Application Note

AN2515/D
Rev. 0, 5/2003

3-Phase Sine Wave Generator – 3 outputs version TPU Function Set (3Sin3)

By Milan Brejl, Ph.D.

Functional Overview

The 3-Phase Sine Wave Generator – 3 outputs version (3Sin3) is a variant of the 3Sin function that, in contrast to 3Sin, generates only the top channel signal of each PWM pair. The bottom channel signal can be derived from the top channel signal by external hardware. The function set consists of 4 TPU functions:

- 3-Phase Sine Wave Generator – 3 outputs version (3Sin3)
- Synchronization signal for 3-Phase Sine Wave Generator – 3 outputs version (3Sin3_sync)
- Resolver Reference Signal for 3-Phase Sine Wave Generator – 3 outputs version (3Sin3_res)
- Fault Input for 3-Phase Sine Wave Generator – 3 outputs version (3Sin3_fault)

The 3Sin3 TPU function generates a 3-channel 3-phase center-aligned PWM signal. The generated signals control external hardware, which outputs a pair of transistor signals (top and bottom) with dead-time inserted.

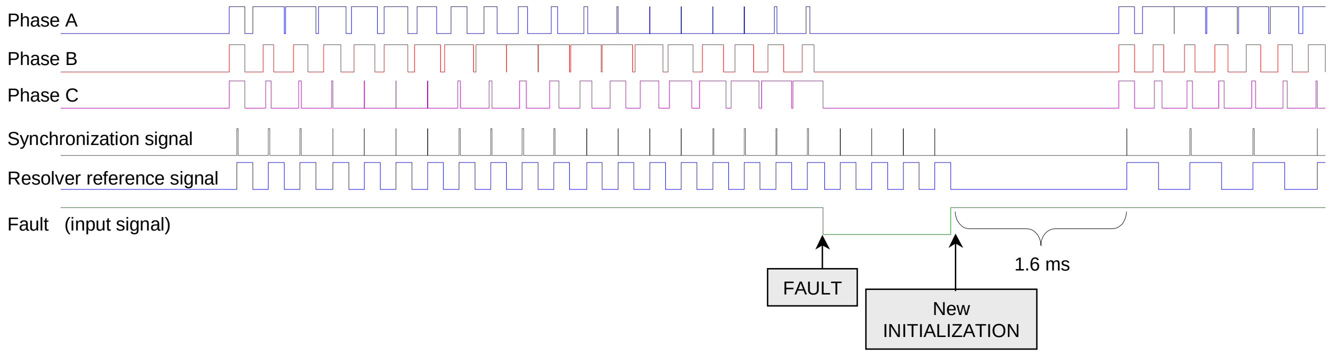


Figure 1. Signals generated by 3Sin3 TPU function set

The Synchronization Signal for the 3Sin3 function can be used to generate one or more adjustable signals for a wide range of uses, which are synchronized to the PWM, and track changes in the PWM period. The Resolver Reference Signal for the 3Sin3 function can be used to generate one or more 50% duty-cycle adjustable signals that are also synchronized to the PWM. The Fault Input for the 3Sin3 function is a TPU input function that sets all PWM outputs low when the input signal goes low. See **Figure 1**.

Function Set Configuration

The 3-Phase Sine Wave Generator – 3 outputs version TPU function is the main function of the set. It can be used either alone, with some of the supporting functions, or with all of them. The 3Sin3 function uses a table of 32 cosine function values. The table is placed in the parameter space of four consecutive channels. One or more channels running a Synchronization Signal for the 3Sin3 as well as Resolver Reference Signals for the 3Sin3 functions can be added to the 3Sin3 function. They can run with different settings on each channel. The function Fault Input for 3Sin3 can also be added to the 3Sin3 function. It is recommended to use it on channel 15, and to select 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 signals, that are disabled in this configuration. The function 3Sin3_fault can run on one of the four channels where the table of cosine function values is placed, because the 3Sin3_fault function does not have any parameters.

Table 1 shows the configuration options and restrictions.

Table 1. 3Sin3 TPU function set configuration options and restrictions

TPU function	Optional/ Mandatory	How many channels	Assignable channels
3Sin3	mandatory	3	any 3 channels
Cosine table	mandatory	4	any 4 consecutive channels
3Sin3_sync	optional	1 or more	any channels
3Sin3_res	optional	1 or more	any channels
3Sin3_fault	optional	1	any, recommended is 15 and DTPU bit set

Table 2 shows an example of configuration.

Table 2. Example of configuration

Channel	TPU function	Priority
0	3Sin3	high
1	3Sin3	high
2	3Sin3	high
10	3Sin3_sync	low
11	3Sin3_res	low
12	Cosine table 1	none
13	Cosine table 2	none
14	Cosine table 3	none
15	3Sin3_fault + Cosine table 4	high

Table 3 shows the TPU function code sizes.

Table 3. TPU function code sizes

TPU function	Code size
3Sin3	198 µ instructions + 8 entries = 206 long words
3Sin3_sync	26 µ instructions + 8 entries = 34 long words
3Sin3_res	38 µ instructions + 8 entries = 46 long words
3Sin3_fault	12 µ instructions + 8 entries = 20 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*, *MPW*, *Theta_H*, *Theta_L* and *sync_presc_addr* must be set before initialization. 32 cosine table values must be set. If an 3Sin3_sync channel or an 3Sin3_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 3Sin3 channels to initialize all PWM channels. Issues an HSR type %10 to the 3Sin3_sync channels, 3Sin3_res channels and 3Sin3_fault channel, if used.
5. Enables servicing by assigning high, middle or low priority to the channel priority bits. All PWM channels must be assigned the same priority to ensure correct operation. The CPU must ensure that the 3Sin3_sync or

3Sin3_res channels are initialized after the initialization of the PWM channels:

– assign a priority to the PWM 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 PWM channels has completed and
– assign a priority to the 3Sin3_sync or 3Sin3_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 – 3 outputs version (3Sin3)

The 3Sin3 TPU function generates a 3-channel, 3-phase PWM signal. Unlike 3Sin, the generated signals are not top-bottom pairs with dead-times but only top-like signals without dead-time. In order to charge the bootstrap transistors, the PWM signals start to run 1.6ms after their initialization (at 20MHz TCR1 clock). The function generates signals corresponding to an 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 minimum pulse width (*MPW*) is not supposed to be changed during run time. 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 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, and 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 $t_{\text{low-high}}$ and $t_{\text{high-low}}$ of each channel are calculated:

$$Theta = Theta + dTheta$$
$$s_A = \cos(Theta)$$
$$s_B = \cos(Theta - 120°)$$
$$s_C = -(s_A + s_B)$$

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 into the first quadrant. The function value is obtained by linear interpolation between two the closest table values. Figure 2 shows the error of the cosine function value calculation. The maximum error is 7 in the amplitude range <–32768, 32767>, that is 0.021%.

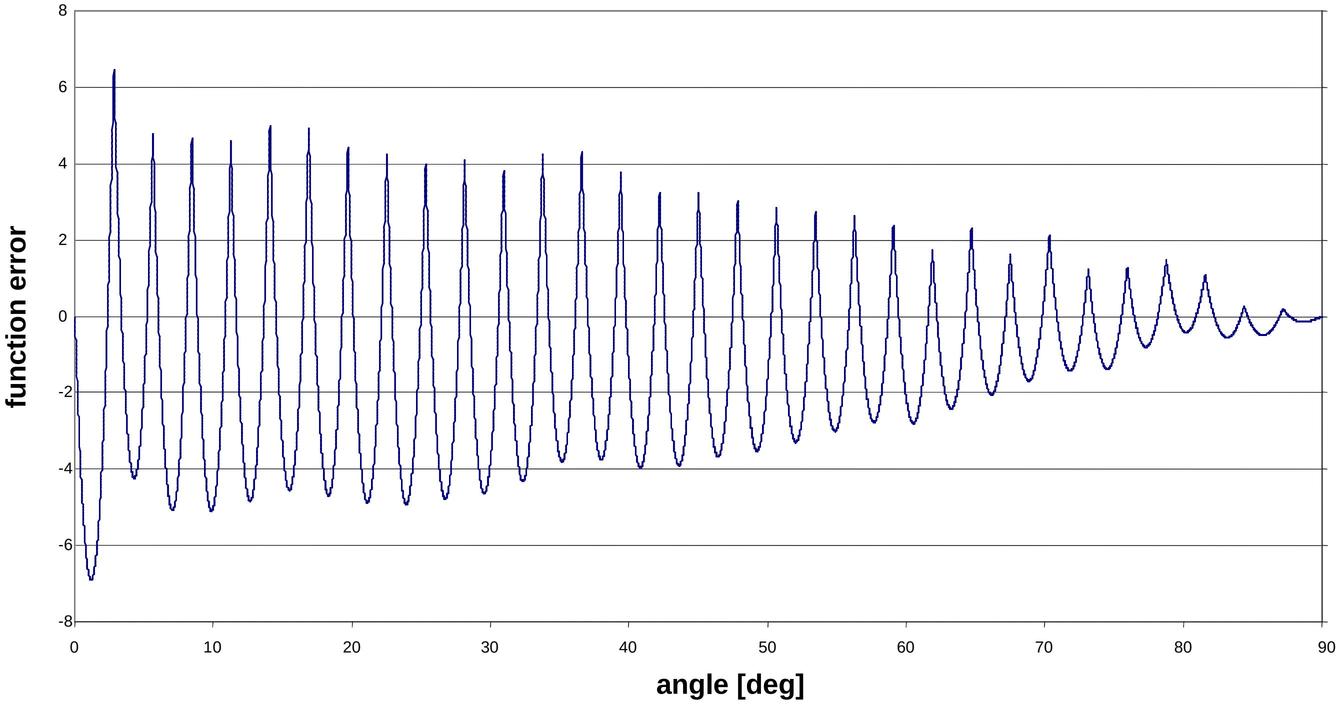
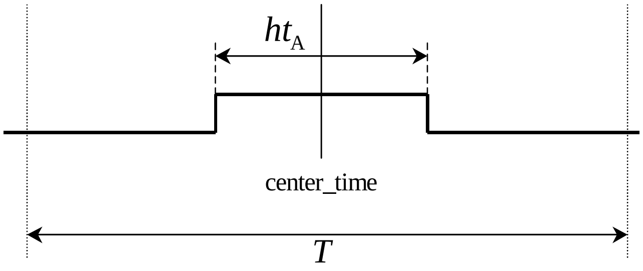


Figure 2. Cosine function value error

$$ht_{\mathrm{A}} = T \cdot \frac{Ampl \cdot s_{\mathrm{A}} + 1}{2}$$

$$ht_{\mathrm{B}} = T \cdot \frac{Ampl \cdot s_{\mathrm{B}} + 1}{2}$$

$$ht_{\mathrm{C}} = T \cdot \frac{Ampl \cdot s_{\mathrm{C}} + 1}{2}$$



Phase A:

$$t_{\text{low-high}} = \text{center\_time} - \frac{ht_{\mathrm{A}}}{2}$$

$$t_{\text{high-low}} = \text{center\_time} + \frac{ht_{\mathrm{A}}}{2}$$

Phase B and Phase C similarly with ht_{B} and ht_{C} substituted to ht_{A}.

Host Interface

□	Written By CPU	🟨	Written by both CPU and TPU
🟪	Written By TPU	🟩	Not Used

Table 4. 3Sin3 Control Bits

Name	Options
3 2 1 0 □□□□ Channel Function Select	3Sin3 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
1 0 □□ Host Service Bits (HSR)	00 – No Host Service Request 01 – Not used 10 – Initialization 11 – Stop
1 0 🟩🟩 Host Sequence Bits (HSQ)	xx – Not used
0 □ Channel Interrupt Enable	0 – Channel Interrupt Disabled 1 – Channel Interrupt Enabled
0 🟪 Channel Interrupt Status	0 – Interrupt Not Asserted 1 – Interrupt Asserted

TPU function 3Sin3 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 PWM channels.

Table 5. 3Sin3 Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Phase A	0	htA, sA															
	1	HLtime_A															
	2	T_copy															
	3	center_time															
	4	LD_OK															
	5	Ampl															
	6	T															
	7	fault_pinstate								sync_presc_addr							
Phase B	0	htB, sB															
	1	HLtime_B															
	2	dec								F_chan							
	3	TA_buf															
	4	Theta_H															
	5	Theta_L															
	6	Theta_buf_H															
	7	Theta_buf_L															
Phase C	0	htC															
	1	HLtime_C															
	2	dTheta_buf_H															
	3	dTheta_buf_L															
	4	dTheta_H															
	5	dTheta_L															
	6	MPW															
	7	prsc_copy								prescaler							

Table 6. 3Sin3 parameter description

Parameter	Format	Description
Parameters written by CPU		
Ampl	16-bit fractional	Stator Reference Voltage Vector amplitude, positive values only!
Theta	32-bit fractional	Stator Ref. Voltage Vector angle range <–1, 1) corresponds to <–180°, 180°)
dTheta	32-bit fractional	Stator Reference Voltage Vector angle increment range <–1, 1) corresponds to <–180°, 180°)
T	16-bit unsigned integer	PWM period in number of TCR1 TPU cycles
prescaler	8-bit unsigned integer	The number of PWM periods per reload of new values
MPW	16-bit unsigned integer	Minimum pulse width in number of TCR1 TPU cycles. See **Performance** for details.
sync_presc_addr	8-bit unsigned integer	address of synchronization channel *prescaler* parameter: $X4, where X is synchronization channel number. $0 if no synchronization channel is used.
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 *Ampl*, *dTheta*, *T* and *prescaler* only $8001 ... load also *Theta* 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 <–1, 1) corresponds to <–180°, 180°)
Other parameters are just for TPU function inner use.		

Performance

The maximum PWM frequency is 35.4 kHz (PWM period *T* = 565 at 20MHz TCR1 clock). This can be achieved when only 3Sin3 runs 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 enabled 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 state lengths, including the time slot transition. Convert the result from IMB clock cycles to TCR1 clock cycles according to TCR1 prescaler settings. The result then indicates how much greater than its minimum value of 565, *T* has to be for the particular case.

Table 7. 3Sin3 State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	72	21
STOP	20	0
HL	2	1
LH_C7	26	7
HL_RLD	58	22
C1	46	4
C2	84	10
C3	82	6
C4	66	7
C5	66	7
C6	66	6

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

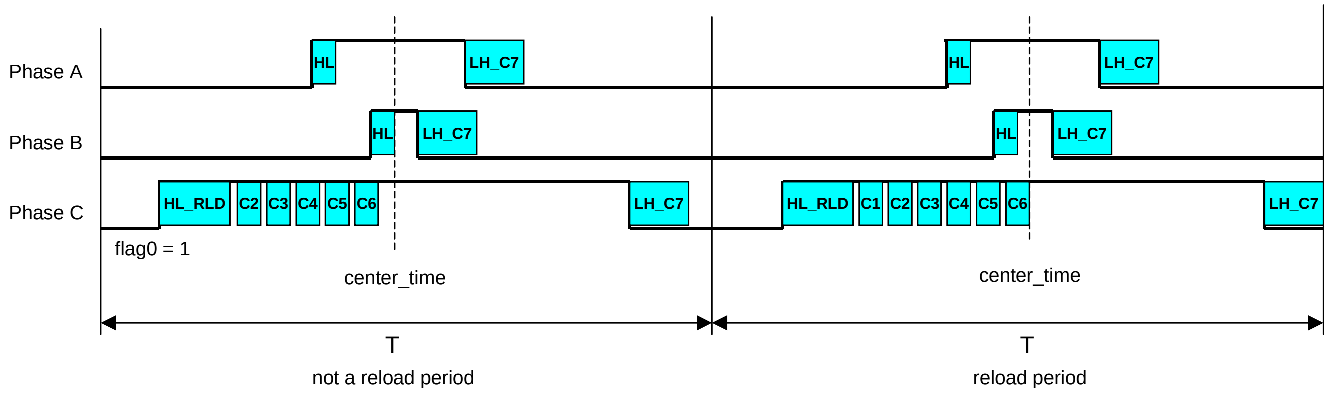


Figure 3. 3Sin3 timing

NOTE: *The bottom channel with longest momentary high-time is marked by a flag0 and runs the HL_RLD and C1, C2, C3, C4, C5, C6 states.*

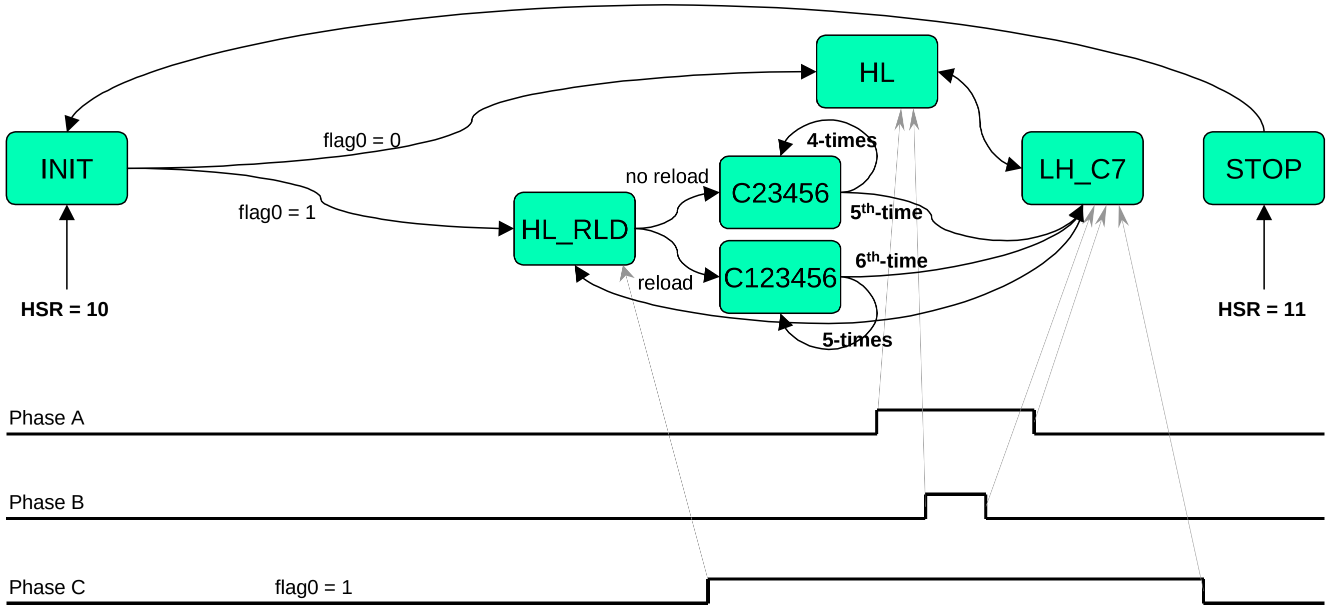


Figure 4. 3Sin3 state diagram

Minimum Pulse Width

The TPU cannot generate PWM signals with duty cycle ratios very close to 0% or 100%. The minimum pulse width that the TPU can be guaranteed to generate correctly is determined by the TPU function itself and by the activity on the other channels. When the TPU function is requested to generate a narrower pulse, a collision can occur. To prevent this, the parameter *MPW* (minimum pulse width) is introduced. The TPU function 3Sin3 limits the narrowest generated pulse widths to *MPW*. The CPU program should check, and limit, the maximum amplitude of the Stator Reference Voltage Vector. The maximum amplitude of the Stator Reference Voltage Vector should be less than

$$1 - \frac{2\,MPW}{T}$$

If this is not the case, the TPU function will start to limit the minimum pulse widths to *MPW* to prevent a collision, and the duty cycle ratio traces will be deformed as shown on **Figure 5**.

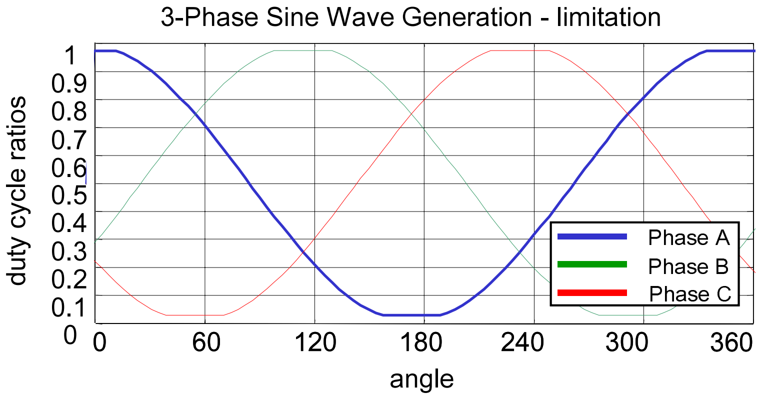


Figure 5. Effect of limitation

The *MPW* is written by the CPU. The *MPW* depends on the whole TPU unit configuration, especially the lengths of the longest states of other functions, and their priorities, running on the same TPU. The *MPW* has to be correctly calculated at the time the whole TPU unit is configured.

Figure 6. Timing of the worst case

When 3Sin3 is running alone on one TPU, the minimum pulse width can be calculated according to **Figure 6**. This illustrates the worst case timing. The high to low transition runs the LH_C7 state that sets the following low to high transition. The LH_C7 state lasts 26 IMB clock cycles (see **Table 7**). Each state is preceded by the Time Slot Transition (TST), which takes 10 or 14 IMB clock cycles. So the time necessary to set the next transition, that corresponds to *MPW*, is 40 IMB clock cycles.

Note that the *MPW* is not entered into the parameter RAM in IMB clock cycles, but in TCR1 clock cycles. It is recommended for the 3Sin3 function that the TCR1 clock is configured for its maximum speed, which is the IMB clock divided by 2. In this case the *MPW* = 20.

When other functions are running concurrently on the same TPU, the longest state of each function with its time-slot transition can increase the calculated *MPW* value. The 3Sin3_fault function does not affect the *MPW*. The 3Sin3_sync, if used, increases the *MPW* value by 22 (44 IMB clock cycles). The 3Sin3_res, if used, increases the *MPW* value by 20 (40 IMB clock cycles).

It is also possible to use the Worst-Case Latency (WCL), which is automatically calculated by the MPC500_Quick_Start Graphical Configuration Tool. It can serve as a good approximation of *MPW*. The calculated WCL is always longer than the real-case is. Let the WCL be calculated after the configuration of TPU channels and then find the longest WCL value within all 3Sin3 PWM channels. Convert the number, from IMB clock cycles to TCR1 clock cycles, to get the *MPW*.

Synchronization signal for 3-Phase Sine Wave Generator – 3 outputs version (3Sin3_sync)

The 3Sin3_sync TPU function uses information obtained from 3Sin3 PWM 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* or *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 pulse width *pw* is another synchronization signal parameter.

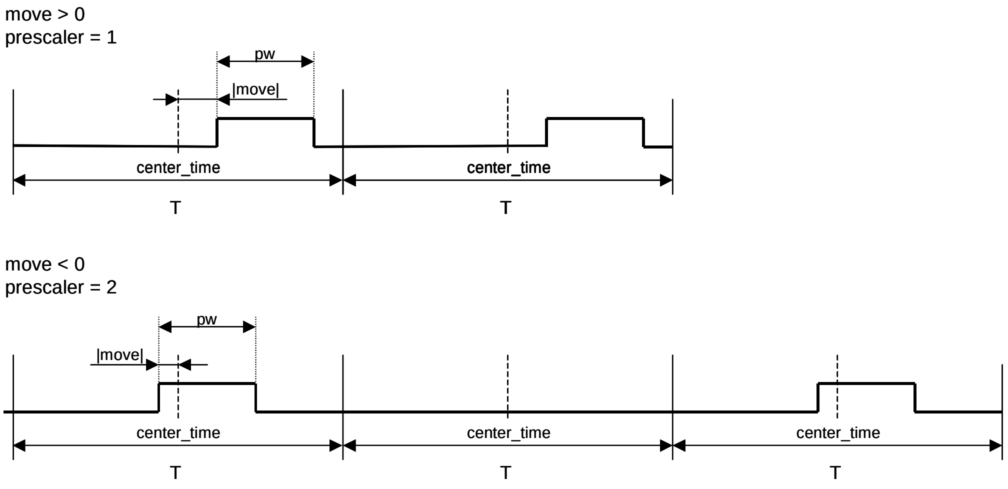


Figure 7. Synchronization signal adjustment examples

Synchronized Change of PWM Prescaler And Synchronization Signal Prescaler

The 3Sin3_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 3Sin3 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. Write the synchronization signal *prescaler* parameter address to the *sync_presc_addr* parameter to enable this mechanism. Write 0 to disable it, and remember to set the synchronization signal *presc_copy* parameter instead of the *prescaler* parameter in this case.

Host Interface

□	Written By CPU	🟨	Written by both CPU and TPU
🟦	Written By TPU	🟩	Not Used

Table 8. 3Sin3_sync Control Bits

Name	Options
3 2 1 0 □□□□ Channel Function Select	3Sin3_sync 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
1 0 □□ Host Service Bits (HSR)	00 – No Host Service Request 01 – Not used 10 – Initialization 11 – Not used
1 0 🟩🟩 Host Sequence Bits (HSQ)	xx – Not used
0 □ Channel Interrupt Enable	0 – Channel Interrupt Disabled 1 – Channel Interrupt Enabled
0 🟦 Channel Interrupt Status	0 – Interrupt Not Asserted 1 – Interrupt Asserted

TPU function 3Sin3_sync generates an interrupt after each low to high transition.

Table 9. 3Sin3_sync Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Synchronization channel	0	move															
	1	pw															
	2	prescaler															
	3	presc_copy															
	4	time															
	5	dec															
	6	T_copy															
	7																

Table 10. 3Sin3_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 – use in case of synchronized prescalers change
presc_copy	16-bit unsigned integer	The number of PWM periods per synchronization pulse – use in case of asynchronized prescalers change
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 11. 3Sin3_sync State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	12	5
S1	12	6
S2	8	3
S3	16	7

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

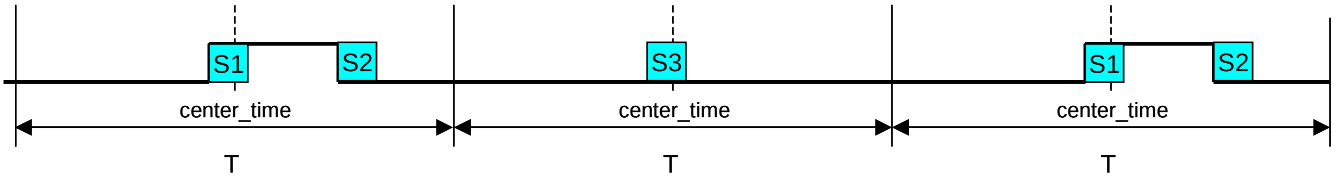


Figure 8. 3Sin3_sync timing

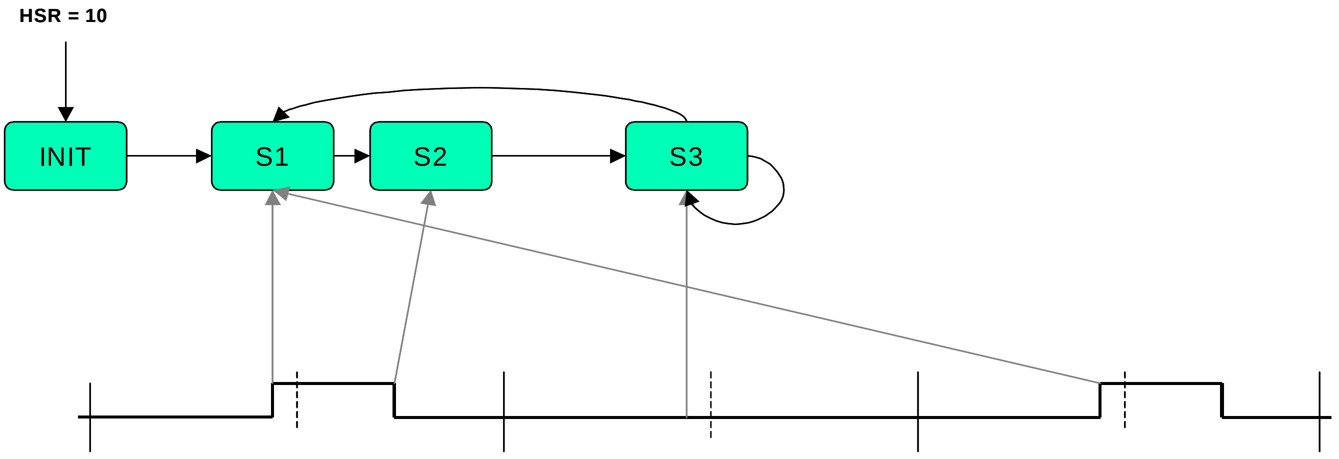


Figure 9. 3Sin3_sync state diagram

Resolver Reference Signal for 3-Phase Sine Wave Generator – 3 outputs version (3Sin3_res)

The 3Sin3_res TPU function uses information read from the 3Sin3 PWM 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.

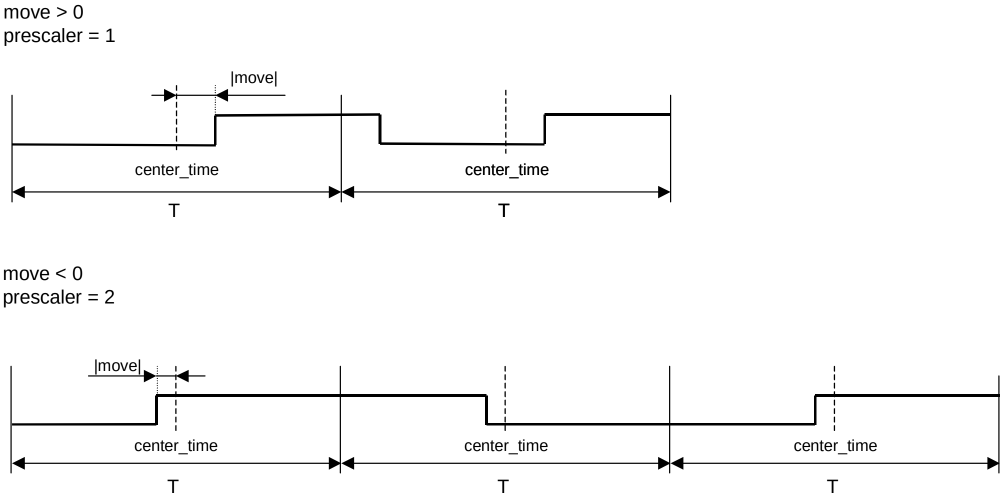


Figure 10. Resolver reference signal adjustment examples

Synchronized Change of PWM Prescaler And Resolver Reference Signals Prescaler

The 3Sin3_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 12. 3Sin3_res Control Bits

Name		Options
3 2 1 0 □□□□	Channel Function Select	3Sin3_res 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
1 0 □□	Host Service Bits (HSR)	00 – No Host Service Request 01 – Not used 10 – Initialization 11 – Not used
1 0 🟩🟩	Host Sequence Bits (HSQ)	xx – Not used
0 🟩	Channel Interrupt Enable	x – Not used
0 🟩	Channel Interrupt Status	x – Not used

Table 13. 3Sin3_res Parameter RAM

<table>
<tr><th>Channel</th><th>Parameter</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td rowspan="8">Resolver</td><td>0</td><td colspan="16">move</td></tr>
<tr><td>1</td><td colspan="16">🟩</td></tr>
<tr><td>2</td><td colspan="16">presc_addr</td></tr>
<tr><td>3</td><td colspan="16">prescaler</td></tr>
<tr><td>4</td><td colspan="16">🟦 time</td></tr>
<tr><td>5</td><td colspan="16">🟦 dec</td></tr>
<tr><td>6</td><td colspan="16">🟦 T_copy</td></tr>
<tr><td>7</td><td colspan="16">🟩</td></tr>
</table>

Table 14. 3Sin3_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	$00X6, where X is a number of Synchronization Signal channel, to inherit Sync. channel prescaler or $0000 to enable direct specification of prescaler value in prescaler parameter
prescaler	1, 2, 4, 6, 8, 10, 12, 14, ...	The number of PWM periods per synchronization pulse – use when apresc_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 15. 3Sin3_res State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	12	5
S1	26	9
S3	18	7

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

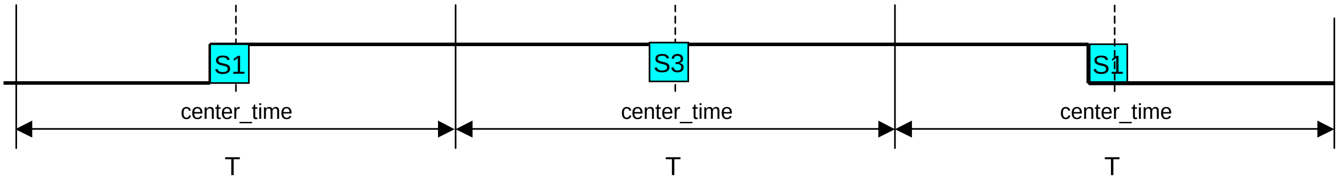


Figure 11. 3Sin3_res timing

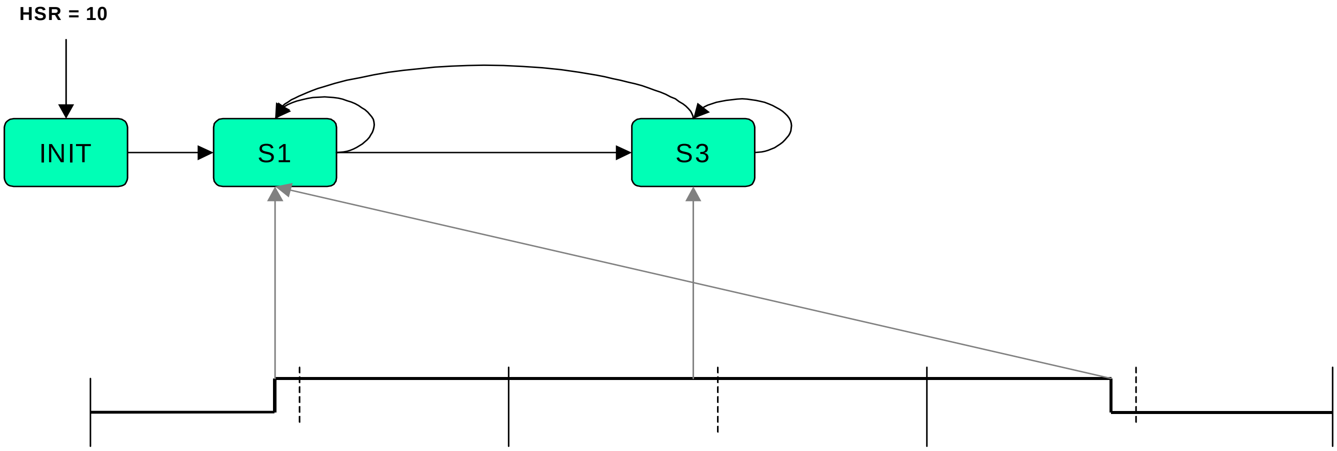


Figure 12. 3Sin3_res state diagram

Fault Input for 3-Phase Sine Wave Generator – 3 outputs version (3Sin3_fault)

The 3Sin3_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 Phase A 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 16. 3Sin3_fault Control Bits

Name		Options
3 2 1 0 □□□□	Channel Function Select	3Sin3_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
1 0 □□	Host Service Bits (HSR)	00 – No Host Service Request 01 – Not used 10 – Initialization 11 – Not used
1 0 🟩🟩	Host Sequence Bits (HSQ)	xx – Not used
0 □	Channel Interrupt Enable	0 – Channel Interrupt Disabled 1 – Channel Interrupt Enabled
0 🟦	Channel Interrupt Status	0 – Interrupt Not Asserted 1 – Interrupt Asserted

TPU function 3Sin3_fault generates an interrupt when a high to low transition appears.

Table 17. 3Sin3_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 18. 3Sin3_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 19. 3Sin3_fault State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	10	3
FAULT	28	2
NO_FAULT	6	2

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

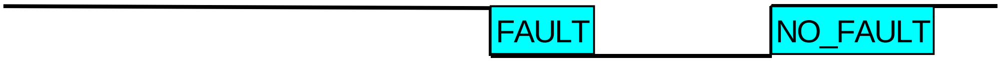


Figure 13. 3Sin3_fault timing

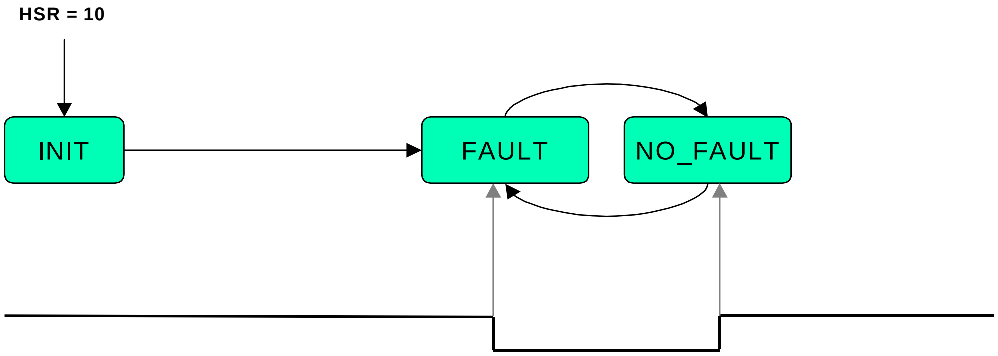


Figure 14. 3Sin3_fault state diagram

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:
Motorola Literature Distribution
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 480-768-2130

JAPAN:
Motorola Japan Ltd.
SPS, Technical Information Center
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-8573, Japan
81-3-3440-3569

ASIA/PACIFIC:
Motorola Semiconductors H.K. Ltd.
Silicon Harbour Centre
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
852-26668334

HOME PAGE:
http://motorola.com/semiconductors

MOTOROLA

Information in this document is provided solely to enable system and software implementers to use Motorola 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.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola 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 that may be provided in Motorola 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. Motorola does not convey any license under its patent rights nor the rights of others. Motorola 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 Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola 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 Motorola was negligent regarding the design or manufacture of the part.

AN2515/D
Rev. 0
5/2003