Application Note

AN2525/D
Rev. 0, 5/2003

DC Motor – XOR version TPU Function Set (DCmXor)

By Milan Brejl, Ph.D.

Functional Overview

The DC Motor – XOR version (DCmXor) TPU function is a version of the DC Motor (DCm) function that uses two TPU channels to generate one PWM output channel. The TPU channel outputs are connected to an XOR gate whos output is the required PWM signal. See **Figure 1**. An advantage of this solution is that the full range (0% to 100%) of PWM duty-cycle ratios is available. There is no *MPW* (minimum pulse width) parameter to limit the edge duty-cycle ratios in this version, unlike in the DCm. A disadvantage is that the number of assigned TPU channels is doubled.

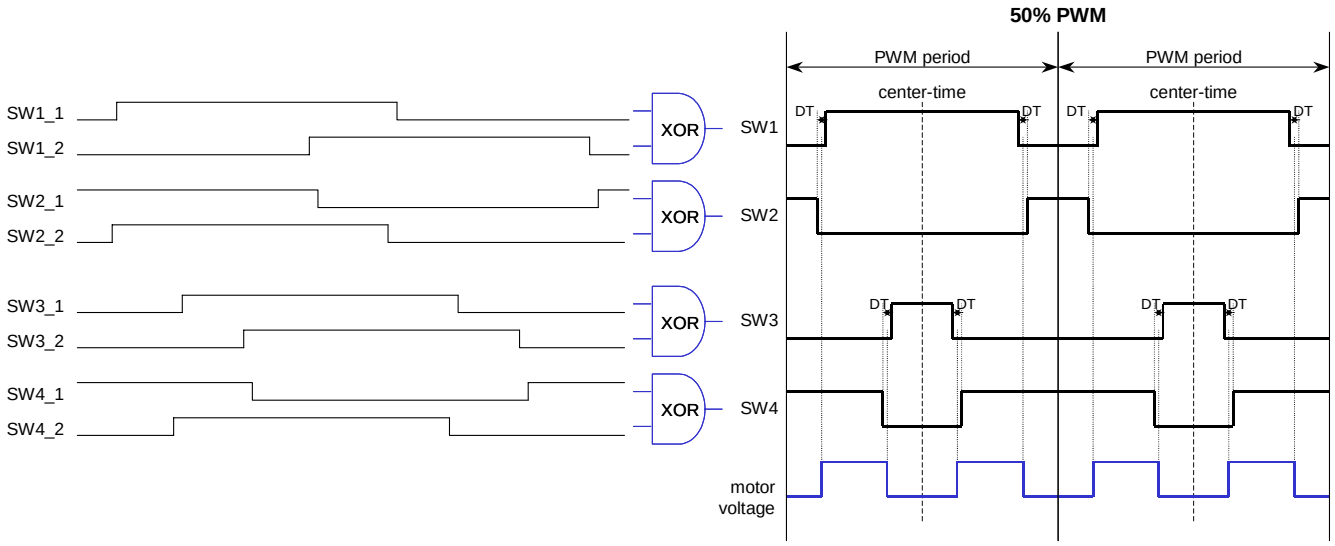

Figure 1. Functionality of XOR version – illustration

The function set consists of 5 TPU functions:

- DC Motor – XOR version – C channels (DCmXor_C)
- DC Motor – XOR version – T channels (DCmXor_T)
- Synchronization Signal for DC Motor – XOR version (DCmXor_sync)
- Resolver Reference Signal for DC Motor – XOR version (DCmXor_res)
- Fault Input for DC Motor – XOR version (DCmXor_fault)

The DCm TPU function set drives a DC Motor, independently of the CPU. The CPU is required only to set a duty-cycle (*dc*) parameter in the range (–1,1). This determines both the speed and the direction. The function generates unipolar-switched center-aligned PWM signals.

The DCmXor_C and DCmXor_T TPU functions work together to generate 4 pairs of XOR gate inputs. The XOR gate outputs then produce a 4-channel 2-phase center-aligned PWM signal with dead-time between the top and bottom channels. The Synchronization Signal for the DCmXor function can be used to generate one or more adjustable signals for a wide range of uses. These are synchronized to the PWM, and track changes in the PWM period. The Resolver Reference Signal for the DCmXor 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 DCmXor function is a TPU input function that sets all PWM outputs low when the input signal goes low.

Function Set Configuration

None of the TPU functions in the DC Motor – XOR version TPU function set can be used separately. The DCmXor_C and DCmXor_T functions have to be used together. The DCmXor_C runs on pins SW1_1 and SW3_1 – see **Figure 1**. The DCmXor_T runs on the other pins. One or more channels running Synchronization Signal for DCmXor as well as Resolver Reference Signals for DCmXor functions can be added. They can run with different settings on each channel. The function Fault Input for DCmXor can also be added. 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.

Table 1 shows the configuration options and restrictions.

Table 1. DCmXor TPU function set configuration options and restrictions

TPU function	Optional/ Mandatory	How many channels	Assignable channels
DCmXor_C	mandatory	2	any 2 channels
DCmXor_T	mandatory	6	any 6 channels
DCmXor_sync	optional	1 or more	any channels
DCmXor_res	optional	1 or more	any channels
DCmXor_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	DCmXor_C	high
1	DCmXor_T	high
2	DCmXor_T	high
3	DCmXor_T	high
4	DCmXor_C	high
5	DCmXor_T	high
6	DCmXor_T	high
7	DCmXor_T	high
10	DCmXor_sync	low
11	DCmXor_res	low
15	DCmXor_fault	high

Table 3 shows the TPU function code sizes.

Table 3. TPU function code sizes

TPU function	Code size
DCmXor_C	134µ instructions + 8 entries = 142 long words
DCmXor_T	3 µ instructions + 8 entries = 11 long words
DCmXor_sync	26 µ instructions + 8 entries = 34 long words
DCmXor_res	38 µ instructions + 8 entries = 46 long words
DCmXor_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*, *DT* and *sync_presc_addr* must be set before initialization. If a DCmXor_sync channel or a DCmXor_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 DCmXor_C channels to initialize all DCmXor_C and DCmXor_T channels. Issues an HSR type %10 to the DCmXor_sync channels, DCmXor_res channels and DCmXor_fault channel, if used.
5. Enables servicing by assigning high, middle or low priority to the channel priority bits. All DCmXor_C and DCmXor_T channels must be assigned the same priority to ensure correct operation. The CPU must ensure that the DCmXor_sync or DCmXor_res function is initialized after the initialization of DCmXor:
   - assign a priority to the DCmXor_C and DCmXor_T channels to enable their initialization
   - if a Synchronization Signal or a Resolver Reference Signal channel is used, wait until the HSR bits are cleared to indicate that initialization of the DCmXor_C and DCmXor_T channels has completed and
   - assign a priority to the DCmXor_sync or DCmXor_res channel to enable its initialization

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

Detailed Function Description

DC Motor – XOR version – C channels (DCmXor_C) and DC Motor – XOR version – T channels (DCmXor_T)

The DCmXor_C and DCmXor_T TPU functions work together to generate 4 pairs of XOR gate inputs. The XOR gate outputs then produce a 4-channel 2-phase center-aligned PWM signal with dead-time between the top and bottom channels. In order to charge the bootstrap transistors, the PWM signals start to run 1.6ms after their initialization (at 20MHz TCR1 clock). The functions generate signals corresponding to a value 0 in duty-cycle ratio *dc* until the first *dc* value is processed, or for at least one PWM period.

The CPU controls the PWM output by setting the TPU parameters. The duty-cycle ratio *dc* and PWM period *T* can be adjusted during run time. Conversely, dead-time (*DT*) is not supposed to be changed during run time. The duty-cycle ratio *dc* can gain a value in the range (–1, 1). The sign controls the motion system direction, while the absolute value controls the amplitude of the applied voltage.

The following figures show the input *dc* value and corresponding XOR gate outputs:

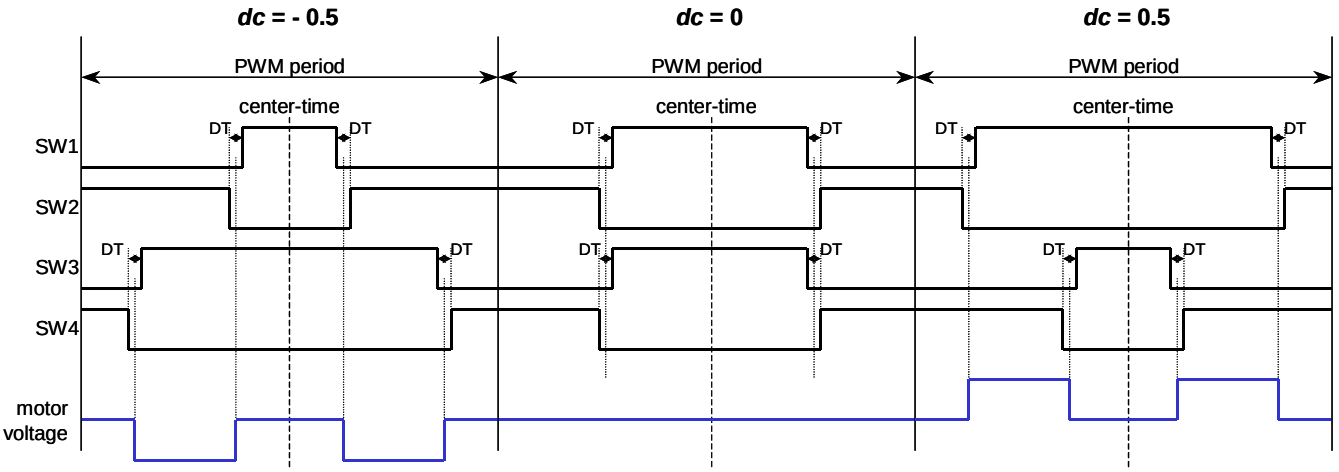

Figure 2. Unipolar switching

The following equations describe how the PWM signal transition times $SW1\_1_T$, $SW1\_2_T$, $SW2\_1_T$, $SW2\_2_T$, $SW3\_1_T$, $SW3\_2_T$, $SW4\_1_T$ and $SW4\_2_T$ are calculated:

$$Tdc = T \cdot dc$$

$$X = \frac{T + Tdc}{2}$$

$$Y = \frac{T - Tdc}{2}$$

$$A = \frac{X - DT}{2} \qquad C = \frac{Y - DT}{2}$$

$$B = \frac{X + DT}{2} \qquad D = \frac{Y + DT}{2}$$

$$SW1\_1_T = center\_time - A \qquad SW1\_2_T = center\_time + A$$

$$SW2\_1_T = center\_time - B \qquad SW2\_2_T = center\_time + B$$

$$SW3\_1_T = center\_time - C \qquad SW3\_2_T = center\_time + C$$

$$SW4\_1_T = center\_time - D \qquad SW4\_2_T = center\_time + D$$

Host Interface

Written By CPU

Written by both CPU and TPU

Written By TPU

Not Used

Table 4. DCmXor_C Control Bits

Name	Options
3 2 1 0 Channel Function Select	DCmXor_C 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	x – Not used
0 Channel Interrupt Status	x – Not used

Table 5. DCmXor_T Control Bits

Name	Options
3 2 1 0 Channel Function Select	DCmXor_T 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 – Not used 11 – Not used
1 0 Host Sequence Bits (HSQ)	xx – Not used

Table 5. DCmXor_T Control Bits

Name	Options
0 Channel Interrupt Enable	x – Not used
0 Channel Interrupt Status	x – Not used

Table 6. DCmXor_C and DCmXor_T Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SW1_1	0	XY_X															
	1	SW13_2_ch_SW1															
	2	SW24_1_ch_SW1															
	3	SW24_2_ch_SW1															
	4	dc															
	5	T															
	6	other_ch_SW1															
	7	fault_pinstate															
SW1_2	0	Ttime_SW1_2															
	1	T_copy															
	2	L															
	3	center_time															
	4	DT															
	5	CPU14															
	6																
	7																
SW2_1	0	Ttime_SW2_1															
	1																
	2																
	3																
	4																
	5	sync_presc_addr															
	6																
	7																
SW2_2	0	Ttime_SW2_2															
	1																
	2																
	3																
	4																
	5																
	6																
	7																

Table 6. DCmXor_C and DCmXor_T Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SW3_1	0	XY_Y															
	1	SW13_2_ch_SW3															
	2	SW24_1_ch_SW3															
	3	SW24_2_ch_SW3															
	4																
	5																
	6	other_ch_SW3															
	7																
SW3_2	0	Ttime_SW3_2															
	1																
	2																
	3																
	4																
	5																
	6																
	7																
SW4_1	0	Ttime_SW4_1															
	1																
	2																
	3																
	4																
	5																
	6																
	7																
SW4_2	0	Ttime_SW4_2															
	1																
	2																
	3																
	4																
	5																
	6																
	7																

Table 7. DCmXor_C and DCmXor_T parameter description

Parameter	Format	Description
Parameters written by CPU		
dc	16-bit fractional	duty-cycle ratio in the range <–1,1)
T	16-bit unsigned integer	PWM period in number of TCR1 TPU cycles
DT	16-bit unsigned integer	Dead-time in number of TCR1 TPU cycles
CPU14	16-bit unsigned integer	Time of 14 IMB clocks in TCR1 clocks.
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 TPU		
fault_pinstate	0 or 1	If fault channel is used, state of fault pin: 0 ... low 1 ... high
Other parameters are just for TPU function inner use.		

Performance

Table 8. DCmXor_T State Statistics

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

Table 9. DCmXor_C State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	88	16
STOP	100	1
C1	76	13
C2	28	9

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

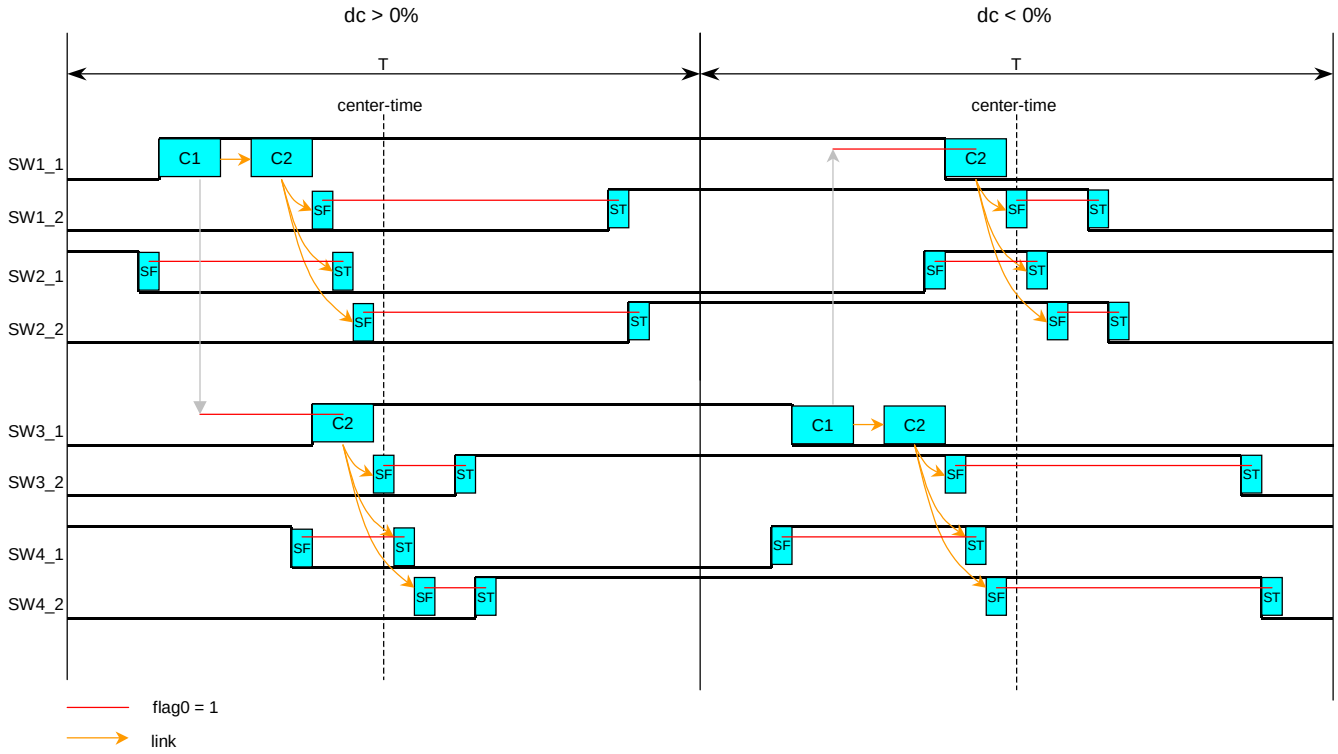

Figure 3. DCmXor_C and DCmXor_T timing

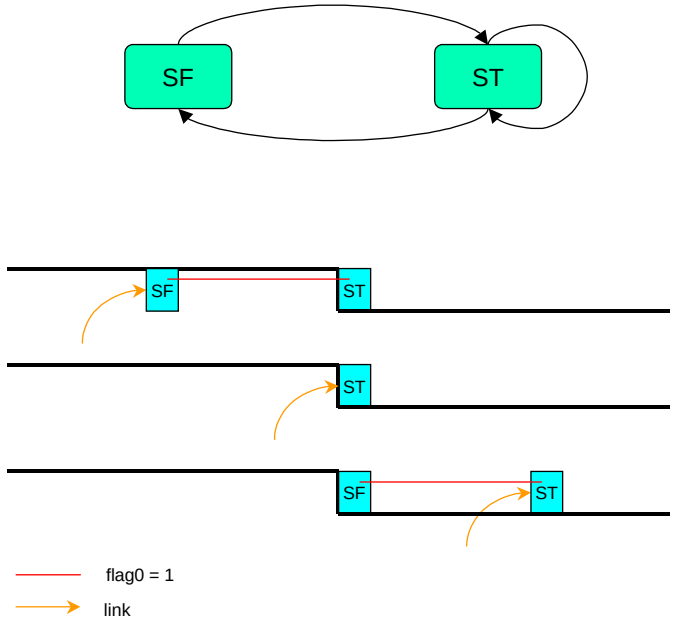

Figure 4. DCmXor_T state diagram and 3 cases of timing

NOTE: *The timing of the link determines which case accurs.*

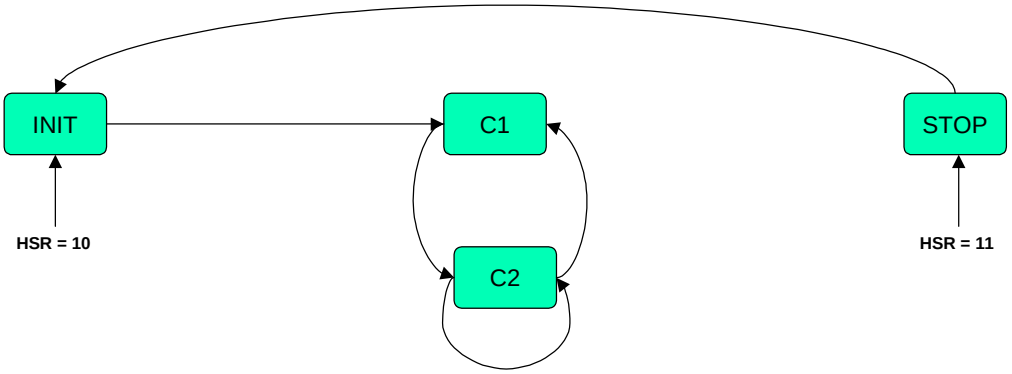

Figure 5. DCmXor_C state diagram

Synchronization signal for DC Motor – XOR version (DCmXor_sync)

The DCmXor_sync TPU function uses information obtained from DCmXor_C and DCmXor_T functions, the actual PWM center times and the PWM periods. This allows a signal to be generated, that 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 before or after the PWM period center time of a number of TCR1 TPU cycles. The pulse width *pw* is another synchronization signal parameter.

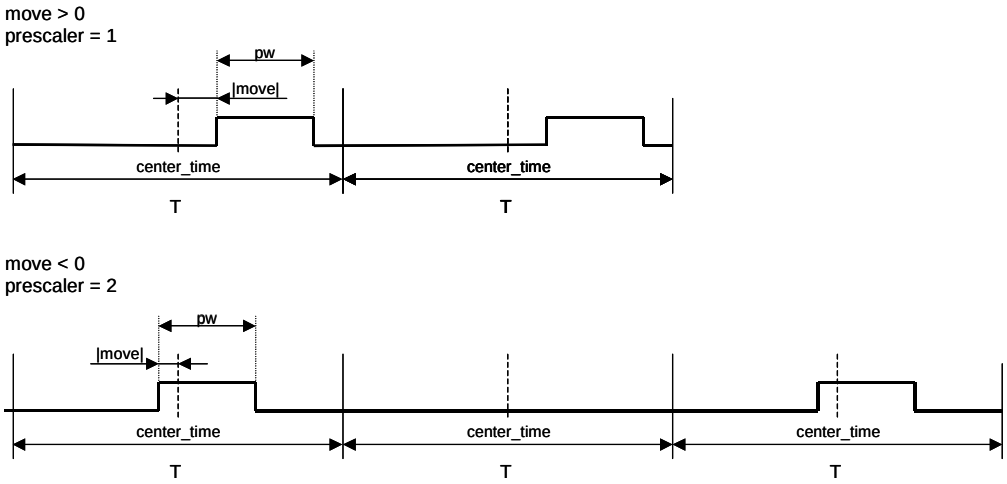

Figure 6. Synchronization signal adjustment examples

Synchronized Change of PWM Prescaler And Synchronization Signal Prescaler

The DCmXor_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 DCmXor_bottom function at the time of the PWM parameters reload. 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 signals *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 10. DCmXor_sync Control Bits

Name	Options
3 2 1 0 Channel Function Select	DCmXor_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 DCmXor_sync generates an interrupt after each low to high transition.

Table 11. DCmXor_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 12. DCmXor_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 then a quarter of the PWM period *T*.

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

Table 13. DCmXor_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)*

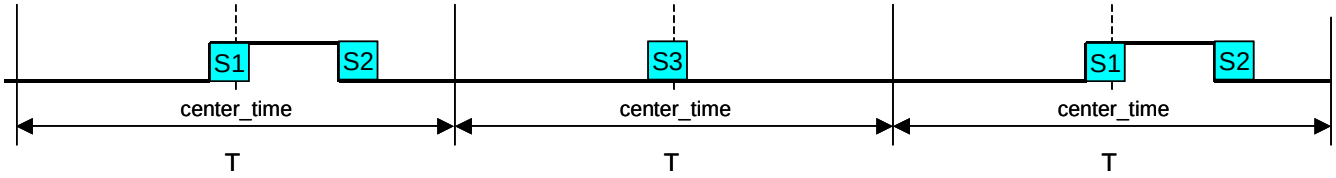

Figure 7. DCmXor_sync timing

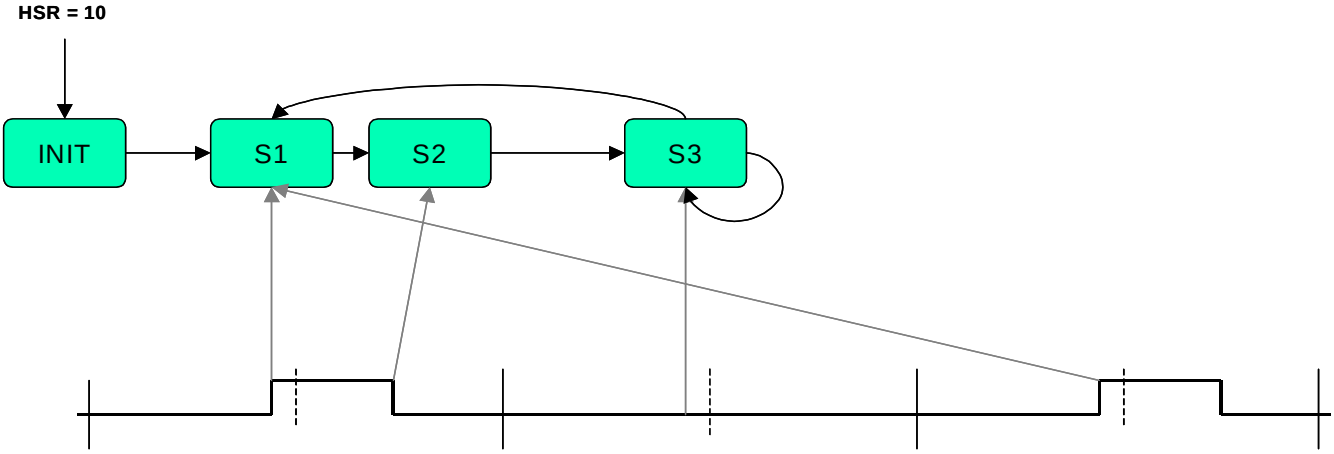

Figure 8. DCmXor_sync state diagram

Resolver Reference Signal for DC Motor – XOR version (DCmXor_res)

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

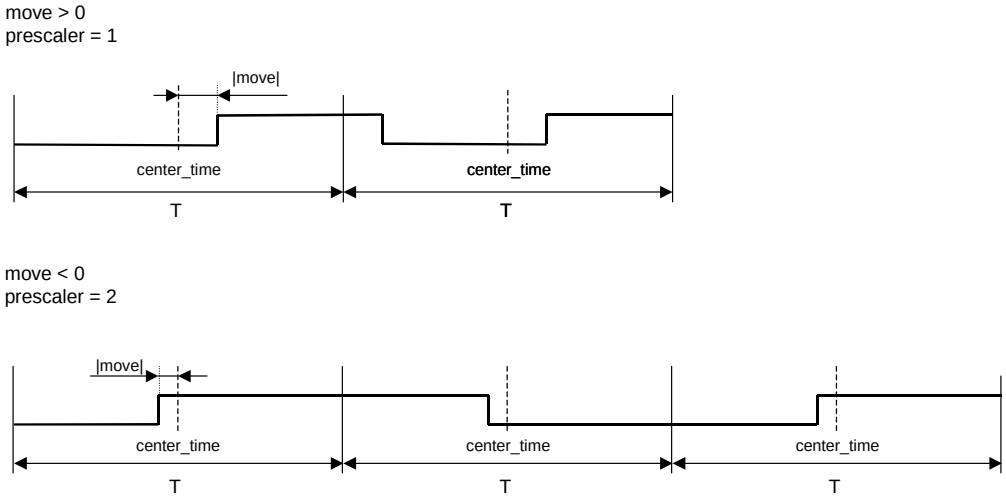

Figure 9. Resolver reference signal adjustment examples

Synchronized Change of PWM Prescaler And Resolver Reference Signals Prescaler

The DCmXor_res TPU function can inherit the Synchronization Signal prescaler that is synchronously changed with 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 *prescaler* parameter to directly specify prescaler value.

Host Interface

Written By CPU

Written by both CPU and TPU

Written By TPU

Not Used

Table 14. DCmXor_res Control Bits

Name	Options
3 2 1 0 Channel Function Select	DCmXor_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 15. DCmXor_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 16. DCmXor_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 17. DCmXor_res State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	12	5
S1	26	9
S3	16	7

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

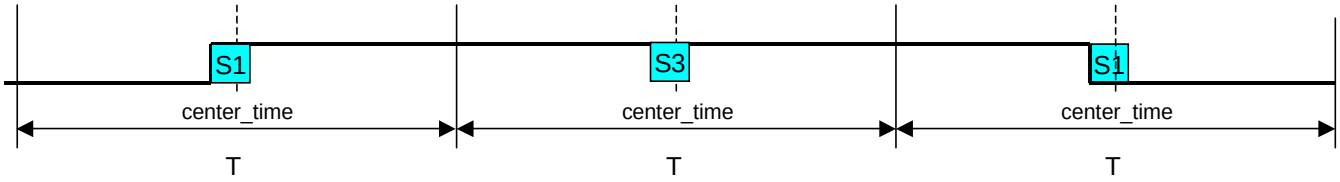

Figure 10. DCmXor_res timing

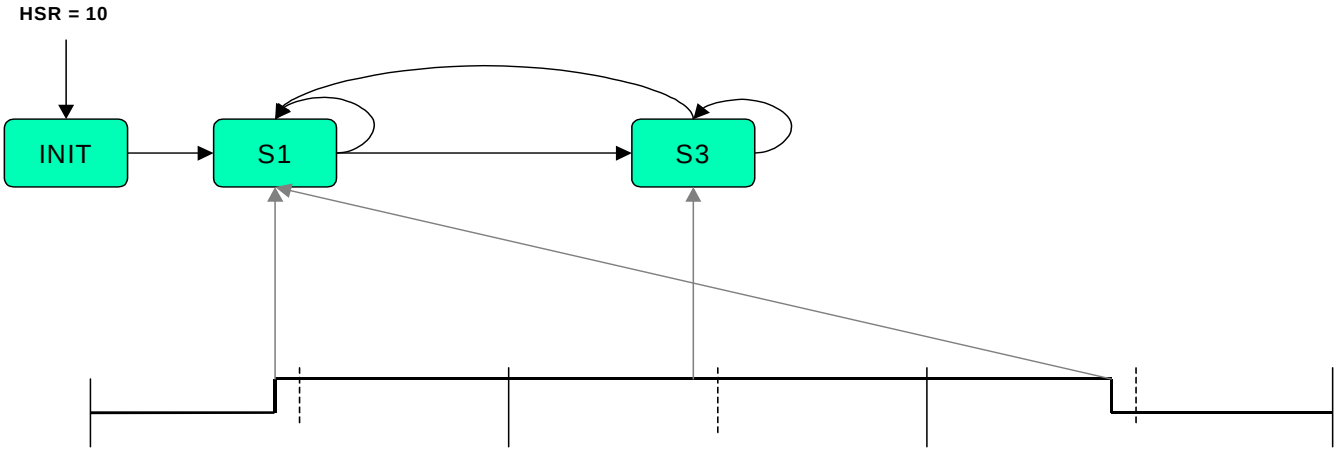

Figure 11. DCmXor_res state diagram

Fault Input for DC Motor – XOR version (DCmXor_fault)

The DCmXor_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 SW1_1 channel to keep the fault channel parameter space free.

Host Interface

Written By CPU

Written by both CPU and TPU

Written By TPU

Not Used

Table 18. DCmXor_fault Control Bits

Name	Options
3 2 1 0 Channel Function Select	DCmXor_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 DCmXor_fault generates an interrupt when a high to low transition appears.

Table 19. DCmXor_fault Parameter RAM

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

Table 20. DCmXor_fault parameter description

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

Performance

Table 21. DCmXor_fault State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	8	2
FAULT	106	2
NO_FAULT	4	1

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

Figure 12. DCmXor_fault timing

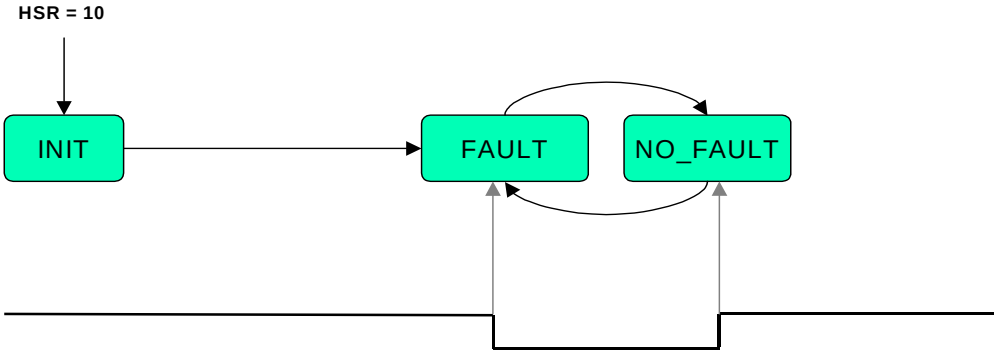

Figure 13. DCmXor_fault state diagram

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

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

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

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

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

freescale™
semiconductor

AN2525/D
Rev. 0
5/2003