AN205494

F²MC-16FX, All Series, Programmable Pulse Generator

This application note reflects the functionality and describes the different modes of the Programmable Pulse Generator. The PPG is a 16-bit down counter with selectable duty cycle (counter value match for output pin state change).

Contents

1 Introduction....1
1.1 Key Features....1
2 The Programmable Pulse Generator....2
2.1 Block Diagram....2
2.2 Simplified Block Diagram....3
2.3 PPG Grouping....4
2.4 Registers....5
2.5 PPG Counter Behaviour....7
2.6 Frequency Examples....8
2.7 External Trigger....10
3 PPG Examples....11
3.1 Basic PPG Functionality....11
3.2 Changing Clock Source....11
3.3 Triggering by Reload Timer....12
3.4 Synchronize PPG Group....13
4 Additional Information....14
Document History....15

1 Introduction

This application note reflects the functionality and describes the different modes of the Programmable Pulse Generator. The PPG is a 16-bit down counter with selectable duty cycle (counter value match for output pin state change).

1.1 Key Features

- Selectable 16-bit reload value and 16-bit duty cycle
- Actual count readable
- Pre scalar dividers: 1, 4, 16, 64
- Frequency Range from 3.8 Hz to 8 MHz with Peripheral Clock at 16 MHz
- Reload Timer 6 Underflow as Clock Source selectable (allows frequencies from about 0.9 μHz (with maximum possible clock divider & period settings of PPG) to 4 MHz with Peripheral Clock at 16 MHz)
- Accuracy of Duty Cycle up to 65536 steps (0.0015%)
- Trigger Inputs with Edge Selection: External, Reload Timer 0 or 1, Internal
- Output polarity and Clamped H/L selectable
- Interrupt at Trigger, Counter Borrow, Duty Value Match, and Borrow or Match.
- One Shot / PWM Mode

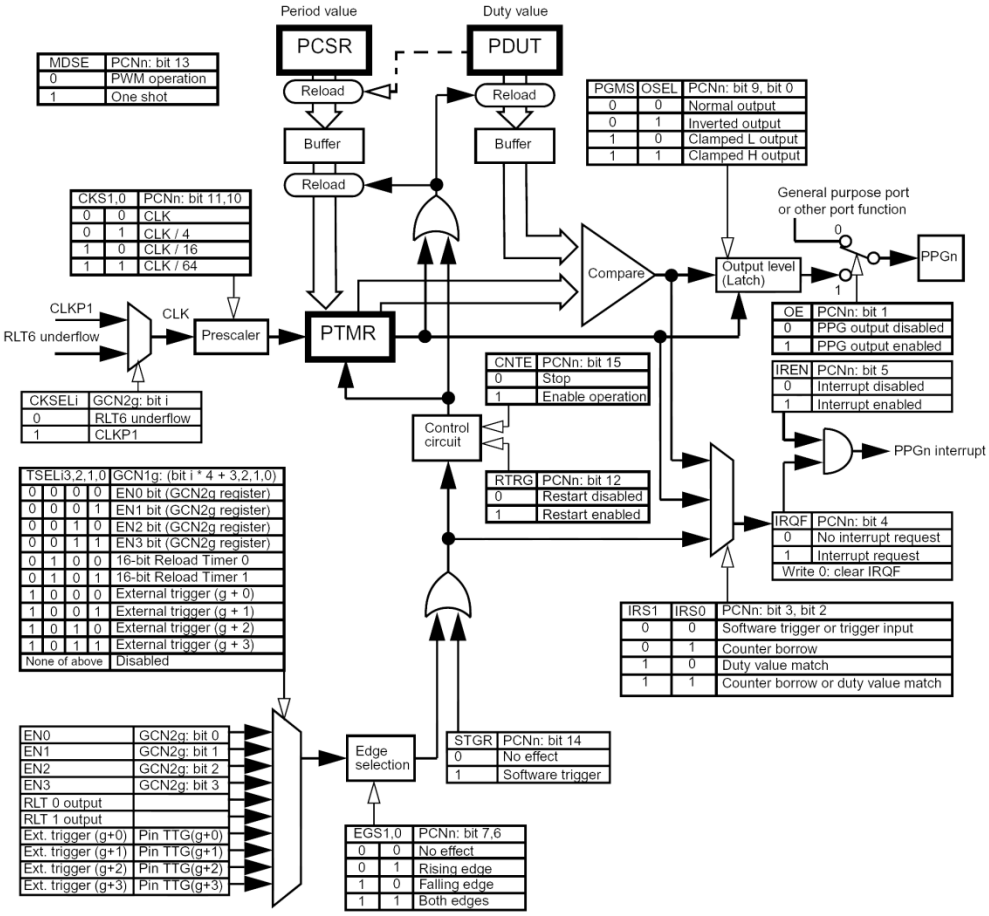
2 The Programmable Pulse Generator

The basic functionality of the PPG

2.1 Block Diagram

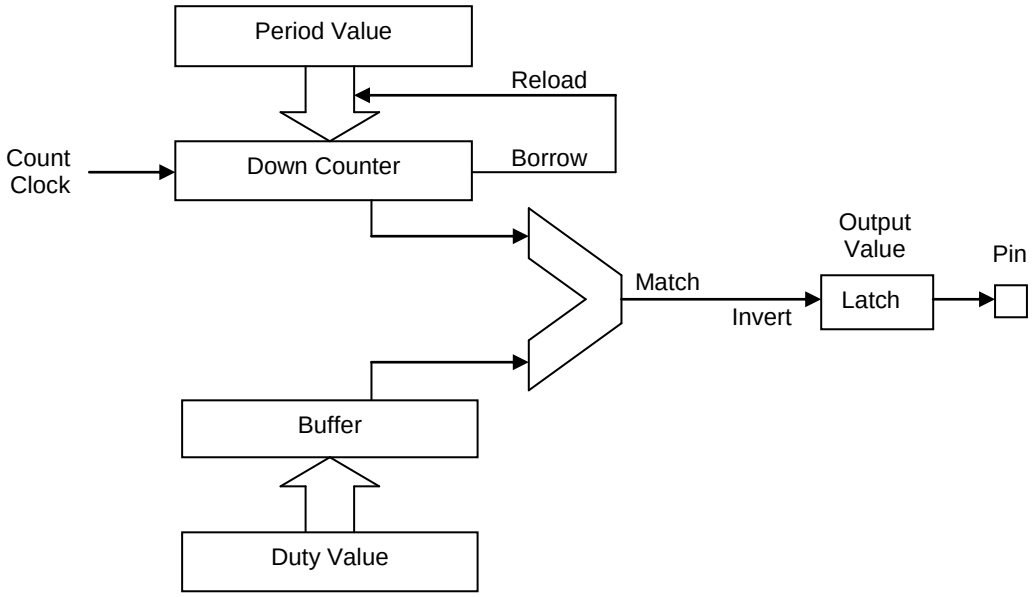
Figure 1 shows the internal block diagram of a PPG channel.

Figure 1. PPG Block Diagram



2.2 Simplified Block Diagram

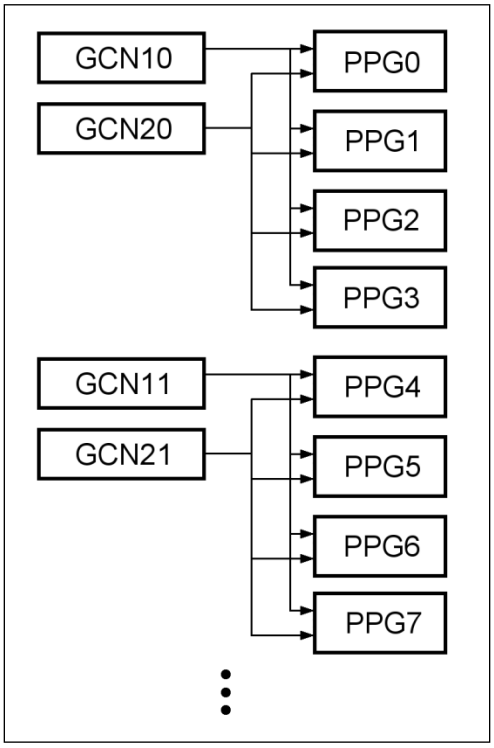
Figure 2. Simplified PPG Block Diagram



2.3 PPG Grouping

The following block diagram shows, how the PPGs are grouped.

Figure 3. Grouping of the PPGs



Different PPGs of this group can share one common trigger and clock source. The trigger type of a group is controlled by corresponding GCN1n register. The clock source for PPGn to PPGn+3 is controlled by corresponding GCN2n register.

Note, while using Reload Timer 0 or 1 as trigger source for all PPGs, all these PPGs can be synchronized to this trigger source.

2.4 Registers

2.4.1 PPG Control Status Register (PCN)

The PCN contains almost all control bits for the functionality of the PPG

Table 1. PCN

Bit No.	Name	Explanation	Value	Operation
15	CNTE	Count Enable	0	PPG disabled
			1	PPG enabled
14	STGR	Software Trigger	0	No trigger
			1	Trigger activated
13	MDSE	Mode Selection	0	Continuous Mode
			1	One-Shot Operation
12	RTRG	Restart Enable	0	Disable Restart
			1	Enable Restart
11, 10	CKS1, 0	Counter Clock Selection	0, 0	Clock selected by CKSEL
			0, 1	Clock selected by CKSEL / 4
			1, 0	Clock selected by CKSEL / 16
			1, 1	Clock selected by CKSEL / 64
9	PGMS	PPG Output Mask Selection	0	No Output Mask
			1	Output Mask (Clamped by OSEL)
8	–	*Undefined*	-	Always write 0
7, 6	EGS1, 0	Trigger Input Edge Selection	0, 0	No Selection
			0, 1	Rising Edge
			1, 0	Falling Edge
			1, 1	Both Edges
5	IREN	Interrupt Enable	0	Interrupt Disabled
			1	Interrupt Enabled
4	IRQF	Interrupt Request Flag	0	Clear Interrupt Request
			1	No Effect
3, 2	IRS1, 0	Interrupt Cause Selection	0, 0	Software or External Trigger
			0, 1	Counter Borrow
			1, 0	Counter matches Duty Value
			1, 1	Counter Borrow or Duty Value match
1	OE	Output Enable	0	Output Disabled
			1	Output Enabled
0	OSEL	Output Polarity	0	Normal Polarity
			1	Inverted Polarity

2.4.2 General Control Register 1 (GCN1)

The GCN1 consists of 4 blocks of Trigger Selection Control Bits. These blocks are related to a group of 4 PPGs. The Trigger Activation works together with the EGS1-0 bits of the 2.4.1 PPG Control Status Register (PCN).

Table 2. GCN1

TSELi3	TSELi2	TSELi1	TSELi0	Activation Trigger Specification
0	0	0	0	EN0 Bit (GCN2 Register)
0	0	0	1	EN1 Bit (GCN2 Register)
0	0	1	0	EN2 Bit (GCN2 Register)
0	0	1	1	EN3 Bit (GCN2 Register)
0	1	0	0	16-Bit Reload Timer Output 0
0	1	0	1	16-Bit Reload Timer Output 1
1	0	0	0	External Trigger (Group*4 + 0)
1	0	0	1	External Trigger (Group*4 + 1)
1	0	1	0	External Trigger (Group*4 + 2)
1	0	1	1	External Trigger (Group*4 + 3)
All other settings				Disabled

Note, that “i” is the PPG number. It rises in foursome blocks from Bit#0 to Bit#15 in the GCN1.

2.4.3 Lower General Control Register 2 (GCN2L)

The lower 4 Bits of the GCN2L contains the Trigger Level Control Bits.

Table 3. GCN2L

EN0 … 3	Internal Triggers EN0 ... 3
0	Set Level to „L“
1	Set Level to „H“

2.4.4 Upper General Control Register 2 (GCN2H)

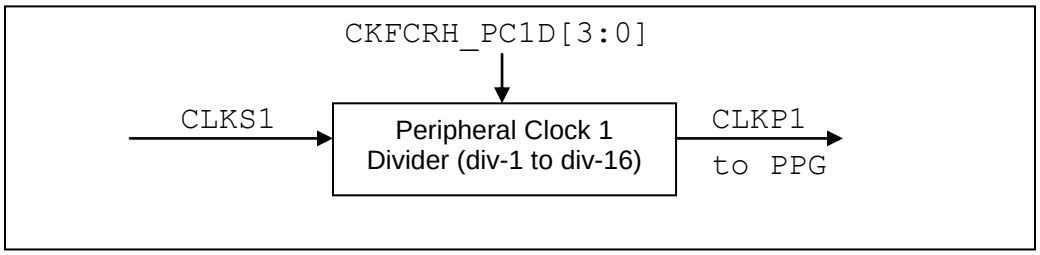
The lower 4 Bits of the GCN2H contains the Pre-scalar Clock Source Selection Control Bits.

Table 4. GCN2H

CKSEL0 … 3	Clock Source
0	CLKP1
1	Reload Timer 6

Please consider that the CLKP1 frequency depends on the setting of the Peripheral Clock 1 Divider.

Figure 4. PPG Clock



2.4.5 PPG Cycle Setting Register (PCSR)

This 16-Bit register contains the period value for a PPG channel.

2.4.6 PPG Duty Setting Register (PDUT)

This 16-Bit register contains the duty time in which the PPG changes its output. Please set a value smaller than the PCSR cycle for normal PPG operation. If these values are equal, the PPG output is “H”, if OSEL=0 or “L”, if OSEL=1 (OSEL is Bit#0 of PCN: 2.4.1).

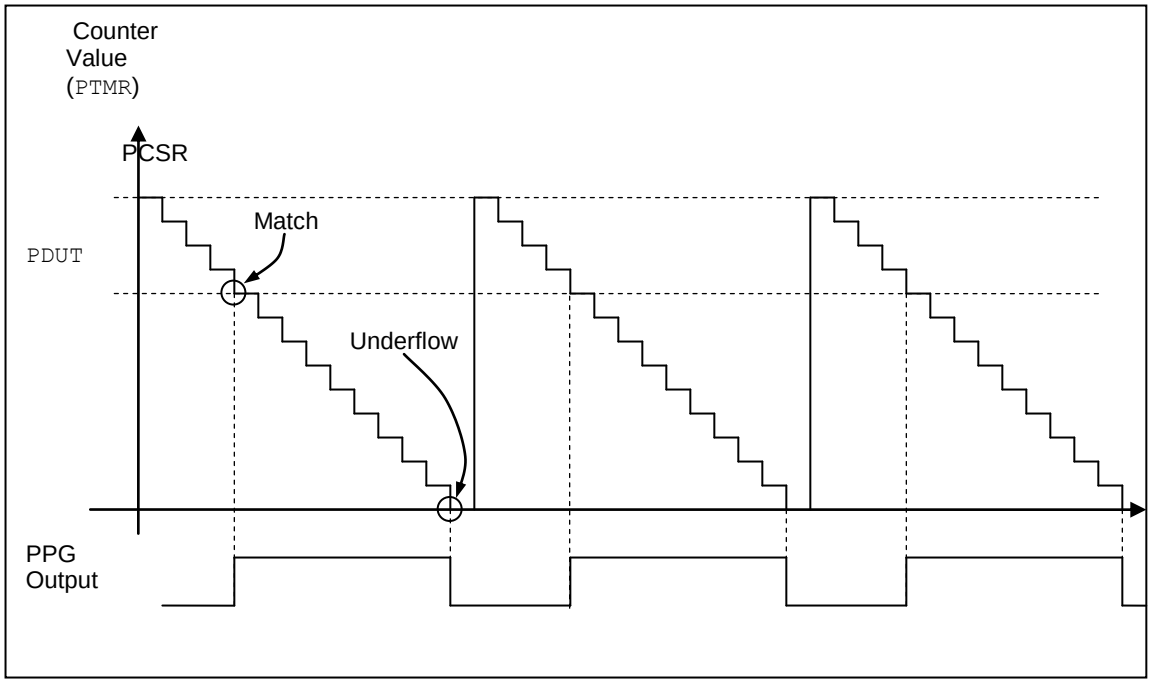
2.4.7 PPG Timer Register (PTMR)

This 16-Bit register is the counter of the PPG. Read access returns the current counter value.

PPG Counter Behavior

The following diagram shows, how the PPG works in the basic mode (free running, no trigger):

Figure 5. PPG Behavior



The non-inverted PPG output is set to “L” if an underflow of the counter value occurs. After this, the period value (PCSR) is reloaded. The PPG output is set to “H”, if the duty cycle value (PDUT) matches the actual counter value.

2.5 Frequency Examples

The following table shows some frequency settings for PPG clocked by CLKP1.

Table 5. Frequency Settings

CLKP1	CKS1, 0 Division	Period Value PCSR	Period Time	Period Frequency	Granulation Time
16 MHz	1	0xFFFF	4.096 ms	244.1 Hz	62.5 ns
		0x0001	125 ns	8 MHz	
	4	0xFFFF	16.38 ms	61 Hz	250 ns
		0x0001	500 ns	2 MHz	
	16	0xFFFF	65.54 ms	15.3 Hz	1 µs
		0x0001	2 µs	500 kHz	
	64	0xFFFF	262 ms	3.8 Hz	4 µs
		0x0001	8 µs	125 kHz	
24 MHz	1	0xFFFF	2.731 ms	366.2 Hz	41.7 ns
		0x0001	83.3 ns	12 MHz	
	4	0xFFFF	10.92 ms	91.6 Hz	166.7 ns
		0x0001	333.3 ns	3 MHz	
	16	0xFFFF	43.69 ms	22.9 Hz	0.67 µs
		0x0001	1.33 µs	750 kHz	
	64	0xFFFF	174.7 ms	5.7 Hz	2.67 µs
		0x0001	5.33 µs	187.5 kHz	
50 MHz	1	0xFFFF	1.311 ms	763 Hz	20 ns
		0x0001	40 ns	25 MHz	
	4	0xFFFF	5.243 ms	190.7Hz	80 ns
		0x0001	160 ns	6.25 MHz	
	16	0xFFFF	20.97 ms	47.7 Hz	0.32 µs
		0x0001	0.64 µs	1.56 MHz	
	64	0xFFFF	83.8 ms	11.92 Hz	1.28 µs
		0x0001	2.56 µs	390.63 kHz	

The formula for the PPG frequency f_{PPG} using f_{CLKP1} (CLKP1) as clock source is:

$$f_{PPG} = \frac{f_{CLKP1}}{div_{CKS1,0} \cdot (P+1)}$$, where $div_{CKS1,0}$ is the CKS1,0 division factor and P the Period Value

Note, that CLKP1 can also be divided by the settings in the Peripheral Clock 1 Divider, so that for high System Clocks (CLKS1) low frequencies for PPG can be used.

For low frequencies it is also possible to use Reload Timer 6 as a pre-scalar. The formula for the resulting PPG frequencies is:

$$f_{PPG} = \frac{f_{CLKP1}}{div_{CKS1,0} \cdot div_{FSEL} \cdot (P+1) \cdot (R+1)}$$, where div_{FSEL} is the division factor of the Reload Timer and R the Period Value

2.5.1 Accuracy of Duty Cycle

The accuracy of the duty cycle depends on the used period value caused on the resulting granularity.

The formula for the accuracy a in per cent is:

$$a = \frac{1}{P+1} \cdot 100\%$$, where P is the Period Value

2.6 External Trigger

Any PPG channel can be triggered by an edge on a port pin. It can be selected which on of the up to 4 trigger pins of a group is assigned to the PPG channels. This is done by the GCN1n and GCN2n registers.

Please note that the trigger pin works as an input pin. All input pins must be enabled by the respective PIER register.

Example:

PPG1 shall be triggered by pin TTG3. Pin TTG3 shares port pin P02_7.

The remaining configuration is arbitrarily selected as follows:

- Period length: 4096 CLKP1 cycles
- Duty length: 2048 CLKP1 cycles
- Output pin enabled

```
/*                            SAMPLE CODE                                  */
/*-------------------------------------------------------------------------*/

void InitPPG0(void)
{
  PCSR0 = 0x0FFF;   // always set cycle value PERIOD 1st
  PDUT0 = 0x07FF;   // set duty value DUTY CYCLE
  PIER02_IE7 = 1;   // enable input on P02_7, TTG3
  GCN11 = 321B       // setting PPG1 trigger to TTG3; other settings as
                    // initial values
  PCNL0 = 0x02;      // Output enable
  PCNH0 = 0x90;      // Count enable, Retrigger, CLK - no div
}
```

3 PPG Examples

Examples for PPG operation

3.1 Basic PPG Functionality

The following code shows how to initialize the PPG for basic operation.

```
/*                            SAMPLE CODE                                    */
/*---------------------------------------------------------------------------*/

void InitPPG0(void)
{
   PCSR0 = 0x1000;         // always set cycle value PERIOD 1st
   PDUT0 = 0x0800;         // set duty value DUTY CYCLE
   PCNL0 = 0x02;           // Output enable
   PCNH0 = 0xD0;           // Count enable, S-Trigger, Retrigger, CLK - no div
}
```

This example code generates a PPG signal with a duty cycle of 50%. The output frequency depends on the MCU clock settings. Here it is clock source (CKSEL) divided by 0x1000 + 1. For a frequency of 16 MHz for CLKP1, the resulting PPG frequency is 3905 Hz.

3.2 Changing Clock Source

The PPG can be clocked by CLKP1 or Reload Timer 6. The following example shows how to operate the PPG with the Reload Timer.

```
/*                               SAMPLE CODE                                 */
/*---------------------------------------------------------------------------*/

void InitReloadTimer6(void)
{
        TMRLR6 = 0x0001;    // set reload value
        TMCSR6 = 0x1013;    // prescaler 1:1, no interrupts
}

void InitPPG0(void)
{
      PCSR0  = 0x1000;     // always set cycle value PERIOD 1st
      PDUT0  = 0x0800;     // set duty value DUTY CYCLE
      PCNL0  = 0x02;       // Output enable
      PCNH0  = 0xD0;       // Count enable, SW-Trigger, Retrigger, CLK - no div
      GCN2H0 = 0x01;       // Clock Source: Reload Timer 6
}
```

3.3 Triggering by Reload Timer

The PPG can be triggered by Reload Timer 0 or 1. The following example shows how to trigger the PPG with the Reload Timer. Note that the PPG is in single shot mode.

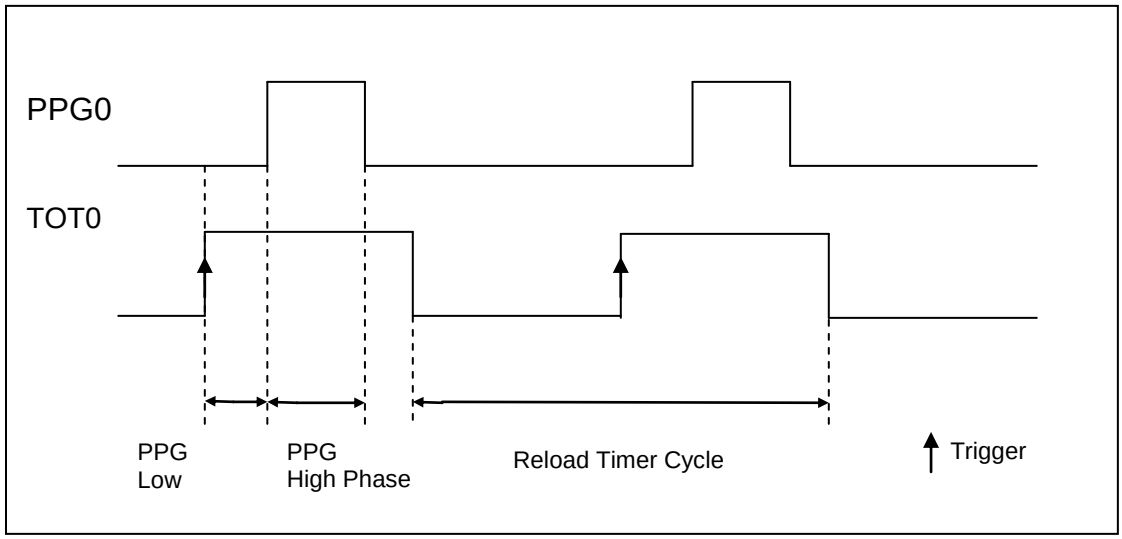
```
/*                              SAMPLE CODE                                   */
/*----------------------------------------------------------------------------*/

void InitReloadTimer0(void)
{
        TMRLR0 = 0x2000;     // set reload value
        TMCSR0 = 0x1053;     // pre-scalar 1:1, no interrupts, output enable
}

void InitPPG0(void)
{
       PCSR0   = 0x1000;    // always set cycle value PERIOD 1st
       PDUT0   = 0x0A00;    // set duty value DUTY CYCLE
       PCNL0   = 0x42;       // Trigger rising edge, Output enable
 PCNH0   = 0xB0;         // Count enable, One Shot, Retrigger,
                         // CLK - no div
 GCN1L0 = 0x04;       // Trigger Source: Reload Timer 0
}
```

The generated waveforms look like the following graphic.

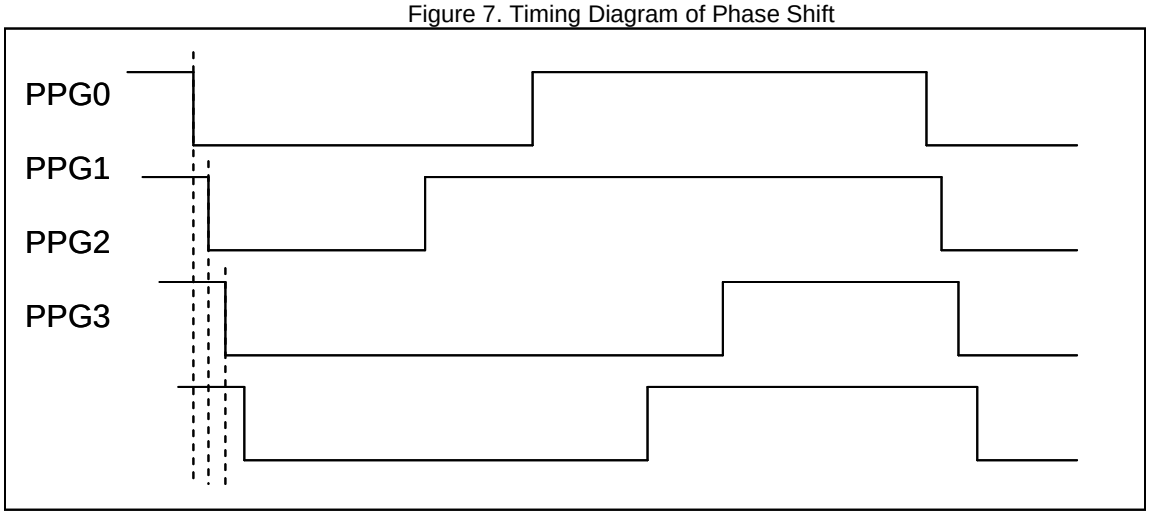
Figure 6. Timing Diagram



3.4 Synchronize PPG Group

If you initialize the PPGs sequentially, but with the same frequency, the starting falling edges (no inversion) of each channel will have a phase shift.

Figure 7. Timing Diagram of Phase Shift



It is possible to synchronize 4 PPG channels with the Internal Triggers.

To synchronize 4 channels, set the EGS*n* Bits of the PPG Control Status Registers (PCNn) to rising edge. Set all 4 channel Trigger Selections to EN0 by setting the General Control Register (GCN1n) to 0x0000. The sequence GCN2Ln = 0x00 and GCN2Ln = 0x01 creates a rising edge on ENn.

```
/*                              SAMPLE CODE                                 */
/*--------------------------------------------------------------------------*/
void InitPPG0(void)
{
   PCSR0 = 0x0FFF;  // always set cycle value PERIOD 1st
   PDUT0 = 0x0700;  // set duty value DUTY CYCLE
   PCNL0 = 0x42;    // Rising edge trigger, Output enable
 PCNH0 = 0xD0;     // Count enable, SW-Trigger, Retrigger, CLK - no div

   PCSR1 = 0x0FFF;
   PDUT1 = 0x0400;
   PCNL1 = 0x42;
 PCNH1 = 0xD0;

 PCSR2 = 0x0FFF;
   PDUT2 = 0x0900;
   PCNL2 = 0x42;
 PCNH2 = 0xD0;

 PCSR3 = 0x0FFF;
   PDUT3 = 0x0800;
   PCNL3 = 0x42;
 PCNH3 = 0xD0;

 GCN10 = 0x0000;  // PPG0 -> EN0, PPG1 -> EN0, ...
 GCN2L0 = 0x00;   // Generate Rising Edge
 GCN2L0 = 0x01;   // Trigger EN0
}
```

4 Additional Information

Information about CYPRESS Microcontrollers can be found on the following Internet page:

http://www.cypress.com/cypress-mcu-product-softwareexamples

The software examples related to this application note is:

96340_ppg0

96340_ppg0_rlt6

96340_ppg0_rlt0_trg

96340_ppg_rlt_adc_dma

Document History

Document Title: AN205494 - F²MC-16FX, All series, Programmable Pulse Generator

Document Number: 002-05494

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	MKEA	04/20/2006	V1.0, First release, MWi
			05/23/2006	V1.1, Group diagram and example added
			12/08/2006	V1.2, Reviewed the document and updated with review findings, MPi
			02/21/2007	V1.3, Fixed typos and clarified various items, MPi
			08/14/2007	V1.4, Added list of tables and figures and fixed typos, MPi
			01/04/2008	V1.5, Fix document number in footer, PHu
			06/24/2008	V1.6; update information on PPG groups; add information about external pin trigger; PHu
*A	5076166	MKEA	03/29/2016	Converted Spansion Application Note “MCU-AN-300201-E-V16” to Cypress format
*B	5869285	AESATMP9	08/31/2017	Updated logo and copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2006-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.