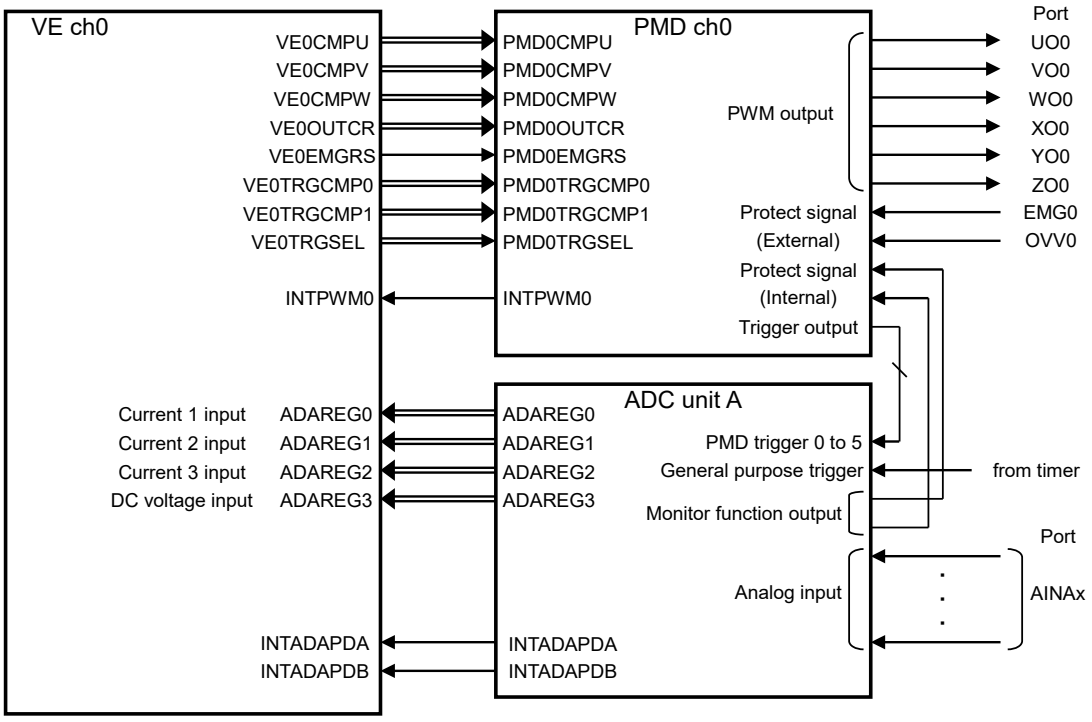


Figure 2.2 Connection diagram of VE and peripherals

3. Description of Operations

3.1. Clock Supply

When you use VE, please set an applicable clock enable bit to "1" (clock supply) in Clock supply and stop register A for fsys (***[CGFSYSENA]***, ***[CGFSYSMENA]***), Clock supply and stop register B for fsys (***[CGFSYSENB]***, ***[CGFSYSMENB]***), and Clock supply and stop register for fc (***[CGFCEN]***).

An applicable register and the bit position vary according to a product. Therefore, the register may not exist with the product. Please refer to "Clock Control and Operation Mode" of the reference manual for the details.

3.2. Schedule Management

Figure 3.1 shows a flowchart of motor control. The VE changes operation status according to the schedule setting (***[VExACTSCH]***) and mode setting (***[VExMODE]***).

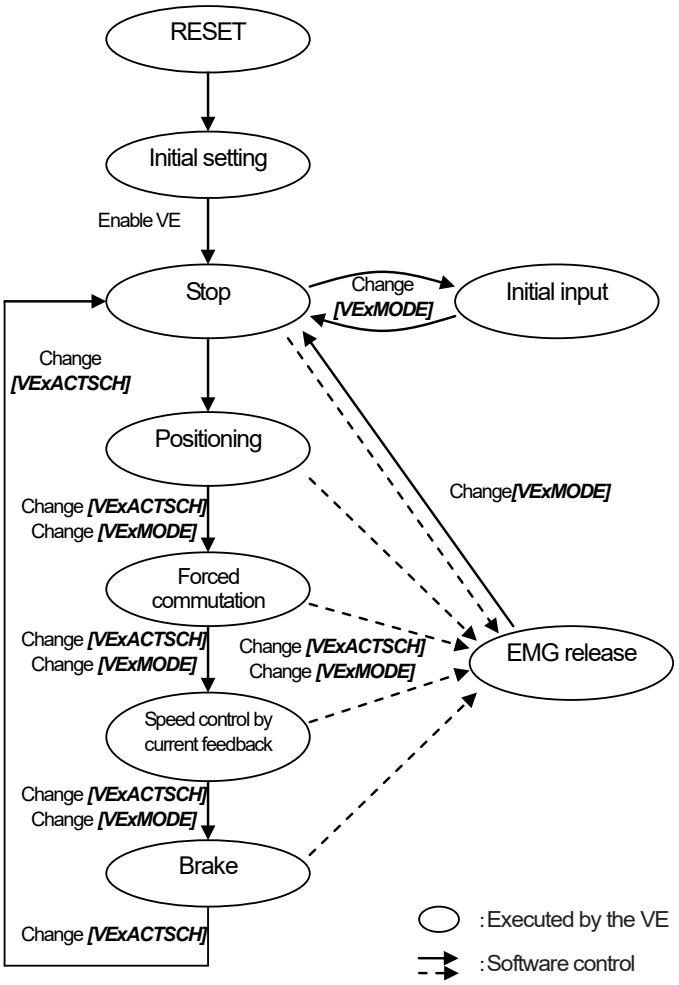


Figure 3.1 Example of Motor Control Operation Status Flowchart

Item	Function
RESET	Resets the MCU.
Initial setting	Specifies the initial setting by software.
Stop	Stops the motor.
Initial input	Samples and stores zero-current data when the motor is at stop.
Positioning	Controls the motor initial position.
Forced commutation	Rotates the motor. The motor is rotated without current feedback control at a specified speed in a motor activation period.
Speed control by current feedback	Control motor rotation speed by current feedback.
Brake	Deceleration control
EMG release	Release the EMG protection state.

3.2.1. Schedule Control

An operation schedule is selected with the ***[VExACTSCH]*** register.

The schedule is comprised of the output schedule handling output process tasks and the input schedule handling input process tasks. Table 3.1 shows the relationship between the schedules and operation tasks.
A task operation is specified with the dedicated register according to the motor control method.

Table 3.1 Execution task in each schedule

Schedule Selection ***[VExACTSCH]*** <VACT>	Output schedule execution tasks								Input schedule execution tasks				Provided by only individual execution	
	Current control	SIN/COS calculation	Output coordinate axis transformation	Output phase transformation 1	Output phase transformation 2	Output control 1	Output control 2	Trigger generation	Input process 1	Input process 2	Input phase transformation	Input coordinate axis transformation	ATAN2 calculation	Square root calculation
	Task 5	Task 6	Task 7	Task 8	Task 11	Task 0	Task 9	Task 1	Task 2	Task 10	Task 3	Task 4	Task 12	Task 13
0:Individual task execution	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)
1: Schedule 1	✓	✓	✓	✓	-	✓	-	✓	✓	-	✓	✓	-	-
2: Schedule 2	✓	✓	✓	✓	-	-	✓	✓	-	✓	✓	✓	-	-
3: Schedule 3	✓	✓	✓	-	✓	-	✓	✓	-	✓	✓	✓	-	-
4: Schedule 4	-	✓	✓	✓	-	✓	-	✓	✓	-	✓	✓	-	-
5: Schedule 5	-	✓	✓	✓	-	-	✓	✓	-	✓	✓	✓	-	-
6: Schedule 6	-	✓	✓	-	✓	-	✓	✓	-	✓	✓	✓	-	-
7: Schedule 7	-	✓	✓	-	✓	✓	-	✓	✓	-	✓	✓	-	-
8: Schedule 8	✓	✓	✓	-	✓	✓	-	✓	✓	-	✓	✓	-	-
9: Schedule 9	-	-	-	-	-	✓	-	✓	✓	-	-	-	-	-
10: Schedule 10	✓	✓	✓	✓	-	✓	-	-	-	-	-	-	-	-
11: Schedule 11	✓	✓	✓	✓	-	-	✓	-	-	-	-	-	-	-
12: Schedule 12	✓	✓	✓	-	✓	-	✓	-	-	-	-	-	-	-
13: Schedule 13	✓	✓	✓	-	✓	✓	-	-	-	-	-	-	-	-
14: Schedule 14	-	-	-	-	-	-	-	✓	✓	-	✓	✓	-	-
15:Schedule 15	-	-	-	-	-	-	-	✓	-	✓	✓	✓	-	-

Note 1: Only the tasks that are specified with ***[VExTASKAPP]*** are executed.
Note 2: ✓: Execution task, -: Non execution task

Table 3.2 Example of setting for typical operation flow

Motor control flow	Setting				
	Schedule setting *[VExACTSCH]* <VACT>	Task selection *[VExTASKAPP]* <VTASK>	Phase interpolation enable *[VExMODE]* <PVIEN>	Output control operation *[VExMODE]* <OCRMD>	Zero-current detection *[VExMODE]* <ZIEN>
Stop	9	0	×	00	0
Initial input	9	0	×	00	1
Positioning	1	5	0	01	0
Forced commutation	1	5	1	01	0
Current feedback speed control	1	5	1	01	0
Brake	4	6	0	01	0
EMG release	9	0	×	11	0
Short circuit brake	4	6	×	10	0

×: don't care

An output schedule starts operation with the command (***[VExCPURUNTRG]***). When all output-related tasks are completed, the VE enters standby state and waits for a start trigger (***[VExTRGMODE]*** setting).

An input schedule starts operation by a start trigger. When all input-related tasks are completed, the VE generates an interrupt to the CPU and enters halt state. However, if the number of repeating of the schedule (***[VExREPTIME]***) is set to "2" or more, the output schedules continue execution. An interrupt does not occur until the schedules have been executed for the specified number of times.

Note: Repeat setting is not available in Schedule 10 to Schedule 15. (The schedule ends once even if the condition is ***[VExREPTIME]*** ≥ 2.)

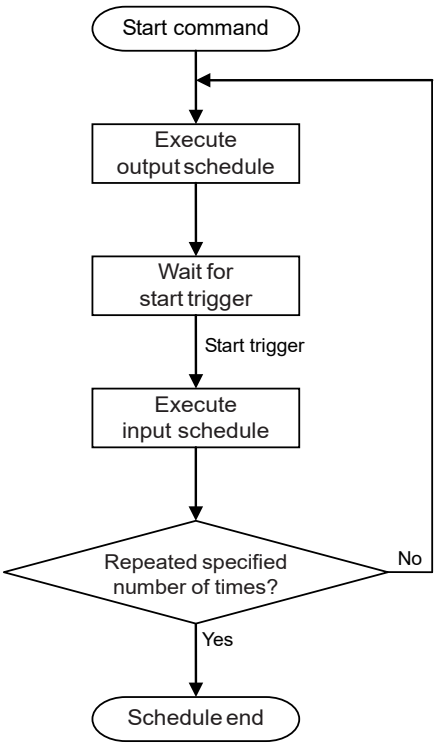


Figure 3.2 Operation schedule flowchart

3.2.2. Start Control

Before the schedule is started, set the VE to enable (***[VExEN]***<VEEN> = 1), and then set the operation schedule selection register (***[VExACTSCH]***), task selection register (***[VExTASKAPP]***) and operation schedule repeat specify register (***[VExREPTIME]***). After that, the schedule can be executed as follows:

A schedule of the VE is comprised of output schedules and input schedules. Typically, the VE enters wait state after an output schedule is complete. At this time, if a startup trigger occurs, an input schedule is executed.

The output schedule and the input schedule are started on different conditions below:

- Start conditions for the output schedule
  - Starting by the command. A task (***[VExTASKAPP]***) is specified with ***[VExCPURUNTRG]***.
  - Repeat setting. Starting after the input schedules have been executed for the number of times (***[VExREPTIME]*** ≥ 1).
- Start conditions for the input schedule
  - Starting an input process task that is started from wait state by a trigger (specified with ***[VExTRGMODE]***). In this case, the VE is in wait state after the output schedule is complete.
  - Starting by the command. A task (***[VExTASKAPP]***) is specified with ***[VExCPURUNTRG]***.

Table 3.3 Schedule-related tasks

Register name	Function	
[VExACTSCH]	Operation schedule selection	0x0: Only the task specified with ***[VExTASKAPP]*** is executed. 0x1: Schedule 1 is executed. 0x2: Schedule 2 is executed. 0x3: Schedule 3 is executed. 0x4: Schedule 4 is executed. 0x5: Schedule 5 is executed. 0x6: Schedule 6 is executed. 0x7: Schedule 7 is executed. 0x8: Schedule 8 is executed. 0x9: Schedule 9 is executed. 0xA: Schedule 10 is executed. 0xB: Schedule 11 is executed. 0xC: Schedule 12 is executed. 0xD: Schedule 13 is executed. 0xE: Schedule 14 is executed. 0xF: Schedule 15 is executed.
[VExTASKAPP]	Starting task designation	Specifies the task number that can be executed among the selected operation schedule.
[VExREPTIME]	The number of repeating of the schedule	Sets the number from "1" to "15". Note: When one-time execution is specified, set "1".If "0" is set, the schedule is not executed.
[VExTRGMODE]	Start trigger mode selection	Selects the input schedule trigger. Sets INTADxPDA or INTADxPDB.

3.2.3. Interrupt Control

The VE has a schedule end interrupt (INTVCNx) that occurs at the completion of the schedule, and a task completion interrupt (INTVCTx) that occurs at the completion of specified task.

- Schedule end interrupt
  (1) Specifies the schedule with the operation schedule selection register (***[VExACTSCH]***). Sets the command (***[VExCPURUNTRG]*** = 1).

  (2) When the schedules have been executed for the specified times (***[VExREPTIME]***), an INTVCNx interrupt occurs.

  (3) If the error detection interrupt control is enabled (***[VExERRINTEN]***<VERREN> = 1), a PWM interrupt of the PMD circuit occurs while output schedule is being executed. Then an INTVCNx interrupt occurs and an error flag (***[VExERRDET]***<VERRD>) is set to "1".

- Task end interrupt
  (1) Specifies a task triggered of a task end interrupt (***[VExTASKAPP]***<VITASK>). Sets task end interrupt control to enable (***[VExERRINTEN]***<INTTEN> = 1).

  (2) Starts the schedule by the command (***[VExCPURUNTRG]*** = 1). Sets a INTVCTx interrupt to occur at the completion of the task specified with <VITASK>

3.3. Description of Tasks

This subsection describes the outline of each task operation in the schedule.
Table 3.4 shows the task numbers that are used to specify the execution task and the start task.

Table 3.4 List of tasks

Task		Task function	Task number
Output schedule	Current control	PI control for d-axis/q-axis (PI control output limit enable) Non-interference control for d-axis and q-axis, voltage scalar limitation	5
	SIN/COS calculation	Sine/cosine calculation Phase interpolation (with clipping function)	6
	Output coordinate axis transformation	Inverse Park transformation	7
	Output phase transformation 1	Transforms from 2-phase to 3-phase [SVM]	8
	Output phase transformation 2	Transforms from 2-phase to 3-phase [Inverse Clarke transformation]	11
	Output control 1	Converts the data to PMD setting format. Switches PWM Shift 1. Limits the PWM output. Sets the dead time compensation control.	0
	Output control 2	Converts the data to PMD setting format. Switches PWM Shift 2. Limits the PWM output. Sets the dead time compensation control.	9
	Trigger generation	Generates the synchronous trigger timing for ADC.	1
Input schedule	Input process 1	Corresponds to sensor, 3-shunt and 1-shunt current detection.1-shunt is corresponded in the case of PWM Shift prohibited or PWM Shift 1. Captures the result of AD conversion and converts them into fixed-point format. Specifies the hysteresis width to determine the current polarity.	2
	Input process 2	Corresponds to sensor, 3-shunt and 1-shunt current detection.1-shunt is corresponded in the case of PWM Shift 2. Captures the result of AD conversion and converts them into fixed-point format. Specifies the hysteresis width to determine the current polarity.	10
	Input phase transformation	Transforms from 3-phase to 2-phase.	3
	Input coordinate axis transformation	Performs Park transformation Calculates the declination angle of the current vector or the induced voltage vector on the d-q coordinate.	4
ATAN2 calculation		Calculates the arctangent.	12
SQRT calculation		Calculates the square root.	13

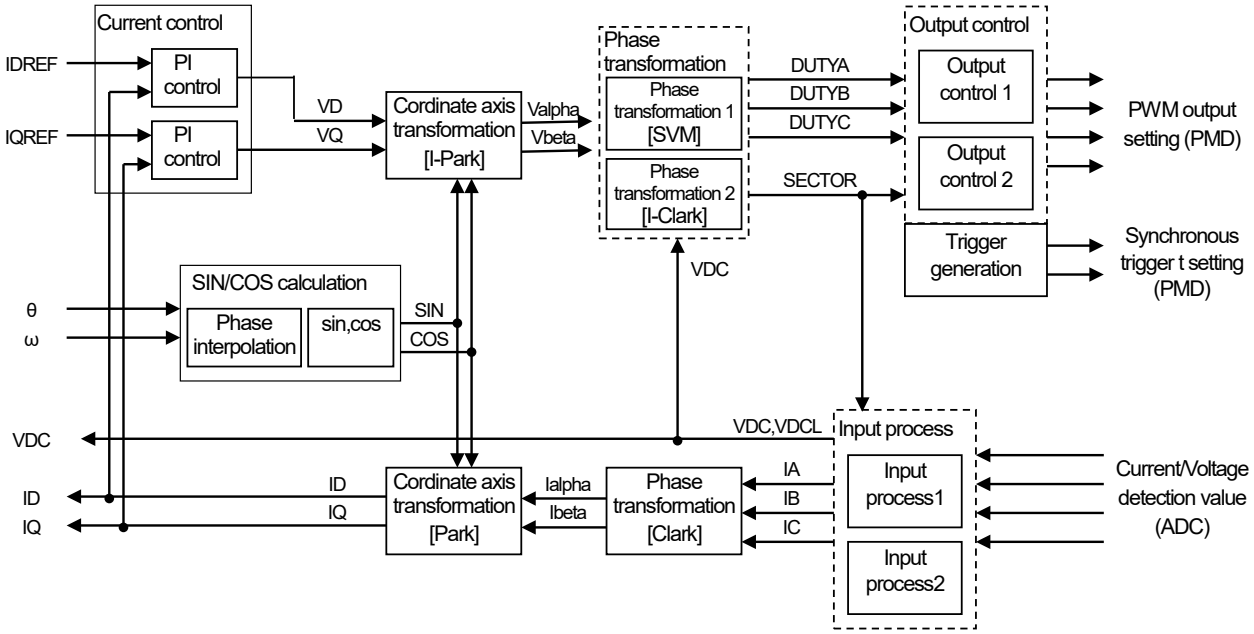


Figure 3.3 Relationship diagram of tasks

3.3.1. Current Control (Task 5)

The current control task is comprised of a PI control unit for d-axis current and a PI control unit for q-axis current, and calculates the d-axis and q-axis voltages respectively.

The expansion control enables the non-interference control and the voltage scalar limitation that controls d-axis and q-axis together.

a. d-axis current PI control

<Equation>

```
[PI control]
Δid = [VExIDREF] - [VExID]                    ; Calculates between the current reference value
                                                and the current feedback
kpg = ([VExCIDKG]<CIDKPG> setting)            ; Range of proportional coefficient
kig = ([VExCIDKG]<CIDKIG> setting)            ; Range of Integral coefficient
limit = [VExPIOLIM]
if (kpg ≥ 2)        n = kpg                   ; Selected range is more than twice
                                                      Save Range of proportional coefficient
                    limit = limit / kpg       ;       Limit correction
                    kig = kig / kpg           ;       Range correction of integral
                    kpg = 1                   ;       Range correction of proportional
else                n = 1
cidkp = [VExCIDKP] × kpg                      ; Proportional coefficient
cidki = [VExCIDKI] × kig                      ; Integral coefficient
vdi0 = cidki × Δid + VDI                      ; Calculates the integral components.
vd0 = cidkp × Δid + vdi0                      ; Calculates the voltage by adding the
                                                proportional components.

[PI control output limitation]
if (vd0 > limit)                              ; The upper-limit value
        vd = limit
        [VExMCTLF]<PIDOVF> = 1
else if (vd0 < -limit)                        ; The lower-limit value
        vd = -limit
        [VExMCTLF]<PIDOVF> = 1
else    vd = vd0
[VExVD] = vd × n                              ; Save more than a double proportional range
                                                to be compensated.

[Anti-windup]
Δvd = vd - vd0                                ; Calculates the difference between the d-axis
                                                voltage and the limit.
VDI = vdi0 + Δvd × ([VExMODE]<AWUMD> setting) ; Reflects the above difference
                                                to the integral component.
```

	Register name	Function	Detail
Input	***[VExID]***	d-axis current	32-bit fixed-point data (31 fractional bits)
	[VExIDREF]	d-axis current reference value	16-bit fixed-point data(15 fractional bits)
	[VExCIDKP]	Proportional coefficient	16-bit data
	[VExCIDKI]	Integral coefficient	16-bit data
	[VExCIDKG]	[3:0] Sets a range of d-axis PI control integral coefficient	<CIDKIG> Range of Integral coefficient 0000: 1 / 1 0001: 1 / 2^4 0010: 1 / 2^8 0011: 1 / 2^{12} 0100: 1 / 2^{16} 0101 to 1111: Reserved
		[11:8] Sets a range of d-axis PI control proportional coefficient	<CIDKPG> Range of proportional coefficient 0000: 1 / 1 0001: 1 / 2^4 0010: 1 / 2^8 0011: 1 / 2^{12} 0100: 1 / 2^{16} 0101 to 1000: Reserved 1001: 2 1010: 2^2 1011: 2^3 1100: 2^4 1101 to 1111: Reserved
	[VExPIOLIM]	PI control output limitation	16-bit fixed-point data (15 fractional bits) Valid range: 0x0000 to 0x7FFF Output limitation is disabled when ***[VExPIOLIM]*** = 0x0000.
	[VExMODE]	[9:8] Anti-windup ratio setting when output is limited.	<AWUMD> 00: Disabled 01: 1 / 4 10: 1 / 2 11: 1 / 1
Output	***[VExVD]***	d-axis voltage	32-bit fixed-point data (31 fractional bits)
	[VExMCTLF]	[8] Flag for d-axis output limitation on PI control	<PIDOVF> 0: |d-axis output on PI control| ≤ ***[VExPIOLIM]*** 1: |d-axis output on PI control| > ***[VExPIOLIM]***
Internal	VDI	Integral component of d-axis voltage	64-bit fixed-point data (63 fractional bits)

Note: The VDI is comprised of 64 bits. The upper is used for the ***[VExVDIH]*** register and the lower is used for the ***[VExVDILH]*** register

b. q-axis current PI control

<Equation>

[PI control]

Δiq = ***[VExIQREF] - [VExIQ]*** ; Calculates between the current reference value and the current feedback

kpg = (***[VExCIQKG]***<CIQKPG> setting) ; Range of proportional coefficient

kig = (***[VExCIQKG]***<CIQKIG> setting) ; Range of Integral coefficient

limit = ***[VExPIOLIM]***

if (kpg ≥ 2) n = kpg ; Selected range is more than twice, Save Range of proportional coefficient

limit = limit / kpg ; Limit correction

kig = kig / kpg ; Range correction of integral

kpg = 1 ; Range correction of proportional

else n = 1

ciqkp = ***[VExCIQKP]*** × kpg ; Proportional coefficient

ciqki = ***[VExCIQKI]*** × kig ; Integral coefficient

vqi0 = ciqki × Δiq + VQI ; Calculates the integral components.

vq0 = ciqkp × Δiq + vqi0 ; Calculates the voltage by adding the proportional components.

[PI control output limitation]
if (vq0 > limit) ; The upper-limit value
vq = limit
[VExMCTLF]<PIQOVF> = 1
else if (vq0 < -limit) ; The lower-limit value
vq = -limit
[VExMCTLF]<PIQOVF> = 1
else vq = vq0
[VExVQ] = vq × n ; Save more than a doubled proportional range to be compensated.

[Anti-windup]
Δvq = vq - vq0 ; Calculates the difference between the q-axis voltage and the limit.
VQI = vqi0 + Δvq × (***[VExMODE]***<AWUMD> setting) ; Reflects the above difference to the integral component.

	Register name	Function	Detail
Input	***[VExIQ]***	q-axis current	32-bit fixed-point data (31 fractional bits)
	[VExIQREF]	q-axis current reference value	16-bit fixed-point data(15 fractional bits)
	[VExCIQKP]	Proportional coefficient	16-bit data
	[VExCIQKI]	Integral coefficient	16-bit data
	[VExCIQKG]	[3:0] Sets a range of q-axis PI control integral coefficient	<CIQKIG> Range of Integral coefficient 0000: 1 / 1 0001: 1 / 2^4 0010: 1 / 2^8 0011: 1 / 2^{12} 0100: 1 / 2^{16} 0101 to 1111: Reserved
		[11:8] Sets a range of q-axis PI control proportional coefficient	<CIQKPG> Range of proportional coefficient 0000: 1 / 1 0001: 1 / 2^4 0010: 1 / 2^8 0011: 1 / 2^{12} 0100: 1 / 2^{16} 0101 to 1000: Reserved 1001: 2 1010: 2^2 1011: 2^3 1100: 2^4 1101 to 1111: Reserved
	[VExPIOLIM]	PI control output limitation	16-bit fixed-point data (15 fractional bits) Valid range: 0x0000 to 0x7FFF Output limitation is disabled when ***[VExPIOLIM]*** = 0x0000.
	[VExMODE]	[9:8] Anti-windup ratio setting when output is limited.	<AWUMD> 00: Disabled 01: 1 / 4 10: 1 / 2 11: 1
Output	***[VExVQ]***	q-axis voltage	32-bit fixed-point data (31 fractional bits)
	[VExMCTLF]	[9] Flag for q-axis output limitation on PI control	<PIQOVF> 0: |q-axis output on PI control| ≤ ***[VExPIOLIM]*** 1: |q-axis output on PI control| > ***[VExPIOLIM]***
Internal	VQI	Integral component of q-axis voltage	64-bit fixed-point data (63 fractional bits)

Note: The VQI is comprised of 64 bits. The upper is used for the ***[VExVQIH]*** register and the lower is used for the ***[VExVQILH]*** register

c. Non-interference control

The result of PI control is corrected using the results of the interference to d-axis and q-axis based on motor voltage equation.

<Equation>

if (***[VExMODE]***<T5ECEN> = 1) ; Expansion control is enabled
  ld = ***[VExCLD]*** × (***[VExCLG]*** setting) ; d-axis inductance
  lq = ***[VExCLQ]*** × (***[VExCLG]*** setting) ; q-axis inductance
  phi = ***[VExCPHI]*** × (***[VExCPHIG]*** setting) ; Interlinkage magnetic flux
  id = ***[VExID]*** ; Feedback current
  iq = ***[VExIQ]***
  if (***[VExFMODE]***<IDQSEL> = 1) id = ***[VExIDREF]*** ; Inputs a command value
    iq = ***[VExIQREF]***
  [VExVDE] = - ***[VExOMEGA]*** × iq × lq ; Calculates the interference to d-axis
  [VExVQE] = ***[VExOMEGA]*** × id × ld + ***[VExOMEGA]*** × phi ; Calculates the interference to q-axis
  if (***[VExMODE]***<NICEN> = 1) ; Non-interference control is enabled
    [VExVD] = ***[VExVD]*** + ***[VExVDE]***
    [VExVQ] = ***[VExVQ]*** + ***[VExVQE]***

	Register name	Function	Detail
Input	***[VExVD]***	d-axis voltage	32-bit fixed-point data (31 fractional bits)
	[VExVQ]	q-axis voltage	
	[VExID]	d-axis current	
	[VExIQ]	q-axis current	
	[VExCLD]	Motor d-axis inductance	16-bit fixed-point data (11 fractional bits)
	[VExCLQ]	Motor q-axis inductance	
	[VExCPHI]	Motor Interlinkage magnetic flux	
	[VExCLG]	Motor Inductance range setting	000: 1 / 1 001: 1 / 2^4 010: 1 / 2^8 011: 1 / 2^{12} 100: 1 / 2^{16} 101 to 111: Reserved
	[VExCPHIG]	Motor Interlinkage magnetic flux range setting	
	[VExOMEGA]	Rotational speed	16-bit fixed-point data (15 fractional bits)
	[VExFMODE]	[4] Selects Non-interference control input	<IDQSEL> 0: Use ***[VExID]***, ***[VExIQ]*** 1: Use ***[VExIDREF]***, ***[VExIQREF]***
	[VExMODE]	[11] Non-interference control enable	<NICEN> 0: Non-interference control is disabled. 1: Non-interference control is enabled.
		[10] Expansion control enable	<T5ECEN> 0: Expansion control is disabled. (Non-interference control is disabled.) 1: Expansion control is enabled.
Output	***[VExVDE]***	Non-interference correction voltage for d-axis	16-bit fixed-point data (15 fractional bits)
	[VExVQE]	Non-interference correction voltage for q-axis	
	[VExVD]	d-axis voltage	32-bit fixed-point data (31 fractional bits)
	[VExVQ]	q-axis voltage	

d. Voltage scalar limitation

The voltage scalar limitation controls d-axis voltage and q-axis voltage, so as to be lower the composite value (the square root of $VD^2 + VQ^2$) that is comprised of d-axis voltage and q-axis voltage than the limit value.

<Equation>

```
if (VExMODE<T5ECEN> = 1)                          ; Expansion control is enabled.
  [VDQ calculation]
  if ((VExVD^2 + VExVQ^2)> VExVSLIM^2)            ; Confirm the excess
    if (VExFMODE<VSLIMMD> = 00)                   ; Voltage scalar limitation is disabled
        VExVDQ = SQRT(VExVD^2 + VExVQ^2)
    else if (VExFMODE<VSLIMMD> = 01)              ; Scalar limitation on the d-axis direction
        VExVDQ = SQRT(VExVSLIM^2 + VExVQ^2)
    else if (VExFMODE<VSLIMMD> = 10)              ; Scalar limitation on the q-axis direction
        VExVDQ = SQRT(VExVSLIM^2 + VExVD^2)
    else if (VExFMODE<VSLIMMD> = 11)              ; dq proportional scalar limitation
        VExVDQ = SQRT(VExVD^2 + VExVQ^2)
  else  VExVDQ = SQRT(VExVD^2 + VExVQ^2)          ; Calculation of voltage scalar value
        Note:  SQRT is the square root calculation

  [Calculation for the declination angle]
  x = |VExVQ|
  y = |VExVD|
  VExVDELTA = ATAN(x, y)
        Note:  ATAN is arctangent calculation

  [Limitation calculation for each axis]
  if (VExFMODE<VSLIMMD> = 00)                     ; Voltage scalar limitation is disabled
    vdlim = VExVSLIM
    vqlim = VExVSLIM
  else if (VExFMODE<VSLIMMD> = 01)                ; Scalar limitation on the d-axis direction
    vdlim = VExVDQ
    vqlim = VExVSLIM
  else if (VExFMODE<VSLIMMD> = 10)                ; Scalar limitation on the q-axis direction
    vdlim = VExVSLIM
    vqlim = VExVDQ
  else if (VExFMODE<VSLIMMD> = 11)                ; dq proportional scalar limitation
    vdlim = VExVSLIM × SIN(VExVDELTA)
    vqlim = VExVSLIM × COS(VExVDELTA)

  [Limitation process]
  if (VExVD > vdlim)
    VExVD = vdlim                                 ; Process for the upper-limit of d-axis
    VExMCTLF<VSOVF> = 1
  else if (VExVD < -vdlim)
    VExVD = -vdlim                                ; Process for the lower-limit of d-axis
```

```
   [VExMCTLF]<VSOVF> = 1
if ([VExVQ] > vqlim)
   [VExVQ] = vqlim                          ; Process for the upper-limit of q-axis
   [VExMCTLF]<VSOVF> = 1
else if ([VExVQ] < -vqlim)
   [VExVQ] = -vqlim                         ; Process for the lower-limit of q-axis
   [VExMCTLF]<VSOVF> = 1

[Correction process]
   [VExVD] = [VExVD] + [VExVDCRC]           ; d-axis correction
   [VExVQ] = [VExVQ] + [VExVQCRC]           ; q-axis correction
```

	Register name	Function	Detail
Input	***[VExVD]***	d-axis voltage	32-bit fixed-point data (31 fractional bits)
	[VExVQ]	q-axis voltage	
	[VExVSLIM]	The limit value of voltage scalar	16-bit fixed-point data (15 fractional bits) 0x0000 to 0x7FFF Limitation is disabled when ***[VExVSLIM]*** = 0x0000
	[VExVDCRC]	The value of d-axis voltage correction	16-bit fixed-point data (15 fractional bits)
	[VExVQCRC]	The value of q-axis voltage correction	
	[VExMODE]	[10] Expansion control enable	<T5ECEN> 0: Expansion control is disabled.(Scalar limitation is disabled.) 1: Expansion control is enabled.
	[VExFMODE]	[11:10] Limitation mode setting	<VSLIMMD> 00: Scalar limitation is disabled. (Limitation in each axis is enabled.) 01: Scalar limitation is enabled. (Limitation on the d-axis direction.) 10: Scalar limitation is enabled. (Limitation on the q-axis direction.) 11: Scalar limitation is enabled. (dq proportional limitation.)
Output	***[VExVDQ]***	The value of voltage scalar or the value of axis direction limitation.	16-bit fixed-point data (15 fractional bits)
	[VExVDELTA]	The voltage declination angle	16-bit data 0x0000 to 0x4000 (0-degree to 90-degree)
	[VExVD]	d-axis voltage	32-bit fixed-point data (31 fractional bits)
	[VExVQ]	q-axis voltage	
	[VExMCTLF]	[10] Indicates the excess flag for voltage scalar limitation.	<VSOVF> 0:non excess 1:excess

3.3.2. SIN/COS calculation (Task 6)

The SIN/COS calculation task executes phase interpolation calculation and SIN/COS calculation.

Phase interpolation calculates the rotation speed by integrating with the PWM period. It is executed only when phase interpolation is enabled (***[VExMODE]***<PVIEN> = 1).

a. Phase interpolation

<Equation>

theta0 = ***[VExOMEGA]*** × ***[VExTPWM]*** + ***[VExTHETA]*** ; Calculates the value of phase interpolation

theta0 = theta0 & 0x0000FFFF

if (***[VExMODE]***<CLPEN> = 1) ; Enables the clipping

  if (***[VExOMEGA]*** ≥ 0) ; On positive rotation

    if (***[VExTHETA]*** ≤ ***[VExTHTCLP]*** ≤ theta0) theta0 = ***[VExTHTCLP]***

    else if (theta0 ≤ ***[VExTHETA]*** ≤ ***[VExTHTCLP]***) theta0 = ***[VExTHTCLP]***

    else if (***[VExTHTCLP]*** ≤ theta0 ≤ ***[VExTHETA]***) theta0 = ***[VExTHTCLP]***

  else if (***[VExOMEGA]*** < 0) ; On inverse rotation

    if (theta0 ≤ ***[VExTHTCLP]*** ≤ ***[VExTHETA]***) theta0 = ***[VExTHTCLP]***

    else if (***[VExTHTCLP]*** ≤ ***[VExTHETA]*** ≤ theta0) theta0 = ***[VExTHTCLP]***

    else if (***[VExTHETA]*** ≤ theta0 ≤ ***[VExTHTCLP]***) theta0 = ***[VExTHTCLP]***

if (***[VExMODE]***<PVIEN> = 1) ***[VExTHETA]*** = theta0 ; Updates the ***[VExTHETA]*** value when phase interpolation is enabled

	Register name	Function	Detail
Input	***[VExTHETA]***	Phase θ	16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)
	[VExOMEGA]	Rotational speed	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	[VExTPWM]	PWM period rate	16-bit data
	[VExTHTCLP]	The value of clipped phase	16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)
	[VExMODE]	[0] Phase interpolation enable	<PVIEN> 0: Phase interpolation is disabled 1: Phase interpolation is enabled
		[7] Phase clipping control	<CLPEN> 0: Clipping is disabled 1: Clipping is enabled
Output	***[VExTHETA]***	Phase θ	16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)

b. SIN/COS calculation

<Equation>

[VExSINM] = ***[VExSIN]*** ; Stores the previous value (for input process)
[VExCOSM] = ***[VExCOS]***
[VExSIN] = SIN (***[VExTHETA]***) ; Calculates the SIN and COS values
[VExCOS] = SIN (***[VExTHETA]*** + 1 / 4)
if (***[VExFMODE]***<MREGDIS> = 1) ; Confirms that the previous value is not maintained
[VExSINM] = ***[VExSIN]***
[VExCOSM] = ***[VExCOS]***

Note: SIN: Sine calculation

	Register name	Function	Detail
Input	***[VExTHETA]***	Phase θ	16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)
	[VExFMODE]	[9] Selects to store the previous value of SIN and COS	<MREGDIS> 0: Previous value is maintained. 1: Previous value is not maintained.
Output	***[VExSIN]***	The sine value of θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	[VExCOS]	The cosine value of θ	
	[VExSINM]	Previous sine value	
	[VExCOSM]	Previous cosine value	

3.3.3. Output Voltage Transformation (Coordinate Axis Transformation/Phase Transformation)

Output voltage transformation is performed in two steps: coordinate axis transformation and phase transformation.
There are two types of phase transformation: Space vector transformation and inverse Clarke transformation.

3.3.3.1. Output coordinate axis transformation (Task 7)

In the output coordinate axis task, α-axis and β-axis voltages are calculated based on d-axis voltage, q-axis voltage, sinθ and cosθ.

<Equation>

$$[VExVALPHA] = [VExCOS] \times [VExVD] - [VExSIN] \times [VExVQ] \quad ; \text{Calculates } V\alpha$$
$$[VExVBETA] = [VExSIN] \times [VExVD] + [VExCOS] \times [VExVQ] \quad ; \text{Calculates } V\beta$$

	Register name	Function	Detail
Input	*[VExVD]*	d-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	[VExVQ]	q-axis voltage	
	[VExSIN]	The sine value of θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	[VExCOS]	The cosine value of θ	
Output	*[VExVALPHA]*	α-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	[VExVBETA]	β-axis voltage	

3.3.3.2. Output phase transformation 1 (Space vector transformation) (Task 8)

Output phase transformation 1 determines a sector using α-axis voltage and β-axis voltage. This task calculates the duties of a-phase voltage, b-phase voltage, and c-phase voltage based on space vector transformation in each sector.

This task can be selected either from 2-phase or 3-phase modulation as a modulation type.

a. Sector determination

<Equation>

```
[VExSECTORM] = [VExSECTOR]                          ; Stores the previous value
valpha = [VExVALPHA]
vbeta = [VExVBETA]
if (valpha ≥ 0 & vbeta ≥ 0)
   if (|valpha| ≥ |vbeta| × √(1 / 3))
      if (|valpha| × √(1 / 3) ≥ |vbeta|)    [VExSECTOR] = 0
      else                                  [VExSECTOR] = 1
   else                                     [VExSECTOR] = 2
else if (valpha < 0 & vbeta ≥ 0)
   if (|valpha| < |vbeta| × √(1 / 3))       [VExSECTOR] = 3
   if (|valpha| × √(1 / 3) < |vbeta|)       [VExSECTOR] = 4
   else                                     [VExSECTOR] = 5
else if (valpha < 0 & vbeta < 0)
   if (|valpha| ≥ |vbeta| × √(1 / 3))
      if (|valpha| × √(1 / 3) ≥ |vbeta|)    [VExSECTOR] = 6
      else                                  [VExSECTOR] = 7
   else                                     [VExSECTOR] = 8
else if (valpha ≥ 0 & vbeta < 0)
   if (|valpha| < |vbeta| × √(1 / 3))       [VExSECTOR] = 9
   else if (|valpha| × √(1 / 3) < |vbeta|)  [VExSECTOR] = 10
   else                                     [VExSECTOR] = 11
if ([VExFMODE]<MREGDIS> = 1)                [VExSECTORM] = [VExSECTOR]   ; Confirm that the previous
                                                                           value is disabled
```

	Register name	Function	Detail
Input	*[VExVALPHA]*	α-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	[VExVBETA]	β-axis voltage	
	[VExFMODE]	[9] Selects to store the previous value of SECTOR	<MREGDIS> 0: Previous sector value is maintained. 1: Previous sector value is not maintained.
Output	*[VExSECTOR]*	Sector information	4-bit data
	[VExSECTORM]	Previous *[VExSECTOR]*	

b. Space vector transformation
(This example describes only the case where 3-phase modulation is used and ***[VExSECTOR]*** = 0, 1.)

<Equation>

kc = √3 ; Relative transformation
if (***[VExFMODE]***<CCVMD> = 1) kc = √2 ; Absolute transformation
if (***[VExSECTOR]*** = 0, 1)
t1 = kc /***[VExVDC]*** × (√3 / 2 × ***[VExVALPHA]*** -1 / 2 × ***[VExVBETA]***) ; Calculates the t1 period.
t2 = kc /***[VExVDC]*** × ***[VExVBETA]***) ; Calculates the t2 period.
t3 = 1 -t1 -t2 ; Calculates the zero-vector period
if (***[VExFMODE]***<C2PEN> = 0) ; 3-phase modulation
dutya = t1 + t2 + t3 / 2
dutyb = t2 + t3 / 2
dutyc = t3 / 2
else ; 2-phase modulation
dutya = t1 + t2
dutyb = t2
dutyc = 0

[VExVDUTYA] = dutya
[VExVDUTYB] = dutyb
[VExVDUTYC] = dutyc
[VExTMPREG5] = t3

	Register name	Function	Detail
Input	***[VExVALPHA]***	α-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	[VExVBETA]	β-axis voltage	
	[VExVDC]	DC supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	[VExSECTOR]	Sector information	4-bit data
	[VExFMODE]	[0] Modulation mode	<C2PEN> 0: 3-phase modulation 1: 2-phase modulation
		[13] Phase transformation mode setting	<CCVMD> 0: Relative transformation 1: Absolute transformation
Output	***[VExVDUTYA]***	a-phase voltage duty	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	[VExVDUTYB]	b-phase voltage duty	
	[VExVDUTYC]	c-phase voltage duty	
	[VExTMPREG5]	Zero-vector duty	

3.3.3.3. Output phase transformation 2 (Inverse Clarke transformation) (Task 11)

Output phase transformation 2 determines a sector using α-axis and β-axis voltages. This task calculates the duties of a-phase, b-phase and c-phase voltages based on inverse Clarke transformation. This task supports only 3-phase modulation as a modulation type.

In addition, if ***[VExFMODE]***<PHCVDIS> is set to "1", this task can calculate the duty of 2- phase voltage.

a. Sector determination

<Equation>

[VExSECTORM] = ***[VExSECTOR]*** ; Stores the previous value
valpha = ***[VExVALPHA]***
vbeta = ***[VExVBETA]***
if (valpha ≥ 0 & vbeta ≥ 0)
  if (|valpha| ≥ |vbeta| × √(1 / 3))
    if (|valpha| × √(1 / 3) ≥ |vbeta|) ***[VExSECTOR]*** = 0
    else ***[VExSECTOR]*** = 1
  else ***[VExSECTOR]*** = 2
else if (valpha < 0 & vbeta ≥ 0)
  if (|valpha| < |vbeta| × √(1 / 3)) ***[VExSECTOR]*** = 3
  if (|valpha| × √(1 / 3) < |vbeta|) ***[VExSECTOR]*** = 4
  else ***[VExSECTOR]*** = 5
else if (valpha < 0 & vbeta < 0)
  if (|valpha| ≥ |vbeta| × √(1 / 3))
    if (|valpha| × √(1 / 3) ≥ |vbeta|) ***[VExSECTOR]*** = 6
    else ***[VExSECTOR]*** = 7
  else ***[VExSECTOR]*** = 8
else if (valpha ≥ 0 & vbeta < 0)
  if (|valpha| < |vbeta| × √(1 / 3)) ***[VExSECTOR]*** = 9
  else if (|valpha| × √(1 / 3) < |vbeta|) ***[VExSECTOR]*** = 10
  else ***[VExSECTOR]*** = 11
if (***[VExFMODE]***<MREGDIS> = 1) ***[VExSECTORM]*** = ***[VExSECTOR]*** ; Confirm that the previous value is disabled

	Register name	Function	Detail
Input	***[VExVALPHA]***	α-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	[VExVBETA]	β-axis voltage	
	[VExFMODE]	[9] Selects to store the previous value of SECTOR	<MREGDIS> 0: Previous sector value is maintained. 1: Previous sector value is not maintained.
Output	***[VExSECTOR]***	Sector information	4-bit data
	[VExSECTORM]	Previous ***[VExSECTOR]***	

b. Inverse Clarke transformation

<Equation>

kc = 1 ; Relative transformation

if (***[VExFMODE]***<CCVMD> = 1) Kc = √(2 / 3) ; Absolute transformation

if (***[VExFMODE]***<PHCVDIS> = 0) ; 3-phase conversion is enabled.

[VExVDUTYA] = kc / ***[VExVDC]*** × ***[VExVALPHA]*** + 1 / 2 ; Va Duty

[VExVDUTYB] = kc / ***[VExVDC]*** × (-1 / 2 × ***[VExVALPHA]*** + √3 / 2 × ***[VExVBETA]***) + 1 / 2 ; Vb Duty

[VExVDUTYC] = kc / ***[VExVDC]*** × (-1 / 2 × ***[VExVALPHA]*** -√3 / 2 × ***[VExVBETA]***) + 1 / 2 ; Vc Duty

else ; Phase conversion is disabled.

[VExVDUTYA] = 1 /***[VExVDC]*** × ***[VExVALPHA]*** + 1 / 2 ; Va Duty

[VExVDUTYB] = 1 /***[VExVDC]*** × ***[VExVBETA]*** + 1 / 2 ; Vb Duty

	Register name	Function	Detail
Input	***[VExVALPHA]***	α-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	[VExVBETA]	β-axis voltage	
	[VExVDC]	DC supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	[VExFMODE]	[12] Phase transformation setting	<PHCVDIS> 0: 2-3 phase transformation is enabled.(3-phase AC output) 1: 2-3 phase transformation is disabled.(2-phase AC output)
		[13] Phase transformation mode setting	<CCVMD> 0: Relative transformation 1: Absolute transformation
Output	***[VExVDUTYA]***	a-phase voltage duty	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	[VExVDUTYB]	b-phase voltage duty	
	[VExVDUTYC]	c-phase voltage duty	

3.3.4. Output Control

Output control unit converts a 3-phase voltage duty into the PMD setting format. The conversion result is set to ***[VExCMPU]***, ***[VExCMPV]***, and ***[VExCMPW]***. According to the output control operation setting, ***[VExOUTCR]*** should be set. Dead time compensation control and PWM output limitation can also be executed.

There are two types of output control task: output control 1 and output control 2. Each task supports different PWM outputs.

Note: When PMD is set to VE mode, the PMD's ***[PMDxCMPU]***/***[PMDxCMPV]***/***[PMDxCMPW]***/***[PMDxMDOUT]*** registers switch to ***[VExCMPU]***/***[VExCMPV]***/***[VExCMPW]***/***[VExOUTCR]***.
Even so, double buffer/triple buffer function of PMD is effective. For details of the buffer function, refer to "Advanced Programmable Motor Control Circuit" in the reference manual.
When double buffers are enabled with PMD, while the output control 1 task/output control 2 task/trigger generation task are running, the execution stage is not updated at the execution stage update timing.

3.3.4.1. Output control 1 (Task 0)

Output control 1 task supports normal PWM outputs and PWM outputs in the PWM Shift 1. When PWM Shift is enabled, if the rotational speed (***[VExOMEGA]***) is lower than the reference of PWM Shift switch (***[VExFPWMCHG]***), PWM outputs changes to PWM Shift 1.

Note: PWM Shift 1 can only be selected in the 1-shunt current detection mode.

a. Output conversion

<Equation>

[VExMCTLF]<LAVFM> = ***[VExMCTLF]***<LAVF> ; Updates a previous value of low-speed flag
[VExMCTLF]<PLSLFM> = ***[VExMCTLF]***<PLSLF> ; Updates a previous value of PWM duty check flag
[VExMCTLF] = & 0xFFEE ; Current flag is cleared
if ((***[VExFMODE]***<IDMODE[1]> = 1) & (***[VExFMODE]***<C2PEN> = 1) & (***[VExFMODE]***<SPWMEN> = 1))
; Shift 1 is enabled in 1-shunt current and 2-phase modulation modes
if (|***[VExOMEGA]***| < ***[VExFPWMCHG]***) ***[VExMCTLF]***<LAVF> = 1
; Sets the low-speed flag for low speed determination
dutya = ***[VExVDUTYA]***
dutyb = ***[VExVDUTYB]***
dutyc = ***[VExVDUTYC]***
if (***[VExMCTLF]***<LAVF> = 1) ; PWM Shift 1 at low-speed
if (***[VExSECTOR]*** = 0, 3, 4, 7, 8, 11) ; Sector determination
dutya = dutya + ***[VExTMPREG5]*** ; Zero-vector V7 transformation
dutyb = dutyb + ***[VExTMPREG5]***
dutyc = dutyc + ***[VExTMPREG5]***
pwma = dutya × 0x8000 ; Transforms the PMD setting value
pwmb = dutyb × 0x8000
pwmc = dutyc × 0x8000

	Register name	Function	Detail
Input	***[VExVDUTYA]***	a-phase voltage duty	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	[VExVDUTYB]	b-phase voltage duty	
	[VExVDUTYC]	c-phase voltage duty	
	[VExSECTOR]	Sector information	4-bit data
	[VExOMEGA]	Rotational speed	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	[VExFPWMCHG]	PWM Shift switch reference speed	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits) Shift switch is disabled when ***[VExFPWMCHG]*** = 0x0000
	[VExFMODE]	[0] Modulation mode	<C2PEN> 0: 3-phase modulation 1: 2-phase modulation
		[1] PWM Shift enable	<SPWMEN> 0: Shift is disabled 1: Shift is enabled
		[3:2] Current detection mode	<IDMODE> 00: 3-shunt 01: 2 sensors 10, 11: 1-shunt
Output	***[VExMCTLF]***	[1:0] Low-speed flag	<LAVFM>, <LAVF>
		[5:4] PWM duty check flag	<PLSLFM>, <PLSLF>

b. PWM output limitation

<Equation>

if (***[VExPWMMAX]*** = 0) max = 0x8000
else max = ***[VExPWMMAX]***
if ((pwma > max) & (0x8000 > max)) ; Checks the PWM upper-limit for U-phase.
  if ((***[VExMODE]***<PWMFLEN> = 0) | (pwma < 0x8000)) ; Confirms the duty of 100% of the output limitation.
    pwma = max
    [VExMCTLF]<PWMOVF> = 1
min = ***[VExPWMMIN]***
if ((pwma < min) & (min > 0)) ; Checks the PWM lower-limit for U-phase.
  if ((***[VExMODE]***<PWMBLEN> = 0) | (pwma > 0)) ; Confirms the duty of 0% of the output limitation.
    pwma = min
    [VExMCTLF]<PWMOVF> = 1

(Calculates the other 2-phase in the same manner)

	Register name	Function	Detail
Input	***[VExPWMMAX]***	Sets the upper-limit value of the PWM.	16-bit data ("0x0000" to "0x8000")
	[VExPWMMIN]	Sets the lower-limit value of the PWM.	
	[VExMODE]	[12] Sets the output duty of 0% when the PWM is limited.	<PWMBLEN> 0: Duty of 0% setting is disabled. 1: Duty of 0% setting is enabled.
		[13] Sets the output duty of 100% when the PWM is limited.	<PWMFLEN> 0: Duty of 100% setting is disabled. 1: Duty of 100% setting is enabled.
Output	***[VExMCTLF]***	[11] Indicates the excess flag of the PWM output limitation.	<PWMOVF>

c. Dead time compensation

<Equation>

```
if (0 < pwma < 0x8000)                          dt = [VExDTC]
else                                            dt = 0
if ([VExDTCS]<IASTS> = 001, 101)                                  ; On Positive current
  if ([VExMODE]<PMDDTCEN> = 1)                                    ; Sets the dead time correction of the PMD
    if (pwma > (0x8000 - 2 × dt))               pwma = (0x8000 + pwma) / 2
  else                                          pwma = pwma + dt
  if (([VExMODE]<PWMFLEN> = 1) & (max < 0x8000))                  ; Confirms the duty of 100%
                                                                    of the output limitation
    if (pwma > (0x8000-1))                      pwma = 0x8000 -1    ; Output limitation after correction.
  else
    if (pwma > 0x8000)                          pwma = 0x8000       ; Output limitation after correction.
else if ([VExDTCS]<IASTS> = 011, 111)                             ; On negative current
  if ([VExMODE]<PMDDTCEN> = 1)                                    ; Sets the dead time correction of the PMD
    if (pwma < (2 × dt))                        pwma = pwma / 2
  else                                          pwma = pwma - dt
  if (([VExMODE]<PWMBLEN> = 1) & (min > 0))                       ; Confirms the duty of 0%
                                                                    of the output limitation.
    if (pwma < 1)                               pwma = 1            ; Output limitation after correction.
  else
    if (pwma < 0)                               pwma = 0            ; Output limitation after correction.
```

(Calculates the other 2-phase in the same manner.)

	Register name	Function	Detail
Input	***[VExDTC]***	The amount of dead time compensation	16-bit data ("0x0000" to "0x8000")
	[VExMODE]	[12] Sets the output duty of 0% when the PWM is limited PWM.	<PWMBLEN> 0: Duty of 0% setting is disabled. 1: Duty of 0% setting is enabled.
		[13] Sets the output duty of 100% when the PWM is limited.	<PWMFLEN> 0: Duty of 100% setting is disabled. 1: Duty of 100% setting is enabled.
		[14] Dead time correction control of the PMD circuit.	<PMDDTCEN> 0: PMD dead time correction is disabled. 1: PMD dead time correction is enabled.
	[VExDTCS]	Dead time compensation control/status	<ICSTS>, <IBSTS>, <IASTS> 000, 010, 100, 110: Undefined 001, 101: Positive current 011, 111: Negative current

d. Output control /PWM Shift 1 transformation

<Equation>

```
outcr = 0x1FF                                   ; All phases center ON
if (IVExMCTLFI<LAVF> = 1)                       ; PWM Shift 1 at low-speed
   if (IVExSECTORI = 0, 1, 2, 11)               ; V-phase center OFF
         outcr = 0x1F3                          ; Reverses a V-phase carrier
   else if (IVExSECTORI = 3, 4, 5, 6)           ; W-phase center OFF
         outcr = 0x1CF                          ; Reverses a W-phase carrier
   else if (IVExSECTORI = 7, 8, 9, 10)          ; U-phase center OFF
         outcr = 0x1FC                          ; Reverses a U-phase carrier
if (IVExMODEI<OCRMD> = 00, 11)                  ; Output OFF
         outcr = 0x000
else if (IVExMODEI<OCRMD> = 10)                 ; Short circuit brake
         outcr = 0x015
IVExCMPUI = pwma
IVExCMPVI = pwmb
IVExCMPWI = pwmc
IVExOUTCRI = outcr

[Pulse width difference check. All OFF-width check]
dif1 = | pwmc - pwma |
dif2 = | pwmb - pwmc |
dif3 = | pwma - pwmb |
difmin = dif1
if (dif2 < difmin)    difmin = dif2
if (dif3 < difmin)    difmin = dif3                          ; Minimum width difference
difmax = (dif1 + dif2 + dif3) / 2                            ; Maximum width difference
if (IVExMCTLFI<LAVF> = 1)                                    ; PWM Shift 1 at low-speed
   if (IVExSECTORI<[0]> = 0)          difmin = difmax        ; Maximum width difference at even sectors
   else                               difmin = difmax-difmin ; Medium width difference at odd sectors
else if (IVExFMODEI<C2PEN> = 1)                              ; Normal PWM at 2-phase modulation
   if (IVExSECTORI = 0, 3, 4, 7, 8, 11)  difmin = difmin × 2 ; Corrected on the sector
                                                             (minimum width = minimum width difference)
      if (difmid < difmin)        difmin = difmid            ; Check the minimum width difference
                                                             after correction

dmax = pwma
if (pwmb > dmax)   dmax = pwmb
if (pwmc > dmax)   dmax = pwmc                               ; Minimum width
offmin = 1-dmax                                              ; All OFF-width

if (IVExMODEI<OCRMD> = 01)                                   ; Output enable
   if (IVExFMODEI<IDMODE[1]> = 1))                           ; 1-shunt current detection
      if (difmin < IVExMINPLSI)   IVExMCTLFI<PLSLF> = 1      ; Check the minimum width difference
   else                                                      ; 3-shunt, 2 sensors
      if (offmin < IVExMINPLSI)   IVExMCTLFI<PLSLF> = 1      ; All OFF-width check
```

	Register name	Function	Detail
Input	***[VExSECTOR]***	Sector information	4-bit data
	[VExMODE]	[3:2] Output control operation	<OCRMD>
	[VExFMODE]	[0] Modulation mode	<C2PEN>
		[3:2] Current detection mode	<IDMODE>
	[VExMINPLS]	Minimum pulse width	16-bit data
Output	***[VExCMPU]***	U-phase PWM duty setting of the PMD	16-bit data ("0x0000" to "0x8000")
	[VExCMPV]	V-phase PWM duty setting of the PMD	
	[VExCMPW]	W-phase PWM duty setting of the PMD	
	[VExOUTCR]	PMD output control setting	9-bit setting
	[VExEMGRS]	PMD EMG release command	1-bit setting
	[VExMCTLF]	[1:0] Low Speed flag	<LAVFM>, <LAVF>
		[5:4] PWM duty check flag	<PLSLFM>, <PLSLF>

3.3.4.2. Output control 2 (Task 9)

Output control 2 task supports two mode outputs: normal PWM outputs and PWM outputs in PWM Shift 2.

When the PWM output in PWM Shift 2 is set, enable PWM Shift (***[VExFMODE]***<SPWMEN> = 1) and select the value other than "00" for PWM Shift mode selection (***[VExFMODE]***<SPWMMD>).

Note: PWM Shift can be selected only in 1-shunt current detection mode.

a. Output conversion

<Equation>

```
mctlfm = [VExMCTLF]                                  ; Read the flag register.
mctlfm = & 0x0011                                    ; Extracts the current value flag
mctlfm = mctlfm << 1                                 ; Shifts to the previous value
if (|[VExOMEGA]| < [VExFPWMCHG])   mctlfm[0] = 1     ; Sets a low-speed flag based on low-speed
                                                       determination
[VExMCTLF] = & 0xFFCC                                ; Current flag is cleared.
[VExMCTLF] = | mctlfm                                ; Update the flag.
pwma = [VExVDUTYA] × 0x8000                          ; The setting value of the PMD is converted.
pwmb = [VExVDUTYB] × 0x8000
pwmc = [VExVDUTYC] × 0x8000
```

	Register name	Function	Detail
Input	***[VExVDUTYA]***	a-phase voltage duty	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	[VExVDUTYB]	b-phase voltage duty	
	[VExVDUTYC]	c-phase voltage duty	
Output	***[VExMCTLF]***	[1:0] Low-speed flag	<LAVFM>, <LAVF>

b. PWM output limitation

<Equation>

```
    if (***[VExPWMMAX]*** = 0)      max = 0x8000
    else                            max = ***[VExPWMMAX]***
if (pwma > max) & (0x8000 > max))                    ; Checks the PWM upper-limit for U-phase.
  if ((***[VExMODE]***<PWMFLEN> = 0) | (pwma < 0x8000))   ; Confirms the duty of 100% of the output
                                                          limitation.
        pwma = max
        ***[VExMCTLF]***<PWMOVF> = 1
    min = ***[VExPWMMIN]***
if ((pwma < min) & (min > 0))                        ; Checks the PWM lower-limit for U-phase.
  if ((***[VExMODE]***<PWMBLEN> = 0) | (pwma > 0))        ; Confirms the duty of 0% of the output
                                                          limitation.
        pwma = min
        ***[VExMCTLF]***<PWMOVF> = 1
```

(Calculates the other 2-phase in the same manner)

	Register name	Function	Detail
Input	***[VExPWMMAX]***	Sets the upper-limit value of the PWM.	16-bit data ("0x0000" to "0x8000")
	[VExPWMMIN]	Sets the lower-limit value of the PWM.	
	[VExMODE]	[12] Sets the output duty of 0% when the PWM is limited.	<PWMBLEN> 0: Duty of 0% setting is disabled. 1: Duty of 0% setting is enabled.
		[13] Sets the output duty of 100% when the PWM is limited.	<PWMFLEN> 0: Duty of 100% setting is disabled. 1: Duty of 100% setting is enabled.
Output	***[VExMCTLF]***	[11] Indicates the excess flag of the PWM output limitation.	<PWMOVF>

c. Dead time compensation

<Equation>

```
      if (0 < pwma < 0x8000)                    dt = [VExDTC]
      else                                      dt = 0
if ([VExDTCS]<IASTS> = 001, 101)                                ; On Positive current
  if ([VExMODE]<PMDDTCEN> = 1)                                  ; Sets the dead time correction of the PMD
        if (pwma > (0x8000- 2 × dt))            pwma = (0x8000 + pwma) / 2
      else                                      pwma = pwma + dt
  if (([VExMODE]<PWMFLEN> = 1) & (max < 0x8000))                ; Confirms the duty of 100%
                                                                  of the output limitation
    if (pwma > (0x8000-1))                      pwma = 0x8000 -1  ; Output limitation after correction.
  else
    if (pwma > 0x8000)                          pwma = 0x8000     ; Output limitation after correction.
else if ([VExDTCS]<IASTS> = 011, 111)                           ; On negative current
  if ([VExMODE]<PMDDTCEN> = 1)                                  ; Sets the dead time correction of the PMD
        if (pwma < (2 × dt))                    pwma = pwma / 2
      else                                      pwma = pwma - dt
  if (([VExMODE]<PWMBLEN> = 1) & (min > 0))                     ; Confirms the duty of 0%
                                                                  of the output limitation.
    if (pwma < 1)                               pwma = 1          ; Output limitation after correction.
  else
    if (pwma < 0)                               pwma = 0          ; Output limitation after correction.
```

(Calculates the other 2-phase in the same manner.)

	Register name	Function	Detail
Input	*[VExDTC]*	The amount of dead time compensation	16-bit data ("0x0000" to "0x8000")
	[VExMODE]	[12] Sets the output duty of 0% when the PWM is limited PWM.	<PWMBLEN> 0: Duty of 0% setting is disabled. 1: Duty of 0% setting is enabled.
		[13] Sets the output duty of 100% when the PWM is limited.	<PWMFLEN> 0: Duty of 100% setting is disabled. 1: Duty of 100% setting is enabled.
		[14] Dead time correction control of the PMD circuit.	<PMDDTCEN> 0: PMD dead time correction is disabled. 1: PMD dead time correction is enabled.
	[VExDTCS]	Dead time compensation control/status	<ICSTS>, <IBSTS>, <IASTS> 000, 010, 100, 110: Undefined 001, 101: Positive current 011, 111: Negative current

d. Output control /PWM Shift 2 transformation
<Equation>

```
outcr = 0x1FF                                                    ; All phases center ON
if (([VExFMODE]<IDMODE[1]> = 1) & ([VExFMODE]<SPWMEN> = 1))   ; PWM Shift 2 is enabled in
                                                                   1-shunt
  if ([VExOMEGA] > [VExFPWMCHG])                                 ; Shift 2 control area determination
    if ([VExFMODE]<SPWMMD> = 01)                                 ; Shift 2, U-phase reference
      if (pwma < 0x4000)                    outcr = 0x1FC      ; Reverses a U-phase carrier
    else if ([VExFMODE]<SPWMMD> = 10)                            ; Shift 2, V-phase reference
      if (pwmb < 0x4000)                    outcr = 0x1F3      ; Reverses a V-phase carrier
    else if ([VExFMODE]<SPWMMD> = 11)                            ; Shift 2, W-phase reference
      if (pwmc < 0x4000)                    outcr = 0x1CF      ; Reverses a W-phase carrier

if ([VExMODE]<OCRMD> = 00, 11)              outcr = 0x000      ; Output OFF
else if ([VExMODE]<OCRMD> = 10)             outcr = 0x015      ; Short circuit brake
[VExCMPU] = pwma
[VExCMPV] = pwmb
[VExCMPW] = pwmc
[VExOUTCR] = outcr

[Pulse width check]
if ([VExFMODE]<IDMODE[1]> = 1 & [VExFMODE]<SPWMEN> = 1)      ; Reference phase for Shift 2
                                                               is double width.
  if ([VExFMODE]<SPWMMD> = 01)              pwma = pwma / 2  ; U-phase reference
  else if ([VExFMODE]<SPWMMD> = 10)         pwmb = pwmb / 2  ; V-phase reference
  else if ([VExFMODE]<SPWMMD> = 11)         pwmc = pwmc / 2  ; W-phase reference

; Detection correction for shift 2 carrier inversion control (Off time/ 2 < [VExMINPLS])
if ([VExOUTCR] = 0x1FC)                     pwma = 0x4000 + pwma
else if ([VExOUTCR] = 0x1F3)                pwmb = 0x4000 + pwmb
else if ([VExOUTCR] = 0x1CF)                pwmc = 0x4000 + pwmc
dmin = pwma
if (pwmb < dmin)            dmin = pwmb
if (pwmc < dmin)            dmin = pwmc                      ; Minimum ON width
dmax = pwma
if (pwmb > dmax)            dmax = pwmb
if (pwmc > dmax)            dmax = pwmc                      ; Maximum ON width
offmin = 1-dmax                                              ; Minimum OFF width (All OFF period)
if (dmin > offmin)          dmin = offmin                    ; Detects the minimum ON/OFF width
minpls = [VExMINPLS]
if ([VExMODE]<OCRMD> = 00, 10, 11)   minpls = 0x0000         ; If output is not enabled
if ([VExFMODE]<IDMODE[1]> = 1)                               ; 1-shunt (PWM Shift 2)
  if (offmin < minpls       [VExMCTLF]<PLSLF> = 1            ; ALL OFF width
else
  if (dmin < minpls)        [VExMCTLF]<PLSLF> = 1            ; Detects the minimum ON/OFF width
```

	Register name	Function	Detail
Input	***[VExFPWMCHG]***	PWM Shift switch reference speed	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits) Shift switch is disabled when ***[VExFPWMCHG]*** = 0x0000
	[VExMODE]	[3:2] Output control operation	<OCRMD> 00: Output is disabled. 01: Output is enabled. 10: Short circuit brake. 11: EMG release.
	[VExMINPLS]	Minimum pulse width	16-bit data
	[VExFMODE]	[1] PWM Shift enable	<SPWMEN> 0: Shift is disabled 1: Shift is enabled
		[3:2] Current detection mode	<IDMODE[1:0]> 00: 3-shunt 01: 2 sensors 10, 11: 1-shunt
		[15:14] PWM Shift mode	<SPWMMD> 00: Shift 1 01: Shift 2 (U-phase Center PWM) 10: Shift 2 (V-phase Center PWM) 11: Shift 2 (W-phase Center PWM)
Output	***[VExCMPU]***	U-phase PWM setting of the PMD	16-bit data ("0x0000" to "0x8000")
	[VExCMPV]	V-phase PWM setting of the PMD	
	[VExCMPW]	W-phase PWM setting of the PMD	
	[VExOUTCR]	PMD output control setting	9-bit setting
	[VExEMGRS]	PMD EMG release command	1-bit setting
	[VExMCTLF]	[5:4] PWM duty check flag	<PLSLFM>, <PLSLF>

3.3.5. Trigger Generation (Task 1)

The trigger generation unit calculates the synchronous trigger timing for ADC from the PWM duty setting values (***[VExCMPU]***, ***[VExCMPV]***, and ***[VExCMPW]***) based on the current detection method, and sets it to the ***[VExTRGCMP0]*** and ***[VExTRGCMP1]*** registers.

Note 1: ***[VExTRGCMP0]*** and ***[VExTRGCMP1]*** are updated only in 1-shunt current detection

Note 2: ***[VExTRGCMP0]*** and ***[VExTRGCMP1]*** are not updated when PWM Shift 2 is selected.

Note 3: Please select "Compare to basic carrier" in the trigger control register comparison carrier selection ***[PMDxTRGCR]***<CARSEL> of the PMD.

Note 4: ***[VExTRGCMP0]*** / ***[VExTRGCMP1]*** / ***[VExTRGSEL]*** switches to ***[PMDxTRGCMP0]*** / ***[PMDxTRGCMP1]*** / ***[PMDxTRGSEL]*** register of PMD when PMD is set to VE mode. Even in that case, double buffer / triple buffer function of PMD is effective. When double buffers / triple buffers are enabled in the PMD, the execution stage is updated even when the update timing of the double buffer / triple buffer execution stage is reached while the output control 1 task/output control 2 task/trigger generation task is being executed not.

Note 5: For details of PMD, refer to "Advanced Programmable Motor Control Circuit" in the reference manual.

Current detection mode <IDMODE>	PWM Shift setting <SPWMEN> <SPWMMD>	Modulation mode <C2PEN>	Low-speed flag <LAVF>	Trigger generation formula
3-shunt	-	2-phase/ 3-phase	-	(No trigger generation)
2 sensors	-	2-phase/ 3-phase	-	(No trigger generation)
1-shunt (Latter half)	Disable PWM Shift	3-phase	-	***[VExTRGCMP0]*** = 0x3FFF + (dmin + dmid) / 4 + ***[VExTRGCRC]*** ***[VExTRGCMP1]*** = 0x3FFF + (dmax + dmid) / 4 + ***[VExTRGCRC]***
		2-phase	-	***[VExTRGCMP0]*** = 0x3FFF + ***[VExTRGCRC]*** ***[VExTRGCMP1]*** = 0x3FFF + (dmax + dmid) / 4 + ***[VExTRGCRC]***
	PWM Shift 1	2-phase	High-speed	***[VExTRGCMP0]*** = 0x3FFF + ***[VExTRGCRC]*** ***[VExTRGCMP1]*** = 0x3FFF + (dmax + dmid) / 4 + ***[VExTRGCRC]***
			Low-speed	***[VExTRGCMP0]*** = 0x3FFF + ***[VExTRGCRC]*** ***[VExTRGCMP1]*** = 0x7FFF - ***[VExTADC]*** + ***[VExTRGCRC]***
	PWM Shift 2	3-phase	-	(No trigger generation)
1-shunt (First half)	Disable PWM Shift	3-phase	-	***[VExTRGCMP0]*** = 0x3FFF - (dmax + dmid) / 4 + ***[VExTRGCRC]*** ***[VExTRGCMP1]*** = 0x3FFF - (dmin + dmid) / 4 + ***[VExTRGCRC]***
		2-phase	-	***[VExTRGCMP0]*** = 0x3FFF - (dmax + dmid) / 4 + ***[VExTRGCRC]*** ***[VExTRGCMP1]*** = 0x3FFF + ***[VExTRGCRC]***
	PWM Shift 1	2-phase	High-speed	***[VExTRGCMP0]*** = 0x3FFF - (dmax + dmid) / 4 + ***[VExTRGCRC]*** ***[VExTRGCMP1]*** = 0x3FFF + ***[VExTRGCRC]***
			Low-speed	***[VExTRGCMP0]*** = 0x0000 + ***[VExTADC]*** + ***[VExTRGCRC]*** ***[VExTRGCMP1]*** = 0x3FFF + ***[VExTRGCRC]***
	PWM Shift 2	3-phase	-	(No trigger generation)

Note: dmin: A value on the minimum duty phase, dmax: A value on the maximum duty phase, dmid: A value on the medium duty phase

	Register name	Function	Detail
Input	***[VExCMPU]***	U-phase PWM duty setting of the PMD	16-bit data ("0x0000" to "0x8000")
	[VExCMPV]	V-phase PWM duty setting of the PMD	
	[VExCMPW]	W-phase PWM duty setting of the PMD	
	[VExTADC]	ADC conversion time	16-bit data ("0x0000" to "0x8000")
	[VExTRGCRC]	Trigger correction value	16-bit data ("0x0000" to "0x8000")
	[VExSECTOR]	Sector information	4-bit data
	[VExMODE]	[0] Zero-current detection	<ZIEN> 0: Normal current detection 1: Zero-current detection
		[3:2] Output control operation	<OCRMD> 00: Output is disabled. 01: Output is enabled. 10: Short circuit brake. 11: EMG release.
	[VExFMODE]	[0] Modulation mode	<C2PEN> 0: 3-phase modulation 1: 2-phase modulation
		[1] PWM Shift enable	<SPWMEN> 0: Shift is disabled 1: Shift is enabled
		[3:2] Current detection mode	<IDMODE[1:0]> 00: 3-shunt 01: 2 sensors 10, 11: 1-shunt
		[8] Trigger correction enable	<CRCEN>
		[15:14] PWM Shift mode	<SPWMMD> 00: Shift 1 01: Shift 2 (U-phase Center PWM) 10: Shift 2 (V-phase Center PWM) 11: Shift 2 (W-phase Center PWM)
	[VExMCTLF]	[0] Low-speed flag	<LAVF>
Output	***[VExTRGCMP0]***	PMD trigger 0 timing setting	16-bit data ("0x0001" to "0x7FFF")
	[VExTRGCMP1]	PMD trigger 1 timing setting	
	[VExTRGSEL]	PMD trigger selection setting	3-bit data: ***[VExSECTOR]*** / 2

3.3.6. Input Process

In the input process, the vector engine reads conversion results and phase information from the ADC. Depending on the current detection method and PWM Shift mode setting, the vector engine converts the phase current data and the voltage data into the fixed-point format and stores them in predefined registers.

In zero-current detection mode, current detection results are stored in the zero-current register.

The vector engine specifies the hysteresis width to determine the current polarity for dead time compensation control.

There are two types of input process: Input process 1 task and Input process 2 task. Each task supports different current detection method.

3.3.6.1. Input process 1 (Task 2)

Input process 1 task supports 3-shunt current detection (only 2-phase current detection (Note 1)) and 1-shunt current detection. However, 1-shunt current detection is not available in PWM Shift 2 (Note 2).

Note 1: Current detection result is used only for two phases. Remained one phase is calculated.

Note 2: PWM Shift can only be selected in 1-shunt current detection mode.

a. Input conversion

<Equation>

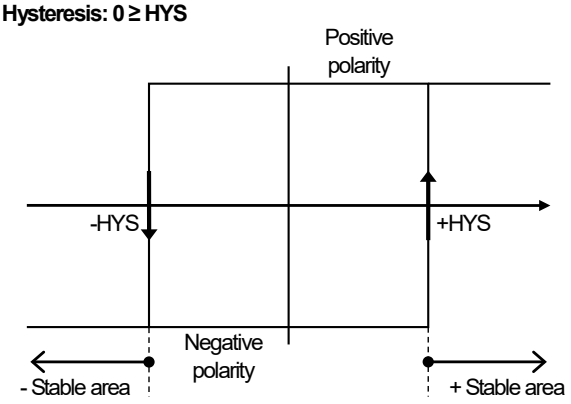
```
[VDC fixed-point transformation/store]
if (/VExMODE/<VDCSEL> = 0)        /VExVDC/ = [DC voltage] >>1
else                              /VExVDCL/ = [DC voltage] >>1
lavfm = /VExMCTLF/<LAVFM>                                  ; Previous low-speed flag
if (/VExFMODE/<MREGDIS> = 1)                               ; Previous value invalid.
    lavfm = /VExMCTLF/<LAVF>                               ; Current low-speed flag

[Current 1 read]
if (/VExFMODE/<IDMODE> = 10, 11)                           ; 1-shunt
  if (lavfm = 0)                                           ; Normal PWM
    if (/VExSECTORM/ = 4, 5, 6, 7)        /VExIAADC/ = [Current 1]
    else if (/VExSECTORM/ = 8, 9, 10, 11) /VExIBADC/ = [Current 1]
    else if (/VExSECTORM/ = 0, 1, 2, 3)   /VExICADC/ = [Current 1]
  else if (lavfm = 1)                                      ; PWM Shift 1
    if (/VExSECTORM/ = 1, 2, 7, 8)        /VExIAADC/ = [Current 1]
    else if (/VExSECTORM/ = 0, 5, 6, 11)  /VExIBADC/ = [Current 1]
    else if (/VExSECTORM/ = 3, 4, 9, 10)  /VExICADC/ = [Current 1]
if (/VExFMODE/<IDMODE> = 00, 01)                           ; 3-shunt, 2 sensors
    if ([Current 1 phase information] = 1)   /VExIAADC/ = [Current 1]
    else if ([Current 1 phase information] = 2) /VExIBADC/ = [Current 1]
    else if ([Current 1 phase information] = 3) /VExICADC/ = [Current 1]

[Current 2 read]
if (/VExFMODE/<IDMODE> = 10, 11)                           ; 1-shunt
  if (lavfm = 0)                                           ; Normal PWM
    if (/VExSECTORM/ = 0, 1, 10, 11)      /VExIAADC/ = [Current 2]
    else if (/VExSECTORM/ = 2, 3, 4, 5)   /VExIBADC/ = [Current 2]
    else if (/VExSECTORM/ = 6, 7, 8, 9)   /VExICADC/ = [Current 2]
```

```
   else if (lavfm = 1)                                  ; PWM Shift 1
      if (***[VExSECTORM]*** = 3, 4, 9, 10)             ***[VExIAADC]*** = [Current 2]
      else if (***[VExSECTORM]*** = 1, 2, 7, 8)         ***[VExIBADC]*** = [Current 2]
      else if (***[VExSECTORM]*** = 0, 5, 6, 11)        ***[VExICADC]*** = [Current 2]
if (***[VExFMODE]***<IDMODE> = 00, 01)                  ; 3-shunt, 2 sensors
      if ([Current 2 phase information] = 1)            ***[VExIAADC]*** = [Current 2]
      else if ([Current 2 phase information] = 2)       ***[VExIBADC]*** = [Current 2]
      else if ([Current 2 phase information] = 3)       ***[VExICADC]*** = [Current 2]

[Current 3 read]
if (***[VExFMODE]***<IDMODE[1]> = 1)                    ; Except 1-shunt
      if ([Current 3 phase information] = 1)            ***[VExIAADC]*** = [Current 3]
      else if ([Current 2 phase information] = 2)       ***[VExIBADC]*** = [Current 3]
      else if ([Current 2 phase information] = 3)       ***[VExICADC]*** = [Current 3]

[Current fixed-point transformation]
ia = ***[VExIAO] - [VExIAADC]***
ib = ***[VExIBO] - [VExIBADC]***
ic = ***[VExICO] - [VExICADC]***
if (***[VExFMODE]***<IDMODE> = 10, 11)                  ; 1-shunt
   if (lavfm = 0)                                       ; Normal PWM
      if (***[VExSECTORM]*** = 0, 1, 10, 11)            ia = -ia
      else if (***[VExSECTORM]*** = 2, 3, 4, 5)         ib = -ib
      else if (***[VExSECTORM]*** = 6, 7, 8, 9)         ic = -ic
   else if (lavfm = 1)                                  ; PWM Shift 1
      if (***[VExSECTORM]*** = 1, 2, 5, 6, 9, 10)       ia = -ia
                                                        ib = -ib
                                                        ic = -ic

[Current 3 calculation]
n = 6 - [Current 2 phase information] -[Current 1 phase information]  ; Calculates the phase number
                                                                        of Current 3
if (n = 1)          ia = -ib -ic
else if (n = 2)     ib = -ic -ia
else if (n = 3)     ic = -ia -ib

[Current storage]
***[VExIA]*** = ia
***[VExIB]*** = ib
***[VExIC]*** = ic
```

	Register name	Function	Detail
Input	(DC voltage)	Input of the ADC conversion result	16-bit data (The conversion result is stored in the upper 12 bits.)
	(Current 1)		
	(Current 2)		
	(Current 3)		
	[VExSECTORM]	Previous ***[VExSECTOR]***	4-bit data
	[VExMODE]	[1] Zero-current detection	<ZIEN> 0: Normal current detection 1: Zero-current detection
		[4] Selection VDC store register	<VDCSEL> 0: ***[VExVDC]*** 1: ***[VExVDCL]***
	[VExFMODE]	[3:2] Current detection mode	<IDMODE> 00: 3-shunt 01: 2 sensors 10, 11: 1-shunt
		[9] Selects to store the previous value(<LAVFM>)	<MREGDIS> 0: Previous value is maintained. 1: Previous value is not maintained.
	[VExMCTLF]	[1:0] Low-speed flag	<LAVFM>, <LAVF>
Output	***[VExVDC]***	DC supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	[VExVDCL]	DC supply voltage	
	[VExIAADC]	The ADC conversion result of a-phase current	16-bit data (The result is stored in the upper 12 bits.)
	[VExIBADC]	The ADC conversion result of b-phase current	
	[VExICADC]	The ADC conversion result of c-phase current	
	[VExIA]	a-phase current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	[VExIB]	b-phase current	
	[VExIC]	c-phase current	
Internal	***[VExIAO]***	a-phase zero-current	16-bit data (The result is stored in the upper 12 bits.)
	[VExIBO]	b-phase zero-current	
	[VExICO]	c-phase zero-current	

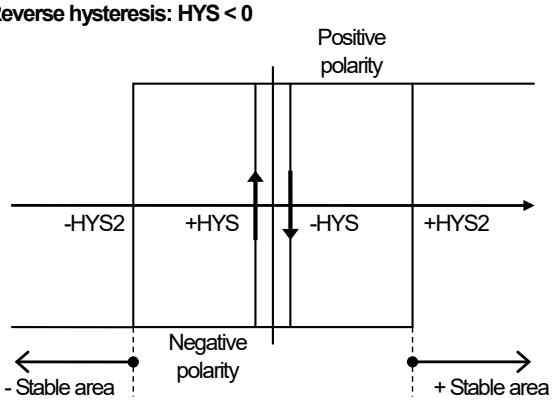
b. Current polarity determination

Hysteresis: 0 ≥ HYS



[|HYS2| ≤ I, I ≤ -|HYS2|, I > HYS, I < -HYS]

Reverse hysteresis: HYS < 0



[|HYS2| ≤ I, I ≤ -|HYS2|, I > HYS, I < -HYS]

	Register name	Function	Detail
Input	***[VExHYS]***	Hysteresis for current polarity determination	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	[VExHYS2]	Hysteresis 2 for current polarity determination	Polarity determination level on reverse hysteresis (***[VExHYS]*** < 0). 16-bit fixed-point data (0 to 1.0, 15 fractional bits) This register is invalid when ***[VExHYS]*** ≥ 0.
	[VExDTCS]	Dead time compensation control/status	<ICSTS>, <IBSTS>, <IASTS> 000, 010, 100, 110: Undefined 001, 101: Positive current 011, 111: Negative current
	[VExMODE]	[15] Current polarity determination control	<IPDEN> 0: Determination is disabled. 1: Determination is enabled.
	[VExIA]	a-phase current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	[VExIB]	b-phase current	
	[VExIC]	c-phase current	
Output	***[VExDTCS]***	Dead time compensation control/status	<ICSTS>, <IBSTS>, <IASTS> 000, 010, 100, 110: Undefined 001, 101: Positive current 011, 111: Negative current

3.3.6.2. Input process 2 (Task 10)

Input process 2 task supports 3-shunt current detection (3-phase detection and 2-phase detection) and 2-sensor current detection. It also supports 1-shunt current detection when PWM outputting in PWM Shift 2.

A current direction can be specified in each phase.

Note: The zero-current detection mode is not available in input process 2 task.

Note: PWM Shift 2 can only be selected in 1-shunt current detection mode.

a. Input conversion

<Equation>

[VDC fixed-point transformation/store]
if (***[VExMODE]***<VDCSEL> = 0) ***[VExVDC]*** = [DC voltage] >> 1
else ***[VExVDCL]*** = [DC voltage] >> 1

[Current 1 read]
if ((***[VExFMODE]***<IDMODE[1]> = 1) & (***[VExMCTLF]***<[1:0]> = 01, 10, 11)) ; 1-shunt at high-speed
  if (***[VExFMODE]***<SPWMMD> = 01) noc = ***[VExOUTCR]***<UOC>
  if (***[VExFMODE]***<SPWMMD> = 10) noc = ***[VExOUTCR]***<VOC>
  if (***[VExFMODE]***<SPWMMD> = 11) noc = ***[VExOUTCR]***<WOC>
if (noc = 11)
  if ([Current 1 phase information] = 1) ***[VExIAADC]*** = [Current 1]
  else if ([Current 1 phase information] = 2) ***[VExIBADC]*** = [Current 1]
  else if ([Current 1 phase information] = 3) ***[VExICADC]*** = [Current 1]
else if (noc = 00)
  if ([Current 2 phase information] = 1) ***[VExIAADC]*** = [Current 1]
  else if ([Current 2 phase information] = 2) ***[VExIBADC]*** = [Current 1]
  else if ([Current 2 phase information] = 3) ***[VExICADC]*** = [Current 1]

[Current 2 read]
if (noc = 11)
  if ([Current 2 phase information] = 1) ***[VExIAADC]*** = [Current 2]
  else if ([Current 2 phase information] = 2) ***[VExIBADC]*** = [Current 2]
  else if ([Current 2 phase information] = 3) ***[VExICADC]*** = [Current 2]
if (noc = 00)
  if ([Current 1 phase information] = 1) ***[VExIAADC]*** = [Current 2]
  else if ([Current 1 phase information] = 2) ***[VExIBADC]*** = [Current 2]
  else if ([Current 1 phase information] = 3) ***[VExICADC]*** = [Current 2]

[Current 3 read]
if (***[VExFMODE]***<IDMODE> = 00 ; Only when 3-phase detection is performed.
  if ([Current 3 phase information] = 1) ***[VExIAADC]*** = [Current 3]
  else if ([Current 3 phase information] = 2) ***[VExIBADC]*** = [Current 3]
  else if ([Current 3 phase information] = 3) ***[VExICADC]*** = [Current 3]

[Current fixed-point transformation]
ia = ***[VExIAO] - [VExIAADC]***
ib = ***[VExIBO] - [VExIBADC]***
ic = ***[VExICO] - [VExICADC]***
if (noc = 11)
  if (***[VExFMODE]***<IAPLMD> = 1) ia = -ia ; Sets direction of Ia current detection
  if (***[VExFMODE]***<IBPLMD> = 1) ib = -ib ; Sets direction of Ib current detection
  if (***[VExFMODE]***<ICPLMD> = 1) ic = -ic ; Sets direction of Ic current detection
if (noc = 00)
  if (***[VExFMODE]***<IAPLMD> = 0) ia = -ia ; Sets direction of Ia current detection
  if (***[VExFMODE]***<IBPLMD> = 0) ib = -ib ; Sets direction of Ib current detection
  if (***[VExFMODE]***<ICPLMD> = 0) ic = -ic ; Sets direction of Ic current detection
n = 6 - [Current 2 phase information] - [Current 1 phase information]
if (***[VExFMODE]***<IDMODE> = 00) ; Current 3 is calculated except when 3-phase detection is performed.

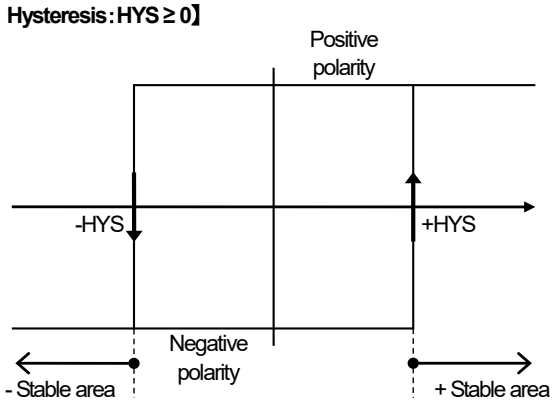
  if (n = 1) ia = -ib -ic
  else if (n = 2) ib = -ic -ia
  else if (n = 3) ic = -ia -ib

[Current storage]
[VExIA] = ia
[VExIB] = ib
[VExIC] = ic

	Register name	Function	Detail
Input	(DC voltage)	Input of the ADC conversion result	16-bit data (The conversion result is stored in the upper 12 bits.)
	(Current 1)		
	(Current 2)		
	(Current 3)		
	[VExSECTORM]	Previous ***[VExSECTOR]***	4-bit data
	[VExMODE]	[4] Selection VDC store register	<VDCSEL> 0: ***[VExVDC]*** 1: ***[VExVDCL]***
	[VExFMODE]	[3:2] Current detection mode	<IDMODE> 00: 3-shunt 01: 2 sensors 10, 11: 1-shunt
		[7:5] Select the current detection	<ICPLMD>, <IBPLMD>, <IAPLMD> 0: Shunt mode (In = VExInO - VExInADC) 1: Sensor mode(In = VExInADC - VExInO) Note: n = A, B, or C
		[15:14] PWM shift mode	<SPWMMD>
	[VExMCTLF]	[1:0] Low-speed flag	<LAVFM>, <LAVF>
	[VExOUTCR]	[1:0] U-phase output control	<UOC>
		[3:2] V-phase output control	<VOC>
		[5:4] W-phase output control	<WOC>
Output	***[VExVDC]***	DC supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	[VExVDCL]	DC supply voltage	
	[VExIAADC]	The ADC conversion result of a-phase current	16-bit data (The result is stored in the upper 12 bits.)
	[VExIBADC]	The ADC conversion result of b-phase current	
	[VExICADC]	The ADC conversion result of c-phase current	
	[VExIA]	a-phase current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	[VExIB]	b-phase current	
	[VExIC]	c-phase current	
Internal	***[VExIAO]***	a-phase zero-current	16-bit data (The result is stored in the upper 12 bits.)
	[VExIBO]	b-phase zero-current	
	[VExICO]	c-phase zero-current	

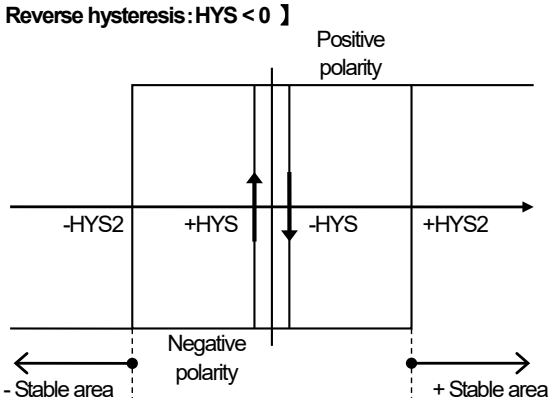
b. Current polarity determination

【 Hysteresis : HYS ≥ 0】



[|HYS2| ≤ I, I ≤ -|HYS2|, I > HYS, I < -HYS]

【 Reverse hysteresis : HYS < 0 】



[|HYS2| ≤ I, I ≤ -|HYS2|, I > HYS, I < -HYS]

	Register name	Function	Detail
Input	***[VExHYS]***	Hysteresis for current polarity determination	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	[VExHYS2]	Hysteresis 2 for current polarity determination	Polarity determination level on reverse hysteresis (***[VExHYS]*** < 0). 16-bit fixed-point data (0 to 1.0, 15 fractional bits) This register is invalid when ***[VExHYS]*** ≥ 0.
	[VExDTCS]	Dead time compensation control/status	<ICSTS>, <IBSTS>, <IASTS> 000, 010, 100, 110: Undefined 001, 101: Positive current 011, 111: Negative current
	[VExMODE]	[15] Current polarity determination control	<IPDEN> 0: Determination is disabled. 1: Determination is enabled.
	[VExIA]	a-phase current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	[VExIB]	b-phase current	
	[VExIC]	c-phase current	
Output	***[VExDTCS]***	Dead time compensation control/status	<ICSTS>, <IBSTS>, <IASTS> 000, 010, 100, 110: Undefined 001, 101: Positive current 011, 111: Negative current

3.3.7. Input Current Transformation (Phase Transformation/Coordinate Axis Transformation)

Input current transformation consists of two tasks: phase transformation and coordinate axis transformation.

3.3.7.1. Input phase transformation (Task 3)

Input phase transformation task calculates Iα and Iβ based on Ia, Ib, and Ic.

<Equation>

kc = 1 ; Relative transformation
if (***[VExFMODE]***<CCVMD> = 1) kc = √(3 / 2) ; Absolute transformation
if (***[VExFMODE]***<PHCVDIS> = 0) ; Phase transformation is enabled
[VExIALPHA] = kc × ***[VExIA]*** ; Calculates the value of Iα
[VExIBETA] = kc × √(1 / 3) × ***[VExIB]*** - kc × √(1 / 3) × ***[VExIC]*** ; Calculates the value of Iβ
else if (***[VExFMODE]***<PHCVDIS> = 1) ; Phase transformation is disabled
[VExIALPHA] = ***[VExIA]***
[VExIBETA] = ***[VExIB]***

	Register name	Function	Detail
Input	***[VExIA]***	a-phase current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	[VExIB]	b-phase current	
	[VExIC]	c-phase current	
	[VExFMODE]	[12] Disables phase transformation	<PHCVDIS> 0: 2-3 phase transformation is enabled. (3-phase AC output) 1: 2-3 phase transformation is disabled. (2-phase AC output)
		[13] Phase transformation mode setting	<CCVMD> 0: Relative transformation 1: Absolute transformation
Output	***[VExIALPHA]***	α-axis current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	[VExIBETA]	β-axis current	

3.3.7.2. Input coordinate axis transformation (Task 4)

In input coordinate axis transformation task, the values of ***[VExID]*** and ***[VExIQ]*** are calculated from ***[VExIALFA]***, ***[VExIBETA]***, ***[VExSINM]***, and ***[VExCOSM]***.

a. Coordinate axis transformation

<Equation>

plslfm = ***[VExMCTLF]***<PLSLFM>
if (***[VExFMODE]***<MREGDIS> = 1) ; Previous value is invalid.
    plslfm = ***[VExMCTLF]***<PLSLF>
if (plslfm = 0) ; Check the small pulse flag.
  [VExID] = ***[VExCOSM]*** × ***[VExIALPHA]*** + ***[VExSINM]*** × ***[VExIBETA]*** ; Calculates the value of Id.
  [VExIQ] = -***[VExSINM]*** × ***[VExIALPHA]*** + ***[VExCOSM]*** × ***[VExIBETA]*** ; Calculates the value of Iq.

	Register name	Function	Detail
Input	***[VExIALPHA]***	α-axis current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	[VExIBETA]	β-axis current	
	[VExSINM]	Previous sine value	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	[VExCOSM]	Previous cosine value	
	[VExSIN]	The sine value of θ	
	[VExCOS]	The cosine value of θ	
	[VExFMODE]	[9] Control for maintain previous value	<MREGDIS> 0: Previous value is maintained. 1: Previous value is not maintained.
	[VExMCTLF]	[4] PWM duty check flag	<PLSLF>
		[5] Previous <PLSLF>	<PLSLFM>
Output	***[VExID]***	d-axis current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	[VExIQ]	q-axis current	

b. ATAN calculation

When <ATANMD> = 10, ATAN task calculates the declination angle of d-axis and q-axis current.

When <ATANMD> = 11, ATAN task calculates the induced voltage of d-axis and q-axis. Then it calculates the declination angle using motor voltage equation based on these results.

<Equation>

if (***[VExMODE]***<ATANMD[1]> = 1) ; Enables calculation of the declination angle
  ld = ***[VExCLD]*** × (***[VExCLG]*** setting) ; d-axis inductance
  lq = ***[VExCLQ]*** × (***[VExCLG]*** setting) ; q-axis inductance
  r = ***[VExCR]*** × (***[VExCRG]*** setting) ; Resistance
  vdiv = ***[VExVD]*** - (***[VExID]*** × r - ***[VExOMEGA]*** × ***[VExIQ]*** × lq) ; Induced voltage for d-axis
  vqiv = ***[VExVQ]*** - (***[VExIQ]*** × r + ***[VExOMEGA]*** × ***[VExID]*** × ld); Induced voltage for q-axis
  [VExTMPREG3] = vdiv ; Stores d-axis induced voltage
  [VExTMPREG4] = vqiv ; Stores q-axis induced voltage
  if (***[VExMODE]***<ATANMD[0]> = 1)
      [VExDELTA] = ATAN(|vqiv|, |vdiv|) ; Calculates the declination angle of the induced voltage.

else ***[VExDELTA]*** = ATAN(|***[VExIQ]***|, |***[VExID]***|) ; Calculates the declination angle of the current

	Register name	Function	Detail
Input	***[VExCLD]***	Motor d-axis inductance	16-bit fixed-point data (11 fractional bits)
	[VExCLQ]	Motor q-axis inductance	
	[VExCR]	Motor resistance value	
	[VExCLG]	Inductance range setting	000: 1 / 1 001: 1 / 2^4 010: 1 / 2^8 011: 1 / 2^{12} 100: 1 / 2^{16} 101 to 111: Reserved
	[VExCRG]	VExCR range setting	
	[VExOMEGA]	Rotational speed	16-bit fixed-point data (15 fractional bits)
	[VExVD]	d-axis voltage	32-bit fixed-point data (-1.0 to 1.0 and 31 fractional bits)
	[VExVQ]	q-axis voltage	
	[VExMODE]	[6:5] ATAN operation mode setting	<ATANMD> 00, 01: Calculation is prohibited. 10: Calculates the declination angle for Id and Iq. 11: Calculates the declination angle of induced voltage of d-axis and q-axis.
Output	***[VExDELTA]***	Output of the current declination angle	16-bit data (-180-degree to 180-degree) 0x8000 to 0x7FFF
	[VExTMPREG3]	d-axis induced voltage	16-bit fixed-point data (15 fractional bits)
	[VExTMPREG4]	q-axis induced voltage	

3.3.8. Other Tasks

3.3.8.1. ATAN2 (Arctangent function 2) (Task 12)

ATAN2 task calculates the angle from the starting point based on X-axis that draws a straight line from the starting point to the point of (X, Y) on the X-Y coordinate.

<Equation>

x = ***[VExTMPREG4]***
y = ***[VExTMPREG5]***
z = ATAN (|y|, |x|) ; Arctangent calculation (0-degree to 90-degree)
if (x < 0 & y ≥ 0) z = 0x00008000 - z ; The second quadrant (90-degree to 180-degree)
if (x < 0 & y < 0) z = 0xFFFF8000 + z ; The third quadrant (-90-degree to -180-degree)
if (x ≥ 0 & y < 0) z = -z ; The forth quadrant (0-degree to -90-degree)
if (x = y = 0) z = 0x00000000 ; Output on the original point (0-degree)
[VExTMPREG5] = z

	Register name	Function	Detail
Input	***[VExTMPREG4]***	Input X	32-bit signed data
	[VExTMPREG5]	Input Y	
Output	***[VExTMPREG5]***	The value of the phase	32-bit data (-180-degree to 180-degree) 0xFFFF8000 to 0x00008000

3.3.8.2. SQRT (Square root function) (Task 13)

SQRT task outputs the value from 0.0 to 2.0 based on the calculation of the square root function using the inputs from 0.0 to 4.0.

<Equation>

x = ***[VExTMPREG5]*** ; Input (0 to 4.0)
n = 0
if (x < 0x2000) n = 1
if (x < 0x0800) n = 2
if (x < 0x0200) n = 3
if (x < 0x0080) n = 4
if (x < 0x0020) n = 5
if (x < 0x0008) n = 6
if (x < 0x0002) n = 7
if (x < 0x8000) n = -1
$x = x \times 2^{2n}$; Normalization (0.25 to 1.0)
if (x ≥ 0x7FFF) x = 0x7FFF
z = SQRT(x) ; Calculates the square root of x,
; Output is the range of 0.5 to 1.0
if (x = 0) z = 0
$z = z / 2^{n}$; Inverted transformation (0 to 2.0)
[VExTMPREG5] = z

	Register name	Function	Detail
Input	***[VExTMPREG5]***	Input value	32-bit fixed-point data (0.0 to 4.0, 15 fractional bits) 0x00000000 to 0x0001FFFF
Output	***[VExTMPREG5]***	The value of square root	32-bit fixed-point data (0.0 to 2.0, 15 fractional bits) 0x00000000 to 0x0000FFFF

4. Registers

The vector engine has the VE control registers and the dedicated registers.

- VE control registers
  The vector engine control registers and temporary registers.

- Dedicated registers
  Computation data registers and computation control registers.

Note: Use word access (32 bits) to the registers.

4.1. List of Registers

The control registers and their addresses are shown as follows:

Peripheral Function		Channel/Unit	Base address		
			TYPE1	TYPE2	TYPE3
Advanced Vector Engine plus	A-VE+	ch0	0x400F8000	0x400EB000	0x4008B000

Note: The Channel/Unit and Base address type are different by products. Please refer to "Product Information" of the reference manual for the details.

VE control register

Register Name		Address (Base+)
VE Operation Enable/Disable Register	***[VExEN]***	0x0000
CPU Start Trigger Selection Register	***[VExCPURUNTRG]***	0x0004
Task Selection Register	***[VExTASKAPP]***	0x0008
Operation Schedule Selection Register	***[VExACTSCH]***	0x000C
Operation Schedule Repeat Number Selection Register	***[VExREPTIME]***	0x0010
Start Trigger Mode Setting Register	***[VExTRGMODE]***	0x0014
Error Interrupt Enable/Disable Setting Register	***[VExERRINTEN]***	0x0018
VE Forcible Termination Register	***[VExCOMPEND]***	0x001C
Error Detection Register	***[VExERRDET]***	0x0020
Schedule Operation Status/Ongoing Task Number Register	***[VExSCHTASKRUN]***	0x0024
Temporary Register 0	***[VExTMPREG0]***	0x002C
Temporary Register 1	***[VExTMPREG1]***	0x0030
Temporary Register 2	***[VExTMPREG2]***	0x0034
Temporary Register 3	***[VExTMPREG3]***	0x0038
Temporary Register 4	***[VExTMPREG4]***	0x003C
Temporary Register 5	***[VExTMPREG5]***	0x0040

Dedicated registers

Register Name		Address (Base+)
Status Register	***[VExMCTLF]***	0x0044
Task Control Mode Register	***[VExMODE]***	0x0048
Flow Control Register	***[VExFMODE]***	0x004C
PWM Period Rate setting Register	***[VExTPWM]***	0x0050
Rotational Speed setting Register	***[VExOMEGA]***	0x0054
Motor Phase setting Register	***[VExTHETA]***	0x0058
d-axis Current Reference setting Register	***[VExIDREF]***	0x005C
q-axis Current Reference setting Register	***[VExIQREF]***	0x0060
d-axis Voltage setting Register	***[VExVD]***	0x0064
q-axis Voltage setting Register	***[VExVQ]***	0x0068
d-axis Current Integral Coefficient setting Register for PI control	***[VExCIDKI]***	0x006C
d-axis Current Proportional Coefficient setting Register for PI control	***[VExCIDKP]***	0x0070
q-axis Current Integral Coefficient setting Register for PI control	***[VExCIQKI]***	0x0074
q-axis Current Proportional Coefficient setting Register for PI control	***[VExCIQKP]***	0x0078
d-axis Voltage Integral component Store Register for PI control (Upper)	***[VExVDIH]***	0x007C
d-axis Voltage Integral component Store Register for PI control (Lower)	***[VExVDILH]***	0x0080
q-axis Voltage Integral component Store Register for PI control (Upper)	***[VExVQIH]***	0x0084
q-axis Voltage Integral component Store Register for PI control (Lower)	***[VExVQILH]***	0x0088
PWM Switching Speed setting Register	***[VExFPWMCHG]***	0x008C
PWM Shift 2 Offset Register	***[VExPWMOFS]***	0x0090
Minimum Pulse Width setting Register	***[VExMINPLS]***	0x0094
Synchronous Trigger Correction setting Register	***[VExTRGCRC]***	0x0098
DC2 Supply Voltage Register	***[VExVDCL]***	0x009C
Cosine value Register of θ	***[VExCOS]***	0x00A0
Sine value Register of θ	***[VExSIN]***	0x00A4
Previous Cosine value Register	***[VExCOSM]***	0x00A8
Previous Sine value Register	***[VExSINM]***	0x00AC
Sector Information Register	***[VExSECTOR]***	0x00B0
Previous Sector Information Register	***[VExSECTORM]***	0x00B4
a-phase Zero-Current Register	***[VExIAO]***	0x00B8
b-phase Zero-Current Register	***[VExIBO]***	0x00BC
c-phase Zero-Current Register	***[VExICO]***	0x00C0
ADC Conversion Result Register for a-phase Current	***[VExIAADC]***	0x00C4
ADC Conversion Result Register for b-phase Current	***[VExIBADC]***	0x00C8
ADC Conversion Result Register for c-phase Current	***[VExICADC]***	0x00CC
DC Supply Voltage Register	***[VExVDC]***	0x00D0
d-axis Current Register	***[VExID]***	0x00D4
q-axis Current Register	***[VExIQ]***	0x00D8
ADC Conversion Time setting Register	***[VExTADC]***	0x0178
U-phase PWM Duty Register	***[VExCMPU]***	0x017C
V-phase PWM Duty Register	***[VExCMPV]***	0x0180

Register Name		Address (Base+)
W-phase PWM Duty Register	***[VExCMPW]***	0x0184
PMD Output Control Register	***[VExOUTCR]***	0x0188
PMD Trigger 0 Timing setting Register	***[VExTRGCMP0]***	0x018C
PMD Trigger 1 Timing setting Register	***[VExTRGCMP1]***	0x0190
Synchronous Trigger Selection setting Register	***[VExTRGSEL]***	0x0194
EMG release Register	***[VExEMGRS]***	0x0198
Output Limitation Register for PI control	***[VExPIOLIM]***	0x01BC
d-axis Coefficient Range setting Register for PI control	***[VExCIDKG]***	0x01C0
q-axis Coefficient Range setting Register for PI control	***[VExCIQKG]***	0x01C4
Voltage Scalar Limitation Register	***[VExVSLIM]***	0x01C8
Voltage Scalar Register	***[VExVDQ]***	0x01CC
Current Declination Angle Register	***[VExDELTA]***	0x01D0
Motor Interlinkage Magnetic Flux Register	***[VExCPHI]***	0x01D4
Motor q-axis Inductance Constant Register	***[VExCLD]***	0x01D8
Motor d-axis Inductance Constant Register	***[VExCLQ]***	0x01DC
Motor Resistance Constant Register	***[VExCR]***	0x01E0
Motor Magnetic Flux Range setting Register	***[VExCPHIG]***	0x01E4
Motor Inductance Range setting Register	***[VExCLG]***	0x01E8
Motor Resistance Range setting Register	***[VExCRG]***	0x01EC
d-axis Voltage Register for Non-interference Control	***[VExVDE]***	0x01F0
q-axis Voltage Register for Non-interference Control	***[VExVQE]***	0x01F4
Dead Time Compensation Register	***[VExDTC]***	0x01F8
Hysteresis Width Register for Current Polarity Determination	***[VExHYS]***	0x01FC
Dead Time Compensation Control/Status Register	***[VExDTCS]***	0x0200
PWM Upper-Limit setting Register	***[VExPWMMAX]***	0x0204
PWM Lower-Limit setting Register	***[VExPWMMIN]***	0x0208
Clipped Phase value setting Register	***[VExTHTCLP]***	0x020C
Hysteresis Width Register 2 for Current Polarity Determination	***[VExHYS2]***	0x0210
α-phase Voltage Register	***[VExVALPHA]***	0x0214
β-phase Voltage Register	***[VExVBETA]***	0x0218
a-phase Voltage Duty Register	***[VExVDUTYA]***	0x021C
b-phase Voltage Duty Register	***[VExVDUTYB]***	0x0220
c-phase Voltage Duty Register	***[VExVDUTYC]***	0x0224
α-phase Current Register	***[VExIALPHA]***	0x0228
β-phase Current Register	***[VExIBETA]***	0x022C
a-phase Current Register	***[VExIA]***	0x0230
b-phase Current Register	***[VExIB]***	0x0234
c-phase Current Register	***[VExIC]***	0x0238
Voltage declination angle Register	***[VExVDELTA]***	0x023C
d-axis Voltage correction Register	***[VExVDCRC]***	0x0240
q-axis Voltage correction Register	***[VExVQCRC]***	0x0244

4.2. Details of VE Control Registers

4.2.1. *[VExEN]* (VE Operation Enable/Disable Register)

Bit	Bit Symbol	After Reset	Type	Function
31:2	-	0	R	Read as "0"
1	-	0	R/W	Write as "0"
0	VEEN	0	R/W	Selects the VE operation. 0: Disabled 1: Enabled

Note1: When the VE is disabled (<VEEN> = 0), access to other registers of the VE is not allowed

Note2: To change the PMD registers from VE, be sure to set the operation schedule repetition count specification register ***[VExREPTIME]***<VREP> to other than "0x0" after enabling the VE operation (<VEEN> = 1).

4.2.2. *[VExCPURUNTRG]* (CPU Start Trigger Selection Register)

Bit	Bit Symbol	After Reset	Type	Function
31:1	-	0	R	Read as "0"
0	VCPURT	0	W	Starts the vector engine by software. 0: - 1: Starts operation Operation starts from the task configured to ***[VExTASKAPP]*** <VTASK>. Specify ***[VExTASKAPP]***, ***[VExACTSCH]***, and ***[VExREPTIME]*** before starting operation.

Note 1: When "1" is written to this bit, it is cleared on the next cycle. This bit always read as "0".

Note 2: If new schedules and the tasks start operation while another schedule is being executed, the vector engine should be terminated with the ***[VExCOMPEND]*** register before it is started with the ***[VExCPURUNTRG]*** register again.

4.2.3. *[VExTASKAPP]* (Task Selection Register)

Bit	Bit Symbol	After Reset	Type	Function
31:12	-	0	R	Read as "0"
11:8	VITASK[3:0]	0x0	R/W	Selects the timing at which the task end interrupt occurs. 0x0: Output control 1 (Task 0) 0x1: Trigger generation (Task 1) 0x2: Input process 1 (Task 2) 0x3: Input phase transformation (Task 3) 0x4: Input coordinate axis transformation (Task 4) 0x5: Current control (Task 5) 0x6: SIN/COS calculation (Task 6) 0x7: Output coordinate axis transformation (Task 7) 0x8: Output phase transformation 1[SVM] (Task 8) 0x9: Output control 2 (Task 9) 0xA: Input process 2 (Task 10) 0xB: Output phase transformation 2[I-Clarke] (Task 11) 0xC: ATAN2 (Task 12) 0xD: SQRT (Task 13) 0xE, 0xF: Reserved
7:4	-	0	R	Read as "0"
3:0	VTASK[3:0]	0x0	R/W	Selects the start task. 0x0: Output control 1 (Task 0) 0x1: Trigger generation (Task 1) 0x2: Input process 1 (Task 2) 0x3: Input phase transformation (Task 3) 0x4: Input coordinate axis transformation (Task 4) 0x5: Current control (Task 5) 0x6: SIN/COS calculation (Task 6) 0x7: Output coordinate axis transformation (Task 7) 0x8: Output phase transformation 1[SVM] (Task 8) 0x9: Output control 2 (Task 9) 0xA: Input process 2 (Task 10) 0xB: Output phase transformation 2[I-Clarke] (Task 11) 0xC: ATAN2 (Task 12) 0xD: SQRT (Task 13) 0xE, 0xF: Reserved Selects the task that is started by software.

Note: Only those tasks that are included in the schedules can be specified.

4.2.4. *[VExACTSCH]* (Operation Schedule Selection Register)

Bit	Bit Symbol	After Reset	Type	Function
31:4	-	0	R	Read as "0"
3:0	VACT[3:0]	0x0	R/W	Selects individual task execution or an operation schedule. 0x0: Executes individual task. 0x1: Schedule 1 0x2: Schedule 2 0x3: Schedule 3 0x4: Schedule 4 0x5: Schedule 5 0x6: Schedule 6 0x7: Schedule 7 0x8: Schedule 8 0x9: Schedule 9 0xA: Schedule 10 0xB: Schedule 11 0xC: Schedule 12 0xD: Schedule 13 0xE: Schedule 14 0xF: Schedule 15 For details, refer to "Table 3.1 Execution task in each schedule".

4.2.5. *[VExREPTIME]* (Operation Schedule Repeat Number Selection Register)

Bit	Bit Symbol	After Reset	Type	Function
31:4	-	0	R	Read as "0"
3:0	VREP[3:0]	0x0	R/W	Selects the number of repeat of the operation schedule. 0x0: No schedule operation is performed. 0x1 to 0xF: Repeats the specified number of times of the schedule operation

Note: When this bit is set to "0x0", do not start schedule operation.

4.2.6. *[VExTRGMODE]* (Start Trigger Mode Setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:2	-	0	R	Read as "0"
1:0	VTRG[1:0]	00	R/W	Selects the input process start condition using an ADC conversion end interrupt. 00: Reserved 01: Input process is started by a INTADxPDA (ADC conversion end interrupt A).(Note) 10: Input process is started by a INTADxPDB (ADC conversion end interrupt B).(Note) 11: Reserved

Note: Connection varies depending on the product. Refer to "Product Information".

4.2.7. *[VExERRINTEN]* (Error Interrupt Enable/Disable Setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:3	-	0	R	Read as "0"
2	INTTEN	0	R/W	Controls a task end interrupt. 0: Disabled 1: Enabled
1	-	0	R	Read as "0"
0	VERREN	0	R/W	Controls an interrupt at error detection. 0: Disabled 1: Enabled If a PWM interrupt is detected when an operation schedule is being executed (an activation trigger wait is not included), "1" is set as an error flag.

4.2.8. *[VExCOMPEND]* (VE Forcible Termination Register)

Bit	Bit Symbol	After Reset	Type	Function
31:1	-	0	R	Read as "0"
0	VCEND	0	W	Ongoing schedule is forcibly terminated. 0: - 1: Stop If "1" is written to this bit, this bit is cleared on next cycle. Always read as "0"

4.2.9. *[VExERRDET]* (Error Detection Register)

Bit	Bit Symbol	After Reset	Type	Function
31:1	-	0	R	Read as "0"
0	VERRD	0	R	This bit indicates the error flag. 0: An error is not detected. 1: An error is detected. When an operation schedule is being executed (excluding a start trigger wait), if a PWM interrupt is detected, "1" is set to this bit. This bit is cleared upon read.

4.2.10. *[VExSCHTASKRUN]* (Schedule Operation Status/Ongoing Task Number Register)

Bit	Bit Symbol	After Reset	Type	Function
31:5	-	0	R	Read as "0"
4:1	VRTASK[3:0]	0x0	R	Indicates the ongoing task number. 0x0: Output control 1 (Task 0) 0x1: Trigger generation (Task 1) 0x2: Input process 1 (Task 2) 0x3: Input phase transformation (Task 3) 0x4: Input coordinate axis transformation (Task 4) 0x5: Current control (Task 5) 0x6: SIN/COS calculation (Task 6) 0x7: Output coordinate axis transformation (Task 7) 0x8: Output phase transformation 1[SVM] (Task 8) 0x9: Output control 2 (Task 9) 0xA: Input process 2 (Task 10) 0xB: Output phase transformation 2[I-Clarke] (Task 11) 0xC: ATAN2 (Task 12) 0xD: SQRT (Task 13) 0xE, 0xF: Reserved
0	VRSCH	0	R	Indicates schedule execution status. 0: Stop 1: Being executed

4.2.11. *[VExTMPREG0]* (Temporary Registers 0)

Example of ***[VExTMPREG0]***. ***[VExTMPREG1]*** to ***[VExTMPREG5]*** have the same configuration.

Bit	Bit Symbol	After Reset	Type	Function
31:0	TMPREG0 [31:0]	0x00000000	R/W	Temporary register 0

4.3. Details of Dedicated Registers

4.3.1. *[VExMCTLF]* (Status Register)

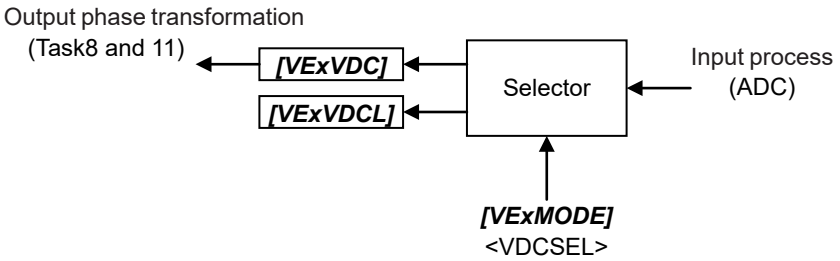
Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15	SFT2STM	0	R/W	Previous value of <SFT2ST> The value is updated when output control 2 (Task 9) is executed.
14	SFT2ST	0	R/W	A PWM carrier reverse flag of PWM Shift 2 The value is updated when output control 2 (Task 9) is executed. 0: Without PWM carrier reverse of reference phase 1: With PWM carrier reverse of reference phase Set "1" to this bit when Shift 2 reference phase duty < 0x4000
13:12	-	00	R/W	Write as "00"
11	PWMOVF	0	R/W	Indicates the excess flag for PWM output limitation. 0: All 3-phase PWM outputs are ***[VExPWMMIN]*** or more, and ***[VExPWMMAX]*** or less. 1: Any 3-phase PWM outputs are less than ***[VExPWMMIN]***, or over ***[VExPWMMAX]***. This bit is updated when output controls (Task 0 and Task 9) are executed.
10	VSOVF	0	R/W	Indicates the excess flag for voltage scalar limitation. 0: Voltage scalar ≤ ***[VExVSLIM]*** 1: Voltage scalar > ***[VExVSLIM]*** This bit is updated when current control (Task 5) is executed.
9	PIQOVF	0	R/W	Indicates the excess flag for q-axis output limitation on PI control. 0: |q-axis output on PI control| ≤ ***[VExPIOLIM]*** 1: |q-axis output on PI control| > ***[VExPIOLIM]*** This bit is updated when current control (Task 5) is executed.
8	PIDOVF	0	R/W	Indicates the excess flag for d-axis output limitation on PI control. 0: |d-axis output on PI control| ≤ ***[VExPIOLIM]*** 1: |d-axis output on PI control| > ***[VExPIOLIM]*** This bit is updated when current control (Task 5) is executed.
7:6	-	00	R/W	Write as "00"
5	PLSLFM	0	R/W	Indicates the previous value of the <PLSLF> register. This bit is updated when output controls (Task 0 and Task 9) are executed.
4	PLSLF	0	R/W	PWM duty check flag. This bit is updated when output controls (Task 0 and Task 9) are executed. When output control 1(Task 0) is executed and 1-shunt current detection is set, 0: ***[VExMINPLS]*** ≤ Difference of minimum pulse width 1: ***[VExMINPLS]*** > Difference of minimum pulse width When output control 2 (Task 9) is executed, 0: ***[VExMINPLS]*** ≤ Minimum On-width or minimum Off-width 1: ***[VExMINPLS]*** > Minimum On-width or minimum Off-width
3	-	0	R/W	Write as "0"
2	LVTF	0	R/W	Low supply voltage flag 0: ***[VExVDC]*** ≥ 1 / 128 (0x0100) 1: ***[VExVDC]*** < 1 / 128 (0x0100) This bit is updated when output phase transformations (Task 8 and 11) are executed.

Bit	Bit Symbol	After Reset	Type	Function
1	LAVFM	0	R/W	Previous value of the <LAVF> register This bit is updated when output controls (Task 0 and Task 9) are executed.
0	LAVF	0	R/W	Low-speed flag This bit is updated when output controls (Task 0 and Task 9) are executed. 0: High speed (***[VExOMEGA]*** ≥ ***[VExFPWMCHG]***) 1: Low speed (***[VExOMEGA]*** < ***[VExFPWMCHG]***)

4.3.2. *[VExMODE]* (Task Control Mode Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15	IPDEN	0	R/W	Disables/enables current polarity determination for Task 2 and Task 10. 0: Disabled 1: Enabled Refer to "3.3.6 Input Process".
14	PMDDTCEN	0	R/W	Setting that corresponds to the dead time correction control of the PMD when the dead time compensation for Task 0 and Task 9 is performed. 0: Dead time correction of the PMD is disabled. 1: Dead time correction of the PMD is enabled. Refer to "3.3.4 Output Control".
13	PWMFLEN	0	R/W	Disables/enables a duty of 100% setting when the upper-limit for Task 0 and Task 9 is set. 0: Duty of 100% setting is disabled. 1: Duty of 100% setting is enabled. Refer to "3.3.4 Output Control".
12	PWMBLEN	0	R/W	Disables/enables a duty of 0% setting when the lower-limit for Task 0 and Task 9 is set. 0: Duty of 0% setting is disabled. 1: Duty of 0% setting is enabled. Refer to "3.3.4 Output Control".
11	NICEN	0	R/W	Disables/enables Non-interference control for Task 5. 0: Disabled 1: Enabled Refer to "3.3.1 Current Control (Task 5)".
10	T5ECEN	0	R/W	Disables/enables expansion control (Non-interference control and voltage scalar limitation) for Task 5. 0: Disabled 1: Enabled Refer to "3.3.1 Current Control (Task 5)".
9:8	AWUMD[1:0]	00	R/W	Specifies anti-windup (AWU) control for Task 5 when PI control output limitation is performed. 00: AWU control is disabled. 01: Substitutes the result from amount of limitation / 4 into the integral term. 10: Substitutes the result from amount of limitation / 2 into the integral term. 11: Substitutes the amount of limitation into the integral term. Refer to "3.3.1 Current Control (Task 5)".

Bit	Bit Symbol	After Reset	Type	Function
7	CLPEN	0	R/W	Disables/enables phase clipping control for Task 6 when phase interpolation is performed. 0: Clipping is disabled. 1: Clipping is enabled. Refer to "3.3.2 SIN/COS calculation (Task 6)".
6:5	ATANMD [1:0]	00	R/W	Specifies ATAN calculation control for Task 4. 00, 01: Calculation is disabled. 10: Calculates the declination angle on d-q coordinate of current vector. 11: Calculates the declination angle on d-q coordinate of induced voltage vector. Refer to "3.3.7 Input Current Transformation (Phase Transformation/Coordinate Axis Transformation)".
4	VDCSEL	0	R/W	Selects the supply voltage store register for Task 2 and Task 10. 0: Stored in ***[VExVDC]***. 1: Stored in ***[VExVDCL]***. Refer to Figure 4.1.
3:2	OCRMD[1:0]	00	R/W	Specifies output control for Task 0 and Task 9. 00: Output is disabled. 01: Output is enabled. 10: Short circuit brake (Outputs are OFF in the upper-phase; outputs are ON in the lower-phase.) 11: EMG release (Outputs are OFF.) Refer to "3.3.4 Output Control".
1	ZIEN	0	R/W	Specifies zero-current detection control for Task 2. 0: Normal current detection 1: Zero-current detection Refer to "3.3.6 Input Process".
0	PVIEN	0	R/W	Disables/enables phase interpolation control for Task 6. 0: Disabled 1: Enabled Refer to "3.3.2 SIN/COS calculation (Task 6)".



Note: When a power supply voltage controlled by ***[VExVDC]*** register is corrected, select ***[VExVDCL]*** register as a storage location, and set a corrected value in ***[VExDVC]*** register.

Figure 4.1 *[VExVDC]*/*[VExVDCL]* store register

4.3.3. *[VExFMODE]* (Flow Control Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:14	SPWMMD[1:0]	00	R/W	Selects a PWM Shift mode. 00: Shift 1 01: Shift 2 (U-phase Center PWM) 10: Shift 2 (V-phase Center PWM) 11: Shift 2 (W-phase Center PWM) Shift 2 cannot be selected when output control 1 (Task 0) is being executed (invalid).
13	CCVMD	0	R/W	Selects a phase transformation mode. 0: Relative transformation 1: Absolute transformation
12	PHCVDIS	0	R/W	Disables/enables phase transformation. 0: 2-3 phase transformation is enabled. (3-phase AC output) 1: 2-3 phase transformation is disabled. (2-phase AC output) Phase transformation cannot be disabled when space vector modulation (Task 8) is running (invalid).
11:10	VSLIMMD[1:0]	00	R/W	Controls voltage scalar limitation of current control (Task 5). 00: Scalar limitation is disabled. (The limitation of each axis is enabled.) 01: Limits the voltage on the d-axis. 10: Limits the voltage on the q-axis. 11: dq proportional limitation
9	MREGDIS	0	R/W	Enables/Disables the use of previous value flags of MCTLF, SIN, COS, and SECTOR registers on the input process 1, 2, and on input coordinate transformation. 0: Enabled 1: Disabled
8	CRCEN	0	R/W	Disables/enables trigger correction 0: Disabled 1: Enabled This bit is enabled when trigger generation (Task 1) is being executed; when PWM Shift is disabled in 1-shunt current detection mode; or when Shift 1 is selected.
7	ICPLMD	0	R/W	Selects the current direction (***[VExIC]***) of input process2 (Task 10). 0: Shunt mode (***[VExIC]*** = ***[VExICO]*** – ***[VExICADC]***) 1: Sensor mode (***[VExIC]*** = ***[VExICADC]*** – ***[VExICO]***)
6	IBPLMD	0	R/W	Selects the current direction (***[VExIB]***) of input process2 (Task 10). 0: Shunt mode (***[VExIB]*** = ***[VExIBO]*** – ***[VExIBADC]***) 1: Sensor mode (***[VExIB]*** = ***[VExIBADC]*** – ***[VExIBO]***)
5	IAPLMD	0	R/W	Selects the current direction (***[VExIA]***) of input process2 (Task 10). 0: Shunt mode (***[VExIA]*** = ***[VExIAO]*** – ***[VExIAADC]***) 1: Sensor mode (***[VExIA]*** = ***[VExIAADC]*** – ***[VExIAO]***)
4	IDQSEL	0	R/W	Selects Non-interference control input current 0: Feedback current (***[VExID]***, ***[VExIQ]***) 1: Command current (***[VExIDREF]***, ***[VExIQREF]***)

Bit	Bit Symbol	After Reset	Type	Function
3:2	IDMODE[1:0]	00	R/W	Selects current detection mode. 00: 3-shunt (Note 1) 01: 2-sensor (Note 2) 10: 1-shunt (PMD TRG in the latter half of PWM(Note 3)) 11: 1-shunt (PMD TRG in the first half of PWM(Note 3)) Note 1: 3-phase current detection is used in input process 2 (Task 10). Note 2: 2-phase current detection is used in input process 2 (Task 10). Note 3: PWM Shift 2 should be used when output control 2 (Task 9) and input process 2 (Task 10) is being executed.
1	SPWMEN	0	R/W	Disables/enables PWM Shift. 0: Disabled 1: Enabled Output control 1 (Task 0) and input process 1 (Task 2) are available only in Shift 1 mode. Output control 2 (Task 9) and input process 2 (Task 10) are available only in Shift 2 mode.
0	C2PEN	0	R/W	Selects the modulation mode. 0: 3-phase modulation 1: 2-phase modulation

4.3.4. *[VExTPWM]* (PWM Period Rate setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	TPWM[15:0]	0x0000	R/W	Sets the PWM period rate and the unit of integration in phase interpolation (16-bit fixed-point data: 0.0 to 1.0). 0x0000 to 0xFFFF (PWM period[s] × Max_Hz × 2^{16}) This indicates the ratio between PWM period and the maximum rotational speed. (Max_Hz: Maximum rotational speed) This bit is used for SIN/COS calculation (Task 6) when phase interpolation is enabled.

4.3.5. *[VExOMEGA]* (Rotational Speed setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	OMEGA[15:0]	0x0000	R/W	Sets a rotational speed (16-bit fixed-point data: -1.0 to 1.0). 0x8000 to 0x7FFF: Rotational speed [Hz] / Max_Hz × 2^{15} (Max_Hz: Maximum rotational speed) This bit is used for SIN/COS calculation (Task 6) when phase interpolation is enabled. This bit is used for output control 1 (Task 0) when PWM Shift 1 is selected for 1-shunt current detection. This bit is used for current control (Task 5) and input coordinate axis transformation (Task 4) in motor voltage equation.

4.3.6. *[VExTHETA]* (Motor Phase setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	THETA[15:0]	0x0000	R/W	Sets the phase (16-bit fixed-point data: 0.0 to 1.0). 0x0000 to 0xFFFF: Phase [deg] / 360 × 2^{16} This bit is used for SIN/COS calculation (Task 6) when phase interpolation is enabled. This bit is updated when SIN/COS calculation (Task 6) is executed while phase interpolation is enabled.

4.3.7. d-axis/q-axis Current Reference setting Registers

4.3.7.1. *[VExIDREF]* (d-axis Current Reference setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	IDREF[15:0]	0x0000	R/W	Sets the reference value of d-axis current (16-bit fixed-point data: -1.0 to 1.0). 0x8000 to 0x7FFF: d-axis current reference [A] / Max_I × 2^{15} Max_I: (The amount of phase current variation when the AD conversion is changed by 1LSB [A]) × 2^{11} This bit is used for current control (Task 5).

4.3.7.2. *[VExIQREF]* (q-axis Current Reference setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	IQREF[15:0]	0x0000	R/W	Sets the reference value of q-axis current (16-bit fixed-point data: -1.0 to 1.0). 0x8000 to 0x7FFF: q-axis current reference [A] / Max_I × 2^{15} Max_I: (The amount of phase current variation when the AD conversion is changed by 1LSB [A]) × 2^{11} This bit is used for current control (Task 5).

4.3.8. d-axis/q-axis Voltage setting Registers

4.3.8.1. *[VExVD]* (d-axis Voltage setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	VD[31:0]	0x00000000	R/W	Sets the value of d-axis voltage (32-bit fixed-point data: -1.0 to 1.0). 0x80000000 to 0x7FFFFFFF: d-axis voltage[V] / Max_V × 2^{31} Max_V: (The amount of supply voltage variation when the AD conversion is changed by 1LSB [V]) × 2^{12} This bit is updated when current control (Task 5) is executed. This bit is used for output coordinate axis transformation (Task 7). This bit is used for input coordinate axis transformation (Task 4) in motor voltage equation.

4.3.8.2. *[VExVQ]* (q-axis Voltage setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	VQ[31:0]	0x00000000	R/W	Sets the value of q-axis voltage (32-bit fixed-point data: -1.0 to 1.0). 0x80000000 to 0x7FFFFFFF: q-axis voltage[V] / Max_V × 2^{31} Max_V: (The amount of supply voltage variation when the AD conversion is changed by 1LSB [V]) × 2^{12} This bit is updated when current control (Task 5) is executed. This bit is used for output coordinate axis transformation (Task 7). This bit is used for input coordinate axis transformation (Task 4) in motor voltage equation.

4.3.8.3. *[VExVDQ]* (Voltage Scalar Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	VDQ[15:0]	0x0000	R/W	Sets value of voltage scalar for d-axis voltage ***[VExVD]*** and q-axis voltage ***[VExVQ]***, or sets the limitation value of axis for d-axis voltage ***[VExVD]*** and q-axis voltage ***[VExVQ]*** when direction scalar is limited. 0x0000 to 0x7FFF: Voltage [V] / Max_V × 2^{15} Max_V: (The amount of supply voltage variation when the AD conversion is changed by 1LSB [V]) × 2^{12} This bit is updated for current control (Task 5) when expansion control is enabled.

4.3.8.4. *[VExVSLIM]* (Voltage Scalar Limitation Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	VSLIM[15:0]	0x0000	R/W	Sets the limitation value of voltage scalar for d-axis voltage ***[VExVD]*** and q-axis voltage ***[VExVQ]***. 0x0000 to 0x7FFF: Voltage [V] / Max_V × 2^{15} Max_V: (The amount of supply voltage variation when the AD conversion is changed by 1LSB [V]) × 2^{12} Limitation of scalar voltage is not effective when "0x0000" is set. This bit is used for current control (Task 5) when expansion control is enabled.

4.3.9. Coefficient Registers for PI control

4.3.9.1. *[VExCIDKI]* (d-axis Current Integral Coefficient setting Register for PI control)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	CIDKI[15:0]	0x0000	R/W	Sets the PI control integral coefficient for d-axis. 0x8000 to 0x7FFF

4.3.9.2. *[VExCIDKP]* (d-axis Current Proportional Coefficient setting Register for PI control)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	CIDKP[15:0]	0x0000	R/W	Sets the PI control proportional coefficient for d-axis. 0x8000 to 0x7FFF

4.3.9.3. *[VExCIDKG]* (d-axis Coefficient Range setting Register for PI control)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:8	CIDKPG[7:0]	0x00	R/W	Selects a PI control d-axis proportional coefficient range. 0x00: 1 / 1 0x01: $1 / 2^4$ 0x02: $1 / 2^8$ 0x03: $1 / 2^{12}$ 0x04: $1 / 2^{16}$ 0x05 to 0x08: Reserved 0x09: 2 0x0A: 2^2 0x0B: 2^3 0x0C: 2^4 0x0D to 0xFF: Reserved This bit is used for current control (Task 5).
7:0	CIDKIG[7:0]	0x00	R/W	Selects a PI control d-axis integral coefficient range. 0x00: 1 / 1 0x01: $1 / 2^4$ 0x02: $1 / 2^8$ 0x03: $1 / 2^{12}$ 0x04: $1 / 2^{16}$ 0x05 to 0xFF: Reserved This bit is used for current control (Task 5).

4.3.9.4. *[VExCIQKI]* (q-axis Current Integral Coefficient setting Register for PI control)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	CIQKI[15:0]	0x0000	R/W	Sets the PI control integral coefficient for q-axis. 0x8000 to 0x7FFF

4.3.9.5. *[VExCIQKP]* (q-axis Current Proportional Coefficient setting Register for PI control)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	CIQKP[15:0]	0x0000	R/W	Sets the PI control proportional coefficient for q-axis. 0x8000 to 0x7FFF

4.3.9.6. *[VExCIQKG]* (q-axis Coefficient Range setting Register for PI control)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:8	CIQKPG[7:0]	0x00	R/W	Selects a PI control q-axis proportional coefficient range. 0x00: 1 / 1 0x01: $1 / 2^4$ 0x02: $1 / 2^8$ 0x03: $1 / 2^{12}$ 0x04: $1 / 2^{16}$ 0x05 to 0x08: Reserved 0x09: 2 0x0A: 2^2 0x0B: 2^3 0x0C: 2^4 0x0D to 0xFF: Reserved This bit is used for current control (Task 5).
7:0	CIQKIG[7:0]	0x00	R/W	Selects a PI control q-axis integral coefficient range. 0x00: 1 / 1 0x01: $1 / 2^4$ 0x02: $1 / 2^8$ 0x03: $1 / 2^{12}$ 0x04: $1 / 2^{16}$ 0x05 to 0xFF: Reserved This bit is used for current control (Task 5).

4.3.9.7. *[VExVDIH]* (d-axis Voltage Integral component Store Register for PI control (Upper))

Bit	Bit Symbol	After Reset	Type	Function
31:0	VDIH[31:0]	0x00000000	R/W	Stores the upper 32 bits of the integral component (VDI) for d-axis on PI control.

4.3.9.8. *[VExVDILH]* (d-axis Voltage Integral component Store Register for PI control (Lower))

Bit	Bit Symbol	After Reset	Type	Function
31:16	VDILH[15:0]	0x0000	R/W	Stores the lower 16 bits of the integral component (VDI) for d-axis on PI control.
15:0	-	0	R	Read as "0"

Note 1: The VDI data is 64-bit fixed-point data (63 fractional bits from "-1.0" to "1.0").

Note 2: The VDI data consists of 48 bits.

4.3.9.9. *[VExVQIH]* (q-axis Voltage Integral component Store Register for PI control (Upper))

Bit	Bit Symbol	After Reset	Type	Function
31:0	VQIH[31:0]	0x00000000	R/W	Stores the upper 32 bits of the integral component (VQI) for q-axis on PI control.

4.3.9.10. *[VExVQILH]* (q-axis Voltage Integral component Store Register for PI control (Lower))

Bit	Bit Symbol	After Reset	Type	Function
31:16	VQILH[15:0]	0x0000	R/W	Stores the lower 16 bits of the integral component (VQI) for q-axis on PI control.
15:0	-	0	R	Read as "0"

Note 1: The VQI data is 64-bit fixed-point data (63 fractional bits from "-1.0" to "1.0").

Note 2: The VQI data consists of 48 bits.

4.3.10. *[VExPIOLIM]* (Output Limitation Register for PI control)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	PIOLIM[15:0]	0x0000	R/W	PI control output limit value setting (16-bit fixed-point data: 0.0 to 1.0). 0x0000 to 0x7FFF This bit is used for current control (Task 5).

4.3.11. *[VExFPWMCHG]* (PWM Switching Speed setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	FPWMCHG [15:0]	0x0000	R/W	Sets the switching speed when PWM Shift is enabled. 0x0000 to 0x7FFF: Switching speed [Hz] / Max_Hz × 2^{15} (Max_Hz: Maximum rotational speed [Hz]) This bit is used when output control 1 (Task 0) is executed and PWM Shift 1 is enabled in 1-shunt current detection.

4.3.12. *[VExPWMOFS]* (PWM Shift 2 Offset Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	PWMOFS [15:0]	0x0000	R/W	An offset value that is added to the duty value when Shift operation is enabled on output control 2 (Task 9). 0x0000 to 0x7FFF

4.3.13. SIN/COS Registers

4.3.13.1. *[VExCOS]* (Cosine value Register of θ)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	COS[15:0]	0x0000	R/W	Sets the cosine value on the ***[VExTHETA]*** (16-bit fixed-point data: -1.0 to 1.0). 0x8000 to 0x7FFF This bit is updated when SIN/COS calculation (Task 6) is executed. This bit is used for output coordinate axis transformation (Task 7).

4.3.13.2. *[VExSIN]* (Sine value Register of θ)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	SIN[15:0]	0x0000	R/W	Sets the sine value on the ***[VExTHETA]*** (16-bit fixed-point data: -1.0 to 1.0). 0x8000 to 0x7FFF This bit is updated when SIN/COS calculation (Task 6) is executed. This bit is used for output coordinate axis transformation (Task 7).

4.3.13.3. *[VExCOSM]* (Previous Cosine value Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	COSM[15:0]	0x0000	R/W	Stores the previous value of the ***[VExCOS]*** register. 0x8000 to 0x7FFF This bit is updated when SIN/COS calculation (Task 6) is executed. This bit is used for input coordinate axis transformation (Task 4).

4.3.13.4. *[VExSINM]* (Previous Sine value Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	SINM[15:0]	0x0000	R/W	Stores the previous value of the ***[VExSIN]*** register. 0x8000 to 0x7FFF This bit is updated when SIN/COS calculation (Task 6) is executed. This bit is used for input coordinate axis transformation (Task 4).

4.3.14. Sector Information Registers

4.3.14.1. *[VExSECTOR]* (Sector Information Register)

Bit	Bit Symbol	After Reset	Type	Function
31:4	-	0	R	Read as "0"
3:0	SECTOR[3:0]	0x0	R/W	Indicates the sector information. Setting value: 0x0 to 0xB A rotational position when output is expressed in the sector that is divided into 12 sectors by 30-degree. This bit is updated when output phase transformations (Task 8 and Task 11) are executed. This bit is used for output control 1 (Task 0).

4.3.14.2. *[VExSECTORM]* (Previous Sector Information Register)

Bit	Bit Symbol	After Reset	Type	Function
31:4	-	0	R	Read as "0"
3:0	SECTORM [3:0]	0x0	R/W	Stores the previous value of the ***[VExSECTOR]*** register. 0x0 to 0xB This bit is updated when output phase transformations (Task 8 and Task 11) are executed. This bit is used in input process (Task 2 and Task10).

4.3.15. 3-phase Current Registers

4.3.15.1. *[VExIAO]* (a-phase Zero-Current Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	IAO[15:0]	0x0000	R/W	Stores the result of AD conversion for a-phase at zero-current. (Stores the result of AD conversion for a-phase current when the motor is at stop.) This bit is updated in the input process 1 (Task 2) when zero-current detection mode is selected. The result of AD conversion for a-phase at zero current is stored in <IAO[15:4]> when the result of AD conversion is captured. <IAO[3:0]> stores "0".

4.3.15.2. *[VExIBO]* (b-phase Zero-Current Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	IBO[15:0]	0x0000	R/W	Stores the result of AD conversion for b-phase at zero-current. (Stores the result of AD conversion for b-phase current when the motor is at stop.) This bit is updated in the input process 1 (Task 2) when zero-current detection mode is selected. The result of AD conversion for a-phase at zero current is stored in <IBO[15:4]> when the result of AD conversion is captured. <IBO[3:0]> stores "0".

4.3.15.3. *[VExICO]* (c-phase Zero-Current Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	ICO[15:0]	0x0000	R/W	Stores the result of AD conversion for c-phase at zero-current. (Stores the result of AD conversion for c-phase current when the motor is at stop.) This bit is updated in the input process 1 (Task 2) when zero-current detection mode is selected. The result of AD conversion for a-phase at zero current is stored in <ICO[15:4]> when the result of AD conversion is captured. <ICO[3:0]> stores "0".

4.3.15.4. *[VExIAADC]* (ADC Conversion Result Register for a-phase Current)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	IAADC[15:0]	0x0000	R/W	Stores the result of AD conversion for a-phase current ("0x0000" to "0xFFFF"). This bit is updated when input processes (Task 2 and Task 10) are executed. The result of AD conversion for a-phase is stored in <IAADC[15:4]> when the result of AD conversion is captured. <IAADC[3:0]> stores "0".

4.3.15.5. *[VExIBADC]* (ADC Conversion Result Register for b-phase Current)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	IBADC[15:0]	0x0000	R/W	Stores the result of AD conversion for b-phase current ("0x0000" to "0xFFFF"). This bit is updated when input processes (Task 2 and Task 10) are executed. The result of AD conversion for b-phase is stored in <IBADC[15:4]> when the result of AD conversion is captured. <IBADC[3:0]> stores "0".

4.3.15.6. *[VExICADC]* (ADC Conversion Result Register for c-phase Current)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	ICADC[15:0]	0x0000	R/W	Stores the result of AD conversion for c-phase current ("0x0000" to "0xFFFF"). This bit is updated when input processes (Task 2 and Task 10) are executed. The result of AD conversion for c-phase is stored in <ICADC[15:4]> when the result of AD conversion is captured. <ICADC[3:0]> stores "0".

4.3.15.7. *[VExIA]* (a-phase Current Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	IA[31:0]	0x00000000	R/W	a-phase current (32-bit fixed point data: -1.0 to 1.0) 0x80000000 to 0x7FFFFFFF The value is updated when input process (Task 2, Task 10) is executed. The value is used on input phase transformation (Task 3).

4.3.15.8. *[VExIB]* (b-phase Current Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	IB[31:0]	0x00000000	R/W	b-phase current (32-bit fixed point data: -1.0 to 1.0) 0x80000000 to 0x7FFFFFFF The value is updated when input process (Task 2, Task 10) is executed. The value is used on input phase transformation (Task 3).

4.3.15.9. *[VExIC]* (c-phase Current Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	IC[31:0]	0x00000000	R/W	c-phase current (32-bit fixed point data: -1.0 to 1.0) 0x80000000 to 0x7FFFFFFF The value is updated when input process (Task 2, Task 10) is executed. The value is used on input phase transformation (Task 3).

4.3.16. DC Supply Voltage Registers

4.3.16.1. *[VExVDC]* (DC Supply Voltage Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	VDC[15:0]	0x0000	R/W	Sets a supply voltage (16-bit fixed-point data: 0 to 1.0). 0x0000 to 0x7FFF The actual voltage value can be calculated as: the value of VDC × the value of Max_V / 2^{15} (Max_V: (The amount of supply voltage variation when AD conversion is changed by 1LSB [V]) × 2^{12}) This bit is updated when ***[VExMODE]***<VDCSEL> is set to "0" and input processes (Task 2 and Task 10) are executed. This bit is used for output phase transformation (Tasks 8 and 11).

4.3.16.2. *[VExVDCL]* (DC2 Supply Voltage Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	VDCL[15:0]	0x0000	R/W	Sets a supply voltage (16-bit fixed-point data: 0 to 1.0). 0x0000 to 0x7FFF The actual voltage value can be calculated as: the value of VDCL × the value of Max_V / 2^{15} (Max_V: (The amount of supply voltage variation when AD conversion is changed by 1LSB [V]) × 2^{12}) This bit is updated when ***[VExMODE]***<VDCSEL> is set to "1" and input processes (Task 2 and Task 10) are executed.

4.3.17. d-axis/q-axis Current Registers

4.3.17.1. *[VExID]* (d-axis Current Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	ID[31:0]	0x00000000	R/W	Sets a d-axis current (32-bit fixed-point data: -1.0 to 1.0). 0x80000000 to 0x7FFFFFFF The actual voltage value can be calculated as: the value of ID × the value of Max_I / 2^{31} (Max_I: (The amount of phase current variation when the AD conversion is changed by 1LSB [A]) × 2^{11}) This bit is updated when input coordinate axis transformation (Task 4) is executed. This bit is used for current control (Task 5).

4.3.17.2. *[VExIQ]* (q-axis Current Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	IQ[31:0]	0x00000000	R/W	Sets a q-axis current (32-bit fixed-point data: -1.0 to 1.0). 0x80000000 to 0x7FFFFFFF The actual voltage value can be calculated as: the value of IQ × the value of Max_I / 2^{31} (Max_I: (The amount of phase current variation when the AD conversion is changed by 1LSB [A]) × 2^{11}) This bit is updated when input coordinate axis transformation (Task 4) is executed. This bit is used for current control (Task 5).

4.3.18. *[VExTADC]* (ADC Conversion Time setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0".
15:0	TADC[15:0]	0x0000	R/W	Sets an ADC conversion time (Refer to Figure 4.2). 0x0000 to 0xFFFF: ADC conversion time [s] / PWM clock period [s] Performs forward correction for a trigger timing at the PWM end when PWM Shift 1 output (Shift 1 is enabled and a low-speed flag is "1") is used in 1-shunt current detection mode. This bit is used in output control 1(Task 0).

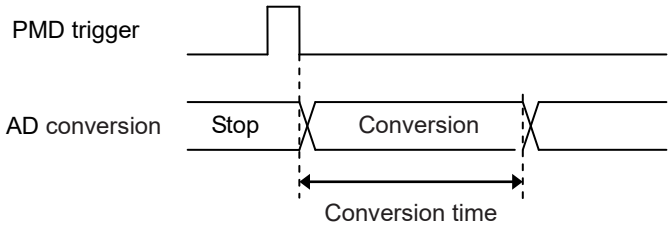


Figure 4.2 ADC conversion time

4.3.19. 3-phase PWM Duty Registers

4.3.19.1. *[VExCMPU]* (U-phase PWM Duty Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	VCMPU[15:0]	0x0000	R/W	Sets the U-phase PWM duty. 0x0000 to 0x8000 This bit is updated when output controls (Task 0 and Task 9) are executed. This bit is used for trigger generation (Task 1).

Note: Setting PMD's mode select register to VE mode can control U-phase PWM duty of PMD with this register. For details, refer to "Advanced Programmable Motor Control Circuit" in the reference manual.

4.3.19.2. *[VExCMPV]* (V-phase PWM Duty Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	VCMPV[15:0]	0x0000	R/W	Sets the V-phase PWM duty. 0x0000 to 0x8000 This bit is updated when output controls (Task 0 and Task 9) are executed. This bit is used for trigger generation (Task 1).

Note: Setting PMD's mode select register to VE mode can control V-phase PWM duty of PMD with this register. For details, refer to "Advanced Programmable Motor Control Circuit" in the reference manual.

4.3.19.3. *[VExCMPW]* (W-phase PWM Duty Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	VCMPW[15:0]	0x0000	R/W	Sets the W-phase PWM duty. 0x0000 to 0x8000 This bit is updated when output controls (Task 0 and Task 9) are executed. This bit is used for trigger generation (Task 1).

Note: Setting PMD's mode select register to VE mode can control W-phase PWM duty of PMD with this register. For details, refer to "Advanced Programmable Motor Control Circuit" in the reference manual.

4.3.19.4. *[VExMINPLS]* (Minimum Pulse Width setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	MINPLS[15:0]	0x0000	R/W	When output control 1 (task 0) is executed: Sets the reference value of the minimum pulse width difference (The minimum value of the duty difference of 3-phase PWM (***[VExCMPU]***, ***[VExCMPV]***, ***[VExCMPW]***)) when PWM shift is enabled in 1-shunt current detection mode. The calculation is as follows: Difference of duty width [s] / PWM clock period[s] × 2^{15} When output control 2 (task 9) is executed: Sets the reference value of the minimum pulse width (The minimum duty of 3-phase PWM (***[VExCMPU]***, ***[VExCMPV]***, ***[VExCMPW]***)). The calculation is as follows: Duty width [s] / PWM clock period[s] × 2^{15}

4.3.20. PWM Output Limitation Registers

4.3.20.1. *[VExPWMMAX]* (PWM Upper-Limit setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	PWMMAX [15:0]	0x0000	R/W	Sets the upper limit value of PWM duty output. 0x0000 to 0x8000 This bit is used in output controls (Task 0 and Task 9).

4.3.20.2. *[VExPWMMIN]* (PWM Lower-Limit setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	PWMMIN [15:0]	0x0000	R/W	Sets the lower limit value of PWM duty output. 0x0000 to 0x8000 This bit is used in output controls (Task 0 and Task 9).

4.3.21. *[VExOUTCR]* (PMD Output Control Register)

Bit	Bit Symbol	After Reset	Type	Function
31:9	-	0	R	Read as "0"
8	WPWM	0	R/W	Select W-phase PWM. 0: ON/OFF output 1: PWM output
7	VPWM	0	R/W	Selects V-phase PWM. 0: ON/OFF output 1: PWM output
6	UPWM	0	R/W	Selects U-phase PWM. 0: ON/OFF output 1: PWM output
5:4	WOC[1:0]	00	R/W	Selects W-phase output control. The decode circuit output is controlled by the combination of this bit, <WOC[1:0]>and <WPWM> (Refer to Table 4.3.). For details, refer to "Conduction Control Circuit" of the reference manual "Advanced Programmable Motor Control Circuit".
3:2	VOC[1:0]	00	R/W	Selects V-phase output control. The decode circuit output is controlled by the combination of this bit, <VOC[1:0]>and <VPWM> (Refer to Table 4.2.). For details, refer to "Conduction Control Circuit" of the reference manual "Advanced Programmable Motor Control Circuit".
1:0	UOC[1:0]	00	R/W	Selects U-phase output control. The decode circuit output is controlled by the combination of this bit, <UOC[1:0]>and <UPWM> (Refer to Table 4.1.). For details, refer to "Conduction Control Circuit" of the reference manual "Advanced Programmable Motor Control Circuit".

Note 1: Setting PMD's mode select register to VE mode can control ***[PMDxMDOUT]*** of PMD with this register. For details, refer to "Advanced Programmable Motor Control Circuit" in the reference manual.

Note 2: This setting is updated when output controls (Task 0 and Task 9) are executed.

Output control of U, V and W-phase of the PMD is shown below: (This table shows only those combinations that are used in the VE.)

Table 4.1 <UPWM>, <UOC[1:0]> PMD setting: Output control of U-phase(UOx, XOx)

Setting		Output	
<UPWM>	<UOC[1:0]>	UOx	XOx
0	00	OFF output	OFF output
0	01	OFF output	ON output
1	00	PWMU inverted output	PWMU output
1	11	PWMU output	PWMU inverted output

Table 4.2 <VPWM>, <VOC[1:0]> PMD setting: Output control of V-phase(VOx, YOx)

Setting		Output	
<VPWM>	<VOC[1:0]>	VOx	YOx
0	00	OFF output	OFF output
0	01	OFF output	ON output
1	00	PWMV inverted output	PWMV output
1	11	PWMV output	PWMV inverted output

Table 4.3 <WPWM>, <WOC[1:0]> PMD setting: Output control of W-phase(WOx, ZOx)

Setting		Output	
<WPWM>	<WOC[1:0]>	WOx	ZOx
0	00	OFF output	OFF output
0	01	OFF output	ON output
1	00	PWMW inverted output	PWMW output
1	11	PWMW output	PWMW inverted output

4.3.22. Trigger Generation Registers

4.3.22.1. *[VExTRGCRC]* (Synchronous Trigger Correction setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	TRGCRC [15:0]	0x0000	R/W	Corrects the trigger timing. 0x0000 to 0xFFFF: Correction time[s] / PWM period[s] This bit is enabled only when PWM Shift is disabled in the 1-shunt current detection or when Shift 1 is enabled. This bit is used to backward correct the trigger timing. This bit is used for trigger generation (Task 1).

4.3.22.2. *[VExTRGCMP0]* (PMD Trigger 0 Timing setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	VTRGCMP0 [15:0]	0x0000	R/W	Specifies the trigger timing for sampling ADC in synchronization with the PMD (PMD setting). 0x0000: Prohibited 0x0001 to 0x7FFF: Trigger timing 0x8000 to 0xFFFF: Prohibited This bit is enabled when one of the following PMD trigger modes is selected: The coincidence in the first half of the PWM cycle; The coincidence in the latter half of the PWM cycle; The coincidence in the first and later halves of the PWM cycle This bit is updated in trigger generation (Task 1) when PWM Shift is disabled in the 1-shunt current detection or when Shift 1 is enabled.

Note: Setting PMD's mode select register to VE mode can control the trigger comparison 0 of PMD with this register. For details, refer to "Advanced Programmable Motor Control Circuit" in the reference manual.

4.3.22.3. *[VExTRGCMP1]* (PMD Trigger 1 Timing setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	VTRGCMP1 [15:0]	0x0000	R/W	Specifies the trigger timing for sampling ADC in synchronization with the PMD (PMD setting). 0x0000: Prohibited 0x0001 to 0x7FFF: Trigger timing 0x8000 to 0xFFFF: Prohibited This bit is enabled when one of the following PMD trigger modes is selected: The coincidence in the first half of the PWM cycle; The coincidence in the latter half of the PWM cycle; The coincidence in the first and later halves of the PWM cycle This register is invalid when the trigger output mode in PMD is set to the trigger selected output (***[PMDxTRGMD]*** <TRGOUT> = 1). This bit is updated in trigger generation (Task 1) when PWM Shift is disabled in the 1-shunt current detection or when Shift 1 is enabled.

Note: Setting PMD's mode select register to VE mode can control the trigger comparison 1 of PMD with this register. For details, refer to "Advanced Programmable Motor Control Circuit" in the reference manual.

4.3.22.4. *[VExTRGSEL]* (Synchronous Trigger Selection setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:3	-	0	R	Read as "0"
2:0	VTRGSEL[2:0]	0x0	R/W	Select the synchronous trigger number that is output at the timing specified in ***[VExTRGCMP0]*** (PMD setting). 0x0 to 0x5: Output trigger number 0x6, 0x7: Prohibited This bit is enabled when trigger selection output (***[PMDxTRGMD]***<TRGOUT> = 1) is selected as PMD trigger output mode. This bit is updated to the value of ***[VExSECTOR]*** / 2 in trigger generation (Task 1).

Note: Setting PMD's mode select register to VE mode can control the trigger output selection of PMD with this register. For details, refer to "Advanced Programmable Motor Control Circuit" in the reference manual.

4.3.23. *[VExEMGRS]* (EMG release Register)

Bit	Bit Symbol	After Reset	Type	Function
31:1	-	0	R	Read as "0"
0	EMGRS	0	W	EMG release command (PMD setting) 0: - 1: EMG release If "1" is written to this bit, this bit is cleared on next cycle. Always read as "0". This bit is set to "1" when output control (Task 0 and Task 9) is executed in EMG release mode.

Note: Setting PMD's mode select register to VE mode can control the return from EMG protection of PMD with this register. For details, refer to "Advanced Programmable Motor Control Circuit" in the reference manual.

4.3.24. *[VExDELTA]* (Current Declination Angle Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	DELTA[15:0]	0x0000	R/W	Sets the current declination angle on the d-q coordinate. (-180-degree to 180-degree) 0x8000 to 0x7FFF: The declination angle: [deg] / 360 × 2^{16} This bit is updated when ATAN calculation is enabled on coordinate axis transformation (Task 4) or when voltage scalar limitation is enabled in current control (Task 5).

4.3.25. Motor Constant Registers

4.3.25.1. *[VExCPHI]* (Motor Interlinkage Magnetic Flux Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	CPHI[15:0]	0x0000	R/W	Sets the back electromotive force constant of the motor [V/rps] (interlinkage flux [Wb/s]) 0x0000 to 0x7FFF: The value of back electromotive force constant [V/rps] / Max_V × Max_Hz × 2^{11} / [CPHIG setting] This bit is used in coordinate axis transformation (Task 4) when the declination angle of induced voltage is calculated or when Non-interference control is enabled in current control (Task 5).

4.3.25.2. *[VExCLD]* (Motor d-axis Inductance Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	CLD[15:0]	0x0000	R/W	Sets the d-axis inductance of the motor constant. 0x0000 to 0x7FFF: The value of inductance [H] × Max_I / Max_V × Max_Hz × 2π × 2^{11} / [CLG setting] This bit is used in coordinate axis transformation (Task 4) when the declination angle of induced voltage is calculated or when Non-interference control is enabled in current control (Task 5).

4.3.25.3. *[VExCLQ]* (Motor q-axis Inductance Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	CLQ[15:0]	0x0000	R/W	Sets the q-axis inductance of the motor constant. 0x0000 to 0x7FFF: The value of inductance [H] × Max_I / Max_V × Max_Hz × 2π × 2^{11} / [CLG setting] This bit is used in coordinate axis transformation (Task 4) when the declination angle of induced voltage is calculated or when Non-interference control is enabled in current control (Task 5).

4.3.25.4. *[VExCR]* (Motor Resistance Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	CR[15:0]	0x0000	R/W	Sets the value of resistance of the motor constant. 0x0000 to 0x7FFF: The value of resistor [Ω] × Max_I / Max_V × 2^{11} /(***[VExCRG]*** setting) This bit is used in coordinate axis transformation (Task 4) when the declination angle of induced voltage is calculated or when Non-interference control is enabled in current control (Task 5).

4.3.25.5. *[VExCPHIG]* (Motor Magnetic Flux Range setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:8	-	0	R	Read as "0"
7:3	-	00000	R/W	Write as "00000"
2:0	CPHIG[2:0]	000	R/W	Selects the range of magnetic flux of the motor constant. 000: 1 / 1 001: 1 / 2^4 010: 1 / 2^8 011: 1 / 2^{12} 100: 1 / 2^{16} 101 to 111: Reserved This bit is used in coordinate axis transformation (Task 4) when the declination angle of induced voltage is calculated or when Non-interference control is enabled in current control (Task 5).

4.3.25.6. *[VExCLG]* (Motor Inductance Range setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:8	-	0	R	Read as "0"
7:3	-	00000	R/W	Write as "00000"
2:0	CLG[2:0]	000	R/W	Selects the range of inductance of the motor constant. 000: 1 / 1 001: 1 / 2^4 010: 1 / 2^8 011: 1 / 2^{12} 100: 1 / 2^{16} 101 to 111: Reserved This bit is used in coordinate axis transformation (Task 4) when the declination angle of induced voltage is calculated or when Non-interference control is enabled in current control (Task 5).

4.3.25.7. *[VExCRG]* (Motor Resistance Range setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:8	-	0	R	Read as "0"
7:3	-	00000	R/W	Write as "00000"
2:0	CRG[2:0]	000	R/W	Selects the range of the resistance of the motor constant. 000: 1 / 1 001: 1 / 2^4 010: 1 / 2^8 011: 1 / 2^{12} 100: 1 / 2^{16} 101 to 111: Reserved This bit is used in coordinate axis transformation (Task 4) when the declination angle of induced voltage is calculated or when Non-interference control is enabled in current control (Task 5).

4.3.26. *[VExVDE]* (d-axis Voltage Register for Non-interference Control)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	VDE[15:0]	0x0000	R/W	Indicates the value of d-axis calculation of non-interference (16-bit fixed-point data: -1.0 to 1.0). 0x8000 to 0x7FFF: Voltage [V] / Max_V × 2^{15} (Max_V: (The amount of voltage variation when the AD conversion is changed by 1LSB [V]) × 2^{12}) This bit is updated when Non-interference control is enabled in current control (Task 5).

4.3.27. *[VExVQE]* (q-axis Voltage Register for Non-interference Control)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	VQE[15:0]	0x0000	R/W	Indicates the value of q-axis calculation of non-interference (16-bit fixed-point data: -1.0 to 1.0). 0x8000 to 0x7FFF: Voltage [V] / Max_V × 2^{15} (Max_V: (The amount of voltage variation when the AD conversion is changed by 1LSB [V]) × 2^{12}) This bit is updated when Non-interference control is enabled in current control (Task 5).

4.3.28. *[VExDTC]* (Dead Time Compensation Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	DTC[15:0]	0x0000	R/W	Sets the amount of compensation on dead time control. 0x0000 to 0x7FFF: Dead time[s] / PWM period[s] × 2^{15} This bit is used when dead time compensation is enabled in output controls (Task 0 and Task 9).

4.3.29. *[VExHYS]* (Hysteresis Width Register for Current Polarity Determination)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	HYS[15:0]	0x0000	R/W	Sets the current hysteresis width when current polarity is determined (16-bit fixed-point data: -1.0 to 1.0). 0x8000 to 0x7FFF: Hysteresis width [A] / Max_I × 2^{15} (Max_I: (The amount of phase current variation when the AD conversion is changed by 1LSB [A]) × 2^{11}) This bit is used in input process (Task 2 and Task 10) when current polarity determination is enabled.

4.3.30. *[VExHYS2]* (Hysteresis Width Register 2 for Current Polarity Determination)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	HYS2[15:0]	0x0000	R/W	Sets a current reverse hysteresis width on polarity determination. (16-bit fixed point data: 0 to 1.0) 0x0000 to 0x7FFF: Hysteresis width [A] / Max_I × 2^{15} (Max_I: Phase current variation when A/D conversion data is changed for 1LSB [A]) × 2^{11}) The value is used on input process (Task2 and Task 10) when current polarity determination is enabled. Note: This register is invalid when ***[VExHYS]*** ≥ 0.

4.3.31. *[VExDTCS]* (Dead Time Compensation Control/Status Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:11	-	00000	R/W	Write as "00000"
10:8	ICSTS[2:0]	000	R/W	Indicates status of the current ***[VExIC]*** polarity determination, or selects dead time compensation control setting for ***[VExCMPW]***. 000, 010, 100, 110: Polarity determination is undefined. Dead time compensation is not performed. 001, 101: Positive current is determined. Adds the ***[VExDTC]*** value when dead time compensation is performed. 011, 111: Negative current is determined. Subtracts the ***[VExDTC]*** value when dead time compensation is performed. This bit is updated in input process (Task 2 and Task 10) when current polarity determination is enabled. This bit is used in output controls (Task 0 and Task 9) when dead time compensation is enabled.
7	-	0	R/W	Write as "0"
6:4	IBSTS[2:0]	000	R/W	Indicates status of the current ***[VExIB]*** polarity determination, or selects dead time compensation control setting for ***[VExCMPV]***. 000, 010, 100, 110: Polarity determination is undefined. Dead time compensation is not performed. 001, 101: Positive current is determined. Adds the ***[VExDTC]*** value when dead time compensation is performed. 011, 111: Negative current is determined. Subtracts the ***[VExDTC]*** value when dead time compensation is performed. This bit is updated in input process (Task 2 and Task 10) when current polarity determination is enabled. This bit is used in output controls (Task 0 and Task 9) when dead time compensation is enabled.
3	-	0	R/W	Write as "0"

2:0	IASTS[2:0]	000	R/W	Indicates status of the current ***[VExIA]*** polarity determination, or selects dead time compensation control setting for ***[VExCMPU]***. 000, 010, 100, 110: Polarity determination is undefined. Dead time compensation is not performed. 001, 101: Positive current is determined. Adds the ***[VExDTC]*** value when dead time compensation is performed. 011, 111: Negative current is determined. Subtracts the ***[VExDTC]*** value when dead time compensation is performed. This bit is updated in input process (Task 2 and Task 10) when current polarity determination is enabled. This bit is used in output controls (Task 0 and Task 9) when dead time compensation is enabled.

4.3.32. *[VExTHTCLP]* (Clipped Phase value setting Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	THTCLP[15:0]	0x0000	R/W	Sets the clipping phase of the angle of ***[VExTHETA]*** when phase interpolation is performed (16-bit fixed-point data: 0.0 to 1.0). 0x0000 to 0xFFFF: Phase [deg] / 360 × 2^{16} This bit is used in SIN/COS calculation (Task 5) when phase interpolation is enabled.

4.3.33. 2-phase Voltage Registers

4.3.33.1. *[VExVALPHA]* (α-phase Voltage Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	VALPHA[31:0]	0x00000000	R/W	α-phase voltage (32-bit fixed point data: -1.0 to 1.0) 0x80000000 to 0x7FFFFFFF To convert an actual voltage value, use the following formula: Register value × Max_V / 2^{31} (Max_V: (Phase voltage variation when A/D conversion data is changed for 1LSB [V]) × 2^{11}) The value is updated when output coordinate axis transformation (Task 7) is executed. The value is used on output phase transformation (Task 8, Task 11).

4.3.33.2. *[VExVBETA]* (β-phase Voltage Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	VBETA[31:0]	0x00000000	R/W	β-phase voltage (32-bit fixed point data: -1.0 to 1.0) 0x80000000 to 0x7FFFFFFF To convert an actual voltage value, use the following formula: Register value × Max_V / 2^{31} (Max_V: (Phase voltage variation when A/D conversion data is changed for 1LSB [V]) × 2^{11}) The value is updated when output coordinate axis transformation (Task 7) is executed. The value is used on output phase transformation (Task 8, Task 11).

4.3.34. 3-phase Voltage Duty Registers

4.3.34.1. *[VExVDUTYA]* (a-phase Voltage Duty Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	VDUTYA[31:0]	0x00000000	R/W	a-phase voltage duty (32-bit fixed point data: -1.0 to 1.0) 0x80000000 to 0x7FFFFFFF The value is updated on output phase transformation (Task 8, Task 11) is executed. The value is used on output control (Task 0, Task 9).

4.3.34.2. *[VExVDUTYB]* (b-phase Voltage Duty Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	VDUTYB[31:0]	0x00000000	R/W	b-phase voltage duty (32-bit fixed point data: -1.0 to 1.0) 0x80000000 to 0x7FFFFFFF The value is updated on output phase transformation (Task 8, Task 11) is executed. The value is used on output control (Task 0, Task 9).

4.3.34.3. *[VExVDUTYC]* (c-phase Voltage Duty Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	VDUTYC[31:0]	0x00000000	R/W	c-phase voltage duty (32-bit fixed point data: -1.0 to 1.0) 0x80000000 to 0x7FFFFFFF The value is updated on output phase transformation (Task 8, Task 11) is executed. The value is used on output control (Task 0, Task 9).

4.3.35. 2-phase Current Registers

4.3.35.1. *[VExIALPHA]* (α-phase Current Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	IALPHA[31:0]	0x00000000	R/W	α-phase current (32-bit fixed point data: -1.0 to 1.0) 0x80000000 to 0x7FFFFFFF The value is updated on input phase transformation (Task 3). The value is used on input coordination axis transformation (Task 4).

4.3.35.2. *[VExIBETA]* (β-phase Current Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	IBETA[31:0]	0x00000000	R/W	β-phase current (32-bit fixed point data: -1.0 to 1.0) 0x80000000 to 0x7FFFFFFF The value is updated on input phase transformation (Task 3). The value is used on input coordination axis transformation (Task 4).

4.3.36. *[VExVDELTA]* (Voltage Declination Angle Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	VDELTA[15:0]	0x0000	R/W	Sets the voltage declination angle on the d-q coordinate. (0-degree to 90-degree) 0x0000 to 0x4000: The declination angle [deg] / 360 × 2^{16} The value is updated when expansion control of current control (Task 5) is enabled and voltage scalar limitation is valid.

4.3.37. *[VExVDCRC]* (d-axis Voltage Correction Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	VDCRC[15:0]	0x0000	R/W	d-axis voltage correction (16-bit fixed point data: -1.0 to 1.0) 0x8000 to 0x7FFF The value is updated when expansion control of current control (Task 5) is enabled and voltage scalar limitation is valid.

4.3.38. *[VExVQCRC]* (q-axis Voltage Correction Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0"
15:0	VQCRC[15:0]	0x0000	R/W	q-axis voltage correction (16-bit fixed point data: -1.0 to 1.0) 0x8000 to 0x7FFF The value is updated when expansion control of current control (Task 5) is enabled and voltage scalar limitation is valid.

5. Precautions

- The Advanced Vector Engine Plus (A-VE+) is used in combination with the Advanced Programmable Motor Driver (A-PMD) and 12-bit Analog to Digital converter (ADC).

  – In order for the A-PMD to use the A-VE+ calculation results, the mode selection register (***[PMDxMODESEL]***) of the A-PMD should be set to the VE mode.

  – In order for the A-PMD to use the trigger timing value (***[VExTRGCMP0]***, ***[VExTRGCMP1]***)(Note) calculated by the Trigger generation task, Trigger comparison carrier selection of the A-PMD should be set to the basic carrier (***[PMDxTRGCR]***<CARSEL> = 0).

    Note: The trigger timing is calculated when the PWM shift prohibition or PWM shift 1 is set in 1-shunt current detection mode.

  – When the vector engine is used, the PMD trigger conversion programs (Note) of the ADC according to the synchronous trigger from the A-PMD should be set.

    Note: PMD trigger enable, AIN selection, and the result register selection.

- When attempting to stop supplying the clock, make sure to check whether the A-VE+ is stopping. Note that when the MCU enters STOP1/STOP2 mode, make sure to check whether the A-VE+ is stopping as well.

6. Revision History

Table 6.1 Revision history

Revision	Date	Description
1.0	2017-11-20	-First release
2.0	2018-01-30	- "Related document" Add line: "12-bit Analog to Digital Converter" - Chapter 1 / Table :Add note: Note1, Note2 - "3.1. Clock Supply" Review of description - "4.1. List of Registers" Add to table of base address: "Type" row - " Precautions" Add section "When attempting to stop supplying the clock, "
3.0	2018-05-22	- "2.1." Title: vector engine→VE "Table 2.1" Modified signal name. - "2.2." Title, body vector engine→VE, motor control circuit→PMD, AD converter→ADC 3rd stage: "conversion result register"→"conversion result storage register" "When the conversion result is read, phase information programmed according to synchronous triggers from the PMD ..." → "When reading the conversion result, phase information set in the conversion program for each synchronous trigger from the PMD ..." Note: Deleted "()" in 2nd and 3rd line. Figure 2.2: Modified terminal name - "3.1." Modified register name. - "3.2." to "3.2.3." vector engine→VE - "3.2.2." "***[VExREPTIME]*** ≥ 2"→"***[VExREPTIME]*** ≥ 1" Table 3.3: ADC unit A interrupt→ INTADxPDA ADC unit B interrupt→INTADxPDB - "3.3.1." a. d-axis current PI control: "cidkp = ***[VExCIDKI]*** × kpg" →"cidkp = ***[VExCIDKP]*** × kpg" ***[VExVDI]*** →VDI b. q-axis current PI control: "ciqkp = ***[VExCIQKI]*** × kpg"→"ciqkp = ***[VExCIQKP]*** × kpg" ***[VExVQI]*** →VQI in Table: "***]]***"→"***]***" - "3.3.4." Note: Deleted "()" in 3rd and 4th line. - "3.3.4.1." in "d. Output control /PWM Shift 1 transformation": ***[VExSECTOR[0]]***" →"***[VExSECTOR]***<[0]>" - "3.3.5." Second line from the bottom of trigger generation formula in Table: TRGCOM1→TRGCOM0, TRGCOM0→TRGCOM1 - "3.3.6." AD converter → ADC - "3.3.6.1." 11 places in <Equation>: Added ")" - "4.1." Table: "Function"→"Peripheral Function", Added TYPE3 - "4.2.1." VEEN/Function, Note1: vector engine→VE Added Note2 - "4.2.5." Function: Changed to hexadecimal notation Note: 0→0x0 - "4.3.1." Bit 13:12, Bit7:6 / After Reset, Function: 0→00 LVTF, LAVFM/Type: W→R/W - "4.3.23." EMGRS/Type: R/W→W - "4.3.25.5." "CPHIG[7:0]" row is separated "-" row and "CPHIG[2:0]" row. - "4.3.25.6." "CLG[7:0]" row is separated "-" row and "CLG[2:0]" row. - "4.3.25.7." "CRG[7:0]" row is separated "-" row and "CRG[2:0]" row. - "5." 1st term: AD converter→12-bit Analog to Digital converter program→conversion program Last term: STOP mode→STOP1/STOP2 mode

Revision	Date	Description
3.1	2021-02-24	- Preface Modified trademark description. - Overall "AD conversion result" →"ADC conversion result" "AD conversion time" →"ADC conversion time" "ADC interrupt" →"ADC conversion end interrupt" "Arc tangent" →"Arctangent" "Schedule completion interrupt", "VE interrupt" →"Schedule end interrupt" "Task completion interrupt" →"Task end interrupt" "EMG return", "EMG protection release" →"EMG release" - 1. Outlines Input process task / Operation in Table: Added "(Note2)" - 2.1. Configuration of the VE Table2.1 Signal name column: "Energization" →"conduction" "Current 0" →"Current 1", "Current 1" →"Current 2", "Current 2" →"Current 3" - 2.2. Interaction between the VE, PMD, and ADC "VE registers." →"VE registers(Note)." Deleted "It can also read the conversion result from the two units ADCs and supports 2-phase current synchronous sampling.". - Figure 2.2: "Timer trigger" →"General purpose trigger" "Monitor output" →"Monitor function output" - 3.2. Schedule Management Table Function: "motor rotation" →"motor rotation speed" - 3.2.1. Schedule Control Table 3.1 Term row: "ATAN2" →"ATAN2 calculation" "Square root" →"Square root calculation" - 3.2.2. Start Control "<VEEN>" →"***[VExEN]***<VEEN>", "task specify register" →"task selection register" "vector engine" →"VE" - 3.3. Description of Tasks Figure 3.3: "VDC, VDC" →"VDC, VDCL" - 3.3.1. Current Control (Task 5) b. term, Table ***[VExCIQKG]*** / Function: Added bit area. ***[VExMCTLF]*** / Function: "[8]" →"[9]" c. term, <Equation>: "***[VExVDE]*** = ***[VExOMEGA]*** × iq × lq" → "***[VExVDE]*** = −***[VExOMEGA]*** × iq × lq" Table ***[VExCLG]*** /Function: "Inductance" →"Motor Inductance" ***[VExCPHIG]*** / Function: "***[VExCPHI]***" → "Motor Interlinkage magnetic flux" - 3.3.2. SIN/COS calculation (Task 6) a. term: "phase" →"phase interpolation", "***[THTCLP]***" →"***[VExTHTCLP]***" - 3.3.3.2. Output phase transformation 1 (Space vector transformation) (Task 8) b. term: "<SECTOR>" →"***[VExSECTOR]***" Table ***[VExVDC]*** / Function: "Supply voltage" →"DC supply voltage" - 3.3.3.3. Output phase transformation 2 (Inverse Clarke transformation) (Task 11) "<PIGSEL>" →"<PHCVDIS>" - 3.3.4. Output Control "control setting" →"control operation setting" - 3.3.4.1. Output control 1 (Task 0) "Shift 1 mode" →"PWM Shift 1" a. term: "duty" →"PWM duty" b. term Table ***[VExMODE]***/Function: "limited PWM" →"limited" c. term <Equation>: "<IASTS> = 01" →"<IASTS> = 001, 101" "<IASTS> = 11" →"<IASTS> = 011, 111" Table ***[VExDTCS]***/Function: Modified 2-bit notation to 3-bit notation - 3.3.4.2. Output control 2 (Task 9) "Shift 2 mode", "PWM Shift 2 mode" →"PWM Shift 2" b. term Table ***[VExMODE]***/Function: "limited PWM" →"limited" c. term <Equation>: "<IASTS> = 01" →"<IASTS> = 001, 101" "<IASTS> = 11" →"<IASTS> = 011, 111" Table ***[VExDTCS]***/Function: Modified 2-bit notation to 3-bit notation - 3.3.5. Trigger Generation (Task 1) Note2: "PWM Shift 2 mode" →"PWM Shift 2" Trigger generation formula column: "0x4000" →"0x3FFF", "0x8000" →"0x7FFF" Table term row: "Shift setting" →"PWM Shift setting" ***[VExTRGCMP0]***, ***[VExTRGCMP1]***/Function: Added "PMD" 1-shunt(First half)/PWM Shift 1/2-phase/Low-speed term: "***[VExTRGCMP0]*** = 0x4000 + ***[VExTRGCRC]***" → "***[VExTRGCMP0]*** = 0x0000 + ***[VExTADC]*** + ***[VExTRGCRC]***" Table [VExMODE]/Detail: "11: EMG protection release" → "EMG release" ***[VExTRGCMP0]***, ***[VExTRGCMP1]*** / Function: Added "PMD" - 3.3.6.1. and 3.3.6.2. "PWM Shift 2 mode" →"PWM Shift 2"

Revision	Date	Description
		a. term Table Function column: "Supply voltage" →"DC supply voltage" b. term Figure: "Hysteresis: 0 ≤ HYS" →"Hysteresis: 0 ≥ HYS" Table Detail: Changed convention. - 3.3.7.1. Input phase transformation (Task 3) Table: "phase transformation setting" →"Disables phase transformation" - 3.3.7.2. Input coordinate axis transformation (Task 4) in b. term "d-axis current" →"d-axis and q-axis current" <Equation>: "|***[VExID]***" →"|***[VExID]***| Table Detail: Changed convention. - 3.3.8.1. and 3.3.8.2. Table Detail: Changed convention. - 4.1. List of Registers VE control register table: "Vector Engine" →"VE" Dedicated registers table: "Trigger Correction"→"Synchronous Trigger Correction" - 4.2.1. and 4.2.8. title: "Vector Engine" →"VE" - 4.3.1. LVTF, LAVF/Function: Changed convention. LAVFM/Function: Deleted "Indicates the" and "and PWM Shift is enabled in 1-shunt current detection". - 4.3.3. VSLIMMD[1:0], ICPLMD, IBPLMD, and IAPLMD/Function: Changed convention. - 4.3.6 THETA[15:0]/Function: Changed convention. - 4.3.10 PIOLIM[15:0]/Function: Changed convention. - 4.3.12 PWMOFS[15:0]/Function: Changed convention. - 4 3.14.2 SECTORM[3:0]/Function: Changed convention. - 4.3.15.7. to 4.3.17.2. Changed convention - 4.3.18 TADC[15:0]/Function: Changed convention. - 4.3.19.1. to 4.3.19.3. Changed convention. - 4.3.20.1. and 4.3.20.2. Changed convention. - 4.3.21. Modified function description of <WOC[1:0]>, <VOC[1:0]>, <UOC[1:0]>. Note1: "U-phase PWM duty" →"***[PMDxMDOUT]***" Added "0 01 OFF Output ON Output" row in Tables 4.1 to Table 4.3. - 4.3.22.1. Title: "Trigger Correction" →"Synchronous Trigger Correction" Table Function: "Setting value" →"0x0000 to 0xFFFF" - 4.3.22.4. Table Function: "Specifies" →"Select", "0x6 to 0x7" →"0x6, 0x7" - 4.3.24. to 4.3.25.4. Changed convention. - 4.3.25.2. to 4.3.25.4. Title: Deleted "Constant". - 4.3.26. and 4.3.27. Changed convention. - 4.3.29. Changed convention. - 4.3.32. Changed convention. - 4.3.33.1 and 4.3.33.2. Changed convention. - 4.3.34.1. to 4.3.34.3. Changed convention. - 4.3.35.1 and 4.3.35.2. Changed convention. - 4.3.36. Changed convention. - 4.3.37. Changed convention. 4.3.38. Changed convention. - 5. Precautions Note: Deleted "PMD trigger program completion interrupt enable,".

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**