Freescale Semiconductor
Application Note

Document Number: AN4163
Rev. 0, 8/2010

PDB Driver for the MC9S08GW64

by: Tanya Malik
Reference Design and Applications Group
Noida
India

Contents

1 Introduction

This document describes a driver for a Programmable Delay Block (PDB), allowing users to customize all the possible configurations for this peripheral.

The software architecture is designed to provide seamless migration between devices that have the same peripheral module.

In this application note, the driver interfaces are explained. Various applications for the MC9S08GW64 can make use of this driver. The following sections describe details and steps for creating an application using the PDB driver.

The primary function of the programmable delay block is simply to provide controllable delays from either an external trigger, or a programmable interval tick, to the sample trigger input of one or more ADCs.

1.1 Programmable Delay Block (PDB) in MC9S08GW64

The MC9S08GW64 series includes a PDB block used to provide a trigger to the two ADC trigger select inputs. The clock to the PDB module can be gated on or off by using the SCGC4_PDB bit of the System clock gating control 4 register. On reset the clock is gated to the PDB module.

© 2010 Freescale Semiconductor, Inc.

1.2 Block Diagram

The following block diagram shows that the PDB block generates four triggers — pre-trigger for channel A, trigger for channel A, pre-trigger for channel B, and trigger for channel B.

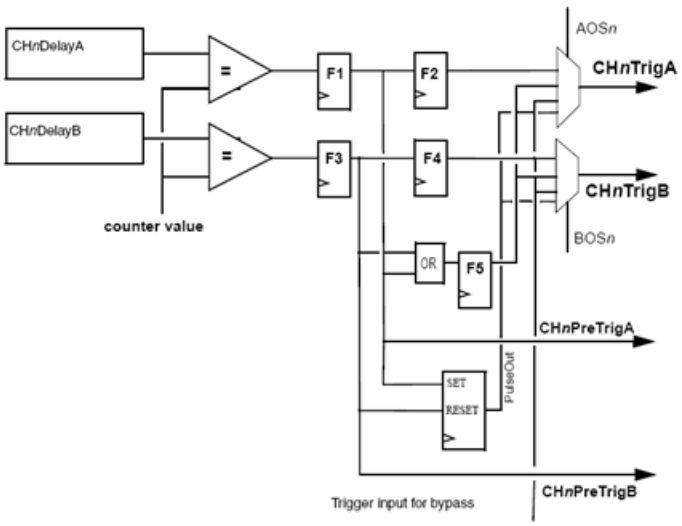

Figure 1. PDB channel block diagram

DelayA and DelayB determine the time between assertion of the trigger input to the point at where changes in the trigger output signals are initiated.

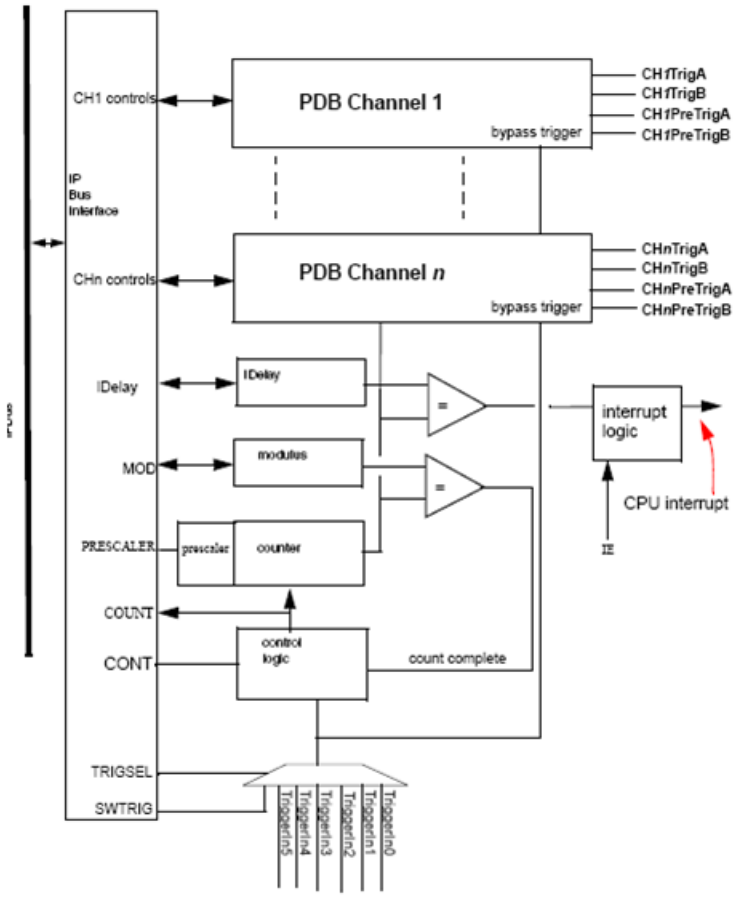

Figure 2. n Channel PDB block diagram

Pre-triggerA and Pre-triggerB are used to precondition the ADC blocks two bus clock periods prior to the actual measurement trigger. The Pre-Trigger signals are used to specify what signal will be sampled next. When PreTriggerA and TriggerA are asserted, the ADC conversion is triggered with set A of the control and results registers. When PreTriggerB and TriggerB are asserted, the ADC conversion is triggered with set B of the control and results register.

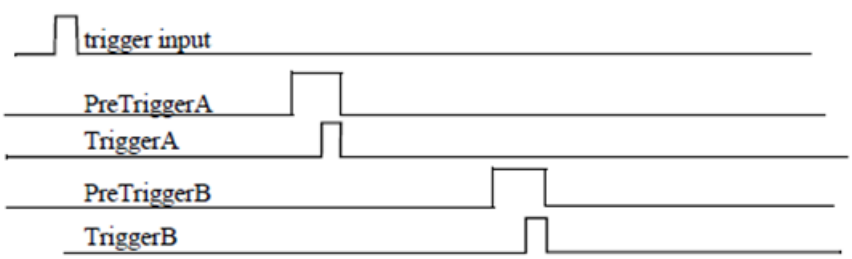

Figure 3. Decoupled A and B trigger generation

The PDB Channeln, PreTriggerA, and PreTriggerB are connected to ADCn Trigger Select Events ADHWTSA and ADHWTSB correspondingly. Either TriggerA or TriggerB can trigger the ADC conversion. When TriggerA triggers the ADC conversion, control and results register set A is used. When TriggerB triggers the ADC conversion, control and results register set B is used.

1.3 Error Conditions

The Delay A and Delay B registers must be configured to make the next trigger asserted after the previous ADC conversion is finished. When one conversion triggered by TriggerA is in progress, the TriggerB output is suppressed until the ADCnSC1A_COCO bit is set. If Delay B is timed-out during the ADC conversion triggered by TriggerA, the Sequence Error bit PDBCHnSC_ERRB will be set, see Figure 4.

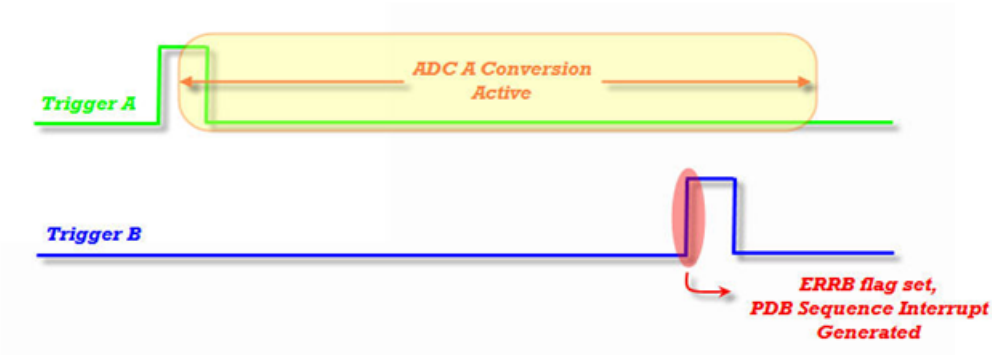

Figure 4. PDB error B

When one conversion, triggered by TriggerB is in progress, the TriggerA output is suppressed until the ADCnSC1B_COCO bit is set. If Delay A is timed out during the ADC conversion triggered by TriggerB, the Sequence Error bit PDBCHnSC_ERRA will be set.

Figure 5 shows the normal conversion without errors.

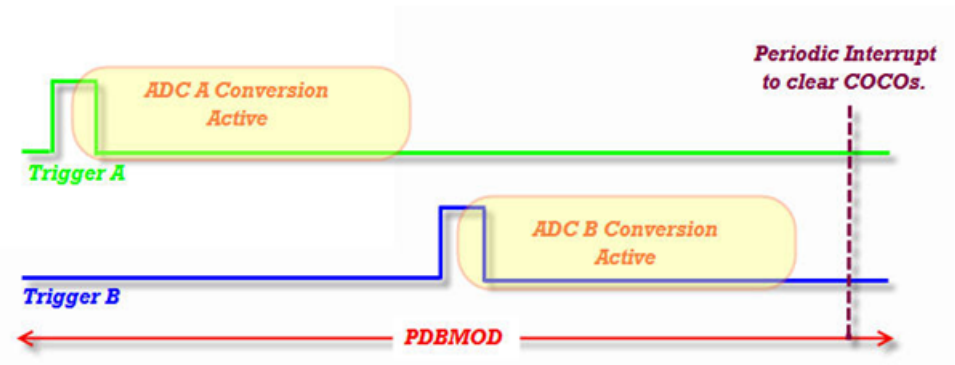

Figure 5. Conversion without PDB error

2 Software Driver Description

The PDB driver is provided as C code files. You can add these files to your applications. With the integration of the PDB driver, PDB driver APIs can be called to use the PDB functionality in your application.

There are three files associated with the PDB driver. The following is a brief description:

pdb.h—It contains all the high level API declarations and the various macros to be used in the functions. It defines the structure of the various PDB registers.

pdb.c—It is the main file for the driver. It contains the various high level API definitions.

2.1 pdb.h

NOTE

The macros provided are passed as arguments to the respective functions to get the required configuration They are explained in detail in Section 2.2 pdb.c

Table 1. Macros

Macros	Descriptions
#define PDB_PRESCALER_1	There are macros used to select the pre-scaler for the pdb clock.
#define PDB_PRESCALER_2	
#define PDB_PRESCALER_4	
#define PDB_PRESCALER_8	
#define PDB_PRESCALER_16	
#define PDB_PRESCALER_32	
#define PDB_PRESACLER_64	
#define PDB_PRESCALER_128	
#define PDB_CONT_MODE	Used to select between running the PDB in continuously or in one shot. In continuous mode, the PDB is configured to provide trigger continuously after the fixed duration.
#define PDB_ONE_SHOT_MODE	
#define PDB_COMP1_TRIG_SEL	Can be used to select the input trigger for the PDB module. The input trigger for the PDB block can be the comparator output, external trigger, or software trigger.
#define PDB_COMP2_TRIG_SEL	
#define PDB_EX_TRIG_SEL	
#define PDB_SW_TRIG_SEL	
#define PDB_IDELAY_INT_ENABLE	Used to enable or disable the idelay interrupt for the PDB module. This feature can be used to schedule an independent interrupt at some point in the PDB cycle.
#define PDB_IDELAY_INT_DISABLE	
#define PDB_CHANNEL_1_TRIG_A	Used to specify the type of error occurred. If a conversion is triggered by macros.
#define PDB_CHANNEL_1_TRIG_B	
#define PDB_CHANNEL_2_TRIG_A	
#define PDB_CHANNEL_2_TRIG_B	
#define PDB_IDELAY_INTERRUPT	

2.2 pdb.c

It contains the definition of various functions to configure and use the various features of the PDB.

2.2.1 PDB_Init

Description:

This function initializes the specific PDB interface by configuring the internal registers. The function is used to set the PDB clock prescaler, enable and disable the interrupt, select the mode, and the type of trigger required to the PDB.

Prototype:

void PDB_Init(unsigned char Prescaler, unsigned char Continuous_OneShot_Mode, unsigned char Trig_Select, unsigned char Idelay_Interrupt_Enable, void (*p)(unsigned char))

Input parameters:

- Prescaler—To select the prescaler of the PDB clock using the macros
  PDB_PRESCALER_1, PDB_PRESCALER_2, PDB_PRESCALER_4,
  PDB_PRESCALER_8, PDB_PRESCALER_16
  PDB_PRESCALER_32, PDB_PRESCALER_64,
  PDB_PRESCALER_128

- Continuous_OneShot_Mode—To select between the continuous trigger mode or one shot mode using the macros.
  PDB_CONT_MODE, PDB_ONE_SHOT_MODE

- Trig_Select—To select the trigger select for the PDB module between the external, internal, comparator 1 output, and comparator 2 output by using the following macros:
  PDB_COMP1_TRIG_SEL, PDB_COMP2_TRIG_SEL
  PDB_EX_TRIG_SEL, PDB_SW_TRIG_SEL

- Idelay_Interrupt_Enable—To enable or disable the idelay interrupt by using the macros:
  PDB_IDELAY_INT_ENABLE, PDB_IDELAY_INT_DISABLE

- p—Callback function address passed as input. The user can either pass the address of the function to be called in case of an interrupt or it can pass 0 if no callback function is required.
- char—Argument to the callback function. It specifies which type of PDB interrupt has occurred.

Output parameters:

None

Example:

```
void callback(void)
      {
       }
PDB_Init(PDB_PRESCALER_2, PDB_CONT_MODE,
              PDB_SW_TRIG_SEL, PDB_IDELAY_INT_ENABLE,
              &callback)
```

Initializes PDB with peripheral clock/2 as the PDB clock, continuous trigger mode, software trigger select, idelay interrupt enabled, and the address of the callback function passed.

NOTE

The Callback function is created by the user according to the user's requirements.

2.2.2 PDB_Set_Mod_Idelay

Description:

This function is used to select the mod value of the PDB counter and the idelay value, after which the interrupt to the MCU is required.

Prototype:

void PDB_Set_Mod_Idelay(unsigned int Mod_Value, unsigned int Idelay)

Input parameters:

- Mod_Value—Enter the mod value of the counter
- Idelay—Enter the idelay value, then the interrupt is required for the MCU

Output parameters:
None

Example:
PDB_Set_Mod_Idelay(0xFFFF,0XFFFF)
Configures the PDB with mod and idelay values.

2.2.3 PDB_Set_Ch1_Delay_A

Description:
This function sets the channel1 delay then pretrigger A is required and enables the interrupt for pdb error A on channel 1.

Prototype:
void PDB_Set_Ch1_Delay_A(unsigned int ChannelA_Delay)

Input parameters:
- ChannelA_Delay—Enter the delay required for pretrigger A in pdb channel1

Output parameters:
None

Example:
PDB_Set_Ch1_Delay_A(0x000F)

2.2.4 PDB_Set_Ch1_Delay_B

Description:
This function sets the delay of channel1 then pretrigger B is required and enables the pdb error B on channel 1.

Prototype:
void PDB_Set_Ch1_Delay_B(unsigned int ChannelB_Delay)

Input parameters:
- ChannelB_Delay—Enter the delay required for pretrigger B in pdb channel1

Output parameters:
None

Example:
PDB_Set_Ch1_Delay_B(0x00FF)

2.2.5 PDB_Set_Ch2_Delay_A

Description:
This function sets the channel2 delay then pretrigger A is required and enables the pdb error A on channel 2.

Prototype:
void PDB_Set_Ch2_Delay_A(unsigned int ChannelA_Delay)

Input parameters:
- ChannelA_Delay—Enter the delay required for pretrigger A in pdb channel2

Output parameters:
None

Example:
PDB_Set_Ch2_Delay_A(0x000F)

2.2.6 PDB_Set_Ch2_Delay_B

Description:
This function sets the channel1 delay then pretrigger B is required and enables the pdb error B on channel 2.

Prototype:
void PDB_Set_Ch2_Delay_B(unsigned int ChannelB_Delay)

Input parameters:
ChannelB_Delay—Enter the delay required for pretrigger B in pdb channel 2

Output parameters:
None

Example:
PDB_Set_Ch2_Delay_B(0x00FF)

2.2.7 PDB_Start_SW_Trigger

Description:
This function starts the conversion by giving software trigger.

NOTE

Only if trigger select is PDB_SW_TRIG_SEL

Prototype:
void PDB_Start_SW_Trigger(void)

Input ´parameters:
None

Output parameters:
None

Example:
PDB_Start_SW_Trigger()

2.2.8 Interrupt Subroutines

There are two types of interrupts:

- Idelay interrupt—It occurs when the value of PDB counter matches the idelay value
- PDB error interrupt—There are two types of errors as described in Error Conditions

There are two interrupt subroutines as follows:

pdb_isr

Description:
This subroutine is called when the pdb counter value matches the idelay value.

Prototype:

void interrupt VectorNumber_Vpdb pdb_isr(void)

Input parameters:
None

Output parameters:
None

pdb_isr_err

Description:
This subroutine is called either errA or errB and occurs on either PDB channel1 or channel2.

Prototype:
void interrupt VectorNumber_Vpdb pdb_isr(void)

Input parameters:
None

Output parameters:
None

NOTE

If the address of the callback function is passed in the PDB_Init function, the interrupt subroutine then jumps to the callback function and the user can write the action taken in the callback function.

3 Assumptions

The descriptions in this document assumes the person reading it has full knowledge of all the configuration registers of all the blocks in 9S08GW64, especially LCD and ICS(Internal Clock Source) blocks.

4 Use Case

Assuming that the clock settings are done and the bus clock is running on 20 Mhz. Include pdb.h in the main file.

Step 1—Initialize the PDB with the required configuration:
PDB_Init(PDB_PRESCALER_2,PDB_ONE_SHOT_MODE
PDB_SW_TRIG_SEL,PDB_IDELAY_INT_DISABLE,&callback);

It initializes the PDB module with prescaler 2, one shot mode, software trigger select, idelay interrupt disabled, and the address of the callback function is passed.

Step 2—Sets the mod value and idelay value
PDB_Set_Mod_Idelay(0xFFFF,0XFFFF);

Step 3—Sets the delay after which pretrigger A and pretrigger B is required on channel 2.
PDB_Set_Ch2_Delay_A(0x000F);
PDB_Set_Ch2_Delay_B(0xFFFF);

Step 4—Starts the PDB module by providing the software trigger
PDB_Start_SW_Trigger();

5 Conclusion

This driver provides a software base for applications that need the implementation of the PDB.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

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)
www.freescale.com/support

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 China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and sofware implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor prodcuts are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to http://www.freescale.com/epp.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2010 Freescale Semiconductor, Inc.

Document Number: AN4163
Rev. 0, 8/2010