

利用MC56F82xx、MC56F84xxx、 MC56F823xx和MC56F827xx DSC系列产生高 分辨率PWM信号

作者: Pavel Grasblum

1 简介

脉宽调制 (PWM) 是基于电压/电流均值控制的基本技术, 电机控制、开关模式电源、照明、无线充电、音频放大器及许多其他应用中都会用到此技术。这种技术能够有效地将电压/电流转换为另一个电平。脉宽调制的原理可从图 1 中的电压转换器示例看出。开关网络产生电压脉冲, 通常该脉冲要么是频率恒定但脉宽可变, 要么是脉宽恒定但频率可变。电压脉冲的幅度与输入源的电压值对应。电压脉冲经过进一步整流和滤波后, 形成平滑的输出电压; 此输出电压的平均值与所产生的电压脉冲的占空比对应。在电机控制等应用中, 电压脉冲不进行整流或滤波, 而是直接作用于负载。

目录

1	简介.....	1
2	细分延迟逻辑.....	3
2.1	高分辨率 PWM 占空比的产生.....	5
2.2	高分辨率 PWN 频率的产生.....	5
2.3	采用细分延迟逻辑产生互补信号....	6
2.4	在软件中使用细分延迟逻辑.....	7
3	结论.....	8
4	参考.....	9
5	修订历史记录.....	9

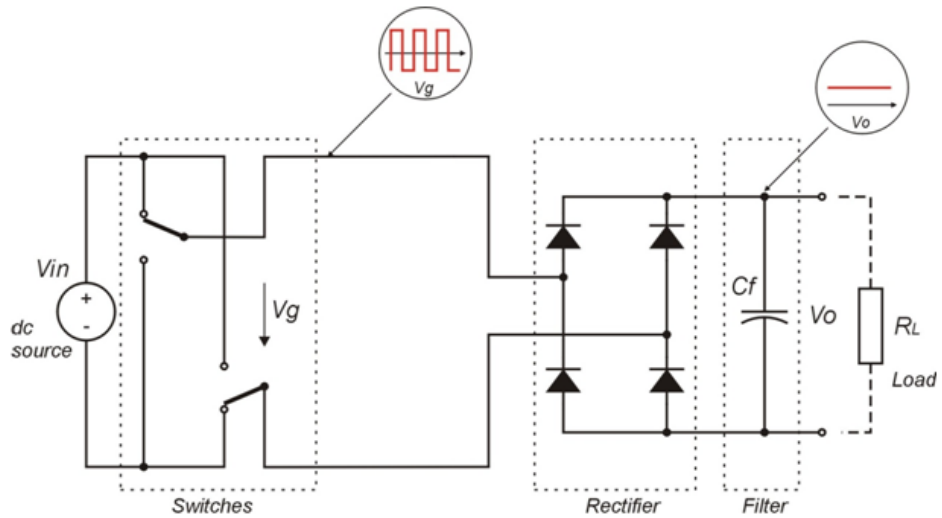


图 1. 电压转换器中的 PWM 原理

PWM 使用的频率范围很宽。加热或电炉应用可以是数秒一次可控硅应用为 50/60 Hz，电机控制应用为 10–20 kHz，开关电源为数百千赫。微控制器利用定时器/PWM 模块产生 PWM 信号。PWM 模块的定时基本单元是其输入时钟。输入时钟通常与外设总线时钟相同，但某些 MCU 可将外设总线时钟的频率数倍倍频后提供给 PWM 模块。PWM 模块的输入时钟决定了产生 PWM 信号的分辨率。PWM 分辨率是闭环控制是否稳定的重要参数。理想情况下，PWM 分辨率必须与模数转换器（ADC）分辨率相当。

低 PWM 分辨率的结果可从图 2 看出。左侧产生的是低分辨率 PWM 信号。控制环路试图让输出值接近参考值。但是，PWM 模块的占空比一旦改变就会引起输出大幅变化，因此输出信号会围绕参考值波动。右侧产生的是高分辨率 PWM 信号，因此输出信号纹波可忽略不计。

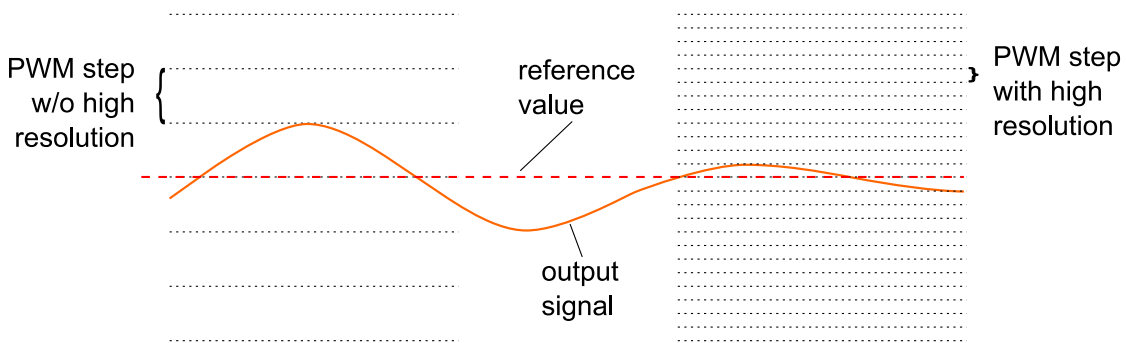


图 2. PWM 分辨率对输出信号的影响

对于低频中频 PWM，市场上大部分 MCU 的 PWM 分辨率是足够的。例如，若 MCU 的外设总线时钟为 100 MHz、PWM 信号频率为 10 kHz，则 PWM 分辨率为 10,000，即超过 13 位。所需的最低 PWM 分辨率取决于应用要求，但一般认为至少是 10 位分辨率才足够。在第二个例子中，PWM 信号的频率为 200 kHz。若外设总线时钟与上例相同，则此 PWM 信号的分辨率低于 9 位。因此，低 PWM 分辨率可能导致出现不需要的输出纹波，如图 2 所示。

为获得与上例相同的 PWM 分辨率，PWM 模块的输入时钟必须至少为 2 GHz，这对于当今 MCU 制造所用的技术而言不太可行。但是，对于开关电源等需要高频 PWM (> 100 kHz) 的应用，某些 MCU 可利用特殊技术实现如此高的分辨率，并且 PWM/定时器模块无需使用非常高的输入时钟频率。以下章节介绍如何利用飞思卡尔数字信号控制器(DSC)系列 MC56F82xx、MC56F84xxx、MC56F823xx 和 MC56F827xx 产生高分辨率的频率和占空比 PWM 信号。

2 细分延迟逻辑

MC56F82xx 系列后的新 DSC 器件利用 eFlexPWM 模块产生 PWM 信号。此模块通过下述方式使 PWM 信号的产生更为灵活：

- 各 PWM 子模块采用独立的时钟基准
- 独立控制每个边沿
- 产生高分辨率的频率和占空比。

这些改进使得 eFlexPWM 模块更适用于电源转换应用的数字控制。

产生高分辨率 PWM 这一特性通过细分延迟逻辑实现。此模块在 eFlexPWM 信号路径中的放置位置如图 3 所示。细分逻辑位于信号路径的末端，输出逻辑之前。细分逻辑能够将输入数字时钟分频为 32 份（5 位），从而提高所产生 PWM 信号的分辨率。根据所选用的 DSC，最终 PWM 分辨率为 520 ps（60 MHz eFlexPWM 模块输入时钟）或 312 ps（100 MHz 输入时钟）。此模块的工作原理可参见图 4。每路 PWM 输出对应一个细分逻辑块，这使得上升沿和下降沿均可独立延迟。此外，细分逻辑还支持以高分辨率控制 PWM 周期，详情如后续章节所述。

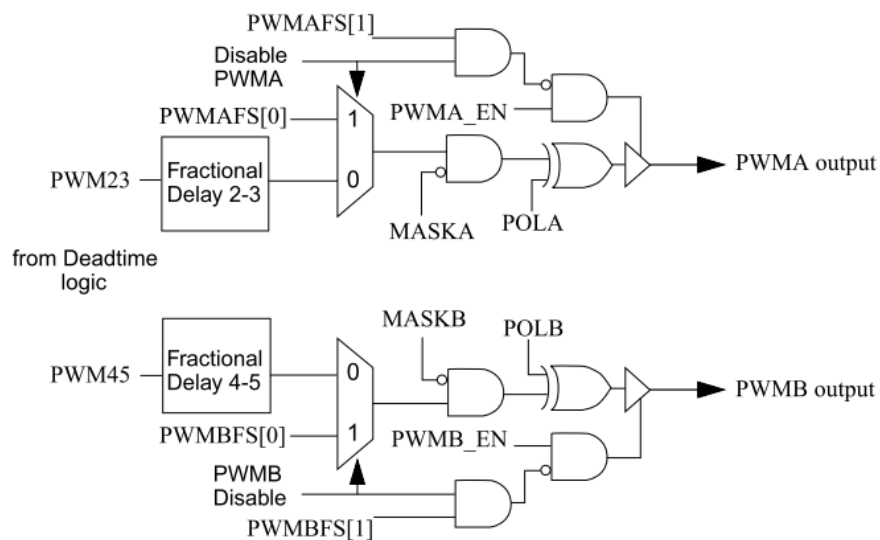


图 3. eFlexPWM 模块中的细分延迟模块

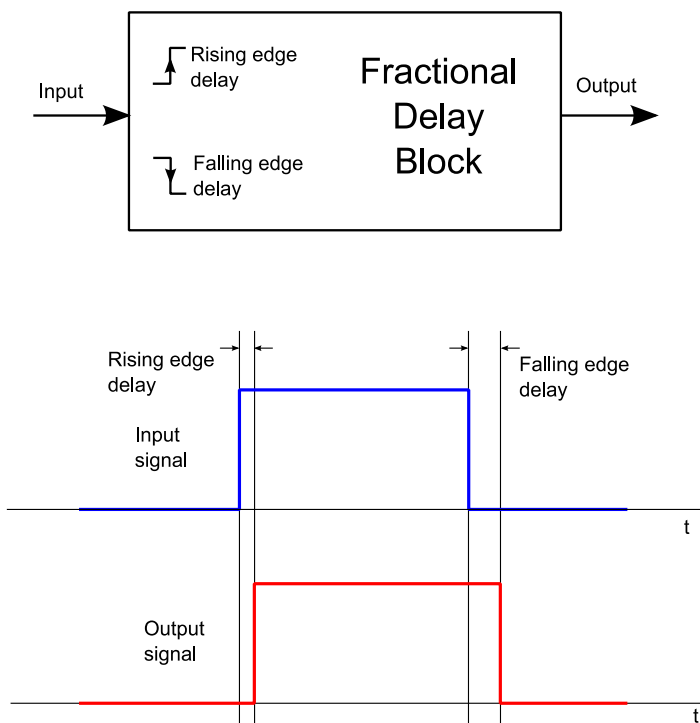


图 4. 细分延迟模块的工作原理

2.1 高分辨率PWM占空比的产生

高分辨率占空比PWM信号的产生过程如图5所示。为了简化此图，仅考虑2位细分延迟逻辑，而实际上，eFlexPWM模块采用5位细分延迟逻辑。图上方显示，计数器以模数为4递增计数。考虑使用数字输入时钟 t_{clk} 。PWM通道可产生占空比为0、25%、50%、75%和100%的信号。采用2位细分延迟逻辑时，每个数字时钟 t_{clk} 可以再细分为四份。这意味着，最小步进为6.25%，而非原来的25%。因此，可以产生占空比为68.75%之类的PWM信号，如图5所示。此信号用占空比值2:3表示，冒号之前的2个数值表示占空比的整数部分，冒号之后的3个数值表示占空比的小数部分。如此设置后，eFlexPWM产生蓝线所示的PWM信号 t_{timer} ，此信号源于数字输入时钟。此信号进入细分延迟逻辑，将下降沿延迟3个小数单位。细分逻辑的输出为红线所示的信号 t_{pwm} ，此信号为最终需要的占空比68.75%。

图5所示为边沿对齐PWM信号的产生过程，因此其上升沿与PWM周期的起始部分保持对齐，小数部分为0。一般而言，上升沿可以延迟0-31个小数单位；所以，在高分辨率下也可以实现相位偏移。还必须注意：对于高分辨率占空比的产生，如果占空比相同，则每个周期上升沿和下降沿的细分延迟均保持不变。

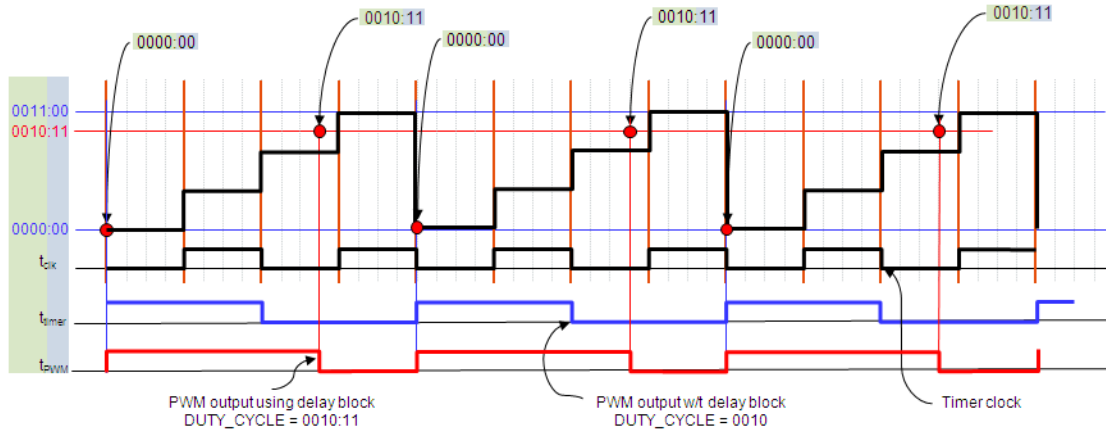


图 5. 高分辨率 PWM 占空比的产生

2.2 高分辨率PWM频率的产生

图6显示了占空比为50%、周期为4:2（即周期的数字部分为4，小数部分为2）的PWM信号的产生过程。细分延迟逻辑以相同方式工作，根据事先设置延迟边沿，但有一个重要差异。如图6所示，每个新边沿的细分值并不相同。图6中的第一个边沿无细分延迟，第二个边沿有一个小数单位的延迟，第三个边沿有两个小数单位的延迟，以此类推。第五个边沿的延迟对应于一整个数字时钟，因此，会多产生一个数字时钟。这说明，细分延迟模块的延迟发生变化，必须计算该延迟才能确定高分辨率频率的产生方式。

新边沿位置的计算可由软件执行，但在很多情况下，这可能非常耗时，原因有二：一是高频信号的产生；二是基于所选拓扑，必须计算许多PWM输出的延迟。因此，配有细分延迟逻辑的电路，无需用户干预即可确保自动安排细分边沿。一旦比较器VAL1启用细分延迟逻辑，此电路就会启用，它与eFlexPWM模块中的INIT寄存器(PWMx_SMnINIT，其中x表示PWM通道，n表示特定PWM

子模块)一同定义PWM周期。边沿调整操作的原理如图7所示。此电路对用户不可见,用户无需执行任何操作。

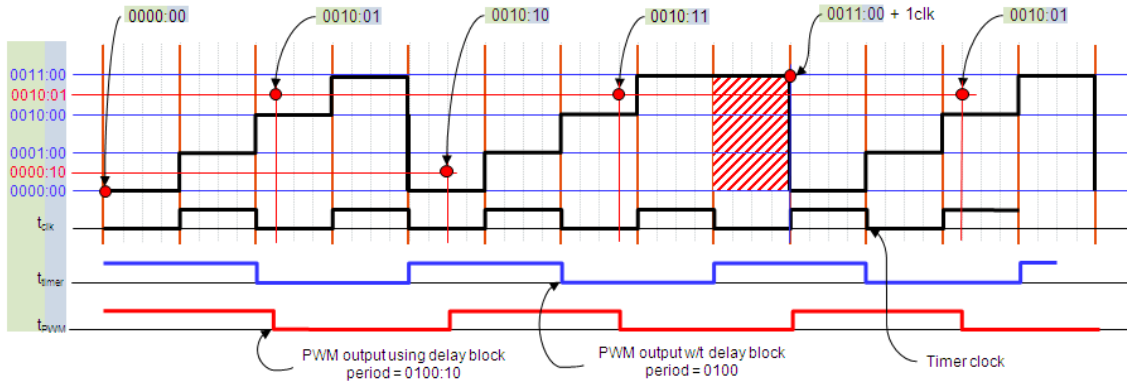


图 6. 高分辨率 PWM 频率的产生

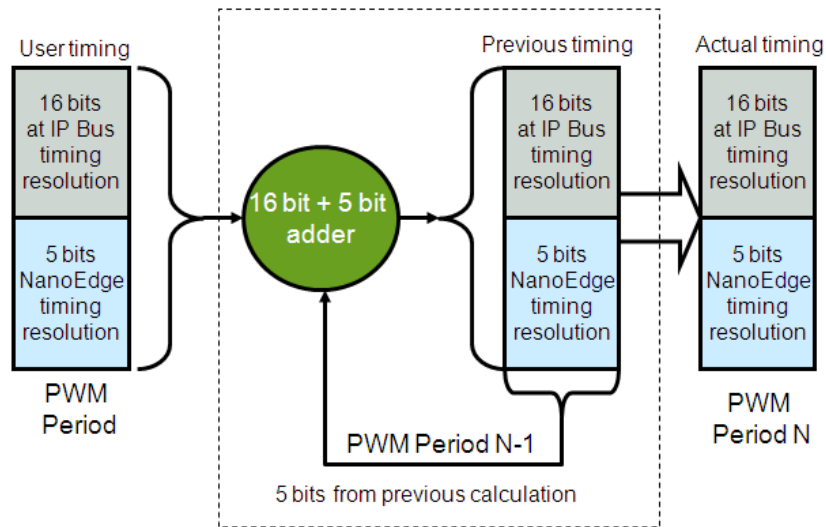


图 7. 自动计算新边沿位置的硬件支持

2.3 采用细分延迟逻辑产生互补信号

由于细分延迟逻辑位于互补和死区逻辑后面,因此仍然可以不受限制地使用产生的互补信号。这样就能确保带死区时间的互补信号能够安全由硬件产生,但用户需要注意细分边沿在互补信号中的位置。

例如,如果PWM23信号用作互补信号产生的来源,则用户不仅要设置PWMx_SMnFRACVAL2和PWMx_SMnFRACVAL3的小数部分,而且要设置PWMx_SMnFRACVAL4和PWMx_SMnFRACVAL5的小数部分。这对MC56F82xx DSC系列有效。MC56F84xxx、MC56F823xx和MC56F827xx等新的DSC系列改进了这一功能。如果选择互补模式,寄存器PWMx_SMnFRACVAL2和PWMx_SMnFRACVAL3中的值会被自动复制到寄存器PWMx_SMnFRACVAL4和PWMx_SMnFRACVAL5中,即PWMx_SMnFRACVAL2 = PWMx_SMnFRACVAL5, PWMx_SMnFRACVAL3 = PWMx_SMnFRACVAL4。

死区分辨率仅由数字时钟决定，但如果用户需要更高的分辨率，可以通过适当设置两个上升沿的细分延迟来实现。例如，要将死区时间设置为10个整数时钟和5个细分单位，可设置 $PWMx_SMnDTCNT0 = 10$ ， $PWMx_SMnFRACVAL2 = PWMx_SMnFRACVAL5 + 5$ 。类似地， $PWMx_SMnDTCNT1 = 10$ ， $PWMx_SMnFRACVAL4 = PWMx_SMnFRACVAL3 + 5$ 。

2.4 在软件中使用细分延迟逻辑

细分延迟逻辑由 $PWMx_MCTRL2$ 和 $PWMx_SMnFRCTRL$ 这两个寄存器控制。

- $PWMx_MCTRL2$ 控制PLL时钟丢失情况下细分延迟逻辑的行为。
 - 位0控制PLL时钟丢失情况下是否禁用细分延迟逻辑。
 - 位1锁定该寄存器的设置，直至下一次复位。
- $PWMx_SMnFRCTRL$ 用于使能细分延迟逻辑的运行。
 - $FRAC_PU$ 区域 ($PWMx_SMnFRCTRL[FRAC_PU]$) 用于为细分延迟逻辑上电。
 - $FRAC45_EN$ 和 $FRAC23_EN$ 区域 ($PWMx_SMnFRCTRL[FRAC45_EN]$ 和 $PWMx_SMnFRCTRL[FRAC23_EN]$) 用于使能各PWM输出 (PWM23和PWM45) 的细分功能。
 - $FRAC1_EN$ 区域 ($PWMx_SMnFRCTRL[FRAC1_EN]$) 用于使能PWM周期/频率的细分功能。

使能细分功能后，所有比较值都包括两部分，值寄存器0 ($PWMx_SMnVAL0$) 中数值除外。

- 第一个部分是比较值的整数部分，存储在 $PWMx_SMnVALy$ 寄存器中，其中 $y = 1, 2 \dots 5$ 。
- 第二个部分是比较值的细分部分，存储在 $PWMx_SMnFRACVALy$ 寄存器中，其中 $y = 1, 2 \dots 5$ 。

这两个部分构成一个21位值，用于定义边沿位置或PWM周期。 $PWMx_SMnVALy$ 和 $PWMx_SMnFRACVALy$ 寄存器彼此相邻，因此可作为单个32位值加以访问。

虽然边沿位置或PWM周期由21位定义，但计算仍可保持在16位范围内。细分延迟逻辑通常用于产生高频率。例如，200 kHz PWM信号的分辨率低于9位（假设输入时钟为100 MHz）。这意味着，该16位变量中有7位不会使用。因此，包含边沿位置的16位变量可按图8所示进行定义。5个最低有效位定义小数部分，其余11位定义整数部分。在边沿位置的最终值写入到值寄存器 ($PWMx_SMnVALy$ 、 $PWMx_SMnFRACVALy$) 之前，该16位值右移5位，然后作为32位值通过32位指令写入两个值寄存器。这种方法可将所有PWM更新计算有效保持在16位范围内。

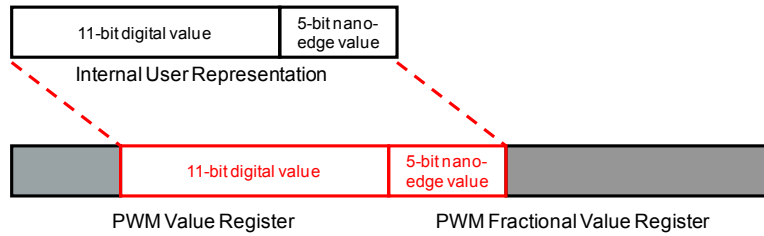


图 8. 高分辨率占空比的 16 位表示

2.4.1 产生高分辨率PWM占空比的软件示例

以下示例代码演示如何使用子模块 0、输出PWM0A、200 kHz的输出频率以及100 MHz DSC内核频率的MC56F84879产生高分辨率PWM信号。

```

/* 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_SMOVAL1 = 499;       // 100MHz/200kHz - 1
PWMA_SMOVAL2 = 0;        // Edge aligned PWM -> rising edge at beginning of period
PWMA_SMOVAL3 = 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)
}

```

注意: 此代码未包含 eFlexPWM 模块的全部初始化过程。以上部分仅与细分延迟逻辑初始化、PWM 周期定义和占空比更新示例有关。

3 结论

细分延迟模块对许多电源转换应用必不可少。本应用笔记详细介绍了飞思卡尔数字信号控制器所用的高分辨率占空比和频率PWM是如何产生的，其中包括细分延迟模块的工作原理、产生高分辨率占空比和频率的区别，以及如何在软件中使用细分延迟模块。最后提供了一个简单的软件示例，说明如何有效使用16位变量来产生高分辨率PWM信号，也可以使用16位变量。

4 参考

有关更多信息，请参阅freescale.com提供的以下文档。

- 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 修订历史记录

修订版本号	日期	重大变更
0	06/2013	初始版本

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

本文档中的信息仅供系统和软件实施方使用 Freescale 产品。本文并未明示或暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。Freescale 保留对本文档中所述任何产品进行更改的权利，恕不另行通知。

Freescale 对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用或使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于因果性或附带损害在内的所有责任。Freescale 数据手册和/或技术规格中可能提供了“典型”参数，这些参数在不同应用中可能并且确实不同，实际性能也可能会随时间变化。所有操作参数，包括“典型值”在内，必须由客户的技术专家根据每个客户应用进行验证。Freescale 未转让与其专利权及其他权利相关的许可。Freescale 销售产品时遵循以下网址中包含的标准销售条款和条件：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.

© 2013 飞思卡尔半导体有限公司

Document Number: AN4746

Rev. 0, 06/2013

June 3, 2013