*Application Note*

*AN2517/D*
*Rev. 0, 5/2003*

# *3-Phase Sine Wave Generator with Dead-Time Correction TPU Function Set (3SinDt)*

**By Milan Brejl, Ph.D.**



## Functional Overview

The 3-Phase Sine Wave Generator with Dead-Time Correction TPU function set (3SinDt) extends the functionality of the 3-Phase Sine Wave Generator TPU function set (3Sin) by the dead-time correction technique. Except for this, its functionality is the same in all aspects.

The dead-time correction technique requires knowledge of the instantaneous direction of the 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 1**.

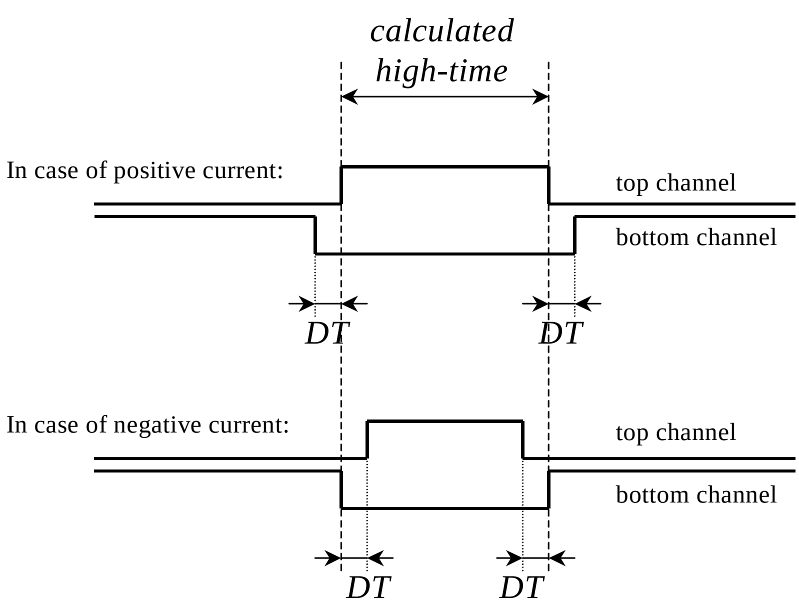



**Figure 1. Dead-Time Correction Technique**

The function set consists of 5 TPU functions:

- 3-Phase Sine Wave Generator with Dead-Time Correction – Top (3SinDt_top)
- 3-Phase Sine Wave Generator with Dead-Time Correction – Bottom (3SinDt_bottom)
- Synchronization Signal for 3-Phase Sine Wave Generator with Dead-Time Correction (3SinDt_sync)
- Resolver Reference Signal for 3-Phase Sine Wave Generator with Dead-Time Correction (3SinDt_res)
- Fault Input for 3-Phase Sine Wave Generator with Dead-Time Correction (3SinDt_fault)

The 3SinDt_top and 3SinDt_bottom TPU functions work together to generate a 6-channel 3-phase center-aligned PWM signal with dead-time between the top and bottom channels. The Synchronization Signal for the 3SinDt function can be used to generate one or more adjustable signals for a wide range of uses, that are synchronized to the PWM, and track changes in the PWM period. The Resolver Reference Signal for the 3SinDt 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 3SinDt function is a TPU input function that sets all PWM outputs low when the input signal goes low. See **Figure 2**.

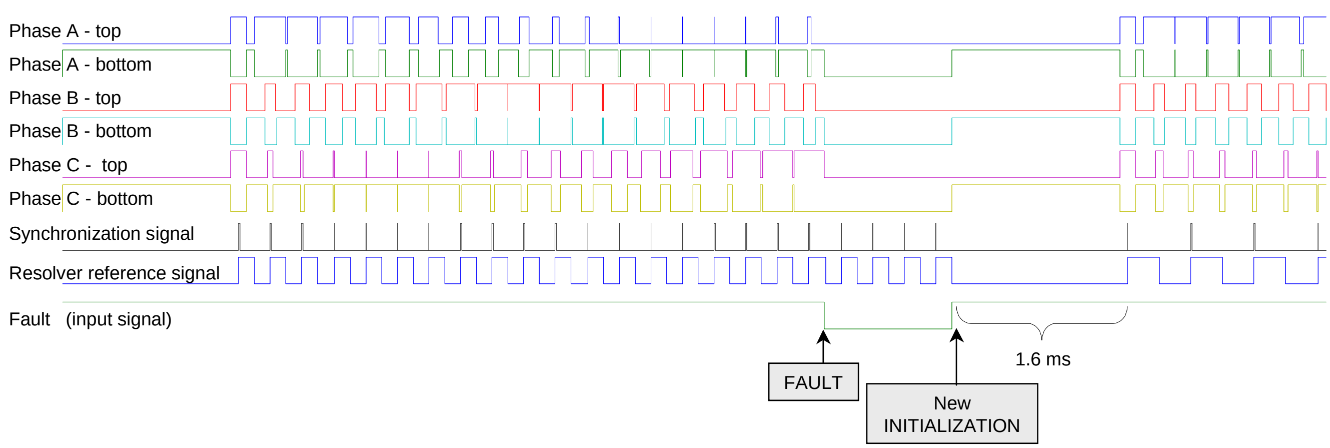



**Figure 2. Signals generated by 3SinDt TPU function set**

## Function Set Configuration

None of the TPU functions in the 3-Phase Sine Wave Generator with Dead-Time Correction TPU function set can be used separately. The 3SinDt_top and 3SinDt_bottom functions have to be used together. The 3SinDt_top is used on 3 channels, the 3SinDt_bottom on a further 3 channels, and within each phase, the function 3SinDt_top has to be assigned on a lower TPU channel than the function 3SinDt_bottom. This is illustrated in the examples in **Table 2** and **Table 3**. The 3SinDt_top and 3SinDt_bottom functions use a table of 32 cosine function values. The table is placed in the parameter space of four consecutive channels. One or more channels running Synchronization Signal for 3SinDt as well as Resolver Reference Signals for 3SinDt functions can be added to the 3SinDt_top and 3SinDt_bottom functions. They can run with different settings on each channel. The function Fault Input for 3SinDt can also be added to the 3SinDt_top and 3SinDt_bottom 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 signals, that are disabled in this configuration. The function 3SinDt_fault can run on one of the four channels where the table of cosine function values is placed, because the 3SinDt_fault function does not have any parameters.

**Table 1** shows the configuration options and restrictions.

**Table 1. 3SinDt TPU function set configuration options and restrictions**

| TPU function | Optional/ Mandatory | How many channels | Assignable channels |
|---|---|---|---|
| 3SinDt_top | mandatory | 3 | any 3 channels, within each phase a lower TPU channel then the same phase 3SinDt_bottom |
| 3SinDt_bottom | mandatory | 3 | any 3 channels, within each phase a higher TPU channel then the same phase 3SinDt_top |
| Cosine table | mandatory | 4 | any 4 consecutive channels |
| 3SinDt_sync | optional | 1 or more | any channels |
| 3SinDt_res | optional | 1 or more | any channels |
| 3SinDt_fault | optional | 1 | any, recommended is 15 and DTPU bit set |

Table 2 and Table 3 show two examples of configuration.

**Table 2. Example of configuration**

| Channel | TPU function | Priority |
|---|---|---|
| 0 | 3SinDt_top | high |
| 1 | 3SinDt_bottom | high |
| 2 | 3SinDt_top | high |
| 3 | 3SinDt_bottom | high |
| 4 | 3SinDt_top | high |
| 5 | 3SinDt_bottom | high |
| 10 | 3SinDt_sync | low |
| 12 | Cosine table 1 | none |
| 13 | Cosine table 2 | none |
| 14 | Cosine table 3 | none |
| 15 | 3SinDt_fault + Cosine table 4 | high |

**Table 3. Example of configuration**

| Channel | TPU function | Priority |
|---|---|---|
| 0 | 3SinDt_top | high |
| 1 | 3SinDt_top | high |
| 2 | 3SinDt_top | high |
| 3 | 3SinDt_bottom | high |
| 4 | 3SinDt_bottom | high |
| 5 | 3SinDt_bottom | high |
| 10 | 3SinDt_sync | low |
| 11 | 3SinDt_res | low |
| 12 | Cosine table 1 | none |
| 13 | Cosine table 2 | none |
| 14 | Cosine table 3 | none |
| 15 | 3SinDt_fault + Cosine table 4 | high |

Table 4 shows the TPU function code sizes.

**Table 4. TPU function code sizes.**

| TPU function | Code size |
|---|---|
| 3SinDt_top | 34 µ instructions + 8 entries = 42 long words |
| 3SinDt_bottom | 217 µ instructions + 8 entries = 225 long words |
| 3SinDt_sync | 26 µ instructions + 8 entries = 34 long words |
| 3SinDt_res | 38 µ instructions + 8 entries = 46 long words |
| 3SinDt_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*, *MPW*, *Theta_H*, *Theta_L* and *sync_presc_addr* must be set before initialization. 32 cosine table values must be set. If a 3SinDt_sync channel or a 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 3SinDt_bottom channels to initialize all PWM channels. Issues an HSR type %10 to the 3SinDt_sync channels, 3SinDt_res channels and 3SinDt_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 3SinDt_sync or 3SinDt_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 3SinDt_sync or 3SinDt_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 – Top (3SinDt_top) and 3-Phase Sine Wave Generator with Dead-Time Correction – Bottom (3SinDt_bottom)

The 3SinDt_top and 3SinDt_bottom TPU functions work together to generate a 6-channel, 3-phase 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 amplitude of 0 (50% duty-cycle) until the first reload 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*) and the minimum pulse width (*MPW*) are 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 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 $t_{low\text{-}high}$ and $t_{high\text{-}low}$ 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 in the first quadrant. The function value is obtained by linear interpolation between two the closest table values. **Figure 3** shows the error of the cosine function value calculation. The maximum error is 7 in the amplitude range <–32768, 32767>, that is 0.021%.

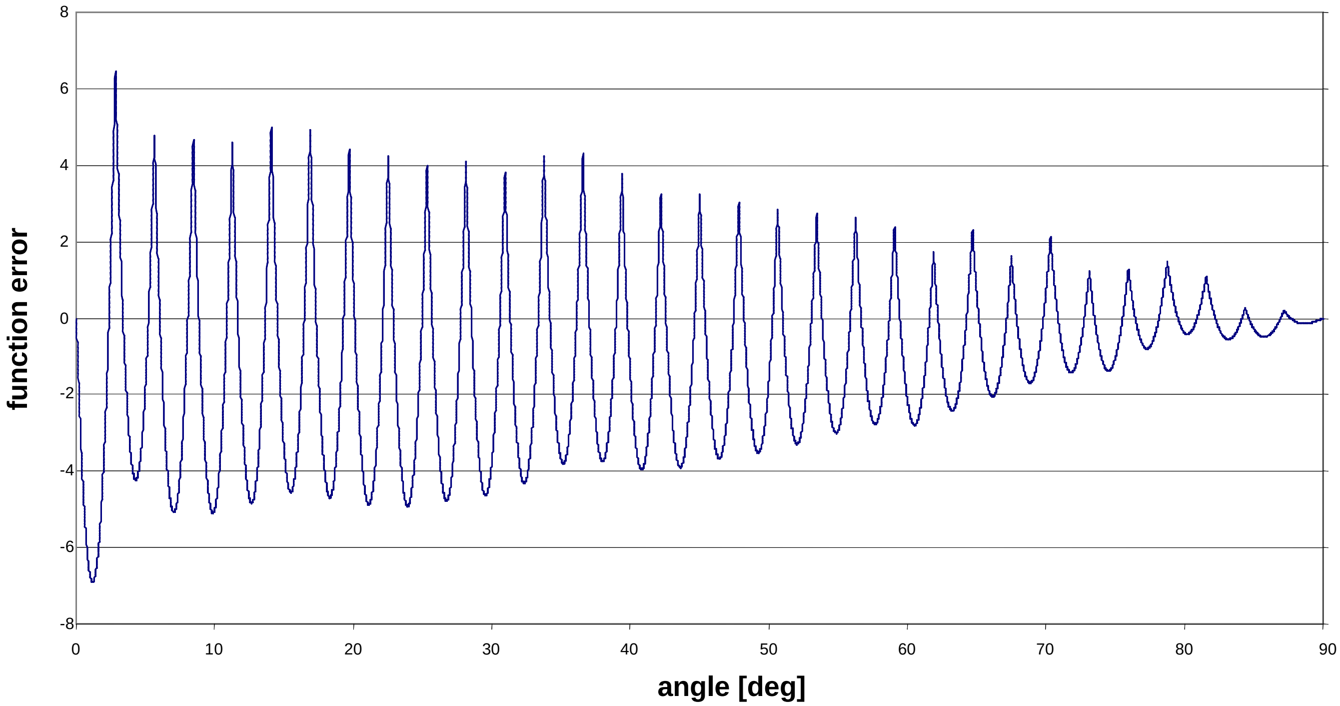
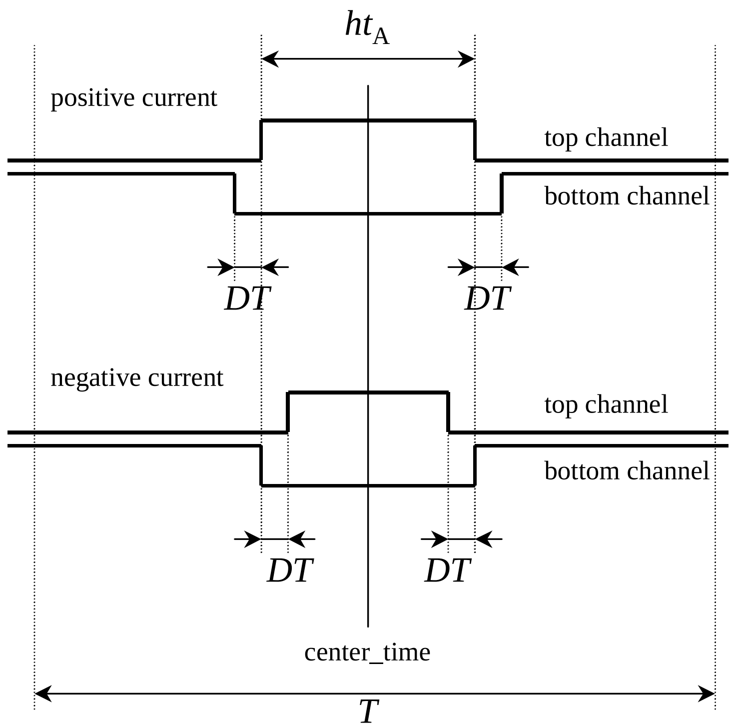



**Figure 3. Cosine function value error**

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

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

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




**Phase A:**

Positive current

top channel

$$t_{\text{low-high}} = \text{center\_time} - \frac{ht_A}{2}$$

$$t_{\text{high-low}} = \text{center\_time} + \frac{ht_A}{2}$$

bottom channel

$$t_{\text{high-low}} = \text{center\_time} - \frac{ht_A}{2} - DT$$

$$t_{\text{low-high}} = \text{center\_time} + \frac{ht_A}{2} + DT$$

Negative current

top channel

$$t_{\text{low-high}} = \text{center\_time} - \frac{ht_A}{2} + DT$$

$$t_{\text{high-low}} = \text{center\_time} + \frac{ht_A}{2} - DT$$

bottom channel

$$t_{\text{high-low}} = \text{center\_time} - \frac{ht_A}{2}$$

$$t_{\text{low-high}} = \text{center\_time} + \frac{ht_A}{2}$$

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 5. 3SinDt_top Control Bits**

| Name | | Options |
|---|---|---|
| 3 2 1 0 | Channel Function Select | 3SinDt_top function number (Assigned during assembly the DPTRAM code from library TPU functions) |
| 1 0 | Channel Priority | 00 – Channel Disabled<br>01 – Low Priority<br>10 – Middle Priority<br>11 – High Priority |
| 1 0 | Host Service Bits (HSR) | 00 – No Host Service Request<br>01 – Not used<br>10 – Not used<br>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 6. 3SinDt_bottom Control Bits**

| Name | | Options |
|---|---|---|
| 3 2 1 0 | Channel Function Select | 3SinDt_bottom function number (Assigned during assembly the DPTRAM code from library TPU functions) |
| 1 0 | Channel Priority | 00 – Channel Disabled<br>01 – Low Priority<br>10 – Middle Priority<br>11 – High Priority |
| 1 0 | Host Service Bits (HSR) | 00 – No Host Service Request<br>01 – Not used<br>10 – Initialization<br>11 – Stop |
| 1 0 | Host Sequence Bits (HSQ) | xx – Not used |
| 0 | Channel Interrupt Enable | 0 – Channel Interrupt Disabled<br>1 – Channel Interrupt Enabled |
| 0 | Channel Interrupt Status | 0 – Interrupt Not Asserted<br>1 – Interrupt Asserted |

TPU function 3SinDt_bottom 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 bottom channels. The top channels do not generate any interrupts.

**Table 7. 3SinDt_top and 3SinDt_bottom 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">Phase A top channel</td><td>0</td><td colspan="16">htA</td></tr>
<tr><td>1</td><td colspan="16">HLtime_AT</td></tr>
<tr><td>2</td><td colspan="16">bottom_chan_A</td></tr>
<tr><td>3</td><td colspan="16">currentA</td></tr>
<tr><td>4</td><td colspan="16">LD_OK</td></tr>
<tr><td>5</td><td colspan="16">TA_buf</td></tr>
<tr><td>6</td><td colspan="16"></td></tr>
<tr><td>7</td><td colspan="16">fault_pinstate</td></tr>
<tr><td rowspan="8">Phase A bottom channel</td><td>0</td><td colspan="16">LHtime_AB</td></tr>
<tr><td>1</td><td colspan="16">HLtime_AB</td></tr>
<tr><td>2</td><td colspan="16">sA</td></tr>
<tr><td>3</td><td colspan="16">T_copy</td></tr>
<tr><td>4</td><td colspan="16">Theta_H</td></tr>
<tr><td>5</td><td colspan="16">Theta_L</td></tr>
<tr><td>6</td><td colspan="16">Theta_buf_H</td></tr>
<tr><td>7</td><td colspan="16">Theta_buf_L</td></tr>
<tr><td rowspan="8">Phase B top channel</td><td>0</td><td colspan="16">htB</td></tr>
<tr><td>1</td><td colspan="16">HLtime_BT</td></tr>
<tr><td>2</td><td colspan="16">bottom_chan_B</td></tr>
<tr><td>3</td><td colspan="16">currentB</td></tr>
<tr><td>4</td><td colspan="16">Ampl</td></tr>
<tr><td>5</td><td colspan="16">sync_presc_addr</td></tr>
<tr><td>6</td><td colspan="16"></td></tr>
<tr><td>7</td><td colspan="16"></td></tr>
<tr><td rowspan="8">Phase B bottom channel</td><td>0</td><td colspan="16">LHtime_BB</td></tr>
<tr><td>1</td><td colspan="16">HLtime_BB</td></tr>
<tr><td>2</td><td colspan="16">sB</td></tr>
<tr><td>3</td><td colspan="16">min_ht</td></tr>
<tr><td>4</td><td colspan="16">T</td></tr>
<tr><td>5</td><td colspan="16">prescaler</td></tr>
<tr><td>6</td><td colspan="16">dec</td></tr>
<tr><td>7</td><td colspan="16">center_time</td></tr>
<tr><td rowspan="8">Phase C top channel</td><td>0</td><td colspan="16">htC</td></tr>
<tr><td>1</td><td colspan="16">HLtime_CT</td></tr>
<tr><td>2</td><td colspan="16">bottom_chan_C</td></tr>
<tr><td>3</td><td colspan="16">currentC</td></tr>
<tr><td>4</td><td colspan="16">dTheta_H</td></tr>
<tr><td>5</td><td colspan="16">dTheta_L</td></tr>
<tr><td>6</td><td colspan="16">dTheta_buf_H</td></tr>
<tr><td>7</td><td colspan="16">dTheta_buf_L</td></tr>
</table>

**Table 7. 3SinDt_top and 3SinDt_bottom Parameter RAM**

| Channel | Parameter | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Phase C bottom channel | 0 | LHtime_CB | | | | | | | | | | | | | | | |
| | 1 | HLtime_CB | | | | | | | | | | | | | | | |
| | 2 | Fchan | | | | | | | | | | | | | | | |
| | 3 | max_ht | | | | | | | | | | | | | | | |
| | 4 | DT | | | | | | | | | | | | | | | |
| | 5 | MPW | | | | | | | | | | | | | | | |
| | 6 | prsc_copy | | | | | | | | | | | | | | | |
| | 7 | | | | | | | | | | | | | | | | |

**Table 8. 3SinDt_top and 3SinDt_bottom 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°) |
| currentA | 0 or 1 | 0 ... positive current on phase A<br>1 ... negative current on phaseA |
| currentB | 0 or 1 | 0 ... positive current on phase B<br>1 ... negative current on phaseB |
| currentC | 0 or 1 | 0 ... positive current on phase C<br>1 ... negative current on phaseC |
| 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 |
| MPW | 16-bit unsigned integer | Minimum pulse width in number of TCR1 TPU cycles. See **Performance** for details. |

**Table 8. 3SinDt_top and 3SinDt_bottom parameter description**

| Parameter | Format | Description |
|---|---|---|
| sync_presc_addr | 8-bit unsigned integer | address of synchronization channel *prescaler* parameter: $X4,<br>where X is synchronization channel number.<br>$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<br><>0 .. TPU can read variables:<br>$0001 ... load *Ampl*, *dTheta*, *T* and *prescaler* only<br>$8001 ... load also *Theta*<br>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:<br>0 ... low<br>1 ... high |
| Theta_buf | 32-bit fractional | Actual Stator Reference Voltage Vector angle<br>range <–1, 1) corresponds to <–180°, 180°) |
| Other parameters are just for TPU function inner use. | | |

*Performance*

The maximum PWM frequency is 32kHz (PWM period *T* = 625). This can be achieved when only 3SinDt_top and 3SinDt_bottom 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 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 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 625, *T* has to be for that particular case.

Table 9. 3SinDt_top State Statistics

| State | Max IMB Clock Cycles | RAM Accesses by TPU |
|---|---|---|
| HL | 2 | 1 |
| LH_C7 | 40 | 10 |

Table 10. 3SinDt_bottom State Statistics

| State | Max IMB Clock Cycles | RAM Accesses by TPU |
|---|---|---|
| INIT | 122 | 36 |
| STOP | 38 | 0 |
| LH | 2 | 1 |
| HL | 2 | 1 |
| LH_RLD | 62 | 23 |
| C1 | 46 | 4 |
| C2 | 84 | 10 |
| C3 | 82 | 6 |
| C4 | 62 | 6 |
| C5 | 62 | 6 |
| C6 | 66 | 6 |

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

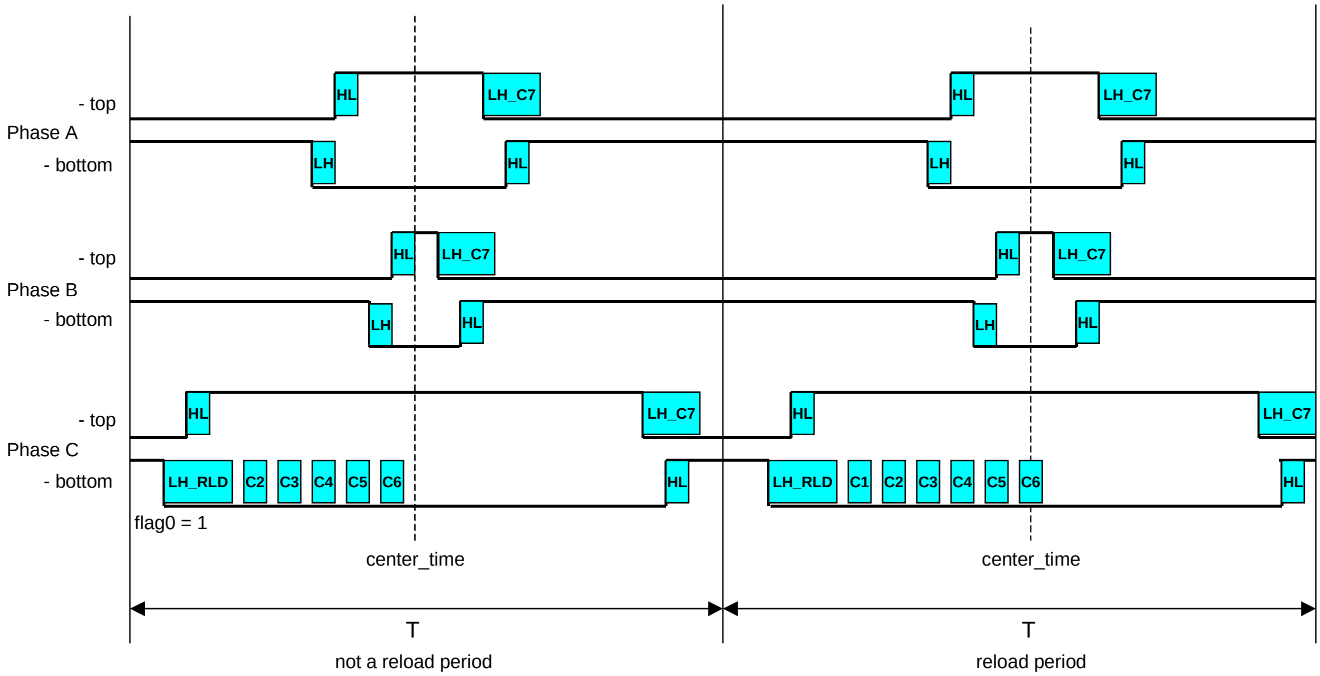



**Figure 4. 3SinDt_top and 3SinDt_bottom timing**

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

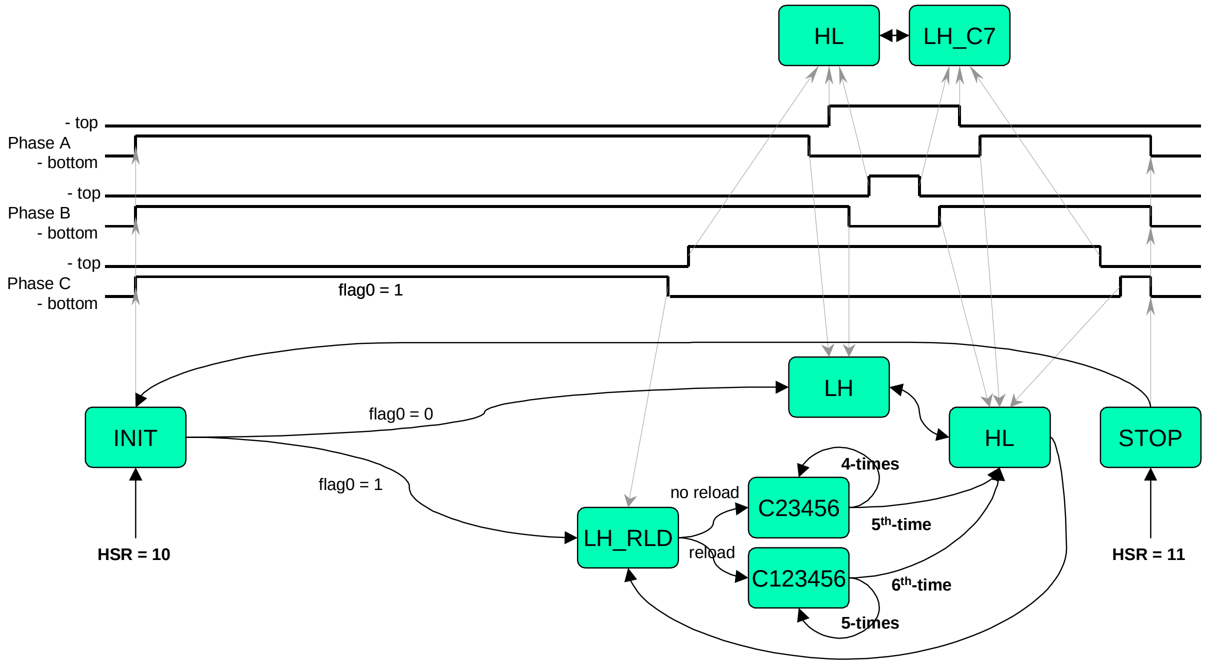



**Figure 5. 3SinDt_top and 3SinDt_bottom state diagram**

*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 correctly generate 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 functions 3SinDt_top and 3SinDt_bottom limit 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

$$1 - \frac{2(MPW + 2DT)}{T}$$

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 6**.

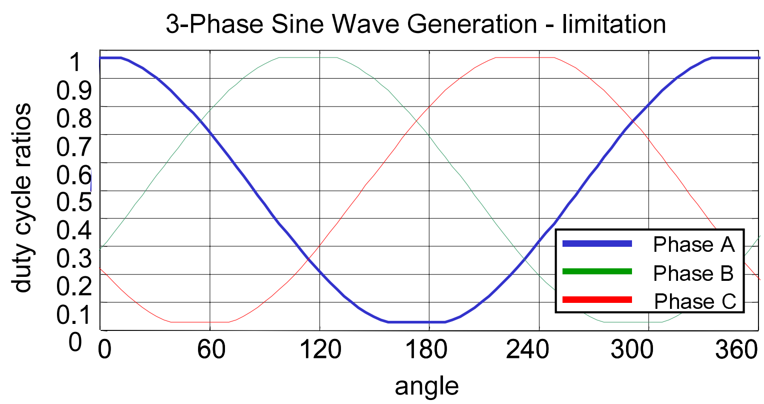



**Figure 6. Effect of limitation**

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.

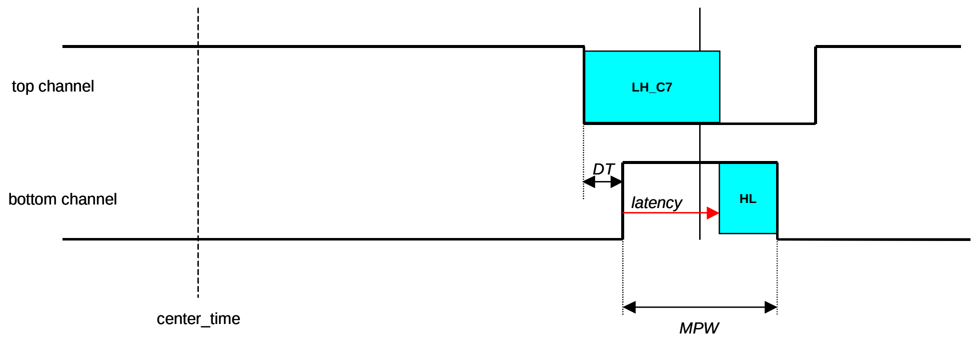



**Figure 7. Timing of the worst case**

When 3SinDt_top and 3SinDt_bottom are running alone on one TPU, the minimum pulse width can be calculated according to **Figure 7**. This illustrates the worst case timing. The bottom channel low to high transition runs the HL state that sets the following high to low transition. The HL state lasts 2 IMB clock cycles (see **Table 10**). Each state is preceded by the Time Slot Transition (TST), which takes 10 IMB clock cycles. So the time necessary to set the next transition on the bottom channel is 12 IMB clock cycles. In addition, there is a latency between the low to high transition and the start of the HL state. The top channel state LH_C7, which is serviced at the time, causes the latency. The LH_C7 state lasts 40 IMB clock cycles (see **Table 9**). Its time slot transition is 10 IMB clock cycles. The service starts immediately after the top channel high to low transition, which occurs at a period of *DT* before the bottom channel low to high transition (see **Figure 7**), so that the latency is 40 IMB clock cycles + 10 IMB clock cycles – *DT*. The 3SinDt functions are designed so that no other 3SinDt state can request service at this time. The *MPW*, in the case when only 3SinDt functions are running on one TPU, is then

latency + 12 IMB clock cycles =
= 40 IMB clock cycles + 10 IMB clock cycles – *DT* + 12 IMB clock cycles =
= 62 IMB clock cycles – *DT*

and has a minimum value of at least 12 IMB clock cycles (when latency = 0).

**Note that the *MPW*, as well as the *DT*, are not entered into the parameter RAM in IMB clock cycles, but in TCR1 clock cycles.** It is recommended for the 3SinDt function that the TCR1 clock is configured for its maximum speed, which is the IMB clock divided by 2. In this case the *MPW* = 31 – *DT*, with a minimum value of 6.

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 3SinDt_fault function does not affect the *MPW*. The 3SinDt_sync, if used, increases the *MPW* value by 22 (44 IMB clock cycles). The 3SinDt_res, if used, increases the *MPW* value by 20 (40 IMB clock cycles).

If a lower value than the one calculated, is set for the *MPW* parameter, the motion system can run with a higher motor voltage amplitude, but with risk, that the dead-time is not maintained.

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 3SinDt 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 with Dead-Time Correction (3SinDt_sync)**

The 3SinDt_sync TPU function uses information obtained from 3SinDt 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.

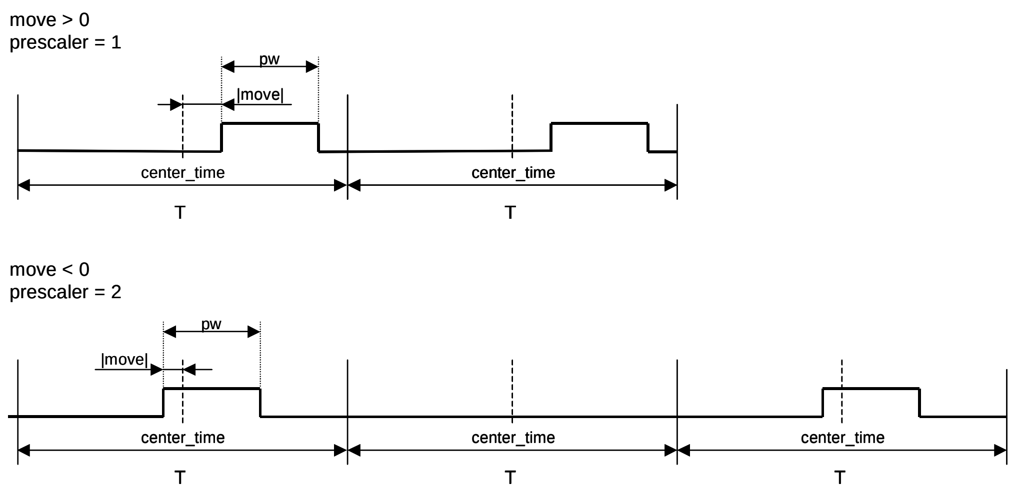



**Figure 8. Synchronization signal adjustment examples**

*Synchronized Change of PWM Prescaler And Synchronization Signal Prescaler*

The 3SinDt_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 3SinDt_bottom 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 (white) | Written by both CPU and TPU (yellow) |
| Written By TPU (light blue) | Not Used (green) |

**Table 11. 3SinDt_sync Control Bits**

| Name | | Options |
|---|---|---|
| 3 2 1 0 | Channel Function Select | 3SinDt_sync function number (Assigned during assembly the DPTRAM code from library TPU functions) |
| 1 0 | Channel Priority | 00 – Channel Disabled<br>01 – Low Priority<br>10 – Middle Priority<br>11 – High Priority |
| 1 0 | Host Service Bits (HSR) | 00 – No Host Service Request<br>01 – Not used<br>10 – Initialization<br>11 – Not used |
| 1 0 (not used) | Host Sequence Bits (HSQ) | xx – Not used |
| 0 | Channel Interrupt Enable | 0 – Channel Interrupt Disabled<br>1 – Channel Interrupt Enabled |
| 0 (written by TPU) | Channel Interrupt Status | 0 – Interrupt Not Asserted<br>1 – Interrupt Asserted |

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

**Table 12. 3SinDt_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 13. 3SinDt_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 14. 3SinDt_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)*

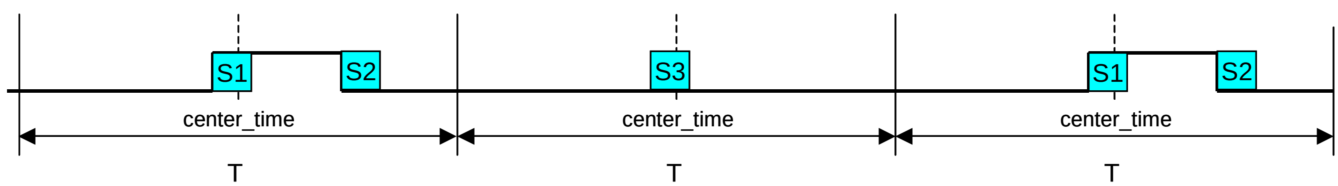



**Figure 9. 3SinDt_sync timing**

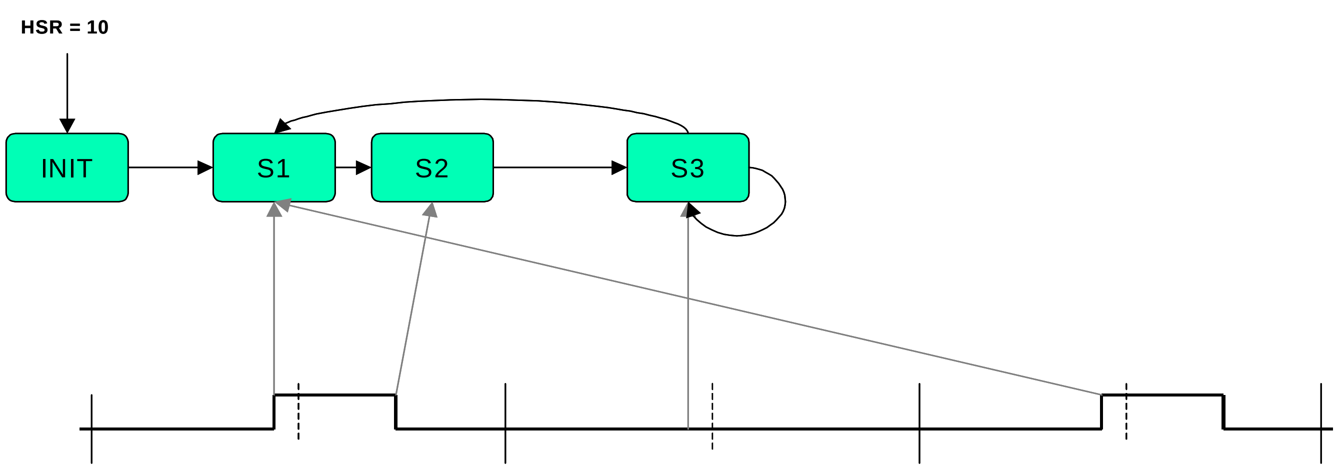



**Figure 10. 3SinDt_sync state diagram**

**Resolver Reference Signal for 3-Phase Sine Wave Generator with Dead-Time Correction (3SinDt_res)**

The 3SinDt_res TPU function uses information read from the 3SinDt 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.

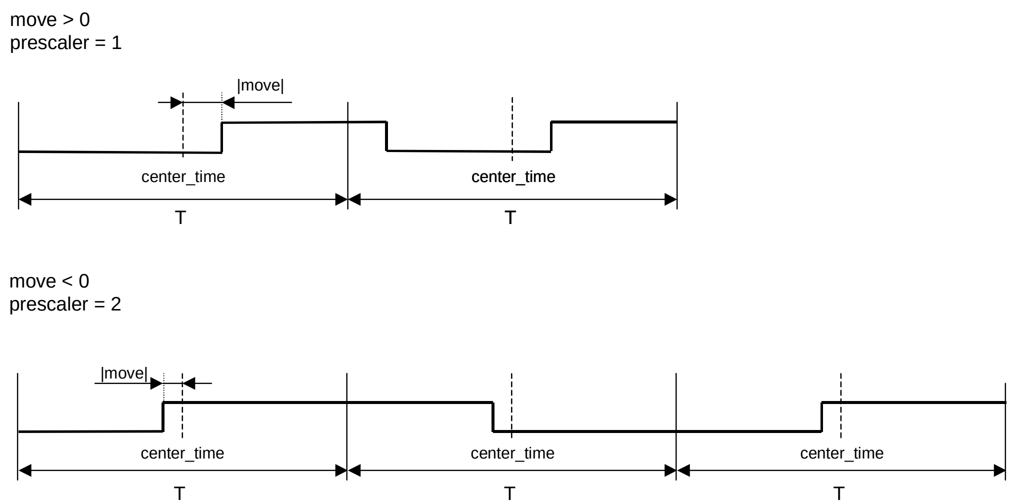



**Figure 11. Resolver reference signal adjustment examples**

*Synchronized Change of PWM Prescaler And Resolver Reference Signals Prescaler*

The 3SinDt_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 15. 3SinDt_res Control Bits**

| Name | Options |
|---|---|
| 3 2 1 0 Channel Function Select | 3SinDt_res function number (Assigned during assembly the DPTRAM code from library TPU functions) |
| 1 0 Channel Priority | 00 – Channel Disabled<br>01 – Low Priority<br>10 – Middle Priority<br>11 – High Priority |
| 1 0 Host Service Bits (HSR) | 00 – No Host Service Request<br>01 – Not used<br>10 – Initialization<br>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 16. 3SinDt_res Parameter RAM**

| Channel | Parameter | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Resolver | 0 | move | | | | | | | | | | | | | | | |
| | 1 | | | | | | | | | | | | | | | | |
| | 2 | presc_addr | | | | | | | | | | | | | | | |
| | 3 | prescaler | | | | | | | | | | | | | | | |
| | 4 | time | | | | | | | | | | | | | | | |
| | 5 | dec | | | | | | | | | | | | | | | |
| | 6 | T_copy | | | | | | | | | | | | | | | |
| | 7 | | | | | | | | | | | | | | | | |

**Table 17. 3SinDt_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<br>or<br>$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<br>– 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 18. 3SinDt_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)*

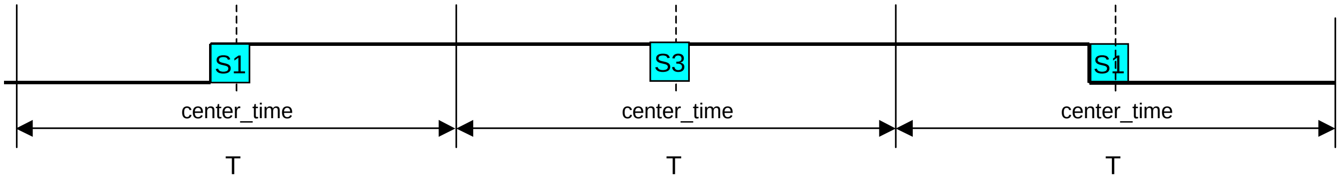



**Figure 12. 3SinDt_res timing**

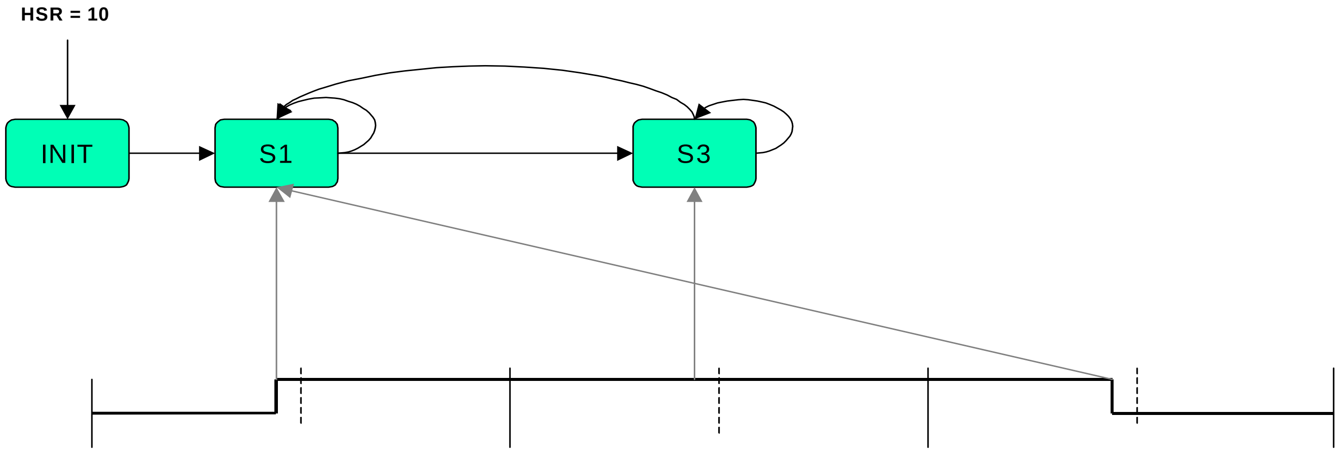



**Figure 13. 3SinDt_res state diagram**

**Fault Input for 3-Phase Sine Wave Generator with Dead-Time Correction (3SinDt_fault)**

The 3SinDt_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 – top channel to keep the fault channel parameter space free.

*Host Interface*

| | |
|---|---|
| (white) Written By CPU | (yellow) Written by both CPU and TPU |
| (blue) Written By TPU | (green) Not Used |

**Table 19. 3SinDt_fault Control Bits**

| Name | Options |
|---|---|
| 3 2 1 0 Channel Function Select | 3SinDt_fault function number (Assigned during assembly the DPTRAM code from library TPU functions) |
| 1 0 Channel Priority | 00 – Channel Disabled<br>01 – Low Priority<br>10 – Middle Priority<br>11 – High Priority |
| 1 0 Host Service Bits (HSR) | 00 – No Host Service Request<br>01 – Not used<br>10 – Initialization<br>11 – Not used |
| 1 0 Host Sequence Bits (HSQ) | xx – Not used |
| 0 Channel Interrupt Enable | 0 – Channel Interrupt Disabled<br>1 – Channel Interrupt Enabled |
| 0 Channel Interrupt Status | 0 – Interrupt Not Asserted<br>1 – Interrupt Asserted |

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

**Table 20. 3SinDt_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 21. 3SinDt_fault parameter description**

| Parameter | Format | Description |
|---|---|---|
| Parameters written by TPU | | |
| fault_pinstate | 0 or 1 | State of fault pin:<br>0 ... low<br>1 ... high |

*Performance*

**Table 22. 3SinDt_fault State Statistics**

| State | Max IMB Clock Cycles | RAM Accesses by TPU |
|---|---|---|
| INIT | 8 | 2 |
| FAULT | 44 | 1 |
| NO_FAULT | 4 | 1 |

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

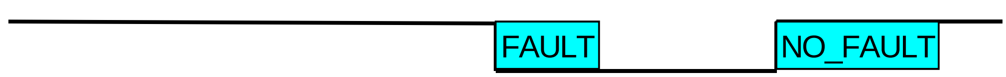



**Figure 14. 3SinDt_fault timing**

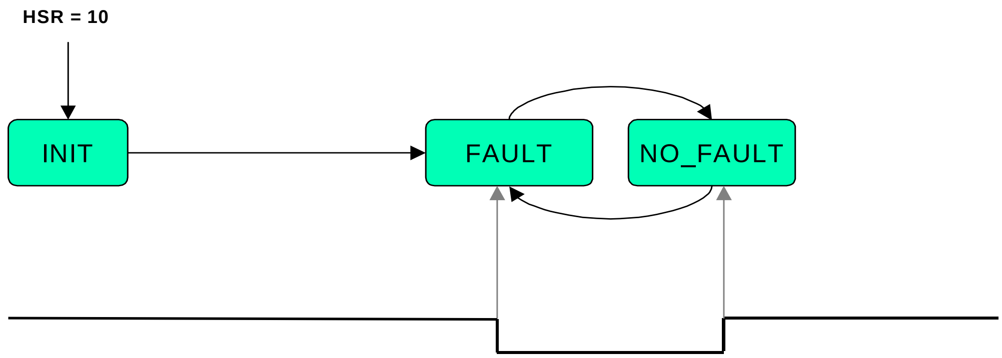



**Figure 15. 3SinDt_fault state diagram**

## 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



AN2517/D
Rev. 0
5/2003