Application Note

AN2527/D
Rev. 0, 5/2003

DC Motor with Dead-Time Correction – XOR version TPU Function Set (DCmDtXor)

By Milan Brejl, Ph.D.

Functional Overview

The DC Motor with Dead-Time Correction – XOR version (DCmDtXor) TPU function is a version of the DC Motor with Dead-Time Correction (DCmDt) 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 DCmDt. A disadvantage is that the number of assigned TPU channels is doubled.

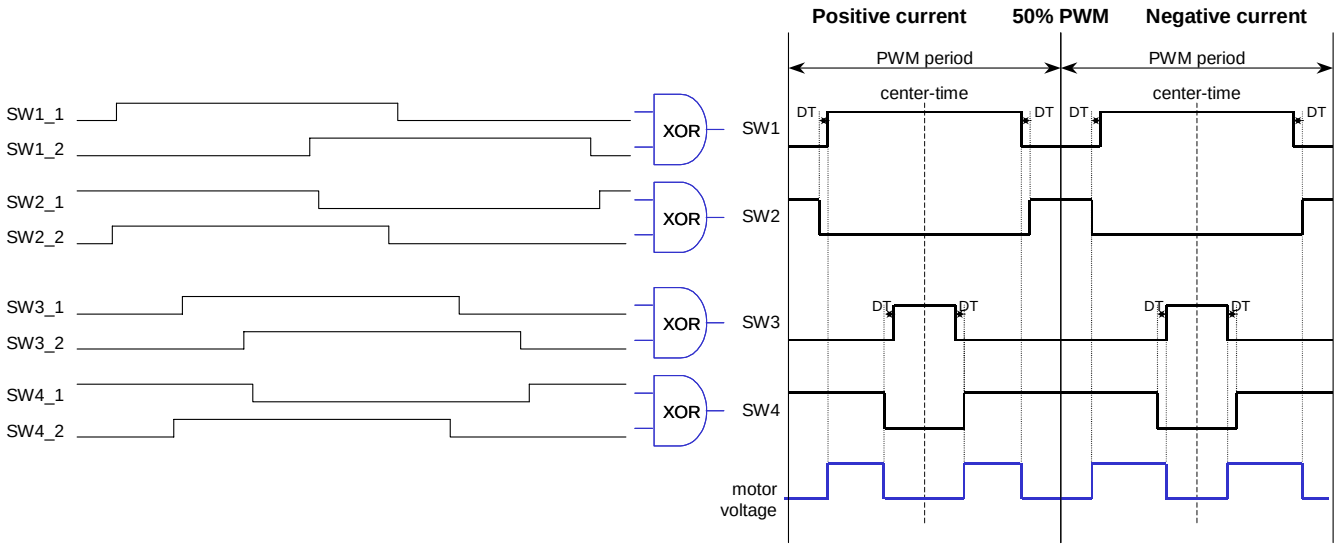

Figure 1. Functionality of XOR version – illustration

The dead-time correction technique requires knowledge the instantaneous direction of the motor current. In case of positive motor current, the SW1 high-time and SW4 low-time are equal to the calculated high-times, and the SW2 and SW3 channels control the dead-time. In the case of negative motor current, the SW2 low-time and SW3 high-time are equal to the calculated high-times, and the SW1 and SW4 channels control the dead-time. See **Figure 2**.

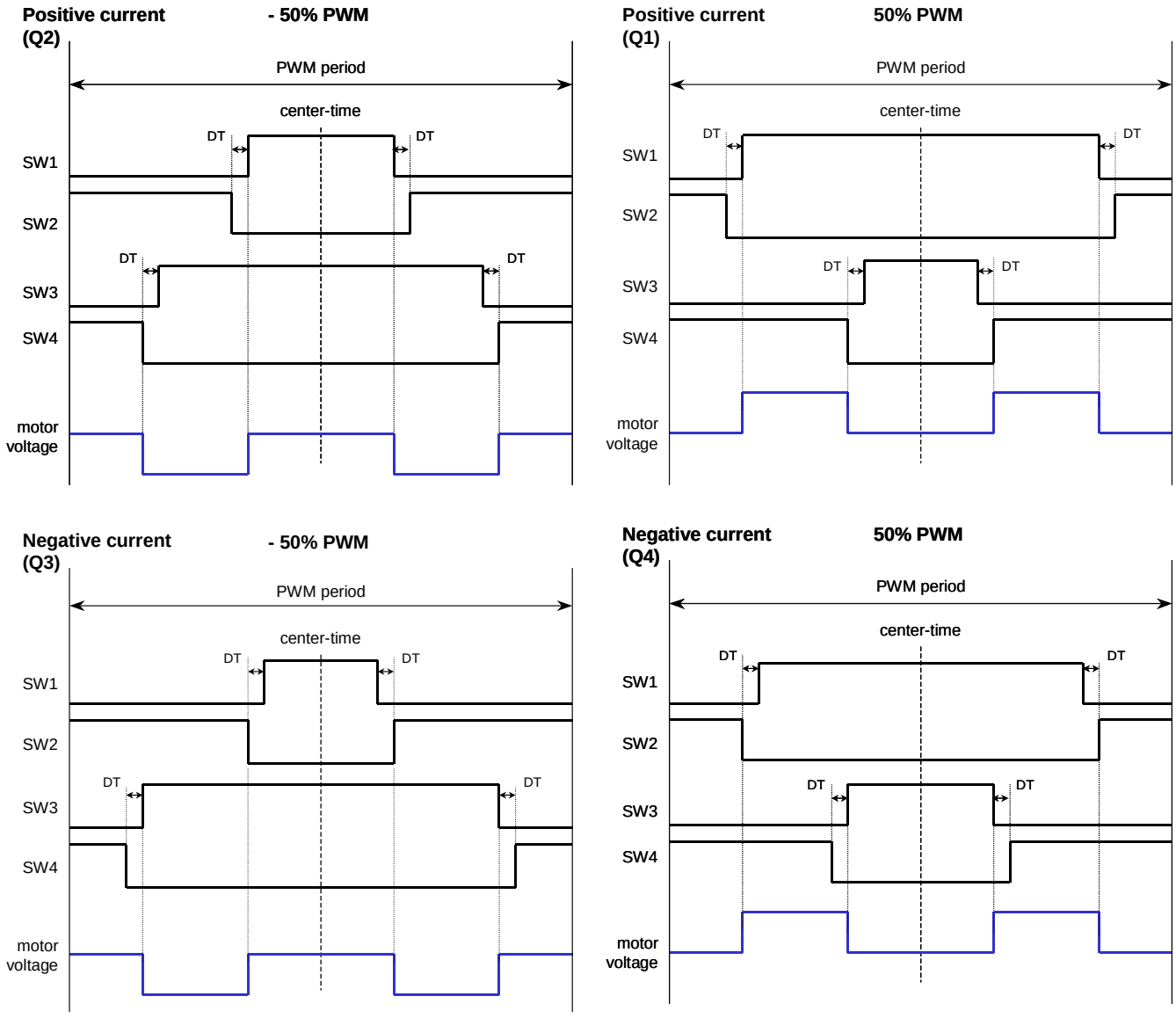

Figure 2. Dead-Time Correction Technique

The function set consists of 5 TPU functions:

- DC Motor with Dead-Time Correction – XOR version – C channels (DCmDtXor_C)
- DC Motor with Dead-Time Correction – XOR version – T channels (DCmDtXor_T)
- Synchronization Signal for DC Motor with Dead-Time Correction – XOR version (DCmDtXor_sync)
- Resolver Reference Signal for DC Motor with Dead-Time Correction – XOR version (DCmDtXor_res)
- Fault Input for DC Motor with Dead-Time Correction – XOR version (DCmDtXor_fault)

The DCmDtXor 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 DCmDtXor_C and DCmDtXor_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 DCmDtXor 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 DCmDtXor 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 DCmDtXor 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 with Dead-Time Correction – XOR version TPU function set can be used separately. The DCmDtXor_C and DCmDtXor_T functions have to be used together. The DCmDtXor_C runs on pins SW1_1 and SW3_1 – see **Figure 1**. The DCmDtXor_T runs on the other pins. One or more channels running Synchronization Signal for DCmDtXor as well as Resolver Reference Signals for DCmDtXor functions can be added. They can run with different settings on each channel. The function Fault Input for DCmDtXor 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. DCmDtXor TPU function set configuration options and restrictions

TPU function	Optional/ Mandatory	How many channels	Assignable channels
DCmDtXor_C	mandatory	2	any 2 channels
DCmDtXor_T	mandatory	6	any 6 channels
DCmDtXor_sync	optional	1 or more	any channels
DCmDtXor_res	optional	1 or more	any channels
DCmDtXor_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	DCmDtXor_C	high
1	DCmDtXor_T	high
2	DCmDtXor_T	high
3	DCmDtXor_T	high
4	DCmDtXor_C	high
5	DCmDtXor_T	high
6	DCmDtXor_T	high
7	DCmDtXor_T	high
10	DCmDtXor_sync	low
11	DCmDtXor_res	low
15	DCmDtXor_fault	high

Table 3 shows the TPU function code sizes.

Table 3. TPU function code sizes

TPU function	Code size
DCmDtXor_C	139µ instructions + 8 entries = 147 long words
DCmDtXor_T	3 µ instructions + 8 entries = 11 long words
DCmDtXor_sync	26 µ instructions + 8 entries = 34 long words
DCmDtXor_res	38 µ instructions + 8 entries = 46 long words
DCmDtXor_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 DCmDtXor_sync channel or a DCmDtXor_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 DCmDtXor_C channels to initialize all DCmDtXor_C and DCmDtXor_T channels. Issues an HSR type %10 to the DCmDtXor_sync channels, DCmDtXor_res channels and DCmDtXor_fault channel, if used.
5. Enables servicing by assigning high, middle or low priority to the channel priority bits. All DCmDtXor_C and DCmDtXor_T channels must be assigned the same priority to ensure correct operation. The CPU must ensure that the DCmDtXor_sync or DCmDtXor_res function is initialized after the initialization of DCmDtXor:
   - assign a priority to the DCmDtXor_C and DCmDtXor_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 DCmDtXor_C and DCmDtXor_T channels has completed and
   - assign a priority to the DCmDtXor_sync or DCmDtXor_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 with Dead-Time Correction – XOR version – C channels (DCmDtXor_C) and DC Motor with Dead-Time Correction – XOR version – T channels (DCmDtXor_T)

The DCmDtXor_C and DCmDtXor_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*, PWM period *T* and *current* 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 (valid for positive motor current):

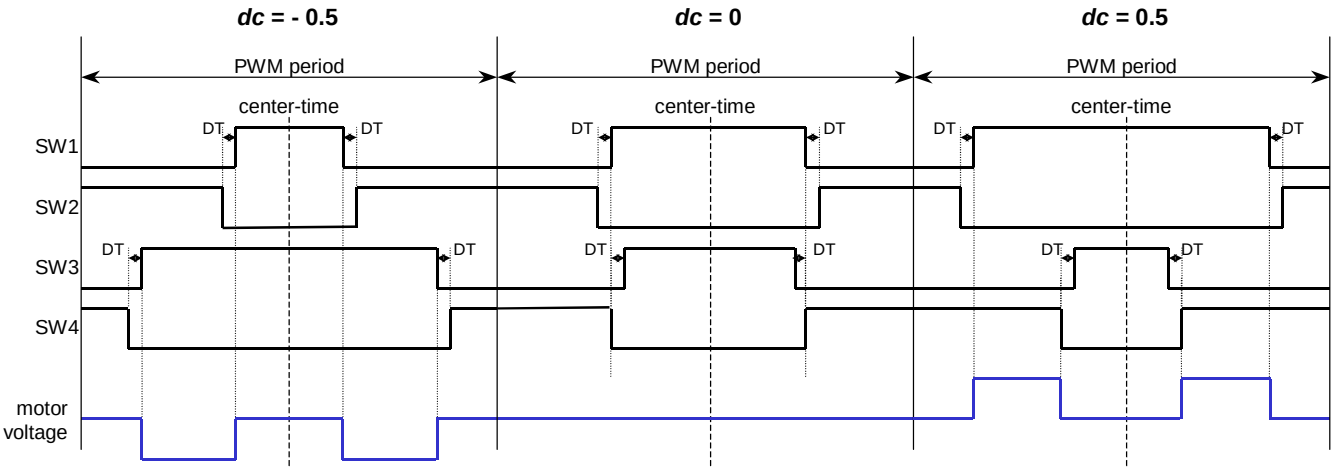

Figure 3. Unipolar switching

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:

$$Tdc = T \cdot dc$$

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

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

Positive current (*current* = 0)

$$A = \frac{X}{2}$$

$$B = \frac{X}{2} + DT$$

$$C = \frac{Y}{2}$$

$$D = \frac{Y}{2} + DT$$

$$SW1\_1_T = center\_time - A$$

$$SW2\_1_T = center\_time - B$$

$$SW3\_1_T = center\_time - C$$

$$SW4\_1_T = center\_time - D$$

Negative current (*current* = 1)

$$A = \frac{X}{2} - DT$$

$$B = \frac{X}{2}$$

$$C = \frac{Y}{2} - DT$$

$$D = \frac{Y}{2}$$

$$SW1\_2_T = center\_time + A$$

$$SW2\_2_T = center\_time + B$$

$$SW3\_2_T = center\_time + C$$

$$SW4\_2_T = center\_time + D$$

Host Interface

Written By CPU

Written by both CPU and TPU

Written By TPU

Not Used

Table 4. DCmDtXor_C Control Bits

Name	Options
3 2 1 0 Channel Function Select	DCmDtXor_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. DCmDtXor_T Control Bits

Name	Options
3 2 1 0 Channel Function Select	DCmDtXor_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. DCmDtXor_T Control Bits

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

Table 6. DCmDtXor_C and DCmDtXor_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	current															
	5	sync_presc_addr															
	6																
	7																
SW2_2	0	Ttime_SW2_2															
	1																
	2																
	3																
	4																
	5																
	6																
	7																

Table 6. DCmDtXor_C and DCmDtXor_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. DCmDtXor_C and DCmDtXor_T parameter description

Parameter	Format	Description
Parameters written by CPU		
dc	16-bit fractional	duty-cycle ratio in the range <–1,1)
current	0 or 1	0 ... positive motor current 1 ... negative motor current
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. DCmDtXor_T State Statistics

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

Table 9. DCmDtXor_C State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	90	16
STOP	100	1
C1	76	13
C2	34	10

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

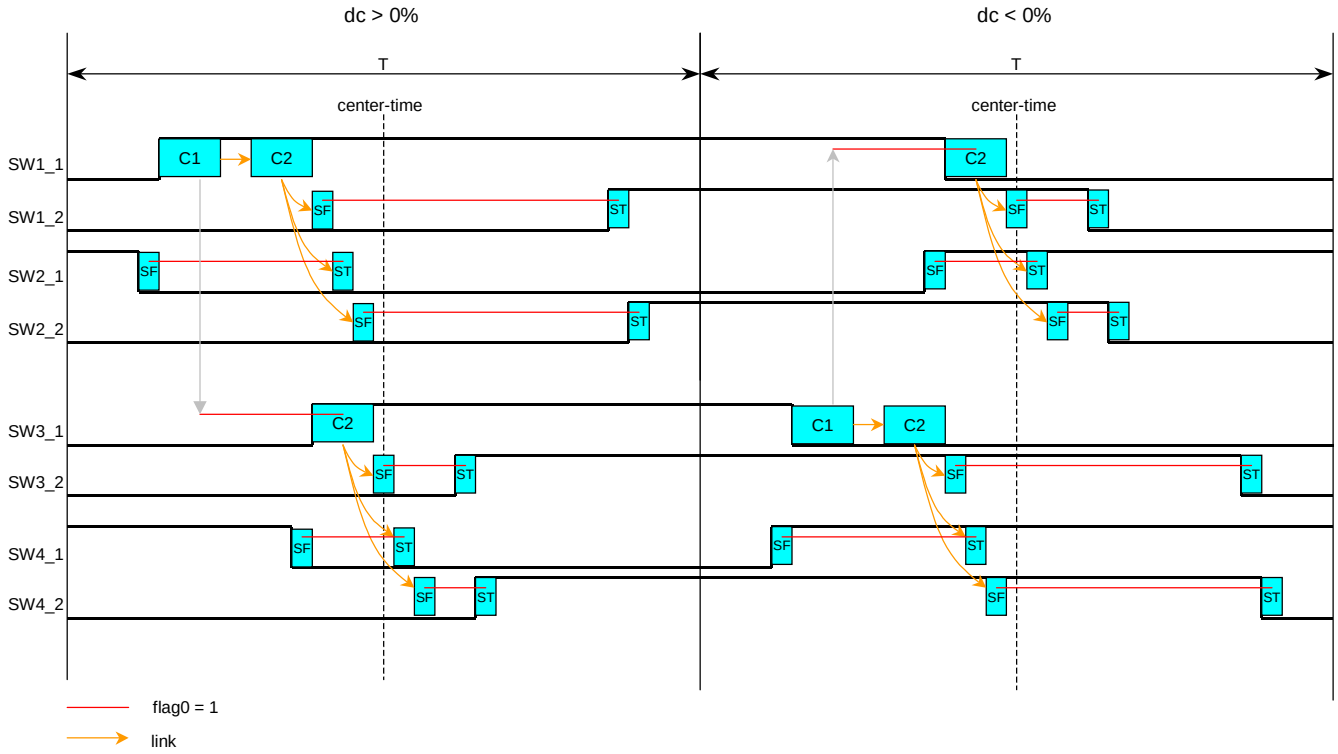

Figure 4. DCmDtXor_C and DCmDtXor_T timing

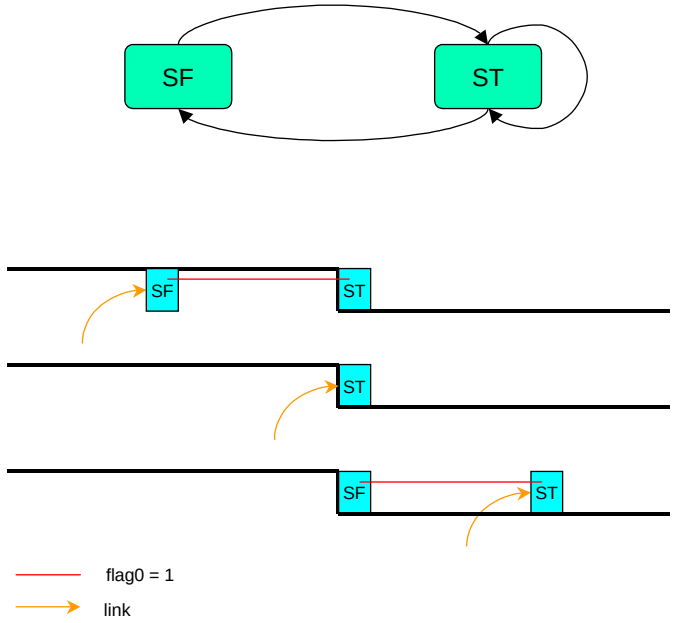

Figure 5. DCmDtXor_T state diagram and 3 cases of timing

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

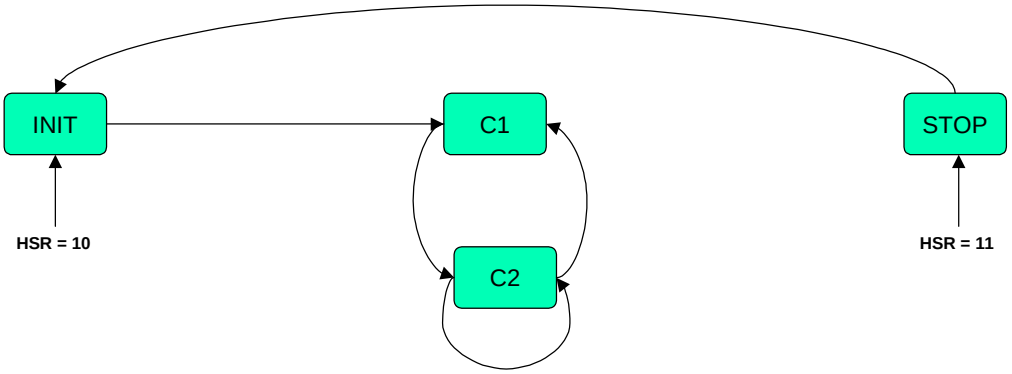

Figure 6. DCmDtXor_C state diagram

Synchronization signal for DC Motor with Dead-Time Correction – XOR version (DCmDtXor_sync)

The DCmDtXor_sync TPU function uses information obtained from DCmDtXor_C and DCmDtXor_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.

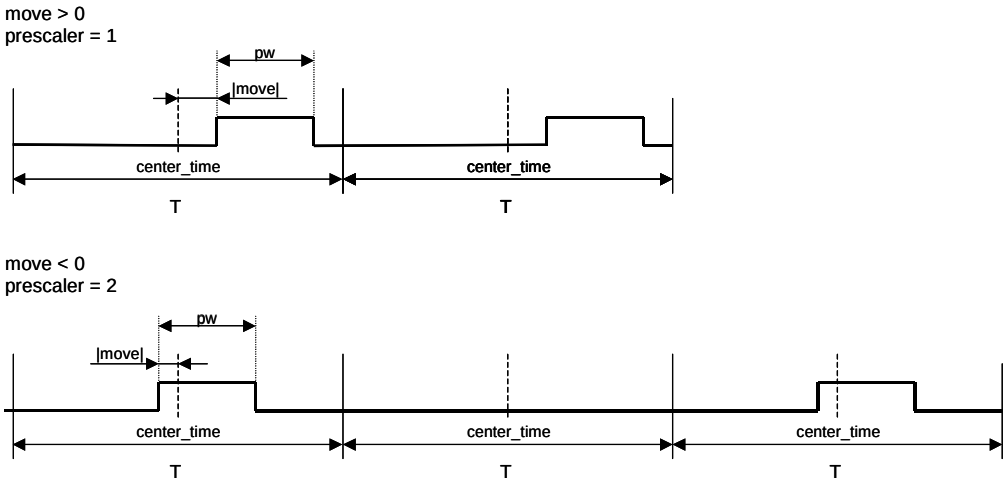

Figure 7. Synchronization signal adjustment examples

Synchronized Change of PWM Prescaler And Synchronization Signal Prescaler

The DCmDtXor_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 DCmDtXor_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. DCmDtXor_sync Control Bits

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

Table 11. DCmDtXor_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. DCmDtXor_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. DCmDtXor_sync State Statistics

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

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

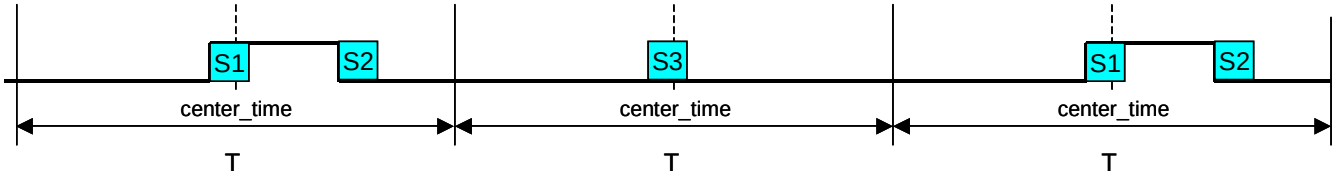

Figure 8. DCmDtXor_sync timing

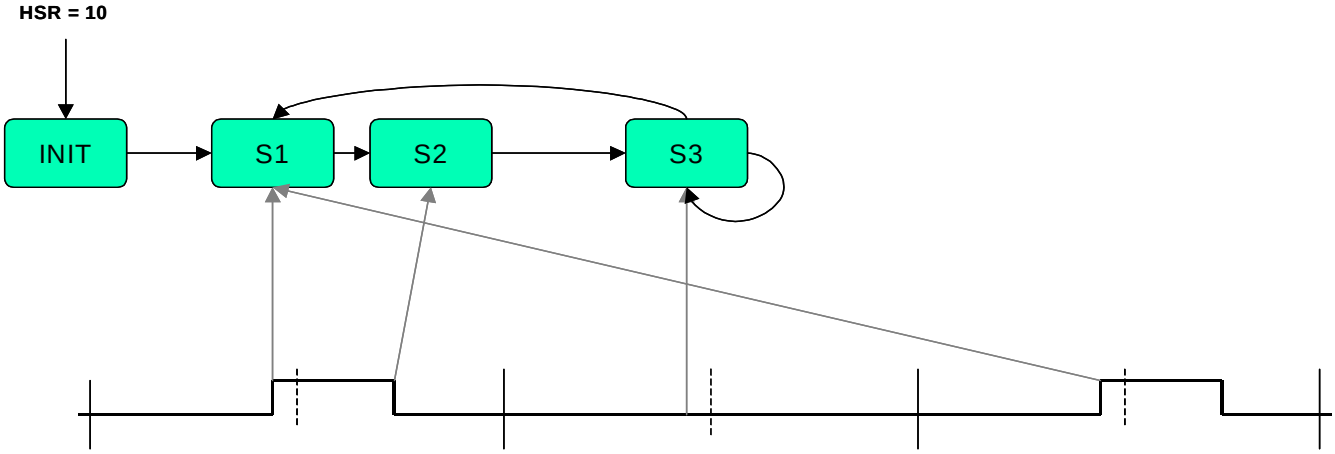

Figure 9. DCmDtXor_sync state diagram

Resolver Reference Signal for DC Motor with Dead-Time Correction – XOR version (DCmDtXor_res)

The DCmDtXor_res TPU function uses information read from the DCmDtXor_C and DCmDtXor_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.

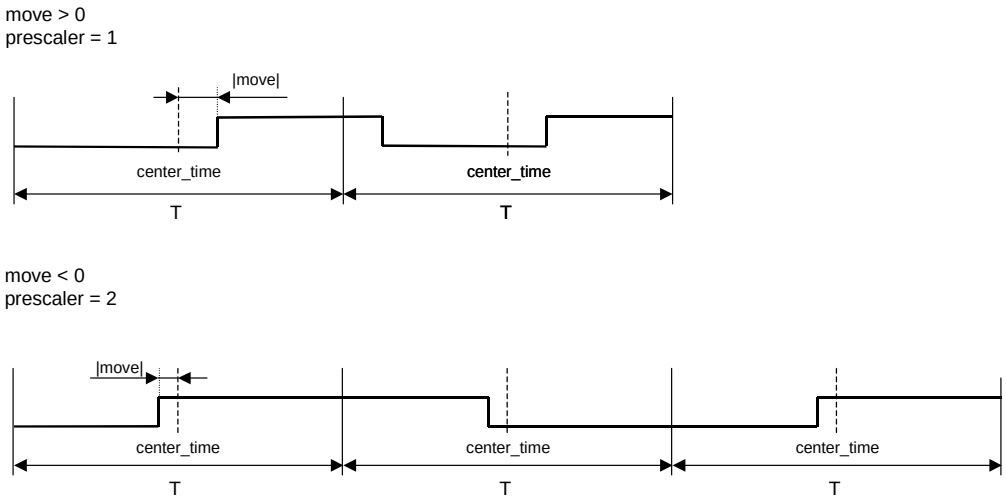

Figure 10. Resolver reference signal adjustment examples

Synchronized Change of PWM Prescaler And Resolver Reference Signals Prescaler

The DCmDtXor_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. DCmDtXor_res Control Bits

Name	Options
3 2 1 0 Channel Function Select	DCmDtXor_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

Table 14. DCmDtXor_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 15. DCmDtXor_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. DCmDtXor_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

Table 16. DCmDtXor_res parameter description

Parameter	Format	Description
prescaler	1, 2, 4, 6, 8, 10, 12, 14, ...	The number of PWM periods per synchronization pulse – use when 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. DCmDtXor_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)*

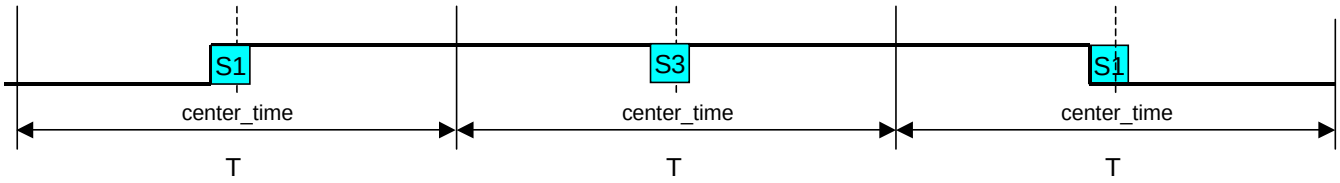

Figure 11. DCmDtXor_res timing

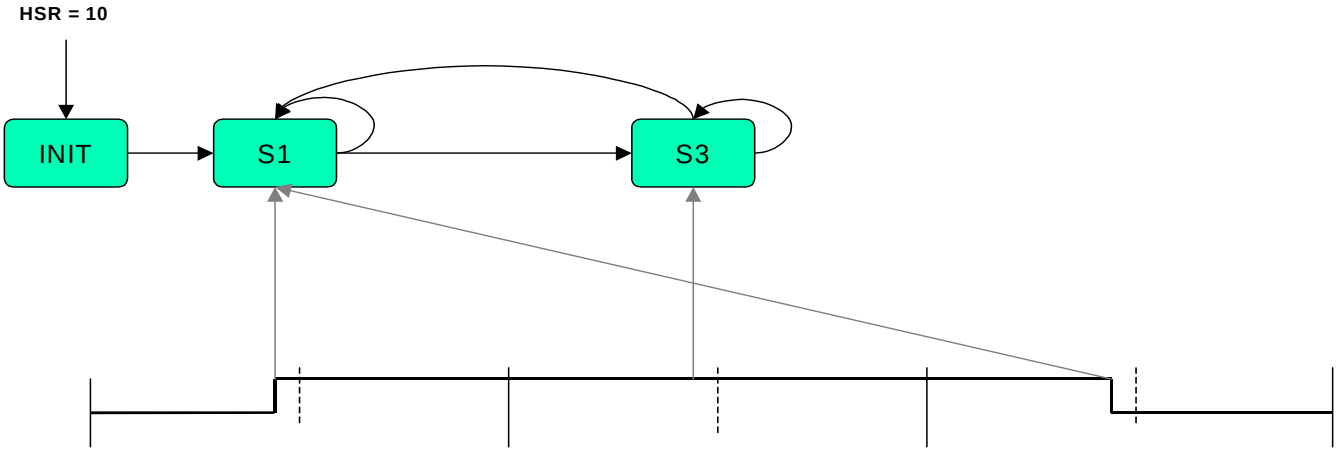

Figure 12. DCmDtXor_res state diagram

Fault Input for DC Motor with Dead-Time Correction – XOR version (DCmDtXor_fault)

The DCmDtXor_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. DCmDtXor_fault Control Bits

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

Table 18. DCmDtXor_fault Control Bits

Name	Options
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 DCmDtXor_fault generates an interrupt when a high to low transition appears.

Table 19. DCmDtXor_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. DCmDtXor_fault parameter description

Parameter	Format	Description
–		
fault_pinstate	0 or 1	State of fault pin: 0 ... low 1 ... high

Performance

Table 21. DCmDtXor_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 13. DCmDtXor_fault timing

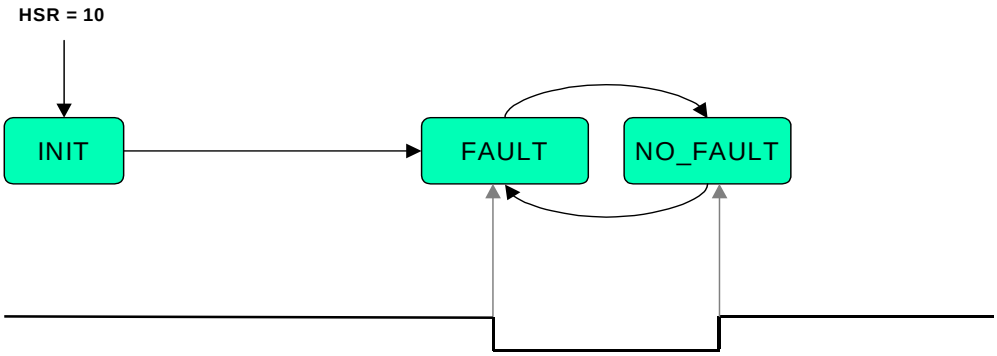

Figure 14. DCmDtXor_fault state diagram

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

AN2527/D
Rev. 0
5/2003

Freescale Semiconductor, Inc.