Freescale Semiconductor
Application Note

Document Number: AN4795
Rev 0, 08/2013

eFlexPWM Fault Handling on MC56F84xxx and MC56F82xxx DSCs

by: Pavel Sustek

1 Introduction

Power transistors used in motor-drive or power management applications need to be protected from failure under external fault conditions. The switching devices are selected to reliably handle circuit currents under normal conditions. However under fault conditions, a device could be subjected to very high surge. Only timely control and removal of the fault current by some internal or external means would save the switching device from the failure. In applications where a system fault is a possibility, a fault protection is used to sense the fault and turn off the transistors by shutting down the gate drive.

Typically inverter fault states are:

- Overcurrent
- Overvoltage
- Over temperature

Freescale MC56F84xxx and MC56F82xxx Digital Signal Controllers (DSCs) contain such fault protection of pulse width modulation (PWM) outputs.

Contents

This application note is supported by the application example code providing ready-to-use PWM module fault configuration. The document extends fault logic description in MC56F84xxx and MC56F82xxx Reference Manuals with main focus on the application use.

2 eFlexPWM fault protection

2.1 Overview

The pulse width modulator (PWM) module contains PWM submodules, each of which is set up to control a single half-bridge power stage. Fault channel support is provided. Freescale MC56F84xxx and MC56F82xxx DSC's have identically implemented fault protection logic.

Fault protection can control any combination of PWM output pins. Faults are generated by logic 1 or 0 on any of the FAULTx pins. Each FAULTx pin can be mapped arbitrarily to any of the PWM outputs. When fault protection hardware disables PWM outputs, the PWM generator continues to run; only the output pins are forced to logic 0, logic 1, or tristated. The fault decoder disables PWM pins selected by the fault logic and the disable mapping (DISMAPn) registers. The following figure shows an example of the fault disable logic.

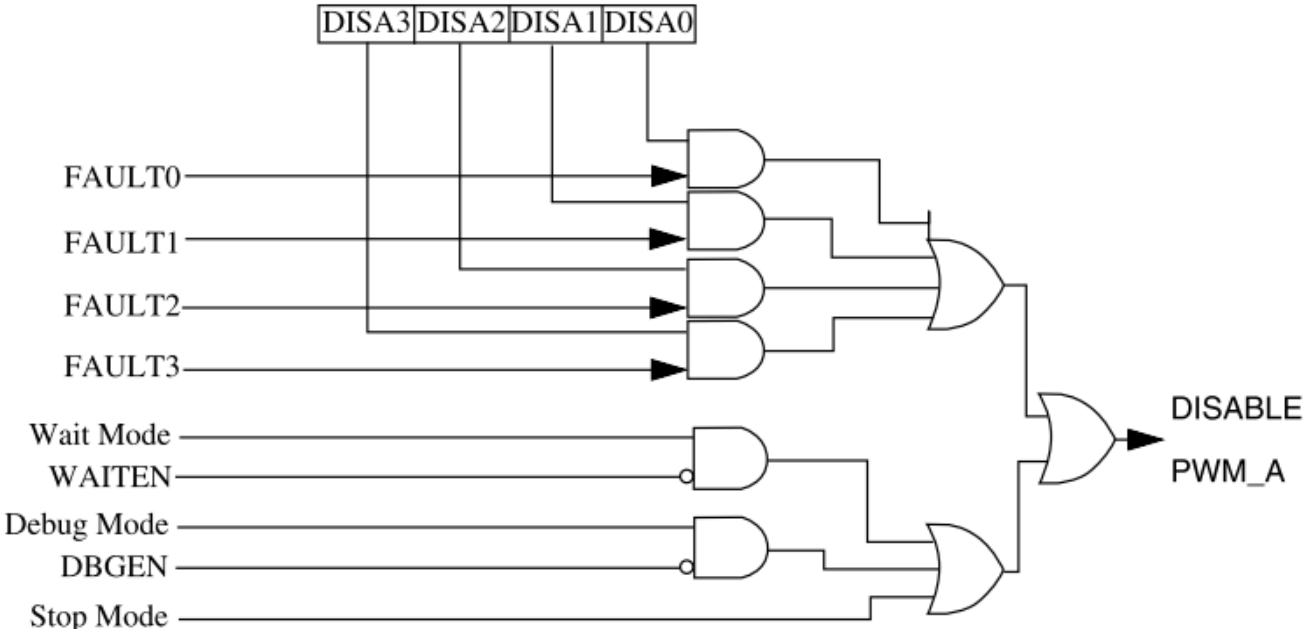


Figure 1. PWM fault logic

Each PWM module on MC56F84xxx and MC56F82xxx contains two sets of registers dedicated for PWM fault setting. The first set with postfix 0 is linked to faults 0–3, the second set with postfix 1 is linked to faults 4–7. Table 1 shows registers concerning PWM faults. For every 4-bit field in a register, the bits correspond to the fault inputs in order. For example, FIE bits correspond to faults 3–0 in PWMA_FCTRL0 and faults 7-4 in PWMA_FCTRL1.

Table 1. PWM fault registers

Register name	Full name	Field with bit position
PWMA(B)_SMxDISMAP0(1)	Fault Disable Mapping	DIS0(1)A [3–0]—Fault Disable Mask DIS0(1)B [7–4] —Fault Disable Mask DIS0(1)X [11–8] —Fault Disable Mask
PWMA(B)_SMxOCTRL	Output Control	PWMXFS [1–0]—PWM_X Fault State PWMBFS [3–2] —PWM_B Fault State PWMAFS [5–4] —PWM_A Fault State
PWMA(B)_FCTRL0(1)	Fault Control	FIE [3–0] —Fault Interrupt Enable FSAFE [7–4] —Fault Safety Mode FAUTO [11–8] —Automatic Fault Clearing FLVL [15–12] —Fault Level
PWMA(B)_FSTS0(1)	Fault Status	FFLAG [3–0] —Fault Flags FFULL [7–4] —Fault Cycle FFPIN [11–8] —Filtered Fault Pins FHALF [15–12]—Half Cycle Fault Recovery
PWMA(B)_FFILT0(1)	Fault Filter	FILT_PER [7–0] —Fault Filter Period FILT_CNT [10–8]—Fault Filter Count GSTR [15] —Fault Glitch Stretch Enable
PWMA(B)_FTST0(1)	Fault Test	FTEST [0]—Fault Test

The eFlexPWM faults can be easily configured using the Graphical Configuration Tool (GCT) which is a component of QuickStart tool [3]. The following figure shows the GCT panel providing fault settings.

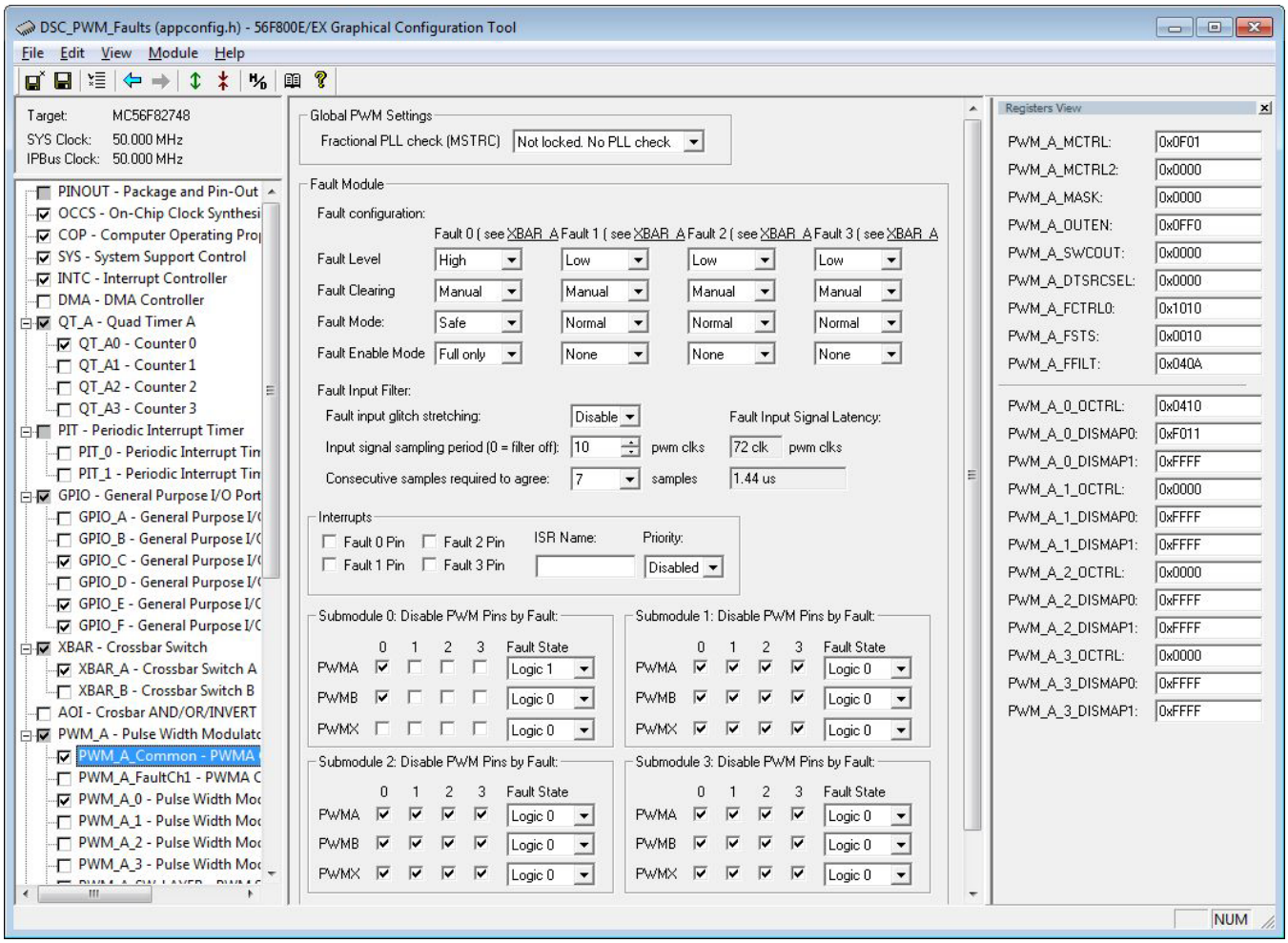


Figure 2. Fault configuration GCT panel

2.2 Fault inputs

The eFlexPWM contains up to 8 fault inputs which can be assigned to control multiple PWM outputs. The fault source can be both internal and external. Internal sources are connected to the PWM peripheral via XBAR channels and the external sources are connected via dedicated GPIO pins. The following table shows available fault sources on MC56F84xxx (containing two PWM modules) and MC56F82xxx (containing one PWM module). Some external fault sources are not available on lower-pin device packages.

Table 2. eFlexPWM fault registers

Fault number	MC56F84xxx		MC56F82xxx
	PWMA	PWMB	PWMA
FAULT0	XBAR_OUT29 / GPIO_E8	XBAR_OUT29 / GPIO_F14	XBAR_OUT29

FAULT1	XBAR_OUT30 / GPIO_E9	XBAR_OUT30 / GPIO_F13	XBAR_OUT30
FAULT2	XBAR_OUT31 / GPIO_G4	XBAR_OUT31 / GPIO_F12	XBAR_OUT31
FAULT3	XBAR_OUT32	XBAR_OUT32	XBAR_OUT32
FAULT4	GPIO_G6	GPIO_G6	GPIO_C14
FAULT5	GPIO_G7	GPIO_G7	GPIO_C15
FAULT6	GPIO_F10	GPIO_F10	GPIO_F4
FAULT7	GPIO_F9	GPIO_F9	GPIO_F5

The selection of GPIO function, as a PWM FAULT input, is set in a particular Peripheral Select Register (SIM_GPSxx). The fault input source selection between XBAR_OUTxx and GPIO_xx (only for MC56F84xxx) is set in the Internal Peripheral Select (SIM_IPS0) register.

The actual filtered FAULTx state reflects read-only bit FFPINx. The state is converted to high-polarity regardless of the fault level (FLVLx). A logic 1 indicates that a fault condition exists on the filtered FAULTx pin. If the filter is disabled, the FFPINx directly reflects changes on FAULTx input.

2.3 Fault glitch stretch logic

The fault glitch stretch logic represents GSTR field in PWMA_FFILTx register. This logic ensures that narrow fault glitches are stretched to be at least two IPBus clock cycles wide. In some cases, a narrow fault input can cause problems due to the short PWM output shutdown/reactivation time.

For example, the FFLAGx bit is set within 2 IPBus clock cycles after a transition on the FAULTx pin. The stretching logic ensures that a glitch on the fault input, when the fault filter is disabled, will be registered in the fault flags.

2.4 Fault level

The active logic level of the individual fault inputs represents FLVL fields in PWMA(B)_FCTRL0 (faults 0–3) and PWMA(B)_FCTRL1 (faults 4–7) registers.

- If FLVLx = 0, a logic 0 on the fault input FAULTx indicates a fault condition.
- If FLVLx = 1, a logic 1 on the fault input FAULTx indicates a fault condition.

Bits defining the fault state of a PWM output during fault conditions and STOP mode are located in PWMA(B)_SMnOCTRL register (n is a number of PWM sub module). The state can be set for each PWM submodule and its outputs A, B, X separately in PWMxFS bit fields.

2.5 PWM output fault state

Bit setting determines following PWM output states:

Bit setting	PWM output fault state
00	The output is forced to logic 0 state prior to consideration of output polarity control.
01	The output is forced to logic 1 state prior to consideration of output polarity control.
10	The output is tristated.
11	The output is tristated.

The resultant output PWM state is affected by the output polarity set in POLx bits (PWMA_SMnOCTRL). If the output polarity is inverted (POLx is logic one), the actual PWM output fault state will be inverted as well (except tristate state).

2.6 Fault test

The Fault Test (FTEST) field enables to simulate a fault condition. It can be used for debug purposes and application testing in case the proper fault signal is difficult to generate.

- Setting this field causes a simulated fault to be sent to all of the fault filters. The condition propagates to the fault flags and possibly, the PWM outputs depending on the DISMAPn settings.
- Clearing this field removes the simulated fault condition.

2.7 Fault clearing

The module supports two modes to clear the existing fault conditions.

- Automatic fault clearing
- Manual fault clearing

The fault clearing mode represents FAUTO fields in PWMA(B)_FCTRL0 (faults 0–3) and PWMA(B)_FCTRL1 (faults 4–7) registers.

- FAUTOx = 0: Manual fault clearing
- FAUTOx = 1: Automatic fault clearing

The PWM output can be enabled at the start of a PWM half cycle (FHALFx bit is set) or a PWM full cycle (FFULLx bit is set). At least one of these control bits must be set to enable PWM outputs, otherwise fault recovery is not possible. If both FHALFx and FFULLx bits are set, the fault recovery occurs at the start of a full cycle and at the start of a half cycle (as defined by VAL0).

2.7.1 Automatic fault clearing

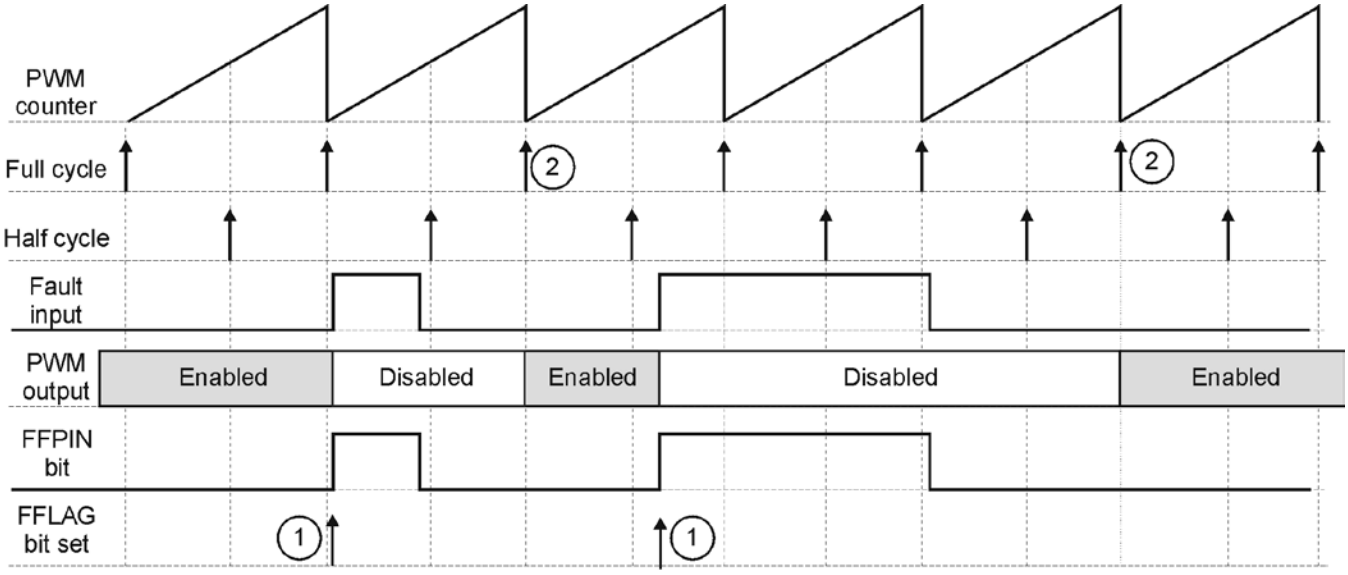
When FAUTOx is set, disabled PWM output pins are enabled when:

- no fault signal is present on FAULTx input, the FAULTx pin returns to logic 1 (FLVLx = 0) or logic 0 (FLVLx = 1)
- new PWM full cycle FFULLx = 1 (Figure 3) or half cycle FHALFx = 1 (Figure 4) begins

Clearing FFLAGx bit does not affect disabled PWM pins when FAUTOx is set.

The signals shown in the following figures are as follows.

- PWM counter—counts PWM input clocks from INIT value to the VAL1 value
- Full cycle—compare value where counter overflows (VAL1 register)
- Half cycle—typically half of PWM period represented by VAL0 register
- Fault input—fault signal from internal or external source
- PWM output—the state of PWM output that can be disabled by the fault input (DISMAPx registers)
- FFPIN bit—bit reflecting the state of filtered fault input
- FFLAG bit—fault flag is after a fault transition on filtered fault pin



1 fault indicated on the fault input, PWM output disabled, fault flag set
2 first full cycle after fault input returned to low level, PWM output re-enabled

Figure 3. Automatic fault clearing at full cycle

These code lines set automatic fault clearing mode.

```
PWM_A_FCTRL0 = 0x1100 /* FLVL = 1, FAUTO = 1, FSAFE = 0 */
PWM_A_FSTS = 0x0010     /* FFULL = 1, FHALF = 0 */
PWM_A_FFILT = 0x0000    /* FILT_COUNT = 0*/
```

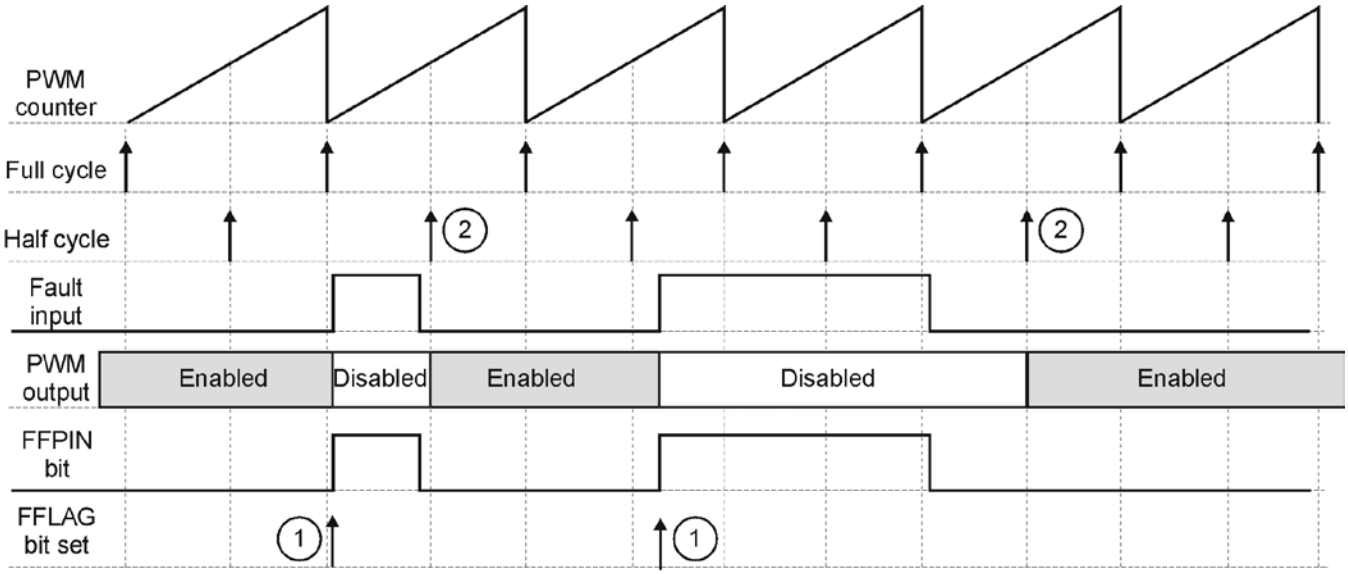


Figure 4. Manual fault clearing at half cycle

These code lines set manual fault clearing mode at half PWM cycle.

```
PWM_A_FCTRL0 = 0x1100 /* FLVL = 1, FAUTO = 1, FSAFE = 0 */
PWM_A_FSTS = 0x1000     /* FFULL = 0, FHALF = 1 */
PWM_A_FFILT = 0x0000     /* FILT_COUNT = 0*/
```

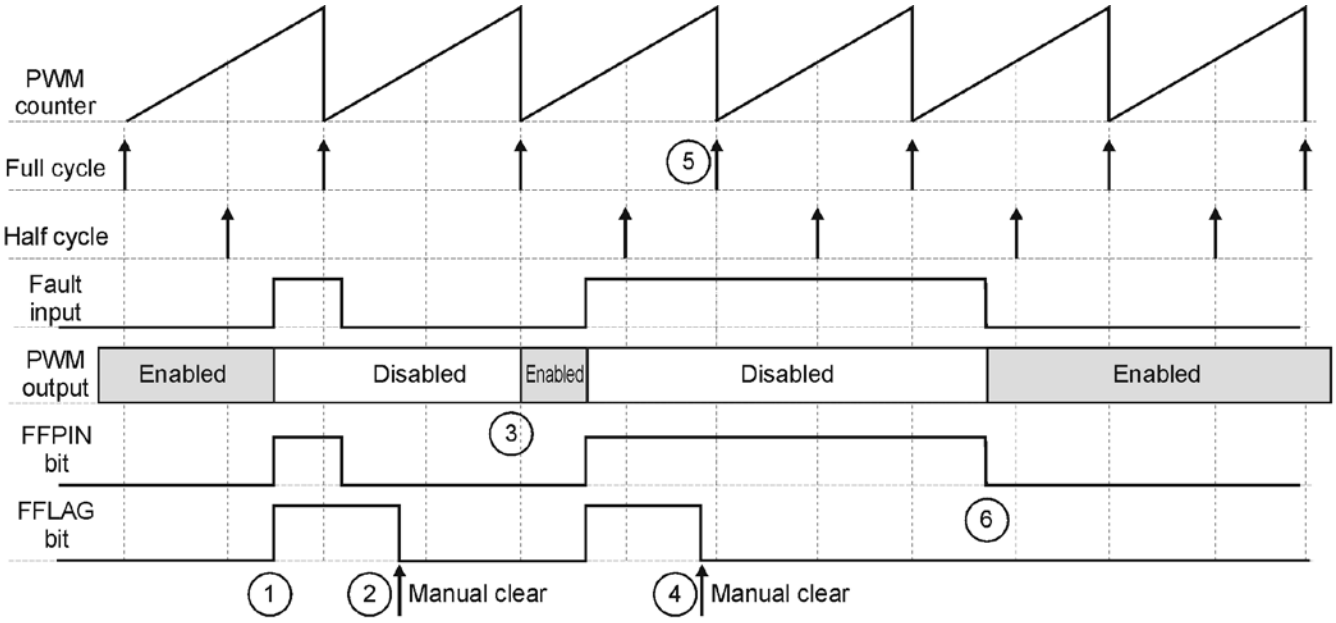
2.7.2 Manual fault clearing

Clearing the automatic clearing mode bit (FAUTOx =0), sets the manual mode.

If safety mode is disabled (FSAFEx =0) then the state of FFPINx is not checked at fault recovering time. Disabled PWM output pins are enabled when:

- software clears the corresponding FFLAGx flag
- new PWM full cycle (FFULLx = 1) or half cycle (FHALFx = 1) begins

For more details, see the following figure.



1 fault indicated on the fault input, PWM output disabled
2 FFLAG cleared manually
3 PWM output re-enabled at first upcoming full cycle
4 FFLAG cleared manually after fault indicated on the fault input and PWM output disabled
5 with disabled safe mode, the PWM output enabled at half or full cycle after FFLAG is clear regardless of FFPIN state. However, there is still combinational path in the module logic that disables PWM output if fault signal presented on the fault input. Thu,s PWM output disabled.
6 fault input returned to low level, PWM output immediately re-enabled

Figure 5. Manual fault clearing at full cycle

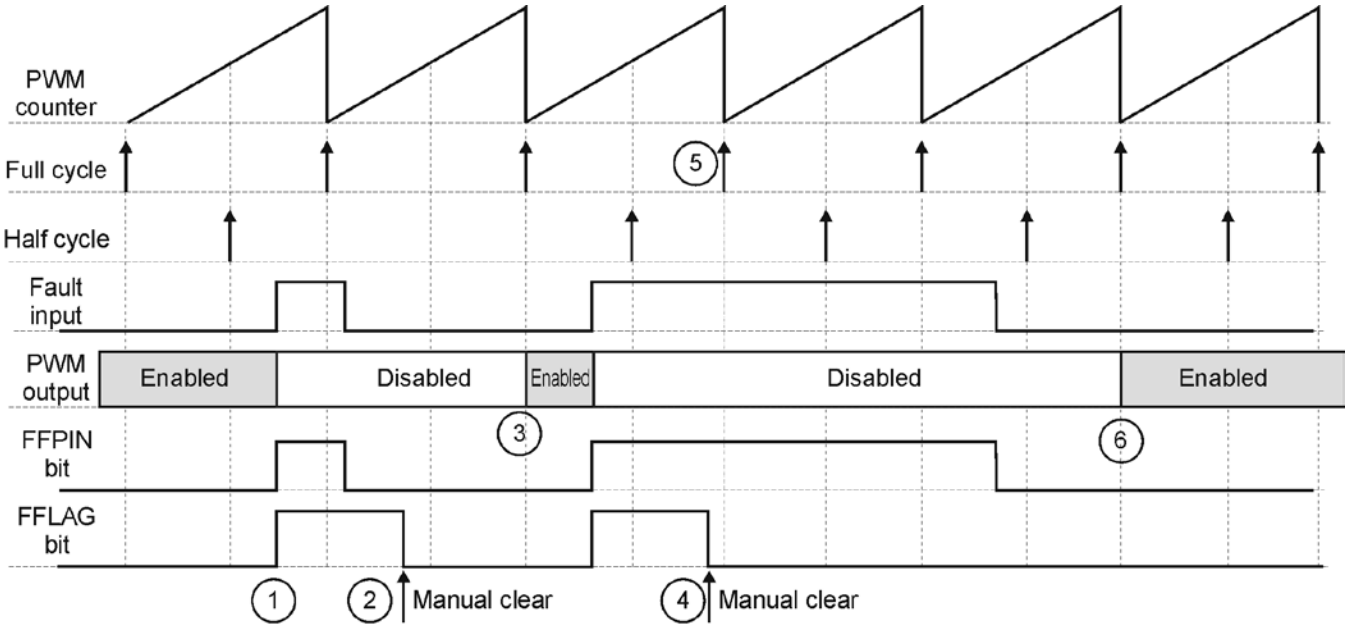
These code lines set manual fault clearing mode at full PWM cycle.

```
PWM_A_FCTRL0 = 0x1000 /* FLVL = 1, FAUTO = 0, FSAFE = 0 */
PWM_A_FSTS = 0x0010     /* FFULL = 1, FHALF = 0 */
PWM_A_FFILT = 0x0000     /* FILT_COUNT = 0*/
```

If safety mode is enabled (FSAFEx =1), then the state of FFPINx must be clear at fault recovering time. Disabled PWM output pins are enabled when:

- software clears the corresponding FFLAGx flag
- new PWM full cycle (FFULLx = 1) or half cycle (FHALFx = 1) begins
- no fault signal is detected on the FAULTx pin at the recovery instant, at the start of the next PWM full or half cycle boundary

For more details, see the following figure.



1 fault indicated on the fault input, PWM output disabled
2 FFLAG cleared manually
3 PWM output re-enabled at first upcoming full cycle
4 FFLAG cleared manually after fault indicated on the fault input and PWM output disabled
5 with safe mode enabled FFLAG and FFPIN checked to be clear at full cycle. Conditions are not accomplished PWM output remains disabled
6 FFLAG =0 and FFPIN = 0 at full cycle, PWM output re-enabled

Figure 6. Manual fault clearing at full cycle, safe mode enabled

These code lines set manual fault clearing mode at full PWM cycle with enabled safe mode.

```
PWM_A_FCTRL0 = 0x1010 /* FLVL = 1, FAUTO = 0, FSAFE = 1 */
PWM_A_FSTS = 0x0010     /* FFULL = 1, FHALF = 0 */
PWM_A_FFILT = 0x0000     /* FILT_COUNT = 0*/
```

2.8 Fault pin filter

Each fault pin has a programmable filter that can be bypassed. It must be noted that even when the filter is enabled, there is still a combinational path to disable the PWM outputs. This is to ensure rapid response to fault conditions and also to ensure fault response if the PWM module loses its clock. It means that PWM outputs are always disabled during the time the FAULTx input signal indicates fault condition regardless of whether PWM filter is enabled, see Figure 7.

The fault filter delays PWM fault interrupt generation but does not eliminate PWM output disabling during filter period.

The setting of a fault filter consists of values of two fields, FILT_PER and FILT_CNT. Both the fields are located in PWMx_FCTRL register.

- PWMx_FCTRL[FILT_PER]—sets the sampling period of the filter. The value should be set such that the sampling period is larger than the period of the expected noise. This way a noise spike will only corrupt one sample. Setting FILT_PER to all 0 disables the input filter for a given FAULTx pin. When changing values for FILT_PER from one non-zero value to another non-zero value, first write a 0 to clear the filter. The maximum number of IPBus clock cycles to be set to FILT_PER field is 255.

- PWMx_FCTRL[FILT_CNT]—sets the number of consecutive samples that must agree before an input transition is recognized. The value should be chosen to reduce the probability of noisy samples causing an incorrect transition to be recognized. The number of samples is the decimal value of this field plus 3: the bit field value of 0–7 represents 3–10 samples, respectively.

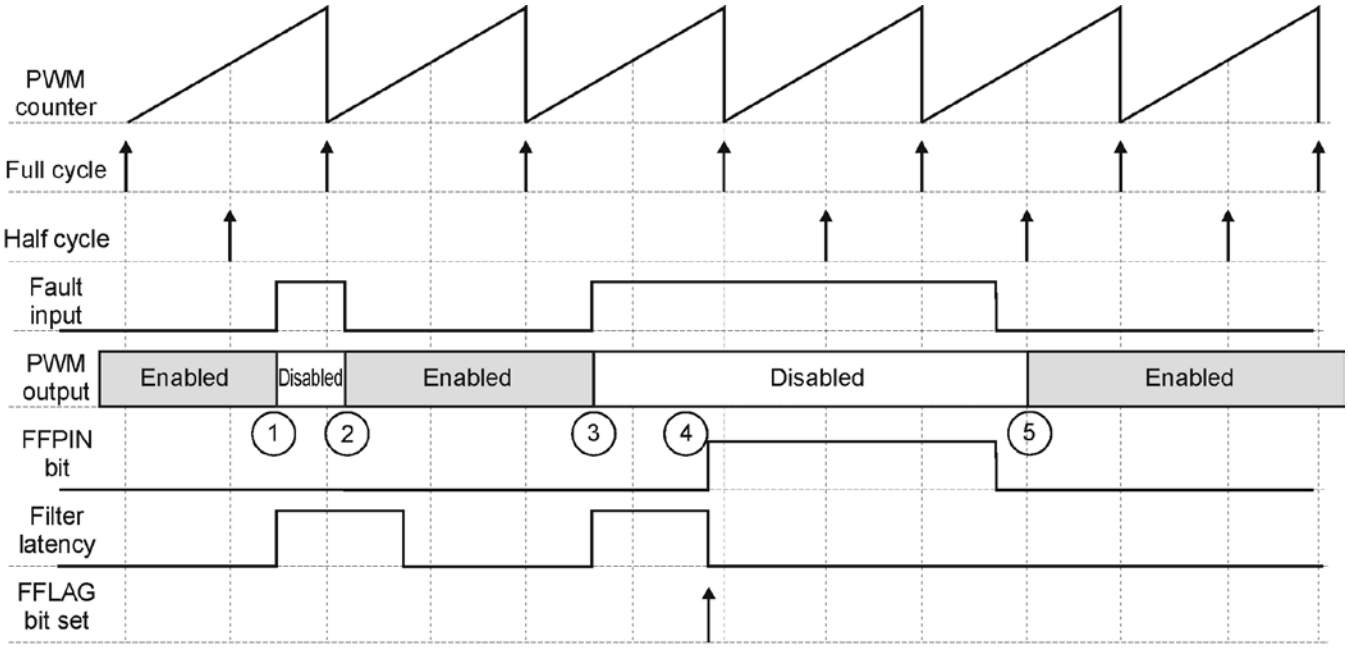
The latency in recognizing input fault transitions can be calculated by the following equation.

$$\text{FAULT FILT LATENCY} = (\text{FILT CNT} + 4) * \text{FILT PER} * \text{IPBUS clock period}$$

The latency induced by the filter will be seen in the time to set FFLAG and FFPIN fields in PWMx_FSTS register.

If the fault filter is enabled (PWMx_FCTRL[FILT_PER] >0), upon detecting a logic 0 on the filtered FAULTx pin (or a logic 1 if FLVLx is set), the corresponding FFPINx bit is set and after the time defined by the filter latency, the fault flag bits, FFLAGx are also set.

FFPINx remains set as long as the filtered FAULTx pin is 0 (or logic 1 FLVLx is set). Writing logic 1 to FFLAGx clears it.



1 fault indicated on fault input, PWM output disabled
2 filter latency longer than fault pulse, however PWM output disabled while fault presenting on fault input
3 fault indicated on the fault input
4 filter propagation delay, FFPIN = 1, FFLAG = 1
5 automatic fault clearing at half cycle, PWM output enabled

Figure 7. Automatic fault clearing at half cycle, filter enabled

These code lines set automatic fault clearing mode at half PWM cycle with filter enabled.

```
PWM_A_FCTRL0 = 0x1010 /* FLVL = 1, FAUTO = 1, FSAFE = 0 */
PWM_A_FSTS = 0x0010     /* FFULL = 1, FHALF = 0 */
PWM_A_FFILT = 0x011E     /* FILT_FILT_COUNT = 4, FILT_PER = 30 */
```

2.9 Fault interrupt

The Fault Interrupt Enable (FIE) bits enable CPU interrupt requests generated by the FAULTx pins (FFLAGx is set). The FIEx bits are located in PWMA(B)_FCTRL0 (faults 0–3) and PWMA(B)_FCTRL1 (faults 4–7) registers.

- If FIEx = 0, FAULTx CPU interrupt request is disabled.
- If FIEx = 1, FAULTx CPU interrupt request is enabled.

If the corresponding FAULTx pin interrupt enable bit is set in FIEx bit field, FFLAGx generates a CPU interrupt request. The interrupt request latch remains set until:

- Software clears FFLAGx by writing logic 1 to the bit.
- Software clears the FIEx bit by writing logic 0 to it.
- A reset occurs.

The fault protection circuit is independent on FIEx bit setting and is always active. If a fault is detected, the PWM outputs are disabled according to the disable mapping register PWMA(B)_SMxDISMAP0(1).

3 Application considerations

3.1 PWM Fault register initialization

The PWM module logic sets FFLAGx flags (registers FSTS0 for faults 0-3 and FSTS1 for faults 4-7) to logic 1 for all FAULT inputs after reset. It is recommended to clear FFLAGx flags by writing a logic 1 to these bits before the interrupts are enabled during the application initialization. Otherwise, the corresponding PWM fault interrupt is immediately generated after the corresponding bit in FIEx is set.

It is recommended to set only DIS0(1)x bits (PWMA(B)_SMnDISMAPx registers) which correspond to FAULTx used in the application. The reset values of DIS0(1)x bits are logic 1s.

3.2 Fault sources

Signals generating fault conditions are as follows.

- Internal—connected to PWM FAULTx inputs via XBAR peripheral
- External—connected to PWM FAULTx inputs from particular GPIO pins

The internal sources of fault inputs are typically peripherals like comparator, program delay block, quad timer or, AOI logic block.

The internal comparator might be used as an overcurrent protection where one input is connected to the internal reference (DAC) and the second input to the signal reflecting measured current. And the comparator output is connected through XBAR channel to the PWM fault input, see this figure.

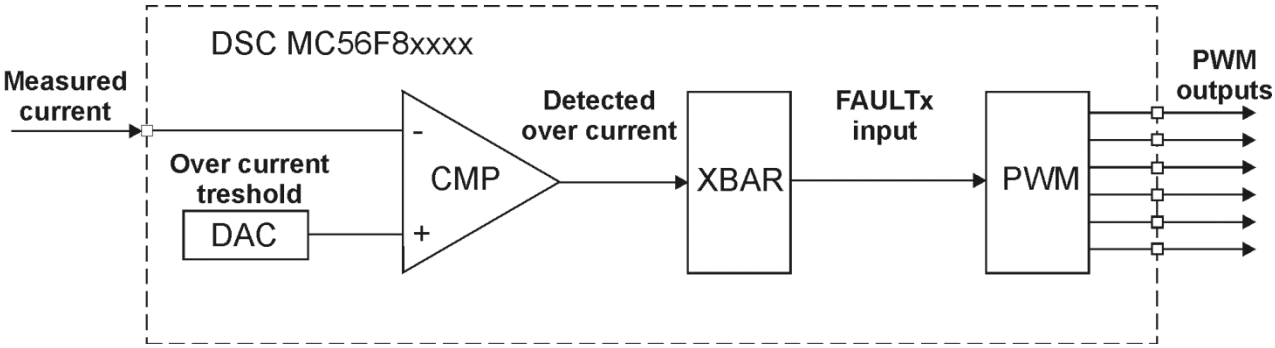


Figure 8. Internal fault source connection

The external sources are typically comparators, temperature sensors, and others. The external RC filter is recommended to be used for a fault signal to avoid random fault signal generation caused by noise on the board. However, since the PWM module contains a fault filter, there is still a combinational path to disable the PWM outputs during the time the fault is active on a FAULTx input (see Figure 5 and Figure 7).

4 Application example

The example application attached with the application note provides a typical configuration of PWM faults with both internal and external fault source signals.

The code is based on CodeWarrior v.10.4, QuickStart 2.6 and FreeMASTER 1.3.15. The TWR-56F8200 is used as a hardware board. Jumper setting for the application example is shown in this table.

Table 3. TWR-56F8200 jumper setting

Jumper name	Position
J4	1-2 3
J5	1 2-3
J8	1 2 3 4 5
J9	1 2 3 4 5

The internal fault signal is generated from internal comparator HSCMP_A. The comparator compares reference signal set by 6-bit DAC to 0.567 V and 12-bit DAC that generates triangle waveform. The periodic comparator output pulse is connected to PWM fault input 1 via XBAR channel 30.
The fault level is set to high polarity (FLVL1 = 1) and disables PWM_A1 and PWM_B1 outputs setting them to logic 0.
The fault clearing is set to the automatic mode with a half cycle recovery time.

The external fault signal is generated by pressing the SW1 button. The fault signal is connected to PWM fault input 0 via XBAR channel 29. The fault level is set to low polarity (FLVL0 = 0) and disables PWM_A0 and PWM_B0 outputs setting them to logic 0. But PWM_A0 output is set to active low polarity which also inverts PWM_A0 output to logic 1 if the output is disabled by the fault. The fault flag generates the interrupt request. The code put to the interrupt service routine Extern_Fault_ISR() turns the LED8 on and waits until SW2 button is pressed. Then the fault flag is cleared, PWM_A0 and PWM_B0 outputs are enabled, and the LED8 is turned off.

PWM outputs A0 and B0 are connected to LED0 and LED1 and A1 and B1 are connected to LED2 and LED3. The peripheral configuration is done in GCT where all peripheral register settings are accessible.

5 Conclusion

This application note describes a detail configuration of eFlexPWM fault logic from the application perspective. An inverter requires proper setting of fault protection to avoid hazard and dangerous states. The application source code provided with the application note supports customer application development using latest tools.

6 References

The following documents are available on freescale.com.

1. *MC56F827xx Reference Manual, (document MC56F8277XRM)*
2. *MC56F847xx Reference Manual, (document MC56F847XXRM)*
3. *DSC56800EX Quick Start User Guide (document DSC56800EXQSUG)*

7 Acronyms and abbreviated terms

This table contains abbreviated terms used in this application note.

Table 4. Acronyms and abbreviated terms

Term	Meaning
LED	Light Emitting Diode
DSC	Digital Signal Controller
IPBus	Internal Peripheral Bus
PWM	Pulse-Width Modulation
XBAR	Inter-Peripheral Crossbar Switch

8 Revision history

Revision number	Date	Substantive changes
0	08/2013	Initial release

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: AN4795
Revision 0, August 2013