*Application Note*

*AN2532/D*
*Rev. 0, 5/2003*

# *Standard Space Vector Modulation – 3 outputs version TPU Function Set (svmStd3)*

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

## Functional Overview

Standard Space Vector Modulation – 3 outputs version (svmStd3) is a variant of the svmStd function that, in contrary to svmStd, generates only top channel signal of each PWM pair. The bottom channel signal could be derived from the top channel signal by an external hardware. The function set consists of 4 TPU functions:

- Standard Space Vector Modulation – 3 outputs version (svmStd3)
- Synchronization signal for Standard Space Vector Modulation – 3 outputs version (svmStd3_sync)
- Resolver Reference Signal for Standard Space Vector Modulation – 3 outputs version (svmStd3_res)
- Fault Input for Standard Space Vector Modulation – 3 outputs version (svmStd3_fault)

The svmStd3 TPU function generates a 3-channel 3-phase center-aligned PWM signal. The generated signals control external hardware, which outputs pair of transistor signals (top and bottom) with dead-time inserted.

The Synchronization Signal for the svmStd3 function can be used to generate one or more adjustable signals for a wide range of uses, which are synchronized to

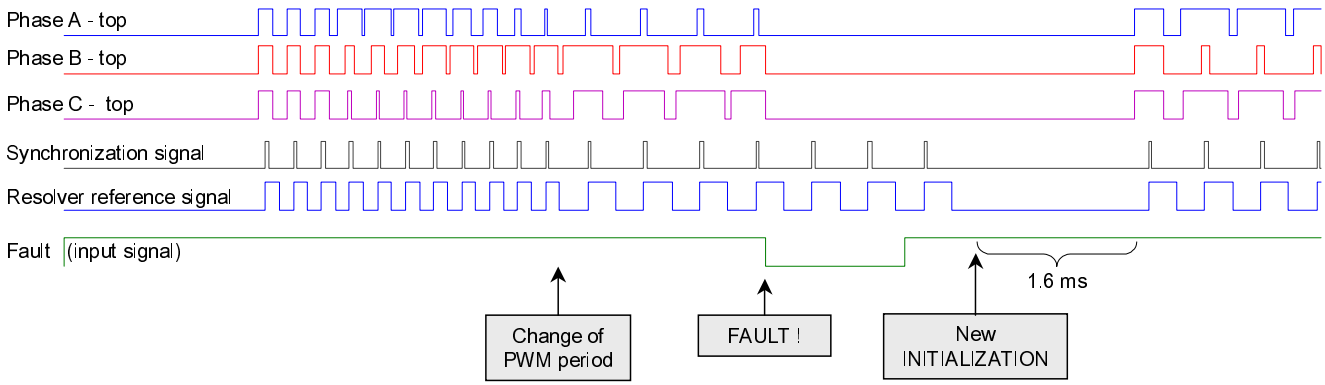



**Figure 1. Signals generated by svmStd3 TPU function set**

the PWM, and track changes in the PWM period. The Resolver Reference Signal for svmStd3 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 svmStd3 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 Standard Space Vector Modulation – 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. One or more channels running a Synchronization Signal for svmStd3 as well as Resolver Reference Signals for svmStd3 functions can be added to the main svmStd3 function. Each channel can run with different settings. When the Fault Input for svmStd3 is added, 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.

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

**Table 1. svmStd3 TPU function set configuration options and restrictions**

| TPU function | Optional/ Mandatory | How many channels | Assignable channels |
|---|---|---|---|
| svmStd3_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 | svmStd3 | high |
| 1 | svmStd3 | high |
| 2 | svmStd3 | high |
| 10 | svmStd3_sync | low |
| 11 | svmStd3_res | low |
| 15 | svmStd3_fault | high |

Table 3 shows the TPU function code sizes.

**Table 3. TPU function code sizes**

| TPU function | Code size |
|---|---|
| svmStd3 | 176 µ instructions + 8 entries = 184 long words |
| svmStd3_sync | 26 µ instructions + 8 entries = 34 long words |
| svmStd3_res | 38 µ instructions + 8 entries = 46 long words |
| svmStd3_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*, *MPW*, *SQRT3* and *sync_presc_addr* must be set before initialization. If an svmStd3_sync channel or an svmStd3_res channel is used, then also its parameters must be set before initialization.
4. Issues an HSR (Host Service Request) type %10 to one of the svmStd3 channels to initialize all PWM channels. Issues an HSR type %10 to the svmStd3_sync channels, svmStd3_res channels and svmStd3_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 svmStd3_sync or svmStd3_res channels are initialized after the initialization of 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 svmStd3_sync or svmStd3_res channels to enable their initialization

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

## Detailed Function Description

**Standard Space Vector Modulation – 3 outputs version (svmStd3)**

The svmStd3 TPU function generates a 3-channel, 3-phase PWM signal. Unlike svmStd, the generated signals are not top-bottom pairs with dead-times but only top-like signals without dead-times. 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 a 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 components $u_{\acute{a}}$ and $u_{\hat{a}}$ have to 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, 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 writes the parameter Sector that indicates the current Stator Reference Voltage Vector position in sector 1 to 6.

The following figures show the input Stator Reference Voltage Vector components $u_{\hat{a}}$ and $u_{\hat{a}}$, corresponding sectors and output PWM signal duty cycle ratios:

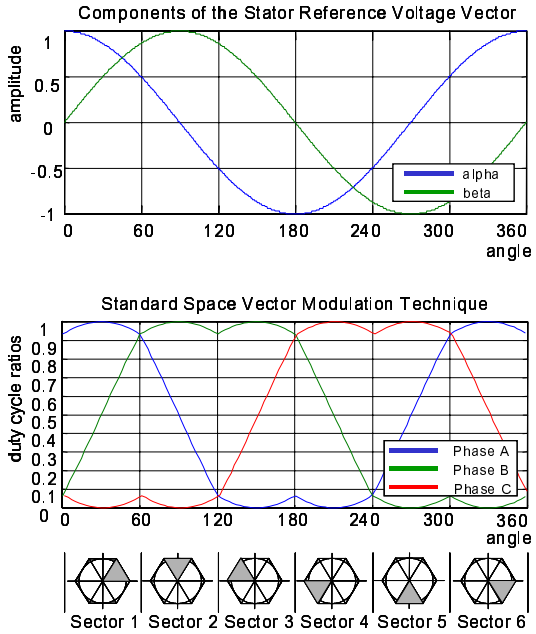



**Figure 2. Standard Space Vector Modulation – 3 outputs version Technique**

The following equations describe how the Space Vector Modulation 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:

$$U_\beta = T \cdot u_\beta$$

$$U_\alpha = T \cdot u_\alpha$$

$$X = U_\beta$$

$$Y = \frac{U_\beta + U_\alpha \sqrt{3}}{2}$$

$$Z = \frac{U_\beta - U_\alpha \sqrt{3}}{2}$$

| | Y < 0 | | | Y >= 0 | | |
|---|---|---|---|---|---|---|
| | Z < 0 | Z >= 0 | | Z < 0 | | Z >= 0 |
| | | X <= 0 | X > 0 | X <= 0 | X > 0 | |
| **Sector:** | **V.** | **IV.** | **III.** | **VI.** | **I.** | **II.** |

Sector I., IV.: $ht_A = \frac{T+X-Z}{2}$

$ht_B = \frac{T+X+Z}{2} = t_A + Z$

$ht_C = \frac{T-X+Z}{2} = t_B - X$

Sector II., V.: $ht_A = \frac{T+Y-Z}{2}$

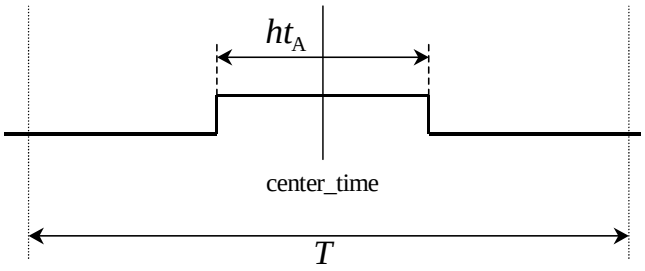
$ht_B = \frac{T+Y+Z}{2} = t_A + Z$

$ht_C = \frac{T-Y-Z}{2} = t_A - Y$

Sector III., VI.: $ht_A = \frac{T-X+Y}{2}$

$ht_B = \frac{T+X-Y}{2} = t_C + X$

$ht_C = \frac{T-X-Y}{2} = t_A - Y$




Phase A:

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

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

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 4. svmStd3 Control Bits**

| Name | Options |
|---|---|
| 3 2 1 0<br>Channel Function Select | svmStd3 function number (Assigned during assembly the DPTRAM code from library TPU functions) |

**Table 4. svmStd3 Control Bits**

| Name | Options |
|---|---|
| 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 svmStd3 generates an interrupt when the current values of *Ualfa*, *Ubeta*, *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* bit to check it has cleared. The interrupt is generated at each reload by one of the PWM channels.

**Table 5. svmStd3 Parameter RAM**

| Channel | Parameter | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Phase A | 0 | htA | | | | | | | | | | | | | | | |
| | 1 | HLtime_A | | | | | | | | | | | | | | | |
| | 2 | center_time | | | | | | | | | | | | | | | |
| | 3 | LD_OK | | | | | | | | | | | | | | | |
| | 4 | SQRT3 | | | | | | | | | | | | | | | |
| | 5 | MPW | | | | | | | | | | | | | | | |
| | 6 | UA3 | | | | | | | | | | | | | | | |
| | 7 | fault_pinstate | | | | | | | | | | | | | | | |
| Phase B | 0 | htB | | | | | | | | | | | | | | | |
| | 1 | HLtime_B | | | | | | | | | | | | | | | |
| | 2 | UA | | | | | | | | | | | | | | | |
| | 3 | UB | | | | | | | | | | | | | | | |
| | 4 | Ualfa | | | | | | | | | | | | | | | |
| | 5 | Ubeta | | | | | | | | | | | | | | | |
| | 6 | Sector | | | | | | | | | | | | | | | |
| | 7 | max_ht | | | | | | | | | | | | | | | |

**Table 5. svmStd3 Parameter RAM**

| Channel | Parameter | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Phase C | 0 | htC | | | | | | | | | | | | | | | |
| | 1 | HLtime_C | | | | | | | | | | | | | | | |
| | 2 | T_copy | | | | | | | | | | | | | | | |
| | 3 | dec | | | | | | | | | | | | | | | |
| | 4 | T | | | | | | | | | | | | | | | |
| | 5 | prescaler | | | | | | | | | | | | | | | |
| | 6 | prsc_copy | | | | | | | | | | | | | | | |
| | 7 | sync_presc_addr | | | | | | | | | | | | | | | |

**Table 6. svmStd3 parameter description**

| Parameter | Format | Description |
|---|---|---|
| Parameters written by CPU | | |
| Ualfa, Ubeta | 16-bit fractional | Stator Reference Voltage Vector components |
| 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 |
| MPW | 16-bit unsigned integer | Minimum pulse width in number of TCR1 TPU cycles. See **Performance** for details. |
| SQRT3 | 16-bit fractional | sqrt(3)/2 = 0.866 = $6EDA constant |
| sync_presc_addr | 8-bit unsigned integer | address of synchronization channel *prescaler* parameter: $X4, where X is synchronization channel number.<br>$0 if no synchronization channel is used. |
| Parameters written by both TPU and CPU | | |
| LD_OK | 1-bit | 0 ... CPU can update variables<br>1 ... TPU can read variables<br>CPU sets 1, TPU sets 0 |
| Parameters written by TPU | | |
| Sector | 16-bit unsigned integer | The position of Stator Reference Voltage Vector in a sector. The Sector can be 1, 2, 3, 4, 5 or 6 |
| fault_pinstate | 0 or 1 | If fault channel is used, state of fault pin:<br>0 ... low<br>1 ... high |
| Other parameters are just for TPU function inner use. | | |

*Performance*

**Table 7. svmStd3 State Statistics**

| State | Max IMB Clock Cycles | RAM Accesses by TPU |
|---|---|---|
| INIT | 66 | 19 |
| STOP | 20 | 0 |
| LH | 26 | 5 |
| HL | 2 | 1 |
| HL_RLD | 44 | 16 |
| C1 | 48 | 3 |
| C2 | 48 | 4 |
| C3 | 50 | 3 |
| C4 | 48 | 8 |

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

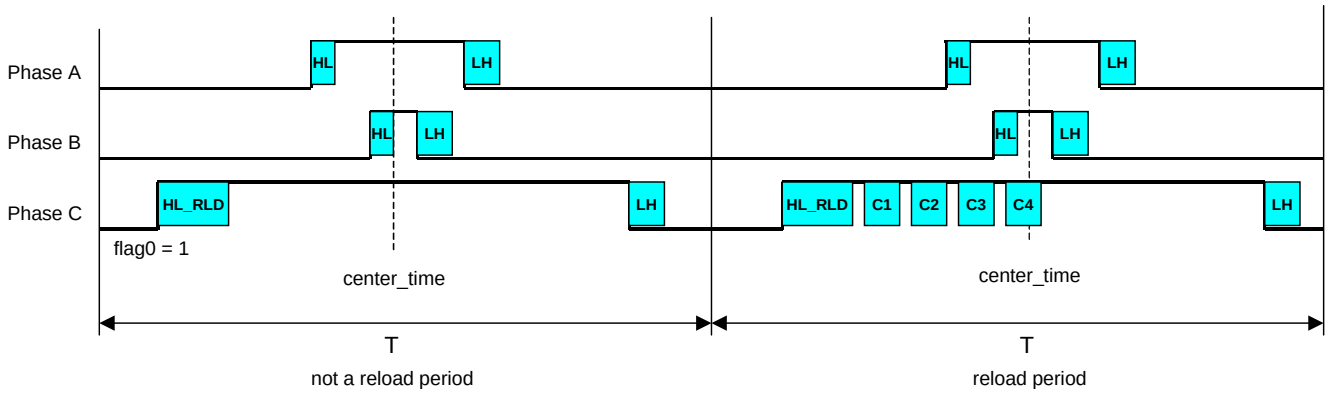



**Figure 3. svmStd3 timing**

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

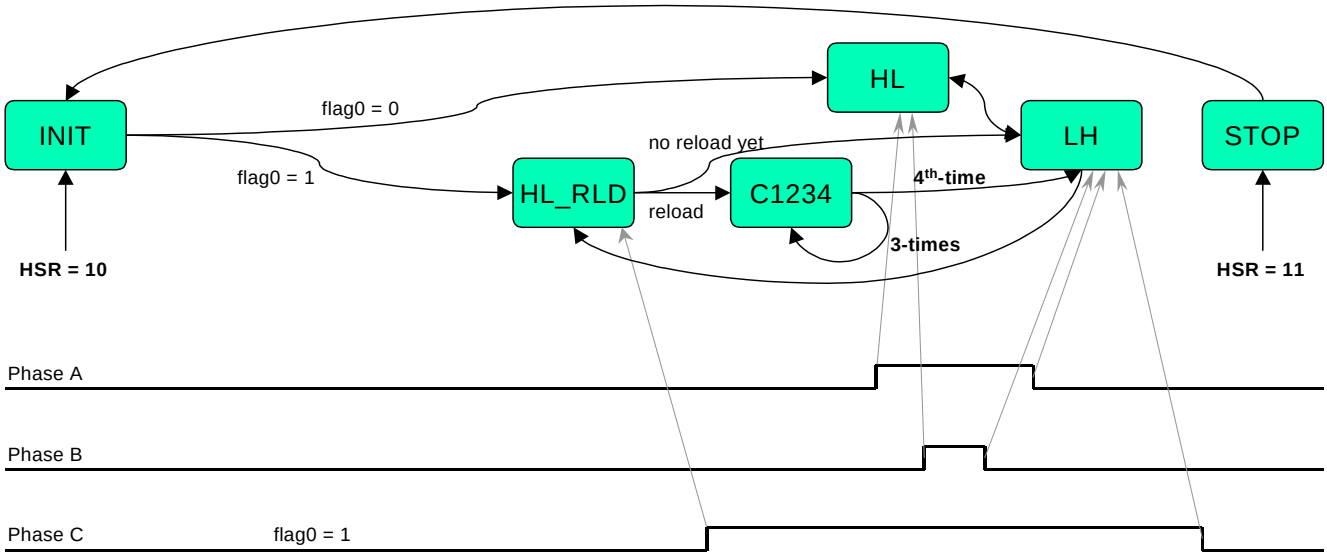



**Figure 4. svmStd3 state diagram**

*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 svmStd3 limits the narrowest generated pulse widths to *MPW*. The CPU program should check, and limit, the maximum amplitude of the Stator Reference Voltage Vector before decomposition to $u_{\acute{a}}$, $u_{\hat{a}}$ components. 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**.

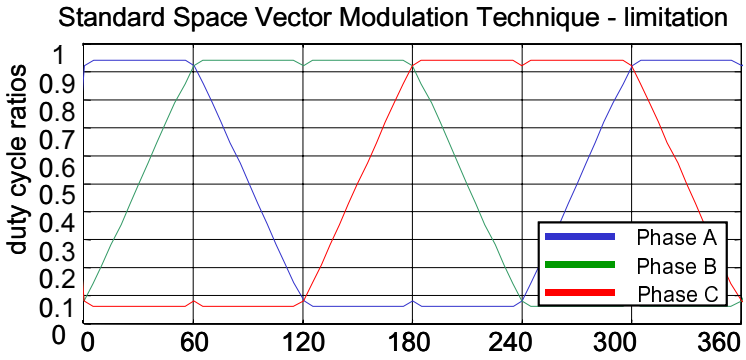



**Figure 5. Effect of limitation**

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 svmStd3 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 state that sets the following low to high transition. The LH 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 svmStd3 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 together, on the same TPU, with the svmStd3 functions, a latency between the high-low transition and the start of the LH state can appear. To maintain sufficiently high performance of svmStd3, it is recommended that the following rules are followed to configure the TPU:

- assign svmStd3 PWM channels high priority
- assign svmStd3 PWM functions on low channel numbers so that no other function with high priority is assigned a channel with a lower number

In this instance, the worst case timing case that can happen is illustrated in **Figure 7**.

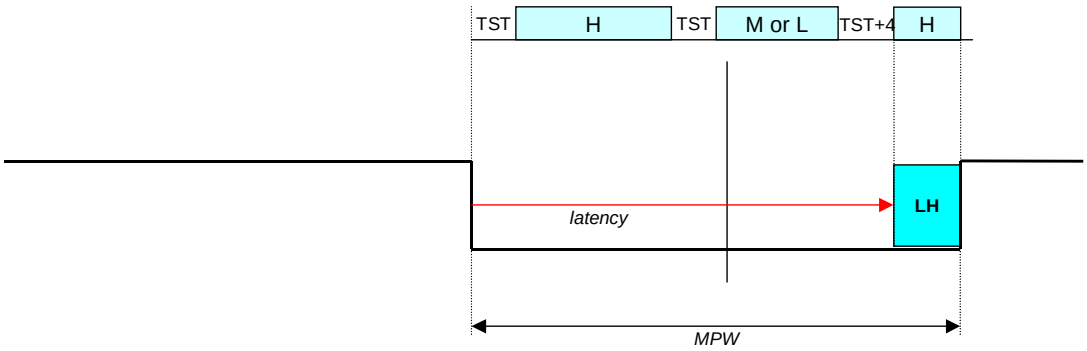



**Figure 7. Worst case timing**

The time slot sequences at the top of the figure shows when a state of a high (H), middle (M) or low (L) priority is serviced in the worst case. To calculate the *MPW* follow these steps:

- Get the lengths of the longest states.
  - It is necessary to know the lengths of the longest states within all functions of each priority group. The initialization states are not considered – only the running states. Let's denote *H* as the time period of the longest state within all functions running on high priority (Do not consider svmStd3 functions). Let's denote *M* as the time period of the longest state within all functions running on middle priority and *L* as the time period of the longest state within all functions running on low priority.
- Calculate *MPW* according to **Figure 7**.
  - TST + *H* + TST + *max*(*M*,*L*) + TST+4 + LH
    that is 60 + *H* + *max*(*M*,*L*) IMB clock cycles.

    *MPW* (in IMB clock cycles) = 60 + *H* + *max*(*M*,*L*)

- Convert *MPW* in IMB clock cycles to *MPW* in TCR1 clock cycles based on TCR1 prescaler settings.

When there are no channels of middle or low priority, simply leave out all the *H* or *L* and the following TST or TST+4 from the formulas.

When the recommended configuration rules are not adhered to, the timing of the worst case is much more complicated. It requires some familiarity with the details of the TPU priority scheme. In this case, the Worst-Case Latency (WCL), which is automatically calculated by the MPC500_Quick_Start Graphical Configuration Tool, can serve as a good approximation. This is always longer than the real-case is. Let the WCL be calculated after the configuration of the TPU channels and then find the longest WCL value within all svmStd3 PWM channels. Convert the number, from IMB clock cycles to TCR1 clock cycles, to get the *MPW*.

**Synchronization signal for Standard Space Vector Modulation – 3 outputs version (svmStd3_sync)**

The svmStd3_sync TPU function uses information obtained from the svmStd3 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.

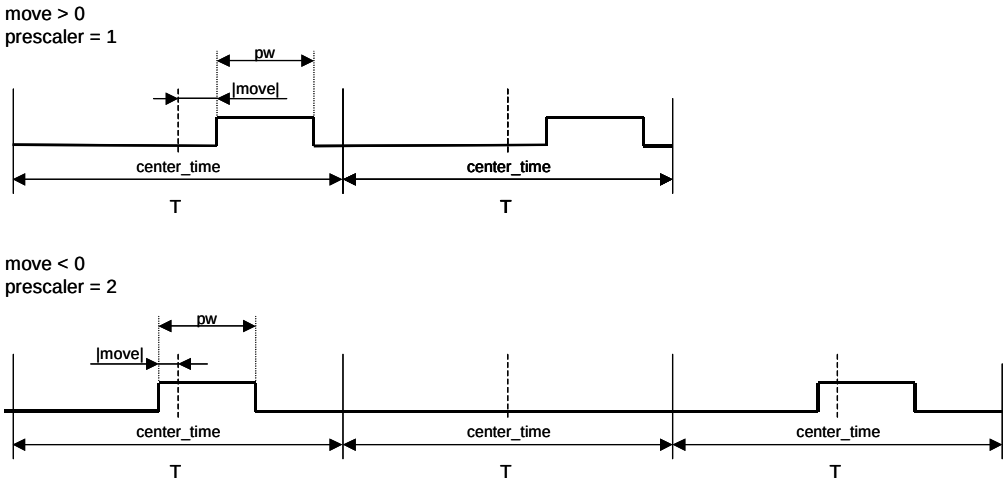



**Figure 8. Synchronization signal adjustment examples**

*Synchronized Change of PWM Prescaler And Synchronization Signal Prescaler*

The svmStd3_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 svmStd3_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*

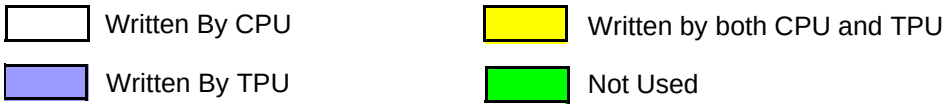



**Table 8. svmStd3_sync Control Bits**

| Name | Options |
|---|---|
| 3 2 1 0 Channel Function Select | svmStd3_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 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 svmStd3_sync generates an interrupt after each low to high transition.

**Table 9. svmStd3_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. svmStd3_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<br>– use in case of synchronized prescalers change |
| presc_copy | 16-bit unsigned integer | The number of PWM periods per synchronization pulse<br>– 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. svmStd3_sync State Statistics**

| State | Max IMB Clock Cycles | RAM Accesses by TPU |
|---|---|---|
| INIT | 12 | 5 |

**Table 11. svmStd3_sync State Statistics**

| State | Max IMB Clock Cycles | RAM Accesses by TPU |
|---|---|---|
| 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)*

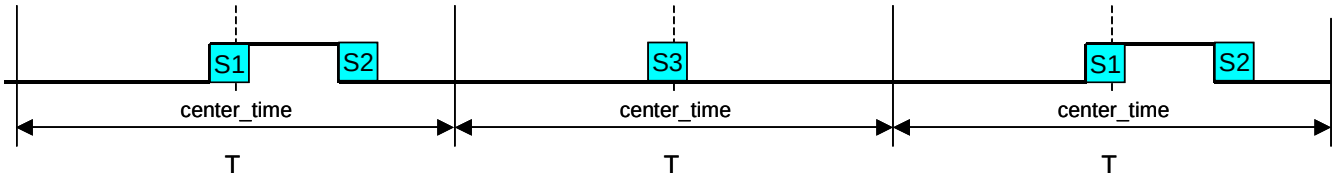



**Figure 9. svmStd3_sync timing**

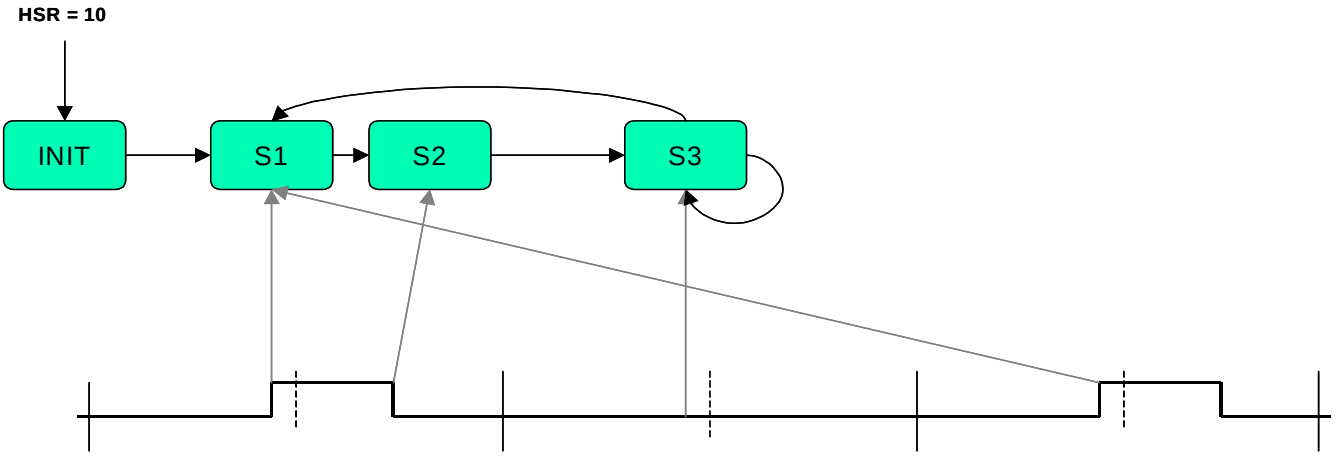



**Figure 10. svmStd3_sync state diagram**

**Resolver Reference Signal for Standard Space Vector Modulation – 3 outputs version (svmStd3_res)**

The svmStd3_res TPU function uses information read from the svmStd3 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.

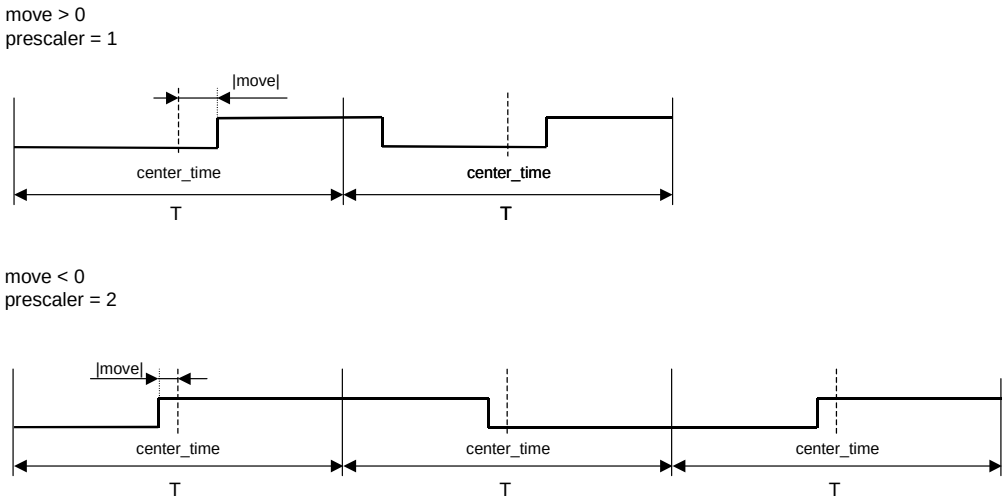



**Figure 11. Resolver reference signal adjustment examples**

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

The svmStd3_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. svmStd3_res Control Bits**

| Name | Options |
|---|---|
| 3 2 1 0<br>Channel Function Select | svmStd3_res function number (Assigned during assembly the DPTRAM code from library TPU functions) |
| 1 0<br>Channel Priority | 00 – Channel Disabled<br>01 – Low Priority<br>10 – Middle Priority<br>11 – High Priority |
| 1 0<br>Host Service Bits (HSR) | 00 – No Host Service Request<br>01 – Not used<br>10 – Initialization<br>11 – Not used |

**Table 12. svmStd3_res Control Bits**

| Name | Options |
|---|---|
| 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. svmStd3_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 14. svmStd3_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 | | |

**Table 14. svmStd3_res 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 15. svmStd3_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)*

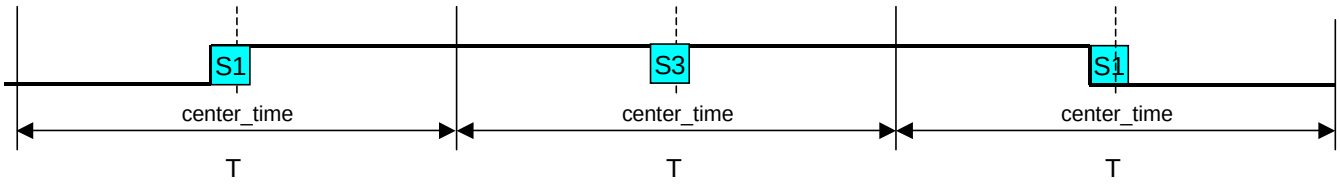



**Figure 12. svmStd3_res timing**

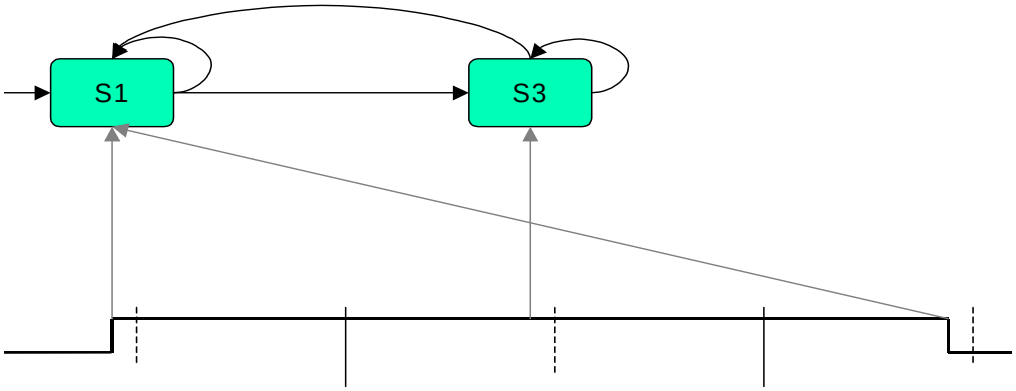



**Figure 13. svmStd3_res state diagram**

**Fault Input for Standard Space Vector Modulation – 3 outputs version (svmStd3_fault)**

The svmStd3_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. svmStd3_fault Control Bits**

| Name | Options |
|---|---|
| 3 2 1 0<br>Channel Function Select | svmStd3_fault function number (Assigned during assembly the DPTRAM code from library TPU functions) |
| 1 0<br>Channel Priority | 00 – Channel Disabled<br>01 – Low Priority<br>10 – Middle Priority<br>11 – High Priority |

**Table 16. svmStd3_fault Control Bits**

| Name | Options |
|---|---|
| 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 svmStd3_fault generates an interrupt when a high to low transition appears.

**Table 17. svmStd3_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. svmStd3_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 19. svmStd3_fault State Statistics**

| State | Max IMB Clock Cycles | RAM Accesses by TPU |
|---|---|---|
| INIT | 8 | 2 |
| FAULT | 26 | 1 |

**Table 19. svmStd3_fault State Statistics**

| State | Max IMB Clock Cycles | RAM Accesses by TPU |
|---|---|---|
| NO_FAULT | 4 | 1 |

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


**Figure 14. svmStd3_fault timing**

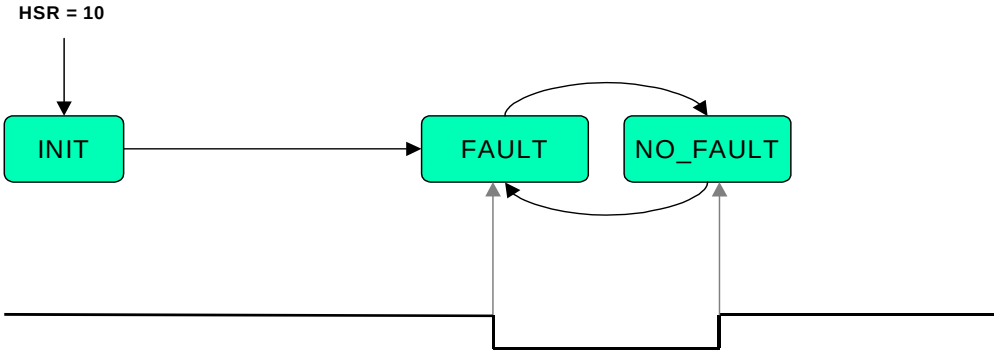



**Figure 15. svmStd3_fault state diagram**

## *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





AN2532/D
Rev. 0
5/2003