Freescale Semiconductor
Application Note

Document Number: AN4746
Rev. 0, 06/2013

High-Resolution PWM Generation Using MC56F82xx, MC56F84xxx, MC56F823xx, and MC56F827xx DSC Families

by: Pavel Grasblum

1 Introduction

Pulse width modulation (PWM) is a basic technique for average control of voltage/current in many applications like motor control, switched mode power supplies, lighting, wireless charging, audio amplifiers, and many others. This technique represents efficient method to convert one level of voltage/current to another level. The principle of pulse width modulation can be seen in Figure 1 with an example of voltage converter. The switching network generates voltage pulses, which usually have either constant frequency and variable pulse width, or constant pulse width and variable frequency. The amplitude of voltage pulses corresponds to the value of input source voltage. The voltage pulses are further rectified and filtered to get smooth output voltage; the average value of this rectified and filtered output voltage corresponds to duty cycle of the generated voltage pulses. In applications such as heating and motor control, the voltage pulses are not rectified or filtered, but directly applied on the load.

Contents

© 2013 Freescale Semiconductor, Inc.

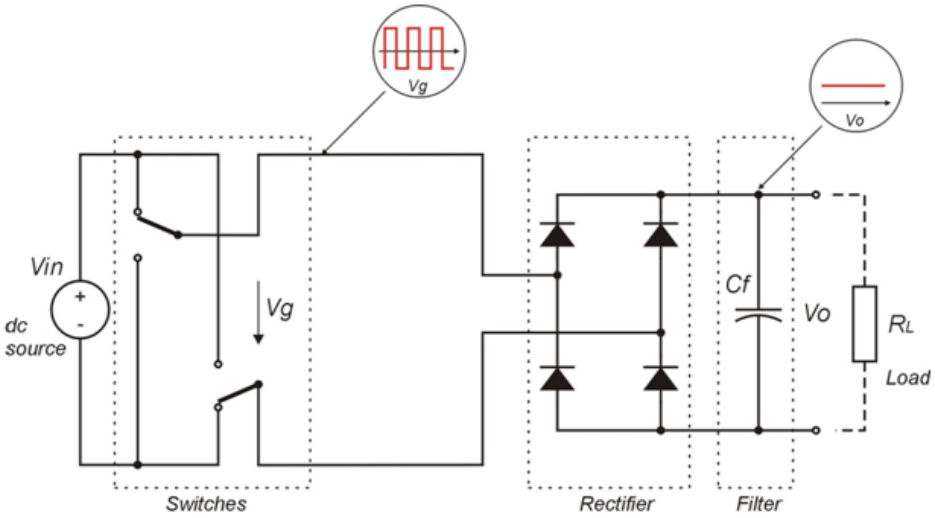

Figure 1. Principle of PWM in voltage converter

The frequency used for PWM varies in wide range. It can be several seconds for heating or electric stove applications, 50/60 Hz for triac control applications, 10–20 kHz for motor control applications, and hundreds of kilohertz for switched mode power supplies. The microcontrollers use more or less complex timer/PWM modules for PWM signal generation. The basic time unit for PWM module is input clock of the PWM module. The input clock is usually equal to the peripheral bus clock but some MCUs can feed PWM module by multiples of peripheral bus clock. The input clock of PWM module defines the resolution of generated PWM signal. The PWM resolution is an important parameter for stable close loop control. Ideally, the PWM resolution must be comparable to analog-to-digital converter (ADC) resolution.

The result of low PWM resolution can be seen in Figure 2. On the left side, there is a PWM signal generation with low resolution. The control loop tries to keep output close to the reference value. But since the one step duty cycle change in PWM module causes high step in the output, the output signal oscillates around the reference level. On the right side, there is the PWM signal generation with high resolution so the output signal ripple is negligible.

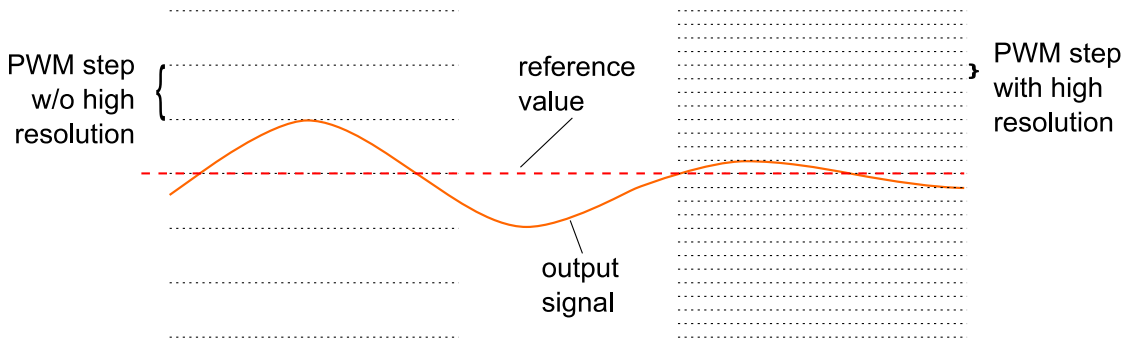

Figure 2. Impact of PWM resolution on output signal

For low and middle PWM frequencies, the PWM resolution is sufficient in most of the MCUs available in market. For example, considering MCU with peripheral bus clock 100 MHz and PWM signal frequency 10 kHz, the PWM resolution is 10, 000, which means more than 13 bits. The required minimal PWM resolution depends on the application requirements but at least 10-bit resolution is considered as a sufficient value. For the second example, consider the PWM signal with frequency 200 kHz. The resolution for this PWM signal at the same previous peripheral bus clock is less than 9 bits. So, low PWM resolution may lead to unwanted high output ripple as shown in Figure 2.

To achieve the same PWM resolution as in the previous case, the input clock to the PWM module has to be at least 2 GHz, which is not feasible for technologies used for the manufacture of MCUs in today's world. However, for applications like switched mode power supplies requiring high PWM frequencies (> 100 kHz), some MCUs use special techniques to achieve such high resolution without using very high input clock frequency to the PWM/timer module. The following sections describe the implementation of a high-resolution frequency and duty cycle PWM generation using Freescale Digital Signal Controllers (DSC) families, MC56F82xx, MC56F84xxx, MC56F823xx, and MC56F827xx.

2 Fractional delay logic

Starting from the MC56F82xx family, the newer DSC devices use eFlexPWM module for PWM generation. This module brings much more flexibility for PWM signal generation by providing:

- Individual time base for each PWM submodule
- Individual control of every edge
- High-resolution frequency and duty cycle generation.

These improvements make eFlexPWM module more suitable for digital control of power conversion applications.

The high-resolution PWM generation feature is implemented by fractional delay logic. The placement of this block in eFlexPWM signal path is shown in Figure 3. The fractional logic is placed at the end of the signal path just in the front of output logic. The fractional logic is capable of dividing the input digital clock into next 32 fractions (5 bits) and thus increase resolution of the generated PWM signals. The final PWM resolution is either 520 ps for 60 MHz eFlexPWM module input clock or 312 ps for 100 MHz, based on the selected DSC. The operating principle of this block can be illustrated in Figure 4. There is one fractional logic block for each PWM output allowing to delay both the rising and falling edges individually. Moreover, the fractional logic also allows to control the PWM period with high resolution as described in the following sections.

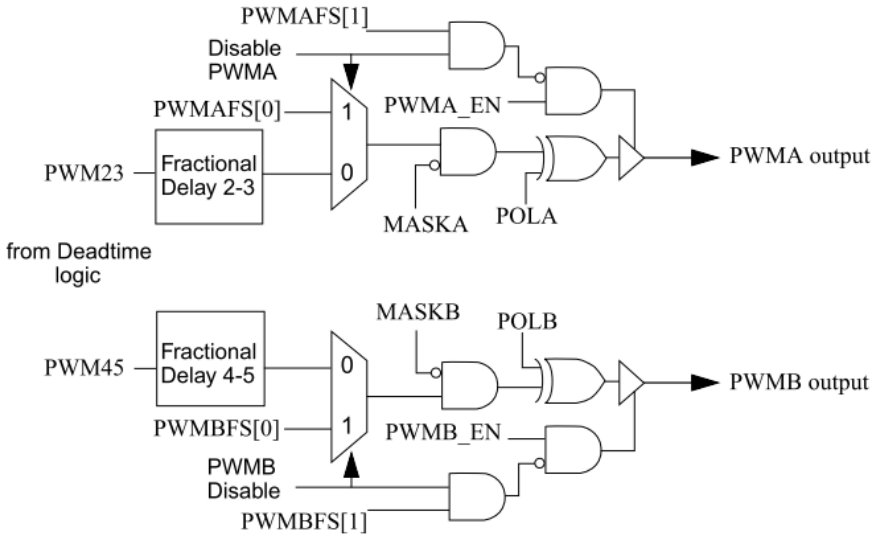

Figure 3. Fractional delay block in eFlexPWM module

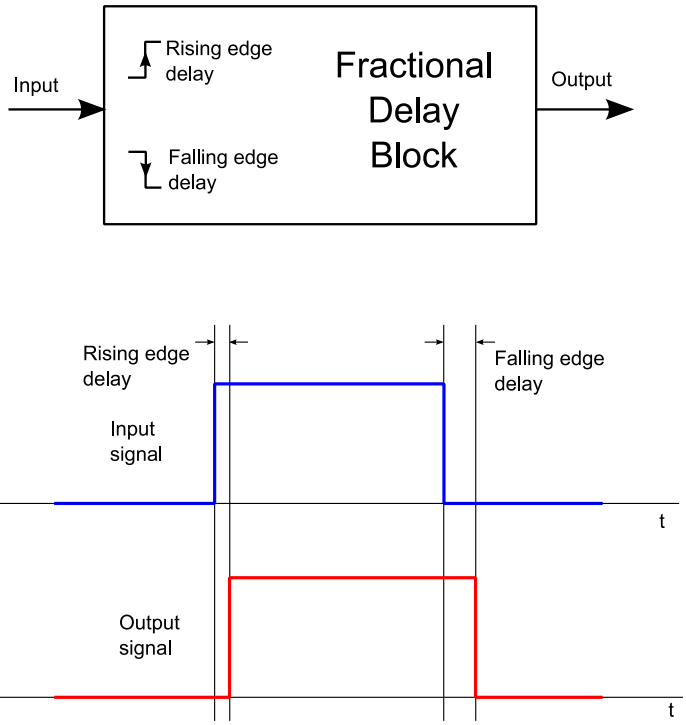

Figure 4. Operating principle of fractional delay block

2.1 High-resolution duty cycle PWM generation

The generation of high-resolution duty cycle PWM signal is demonstrated in Figure 5. To simplify this figure, only 2-bit fractional delay logic is considered while in reality, the eFlexPWM module

implements 5-bit fractional delay logic. The top of the figure shows counter counting up with modulo 4. Considering digital input clock t_{clk}, the PWM channel can generate signal with 0, 25, 50, 75, and 100% duty cycle. Using 2-bit fractional delay logic, every digital clock t_{clk} can be divided into the next four fractions. This means that minimal step will be 6.25% instead of original 25%. Therefore, the PWM signal with, for example, a duty cycle of 68.75% can be generated, as shown in Figure 5. This signal is represented by duty cycle value 2:3, where value 2 before colon means digital value of the duty cycle and value 3 after colon is fractional part of the duty cycle. For this setting, the eFlexPWM generates blue PWM signal t_{timer}, which is derived from digital input clock. This signal enters the fractional delay logic, which delays falling-edge by three fractions. The output of fractional logic results in red signal t_{PWM}, which has final required duty cycle of 68.75%.

Figure 5 depicts edge-aligned PWM signal generation; therefore the rising-edge is kept aligned with the start of PWM period and fractional part has zero value. Generally, the rising-edge can be delayed in the range of 0-31 fractions; so phase shift can also be implemented with high resolution. It is also important to note that for high-resolution duty cycle generation, the fractional delays remain constant period by period for both rising and falling edges at the same duty cycle.

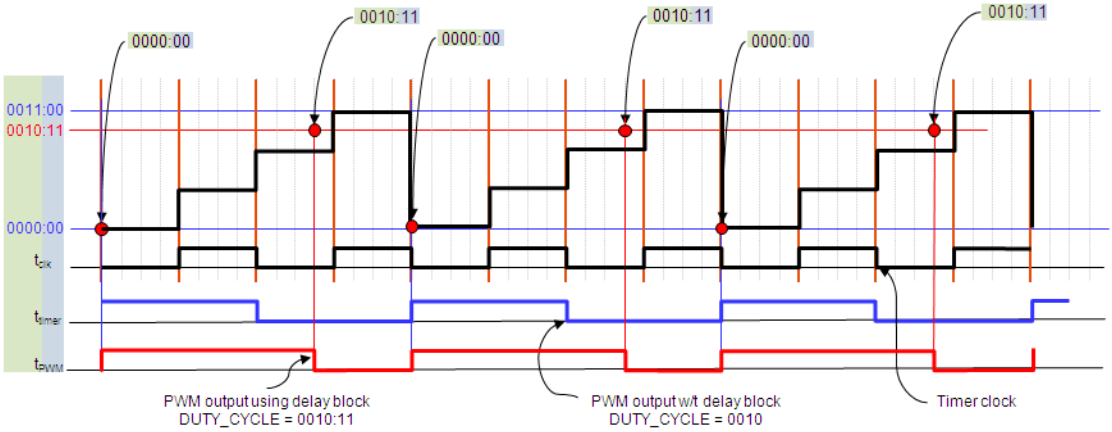

Figure 5. High-resolution PWM duty cycle generation

2.2 High-resolution frequency PWM generation

Figure 6 depicts the generation of PWM signal with 50% duty cycle and period 4:2, which means that digital part of period is equal to 4 and fractional part is equal to 2. The fractional delay logic works in the same way delaying the edges according to settings; however there is one important difference. Figure 6 shows that the fractional value of each edge changes with every new edge. The first edge in Figure 6 has no fractional delay, the second one has delay of one fraction, the third one has delay of two fractions, and so on. The fifth edge has delay corresponding to the whole digital clock; therefore one more digital clock is generated. It means that the delay in fractional delay block changes and need to be calculated for high-resolution frequency generation.

The new edge position calculation can be provided by the software but in many cases, it can be very time-consuming, firstly due to the generation of high frequency, and secondly, the delays have to be calculated for many PWM outputs based on selected topology. Therefore, the fractional delay logic is also equipped by circuit, which ensures automatic fractional edge placement without user intervention. This circuit is enabled once the fractional delay logic is enabled for comparator VAL1, which defines PWM period together with the INIT register in the eFlexPWM module (PWMx_SMnINIT), where x

denotes the PWM channel and n denotes the specific PWM submodule. The principle of edge adjustment operation can be seen in Figure 7. This circuit is invisible for the user and doesn't need any user attention.

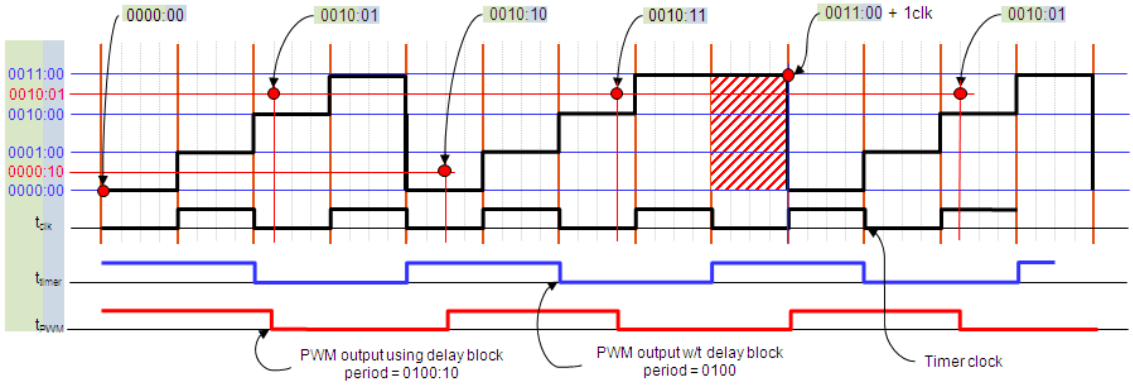

Figure 6. High-resolution PWM frequency generation

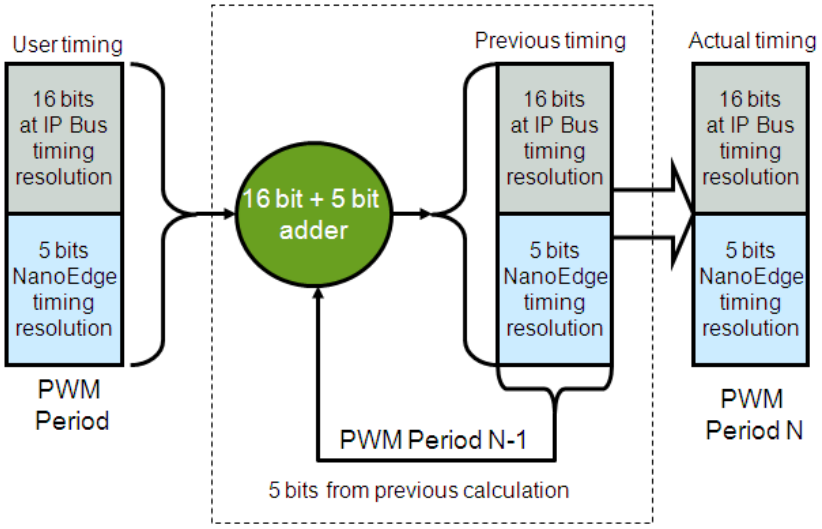

Figure 7. Hardware support for automatic new edge position calculation

2.3 Complementary signal generation with fractional delay logic

Since the fractional delay logic is located behind complementary and deadtime logic, the complementary signal generation can still be used without any limitation. This ensures safe hardware generation of complementary signals with deadtime, but the user needs to take care about fractional edge positioning of the complementary signal.

For example, if PWM23 signal is used as the source for complementary signal generation, the user has to set fractional part not only into PWMx_SMnFRACVAL2 and PWMx_SMnFRACVAL3 but also into PWMx_SMnFRACVAL4 and PWMx_SMnFRACVAL5. This is valid for MC56F82xx DSC family. The newer DSC families like MC56F84xxx, MC56F823xx, and MC56F827xx improve this functionality. If complementary mode is selected, the values in registers PWMx_SMnFRACVAL2 and PWMx_SMnFRACVAL3 are automatically copied into PWMx_SMnFRACVAL4 and

PWMx_SMnFRACVAL5 registers in the way that PWMx_SMnFRACVAL2 = PWMx_SMnFRACVAL5 and PWMx_SMnFRACVAL3 = PWMx_SMnFRACVAL4.

The resolution of deadtime is given by digital clock only, but if the user needs higher resolution, it can be achieved by proper setting of fractional delay for both the rising edges. For example, setting deadtime to 10 digital clocks and 5 fractions can be achieved by setting PWMx_SMnDTCNT0 = 10 and PWMx_SMnFRACVAL2 = PWMx_SMnFRACVAL5 + 5. Similarly, PWMx_SMnDTCNT1 = 10 and PWMx_SMnFRACVAL4 = PWMx_SMnFRACVAL3 + 5.

2.4 Using fractional delay logic in the software

The fractional delay logic is controlled by two registers PWMx_MCTRL2 and PWMx_SMnFRCTRL.

- PWMx_MCTRL2 controls the behaviour of fractional delay logic in case of PLL clock loss.
    - The bit 0 controls whether the fractional delay logic is disabled or not in case of PLL clock loss.
    - The bit 1 locks the setting of the register until the next reset.
- PWMx_SMnFRCTRL enables the operation of fractional delay logic.
    - FRAC_PU field (PWMx_SMnFRCTRL[FRAC_PU]) powers up the fractional delay logic.
    - FRAC45_EN, and FRAC23_EN fields (PWMx_SMnFRCTRL[FRAC45_EN] and PWMx_SMnFRCTRL[FRAC23_EN) enable fractional functionality individually for each PWM output (PWM23, PWM45)
    - FRAC1_EN field (PWMx_SMnFRCTRL[FRAC1_EN]) enables fractional functionality for PWM period/frequency.

Enabling fractional functionality every compare value except value in the Value Register 0 (PWMx_SMnVAL0) consists of two parts.

- The first part represents digital part of compare value and is stored in PWMx_SMnVALy register, where y = 1, 2…5.
- The second part represents fractional part of compare value stored in PWMx_SMnFRACVALy register, where y = 1, 2..5.

Both the parts create 21-bit value defining edge positioning or PWM period. The PWMx_SMnVALy and PWMx_SMnFRACVALy registers sit next to each other; so they can be accessed as single 32-bit value.

Even if the edge position or PWM period is defined by 21 bits, the calculation can be still kept in 16-bit range. The fractional delay logic is usually used to generate high frequency. For example, 200 kHz PWM signal has resolution less than 9 bits (considering 100 MHz input clock). It means that 7 bits will be unused in the 16-bit variable. Therefore, the 16-bit variable containing edge position can be defined as shown in Figure 8. The five least significant bits define the fractional part and the rest 11 bits define the digital part. Before the final value of edge position is written into the value registers

(PWMx_SMnVALy, PWMx_SMnFRACVALy), the 16-bit value is shifted by five bits right and written as 32-bit value into both the value registers by 32-bit move instruction. This approach efficiently allows keeping all PWM update calculations in 16-bit range.

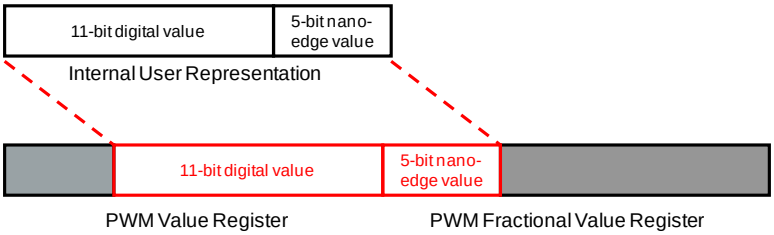

Figure 8. 16-bit representation of high-resolution duty cycle

2.4.1 Software example of high-resolution PWM duty cycle generation

The following sample code demonstrates the generation of PWM signals with high resolution using submodule 0, output PWM0A, output frequency 200 kHz, and DSC core frequency 100 MHz on MC56F84879 DSC.

```
/* variable declaration */
unsigned int duty_cycle;

/* fractional delay logic initialization */
PWMA_MCTRL2 = 0x0003;      // Disable fractional logic when PLL lock lost
PWMA_SM0FRCTRL = 0x0104; // FRAC_PU=1, FRAC23_EN=1

/* Period initialization */
PWMA_SM0INIT = 0;
PWMA_SM0VAL1 = 499;        // 100MHz/200kHz – 1
PWMA_SM0VAL2 = 0;          //Edge aligned PWM -> rising edge at beginning of period
PWMA_SM0VAL3 = 0;          // 0% duty cycle during PWM initialization

/* example of PWM update */
asm
{
  move.w  duty_cycle, A                // duty_cycle can be in range 0 – 16000 (500*32)
  asrr.l  #0x5,A                       // move 5 LSB bits to fractional register position
  move.l  A10, X:FPWMA_SM0FRACVAL3     // write to VAL3 and FRACVAL3 registers
  bfset   #0x0001, X:FPWMA_MCTRL // Set LDOK (update VAL registers for submodule 0
}
```

Note: This code doesn’t contain whole eFlexPWM module initialization. There are parts related to fractional delay logic initialization, PWM period definition and example of duty cycle update only.

3 Conclusion

The fractional delay block is essential for many power conversion applications. This application note describes the implementation of high-resolution duty cycle and frequency PWM generation used on

Freescale digital signal controllers in detail including the principle of fractional delay block operation, difference between high-resolution duty cycle and frequency generation, and using fractional delay block in the software. Finally, a simple software example is presented demonstrating efficient using of 16-bit variable even for high-resolution PWM signal generation.

4 References

For more information, see the following documents, available on **freescale.com**.

- MC56F825XRM: MC56F825x/4x Reference Manual
- MC56F8455XRM: MC56F8455x Reference Manual
- MC56F8458XRM: MC56F8458x Reference Manual
- MC56F847XXRM: MC56F847xx Reference Manual
- MC56F823XXRM: MC56F823xx Reference Manual
- MC56F827XXRM: MC56F827xx Reference Manual

5 Revision history

Revision number	Date	Substantive changes
0	06/2013	Initial release

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. Freescale reserves the right to make changes without further notice to any products herein.

Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale 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 does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale, and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners.

© 2013 Freescale Semiconductor, Inc. All rights reserved.

Document Number: AN4746
Rev. 0, 06/2013
June 3, 2013