

Application Note

AN2519/D
Rev. 0, 5/2003

3Sin with Dead-Time
Correction – XOR version
TPU Function Set (3SinDtXor)

By Milan Brejl, Ph.D.

Functional Overview

The 3-Phase Sine Wave Generator with Dead-Time Correction – XOR version (3SinDtXor) is a version of the 3-Phase Sine Wave Generator with Dead-Time Correction (3SinDt) function that uses two TPU channels to generate one PWM output channel. The TPU channel outputs are to be connected to a XOR gate whose output is the required PWM signal. See **Figure 1**. An advantage of this solution is that it provides the full range 0% to 100% of PWM duty-cycle ratios. There is no *MPW* (minimum pulse width) parameter to limit the edge duty-cycle ratios in this version, unlike in the 3SinDt. A disadvantage is that the number of assigned TPU channels is doubled.

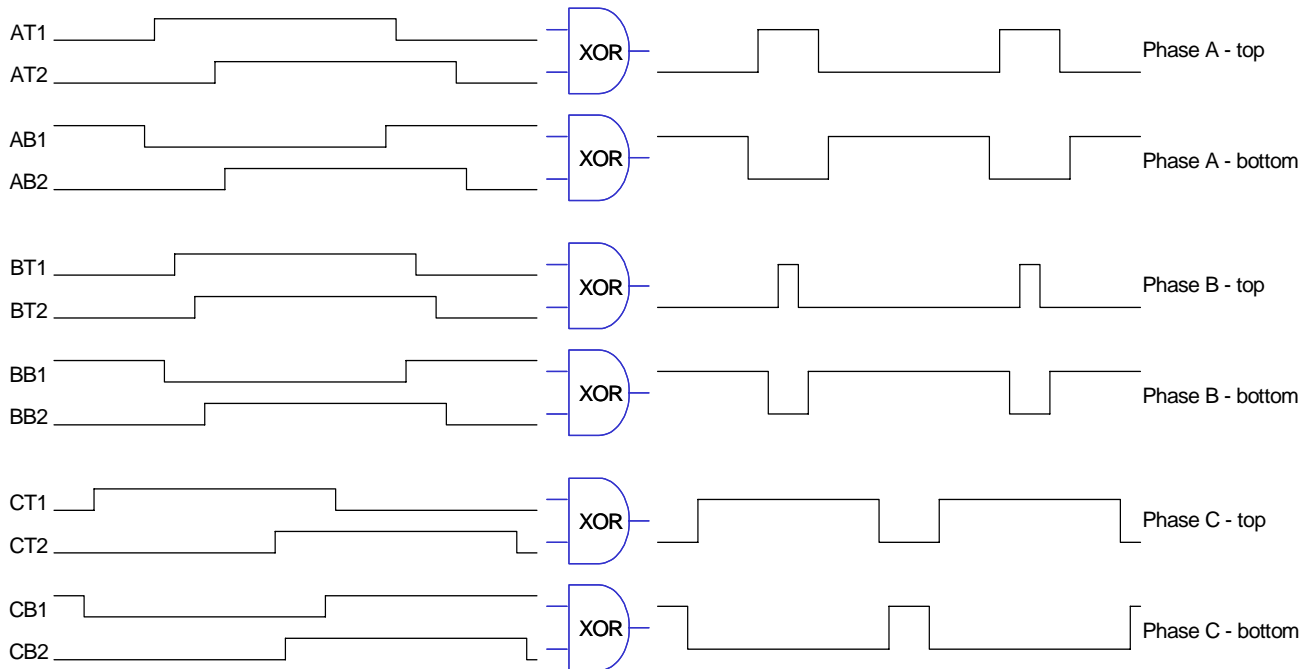


Figure 1. Functionality of XOR version – illustration

The dead-time correction technique requires knowledge of the instantaneous direction of phase currents. In the case of positive phase current the top channel high-time is equal to the calculated high-time, and the bottom channel has to control the dead-time. In case of negative phase current the bottom channel low-time is equal to the calculated high-time, and the top channel has to control the dead-time. See [Figure 2](#).

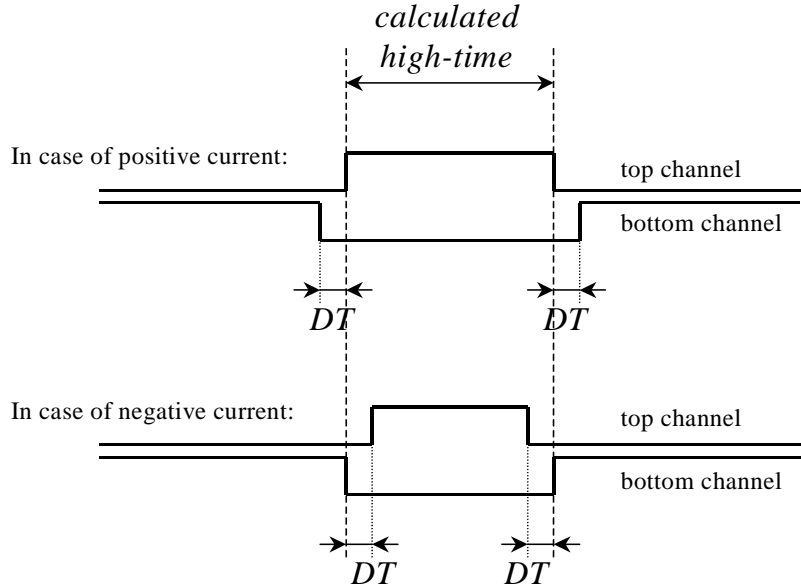


Figure 2. Dead-Time Correction Technique

The function set consists of 5 TPU functions:

- 3-Phase Sine Wave Generator with Dead-Time Correction – XOR version – R channels (3SinDtXor_R)
- 3-Phase Sine Wave Generator with Dead-Time Correction – XOR version – T channels (3SinDtXor_T)
- Synchronization Signal for 3-Phase Sine Wave Generator with Dead-Time Correction – XOR version (3SinDtXor_sync)
- Resolver Reference Signal for 3-Phase Sine Wave Generator with Dead-Time Correction – XOR version (3SinDtXor_res)
- Fault Input for 3-Phase Sine Wave Generator with Dead-Time Correction – XOR version (3SinDtXor_fault)

The 3SinDtXor_R and 3SinDtXor_T TPU functions work together to generate 6 pairs of XOR gate inputs. The XOR gate outputs then produce a 6-channel 3-phase center-aligned PWM signal with dead-time between the top and bottom channels. The Synchronization Signal for the 3SinDtXor function can be used to generate one adjustable signal for a wide range of uses, which is synchronized to the PWM, and tracks changes in the PWM period. The

Resolver Reference Signal for the 3SinDtXor function can be used to generate one 50% duty-cycle adjustable signal, which is also synchronized to the PWM. The Fault Input for the 3SinDtXor function is a TPU input function that sets all XOR gate outputs low when the input signal goes low.

Function Set Configuration

None of the TPU functions in the 3-Phase Sine Wave Generator with Dead-Time Correction – XOR version TPU function set can be used separately. The 3SinDtXor_R and 3SinDtXor_T functions have to be used together. The 3SinDtXor_R runs on pins AB1, BB1, CB1 – see [Figure 1](#). 3SinDtXor_T runs on the other pins. The 3SinDtXor_R and 3SinDtXor_T functions use a table of 32 cosine function values. The table is placed in the parameter space of four consecutive channels. One channel running Synchronization Signal for 3SinDtXor and one channel running Resolver Reference Signal for 3SinDtXor functions can be added to the 3SinDtXor_R and 3SinDtXor_T functions. They can run on one of the channels where the cosine table values are placed, because the 3SinDtXor_sync and 3SinDtXor_res parameters are placed on two 3SinDtXor_T channels. The function Fault Input for 3SinDtXor can also be added to the 3SinDtXor_R and 3SinDtXor_T functions. It is recommended to use it on channel 15, and to set the hardware option that disables all TPU output pins when the channel 15 input signal is low (DTPU bit = 1). This ensures that the hardware reacts quickly to a pin fault state. Note that it is not only the PWM channels, but all TPU output channels, including the synchronization and resolver reference signals, that are disabled in this configuration. The function 3SinDtXor_fault can run on one of the four channels where the table of cosine function values is placed, because the 3SinDtXor_fault function does not have any parameters.

[Table 1](#) shows the configuration options and restrictions.

Table 1. 3SinDtXor TPU function set configuration options and restrictions

TPU function	Optional/Mandatory	How many channels	Assignable channels
3SinDtXor_R	mandatory	3	any 3 channels
3SinDtXor_T	mandatory	9	any 9 channels
Cosine table	mandatory	4	any 4 consecutive channels
3SinDtXor_sync	optional	1	one of Cosine Table channels
3SinDtXor_res	optional	1	one of Cosine Table channels
3SinDtXor_fault	optional	1	one of Cosine Table channels, recommended is 15 and DTPU bit set

Table 2 shows an example of configuration.

Table 2. Example of configuration

Channel	TPU function	Priority
0	3SinDtXor_T	middle
1	3SinDtXor_T	middle
2	3SinDtXor_R	middle
3	3SinDtXor_T	middle
4	3SinDtXor_T	middle
5	3SinDtXor_T	middle
6	3SinDtXor_R	middle
7	3SinDtXor_T	middle
8	3SinDtXor_T	middle
9	3SinDtXor_T	middle
10	3SinDtXor_R	middle
11	3SinDtXor_T	middle
12	Cosine table 1	none
13	3SinDtXor_sync, Cosine table 2	low
14	3SinDtXor_res, Cosine table 3	low
15	3SinDtXor_fault, Cosine table 4	high

Table 3 shows the TPU function code sizes.

Table 3. TPU function code sizes

TPU function	Code size
3SinDtXor_R	317 μ instructions + 8 entries = 325 long words
3SinDtXor_T	3 μ instructions + 8 entries = 11 long words
3SinDtXor_sync	30 μ instructions + 8 entries = 38 long words
3SinDtXor_res	41 μ instructions + 8 entries = 49 long words
3SinDtXor_fault	9 μ instructions + 8 entries = 17 long words

Configuration Order

The CPU configures the TPU as follows.

1. Disables the channels by clearing the two channel priority bits on each channel used (not necessary after reset).
2. Selects the channel functions on all used channels by writing the function numbers to the channel function select bits.
3. Initializes function parameters. The parameters *T*, *prescaler*, *DT*, *Theta_H*, *Theta_L* and *sync_presc_addr* must be set before initialization. 32 cosine table values must be set. If an 3SinDt_sync channel or an 3SinDt_res channel is used, then its parameters must also be set before initialization.

4. Issues an HSR (Host Service Request) type %10 to one of the 3SinDtXor_R channels to initialize all 3SinDtXor_R and 3SinDtXor_T channels. Issues an HSR type %10 to the 3SinDtXor_sync channel, 3SinDtXor_res channel and 3SinDtXor_fault channel, if used.
5. Enables servicing by assigning a high, middle or low priority to the channel priority bits. All 3SinDtXor_R and 3SinDtXor_T channels must be assigned the same priority to ensure correct operation. The CPU must ensure that the 3SinDtXor_sync or 3SinDtXor_res channels are initialized after the initialization of 3SinDtXor_R and 3SinDtXor_T channels:
 - assign a priority to the 3SinDtXor_R and 3SinDtXor_T channels to enable their initialization
 - if a Synchronization Signal or a Resolver Reference Signal channel is used, wait until the HSR bits are cleared to indicate that initialization of the 3SinDtXor_R and 3SinDtXor_T channels has completed and
 - assign a priority to the 3SinDtXor_sync or 3SinDtXor_res channels to enable their initialization

NOTE: A CPU routine that configures the TPU can be generated automatically using the MPC500_Quick_Start Graphical Configuration Tool.

Detailed Function Description

3-Phase Sine Wave Generator with Dead-Time Correction – XOR version – R channels (3SinDtXor_R) and 3-Phase Sine Wave Generator with Dead-Time Correction – XOR version – T channels (3SinDtXor_T)

The 3SinDtXor_R and 3SinDtXor_T TPU functions work together to generate 6 pairs of XOR gate inputs. The XOR gate outputs then produce a 6-channel 3-phase center-aligned PWM signal with dead-time between the top and bottom channels. In order to charge the bootstrap transistors, the PWM signals start to run 1.6ms after their initialization (at 20MHz TCR1 clock). The functions generate signals corresponding to Reference Voltage Vector Amplitude of 0 (50% duty-cycle) until the first reloaded values are processed.

The CPU controls the PWM output by setting the TPU parameters. The Stator Reference Voltage Vector Amplitude *Ampl*, the Stator Reference Voltage Vector angle *Theta* (32-bit) and the angle increment *dTheta* (32-bit), can be adjusted during run time. The PWM period *T* and the *prescaler* – the number of PWM periods per reload of new values – are also read at each reload, so these parameters can be changed during run time. Conversely, the dead-time (*DT*) is not supposed to be changed during run time. The phase currents *currentA*, *currentB* and *currentC* are read by the TPU asynchronously to the PWM parameters reload. They are read in the last part of the edge-time calculation to reflect the latest state of the phase currents. The CPU notifies the TPU that the new reload values are prepared by setting the LD_OK parameter. The TPU

notifies the CPU that the reload values have been read and new values can be written by clearing the LD_OK parameter.

The TPU function rotates the Stator Reference Voltage Vector by $dTheta$ angle each period so that the TPU can drive the motor with constant amplitude and constant speed independently of the CPU. The CPU can adjust the *Ampl* parameter to change the Stator Reference Voltage Vector amplitude, the *dTheta* parameter to change the rotation speed. The CPU can also set the absolute value of Stator Reference Voltage Vector angle *Theta*. To notify the TPU that the *Theta* parameter should be loaded instead of using the buffered value, the CPU must set $LD_OK = \$8001$ instead of $\$0001$.

The following equations describe how the 3-phase sine wave PWM signal high-times ht_A , ht_B , ht_C and transition times t_{trans} of each channel are calculated:

$$Theta = Theta + dTheta$$

$$s_A = \cos(Theta)$$

$$s_B = \cos(Theta - 120^\circ)$$

$$s_C = -(s_A + s_B)$$

The function **cos** is calculated using a table of 32 values from the first quadrant of one cosine wave period. The function parameter is mirrored in the first quadrant. The function value is obtained by linear interpolation between the two closest table values. **Figure 3** shows the error of the cosine function value calculation. The maximum error is 7 in the amplitude range $\langle -32768, 32767 \rangle$, that is 0.021%.

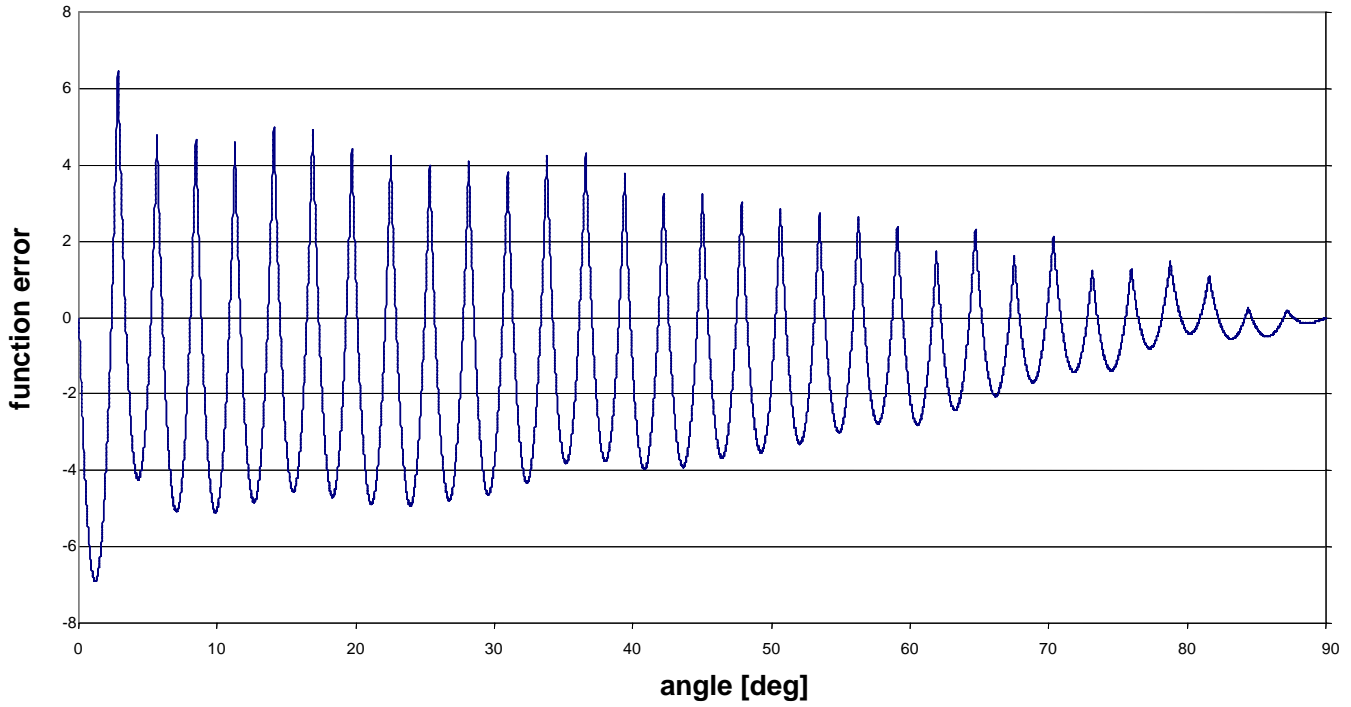
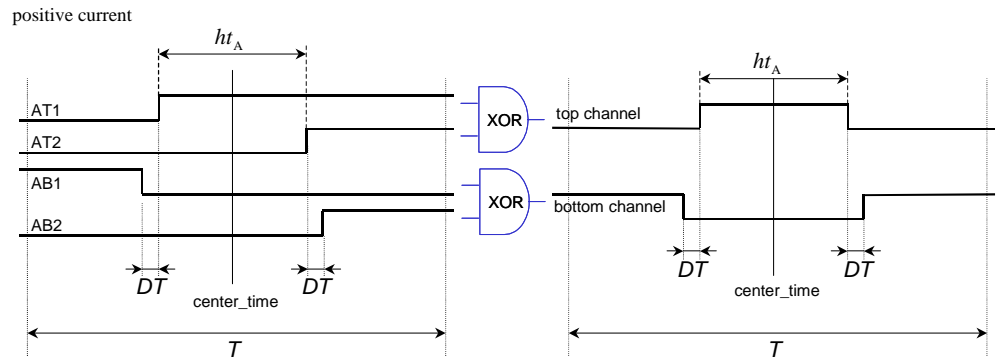


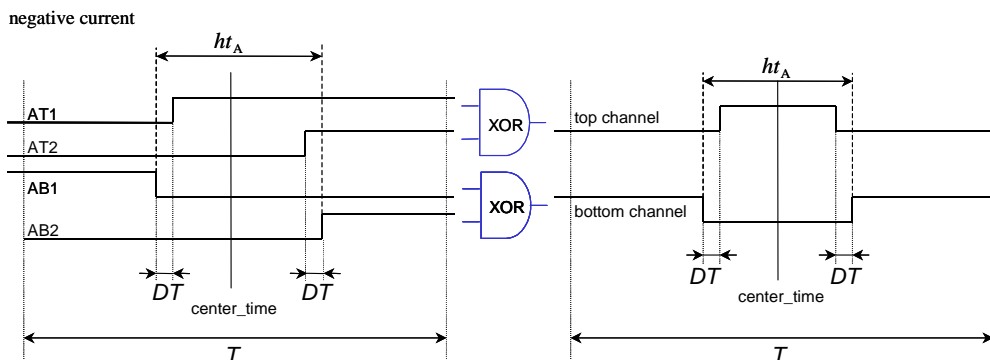
Figure 3. Cosine function value error

$$ht_A = T \cdot \frac{Ampl \cdot s_A + 1}{2}$$

$$ht_B = T \cdot \frac{Ampl \cdot s_B + 1}{2}$$

$$ht_C = T \cdot \frac{Ampl \cdot s_C + 1}{2}$$





Phase A:

Positive current

– T1 channel

$$t_{trans} = center_time - \frac{ht_A}{2}$$

– T2 channel

$$t_{trans} = center_time + \frac{ht_A}{2}$$

– B1 channel

$$t_{trans} = center_time - \frac{ht_A}{2} - DT$$

– B2 channel

$$t_{trans} = center_time + \frac{ht_A}{2} + DT$$

Negative current

– T1 channel

$$t_{trans} = center_time - \frac{ht_A}{2} + DT$$

– T2 channel

$$t_{trans} = center_time + \frac{ht_A}{2} - DT$$

– B1 channel

$$t_{trans} = center_time - \frac{ht_A}{2}$$

– B2 channel

$$t_{trans} = center_time + \frac{ht_A}{2}$$

Phase B and Phase C similarly with ht_B and ht_C substituted to ht_A .

Host Interface

<input type="checkbox"/>	Written By CPU	<input type="checkbox"/>	Written by both CPU and TPU
<input type="checkbox"/>	Written By TPU	<input type="checkbox"/>	Not Used

Table 4. 3SinDtXor_T Control Bits

Name	Options
<div style="display: flex; justify-content: space-around; width: 100px;"> 3210 </div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 20px; margin-right: 5px;"></div> <div style="border: 1px solid black; width: 20px; height: 20px; margin-right: 5px;"></div> <div style="border: 1px solid black; width: 20px; height: 20px; margin-right: 5px;"></div> <div style="border: 1px solid black; width: 20px; height: 20px;"></div> Channel Function Select </div>	3SinDtXor_T function number (Assigned during assembly the DPTRAM code from library TPU functions)
<div style="display: flex; justify-content: space-around; width: 100px;"> 10 </div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 20px; margin-right: 5px;"></div> <div style="border: 1px solid black; width: 20px; height: 20px;"></div> Channel Priority </div>	00 – Channel Disabled 01 – Low Priority 10 – Middle Priority 11 – High Priority

Table 4. 3SinDtXor_T Control Bits

Name	Options
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="display: flex; justify-content: space-around; width: 40px;"> 10 </div> <div style="display: flex; justify-content: space-around; width: 40px;"> <div style="width: 15px; height: 15px; border: 1px solid black;"></div> <div style="width: 15px; height: 15px; border: 1px solid black;"></div> </div> </div> <div>Host Service Bits (HSR)</div> </div>	00 – No Host Service Request 01 – Not used 10 – Not used 11 – Not used
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="display: flex; justify-content: space-around; width: 40px;"> 10 </div> <div style="display: flex; justify-content: space-around; width: 40px;"> <div style="width: 15px; height: 15px; background-color: green; border: 1px solid black;"></div> <div style="width: 15px; height: 15px; background-color: green; border: 1px solid black;"></div> </div> </div> <div>Host Sequence Bits (HSQ)</div> </div>	xx – Not used
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="display: flex; justify-content: space-around; width: 40px;"> 0 </div> <div style="display: flex; justify-content: space-around; width: 40px;"> <div style="width: 15px; height: 15px; background-color: green; border: 1px solid black;"></div> </div> </div> <div>Channel Interrupt Enable</div> </div>	x – Not used
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="display: flex; justify-content: space-around; width: 40px;"> 0 </div> <div style="display: flex; justify-content: space-around; width: 40px;"> <div style="width: 15px; height: 15px; background-color: green; border: 1px solid black;"></div> </div> </div> <div>Channel Interrupt Status</div> </div>	x – Not used

Table 5. 3SinDtXor_R Control Bits

Name	Options
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="display: flex; justify-content: space-around; width: 60px;"> 3210 </div> <div style="display: flex; justify-content: space-around; width: 60px;"> <div style="width: 15px; height: 15px; border: 1px solid black;"></div> <div style="width: 15px; height: 15px; border: 1px solid black;"></div> <div style="width: 15px; height: 15px; border: 1px solid black;"></div> <div style="width: 15px; height: 15px; border: 1px solid black;"></div> </div> </div> <div>Channel Function Select</div> </div>	3SinDtXor_R function number (Assigned during assembly the DPTRAM code from library TPU functions)
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="display: flex; justify-content: space-around; width: 40px;"> 10 </div> <div style="display: flex; justify-content: space-around; width: 40px;"> <div style="width: 15px; height: 15px; border: 1px solid black;"></div> <div style="width: 15px; height: 15px; border: 1px solid black;"></div> </div> </div> <div>Channel Priority</div> </div>	00 – Channel Disabled 01 – Low Priority 10 – Middle Priority 11 – High Priority
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="display: flex; justify-content: space-around; width: 40px;"> 10 </div> <div style="display: flex; justify-content: space-around; width: 40px;"> <div style="width: 15px; height: 15px; border: 1px solid black;"></div> <div style="width: 15px; height: 15px; border: 1px solid black;"></div> </div> </div> <div>Host Service Bits (HSR)</div> </div>	00 – No Host Service Request 01 – Not used 10 – Initialization 11 – Stop
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="display: flex; justify-content: space-around; width: 40px;"> 10 </div> <div style="display: flex; justify-content: space-around; width: 40px;"> <div style="width: 15px; height: 15px; background-color: green; border: 1px solid black;"></div> <div style="width: 15px; height: 15px; background-color: green; border: 1px solid black;"></div> </div> </div> <div>Host Sequence Bits (HSQ)</div> </div>	xx – Not used
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="display: flex; justify-content: space-around; width: 40px;"> 0 </div> <div style="display: flex; justify-content: space-around; width: 40px;"> <div style="width: 15px; height: 15px; border: 1px solid black;"></div> </div> </div> <div>Channel Interrupt Enable</div> </div>	0 – Channel Interrupt Disabled 1 – Channel Interrupt Enabled
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="display: flex; justify-content: space-around; width: 40px;"> 0 </div> <div style="display: flex; justify-content: space-around; width: 40px;"> <div style="width: 15px; height: 15px; background-color: blue; border: 1px solid black;"></div> </div> </div> <div>Channel Interrupt Status</div> </div>	0 – Interrupt Not Asserted 1 – Interrupt Asserted

TPU function 3SinDtXor_R generates an interrupt when the current values of *Ampl*, *dTheta* (optionally also *Theta*), *T* and *prescaler* have been read by the TPU and indicates to the CPU that it can write new variables. The CPU program can

either wait for this interrupt to occur, or poll the *LD_OK* parameter to check it has cleared. The interrupt is generated at each reload by one of the R channels. The T channels do not generate any interrupts.

Table 6. 3SinDtXor_T and 3SinDtXor_R Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Phase A T1 channel	0	Ttime_AT1																
	1	state																
	2	[Redacted]																
	3	prsc_copy																
	4	LD_OK																
	5	prescaler																
	6	[Redacted]																
	7	fault_pinstat																
Phase A T2 channel	0	Ttime_AT2																
	1	[Redacted]																
	2	max_ht																
	3	dec																
	4	Theta_H																
	5	Theta_L																
	6	[Redacted]																
	7	[Redacted]																
Phase A B1 channel	0	htA																
	1	B2_chan_A																
	2	T1_chan_A																
	3	T2_chan_A																
	4	B1a_chan_A																
	5	B1b_chan_A																
	6	currentA																
	7	[Redacted]																
Phase A B2 channel	0	Ttime_AB2																
	1	[Redacted]																
	2	center_time																
	3	TA_buf																
	4	Theta_buf_H																
	5	Theta_buf_L																
	6	[Redacted]																
	7	[Redacted]																

Table 6. 3SinDtXor_T and 3SinDtXor_R Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Phase B T1 channel	0	Ttime_BT1																
	1																	
	2	F_chan																
	3	T_copy																
	4	Ampl																
	5	T																
	6																	
	7																	
Phase B T2 channel	0	Ttime_BT2																
	1																	
	2	dTheta_buf_H																
	3	dTheta_buf_L																
	4	dTheta_H																
	5	dTheta_L																
	6																	
	7																	
Phase B B1 channel	0	htB																
	1	B2_chan_B																
	2	T1_chan_B																
	3	T2_chan_B																
	4	B1a_chan_B																
	5	B1b_chan_B																
	6	currentB																
	7																	
Phase B B2 channel	0	Ttime_BB2																
	1																	
	2																	
	3																	
	4	CPU14																
	5																	
	6																	
	7																	
Phase C T1 channel	0	Ttime_CT1																
	1																	
	2																	
	3																	
	4	DT																
	5																	
	6																	
	7																	

Table 6. 3SinDtXor_T and 3SinDtXor_R Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Phase C T2 channel	0	Ttime_CT2																
	1	move_res																
	2																	
	3	presc_addr_res																
	4	prescaler_res																
	5	time_res																
	6	dec_res																
	7	T_copy_res																
Phase C B1 channel	0	htC																
	1	B2_chan_C																
	2	T1_chan_C																
	3	T2_chan_C																
	4	B1a_chan_C																
	5	B1b_chan_C																
	6	currentC																
	7																	
Phase C B2 channel	0	Ttime_CB2																
	1	move_sync																
	2	pw_sync																
	3	prescaler_sync																
	4	presc_copy_sync																
	5	time_sync																
	6	dec_sync																
	7	T_copy_sync																

Table 7. 3SinDtXor_T and 3SinDtXor_R parameter description

Parameter	Format	Description
Parameters written by CPU		
Ampl	16-bit fractional	Stator Reference Voltage Vector amplitude, positive values only!
currentA	0 or 1	0 ... positive current on phase A 1 ... negative current on phaseA
currentB	0 or 1	0 ... positive current on phase B 1 ... negative current on phaseB
currentC	0 or 1	0 ... positive current on phase C 1 ... negative current on phaseC

Table 7. 3SinDtXor_T and 3SinDtXor_R parameter description

Parameter	Format	Description
Theta	32-bit fractional	Stator Ref. Voltage Vector angle range $\langle -1, 1 \rangle$ corresponds to $\langle -180^\circ, 180^\circ \rangle$
dTheta	32-bit fractional	Stator Reference Voltage Vector angle increment range $\langle -1, 1 \rangle$ corresponds to $\langle -180^\circ, 180^\circ \rangle$
T	16-bit unsigned integer	PWM period in number of TCR1 TPU cycles
prescaler	16-bit unsigned integer	The number of PWM periods per reload of new values
DT	16-bit unsigned integer	Dead-time in number of TCR1 TPU cycles
CPU14	16-bit unsigned integer	Time of 14 IMB clocks in TCR1 clocks.
Parameters written by both TPU and CPU		
LD_OK	16-bit unsigned integer	0 CPU can update variables <>0 .. TPU can read variables: \$0001 ... load <i>Ampl</i> , <i>dTheta</i> , <i>T</i> and <i>prescaler</i> only \$8001 ... load also <i>Theta</i> CPU sets \$0001 or \$8001, TPU sets 0
Parameters written by TPU		
fault_pinstate	0 or 1	If fault channel is used, state of fault pin: 0 ... low 1 ... high
Theta_buf	32-bit fractional	Actual Stator Reference Voltage Vector angle range $\langle -1, 1 \rangle$ corresponds to $\langle -180^\circ, 180^\circ \rangle$
Other parameters are just for TPU function inner use.		

Performance

The maximum PWM frequency is 38kHz (PWM period $T = 525$). This can be achieved when only 3SinDtXor_R and 3SinDtXor_T run on the TPU and the IMB clock is 40MHz. When other functions run on the same TPU the minimum

PWM period T has to be greater. Get all the other running function states that can be served during one PWM period. Get their lengths (number of IMB clock cycles) and add a time slot transition of 10 IMB clock cycles to each one. Sum all the states lengths including the time slot transition. Convert the result from IMB clock cycles to TCR1 clock cycles according to TCR1 prescaler settings. The result indicates how much greater than the minimum value of 525 T has to be for that particular case.

Table 8. 3SinDtXor_T State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
ST	2	1
SF	2	0

Table 9. 3SinDtXor_R State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	154	37
STOP	166	4
SFR ₀	6	1
SFR	64	24
C7	44	15
SFC ₀	6	1
SFC ₁	58	7
SFC ₂	96	12
SFC ₃	88	7
SFC ₄	68	8
SFC ₅	68	8
SFC ₆	80	9

NOTE: Execution times do not include the time slot transition time ($TST = 10$ or 14 IMB clocks)

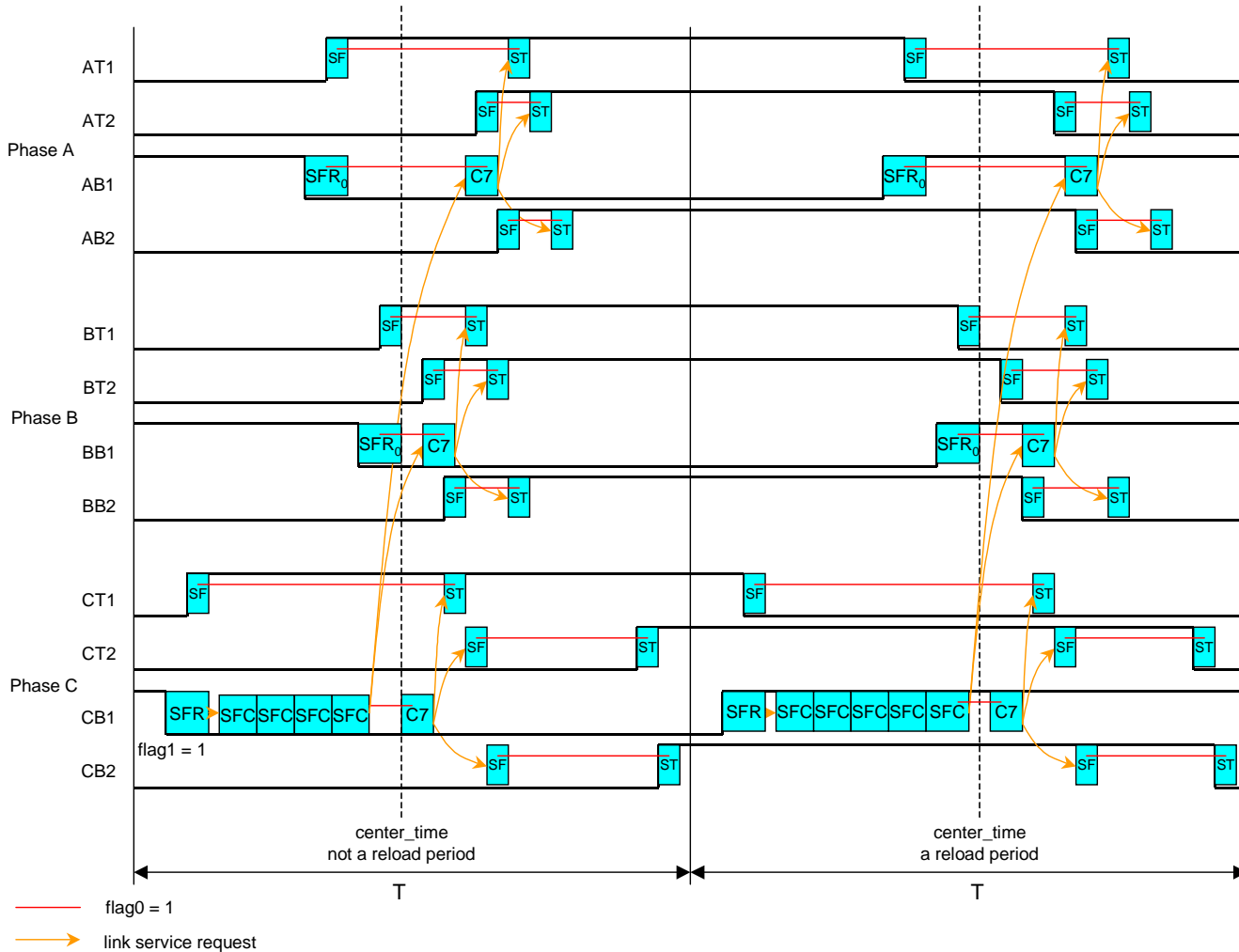


Figure 4. 3SinDtXor_T and 3SinDtXor_R timing

NOTE: The R channel with the momentary earliest transition within the PWM period is marked by a flag1 and runs the SFR and SFC states.

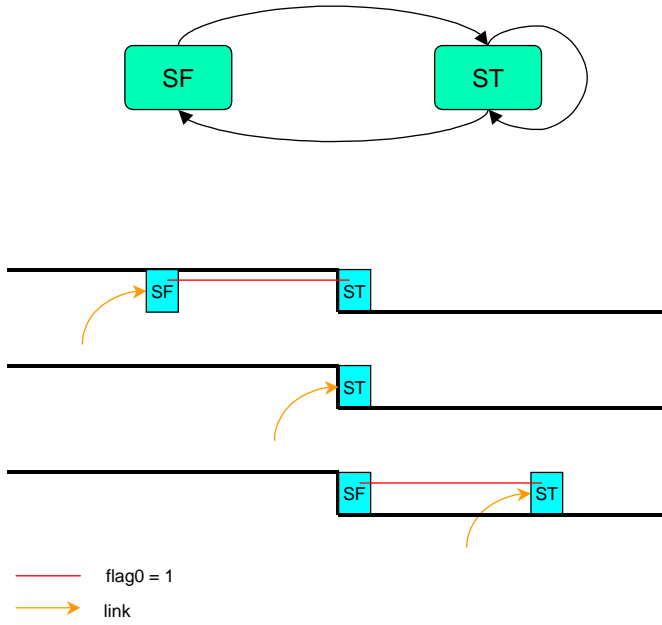


Figure 5. 3SinDtXor_T state diagram and 3 cases of timing

NOTE: The case that happens is determined by the time when the link comes.

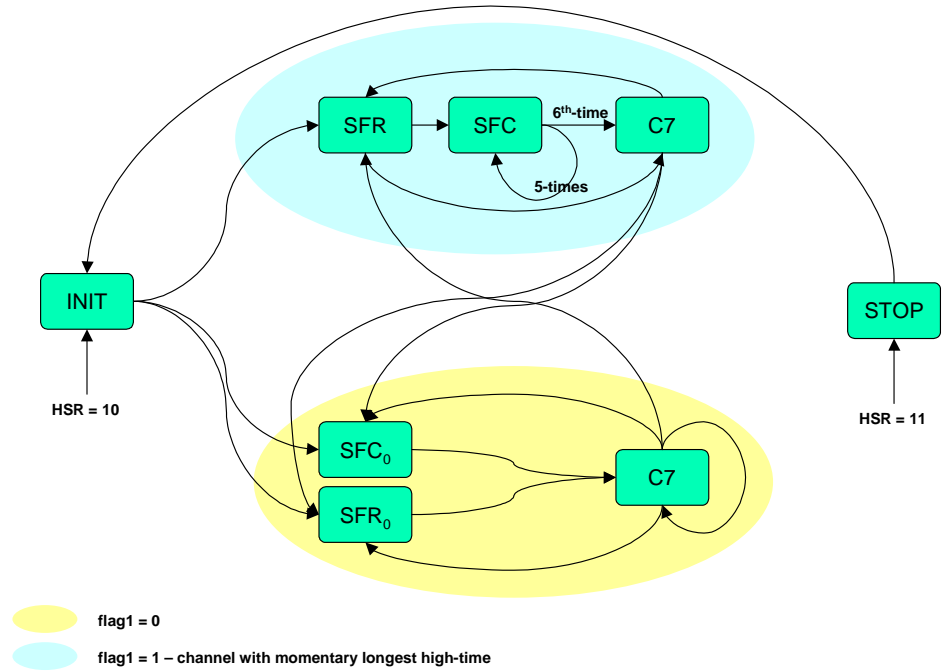


Figure 6. 3SinDtXor_R state diagram

Synchronization signal for 3-Phase Sine Wave Generator with Dead-Time Correction – XOR version (3SinDtXor_sync)

The 3SinDtXor_sync TPU function uses information obtained from 3SinDtXor_R and 3SinDtXor_T functions, the actual PWM center times and the PWM periods. This allows a signal to be generated, which tracks the changes in the PWM period and is always synchronized with the PWM. The synchronization signal is a positive pulse generated repeatedly after the *prescaler* PWM periods. The low to high transition of the pulse can be adjusted by a parameter, either negative or positive, to go a number of TCR1 TPU cycles before or after the PWM period center time. The pulse width *pw* is another synchronization signal parameter.

The 3SinDtXor_sync parameters are placed on the CB2 channel to keep the channel parameter space free, available for the table of cosine values.

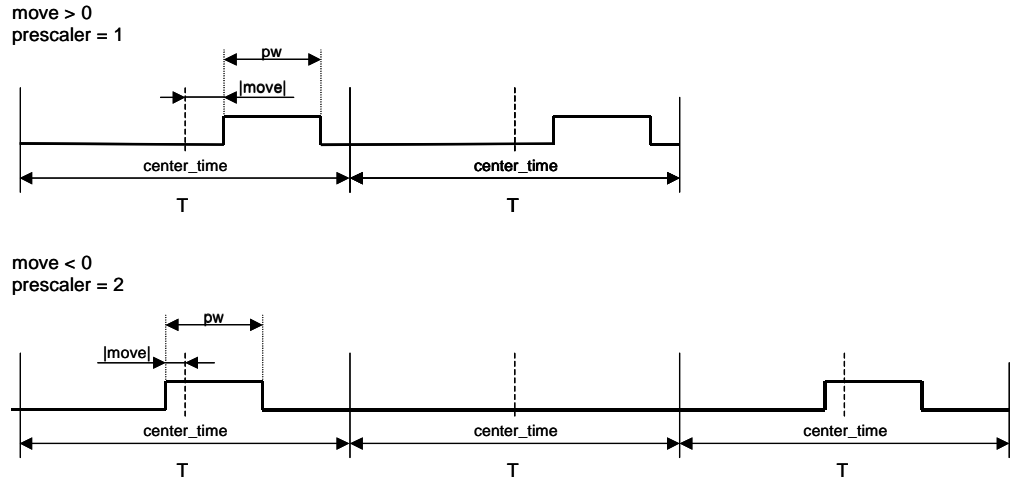


Figure 7. Synchronization signal adjustment examples

Synchronized Change of PWM Prescaler And Synchronization Signal Prescaler

The 3SinDtXor_sync TPU function actually uses the *presc_copy* parameter instead of the *prescaler* parameter. The *prescaler* parameter holds the prescaler value that is copied to the *presc_copy* by the 3SinDtXor_R function at the time the PWM parameters are reloaded. This ensures that new prescaler values for the PWM signals, as well as the synchronization signal, are applied at the same time.

Host Interface

- Written By CPU
- Written by both CPU and TPU
- Written By TPU
- Not Used

Table 10. 3SinDtXor_sync Control Bits

Name	Options
<div style="display: flex; justify-content: space-around; width: 100px;"> 3210 </div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 20px; margin-right: 5px;"></div> <div style="border: 1px solid black; width: 20px; height: 20px; margin-right: 5px;"></div> <div style="border: 1px solid black; width: 20px; height: 20px; margin-right: 5px;"></div> <div style="border: 1px solid black; width: 20px; height: 20px;"></div> </div> Channel Function Select	3SinDtXor_sync function number (Assigned during assembly the DPTRAM code from library TPU functions)
<div style="display: flex; justify-content: space-around; width: 100px;"> 10 </div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 20px; margin-right: 5px;"></div> <div style="border: 1px solid black; width: 20px; height: 20px;"></div> </div> Channel Priority	00 – Channel Disabled 01 – Low Priority 10 – Middle Priority 11 – High Priority
<div style="display: flex; justify-content: space-around; width: 100px;"> 10 </div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 20px; margin-right: 5px;"></div> <div style="border: 1px solid black; width: 20px; height: 20px;"></div> </div> Host Service Bits (HSR)	00 – No Host Service Request 01 – Not used 10 – Initialization 11 – Not used

Table 10. 3SinDtXor_sync Control Bits

Name	Options
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="display: flex; justify-content: space-around; width: 20px;"> 10 </div> <div style="display: flex; justify-content: space-around; width: 20px;"> <div style="width: 10px; height: 10px; background-color: green; border: 1px solid black;"></div> <div style="width: 10px; height: 10px; background-color: green; border: 1px solid black;"></div> </div> </div> <div>Host Sequence Bits (HSQ)</div> </div>	xx – Not used
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="display: flex; justify-content: center; width: 20px;"> 0 </div> <div style="width: 15px; height: 15px; border: 1px solid black; margin: 0 auto;"></div> </div> <div>Channel Interrupt Enable</div> </div>	0 – Channel Interrupt Disabled 1 – Channel Interrupt Enabled
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="display: flex; justify-content: center; width: 20px;"> 0 </div> <div style="width: 15px; height: 15px; background-color: blue; border: 1px solid black; margin: 0 auto;"></div> </div> <div>Channel Interrupt Status</div> </div>	0 – Interrupt Not Asserted 1 – Interrupt Asserted

TPU function 3SinDtXor_sync generates an interrupt after each low to high transition.

Table 11. 3SinDtXor_sync Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Synchronization channel	0																
	1																
	2																
	3																
	4																
	5																
	6																
	7																

Table 12. 3SinDtXor_sync parameter description

Parameter	Format	Description
Parameters written by CPU		
move	16-bit signed integer	The number of TCR1 TPU cycles to forego (negative) or come after (positive) the PWM period center time
pw	16-bit unsigned integer	Synchronization pulse width in number of TCR1 TPU cycles.
prescaler	16-bit unsigned integer	The number of PWM periods per synchronization pulse
Parameters written by TPU		

Table 12. 3SinDtXor_sync parameter description

Parameter	Format	Description
Other parameters are just for TPU function inner use.		

Performance

There is one limitation. The absolute value of parameter *move* has to be less than a quarter of the PWM period *T*.

$$|move| < \frac{T}{4}$$

Table 13. 3SinDtXor_sync State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	14	5
S1	14	6
S2	10	3
S3	18	7

NOTE: Execution times do not include the time slot transition time ($TST = 10$ or 14 IMB clocks)

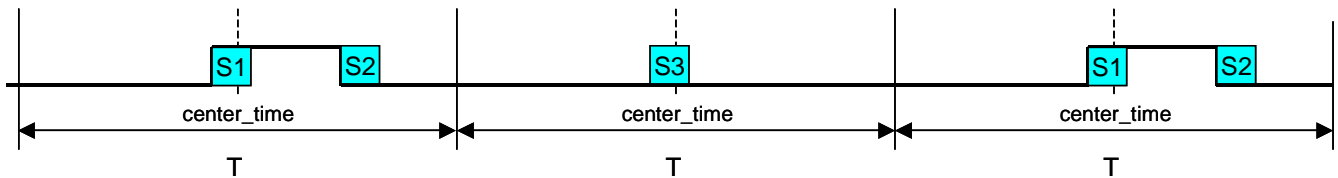


Figure 8. 3SinDtXor_sync timing

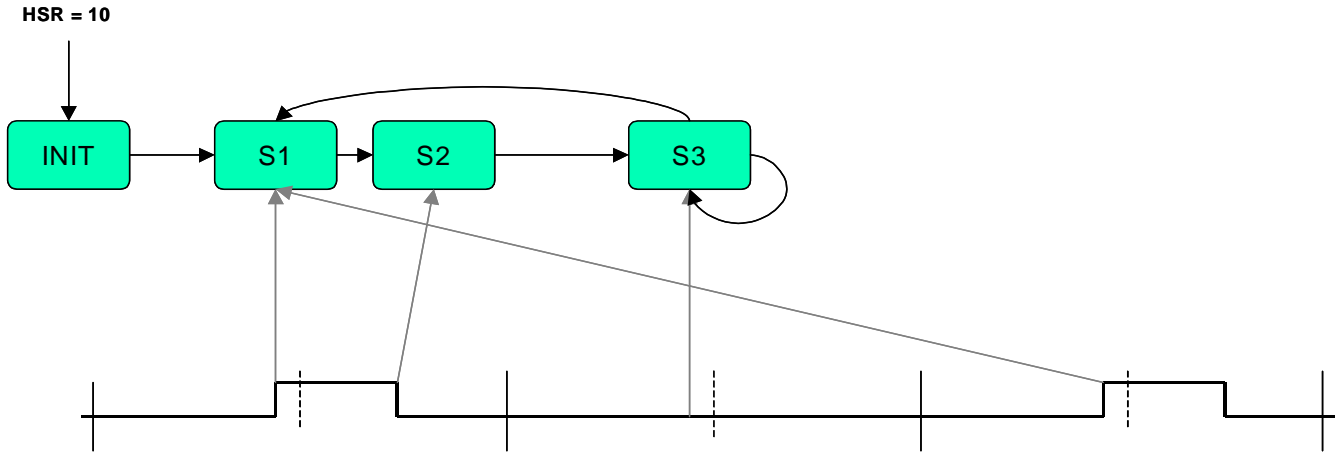


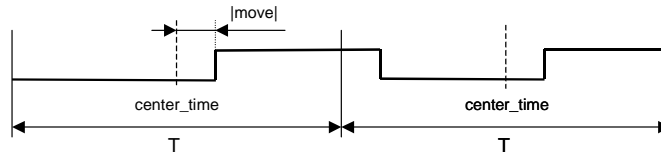
Figure 9. 3SinDtXor_sync state diagram

Resolver Reference Signal for 3-Phase Sine Wave Generator with Dead-Time Correction – XOR version (3SinDtXor_res)

The 3SinDtXor_res TPU function uses information read from the 3SinDtXor_R and 3SinDtXor_T functions, the actual PWM center times and the PWM periods. This allows a signal to be generated, which tracks the changes of the PWM period and is always synchronized with the PWM. The resolver reference signal is a 50% duty-cycle signal with a period equal to *prescaler* or synchronization channel *presc_copy* PWM periods (see next paragraph). The low to high transition of the pulse can be adjusted by a parameter, either negative or positive, to go a number of TCR1 TPU cycles before or after the PWM period center time.

The 3SinDtXor_res parameters are placed on the CT2 channel to keep the channel parameter space free, available for the table of cosine values.

move > 0
prescaler = 1



move < 0
prescaler = 2

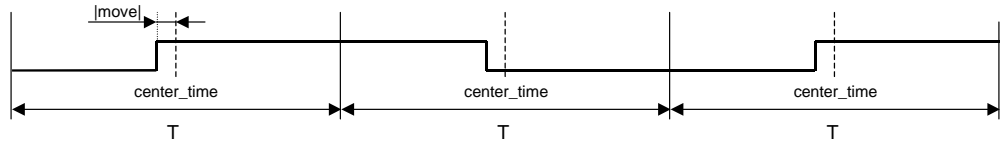


Figure 10. Resolver reference signal adjustment examples

Synchronized Change of PWM Prescaler And Resolver Reference Signal Prescaler

The 3SinDtXor_res TPU function can inherit the Synchronization Signal prescaler that is synchronously changed with the PWM prescaler. Write the synchronization signals *presc_copy* parameter address to the *presc_addr* parameter to enable this mechanism. Write 0 to disable it, and in this case set the *prescaler* parameter to directly specify prescaler value.

Host Interface

- Written By CPU
- Written by both CPU and TPU
- Written By TPU
- Not Used

Table 14. 3SinDtXor_res Control Bits

Name	Options
<div style="display: flex; justify-content: space-around; width: 100px;"> 3210 </div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 20px; margin-right: 5px;"></div> <div style="border: 1px solid black; width: 20px; height: 20px; margin-right: 5px;"></div> <div style="border: 1px solid black; width: 20px; height: 20px; margin-right: 5px;"></div> <div style="border: 1px solid black; width: 20px; height: 20px;"></div> </div> Channel Function Select	3SinDtXor_res function number (Assigned during assembly the DPTRAM code from library TPU functions)
<div style="display: flex; justify-content: space-around; width: 100px;"> 10 </div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 20px; margin-right: 5px;"></div> <div style="border: 1px solid black; width: 20px; height: 20px;"></div> </div> Channel Priority	00 – Channel Disabled 01 – Low Priority 10 – Middle Priority 11 – High Priority
<div style="display: flex; justify-content: space-around; width: 100px;"> 10 </div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 20px; margin-right: 5px;"></div> <div style="border: 1px solid black; width: 20px; height: 20px;"></div> </div> Host Service Bits (HSR)	00 – No Host Service Request 01 – Not used 10 – Initialization 11 – Not used

Table 14. 3SinDtXor_res Control Bits

Name	Options
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="display: flex; justify-content: space-around; width: 20px;"> 10 </div> <div style="display: flex; justify-content: space-around; width: 20px;"> <div style="width: 10px; height: 10px; background-color: red;"></div> <div style="width: 10px; height: 10px; background-color: red;"></div> </div> </div> <div> Host Sequence Bits (HSQ) </div> <div style="margin-left: 20px;"> xx – Not used </div> </div>	
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="width: 10px; height: 10px; background-color: red;"></div> </div> <div> Channel Interrupt Enable </div> <div style="margin-left: 20px;"> x – Not used </div> </div>	
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <div style="width: 10px; height: 10px; background-color: red;"></div> </div> <div> Channel Interrupt Status </div> <div style="margin-left: 20px;"> x – Not used </div> </div>	

Table 15. 3SinDtXor_res Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resolver	0																
	1																
	2																
	3																
	4																
	5																
	6																
	7																

Table 16. 3SinDtXor_res parameter description

Parameter	Format	Description
Parameters written by CPU		
move	16-bit signed integer	The number of TCR1 TPU cycles to forego (negative) or come after (positive) the PWM period center time
presc_addr	16-bit unsigned integer	\$00X8, where X is a number of CB2 channel, to inherit Sync. channel prescaler or \$0000 to enable direct specification of prescaler value in prescaler parameter

Table 16. 3SinDtXor_res parameter description

Parameter	Format	Description
prescaler	1, 2, 4, 6, 8, 10, 12, 14, ...	The number of PWM periods per synchronization pulse – use when presc_addr = 0
Parameters written by TPU		
Other parameters are just for TPU function inner use.		

Performance

There is one limitation. The absolute value of parameter *move* has to be less than a quarter of the PWM period *T*.

$$|move| < \frac{T}{4}$$

Table 17. 3SinDtXor_res State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	14	5
S1	28	9
S3	20	7

NOTE: Execution times do not include the time slot transition time ($TST = 10$ or 14 IMB clocks)

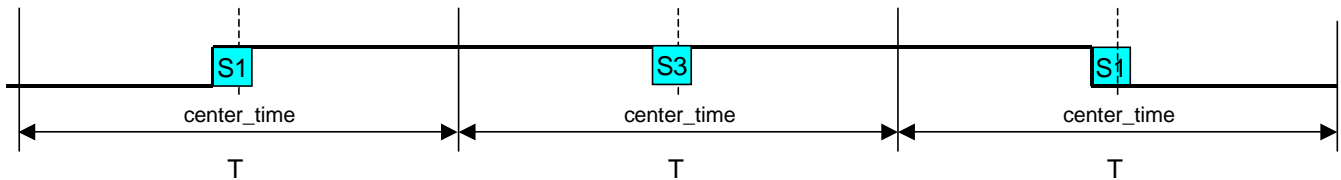


Figure 11. 3SinDtXor_res timing

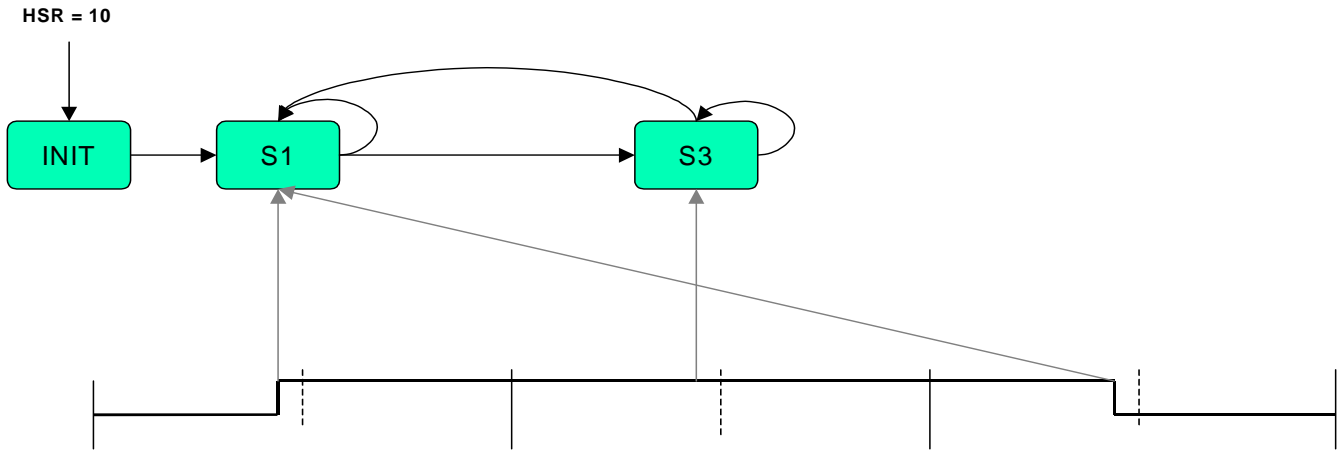


Figure 12. 3SinDtXor_res state diagram

Fault Input for 3-Phase Sine Wave Generator with Dead-Time Correction – XOR version (3SinDtXor_fault)

The 3SinDtXor_fault is an input TPU function that monitors the pin, and if a high to low transition occurs, immediately sets all PWM channels low and cancels all further transitions on them. The PWM channels, as well as the synchronization and resolver reference signal channels (if used), have to be initialized again to start them running.

The function returns the actual pinstate as a value of 0 (low) or 1 (high) in the parameter *fault_pinstate*. The parameter is placed on the AT1 channel to keep the fault channel parameter space free.

Host Interface

- Written By CPU
- Written by both CPU and TPU
- Written By TPU
- Not Used

Table 18. 3SinDtXor_fault Control Bits

Name		Options
3 2 1 0	Channel Function Select	3SinDtXor_fault function number (Assigned during assembly the DPTRAM code from library TPU functions)
□ □ □ □		
1 0	Channel Priority	00 – Channel Disabled 01 – Low Priority 10 – Middle Priority 11 – High Priority
□ □		

Table 18. 3SinDtXor_fault Control Bits

Name		Options
<div style="display: flex; justify-content: space-around;"> 1 0 </div> <div style="display: flex; justify-content: space-around;"> <input type="checkbox"/> <input type="checkbox"/> </div>	Host Service Bits (HSR)	00 – No Host Service Request 01 – Not used 10 – Initialization 11 – Not used
<div style="display: flex; justify-content: space-around;"> 1 0 </div> <div style="display: flex; justify-content: space-around;"> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> </div>	Host Sequence Bits (HSQ)	xx – Not used
<div style="display: flex; justify-content: space-around;"> 0 </div> <div style="display: flex; justify-content: space-around;"> <input type="checkbox"/> </div>	Channel Interrupt Enable	0 – Channel Interrupt Disabled 1 – Channel Interrupt Enabled
<div style="display: flex; justify-content: space-around;"> 0 </div> <div style="display: flex; justify-content: space-around;"> <input checked="" type="checkbox"/> </div>	Channel Interrupt Status	0 – Interrupt Not Asserted 1 – Interrupt Asserted

TPU function 3SinDtXor_fault generates an interrupt when a high to low transition appears.

Table 19. 3SinDtXor_fault Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Fault input	0																
	1																
	2																
	3																
	4																
	5																
	6																
	7																

Table 20. 3SinDtXor_fault parameter description

Parameter	Format	Description
Parameters written by TPU		
fault_pinstate	0 or 1	State of fault pin: 0 ... low 1 ... high

Performance

Table 21. 3SinDtXor_fault State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	8	2
FAULT	172	5
NO_FAULT	4	1

NOTE: Execution times do not include the time slot transition time (TST = 10 or 14 IMB clocks)



Figure 13. 3SinDtXor_fault timing

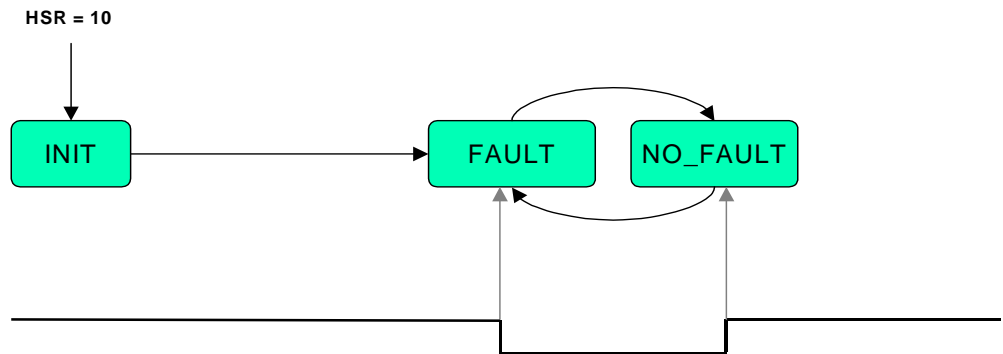


Figure 14. 3SinDtXor_fault state diagram

How to Reach Us:

Home Page:

www.freescale.com

E-mail:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor
 Technical Information Center, CH370
 1300 N. Alma School Road
 Chandler, Arizona 85224
 +1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
 Technical Information Center
 Schatzbogen 7
 81829 Muenchen, Germany
 +44 1296 380 456 (English)
 +46 8 52200080 (English)
 +49 89 92103 559 (German)
 +33 1 69 35 48 48 (French)
support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd.
 Headquarters
 ARCO Tower 15F
 1-8-1, Shimo-Meguro, Meguro-ku,
 Tokyo 153-0064
 Japan
 0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.
 Technical Information Center
 2 Dai King Street
 Tai Po Industrial Estate
 Tai Po, N.T., Hong Kong
 +800 2666 8080
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
 P.O. Box 5405
 Denver, Colorado 80217
 1-800-441-2447 or 303-675-2140
 Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document. Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

