UM10503

LPC43xx ARM Cortex-M4/M0 双核微处理器

修订版：1 — 2012 年 2 月 1 日　　初版用户手册

文档信息

信息	内容
关键字	LPC43xx、 LPC4350、 LPC4330、 LPC4320、 LPC4310、 LPC4357、 LPC4353、 LPC4337、 LPC4333、 LPC4327、 LPC4325、 LPC4323、 LPC4322、LPC4317、LPC4315、LPC4313、LPC4312、LPC4310、ARM Cortex-M4、 ARM Cortex-M0、 SPIFI、 SCT、 USB、以太网
摘要	LPC4300 初版用户手册

This translated version is for reference only, and the English version shall prevail in case of any discrepancy between the translated and English versions.
版权所有 2012 恩智浦有限公司 未经许可，禁止转载

修订记录

版本	日期	描述
1	20111212	初版 LPC43xx 用户手册。

联系信息

有关详细信息，请访问：http://www.nxp.com

欲咨询销售办事处地址，请发送电子邮件至：salesaddresses@nxp.com

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

1.1 简介

LPC43xx 是针对嵌入式应用的 ARM Cortex-M4 微控制器，配有 1 个 ARM Cortex-M0 协处理器、高达 264 kB SRAM、高级可配置外设（如状态可配置定时器 (SCT) 和串行通用 I/O (SGPIO) 接口）、2 个高速 USB 控制器、以太网、液晶显示器、1 个外部存储控制器和多个数字和模拟外设。 LPC43xx 系列的 CPU 工作频率高达 204 MHz。

ARM Cortex-M4 是下一代 32 位微控制器内核，具有低功耗、易调试、易集成等多种系统增强优势。ARM Cortex-M4 内核 CPU 采用 3 级流水线和哈佛架构，具有独立的本地指令和数据总线以及用于系统外设的第三总线，同时还包含一个支持不确定分支操作的内部预取单元。 ARM Cortex-M4 支持单周期数字信号处理和 SIMD 指令。内核集成硬件浮点处理器。

ARM Cortex-M0 协处理器是一款易用性能突出的高能效 32 位内核，其代码和工具均与 Cortex-M4 兼容。作为现有 8/16 位微处理器的替代器件，Cortex-M0 协处理器提供高达 204 MHz 的卓越性能，采用简单的指令集，代码尺寸也有所降低。

1.2 功能要点

- Cortex-M4 处理器内核
  - ARM Cortex-M4 处理器，工作频率高达 204 MHz。
  - ARM Cortex-M4 内建的存储器保护单元 (MPU) 支持 8 个区。
  - ARM Cortex-M4 内建可嵌套向量中断控制器 (NVIC)。
  - 硬件浮点单元。
  - 非屏蔽中断 (NMI) 输入。
  - JTAG 和串行线调试 (SWD)、串行线路、 8 个中断点及 4 个观察点。
  - 强化的跟踪模块 (ETM) 和强化的跟踪缓冲区 (ETB) 支持。
  - 系统定时器。
- Cortex-M0 处理器内核
  - ARM Cortex-M0 协处理器能分担 ARM Cortex-M4 主应用处理器的负荷。
  - 工作频率高达 204 MHz。
  - JTAG、串行线调试和内建 NVIC。
- 片内存储器
  - 最高 264 kB SRAM，供代码和数据使用。
  - 多个 SRAM 块，支持独立总线存取。 2 个 SRAM 块可以独立进入掉电模式。
  - 64 kB ROM，含引导代码和片内软件驱动程序。
  - 128 位通用一次性可编程 (OTP) 存储器。
- 可配置的数字外设
  - 串行 GPIO (SGPIO) 接口。
  - 位于 AHB 上的状态可配置定时器 (SCT) 子系统。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- 全局输入多路复用器阵列(GIMA)允许将多个输入和输出交叉连接至事件驱动型外设（如定时器、 SCT 和 ADC0/1）。

- 串行接口
  - 四通道 SPI 闪存接口 (SPIFI)，其 1 位、 2 位或 4 位数据速率最高可达 40 Mb/ 秒。
  - 10/100T 以太网 MAC，配有 RMII 和 MII 接口以及 DMA 支持，在低 CPU 负载下也可实现高吞吐量。支持 IEEE 1588 时间戳 / 高级时间戳 (IEEE 1588-2008 v2)。
  - 一个高速 USB 2.0 主机 / 设备 /OTG 接口，支持 DMA 并带有片内高速 PHY。
  - 1 个高速 USB 2.0 主机 / 设备接口，配有 DMA 支持、片内全速 PHY 和可连接外部高速 PHY 的 ULPI 接口。
  - ROM USB 协议栈集成 USB 接口电气测试软件。
  - 1 个 550 UART，配有 DMA 支持和全调制解调器接口。
  - 3 个 550 USART，配有 DMA 和同步模式支持，并配备 1 个符合 ISO7816 规范的智能卡接口。 1 个 USART，配有 IrDA 接口。
  - 2 个 C_CAN 2.0B 控制器，各配有一个通道。
  - 2 个 SSP 控制器，配有 FIFO 和多协议支持。 2 个 SSP 控制器均支持 DMA。
  - 1 个 SPI 控制器。
  - 一个 Fast-mode Plus I^2C 总线接口，具备监控模式和符合完整 I^2C 总线规范的开漏 I/O 引脚。处理高达 1 Mbit/s 的数据速率。
  - 一个标准的 I^2C 总线接口，具备监控模式和标准 I/O 引脚。
  - 2 个 I^2S 接口，均配有 DMA 支持和 1 个输入和 1 个输出。

- 数字外设
  - 外部存储控制器 (EMC)，支持外部 SRAM、 ROM、 NOR 闪存和 SDRAM 设备。
  - LCD 控制器具有 DMA 支持以及高达 1024H × 768V 的可编程显示分辨率。支持单色和彩色 STN 面板和 TFT 彩色面板；支持 1/2/4/8 bpp 彩色查找表 (CLUT) 和 16/24 位直接像素映射。
  - 安全数字输入输出 (SD/MMC) 卡接口。
  - 八通道通用DMA (GPDMA)控制器，可以存取AHB上的所有存储器和所有支持DMA的 AHB 从属存储器。
  - 多达 164 个通用输入 / 输出 (GPIO) 引脚，配有可配置上拉 / 下拉电阻和开漏模式。
  - GPIO 寄存器位于 AHB 上，以支持快速存取。 GPIO 端口支持 DMA。
  - 最多可以从所有 GPIO 引脚中选择 8 个 GPIO 引脚，作为边缘和电平触发型中断源。
  - 2 个 GPIO 分组中断模块，支持基于一组 GPIO 引脚的输入状态的可编程范式进行中断。
  - 4 个通用定时器 / 计数器，支持捕获和匹配。
  - 1 个电机控制脉冲宽度调制器 (PWM)，用于三相电机控制。
  - 1 个正交编码器接口 (QEI)。
  - 重复性中断定时器 （RI 定时器）。
  - 窗口看门狗定时器 (WWDT)。
  - 独立的电源域配有超低功耗实时时钟 (RTC)，配有 256 字节的电池供电型备份寄存器。
  - 报警定时器；可通过电池供电。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- 模拟外设
  - 1 个 10 位 DAC，支持 DMA，数据转换采样速率为 400 k/s。
  - 2 个 10 位 ADCs，具有 DMA 支持，数据转换采样速率为 400 k/s。
- 安全
  - AES 解密，可通过片内 API 进行编程。
  - 2 个 128 位安全 OTP 存储器，供存储 AES 密钥和客户使用。
  - 随机数生成器 (RNG) 可通过 AES API 访问。
  - 每只芯片有唯一的 ID。
- 时钟产生单元
  - 晶振，工作频率范围为 1 MHz 至 25 MHz。
  - 12 MHz 内部 RC (IRC) 振荡器，在全部温度和电压范围内精度可达 1%。
  - 超低功耗实时时钟 (RTC) 晶振。
  - 3 个 PLL，使 CPU 可达最高速率，而无需使用高频晶振。第二 PLL 专门用于高速 USB，第三 PLL 可用作音频 PLL。
  - 时钟输出。
- 电源
  - 采用 3.3 V（2.2 V 至 3.6 V）单电源供电，配有片内 DC-DC 转换器，为内核和 RTC 电源域供电。
  - RTC 功耗域可由 3 V 电池单独供电。
  - 4 种节能模式：睡眠模式、深度睡眠模式、掉电模式和深度掉电模式。
  - 可通过多种外设的唤醒中断从睡眠模式唤醒处理器。
  - 支持通过外部中断和 RTC 电源域中电池供电模块产生的中断从深度睡眠、掉电和深度掉电三种模式唤醒。
  - 掉电检测，为中断和强制复位设有 4 个独立的阈值。
  - 上电复位 (POR)。
  - 提供 256 引脚、180 引脚和 100 引脚 LBGA 封装，以及 208 引脚、144 引脚和 100 引脚 LQFP 封装。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

1.3 订购信息 （无闪存的器件 LPC4350/30/20/10）

表 1. 订购信息

产品型号	封装		
	名称	描述	版本
LPC4350FET256	LBGA256	塑封薄型球栅阵列封装； 256 引脚；主体尺寸 17 × 17 × 1 mm	SOT740-2
LPC4350FET180	TFBGA180	薄型小间距球栅阵列封装； 180 引脚	SOT570-3
LPC4350FBD208	LQFP208	塑封薄型四方扁平封装； 208 引脚；主体尺寸 × 28 × 1.4 mm	SOT459-1
LPC4330FET256	LBGA256	塑封薄型球栅阵列封装； 256 引脚；主体尺寸 17 × 17 × 1 mm	SOT740-2
LPC4330FET180	TFBGA180	薄型小间距球栅阵列封装； 180 引脚	SOT570-3
LPC4330FET100	TFBGA100	塑封精细间距球栅阵列封装； 100 引脚；主体尺寸 9 × 9 × 0.7 mm	SOT926-1
LPC4330FBD144	LQFP144	塑封薄型四方扁平封装； 144 引脚；主体尺寸 20 × 20 × 1.4 mm	SOT486-1
LPC4320FET100	TFBGA100	塑封精细间距球栅阵列封装； 100 引脚；主体尺寸 9 × 9 × 0.7 mm	SOT926-1
LPC4320FBD144	LQFP144	塑封薄型四方扁平封装； 144 引脚；主体尺寸 20 × 20 × 1.4 mm	SOT486-1
LPC4320FBD100	LQFP100	塑封薄型四方扁平封装； 100 引脚；主体尺寸 14 × 14 × 1.4 mm	SOT407-1
LPC4310FET100	TFBGA100	塑封精细间距球栅阵列封装； 100 引脚；主体尺寸 9 × 9 × 0.7 mm	SOT926-1
LPC4310FBD144	LQFP144	塑封薄型四方扁平封装； 144 引脚；主体尺寸 20 × 20 × 1.4 mm	SOT486-1

表 2. 订购选项

产品型号	总SRAM	LCD	以太网	USB0（主机、设备、OTG）	USB1（主机、设备）/ULPI 接口	ADC 通道	PWM	QEI	GPIO	封装
LPC4350FET256	264 kB	是	是	是	有 / 有	8	是	是	164	LBGA256
LPC4350FET180	264 kB	是	是	是	有 / 有	8	是	是	118	TFBGA180
LPC4350FBD208	264 kB	是	是	是	有 / 有	8	是	是	142	LQFP208
LPC4330FET256	264 kB	否	是	是	有 / 有	8	是	是	164	LBGA256
LPC4330FET180	264 kB	否	是	是	有 / 有	8	是	是	118	TFBGA180
LPC4330FET100	264 kB	否	是	是	有 / 无	4	否	否	49	TFBGA100
LPC4330FBD144	264 kB	否	是	是	有 / 无	8	是	否	83	LQFP144
LPC4320FET100	200 kB	否	否	是	否	4	否	否	49	TFBGA100
LPC4320FBD144	200 kB	否	否	是	否	8	是	否	83	LQFP144
LPC4320FBD100	200 kB	否	否	是	否	5	否	否	49	LQFP100
LPC4310FET100	168 kB	否	否	否	否	4	否	否	49	TFBGA100
LPC4310FBD144	168 kB	否	否	否	否	8	是	否	83	LQFP144

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

1.4 功能框图

图 1. LPC43xx 框图

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

2.1 本章导读

ARM Cortex-M0 协处理器可用于所有 LPC43xx 器件。

2.2 基本配置

ARM Cortex-M0 协处理器的配置如下所示：

- 参见表 3 以了解时钟与电源控制。
- ARM Cortex-M0 通过 M0APP_RST （复位 # 56）或通用复位进行复位。
- 上电后，必须通过清除相应的 RESET_CTRL1 位来释放 ARM Cortex-M0 复位（参见表 106）。
- ARM Cortex-M0 中断连接到 ARM Cortex-M4 NVIC 中的中断插槽 # 1。有关连接到 ARM Cortex-M0 的外设中断，请参见表 22。
- 要清除 ARM-Cortex-M0 中断，使用 M0TXEVENT 寄存器（表 46）。参见第 2.4.2 节。

表 3. ARM Cortex-M0 时钟和电源控制

	基本时钟	分支时钟	工作频率
ARM CORTEX-M0 时钟	BASE_M4_CLK	CLK_M4_M0	高达 204 MHz

2.3 简介

在 LPC43xx 等微处理器中， ARM Cortex-M4 用作应用处理器。另有一个第二处理器，即 ARM Cortex-M0，它可用作可减轻 ARM Cortex-M4 的负载并执行串行 I/O 任务的协处理器。两个处理器之间需要一种通信协议。本章说明 LPC43xx 的内核间通信 (IPC) 协议。

2.4 简介

在 LPC43xx 上，ARM Cortex-M4 主机 CPU 用作顶级系统控制器。LPC43xx 还包含另一个 CPU，即 ARM Cortex-M0。ARM Cortex-M0 CPU 受主机 CPU 控制。这两个 CPU 之间的通信利用共享的存储器空间和中断。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 2. 双核框图

2.4.1 硬件

IPC 使用的并非是专用硬件，而是现有的硬件组件。共用存储器中的缓冲区可使用任何可用的 SRAM。缓冲区指针保持在软件中。中断在处理器 NVIC 中捕获，在 CREG 模块中清除（参见表 44 和表 46）。

2.4.2 中断处理

ARM Cortex-M4 和 ARM Cortex-M0 通过 CREG 寄存器 M4TXEVENT 和 M0TXEVENT 相互启动中断（参见表 44 和表 46）。M4 至 M0 中断和 M0 至 M4 中断使用 SendEvent 指令 (SEV) 来产生信号 TXEV。该信号由 CREG 捕获。应当通过接收内核的中断处理器来清除。

2.5 IPC 协议描述

IPC 支持寄存器级接口等低级别接口，但也支持 API 等级别较高的接口。

ARM Cortex-M4 主机 CPU 始终为主机。它发起命令到模拟硬件寄存器级接口的 ARM Cortex-M0。命令可以同步 （等待回复消息）或异步 （不等待回复消息）方式发出，具体取决于主机应用。

ARM Cortex-M0 通过发出消息来响应 ARM Cortex-M4 给出的命令。

由于 ARM Cortex-M4 和 ARM Cortex-M0 无法在同一时间写入相同的位置，因此这一 IPC 中不需要同步对象 （如信号量）。

基本的 IPC 特性为：

- ARM Cortex-M4 初始化 ARM Cortex-M0 系统。
- ARM Cortex-M4 通过命令队列与 ARM Cortex-M0 系统通信。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

- 消息队列位于ARM Cortex-M4地址空间中，因为ARM Cortex-M4可以被禁止访问ARM Cortex-M0 硬件子系统。

2.5.1 IPC 队列

ARM Cortex-M4 具有一个输出命令队列和一个输入消息队列。一个队列由 4 个寄存器来定义：

1. 队列起始地址
2. 队列结束地址
3. 写指针
4. 读指针

ARM Cortex-M4 初始化这 4 个寄存器。这些寄存器驻留在与队列相同的共享 SRAM 中，以便确保数据和寄存器更改处于同步状态。其位置是静态的，并且对 ARM Cortex-M0 来说是已知的。

队列之间使用循环缓冲来传递消息。一个队列从起始地址到结束地址填充了命令或消息。当缓冲指针超出结束地址时，它会返回来指向起始地址。当读指针等于写指针时，队列可以为空或全满。为了避免这种双重性，队列不可完全填满。因此最小的队列为 3 个字 （最长的命令 / 消息 +1 个字）。写指针和读指针相等将表示队列为空。

命令队列由 ARM Cortex-M4 填充，并由 ARM Cortex-M0 清空；每次 ARM Cortex-M4 添加一个新命令到队列中时，会将写指针前进一位。每次 ARM Cortex-M0 从队列中删除一个命令时，会将读指针前进一位。

消息队列由 ARM Cortex-M0 填充，并由 ARM Cortex-M4 清空；每当 ARM Cortex-M0 添加一个新消息到队列中时，会将写指针前进一位。每次 ARM Cortex-M4 从队列中删除一个消息时，会将读指针前进一位。

当新命令或消息添加到队列中并且写指针被更新后，会向另一处理器产生一个中断。命令通过返回消息 （接受或失败）来确认。

ARM Cortex-M4 和 ARM Cortex-M0 仅具有一个 IPC 写任务和一个 IPC 读任务。如果存在多个实例，则有一个本地仲裁器来确保所有写入和读取操作都是原子性的；在数据被写入（读取）后，写 （读）指针将被更新，而后再启动另一个写入 （读取）操作。

写入队列的进程负责确保队列不被彻底填满；在加载新的项目前，该进程应确认写指针不等于 （或者不超过）读指针并且保留至少一个可用空间。另一方面，接收的一端应及时处理和删除队列中的项目。

不执行任何显式错误处理。假定 ARM Cortex-M0 会始终响应 ARM Cortex-M4 命令。

2.5.2 协议

ARM Cortex-M0 用作减轻 ARM Cortex-M4 负载并执行串行 IO 任务的协处理器。 ARM Cortex-M4 应当能初始化 ARM Cortex-M0 上执行的任务。当这些任务完成或失败时，ARM Cortex-M0 应当能够将信号发送给 ARM Cortex-M4。这是通过 ARM Cortex-M4 向 ARM Cortex-M0 发出命令而 ARM Cortex-M0 用消息进行响应来实现的。这一命令和消息接口类似于一种带有命令和状态寄存器的硬件寄存器级接口。

ARM Cortex-M4 向 ARM Cortex-M0 发出 32 位命令。每个命令以一个 16 位 ID 开头，用来定义所指代的任务。 LSBit 表示命令类型。写命令后接一个 32 位操作数。有新的可用命令时， ARM Cortex-M4 通过产生中断将它通知给 ARM Cortex-M0。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

ARM Cortex-M0 向 ARM Cortex-M4 返回 32 位消息。每条消息以一个 16 位 ID 开头，用来表示消息所指代的任务。LSByte 表示消息类型。读响应消息后接一个 32 位读取操作数。有新的可用消息时 ,ARM Cortex-M0 通过产生中断将它通知给 ARM Cortex-M4。

小规模的数据传输可通过单个 32 位数据读取 CMD_RD_ID 和写入 CMD_WR_ID 命令来执行。这些命令使用 3 字节寻址方案来支持 212 = 4096 32 位字的自变量空间。大规模的数据传输可通过使用指针来更为高效地进行处理。

另外，使用 API 调用的更高级别的接口通常使用间接、基于指针的读取和写入。

当同时运行多个任务时，使用 ID 来区分属于特定任务的命令和消息。应当使用一种全局命令解析器将命令传送到 ARM Cortex-M0 上运行的任务。

ARM Cortex-M4 一端亦是如此，通过全局消息解析器将消息传回到 ARM Cortex-M4 一端运行的任务调度程序。

表 4. 命令列表

命令	位屏蔽	描述
CMD_RD_ID	0xTTTT.PPP0	从 ID = 0xTTTT 的任务读取自变量 ID=0xPPP 的 32 位 WORD
CMD_WR_ID	0xTTTT.PPP1, WORD	将自变量 ID=0xPPP 的 32 位 WORD 写入到 ID = 0xTTTT 的任务

表 5. 消息列表

消息	位屏蔽	描述
MSG_SRV_ID	0xTTTT.SS00	ARM Cortex-M0 请求服务 ID = 0xTTTT 的任务。服务类型以字节 SS 进行编码。SS 的含义专用于各个任务。 SS=0x00 代表任务已完成。
MSG_RD_ID	0xTTTT.PPP1, VALUE	ARM Cortex-M0 使用 VALUE 来响应从 ID = 0xTTTT 的任务读取自变量 ID=0xPPP* 的 WORD 的命令。
MSG_RD_STS_ID	0xTTTT.PPPR	ARM Cortex-M0 响应从 ID = 0xTTTT 的任务读取自变量 ID=0xPPP* 的 WORD 的命令失败。失败的原因以 R 进行编码； R = 2...4 2 = 无效自变量 3 = 保留 4 = 保留
MSG_WR_STS_ID	0xTTTT.PPPW	ARM Cortex-M0 响应向 ID = 0xTTTT 的任务写入自变量 ID=0xPPP* 的 WORD 的命令。响应以 W 进行编码； W = 5...7 5 = 写成功 6 = 写失败 7 = 保留

表 6. 命令响应

命令	可能的响应	描述
CMD_RD_ID	MSG_RD_ID, VALUE	读取已确认
	MSG_RD_STS_ID	读取已失败
CMD_WR, WORD	MSG_WR_STS_ID	写入已确认成功或失败

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

2.5.3 示例

假设某个 ID 为 0x1234 的任务应当由 ARM Cortex-M0 执行。例如，从由 ARM Cortex-M0 控制的寄存器级接口读取数据。粗体的文字代表寄存器。

寄存器可位于 ARM Cortex-M0 SRAM 中 （访问时间更具确定性）或位于共享 SRAM 中。如果使用 ARM Cortex-M0 SRAM，则需要在初始化阶段复制寄存器数据。此复制操作会占用一些时间。 ARM Cortex-M4 可以轮询状态寄存器来确定该传输完成的时间。

ARM Cortex-M4 通过载入起始和结束地址以及写和读指针来初始化命令和消息队列。

然后，ARM Cortex-M4 在通用 SRAM 存储器的保留区域中加载寄存器值。（另一个方式是 ARM Cortex-M4 逐个寄存器执行写入，但这与一次性加载所有数据相比要求更多的通信开销。）所有数据都设好后，即可启动 ARM Cortex-M0 任务。

表 7. IPC 示例

	命令	消息	字节值	描述
1	CMD_WR		0x1234<rst>1, 指针	初始化任务的命令，该指针通知 ARM Cortex-M0 寄存器值的位置。
ARM Cortex-M0 处理寄存器				
2		MSG_WR_STS	0x1234<rst>2	ARM Cortex-M0 通知写入数据操作已被处理
3		MSR_SRV	0x1234<serve>00	ARM Cortex-M0 请求服务，例如由于数据已被捕获并且可用
4	CMD_RD		0x1234< 状态 >1	ARM Cortex-M4 读取状态
5		MSG_RD,VALUE	0x1234< 状态 >1, VALUE	ARM Cortex-M0 用状态进行响应
根据该状态， ARM Cortex-M4 可能会决定读取更多数据				
6	CMD_RD		0x1234< 结果 >,1	ARM Cortex-M4 读取结果
7		MSG_RD,VALUE	0x1234< 结果 >1, 指针	ARM Cortex-M0 用指针响应结果。
:	:	:	:	:
n	CMD_WR		0x1234< 停止 >1, 值	用于停止任务的命令
n+1		MSG_WR_STS	0x1234< 停止 >2	ARM Cortex-M0 通知停止操作已被处理

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

3.1 本章导读

可用的外围设备及其存储器随器件不同而有所差异。

- 以太网：仅用于 LPC435x/3x。
- USB0：仅用于 LPC435x/3x/2x。
- USB1：仅用于 LPC435x/3x。
- SRAM：参见表 8。

与不可用外围设备和存储器模块对应的寄存器和存储器区域将保留。

3.2 基本配置

在 CREG 模块 （参见表 42）中，选择访问地址 0x2000 C000 中的 16 kB RAM 模块的接口。该 RAM 存储器模块可通过嵌入式跟踪缓冲区 (ETB)（这是默认值）访问，或用作 AHB 总线上的普通 SRAM。

注：使用 ETB 时，其他任何进程都不得使用 0x2000 C000 的 16 kB 存储器空间。

3.3 存储器配置

3.3.1 片内静态 RAM

LPC43xx 支持高达 264 kB 的 SRAM，并提供分块的总线主机访问以实现更高的吞吐量，以及单独的电源控制以实现低功耗操作。

表 8. LPC4350/30/20/10 SRAM 配置

器件	本地 SRAM 0x1000 0000	本地 SRAM 0x1008 0000	AHB SRAM 0x2000 0000	AHB SRAM 0x2000 8000	AHB SRAM/ ETB SRAM[1] 0x2000 C000	
LPC4350	128 kB	72 kB	32 kB	16 kB	16 kB	图 3
LPC4330	128 kB	72 kB	32 kB	16 kB	16 kB	图 3
LPC4320	96 kB	40 kB	32 kB	16 kB	16 kB	图 3
LPC4310	96 kB	40 kB	16 kB	-	16 kB	图 3

[1] 要配置用于 AHB 或 ETB 的 SRAM 存储器，请参见表 42。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

3.3.2 位带机制

位带机制提供高效的位访问。位带区域（0x2000 0000 到 0x2010 0000 和 0x4000 0000 到 0x40100000）的位可在所谓的别名区域（0x22000 0000 和 0x42000 0000）中访问。读取操作从位带区域返回相应位。写入操作在位带区域的相应位上进行原子性“读 - 修改 - 写”操作。更多详情，请参见 *ARM Cortex-M4 技术参考手册*。

注：位带机制不可用于 MAC_RWAKE_FRFLT 寄存器 （参见第 26.6.10 节）。

注：尽管 EEPROM 也映射到了支持位带的区域，但是往此位带别名存储区的写操作不会导致位写入

3.3.3 掉电模式下的存储器保持

深度睡眠模式下，所有 SRAM 内容都会保留。在唤醒模式下，系统可立即重新启动。

在掉电模式下，只有开始于 0x1008 0000 上 SRAM 块的最高 8 kB 保留 - 即位于 0x1009 0000 的 SRAM 的 8 kB。其他所有 SRAM 内容都会丢失。常见做法是存储需要在该 8 kB 存储器空间中保留的协议栈和其他变量以及代码以重新启动系统的其余部分。

深度掉电模式下，不会保留任何 SRAM 内容。深度掉电模式下需要保留的变量可存储在位于 0x4004 1000 的 RTC 域的 256 字节寄存器文件中。

3.3.4 存储器保护单元 (MPU)

MPU 是 ARM Cortex-M4 的重要组成部分，用于保护存储器，受所有 LPC43xx 器件的支持。处理器支持标准的 ARMv7 受保护存储器系统架构模型。 MPU 全面支持：

- 保护区域
- 重叠保护区域，具有递增的区域优先级 （7 = 最高优先级， 0 = 最低优先级）
- 访问权限
- 将存储器属性导出到系统

MPU 不匹配和权限违反会调用可编程优先级 MemManage 故障处理程序。参见 *ARMv7-M 架构参考手册*以了解更多信息。

区域访问控制寄存器的访问权限位 TEX、C、B、AP 以及 XN 控制对相应存储器区域的访问。如果在没有所需权限的情况下访问存储器的某个区域，会产生权限错误。有关更多信息，请参见 *ARMv7-M 架构参考手册*。

MPU 用于实施特权规则、分离进程并实施访问规则。有关 MPU 使用方法的详细信息以及寄存器描述，请参阅 *ARM Cortex-M4 技术参考手册*。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

3.4 存储器映射

图 3. 系统存储器映射 （有关所有外围设备的详细地址，请参见图 4）

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

图 4.　外围设备存储器映射 （有关存储器模块的详细地址，请参见图 3）

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

3.5 AHB 多层矩阵配置

所有总线主机可通过AHB多层矩阵访问任何嵌入式存储器以及连接至SPIFI接口的外部SPI闪存。两个或两个以上的总线主机尝试访问相同的从机时，采用是循环仲裁方案，每个主机按循环顺序轮流访问从机。访问长度由主机的连发访问长度决定。对于 CPU，连发大小为 1；对于 GP-DMA，连发大小最大可达到 8。为优化 CPU 性能，低延迟代码应存储在其他总线主机 （尤其是所用连发大小较长的主机）不访问的存储器中

为优化 CPU 性能， ARM Cortex-M4 有三条总线用于指令（代码） (I) 访问、数据 (D) 访问以及系统 (S) 访问。I 总线和 D 总线访问存储器空间位于 0x2000 0000 下方，S 总线访问开始于 0x2000 0000 的存储器空间。如果指令和数据保存在不同的存储器中，则可在一个周期内进行代码和数据的并行访问。如果代码和数据保存在相同的存储器中，则加载或存储数据的指令需要经过两个周期。

使用嵌入式跟踪缓冲区时 （参见 ETBCFG 寄存器，表 42），其他任何进程都不得使用 0x2000 C000 的 16 kB 存储器空间。

图 5. LPC43xx AHB 多层矩阵连接

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

4.1 本章导读

本章适用于所有 LPC43xx 器件。

4.2 功能要点

- OTP 存储器存储下列信息：
  - 器件 ID
  - 唯一 ID 包含每个微控制器的唯一 ID。
  - 用户可编程的为启动源、 USB 供应商和产品 ID 以及 AES 密钥。
  - 未使用的域可用于存储其他数据。
- API 支持对 OTP 进行编程

4.3 简介

OTP 包含使用两个 OTP 组别的预编程设备特定信息器件 ID 和唯一 ID。必须由用户来对所有其他 384 个 OTP 位进行编程。

原始 OTP 状态为全零。这表示，一个零值将被 1 值覆盖，但 1 值无法被更改。

OTP 编程电压需要高于读取时的电压。读取电压由内部生成。编程电压通过引脚 VPP 供应。 OTP 控制器将自动选择正确的电压。如果未连接 VPP 引脚，则无法对 OTP 编程。

OTP 存储器中的 AES 密钥无法由软件读取。

4.4 寄存器描述

表 9. OTP 存储器描述 （OTP 基址 0x4004 5000）

OTP 组别	字	访问类型	地址偏移	尺寸	描述	参考
0	0	预编程；无法由用户更改。	0x000	32 位	器件 ID	-
0	1	预编程；无法由用户更改。	0x004	32 位	保留	-
0	2	预编程；无法由用户更改。	0x008	32 位	唯一 ID	-
0	3	预编程；无法由用户更改。	0x00C	32 位	保留	-
1	0	用户可编程；初始状态 = 0	0x010	32 位	通用 OTP 存储器 0，字 0 或 AES 密钥 0，字 0	-
1	1	用户可编程；初始状态 = 0	0x014	32 位	通用 OTP 存储器 0，字 1 或 AES 密钥 0，字 1	-
1	2	用户可编程；初始状态 = 0	0x018	32 位	通用 OTP 存储器 0，字 2 或 AES 密钥 0，字 2	-
1	3	用户可编程；初始状态 = 0	0x01C	32 位	通用 OTP 存储器 0，字 3 或 AES 密钥 0，字 3	-
2	0	用户可编程；初始状态 = 0	0x020	32 位	通用 OTP 存储器 1，字 0 或 AES 密钥 1，字 0	-
2	1	用户可编程；初始状态 = 0	0x024	32 位	通用 OTP 存储器 1，字 1 或 AES 密钥 1，字 1	-
2	2	用户可编程；初始状态 = 0	0x028	32 位	通用 OTP 存储器 1，字 2 或 AES 密钥 1，字 2	-
2	3	用户可编程；初始状态 = 0	0x02C	32 位	通用 OTP 存储器 1，字 3 或 AES 密钥 1，字 3	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 9. OTP 存储器描述 （OTP 基址 0x4004 5000）*（续）*

OTP 组别	字	访问类型	地址偏移	尺寸	描述	参考
3	0	用户可编程；初始状态 = 0	0x030	32 位	客户控制数据	表 10
3	1	用户可编程；初始状态 = 0	0x034	32 位	通用 OTP 存储器 2，字 0 或 USB ID	表 11
3	2	用户可编程；初始状态 = 0	0x038	32 位	通用 OTP 存储器 2，字 1	表 12
3	3	用户可编程；初始状态 = 0	0x03C	32 位	通用 OTP 存储器 2，字 2	-

表 10. OTP 存储组别 3，字 0 – 客户控制数据 （地址偏移 0x030）

位	符号	值	描述
22:0	-	-	保留
23	USB_ID_ENABLE		设置此位将允许使能 OTP 定义的 USB 供应商以及产品 ID。使能时，USB 驱动器使用 USB_VENDOR_ID 和 USB_PRODUCT_ID 值。如果禁用，则使用恩智浦供应商 ID (0x1FC9) 和产品 ID (0x000C)。
		0	已禁用
		1	使能
24	-	-	保留
28:25	BOOT_SRC		OTP 中的启动源选择。有关详情，请参阅表 15。
		0000	外部引脚
		0001	UART0
		0010	保留
		0011	EMC 8 位
		0100	EMC 16 位
		0101	EMC 32 位
		0110	USB0
		0111	USB1
		1000	SPI （通过 SSP）
		1001	UART3
29	-		保留。不要向此位写入。
30	-		保留。不要向此位写入。
31	JTAG_DISABLE		如果设置了该位，则 JTAG 不能由软件使能并将保持禁用状态。

表 11. OTP 存储组别 3，字 1 - 通用 OTP 存储组别 2，字 0 或 USB ID （地址偏移 0x034）

位	符号	描述
15:0	USB_VENDOR_ID	如果未设置 USB_ID_ENABLE 位，则它将用作通用 OTP 存储器 2，字 0， GP2_0。
31:16	USB_PRODUCT_ID	

表 12. OTP 存储组别 3，字 2 - 通用 OTP 存储器 2，字 1 或以太网 MAC （地址 0x038）

位	符号	描述
31:0	ETH_MAC	如果未设置 ETH_MAC_ENABLE，则它将用作通用 OTP 存储器 2，字 1， GP2_1。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

4.5 OTP API

OTP 存储器通过 LPC43xx ROM 中的一组简单的 API 调用进行控制。

执行 ROM 驱动器表中指针所指向的功能来进行 ROM 的 API 调用。

图 6. OTP 驱动器指针结构

表 13. ROM 驱动器指针 （主要 API 输入点 0x1040 0100）

API	地址	描述
OTP	0x04	指向 OTP 驱动程序基址的指针

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

4.5.1 OTP 功能分配

表 14. OTP 功能分配

函数	偏移	描述
otp_Init	0x00	初始化 OTP 控制器。 参数 - void 返回 – unsigned：参见常见的错误代码 <tbd>。
otp_ProgBootSrc	0x04	对启动源进行编程。 参数 - unsigned <src> 返回 – unsigned：参见常见的错误代码 <tbd>。
otp_ProgJTAGDis	0x08	设置 JTAG 禁用。该指令仅在设备支持 AES 时禁用 JTAG。 参数 - void 返回 – unsigned：参见常见的错误代码 <tbd>。
otp_ProgUSBID	0x0C	对 USB_ID 进行编程。 参数 - unsigned <prod_id>、 unsigned <vend_id> 返回 – unsigned：参见常见的错误代码 <tbd>。
-	0x10	保留
-	0x14	保留
-	0x18	保留
otp_ProgGP0	0x1C	对通用 OTP 存储器 GP0 进行编程。仅在设备不支持 AES 的情况下使用。 参数 - unsigned <data>、 unsigned <mask> 返回 – unsigned：参见常见的错误代码 <tbd>。
otp_ProgGP1	0x20	对通用 OTP 存储器 GP1 进行编程。仅在设备不支持 AES 的情况下使用。 参数 - unsigned <data>、 unsigned <mask> 返回 – unsigned：参见常见的错误代码 <tbd>。
otp_ProgGP2	0x24	对通用 OTP 存储器 GP2 进行编程。用于客户特定数据。 参数 - unsigned <data>、 unsigned <mask> 返回 – unsigned：参见常见的错误代码 <tbd>。
otp_ProgKey1	0x28	对 AES 密钥 1 进行编程。 参数 - unsigned char *key （预计 16 字节） 返回 – unsigned：参见常见的错误代码 <tbd>。
otp_ProgKey2	0x2C	对 AES 密钥 2 进行编程 参数 - unsigned char *key （预计 16 字节） 返回 – unsigned：参见常见的错误代码 <tbd>。
otp_GenRand	0x30	使用硬件随机数生成器 (RNG) 生成新的随机数。 参数 - void 返回 – unsigned：参见常见的错误代码 <tbd>。

4.5.2 OTP API 调用

< 待定 >

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

5.1 本章导读

本章适用于所有 LPC4350/30/20/10 器件。

5.2 功能要点

Boot ROM 存储器包括以下功能要点：

- ROM 存储器的大小为 64 KB。
- 支持从UART接口和外部静态存储器（如NOR闪存、SPI闪存和四通道SPI闪存）启动。
- 包含用于 OTP 编程的 API。
- 包括 USB 驱动器。
- ISP 模式用于将数据加载到片内 SRAM 并执行来自片内 SRAM 的代码。

支持 AES 的器件还支持：

- 对引导映像进行 CMAC 验证。
- 从加密映像进行安全引导。
- 支持允许从纯文本映像启动的开发模式。通过对 AES 密钥进行编程可终止开发模式。
- AES 编程的 API。

5.3 功能说明

内部 ROM 存储器用于存储启动代码。复位之后， ARM 处理器将从此存储器开始执行其代码。

ARM 内核配置为在复位后开始执行代码，且程序计数器设为值 0x0000 0000。LPC43xx 包含一个影子指针，它允许存储器区域映射到地址 0x0000 0000。影子指针的默认值为 0x1040 0000，确保在复位后执行引导 ROM 中包含的代码。

根据 OTP 位 BOOT_SRC 的值，可使用若干启动模式（参见第 4.4 节）。如果没有对 OTP 存储器进行编程，或者 BOOT_SRC 位全部为零，则引导模式将由引导引脚 P2_9、P2_8、P1_2 和 P1_1 的状态决定。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 15. OTP BOOT_SRC 位已编程时的引导模式

启动模式	BOOT_SRC 位 3	BOOT_SRC 位 2	BOOT_SRC 位 1	BOOT_SRC 位 0	描述
启动引脚	0	0	0	0	引导源由 P1_1、P1_2、P2_9 和 P2_8 引脚的复位状态进行定义。参见表 16。
USART0	0	0	0	1	从使用 P2_0 和 P2_1 连接到 USART0 的设备启动。
SPIFI	0	0	1	0	从使用 P3_3 至 P3_8 连接到 SPIFI 接口的四通道 SPI 闪存启动。
EMC 8 位	0	0	1	1	从使用 CS0 和 8 位数据总线的外部静态存储器（例如 NOR 闪存）启动。
EMC 16 位	0	1	0	0	从使用 CS0 和 16 位数据总线的外部静态存储器（例如 NOR 闪存）启动。
EMC 32 位	0	1	0	1	从使用 CS0 和 32 位数据总线的外部静态存储器（例如 NOR 闪存）启动。
USB0	0	1	1	0	从 USB0 启动。
USB1	0	1	1	1	从 USB1 启动。
SPI (SSP)	1	0	0	0	从连接到 P3_3（功能 SSP0_SCK）、P3_6（功能 SSP0_MISO）、P3_7（功能 SSP0_MOSI）和 P3_8（功能 SSP0_SSEL）上的 SSP0 接口的 SPI 闪存启动[1]。
USART3	1	0	0	1	从使用引脚 P2_3 和 P2_4 连接到 USART3 的设备启动。

[1] 启动引导程序对复位时相应的引脚功能编程，以使用 SSP0 进行启动。

表 16. OTP BOOT_SRC 位为零时的启动模式

启动模式	P2_9	P2_8	P1_2	P1_1	描述
USART0	低	低	低	低	从使用 P2_0 和 P2_1 连接到 USART0 的设备启动。
SPIFI	低	低	低	高	从连接到 P3_3 到 P3_8[1] 上的 SPIFI 接口的四通道 SPI 闪存启动。
EMC 8 位	低	低	高	低	从使用 CS0 和 8 位数据总线的外部静态存储器 （例如 NOR 闪存）启动。
EMC 16 位	低	低	高	高	从使用 CS0 和 16 位数据总线的外部静态存储器（例如 NOR 闪存）启动。
EMC 32 位	低	高	低	低	从使用 CS0 和 32 位数据总线的外部静态存储器（例如 NOR 闪存）启动。
USB0	低	高	低	高	从 USB0 启动。
USB1	低	高	高	低	从 USB1 启动。
SPI (SSP)	低	高	高	高	从连接到 P3_3（功能 SSP0_SCK）、P3_6（功能 SSP0_MISO）、P3_7 （功能 SSP0_MOSI）和 P3_8 （功能 SSP0_SSEL）上的 SSP0 接口的 SPI 闪存启动[1]。
USART3	高	低	低	低	从使用引脚 P2_3 和 P2_4 连接到 USART3 的设备启动。

[1] 启动引导对复位时相应的引脚功能编程，以从 SPIFI 或 SSP0 进行启动。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

5.3.1 AES 兼容设备

通常，支持 AES 的产品总是从安全 （加密的）映像或使用 CMAC 认证启动。不过，特殊的开发模式允许从纯文本映像启动。这种开发模式在尚未对 AES 密钥进行编程的情况下有效。此时， AES 密钥为全零。一旦对密钥进行编程后 （为非零值），该开发模式终止。

5.3.2 启动流程 (仅适用于无片上闪存的器件)

全面的启动过程在图 7 中描述。复位完成后，启动即开始。选择 IRC 作为 CPU 时钟，通过执行 Boot ROM 启动 Cortex-M4。默认情况下，复位时会禁止 JTAG 对芯片的访问。如果器件不支持 AES 或支持 AES 但尚未对 AES 密钥编程，则会使能 JTAG 访问。

如图 7 所示，Boot ROM 根据 OTP BOOT_SRC 值或引脚 P1_1、P1_2、P2_8 和 P2_9 的复位状态决定启动模式。BOOT ROM 会将映像复制到位于 0x1000 0000 的内部 SRAM 中，并在映像验证后跳到该位置（将 ARM 的影子指针设到 0x1000 0000）。因此，LPC43xx 的映像应当用入口点 0x0000 0000 进行编译。在编入了 AES 密钥的 AES 兼容 LPC43xx 上，映像和标头使用 CMAC 算法进行认证。如果认证失败，该设备将会复位。

在处于开发模式的 AES 兼容 LPC43xx 和不兼容 AES 的 LPC43xx 上，不对映像和标头进行认证。如果映像前面没有标头，则不会将该映像复制到 SRAM，而是假设其能够按原样执行。在那种情况下，影子指针将设到外部启动存储器的第一个地址位置。 LPC43xx 的无标头映像也应当用入口点 0x0000 0000 进行编译，与具有标头的映像相同。

注：启动过程失败时，引脚 P1_1 以 1 Hz 频率切换 60 秒。 60 秒后，设备复位。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 7.　启动流程 (仅适用于无片上闪存的器件)

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

5.3.3 启动映像格式

支持 AES 且 AES 密钥已编程的产品总是从安全映像或使用 CMAC 认证启动。安全映像应始终包括标头。

不支持 AES 的产品可以从含有标头的映像启动，或直接从启动源执行（当启动源为映射的存储器、SPIFI 或 EMC 时）。如果未找到有效的标头，则 CPU 将尝试执行存储器映射的启动源的第一个位置的代码。用户应注意确保此位置包含可执行的代码，否则将会出现硬错误异常。该异常将跳到 while(1) loop 锁相环。

映像前面必须带有呈表 17 中所述布局的标头。非加密的映像可以忽略标头。

表 17. 映像标头

地址	名称	描述	大小 [位]
5:0	AES_ACTIVE[1]	AES 加密有效 0x25 (100101)：AES 加密有效 0x1A (011010)：AES 加密无效 所有其他值：无效映像	6
7:6	HASH_ACTIVE[1]	指示是否使用了 Hash： 00: 使用 CMAC 哈希，值为 HASH_VALUE 01: 保留 10: 保留 11: 未使用 Hash	2
13:8	保留	11...11 （二进制）	6
15:14	AES_CONTROL	这 2 位可以设置为使 AES 加密有效的值，以便在 AES 加密后，AES_ACTIVE 字段不等于值 0x1A （AES 加密无效）。	2
31:16	HASH_SIZE[3]	映像中用于计算哈希值的部分的大小，以 512 字节帧的数目为单位。另外，在启动时复制映像大小到内部 SRAM 中。 Hash 大小 = 16[2]+ HASH_SIZE x 512 字节。	16
95:32	HASH_VALUE	如 HASH_SIZE 所示，CMAC hash 值基于映像的第一个字节（从标头开始）进行计算。该值截断到 64 MSB。	64
127:96	保留	11...11 （二进制）	32

[1] 只有在设备兼容 AES 时有效，否则视为无效映像。

[2] 标头字节数中需要 16 个额外字节。

[3] 映像大小应设置为不超过 0x1000 0000 处的 SRAM 大小。

5.3.4 启动映像创建

5.3.4.1 CMAC

用于映像认证的标签使用 CMAC 算法计算。标签存储在标头字段 HASH_VALUE 中。

认证过程原理如下所示：

1. 使用 CMAC 算法生成 128 位标签。将标签截断成 64 MSB，并在标头中插入该截断的标签。
2. 在启动时，将会重新计算标签。如果计算的标签等于映像标头中收到的标签，则认证通过。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

要使用 128 位的模块密码 AES 和密钥 K 为消息 M 生成 l 位的 CMAC 标签 T， CMAC 标签生成过程原理如下所示：

1. 生成子密钥 K_1：
   - 计算临时值 $K_0 = AES_K(0)$。
   - 如果 $msb(K_0) = 0$，则 $K_1 = (K_0 << 1)$，否则 $K_1 = (K_0 << 1) \oplus 0x87$
2. 将消息分成 128 位分块 $M_1 || ... || M_{n-1} || M_n^*$，其中 $M_1 ... M_{n-1}$ 为完整的模块。
3. 最后一个模块 M_n^* 应当填补为完整的模块，然后 $M_n = K_1 \oplus M_n^*$。
4. 使 $c_0 = 00...0$。
5. 对于 i = 1, ..., n，计算 $c_i = AES_K(c_{i-1} \oplus M_i)$。
6. 输出 $T = msb_l(c_n)$。

第一个消息模块为标头。由于 CMAC 标签存储在标头字段 HASH_VALUE 中，且此标签在 CMAC 计算之前未知，所以在 CMAC 计算期间将使用临时的假想标签值 0x3456789A 的临时标头。在最终的标头字段 HASH_VALUE 中，应使用计算的标签值替代此假想的值。

对于 LPC43xx，选定的 CMAC 参数为：加密密钥 K = 用户密钥（与用于解密的密钥相同），且标签长度 l = 64。数据按低位优先模式处理。这意味着从映像读取的第一个字节将作为最低有效字节集成到 AES 代码字中。从映像读取的第 16 个字节是第一个 AES 代码字中最高有效字节。

CMAC 基于标头和加密的映像进行计算。

图 8. CMAC 生成

5.3.4.2 UART 引导模式

图 9 详细介绍了 UART 启动模式的启动流程步骤。只有在相应设置了启动模式的情况下才会执行此模式 （参见启动模式表 15 和表 16）。

使用以下设置配置 UART （如图 9 所示）：

- 波特率 = 115200、 57600、 38400、 19200 或 9600
- 数据位 = 8
- 奇偶位 = 无
- 停止位 = 1

自动识别波特率功能被启用：等待收到 0x3F 后启动，并回复 “OK”。后面应跟标头和映像。 Boot ROM 在文件传输期间不会实施任何流程控制或握手机制。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

下载启动映像后，将（根据标头信息) 检查其是否为有效映像，再分别将 OK 或 FAILED 发送至主机，后跟 CR 和 LF。最后，将完整的引导映像复制、处理到地址 0x10000000 并从此处执行。

图 9. UART 启动流程 （仅适用于无片上闪存的器件）

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

5.3.4.3 EMC 引导模式

EMC 启动过程遵循图 10 中所示的主流程。CPU 时钟设置为 96 MHz，并且不支持 AES 的器件在映像不包含标头时，将直接从 EMC 启动。 EMC 使用 0xE 等待状态，在捕获数据前提供大约 156 ns 延迟。

请注意，引脚配置中所选的地址位数最初为 EMC_A[13:0]。所有高地址位引脚均被配置为下拉，但不是主动驱动。读取标头后，地址位将根据 HASH_SIZE 定义的映像大小进行扩展，例如，如果 HASH_SIZE 为 100 kB，则因为 2^{17} > 100 kB，将配置引脚 EMC_A[16:14]。如果启动时没有标头，则应为映像配置额外的地址引脚（如果需要更多），大于最初配置的 EMC_A[13:0]。该配置应在映像的初始 16 kB 区域发生。

如果不存在标头，则假设映像位于地址 0x1C00 0000 处，并从此处执行。

图 10. EMC 启动流程 （仅适用于无片上闪存的器件）

5.3.4.4 SPI 引导模式

在 SPI 模式下，启动使用 SSP0。 SPI 时钟为 18 MHz。

图 11 详细介绍了 SPI 闪存启动模式的启动流程步骤。只有在相应设置了启动模式的情况下才会执行此模式 （参见启动模式表 15 和表 16）。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 11. SPI 启动流程 （仅适用于无片上闪存的器件）

5.3.4.5 SPIFI 引导模式

图 12 详细介绍了四通道 SPI 闪存启动模式的启动流程步骤。只有

在相应设置了引导模式的情况下才会执行此模式（参见表 15 和表 16 中的启动模式）。SPIFI 时钟为 32 MHz。如果检测到的设备未知， SPIFI 时钟将减至 18 MHz。

如果不存在标头，则假设映像位于地址 0x80000000 处，并从此处执行。

图 12. SPIFI 启动流程 （仅适用于无片上闪存的器件）

5.3.4.6 USB 启动模式

从 USB 启动时，有两个 USB 接口可用。USB0 支持高速和全速，而 USB1 只支持全速。将 USB 时钟分别设为 480 MHz 或 60 MHz。 USB1 要求正确设置 VBUS 引脚。最初，禁用 USB0 PHY 以节约用电。使能后，可以开始枚举。 DFU 类用于下载启动映像。从主机接收启动映像后，根据上文所述的一组规则验证映像。如果有效，映像会被相应地处理到地址 0x10000000 并从此处执行。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

USB 产品和供应商 ID 由 OTP 定义（参见表 10 和表 11）。

图 13. USB 启动流程（仅适用于无片上闪存的器件）

5.3.5 引导流程时序

下列参数描述了启动过程的时序：

表 18. 引导流程时序参数

参数	描述	值
t_a	检查启动选择引脚	< 1.25 µs <tbd>
t_b	初始化设备	250 µs <tbd>
t_c	复制映像到嵌入式 SRAM	< 0.3 µs <tbd>
	如果器件从外部闪存执行，不含副本	
	如果映像已加密或必须复制	< 1 µs 到 10000 µs <tbd> 取决于映像大小和启动存储器的速度

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 14. 引导流程时序

5.3.6 ISP

在系统编程 (ISP) 是指利用启动引导程序软件和 USART0 串行端口，对片内 SRAM 存储器进行编程或重新编程。当器件位于最终用户端时，可执行 ISP。ISP 可以将数据加载到片内 SRAM 中并执行来自片内 SRAM 的代码。有关详情，请参阅第 45 章。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

6.1 本章导读

所有 LPC43xx 器件均支持 AES 解密。另请参见第 5 章。

6.2 功能要点

- 外部映像数据解密。
- 解密密钥的安全存储。
- 支持对待验证数据进行 CMAC Hash 计算。
- 支持无法读取的一个硬件密钥。
- 1 个字节 / 时钟周期的 AES 引擎性能。
- AES 引擎支持：
  - 采用 128 位密钥的 ECB 解码模式。
  - 采用 128 位密钥的 CBC 解码模式。
  - CMAC Hash 计算 （参见第 5.3.4.1 节）。

与启动过程有关的 AES 解密的详细信息如第 5 章中所述。

6.3 简介

LPC43xx 使用外部映像来存储指令代码和数据。 LPC43xx 提供硬件来保护外部映像内容，并加速处理数据解密、数据完整性和来源证明。

硬件包括：

- 一次性可编程 (OTP) 非易失性存储器，用于存储 AES 密钥。提供两个实例 (OTP1/2)，使用 OTP API 存储两个密钥 (表 14)。
- AES 引擎，用于执行 AES 解密。此引擎支持采用外部 GPDMA 模块来读取和写入数据。该引擎使用 128 位密钥，处理 128 位的分块。使用 AES API，密钥可以存储在软件不可见专用的硬件接口中。

可使用四种不同密钥加载 AES 引擎：

1. 密钥 1 – 存储在 OTP 中
2. 密钥 2 – 存储在 OTP 中
3. 随机生成的密钥
4. 软件定义密钥

注：在深度掉电和复位时不保留随机生成密钥和软件定义密钥，并且必须重新加载。

注：要更新随机数生成器 (RNG) 并加载新的随机数，则首先使用 otp_GenRand() API 调用，然后使用 aes_LoadKeyRNG() 调用 （参见第 6.5.5 节）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

6.4 AES API

AES 通过 LPC43xx ROM 中的一组简单的 API 调用进行控制。

执行 ROM 驱动器表中指针所指向的功能来进行 ROM 的 API 调用。

图 15. AES 驱动器指针结构

6.4.1 AES 功能分配

安全 API 控制着 AES 模块。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 19. 安全 API 调用

函数	相对于 API 输入点的偏移	描述
aes_Init	0x00	初始化 AES 引擎 参数 - void 返回 - void
aes_SetMode	0x04	定义 AES 引擎操作模式 参数：unsigned cmd 及值： 0 - 保留。不使用。 1 - AES_API_CMD_DECODE_ECB 2 - 保留。不使用。 3 - AES_API_CMD_DECODE_CBC 返回 - unsigned：参见常见的错误代码。
aes_LoadKey1	0x08	加载 128 位 AES 用户密钥 1 参数 - void 返回 - void
aes_LoadKey2	0x0C	加载 128 位 AES 用户密钥 2 参数 - void 返回 - void
aes_LoadKeyRNG	0x10	加载AES引擎中随机生成的密钥要更新RNG并加载新的随机数，则应在 aes_LoadKeyRNG 之前使用 API 调用 otp_GenRand。 参数 - void 返回 - void
aes_LoadKeySW	0x14	加载 128 位 AES 软件定义的用户密钥 参数 - unsigned char *key （16 字节） 返回 - void
aes_LoadIV_SW	0x18	加载 128 位 AES 初始化向量 参数 - unsigned char *iv （16 字节） 返回 - void
aes_LoadIV_IC	0x1C	加载 128 位 AES IC 特定的初始化向量，用于对启动映像进行解密。 参数 - void 返回 - void
aes_Operate	0x1C	使用 aes_Set_Mode 设置 AES 模式并加载合适的密钥和初始化向量后，执行 AES 解密。 参数 1 - unsigned char *data_out 参数 2 - unsigned char *data_in 参数 3 - unsigned size （128 位字 - 16 字节） 返回 - unsigned：参见常见的错误代码。
aes_ProgramKey1	0x20	对 OTP 中的 128 位 AES 密钥编程。 参数：unsigned char *key （16 字节） 返回 - unsigned：参见常见的错误代码。
aes_ProgramKey2	0x24	对 OTP 中的 128 位 AES 密钥编程。 参数：unsigned char *key （16 字节） 返回 - unsigned：参见常见的错误代码。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

6.5 功能说明

6.5.1 AES 解密

使用以下步骤对数据进行解密 （参见图 16）：

1. 使用 AES，用户密钥（OTP 中存储的 AES 用户密钥 1（参见表 9））作为 AES 密钥，iv=0来对标头进行解密。标头提供计算CMAC所依据的HASH_VALUE和HASH_SIZE。
2. 在标头中，用常数 0x3456789A 替换 HASH_VALUE。
3. 使用 AES，用户密钥作为 AES 密钥， iv=0 来对标头进行加密。
4. 计算第 6.5.2 节中定义的 CMAC 标签，并确认 64 MSB 与 HASH_VALUE 相等，如果不相等则复位设备。
5. 使用 CBC AES，用户密钥作为 AES 密钥， iv 如第 6.5.3 节中定义来对密文帧进行解密。待解密的帧数由值 HASH_SIZE 给定。如果需要解密更多的帧，则需要通过应用程序来完成。

可独立于其他密文帧来解密某一密文帧。这在需要访问随机帧时十分有用。

图 16. AES 解密流程

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

6.5.2 使用 AES 硬件加速的 CMAC

CMAC 是使用 AES 引擎的验证算法。

CMAC 基于密文进行计算。这要优于基于纯文本计算 CMAC，因为即使纯文本并非随机，密文也将更为随机 （使用 CBC 的缘故）。

生成子密钥

要使用 b 位模块密码 E 和密钥 k 为消息 m 生成 1 位 CMAC 标签 T，则首先用下列算法生成两个 b 位子密钥 k1 和 k2 （这相当于在有限域 GF(2b) 中乘以 x 和 x2）。让 << 指示标准左移位运算符：

1. 计算临时值 $k_0 = E_k(0)$。
2. 如果 $msb(k_0) = 0$，则 k1 = k0 << 1，否则 k1 = (k0 << 1) ⊕ C，其中 C 为某个只取决于 b 的常数。具体来说，C 是具有最少 1 的字典最靠前不可约 b 次二进制多项式的非首项系数。对于 b =128， C = 00...010000111。
3. 如果 $msb(k_1) = 0$，则 k2 = k1 << 1，否则 k2 = (k1 << 1) ⊕ C。

注：图像应由完整的模块组成； Mn* 为 128 位，因此从不使用 K2。

生成 CMAC 标签

要生成 CMAC 标签，请遵照以下步骤：

1. 将消息分为 b 位模块 $M = M_1 || ... || M_{n-1} || M_n^*$，其中 $M_1, ..., M_{n-1}$ 为完整模块。
2. $M_n = K_1 \oplus M_n^*$
3. 设置 $c_0 = 00...0$ （二进制）。
4. 对于 i = 1,..., n，计算 $c_i = E_k(c_{i-1} \oplus M_i)$.
5. 输出 $T = MSB_l(c_n)$。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 17. CMAC 生成

验证 CMAC 标签

已加密图像由启动代码验证。

6.5.3 AES 密钥的使用

软件密钥是软件定义的 AES 密钥。由于该密钥对软件可见，因此要比 OTP 中硬件定义密钥的安全性低。但是 OTP 仅可存储两个密钥，而在软件内可存储多个密钥。

如果相同数据被多次加密，则使用 128 位 AES 初始化向量 iv 来随机加密，初始化向量无需加密，也用于解密数据。对于 CMAC 计算，使用 AES 初始化向量 iv = 0。

对于 LPC43xx，使用用户特定的 iv：

iv = AES^{-1}（用户密钥， 1）

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

6.5.4 字节顺序

AES 引擎每个运算可处理 128 位 （16 字节）模块。要加载 / 存储 AES 模块，需充分使用 32 位基础设施。为方便起见，API 接口使用字节序而非字顺序。API 传递 / 获得字节数组的指针，并且 AES 低电平驱动器将指针类型转换为无符号的 32 位数组。显示使用 16 字节密钥的 16 字节数据 AES 加密。为简单起见，从 00 开始按递增顺序选择数据和密钥字节。

图 18. AES 字节顺序

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

6.5.5 在深度掉电模式下存储 AES 密钥

深度掉电模式下，所有 AES 信息都会丢失。唤醒后，需要重新加载 AES 密钥。如果在进入深度掉电模式前要使用相同的 RNG 密钥，则该 RNG 密钥可能会被存储在 0x4004 1000 的备份寄存器内。应保持下列处理顺序：

1. 通过调用 otp_GenRand() API 生成随机数。
2. 将该数存储在 RTC REGFILE 寄存器中。
3. 使用 aes_LoadKeySW 在 AES 引擎中加载该数。

每一次唤醒后，执行下列操作：

1. 从备份寄存器加载存储的随机数。
2. 使用 aes_LoadKeySW 在 AES 引擎中加载该数。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

7.1 本章导读

NVIC 中断源随器件不同而有所差异。

- 以太网中断：仅用于 LPC4350/30。
- USB0 中断：仅用于 LPC4350/30/20。
- USB1 中断：仅用于 LPC4350/30。

7.2 基本配置

在 LPC43xx 上，每个内核连接到各自的 NVIC、ARM Cortex-M4 NVIC 和 ARM Cortex-M0 NVIC 上。

7.3 功能要点

- 可嵌套向量中断控制器是 ARM Cortex-M4 和 M0 处理器的重要组成部分。
- 紧耦合方式使中断延迟大大缩短。
- NVIC 可控制系统的异常及外设中断。
- 支持软件中断生成。
- Cortex-M4 内核：
  - 最多 52 个中断
  - 可重定位向量表
  - 非屏蔽中断
  - 8 个可编程的中断优先级，带硬件优先级屏蔽
- Cortex-M0 内核：
  - 最多 32 个中断
  - 4 <tbd> 个可编程的中断优先级 （带硬件优先级屏蔽功能）

7.4 简介

可嵌套向量中断控制器 (NVIC) 是 Cortex-M4 和 M0 的一个重要组成部分。它与 CPU 紧密结合，降低了中断延时，并让新进中断可以得到高效处理。

有关 NVIC 操作的详细信息，请参阅 《Cortex-M4 用户指南》。

7.5 引脚描述

表 20. NVIC 引脚描述

功能	方向	描述
NMI	I	外部非屏蔽中断 (NMI) 输入

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

7.6 中断源

表 21 列出了每种外设功能的中断源。每个外设可以有一条或多条中断线连接到中断向量控制器。请注意，每条线可能代表多个源。

异常号码与条目在异常向量表中的存储位置相关。中断号码也用于一些其它环境，例如软件中断。

除了表 21 和表 22 中列出的信号外，NVIC 还处理非屏蔽中断 (NMI)。为使 NMI 基于外部信号操作， NMI 功能必须连接到相关的设备引脚 （P4_0 或 PE_4）。连接后，引脚中的逻辑信号将使 NMI 得到处理。有关详细信息，请参阅 《Cortex-M4 或 Cortex-M0 用户指南》。

7.6.1 Cortex-M4 中断源

表 21. 中断源与 Cortex-M4 NVIC 的连接

中断 ID	异常编号	向量偏移	函数	标志
0	16	0x40	DAC	
1	17	0x44	M0CORE	Cortex-M0 ；锁存 TXEV ；用于 M4-M0 通信
2	18	0x48	DMA	
3	19	0x4C	-	保留
4	20	0x50	-	保留
5	21	0x54	以太网	以太网中断
6	22	0x58	SDIO	SD/MMC 中断
7	23	0x5C	LCD	
8	24	0x60	USB0	OTG 中断
9	25	0x64	USB1	< 待定 >
10	26	0x68	SCT	SCT 共用中断
11	27	0x6C	RITIMER	
12	28	0x70	定时器 0	
13	29	0x74	定时器 1	
14	30	0x78	定时器 2	
15	31	0x7C	定时器 3	
16	32	0x80	MCPWM	电机控制 PWM
17	33	0x84	ADC0	
18	34	0x88	I2C0	
19	35	0x8C	I2C1	
20	36	0x90	SPI	
21	37	0x94	ADC1	
22	38	0x98	SSP0	
23	39	0x9C	SSP1	
24	40	0xA0	USART0	
25	41	0xA4	UART1	UART 和调制解调器共用中断
26	42	0xA8	USART2	
27	43	0xAC	USART3	UART 和调制解调器共用中断
28	44	0xB0	I2S0	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 21. 中断源与 Cortex-M4 NVIC 的连接 *（续）*

中断 ID	异常编号	向量偏移	函数	标志
29	45	0xB4	I2S1	
30	46	0xB8	SPIFI	
31	47	0xBC	SGPIO	
32	48	0xC0	PIN_INT0	GPIO 引脚中断 0
33	49	0xC4	PIN_INT1	GPIO 引脚中断 1
34	50	0xC8	PIN_INT2	GPIO 引脚中断 2
35	51	0xCC	PIN_INT3	GPIO 引脚中断 3
36	52	0xD0	PIN_INT4	GPIO 引脚中断 4
37	53	0xD4	PIN_INT5	GPIO 引脚中断 5
38	54	0xD8	PIN_INT6	GPIO 引脚中断 6
39	55	0xDC	PIN_INT7	GPIO 引脚中断 7
40	56	0xE0	GINT0	GPIO 全局中断 0
41	57	0xE4	GINT1	GPIO 全局中断 1
42	58	0xE8	EVENTROUTER	事件路由器中断
43	59	0xEC	C_CAN1	
44	60	0xF0	保留	
45	61	0xF4	保留	
46	62	0xF8	ATIMER	报警定时器中断
47	63	0xFC	RTC	
48	64	0x100	保留	
49	65	0x104	WWDT	
50	66	0x108	保留	
51	67	0x10C	C_CAN0	
52	68	0x110	QEI	

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

7.6.2 Cortex-M0 中断源

表 22. 中断源与 Cortex-M0 NVIC 的连接

中断 ID	异常编号	向量偏移	函数	标志
0	16	0x40	M0_RTC	
1	17	0x44	M0_M4CORE	来自 M4 内核的中断
2	18	0x48	M0_DMA	
3	19	0x4C	-	保留
4	20	0x50	-	保留
5	21	0x54	M0_ETHERNET	以太网中断
6	22	0x58	M0_SDIO	
7	23	0x5C	M0_LCD	
8	24	0x60	M0_USB0	OTG 中断
9	25	0x64	M0_USB1	
10	26	0x68	M0_SCT	SCT 共用中断
11	27	0x6C	M0_RITIMER_OR_ WWDT	RI 定时器中断或者 WWDT 中断
12	28	0x70	M0_TIMER0	
13	29	0x74	M0_GINT1	GPIO 全局中断 1
14	30	0x78	PIN_INT4	GPIO 引脚中断 4<tbd>
15	31	0x7C	M0_TIMER3	
16	32	0x80	M0_MCPWM	电机控制 PWM
17	33	0x84	M0_ADC0	
18	34	0x88	M0_I2C0_OR_I2C1	
19	35	0x8C	M0_SGPIO	
20	36	0x90	M0_SPI_OR_DAC	SPI 中断或者 DAC 中断
21	37	0x94	M0_ADC1	
22	38	0x98	M0_SSP0_OR_SSP1	SSP0 中断或者 SSP1 中断
23	39	0x9C	M0_EVENTROUTER	事件路由器
24	40	0xA0	M0_USART0	
25	41	0xA4	M0_UART1	调制解调器中断
26	42	0xA8	M0_USART_OR_ C_CAN1	USART2 中断或者 C_CAN1 中断
27	43	0xAC	M0_USART3	
28	44	0xB0	M0_I2S0_OR_I2S1_QEI	I2S0 OR I2S1 OR QEI 中断
29	45	0xB4	M0_C_CAN0	
30	46	0xB8	保留	
31	47	0xBC	保留	

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

8.1 本章导读

注：事件路由器控制着 NVIC 的唤醒流程以及其各种事件输入。

事件路由器源因不同的器件而异。

- 以太网：仅用于 LPC4350/30。
- USB0：仅用于 LPC4350/30/20。
- USB1：仅用于 LPC4350/30。

8.2 基本配置

- 关于时钟请参见表 23。
- 事件路由器连接到 NVIC 中的中断 #42 （参见表 21）。
- CREG0 寄存器将 WAKEUP0/1 引脚配置为事件路由器输入或输出，以监控事件路由器的输出 （参见表 38）。
- 要检测连接至任何外围中断的事件，将 HILO 寄存器中的相应位设置为高位（该寄存器的默认设置为低位）。
- 对于事件 #4 和 #5，激活 CREG0 寄存器中的 32 kHz 振荡器 （表 38）。

表 23. 事件路由器时钟和电源控制

	基础时钟	分支时钟	工作频率
事件路由器的时钟	BASE_M4_CLK	CLK_M4_BUS	最大 204 MHz

8.3 简介

事件路由器用于处理唤醒事件，例如特定的中断和外部或内部输入，以便从任何掉电模式（睡眠、深度睡眠、掉电和深度掉电模式）中唤醒。

在深度睡眠、掉电或深度掉电模式中，如果 32 kHz 振荡器在运行，只有四分之一 WAKEUP 引脚和 RTC 上的事件以及报警定时器事件有效 （参见表 24）。

所有事件都可以将器件从睡眠模式唤醒。但是，RTC 和报警定时器事件要求 32 kHz 振荡器在运行。

事件路由器具有来自各种外围设备的多种事件输入。如果 EDGE 配置寄存器中设置了合理的边沿检测，则事件路由器可以唤醒器件或在 NVIC 中产生中断。

每个到事件路由器的事件输入均可配置为，在上升沿或下降沿，或在高或低电平状况下启动输出信号。事件路由器会将所有事件合并成一个输出信号，用于以下目的：

- 如果在 NVIC 中使能了事件路由器中断，则创建中断。
- 向电源管理单元发送唤醒信号，以便从 “深度睡眠”、“掉电” 和 “深度掉电” 模式中唤醒。
- 向 CCU1 和 CCU2 发送唤醒信号，以便从睡眠模式中唤醒 （参见表 12.5.3）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 19. 事件路由器框图

8.4 事件路由器输入

表 24. 事件路由器输入

事件编号	源	备注
0	WAKEUP0 引脚	WAKEUP0 引脚。总是有效。用于从深度掉电和所有其他掉电模式中唤醒。
1	WAKEUP1 引脚	WAKEUP1 引脚。总是有效。用于从深度掉电和所有其他掉电模式中唤醒。
2	WAKEUP2 引脚	WAKEUP2 引脚。总是有效。用于从深度掉电和所有其他掉电模式中唤醒。
3	WAKEUP3 引脚	WAKEUP3 引脚。总是有效。用于从深度掉电和所有其他掉电模式中唤醒。
4	报警定时器外设	报警定时器中断。 32 kHz 振荡器在运行时激活。
5	RTC 外设	RTC 中断。 32 kHz 振荡器在运行时激活。
6	BOD 断路电平 1	BOD 中断；从低功耗模式中唤醒。深度掉电模式下不活动。用于从睡眠、深度睡眠和掉电模式唤醒。
7	WWDT 外设	WWDT 中断。深度睡眠、掉电和深度掉电模式下不活动。用于从睡眠模式唤醒。
8	以太网外设	唤醒包指示器。深度睡眠、掉电和深度掉电模式下不活动。用于从睡眠模式唤醒。
9	USB0 外设	唤醒请求信号。深度睡眠、掉电和深度掉电模式下不活动。用于从睡眠模式唤醒。
10	USB1 外设	ahb_needclk 信号。深度睡眠、掉电和深度掉电模式下不活动。用于从睡眠模式唤醒。
11	SD/MMC 外设	SD/MMC 中断。深度睡眠、掉电和深度掉电模式下不活动。用于从睡眠模式唤醒。
12	C_CAN0/1 外设	ORed C_CAN0 和 C_CAN1 中断。深度睡眠、掉电和深度掉电模式下不活动。用于从睡眠模式唤醒。
13	GIMA 输出 25	定时器共用输出 2（SCT 输出 2 的 OR 关系输出和定时器 0 的匹配通道 2）。参见表 139。深度睡眠、掉电和深度掉电模式下不活动。用于从睡眠模式唤醒。
14	GIMA 输出 26	定时器共用输出 6（SCT 输出 6 的 ORed 输出和定时器 1 的匹配通道 2）。参见表 139。深度睡眠、掉电和深度掉电模式下不活动。用于从睡眠模式唤醒。
15	QEI 外设	QEI 中断

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 24. 事件路由器输入 （续）

事件编号	源	备注
16	GIMA 输出 27	定时器共用输出 14（SCT 输出 14 的 ORed 输出和定时器 3 的匹配通道 2）。参见表 139。深度睡眠、掉电和深度掉电模式下不活动。用于从睡眠模式唤醒。
17	WAKEUP0 引脚	保留
18	-	保留
19	复位	< 待定 >

8.5 引脚描述

表 25. 事件路由器引脚描述

引脚	方向	描述
WAKEUP0/1	I/O	外部唤醒输入；可以提出事件路由器中断并从任何低功耗模式中唤醒。可以通过 CREG0 寄存器配置这些引脚以监控事件路由器输出 （表 38）。
WAKEUP2/3	I	外部唤醒输入；可以提出事件路由器中断并从任何低功耗模式中唤醒。

8.6 寄存器描述

表 26. 寄存器简介：事件路由器 （基址 0x4004 4000）

名称	访问类型	地址偏移	描述	复位值	参考
HILO	R/W	0x000	电平配置寄存器	0x000	表 27
EDGE	R/W	0x004	边沿配置	0x000	表 29
-	-	0x008 - 0xFD4	保留	-	-
CLR_EN	W	0xFD8	清除事件使能寄存器	0x0	表 30
SET_EN	W	0xFDC	设置事件使能寄存器	0x0	表 31
STATUS	R	0xFE0	事件状态寄存器	0x0	表 32
ENABLE	R	0xFE4	事件使能寄存器	0x0	表 33
CLR_STAT	W	0xFE8	清除事件状态寄存器	0x0	表 34
SET_STAT	W	0xFEC	设置事件状态寄存器	0x0	表 35

8.6.1 电平配置寄存器

此寄存器与边沿配置寄存器 EDGE （参见表 29）联合使用，可配置到事件路由器的每个输入的电平和边沿检测。

表 27. 电平配置寄存器 （HILO – 地址 0x4004 4000）位描述

位	符号	值	描述	复位值
0	WAKEUP0_L		WAKEUP0 事件的电平检测模式。	0
		0	如果 EDGE 寄存器中的位 0 为 0，则检测 WAKEUP0 引脚的低电平。如果EDGE 寄存器中的位 0 为 1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 0 为 0，则检测 WAKEUP0 引脚的高电平。如果EDGE 寄存器中的位 0 为 1，则检测上升沿。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 27. 电平配置寄存器 （HILO － 地址 0x4004 4000）位描述 *（续）*

位	符号	值	描述	复位值
1	WAKEUP1_L		WAKEUP1 事件的电平检测模式。EDGE 寄存器中的对应位必须为 0。	0
		0	如果 EDGE 寄存器的位 1 为 0，检测 WAKEUP1 引脚的低电平。	
		1	如果 EDGE 寄存器中的位 1 为 0，则检测 WAKEUP1 引脚的高电平。如果EDGE寄存器中的位1为1，则检测上升沿。	
2	WAKEUP2_L		WAKEUP2 事件的电平检测模式。	0
		0	如果 EDGE 寄存器中的位 2 为 0，则检测 WAKEUP2 引脚的低电平。如果EDGE寄存器中的位2为1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 2 为 0，则检测 WAKEUP2 引脚的高电平。如果EDGE寄存器中的位2为1，则检测上升沿。	
3	WAKEUP3_L		WAKEUP3 事件的电平检测模式。	0
		0	如果 EDGE 寄存器中的位 3 为 0，则检测 WAKEUP3 引脚的低电平。如果EDGE寄存器中的位3为1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 3 为 0，则检测 WAKEUP3 引脚的高电平。如果EDGE寄存器中的位3为1，则检测上升沿。	
4	ATIMER_L		报警定时器事件的电平检测模式。	0
		0	如果 EDGE 寄存器中的位 4 为 0，则检测报警定时器中断的低电平。如果 EDGE 寄存器中的位 4 为 1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 4 为 0，则检测报警定时器中断的高电平。如果 EDGE 寄存器中的位 4 为 1，则检测上升沿。	
5	RTC_L		RTC 事件的电平检测模式。	0
		0	如果 EDGE 寄存器中的位 5 为 0，则检测 RTC 中断的低电平。如果 EDGE 寄存器中的位 5 为 1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 5 为 0，则检测 RTC 中断的高电平。如果 EDGE 寄存器中的位 5 为 1，则检测上升沿。	
6	BOD_L		BOD 事件的电平检测模式。	0
		0	如果 EDGE 寄存器中的位 6 为 0，则检测 BOD 中断的低电平。如果 EDGE 寄存器中的位 6 为 1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 6 为 0，则检测 BOD 中断的高电平。如果 EDGE 寄存器中的位 6 为 1，则检测上升沿。	
7	WWDT_L		WWDT 事件的电平检测模式。	0
		0	如果 EDGE 寄存器中的位 7 为 0，则检测 WWDT 中断的低电平。如果 EDGE 寄存器中的位 7 为 1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 7 为 0，则检测 WWDT 中断的高电平。如果 EDGE 寄存器中的位 7 为 1，则检测上升沿。	
8	ETH_L		以太网事件的电平检测模式	0
		0	如果 EDGE 寄存器中的位 8 为 0，则检测以太网中断的低电平。如果 EDGE 寄存器中的位 8 为 1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 8 为 0，则检测以太网中断的高电平。如果 EDGE 寄存器中的位 8 为 1，则检测上升沿。	
9	USB0_L		USB0 事件的电平检测模式	0
		0	如果 EDGE 寄存器中的位 9 为 0，则检测 USB0 中断的低电平。如果 EDGE 寄存器中的位 9 为 1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 9 为 0，则检测 USB0 中断的高电平。如果 EDGE 寄存器中的位 9 为 1，则检测上升沿。	

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 27. 电平配置寄存器 （HILO － 地址 0x4004 4000）位描述 *（续）*

位	符号	值	描述	复位值
10	USB1_L		USB1 事件的电平检测模式	0
		0	如果 EDGE 寄存器中的位 10 为 0，则检测 USB1 中断的低电平。如果 EDGE 寄存器中的位 10 为 1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 10 为 0，则检测 USB1 中断的高电平。如果 EDGE 寄存器中的位 10 为 1，则检测上升沿。	
11	SDMMC_L		SD/MMC 事件的电平检测模式	0
		0	如果 EDGE 寄存器中的位 11 为 0，则检测 SD/MMC 中断的低电平。如果 EDGE 寄存器中的位 11 为 1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 11 为 0，则检测 SD/MMC 中断的高电平。如果 EDGE 寄存器中的位 11 为 1，则检测上升沿。	
12	CAN_L		C_CAN 事件的电平检测模式。	0
		0	如果 EDGE 寄存器中的位 12 为 0，则检测 C_CAN 共用中断的低电平。如果 EDGE 寄存器中的位 12 为 1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 12 为 0，则检测 C_CAN 共用中断的高电平。如果 EDGE 寄存器中的位 12 为 1，则检测上升沿。	
13	TIM2_L		组合定时器输出 2 事件的电平检测模式。	0
		0	如果 EDGE 寄存器中的位 13 为 0，则检测低电平 GIMA 输出 25。如果 EDGE 寄存器中的位 13 为 1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 13 为 0，则检测高电平 GIMA 输出 25。如果 EDGE 寄存器中的位 13 为 1，则检测上升沿。	
14	TIM6_L		组合定时器输出 6 事件的电平检测模式。	0
		0	如果 EDGE 寄存器中的位 14 为 0，则检测 GIMA 输出 26 的低电平。如果 EDGE 寄存器中的位 14 为 1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 14 为 0，则检测 GIMA 输出 26 的高电平。如果 EDGE 寄存器中的位 14 为 1，则检测上升沿。	
15	QEI_L		QEI 事件的电平检测模式。	0
		0	如果 EDGE 寄存器中的位 15 为 0，则检测 QEI 中断的低电平。如果 EDGE 寄存器中的位 15 为 1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 15 为 0，则检测 QEI 中断的高电平。如果 EDGE 寄存器中的位 15 为 1，则检测上升沿。	
16	TIM14_L		组合定时器输出 14 事件的电平检测模式。	0
		0	如果 EDGE 寄存器中的位 16 为 0，则检测 GIMA 输出 27 的低电平。如果 EDGE 寄存器中的位 16 为 1，则检测下降沿。	
		1	如果 EDGE 寄存器中的位 16 为 0，则检测 GIMA 输出 27 的高电平。如果 EDGE 寄存器中的位 16 为 1，则检测上升沿。	
18:17	-	-	保留。	
19	RESET_L		复位电平检测模式	0
		0	如果 EDGE 寄存器的位 17 为 0，检测低电平。如果 EDGE 寄存器的位 17 为 1，检测下降沿。	
		1	如果 EDGE 寄存器的位 17 为 0，检测高电平。如果 EDGE 寄存器的位 17 为 1，检测上升沿。	
31:20	-	-	保留。	

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

8.6.2 边沿配置寄存器

此寄存器与电平配置寄存器 HILO （参见表 27）联合使用，可配置到事件路由器的每个输入的电平或边沿检测。

EDGE 配置寄存器决定事件路由器是响应电平变更 (EDGEn=1)，还是保持常量电平 (EDGEn=0)。 HILOn 位确定对上升沿 (HILOn=1) 或下降沿 (HILOn=0) 的响应。

表 28. EDGE 和 HILO 寄存器设置组合

HILO 位 n	EDGE 位 n	描述
0	0	检测低电平
0	1	检测下降沿
1	0	检测高电平
1	1	检测上升沿

如果高电平检测有效，则在信号改为 “低” 之前无法清除事件路由器状态位。如果上升沿检测有效，则在事件发生之后可以清除事件路由器状态位。

表 29. 边沿配置寄存器 （EDGE - 地址 0x4004 4004）位描述

位	符号	值	描述	复位值
0	WAKEUP0_E		WAKEUP0 事件的边沿检测模式。EDGE 寄存器中的对应位必须为 0。	0
		0	电平检测。	
		1	WAKEUP0 引脚的边沿检测。如果 HILO 寄存器的位 0 为 0，检测下降沿。如果HILO寄存器的位0为1，检测上升沿。	
1	WAKEUP1_E		WAKEUP1 事件的边沿 / 电平检测模式。 EDGE 寄存器中的对应位必须为 0。	0
		0	电平检测。	
		1	WAKEUP1 引脚的边沿检测。如果 HILO 寄存器的位 1 为 0，检测下降沿。如果HILO寄存器的位1为1，检测上升沿。	
2	WAKEUP2_E		WAKEUP2 事件的边沿 / 电平检测模式。 EDGE 寄存器中的对应位必须为 0。	0
		0	电平检测。	
		1	WAKEUP2 引脚的边沿检测。如果 HILO 寄存器的位 2 为 0，检测下降沿。如果HILO寄存器的位2为1，检测上升沿。	
3	WAKEUP3_E		WAKEUP3 事件的边沿 / 电平检测模式。 EDGE 寄存器中的对应位必须为 0。	0
		0	电平检测。	
		1	WAKEUP3 引脚的边沿检测。如果 HILO 寄存器的位 30 为 0，检测下降沿。如果HILO寄存器的位3为1，检测上升沿。	
4	ATIMER_E		报警定时器事件的边沿 / 电平检测模式。 EDGE 寄存器中的对应位必须为 0。	0
		0	电平检测。	
		1	报警定时器中断的边沿检测。如果 HILO 寄存器的位 4 为 0，检测下降沿。如果HILO寄存器的位4为1，检测上升沿。	
5	RTC_E		RTC 事件的边沿 / 电平检测模式。EDGE 寄存器中的对应位必须为 0。	0
		0	电平检测。	
		1	RTC 中断的边沿检测。如果 HILO 寄存器的位 5 为 0，检测下降沿。如果 HILO 寄存器的位 5 为 1，检测上升沿。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 29. 边沿配置寄存器 （EDGE - 地址 0x4004 4004）位描述 *（续）*

位	符号	值	描述	复位值
6	BOD_E		BOD 事件的边沿 / 电平检测模式。EDGE 寄存器中的对应位必须为 0。	0
		0	电平检测。	
		1	BOD 中断的边沿检测。如果 HILO 寄存器的位 6 为 0，检测下降沿。如果 HILO 寄存器的位 6 为 1，检测上升沿。	
7	WWDT_E		WWDTD 事件的边沿 / 电平检测模式。EDGE 寄存器中的对应位必须为 0。	0
		0	电平检测。	
		1	WWDT 中断的边沿检测。如果 HILO 寄存器的位 7 为 0，检测下降沿。如果 HILO 寄存器的位 7 为 1，检测上升沿。	
8	ETH_E		EDGE 寄存器中对应的位必须为 0。	0
		0	电平检测。	
		1	以太网中断的边沿检测。如果 HILO 寄存器的位 8 为 0，检测下降沿。如果 HILO 寄存器的位 8 为 1，检测上升沿。	
9	USB0_E		EDGE 寄存器中对应的位必须为 0。	0
		0	电平检测。	
		1	USB0 中断的边沿检测。如果 HILO 寄存器的位 9 为 0，检测下降沿。如果 HILO 寄存器的位 9 为 1，检测上升沿。	
10	USB1_E		EDGE 寄存器中对应的位必须为 0。	0
		0	电平检测。	
		1	USB1 中断的边沿检测。如果 HILO 寄存器的位 10 为 0，检测下降沿。如果 HILO 寄存器的位 10 为 1，检测上升沿。	
11	SDMMC_E		SD/MMC 事件的边沿 / 电平检测模式。EDGE 寄存器中的对应位必须为 0。	
		0	电平检测。	
		1	SD/MMC 中断的边沿检测。如果 HILO 寄存器的位 10 为 0，检测下降沿。如果 HILO 寄存器的位 10 为 1，检测上升沿。	
12	CAN_E		C_CAN 事件的边沿 / 电平检测模式。EDGE 寄存器中的对应位必须为 0。	0
		0	电平检测。	
		1	C_CAN 共用中断的边沿检测。如果 HILO 寄存器的位 12 为 0，检测下降沿。如果 HILO 寄存器的位 12 为 1，检测上升沿。	
13	TIM2_E		组合定时器输出 2 事件的边沿 / 电平检测模式。 EDGE 寄存器中的对应位必须为 0。	0
		0	电平检测。	
		1	GIMA 输出 25 的边沿检测。如果 HILO 寄存器中的位 13 为 0，则检测下降沿。如果 HILO 寄存器中的位 13 为 1，则检测上升沿。	
14	TIM6_E		组合定时器输出 6 事件的边沿 / 电平检测模式。 EDGE 寄存器中的对应位必须为 0。	0
		0	电平检测。	
		1	GIMA 输出 26 的边沿检测。如果 HILO 寄存器中的位 14 为 0，则检测下降沿。如果 HILO 寄存器中的位 14 为 1，则检测上升沿。	

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 29. 边沿配置寄存器 （EDGE - 地址 0x4004 4004）位描述 *（续）*

位	符号	值	描述	复位值
15	QEI_E		QEI 中断信号的边沿 / 电平检测模式。 EDGE 寄存器中的对应位必须为 0。	0
		0	电平检测。	
		1	QEI 中断的边沿检测。如果 HILO 寄存器的位 15 为 0，检测下降沿。如果 HILO 寄存器的位 15 为 1，检测上升沿。	
16	TIM14_E		组合定时器输出 14 事件的边沿 / 电平检测模式。EDGE 寄存器中的对应位必须为 0。	0
		0	电平检测。	
		1	GIMA 输出 27 的边沿检测。如果 HILO 寄存器中的位 16 为 0，则检测下降沿。如果 HILO 寄存器中的位 16 为 1，则检测上升沿。	
18:17	-	-	保留。	
19	RESET_E		复位的边沿 / 电平检测模式。 EDGE 寄存器中的对应位必须为 0。	0
		0	电平检测。	
		1	复位信号的边沿检测。如果 HILO 寄存器的位 19 为 0，检测下降沿。如果 HILO 寄存器的位 19 为 1，检测上升沿。	
31:20	-	-	保留。	

8.6.3 清除事件使能寄存器

表 30. 清除事件使能寄存器 （CLR_EN – 地址 0x4004 4FD8）位描述

位	符号	描述	复位值
0	WAKEUP0_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 0。	-
1	WAKEUP1_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 1。	-
2	WAKEUP2_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 2。	-
3	WAKEUP3_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 3。	-
4	ATIMER_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 4。	-
5	RTC_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 5。	-
6	BOD_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 6。	-
7	WWDT_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 7。	-
8	ETH_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 8。	-
9	USB0_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 9。	-
10	USB1_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 10。	-
11	SDMMC_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 11。	-
12	CAN_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 12。	-
13	TIM2_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 13。	-
14	TIM6_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 14。	-
15	QEI_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 15。	-
16	TIM14_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 16。	-
18:17	-	保留。	-
19	RESET_CLREN	向此位写入 1 将清除 ENABLE 寄存器中的事件使能位 19。	-
31:20	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

8.6.4 设置事件使能寄存器

表 31. 事件设置使能寄存器 （SET_EN – 地址 0x4004 4FDC）位描述

位	符号	描述	复位值
0	WAKEUP0_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 0。	-
1	WAKEUP1_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 1。	-
2	WAKEUP2_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 2。	-
3	WAKEUP3_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 3。	-
4	ATIMER_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 4。	-
5	RTC_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 5。	-
6	BOD_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 6。	-
7	WWDT_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 7。	-
8	ETH_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 8。	-
9	USB0_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 9。	-
10	USB1_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 10。	-
11	SDMMC_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 11。	-
12	CAN_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 12。	-
13	TIM2_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 13。	-
14	TIM6_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 14。	-
15	QEI_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 15。	-
16	TIM14_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 16。	-
18:17	-	保留。	-
19	RESET_SETEN	向此位写入 1 将设置 ENABLE 寄存器中的事件使能位 19。	-
31:20	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

8.6.5 事件状态寄存器

表 32. 事件状态寄存器（STATUS – 地址 0x4004 4FE0）位描述

位	符号	描述	复位值
0	WAKEUP0_ST	此位为 1 表示已引发 WAKEUP0 事件。	-
1	WAKEUP1_ST	此位为 1 表示已引发 WAKEUP1 事件。	-
2	WAKEUP2_ST	此位为 1 表示已引发 WAKEUP2 事件。	-
3	WAKEUP3_ST	此位为 1 表示已引发 WAKEUP3 事件。	-
4	ATIMER_ST	此位为 1 表示已引发 ATIMER 事件。	-
5	RTC_ST	此位为 1 表示已引发 RTC 事件。	-
6	BOD_ST	此位为 1 表示已引发 BOD 事件。	-
7	WWDT_ST	此位为 1 表示已引发 WWDT 事件。	-
8	ETH_ST	此位为 1 表示已引发 ETHERNET 事件。	-
9	USB0_ST	此位为 1 表示已引发 USB0 事件。	-
10	USB1_ST	此位为 1 表示已引发 USB1 事件。	-
11	SDMMC_ST	此位为 1 表示已引发 SDMMC 事件。	-
12	CAN_ST	此位为 1 表示已引发 C_CAN 事件。	-
13	TIM2_ST	此位为 1 表示已引发组合定时器 2 输出事件。	-
14	TIM6_ST	此位为 1 表示已引发组合定时器 6 输出事件。	-
15	QEI_ST	此位为 1 表示已引发 QEI 事件。	-
16	TIM14_ST	此位为 1 表示已引发组合定时器 14 输出事件。	-
18:17	-	保留。	-
19	RESET_ST	此位为 1 表示已引发 <tbd> 事件。	-
31:20	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

8.6.6 事件使能寄存器

表 33. 事件使能寄存器 （ENABLE – 地址 0x4004 4FE4）位描述

位	符号	描述	复位值
0	WAKEUP0_EN	此位为 1 表示已使能 WAKEUP0 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
1	WAKEUP1_EN	此位为 1 表示已使能 WAKEUP1 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
2	WAKEUP2_EN	此位为 1 表示已使能 WAKEUP2 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
3	WAKEUP3_EN	此位为 1 表示已使能 WAKEUP3 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
4	ATIMER_EN	此位为 1 表示已使能 ATIMER 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
5	RTC_EN	此位为 1 表示已使能 RTC 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
6	BOD_EN	此位为 1 表示已使能 BOD 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
7	WWDT_EN	此位为 1 表示已使能 WWDT 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
8	ETH_EN	此位为 1 表示已使能 ETHERNET 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
9	USB0_EN	此位为 1 表示已使能 USB0 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
10	USB1_EN	此位为 1 表示已使能 USB1 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
11	SDMMC_EN	此位为 1 表示已使能 SDMMC 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
12	CAN_EN	此位为 1 表示已使能 CAN 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
13	TIM2_EN	此位为 1 表示已使能 TIM2 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
14	TIM6_EN	此位为 1 表示已使能 TIM6 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
15	QEI_EN	此位为 1 表示已使能 QEI 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
16	TIM14_EN	此位为 1 表示已使能 TIM14 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
18:17	-	保留	-
19	RESET_EN	此位为 1 表示已使能 RESET 事件。如果 STATUS 寄存器中的位 0 = 1，则此事件会唤醒芯片并可产生事件路由器中断。	0
31:20	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

8.6.7 清除事件状态寄存器

表 34. 清除事件状态寄存器 （CLR_STAT – 地址 0x4004 4FE8）位描述

位	符号	描述	复位值
0	WAKEUP0_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 0。	-
1	WAKEUP1_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 1。	-
2	WAKEUP2_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 2。	-
3	WAKEUP3_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 3。	-
4	ATIMER_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 4。	-
5	RTC_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 5。	-
6	BOD_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 6。	-
7	WWDT_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 7。	-
8	ETH_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 8。	-
9	USB0_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 9。	-
10	USB1_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 10。	-
11	SDMMC_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 11。	-
12	CAN_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 12。	-
13	TIM2_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 13。	-
14	TIM6_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 14。	-
15	QEI_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 15。	-
16	TIM14_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 16。	-
18:17	-		-
19	RESET_CLRST	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 事件位 19。	-
31:20	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

8.6.8 设置事件状态寄存器

表 35. 设置事件状态寄存器 （SET_STAT – 地址 0x4004 4FEC）位描述

位	符号	描述	复位值
0	WAKEUP0_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 0。	-
1	WAKEUP1_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 1。	-
2	WAKEUP2_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 2。	-
3	WAKEUP3_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 3。	-
4	ATIMER_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 4。	-
5	RTC_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 5。	-
6	BOD_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 6。	-
7	WWDT_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 7。	-
8	ETH_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 8。	-
9	USB0_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 9。	-
10	USB1_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 10。	-
11	SDMMC_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 11。	-
12	CAN_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 12。	-
13	TIM2_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 13。	-
14	TIM6_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 14。	-
15	QEI_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 15。	-
16	TIM14_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 16。	-
18:17	-	保留。	-
19	RESET_SETST	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 事件位 19。	-
31:20	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

9.1 本章导读

可用的外围设备随器件不同而有所差异。

- 以太网：仅用于 LPC4350/30。
- USB0：仅用于 LPC4350/30/20。
- USB1：仅用于 LPC4350/30。

如果外围设备不可用，则 CREG 寄存器中对应的位将保留。

9.2 基本配置

CREG 模块的配置方式如下：

- 参见表 36 以了解时钟与电源控制。
- CREG 模块无法通过软件复位。

表 36. CREG 时钟和电源控制

	基础时钟	分支时钟	工作频率
CREG	BASE_M4_CLK	CLK_M4_CREG	最大 204 MHz

9.3 功能要点

以下设置在配置寄存器模块中控制：

- ETB SRAM 配置
- BOD 断路电平设置
- 振荡器输出
- DMA 至外设复用
- 以太网模式
- 存储器映射
- 定时器 /UART 输入
- USB1 PHY 控制
- RTC_ALARM 和 WAKEUP0/1 引脚功能

此外， Creg 模块包含器件 ID 和器件配置信息。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

9.4 寄存器描述

表 37. 寄存器简介：配置寄存器 （基址 0x4004 3000）

名称	访问类型	地址偏移	描述	复位值	EMC、UART0/3 启动后的复位值	USB0/1 启动后的复位值	参考
-	-	0x000	保留	-	-	-	-
CREG0	R/W	0x004	芯片配置寄存器 32 kHz 振荡器输出和 BOD 控制寄存器。	< 待定 >	0xF3C	0xF1C	表 38
-	-	0x008	保留	-	-	-	-
-	-	0x008 - 0x0FC	保留	-	-	-	-
M4MEMMAP	R/W	0x100	ARM Cortex-M4 存储器映射	0x10400	0x1000 0000	0x1000 0000	表 39
-	-	0x104	保留	< 待定 >	< 待定 >	< 待定 >	< 待定 >
CREG1	RO	0x108	芯片配置寄存器 1	< 待定 >	< 待定 >	< 待定 >	< 待定 >
CREG2	RO	0x10C	芯片配置寄存器 2	< 待定 >	< 待定 >	< 待定 >	< 待定 >
CREG3	RO	0x110	芯片配置寄存器 3	< 待定 >	< 待定 >	< 待定 >	< 待定 >
CREG4	RO	0x114	芯片配置寄存器 4	< 待定 >	< 待定 >	< 待定 >	< 待定 >
CREG5	R/W	0x118	芯片配置寄存器 5。控制 JTAG 访问。	< 待定 >	< 待定 >	< 待定 >	表 40
DMAMUX	R/W	0x11C	DMA 复用控制	< 待定 >	< 待定 >	< 待定 >	表 41
-	-	0x120 - 0x124	保留	-	-	-	-
ETBCFG	R/W	0x128	ETB RAM 配置	0	0x1	< 待定 >	表 42
CREG6	R/W	0x12C	芯片配置寄存器 6。控制多重功能：以太网接口、 SCT 输出、 I2S0/1 输入、 EMC 时钟。	0	< 待定 >	< 待定 >	表 43
M4TXEVENT	R/W	0x130	Cortex-M4 TXEV 事件清除	0	< 待定 >	< 待定 >	表 44
-	-	0x134 - 0x1FC	保留	-	-	-	-
CHIPID	RO	0x200	器件 ID	< 待定 >	< 待定 >	< 待定 >	表 45
-	-	0x204 - 0x2FC	保留	-	-	-	-
-	-	0x300	保留	-	-	-	-
-	-	0x304	保留	-	-	-	-
-	-	0x308	保留	-	-	-	-
-	-	0x30C - 0x3FC	保留	-	-	-	-
M0TXEVENT	R/W	0x400	Cortex-M0 TXEV 事件清除	0	< 待定 >	< 待定 >	表 46
M0APPMEMMAP	R/W	0x404	ARM Cortex-M0 存储器映射	0x20000	< 待定 >	< 待定 >	表 47

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

9.4.1 CREG0 控制寄存器

表 38. CREG0 寄存器 （CREG0，地址 0x4004 3004）位描述

位	符号	值	描述	复位值	访问类型
0	EN1KHZ		使能 1 kHz 输出。	0	R/W
		0	禁用 1 kHz 输出。		
		1	1 kHz 输出已使能。		
1	EN32KHZ		使能 32 kHz 输出。	0	R/W
		0	禁用 32 kHz 输出。		
		1	32 kHz 输出已使能。		
2	RESET32KHZ		32 kHz 振荡器复位	1	R/W
		0	清除复位。		
		1	激活复位。		
3	PD32KHZ		32 kHz 电源控制。	1	R/W
		0	上电。		
		1	掉电。		
4	-		保留	-	-
5	USB0PHY		USB0 PHY 电源控制。	1	R/W
		0	使能 USB0 PHY 功耗。		
		1	禁用 USB0 PHY。 PHY 掉电。		
7:6	ALARMCTRL		RTC_ALARM 引脚输出控制	0	R/W
		0x0	RTC 报警。		
		0x1	Event 路由器事件。		
		0x2	保留。		
		0x3	不工作。		
9:8	BODLVL1		BOD 断路电平以产生中断。	0x3	R/W
		0x0	2.8 V		
		0x1	2.9 V		
		0x2	3.0 V		
		0x3	3.1 V		
11:10	BODLVL2		BOD 断路电平以进行复位。	0x3	R/W
		0x0	1.75 V		
		0x1	1.85 V		
		0x2	1.95 V		
		0x3	2.05 V		
13:12	-		保留	-	-
15:14	WAKEUP0CTRL		WAKEUP0 引脚输入 / 输出控制	0	R/W
		0x0	事件路由器输入。		
		0x1	事件路由器输出。		
		0x2	保留。		
		0x3	事件路由器输入。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 38. CREG0 寄存器 （CREG0，地址 0x4004 3004）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
17:16	WAKEUP1CTRL		WAKEUP1 引脚输入 / 输出控制	0	R/W
		0x0	事件路由器输入。		
		0x1	事件路由器输出。		
		0x2	保留		
		0x3	事件路由器输入。		
31:18	-		保留	-	-

9.4.2 ARM Cortex-M4 存储器映射寄存器

表 39. 存储器映射寄存器 （M4MEMMAP，地址 0x4004 3100）位描述

位	符号	描述	复位值	访问类型
11:0		保留	0x000	-
31:12	M4MAP	访问位于地址 0x0000 0000 的存储器时的影子地址	0x10400	R/W

9.4.3 CREG5 控制寄存器

表 40. CREG5 控制寄存器 （CREG5，地址 0x4004 3118）位描述

位	符号	值	描述	复位值	访问类型
4:0	-		保留。	-	-
5	-		保留。	0	-
6	M4TAPSEL		JTAG 调试选择 <tbd>	< 待定 >	R/W
		0	旁路 JTAG 调试。		
		1	使能 JTAG 调试。		
7	-		保留。	0	-
8	-		保留。	0	-
31:9	-		保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

9.4.4 DMA 复用寄存器

此寄存器控制将哪组外围设备连接到 DMA 控制器 （参见表 260）。

表 41. DMA 复用寄存器 （DMAMUX，地址 0x4004 311C）位描述

位	符号	值	描述	复位值	访问类型
1:0	DMAMUXCH0		选择外围设备连接的 DMA，用于 DMA 外围设备 0。	0	R/W
		0x0	SPIFI		
		0x1	SCT 匹配 2		
		0x2	保留		
		0x3	T3 匹配 1		
3:2	DMAMUXCH1		选择外围设备连接的 DMA，用于 DMA 外围设备 1。	0	R/W
		0x0	定时器 0 匹配 0		
		0x1	USART0 发送		
		0x2	保留		
		0x3	保留		
5:4	DMAMUXCH2		选择外围设备连接的 DMA，用于 DMA 外围设备 2。	0	R/W
		0x0	定时器 0 匹配 1		
		0x1	USART0 接收		
		0x2	保留		
		0x3	保留		
7:6	DMAMUXCH3		选择外围设备连接的 DMA，用于 DMA 外围设备 3。	0	R/W
		0x0	定时器 1 匹配 0		
		0x1	UART1 发送		
		0x2	I2S1 通道 0		
		0x3	SSP1 发送		
9:8	DMAMUXCH4		选择外围设备连接的 DMA，用于 DMA 外围设备 4。	0	R/W
		0x0	定时器 1 匹配 1		
		0x1	UART1 接收		
		0x2	I2S1 通道 1		
		0x3	SSP1 接收		
11:10	DMAMUXCH5		选择外围设备连接的 DMA，用于 DMA 外围设备 5。	0	R/W
		0x0	定时器 2 匹配 0		
		0x1	USART2 发送		
		0x2	SSP1 发送		
		0x3	保留		

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 41. DMA 复用寄存器 （DMAMUX，地址 0x4004 311C）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
13:12	DMAMUXCH6		选择外围设备连接的 DMA，用于 DMA 外围设备 6。	0	R/W
		0x0	定时器 2 匹配 1		
		0x1	USART2 接收		
		0x2	SSP1 接收		
		0x3	保留		
15:14	DMAMUXCH7		选择外围设备连接的 DMA，用于 DMA 外围设备 7。	0	R/W
		0x0	定时器 3 匹配 I 0		
		0x1	USART3 发送		
		0x2	SCT 匹配输出 0		
		0x3	保留		
17:16	DMAMUXCH8		选择外围设备连接的 DMA，用于 DMA 外围设备 8。	0	R/W
		0x0	定时器 3 匹配 1		
		0x1	USART3 接收		
		0x2	SCT 匹配输出 1		
		0x3	保留		
19:18	DMAMUXCH9		选择外围设备连接的 DMA，用于 DMA 外围设备 9。	0	R/W
		0x0	SSP0 接收		
		0x1	I2S0 通道 0		
		0x2	SCT 匹配输出 1		
		0x3	保留		
21:20	DMAMUXCH10		选择外围设备连接的 DMA，用于 DMA 外围设备 10。	0	R/W
		0x0	SSP0 发送		
		0x1	I2S0 通道 1		
		0x2	SCT 匹配输出 0		
		0x3	保留		
23:22	DMAMUXCH11		选择外围设备连接的 DMA，用于 DMA 外围设备 11。	0	R/W
		0x0	SSP1 接收		
		0x1	保留		
		0x2	USART0 发送		
		0x3	保留		
25:24	DMAMUXCH12		选择外围设备连接的 DMA，用于 DMA 外围设备 12。	0	R/W
		0x0	SSP1 发送		
		0x1	保留		
		0x2	USART0 接收		
		0x3	保留		

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 41. DMA 复用寄存器 （DMAMUX，地址 0x4004 311C）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
27:26	DMAMUXCH13		选择外围设备连接的 DMA，用于 DMA 外围设备 13。	0	R/W
		0x0	ADC0		
		0x1	保留		
		0x2	SSP1 接收		
		0x3	USART3 接收		
29:28	DMAMUXCH14		选择外围设备连接的 DMA，用于 DMA 外围设备 14。	0	R/W
		0x0	ADC1		
		0x1	保留		
		0x2	SSP1 发送		
		0x3	USART3 发送		
31:30	DMAMUXCH15		选择外围设备连接的 DMA，用于 DMA 外围设备 15。	0	R/W
		0x0	DAC		
		0x1	SCT 匹配输出 3		
		0x2	保留		
		0x3	定时器 3 匹配 0		

9.4.5 ETB SRAM 配置寄存器

此寄存器选择用于地址为 0x2000 C000 的 RAM 16 kB 块的接口。此 RAM 存储器块可通过 ETB 访问，或用作 AHB 总线上的普通 SRAM。

请注意，默认情况下是通过 ETB 访问此存储器。

表 42. ETB SRAM 配置寄存器 （ETBCFG，地址 0x4004 3128）位描述

位	符号	值	描述	复位值	访问类型
0	ETB		选择 SRAM 接口	0	R/W
		0	ETB 访问位于地址 0x2000 C000 的 SRAM。		
		1	AHB 访问位于地址 0x2000 C000 的 SRAM。		
31:1	-		保留。	-	-

9.4.6 CREG6 控制寄存器

此寄存器控制 LPC43xx 的各个方面：

- 位 2:0 控制以太网 PHY 接口。以太网模块在设置期间会读取此寄存器。因此在更改 PHY 接口后必须复位以太网。
- 位 4 选择连接至 CTOUT_n 引脚的 SCT 输出以及选定的 GIMA 输入的功能：
  - SCT 输出或者定时器匹配输出 （默认）。参见 <tbd>
  - 仅 SCT 输出。
- 位 12:15 控制 I2S 时钟连接。
- 位 16 控制外部存储控制器时钟。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 43. CREG6 控制寄存器 （CREG6，地址 0x4004 312C）位描述

位	符号	值	描述	复位值	访问类型
2:0	ETHMODE		选择以太网模式。更改 PHY 接口之后复位以太网。 保留所有其它设置。		R/W
		0x0	MII		
		0x4	RMII		
3	-		保留。		R/W
4	CTOUTCTRL		选择 SCT 输出的功能。	0	R/W
		0	SCT 和定时器共用匹配输出。 SCT 输出或者定时器输出。		
		1	仅 SCT 输出。 SCT 输出用于无定时器匹配输出时。		
11:5	-		保留。	-	-
12	I2S0_TX_SCK_IN_SEL		I2S0_TX_SCK 输入选择	0	R/W
		0	按 I2 S 发送模式寄存器表 895 定义，选择 I2 S 时钟。		
		1	用于 I2S 发送时钟 MCLK 输入和 MCLK 输出的音频 PLL。 I2S 必须以从机模式配置。		
13	I2S0_RX_SCK_IN_SEL		I2S0_RX_SCK 输入选择	0	R/W
		0	按 I2 S 接收模式寄存器表 896 定义，选择 I2 S 时钟。		
		1	用于 I2S 接收时钟 MCLK 输入和 MCLK 输出的音频 PLL。 I2S 必须以从机模式配置。		
14	I2S1_TX_SCK_IN_SEL		I2S1_TX_SCK 输入选择	0	R/W
		0	按 I2 S 发送模式寄存器表 895 定义，选择 I2 S 时钟。		
		1	用于 I2S 发送时钟 MCLK 输入和 MCLK 输出的音频 PLL (PLL0AUDIO)。 I2S 必须以从机模式配置。		
15	I2S1_RX_SCK_IN_SEL		I2S1_RX_SCK 输入选择	0	R/W
		0	按 I2 S 接收模式寄存器表 896 定义，选择 I2 S 时钟。		
		1	用于 I2S 接收时钟 MCLK 输入和 MCLK 输出的音频 PLL (PLL0AUDIO)。 I2S 必须以从机模式配置。		
16	EMC_CLK_SEL		EMC_CLK 分频时钟选择 （参见第 21.1 章）。	0	R/W
		0	EMC_CLK_DIV 未分频。		
		1	EMC_CLK_DIV 二分频。		
31:17	-		保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

9.4.7 Cortex-M4 TXEV 事件清除寄存器

该寄存器捕获来自 ARM Cortex-M4 处理器的信号 TXEV （参见第 2.4.2 节）。

表 44. M4 TXEV 清除寄存器 （M4TXEVENT，地址 0x4004 3130）位描述

位	符号	值	描述	复位值	访问类型
0	TXEVCLR		Cortex-M4 TXEV 事件。	0	R/W
		0	清除 TXEV 事件。		
		1	无效。		
31:1	-		保留。	-	-

9.4.8 器件 ID 寄存器

表 45. 器件 ID 寄存器 （CHIPID，地址 0x4004 3200）位描述

位	符号	描述	复位值	访问类型
31:0	ID	< 待定 >	-	RO

9.4.9 Cortex-M0 TXEV 事件清除寄存器

该寄存器捕获来自 ARM Cortex-M0 处理器的信号 TXEV （参见第 2.4.2 节）。

表 46. Cortex-M0 TXEV 清除寄存器 （M0TXEVENT，地址 0x4004 3400）位描述

位	符号	值	描述	复位值	访问类型
0	TXEVCLR		Cortex-M0 TXEV 事件。	0	R/W
		0	清除 TXEV 事件。		
		1	无效。		
31:1	-		保留。	-	-

9.4.10 ARM Cortex-M0 存储器映射寄存器

表 47. 存储器映射寄存器 （M0APPMEMMAP，地址 0x4004 3404）位描述

位	符号	描述	复位值	访问类型
11:0		保留	0	-
31:12	M0APPMAP	访问位于地址 0x0000 0000 的存储器时的影子地址	0x20000	R/W

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

10.1 本章导读

所有 LPC43xx 器件上的电源管理控制器都相同。

10.2 基本配置

PMC 控制掉电模式 （深度睡眠、掉电和深度掉电）。

10.3 简介

PMC 执行控制序列，以便能够在不同的功耗模式之间跳变并控制每个外围设备的功耗状态。此外，支持根据硬件事件从掉电模式中唤醒。

掉电模式只能从工作模式中进入，且不允许在掉电模式之间跳变。通过 WFI 或 WFE 指令进入掉电模式。唤醒由中断或事件引起。中断原因在 NVIC 中捕获（第 7 章），事件在事件路由器中捕获 （第 8 章）。

PMC 支持以下掉电模式：深度睡眠、掉电和深度掉电。从任何掉电模式唤醒都始终会进入工作模式。

LPC43xx 按最高到最低功耗的顺序，支持五种功耗模式：

1. 工作模式
2. 睡眠模式 （由 ARM Cortex-M4 内核控制）
3. 掉电模式：
   a. 深度睡眠模式 （由 ARM Cortex-M4 内核控制）
   b. 掉电模式 （由 ARM Cortex-M4 内核控制）
   c. 深度掉电模式

注：在可以选择深度睡眠模式或掉电模式前，必须通过 CGU 寄存器将 IRC 选为所有输出时钟的时钟源 （参见第 11.8.1 节），并且所有 PLL 必须为掉电状态。

10.3.1 工作模式

默认情况下， LPC43xx 位于工作模式，指每个外围设备可以在标称的操作条件下执行功能操作。其他低功耗模式属于待机模式，在这些模式下，禁用外围设备时钟并会调整操作条件以进一步节约用电。在唤醒时，外围设备时钟会重新使能。

在工作 （或睡眠模式）下，支持以下工作模式。这些操作模式可通过 API 进行编程。

- 低功耗模式：CPU 和内核逻辑经过优化以实现高功效比而非最高性能。
- 正常功耗模式：CPU 以标称性能水平运行。这是复位后的默认设置。
- 高性能模式：CPU 以高性能水平运行。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

10.3.2 睡眠模式

在睡眠模式下，CPU 时钟将关闭以节约用电，而外设仍然保持工作且功能正常。如果 ARM Cortex-M4 系统控制寄存器中的 SLEEPDEEP 位设为 0，可以通过 WFI 或 WFE 指令进入睡眠模式。

支持与工作模式中相同的操作模式 （降低功耗、标称和高性能）。

10.3.3 深度睡眠模式

在深度睡眠模式中，CPU 时钟和外围设备时钟均会关闭以节约用电；逻辑状态和 SRAM 存储器会保持正常。所有模拟模块和 BOD 控制电路均会掉电。如果 ARM Cortex-M4 系统控制寄存器中的 SLEEPDEEP 位设为 1，且 PD0_SLEEP0_MODE 寄存器 （参见表 51）使用深度睡眠模式值编程，可以通过 WFI 或 WFE 指令进入深度睡眠模式。

当 LPC43xx 从深度睡眠模式中唤醒时， 12 MHz IRC 用作所有基本时钟的时钟源。

注：进入深度睡眠模式之前，按以下方式对 CGU 编程：

- 将所有基础时钟的时钟源切换到 IRC。
- 将 PLL 置于掉电模式。

对 CGU 重新编程可避免唤醒时使用任何未定义或未锁定的 PLL 时钟，并将深度睡眠模式下的功耗降至最低。

10.3.4 掉电模式

在掉电模式下， CPU 时钟和外围设备时钟均会关闭，但逻辑状态将保持正常。所有 SRAM 存储器（位于 0x1008 0000 的上 8 kB 本地 SRAM 除外）、模拟模块和 BOD 控制电路均处于掉电状态。如果 ARM Cortex-M4 系统控制寄存器中的 SLEEPDEEP 位设为 1，且 PD0_SLEEP0_MODE 寄存器 （参见表 51）使用掉电模式值编程，可以通过 WFI 或 WFE 指令进入掉电模式。

当 LPC43xx 从掉电模式中唤醒时， 12 MHz IRC 用作所有基本时钟的时钟源。

注：进入掉电模式之前，按以下方式对 CGU 编程：

- 将所有基础时钟的时钟源切换到 IRC。
- 将 PLL 置于掉电模式。

对 CGU 重新编程可避免唤醒时使用任何未定义或未锁定的 PLL 时钟，并将掉电模式下的功耗降至最低。

10.3.5 深度掉电模式

在深度掉电模式下，整个内核逻辑均处于掉电状态，且包括 I/O 管脚在内的整个系统的逻辑状态丢失。只有 RTC 电源域中的逻辑保持工作。如果 ARM Cortex-M4 系统控制寄存器中的 SLEEPDEEP 位设为 1，且 PD0_SLEEP0_MODE 寄存器 （参见表 51）使用深度掉电模式值编程，可以通过 WFI 或 WFE 指令进入深度掉电模式。

当 LPC43xx 从深度掉电模式中唤醒时，启动引导程序将 PLL1 配置为以 96 MHz 运行的时钟源。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

10.3.6 掉电模式下的存储器保持

表 48 显示了在睡眠模式和各种掉电模式中保留 SRAM 的哪些部分。

此外，外设模块（USB0/1、LCD、CAN、以太网、USART0/2/3、UART）中包含的全部 FIFO 存储器在睡眠模式和深度睡眠模式中保持，而在掉电模式和深度掉电模式中则不保持。

表 48. 存储器保持

模式	开始于 0x1000 0000 的 128 kB 本地 SRAM	开始于 0x1008 0000 的 64 kB 本地 SRAM	开始于 0x1009 0000 的 8 kB 本地 SRAM	开始于 0x2000 0000 的 64 kB AHB SRAM	位于 0x4004 1000（RTC 电源域）的 256 字节备份寄存器
睡眠模式	是	是	是	是	是
深度睡眠模式	是	是	是	是	是
掉电模式	否	否	是	否	是
深度掉电模式	否	否	否	否	是

10.4 寄存器描述

表 49. 寄存器简介：功耗模式控制器 (PMC)（基址 0x4004 2000）

名称	访问类型	地址偏移	描述	复位值	参考
PD0_SLEEP0_HW_ENA	R/W	0x000	硬件睡眠事件使能寄存器	0x0000 0001	表 50
-	-	0x004 - 0x018	保留	-	-
PD0_SLEEP0_MODE	R/W	0x01C	掉电模式控制寄存器	0x003F FF7F	表 51

10.4.1 硬件睡眠事件使能寄存器 PD0_SLEEP0_HW_ENA

表 50. 硬件睡眠事件使能寄存器（PD0_SLEEP0_HW_ENA – 地址 0x4004 2000）位描述

位	符号	描述	复位值	访问类型
0	ENA_EVENT0	写入 1 将为 Cortex-M4 使能任何掉电模式。	1	R/W
31:1	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-	-

10.4.2 掉电模式寄存器 PD0_SLEEP0_MODE

发出 ARM WFE/WFI 指令且 SLEEPDEEP 位设置为 1 时，PD0_SLEEP0_MODE 寄存器控制进入三种低功耗模式（深度睡眠、掉电或深度掉电）中的哪一种。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 51. 掉电模式寄存器 （PD0_SLEEP0_MODE – 地址 0x4004 201C）位描述

位	符号	描述	复位值	访问类型
31:0	PWR_STATE	在深度睡眠、掉电和深度掉电模式之间选择。 在此寄存器中只能对以下三个值之一进行编程： 0x0030 00AA = 深度睡眠模式 0x0030 FCBA = 掉电模式 0x003F FF7F = 深度掉电模式	0x003F FF7F	R/W

10.5 功能说明

10.5.1 运行时编程

PD0_SLEEP0_MODE 寄存器在运行时可进行编程，以便在下次跳变到降低功耗状态之后更改 LPC43xx 的默认功耗状态。

表 52. PMC 功耗模式的典型设置

掉电模式	描述	PD0_SLEEP0_MODE 寄存器位设置
深度睡眠	CPU、外设、模拟、USB PHY 处于保持模式；所有 SRAM 设备处于工作模式； BOD 处于掉电模式。	0x0030 00AA
掉电	CPU、外设、模拟设备处于保持模式； USB PHY 处于掉电模式； SRAM1 处于工作模式；所有其他 SRAM 处于掉电模式； BOD 处于掉电模式。	0x0030 FCBA
深度掉电模式	CPU、外设、模拟、USB PHY 处于掉电模式；所有 SRAM 处于掉电模式； BOD 处于掉电模式。	0x003F FF7F

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

11.1 本章导读

以太网、USB0、USB1 和 LCD 相关时钟并非在所有封装中均可用。参见第 1.3 节。对应的时钟控制寄存器保留。

VADC 外设仅适用于 <tbd> 器件。

11.2 基本配置

CGU 的配置方式如下：

- 参见表 53 以了解时钟与电源控制。
- 请勿在正常操作期间复位 CGU。

表 53. CGU 时钟和电源控制

	基础时钟	分支时钟	工作频率
CGU	BASE_M4_CLK	CLK_M4_BUS	最大 204 MHz

11.3 功能要点

- PLL 控制
- 支持三个 PLL：
  - 用于为高速 USB0 创建 480 MHz 时钟的 PLL0USB
  - 用于为高精度音频应用创建各种频率的带小数分频器的 PLL0AUDIO
  - 用于创建内核和外设时钟的 PLL1。
- 振荡器控制
- 时钟产生和时钟源多路复用
- 用于时钟输出级的整数分频器

11.4 简介

CGU 为 LPC43xx 的内核和外围设备模块生成多个独立的时钟。每个独立时钟称为基础时钟，其本身是两个时钟控制单元 (CCU) 的输入之一，这两个控制单元控制着各个外围设备的分支时钟 （参见第 12 章）。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 20. CGU 和 CCU1/2 框图

CGU 从多个时钟源中选择输入到时钟发生器的时钟，控制时钟产生并通过时钟源总线将时钟发生器的输出路由到输出级。每个输出级提供独立的时钟源，并与 LPC43xx 的基础时钟之一相对应。请参见表 54 和表 56 了解各个基础时钟的描述以及各个基础时钟可能的时钟源。

CGU 包含四种类型的时钟发生器：

1. 外部时钟输入和内部时钟：外部时钟输入指以太网PHY时钟和通用输入时钟GP_CLKIN。内部振荡器的时钟是指 IRC 和 32 kHz 振荡器输出时钟。这些时钟发生器不从时钟源总线中选择输入，而是对每个时钟源总线提供一个时钟输出。
2. 晶体振荡器：晶体振荡器由 CGU 控制。晶体振荡器的输入是 XTAL 引脚。晶体振荡器创建一个到时钟源总线的输出。
3. PLL：PLL0USB0、 PLL0AUDIO 和 PLL1 由 CGU 控制。每个 PLL 都可以从时钟源总线中选择一个输入，并向时钟源总线提供一个输出。PLL 的输入可以从所有外部和内部时钟和振荡器、所有其他 PLL 以及任何整数分频器的输出中选择 （参见表 55）。一个 PLL0 不能选择其他 PLL0 作为输入。
4. 整数分频器：五个整数分频器中，每个都可以从时钟源总线中选择一个输入并向时钟源总线创建一个分频的输出时钟。所有整数分频器的输入都可以从所有外部和内部时钟及振荡器以及全部三个 PLL 中选择。此外，第一个整数分频器的输出还可以选作所有其他整数分频器的输入 （参见表 55）。
   - 整数分频器 A：最大分频系数 = 4 （参见表 72）。
   - 整数分频器 B、 C、 D：最大分频系数 = 16 （参见表 73）。
   - 整数分频器 E：最大分频系数 = 256 （参见表 74）。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

输出级为每个基础时钟从时钟源总线中选择时钟源（参见表 56）。除 WWDT (BASE_SAFE_CLK) 和 USB0 (BASE_USB0_CLK) 的基础时钟之外，每个输出级的时钟源可以直接是任何外部和内部时钟和振荡器或是 PLL 输出或整数分频器的任何输出之一。

表 54. CGU0 基础时钟

编号	名称	频率 [1]	描述
0	BASE_SAFE_CLK	12 MHz	WDT 的基础安全时钟（始终打开）
1	BASE_USB0_CLK	480 MHz	USB0 的基础时钟
2	BASE_PERIPH_CLK	204 MHz	SGPIO 外围设备的基础时钟
3	BASE_USB1_CLK	204 MHz	USB1 的基础时钟
4	BASE_M4_CLK	204 MHz	ARM Cortex-M4 内核和 APB 外设模块 #0 和 #2 的系统基础时钟
5	BASE_SPIFI_CLK	204 MHz	SPIFI 的基础时钟
6	BASE_SPI_CLK	204 MHz	SPI 的基础时钟
7	BASE_PHY_RX_CLK	75 MHz	以太网 PHY 接收时钟的基础时钟
8	BASE_PHY_TX_CLK	75 MHz	以太网 PHY 发送时钟的基础时钟
9	BASE_APB1_CLK	204 MHz	APB 外设块 # 1 的基础时钟
10	BASE_APB3_CLK	204 MHz	APB 外设块 # 3 的基础时钟
11	BASE_LCD_CLK	204 MHz	LCD 的基础时钟
12	BASE_VADC_CLK	204 MHz	VADC 的基础时钟
13	BASE_SDIO_CLK	204 MHz	SD/MMC 的基础时钟
14	BASE_SSP0_CLK	204 MHz	SSP0 的基础时钟
15	BASE_SSP1_CLK	204 MHz	SSP1 的基础时钟
16	BASE_UART0_CLK	204 MHz	UART0 的基础时钟
17	BASE_UART1_CLK	204 MHz	UART1 的基础时钟
18	BASE_UART2_CLK	204 MHz	UART2 的基础时钟
19	BASE_UART3_CLK	204 MHz	UART3 的基础时钟
20	BASE_OUT_CLK	204 MHz	CLKOUT 引脚的基础时钟
24-21	-	-	保留
25	BASE_APLL_CLK	204 MHz	音频 PLL 的基础时钟
26	BASE_CGU_OUT0_CLK	204 MHz	CGU_OUT0 时钟输出的基础时钟
27	BASE_CGU_OUT1_CLK	204 MHz	CGU_OUT1 时钟输出的基础时钟

[1] 保证 LPC43xx 稳定操作的最大频率。

表 55 显示了每个时钟发生器可用的所有输入时钟源。

表 55. 带可选择输入的时钟发生器的时钟源

时钟源	时钟发生器							
	PLL0 USB	PLL0 AUDIO	PLL1	IDIVA /4	IDIVB /16	IDIVC /16	IDIVD /16	IDIVE /256
32 kHz 振荡器	是	是	是	是	是	是	是	是
IRC 12 MHz	是	是	是	是	是	是	是	是
ENET_RX_CLK	是	是	是	是	是	是	是	是
ENET_TX_CLK	是	是	是	是	是	是	是	是
GP_CLKIN	是	是	是	是	是	是	是	是

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 55. 带可选择输入的时钟发生器的时钟源 *（续）*

时钟源	时钟发生器 PLL0 USB	PLL0 AUDIO	PLL1	IDIVA /4	IDIVB /16	IDIVC /16	IDIVD /16	IDIVE /256
晶体振荡器	是	是	是	是	是	是	是	是
PLL0USB	否	否	是	是	否	否	否	否
PLL0AUDIO	否	否	是	是	是	是	是	是
PLL1	是	是	否	是	是	是	是	是
IDIVA	是	是	是	否	是	是	是	是
IDIVB	是	是	是	否	否	否	否	否
IDIVC	是	是	是	否	否	否	否	否
IDIVD	是	是	是	否	否	否	否	否
IDIVE	是	是	是	否	否	否	否	否

表 56. 输出级的时钟源

时钟源	输出级 （d = 默认时钟源， y = 是 （可用时钟源）， n = 否 （不可用时钟源））																							
	BASE_SAFE_CLK	BASE_USB0_CLK	BASE_PERIPH_CLK	BASE_USB1_CLK	BASE_M4_CLK	BASE_SPIFI_CLK	BASE_SPI_CLK	BASE_PHY_RX_CLK	BASE_PHY_TX_CLK	BASE_APB1_CLK	BASE_APB3_CLK	BASE_LCD_CLK	BASE_VADC_CLK	BASE_SDIO_CLK	BASE_SSP0_CLK	BASE_SSP1_CLK	BASE_UART0_CLK	BASE_UART1_CLK	BASE_UART2_CLK	BASE_UART3_CLK	BASE_OUT_CLK	BASE_APLL_CLK	BASE_CGU_OUT0_CLK	BASE_CGU_OUT1_CLK
32 kHz振荡器	n	n	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
IRC 12 MHz	d	n	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d
ENET_ RX_CLK	n	n	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
ENET_ TX_CLK	n	n	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
GP_ CLKIN	n	n	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
晶体振荡器	n	n	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
PLL0 USB	n	d	n	y	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	y	n	y	y
PLL0 AUDIO	n	n	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
PLL1	n	n	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
IDIVA	n	n	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
IDIVB	n	n	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
IDIVC	n	n	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
IDIVD	n	n	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
IDIVE	n	n	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

图 21. CGU 框图

11.5 引脚描述

表 57. CGU 引脚描述

引脚名称 / 功能名称	方向	描述
XTAL1	I	晶体振荡器输入
XTAL2	O	晶体振荡器输出
RTCX1	I	RTC 32 kHz 振荡器输入
RTCX2	O	RTC 32 kHz 振荡器输出
GP_CLKIN	I	通用输入时钟
ENET_TX_CLK	I	以太网 PHY 发送时钟
ENET_RX_CLK	I	以太网 PHY 接收时钟
CLKOUT	O	时钟输出引脚
CGU_OUT0	O	CGU 备用输出 0
CGU_OUT1	O	CGU 备用输出 1

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

11.6 寄存器描述

表 58 所示为 CGU 的寄存器地址。

注：CGU 在复位以及从深度掉电模式中唤醒时由启动引导程序配置，以使用 PLL1 生成 72 MHz 的时钟。请注意，表 58 中给定的复位值不反映此配置。

表 58. 寄存器简介：CGU （基址 0x4005 0000）

名称	访问类型	地址偏移	描述	复位值	EMC、UART0/3 启动后的复位值	USB0/1 启动后的复位值	参考
-	R	0x000	保留	-	-	-	-
-	R	0x004	保留	-	-	-	-
-	R	0x008	保留	-	-	-	-
-	R	0x00C	保留	-	-	-	-
-	-	0x010	保留	-			-
FREQ_MON	R/W	0x014	频率监控寄存器	0	0	0	表 59
XTAL_OSC_CTRL	R/W	0x018	晶体振荡器控制寄存器	0x0000 0005	0x0100 0000	0 (USB0)	表 60
PLL0USB_STAT	R	0x01C	PLL0USB 状态寄存器	0x0100 0000	0x0100 0000	0x1 (USB0)	表 61
PLL0USB_CTRL	R/W	0x020	PLL0USB 控制寄存器	0x0100 0003	0x0100 0000	0x0600 0818 (USB0)	表 62
PLL0USB_MDIV	R/W	0x024	PLL0USB M 分频器寄存器	0x05F8 5B6A	0x0100 0000	0x0196 7FFA (USB0)	表 63
PLL0USB_NP_DIV	R/W	0x028	PLL0USB N/P 分频器寄存器	0x000B 1002	0x0100 0000	0x0030 2062 (USB0)	表 64
PLL0AUDIO_STAT	R	0x02C	PLL0AUDIO 状态寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 65
PLL0AUDIO_CTRL	R/W	0x030	PLL0AUDIO 控制寄存器	0x0100 4003	0x0100 0000	0x0100 0000	表 66
PLL0AUDIO_MDIV	R/W	0x034	PLL0AUDIO M 分频器寄存器	0x05F8 5B6A	0x0100 0000	0x0100 0000	表 67
PLL0AUDIO_NP_DIV	R/W	0x038	PLL0AUDIO N/P 分频器寄存器	0x000B 1002	0x0100 0000	0x0100 0000	表 68
PLLAUDIO_FRAC	R/W	0x03C	PLL0AUDIO 小数分频器寄存器	0x0020 0000	0x0100 0000	0x0100 0000	表 69
PLL1_STAT	R	0x040	PLL1 状态寄存器	0x0100 0000	0x1	0x1 (USB1)	表 70
PLL1_CTRL	R/W	0x044	PLL1 控制寄存器	0x0100 0003	0x0117 0880	0x0117 0880 (USB1)	表 71
IDIVA_CTRL	R/W	0x048	整数分频器 A 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 72
IDIVB_CTRL	R/W	0x04C	整数分频器 B 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 73

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 58. 寄存器简介：CGU （基址 0x4005 0000）*（续）*

名称	访问类型	地址偏移	描述	复位值	EMC、UART0/3 启动后的复位值	USB0/1 启动后的复位值	参考
IDIVC_CTRL	R/W	0x050	整数分频器 C 控制寄存器	0x0100 0000	0x0900 0808	0x0900 0808	表 73
IDIVD_CTRL	R/W	0x054	整数分频器 D 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 73
IDIVE_CTRL	R/W	0x058	整数分频器 E 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 74
BASE_SAFE_CLK	R/W	0x05C	基础时钟 BASE_SAFE_CLK 的输出级 0 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 75
BASE_USB0_CLK	R/W	0x060	基础时钟 BASE_USB0_CLK 的输出级 1 控制寄存器	0x0700 0000	0x0100 0000	0x0700 0800 (USB0)	表 76
BASE_PERIPH_CLK	R/W	0x064	基础时钟 BASE_PERIPH_CLK 的输出级 2 控制寄存器	0x0700 0000	< 待定 >	< 待定 >	表 78
BASE_USB1_CLK	R/W	0x068	基础时钟 BASE_USB1_CLK 的输出级 3 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 78
BASE_M4_CLK	R/W	0x06C	基础时钟 BASE_M4_CLK 的输出级 4 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_SPIFI_CLK	R/W	0x070	基础时钟 BASE_SPIFI_CLK 的输出级 5 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_SPI_CLK	R/W	0x074	基础时钟 BASE_SPI_CLK 的输出级 6 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_PHY_RX_CLK	R/W	0x078	基础时钟 BASE_PHY_RX_CLK 的输出级 7 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_PHY_TX_CLK	R/W	0x07C	基础时钟 BASE_PHY_TX_CLK 的输出级 8 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_APB1_CLK	R/W	0x080	基础时钟 BASE_APB1_CLK 的输出级 9 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_APB3_CLK	R/W	0x084	基础时钟 BASE_APB3_CLK 的输出级 10 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_LCD_CLK	R/W	0x088	基础时钟 BASE_LCD_CLK 的输出级 11 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_VADC_CLK	R/W	0x08C	基础时钟 BASE_VADC_CLK 的输出级 12 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_SDIO_CLK	R/W	0x090	基础时钟 BASE_SDIO_CLK 的输出级 13 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_SSP0_CLK	R/W	0x094	基础时钟 BASE_SSP0_CLK 的输出级 14 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_SSP1_CLK	R/W	0x098	基础时钟 BASE_SSP1_CLK 的输出级 15 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_UART0_CLK	R/W	0x09C	基础时钟 BASE_UART0_CLK 的输出级 16 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 58. 寄存器简介：CGU （基址 0x4005 0000）*（续）*

名称	访问类型	地址偏移	描述	复位值	EMC、UART0/3 启动后的复位值	USB0/1 启动后的复位值	参考
BASE_UART1_CLK	R/W	0x0A0	基础时钟 BASE_UART1_CLK 的输出级 17 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_UART2_CLK	R/W	0x0A4	基础时钟 BASE_UART2_CLK 的输出级 18 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_UART3_CLK	R/W	0x0A8	基础时钟 BASE_UART3_CLK 的输出级 19 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 79
BASE_OUT_CL	R/W	0x0AC	基础时钟 BASE_OUT_CLK 的输出级 20 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 80
OUTCLK_21_CTRL 至 OUTCLK_24_CTRL	R/W	0x0B0 至 0x0BC	保留输出级	-	-	-	-
BASE_APLL_CLK	R/W	0x0C0	基础时钟 BASE_APLL_CLK 的输出级 25 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 81
BASE_CGU_OUT0_CLK	R/W	0x0C4	基础时钟 BASE_CGU_OUT0_CLK 的输出级 26 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 82
BASE_CGU_OUT1_CLK	R/W	0x0C8	基础时钟 BASE_CGU_OUT1_CLK 的输出级 27 控制寄存器	0x0100 0000	0x0100 0000	0x0100 0000	表 82

11.6.1 频率监控寄存器

CGU 可报告任何工作时钟的相对频率。要测量的时钟必须由软件选择，而固定频率的 IRC 时钟频率用作参考频率。然后，由 14 位计数器计数测量的时钟在用户定义的参考时钟周期数期间发生的周期数。如果设置了 MEAS 位，测量的时钟计数器将复位为 0 并向上计数，而 9 位参考时钟计数器将加载 RCNT 中的值，然后向下计数到 0。在任一计数器到达其终结值后，两个计数器都将被禁用，然后 MEAS 位复位为 0。接下来，可读取计数器的当前值以及通过以下等式获得的选定频率：

$$fselected = \frac{FCNT}{RCNT} \times fref$$

请注意，此测量的精确度可能受若干因素影响：

1. 如果两个时钟之间的比率较大（如 100 kHz 和 1 kHz），则量化误差会比较显著，因为一个计数器会在另一个计数器仍只有很小的计数值时就已饱和。
2. 由于同步化，各计数器不会精确地在同一时间开始和停止。
3. 测量频率只能达到与参考频率相同级别的精度。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 59. FREQ_MON 寄存器 （FREQ_MON，地址 0x4005 0014）位描述

位	符号	值	描述	复位值	访问类型
8:0	RCNT		9 位参考时钟计数器值	0	R/W
22:9	FCNT		14 位选定时钟计数器值	0	R
23	MEAS		测量频率	0	R/W
		0	RCNT 和 FCNT 已禁用		
		1	频率计数器已起动		
28:24	CLK_SEL		要测量的时钟源选择。保留所有其它值。	0	R/W
		0x00	32 kHz 振荡器 （默认）		
		0x01	IRC		
		0x02	ENET_RX_CLK		
		0x03	ENET_TX_CLK		
		0x04	GP_CLKIN		
		0x05	保留		
		0x06	晶体振荡器		
		0x07	PLL0USB		
		0x08	PLL0AUDIO		
		0x09	PLL1		
		0x0A	保留		
		0x0B	保留		
		0x0C	IDIVA		
		0x0D	IDIVB		
		0x0E	IDIVC		
		0x0F	IDIVD		
		0x10	IDIVE		
31:29	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

11.6.2 晶体振荡器控制寄存器

寄存器 XTAL_OSC_CONTROL 包含晶体振荡器的控制位。

表 60. XTAL_OSC_CTRL 寄存器 （XTAL_OSC_CTRL，地址 0x4005 0018）位描述

位	符号	值	描述	复位值	访问类型
0	ENABLE		使能振荡器垫。不能在一次写入操作中更改 BYPASS 和 ENABLE 位：这样将导致设备操作不稳！	1	R/W
		0	使能		
		1	掉电 （默认）		
1	BYPASS		配置晶振操作或外部时钟输入引脚XTAL1。不能在一次写入操作中更改 BYPASS 和 ENABLE 位：这样将导致设备操作不稳！	0	R/W
		0	在连接晶体的状况下操作 （默认）。		
		1	旁路模式。在使用外部时钟源代替晶体时，使用此模式。		
2	HF		选择频率范围	1	R/W
		0	振荡器低频模式（晶体或外部时钟源1至20 MHz）。在15 MHz与20 MHz之间，HF位的状态无关紧要。		
		1	振荡器高频模式；晶体或外部时钟源15至25 MHz。在15 MHz与20 MHz之间，HF位的状态无关紧要。		
31:3	-		保留	-	-

11.6.3 PLL0USB 寄存器

PLL0 向高速 USB0 接口提供专用时钟。

有关如何设置 PLL0 的说明，请参见第 11.7.4.5 节。

11.6.3.1 PLL0USB 状态寄存器

表 61. PLL0USB 状态寄存器 （PLL0USB_STAT，地址 0x4005 001C）位描述

位	符号	描述	复位值	访问类型
0	LOCK	PLL0 锁定指示器	0	R
1	FR	PLL0 自由运行指示器	0	R
31:2	-	保留		-

11.6.3.2 PLL0USB 控制寄存器

表 62. PLL0USB 状态寄存器 （PLL0USB_CTRL，地址 0x4005 0020C）位描述

位	符号	值	描述	复位值	访问类型
0	PD		PLL0 掉电	1	R/W
		0	PLL0 已使能		
		1	PLL0 已掉电		
1	BYPASS		输入时钟旁路控制	1	R/W
		0	CCO 时钟发送到后分频器。在正常操作中使用此设置。		
		1	PLL0 时钟发送到后分频器 （默认）。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 62. PLL0USB 状态寄存器 （PLL0USB_CTRL，地址 0x4005 0020C）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
2	DIRECTI		PLL0 直接输入	0	R/W
3	DIRECTO		PLL0 直接输出	0	R/W
4	CLKEN		PLL0 时钟使能	0	R/W
5	-		保留	-	-
6	FRM		自由运行模式	0	R/W
7	-		保留	0	R/W
8	-		保留。读取为零。不要向此寄存器写入 1。	0	R/W
9	-		保留。读取为零。不要向此寄存器写入 1。	0	R/W
10	-		保留。读取为零。不要向此寄存器写入 1。	0	R/W
11	AUTOBLOCK		在频率变更期间自动锁定时钟	0	R/W
		0	自动锁定已禁用。		
		1	自动锁定已使能。		
23:12	-		保留	-	-
28:24	CLK_SEL		时钟源选择。保留所有其它值。	0x01	R/W
		0x00	32 kHz 振荡器		
		0x01	IRC （默认）		
		0x02	ENET_RX_CLK		
		0x03	ENET_TX_CLK		
		0x04	GP_CLKIN		
		0x06	晶体振荡器		
		0x09	PLL1		
		0x0C	IDIVA		
		0x0D	IDIVB		
		0x0E	IDIVC		
		0x0F	IDIVD		
		0x10	IDIVE		
31:29	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

11.6.3.3 PLL0USB M 分频器寄存器

注：PLL M 分频器寄存器不会直接使用 M 的直接二进制表示。相反，它使用 M 的编码版 MDEC。 M 的有效范围是 1 至 2^15。该值编码成一个 17 位 MDEC 值。

M 和 MDEC 之间的关系通过下列伪代码表示。具体示例请参见第 11.8.3 节和第 11.8.4 节。

```
M_max=0x00008000, x=0x00004000;
   switch (M) {
       case 0:x = 0xFFFFFFFF;
       case 1:x = 0x00018003;
       case 2:x = 0x00010003;

       default:for (i = M; i <= M_max; i++)
                 x = (((x ^ (x>>1)) & 1) << 14) | ((x>>1) & 0x3FFF); }
   MDEC[16:0] = x;
```

SELP、 SELI 和 SELR 的值取决于 M 的值，由以下伪代码表示：

```
if (M < 60) then
     SELP = (M>>1) + 1
   else
     SELP = 31;
if (M > 16384) then
     SELI = 1
   else if (M > 8192) then
     SELI = 2
   else if (M > 2048) then
     SELI = 4
   else if (M >= 501) then
     SELI = 8
   else if (M >= 60) then
     SELI = 4*(1024/(M+9))
   else
     SELI = (M & 0x3C) + 4; /* & denotes bitwise AND */
   SELR = 0;
```

表 63. PLL0USB M 分频器寄存器 （PLL0USB_MDIV，地址 0x4005 0024）位描述

位	符号	描述	复位值	访问类型
16:0	MDEC	解码的 M 分频器系数值。为 M 分频器选择介于 1 和 131071 之间的值。	0x05F85B6A	R/W
21:17	SELP	带宽选择 P 值	11100	R/W
27:22	SELI	带宽选择 I 值	010111	R/W
31:28	SELR	带宽选择 R 值	0000	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

11.6.3.4 PLL0USB NP 分频器寄存器

注：PLL NP 分频器寄存器不会直接使用 N 和 P 的直接二进制表示。相反，它将分别使用 N 和 P 的编码版 NDEC 和 PDEC。

- N的有效范围是1至2^8。该值编码成一个10位NDEC值。关系可通过下列伪代码表示：

```
N_max=0x00000100, x=0x00000080;
    switch (N) {
        case 0:x = 0xFFFFFFFF;
        case 1:x = 0x00000302;
        case 2:x = 0x00000202;

        default:for (i = N; i <= N_max; i++)
                    x = (((x ^ (x>>2) ^ (x>>3) ^ (x>>4)) & 1) << 7) | ((x>>1) & 0x7F); }
    NENC[9:0] = x;
```

- P 的有效范围是 1 至 2^5。该值编码成一个 7 位 PDEC 值。关系可通过下列伪代码表示：

```
P_max=0x00000200, x=0x00000010;
    switch (P) {
        case 0:x = 0xFFFFFFFF;
        case 1:x = 0x00000062;
        case 2:x = 0x00000042;

        default:for (i = P; i <= P_max; i++)
                    x = (((x ^ (x>>2)) & 1) << 4) | ((x>>1) & 0xF); }
    PDEC[6:0] = x;
```

具体示例请参见第 11.8.3 节和第 11.8.4 节。

表 64. PLL0USB NP 分频器寄存器 （PLL0USB_NP_DIV，地址 0x4005 0028）位描述

位	符号	描述	复位值	访问类型
6:0	PDEC	解码的 P 分频器系数值	000 0010	R/W
11:7	-	保留	-	-
21:12	NDEC	解码的 N 分频器系数值	1011 0001	R/W
31:22	-	保留	-	-

11.6.4 PLL0AUDIO 寄存器

PLL0AUDIO 为音频应用提供广泛的频率，并可连接至多个基础时钟。PLL0AUDIO 可搭配或不搭配小数分频器使用。

有关如何设置 PLL0 的说明，请参见第 11.7.4.5 节。

11.6.4.1 PLL0AUDIO 状态寄存器

表 65. PLL0AUDIO 状态寄存器 （PLL0AUDIO_STAT，地址 0x4005 002C）位描述

位	符号	描述	复位值	访问类型
0	LOCK	PLL0 锁定指示器	0	R
1	FR	PLL0 自由运行指示器	0	R
31:2	-	保留		-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

11.6.4.2 PLL0AUDIO 控制寄存器

表 66. PLL0AUDIO 状态寄存器 （PLL0AUDIO_CTRL，地址 0x4005 0030C）位描述

位	符号	值	描述	复位值	访问类型
0	PD		PLL0 掉电	1	R/W
		0	PLL0 已使能		
		1	PLL0 已掉电		
1	BYPASS		输入时钟旁路控制	1	R/W
		0	CCO 时钟发送到后分频器。在正常操作中使用此设置。		
		1	PLL0 时钟发送到后分频器 （默认）。		
2	DIRECTI		PLL0 直接输入	0	R/W
3	DIRECTO		PLL0 直接输出	0	R/W
4	CLKEN		PLL0 时钟使能	0	R/W
5	-		保留	-	-
6	FRM		自由运行模式	0	R/W
7	-		保留	0	R/W
8	-		保留。读取为零。不要向此寄存器写入 1。	0	R/W
9	-		保留。读取为零。不要向此寄存器写入 1。	0	R/W
10	-		保留。读取为零。不要向此寄存器写入 1。	0	R/W
11	AUTOBLOCK		在频率变更期间自动锁定时钟	0	R/W
		0	自动锁定已禁用。		
		1	自动锁定已使能。		
12	PLLFRAQ_REQ		小数 PLL 字写入请求。在 SEL_EXT 位使能小数分频器时，将此位设为 1。	0	R/W
13	SEL_EXT		选择小数分频器。	0	R/W
		0	使能小数分频器。		
		1	MDEC 已使能。未使用小数分频器。		
14	MOD_PD		Σ - Δ 调制器掉电	1	R/W
		0	Σ - Δ 调制器已使能		
		1	Σ - Δ 调制器已掉电		
23:15	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 66. PLL0AUDIO 状态寄存器 （PLL0AUDIO_CTRL，地址 0x4005 0030C）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
28:24	CLK_SEL		时钟源选择。保留所有其它值。	0x01	R/W
		0x00	32 kHz 振荡器		
		0x01	IRC （默认）		
		0x02	ENET_RX_CLK		
		0x03	ENET_TX_CLK		
		0x04	GP_CLKIN		
		0x06	晶体振荡器		
		0x09	PLL1		
		0x0C	IDIVA		
		0x0D	IDIVB		
		0x0E	IDIVC		
		0x0F	IDIVD		
		0x10	IDIVE		
31:29	-		保留	-	-

11.6.4.3 PLL0AUDIO M 分频器寄存器

未在小数模式中使用 PLL0AUDIO 时，可直接设置 PLL0AUDIO M 分频器寄存器。如果小数分频器已使能 （参见表 69），则小数分频器会设置 MDEC 值。

注：PLL M 分频器寄存器不会直接使用 M 的直接二进制表示。相反，它使用 M 的编码版 MDEC。 M 的有效范围是 1 至 2^15。该值编码成一个 17 位 MDEC 值。

M 和 MDEC 之间的关系通过下列伪代码表示。具体示例请参见第 11.8.3 节和第 11.8.4 节。

```
M_max=0x00008000, x=0x00004000;
   switch (M) {
       case 0:x = 0xFFFFFFFF;
       case 1:x = 0x00018003;
       case 2:x = 0x00010003;

       default:for (i = M; i <= M_max; i++)
                   x = (((x ^ (x>>1)) & 1) << 14) | ((x>>1) & 0x3FFF); }
   MDEC[16:0] = x;
```

注：SELP、 SELI 和 SELR 的值由编码模块生成，无需显性编程。

表 67. PLL0AUDIO M 分频器寄存器 （PLL0AUDIO_MDIV，地址 0x4005 0034）位描述

位	符号	描述	复位值	访问类型
16:0	MDEC	解码的 M 分频器系数值。为 M 分频器选择介于 1 和 131071 之间的值。	0x05F85B6A	R/W
31:17	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

11.6.4.4 PLL0AUDIO NP 分频器寄存器

注：PLL NP 分频器寄存器不会直接使用 N 和 P 的直接二进制表示。相反，它将分别使用 N 和 P 的编码版 NDEC 和 PDEC。

- N的有效范围是1至2^8。该值编码成一个10位NDEC值。关系可通过下列伪代码表示：

```
N_max=0x00000100, x=0x00000080;
    switch (N) {
        case 0:x = 0xFFFFFFFF;
        case 1:x = 0x00000302;
        case 2:x = 0x00000202;

        default:for (i = N; i <= N_max; i++)
                    x = (((x ^ (x>>2) ^ (x>>3) ^ (x>>4)) & 1) << 7) | ((x>>1) & 0x7F); }
    NENC[9:0] = x;
```

- P 的有效范围是 1 至 2^5。该值编码成一个 7 位 PDEC 值。关系可通过下列伪代码表示：

```
P_max=0x00000200, x=0x00000010;
    switch (P) {
        case 0:x = 0xFFFFFFFF;
        case 1:x = 0x00000062;
        case 2:x = 0x00000042;

        default:for (i = P; i <= P_max; i++)
                    x = (((x ^ (x>>2)) & 1) << 4) | ((x>>1) & 0xF); }
    PDEC[6:0] = x;
```

表 68. PLL0 AUDIO NP 分频器寄存器 （PLL0AUDIO_NP_DIV，地址 0x4005 0038）位描述

位	符号	描述	复位值	访问类型
6:0	PDEC	解码的 P 分频器系数值	000 0010	R/W
11:7	-	保留	-	-
21:12	NDEC	解码的 N 分频器系数值	1011 0001	R/W
31:22	-	保留	-	-

11.6.4.5 PLL0AUDIO 小数分频器寄存器

小数分频器激活后，sigma-delta 调制器模块生成正确比例的分频器值 M 和 M+1，从而实现 M+K/L 的平均分频比，其中 0<=K<=L，且 M、K 和 L 为整数值。M 由 PLLFRACT_CTRL 寄存器 (PLLFRACT[21:15]) 的整数部分决定，K 由 PLLFRACT_CTRL 寄存器 (PLLFRACT[14:0]) 的小数部分决定。显示成到 M 分频器的输入前，要将连续 M 和 M+1 值进一步编码成适当的 MDEC 值。

表 69. PLL0AUDIO 小数分频器寄存器 （PLL0AUDIO_FRAC，地址 0x4005 003C）位描述

位	符号	描述	复位值	访问类型
21:0	PLLFRACT_CTRL	PLL 小数分频器控制字	000 0000	R/W
31:22	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

11.6.5 PLL1 寄存器

PLL1 用于内核及所有外设块。

11.6.5.1 PLL1 状态寄存器

表 70. PLL1 状态寄存器 （PLL1_STAT，地址 0x4005 0040）位描述

位	符号	描述	复位值	访问类型
0	LOCK	PLL1 锁定指示器	0	R
31:1	-	保留	-	-

11.6.5.2 PLL1 控制寄存器

表 71. PLL1_CTRL 寄存器 （PLL1_CTRL，地址 0x4005 0044）位描述

位	符号	值	描述	复位值	访问类型
0	PD		PLL1 掉电	1	R/W
		0	PLL1 已使能		
		1	PLL1 已掉电		
1	BYPASS		输入时钟旁路控制	1	R/W
		0	CCO 时钟发送到后分频器。用于正常操作。		
		1	PLL1 时钟发送到后分频器 （默认）。		
2	-		保留。不要向此位写入 1。	0	R/W
5:3	-		保留。不要向这些位写入 1。	-	-
6	FBSEL		PLL 反馈选择 （参见 图 24. PLL1 框图）。	0	R/W
		0	CCO 输出用作反馈分频器输入时钟。		
		1	PLL 输出时钟 (clkout) 用作反馈分频器输入时钟。用于正常操作。		
7	DIRECT		PLL 直接 CCO 输出	0	R/W
		0	已禁用		
		1	已使能		
9:8	PSEL[		后分频器分频比 P。应用的值为 2xP。	01	R/W
		0x0	1		
		0x1	2 （默认值）		
		0x2	4		
		0x3	8		
10	-		保留	-	-
11	AUTOBLOCK		在频率变更期间自动锁定时钟	0	R/W
		0	自动锁定已禁用。		
		1	自动锁定已使能。		
13:12	NSEL		预分频器分频比	10	R/W
		0x0	1		
		0x1	2		
		0x2	3 （默认值）		
		0x3	4		
15:14	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 71. PLL1_CTRL 寄存器 （PLL1_CTRL，地址 0x4005 0044）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
23:16	MSEL		反馈分频器标度比 (M) 00000000 = 1 00000001 = 2 ... 11111111 = 256	11000	R/W
28:24	CLK_SEL		时钟源选择。	0x01	R/W
		0x00	32 kHz 振荡器		
		0x01	IRC （默认）		
		0x02	ENET_RX_CLK		
		0x03	ENET_TX_CLK		
		0x04	GP_CLKIN		
		0x05	保留		
		0x06	晶体振荡器		
		0x07	PLL0USB		
		0x08	PLL0AUDIO		
		0x09	保留		
		0x0A	保留		
		0x0C	IDIVA		
		0x0D	IDIVB		
		0x0E	IDIVC		
		0x0F	IDIVD		
		0x10	IDIVE		
31:29	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

11.6.6 整数分频器寄存器 A

表 72. IDIVA 状态寄存器 （IDIVA_CTRL，地址 0x4005 0048）位描述

位	符号	值	描述	复位值	访问类型
0	PD		整数分频器 A 掉电	0	R/W
		0	IDIVA 已使能 （默认）		
		1	掉电		
1	-		保留	-	-
3:2	IDIV		整数分频器 A 分频器值 (1/(IDIV + 1))	00	R/W
		0x0	1 （默认值）		
		0x1	2		
		0x2	3		
		0x3	4		
10:4	-		保留	-	-
11	AUTOBLOCK		在频率变更期间自动锁定时钟	0	R/W
		0	自动锁定已禁用。		
		1	自动锁定已使能。		
23:12	-		保留	-	-
28:24	CLK_SEL		时钟源选择。保留所有其它值。	0x01	R/W
		0x00	32 kHz 振荡器		
		0x01	IRC （默认）		
		0x02	ENET_RX_CLK		
		0x03	ENET_TX_CLK		
		0x04	GP_CLKIN		
		0x06	晶体振荡器		
		0x07	PLL0USB		
		0x08	PLL0AUDIO		
		0x09	PLL1		
31:29	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

11.6.7 整数分频器寄存器 B、 C、 D

表 73. IDIVB/C/D 控制寄存器（IDIVB_CTRL，地址 0x4005 004C ； IDIVC_CTRL，地址 0x4005 0050 ； IDIVC_CTRL，地址 0x4005 0054）位描述

位	符号	值	描述	复位值	访问类型
0	PD		整数分频器掉电	0	R/W
		0	IDIV 已使能 （默认）		
		1	掉电		
1	-		保留	-	-
5:2	IDIV		整数分频器 B、C、D 分频器值 (1/(IDIV + 1)) 0000 = 1 （默认） 0001 = 2 ... 1111 = 16	0000	R/W
10:6	-		保留	-	-
11	AUTOBLOCK		在频率变更期间自动锁定时钟	0	R/W
		0	自动锁定已禁用。		
		1	自动锁定已使能。		
23:12	-		保留	-	-
28:24	CLK_SEL		时钟源选择。保留所有其它值。	0x01	R/W
		0x00	32 kHz 振荡器		
		0x01	IRC （默认）		
		0x02	ENET_RX_CLK		
		0x03	ENET_TX_CLK		
		0x04	GP_CLKIN		
		0x06	晶体振荡器		
		0x08	PLL0AUDIO		
		0x09	PLL1		
		0x0C	IDIVA		
31:29	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

11.6.8 整数分频器寄存器 E

表 74. IDIVE 状态寄存器 （IDIVE_CTRL，地址 0x4005 0058）位描述

位	符号	值	描述	复位值	访问类型
0	PD		整数分频器掉电	0	R/W
		0	IDIV 已使能 （默认）		
		1	掉电		
1	-		保留	-	-
9:2	IDIV		整数分频器 E 分频器值 (1/(IDIV + 1)) 00000000 = 1 （默认） 00000001 = 2 ... 111111111 = 256	00000000	R/W
10	-		保留	-	-
11	AUTOBLOCK		在频率变更期间自动锁定时钟	0	R/W
		0	自动锁定已禁用。		
		1	自动锁定已使能。		
23:12	-		保留	-	-
28:24	CLK_SEL		时钟源选择。保留所有其它值。	0x01	R/W
		0x00	32 kHz 振荡器		
		0x01	IRC （默认）		
		0x02	ENET_RX_CLK		
		0x03	ENET_TX_CLK		
		0x04	GP_CLKIN		
		0x06	晶体振荡器		
		0x08	PLL0AUDIO		
		0x09	PLL1		
		0x0C	IDIVA		
31:29	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

11.6.9 BASE_SAFE_CLK 控制寄存器

此寄存器控制看门狗振荡器的 BASE_SAFE_CLK。此基础时钟唯一可能的时钟源是 IRC。

表 75. BASE_SAFE_CLK 控制寄存器 （BASE_SAFE_CLK - 地址 0x4005 005C）位描述

位	符号	值	描述	复位值	访问类型
0	PD		输出级掉电	0	R/W
		0	输出级已使能 （默认）		
		1	掉电		
10:1	-		保留	-	-
11	AUTOBLOCK		在频率变更期间自动锁定时钟	0	R/W
		0	自动锁定已禁用。		
		1	自动锁定已使能。		
23:12	-		保留	-	-
28:24	CLK_SEL		时钟源选择。保留所有其它值。	0x01	R/W
		0x01	IRC （默认）		
31:29	-		保留	-	-

11.6.10 BASE_USB0_CLK 控制寄存器

此寄存器控制高速USB0的BASE_USB0_CLK。此基础时钟唯一可能的时钟源是PLL0USB输出。

表 76. BASE_USB0_CLK 控制寄存器 （BASE_USB0_CLK - 地址 0x4005 0060）位描述

位	符号	值	描述	复位值	访问类型
0	PD		输出级掉电	0	R/W
		0	输出级已使能 （默认）		
		1	掉电		
10:1	-		保留	-	-
11	AUTOBLOCK		在频率变更期间自动锁定时钟	0	R/W
		0	自动锁定已禁用。		
		1	自动锁定已使能。		
23:12	-		保留	-	-
28:24	CLK_SEL		时钟源选择。	0x07	R/W
		0x07	PLL0USB （默认）		
31:29	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

11.6.11 BASE_PERIPH_CLK 控制寄存器

该寄存器控制 SGPIO 模块的基础时钟 2。

表 77. BASE_PERIPH_CLK 控制寄存器 （BASE_PERIPH_CLK - 地址 0x4005 0064）位描述

位	符号	值	描述	复位值	访问类型
0	PD		输出级掉电	0	R/W
		0	输出级已使能 （默认）		
		1	掉电		
10:1	-		保留	-	-
11	AUTOBLOCK		在频率变更期间自动锁定时钟	0	R/W
		0	自动锁定已禁用。		
		1	自动锁定已使能。		
23:12	-		保留	-	-
28:24	CLK_SEL		时钟源选择。保留所有其它值。	0x01	R/W
		0x00	32 kHz 振荡器		
		0x01	IRC （默认）		
		0x02	ENET_RX_CLK		
		0x03	ENET_TX_CLK		
		0x04	GP_CLKIN		
		0x06	晶体振荡器		
		0x08	PLL0AUDIO		
		0x09	PLL1		
		0x0C	IDIVA		
		0x0D	IDIVB		
		0x0E	IDIVC		
		0x0F	IDIVD		
		0x10	IDIVE		
31:29	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

11.6.12 BASE_USB1_CLK 控制寄存器

这些寄存器控制基础时钟 3 (USB1)。

表 78. BASE_USB1_CLK 控制寄存器 （BASE_USB1_CLK - 地址 0x4005 0068）位描述

位	符号	值	描述	复位值	访问类型
0	PD		输出级掉电	0	R/W
		0	输出级已使能 （默认）		
		1	掉电		
10:1	-		保留	-	-
11	AUTOBLOCK		在频率变更期间自动锁定时钟	0	R/W
		0	自动锁定已禁用。		
		1	自动锁定已使能。		
23:12	-		保留	-	-
28:24	CLK_SEL		时钟源选择。保留所有其它值。	0x01	R/W
		0x00	32 kHz 振荡器		
		0x01	IRC （默认）		
		0x02	ENET_RX_CLK		
		0x03	ENET_TX_CLK		
		0x04	GP_CLKIN		
		0x06	晶体振荡器		
		0x07	PLL0USB		
		0x08	PLL0AUDIO		
		0x09	PLL1		
		0x0C	IDIVA		
		0x0D	IDIVB		
		0x0E	IDIVC		
		0x0F	IDIVD		
		0x10	IDIVE		
31:29	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

11.6.13 BASE_M4_CLK 至 BASE_UART3_CLK 控制寄存器

这些寄存器控制基础时钟 4 至 19。

表 79. BASE_M4_CLK 至 BASE_UART3_CLK 控制寄存器 （BASE_M4_CLK 至 BASE_UART3_CLK，地址 0x4005 006C 至 0x4005 00A8）位描述

位	符号	值	描述	复位值	访问类型
0	PD		输出级掉电	0	R/W
		0	输出级已使能 （默认）		
		1	掉电		
10:1	-		保留	-	-
11	AUTOBLOCK		在频率变更期间自动锁定时钟	0	R/W
		0	自动锁定已禁用。		
		1	自动锁定已使能。		
23:12	-		保留	-	-
28:24	CLK_SEL		时钟源选择。保留所有其它值。	0x01	R/W
		0x00	32 kHz 振荡器		
		0x01	IRC （默认）		
		0x02	ENET_RX_CLK		
		0x03	ENET_TX_CLK		
		0x04	GP_CLKIN		
		0x06	晶体振荡器		
		0x08	PLL0AUDIO		
		0x09	PLL1		
		0x0C	IDIVA		
		0x0D	IDIVB		
		0x0E	IDIVC		
		0x0F	IDIVD		
		0x10	IDIVE		
31:29	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

11.6.14 BASE_OUT_CLK 寄存器

此寄存器控制 CLKOUT 引脚的时钟输出。所有时钟发生器输出都可以通过此引脚监控。

表 80. BASE_OUT_CLK 控制寄存器 （BASE_OUT_CLK，地址 0x4005 00AC）位描述

位	符号	值	描述	复位值	访问类型
0	PD		输出级掉电	0	R/W
		0	输出级已使能 （默认）		
		1	掉电		
10:1	-		保留	-	-
11	AUTOBLOCK		在频率变更期间自动锁定时钟	0	R/W
		0	自动锁定已禁用。		
		1	自动锁定已使能。		
23:12	-		保留	-	-
28:24	CLK_SEL		时钟源选择。	0x01	R/W
		0x00	32 kHz 振荡器		
		0x01	IRC （默认）		
		0x02	ENET_RX_CLK		
		0x03	ENET_TX_CLK		
		0x04	GP_CLKIN		
		0x05	保留		
		0x06	晶体振荡器		
		0x07	PLL0USB		
		0x08	PLL0AUDIO		
		0x09	PLL1		
		0x0C	IDIVA		
		0x0D	IDIVB		
		0x0E	IDIVC		
		0x0F	IDIVD		
		0x10	IDIVE		
31:29	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

11.6.15 BASE_APLL_CLK 寄存器

此寄存器控制 <tbd> 的时钟输出。

表 81. BASE_APLL_CLK 控制寄存器 （BASE_APLL_CLK - 地址 0x4005 00C0）位描述

位	符号	值	描述	复位值	访问类型
0	PD		输出级掉电	0	R/W
		0	输出级已使能 （默认）		
		1	掉电		
10:1	-		保留	-	-
11	AUTOBLOCK		在频率变更期间自动锁定时钟	0	R/W
		0	自动锁定已禁用。		
		1	自动锁定已使能。		
23:12	-		保留	-	-
28:24	CLK_SEL		时钟源选择。	0x01	R/W
		0x00	32 kHz 振荡器		
		0x01	IRC （默认）		
		0x02	ENET_RX_CLK		
		0x03	ENET_TX_CLK		
		0x04	GP_CLKIN		
		0x05	保留		
		0x06	晶体振荡器		
		0x07	保留		
		0x08	PLL0AUDIO		
		0x09	PLL1		
		0x0C	IDIVA		
		0x0D	IDIVB		
		0x0E	IDIVC		
		0x0F	IDIVD		
		0x10	IDIVE		
31:29	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

11.6.16 BASE_CGU_OUT0_CLK 至 BASE_CGU_OUT1_CLK 寄存器

此寄存器控制备用 CGU 输出引脚 CGU_OUT0 和 CGU_OUT1 的时钟输出。所有时钟发生器输出都可以通过此引脚监控。

表 82. BASE_CGU_OUT0_CLK 至 BASE_CGU_OUT1_CLK 控制寄存器（BASE_CGU_OUT0_CLK 至 BASE_CGU_OUT1_CLK，地址 0x4005 00C4 至 0x4005 00C8）位描述

位	符号	值	描述	复位值	访问类型
0	PD		输出级掉电	0	R/W
		0	输出级已使能 （默认）		
		1	掉电		
10:1	-		保留	-	-
11	AUTOBLOCK		在频率变更期间自动锁定时钟	0	R/W
		0	自动锁定已禁用。		
		1	自动锁定已使能。		
23:12	-		保留	-	-
28:24	CLK_SEL		时钟源选择。	0x01	R/W
		0x00	32 kHz 振荡器		
		0x01	IRC （默认）		
		0x02	ENET_RX_CLK		
		0x03	ENET_TX_CLK		
		0x04	GP_CLKIN		
		0x05	保留		
		0x06	晶体振荡器		
		0x07	PLL0USB		
		0x08	PLL0AUDIO		
		0x09	PLL1		
		0x0C	IDIVA		
		0x0D	IDIVB		
		0x0E	IDIVC		
		0x0F	IDIVD		
		0x10	IDIVE		
31:29	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

11.7 功能描述

11.7.1 32 kHz 振荡器

32 kHz 振荡器输出由 CREG 模块控制 （参见表 38）。 RTC 和报警定时器直接连接到 32 kHz 振荡器。

11.7.2 IRC

IRC 调整为 12 MHz 内部振荡器。尽管它属于 CGU 的一部分，但 CGU 不对此时钟源进行控制。 IRC 根据掉电模式的设置进入掉电模式。

11.7.3 晶体振荡器

晶体振荡器由 CGU 中的 XTAL_OSC_CTRL 寄存器控制 （参见表 60）。

11.7.4 PLL0 （PLL0USB 和 PLL0AUDIO）

PLL0USB 和 PLL0AUDIO 的 PLL 模块相同。PLL0AUDIO 支持额外的小数分频器，为音频应用获取更多更高精度的 PLL 频率。

11.7.4.1 功能要点

- 输入频率：14 kHz 至 150 MHz。来自外部晶体的输入上限为 25 MHz。
- CCO 频率：275 MHz 至 550 MHz。
- 输出时钟范围：4.3 MHz 至 550 MHz。
- 可编程分频器：
  - 预分频器 N （N， 1 至 2^8）
  - 反馈分频器 2 x M （M， 1 至 2^{15}）
  - 后分频器 P x 2 （P， 1 至 2^5）。
- 可编程带宽 （集成操作、正比操作、高频极）。
- 时钟可以随时调整 （带信号握手控制的分频器）。
- 正沿时钟。
- 用于避免 PLL 挂起的频率限制器。
- 锁定探测器。
- 掉电模式。
- 自由运行模式

注：两个 PLL0 模块功能完全相同。音频应用的 PLL0 （用于音频的 PLL0）支持额外的小数分频器级 （参见第 11.7.5 节）。

11.7.4.2 PLL0 描述

此 PLL 的框图如图 22 所示。时钟输入必须馈入引脚 clkin。引脚 clkout 是 PLL 时钟输出。PLL 的模拟部分包括相位频率检测器 (PFD)、滤波器和电流控制振荡器 (CCO)。 PFD 有两种输入，来自（分频）外部时钟的参比输入以及来自分频 CCO 输出时钟的输入。PFD 比较这些输入信号的相位 / 频率，并在它们不匹配时产生控制信号。此控制信号将馈入滤波器，从而驱动 CCO。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 22. PLL0 框图

PLL 包含三种可编程分频器：预分频器 (N)、反馈分频器 (M) 和后分频器 (P)。 PLL 包含锁定探测器，用来测量输入时钟上升沿与反馈时钟上升沿之间的相位差异。只有当此差异小于连续七个以上输入时钟期间的所谓锁定标准时，锁定输出才会从低切换到高。单一过大的相位差异会立即复位计数器，并导致锁定信号降落（如果原来较高）。每行要求有七次相位测量值低于某个特定的数，以确保锁定探测器在输入时钟和反馈时钟两者的相位和频率极其相符时，才会指示锁定。这样，可有效防止虚假的锁定指示，由此确保锁定信号不会出错。

为避免频率中断， PLL 包含有频率限制器。此为内置功能，以防止 CCO 运行过快。例如，如果对 PLL 应用了错误的反馈分频 (M) 比率，即可能会出现这种状况。

11.7.4.3 PLL0 工作模式的使用

表 83. PLL 工作模式

	PLL0_Mode 位设置：					
模式	PD	CLKEN	BYPASS	DIRECTI	DIRECTO	FRM
1: 普通	0	1	0	1/0	1/0	0
3: 掉电	1	x	x	x	x	x

11.7.4.3.1 正常模式

模式 1 为正常工作模式。

可选择预分频器和后分频器以提供：

- 模式 1a：正常工作模式，无需后分频器和预分频器
- 模式 1b：正常工作模式，需要后分频器，无需预分频器
- 模式 1c：正常工作模式，无需后分频器，需要预分频器
- 模式 1d：正常工作模式，需要后分频器和预分频器

为了在 PLL (clkout) 的输出中获得最佳的相位噪声和抖动性能，必须在 PFD 使用最高的可能参比时钟 (clkref)。因此，建议使用模式 1a 和 1b，这时无需预分频器即能够获得合适的输出频率。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

通过在 PLL (clkout) 的输出中使用后分频器，可使分频比始终均等，因为在后分频后分频器除以 2。

表 84. HP0/1_Mode 寄存器中的 DIRECTL 和 DIRECTO 位设置

模式	DIRECTI	DIRECTO
1a	1	1
1b	1	0
1c	0	1
1d	0	0

11.7.4.3.2 模式 1a：正常工作模式，无需后分频器和预分频器

在正常工作模式 1a 中，将绕过后分频器和预分频器。工作频率为：

Fout = Fcco = 2 x M x Fin Λ (275 MHz ≤Fcco ≤550 MHz, 4 kHz ≤ Fin ≤150 MHz)

反馈分频比可进行编程：

- 反馈分频器 M （M， 1 至 2^{15}）

11.7.4.3.3 模式 1b：正常工作模式，需要后分频器，无需预分频器

在正常工作模式 1b 中，将绕过预分频器。工作频率为：

Fout = Fcco /(2 x P) = (M / P) x Fin Λ (275 MHz ≤Fcco ≤550 MHz, 4 kHz ≤Fin ≤ 150 MHz)

分频比可进行编程：

- 反馈分频器 M （M， 1 至 2^{15}）
- 后分频器 P （P、 1 至 32）

11.7.4.3.4 模式 1c：正常工作模式，无需后分频器，需要预分频器

在正常工作模式 1c 中，将绕过除以 2 分频的后分频器。工作频率为：

Fout = Fcco = 2 x M x Fin / N Λ (275 MHz ≤Fcco ≤550 MHz, 4 kHz ≤Fin/N ≤150 MHz)

分频比可进行编程：

- 预分频器 N （N、 1 至 256）
- 反馈分频器 M （M， 1 至 2^{15}）

11.7.4.3.5 模式 1d：正常工作模式，需要后分频器和预分频器

在正常工作模式 1d 中，不会绕过任何分频器。工作频率为：

Fout = Fcco /(2 x P) = M x Fin /(N x P) Λ (275 MHz ≤Fcco ≤550 MHz, 4 kHz ≤Fin/N ≤150 MHz)

分频比可进行编程：

- 预分频器 N （N、 1 至 256）
- 反馈分频器 M （M， 1 至 2^{15}）
- 后分频器 P （P、 1 至 32）

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

11.7.4.3.6 模式 3：掉电模式 (pd)

在此模式 (pd = '1') 下，将停止振荡器，降低锁定输出，并且将关闭内部电流参考。在 pd 期间，还可以在输入总线（msel、psel、nsel）上加载新的分频比。掉电模式通过使 pd 降低将结束，导致 PLL 启动。 PLL 重新获得输入时钟的锁定后，锁定信号将变高。

11.7.4.4 USB0 的设置

表 85 显示了用于配置 USB0 使用的输出频率 480 MHz 的 F_{out} 的分频器设置。

11.7.4.5 使用说明

为了设置 PLL0，请遵循以下步骤：

1. 将 PLL0_CTRL 中的位 1 设置为 1，使 PLL0 掉电。只有当前使能了 PLL0 时，才需要此步骤。
2. 配置 PLL0_M 和 PLL0_NP 寄存器中的 PLL0 m、 n 和 p 分频器值。
3. 通过将 PLL0_CTRL 寄存器中的位 1 设置为 0，给 PLL0 加电。
4. 通过监控 PLL0_STAT 寄存器中的 LOCK 位，等待 PLL0 锁定。
5. 使能 PLL0_CTRL 寄存器中的 PLL0 时钟输出。

11.7.5 PLL0AUDIO 的小数分频器

音频应用的 PLL0 (PLL0AUDIO) 包括额外的小数分频器。 PLL0AUDIO 控制寄存器的 SEL_EXT 位决定是使用 (SEL_EXT=0) 还是绕过小数分频器 (SEL_EXT=1)。后一种情况下， PLL0AUDIO 的工作方式与 PLL0USB 完全相同，且 MDEC 值直接被用于控制反馈分频器。

小数分频器激活后，sigma-delta 调制器模块生成正确比例的分频器值 M 和 M+1，从而实现 M+K/L 的平均分频比，其中 0<=K<=L，且 M、 K 和 L 为整数值。 M 由 PLLFRACT_CTRL 寄存器(PLLFRACT[21:15])的整数部分决定，K由PLLFRACT_CTRL寄存器(PLLFRACT[14:0])的小数部分决定。显示成到 M 分频器的输入前，要将连续 M 和 M+1 值进一步编码成适当的 MENC 值。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 23. 带小数分频器的 PLL0

11.7.6 PLL1

11.7.6.1 功能要点

- 1 MHz 至 50 MHz 输入频率。来自外部晶体的输入上限为 25 MHz。
- 可选择输出频率 9.75 MHz 至 320 MHz，占空比为 50%。
- 156 MHz 至 320 MHz 电流控制振荡器 (CCO) 频率。
- 掉电模式。
- 锁定探测器。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

11.7.6.2 PLL1 描述

图 24. PLL1 框图

此 PLL 的功能框图如图 24 所示。输入频率的范围从 10 MHz 到 25 MHz。输入时钟直接馈入相位频率检测器 (PFD)。此块将比较其输入的相位和频率，如果它们不匹配，将生成控制信号。锁相环滤波器对这些控制信号进行过滤，并驱动生成主时钟的电流控制振荡器 (CCO)。 CCO 频率范围为 156 MHz 到 320 MHz。这些时钟通过可编程的后分频器按 2xP 分频以创建输出时钟，或者直接发送到输出。然后，可编程的反馈分频器将主输出时钟按 M 分频，以生成反馈时钟。锁定探测器还会监控相位频率检测器的输出信号，在 PLL 已锁定到输入时钟时发出信号。

11.7.6.3 锁定检测器

锁定探测器测量输入时钟上升沿与反馈时钟上升沿之间的相位差异。只有当此差异小于八个以上连续输入时钟期间所谓的锁定标准时，锁定输出才会从低切换到高。单一过大的相位差异会立即复位计数器，并导致锁定信号降落（如果原来较高）。每行要求有八次相位测量值低于某个特定的数，以确保锁定探测器在输入时钟和反馈时钟两者的相位和频率极其相符时，才会指示锁定。这样，可有效防止虚假的锁定指示，由此确保锁定信号不会出错。

11.7.6.4 掉电控制

为降低无需 PLL 时钟时的功耗，引入了掉电模式。在此模式下，将关闭内部电流参考，停止振荡器和相位频率检测器，并且分频器将进入复位状态。同时，在掉电模式下，锁定输出将变低以指示 PLL 未处于锁定状态。掉电模式终止后， PLL 将恢复正常操作，并且在重新获得对输入时钟的锁定后，使锁定信号变高。

11.7.6.5 可选择的反馈分频器时钟

为使得功能和功耗之间可以达到平衡，通过将 FBSEL 设置为 0 或 1，可以将反馈分频器连接到 CCO 时钟或输出时钟。如果使用后分频器来缩减 CCO 时钟，则通过在较低的输出时钟（而非 CCO 时钟）上运行反馈分频器，可降低其电流消耗，但这样做，会将输出和相位检测器时钟频率之间的关联限制为整数值。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

11.7.6.6 直接输出模式

在正常工作模式下 （DIRECT 设置为 0）， CCO 时钟将按 2、 4、 8 或 16 分频 （取决于 PSEL[1:0] 的值），并自动给定占空比为 50% 的输出时钟。如果需要更高的输出频率，通过将 DIRECT 设置为 1，可以将 CCO 时钟直接发送到输出。由于 CCO 旨在直接生成 50% 占空比的时钟，所以输出时钟占空比在直接模式下也将为 50%。

11.7.6.7 分频器比率编程

预分频器

预分频器的分频比由NSEL[1:0]输入控制。PLL 的输入时钟和相位检测器时钟之间的分频比是 NSEL[1:0] 上的十进制值
加一。

后分频器

后分频器的分频比由 PSEL 位控制。分频比是 PSEL 位选定的 P 值的 2 倍。这样可保证输出时钟达到 50% 的占空比。

反馈分频器

反馈分频器的分频比由 MSEL 位控制。 PLL 的输出时钟和输入时钟之间的分频比是 MSEL 位上的十进制值加一。

更改分频器值

不建议在 PLL 运行时更改分频比。因为无法将 NSEL、 MSEL 和 PSEL 值的更改与分频器同步，所以存在计数器可能读取未定义值的风险，从而可能导致输出时钟频率的意外陡升或降落。要在分频器之间更改设置，建议使 PLL 掉电，调整分频器设置，然后再次启动 PLL。

11.7.6.8 频率选择

PLL 频率的计算公式使用下列参数 （另请参见图 24）：

整数模式

在此模式下，将使能后分频器，将反馈分频器设置为在 PLL 输出时钟上运行，从而提供以下频率关联：

$$FCLKOUT = M \times \frac{FCLKIN}{N} \quad (1)$$

$$FCCO = 2 \times P \times FCLKOUT = 2 \times P \times M \times \frac{FCLKIN}{N} \quad (2)$$

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

非整数模式

在此模式下，将使能后分频器，将反馈分频器设置为直接在 CCO 时钟上运行，从而提供以下频率关联：

$$FCLKOUT = \frac{FCCO}{2 \times P} = \frac{M}{2 \times P} \times \frac{FCLKIN}{N} \quad (3)$$

$$FCCO = M \times \frac{FCLKIN}{N} \quad (4)$$

直接模式

在此模式下，将禁用后分频器， CCO 时钟直接发送到输出，从而可使用以下频率等式：

$$FCLKOUT = FCCO = M \times \frac{FCLKIN}{N} \quad (5)$$

掉电模式

在此模式下，将关闭内部电流参考，停止振荡器和相位频率检测器，并且分频器将进入复位状态。同时，在掉电模式下，锁定输出将变低以指示 PLL 未处于锁定状态。掉电模式终止后， PLL 将恢复正常操作，并且在重新获得对输入时钟的锁定后，使锁定信号变高。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

11.8 CGU 配置示例

11.8.1 深度睡眠和掉电模式的 CGU 编程

在 LPC43xx 进入深度睡眠或掉电模式之前，必须编程所有输出级的 (OUTCLK_0 至 OUTCLK_27) 控制寄存器，以选择 IRC 作为时钟源。此外， PLL 必须处于掉电模式。

当 LPC43xx 从深度睡眠模式或掉电模式中唤醒时，IRC 将用作所有输出级的时钟源。另请参见第 10 章。

11.8.2 对 CGU 进行编程以使 I2S 的外围设备时钟速率达到 30 MHz

在此示例中，I2S 接口的外设时钟将设置为 30 MHz。外设 I2S 时钟属于 BASE_APB1_CLK 的分支。使用 12 MHz 的晶体作为时钟源，PLL1 乘数为 10 且整数分频器为 4，可提供所需的时钟速率。

对此示例，可按以下方式对 CGU 编程：

1. 使能 XTAL_OSC_CTRL 寄存器中的晶体振荡器 （表 60）。
2. 等待晶体稳定下来。
3. 选择晶体振荡器作为到 PLL1 的输入，并设置 PLL1_CTRL 寄存器中的分频器（参见表 71）：
   - 将 PLL1_CTRL 寄存器中的 CLK_SEL 位设置为 0x6。
   - 设置 MSEL = 9。
   - 设置 NSEL = 0。
   - 设置 PSEL = 1。
   - 设置 FBSEL = 1。
   - 设置 BYPASS = 0、 DIRECT = 0。
4. 等待 PLL1 锁定。
5. 在IDIVA寄存器中，选择PLL1作为整数分频器A (IDIVA)的时钟源，并设置AUTOBLOCK = 1 （参见表 71）。
6. 选择 IDIVA 作为基础时钟 BASE_APB1_CLK 的时钟源，并设置 AUTOBLOCK = 1（参见表 72）。
7. 确保在 CCU 中使能 I2S 分支时钟 CLK_APB1_I2S （参见表 91）。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

11.8.3 针对 USB 应用的 PLL0USB 设置

表 85 显示了为 USB0 产生 480 MHz 输出时钟的 M 分频器和 NP 分频器寄存器设置示例。

表 85. 480 MHz 输出时钟的 PLL0 （用于 USB）设置

Fclkin [MHz]	PLL0USB_MDIV 表 63	PLL0USB_NP_DIV 表 64
1	0x073E 56C9	0x0030 2062
2	0x073E 2DAD	0x0030 2062
3	0x0B3E 34B1	0x0030 2062
4	0x0E3E 7777	0x0030 2062
5	0x0D32 6667	0x0030 2062
6	0x0B2A 2A66	0x0030 2062
8	0x0820 6AAA	0x0030 2062
10	0x071A 7FAA	0x0030 2062
12	0x0616 7FFA	0x0030 2062
15	0x0512 3FFF	0x0030 2062
16	0x0410 1FFF	0x0030 2062
20	0x040E 03FF	0x0030 2062
24	0x030C 00FF	0x0030 2062

11.8.4 音频应用的 PLL0AUDIO 设置

11.8.4.1 使用小数分频器

表 86 显示了小数分频器激活时，音频 PLL0 的典型分频器设置。

要使用小数分频器，请遵照以下步骤：

1. 设置PLL0AUDIO_CTRL寄存器中的位SEL_EXT = 0和PLLFRACT_REQ = 1（表66）。
2. 计算输出频率 Fout 的 NDEC、 PDEC 和 PLLFRACT_CTRL。
3. 将计算的 NDEC 和 PDEC 值写入 PLL0AUDIO_NP_DIV 寄存器。
4. 将计算得到的 PLLFRACT_CTRL 值写入 PLL0AUDIOFRAC 寄存器。

表 86. 12 MHz 输入的 PLL0AUDIO 分频器设置

Fs [kHz]	Fout [MHz]	Fcco [MHz]	Error [Hz]	NDEC	PDEC	PLL0AUDIO_NP_DIV 表 68	PLLF0RACT_CTRL 表 69
128Fs							
192	24.576	540.672	1	514	29	0x0000201d	0x16872b
96	12.288	417.792	1	1	3	0x00001003	0x1a1cac
88.2	11.2896	338.688	1	0	24	0x00000018	0x070e56
64	8.192	344.064	1	0	30	0x0000001e	0x072b02
48	6.144	307.2	1	1	6	0x00001006	0x133333
44.1	5.6448	282.24	1	1	6	0x00001006	0x11a3d7
256Fs							
192	49.152	491.52	11	514	5	0x00002005	0x147ae1
96	24.576	540.672	1	514	29	0x0000201d	0x16872b

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 86. 12 MHz 输入的 PLL0AUDIO 分频器设置 *（续）*

Fs [kHz]	Fout [MHz]	Fcco [MHz]	Error [Hz]	NDEC	PDEC	PLL0AUDIO_NP_DIV	PLLF0RACT_CTRL
						表 68	表 69
88.2	22.5792	451.584	1	1	14	0x0000100e	0x1c3958
64	16.384	360.448	1	1	29	0x0000101d	0x16872b
48	12.288	417.792	1	1	3	0x00001003	0x1a1cac
44.1	11.2896	338.688	1	0	24	0x00000018	0x070e56
32	8.192	344.064	1	0	30	0x0000001e	0x072b02
24	6.144	307.2	1	1	6	0x00001006	0x133333
22.05	5.6448	282.24	1	1	6	0x00001006	0x11a3d7
384Fs							
192	73.728	442.368	4	2		0x00002001	0x24dd2f
96	36.864	442.368	2	2		0x0000200a	0x24dd2f
88.2	33.8688	338.688	2	0		0x00000005	0x070e56
64	24.576	540.672	1	514		0x0000201d	0x16872b
48	18.432	442.368	1	2		0x0000201b	0x24dd2f
44.1	16.9344	338.688	1	0		0x0000000e	0x070e56
32	12.288	417.792	1	1		0x00001003	0x1a1cac
24	9.216	276.48	1	1		0x00001018	0x1147ae
22.05	8.4672	338.688	1	0		0x0000001f	0x070e56
16	6.144	307.2	1	1		0x00001006	0x133333
12	4.608	276.48	1	1		0x00001012	0x1147ae
512Fs							
192	98.304	393.216	30	2	66	0x00002042	0x20c49b
96	49.152	491.52	11	514	5	0x00002005	0x147ae1
88.2	45.1584	451.584	2	1	5	0x00001005	0x1c3958
64	32.768	458.752	4	1	21	0x00001015	0x1cac08
48	24.576	540.672	1	514	29	0x0000201d	0x16872b
44.1	22.5792	451.584	1	1	14	0x0000100e	0x1c3958
32	16.384	360.448	1	1	29	0x0000101d	0x16872b
24	12.288	417.792	1	1	3	0x00001003	0x1a1cac
22.05	11.2896	338.688	1	0	24	0x00000018	0x070e56
16	8.192	344.064	1	0	30	0x0000001e	0x072b02
12	6.144	307.2	1	1	6	0x00001006	0x133333
11.025	5.6448	282.24	1	1	6	0x00001006	0x11a3d7
1024Fs							
192	196.608	393.216	60	2	98	0x00002062	0x20c49b
96	98.304	393.216	30	2	66	0x00002042	0x20c49b
88.2	90.3168	541.9008	28	0	1	0x00000001	0x0b4a23
64	65.536	524.288	13	2	2	0x00002002	0x2bb0cf
48	49.152	491.52	11	514	5	0x00002005	0x147ae1
44.1	45.1584	451.584	2	1	5	0x00001005	0x1c3958
32	32.768	458.752	4	1	21	0x00001015	0x1cac08

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 86. 12 MHz 输入的 PLL0AUDIO 分频器设置 （续）

Fs [kHz]	Fout [MHz]	Fcco [MHz]	Error [Hz]	NDEC	PDEC	PLL0AUDIO_NP_DIV 表 68	PLLF0RACT_CTRL 表 69
24	24.576	540.672	1	514	29	0x0000201d	0x16872b
22.05	22.5792	451.584	1	1	14	0x0000100e	0x1c3958
16	16.384	360.448	1	1	29	0x0000101d	0x16872b
12	12.288	417.792	1	1	3	0x00001003	0x1a1cac
11.025	11.2896	338.688	1	0	24	0x00000018	0x070e56
8	8.192	344.064	1	0	30	0x0000001e	0x072b02

11.8.4.2 绕过小数分频器

要绕过小数分频器并直接使用 MDEC 值，请遵照以下步骤：

1. 设置 PLL0AUDIO_CTRL 寄存器中的位 SEL_EXT = 1 （表 66）。
2. 计算输出频率 Fout 的 NDEC、 PDEC 和 MDEC。
3. 将计算的 NDEC 和 PDEC 值写入 PLL0AUDIO_NP_DIV 寄存器。
4. 将计算得到的 MDEC 值写入 PLL0AUDIO_MDIV 寄存器。

表 87. 绕过小数分频器的 12 MHz 的 PLL0AUDIO 分频器设置

Fs [KHz]	Fout [MHz]	Fcco [MHz]	Error [Hz]	NDEC	MDEC	PDEC	PLL0AUDIO_MDIV 表 67	PLL0AUDIO_NP_DIV 表 68
128 Fs								
192	24.576	491.52	0	63	13523	14	0x000034d3	0x0003f00e
96	12.288	368.64	0	63	2665	24	0x00000a69	0x0003f018
88.2	11.2896	338.688	0	45	18810	24	0x0000497a	0x0002d018
64	8.192	409.6	0	61	18724	6	0x00004924	0x0003d006
48	6.144	307.2	0	5	30580	6	0x00007774	0x00005006
44.1	5.6448	282.24	0	63	31356	6	0x00007a7c	0x0003f006
256 Fs								
192	49.152	491.52	0	63	13523	5	0x000034d3	0x0003f005
96	24.576	491.52	0	63	13523	14	0x000034d3	0x0003f00e
88.2	22.5792	451.584	0	45	29122	14	0x000071c2	0x0002d00e
64	16.384	491.52	0	63	13523	24	0x000034d3	0x0003f018
48	12.288	368.64	0	63	2665	24	0x00000a69	0x0003f018
44.1	11.2896	338.688	0	45	18810	24	0x0000497a	0x0002d018
32	8.192	409.6	0	61	18724	6	0x00004924	0x0003d006
24	6.144	307.2	0	5	30580	6	0x00007774	0x00005006
22.05	5.6448	282.24	0	63	31356	6	0x00007a7c	0x0003f006
384 Fs								
192	73.728	294.912	0	45	4770	66	0x000012a2	0x0002d042
96	36.864	368.64	0	63	2665	5	0x00000a69	0x0003f005

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 87. 绕过小数分频器的 12 MHz 的 PLL0AUDIO 分频器设置 *（续）*

Fs [KHz]	Fout [MHz]	Fcco [MHz]	Error [Hz]	NDEC	MDEC	PDEC	PLL0AUDIO_MDIV	PLL0AUDIO_NP_DIV
							表 67	表 68
88.2	33.8688	338.688	0	45	18810	5	0x0000497a	0x0002d005
64	24.576	491.52	0	63	13523	14	0x000034d3	0x0003f00e
48	18.432	368.64	0	63	2665	14	0x00000a69	0x0003f00e
44.1	16.9344	338.688	0	45	18810	14	0x0000497a	0x0002d00e
32	12.288	368.64	0	63	2665	24	0x00000a69	0x0003f018
24	9.216	460.8	0	5	12733	6	0x000031bd	0x00005006
22.05	8.4672	423.36	0	63	17692	6	0x0000451c	0x0003f006
16	6.144	307.2	0	5	30580	6	0x00007774	0x00005006
12	4.608	276.48	0	63	1513	18	0x000005e9	0x0003f012
512 Fs								
192	98.304	393.216	0	45	10784	66	0x00002a20	0x0002d042
96	49.152	491.52	0	63	13523	5	0x000034d3	0x0003f005
88.2	45.1584	451.584	0	45	29122	5	0x000071c2	0x0002d005
64	32.768	327.68	0	102	7482	5	0x00001d3a	0x00066005
48	24.576	491.52	0	63	13523	14	0x000034d3	0x0003f00e
44.1	22.5792	451.584	0	45	29122	14	0x000071c2	0x0002d00e
32	16.384	491.52	0	63	13523	24	0x000034d3	0x0003f018
24	12.288	368.64	0	63	2665	24	0x00000a69	0x0003f018
22.05	11.2896	338.688	0	45	18810	24	0x0000497a	0x0002d018
16	8.192	409.6	0	61	18724	6	0x00004924	0x0003d006
12	6.144	307.2	0	5	30580	6	0x00007774	0x00005006
11.025	5.6448	282.24	0	63	31356	6	0x00007a7c	0x0003f006
1024 Fs								
192	196.608	393.216	0	45	10784	98	0x00002a20	0x0002d062
96	98.304	393.216	0	45	10784	66	0x00002a20	0x0002d042
88.2	90.3168	541.901	31	181	3535	1	0x00000dcf	0x000b5001
64	65.536	393.216	0	45	10784	1	0x00002a20	0x0002d001
48	49.152	491.52	0	63	13523	5	0x000034d3	0x0003f005
44.1	45.1584	451.584	0	45	29122	5	0x000071c2	0x0002d005
32	32.768	327.68	0	102	7482	5	0x00001d3a	0x00066005
24	24.576	491.52	0	63	13523	14	0x000034d3	0x0003f00e
22.05	22.5792	451.584	0	45	29122	14	0x000071c2	0x0002d00e
16	16.384	491.52	0	63	13523	24	0x000034d3	0x0003f018
12	12.288	368.64	0	63	2665	24	0x00000a69	0x0003f018
11.025	11.2896	338.688	0	45	18810	24	0x0000497a	0x0002d018
8	8.192	409.6	0	61	18724	6	0x00004924	0x0003d006

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

12.1 本章导读

注：VADC 在器件 LPC4350/30/20/10 上不可用。

以太网、 USB0、 USB1 和 LCD 相关时钟并非在所有封装中均可用。参见表 2。

12.2 基本配置

CCU1/2 的配置方式如下：

- 参见表 88 以了解时钟与电源控制。
- 所有分支时钟都默认使能。
- 请勿在正常操作期间复位 CCU。
- 将 EMC 时钟分频器的输出时钟（表 97）与 CREG6 寄存器（表 43）中的位 16 一起配置。

表 88. CCU 时钟和电源控制

	基础时钟	分支时钟	工作频率
CCU1	BASE_M4_CLK	CLK_M4_BUS	最大 204 MHz
CCU2	BASE_M4_CLK	CLK_M4_BUS	最大 204 MHz

12.3 功能要点

CCU 可关闭或打开各个外设的时钟。

- 在关闭分支时钟之前，使用自动模式激活 AHB 禁用协议。
- 在唤醒模式中，可以选择要在唤醒事件后自动运行的时钟。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

12.4 简介

每个 CGU 基础时钟都有若干时钟分支，可通过时钟控制单元 CCU1 或 CCU2 单独打开或关闭。分支时钟在 CCU1 和 CCU2 之间分配。

表 89. CCU1 分支时钟

基础时钟	分支时钟	描述
BASE_APB3_CLK	CLK_APB3_BUS	APB3 总线时钟。
	CLK_APB3_I2C1	I2C1 寄存器接口时钟和 I2C1 外设时钟。
	CLK_APB3_DAC	DAC 寄存器接口的时钟。
	CLK_APB3_ADC0	ADC0 寄存器接口的时钟和 ADC0 外设时钟。
	CLK_APB3_ADC1	ADC1 寄存器接口的时钟和 ADC1 外设时钟。
	CLK_APB3_CAN0	C_CAN0 寄存器接口的时钟和 C_CAN0 外围设备时钟。
BASE_APB1_CLK	CLK_APB1_BUS	APB1 总线时钟。
	CLK_APB1_MOTOCON	PWM 电机控制模块时钟和 PWM 电机控制外设时钟。
	CLK_APB1_I2C0	I2C0 寄存器接口时钟和 I2C0 外设时钟。
	CLK_APB1_I2S	I2S0 和 I2S1 寄存器接口的时钟以及 I2S0 和 I2S1 外围设备时钟。
	CLK_APB1_CAN1	C_CAN1 寄存器接口的时钟和 C_CAN1 外围设备时钟。
BASE_SPIFI_CLK	CLK_SPIFI	SPIFI SCKI 时钟输入的时钟。
BASE_M4_CLK	CLK_M4_BUS	M4 总线时钟。
	CLK_M4_SPIFI	SPIFI 寄存器接口的时钟。
	CLK_M4_GPIO	GPIO 寄存器接口的时钟
	CLK_M4_LCD	LCD 寄存器接口的时钟。
	CLK_M4_ETHERNET	以太网寄存器接口的时钟。
	CLK_M4_USB0	USB0 寄存器接口的时钟。
	CLK_M4_EMC	外部存储控制器的时钟。
	CLK_M4_SDIO	SD/MMC 寄存器接口的时钟。
	CLK_M4_DMA	DMA 寄存器接口的时钟。
	CLK_M4_M4CORE	Cortex-M4 内核的时钟
	CLK_M4_SCT	SCT 寄存器接口的时钟。
	CLK_M4_USB1	USB1 寄存器接口的时钟。
	CLK_M4_EMC_DIV	EMC （带时钟分频器）的时钟。
	CLK_M4_M0APP	M0APP 协处理器的时钟。
	CLK_M4_VADC	VADC 的时钟。
	CLK_M4_WWDT	WWDT 寄存器接口的时钟。
	CLK_M4_UART0	USART0 寄存器接口的时钟。
	CLK_M4_UART1	UART1 寄存器接口的时钟。
	CLK_M4_SSP0	SSP0 寄存器接口的时钟。
	CLK_M4_TIMER0	定时器 0 寄存器接口的时钟和定时器 0 外设时钟。
	CLK_M4_TIMER1	定时器 1 寄存器接口的时钟和定时器 1 外设时钟。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 89. CCU1 分支时钟 *（续）*

基础时钟	分支时钟	描述
BASE_M4_CLK	CLK_M4_SCU	系统控制单元寄存器接口的时钟。
	CLK_M4_CREG	CREG 寄存器接口的时钟。
	CLK_M4_RITIMER	RI 定时器寄存器接口的时钟和 RI 定时器外设时钟
	CLK_M4_UART2	UART2 寄存器接口的时钟。
	CLK_M4_UART3	UART3 寄存器接口的时钟。
	CLK_M4_TIMER2	定时器 2 寄存器接口的时钟和定时器 2 外设时钟。
	CLK_M4_TIMER3	定时器 3 寄存器接口的时钟和定时器 3 外设时钟。
	CLK_M4_SSP1	SSP1 寄存器接口的时钟。
	CLK_M4_QEI	QEI 寄存器接口时钟和 QEI 外设时钟。
BASE_PERIPH_CLK	CLK_PERIPH_BUS	外设总线的时钟。
	CLK_PERIPH_CORE	外设内核的时钟。
	CLK_PERIPH_SGPIO	SGPIO 接口的时钟。
BASE_USB0_CLK	CLK_USB0	USB0 外设时钟。
BASE_USB1_CLK	CLK_USB1	USB1 外设时钟。
BASE_SPI_CLK	CLK_SPI	SPI 接口的时钟。
BASE_VADC_CLK	CLK_VADC	VADC 时钟。

表 90. CCU2 分支时钟

基础时钟	分支时钟	描述
BASE_APLL_CLK	CLK_APLL	音频 PLL 时钟
BASE_UART3_CLK	CLK_APB2_UART3	USART3 外设时钟。
BASE_UART2_CLK	CLK_APB2_UART2	USART2 外设时钟。
BASE_UART1_CLK	CLK_APB0_UART1	UART1 外设时钟。
BASE_UART0_CLK	CLK_APB0_UART0	USART0 外设时钟。
BASE_SSP1_CLK	CLK_APB2_SSP1	SSP1 外设时钟。
BASE_SSP0_CLK	CLK_APB0_SSP0	SSP0 外设时钟。
BASE_SDIO_CLK	CLK_SDIO	SD/MMC 外设时钟。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

12.5 寄存器描述

表 91. 寄存器简介：CCU1 （基址 0x4005 1000）

名称	访问类型	地址偏移	描述	复位值	参考
PM	R/W	0x000	CCU1 电源模式寄存器	0x0000 0000	表 93
BASE_STAT	R	0x004	CCU1 基础时钟状态寄存器	0x0000 0FFF	表 94
-	-	0x008 至 0x0FC	保留	-	
CLK_APB3_BUS_CFG	R/W	0x100	CLK_APB3_BUS 时钟配置寄存器	0x0000 0001	表 96
CLK_APB3_BUS_STAT	R	0x104	CLK_APB3_BUS 时钟状态寄存器	0x0000 0001	表 99
CLK_APB3_I2C1_CFG	R/W	0x108	CLK_APB3_I2C1 配置寄存器	0x0000 0001	表 96
CLK_APB3_I2C1_STAT	R	0x10C	CLK_APB3_I2C1 状态寄存器	0x0000 0001	表 99
CLK_APB3_DAC_CFG	R/W	0x110	CLK_APB3_DAC 配置寄存器	0x0000 0001	表 96
CLK_APB3_DAC_STAT	R	0x114	CLK_APB3_DAC 状态寄存器	0x0000 0001	表 99
CLK_APB3_ADC0_CFG	R/W	0x118	CLK_APB3_ADC0 配置寄存器	0x0000 0001	表 96
CLK_APB3_ADC0_STAT	R	0x11C	CLK_APB3_ADC0 状态寄存器	0x0000 0001	表 99
CLK_APB3_ADC1_CFG	R/W	0x120	CLK_APB3_ADC1 配置寄存器	0x0000 0001	表 96
CLK_APB3_ADC1_STAT	R	0x124	CLK_APB3_ADC1 状态寄存器	0x0000 0001	表 99
CLK_APB3_CAN0_CFG	R/W	0x128	CLK_APB3_CAN0 配置寄存器	0x0000 0001	表 96
CLK_APB3_CAN0_STAT	R	0x12C	CLK_APB3_CAN0 状态寄存器	0x0000 0001	表 99
-	-	0x130 至 0x1FC	保留	-	-
CLK_APB1_BUS_CFG	R/W	0x200	CLK_APB1_BUS 配置寄存器	0x0000 0001	表 96
CLK_APB1_BUS_STAT	R	0x204	CLK_APB1_BUS 状态寄存器	0x0000 0001	表 99
CLK_APB1_MOTOCON PWM_CFG	R/W	0x208	CLK_APB1_MOTOCON 配置寄存器	0x0000 0001	表 96
CLK_APB1_MOTOCON PWM_STAT	R	0x20C	CLK_APB1_MOTOCON 状态寄存器	0x0000 0001	表 99
CLK_APB1_I2C0_CFG	R/W	0x210	CLK_APB1_I2C0 配置寄存器	0x0000 0001	表 96
CLK_APB1_I2C0_STAT	R	0x214	CLK_APB1_I2C0 状态寄存器	0x0000 0001	表 99
CLK_APB1_I2S_CFG	R/W	0x218	CLK_APB1_I2S 配置寄存器	0x0000 0001	表 96
CLK_APB1_I2S_STAT	R	0x21C	CLK_APB1_I2S 状态寄存器	0x0000 0001	表 99
CLK_APB1_CAN1_CFG	R/W	0x220	CLK_APB3_CAN1 配置寄存器	0x0000 0001	表 96
CLK_APB1_CAN1_STAT	R	0x224	CLK_APB3_CAN1 状态寄存器	0x0000 0001	表 99
-	-	0x220 至 0x2FC	保留	-	-
CLK_SPIFI_CFG	R/W	0x300	CLK_SPIFI 配置寄存器	0x0000 0001	表 96
CLK_SPIFI_STAT	R	0x304	CLK_SPIFI 状态寄存器	0x0000 0001	表 99
-	-	0x308 至 0x3FC	保留	-	-
CLK_M4_BUS_CFG	R/W	0x400	CLK_M4_BUS 配置寄存器	0x0000 0001	表 96
CLK_M4_BUS_STAT	R	0x404	CLK_M4_BUS 状态寄存器	0x0000 0001	表 99
CLK_M4_SPIFI_CFG	R/W	0x408	CLK_M4_SPIFI 配置寄存器	0x0000 0001	表 96
CLK_M4_SPIFI_STAT	R	0x40C	CLK_M4_SPIFI 状态寄存器	0x0000 0001	表 99

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 91. 寄存器简介：CCU1 （基址 0x4005 1000） *（续）*

名称	访问类型	地址偏移	描述	复位值	参考
CLK_M4_GPIO_CFG	R/W	0x410	CLK_M4_GPIO 配置寄存器	0x0000 0001	表 96
CLK_M4_GPIO_STAT	R	0x414	CLK_M4_GPIO 状态寄存器	0x0000 0001	表 99
CLK_M4_LCD_CFG	R/W	0x418	CLK_M4_LCD 配置寄存器	0x0000 0001	表 96
CLK_M4_LCD_STAT	R	0x41C	CLK_M4_LCD 状态寄存器	0x0000 0001	表 99
CLK_M4_ETHERNET_CFG	R/W	0x420	CLK_M4_ETHERNET 配置寄存器	0x0000 0001	表 96
CLK_M4_ETHERNET_STAT	R	0x424	CLK_M4_ETHERNET 状态寄存器	0x0000 0001	表 99
CLK_M4_USB0_CFG	R/W	0x428	CLK_M4_USB0 配置寄存器	0x0000 0001	表 96
CLK_M4_USB0_STAT	R	0x42C	CLK_M4_USB0 状态寄存器	0x0000 0001	表 99
CLK_M4_EMC_CFG	R/W	0x430	CLK_M4_EMC 配置寄存器	0x0000 0001	表 96
CLK_M4_EMC_STAT	R	0x434	CLK_M4_EMC 状态寄存器	0x0000 0001	表 99
CLK_M4_SDIO_CFG	R/W	0x438	CLK_M4_SDIO 配置寄存器	0x0000 0001	表 96
CLK_M4_SDIO_STAT	R	0x43C	CLK_M4_SDIO 状态寄存器	0x0000 0001	表 99
CLK_M4_DMA_CFG	R/W	0x440	CLK_M4_DMA 配置寄存器	0x0000 0001	表 96
CLK_M4_DMA_STAT	R	0x444	CLK_M4_DMA 状态寄存器	0x0000 0001	表 99
CLK_M4_M4CORE_CFG	R/W	0x448	CLK_M4_M4CORE 配置寄存器	0x0000 0001	表 96
CLK_M4_M4CORE_STAT	R	0x44C	CLK_M4_M4CORE 状态寄存器	0x0000 0001	表 99
-	-	0x450 至 0x45C	保留	-	-
-	-	0x460 至 0x464	保留	-	-
CLK_M4_SCT_CFG	R/W	0x468	CLK_M4_SCT 配置寄存器	0x0000 0001	表 96
CLK_M4_SCT_STAT	R	0x46C	CLK_M4_SCT 状态寄存器	0x0000 0001	表 99
CLK_M4_USB1_CFG	R/W	0x470	CLK_M4_USB1 配置寄存器	0x0000 0001	表 96
CLK_M4_USB1_STAT	R	0x474	CLK_M4_USB1 状态寄存器	0x0000 0001	表 99
CLK_M4_EMCDIV_CFG	R/W	0x478	CLK_M4_EMCDIV 配置寄存器	0x0000 0001	表 97
CLK_M4_EMCDIV_STAT	R	0x47C	CLK_M4_EMCDIV 状态寄存器	0x0000 0001	表 99
-	-	0x480	-	-	-
-	-	0x484	-	-	-
-	-	0x488	-	-	-
-	-	0x48C	-	-	-
CLK_M4_M0APP_CFG	-	0x490	CLK_M4_M0_CFG 配置寄存器	0x0000 0001	表 97
CLK_M4_M0APP_STAT	-	0x494	CLK_M4_M0_STAT 状态寄存器	0x0000 0001	表 99
CLK_M4_VADC_CFG	-	0x498	CLK_M4_VADC_CFG 配置寄存器	0x0000 0001	表 97
CLK_M4_VADC_STAT		0x49C	CLK_M4_VADC_STAT 配置寄存器	0x0000 0001	表 99
-	-	0x480 至 0x4FC	保留	-	
CLK_M4_WWDT_CFG	R/W	0x500	CLK_M4_WWDT 配置寄存器	0x0000 0001	表 96
CLK_M4_WWDT_STAT	R	0x504	CLK_M4_WWDT 状态寄存器	0x0000 0001	表 99
CLK_M4_USART0_CFG	R/W	0x508	CLK_M4_UART0 配置寄存器	0x0000 0001	表 96
CLK_M4_USART0_STAT	R	0x50C	CLK_M4_UART0 状态寄存器	0x0000 0001	表 99
CLK_M4_UART1_CFG	R/W	0x510	CLK_M4_UART1 配置寄存器	0x0000 0001	表 96
CLK_M4_UART1_STAT	R	0x514	CLK_M4_UART1 状态寄存器	0x0000 0001	表 99

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 91. 寄存器简介：CCU1 （基址 0x4005 1000） *（续）*

名称	访问类型	地址偏移	描述	复位值	参考
CLK_M4_SSP0_CFG	R/W	0x518	CLK_M4_SSP0 配置寄存器	0x0000 0001	表 96
CLK_M4_SSP0_STAT	R	0x51C	CLK_M4_SSP0 状态寄存器	0x0000 0001	表 99
CLK_M4_TIMER0_CFG	R/W	0x520	CLK_M4_TIMER0 配置寄存器	0x0000 0001	表 96
CLK_M4_TIMER0_STAT	R	0x524	CLK_M4_TIMER0 状态寄存器	0x0000 0001	表 99
CLK_M4_TIMER1_CFG	R/W	0x528	CLK_M4_TIMER1 配置寄存器	0x0000 0001	表 96
CLK_M4_TIMER1_STAT	R	0x52C	CLK_M4_TIMER1 状态寄存器	0x0000 0001	表 99
CLK_M4_SCU_CFG	R/W	0x530	CLK_M4_SCU 配置寄存器	0x0000 0001	表 96
CLK_M4_SCU_STAT	R	0x534	CLK_M4_SCU 状态寄存器	0x0000 0001	表 99
CLK_M4_CREG_CFG	R/W	0x538	CLK_M4_CREG 配置寄存器	0x0000 0001	表 96
CLK_M4_CREG_STAT	R	0x53C	CLK_M4_CREG 状态寄存器	0x0000 0001	表 99
-	-	0x540 至 0x5FC	保留	-	-
CLK_M4_RITIMER_CFG	R/W	0x600	CLK_M4_RITIMER 配置寄存器	0x0000 0001	表 96
CLK_M4_RITIMER_STAT	R	0x604	CLK_M4_RITIMER 状态寄存器	0x0000 0001	表 99
CLK_M4_USART2_CFG	R/W	0x608	CLK_M4_UART2 配置寄存器	0x0000 0001	表 96
CLK_M4_USART2_STAT	R	0x60C	CLK_M4_UART2 状态寄存器	0x0000 0001	表 99
CLK_M4_USART3_CFG	R/W	0x610	CLK_M4_UART3 配置寄存器	0x0000 0001	表 96
CLK_M4_USART3_STAT	R	0x614	CLK_M4_UART3 状态寄存器	0x0000 0001	表 99
CLK_M4_TIMER2_CFG	R/W	0x618	CLK_M4_TIMER2 配置寄存器	0x0000 0001	表 96
CLK_M4_TIMER2_STAT	R	0x61C	CLK_M4_TIMER2 状态寄存器	0x0000 0001	表 99
CLK_M4_TIMER3_CFG	R/W	0x620	CLK_M4_TIMER3 配置寄存器	0x0000 0001	表 96
CLK_M4_TIMER3_STAT	R	0x624	CLK_M4_TIMER3 状态寄存器	0x0000 0001	表 99
CLK_M4_SSP1_CFG	R/W	0x628	CLK_M4_SSP1 配置寄存器	0x0000 0001	表 96
CLK_M4_SSP1_STAT	R	0x62C	CLK_M4_SSP1 状态寄存器	0x0000 0001	表 99
CLK_M4_QEI_CFG	R/W	0x630	CLK_M4_QEI 配置寄存器	0x0000 0001	表 96
CLK_M4_QEI_STAT	R	0x634	CLK_M4_QEI 状态寄存器	0x0000 0001	表 99
-	R/W	0x638 至 0x6FC	保留	-	-
CLK_PERIPH_BUS_CFG	R/W	0x700	CLK_PERIPH_BUS 配置寄存器	0x0000 0001	表 96
CLK_PERIPH_BUS_STAT	R	0x704	CLK_PERIPH_BUS 状态寄存器	0x0000 0001	表 99
-	R/W	0x708 至 0x70C	保留	-	-
CLK_PERIPH_CORE_CFG	R/W	0x710	CLK_PERIPH_CORE 配置寄存器	0x0000 0001	表 96
CLK_PERIPH_CORE_STAT	R	0x714	CLK_PERIPH_CORE 状态寄存器	0x0000 0001	表 99
CLK_PERIPH_SGPIO_CFG	R/W	0x718	CLK_PERIPH_SGPIO 配置寄存器	0x0000 0001	表 96
CLK_PERIPH_SGPIO_STAT	R	0x71C	CLK_PERIPH_SGPIO 状态寄存器	0x0000 0001	表 99
-	R/W	0x720 至 0x7FC	保留	-	-
CLK_USB0_CFG	R/W	0x800	CLK_USB0 配置寄存器	0x0000 0001	表 96
CLK_USB0_STAT	R	0x804	CLK_USB0 状态寄存器	0x0000 0001	表 99
-	-	0x808 至 0x8FC	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 91. 寄存器简介：CCU1 （基址 0x4005 1000） *（续）*

名称	访问类型	地址偏移	描述	复位值	参考
CLK_USB1_CFG	R/W	0x900	CLK_USB1 配置寄存器	0x0000 0001	表 96
CLK_USB1_STAT	R	0x904	CLK_USB1 状态寄存器	0x0000 0001	表 99
-	-	0x908 至 0x9FC	保留	-	-
CLK_SPI_CFG	R/W	0xA00	CLK_SPI 配置寄存器	0x0000 0001	表 96
CLK_SPI_STAT	R	0xA04	CLK_SPI 状态寄存器	0x0000 0001	表 99
CLK_VADC_CFG	R/W	0xB00	CLK_VADC 配置寄存器	0x0000 0001	表 96
CLK_VADC_STAT	R	0xB04	CLK_VADC 状态寄存器	0x0000 0001	表 99

表 92. 寄存器简介：CCU2 （基址 0x4005 2000）

名称	访问类型	地址偏移	描述	复位值	参考
PM	R/W	0x000	CCU2 电源模式寄存器	0x0000 0000	表 93
BASE_STAT	R	0x004	CCU2 基础时钟状态寄存器	0x0000 0FFF	表 95
-	-	0x008 至 0x0FC	保留	-	
CLK_APLL_CFG	R/W	0x100	CLK_APLL 配置寄存器	0x0000 0001	表 98
CLK_APLL_STAT	R	0x104	CLK_APLL 状态寄存器	0x0000 0001	表 100
-	-	0x108 至 0x1FC	保留	-	
CLK_APB2_USART3_CFG	R/W	0x200	CLK_APB2_UART3 配置寄存器	0x0000 0001	表 98
CLK_APB2_USART3_STAT	R	0x204	CLK_APB2_UART3 状态寄存器	0x0000 0001	表 100
-	-	0x208 至 0x2FC	保留	-	
CLK_APB2_USART2_CFG	R/W	0x300	CLK_APB2_UART2 配置寄存器	0x0000 0001	表 98
CLK_APB2_USART2_STAT	R	0x304	CLK_APB2_UART2 状态寄存器	0x0000 0001	表 100
-	-	0x308 至 0x3FC	保留	-	
CLK_APB0_UART1_CFG	R/W	0x400	CLK_APB0_UART1 配置寄存器	0x0000 0001	表 98
CLK_APB0_UART1_STAT	R	0x404	CLK_APB0_UART1 状态寄存器	0x0000 0001	表 100
-	-	0x408 至 0x4FC	保留	-	
CLK_APB0_USART0_CFG	R/W	0x500	CLK_APB0_UART0 配置寄存器	0x0000 0001	表 98
CLK_APB0_USART0_STAT	R	0x504	CLK_APB0_UART0 状态寄存器	0x0000 0001	表 100
-	-	0x508 至 0x5FC	保留	-	
CLK_APB2_SSP1_CFG	R/W	0x600	CLK_APB2_SSP1 配置寄存器	0x0000 0001	表 98
CLK_APB2_SSP1_STAT	R	0x604	CLK_APB2_SSP1 状态寄存器	0x0000 0001	表 100
-	-	0x608 至 0x6FC	保留	-	
CLK_APB0_SSP0_CFG	R/W	0x700	CLK_APB0_SSP0 配置寄存器	0x0000 0001	表 98
CLK_APB0_SSP0_STAT	R	0x704	CLK_APB0_SSP0 状态寄存器	0x0000 0001	表 100

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 92. 寄存器简介：CCU2 （基址 0x4005 2000）*（续）*

名称	访问类型	地址偏移	描述	复位值	参考
-	-	0x708 至 0x7FC	保留	-	
CLK_SDIO_CFG	R/W	0x800	CLK_SDIO 配置寄存器 （用于 SD/MMC）	0x0000 0001	表 98
CLK_SDIO_STAT	R	0x804	CLK_SDIO 状态寄存器 （用于 SD/MMC）	0x0000 0001	表 100

12.5.1 电源模式寄存器

此寄存器包含单个位 PD，将禁用所有已使能唤醒的输出时钟。（CCU 分支时钟配置寄存器中的 W = 1，第 12.5.3 节）。在检测到唤醒中断或向该 PD 位写入 0 后，通过向此寄存器写入值而被禁用的时钟将重新被激活。

表 93. CCU1/2 功耗模式寄存器 （CCU1_PM，地址 0x4005 1000 和 CCU2_PM，地址 0x4005 2000）位描述

位	符号	值	描述	复位值	访问类型
0	PD		掉电模式初始化	0	R/W
		0	正常操作。		
		1	已使能唤醒模式 (W = 1) 的时钟被禁用。		
31:1	-		保留。	-	-

12.5.2 基础时钟状态寄存器

此寄存器中的各个位表示指定的基础时钟是否可以安全关闭。逻辑零指示此基础时钟生成的所有分支时钟都被禁用。因此，也可以安全关闭基础时钟。逻辑 1 指示至少还有一个分支时钟在运行。

注：在写入到分支时钟的配置寄存器前，先重新激活基础时钟。

表 94. CCU1 基础时钟状态寄存器 （CCU1_BASE_STAT，地址 0x4005 1004）位描述

位	符号	描述	复位值	访问类型
0	BASE_APB3_CLK_IND	BASE_APB3_CLK 的基础时钟指示器 0 = 所有分支时钟均关闭。 1 = 至少有一个分支时钟在运行。	1	R
1	BASE_APB1_CLK_IND	BASE_APB1_CLK 的基础时钟指示器 0 = 所有分支时钟均关闭。 1 = 至少有一个分支时钟在运行。	1	R
2	BASE_SPIFI_CLK_IND	BASE_SPIFI_CLK 的基础时钟指示器 0 = 所有分支时钟均关闭。 1 = 至少有一个分支时钟在运行。	1	R
3	BASE_M4_CLK_IND	BASE_M4_CLK 的基础时钟指示器 0 = 所有分支时钟均关闭。 1 = 至少有一个分支时钟在运行。	1	R
5:4	-	保留	-	-
6	BASE_PERIPH_CLK_IND	BASE_PERIPH_CLK 的基础时钟指示器 0 = 所有分支时钟均关闭。 1 = 至少有一个分支时钟在运行。	1	R

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 94. CCU1 基础时钟状态寄存器 （CCU1_BASE_STAT，地址 0x4005 1004）位描述 *（续）*

位	符号	描述	复位值	访问类型
7	BASE_USB0_CLK_IND	BASE_USB0_CLK 的基础时钟指示器 0 = 所有分支时钟均关闭。 1 = 至少有一个分支时钟在运行。	1	R
8	BASE_USB1_CLK_IND	BASE_USB1_CLK 的基础时钟指示器 0 = 所有分支时钟均关闭。 1 = 至少有一个分支时钟在运行。	1	R
9	BASE_SPI_CLK_IND	BASE_SPI_CLK 的基础时钟指示器 0 = 所有分支时钟均关闭。 1 = 至少有一个分支时钟在运行。	1	R
31:10	-	保留	-	-

表 95. CCU2 基础时钟状态寄存器 （CCU2_BASE_STAT，地址 0x4005 2004）位描述

位	符号	描述	复位值	访问类型
0	-	保留。	-	-
1	BASE_UART3_CLK	BASE_UART3_CLK 的基础时钟指示器 0 = 所有分支时钟均关闭。 1 = 至少有一个分支时钟在运行。	1	R
2	BASE_UART2_CLK	BASE_UART2_CLK 的基础时钟指示器 0 = 所有分支时钟均关闭。 1 = 至少有一个分支时钟在运行。	1	R
3	BASE_UART1_CLK	BASE_UART1_CLK 的基础时钟指示器 0 = 所有分支时钟均关闭。 1 = 至少有一个分支时钟在运行。	1	R
4	BASE_UART0_CLK	BASE_UART0_CLK 的基础时钟指示器 0 = 所有分支时钟均关闭。 1 = 至少有一个分支时钟在运行。	1	R
5	BASE_SSP1_CLK	BASE_SSP1_CLK 的基础时钟指示器 0 = 所有分支时钟均关闭。 1 = 至少有一个分支时钟在运行。	1	R
6	BASE_SSP0_CLK	BASE_SSP0_CLK 的基础时钟指示器 0 = 所有分支时钟均关闭。 1 = 至少有一个分支时钟在运行。	1	R
7	-	保留。	-	-
31:8	-	保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

12.5.3 CCU1/2 分支时钟配置寄存器

从 CCU 生成的每个输出时钟均有配置寄存器。它们均遵循表 96 和表 98 中所述的格式。

在 LPC43xx 上，所有分支时钟在复位后都处于运行模式。自动和唤醒功能将被禁用。

该时钟可配置为在 CLK_XXX_CFG 寄存器中的位 RUN、AUTO 和 WAKEUP 说明的模式下运行，如下所示：

RUN — WAKEUP、PD 和 AUTO 控制位确定分支时钟的激活。如果设置了寄存器位 AUTO，则必须先完成 AHB 禁用协议，才能关闭时钟。 PD 位在表 93 中设置。

AUTO — 使能自动 （AHB 禁用机制）。在关闭时钟之前， PMU 将初始化 AHB 禁用协议。本协议可确保所有 AHB 事务都在关闭时钟前完成。

WAKEUP — 分支时钟在以下条件下将使能唤醒机制：

- 设置了功耗模式寄存器 （参见表 93）中的 PD 位。
- 使能了唤醒机制的时钟已关闭。

检测到唤醒机制事件或者清除 PD 位时，使能了唤醒机制的时钟将开启。如果设置了寄存器位 AUTO，则必须先完成 AHB 禁用协议，才能关闭时钟。

注：为安全禁用任何分支时钟，将向 CLK_XXX_CFG 寄存器进行两次单独的写入：首先设置 AUTO 位，然后在下次写操作时，通过将 RUN 位设置为零禁用时钟。

表 96. CCU1 分支时钟配置寄存器 （CLK_XXX_CFG，地址 0x4005 1100、 0x4005 1104、 ...、 0x4005 1A00）位描述

位	符号	值	描述	复位值	访问类型
0	RUN		运行使能	1	R/W
		0	时钟已禁用。		
		1	时钟已使能。		
1	AUTO		自动 （AHB 禁用机制）使能	0	R/W
		0	自动已禁用。		
		1	自动已使能。		
2	WAKEUP		唤醒机制机制使能	0	R/W
		0	唤醒机制已禁用。		
		1	唤醒机制已使能。		
31:3	-		保留	-	-

注：将 EMC 时钟分频器的输出时钟（表 97）与 CREG6 寄存器（表 43）中的位 16 一起配置。

表 97. CCU1 分支时钟配置寄存器 （CLK_EMCDIV_CFG，地址 0x4005 1478）位描述

位	符号	值	描述	复位值	访问类型
0	RUN		运行使能	1	R/W
		0	时钟已禁用。		
		1	时钟已使能。		
1	AUTO		自动 （AHB 禁用机制）使能	0	R/W
		0	自动已禁用。		
		1	自动已使能。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 97. CCU1 分支时钟配置寄存器 （CLK_EMCDIV_CFG，地址 0x4005 1478）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
2	WAKEUP		唤醒机制使能	0	R/W
		0	唤醒机制已禁用。		
		1	唤醒机制已使能。		
3	-		保留	-	-
4	-		保留	-	-
7:5	DIV		时钟分频器值	0	R/W
		0x0	无分频 （除以 1）。		
		0x1	2 分频		
		0x2	保留		
		0x3	保留		
		0x4	保留		
31:8	-		保留	-	-

表 98. CCU2 分支时钟配置寄存器 （CLK_XXX_CFG，地址 0x4005 2100、 0x4005 2200、 ...、 0x4005 2800）位描述

位	符号	值	描述	复位值	访问类型
0	RUN		运行使能	1	R/W
		0	时钟已禁用。		
		1	时钟已使能。		
1	AUTO		自动 （AHB 禁用机制）使能	0	R/W
		0	自动已禁用。		
		1	自动已使能。		
2	WAKEUP		唤醒机制使能	0	R/W
		0	唤醒机制已禁用。		
		1	唤醒机制已使能。		
31:3	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

12.5.4 CCU1/2 分支时钟状态寄存器

每个从 CCU 产生的输出时钟都有一个状态寄存器。在向输出时钟的配置寄存器写入时，自动或唤醒机制可延迟实际硬件信号的更新。状态寄存器显示这些信号的当前值。所有输出状态寄存器均遵循表 99 和表 100 中所述的格式。

表 99. CCU1 分支时钟状态寄存器 （CLK_XXX_STAT，地址 0x4005 1104、 0x4005 110C、 ...、0x4005 1A04）位描述

位	符号	描述	复位值	访问类型
0	RUN	运行使能状态 0 = 时钟已禁用。 1 = 时钟已使能。	1	R
1	AUTO	自动 （AHB 禁用机制）使能状态 0 = 自动已禁用。 1 = 自动已使能。	0	R
2	WAKEUP	唤醒机制使能状态 0 = 唤醒机制已禁用。 1 = 唤醒机制已使能。	0	R
31:3	-	保留	-	-

表 100. CCU2 分支时钟状态寄存器 （CLK_XXX_STAT，地址 0x4005 2104、 0x4005 2204、 ...、0x4005 2804）位描述

位	符号	描述	复位值	访问类型
0	RUN	运行使能状态 0 = 时钟已禁用 1 = 时钟已使能	1	R
1	AUTO	自动 （AHB 禁用机制）使能状态 0 = 自动已禁用 1 = 自动已使能	0	R
2	WAKEUP	唤醒机制使能状态 0 = 唤醒机制已禁用。 1 = 唤醒机制已使能。	0	R
31:3	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

13.1 本章导读

所有 LPC43xx 器件中的 RGU 都是一样的。以太网、USB0、USB1 和 LCD 相关时钟并非在所有封装中均可用。参见第 1.3 节。保留对应的复位寄存器。

13.2 基本配置

表 101. RGU 时钟和电源控制

	基本时钟	分支时钟	工作频率
RGU	BASE_M4_CLK	CLK_M4_BUS	最大 204 MHz
RGU 延迟时钟	BASE_SAFE_CLK	-	12 MHz

RGU 通过 BUS_RST (reset #8) 复位。

13.3 简介

RGU 允许为 LPC43xx 上的各个模块和外围设备产生独立的复位信号。每个复位信号由复位发生器确定，它具有一个输出 （复位信号）以及一个或多个输入，从而将复位发生器链接在一起，并创建复位等级。

注：ARM Cortex-M4 SYSRESETREQ 启动一个外围复位 PERIPH_RST。

图 25. RGU 框图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 102. 复位输出配置

复位输出发生器	复位输出编号	复位源	激活时复位的设备器件
CORE_RST	0	外部复位、BOD 复位、WWDT 超时复位	整个芯片，包括电池供电域中的外围设备：CGU、电源管理控制器、通用寄存器、报警定时器、 CREG 模块的器件和 RTC。
PERIPH_RST	1	CORE_RST	使用复位源PERIPH_RST和MASTER_RST的所有外围设备
MASTER_RST	2	PERIPH_RST	使用复位源MASTER_RST的所有外围设备
保留	3	-	-
WWDT_RST	4	CORE_RST	WWDT。无软件复位。
CREG_RST	5	CORE_RST	配置寄存器模块、Event 路由器、备份寄存器、 RTC、报警定时器。无软件复位。
保留	6 - 7	-	-
BUS_RST	8	PERIPH_RST	总线、RGU、CCU 和 CGU 寄存器、存储控制器、总线桥接。请勿在正常操作期间使用。
SCU_RST	9	PERIPH_RST	系统控制单元
保留	10 - 12	-	-
M4_RST	13	MASTER_RST	Cortex-M4 系统复位
保留	14	-	-
保留	15	-	-
LCD_RST	16	MASTER_RST	LCD 控制器复位
USB0_RST	17	MASTER_RST	USB0 复位
USB1_RST	18	MASTER_RST	USB1 复位
DMA_RST	19	MASTER_RST	DMA 复位
SDIO_RST	20	MASTER_RST	SDIO 复位
EMC_RST	21	MASTER_RST	外部存储控制器复位
ETHERNET_RST	22	MASTER_RST	以太网复位
保留	23	-	-
保留	24 - 27	-	-
GPIO_RST	28	PERIPH_RST	GPIO 复位
保留	29 - 31	-	-
TIMER0_RST	32	PERIPH_RST	定时器 0 复位
TIMER1_RST	33	PERIPH_RST	定时器 1 复位
TIMER2_RST	34	PERIPH_RST	定时器 2 复位
TIMER3_RST	35	PERIPH_RST	定时器 3 复位
RITIMER_RST	36	PERIPH_RST	重复中断定时器复位
SCT_RST	37	PERIPH_RST	状态可配置定时器复位
MOTOCONPWM_RST	38	PERIPH_RST	电机控制 PWM 复位
QEI_RST	39	PERIPH_RST	QEI 复位
ADC0_RST	40	PERIPH_RST	ADC0 复位（ADC 寄存器接口和模拟模块）
ADC1_RST	41	PERIPH_RST	ADC1 复位（ADC 寄存器接口和模拟模块）
DAC_RST	42	PERIPH_RST	DAC 复位（DAC 寄存器接口和模拟模块）
保留	43	-	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 102. 复位输出配置 *（续）*

复位输出发生器	复位输出编号	复位源	激活时复位的设备器件
UART0_RST	44	PERIPH_RST	USART0 复位
UART1_RST	45	PERIPH_RST	UART1 复位
UART2_RST	46	PERIPH_RST	USART2 复位
UART3_RST	47	PERIPH_RST	USART3 复位
I2C0_RST	48	PERIPH_RST	I2C0 复位
I2C1_RST	49	PERIPH_RST	I2C1 复位
SSP0_RST	50	PERIPH_RST	SSP0 复位
SSP1_RST	51	PERIPH_RST	SSP1 复位
I2S_RST	52	PERIPH_RST	I2S0 和 I2S1 复位
SPIFI_RST	53	PERIPH_RST	SPIFI 复位
CAN1_RST	54	PERIPH_RST	C_CAN1 复位
CAN0_RST	55	PERIPH_RST	C_CAN0 复位
M0APP_RST	56	MASTER_RST	ARM Cortex-M0 协处理器复位。 **注：**软件必须通过写入 RESET_CTRL1 寄存器，清除 M0 协处理器复位。
SGPIO_RST	57	PERIPH_RST	SGPIO 复位
SPI_RST	58	PERIPH_RST	SPI 复位
保留	59 - 63	-	-

RGU 还将监控每个复位输出的复位原因。复位原因可以使用两级粒度检索。

第一级由 3 个寄存器的 RESET_STATUS0 监控，并指示下列复位原因之一 （参见表 107 至表 110）：

- 未发生复位。
- 复位由软件生成 （使用寄存器 RESET_CTRL0 和 RESET_CTRL1）。
- 复位由任何复位源生成。

第二级粒度由每个复位输出 (RESET_EXT_STATUSn) 的单个寄存器监控，在其中将指示详细的复位原因，即是否激活每个复位发生器的任何潜在输入。下面列出了所有输入，但请注意。只有一小部分输入会连接到各个复位发生器：

- 外部复位 （从外部复位引脚）
- CORE_RST 输出
- PERIPH_RST 输出
- MASTER_RST 输出
- BOD 复位信号
- WWDT 超时信号

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

13.3.1 复位等级

等级如下所示 （参见表 103）：

1. 外部复位、 BOD 复位信号、 WWDT 超时和 PMU 的复位信号。
2. CORE_RST（输入为外部复位引脚、BOD 复位和 WWDT 超时复位）；复位整个芯片，包括 WWDT 和配置寄存器模块 CREG。
3. PERIPH_RST（输入为 CORE_RST）；复位所有 APB 外围设备和 ARM 内核，但不包括 WWDT 和 CREG 模块。
4. MASTER_RST（输入为 PERIPH_RST）；复位 ARM Cortex-M4 内核和 AHB 外围设备（DMA、 USB0/1、 LCD、 SDIO、 EMC）。

表 103. 复位优先级

优先级	复位输入	WWDT	CREG/ RTC/ Event router	ABP 外围设备	Cortex-M4 Core	AHB 外围设备	RGU	EMC	GPIO	SRAM 控制器
1	外部复位引脚、BOD、 WWDT	是	是	是	是	是	是	是	是	是
2	CORE_RST	是	是	是	是	是	是	是	是	是
3	PERIPH_RST	否	否	是	是	是	是	是	是	是
4	MASTER_RST	否	否	否	是	是	是	是	是	是

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 26. RGU 复位结构

13.4 寄存器概述

表 104. 寄存器简介：RGU （基址：0x4005 3000)

名称	访问类型	地址偏移	描述	复位值	参考
RESET_CTRL0	W	0x100	复位控制寄存器 0	-	参见表 105
RESET_CTRL1	W	0x104	复位控制寄存器 1	-	参见表 106
RESET_STATUS0	R/W	0x110	复位状态寄存器 0	0x5555 0050	参见表 107
RESET_STATUS1	R/W	0x114	复位状态寄存器 1	0x5555 5555	参见表 108
RESET_STATUS2	R/W	0x118	复位状态寄存器 2	0x5555 5555	参见表 109
RESET_STATUS3	R/W	0x11C	复位状态寄存器 3	0x5555 5555	参见表 110
RESET_ACTIVE_STATUS0	R	0x150	复位活动状态寄存器 0	0x0	参见表 111
RESET_ACTIVE_STATUS1	R	0x154	复位活动状态寄存器 1	0x0	参见表 112
RESET_EXT_STAT0	R/W	0x400	CORE_RST 的复位外部状态寄存器 0	0x0	参见表 113
RESET_EXT_STAT1	R/W	0x404	PERIPH_RST 的复位外部状态寄存器 1	0x0	参见表 114
RESET_EXT_STAT2	R/W	0x408	MASTER_RST的复位外部状态寄存器2	0x0	参见表 115
RESET_EXT_STAT3	-	0x40C	保留	-	
RESET_EXT_STAT4	R/W	0x410	WWDT_RST 的复位外部状态寄存器 4	0x0	参见表 116
RESET_EXT_STAT5	R/W	0x414	CREG_RST 的复位外部状态寄存器 5	0x0	参见表 117
RESET_EXT_STAT6	-	0x418	保留	-	-
RESET_EXT_STAT7	-	0x41C	保留	-	-
RESET_EXT_STAT8	R/W	0x420	BUS_RST 的复位外部状态寄存器 8	0x0	参见表 118
RESET_EXT_STAT9	R/W	0x424	SCU_RST 的复位外部状态寄存器 9	0x0	参见表 118
RESET_EXT_STAT10	-	0x428	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 104. 寄存器简介：RGU （基址：0x4005 3000） *（续）*

名称	访问类型	地址偏移	描述	复位值	参考
RESET_EXT_STAT11	-	0x42C	保留	-	-
RESET_EXT_STAT12	-	0x430	保留	-	-
RESET_EXT_STAT13	R/W	0x434	M4_RST 复位外部状态寄存器 13	0x0	参见表 119
RESET_EXT_STAT14	-	0x438	保留	-	-
RESET_EXT_STAT15	-	0x43C	保留	-	-
RESET_EXT_STAT16	R/W	0x440	LCD_RST 的复位外部状态寄存器 16	0x0	参见表 119
RESET_EXT_STAT17	R/W	0x444	USB0_RST 的复位外部状态寄存器 17	0x0	参见表 119
RESET_EXT_STAT18	R/W	0x448	USB1_RST 的复位外部状态寄存器 18	0x0	参见表 119
RESET_EXT_STAT19	R/W	0x44C	DMA_RST 的复位外部状态寄存器 19	0x0	参见表 119
RESET_EXT_STAT20	R/W	0x450	SDIO_RST 的复位外部状态寄存器 20	0x0	参见表 119
RESET_EXT_STAT21	R/W	0x454	EMC_RST 的复位外部状态寄存器 21	0x0	参见表 119
RESET_EXT_STAT22	R/W	0x458	ETHERNET_RST 的复位外部状态寄存器 22	0x0	参见表 119
RESET_EXT_STAT23	-	0x45C	保留	-	-
RESET_EXT_STAT24	-	0x460	保留	-	-
RESET_EXT_STAT25	-	0x464	保留	-	-
RESET_EXT_STAT26	-	0x468	保留	-	-
RESET_EXT_STAT27	-	0x46C	保留	-	-
RESET_EXT_STAT28	R/W	0x470	GPIO_RST 的复位外部状态寄存器 28	0x0	参见表 118
RESET_EXT_STAT29	-	0x474	保留	-	-
RESET_EXT_STAT30	-	0x478	保留	-	-
RESET_EXT_STAT31	-	0x47C	保留	-	-
RESET_EXT_STAT32	R/W	0x480	TIMER0_RST 的复位外部状态寄存器 32	0x0	参见表 118
RESET_EXT_STAT33	R/W	0x484	TIMER1_RST 的复位外部状态寄存器 33	0x0	参见表 118
RESET_EXT_STAT34	R/W	0x488	TIMER2_RST 的复位外部状态寄存器 34	0x0	参见表 118
RESET_EXT_STAT35	R/W	0x48C	TIMER3_RST 的复位外部状态寄存器 35	0x0	参见表 118
RESET_EXT_STAT36	R/W	0x490	RITIMER_RST 的复位外部状态寄存器 36	0x0	参见表 118
RESET_EXT_STAT37	R/W	0x494	SCT_RST 的复位外部状态寄存器 37	0x0	参见表 118
RESET_EXT_STAT38	R/W	0x498	MOTOCONPWM_RST的复位外部状态寄存器 38	0x0	参见表 118
RESET_EXT_STAT39	R/W	0x49C	QEI_RST 的复位外部状态寄存器 39	0x0	参见表 118
RESET_EXT_STAT40	R/W	0x4A0	ADC0_RST 的复位外部状态寄存器 40	0x0	参见表 118
RESET_EXT_STAT41	R/W	0x4A4	ADC1_RST 的复位外部状态寄存器 41	0x0	参见表 118
RESET_EXT_STAT42	R/W	0x4A8	DAC_RST 的复位外部状态寄存器 42	0x0	参见表 118
RESET_EXT_STAT43	-	0x4AC	保留	0x0	-
RESET_EXT_STAT44	R/W	0x4B0	UART0_RST 的复位外部状态寄存器 44	0x0	参见表 118
RESET_EXT_STAT45	R/W	0x4B4	UART1_RST 的复位外部状态寄存器 45	0x0	参见表 118
RESET_EXT_STAT46	R/W	0x4B8	UART2_RST 的复位外部状态寄存器 46	0x0	参见表 118

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 104. 寄存器简介：RGU （基址：0x4005 3000） *（续）*

名称	访问类型	地址偏移	描述	复位值	参考
RESET_EXT_STAT47	R/W	0x4BC	UART3_RST 的复位外部状态寄存器 47	0x0	参见表 118
RESET_EXT_STAT48	R/W	0x4C0	I2C0_RST 的复位外部状态寄存器 48	0x0	参见表 118
RESET_EXT_STAT49	R/W	0x4C4	I2C1_RST 的复位外部状态寄存器 49	0x0	参见表 118
RESET_EXT_STAT50	R/W	0x4C8	SSP0_RST 的复位外部状态寄存器 50	0x0	参见表 118
RESET_EXT_STAT51	R/W	0x4CC	SSP1_RST 的复位外部状态寄存器 51	0x0	参见表 118
RESET_EXT_STAT52	R/W	0x4D0	I2S_RST 的复位外部状态寄存器 52	0x0	参见表 118
RESET_EXT_STAT53	R/W	0x4D4	SPIFI_RST 的复位外部状态寄存器 53	0x0	参见表 118
RESET_EXT_STAT54	R/W	0x4D8	CAN1_RST 的复位外部状态寄存器 54	0x0	参见表 118
RESET_EXT_STAT55	R/W	0x4DC	CAN0_RST 的复位外部状态寄存器 55	0x0	参见表 118
RESET_EXT_STAT56	R/W	0x4E0	M0APP_RST的复位外部状态寄存器56	< 待定 >	参见表 118
RESET_EXT_STAT57	R/W	0x4E4	SGPIO_RST 的复位外部状态寄存器 57	0x0	参见表 118
RESET_EXT_STAT58	R/W	0x4E8	SPI_RST 的复位外部状态寄存器 58	0x0	参见表 118
RESET_EXT_STAT59	-	0x4EC	保留	-	-
RESET_EXT_STAT60	-	0x4F0	保留	-	-
RESET_EXT_STAT61	-	0x4F4	保留	-	-
RESET_EXT_STAT62	-	0x4F8	保留	-	-
RESET_EXT_STAT63	-	0x4FC	保留	-	-

13.4.1 RGU 复位控制寄存器

RGU 复位控制寄存器允许软件激活并清除各个复位输出。每个位与各个复位输出相对应，并且写入 1 可激活该输出。除 M0APP_RST 之外，在固定的延迟期间之后，将自动取消激活复位输出。如果复位输出可以手动发出，则它在推到低位后仍然保持激活状态，直到向此寄存器中的相应位写入 0。这取决于复位激活是来自复位控制寄存器还是任何其他源

表 105. 复位控制寄存器 0 （RESET_CTRL0，地址 0x4005 3100）位描述

位	符号	描述	复位值	访问类型
0	CORE_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
1	PERIPH_RST	写入 1 将激活复位。三个时钟周期后，此位自动清除为 0。	0	W
2	MASTER_RST	写入 1 将激活复位。三个时钟周期后，此位自动清除为 0。	0	W
3	-	保留	0	-
4	WWDT_RST	向此位写入 1 不起作用。	0	-
5	CREG_RST	向此位写入 1 不起作用。	0	-
6	-	保留	0	-
7	-	保留	0	-
8	BUS_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。请勿在正常操作期间使用	0	W
9	SCU_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
10	PINMUX_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
11	-	保留	0	-
12	-	保留	0	-
13	M4_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
14	-	保留	0	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 105. 复位控制寄存器 0 （RESET_CTRL0，地址 0x4005 3100）位描述 *（续）*

位	符号	描述	复位值	访问类型
15	-	保留	0	-
16	LCD_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
17	USB0_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
18	USB1_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
19	DMA_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
20	SDIO_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
21	EMC_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
22	ETHERNET_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
23	-	保留	-	-
24	-	保留	-	-
25	-	保留	-	-
26	-	保留	-	-
27	-	保留	-	-
28	GPIO_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
29	-	保留	-	-
30	-	保留	-	-
31	-	保留	-	-

表 106. 复位控制寄存器 1 （RESET_CTRL1，地址 0x4005 3104）位描述

位	符号	描述	复位值	访问类型
0	TIMER0_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
1	TIMER1_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
2	TIMER2_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
3	TIMER3_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
4	RITIMER_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
5	SCT_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
6	MOTOCONPWM_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
7	QEI_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
8	ADC0_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
9	ADC1_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
10	DAC_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
11	-	保留	-	-
12	UART0_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
13	UART1_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
14	UART2_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
15	UART3_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
16	I2C0_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
17	I2C1_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
18	SSP0_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
19	SSP1_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
20	I2S_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 106. 复位控制寄存器 1 （RESET_CTRL1，地址 0x4005 3104）位描述 *（续）*

位	符号	描述	复位值	访问类型
21	SPIFI_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
22	CAN1_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
23	CAN0_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
24	M0APP_RST	写入1将激活复位。写0可以清除该复位。**该位必须由软件清除。**	1	W
25	SGPIO_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
26	SPI_RST	写入 1 将激活复位。一个时钟周期后，此位自动清除为 0。	0	W
27	-	保留	-	-
28	-	保留	-	-
29	-	保留	-	-
30	-	保留	-	-
31	-	保留	-	-

13.4.2 RGU 复位状态寄存器

复位状态寄存器显示哪个源 （如有）导致 RGU 各个复位输出的最后一次复位激活。如果 RGU 的一个或多个输入导致复位输出转为有效 （指示值为 01），则可读取对应的 RESET_EXT_STATUS 寄存器 （参见第 13.4.4 节）。

通过向每个状态位写入 0，可清除 RESET_STATUS 寄存器。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 107. 复位状态寄存器 0 （RESET_STATUS0，地址 0x4005 3110）位描述

位	符号	描述	复位值	访问类型
1:0	CORE_RST	CORE_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	00	R/W
3:2	PERIPH_RST	PERIPH_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 – 该复位为自动清除 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	00	R/W
5:4	MASTER_RST	MASTER_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 – 该复位为自动清除 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
7:6	-	保留	01	
9:8	WWDT_RST	WWDT_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 保留	00	R/W
11:10	CREG_RST	CREG_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 保留	00	R/W
13:12	-	保留	01	-
15:14	-	保留	01	-
17:16	BUS_RST	BUS_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
19:18	SCU_RST	SCU_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
21:20	-	保留	01	-
23:22	-	保留	01	-
25:24	-	保留	01	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 107. 复位状态寄存器 0 （RESET_STATUS0，地址 0x4005 3110）位描述 *（续）*

位	符号	描述	复位值	访问类型
27:26	M4_RST	M4_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
29:28	-	保留	01	-
31:30	-	保留	01	-

表 108. 复位状态寄存器 1 （RESET_STATUS1，地址 0x4005 3114）位描述

位	符号	描述	复位值	访问类型
1:0	LCD_RST	LCD_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
3:2	USB0_RST	USB0_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
5:4	USB1_RST	USB1_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
7:6	DMA_RST	DMA_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
9:8	SDIO_RST	SDIO_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
11:10	EMC_RST	EMC_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 108. 复位状态寄存器 1 （RESET_STATUS1，地址 0x4005 3114）位描述 *（续）*

位	符号	描述	复位值	访问类型
13:12	ETHERNET_RST	ETHERNET_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
15:14	-	保留	01	-
17:16	-	保留	01	-
19:18	-	保留	01	-
21:20	-	保留	01	-
23:22	-	保留	01	-
25:24	GPIO_RST	GPIO_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
27:26	-	保留	01	-
29:28	-	保留	01	-
31:30	-	保留	01	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 109. 复位状态寄存器 2 （RESET_STATUS2，地址 0x4005 3118）位描述

位	符号	描述	复位值	访问类型
1:0	TIMER0_RST	TIMER0_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
3:2	TIMER1_RST	TIMER1_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
5:4	TIMER2_RST	TIMER2_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
7:6	TIMER3_RST	TIMER3_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
9:8	RITIMER_RST	RITIMER_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
11:10	SCT_RST	SCT_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
13:12	MOTOCONPWM_RST	MOTOCONPWM_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
15:14	QEI_RST	QEI_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 109. 复位状态寄存器 2（RESET_STATUS2，地址 0x4005 3118）位描述 *（续）*

位	符号	描述	复位值	访问类型
17:16	ADC0_RST	ADC0_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
19:18	ADC1_RST	ADC1_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
21:20	DAC_RST	DAC_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
23:22	-	保留	01	R/W
25:24	UART0_RST	UART0_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
27:26	UART1_RST	UART1_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
29:28	UART2_RST	UART2_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
31:30	UART3_RST	UART3_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 110. 复位状态寄存器 3 （RESET_STATUS3，地址 0x4005 311C）位描述

位	符号	描述	复位值	访问类型
1:0	I2C0_RST	I2C0_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
3:2	I2C1_RST	I2C1_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
5:4	SSP0_RST	SSP0_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
7:6	SSP1_RST	SSP1_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
9:8	I2S_RST	I2S_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
11:10	SPIFI_RST	SPIFI_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
13:12	CAN1_RST	CAN1_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
15:14	CAN0_RST	CAN0_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 110. 复位状态寄存器 3 （RESET_STATUS3，地址 0x4005 311C）位描述 *（续）*

位	符号	描述	复位值	访问类型
17:16	M0APP_RST	M0APP_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	11	R/W
19:18	SGPIO_RST	SGPIO_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
21:20	SPI_RST	SPI_RST 复位发生器输出的状态 00 = 未发生过复位 01 = 上次复位由复位发生器的输入导致 10 = 保留 11 = 上次复位由软件写入 RESET_CTRL 寄存器导致	01	R/W
23:22	-	保留	01	-
25:24	-	保留	01	-
27:26	-	保留	01	-
29:28	-	保留	01	-
31:30	-	保留	01	-

13.4.3 RGU 复位活动状态寄存器

复位活动状态寄存器显示 RGU 复位输出的当前值。请注意，复位活动状态为 LOW。

表 111. 复位活动状态寄存器 0 （RESET_ACTIVE_STATUS0，地址 0x4005 3150）位描述

位	符号	描述	复位值	访问类型
0	CORE_RST	CORE_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
1	PERIPH_RST	PERIPH_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
2	MASTER_RST	MASTER_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
3	-	保留	0	
4	WWDT_RST	WWDT_RS 的当前状态 0 = 复位中 1 = 无复位	0	R
5	CREG_RST	CREG_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
6	-	保留	0	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 111. 复位活动状态寄存器 0 （RESET_ACTIVE_STATUS0，地址 0x4005 3150）位描述 *（续）*

位	符号	描述	复位值	访问类型
7	-	保留	0	
8	BUS_RST	BUS_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
9	SCU_RST	SCU_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
10	PINMUX_RST	PINMUX_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
11	-	保留	0	-
12	-	保留	0	-
13	M4_RST	M4_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
14	-	保留	0	
15	-	保留	0	
16	LCD_RST	LCD_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
17	USB0_RST	USB0_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
18	USB1_RST	USB1_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
19	DMA_RST	DMA_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
20	SDIO_RST	SDIO_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
21	EMC_RST	EMC_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
22	ETHERNET_RST	ETHERNET_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
23	-	保留	-	-
24	-	保留	-	-
25	-	保留	-	-
26	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 111. 复位活动状态寄存器 0 （RESET_ACTIVE_STATUS0，地址 0x4005 3150）位描述 *（续）*

位	符号	描述	复位值	访问类型
27	-	保留	-	-
28	GPIO_RST	GPIO_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
29	-	保留	-	-
30	-	保留	-	-
31	-	保留	-	-

表 112. 复位活动状态寄存器 1 （RESET_ACTIVE_STATUS1，地址 0x4005 3154）位描述

位	符号	描述	复位值	访问类型
0	TIMER0_RST	TIMER0_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
1	TIMER1_RST	TIMER1_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
2	TIMER2_RST	TIMER2_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
3	TIMER3_RST	TIMER3_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
4	RITIMER_RST	RITIMER_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
5	SCT_RST	SCT_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
6	MOTOCONPWM_RST	MOTOCONPWM_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
7	QEI_RST	QEI_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
8	ADC0_RST	ADC0_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
9	ADC1_RST	ADC1_RST 的当前状态 0 = 复位中 1 = 无复位	0	R

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 112. 复位活动状态寄存器 1 （RESET_ACTIVE_STATUS1，地址 0x4005 3154）位描述 *（续）*

位	符号	描述	复位值	访问类型
10	DAC_RST	DAC_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
11	-		-	-
12	UART0_RST	UART0_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
13	UART1_RST	UART1_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
14	UART2_RST	UART2_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
15	UART3_RST	UART3_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
16	I2C0_RST	I2C0_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
17	I2C1_RST	I2C1_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
18	SSP0_RST	SSP0_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
19	SSP1_RST	SSP1_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
20	I2S_RST	I2S_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
21	SPIFI_RST	SPIFI_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
22	CAN1_RST	CAN1_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
23	CAN0_RST	CAN0_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
24	M0APP_RST	M0APP_RST 的当前状态 0 = 复位中 1 = 无复位	0	R

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 112. 复位活动状态寄存器 1 （RESET_ACTIVE_STATUS1，地址 0x4005 3154）位描述 *（续）*

位	符号	描述	复位值	访问类型
25	SGPIO_RST	SGPIO_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
26	SPI_RST	SPI_RST 的当前状态 0 = 复位中 1 = 无复位	0	R
27	-	保留	-	-
28	-	保留	-	-
29	-	保留	-	-
30	-	保留	-	-
31	-	保留	-	-

13.4.4 复位外部状态寄存器

外部状态寄存器指示哪个到复位发生器的输入导致复位输出有效。复位外部状态寄存器中设为 1 的任何位在读取操作后都将清除为 0，以便能够检测下一次复位。

所有复位发生器（WWDT 超时复位、BOD 复位和来自 PMU 的复位信号除外）和无输入的软件复位，都有相关联的外部状态寄存器。 CORE_RST 复位发生器有三个可能的输入（WWDT 超时复位、BOD 复位和 PMU），并将在外部状态寄存器中指示由哪个输入导致复位。所有其他复位发生器都只有一个输入，可能是 CORE_RST、 PERIPHERAL_RST 或 MASTER_RST，具体取决于等级。

请注意，外部状态寄存器不会显示复位是否由软件复位激活。软件复位将在复位状态寄存器 0 至 3 中指示 （参见表 107 至表 110）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

13.4.4.1 CORE_RST 的复位外部状态寄存器 0

此寄存器显示 CORE_RST 是否由 CORE_RST 复位发生器的任何输入激活。 CORE_RST 可以由外部复位引脚、WWDT 超时、BOD 复位或通过向 RESET_CTRL0 寄存器的位 0 写入进行激活。

表 113. 复位外部状态寄存器 0 （RESET_EXT_STAT0，地址 0x4005 3400）位描述

位	符号	描述	复位值	访问类型
0	EXT_RESET	复位由来自复位引脚的外部复位激活。写入0进行清除。 0 = 复位非由复位引脚激活 1 = 曾激活复位	0	R/W
1	-	保留。请勿修改，读为逻辑 0。	0	-
2	-	保留。请勿修改，读为逻辑 0。	0	-
3	-	保留。请勿修改，读为逻辑 0。	0	-
4	BOD_RESET	复位由 BOD 复位激活。写入 0 进行清除。 0 = 复位非由 BOD 激活 1 = 曾激活复位	0	R/W
5	WWDT_RESET	复位由 WWDT 超时激活。写入 0 进行清除。 0 = 复位非由 WWDT 激活 1 = 曾激活复位	0	R/W
31:6	-	保留。请勿修改，读为逻辑 0。	0	-

13.4.4.2 PERIPH_RST 的复位外部状态寄存器 1

此寄存器显示 PERIPH_RST 是否由 CORE_RST 输出激活。除通过向 RESET_CTRL 寄存器写入进行软件复位外，由 CORE_RST 生成的复位是 PERIPH_RST 唯一可能的复位源。

表 114. 复位外部状态寄存器 1 （RESET_EXT_STAT1，地址 0x4005 3404）位描述

位	符号	描述	复位值	访问类型
0	-	保留。请勿修改，读为逻辑 0。	0	-
1	CORE_RESET	复位由 CORE_RST 输出激活。写入 0 进行清除。 0 = 复位未激活 1 = 曾激活复位	0	R/W
31:2	-	保留。请勿修改，读为逻辑 0。	0	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

13.4.4.3 MASTER_RST 的复位外部状态寄存器 2

表 115. 复位外部状态寄存器 2 （RESET_EXT_STAT2，地址 0x4005 3408）位描述

位	符号	描述	复位值	访问类型
1:0	-	保留。请勿修改，读为逻辑 0。	0	-
2	PERIPHERAL_RESET	复位由 PERIPHERAL_RST 输出激活。写入 0 进行清除。 0 = 复位未激活 1 = 曾激活复位	0	R/W
31:3	-	保留。请勿修改，读为逻辑 0。	0	-

13.4.4.4 WWDT_RST 的复位外部状态寄存器 4

表 116. 复位外部状态寄存器 4 （RESET_EXT_STAT4，地址 0x4005 3410）位描述

位	符号	描述	复位值	访问类型
0	-	保留。请勿修改，读为逻辑 0。	0	-
1	CORE_RESET	复位由CORE_RST输出激活。写入0进行清除。 0 = 复位未激活 1 = 曾激活复位	0	R/W
31:2	-	保留。请勿修改，读为逻辑 0。	0	-

13.4.4.5 CREG_RST 的复位外部状态寄存器 5

表 117. 复位外部状态寄存器 5 （RESET_EXT_STAT5，地址 0x4005 3414）位描述

位	符号	描述	复位值	访问类型
0	-	保留。请勿修改，读为逻辑 0。	0	-
1	CORE_RESET	复位由CORE_RST输出激活。写入0进行清除。 0 = 复位未激活 1 = 曾激活复位	0	R/W
31:2	-	保留。请勿修改，读为逻辑 0。	0	-

13.4.4.6 PERIPHERAL_RESET 的复位外部状态寄存器

有关具有 PERIPH_RST 输出作为复位源的复位发生器，请参阅表 104。

表 118. 复位外部状态寄存器 x （RESET_EXT_STATx，地址 0x4005 34xx）位描述

位	符号	描述	复位值	访问类型
1:0	-	保留。请勿修改，读为逻辑 0。	0	-
2	PERIPHERAL_RESET	复位由 PERIPHERAL_RST 输出激活。写入 0 进行清除。 0 = 复位未激活 1 = 曾激活复位	0	R/W
31:3	-	保留。请勿修改，读为逻辑 0。	0	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

13.4.4.7 MASTER_RESET 的复位外部状态寄存器

有关具有MASTER_RST作为复位源的复位发生器，请参阅表 104。它们是ARM Cortex-M4内核、LCD 控制器、USB0、GPDMA、SDIO 控制器、外部存储控制器、以太网控制器。

复位值取决于外围设备，请参见表 104。

表 119. 复位外部状态寄存器 y （RESET_EXT_STATy，地址 0x4005 34yy）位描述

位	符号	描述	复位值	访问类型
2:0	-	保留。请勿修改，读为逻辑 0。	0	-
3	MASTER_RESET	复位由 MASTER_RST 输出激活。写入 0 进行清除。 0 = 复位未激活 1 = 曾激活复位	0	R/W
31:4	-	保留。请勿修改，读为逻辑 0。	0	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

14.1 本章导读

本章适用于所有器件。

14.2 引脚说明

在 LPC43xx 上，数字引脚将分组到 16 个引脚组，即 P0 到 P9 和 PA 到 PF，每组最多使用 20 个引脚。每个数字引脚可以支持多达 8 种不同的数字引脚功能，包括通用 I/O (GPIO)，可通过 SCU 引脚配置寄存器进行选择。一些数字引脚支持附加模拟功能，可通过 SCU 中的 ENAIO 寄存器进行选择。

注：请注意，引脚名称并不指代所分配的 GPIO 端口。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
多路复用数字引脚										
P0_0	L3	x	G2	47	32	22	[3]	I; PU	I/O	**GPIO0[0]** — 通用数字输入 / 输出引脚。
									I/O	**SSP1_MISO** — SSP1 主机输入、从机输出。
									I	**ENET_RXD1** — 以太网接收数据 1（RMII/MII 接口）。
									I/O	**SGPIO0** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**I2S0_TX_WS** — 发送字选择。由主机驱动，从机接收。相当于 *I²S 总线规范中的信号 WS*。
									I/O	**I2S1_TX_WS** — 发送字选择。由主机驱动，从机接收。相当于 *I²S 总线规范中的信号 WS*。
P0_1	M2	x	G1	50	34	23	[3]	I; PU	I/O	**GPIO0[1]** — 通用数字输入 / 输出引脚。
									I/O	**SSP1_MOSI** — SSP1 主机输出，从机输入。
									I	**ENET_COL** — 以太网冲突检测 （MII 接口）。
									I/O	**SGPIO1** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
										ENET_TX_EN — 以太网发送使能（RMII/MII 接口）。
									I/O	**I2S1_TX_SDA** — I2S 发送数据。由发送器驱动，接收器读取。相当于 *I²S 总线规范中的信号 SD*。
P1_0	P2	x	H1	54	38	25	[3]	I; PU	I/O	**GPIO0[4]** — 通用数字输入 / 输出引脚。
									I	**CTIN_3** — SCT 输入 3。定时器 1 的捕获输入 1。
									I/O	**EMC_A5** — 外部存储器地址线 5。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SSP0_SSEL** — SSP0 从机选择。
									I/O	**SGPIO7** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
P1_1	R2	x	K2	58	42	28	[3]	I; PU	I/O	**GPIO0[8]** — 通用数字输入 / 输出引脚。启动引脚（见表 16）。
									O	**CTOUT_7** — SCT 输出 7。定时器 1 的匹配输出 3。
									I/O	**EMC_A6** — 外部存储器地址线 6。
									I/O	**SGPIO8** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I/O	**SSP0_MISO** — SSP0 主机输入、从机输出。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P1_2	R3	x	K1	60	43	29	[3]	I; PU	I/O	**GPIO0[9]** — 通用数字输入 / 输出引脚。启动引脚（见表 16）。
									O	**CTOUT_6** — SCT 输出 6。定时器 1 的匹配输出 2。
									I/O	**EMC_A7** — 外部存储器地址线 7。
									I/O	**SGPIO9** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I/O	**SSP0_MOSI** — SSP0 主机输出，从机输入。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P1_3	P5	x	J1	61	44	30	[3]	I; PU	I/O	**GPIO0[10]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_8** — SCT 输出 8。定时器 2 的匹配输出 0。
									I/O	**SGPIO10** — 通用数字输入 / 输出引脚。
									O	**EMC_OE** — 低电平有效输出使能信号。
									O	**USB0_IND1** — USB0 端口 LED 指示灯控制输出 1。
									I/O	**SSP1_MISO** — SSP1 主机输入、从机输出。
									-	**R** — 保留功能。
									O	**SD_RST** — MMC4.4 卡 SD/MMC 复位信号。
P1_4	T3	x	J2	64	47	32	[3]	I; PU	I/O	**GPIO0[11]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_9** — SCT 输出 9。定时器 2 的匹配输出 1。
									I/O	**SGPIO11** — 通用数字输入 / 输出引脚。
									O	**EMC_BLS0** — 低电平有效“字节通道”选择信号 0。
									O	**USB0_IND0** — USB0 端口 LED 指示灯控制输出 0。
									I/O	**SSP1_MOSI** — SSP1 主机输出，从机输入。
									-	**R** — 保留功能。
									O	**SD_VOLT1** — SD/MMC 总线电压选择输出 1。
P1_5	R5	x	J4	65	48	33	[3]	I; PU	I/O	**GPIO1[8]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_10** — SCT 输出 10。定时器 2 的匹配输出 2。
									-	**R** — 保留功能。
									O	**EMC_CS0** — 低电平有效芯片选择 0 信号。
									O	**USB0_PWR_FAULT** — 指示过流状况的端口电源故障信号，该信号监控 USB 总线上的过流状况（检测过电流状况需要外部电路）。
									I/O	**SSP1_SSEL** — SSP1 从机选择。
									I/O	**SGPIO15** — 通用数字输入 / 输出引脚。
									O	**SD_POW** — SD/MMC 电源监控输出。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P1_6	T4	x	K4	67	49	34	[3]	I; PU	I/O	**GPIO1[9]** — 通用数字输入 / 输出引脚。
									I	**CTIN_5** — SCT 输入 5。定时器 2 的捕获输入 2。
									-	**R** — 保留功能。
									O	**EMC_WE** — 低电平有效写使能信号。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SGPIO14** — 通用数字输入 / 输出引脚。
									I/O	**SD_CMD** — SD/MMC 命令信号。
P1_7	T5	x	G4	69	50	35	[3]	I; PU	I/O	**GPIO1[0]** — 通用数字输入 / 输出引脚。
									I	**U1_DSR** — UART1 数据设置就绪输入。
									O	**CTOUT_13** — SCT 输出 13。定时器 3 的匹配输出 1。
									I/O	**EMC_D0** — 外部存储器数据线 0。
									O	**USB0_PPWR** — VBUS 驱动信号 （发送至外部充电泵或电源管理单元）；指示 VBUS 必须被驱动（有效高电平）。 复位时，添加一个下拉电阻以禁用电源开关。和其他恩智浦 LPC 器件使用的 USB_PPWR 相比，该信号具有相反的极性。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P1_8	R7	x	H5	71	51	36	[3]	I; PU	I/O	**GPIO1[1]** — 通用数字输入 / 输出引脚。
									O	**U1_DTR** — UART1 数据终结就绪输出。
									O	**CTOUT_12** — SCT 输出 12。定时器 3 的匹配输出 0。
									I/O	**EMC_D1** — 外部存储器数据线 1。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**SD_VOLT0** — SD/MMC 总线电压选择输出 0。
P1_9	T7	x	J5	73	52	37	[3]	I; PU	I/O	**GPIO1[2]** — 通用数字输入 / 输出引脚。
									O	**U1_RTS** — UART1 请求发送输出。
									O	**CTOUT_11** — SCT 输出 11。定时器 3 的匹配输出 2。
									I/O	**EMC_D2** — 外部存储器数据线 2。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SD_DAT0** — SD/MMC 数据总线 0。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态 [2]	类型	描述
P1_10	R8	x	H6	75	53	38	[3]	I; PU	I/O	**GPIO1[3]** — 通用数字输入 / 输出引脚。
									I	**U1_RI** — UART1 振铃指示器输入。
									O	**CTOUT_14** — SCT 输出 14。定时器 3 的匹配输出 2。
									I/O	**EMC_D3** — 外部存储器数据线 3。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SD_DAT1** — SD/MMC 数据总线 1。
P1_11	T9	x	J7	77	55	39	[3]	I; PU	I/O	**GPIO1[4]** — 通用数字输入 / 输出引脚。
									I	**U1_CTS** — UART1 准许发送输入。
									O	**CTOUT_15** — SCT 输出 15。定时器 3 的匹配输出 3。
									I/O	**EMC_D4** — 外部存储器数据线 4。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SD_DAT2** — SD/MMC 数据总线 2。
P1_12	R9	x	K7	78	56	40	[3]	I; PU	I/O	**GPIO1[5]** — 通用数字输入 / 输出引脚。
									I	**U1_DCD** — UART1 数据载波检测输入。
									-	**R** — 保留功能。
									I/O	**EMC_D5** — 外部存储器数据线 5。
									I	**T0_CAP1** — 定时器 1 的捕获输入 0。
									-	**R** — 保留功能。
									I/O	**SGPIO8** — 通用数字输入 / 输出引脚。
									I/O	**SD_DAT3** — SD/MMC 数据总线 3。
P1_13	R10	x	H8	83	60	41	[3]	I; PU	I/O	**GPIO1[6]** — 通用数字输入 / 输出引脚。
									O	**U1_TXD** — UART1 的发送器输出。
									-	**R** — 保留功能。
									I/O	**EMC_D6** — 外部存储器数据线 6。
									I	**T0_CAP0** — 定时器 0 的捕获输入 0。
									-	**R** — 保留功能。
									I/O	**SGPIO9** — 通用数字输入 / 输出引脚。
									I	**SD_CD** — SD/MMC 卡检测输入。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态 [2]	类型	描述
P1_14	R11	x	J8	85	61	42	[3]	I; PU	I/O	**GPIO1[7]** — 通用数字输入 / 输出引脚。
									I	**U1_RXD** — UART1 的接收器输入。
									-	**R** — 保留功能。
									I/O	**EMC_D7** — 外部存储器数据线 7。
									O	**T0_MAT2** — 定时器 0 的匹配输出 0。
									-	**R** — 保留功能。
									I/O	**SGPIO10** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
P1_15	T12	x	K8	87	62	43	[3]	I; PU	I/O	**GPIO0[2]** — 通用数字输入 / 输出引脚。
									O	**U2_TXD** — USART2 的发送器输出。
									I/O	**SGPIO2** — 通用数字输入 / 输出引脚。
									I	**ENET_RXD0** — 以太网接收数据 0（RMII/MII 接口）。
									O	**T0_MAT1** — 定时器 0 的匹配输出 0。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P1_16	M7	x	H9	90	64	44	[3]	I; PU	I/O	**GPIO0[3]** — 通用数字输入 / 输出引脚。
									I	**U2_RXD** — UART2 的接收器输入。
									I/O	**SGPIO3** — 通用数字输入 / 输出引脚。
									I	**ENET_CRS** — 以太网载波侦听 （MII 接口）。
									O	**T0_MAT0** — 定时器 0 的匹配输出 0。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I	**ENET_RX_DV** — 以太网接收数据有效 （RMII/MII 接口）。
P1_17	M8	x	H10	93	66	45	[4]	I; PU	I/O	**GPIO0[12]** — 通用数字输入 / 输出引脚。
									I/O	**U2_UCLK** — 同步模式下 USART2 的串行时钟输入 / 输出。
									-	**R** — 保留功能。
									I/O	**ENET_MDIO** — 以太网 MIIM 数据输入和输出。
									I	**T0_CAP3** — 定时器 3 的捕获输入 0。
									O	**CAN1_TD** — CAN1 发送器输出。
									I/O	**SGPIO11** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P1_18	N12	x	J10	95	67	46	[3]	I; PU	I/O	**GPIO0[13]** — 通用数字输入 / 输出引脚。
									I/O	**U2_DIR** — USART2 RS-485/EIA-485 输出使能 / 方向控制。
									-	**R** — 保留功能。
									O	**ENET_TXD0** — 以太网发送数据 0（RMII/MII 接口）。
									O	**T0_MAT3** — 定时器 3 的匹配输出 0。
									I	**CAN1_RD** — CAN1 接收器输入。
									I/O	**SGPIO12** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
P1_19	M11	x	K9	96	68	47	[3]	I; PU	I	**ENET_TX_CLK (ENET_REF_CLK)** — 以太网发送时钟（MII 接口）或以太网基准时钟（RMII 接口）。
									I/O	**SSP1_SCK** — SSP1 的串行时钟。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**CLKOUT** — 时钟输出引脚。
									-	**R** — 保留功能。
									O	**I2S0_RX_MCLK** — I2S 接收主机时钟。
									I/O	**I2S1_TX_SCK** — 发送时钟。由主机驱动，从机接收。相当于 I^2S 总线规范中的信号 SCK。
P1_20	M10	x	K10	100	70	48	[3]	I; PU	I/O	**GPIO0[15]** — 通用数字输入 / 输出引脚。
									I/O	**SSP1_SSEL** — SSP1 从机选择。
									-	**R** — 保留功能。
									O	**ENET_TXD1** — 以太网发送数据 1（RMII/MII 接口）。
									I	**T0_CAP2** — 定时器 2 的捕获输入 0。
									-	**R** — 保留功能。
									I/O	**SGPIO13** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P2_0	T16	x	G10	108	75	50	[3]	I; PU	I/O	**SGPIO4** — 通用数字输入 / 输出引脚。
									O	**U0_TXD** — USART0 的发送器输出。
									I/O	**EMC_A13** — 外部存储器地址线 13。
									O	**USB0_PPWR** — VBUS 驱动信号（发送至外部充电泵或电源管理单元）；指示 VBUS 必须被驱动（有效高电平）。 复位时，添加一个下拉电阻以禁用电源开关。和其他恩智浦 LPC 器件使用的 USB_PPWR 相比，该信号具有相反的极性。
									I/O	**GPIO5[0]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I	**T3_CAP0** — 定时器 0 的捕获输入 3。
									O	**ENET_MDC** — 以太网 MIIM 时钟。
P2_1	N15	x	G7	116	81	54	[3]	I; PU	I/O	**SGPIO5** — 通用数字输入 / 输出引脚。
									I	**U0_RXD** — UART0 的接收器输入。
									I/O	**EMC_A12** — 外部存储器地址线 12。
									O	**USB0_PWR_FAULT** — 指示过流状况的端口电源故障信号，该信号监控 USB 总线上的过流状况（检测过电流状况需要外部电路）。
									I/O	**GPIO5[1]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I	**T3_CAP1** — 定时器 1 的捕获输入 3。
									-	**R** — 保留功能。
P2_2	M15	x	F5	121	84	56	[3]	I; PU	I/O	**SGPIO6** — 通用数字输入 / 输出引脚。
									I/O	**U0_UCLK** — 同步模式下 USART0 的串行时钟输入 / 输出。
									I/O	**EMC_A11** — 外部存储器地址线 11。
									O	**USB0_IND1** — USB0 端口 LED 指示灯控制输出 1。
									I/O	**GPIO5[2]** — 通用数字输入 / 输出引脚。
									I	**CTIN_6** — SCT 输入 6。定时器 3 的捕获输入 1。
									I	**T3_CAP2** — 定时器 2 的捕获输入 3。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P2_3	J12	x	D8	127	87	57	[4]	I; PU	I/O	**SGPIO12** — 通用数字输入 / 输出引脚。
									I/O	**I2C1_SDA** — I²C1 数据输入 / 输出（此引脚不使用专门的 I²C 缓冲器）。
									O	**U3_TXD** — USART3 的发送器输出。
									I	**CTIN_1** — SCT 输入 1。定时器 0 的捕获输入 1。定时器 2 的捕获输入 1。
									I/O	**GPIO5[3]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									O	**T3_MAT0** — 定时器 3 的匹配输出 0。
									O	**USB0_PPWR** — VBUS 驱动信号（发送至外部充电泵或电源管理单元）；指示 VBUS 必须被驱动（有效高电平）。 复位时，添加一个下拉电阻以禁用电源开关。和其他恩智浦 LPC 器件使用的 USB_PPWR 相比，该信号具有相反的极性。
P2_4	K11	x	D9	128	88	58	[4]	I; PU	I/O	**SGPIO13** — 通用数字输入 / 输出引脚。
									I/O	**I2C1_SCL** — I²C1 时钟输入 / 输出（此引脚不使用专门的 I²C 缓冲器）。
									I	**U3_RXD** — UART3 的接收器输入。
									I	**CTIN_0** — SCT 输入 0。定时器 0、1、2、3 的捕获输入 0。
									I/O	**GPIO5[4]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									O	**T3_MAT1** — 定时器 3 的匹配输出 1。
									O	**USB0_PWR_FAULT** — 指示过流状况的端口电源故障信号，该信号监控 USB 总线上的过流状况（检测过电流状况需要外部电路）。
P2_5	K14	x	D10	131	91	61	[4]	I; PU	I/O	**SGPIO14** — 通用数字输入 / 输出引脚。
									I	**CTIN_2** — SCT 输入 2。定时器 0 的捕获输入 2。
									I	**USB1_VBUS** — 监控是否存在 USB1 总线供电。 **注：**要进行 USB 复位，该信号必须为高电平。
									I	**ADCTRIG1** — ADC 启动器输入 1。
									I/O	**GPIO5[5]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									O	**T3_MAT2** — 定时器 3 的匹配输出 2。
									O	**USB0_IND0** — USB0 端口 LED 指示灯控制输出 0。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P2_6	K16	x	G9	137	95	64	[3]	I; PU	I/O	**SGPIO7** — 通用数字输入 / 输出引脚。
									I/O	**U0_DIR** — USART0 RS-485/EIA-485 输出使能 / 方向控制。
									I/O	**EMC_A10** — 外部存储器地址线 10。
									O	**USB0_IND0** — USB0 端口 LED 指示灯控制输出 0。
									I/O	**GPIO5[6]** — 通用数字输入 / 输出引脚。
									I	**CTIN_7** — SCT 输入 7。
									I	**T3_CAP3** — 定时器 3 的捕获输入 3。
									-	**R** — 保留功能。
P2_7	H14	x	C10	138	96	65	[3]	I; PU	I/O	**GPIO0[7]** — 通用数字输入 / 输出引脚。如果该引脚在复位时被拉低，则器件使用 USART0 进入 ISP 模式。
									O	**CTOUT_1** — SCT 输出 1。定时器 0 的匹配输出 1。
									I/O	**U3_UCLK** — 同步模式下 USART3 的串行时钟输入 / 输出。
									I/O	**EMC_A9** — 外部存储器地址线 9。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**T3_MAT3** — 定时器 3 的匹配输出 3。
									-	**R** — 保留功能。
P2_8	J16	x	C6	140	98	67	[3]	I; PU	I/O	**SGPIO15** — 通用数字输入 / 输出引脚。启动引脚（见表 16）。
									O	**CTOUT_0** — SCT 输出 0。定时器 0 的匹配输出 2。
									I/O	**U3_DIR** — USART3 RS-485/EIA-485 输出使能 / 方向控制。
									I/O	**EMC_A8** — 外部存储器地址线 8。
									I/O	**GPIO5[7]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P2_9	H16	x	B10	144	102	70	[3]	I; PU	I/O	**GPIO1[10]** — 通用数字输入/输出引脚。启动引脚（见表 16）。
									O	**CTOUT_3** — SCT 输出 3。定时器 0 的匹配输出 3。
									I/O	**U3_BAUD** — USART3 波特引脚。
									I/O	**EMC_A0** — 外部存储器地址线 0。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P2_10	G16	x	E8	146	104	71	[3]	I; PU	I/O	**GPIO0[14]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_2** — SCT 输出 2。定时器 0 的匹配输出 2。
									O	**U2_TXD** — USART2 的发送器输出。
									I/O	**EMC_A1** — 外部存储器地址线 1。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P2_11	F16	x	A9	148	105	72	[3]	I; PU	I/O	**GPIO1[11]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_5** — SCT 输出 5。定时器 1 的匹配输出 1。
									I	**U2_RXD** — UART2 的接收器输入。
									I/O	**EMC_A2** — 外部存储器地址线 2。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P2_12	E15	x	B9	153	106	73	[3]	I; PU	I/O	**GPIO1[12]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_4** — SCT 输出 4。定时器 1 的匹配输出 0。
									-	**R** — 保留功能。
									I/O	**EMC_A3** — 外部存储器地址线 3。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**U2_UCLK** — 同步模式下 USART2 的串行时钟输入 / 输出。
P2_13	C16	x	A10	156	108	75	[3]	I; PU	I/O	**GPIO1[13]** — 通用数字输入 / 输出引脚。
									I	**CTIN_4** — SCT 输入 4。定时器 1 的捕获输入 2。
									-	**R** — 保留功能。
									I/O	**EMC_A4** — 外部存储器地址线 4。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**U2_DIR** — USART2 RS-485/EIA-485 输出使能 / 方向控制。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P3_0	F13	x	A8	161	112	78	[3]	I; PU	I/O	**I2S0_RX_SCK** — 发送字选择。由主机驱动，从机接收。相当于 *I²S 总线规范中的信号 SCK*。
									O	**I2S0_RX_MCLK** — I2S 接收主机时钟。
									I/O	**I2S0_TX_SCK** — 发送时钟。由主机驱动，从机接收。相当于 I²S 总线规范中的信号 SCK。
									O	**I2S0_TX_MCLK** — I2S 发送主机时钟。
									I/O	**SSP0_SCK** — SSP0 的串行时钟。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P3_1	G11	x	F7	163	114	79	[3]	I; PU	I/O	**I2S0_TX_WS** — 发送字选择。由主机驱动，从机接收。相当于 *I²S 总线规范中的信号 WS*。
									I/O	**I2S0_RX_WS** — 接收字选择。由主机驱动，从机接收。相当于 *I²S 总线规范中的信号 WS*。
									I	**CAN0_RD** — CAN 接收器输入。
									O	**USB1_IND1** — USB1 端口 LED 指示灯控制输出 1。
									I/O	**GPIO5[8]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									O	**LCD_VD15** — LCD 数据。
									-	**R** — 保留功能。
P3_2	F11	x	G6	166	116	80	[3]	I; PU	I/O	**I2S0_TX_SDA** — I2S1 传输数据。由发送器驱动，接收器读取。相当于 *I²S 总线规范中的信号 SD*。
									I/O	**I2S0_RX_SDA** — I2S 接收数据。由发送器驱动，接收器读取。相当于 *I²S 总线规范中的信号 SD*。
									O	**CAN0_TD** — CAN 发送器输出。
									O	**USB1_IND0** — USB1 端口 LED 指示灯控制输出 0。
									I/O	**GPIO5[9]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									O	**LCD_VD14** — LCD 数据。
									-	**R** — 保留功能。
P3_3	B14	x	A7	169	118	81	[5]	I; PU	-	**R** — 保留功能。
									I/O	**SPI_SCK** — SPI 的串行时钟。
									I/O	**SSP0_SCK** — SSP0 的串行时钟。
									O	**SPIFI_SCK** — SPIFI 的串行时钟。
									O	**CGU_OUT1** — CGU 备用时钟输出 1。
									-	**R** — 保留功能。
									O	**I2S0_TX_MCLK** — I2S 发送主机时钟。
									I/O	**I2S1_TX_SCK** — 发送时钟。由主机驱动，从机接收。相当于 I²S 总线规范中的信号 SCK。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P3_4	A15	x	B8	171	119	82	[3]	I; PU	I/O	**GPIO1[14]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SPIFI_SIO3** — SPIFI 的 I/O 通道 3。
									O	**U1_TXD** — UART 1 的发送器输出。
									I/O	**I2S0_TX_WS** — 发送字选择。由主机驱动，从机接收。相当于 *I²S 总线规范中的信号 WS*。
									I/O	**I2S1_RX_SDA** — I2S 接收数据。由发送器驱动，接收器读取。相当于 *I²S 总线规范中的信号 SD*。
									O	**LCD_VD13** — LCD 数据。
P3_5	C12	x	B7	173	121	84	[3]	I; PU	I/O	**GPIO1[15]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SPIFI_SIO2** — SPIFI 的 I/O 通道 2。
									I	**U1_RXD** — UART 1 的接收器输入。
									I/O	**I2S0_TX_SDA** — I2S 发送数据。由发送器驱动，接收器读取。相当于 *I²S 总线规范中的信号 SD*。
									I/O	**I2S1_RX_WS** — 接收字选择。由主机驱动，从机接收。相当于 *I²S 总线规范中的信号 WS*。
									O	**LCD_VD12** — LCD 数据。
P3_6	B13	x	C7	174	122	85	[3]	I; PU	I/O	**GPIO0[6]** — 通用数字输入 / 输出引脚。
									I/O	**SPI_MISO** — SPI 主机输入、从机输出。
									I/O	**SSP0_SSEL** — SSP0 从机选择。
									I/O	**SPIFI_MISO** — SPIFI 四通道模式下的输入 1； SPIFI 输出 IO1。
									-	**R** — 保留功能。
									I/O	**SSP0_MISO** — SSP0 主机输入、从机输出。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P3_7	C11	x	D7	176	123	86	[3]	I; PU	-	**R** — 保留功能。
									I/O	**SPI_MOSI** — SPI 主机输出、从机输入。
									I/O	**SSP0_MISO** — SSP0 主机输入、从机输出。
									I/O	**SPIFI_MOSI** — SPIFI 四通道模式下的输入 I0； SPIFI 输出 IO0。
									I/O	**GPIO5[10]** — 通用数字输入 / 输出引脚。
									I/O	**SSP0_MOSI** — SSP0 主机输出，从机输入。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态 [2]	类型	描述
P3_8	C10	x	E7	179	124	87	[3]	I; PU	-	**R** — 保留功能。
									I	**SPI_SSEL** — SPI 从机选择。请注意，该引脚仅为输入引脚。 SPI 在主机模式下不能驱动从机上的 CS 输入。主机模式下，可通过任何 GPIO 引脚为 SPI 选择芯片。
									I/O	**SSP0_MOSI** — SSP0 主机输出，从机输入。
									I/O	**SPIFI_CS** — SPIFI 串行闪存芯片选择。
									I/O	**GPIO5[11]** — 通用数字输入 / 输出引脚。
									I/O	**SSP0_SSEL** — SSP0 从机选择。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P4_0	D5	x	-	1	1	-	[3]	I; PU	I/O	**GPIO2[0]** — 通用数字输入 / 输出引脚。
									O	**MCOA0** — 电机控制 PWM 通道 0，输出 A。
									I	**NMI** — NMI 外部中断输入。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**LCD_VD13** — LCD 数据。
									I/O	**U3_UCLK** — 同步模式下 USART3 的串行时钟输入 / 输出。
									-	**R** — 保留功能。
P4_1	A1	x	-	3	3	-	[6]	I; PU	I/O	**GPIO2[1]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_1** — SCT 输出 1。定时器 0 的匹配输出 1。
									O	**LCD_VD0** — LCD 数据。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**LCD_VD19** — LCD 数据。
									O	**U3_TXD** — USART3 的发送器输出。
									I	**ENET_COL** — 以太网冲突检测 （MII 接口）。
									I	**ADC0_1** — ADC0，输入通道 1。
P4_2	D3	x	-	12	8	-	[3]	I; PU	I/O	**GPIO2[2]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_0** — SCT 输出 0。定时器 0 的匹配输出 0。
									O	**LCD_VD3** — LCD 数据。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**LCD_VD12** — LCD 数据。
									I	**U3_RXD** — UART3 的接收器输入。
									I/O	**SGPIO8** — 通用数字输入 / 输出引脚。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P4_3	C2	x	-	10	7	-	[6]	I; PU	I/O	**GPIO2[3]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_3** — SCT 输出 3。定时器 0 的匹配输出 3。
									O	**LCD_VD2** — LCD 数据。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**LCD_VD21** — LCD 数据。
									I/O	**U3_BAUD** — USART3 波特引脚。
									I/O	**SGPIO9** — 通用数字输入 / 输出引脚。
									I	**ADC0_0** — ADC0，输入通道 0。
P4_4	B1	x	-	14	9	-	[6]	I; PU	I/O	**GPIO2[4]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_2** — SCT 输出 2。定时器 0 的匹配输出 2。
									O	**LCD_VD1** — LCD 数据。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**LCD_VD20** — LCD 数据。
									I/O	**U3_DIR** — USART3 RS-485/EIA-485 输出使能 / 方向控制。
									I/O	**SGPIO10** — 通用数字输入 / 输出引脚。
									O	**DAC** — DAC 输出。
P4_5	D2	x	-	15	10	-	[3]	I; PU	I/O	**GPIO2[5]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_5** — SCT 输出 5。定时器 1 的匹配输出 1。
									O	**LCD_FP** — 帧脉冲 (STN)。垂直同步脉冲 (TFT)。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SGPIO11** — 通用数字输入 / 输出引脚。
P4_6	C1	x	-	17	11	-	[3]	I; PU	I/O	**GPIO2[6]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_4** — SCT 输出 4。定时器 1 的匹配输出 0。
									O	**LCD_ENAB/LCDM** — STN 交流偏置驱动或 TFT 数据使能输入。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SGPIO12** — 通用数字输入 / 输出引脚。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P4_7	H4	x	-	21	14	-	[3]	O; PU	O	**LCD_DCLK** — LCD 面板时钟。
									I	**GP_CLKIN** — CGU 通用时钟输入。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**I2S1_TX_SCK** — 发送时钟。由主机驱动，从机接收。相当于 I^2S 总线规范中的信号 SCK。
									I/O	**I2S0_TX_SCK** — 发送时钟。由主机驱动，从机接收。相当于 I^2S 总线规范中的信号 SCK。
P4_8	E2	x	-	23	15	-	[3]	I; PU	-	**R** — 保留功能。
									I	**CTIN_5** — SCT 输入 5。定时器 2 的捕获输入 2。
									O	**LCD_VD9** — LCD 数据。
									-	**R** — 保留功能。
									I/O	**GPIO5[12]** — 通用数字输入 / 输出引脚。
									O	**LCD_VD22** — LCD 数据。
									O	**CAN1_TD** — CAN1 发送器输出。
									I/O	**SGPIO13** — 通用数字输入 / 输出引脚。
P4_9	L2	x	-	48	33	-	[3]	I; PU	-	**R** — 保留功能。
									I	**CTIN_6** — SCT 输入 6。定时器 1 的捕获输入 1。
									O	**LCD_VD11** — LCD 数据。
									-	**R** — 保留功能。
									I/O	**GPIO5[13]** — 通用数字输入 / 输出引脚。
									O	**LCD_VD15** — LCD 数据。
									I	**CAN1_RD** — CAN1 接收器输入。
									I/O	**SGPIO14** — 通用数字输入 / 输出引脚。
P4_10	M3	x	-	51	35	-	[3]	I; PU	-	**R** — 保留功能。
									I	**CTIN_2** — SCT 输入 2。定时器 0 的捕获输入 2。
									O	**LCD_VD10** — LCD 数据。
									-	**R** — 保留功能。
									I/O	**GPIO5[14]** — 通用数字输入 / 输出引脚。
									O	**LCD_VD14** — LCD 数据。
									-	**R** — 保留功能。
									I/O	**SGPIO15** — 通用数字输入 / 输出引脚。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P5_0	N3	x	-	53	37	-	[3]	I; PU	I/O	**GPIO2[9]** — 通用数字输入 / 输出引脚。
									O	**MCOB2** — 电机控制 PWM 通道 2，输出 B。
									I/O	**EMC_D12** — 外部存储器数据线 12。
									-	**R** — 保留功能。
									I	**U1_DSR** — UART 1 数据设置就绪输入。
									I	**T1_CAP0** — 定时器 0 的捕获输入 1。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P5_1	P3	x	-	55	39	-	[3]	I; PU	I/O	**GPIO2[10]** — 通用数字输入 / 输出引脚。
									I	**MCI2** — 电机控制 PWM 通道 2，输入。
									I/O	**EMC_D13** — 外部存储器数据线 13。
									-	**R** — 保留功能。
									O	**U1_DTR** — UART 1 的数据终结就绪输出。也可以配置为 UART 1 的 RS-485/EIA-485 输出使能信号。
									I	**T1_CAP1** — 定时器 1 的捕获输入 1。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P5_2	R4	x	-	63	46	-	[3]	I; PU	I/O	**GPIO2[11]** — 通用数字输入 / 输出引脚。
									I	**MCI1** — 电机控制 PWM 通道 1，输入。
									I/O	**EMC_D14** — 外部存储器数据线 14。
									-	**R** — 保留功能。
									O	**U1_RTS** — UART 1 的请求发送输出。也可以配置为 UART 1 的 RS-485/EIA-485 输出使能信号。
									I	**T1_CAP2** — 定时器 2 的捕获输入 1。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P5_3	T8	x	-	76	54	-	[3]	I; PU	I/O	**GPIO2[12]** — 通用数字输入 / 输出引脚。
									I	**MCI0** — 电机控制 PWM 通道 0，输入。
									I/O	**EMC_D15** — 外部存储器数据线 15。
									-	**R** — 保留功能。
									I	**U1_RI** — UART 1 的振铃指示器输入。
									I	**T1_CAP3** — 定时器 3 的捕获输入 1。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P5_4	P9	x	-	80	57	-	[3]	I; PU	I/O	**GPIO2[13]** — 通用数字输入 / 输出引脚。
									O	**MCOB0** — 电机控制 PWM 通道 0，输出 B。
									I/O	**EMC_D8** — 外部存储器数据线 8。
									-	**R** — 保留功能。
									I	**U1_CTS** — UART 1 的准许发送输入。
									O	**T1_MAT0** — 定时器 1 的匹配输出 0。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P5_5	P10	x	-	81	58	-	[3]	I; PU	I/O	**GPIO2[14]** — 通用数字输入 / 输出引脚。
									O	**MCOA1** — 电机控制 PWM 通道 1，输出 A。
									I/O	**EMC_D9** — 外部存储器数据线 9。
									-	**R** — 保留功能。
									I	**U1_DCD** — UART 1 的数据载波检测输入。
									O	**T1_MAT1** — 定时器 1 的匹配输出 1。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P5_6	T13	x	-	89	63	-	[3]	I; PU	I/O	**GPIO2[15]** — 通用数字输入 / 输出引脚。
									O	**MCOB1** — 电机控制 PWM 通道 1，输出 B。
									I/O	**EMC_D10** — 外部存储器数据线 10。
									-	**R** — 保留功能。
									O	**U1_TXD** — UART 1 的发送器输出。
									O	**T1_MAT2** — 定时器 1 的匹配输出 2。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P5_7	R12	x	-	91	65	-	[3]	I; PU	I/O	**GPIO2[7]** — 通用数字输入 / 输出引脚。
									O	**MCOA2** — 电机控制 PWM 通道 2，输出 A。
									I/O	**EMC_D11** — 外部存储器数据线 11。
									-	**R** — 保留功能。
									I	**U1_RXD** — UART 1 的接收器输入。
									O	**T1_MAT3** — 定时器 3 的匹配输出 3。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P6_0	M12	x	H7	105	73	-	[3]	I; PU	-	**R** — 保留功能。
									O	**I2S0_RX_MCLK** — I2S 接收主机时钟。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**I2S0_RX_SCK** — 接收时钟。由主机驱动，从机接收。相当于 *I²S 总线规范中的信号 SCK*。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P6_1	R15	x	G5	107	74	-	[3]	I; PU	I/O	**GPIO3[0]** — 通用数字输入 / 输出引脚。
									O	**EMC_DYCS1** — SDRAM 芯片选择 1。
									I/O	**U0_UCLK** — 同步模式下 USART0 的串行时钟输入 / 输出。
									I/O	**I2S0_RX_WS** — 接收字选择。由主机驱动，从机接收。相当于 *I²S 总线规范中的信号 WS*。
									-	**R** — 保留功能。
									I	**T2_CAP0** — 定时器 2 的捕获输入 2。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P6_2	L13	x	J9	111	78	-	[3]	I; PU	I/O	**GPIO3[1]** — 通用数字输入 / 输出引脚。
									O	**EMC_CKEOUT1** — SDRAM 时钟使能 1。
									I/O	**U0_DIR** — USART0 RS-485/EIA-485 输出使能 / 方向控制。
									I/O	**I2S0_RX_SDA** — I2S 接收数据。由发送器驱动，接收器读取。相当于 *I²S 总线规范中的信号 SD*。
									-	**R** — 保留功能。
									I	**T2_CAP1** — 定时器 1 的捕获输入 2。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P6_3	P15	x	-	113	79	-	[3]	I; PU	I/O	**GPIO3[2]** — 通用数字输入 / 输出引脚。
									O	**USB0_PPWR** — VBUS 驱动信号（发送至外部充电泵或电源管理单元）；指示VBUS信号必须被驱动（有效高电平）。 复位时，添加一个下拉电阻以禁用电源开关。和其他恩智浦 LPC 器件使用的 USB_PPWR 相比，该信号具有相反的极性。
									I/O	**SGPIO4** — 通用数字输入 / 输出引脚。
									O	**EMC_CS1** — 低电平有效芯片选择 1 信号。
									-	**R** — 保留功能。
									I	**T2_CAP2** — 定时器 2 的捕获输入 2。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P6_4	R16	x	F6	114	80	53	[3]	I; PU	I/O	**GPIO3[3]** — 通用数字输入 / 输出引脚。
									I	**CTIN_6** — SCT 输入 6。定时器 1 的捕获输入 1。
									O	**U0_TXD** — USART0 的发送器输出。
									O	**EMC_CAS** — 低电平有效 SDRAM 列地址选通。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P6_5	P16	x	F9	117	82	55	[3]	I; PU	I/O	**GPIO3[4]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_6** — SCT 输出 6。定时器 1 的匹配输出 2。
									I	**U0_RXD** — UART0 的接收器输入。
									O	**EMC_RAS** — 低电平有效 SDRAM 行地址选通。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P6_6	L14	x	-	119	83	-	[3]	I; PU	I/O	**GPIO0[5]** — 通用数字输入 / 输出引脚。
									O	**EMC_BLS1** — 低电平有效“字节通道”选择信号 1。
									I/O	**SGPIO5** — 通用数字输入 / 输出引脚。
									O	**USB0_PWR_FAULT** — 指示过流状况的端口电源故障信号，该信号监控 USB 总线上的过流状况（检测过电流状况需要外部电路）。
									-	**R** — 保留功能。
									I	**T2_CAP3** — 定时器 3 的捕获输入 2。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态 [2]	类型	描述
P6_7	J13	x	-	123	85	-	[3]	I; PU	-	**R** — 保留功能。
									I/O	**EMC_A15** — 外部存储器地址线 15。
									I/O	**SGPIO6** — 通用数字输入 / 输出引脚。
									O	**USB0_IND1** — USB0 端口 LED 指示灯控制输出 1。
									I/O	**GPIO5[15]** — 通用数字输入 / 输出引脚。
									O	**T2_MAT0** — 定时器 2 的匹配输出 0。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P6_8	H13	x	-	125	86	-	[3]	I; PU	-	**R** — 保留功能。
									I/O	**EMC_A14** — 外部存储器地址线 14。
									I/O	**SGPIO7** — 通用数字输入 / 输出引脚。
									O	**USB0_IND0** — USB0 端口 LED 指示灯控制输出 0。
									I/O	**GPIO5[16]** — 通用数字输入 / 输出引脚。
									O	**T2_MAT1** — 定时器 2 的匹配输出 1。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P6_9	J15	x	F8	139	97	66	[3]	I; PU	I/O	**GPIO3[5]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**$\overline{\text{EMC_DYCS0}}$** — SDRAM 芯片选择 0。
									-	**R** — 保留功能。
									O	**T2_MAT2** — 定时器 2 的匹配输出 2。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P6_10	H15	x	-	142	100	-	[3]	I; PU	I/O	**GPIO3[6]** — 通用数字输入 / 输出引脚。
									O	**$\overline{\text{MCABORT}}$** — 电机控制 PWM，低电平有效紧急停机。
									-	**R** — 保留功能。
									O	**EMC_DQMOUT1** — SDRAM 和静态器件使用的数据屏蔽 1。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P6_11	H12	x	C9	143	101	69	[3]	I; PU	I/O	**GPIO3[7]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**EMC_CKEOUT0** — SDRAM 时钟使能 0。
									-	**R** — 保留功能。
									O	**T2_MAT3** — 定时器 3 的匹配输出 3。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P6_12	G15	x	-	145	103	-	[3]	I; PU	I/O	**GPIO2[8]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_7** — SCT 输出 7。定时器 1 的匹配输出 3。
									-	**R** — 保留功能。
									O	**EMC_DQMOUT0** — SDRAM 和静态器件使用的数据屏蔽 0。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P7_0	B16	x	-	158	110	-	[3]	I; PU	I/O	**GPIO3[8]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_14** — SCT 输出 14。定时器 3 的匹配输出 2。
									-	**R** — 保留功能。
									O	**LCD_LE** — 行结束信号。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SGPIO4** — 通用数字输入 / 输出引脚。
P7_1	C14	x	-	162	113	-	[3]	I; PU	I/O	**GPIO3[9]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_15** — SCT 输出 15。定时器 3 的匹配输出 3。
									I/O	**I2S0_TX_WS** — 发送字选择。由主机驱动，从机接收。相当于 *I^2S 总线规范中的信号 WS*。
									O	**LCD_VD19** — LCD 数据。
									O	**LCD_VD7** — LCD 数据。
									-	**R** — 保留功能。
									O	**U2_TXD** — USART2 的发送器输出。
									I/O	**SGPIO5** — 通用数字输入 / 输出引脚。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P7_2	A16	x	-	165	115	-	[3]	I; PU	I/O	**GPIO3[10]** — 通用数字输入 / 输出引脚。
									I	**CTIN_4** — SCT 输入 4。定时器 1 的捕获输入 2。
									I/O	**I2S0_TX_SDA** — I2S 发送数据。由发送器驱动，接收器读取。相当于 *I^2S 总线规范中的信号 SD*。
									O	**LCD_VD18** — LCD 数据。
									O	**LCD_VD6** — LCD 数据。
									-	**R** — 保留功能。
									I	**U2_RXD** — UART2 的接收器输入。
									I/O	**SGPIO6** — 通用数字输入 / 输出引脚。
P7_3	C13	x	-	167	117	-	[3]	I; PU	I/O	**GPIO3[11]** — 通用数字输入 / 输出引脚。
									I	**CTIN_3** — SCT 输入 3。定时器 1 的捕获输入 1。
									-	**R** — 保留功能。
									O	**LCD_VD17** — LCD 数据。
									O	**LCD_VD5** — LCD 数据。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P7_4	C8	x	-	189	132	-	[6]	I; PU	I/O	**GPIO3[12]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_13** — SCT 输出 13。定时器 3 的匹配输出 1。
									-	**R** — 保留功能。
									O	**LCD_VD16** — LCD 数据。
									O	**LCD_VD4** — LCD 数据。
									O	**TRACEDATA[0]** — 跟踪数据、位 0。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I	**ADC0_4** — ADC0，输入通道 4。
P7_5	A7	x	-	191	133	-	[6]	I; PU	I/O	**GPIO3[13]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_12** — SCT 输出 12。定时器 3 的匹配输出 3。
									-	**R** — 保留功能。
									O	**LCD_VD8** — LCD 数据。
									O	**LCD_VD23** — LCD 数据。
									O	**TRACEDATA[1]** — 跟踪数据、位 1。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I	**ADC0_3** — ADC0，输入通道 3。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P7_6	C7	x	-	194	134	-	[3]	I; PU	I/O	**GPIO3[14]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_11** — SCT 输出 1。定时器 2 的匹配输出 3。
									-	**R** — 保留功能。
									O	**LCD_LP** — 行同步脉冲 (STN)。水平同步脉冲 (TFT)。
									-	**R** — 保留功能。
									O	**TRACEDATA[2]** — 跟踪数据、位 2。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
P7_7	B6	x	-	201	140	-	[6]	I; PU	I/O	**GPIO3[15]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_8** — SCT 输出 8。定时器 2 的匹配输出 0。
									-	**R** — 保留功能。
									O	**LCD_PWR** — LCD 面板电源使能。
									-	**R** — 保留功能。
									O	**TRACEDATA[3]** — 跟踪数据、位 3。
									O	**ENET_MDC** — 以太网 MIIM 时钟。
									I/O	**SGPIO7** — 通用数字输入 / 输出引脚。
									I	**ADC1_6** — ADC1，输入通道 6。
P8_0	E5	x	-	2	-	-	[4]	I; PU	I/O	**GPIO4[0]** — 通用数字输入 / 输出引脚。
									O	**USB0_PWR_FAULT** — 指示过流状况的端口电源故障信号，该信号监控 USB 总线上的过流状况（检测过电流状况需要外部电路）。
									-	**R** — 保留功能。
									I	**MCI2** — 电机控制 PWM 通道 2，输入。
									I/O	**SGPIO8** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**T0_MAT0** — 定时器 0 的匹配输出 1。
P8_1	H5	x	-	34	-	-	[4]	I; PU	I/O	**GPIO4[1]** — 通用数字输入 / 输出引脚。
									O	**USB0_IND1** — USB0 端口 LED 指示灯控制输出 1。
									-	**R** — 保留功能。
									I	**MCI1** — 电机控制 PWM 通道 1，输入。
									I/O	**SGPIO9** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**T0_MAT1** — 定时器 0 的匹配输出 1。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P8_2	K4	x	-	36	-	-	[4]	I; PU	I/O	**GPIO4[2]** — 通用数字输入 / 输出引脚。
									O	**USB0_IND0** — USB0 端口 LED 指示灯控制输出 0。
									-	**R** — 保留功能。
									I	**MCI0** — 电机控制 PWM 通道 0，输入。
									I/O	**SGPIO10** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**T0_MAT2** — 定时器 0 的匹配输出 2。
P8_3	J3	x	-	37	-	-	[3]	I; PU	I/O	**GPIO4[3]** — 通用数字输入 / 输出引脚。
									I/O	**USB1_ULPI_D2** — ULPI 链接双向数据线 2。
									-	**R** — 保留功能。
									O	**LCD_VD12** — LCD 数据。
									O	**LCD_VD19** — LCD 数据。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**T0_MAT3** — 定时器 3 的匹配输出 0。
P8_4	J2	x	-	39	-	-	[3]	I; PU	I/O	**GPIO4[4]** — 通用数字输入 / 输出引脚。
									I/O	**USB1_ULPI_D1** — ULPI 链接双向数据线 1。
									-	**R** — 保留功能。
									O	**LCD_VD7** — LCD 数据。
									O	**LCD_VD16** — LCD 数据。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I	**T0_CAP0** — 定时器 0 的捕获输入 0。
P8_5	J1	x	-	40	-	-	[3]	I; PU	I/O	**GPIO4[5]** — 通用数字输入 / 输出引脚。
									I/O	**USB1_ULPI_D0** — ULPI 链接双向数据线 0。
									-	**R** — 保留功能。
									O	**LCD_VD6** — LCD 数据。
									O	**LCD_VD8** — LCD 数据。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I	**T0_CAP1** — 定时器 1 的捕获输入 0。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P8_6	K3	x	-	43	-	-	[3]	I; PU	I/O	**GPIO4[6]** — 通用数字输入 / 输出引脚。
									I	**USB1_ULPI_NXT** — ULPI 链接 NXT 信号。来自 PHY 的数据流控制信号。
									-	**R** — 保留功能。
									O	**LCD_VD5** — LCD 数据。
									O	**LCD_LP** — 行同步脉冲 (STN)。水平同步脉冲 (TFT)。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I	**T0_CAP2** — 定时器 2 的捕获输入 0。
P8_7	K1	x	-	45	-	-	[3]	I; PU	I/O	**GPIO4[7]** — 通用数字输入 / 输出引脚。
									O	**USB1_ULPI_STP** — ULPI 链接 STP 信号。断言结束或中断到 PHY 的传输。
									-	**R** — 保留功能。
									O	**LCD_VD4** — LCD 数据。
									O	**LCD_PWR** — LCD 面板电源使能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I	**T0_CAP3** — 定时器 3 的捕获输入 0。
P8_8	L1	x	-	49	-	-	[3]	I; PU	-	**R** — 保留功能。
									I	**USB1_ULPI_CLK** — ULPI 链接 CLK 信号。 PHY 生成的 60 MHz 时钟。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**CGU_OUT0** — CGU 备用时钟输出 0。
									O	**I2S1_TX_MCLK** — I2S1 发送主机时钟。
P9_0	T1	x	-	59	-	-	[3]	I; PU	I/O	**GPIO4[12]** — 通用数字输入 / 输出引脚。
									O	**$\overline{\text{MCABORT}}$** — 电机控制 PWM，低电平有效紧急停机。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I	**ENET_CRS** — 以太网载波侦听 （MII 接口）。
									I/O	**SGPIO0** — 通用数字输入 / 输出引脚。
									I/O	**SSP0_SSEL** — SSP0 从机选择。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P9_1	N6	x	-	66	-	-	[3]	I; PU	I/O	**GPIO4[13]** — 通用数字输入 / 输出引脚。
									O	**MCOA2** — 电机控制 PWM 通道 2，输出 A。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**I2S0_TX_WS** — 发送字选择。由主机驱动，从机接收。相当于 *I²S 总线规范中的信号 WS*。
									I	**ENET_RX_ER** — 以太网接收错误 （MII 接口）。
									I/O	**SGPIO1** — 通用数字输入 / 输出引脚。
									I/O	**SSP0_MISO** — SSP0 主机输入、从机输出。
P9_2	N8	x	-	70	-	-	[3]	I; PU	I/O	**GPIO4[14]** — 通用数字输入 / 输出引脚。
									O	**MCOB2** — 电机控制 PWM 通道 2，输出 B。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**I2S0_TX_SDA** — I2S 发送数据。由发送器驱动，接收器读取。相当于 *I²S 总线规范中的信号 SD*。
									I	**ENET_RXD3** — 以太网接收数据 3 （MII 接口）。
									I/O	**SGPIO2** — 通用数字输入 / 输出引脚。
									I/O	**SSP0_MOSI** — SSP0 主机输出，从机输入。
P9_3	M6	x	-	79	-	-	[3]	I; PU	I/O	**GPIO4[15]** — 通用数字输入 / 输出引脚。
									O	**MCOA0** — 电机控制 PWM 通道 0，输出 A。
									O	**USB1_IND1** — USB1 端口 LED 指示灯控制输出 1。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I	**ENET_RXD2** — 以太网接收数据 2 （MII 接口）。
									I/O	**SGPIO9** — 通用数字输入 / 输出引脚。
									O	**U3_TXD** — USART3 的发送器输出。
P9_4	N10	x	-	92	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**MCOB0** — 电机控制 PWM 通道 0，输出 B。
									O	**USB1_IND0** — USB1 端口 LED 指示灯控制输出 0。
									-	**R** — 保留功能。
									I/O	**GPIO5[17]** — 通用数字输入 / 输出引脚。
									O	**ENET_TXD2** — 以太网发送数据 2 （MII 接口）。
									I/O	**SGPIO4** — 通用数字输入 / 输出引脚。
									I	**U3_RXD** — UART3 的接收器输入。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
P9_5	M9	x	-	98	69	-	[3]	I; PU	-	**R** — 保留功能。
									O	**MCOA1** — 电机控制 PWM 通道 1，输出 A。
									O	**USB1_PPWR** — VBUS 驱动信号（发送至外部充电泵或电源管理单元）；指示 VBUS 必须被驱动（有效高电平）。 复位时，添加一个下拉电阻以禁用电源开关。和其他恩智浦 LPC 器件使用的 USB_PPWR 相比，该信号具有相反的极性。
									-	**R** — 保留功能。
									I/O	**GPIO5[18]** — 通用数字输入 / 输出引脚。
									O	**ENET_TXD3** — 以太网发送数据 3（MII 接口）。
									I/O	**SGPIO3** — 通用数字输入 / 输出引脚。
									O	**U0_TXD** — USART0 的发送器输出。
P9_6	L11	x	-	103	72	-	[3]	I; PU	I/O	**GPIO4[11]** — 通用数字输入 / 输出引脚。
									O	**MCOB1** — 电机控制 PWM 通道 1，输出 B。
									O	**USB1_PWR_FAULT** — 指示过电流状况的 USB1 端口电源故障信号；此信号监控 USB1 总线上的过电流状况（检测过电流状况需要外部电路）。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I	**ENET_COL** — 以太网冲突检测（MII 接口）。
									I/O	**SGPIO8** — 通用数字输入 / 输出引脚。
									I	**U0_RXD** — UART0 的接收器输入。
PA_0	L12	x	-	126	-	-	[3]	I; PU	-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**I2S1_RX_MCLK** — I2S1 接收主机时钟。
									O	**CGU_OUT1** — CGU 备用时钟输出 1。
									-	**R** — 保留功能。
PA_1	J14	x	-	134	-	-	[4]	I; PU	I/O	**GPIO4[8]** — 通用数字输入 / 输出引脚。
									I	**QEI_IDX** — 正交编码器接口 INDEX 输入。
									-	**R** — 保留功能。
									O	**U2_TXD** — USART2 的发送器输出。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
PA_2	K15	x	-	136	-	-	[4]	I; PU	I/O	**GPIO4[9]** — 通用数字输入 / 输出引脚。
									I	**QEI_PHB** — 正交编码器接口 PHB 输入。
									-	**R** — 保留功能。
									I	**U2_RXD** — UART2 的接收器输入。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PA_3	H11	x	-	147	-	-	[4]	I; PU	I/O	**GPIO4[10]** — 通用数字输入 / 输出引脚。
									I	**QEI_PHA** — 正交编码器接口 PHA 输入。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PA_4	G13	x	-	151	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_9** — SCT 输出 9。定时器 2 的匹配输出 1。
									-	**R** — 保留功能。
									I/O	**EMC_A23** — 外部存储器地址线 23。
									I/O	**GPIO5[19]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PB_0	B15	x	-	164	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_10** — SCT 输出 10。定时器 2 的匹配输出 2。
									O	**LCD_VD23** — LCD 数据。
									-	**R** — 保留功能。
									I/O	**GPIO5[20]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
PB_1	A14	x	-	175	-	-	[3]	I; PU	-	**R** — 保留功能。
									I	**USB1_ULPI_DIR** — ULPI 链接 DIR 信号。控制 ULP 数据线方向。
									O	**LCD_VD22** — LCD 数据。
									-	**R** — 保留功能。
									I/O	**GPIO5[21]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_6** — SCT 输出 6。定时器 1 的匹配输出 2。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PB_2	B12	x	-	177	-	-	[3]	I; PU	-	**R** — 保留功能。
									I/O	**USB1_ULPI_D7** — ULPI 链接双向数据线 7。
									O	**LCD_VD21** — LCD 数据。
									-	**R** — 保留功能。
									I/O	**GPIO5[22]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_7** — SCT 输出 7。定时器 1 的匹配输出 3。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PB_3	A13	x	-	178	-	-	[3]	I; PU	-	**R** — 保留功能。
									I/O	**USB1_ULPI_D6** — ULPI 链接双向数据线 6。
									O	**LCD_VD20** — LCD 数据。
									-	**R** — 保留功能。
									I/O	**GPIO5[23]** — 通用数字输入 / 输出引脚。
									O	**CTOUT_8** — SCT 输出 8。定时器 2 的匹配输出 0。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PB_4	B11	x	-	180	-	-	[3]	I; PU	-	**R** — 保留功能。
									I/O	**USB1_ULPI_D5** — ULPI 链接双向数据线 5。
									O	**LCD_VD15** — LCD 数据。
									-	**R** — 保留功能。
									I/O	**GPIO5[24]** — 通用数字输入 / 输出引脚。
									I	**CTIN_5** — SCT 输入 5。定时器 2 的捕获输入 2。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态 [2]	类型	描述
PB_5	A12	x	-	181	-	-	[3]	I; PU	-	**R** — 保留功能。
									I/O	**USB1_ULPI_D4** — ULPI 链接双向数据线 4。
									O	**LCD_VD14** — LCD 数据。
									-	**R** — 保留功能。
									I/O	**GPIO5[25]** — 通用数字输入 / 输出引脚。
									I	**CTIN_7** — SCT 输入 7。
									O	**LCD_PWR** — LCD 面板电源使能。
									-	**R** — 保留功能。
PB_6	A6	x	-	-	-	-	[6]	I; PU	-	**R** — 保留功能。
									I/O	**USB1_ULPI_D3** — ULPI 链接双向数据线 3。
									O	**LCD_VD13** — LCD 数据。
									-	**R** — 保留功能。
									I/O	**GPIO5[26]** — 通用数字输入 / 输出引脚。
									I	**CTIN_6** — SCT 输入 6。定时器 3 的捕获输入 1。
									O	**LCD_VD19** — LCD 数据。
									-	**R** — 保留功能。
									I	**ADC0_6** — ADC0，输入通道 6。
PC_0	D4	x	-	7	-	-	[6]	I; PU	-	**R** — 保留功能。
									I	**USB1_ULPI_CLK** — ULPI 链接 CLK 信号。 PHY 生成的 60 MHz 时钟。
									-	**R** — 保留功能。
									I/O	**ENET_RX_CLK** — 以太网接收时钟 （MII 接口）。
									O	**LCD_DCLK** — LCD 面板时钟。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SD_CLK** — SD/MMC 卡时钟。
									I	**ADC1_1** — ADC1，输入通道 1。
PC_1	E4	-	-	9	-	-	[3]	I; PU	I/O	**USB1_ULPI_D7** — ULPI 链接双向数据线 7。
									-	**R** — 保留功能。
									I	**U1_RI** — UART 1 的振铃指示器输入。
									O	**ENET_MDC** — 以太网 MIIM 时钟。
									I/O	**GPIO6[0]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I	**T3_CAP0** — 定时器 0 的捕获输入 3。
									O	**SD_VOLT0** — SD/MMC 总线电压选择输出 0。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
PC_2	F6	-	-	13	-	-	[3]	I; PU	I/O	**USB1_ULPI_D6** — ULPI 链接双向数据线 6。
									-	**R** — 保留功能。
									I	**U1_CTS** — UART 1 的准许发送输入。
									O	**ENET_TXD2** — 以太网发送数据 2 （MII 接口）。
									I/O	**GPIO6[1]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**SD_RST** — MMC4.4 卡 SD/MMC 复位信号。
PC_3	F5	-	-	11	-	-	[6]	I; PU	I/O	**USB1_ULPI_D5** — ULPI 链接双向数据线 5。
									-	**R** — 保留功能。
									O	**U1_RTS** — UART 1 的请求发送输出。也可以配置为 UART 1 的 RS-485/EIA-485 输出使能信号。
									O	**ENET_TXD3** — 以太网发送数据 3 （MII 接口）。
									I/O	**GPIO6[2]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**SD_VOLT1** — SD/MMC 总线电压选择输出 1。
									I	**ADC1_0** — ADC1，输入通道 0。
PC_4	F4	-	-	16	-	-	[3]	I; PU	-	**R** — 保留功能。
									I/O	**USB1_ULPI_D4** — ULPI 链接双向数据线 4。
									-	**R** — 保留功能。
										ENET_TX_EN — 以太网发送使能（RMII/MII 接口）。
									I/O	**GPIO6[3]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I	**T3_CAP1** — 定时器 1 的捕获输入 3。
									I/O	**SD_DAT0** — SD/MMC 数据总线 0。
PC_5	G4	-	-	20	-	-	[3]	I; PU	-	**R** — 保留功能。
									I/O	**USB1_ULPI_D3** — ULPI 链接双向数据线 3。
									-	**R** — 保留功能。
									O	**ENET_TX_ER** — 以太网发送错误 （MII 接口）。
									I/O	**GPIO6[4]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I	**T3_CAP2** — 定时器 2 的捕获输入 3。
									I/O	**SD_DAT1** — SD/MMC 数据总线 1。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
PC_6	H6	-	-	22	-	-	[3]	I; PU	-	**R** — 保留功能。
									I/O	**USB1_ULPI_D2** — ULPI 链接双向数据线 2。
									-	**R** — 保留功能。
									I	**ENET_RXD2** — 以太网接收数据 2 （MII 接口）。
									I/O	**GPIO6[5]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I	**T3_CAP3** — 定时器 3 的捕获输入 3。
									I/O	**SD_DAT2** — SD/MMC 数据总线 2。
PC_7	G5	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									I/O	**USB1_ULPI_D1** — ULPI 链接双向数据线 1。
									-	**R** — 保留功能。
									I	**ENET_RXD3** — 以太网接收数据 3 （MII 接口）。
									I/O	**GPIO6[6]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									O	**T3_MAT0** — 定时器 3 的匹配输出 0。
									I/O	**SD_DAT3** — SD/MMC 数据总线 3。
PC_8	N4	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									I/O	**USB1_ULPI_D0** — ULPI 链接双向数据线 0。
									-	**R** — 保留功能。
									I	**ENET_RX_DV** — 以太网接收数据有效 （RMII/MII 接口）。
									I/O	**GPIO6[7]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									O	**T3_MAT1** — 定时器 3 的匹配输出 1。
									I	**SD_CD** — SD/MMC 卡检测输入。
PC_9	K2	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									I	**USB1_ULPI_NXT** — ULPI 链接 NXT 信号。来自 PHY 的数据流控制信号。
									-	**R** — 保留功能。
									I	**ENET_RX_ER** — 以太网接收错误 （MII 接口）。
									I/O	**GPIO6[8]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									O	**T3_MAT2** — 定时器 3 的匹配输出 2。
									O	**SD_POW** — SD/MMC 电源监控输出。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
PC_10	M5	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**USB1_ULPI_STP** — ULPI 链接 STP 信号。断言结束或中断到 PHY 的传输。
									I	**U1_DSR** — UART 1 数据设置就绪输入。
									-	**R** — 保留功能。
									I/O	**GPIO6[9]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									O	**T3_MAT3** — 定时器 3 的匹配输出 3。
									I/O	**SD_CMD** — SD/MMC 命令信号。
PC_11	L5	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									I	**USB1_ULPI_DIR** — ULPI 链接 DIR 信号。控制 ULP 数据线方向。
									I	**U1_DCD** — UART 1 的数据载波检测输入。
									-	**R** — 保留功能。
									I/O	**GPIO6[10]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SD_DAT4** — SD/MMC 数据总线 4。
PC_12	L6	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**U1_DTR** — UART 1 的数据终结就绪输出。也可以配置为 UART 1 的 RS-485/EIA-485 输出使能信号。
									-	**R** — 保留功能。
									I/O	**GPIO6[11]** — 通用数字输入 / 输出引脚。
									I/O	**SGPIO11** — 通用数字输入 / 输出引脚。
									I/O	**I2S0_TX_SDA** — I2S 发送数据。由发送器驱动，接收器读取。相当于 *I²S 总线规范中的信号 SD*。
									I/O	**SD_DAT5** — SD/MMC 数据总线 5。
PC_13	M1	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**U1_TXD** — UART 1 的发送器输出。
									-	**R** — 保留功能。
									I/O	**GPIO6[12]** — 通用数字输入 / 输出引脚。
									I/O	**SGPIO12** — 通用数字输入 / 输出引脚。
									I/O	**I2S0_TX_WS** — 发送字选择。由主机驱动，从机接收。相当于 *I²S 总线规范中的信号 WS*。
									I/O	**SD_DAT6** — SD/MMC 数据总线 6。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
PC_14	N1	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									-	**R** — 保留功能。
									I	**U1_RXD** — UART 1 的接收器输入。
									-	**R** — 保留功能。
									I/O	**GPIO6[13]** — 通用数字输入 / 输出引脚。
									I/O	**SGPIO13** — 通用数字输入 / 输出引脚。
									O	**ENET_TX_ER** — 以太网发送错误 （MII 接口）。
									I/O	**SD_DAT7** — SD/MMC 数据总线 7。
PD_0	N2	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_15** — SCT 输出 15。定时器 3 的匹配输出 3。
									O	**EMC_DQMOUT2** — SDRAM 和静态器件使用的数据屏蔽 2。
									-	**R** — 保留功能。
									I/O	**GPIO6[14]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SGPIO4** — 通用数字输入 / 输出引脚。
PD_1	P1	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**EMC_CKEOUT2** — SDRAM 时钟使能 2。
									-	**R** — 保留功能。
									I/O	**GPIO6[15]** — 通用数字输入 / 输出引脚。
									O	**SD_POW** — SD/MMC 电源监控输出。
									-	**R** — 保留功能。
									I/O	**SGPIO5** — 通用数字输入 / 输出引脚。
PD_2	R1	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_7** — SCT 输出 7。定时器 1 的匹配输出 3。
									I/O	**EMC_D16** — 外部存储器数据线 16。
									-	**R** — 保留功能。
									I/O	**GPIO6[16]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SGPIO6** — 通用数字输入 / 输出引脚。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
PD_3	P4	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_6** — SCT 输出 7。定时器 1 的匹配输出 2。
									I/O	**EMC_D17** — 外部存储器数据线 17。
									-	**R** — 保留功能。
									I/O	**GPIO6[17]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SGPIO7** — 通用数字输入 / 输出引脚。
PD_4	T2	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_8** — SCT 输出 8。定时器 2 的匹配输出 0。
									I/O	**EMC_D18** — 外部存储器数据线 18。
									-	**R** — 保留功能。
									I/O	**GPIO6[18]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SGPIO8** — 通用数字输入 / 输出引脚。
PD_5	P6	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_9** — SCT 输出 9。定时器 2 的匹配输出 1。
									I/O	**EMC_D19** — 外部存储器数据线 19。
									-	**R** — 保留功能。
									I/O	**GPIO6[19]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SGPIO9** — 通用数字输入 / 输出引脚。
PD_6	R6	-	-	68	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_10** — SCT 输出 10。定时器 2 的匹配输出 2。
									I/O	**EMC_D20** — 外部存储器数据线 20。
									-	**R** — 保留功能。
									I/O	**GPIO6[20]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SGPIO10** — 通用数字输入 / 输出引脚。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态 [2]	类型	描述
PD_7	T6	-	-	72	-	-	[3]	I; PU	-	**R** — 保留功能。
									I	**CTIN_5** — SCT 输入 5。定时器 2 的捕获输入 2。
									I/O	**EMC_D21** — 外部存储器数据线 21。
									-	**R** — 保留功能。
									I/O	**GPIO6[21]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SGPIO11** — 通用数字输入 / 输出引脚。
PD_8	P8	-	-	74	-	-	[3]	I; PU	-	**R** — 保留功能。
									I	**CTIN_6** — SCT 输入 6。定时器 3 的捕获输入 1。
									I/O	**EMC_D22** — 外部存储器数据线 22。
									-	**R** — 保留功能。
									I/O	**GPIO6[22]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SGPIO12** — 通用数字输入 / 输出引脚。
PD_9	T11	-	-	84	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_13** — SCT 输出 13。定时器 3 的匹配输出 1。
									I/O	**EMC_D23** — 外部存储器数据线 23。
									-	**R** — 保留功能。
									I/O	**GPIO6[23]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SGPIO13** — 通用数字输入 / 输出引脚。
PD_10	P11	-	-	86	-	-	[3]	I; PU	-	**R** — 保留功能。
									I	**CTIN_1** — SCT 输入 1。定时器 0 的捕获输入 1。定时器 2 的捕获输入 1。
									O	**$\overline{\text{EMC_BLS3}}$** — 低电平有效“字节通道”选择信号 3。
									-	**R** — 保留功能。
									I/O	**GPIO6[24]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态 [2]	类型	描述
PD_11	N9	x	-	88	-	-	[3]	I; PU	-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**EMC_CS3** — 低电平有效芯片选择 3 信号。
									-	**R** — 保留功能。
									I/O	**GPIO6[25]** — 通用数字输入 / 输出引脚。
									I/O	**USB1_ULPI_D0** — ULPI 链接双向数据线 0。
									O	**CTOUT_14** — SCT 输出 14。定时器 3 的匹配输出 2。
									-	**R** — 保留功能。
PD_12	N11	x	-	94	-	-	[3]	I; PU	-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**EMC_CS2** — 低电平有效芯片选择 2 信号。
									-	**R** — 保留功能。
									I/O	**GPIO6[26]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									O	**CTOUT_10** — SCT 输出 10。定时器 2 的匹配输出 2。
									-	**R** — 保留功能。
PD_13	T14	x	-	97	-	-	[3]	I; PU	-	**R** — 保留功能。
									I	**CTIN_0** — SCT 输入 0。定时器 0、1、2、3 的捕获输入 0。
									O	**EMC_BLS2** — 低电平有效“字节通道”选择信号 2。
									-	**R** — 保留功能。
									I/O	**GPIO6[27]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									O	**CTOUT_13** — SCT 输出 13。定时器 3 的匹配输出 1。
									-	**R** — 保留功能。
PD_14	R13	x	-	99	-	-	[3]	I; PU	-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**EMC_DYCS2** — SDRAM 芯片选择 2。
									-	**R** — 保留功能。
									I/O	**GPIO6[28]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									O	**CTOUT_11** — SCT 输出 11。定时器 2 的匹配输出 3。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态 [2]	类型	描述
PD_15	T15	x	-	101	-	-	[3]	I; PU	-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**EMC_A17** — 外部存储器地址线 17。
									-	**R** — 保留功能。
									I/O	**GPIO6[29]** — 通用数字输入 / 输出引脚。
									I	**SD_WP** — SD/MMC 卡写入保护输入。
									O	**CTOUT_8** — SCT 输出 8。定时器 2 的匹配输出 0。
									-	**R** — 保留功能。
PD_16	R14	x	-	104	-	-	[3]	I; PU	-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**EMC_A16** — 外部存储器地址线 16。
									-	**R** — 保留功能。
									I/O	**GPIO6[30]** — 通用数字输入 / 输出引脚。
									O	**SD_VOLT2** — SD/MMC 总线电压选择输出 2。
									O	**CTOUT_12** — SCT 输出 12。定时器 3 的匹配输出 0。
									-	**R** — 保留功能。
PE_0	P14	x	-	106	-	-	[3]	I; PU	-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**EMC_A18** — 外部存储器地址线 18。
									I/O	**GPIO7[0]** — 通用数字输入 / 输出引脚。
									O	**CAN1_TD** — CAN1 发送器输出。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PE_1	N14	x	-	112	-	-	[3]	I; PU	-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**EMC_A19** — 外部存储器地址线 19。
									I/O	**GPIO7[1]** — 通用数字输入 / 输出引脚。
									I	**CAN1_RD** — CAN1 接收器输入。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
PE_2	M14	x	-	115	-	-	[3]	I; PU	I	**ADCTRIG0** — ADC 启动器输入 0。
									I	**CAN0_RD** — CAN 接收器输入。
									-	**R** — 保留功能。
									I/O	**EMC_A20** — 外部存储器地址线 20。
									I/O	**GPIO7[2]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PE_3	K12	x	-	118	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CAN0_TD** — CAN 发送器输出。
									I	**ADCTRIG1** — ADC 启动器输入 1。
									I/O	**EMC_A21** — 外部存储器地址线 21。
									I/O	**GPIO7[3]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PE_4	K13	x	-	120	-	-	[3]	I; PU	-	**R** — 保留功能。
									I	**NMI** — NMI 外部中断输入。
									-	**R** — 保留功能。
									I/O	**EMC_A22** — 外部存储器地址线 22。
									I/O	**GPIO7[4]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PE_5	N16	-	-	122	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_3** — SCT 输出 3。定时器 0 的匹配输出 3。
									O	**U1_RTS** — UART 1 的请求发送输出。也可以配置为 UART 1 的 RS-485/EIA-485 输出使能信号。
									I/O	**EMC_D24** — 外部存储器数据线 24。
									I/O	**GPIO7[5]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
PE_6	M16	-	-	124	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_2** — SCT 输出 2。定时器 0 的匹配输出 2。
									I	**U1_RI** — UART 1 的振铃指示器输入。
									I/O	**EMC_D25** — 外部存储器数据线 25。
									I/O	**GPIO7[6]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PE_7	F15	-	-	149	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_5** — SCT 输出 5。定时器 1 的匹配输出 1。
									I	**U1_CTS** — UART1 准许发送输入。
									I/O	**EMC_D26** — 外部存储器数据线 26。
									I/O	**GPIO7[7]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PE_8	F14	-	-	150	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_4** — SCT 输出 4。定时器 0 的匹配输出 0。
									I	**U1_DSR** — UART 1 数据设置就绪输入。
									I/O	**EMC_D27** — 外部存储器数据线 27。
									I/O	**GPIO7[8]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PE_9	E16	-	-	152	-	-	[3]	I; PU	-	**R** — 保留功能。
									I	**CTIN_4** — SCT 输入 4。定时器 1 的捕获输入 2。
									I	**U1_DCD** — UART 1 的数据载波检测输入。
									I/O	**EMC_D28** — 外部存储器数据线 28。
									I/O	**GPIO7[9]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
PE_10	E14	-	-	154	-	-	[3]	I; PU	-	**R** — 保留功能。
									I	**CTIN_3** — SCT 输入 3。定时器 1 的捕获输入 1。
									O	**U1_DTR** — UART 1 的数据终结就绪输出。也可以配置为 UART 1 的 RS-485/EIA-485 输出使能信号。
									I/O	**EMC_D29** — 外部存储器数据线 29。
									I/O	**GPIO7[10]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PE_11	D16	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_12** — SCT 输出 12。定时器 3 的匹配输出 0。
									O	**U1_TXD** — UART 1 的发送器输出。
									I/O	**EMC_D30** — 外部存储器数据线 30。
									I/O	**GPIO7[11]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PE_12	D15	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_11** — SCT 输出 11。定时器 3 的匹配输出 3。
									I	**U1_RXD** — UART 1 的接收器输入。
									I/O	**EMC_D31** — 外部存储器数据线 31。
									I/O	**GPIO7[12]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PE_13	G14	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_14** — SCT 输出 14。定时器 2 的匹配输出 3。
									I/O	**I2C1_SDA** — I^2C1 数据输入 / 输出（此引脚不使用专门的 I^2C 缓冲器）。
									O	**EMC_DQMOUT3** — SDRAM 和静态器件使用的数据屏蔽 3。
									I/O	**GPIO7[13]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
PE_14	C15	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**EMC_DYCS3** — SDRAM 芯片选择 3。
									I/O	**GPIO7[14]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PE_15	E13	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**CTOUT_0** — SCT 输出 0。定时器 0 的匹配输出 0。
									I/O	**I2C1_SCL** — I^2C1 时钟输入 / 输出（此引脚不使用专门的 I^2C 缓冲器）。
									O	**EMC_CKEOUT3** — SDRAM 时钟使能 3。
									I/O	**GPIO7[15]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
PF_0	D12	-	-	159	-	-	[3]	O; PU	I/O	**SSP0_SCK** — SSP0 的串行时钟。
									I	**GP_CLKIN** — CGU 通用时钟输入。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**I2S1_TX_MCLK** — I2S1 发送主机时钟。
PF_1	E11	-	-	-	-	-	[3]	I; PU	-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SSP0_SSEL** — SSP0 从机选择。
									-	**R** — 保留功能。
									I/O	**GPIO7[16]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I/O	**SGPIO0** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态 [2]	类型	描述
PF_2	D11	-	-	168	-	-	[3]	I; PU	-	**R** — 保留功能。
									O	**U3_TXD** — USART3 的发送器输出。
									I/O	**SSP0_MISO** — SSP0 主机输入、从机输出。
									-	**R** — 保留功能。
									I/O	**GPIO7[17]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I/O	**SGPIO1** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
PF_3	E10	-	-	170	-	-	[3]	I; PU	-	**R** — 保留功能。
									I	**U3_RXD** — UART3 的接收器输入。
									I/O	**SSP0_MOSI** — SSP0 主机输出，从机输入。
									-	**R** — 保留功能。
									I/O	**GPIO7[18]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I/O	**SGPIO2** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
PF_4	D10	x	H4	172	120	83	[3]	O; PU	I/O	**SSP1_SCK** — SSP1 的串行时钟。
									I	**GP_CLKIN** — CGU 通用时钟输入。
									O	**TRACECLK** — 跟踪时钟。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**I2S0_TX_MCLK** — I2S 发送主机时钟。
									I/O	**I2S0_RX_SCK** — I2S 接收时钟。由主机驱动，从机接收。相当于 *I^2S 总线规范中的信号 SCK*。
PF_5	E9	-	-	190	-	-	[6]	I; PU	-	**R** — 保留功能。
									I/O	**U3_UCLK** — 同步模式下 USART3 的串行时钟输入 / 输出。
									I/O	**SSP1_SSEL** — SSP1 从机选择。
									O	**TRACEDATA[0]** — 跟踪数据、位 0。
									I/O	**GPIO7[19]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I/O	**SGPIO4** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I	**ADC1_4** — ADC1，输入通道 4。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
PF_6	E7	-	-	192	-	-	[6]	I; PU	-	**R** — 保留功能。
									I/O	**U3_DIR** — USART3 RS-485/EIA-485 输出使能 / 方向控制。
									I/O	**SSP1_MISO** — SSP1 主机输入、从机输出。
									O	**TRACEDATA[1]** — 跟踪数据、位 1。
									I/O	**GPIO7[20]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I/O	**SGPIO5** — 通用数字输入 / 输出引脚。
									I/O	**I2S1_TX_SDA** — I2S 发送数据。由发送器驱动，接收器读取。相当于 *I^2S 总线规范中的信号 SD*。
									I	**ADC1_3** — ADC1，输入通道 3。
PF_7	B7	-	-	193	-	-	[6]	I; PU	-	**R** — 保留功能。
									I/O	**U3_BAUD** — USART3 波特引脚。
									I/O	**SSP1_MOSI** — SSP1 主机输出，从机输入。
									O	**TRACEDATA[2]** — 跟踪数据、位 2。
									I/O	**GPIO7[21]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I/O	**SGPIO6** — 通用数字输入 / 输出引脚。
									I/O	**I2S1_TX_WS** — 发送字选择。由主机驱动，从机接收。相当于 *I^2S 总线规范中的信号 WS*。
									I/O	**ADC1_7** — ADC1，输入通道 7 或带隙输出。
PF_8	E6	-	-	-	-	-	[6]	I; PU	-	**R** — 保留功能。
									I/O	**U0_UCLK** — 同步模式下 USART0 的串行时钟输入 / 输出。
									I	**CTIN_2** — SCT 输入 2。定时器 0 的捕获输入 2。
									O	**TRACEDATA[3]** — 跟踪数据、位 3。
									I/O	**GPIO7[22]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I/O	**SGPIO7** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I	**ADC0_2** — ADC0，输入通道 2。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
PF_9	D6	-	-	203	-	-	[6]	I; PU	-	**R** — 保留功能。
									I/O	**U0_DIR** — USART0 RS-485/EIA-485 输出使能 / 方向控制。
									O	**CTOUT_1** — SCT 输出 1。定时器 0 的匹配输出 1。
									-	**R** — 保留功能。
									I/O	**GPIO7[23]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I/O	**SGPIO3** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I	**ADC1_2** — ADC1，输入通道 2。
PF_10	A3	-	-	205	-	98	[6]	I; PU	-	**R** — 保留功能。
									O	**U0_TXD** — USART0 的发送器输出。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**GPIO7[24]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									I	**SD_WP** — SD/MMC 卡写入保护输入。
									-	**R** — 保留功能。
									I	**ADC0_5** — ADC0，输入通道 5。
PF_11	A2	-	-	207	-	100	[6]	I; PU	-	**R** — 保留功能。
									I	**U0_RXD** — UART0 的接收器输入。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**GPIO7[25]** — 通用数字输入 / 输出引脚。
									-	**R** — 保留功能。
									O	**SD_VOLT2** — SD/MMC 总线电压选择输出 2。
									-	**R** — 保留功能。
									I	**ADC1_5** — ADC1，输入通道 5。
时钟引脚										
CLK0	N5	x	K3	62	45	31	[5]	O; PU	O	**EMC_CLK0** — SDRAM 时钟 0。
									O	**CLKOUT** — 时钟输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SD_CLK** — SD/MMC 卡时钟。
									O	**EMC_CLK01** — SDRAM 时钟 0 和时钟 1 组合。
									I/O	**SSP1_SCK** — SSP1 的串行时钟。
									I	**ENET_TX_CLK (ENET_REF_CLK)** — 以太网发送时钟（MII 接口）或以太网基准时钟（RMII 接口）。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
CLK1	T10	x	-	-	-	-	[5]	O; PU	O	**EMC_CLK1** — SDRAM 时钟 1。
									O	**CLKOUT** — 时钟输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**CGU_OUT0** — CGU 备用时钟输出 0。
									-	**R** — 保留功能。
									O	**I2S1_TX_MCLK** — I2S1 发送主机时钟。
CLK2	D14	x	K6	141	99	68	[5]	O; PU	O	**EMC_CLK3** — SDRAM 时钟 3。
									O	**CLKOUT** — 时钟输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									I/O	**SD_CLK** — SD/MMC 卡时钟。
									O	**EMC_CLK23** — SDRAM 时钟 2 和时钟 3 组合。
									O	**I2S0_TX_MCLK** — I2S 发送主机时钟。
									I/O	**I2S1_RX_SCK** — 接收时钟。由主机驱动，从机接收。相当于 I^2S 总线规范中的信号 SCK。
CLK3	P12	x	-	-	-	-	[5]	O; PU	O	**EMC_CLK2** — SDRAM 时钟 2。
									O	**CLKOUT** — 时钟输出引脚。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									-	**R** — 保留功能。
									O	**CGU_OUT1** — CGU 备用时钟输出 1。
									-	**R** — 保留功能。
									I/O	**I2S1_RX_SCK** — 接收时钟。由主机驱动，从机接收。相当于 I^2S 总线规范中的信号 SCK。
调试引脚										
DBGEN	L4	x	A6	41	28	18	[3]	I	I	JTAG 接口控制信号，也用于边界扫描。
TCK/SWDCLK	J5	x	H2	38	27	17	[3]	I; F	I	JTAG 接口的测试时钟 （默认）或串行 (SW) 时钟。
$\overline{TRST}$	M4	x	B4	42	29	19	[3]	I; PU	I	JTAG 接口的测试复位。
TMS/SWDIO	K6	x	C4	44	30	20	[3]	I; PU	I	JTAG 接口的测试模式选择 （默认）或串行调试数据输入 / 输出。
TDO/SWO	K5	x	H3	46	31	21	[3]	O	O	JTAG 接口的测试数据输出（默认）或 SW 跟踪输出。
TDI	J4	x	G3	35	26	16	[3]	I; PU	I	JTAG 接口的测试数据输入。
USB0 引脚										
USB0_DP	F2	x	E1	26	18	9	[7]	-	I/O	USB0 双向 D+ 线。
USB0_DM	G2	x	E2	28	20	11	[7]	-	I/O	USB0 双向 D– 线路。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
USB0_VBUS	F1	x	E3	29	21	12	[7] [8]	-	I/O	VBUS 引脚（USB 线缆电源）。该引脚包括一个 64 kΩ（典型值）± 16 kΩ 的内部下拉电阻。
USB0_ID	H2	x	F1	30	22	13	[9]	-	I	向收发器指明是连接到器件 A（USB0_ID 低）还是器件 B（USB0_ID 高）。对于 OTG，该引脚具有一个内部上拉电阻。
USB0_RREF	H1	x	F3	32	24	15	[9]	-		用于电流基准的 12.0 kΩ（精度 1 %）板载接地电阻。
USB1 引脚										
USB1_DP	F12	x	E9	129	89	59	[10]	-	I/O	USB1 双向 D+ 线。
USB1_DM	G12	x	E10	130	90	60	[10]	-	I/O	USB1 双向 D– 线路。
I^2C 总线引脚										
I2C0_SCL	L15	x	D6	132	92	62	[11]	I; F	I/O	I^2C 时钟输入 / 输出。开漏输出（符合 I^2C 总线规范）。
I2C0_SDA	L16	x	E6	133	93	63	[11]	I; F	I/O	I^2C 数据输入 / 输出。开漏输出（符合 I^2C 总线规范）。
复位和唤醒引脚										
$\overline{\text{RESET}}$	D9	x	B6	185	128	91	[12]	I; IA	I	外部复位输入：此引脚上的低电平将会复位器件，从而导致 I/O 端口和外围设备采用其默认状态，并且处理器从地址 0 开始执行。
WAKEUP0	A9	x	A4	187	130	93	[12]	I; IA	I	外部唤醒输入；可以产生中断并从任何低功耗模式中唤醒。
WAKEUP1	A10	x	-	-	-	-	[12]	I; IA	I	外部唤醒输入；可以产生中断并从任何低功耗模式中唤醒。
WAKEUP2	C9	x	-	-	-	-	[12]	I; IA	I	外部唤醒输入；可以产生中断并从任何低功耗模式中唤醒。
WAKEUP3	D8	x	-	-	-	-	[12]	I; IA	I	外部唤醒输入；可以产生中断并从任何低功耗模式中唤醒。
ADC 引脚										
ADC0_0/ ADC1_0/DAC	E3	x	A2	8	6	4	[9]	I; IA	I	ADC 输入通道 0。在 10 位 ADC0/1 和 DAC 之间共享。
ADC0_1/ ADC1_1	C3	x	A1	4	2	1	[9]	I; IA	I	ADC 输入通道 1。在 10 位 ADC0/1 之间共享。
ADC0_2/ ADC1_2	A4	x	B3	206	143	99	[9]	I; IA	I	ADC 输入通道 2。在 10 位 ADC0/1 之间共享。
ADC0_3/ ADC1_3	B5	x	A3	200	139	96	[9]	I; IA	I	ADC 输入通道 3。在 10 位 ADC0/1 之间共享。
ADC0_4/ ADC1_4	C6	x	-	199	138	-	[9]	I; IA	I	ADC 输入通道 4。在 10 位 ADC0/1 之间共享。
ADC0_5/ ADC1_5	B3	x	-	208	144	-	[9]	I; IA	I	ADC 输入通道 5。在 10 位 ADC0/1 之间共享。
ADC0_6/ ADC1_6	A5	x	-	204	142	-	[9]	I; IA	I	ADC 输入通道 6。在 10 位 ADC0/1 之间共享。
ADC0_7/ ADC1_7	C5	x	-	197	136	-	[9]	I; IA	I	ADC 输入通道 7。在 10 位 ADC0/1 之间共享。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
RTC										
RTC_ALARM	A11	x	C3	186	129	92	[12]	-	O	RTC 控制输出，
RTCX1	A8	x	A5	182	125	88	[9]	-	I	RTC 32 kHz 超低功耗振荡器电路的输入。
RTCX2	B8	x	B5	183	126	89	[9]	-	O	RTC 32 kHz 超低功耗振荡器电路的输出。
晶体振荡器引脚										
XTAL1	D1	x	B1	18	12	5	[9]	-	I	振荡器电路和内部时钟发生器电路的输入。
XTAL2	E1	x	C1	19	13	6	[9]	-	O	振荡器放大器的输出。
电源和接地引脚										
USB0_VDDA 3V3_DRIVER	F3	x	D1	24	16	7		-	-	驱动器的独立模拟 3.3 V 电源。
USB0 _VDDA3V3	G3	x	D2	25	17	8		-	-	USB 3.3 V 独立电源电压。
USB0_VSSA _TERM	H3	x	D3	27	19	10		-	-	用于终结电阻独立基准的专用模拟接地。
USB0_VSSA _REF	G1	x	F2	31	23	14		-	-	基准电流和电压产生的专用独立模拟接地。
VDDA	B4	x	B2	198	137	95		-	-	模拟电源和 ADC 基准电压。
VBAT	B10	x	C5	184	127	90		-	-	RTC 电源：3.3 此引脚上的 V 向 RTC 供电。
VDDREG	F10, F9, L8, L7	x	E4, E5, F4	135, 188, 195, 82, 33	94, 131, 59, 25	-			-	主调压器电源。将 VDDREG 和 VDDIO 引脚连接至共同电源，确保两个电源电压的上升时间相同。
VPP	E8	-	-	-	-	-	[13]	-	-	OTP 编程电压。
VDDIO	D7, E12, F7, F8, G10, H10, J6, J7, K7, L9, L10, N7, N13	x	F10, K5	6, 52, 57, 102, 110, 155, 160, 202	5, 36, 41, 71, 77, 107, 111, 141	-	[13]	-	-	I/O 电源。将 VDDREG 和 VDDIO 引脚连接至共同电源，确保两个电源电压的上升时间相同。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 120. 引脚描述 *（续）*

LCD、以太网、 USB0 以及 USB1 功能并非在所有器件上都可用。参见表 2。

符号	LBGA256	TFBGA180[1]	TFBGA100	LQFP208[1]	LQFP144	LQFP100[1]		复位状态[2]	类型	描述
VDD	-	-	-	-	-	3, 24, 27, 49, 52, 74, 77, 97				主调压器、 I/O 和 OTP 的电源。
VSS	G9, H7, J10, J11, K8	x	-	-	-	2, 26, 51, 76	[14] [15]	-	-	地线。
VSSIO	C4, D13, G6, G7, G8, H8, H9, J8, J9, K9, K10, M13, P7, P13	x	C8, D4, D5, G8, J3, J6	5, 56, 109, 157	4, 40, 76, 109	-	[14] [15]	-	-	地线。
VSSA	B2	x	C2	196	135	94		-	-	模拟接地。
未连接										
-	B9	-	-	-	-	-		-	-	n.c.

[1] x = 可用； - = 未引出。

[2] I = 输入， O = 输出， IA = 无效； PU = 使能上拉电阻（弱上拉电阻上拉引脚至 $V_{DD(IO)}$）； F = 悬浮；复位状态反映引脚在复位时无启动代码操作的状态。

[3] 具有 15 ns 干扰滤波器的 5 V 容限管脚（$V_{DD(IO)}$ 存在时，容限为 5 V； $V_{DD(IO)}$ 不存在时，容限不超过 3.3 V）；提供具有 TTL 电平和滞回的数字 I/O 功能；标准电平驱动强度。

[4] 具有 15 ns 干扰滤波器的 5 V 容限管脚（$V_{DD(IO)}$ 存在时，容限为 5 V； $V_{DD(IO)}$ 不存在时，容限不超过 3.3 V）；提供具有 TTL 电平和滞回的数字 I/O 功能；高驱动能力强度。

[5] 具有 15 ns 干扰滤波器的 5 V 容限管脚（$V_{DD(IO)}$ 存在时，容限为 5 V； $V_{DD(IO)}$ 不存在时，容限不超过 3.3 V）；提供具有 TTL 电平和滞回的高速数字 I/O 功能。

[6] 5V 容限管脚提供数字 I/O 功能（具有 TTL 电平和滞回）和模拟输入或输出（$V_{DD(IO)}$ 存在时，容限为 5V； $V_{DD(IO)}$ 不存在时，容限不超过 3.3 V）。当配置为 ADC 输入或 DAC 输出时，引脚非 5 V 容限，必须禁用 pad 的数字部分，方法为：将引脚设置为输入函数并通过引脚的 SFSP 寄存器禁用上拉电阻。

[7] 5 V 容限的透明模拟管脚。

[8] 对于最大负载 C_L=6.5μF 和最大电阻 R_{pd}=80kΩ，当 VBUS 信号不再被驱动时，从 VBUS = 5 V 下降至 VBUS = 0.2 V 约需 2 秒。

[9] 透明模拟管脚。非 5 V 容限。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

[10] 管脚提供 USB 功能 （$V_{DD(IO)}$ 存在时，容限为 5V ； $V_{DD(IO)}$ 不存在时，容限不超过 3.3 V）。其设计符合 USB 规范， 2.0 版 （仅限于全速和低速模式）。此管脚非 5 V 容限。

[11] 5 V 容限的开漏数字 I/O 管脚，符合 I^2C 总线超快速模式规范。此 pad 要求进行外部上拉，以提供输出功能。电源关闭时，连接到 I^2C 总线的此引脚将处于悬浮状态，并且不会干扰 I^2C 线路。

[12] 5V 容限的管脚以及 20ns 的干扰滤波器；提供数字 I/O 功能，带有低上拉电阻和滞回的开漏输出。

[13] 在 TFBGA100 和 LQFP208 封装中， VPP 内部连接至 VDDIO。

[14] 在 LQFP144 封装中， VSSIO 和 VSS 连接至同一个接地层。

[15] 在 TFBGA100 和 LQFP100/208 封装中， VSS 内部连接至 VSSIO。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

15.1 本章导读

下列外围设备并非在所有器件上均可用，且选择 SFSP 寄存器的这些功能的对应位值会保留：

- 以太网：仅用于 LPC4350/30。
- USB0：仅用于 LPC4350/30/20。
- USB1：仅用于 LPC4350/30。
- LCD：仅用于 LPC4350。

15.2 基本配置

SCU 的配置如下：

- 参见表 121 以了解时钟与电源控制。
- SCU 通过 SCU_RST 复位 （复位 # 9）。

表 121. SCU 时钟和电源控制

	基本时钟	分支时钟	工作频率
SCU 寄存器接口时钟	BASE_M4_CLK	CLK_M4_SCU	高达 204 MHz

注：在使用任何多路复用引脚或 I2C0 引脚**作为输入**之前，应按以下步骤设置对应的引脚配置寄存器：

- 将位 EZI 设为 1，使能输入缓冲。
- 对于高频信号，将位 ZIF 设为 1，禁用输入干扰滤波器。

15.3 简介

系统控制单元决定了多数数字引脚的功能和电气模式。默认情况下，所有引脚选择使能上拉的数字功能 0。

注：某些管脚支持数字功能和模拟功能的引脚多路复用。ADC 和 DAC 的所有模拟 I/O 也通过引脚输出在纯模拟管脚上，无引脚多路复用。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 27. I/O 管脚的框图

15.3.1 数字引脚功能

SFS 寄存器的 FUNC 位可以控制各个引脚的功能。每个引脚的数字功能可多达 8 种，并且某些引脚支持附加的模拟功能。如果该功能设为 GPIO，则 DIR 寄存器会决定该引脚是配置为输入还是输出（参见表 191）。对于任何外设功能，引脚方向根据引脚的功能自动控制。GPIO DIR 寄存器不会影响外设功能。

15.3.2 数字引脚模式

SFS 寄存器的 EPUN 和 EPD 位（见图 27）可为每个引脚选择典型值为 50 kΩ 的弱片内上拉或下拉电阻，或者选择中继模式。

片内电阻配置可能是上拉使能、下拉使能，或是两者都不使能。默认值为上拉使能。

当引脚处于逻辑高电平时，中继模式会使能上拉电阻；当引脚处于逻辑低电平时，则使能下拉电阻。这样，当引脚配置为输入且不由外部驱动时，会使引脚保持其上一已知状态。如果暂时不驱动引脚，通常可用中继模式来防止引脚悬空（当引脚处于不确定状态时，可能会使用大量电量）。

要选择中继模式，则配置 SFS 寄存器中的上拉电阻和下拉电阻。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

15.3.3 输入缓冲

要能够接收数字信号，必须通过引脚配置寄存器的位 EZI 使能输入缓冲 （见图 27）。默认情况下，禁用输入缓冲。

对于支持数字和模拟两种功能的管脚，必须在使能模拟功能前禁用输入缓冲 （见第 15.4.6 节至第 15.4.8 节）。

15.3.4 可编程干扰滤波器

所有数字引脚都支持可开启或关闭的可编程干扰滤波器 （位 ZIF）（见图 27）。默认情况下，开启干扰滤波器。当时钟信号频率高于 30 MHz 时，应禁用干扰滤波器。

15.3.5 可编程转换速率

标准电平驱动和高速引脚支持可编程转换速率 （位 EHS）在低噪声和低速或者高噪声和高速之间选择 （见图 27）。支持的典型频率为 50 MHz/80 MHz （标准电平驱动引脚）和 75 MHz/204 MHz （高速引脚）。

15.3.6 高速引脚

时钟引脚 CLK0 至 CLK3 和 P3_3 支持典型频率为 75 MHz 或 204 MHz （具体取决于转换速率设置）的可编程高速输出 （参见第 15.3.5 节）。

15.3.7 高驱动能力引脚

选定引脚 （见第 15.4.2 节）支持具有 4 个可编程电平的高驱动能力输出。

高驱动能力引脚支持可编程干扰滤波器，但不支持可编程转换速率。

15.3.8 I^2C0 总线引脚

SFSI2C0 寄存器 （表 127）允许为 I^2C0 总线接口配置不同模式：

- 标准模式 / 快速模式 I^2C （根据 I^2C 总线规范，其中包括开漏输出）。这是默认模式。
- 超快速模式 （根据 I^2C 总线规范，其中包括开漏输出）。

I2C0 引脚使用可编程干扰滤波器 （位 ZIF）。

注：要让 I2C0 引脚 SDA 和 SCL 正常工作，必须使能输入缓冲。

15.3.9 USB1 USB1_DP/USB1_DM 引脚

USB1 的输入信号由 SFSUSB 寄存器控制（表 126）。USB_ESEA 位必须设为 1，以使能 USB1 模块。

15.3.10 EMC 信号延迟控制

SCU 包含所有 EMC SDRAM 时钟的可编程延迟控制 （参见表 134）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

15.3.11 引脚复用

多路复用数字引脚将分组到 16 个引脚组，即 P0 到 P9 和 PA 到 PF，每组最多使用 20 个引脚。每个数字引脚可以支持多达 8 种不同的数字功能，包括通用 I/O (GPIO)，可通过 SCU 寄存器进行选择。此外，某些引脚支持通过 ENAIO 寄存器可选的模拟功能 （ADC 输入和 DAC 输出）。请注意，引脚名称并不表示所指派的 GPIO 端口。

15.4 寄存器描述

系统控制单元所含的寄存器可配置多路复用数字引脚的引脚功能、 EMC 时钟延迟和 GPIO 引脚中断。

注：器件启动时，启动引导程序可配置启动过程中包括的引脚（见第 5.3.2 节）。表 122 中针对引脚配置寄存器和 EMC 延迟寄存器给定的复位值不考虑启动引导程序执行的更改。

表 122. 寄存器简介：系统控制单元 (SCU) （基址 0x4008 6000）

名称	访问类型	地址偏移	描述	复位值	EMC 启动后的复位值	UART 启动后的复位值	参考
引脚 P0_n							
SFSP0_0	R/W	0x000	引脚 P0_0 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP0_1	R/W	0x004	引脚 P0_1 的引脚配置寄存器	0x00	0x00	0x00	表 123
-	-	0x008 - 0x07C	保留	-	-	-	
引脚 P1_n							
SFSP1_0	R/W	0x080	引脚 P1_0 的引脚配置寄存器	0x00	0xD2	0x00	表 123
SFSP1_1	R/W	0x084	引脚 P1_1 的引脚配置寄存器	0x00	0xD2	0x00	表 123
SFSP1_2	R/W	0x088	引脚 P1_2 的引脚配置寄存器	0x00	0xD2	0x00	表 123
SFSP1_3	R/W	0x08C	引脚 P1_3 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP1_4	R/W	0x090	引脚 P1_4 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP1_5	R/W	0x094	引脚 P1_5 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP1_6	R/W	0x098	引脚 P1_6 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP1_7	R/W	0x09C	引脚 P1_7 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP1_8	R/W	0x0A0	引脚 P1_8 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP1_9	R/W	0x0A4	引脚 P1_9 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP1_10	R/W	0x0A8	引脚 P1_10 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP1_11	R/W	0x0AC	引脚 P1_11 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP1_12	R/W	0x0B0	引脚 P1_12 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP1_13	R/W	0x0B4	引脚 P1_13 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP1_14	R/W	0x0B8	引脚 P1_14 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP1_15	R/W	0x0BC	引脚 P1_15 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP1_16	R/W	0x0C0	引脚 P1_16 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP1_17	R/W	0x0C4	引脚 P1_17 的引脚配置寄存器	0x00	0x00	0x00	表 124
SFSP1_18	R/W	0x0C8	引脚 P1_18 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP1_19	R/W	0x0CC	引脚 P1_19 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP1_20	R/W	0x0D0	引脚 P1_20 的引脚配置寄存器	0x00	0x00	0x00	表 123

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 122. 寄存器简介：系统控制单元 (SCU) （基址 0x4008 6000） *（续）*

名称	访问类型	地址偏移	描述	复位值	EMC 启动后的复位值	UART 启动后的复位值	参考
-	-	0x0D4 - 0x0FC	保留	-	-	-	
引脚 P2_n							
SFSP2_0	R/W	0x100	引脚 P2_0 的引脚配置寄存器	0x00	0xD8	0xD1 (UART0)	表 123
SFSP2_1	R/W	0x104	引脚 P2_1 的引脚配置寄存器	0x00	0xD2	0xD1 (UART0)	表 123
SFSP2_2	R/W	0x108	引脚 P2_2 的引脚配置寄存器	0x00	0xD2		表 123
SFSP2_3	R/W	0x10C	引脚 P2_3 的引脚配置寄存器	0x00	0x00	0xD2 (UART3)	表 124
SFSP2_4	R/W	0x110	引脚 P2_4 的引脚配置寄存器	0x00	0x00	0xD2 (UART3)	表 124
SFSP2_5	R/W	0x114	引脚 P2_5 的引脚配置寄存器	0x00	0x00	0x00	表 124
SFSP2_6	R/W	0x118	引脚 P2_6 的引脚配置寄存器	0x00	0xD2	0x00	表 123
SFSP2_7	R/W	0x11C	引脚 P2_7 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP2_8	R/W	0x120	引脚 P2_8 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP2_9	R/W	0x124	引脚 P2_9 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP2_10	R/W	0x128	引脚 P2_10 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP2_11	R/W	0x12C	引脚 P2_11 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP2_12	R/W	0x130	引脚 P2_12 的引脚配置寄存器	0x00	0xD3	0x00	表 123
SFSP2_13	R/W	0x134	引脚 P2_13 的引脚配置寄存器	0x00	0xD3	0x00	表 123
-	-	0x138 - 0x17C	保留	-	-	-	-
引脚 P3_n							
SFSP3_0	R/W	0x180	引脚 P3_0 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP3_1	R/W	0x184	引脚 P3_1 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP3_2	R/W	0x188	引脚 P3_2 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP3_3	R/W	0x18C	引脚 P3_3 的引脚配置寄存器	0x00	0x00	0x00	表 125
SFSP3_4	R/W	0x190	引脚 P3_4 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP3_5	R/W	0x194	引脚 P3_5 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP3_6	R/W	0x198	引脚 P3_6 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP3_7	R/W	0x19C	引脚 P3_7 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP3_8	R/W	0x1A0	引脚 P3_8 的引脚配置寄存器	0x00	0x00	0x00	表 123
-	-	0x1A4 - 0x1FC	保留	-	-	-	
引脚 P4_n							
SFSP4_0	R/W	0x200	引脚 P4_0 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP4_1	R/W	0x204	引脚 P4_1 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 129
SFSP4_2	R/W	0x208	引脚 P4_2 的引脚配置寄存器	0x00	0x00	0x00	表 123

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 122. 寄存器简介：系统控制单元 (SCU) （基址 0x4008 6000） *（续）*

名称	访问类型	地址偏移	描述	复位值	EMC 启动后的复位值	UART 启动后的复位值	参考
SFSP4_3	R/W	0x20C	引脚 P4_3 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 129
SFSP4_4	R/W	0x210	引脚 P4_4 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 133
SFSP4_5	R/W	0x214	引脚 P4_5 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP4_6	R/W	0x218	引脚 P4_6 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP4_7	R/W	0x21C	引脚 P4_7 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP4_8	R/W	0x220	引脚 P4_8 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP4_9	R/W	0x224	引脚 P4_9 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP4_10	R/W	0x228	引脚 P4_10 的引脚配置寄存器	0x00	0x00	0x00	表 123
-	-	0x22C - 0x27C	保留	-	-	-	
引脚 P5_n							
SFSP5_0	R/W	0x280	引脚 P5_0 的引脚配置寄存器	0x00	0xD2	0x00	表 123
SFSP5_1	R/W	0x284	引脚 P5_1 的引脚配置寄存器	0x00	0xD2	0x00	表 123
SFSP5_2	R/W	0x288	引脚 P5_2 的引脚配置寄存器	0x00	0xD2	0x00	表 123
SFSP5_3	R/W	0x28C	引脚 P5_3 的引脚配置寄存器	0x00	0xD2	0x00	表 123
SFSP5_4	R/W	0x290	引脚 P5_4 的引脚配置寄存器	0x00	0xD2	0x00	表 123
SFSP5_5	R/W	0x294	引脚 P5_5 的引脚配置寄存器	0x00	0xD2	0x00	表 123
SFSP5_6	R/W	0x298	引脚 P5_6 的引脚配置寄存器	0x00	0xD2	0x00	表 123
SFSP5_7	R/W	0x29C	引脚 P5_7 的引脚配置寄存器	0x00	0xD2	0x00	表 123
-	-	0x2A0 - 0x2FC	保留	-	-	-	-
引脚 P6_n							
SFSP6_0	R/W	0x300	引脚 P6_0 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP6_1	R/W	0x304	引脚 P6_1 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP6_2	R/W	0x308	引脚 P6_2 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP6_3	R/W	0x30C	引脚 P6_3 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP6_4	R/W	0x310	引脚 P6_4 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP6_5	R/W	0x314	引脚 P6_5 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP6_6	R/W	0x318	引脚 P6_6 的引脚配置寄存器	0x00	0xD1	0x00	表 123
SFSP6_7	R/W	0x31C	引脚 P6_7 的引脚配置寄存器	0x00	0xD8	0x00	表 123
SFSP6_8	R/W	0x320	引脚 P6_8 的引脚配置寄存器	0x00	0xD8	0x00	表 123
SFSP6_9	R/W	0x324	引脚 P6_9 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP6_10	R/W	0x328	引脚 P6_10 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP6_11	R/W	0x32C	引脚 P6_11 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP6_12	R/W	0x330	引脚 P6_12 的引脚配置寄存器	0x00	0x00	0x00	表 123
-	-	0x334 - 0x37C	保留	-	-	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 122. 寄存器简介：系统控制单元 (SCU) （基址 0x4008 6000） *（续）*

名称	访问类型	地址偏移	描述	复位值	EMC 启动后的复位值	UART 启动后的复位值	参考
引脚 P7_n							
SFSP7_0	R/W	0x380	引脚 P7_0 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP7_1	R/W	0x384	引脚 P7_1 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP7_2	R/W	0x388	引脚 P7_2 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP7_3	R/W	0x38C	引脚 P7_3 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP7_4	R/W	0x390	引脚 P7_4 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 129
SFSP7_5	R/W	0x394	引脚 P7_5 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 129
SFSP7_6	R/W	0x398	引脚 P7_6 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP7_7	R/W	0x39C	引脚 P7_7 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 131
-	-	0x3A0 - 0x3FC	保留	-	-	-	-
引脚 P8_n							
SFSP8_0	R/W	0x400	引脚 P8_0 的引脚配置寄存器	0x00	0x00	0x00	表 124
SFSP8_1	R/W	0x404	引脚 P8_1 的引脚配置寄存器	0x00	0x00	0x00	表 124
SFSP8_2	R/W	0x408	引脚 P8_2 的引脚配置寄存器	0x00	0x00	0x00	表 124
SFSP8_3	R/W	0x40C	引脚 P8_3 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP8_4	R/W	0x410	引脚 P8_4 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP8_5	R/W	0x414	引脚 P8_5 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP8_6	R/W	0x418	引脚 P8_6 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP8_7	R/W	0x41C	引脚 P8_7 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP8_8	R/W	0x420	引脚 P8_8 的引脚配置寄存器	0x00	0x00	0x00	表 123
-	-	0x424 - 0x47C	保留	-	-	-	-
引脚 P9_n							
SFSP9_0	R/W	0x480	引脚 P9_0 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP9_1	R/W	0x484	引脚 P9_1 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP9_2	R/W	0x488	引脚 P9_2 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP9_3	R/W	0x49C	引脚 P9_3 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP9_4	R/W	0x490	引脚 P9_4 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP9_5	R/W	0x494	引脚 P9_5 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSP9_6	R/W	0x498	引脚 P9_6 的引脚配置寄存器	0x00	0x00	0x00	表 123
-	-	0x49C - 0x4FC	保留	-	-	-	-
引脚 PA_n							
SFSPA_0	R/W	0x500	引脚 PA_0 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPA_1	R/W	0x504	引脚 PA_1 的引脚配置寄存器	0x00	0x00	0x00	表 124
SFSPA_2	R/W	0x508	引脚 PA_2 的引脚配置寄存器	0x00	0x00	0x00	表 124

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 122. 寄存器简介：系统控制单元 (SCU) （基址 0x4008 6000） *（续）*

名称	访问类型	地址偏移	描述	复位值	EMC 启动后的复位值	UART 启动后的复位值	参考
SFSPA_3	R/W	0x50C	引脚 PA_3 的引脚配置寄存器	0x00	0x00	0x00	表 124
SFSPA_4	R/W	0x510	引脚 PA_4 的引脚配置寄存器	0x00	0xD8	0x00	表 123
-	-	0x514 - 0x57C	保留	-	-	-	-
引脚 PB_n							
SFSPB_0	R/W	0x580	引脚 PB_0 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPB_1	R/W	0x584	引脚 PB_1 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPB_2	R/W	0x588	引脚 PB_2 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPB_3	R/W	0x58C	引脚 PB_3 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPB_4	R/W	0x590	引脚 PB_4 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPB_5	R/W	0x594	引脚 PB_5 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPB_6	R/W	0x598	引脚 PB_6 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 129
-	-	0x59C - 0x5FC	保留	-	-	-	-
引脚 PC_n							
SFSPC_0	R/W	0x600	引脚 PC_0 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 131
SFSPC_1	R/W	0x604	引脚 PC_1 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPC_2	R/W	0x608	引脚 PC_2 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPC_3	R/W	0x60C	引脚 PC_3 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 131
SFSPC_4	R/W	0x610	引脚 PC_4 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPC_5	R/W	0x614	引脚 PC_5 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPC_6	R/W	0x618	引脚 PC_6 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPC_7	R/W	0x61C	引脚 PC_7 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPC_8	R/W	0x620	引脚 PC_8 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPC_9	R/W	0x624	引脚 PC_9 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPC_10	R/W	0x628	引脚 PC_10 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPC_11	R/W	0x62C	引脚 PC_11 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPC_12	R/W	0x630	引脚 PC_12 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPC_13	R/W	0x634	引脚 PC_13 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPC_14	R/W	0x638	引脚 PC_14 的引脚配置寄存器	0x00	0x00	0x00	表 123
-	-	0x63C - 0x67C	保留	-	-	-	-
引脚 PD_n							
SFSPD_0	R/W	0x680	引脚 PD_0 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPD_1	R/W	0x684	引脚 PD_1 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPD_2	R/W	0x688	引脚 PD_2 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPD_3	R/W	0x68C	引脚 PD_3 的引脚配置寄存器	0x00	0x00	0x00	表 123

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 122. 寄存器简介：系统控制单元 (SCU) （基址 0x4008 6000） *（续）*

名称	访问类型	地址偏移	描述	复位值	EMC 启动后的复位值	UART 启动后的复位值	参考
SFSPD_4	R/W	0x690	引脚 PD_4 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPD_5	R/W	0x694	引脚 PD_5 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPD_6	R/W	0x698	引脚 PD_6 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPD_7	R/W	0x69C	引脚 PD_7 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPD_8	R/W	0x6A0	引脚 PD_8 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPD_9	R/W	0x6A4	引脚 PD_9 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPD_10	R/W	0x6A8	引脚 PD_10 的引脚配置寄存器	0x00	0xD2	0x00	表 123
SFSPD_11	R/W	0x6AC	引脚 PD_11 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPD_12	R/W	0x6B0	引脚 PD_12 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPD_13	R/W	0x6B4	引脚 PD_13 的引脚配置寄存器	0x00	0xD2	0x00	表 123
SFSPD_14	R/W	0x6B8	引脚 PD_14 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPD_15	R/W	0x6BC	引脚 PD_15 的引脚配置寄存器	0x00	0xD8	0x00	表 123
SFSPD_16	R/W	0x6C0	引脚 PD_16 的引脚配置寄存器	0x00	0xD8	0x00	表 123
-	-	0x6C4 - 0x6FC	保留	-			
引脚 PE_n							
SFSPE_0	R/W	0x700	引脚 PE_0 的引脚配置寄存器	0x00	0xD8	0x00	表 123
SFSPE_1	R/W	0x704	引脚 PE_1 的引脚配置寄存器	0x00	0xD8	0x00	表 123
SFSPE_2	R/W	0x708	引脚 PE_2 的引脚配置寄存器	0x00	0xD8	0x00	表 123
SFSPE_3	R/W	0x70C	引脚 PE_3 的引脚配置寄存器	0x00	0xD8	0x00	表 123
SFSPE_4	R/W	0x710	引脚 PE_4 的引脚配置寄存器	0x00	0xD8	0x00	表 123
SFSPE_5	R/W	0x714	引脚 PE_5 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPE_6	R/W	0x718	引脚 PE_6 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPE_7	R/W	0x71C	引脚 PE_7 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPE_8	R/W	0x720	引脚 PE_8 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPE_9	R/W	0x724	引脚 PE_9 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPE_10	R/W	0x728	引脚 PE_10 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPE_11	R/W	0x72C	引脚 PE_11 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPE_12	R/W	0x730	引脚 PE_12 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPE_13	R/W	0x734	引脚 PE_13 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPE_14	R/W	0x738	引脚 PE_14 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPE_15	R/W	0x73C	引脚 PE_15 的引脚配置寄存器	0x00	0x00	0x00	表 123
-	-	0x740 - 0x77C	保留	-			
引脚 PF_n							
SFSPF_0	R/W	0x780	引脚 PF_0 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPF_1	R/W	0x784	引脚 PF_1 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPF_2	R/W	0x788	引脚 PF_2 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPF_3	R/W	0x78C	引脚 PF_3 的引脚配置寄存器	0x00	0x00	0x00	表 123

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 122. 寄存器简介：系统控制单元 (SCU) （基址 0x4008 6000）
（续）

名称	访问类型	地址偏移	描述	复位值	EMC 启动后的复位值	UART 启动后的复位值	参考
SFSPF_4	R/W	0x790	引脚 PF_4 的引脚配置寄存器	0x00	0x00	0x00	表 123
SFSPF_5	R/W	0x794	引脚 PF_5 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 131
SFSPF_6	R/W	0x798	引脚 PF_6 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 131
SFSPF_7	R/W	0x79C	引脚 PF_7 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 131
SFSPF_8	R/W	0x7A0	引脚 PF_8 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 129
SFSPF_9	R/W	0x7A4	引脚 PF_9 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 131
SFSPF_10	R/W	0x7A8	引脚 PF_10 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 129
SFSPF_11	R/W	0x7AC	引脚 PF_11 的引脚配置寄存器	0x00	0x00	0x00	表 123/ 表 131
-	-	0x7B0 - 0xBFC	保留	-	-	-	-
CLKn 引脚							
SFSCLK0	R/W	0xC00	引脚 CLK0 的引脚配置寄存器	0x00	< 待定 >	0x00	表 125
SFSCLK1	R/W	0xC04	引脚 CLK1 的引脚配置寄存器	0x00	< 待定 >	0x00	表 125
SFSCLK2	R/W	0xC08	引脚 CLK2 的引脚配置寄存器	0x00	< 待定 >	0x00	表 125
SFSCLK3	R/W	0xC0C	引脚 CLK3 的引脚配置寄存器	0x00	< 待定 >	0x00	表 125
-	-	0xC10 - 0xC7C	保留	-	-	-	-
USB1 USB1_DP/USB1_DM 引脚和 I^2C 总线开漏引脚							
SFSUSB	R/W	0xC80	引脚 USB1_DM 和 USB1_DP 的引脚配置寄存器	0x02	0x00	0x00	表 126
SFSI2C0	R/W	0xC84	I^2C0 总线引脚的引脚配置寄存器	0x00	0x00	0x00	表 127
ADC 引脚选择寄存器							
ENAIO0	R/W	0xC88	ADC0 功能选择寄存器	0x00	0x00	0x00	表 129
ENAIO1	R/W	0xC8C	ADC1 功能选择寄存器	0x00	0x00	0x00	表 131
ENAIO2	R/W	0xC90	模拟功能选择寄存器	0x00	0x00	0x00	表 133
EMC 延迟寄存器							
EMCDELAYCLK	R/W	0xD00	EMC 时钟延迟寄存器	0x00	0x00	0x00	表 134
引脚中断选择寄存器							
PINTSEL0	R/W	0xE00	引脚中断 0 至 3 的引脚中断选择寄存器。	0x00	0x00	0x00	表 135
PINTSEL1	R/W	0xE04	引脚中断 4 至 7 的引脚中断选择寄存器。	0x00	0x00	0x00	表 136

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

15.4.1 标准电平驱动引脚的引脚配置寄存器

LPC43xx 上的每个数字引脚和时钟引脚都拥有一个相关的引脚配置寄存器，可以决定引脚的功能和电气特性。每个引脚分配到的功能列示在 <tbd>。

标准电平驱动引脚的引脚配置寄存器控制以下引脚：

- P0_0 和 P0_1
- P1_0 到 P1_16 和 P1_18 到 P1_20
- P2_0 到 P2_2 和 P2_6 到 P2_13
- P3_0 到 P3_2 和 P3_4 到 P3_8
- P4_0 到 P4_10
- P5_0 到 P5_7
- P6_0 到 P6_12
- P7_0 到 P7_7
- P8_3 到 P8_8
- P9_0 到 P9_6
- PA_0 和 PA_4
- PB_0 到 PB_6
- PC_0 到 PC_14
- PE_0 到 PE_15
- PF_0 到 PF_11

表 123. 标准电平驱动引脚的引脚配置寄存器（SFS，地址0x4008 6000 (SPSP0_0)至0x4008 67AC (SFSPF_11)）位描述

位	符号	值	描述	复位值	访问类型
2:0	MODE		选择引脚功能。	0	R/W
		0x0	功能 0 （默认值）		
		0x1	功能 1		
		0x2	功能 2		
		0x3	功能 3		
		0x4	功能 4		
		0x5	功能 5		
		0x6	功能 6		
		0x7	功能 7		
3	EPD		使能管脚上的下拉电阻。	0	R/W
		0	禁用下拉。		
		1	使能下拉。中继模式需同时启用上拉与下拉。		
4	EPUN		禁用管脚上的上拉电阻。上拉电阻复位时默认为使能。	0	R/W
		0	使能上拉。为中继模式同时使能上拉电阻和下拉电阻。		
		1	禁用上拉。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 123. 标准电平驱动引脚的引脚配置寄存器（SFS，地址0x4008 6000 (SPSP0_0)至0x4008 67AC (SFSPF_11)）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
5	EHS		选择转换速率。	0	R/W
		0	慢 （中速低噪声）		
		1	快 （快速中噪声）		
6	EZI		输入缓冲使能。输入缓冲复位时默认为禁用，接收时则必须使能。	0	R/W
		0	禁用输入缓冲		
		1	使能输入缓冲		
7	ZIF		输入干扰滤波器。当时钟信号高于 30 MHz 时，禁用干扰滤波器。	0	R/W
		0	使能输入干扰滤波器		
		1	禁用输入干扰滤波器		
31:8	-		保留	-	-

15.4.2 高驱动能力引脚的引脚配置寄存器

LPC43xx 上的每个数字引脚和时钟引脚都拥有一个相关的引脚配置寄存器，可以决定引脚的功能和电气特性。每个引脚分配到的功能列示在 <tbd>。

高驱动能力引脚的引脚配置寄存器控制以下引脚：

- P1_17
- P2_3 至 P2_5
- P8_0 至 P8_2
- PA_1 至 PA_3

表 124. 高驱动能力引脚的引脚配置寄存器（SFS，地址 0x4008 60C4 (SFSP1_17) 至 0x4008 650C (SFSPA_3)）位描述

位	符号	值	描述	复位值	访问类型
2:0	MODE		选择引脚功能。	0	R/W
		0x0	功能 0 （默认值）		
		0x1	功能 1		
		0x2	功能 2		
		0x3	功能 3		
		0x4	功能 4		
		0x5	功能 5		
		0x6	功能 6		
		0x7	功能 7		
3	EPD		使能管脚上的下拉电阻。	0	R/W
		0	禁用下拉。		
		1	使能下拉。为中继模式同时使能上拉电阻和下拉电阻。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 124. 高驱动能力引脚的引脚配置寄存器（SFS，地址 0x4008 60C4 (SFSP1_17) 至 0x4008 650C (SFSPA_3)）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
4	EPUN		禁用管脚上的上拉电阻。上拉电阻复位时默认为使能。	0	R/W
		0	使能上拉。为中继模式同时使能上拉电阻和下拉电阻。		
		1	禁用上拉		
5	-		保留	-	-
6	EZI		输入缓冲使能。输入缓冲复位时默认为禁用，但将数据从 I/O 缓冲发送到管脚时必须使能。	0	R/W
		0	禁用输入缓冲		
		1	使能输入缓冲		
7	ZIF		输入干扰滤波器。当时钟信号高于 30 MHz 时，禁用干扰滤波器。	0	R/W
		0	使能输入干扰滤波器		
		1	禁用输入干扰滤波器		
9:8	EHD		选择驱动强度。	0	R/W
		0x0	标准电平驱动：4 mA 驱动强度		
		0x1	中电平驱动：8 mA 驱动强度		
		0x2	高驱动能力：14 mA 驱动强度		
		0x3	超高驱动能力：20 mA 驱动强度		
31:10	-		保留	-	-

15.4.3 高速引脚的引脚配置寄存器

LPC43xx 上的每个数字引脚和时钟引脚都拥有一个相关的引脚配置寄存器，可以决定引脚的功能和电气特性。每个引脚分配到的功能列示在 <tbd>。

该寄存器控制以下引脚：P3_3 和引脚 CLK0 到 CLK3。

表 125. 高速引脚的引脚配置寄存器（SFS，地址 0x4008 618C (SPSP3_3)； 0x4008 6C00 (SFSCLK0) 至 0x4008 6C0C (SFSCLK3)）位描述

位	符号	值	描述	复位值	访问类型
2:0	MODE		选择引脚功能。	0	R/W
		0x0	功能 0 （默认值）		
		0x1	功能 1		
		0x2	功能 2		
		0x3	功能 3		
		0x4	功能 4		
		0x5	功能 5		
		0x6	功能 6		
		0x7	功能 7		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 125. 高速引脚的引脚配置寄存器（SFS，地址 0x4008 618C (SPSP3_3)； 0x4008 6C00 (SFSCLK0) 至 0x4008 6C0C (SFSCLK3)）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
3	EPD		使能管脚上的下拉电阻。	0	R/W
		0	禁用下拉。		
		1	使能下拉。为中继模式同时使能上拉电阻和下拉电阻。		
4	EPUN		禁用管脚上的上拉电阻。上拉电阻复位时默认为使能。	0	R/W
		0	使能上拉。为中继模式同时使能上拉电阻和下拉电阻。		
		1	禁用上拉。		
5	EHS		转换速率	0	R/W
		0	快 （快速低噪声）		
		1	高速 （高速中噪声）		
6	EZI		输入缓冲使能。输入缓冲复位时默认为禁用，接收时则必须使能。	0	R/W
		0	禁用输入缓冲		
		1	使能输入缓冲		
7	ZIF		输入干扰滤波器。当时钟信号高于 30 MHz 时，禁用干扰滤波器。	0	R/W
		0	使能输入滤波器		
		1	禁用输入滤波器		
31:8	-		保留	-	-

15.4.4 USB1 引脚 USB1_DP/USB1_DM 的引脚配置寄存器

注：如果使用 USB1， USB_ESEA 位必须设为 1。

表 126. 引脚 USB1_DP/USB1_DM 寄存器引脚配置 （SFSUSB，地址 0x4008 6C80）位描述

位	符号	值	描述	复位值	访问类型
0	USB_AIM		差分数据输入 AIP/AIM。	0	R/W
		0	以全边沿速率下降		
		1	以全边沿速率上升		
1	USB_ESEA		差分输入或单端输入的控制信号。	1	R/W
		0	保留。不使用。		
		1	单端输入。使能 USB1。适用于片内全速 PHY。		
2	USB_EPD		使能下拉连接。	0	R/W
		0	下拉已断开		
		1	下拉已连接		
3	-		保留	-	-
4	USB_EPWR		功耗模式。	0	R/W
		0	掉电模式 （挂起模式）		
		1	正常模式		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 126. 引脚 USB1_DP/USB1_DM 寄存器引脚配置（SFSUSB，地址 0x4008 6C80）位描述（续）

位	符号	值	描述	复位值	访问类型
5	USB_VBUS		使能 vbus_valid 信号。该信号由 USB1 模块监控。该位可用于 VBUS 感应信号的软件去抖，或当存在 VBUS 信号但 SFSP2_5 寄存器中未连接 USB1_VBUS 功能时，用于向 USB1 控制器指示 VBUS 状态。 **注：**如果通过 SFSP2_5 寄存器使能引脚 P2_5 的 USB1_VBUS 功能，该位的设置无效。	0	R/W
		0	VBUS 信号处于低电平或不工作状态		
		1	VBUS 信号处于高电平或工作状态		
31:6	-		保留	-	-

15.4.5 开漏 I²C 总线引脚的引脚配置寄存器

表 127. 开漏 I²C 总线引脚寄存器的引脚配置 （SFSI2C0，地址 0x4008 6C84）位描述

位	符号	值	描述	复位值	访问类型
0	SCL_EFP		选择 SCL 引脚的输入干扰滤波器时间常数。	0	R/W
		0	50 ns 干扰滤波器		
		1	3 ns 干扰滤波器		
1	-		保留。始终向该位写入 0。	0	R/W
2	SCL_EHD		选择 SCL 引脚的 I2C 模式。	0	R/W
		0	标准 / 快速模式发送		
		1	超快速模式发送		
3	SCL_EZI		使能 SCL 引脚的输入接收器。使用 I2C0 时，始终向该位写入 1。	0	R/W
		0	已禁用		
		1	使能		
6:4	-		保留	-	-
7	SCL_ZIF		使能或禁用 SCL 引脚的输入干扰滤波器。滤波器时间常数由位 EFP 确定。	0	R/W
		0	使能输入滤波器		
		1	禁用输入滤波器		
8	SDA_EFP		选择 SDA 引脚的输入干扰滤波器时间常数。	0	R/W
		0	50 ns 干扰滤波器		
		1	3 ns 干扰滤波器		
9	-		保留。始终向该位写入 0。	0	R/W
10	SDA_EHD		选择 SDA 引脚的 I2C 模式。	0	R/W
		0	标准 / 快速模式发送		
		1	超快速模式发送		
11	SDA_EZI		使能 SDA 引脚的输入接收器。使用 I2C0 时，始终向该位写入 1。	0	R/W
		0	已禁用		
		1	使能		
14:12	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 127. 开漏 I²C 总线引脚寄存器的引脚配置 （SFSI2C0，地址 0x4008 6C84）位描述 （续）

位	符号	值	描述	复位值	访问类型
15	SDA_ZIF		使能或禁用 SDA 引脚的输入干扰滤波器。滤波器时间常数由位 SDA_EFP 确定。	0	R/W
		0	使能输入滤波器		
		1	禁用输入滤波器		
31:16	-		保留	-	-

15.4.6 ADC0 功能选择寄存器

对于具有数字和模拟功能的引脚，该寄存器为任何可用的数字功能选择 ADC0 输入通道。该选项对通道 ADC0_7 不适用。

此外，每种模拟功能由不受该寄存器影响的专用模拟引脚输出。

以下引脚由 ENAIO0 寄存器控制：

表 128. ENAIO0 寄存器控制的引脚

引脚	ADC 功能	ENAIO0 寄存器位
P4_3	ADC0_0	0
P4_1	ADC0_1	1
PF_8	ADC0_2	2
P7_5	ADC0_3	3
P7_4	ADC0_4	4
PF_10	ADC0_5	5
PB_6	ADC0_6	6

所有引脚都默认连接至数字功能 0，仅数字管脚可用。

要选择模拟功能，必须使用对应的 SFSP 寄存器按以下步骤设置管脚：

1. 通过将引脚多路复用（如 GPIO 功能）选择为输入模式，让输出驱动器处于三态模式。
2. 通过将 EZI 位设为 0，禁用接收器 （见表 123 或表 124）。这是默认设置。
3. 通过将 EPUN 位设为 1，禁用上拉电阻，同时将 EPD 位设为 0，禁用下拉电阻。
4. 在 ENAIO0 寄存器中，将模拟功能的对应位设为 1。

表 129. ADC0 功能选择寄存器 （ENAIO0，地址 0x4008 6C88）位描述

位	符号	值	描述	复位值	访问类型
0	ADC0_0		选择 ADC0_0	0	R/W
		0	引脚 P4_3 上选择的数字功能。		
		1	引脚 P4_3 上选择的模拟功能 ADC0_0		
1	ADC0_1		选择 ADC0_1	0	R/W
		0	引脚 P4_1 上选择的数字功能。		
		1	引脚 P4_1 上选择的模拟功能 ADC0_1。		
2	ADC0_2		选择 ADC0_2	0	R/W
		0	引脚 PF_8 上选择的数字功能。		
		1	引脚 PF_8 上选择的模拟功能 ADC0_2。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 129. ADC0 功能选择寄存器 （ENAIO0，地址 0x4008 6C88）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
3	ADC0_3		选择 ADC0_3	0	R/W
		0	引脚 P7_5 上选择的数字功能。		
		1	引脚 P7_5 上选择的模拟功能 ADC0_3。		
4	ADC0_4		选择 ADC0_4	0	R/W
		0	引脚 P7_4 上选择的数字功能。		
		1	引脚 P7_4 上选择的模拟功能 ADC0_4。		
5	ADC0_5		选择 ADC0_5	0	R/W
		0	引脚 PF_10 上选择的数字功能。		
		1	引脚 PF_10 上选择的模拟功能 ADC0_5。		
6	ADC0_6		选择 ADC0_6	0	R/W
		0	引脚 PB_6 上选择的数字功能。		
		1	引脚 PB_6 上选择的模拟功能 ADC0_6。		
31:7			保留	-	-

15.4.7 ADC1 功能选择寄存器

对于具有数字和模拟功能的引脚，该寄存器为任何可用的数字功能选择 ADC1 功能。

此外，每种模拟功能由不受该寄存器影响的专用模拟引脚输出。

以下引脚由 ENAIO1 寄存器控制：

表 130. ENAIO1 寄存器控制的引脚

引脚	ADC 功能	ENAIO1 寄存器位
PC_3	ADC1_0	0
PC_0	ADC1_1	1
PF_9	ADC1_2	2
PF_6	ADC1_3	3
PF_5	ADC1_4	4
PF_11	ADC1_5	5
P7_7	ADC1_6	6
PF_7	ADC1_7	7

所有引脚都默认连接至数字功能 0，仅数字管脚可用。

要选择模拟功能，必须使用对应的 SFSP 寄存器按以下步骤设置管脚：

1. 通过将引脚多路复用（如 GPIO 功能）选择为输入模式，让输出驱动器处于三态模式。
2. 通过将 EZI 位设为 0，禁用接收器 （见表 123 或表 124）。这是默认设置。
3. 通过将 EPUN 位设为 1，禁用上拉电阻，同时将 EPD 位设为 0，禁用下拉电阻。
4. 在 ENAIO1 寄存器中，将模拟功能的对应位设为 1。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 131. ADC1 功能选择寄存器 （ENAIO1，地址 0x4008 6C8C）位描述

位	符号	值	描述	复位值	访问类型
0	ADC1_0		选择 ADC1_0	0	R/W
		0	引脚 PC_3 上选择的数字功能。		
		1	引脚 PC_3 上选择的模拟功能 ADC1_0。		
1	ADC1_1		选择 ADC1_1	0	R/W
		0	引脚 PC_0 上选择的数字功能。		
		1	引脚 PC_0 上选择的模拟功能 ADC1_1。		
2	ADC1_2		选择 ADC1_2	0	R/W
		0	引脚 PF_9 上选择的数字功能。		
		1	引脚 PF_9 上选择的模拟功能 ADC1_2。		
3	ADC1_3		选择 ADC1_3	0	R/W
		0	引脚 PF_6 上选择的数字功能。		
		1	引脚 PF_6 上选择的模拟功能 ADC1_3。		
4	ADC1_4		选择 ADC1_4	0	R/W
		0	引脚 PF_5 上选择的数字功能。		
		1	引脚 PF_5 上选择的模拟功能 ADC1_4。		
5	ADC1_5		选择 ADC1_5	0	R/W
		0	引脚 PF_11 上选择的数字功能。		
		1	引脚 PF_11 上选择的模拟功能 ADC1_5。		
6	ADC1_6		选择 ADC1_6	0	R/W
		0	引脚 P7_7 上选择的数字功能。		
		1	引脚 P7_7 上选择的模拟功能 ADC1_6。		
7	ADC1_7		选择 ADC1_7。	0	R/W
		0	引脚 PF_7 上选择的数字功能。		
		1	引脚 PF_7 上选择的模拟功能 ADC1_7。		
31:8			保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

15.4.8 模拟功能选择寄存器

对于具有数字和模拟功能的引脚，该寄存器为任何可用的数字功能选择模拟 DAC 和带隙功能。

此外， DAC 功能由不受该寄存器影响的专用模拟引脚输出。

以下引脚由 ENAIO2 寄存器控制：

表 132. ENAIO2 寄存器控制的引脚

引脚	ADC 功能	ENAIO2 寄存器位
P4_4	DAC	0
PF_7	BG （带隙输出）	4

所有引脚都默认连接至数字功能 0，仅数字管脚可用。

要选择模拟功能，必须使用对应的 SFSP 寄存器按以下步骤设置管脚：

1. 通过将引脚多路复用（如 GPIO 功能）选择为输入模式，让输出驱动器处于三态模式。
2. 通过将 EZI 位设为 0，禁用接收器 （见表 123 或表 124）。这是默认设置。
3. 通过将 EPUN 位设为 1，禁用上拉电阻，同时将 EPD 位设为 0，禁用下拉电阻。
4. 在 ENAIO2 寄存器中，将模拟功能的对应位设为 1。

表 133. 模拟功能选择寄存器 （ENAIO2，地址 0x4008 6C90）位描述

位	符号	值	描述	复位值	访问类型
0	DAC		选择 DAC	0	R/W
		0	引脚 P4_4 上选择的数字功能。		
		1	引脚 P4_4 上选择的模拟功能 DAC。		
3:1			保留	-	-
4	BG		选择带隙输出。要测量带隙，需禁用引脚 PF_7 上的上拉电阻并将 PF_7 连接到数字管脚。测量带隙时，不要使用数字管脚或线路板上的 ADC1_7（见第 15.4.8.1 节）。	0	R/W
		0	引脚 PF_7 上选择的数字功能。		
		1	引脚 PF_7 上选择的带隙输出。		
31:5			保留	-	-

15.4.8.1 测量带隙

要测量带隙，需按以下步骤设置引脚 PF_7 的引脚配置和 ADC 功能选择寄存器：

1. 禁用上拉电阻和输入缓冲：将 0x4008 679C 的寄存器 SFSPF_7 设为 0x10。
2. 将 ADC1_7 输入连接到数字管脚：将 0x4008 6C8C 的寄存器 ENAIO1 设为 0x80。
3. 将带隙连接到数字管脚：将 0x4008 6C90 的寄存器 ENAIO2 设为 0x10。
4. 不要连接线路板上的引脚 PF_7。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

15.4.9 EMC 时钟延迟寄存器

此寄存器为 EMC 时钟输出提供可编程的延迟。所有 EMC_CLKn 时钟输出的延迟都相同，并从 0 开始以大约 0.5 ns 步幅增加（CLK_DELAY = 0 到 3.5 ns (CLK_DELAY = 0x7777)）。延迟的确切值随温度和处理过程而异。

表 134. EMC 功能选择寄存器 （EMCDELAYCLK，地址 0x4008 6D00）位描述

位	符号	描述	复位值	访问类型
15:0	CLK_DELAY	EMC_CLKn SDRAM 时钟输出延迟。 0x0 = 无延迟 0x1111 ≈ 0.5 ns 延迟 0x2222 ≈ 1.0 ns 延迟 0x3333 ≈ 1.5 ns 延迟 0x4444 ≈ 2.0 ns 延迟 0x5555 ≈ 2.5 ns 延迟 0x6666 ≈ 3.0 ns 延迟 0x7777 ≈ 3.5 ns 延迟	0	R/W
31:16	-	保留。切勿对寄存器保留位写入 1。	-	-

15.4.10 引脚中断选择寄存器 0

该寄存器从所有端口的所有 GPIO 引脚中选出一个引脚，作为引脚中断 0 至 3 的源。

示例：对于引脚中断 1，INTPIN1 = 0xA 和 PORTSEL1 = 1 选择位于引脚 P2_9 的 GPIO 引脚 GPIO1[10] 以生成中断。每个引脚中断必须在 NVIC 中使能。

如需使能每个引脚中断并配置其边沿或电平触发性，使用 GPIO 引脚中断寄存器 （见第 17.4.1 节）。

表 135. 引脚中断选择寄存器 0 （PINTSEL0，地址 0x4008 6E00）位描述

位	符号	值	描述	复位值
4:0	INTPIN0		Pint 中断 0：选择该寄存器的 PORTSEL0 位所选择的 GPIO 端口中的引脚编号。	0
7:5	PORTSEL0		引脚中断 0：为该寄存器的 INTPIN0 位要选择的引脚编号选择端口。	0
		0x0	GPIO 端口 0	
		0x1	GPIO 端口 1	
		0x2	GPIO 端口 2	
		0x3	GPIO 端口 3	
		0x4	GPIO 端口 4	
		0x5	GPIO 端口 5	
		0x6	GPIO 端口 6	
		0x7	GPIO 端口 7	
12:8	INTPIN1		Pint 中断 1：选择该寄存器的 PORTSEL1 位所选择的 GPIO 端口中的引脚编号。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 135. 引脚中断选择寄存器 0 （PINTSEL0，地址 0x4008 6E00）位描述 *（续）*

位	符号	值	描述	复位值
15:13	PORTSEL1		引脚中断 1：为该寄存器的 INTPIN1 位要选择的引脚编号选择端口。	0
		0x0	GPIO 端口 0	
		0x1	GPIO 端口 1	
		0x2	GPIO 端口 2	
		0x3	GPIO 端口 3	
		0x4	GPIO 端口 4	
		0x5	GPIO 端口 5	
		0x6	GPIO 端口 6	
		0x7	GPIO 端口 7	
20:16	INTPIN2		Pint 中断 2：选择该寄存器的 PORTSEL2 位所选择的 GPIO 端口中的引脚编号。	0
23:21	PORTSEL2		引脚中断 2：为该寄存器的 INTPIN2 位要选择的引脚编号选择端口。	0
		0x0	GPIO 端口 0	
		0x1	GPIO 端口 1	
		0x2	GPIO 端口 2	
		0x3	GPIO 端口 3	
		0x4	GPIO 端口 4	
		0x5	GPIO 端口 5	
		0x6	GPIO 端口 6	
		0x7	GPIO 端口 7	
28:24	INTPIN3		Pint 中断 3：选择该寄存器的 PORTSEL3 位所选择的 GPIO 端口中的引脚编号。	0
31:29	PORTSEL3		引脚中断 3：为该寄存器的 INTPIN3 位要选择的引脚编号选择端口。	0
		0x0	GPIO 端口 0	
		0x1	GPIO 端口 1	
		0x2	GPIO 端口 2	
		0x3	GPIO 端口 3	
		0x4	GPIO 端口 4	
		0x5	GPIO 端口 5	
		0x6	GPIO 端口 6	
		0x7	GPIO 端口 7	

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

15.4.11 引脚中断选择寄存器 1

该寄存器从所有端口的所有 GPIO 引脚中选出一个引脚，作为引脚中断 4 至 7 的源。

示例：对于引脚中断 4，INTPIN4 = 0xA 和 PORTSEL4 = 1 选择位于引脚 P2_9 的 GPIO 引脚 GPIO1[10] 以生成中断。每个引脚中断必须使用中断槽 32 至 39 在 NVIC 中使能。

如需使能每个引脚中断并配置其边沿或电平触发性，使用 GPIO 引脚中断寄存器（见第 17.4.1 节）。

表 136. 引脚中断选择寄存器 1（PINTSEL1，地址 0x4008 6E04）位描述

位	符号	值	描述	复位值
6:0	INTPIN4		Pint 中断 4：选择该寄存器的 PORTSEL4 位所选择的 GPIO 端口中的引脚编号。	0
7:5	PORTSEL4		引脚中断 4：为该寄存器的 INTPIN4 位要选择的引脚编号选择端口。	0
		0x0	GPIO 端口 0	
		0x1	GPIO 端口 1	
		0x2	GPIO 端口 2	
		0x3	GPIO 端口 3	
		0x4	GPIO 端口 4	
		0x5	GPIO 端口 5	
		0x6	GPIO 端口 6	
		0x7	GPIO 端口 7	
12:8	INTPIN5		Pint 中断 5：选择该寄存器的 PORTSEL5 位所选择的 GPIO 端口中的引脚编号。	0
15:13	PORTSEL5		引脚中断 5：为该寄存器的 INTPIN5 位要选择的引脚编号选择端口。	0
		0x0	GPIO 端口 0	
		0x1	GPIO 端口 1	
		0x2	GPIO 端口 2	
		0x3	GPIO 端口 3	
		0x4	GPIO 端口 4	
		0x5	GPIO 端口 5	
		0x6	GPIO 端口 6	
		0x7	GPIO 端口 7	
20:16	INTPIN6		Pint 中断 6：选择该寄存器的 PORTSEL6 位所选择的 GPIO 端口中的引脚编号。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 136. 引脚中断选择寄存器 1 （PINTSEL1，地址 0x4008 6E04）位描述 *（续）*

位	符号	值	描述	复位值
23:21	PORTSEL6		引脚中断 6：为该寄存器的 INTPIN6 位要选择的引脚编号选择端口。	0
		0x0	GPIO 端口 0	
		0x1	GPIO 端口 1	
		0x2	GPIO 端口 2	
		0x3	GPIO 端口 3	
		0x4	GPIO 端口 4	
		0x5	GPIO 端口 5	
		0x6	GPIO 端口 6	
		0x7	GPIO 端口 7	
28:24	INTPIN7		Pint 中断 7：选择该寄存器的 PORTSEL7 位所选择的 GPIO 端口中的引脚编号。	0
31:29	PORTSEL7		引脚中断 7：为该寄存器的 INTPIN7 位要选择的引脚编号选择端口。	0
		0x0	GPIO 端口 0	
		0x1	GPIO 端口 1	
		0x2	GPIO 端口 2	
		0x3	GPIO 端口 3	
		0x4	GPIO 端口 4	
		0x5	GPIO 端口 5	
		0x6	GPIO 端口 6	
		0x7	GPIO 端口 7	

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

16.1 本章导读

注：VADC 模块在 LPC4350/30/20/10 修订版 “A” 上不可用。

16.2 基本配置

GIMA 的配置如下：

- 参见表 137 以了解时钟与电源控制。
- GIMA 通过 BUS_RST （复位号 8）复位。请勿在正常操作期间复位 GIMA
- GIMA 输出连接至定时器、 SCT、 ADC 和事件路由外围设备 （参见图 28）。
- 如需配置定时器和SCT的GIMA输入，设置CREG6中的CTOUTCTRL位（表43）。该位控制使用 SCT 输出或者定时器匹配输出，还是仅使用 SCT 输出。

表 137. GIMA 时钟和电源控制

	基础时钟	分支时钟	最大频率
GIMA 寄存器接口的时钟	BASE_M4_CLK	CLK_M4_BUS	204 MHz

16.3 简介

全局输入多路复用器阵列 (GIMA) 将事件连接到各种事件启动外设，如 ADC、 SCT 或定时器。

每个 GIMA 输出都连接到一个外设功能 （例如，定时器捕获输入或 ADC 转换启动器输入）并通过一个寄存器配置 （选择事件启动器并配置时钟同步）。

例如，ADC 转换可在 SCT 输出或者定时器匹配输出上启动。如需选择启动器事件，使用连接到 ADC0 和 ADC1 启动 0 转换输入的 GIMA 输出 28。相应的 GIMA 输出寄存器 ADCSTART0_IN 选择 SCT 输出 15 或者定时器 0 的匹配输出 0 作为转换寄存器 （参见表 138）。

16.3.1 GIMA 事件输入选择

可启动外设功能的事件 （例如， ADC 转换或定时器捕获）可从以下源选择：

- 定时器捕获引脚
- SCT 输入引脚
- 定时器 0/1/2/3 匹配输出
- SCT 输出
- I2S0/1 MWS 信号
- USART0/2/3 RX/TX 工作信号
- USB0/1 SOF 信号

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

以下外设功能连接到 GIMA 输出：

- 定时器 0/1/2/3 捕获输入
- SCT 输入
- ADC0/1 启动转换
- 事件路由器事件

图 28 显示通过 GIMA 连接的外设。有关详情，请参阅表 138。

图 28. GIMA 和外设之间的连接

表 138 显示 GIMA 输出编号、 GIMA 输出连接的外设功能以及可为该 GIMA 输出选择的事件。对于从外部引脚产生的信号，从引脚输出中选择一个引脚 （可能有多个引脚）并在引脚配置寄存器中编程相应的引脚功能。来自外部引脚的输入包括任何定时器捕获引脚（T0/1/2/3_CAPx 和 SCT 输入引脚 CTIN_x）。

每个 GIMA 输出使用一个寄存器 （参见表 139）控制同步阶段并从表 138 中显示的所有可用事件启动器信号的子集选择一个事件。

表 138. GIMA 输出

GIMA 输出	外设	GIMA 输入				参考
0	T0 捕获通道 0	引脚 CTIN_0	SGPIO3	引脚 T0_CAP0	-	表 140
1	T0 捕获通道 1	引脚 CTIN_1	USART2 TX 有效	引脚 T0_CAP1	-	表 141
2	T0 捕获通道 2	引脚 CTIN_2	SGPIO3_DIV	引脚 T0_CAP2	-	表 142
3	T0 捕获通道 3	SCT 输出 15 或 T3 匹配通道 3	引脚 T0_cap3	T3 匹配通道 3	-	表 143
4	T1 捕获通道 0	引脚 CTIN_0	SGPIO12	引脚 T1_CAP0	-	表 144
5	T1 捕获通道 1	引脚 CTIN_3	USART0 TX 有效	引脚 T1_CAP1	-	表 145
6	T1 捕获通道 2	引脚 CTIN_4	USART0 RX 有效	引脚 T1_CAP2	-	表 146
7	T1 捕获通道 3	SCT 输出 3 或 T0 匹配通道 3	引脚 T1_CAP3	T0 匹配通道 3	-	表 147

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 138. GIMA 输出 *（续）*

GIMA 输出	外设	GIMA 输入				参考
8	T2 捕获通道 0	引脚 CTIN_0	SGPIO12_DIV	引脚 T2_CAP0	-	表 148
9	T2 捕获通道 1	引脚 CTIN_1	USART2 TX 有效	I2S1_RX_MSW	引脚 T2_CAP1	表 149
10	T2 捕获通道 2	引脚 CTIN_5	USART2 RX 有效	I2S1_TX_MSW	引脚 T2_CAP2	表 150
11	T2 捕获通道 3	SCT 输出 7 或 T1 匹配通道 3	引脚 T2_CAP3		-	表 151
12	T3 捕获通道 0	引脚 CTIN_0	I2S0_RX_MSW	引脚 T3_CAP0	-	表 152
13	T3 捕获通道 1	引脚 CTIN_6	USART3 TX 有效	I2S0_TX_MSW	引脚 T3_CAP1	表 153
14	T3 捕获通道 2	引脚 CTIN_7	USART3 RX 有效	SOF0	引脚 T3_CAP2	表 154
15	T3 捕获通道 3	SCT 输出 11 或 T2 匹配通道 3	SOF1	引脚 T3_CAP3	-	表 155
16	SCT 输入 0	引脚 CTIN_0	SGPIO3	SGPIO3_DIV	-	表 156
17	SCT 输入 1	引脚 CTIN_1	USART2 TX 有效	SGPIO12	-	表 157
18	SCT 输入 2	引脚 CTIN_2	SGPIO12	SGPIO12_DIV	-	表 158
19	SCT 输入 3	引脚 CTIN_3	USART0 TX 有效	I2S1_RX_MSW	I2S1_TX_MSW	表 159
20	SCT 输入 4	引脚 CTIN_4	USART0 RX 有效	I2S1_RX_MSW	I2S1_TX_MSW	表 160
21	SCT 输入 5	引脚 CTIN_5	USART2 TX 有效	SGPIO12_DIV	-	表 161
22	SCT 输入 6	引脚 CTIN_6	USART3 TX 有效	I2S0_RX_MSW	I2S0_TX_MSW	表 162
23	SCT 输入 7	引脚 CTIN_7	USART3 RX 有效	SOF0	SOF1	表 163
24	VADC 启动器	参见表 164				表 164
25	事件路由器输入 13	SCT 输出 2 或 T0 匹配通道 2	SGPIO3	T0 匹配通道 2	-	表 165
26	事件路由器输入 14	SCT 输出 6 或 T1 匹配通道 2	SGPIO12	T1 匹配通道 2	-	表 166
27	事件路由器输入 16	SCT 输出 14 或 T3 匹配通道 2	T3 匹配通道 2	-	-	表 167
28	ADC 启动 0 输入（ADC CR 寄存器 START 位 = 0x2）	SCT 输出 15 或 T3 匹配通道 3	T0 匹配通道 0	-	-	表 168
29	ADC 启动 1 输入（ADC CR 寄存器位 START = 0x3）	SCT 输出 8 或 T2 匹配通道 0	T2 匹配通道 0	-	-	表 169

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

16.3.2 GIMA 时钟同步

时钟同步对每个 GIMA 输出的控制分为 5 个步骤 （图 29）：

1. 输入选择
2. 输入反转：将源和目标之间的通道反转。
3. 异步捕获
4. 与外围设备时钟同步
5. 脉冲生成

图 29. GIMA 输入阶段

16.4 寄存器描述

表 139. 寄存器简介：GIMA （基址：0x400C 7000）

名称	访问类型	地址偏移	描述	复位值	参考
CAP0_0_IN	R/W	0x000	定时器0 CAP0_0捕获输入多路复用器（GIMA输出0）	0	表 140
CAP0_1_IN	R/W	0x004	定时器0 CAP0_1捕获输入多路复用器（GIMA输出1）	0	表 141
CAP0_2_IN	R/W	0x008	定时器0 CAP0_2捕获输入多路复用器（GIMA输出2）	0	表 142
CAP0_3_IN	R/W	0x00C	定时器0 CAP0_3捕获输入多路复用器（GIMA输出3）	0	表 143
CAP1_0_IN	R/W	0x010	定时器1 CAP1_0捕获输入多路复用器（GIMA输出4）	0	表 144
CAP1_1_IN	R/W	0x014	定时器1 CAP1_1捕获输入多路复用器（GIMA输出5）	0	表 145
CAP1_2_IN	R/W	0x018	定时器1 CAP1_2捕获输入多路复用器（GIMA输出6）	0	表 146
CAP1_3_IN	R/W	0x01C	定时器1 CAP1_3捕获输入多路复用器（GIMA输出7）	0	表 147
CAP2_0_IN	R/W	0x020	定时器2 CAP2_0捕获输入多路复用器（GIMA输出8）	0	表 148
CAP2_1_IN	R/W	0x024	定时器2 CAP2_1捕获输入多路复用器（GIMA输出9）	0	表 149
CAP2_2_IN	R/W	0x028	定时器 2 CAP2_2 捕获输入多路复用器 （GIMA 输出 10）	0	表 150
CAP2_3_IN	R/W	0x02C	定时器 2 CAP2_3 捕获输入多路复用器 （GIMA 输出 11）	0	表 151
CAP3_0_IN	R/W	0x030	定时器3 CAP3_0捕获输入多路复用器（GIMA输出12)	0	表 152
CAP3_1_IN	R/W	0x034	定时器3 CAP3_1捕获输入多路复用器（GIMA输出13)	0	表 153
CAP3_2_IN	R/W	0x038	定时器3 CAP3_2捕获输入多路复用器（GIMA输出14)	0	表 154
CAP3_3_IN	R/W	0x03C	定时器3 CAP3_3捕获输入多路复用器（GIMA输出15)	0	表 155
CTIN_0_IN	R/W	0x040	SCT CTIN_0 捕获输入多路复用器 （GIMA 输出 16）	0	表 156

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 139. 寄存器简介：GIMA （基址：0x400C 7000）*（续）*

名称	访问类型	地址偏移	描述	复位值	参考
CTIN_1_IN	R/W	0x044	SCT CTIN_1 捕获输入多路复用器 （GIMA 输出 17）	0	表 157
CTIN_2_IN	R/W	0x048	SCT CTIN_2 捕获输入多路复用器 （GIMA 输出 18）	0	表 158
CTIN_3_IN	R/W	0x04C	SCT CTIN_3 捕获输入多路复用器 （GIMA 输出 19）	0	表 159
CTIN_4_IN	R/W	0x050	SCT CTIN_4 捕获输入多路复用器 （GIMA 输出 20）	0	表 160
CTIN_5_IN	R/W	0x054	SCT CTIN_5 捕获输入多路复用器 （GIMA 输出 21）	0	表 161
CTIN_6_IN	R/W	0x058	SCT CTIN_6 捕获输入多路复用器 （GIMA 输出 22）	0	表 162
CTIN_7_IN	R/W	0x05C	SCT CTIN_7 捕获输入多路复用器 （GIMA 输出 23）	0	表 163
VADC_TRIGGER_IN	R/W	0x060	VADC 启动输入多路复用器 （GIMA 输出 24）	0	表 164
EVENTROUTER_13_IN	R/W	0x064	事件路由器输入 13 多路复用器 （GIMA 输出 25）	0	表 165
EVENTROUTER_14_IN	R/W	0x068	事件路由器输入 14 多路复用器 （GIMA 输出 26）	0	表 166
EVENTROUTER_16_IN	R/W	0x06C	事件路由器输入 16 多路复用器 （GIMA 输出 27）	0	表 167
ADCSTART0_IN	R/W	0x070	ADC 启动 0 输入多路复用器 （GIMA 输出 28）	0	表 168
ADCSTART1_IN	R/W	0x074	ADC 启动 1 输入多路复用器 （GIMA 输出 29）	0	表 169

16.4.1 定时器 0 CAP0_0 捕获输入多路复用器 (CAP0_0_IN)

表 140. 定时器 0 CAP0_0 捕获输入多路复用器 （CAP0_0_IN，地址 0x400C 7000）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTIN_0	
		0x1	SGPIO3	
		0x2	T0_CAP0	
31:8	-		保留	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

16.4.2 定时器 0 CAP0_1 捕获输入多路复用器 (CAP0_1_IN)

表 141. 定时器 0 CAP0_1 捕获输入多路复用器 （CAP0_1_IN，地址 0x400C 7004）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTIN_1	
		0x1	USART2 TX 有效	
		0x2	T0_CAP1	
31:8	-		保留	-

16.4.3 定时器 0 CAP0_2 捕获输入多路复用器 (CAP0_2_IN)

表 142. 定时器 0 CAP0_2 捕获输入多路复用器 （CAP0_2_IN，地址 0x400C 7008）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTIN_2	
		0x1	SGPIO3_DIV	
		0x2	T0_CAP2	
31:8	-		保留	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

16.4.4 定时器 0 CAP0_3 捕获输入多路复用器 (CAP0_3_IN)

表 143. 定时器 0 CAP0_3 捕获输入多路复用器 （CAP0_3_IN，地址 0x400C 700C）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTOUT_15 或 T3_MAT3	
		0x1	T0_CAP3	
		0x2	T3_MAT3	
31:8	-		保留	-

16.4.5 定时器 1 CAP1_0 捕获输入多路复用器 (CAP1_0_IN)

表 144. 定时器 1 CAP1_0 捕获输入多路复用器 （CAP1_0_IN，地址 0x400C 7010）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTIN_0	
		0x1	SGPIO12	
		0x2	T1_CAP0	
31:8	-		保留	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

16.4.6 定时器 1 CAP1_1 捕获输入多路复用器 (CAP1_1_IN)

表 145. 定时器 1 CAP1_1 捕获输入多路复用器 （CAP1_1_IN，地址 0x400C 7014）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTIN_3	
		0x1	USART0 TX 有效	
		0x2	T1_CAP1	
31:8	-		保留	-

16.4.7 定时器 1 CAP1_2 捕获输入多路复用器 (CAP1_2_IN)

表 146. 定时器 1 CAP1_2 捕获输入多路复用器 （CAP1_2_IN，地址 0x400C 7018）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTIN_4	
		0x1	USART0 RX 有效	
		0x2	T1_CAP2	
31:8	-		保留	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

16.4.8 定时器 1 CAP1_3 捕获输入多路复用器 (CAP1_3_IN)

表 147. 定时器 1 CAP1_3 捕获输入多路复用器 （CAP1_3_IN，地址 0x400C 701C）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTOUT_3 或 T0_MAT3	
		0x1	T1_CAP3	
		0x2	T0_MAT3	
31:8	-		保留	-

16.4.9 定时器 2 CAP2_0 捕获输入多路复用器 (CAP2_0_IN)

表 148. 定时器 2 CAP2_0 捕获输入多路复用器 （CAP2_0_IN，地址 0x400C 7020）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x4 至 0xF 保留。	0
		0x0	CTIN_0	
		0x1	SGPIO12_DIV	
		0x2	T2_CAP0	
31:8	-		保留	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

16.4.10 定时器 2 CAP2_1 捕获输入多路复用器 (CAP2_1_IN)

表 149. 定时器 2 CAP2_1 捕获输入多路复用器 （CAP2_1_IN，地址 0x400C 7024）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x4 至 0xF 保留。	0
		0x0	CTIN_1	
		0x1	USART2 TX 有效	
		0x2	<tbd> - I2S1_RX_MWS	
		0x3	T2_CAP1	
31:8	-		保留	-

16.4.11 定时器 2 CAP2_2 捕获输入多路复用器 (CAP2_2_IN)

表 150. 定时器 2 CAP2_2 捕获输入多路复用器 （CAP2_2_IN，地址 0x400C 7028）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 150. 定时器 2 CAP2_2 捕获输入多路复用器 （CAP2_2_IN，地址 0x400C 7028）位描述 *（续）*

位	符号	值	描述	复位值
7:4	SELECT		选择输入。值 0x4 至 0xF 保留。	0
		0x0	CTIN_5	
		0x1	USART2 RX 有效	
		0x2	<tbd> - I2S1_TX_MWS	
		0x3	T2_CAP2	
31:8	-		保留	-

16.4.12 定时器 2 CAP2_3 捕获输入多路复用器 (CAP2_3_IN)

表 151. 定时器 2 CAP2_3 捕获输入多路复用器 （CAP2_3_IN，地址 0x400C 702C）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTOUT_7 或 T1_MAT3	
		0x1	T2_CAP3	
		0x2	T1_MAT3	
31:8	-		保留	-

16.4.13 定时器 3 CAP3_0 捕获输入多路复用器 (CAP3_0_IN)

表 152. 定时器 3 CAP3_0 捕获输入多路复用器 （CAP3_0_IN，地址 0x400C 7030）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 152. 定时器 3 CAP3_0 捕获输入多路复用器 （CAP3_0_IN，地址 0x400C 7030）位描述 *（续）*

位	符号	值	描述	复位值
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTIN_0	
		0x1	<tbd> I2S0_RX_MWS	
		0x2	T3_CAP0	
31:8	-		保留	-

16.4.14 定时器 3 CAP3_1 捕获输入多路复用器 (CAP3_1_IN)

表 153. 定时器 3 CAP3_1 捕获输入多路复用器 （CAP3_1_IN，地址 0x400C 7034）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x4 至 0xF 保留。	0
		0x0	CTIN_6	
		0x1	USART3 TX 有效	
		0x2	待定 - I2S0_TX_MWS	
		0x3	T3_CAP1	
31:8	-		保留	-

16.4.15 定时器 3 CAP3_2 捕获输入多路复用器 (CAP3_2_IN)

表 154. 定时器 3 CAP3_2 捕获输入多路复用器 （CAP3_2_IN，地址 0x400C 7038）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 154. 定时器 3 CAP3_2 捕获输入多路复用器 （CAP3_2_IN，地址 0x400C 7038）位描述 *（续）*

位	符号	值	描述	复位值
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x4 至 0xF 保留。	0
		0x0	CTIN_7	
		0x1	USART3 RX 有效	
		0x2	SOF0 （帧起始 USB0）	
		0x3	T3_CAP2	
31:8	-		保留	-

16.4.16 定时器 3 CAP3_3 捕获输入多路复用器 (CAP3_3_IN)

表 155. 定时器 3 CAP3_3 捕获输入多路复用器 （CAP3_3_IN，地址 0x400C 703C）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x4 至 0xF 保留。	0
		0x0	CTOUT11 或 T2_MAT3	
		0x1	SOF1 （帧起始 USB1）	
		0x2	T3_CAP3	
		0x3	T2_MAT3	
31:8	-		保留	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

16.4.17 SCT CTIN_0 捕获输入多路复用器 (CTIN_0_IN)

表 156. SCT CTIN_0 捕获输入多路复用器 （CTIN_0_IN，地址 0x400C 7040）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTIN_0	
		0x1	SGPIO3	
		0x2	SGPIO3_DIV	
31:8	-		保留	-

16.4.18 SCT CTIN_1 捕获输入多路复用器 (CTIN_1_IN)

表 157. SCT CTIN_1 捕获输入多路复用器 （CTIN_1_IN，地址 0x400C 7044）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTIN_1	
		0x1	USART2 TX 有效	
		0x2	SGPIO12	
31:8	-		保留	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

16.4.19 SCT CTIN_2 捕获输入多路复用器 (CTIN_2_IN)

表 158. SCT CTIN_2 捕获输入多路复用器 （CTIN_2_IN，地址 0x400C 7048）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTIN_2	
		0x1	SGPIO12	
		0x2	SGPIO12_DIV	
31:8	-		保留	-

16.4.20 SCT CTIN_3 捕获输入多路复用器 (CTIN_3_IN)

表 159. SCT CTIN_3 捕获输入多路复用器 （CTIN_3_IN，地址 0x400C 704C）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x4 至 0xF 保留。	0
		0x0	CTIN_3	
		0x1	USART0 TX 有效	
		0x2	保留	
		0x3	保留	
31:8	-		保留	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

16.4.21 SCT CTIN_4 捕获输入多路复用器 (CTIN_4_IN)

表 160. SCT CTIN_4 捕获输入多路复用器 （CTIN_4_IN，地址 0x400C 7050）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x4 至 0xF 保留。	0
		0x0	CTIN_4	
		0x1	USART0 RX 有效	
		0x2	<tbd> - I2S1_RX_MWS1	
		0x3	<tbd> - I2S1_TX_MWS1	
31:8	-		保留	-

16.4.22 SCT CTIN_5 捕获输入多路复用器 (CTIN_5_IN)

表 161. SCT CTIN_5 捕获输入多路复用器 （CTIN_5_IN，地址 0x400C 7054）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTIN_5	
		0x1	USART2 RX 有效	
		0x2	SGPIO12_DIV	
31:8	-		保留	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

16.4.23 SCT CTIN_6 捕获输入多路复用器 (CTIN_6_IN)

表 162. SCT CTIN_6 捕获输入多路复用器 （CTIN_6_IN，地址 0x400C 7058）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x4 至 0xF 保留。	0
		0x0	CTIN_6	
		0x1	USART3 TX 有效	
		0x2	<tbd> - I2S0_RX_MWS	
		0x3	<tbd> - I2S0_TX_MWS	
31:8	-		保留	-

16.4.24 SCT CTIN_7 捕获输入多路复用器 (CTIN_7_IN)

表 163. SCT CTIN_7 捕获输入多路复用器 （CTIN_7_IN，地址 0x400C 705C）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 163. SCT CTIN_7 捕获输入多路复用器 （CTIN_7_IN，地址 0x400C 705C）位描述 *（续）*

位	符号	值	描述	复位值
7:4	SELECT		选择输入。值 0x4 至 0xF 保留。	0
		0x0	CTIN_7	
		0x1	USART3 RX 有效	
		0x2	SOF0 （帧起始 USB0）	
		0x3	SOF1 （帧起始 USB1）	
31:8	-		保留	-

16.4.25 VADC 启动输入多路复用器 (VADC_TRIGGER_IN)

表 164. ADC 启动输入多路复用器 （VADC_TRIGGER_IN，地址 0x400C 7060）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0xA 至 0xF 保留。	0
		0x0	GPIO6[28]	
		0x1	GPIO5[3]	
		0x2	SGPIO10	
		0x3	SGPIO12	
		0x4	保留	
		0x5	MCOB2	
		0x6	CTOUT_0 或 T0_MAT0	
		0x7	CTOUT_8 或 T2_MAT0	
		0x8	T0_MAT0	
		0x9	T2_MAT0	
31:8	-		保留	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

16.4.26 事件路由器输入 13 多路复用器 (EVENTROUTER_13_IN)

表 165. 事件路由器输入 13 多路复用器 （EVENTROUTER_13_IN，地址 0x400C 7064）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTOUT_2 或 T0_MAT2	
		0x1	SGPIO3	
		0x2	T0_MAT2	
31:8	-		保留	-

16.4.27 事件路由器输入 14 多路复用器 (EVENTROUTER_14_IN)

表 166. 事件路由器输入 14 多路复用器 （EVENTROUTER_14_IN，地址 0x400C 7068）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x3 至 0xF 保留。	0
		0x0	CTOUT_6 或 T1_MAT2	
		0x1	SGPIO12	
		0x2	T1_MAT2	
31:8	-		保留	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

16.4.28 事件路由器输入 16 多路复用器 (EVENTROUTER_16_IN)

表 167. 事件路由器输入 16 多路复用器 （EVENTROUTER_16_IN，地址 0x400C 706C）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x2 至 0xF 保留。	0
		0x0	CTOUT_14 或 T3_MAT2	
		0x1	T3_MAT2	
31:8	-		保留	-

16.4.29 ADC 启动 0 输入多路复用器 (ADCSTART0_IN)

表 168. ADC 启动 0 输入多路复用器 （ADCSTART0_IN，地址 0x400C 7070）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x2 至 0xF 保留。	0
		0x0	CTOUT_15 或 T3_MAT3	
		0x1	T0_MAT0	
31:8	-		保留	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

16.4.30 ADC 启动 1 输入多路复用器 (ADCSTART1_IN)

表 169. ADC 启动 1 输入多路复用器 （ADCSTART1_IN，地址 0x400C 7074）位描述

位	符号	值	描述	复位值
0	INV		反转输入	0
		0	不反转。	
		1	已反转输入。	
1	EDGE		使能上升沿检测	0
		0	不检测边沿。	
		1	上升沿检测已使能。	
2	SYNCH		使能同步	0
		0	禁用同步。	
		1	使能同步。	
3	PULSE		使能单脉冲生成。	0
		0	禁用单脉冲生成。	
		1	使能单脉冲生成。	
7:4	SELECT		选择输入。值 0x2 至 0xF 保留。	0
		0x0	CTOUT_8 或 T2_MAT0	
		0x1	T2_MAT0	
31:8	-		保留	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

17.1 本章导读

对于每个 GPIO 端口，所有 GPIO 寄存器位描述最多涵盖 31 个引脚。并非所有引脚均可用，这要取决于封装类型；另外， GPIO 寄存器中的对应位保留 （参见表 170）。

表 170. 不同引脚封装的 GPIO 引脚

	LBGA256	TFBGA180	TFBGA100	LQFP208	LQFP144	LQFP100
GPIO 端口 0	GPIO0[15:0]	GPIO0[15:0]	GPIO0[4:0]; GPIO0[15:6]	GPIO0[15:0]	GPIO0[15:0]	GPIO0[4:0]; GPIO0[15:6]
GPIO 端口 1	GPIO1[15:0]	GPIO1[15:0]	GPIO1[15:0]	GPIO1[15:0]	GPIO1[15:0]	GPIO1[15:0]
GPIO 端口 2	GPIO2[15:0]	GPIO2[15:0]	-	GPIO2[15:0]	GPIO2[15:0]	-
GPIO 端口 3	GPIO3[15:0]	GPIO3[15:0]	GPIO3[1:0]; GPIO3[5:3]; GPIO3[7]	GPIO3[15:0]	GPIO3[15:0]	GPIO3[1:0]; GPIO3[5:3]; GPIO3[7]
GPIO 端口 4	GPIO4[15:0]	GPIO4[15:0]	-	GPIO4[15:0]	GPIO4[11]	-
GPIO 端口 5	GPIO5[26:0]	GPIO5[26:0]	GPIO5[11:0]	GPIO5[25:0]	GPIO5[16:0]; GPIO5[18]	GPIO5[11:0]
GPIO 端口 6	GPIO6[30:0]	GPIO6[30:25]	-	GPIO6[5:0]; GPIO6[30:20]	-	-
GPIO 端口 7	GPIO7[25:0]	GPIO7[4:0]	-	GPIO7[10:0]; GPIO7[21:17]; GPIO7[25:23]	-	-

17.2 基本配置

各个 GPIO 功能块共享一个通用的时钟和复位连接，配置如下：

- 参见表 171 以了解时钟与电源控制。
- GPIO 通过 GPIO_RST 复位 （复位 #28）。
- 所有 GPIO 引脚默认设为输入。
- 对于引脚中断，从SCU的所有GPIO端口引脚中选择多达8个外部中断引脚（参见表135和表 136）。引脚中断必须在 NVIC 中使能 （参见表 21）。
- GPIO 分组中断必须在 NVIC 中使能 （参见表 21）。
- GPIO 端口寄存器可由 GPDMA 作为存储器到存储器传输来访问。

表 171. GPIO 时钟和电源控制

	基础时钟	分支时钟	工作频率
GPIO、GPIO 引脚中断、GPIO 组 0 中断、GPIO 组 1 中断	BASE_M4_CLK	CLK_M4_GPIO	最大 204 MHz

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

17.3 功能要点

17.3.1 GPIO 引脚中断功能要点

- 可从所有 GPIO 引脚中选择多达 8 个引脚，作为边沿或电平触发中断请求。每个请求在 NVIC 中创建一个单独的中断。
- 边沿敏感中断引脚可以在上升沿和 / 或下降沿发生中断。
- 电平触发中断引脚可以是高电平或低电平有效。

17.3.2 GPIO 分组中断功能要点

- 可以使能任意数量的 GPIO 引脚的输入，以构成一个共用的分组中断。
- 为分组中断使能的每个输入的极性可以配置为高电平或低电平。
- 使能中断可以通过 OR 或 AND 操作进行逻辑组合。
- 支持两种分组中断，体现两种不同的中断模式。
- GPIO 分组中断可以将器件从睡眠、深度睡眠或掉电模式下唤醒。

17.3.3 GPIO 端口功能要点

- GPIO 引脚可以通过软件配置为输入或输出。
- 所有 GPIO 引脚都默认设置为输入，复位时禁用中断。
- 引脚寄存器允许单独感测和设置各引脚。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

17.4 简介

GPIO 引脚可用于通过多种方式将引脚设置为输入或输出，并将输入用作电平和边沿敏感中断的组合。

17.4.1 GPIO 引脚中断

在所有可用的 GPIO 引脚中，可在系统控制模块中选择多达八个引脚作为外部中断引脚（参见 <tbd>）。外部中断引脚连接至 NVIC 中 8 个独立的中断，并根据上升 / 下降沿或引脚输入电平创建。

17.4.2 GPIO 分组中断

对于连接至两个 GPIO 分组中断模块（组 0 和组 1）的每个端口 / 引脚，GPIO 分组中断寄存器决定使能哪些引脚用于产生中断以及每个此类引脚的活动极性。

此外，GPIO 分组中断寄存器还会选择中断输出是由电平触发还是边沿启动，以及是基于所有使能输入的 OR 还是 AND 操作。

所选的输入引脚上检测到指定模式后，GPIO 分组中断模块将产生一个中断。如果器件处于掉电模式，则将先异步唤醒器件，然后才产生中断请求。向控制寄存器的中断状态位写入 1，可以清除中断请求线。

17.4.3 GPIO 端口

GPIO 端口寄存器可用于将每个 GPIO 引脚配置为输入或输出；引脚配置为输入时，读取每个引脚的状态；或引脚配置为输出时，设置每个引脚的状态。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

17.5 寄存器描述

GPIO 由以下模块组成：

- GPIO引脚中断模块位于地址0x4008 7000。该模块的寄存器使能表135或表136中选择的多达 8 个引脚中断，并为每个所选的引脚中断配置电平和边沿敏感性。GPIO 中断寄存器列示于 <tbd> 表 176 到表 185。
- GPIO组0中断模块位于地址0x4008 8000。该模块的寄存器可以配置端口0和1上的任何引脚，以形成共用中断。 GPIO 组 0 寄存器列示于表 173 和第 17.5.2 节。
- GPIO组1中断模块位于地址0x4008 9000。该模块的寄存器可以配置端口0和1上的任何引脚，以形成共用中断。 GPIO 组 1 寄存器列示于表 174 和第 17.5.2 节。
- GPIO 端口模块位于地址 0x400F 4000。该模块的寄存器可以读写端口引脚，并将端口引脚配置为输入或输出。 GPIO 端口寄存器列示于表 175 和第 17.5.3 节。

注：在所有 GPIO 寄存器中，未显示的位为**保留位**。

表 172. 寄存器简介：GPIO 引脚中断 （基址：0x4008 7000）

名称	访问类型	地址偏移	描述	复位值	参考
ISEL	R/W	0x000	引脚中断模式寄存器	0	表 176
IENR	R/W	0x004	引脚中断电平 （或上升沿）中断启用寄存器	0	表 177
SIENR	WO	0x008	引脚中断电平 （或上升沿）中断设置为启用寄存器	不适用	表 178
CIENR	WO	0x00C	引脚中断电平 （或上升沿）中断设置为禁用寄存器	不适用	表 179
IENF	R/W	0x010	引脚中断有效电平选择 （或下降沿中断启用）	0	表 180
SIENF	WO	0x014	引脚中断选择高电平有效（或下降沿中断设置为启用）寄存器	不适用	表 181
CIENF	WO	0x018	引脚中断选择低电平有效（或下降沿中断设置为禁用）寄存器	不适用	表 182
RISE	R/W	0x01C	引脚中断上升沿寄存器	0	表 183
FALL	R/W	0x020	引脚中断下降沿寄存器	0	表 184
IST	R/W	0x024	引脚中断状态寄存器	0	表 185

表 173. 寄存器简介：GPIO 组 0 中断 （基址 0x4008 8000）

名称	访问类型	地址偏移	描述	复位值	参考
CTRL	R/W	0x000	GPIO 分组中断控制寄存器	0	表 186
PORT_POL0	R/W	0x020	GPIO 分组中断端口 0 极性寄存器	0xFFFF FFFF	表 187
PORT_POL1	R/W	0x024	GPIO 分组中断端口 1 极性寄存器	0xFFFF FFFF	表 187
PORT_POL2	R/W	0x028	GPIO 分组中断端口 2 极性寄存器	0xFFFF FFFF	表 187
PORT_POL3	R/W	0x02C	GPIO 分组中断端口 3 极性寄存器	0xFFFF FFFF	表 187
PORT_POL4	R/W	0x030	GPIO 分组中断端口 4 极性寄存器	0xFFFF FFFF	表 187
PORT_POL5	R/W	0x034	GPIO 分组中断端口 5 极性寄存器	0xFFFF FFFF	表 187
PORT_POL6	R/W	0x038	GPIO 分组中断端口 6 极性寄存器	0xFFFF FFFF	表 187
PORT_POL7	R/W	0x03C	GPIO 分组中断端口 7 极性寄存器	0xFFFF FFFF	表 187
PORT_ENA0	R/W	0x040	GPIO 分组中断端口 0 使能寄存器	0	表 188
PORT_ENA1	R/W	0x044	GPIO 分组中断端口 1 使能寄存器	0	表 188

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 173. 寄存器简介：GPIO 组 0 中断 （基址 0x4008 8000） *（续）*

名称	访问类型	地址偏移	描述	复位值	参考
PORT_ENA2	R/W	0x048	GPIO 分组中断端口 2 使能寄存器	0	表 188
PORT_ENA3	R/W	0x04C	GPIO 分组中断端口 3 使能寄存器	0	表 188
PORT_ENA4	R/W	0x050	GPIO 分组中断端口 4 使能寄存器	0	表 188
PORT_ENA5	R/W	0x054	GPIO 分组中断端口 5 使能寄存器	0	表 188
PORT_ENA6	R/W	0x058	GPIO 分组中断端口 5 使能寄存器	0	表 188
PORT_ENA7	R/W	0x05C	GPIO 分组中断端口 5 使能寄存器	0	表 188

表 174. 寄存器简介：GPIO 组 1 中断 （基址 0x4008 9000）

名称	访问类型	地址偏移	描述	复位值	参考
CTRL	R/W	0x000	GPIO 分组中断控制寄存器	0	表 186
PORT_POL0	R/W	0x020	GPIO 分组中断端口 0 极性寄存器	0xFFFF FFFF	表 187
PORT_POL1	R/W	0x024	GPIO 分组中断端口 1 极性寄存器	0xFFFF FFFF	表 187
PORT_POL2	R/W	0x028	GPIO 分组中断端口 2 极性寄存器	0xFFFF FFFF	表 187
PORT_POL3	R/W	0x02C	GPIO 分组中断端口 3 极性寄存器	0xFFFF FFFF	表 187
PORT_POL4	R/W	0x030	GPIO 分组中断端口 4 极性寄存器	0xFFFF FFFF	表 187
PORT_POL5	R/W	0x034	GPIO 分组中断端口 5 极性寄存器	0xFFFF FFFF	表 187
PORT_POL6	R/W	0x038	GPIO 分组中断端口 6 极性寄存器	0xFFFF FFFF	表 187
PORT_POL7	R/W	0x03C	GPIO 分组中断端口 7 极性寄存器	0xFFFF FFFF	表 187
PORT_ENA0	R/W	0x040	GPIO 分组中断端口 0 使能寄存器	0	表 188
PORT_ENA1	R/W	0x044	GPIO 分组中断端口 1 使能寄存器	0	表 188
PORT_ENA2	R/W	0x048	GPIO 分组中断端口 2 使能寄存器	0	表 188
PORT_ENA3	R/W	0x04C	GPIO 分组中断端口 3 使能寄存器	0	表 188
PORT_ENA4	R/W	0x050	GPIO 分组中断端口 4 使能寄存器	0	表 188
PORT_ENA5	R/W	0x054	GPIO 分组中断端口 5 使能寄存器	0	表 188
PORT_ENA6	R/W	0x058	GPIO 分组中断端口 5 使能寄存器	0	表 188
PORT_ENA7	R/W	0x05C	GPIO 分组中断端口 5 使能寄存器	0	表 188

GPIO 端口地址可以字节、半字或字进行读写。

表 175. 寄存器简介：GPIO 端口 （基址 0x400F 4000）
X 表示每个端口的最高引脚编号，取决于封装尺寸 （参见表 170）。

名称	访问类型	地址偏移	描述	复位值	宽度	参考
B0 至 B31	R/W	0x0000 至 x001F	字节引脚寄存器端口 0；引脚 PIO0_0 至 PIO0_31	ext[1]	字节 （8 位）	表 189
B32 至 Bx	R/W	0x0020 至 0x003F	字节引脚寄存器端口 1	ext[1]	字节 （8 位）	表 189
B64 至 Bx	R/W	0x0040 至 0x005F	字节引脚寄存器端口 2	ext[1]	字节 （8 位）	
B96 至 Bx	R/W	0x0060 至 0x007F	字节引脚寄存器端口 3	ext[1]	字节 （8 位）	表 189
B128 至 Bx	R/W	0x0080 至 0x009F	字节引脚寄存器端口 4	ext[1]	字节 （8 位）	表 189
B160 至 Bx	R/W	0x00A0 至 0x00BF	字节引脚寄存器端口 5	ext[1]	字节 （8 位）	表 189
B192 至 Bx	R/W	0x00C0 to0x00DF	字节引脚寄存器端口 6	ext[1]	字节 （8 位）	表 189
B224 至 Bx	R/W	0x00E0 至 0x00FC	字节引脚寄存器端口 7	ext[1]	字节 （8 位）	表 189
W0 至 Wx	R/W	0x1000 至 0x107C	字引脚寄存器端口 0	ext[1]	字 （32 位）	表 190
W32 至 Wx	R/W	0x1080 至 0x10FC	字引脚寄存器端口 1	ext[1]	字 （32 位）	表 190

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 175. 寄存器简介：GPIO 端口 （基址 0x400F 4000） *（续）*

X 表示每个端口的最高引脚编号，取决于封装尺寸 （参见表 170）。

名称	访问类型	地址偏移	描述	复位值	宽度	参考
W64 至 Wx	R/W	0x1100 至 0x11FC	字引脚寄存器端口 2	ext[1]	字 （32 位）	表 190
W96 至 Wx	R/W	0x1180 至 0x11FC	字引脚寄存器端口 3	ext[1]	字 （32 位）	表 190
W128 至 Wx	R/W	0x1200 至 0x12FC	字引脚寄存器端口 4	ext[1]	字 （32 位）	表 190
W160 至 Wx	R/W	0x1280 至 0x12FC	字引脚寄存器端口 5	ext[1]	字 （32 位）	表 190
W192 至 Wx	R/W	0x1300 至 0x137C	字引脚寄存器端口 6	ext[1]	字 （32 位）	表 190
W224 至 Wx	R/W	0x1380 至 0x13FC	字引脚寄存器端口 7	ext[1]	字 （32 位）	表 190
DIR0	R/W	0x2000	方向寄存器端口 0	0	字 （32 位）	表 191
DIR1	R/W	0x2004	方向寄存器端口 1	0	字 （32 位）	表 191
DIR2	R/W	0x2008	方向寄存器端口 2	0	字 （32 位）	表 191
DIR3	R/W	0x200C	方向寄存器端口 3	0	字 （32 位）	表 191
DIR4	R/W	0x2010	方向寄存器端口 4	0	字 （32 位）	表 191
DIR5	R/W	0x2014	方向寄存器端口 5	0	字 （32 位）	表 191
DIR6	R/W	0x2018	方向寄存器端口 6	0	字 （32 位）	表 191
DIR7	R/W	0x201C	方向寄存器端口 7	0	字 （32 位）	表 191
MASK0	R/W	0x2080	掩码寄存器端口 0	0	字 （32 位）	表 192
MASK1	R/W	0x2084	掩码寄存器端口 1	0	字 （32 位）	表 192
MASK2	R/W	0x2088	掩码寄存器端口 2	0	字 （32 位）	表 192
MASK3	R/W	0x208C	掩码寄存器端口 3	0	字 （32 位）	表 192
MASK4	R/W	0x2090	掩码寄存器端口 4	0	字 （32 位）	表 192
MASK5	R/W	0x2094	掩码寄存器端口 5	0	字 （32 位）	表 192
MASK6	R/W	0x2098	掩码寄存器端口 6	0	字 （32 位）	表 192
MASK7	R/W	0x209C	掩码寄存器端口 7	0	字 （32 位）	表 192
PIN0	R/W	0x2100	端口引脚寄存器端口 0	ext[1]	字 （32 位）	表 193
PIN1	R/W	0x2104	端口引脚寄存器端口 1	ext[1]	字 （32 位）	表 193
PIN2	R/W	0x2108	端口引脚寄存器端口 2	ext[1]	字 （32 位）	表 193
PIN3	R/W	0x210C	端口引脚寄存器端口 3	ext[1]	字 （32 位）	表 193
PIN4	R/W	0x2110	端口引脚寄存器端口 4	ext[1]	字 （32 位）	表 193
PIN5	R/W	0x2114	端口引脚寄存器端口 5	ext[1]	字 （32 位）	表 193
PIN6	R/W	0x2118	端口引脚寄存器端口 6	ext[1]	字 （32 位）	表 193
PIN7	R/W	0x211C	端口引脚寄存器端口 7	ext[1]	字 （32 位）	表 193
MPIN0	R/W	0x2180	掩码后端口寄存器端口 0	ext[1]	字 （32 位）	表 194
MPIN1	R/W	0x2184	掩码后端口寄存器端口 1	ext[1]	字 （32 位）	表 194
MPIN2	R/W	0x2188	掩码后端口寄存器端口 2	ext[1]	字 （32 位）	表 194
MPIN3	R/W	0x218C	掩码后端口寄存器端口 3	ext[1]	字 （32 位）	表 194
MPIN4	R/W	0x2190	掩码后端口寄存器端口 4	ext[1]	字 （32 位）	表 194
MPIN5	R/W	0x2194	掩码后端口寄存器端口 5	ext[1]	字 （32 位）	表 194
MPIN6	R/W	0x2198	掩码后端口寄存器端口 6	ext[1]	字 （32 位）	表 194
MPIN7	R/W	0x219C	掩码后端口寄存器端口 7	ext[1]	字 （32 位）	表 194
SET0	R/W	0x2200	写入：端口 0 的设置寄存器 读：端口 0 的输出位	0	字 （32 位）	表 195

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 175. 寄存器简介：GPIO 端口 （基址 0x400F 4000） *（续）*

X 表示每个端口的最高引脚编号，取决于封装尺寸 （参见表 170）。

名称	访问类型	地址偏移	描述	复位值	宽度	参考
SET1	R/W	0x2204	写入：端口 1 的设置寄存器 读：端口 1 的输出位	0	字 （32 位）	表 195
SET2	R/W	0x2208	写入：端口 2 的设置寄存器 读：端口 2 的输出位	0	字 （32 位）	表 195
SET3	R/W	0x220C	写入：端口 3 的设置寄存器 读：端口 3 的输出位	0	字 （32 位）	表 195
SET4	R/W	0x2210	写入：端口 4 的设置寄存器 读：端口 4 的输出位	0	字 （32 位）	表 195
SET5	R/W	0x2214	写入：端口 5 的设置寄存器 读：端口 5 的输出位	0	字 （32 位）	表 195
SET6	R/W	0x2218	写入：端口 6 的设置寄存器 读：端口 6 的输出位	0	字 （32 位）	表 195
SET7	R/W	0x221C	写入：端口 7 的设置寄存器 读：端口 7 的输出位	0	字 （32 位）	表 195
CLR0	WO	0x2280	清除端口 0	不适用	字 （32 位）	表 196
CLR1	WO	0x2284	清除端口 1	不适用	字 （32 位）	表 196
CLR2	WO	0x2288	清除端口 2	不适用	字 （32 位）	表 196
CLR3	WO	0x228C	清除端口 3	不适用	字 （32 位）	表 196
CLR4	WO	0x2290	清除端口 4	不适用	字 （32 位）	表 196
CLR5	WO	0x2294	清除端口 5	不适用	字 （32 位）	表 196
CLR6	WO	0x2298	清除端口 6	不适用	字 （32 位）	表 196
CLR7	WO	0x229C	清除端口 7	不适用	字 （32 位）	表 196
NOT0	WO	0x2300	切换端口 0	不适用	字 （32 位）	表 197
NOT1	WO	0x2304	切换端口 1	不适用	字 （32 位）	表 197
NOT2	WO	0x2308	切换端口 2	不适用	字 （32 位）	表 197
NOT3	WO	0x230C	切换端口 3	不适用	字 （32 位）	表 197
NOT4	WO	0x2310	切换端口 4	不适用	字 （32 位）	表 197
NOT5	WO	0x2314	切换端口 5	不适用	字 （32 位）	表 197
NOT6	WO	0x2318	切换端口 6	不适用	字 （32 位）	表 197
NOT7	WO	0x231C	切换端口 7	不适用	字 （32 位）	表 197

[1] 此表和后续表中的 ext 是指复位后的数据读取取决于引脚的状态，这进一步又可能取决于外部源。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

17.5.1 GPIO 引脚中断寄存器描述

17.5.1.1 引脚中断模式寄存器

对于表 135 和表 136 中选择的八个引脚中断，ISEL 寄存器的某个位决定了这些中断是边沿还是电平触发。

表 176. 引脚中断模式寄存器 （ISEL，地址 0x4008 7000）位描述

位	符号	描述	复位值	访问类型
7:0	PMODE	选择每个引脚中断的中断模式。位 n 配置 PINTSELn 中所选的引脚中断。 0 = 边沿敏感 1 = 电平触发	0	R/W
31:8	-	保留。	-	-

17.5.1.2 引脚中断电平 （上升沿中断）使能寄存器

对于 PINTSEL 寄存器中所选的八个引脚中断 （参见表 135 和表 136）， IENR 寄存器的某个位会根据 ISEL 寄存器中配置的引脚中断模式使能中断：

- 如果引脚中断模式是边沿敏感 (PMODE = 0)，则使能上升沿中断。
- 如果引脚中断模式是电平触发 (PMODE = 1)，则使能电平中断。PINTEN_F 寄存器配置该中断的有效电平 （高或低电平）。

表 177. 引脚中断电平 （上升沿中断使能）寄存器 （IENR，地址 0x4008 7004）位描述

位	符号	描述	复位值	访问类型
7:0	ENRL	使能每个引脚中断的上升沿或电平中断。位n配置PINTSELn中所选的引脚中断。 0 = 禁用上升沿或电平中断。 1 = 使能上升沿或电平中断。	0	R/W
31:8	-	保留。	-	-

17.5.1.3 引脚中断电平 （上升沿中断）设置寄存器

对于 PINTSEL 寄存器中所选的八个引脚中断 （参见表 135 和表 136）， SIENR 寄存器的某个位会根据 PINTMODE 寄存器中配置的引脚中断模式设置 IENR 寄存器中的对应位：

- 如果引脚中断模式是边沿敏感 (PMODE = 0)，则设置为上升沿中断。
- 如果引脚中断模式是电平触发 (PMODE = 1)，则设置为电平中断。

表 178. 引脚中断电平 （上升沿中断）设置寄存器 （SIENR，地址 0x4008 7008）位描述

位	符号	描述	复位值	访问类型
7:0	SETENRL	向该地址写 1 会置位 PINTEN_R 中的位，从而使能中断。位 n 设置 PINTEN_R 寄存器中的位 n。 0 = 无操作。 1 = 使能上升沿或电平中断。	不适用	WO
31:8	-	保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

17.5.1.4 引脚中断电平 （上升沿中断）清除寄存器

对于 PINTSEL 寄存器中所选的八个引脚中断 （参见表 135 和表 136）， CIENR 寄存器的某个位会根据 ISEL 寄存器中配置的引脚中断模式清除 IENR 寄存器中的对应位：

- 如果引脚中断模式是边沿敏感 (PMODE = 0)，则清除上升沿中断。
- 如果引脚中断模式是电平触发 (PMODE = 1)，则清除电平中断。

表 179. 引脚中断电平 （上升沿中断）清除寄存器 （PCIENR，地址 0x4008 700C）位描述

位	符号	描述	复位值	访问类型
7:0	CENRL	向该地址写 1 会清除 IENR 中的位，从而禁用中断。位 n 清除 IENR 寄存器中的位 n。 0 = 无操作。 1 = 禁用上升沿或电平中断。	不适用	WO
31:8	-	保留。	-	-

17.5.1.5 引脚中断有效电平 （下降沿中断使能）寄存器

对于 PINTSEL 寄存器中所选的八个引脚中断 （参见表 135 和表 136）， PINTSEN_F 寄存器的某个位会根据 ISEL 寄存器中配置的引脚中断模式使能下降沿中断或配置电平触发性：

- 如果引脚中断模式是边沿敏感 (PMODE = 0)，则使能下降沿中断。
- 如果引脚中断模式是电平触发 (PMODE = 1)，则配置电平中断的有效电平 （高电平或低电平）。

表 180. 引脚中断有效电平 （下降沿中断使能）寄存器 （IENF，地址 0x4008 7010）位描述

位	符号	描述	复位值	访问类型
7:0	ENAF	使能每个引脚中断的下降沿中断或配置有效电平中断。位 n 配置 PINTSELn 中所选的引脚中断。 0 = 禁用下降沿中断或将有效中断电平设置为低电平。 1 = 使能下降沿中断或将有效中断电平设置为高电平。	0	R/W
31:8	-	保留。	-	-

17.5.1.6 引脚中断有效电平 （下降沿中断）设置寄存器

对于 PINTSEL 寄存器中所选的八个引脚中断（参见表 135 和表 136），SIENF 寄存器的某个位会根据 ISEL 寄存器中配置的引脚中断模式设置 IENF 寄存器中的对应位：

- 如果引脚中断模式是边沿敏感 (PMODE = 0)，则设置为下降沿中断。
- 如果引脚中断模式是电平触发 (PMODE = 1)，则选择高电平有效中断。

表 181. 引脚中断有效电平 （下降沿中断）设置寄存器 （SIENF，地址 0x4008 7014）位描述

位	符号	描述	复位值	访问类型
7:0	SETENAF	向该地址写 1 会置位 IENF 中的位，从而使能中断。位 n 设置 IENF 寄存器中的位 n。 0 = 无操作。 1 = 选择高电平有效中断或使能下降沿中断。	不适用	WO
31:8	-	保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

17.5.1.7 引脚中断有效电平 （下降沿中断）清除寄存器

对于 PINTSEL 寄存器中所选的八个引脚中断（参见表 135 和表 136），CIENF 寄存器的某个位会根据 ISEL 寄存器中配置的引脚中断模式设置 IENF 寄存器中的对应位：

- 如果引脚中断模式是边沿敏感 (PMODE = 0)，则清除下降沿中断。
- 如果引脚中断模式是电平触发 (PMODE = 1)，则选择低电平有效中断。

表 182. 引脚中断有效电平 （下降沿中断）清除寄存器 （CIENF，地址 0x4008 7018）位描述

位	符号	描述	复位值	访问类型
7:0	CENAF	向该地址写 1 会清除 IENF 中的位，从而禁用中断。位 n 设置 IENF 寄存器中的位 n。 0 = 无操作。 1 = 选择低电平有效中断或禁用下降沿中断。	不适用	WO
31:8	-	保留。	-	-

17.5.1.8 引脚中断上升沿寄存器

该寄存器的 1 表示 PINTSEL 寄存器中所选的引脚中断 （参见表 135 和表 136）上已检测到上升沿。向该寄存器写 1，清除上升沿检测。对于上升沿中断使能的引脚，该寄存器中的 1 产生一个中断请求。为 PINTSEL 寄存器中所选的所有引脚检测所有边沿，无论这些引脚是否是中断使能。

表 183. 引脚中断上升沿寄存器 （RISE，地址 0x4008 701C）位描述

位	符号	描述	复位值	访问类型
7:0	RDET	上升沿检测。位 n 检测 PINTSELn 中所选引脚的上升沿。 读 0：复位或上一次向该位写 1 后，未在该引脚上检测到上升沿。 写 0：无操作。 读 1：自复位或上一次向该位写 1 起，已检测到上升沿。 写 1：清除该引脚已检测到上升沿的标志。	0	R/W
31:8	-	保留。	-	-

17.5.1.9 引脚中断下降沿寄存器

该寄存器的 1 表示 PINTSEL 寄存器中所选的引脚中断 （参见表 135 和表 136）上已检测到下降沿。向该寄存器写 1，清除下降沿检测。对于下降沿中断使能的引脚，该寄存器中的 1 产生一个中断请求。为 PINTSEL 寄存器中所选的所有引脚检测所有边沿，无论这些引脚是否是中断使能。

表 184. 引脚中断下降沿寄存器 （FALL，地址 0x4008 7020）位描述

位	符号	描述	复位值	访问类型
7:0	FDET	下降沿检测。位 n 检测 PINTSELn 中所选引脚的下降沿。 读 0：复位或上一次向该位写 1 后，未在该引脚上检测到下降沿。 写 0：无操作。 读 1：自复位或上一次向该位写 1 起，已检测到下降沿。 写 1：清除该引脚已检测到下降沿的标志。	0	R/W
31:8	-	保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

17.5.1.10 引脚中断状态寄存器

引脚中断当前正在请求中断时，读该寄存器会返回 1。对于在中断选择寄存器中确定为边沿敏感的引脚，向该寄存器中写 1 会清除该引脚上已检测到的上升或下降沿标志。对于电平触发引脚，写 1 会反转有效电平寄存器中的对应位，从而切换引脚的有效电平。

表 185. 引脚中断状态寄存器 （IST，地址 0x4008 7024）位描述

位	符号	描述	复位值	访问类型
7:0	PSTAT	引脚中断状态。位 n 返回状态、清除边沿中断，或反转 PINTSELn 所选引脚的有效电平。 读 0：该中断引脚无正在请求的中断。 写 0：无操作。 读 1：该中断引脚有正在请求的中断。 写 1 （边沿敏感）：清除该引脚上已检测到的上升或下降沿标志。 写 1 （电平触发）：切换 （PINTENT_F 寄存器中）该引脚的有效电平。	0	R/W
31:8	-	保留。	-	-

17.5.2 GPIO 组 0/ 组 1 中断寄存器描述

17.5.2.1 分组中断控制寄存器

表 186. GPIO 分组中断控制寄存器 （CTRL，地址 0x4008 8000 (GROUP0 INT) 和 0x4008 9000 (GROUP1 INT)）位描述

位	符号	值	描述	复位值
0	INT		分组中断状态。写 1 则清除该位。写入 0 无效。	0
		0	无中断请求挂起。	
		1	中断请求有效。	
1	COMB		组合分组中断的使能输入	0
		0	OR 功能：当任意一个使能输入有效时 （基于可编程的极性），生成一个分组中断。	
		1	AND 功能：当所有使能位有效时 （基于可编程的极性），生成一个中断。	
2	TRIG		分组中断启动	0
		0	边沿启动	
		1	电平触发	
31:3	-	-	保留	0

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

17.5.2.2 GPIO 分组中断端口极性寄存器

分组中断端口极性寄存器决定每个使能引脚的极性如何用于产生分组中断。每个端口 n (n = 0 至 7) 与其端口极性寄存器相关，且所有寄存器的值一起决定分组中断。

表 187. GPIO 分组中断端口极性寄存器（PORT_POL，地址 0x4008 8020 (PORT_POL0) 至 0x4008 803C (PORT_POL7) （组 0 INT）和 0x4008 9020 (PORT_POL0) 至 0x4008 903C (PORT_POL7) （组 1 INT））位描述

位	符号	描述	复位值	访问类型
31:0	POL	配置分组中断的端口 n 引脚的引脚极性。位 m 对应端口 n 的引脚 GPIOn[m]。 0 = 引脚为有效低电平。如果该引脚为低电平，则该引脚用于产生分组中断。 1 = 引脚为有效高电平。如果该引脚为高电平，则该引脚用于产生分组中断。	1	-

17.5.2.3 GPIO 分组中断端口使能寄存器

分组中断端口使能寄存器使能构成分组中断的引脚。每个端口 n (n = 0 至 7) 都关联一个它自己的端口启用寄存器，且所有寄存器的值一起决定构成分组中断的引脚。

表 188. GPIO 分组中断端口 n 使能寄存器 （PORT_ENA，地址 0x4008 8040 (PORT_ENA0) 至 0x4008 805C (PORT_ENA7)（组 0 INT）和 0x4008 9040 (PORT_ENA0) 至 0x4008 905C (PORT_ENA7) （组 1 INT））位描述

位	符号	描述	复位值	访问类型
31:0	ENA	使能分组中断的端口n引脚。位m对应端口n的引脚GPIOPn[m]。 0 = 端口 n 引脚禁用，不构成分组中断。 1 = 端口 n 引脚使能，构成分组中断。	0	-

17.5.3 GPIO 端口寄存器描述

17.5.3.1 GPIO 端口字节引脚寄存器

每个 GPIO 引脚 GPIOn[m] 在该地址范围内都拥有一个字节寄存器。GPIO 端口 0 的字节引脚寄存器对应寄存器 B0 至 B31，GPIO 端口 1 的字节寄存器对应寄存器 B32 至 B63 等。为未使用的 GPIO 端口引脚保留字节地址 （参见表 170）。

通常，软件通过读取和写入字节来访问各个引脚，也可以读取或写入半字来感测或设置 2 个引脚的状态，读取或写入字来感测或设置 4 个引脚的状态。

表 189. GPIO 端口字节引脚寄存器 （B，地址 0x400F 4000 (B0) 至 0x400F 00FC (B255)）位描述

位	符号	描述	复位值	访问类型
0	PBYTE	读：引脚 GPIOn[m] 的状态，无论其方向、屏蔽或可选功能如何。配置为模拟 I/O 的引脚总是读为 0。 写入：加载引脚的输出位。	ext	R/W
7:1		保留 （读取时为 0，写入时忽略）	0	-

17.5.3.2 GPIO 端口字引脚寄存器

每个 GPIO 引脚 GPIOn[m] 在该地址范围内都拥有一个字寄存器。GPIO 端口 0 的字引脚寄存器对应寄存器 W0 至 W31， GPIO 端口 1 的字寄存器对应寄存器 W32 至 W63 等。为未使用的 GPIO 端口引脚保留字地址 （参见表 170）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

如果引脚为低电平，则该范围内的任何字节、半字或字读取为全 0；如果引脚为高电平，则为全 1，引脚的方向、屏蔽或可选功能都不会影响结果，除非引脚配置为模拟 I/O，则会始终读为 0。如果写入的值均为 0，则任何写入都将清除引脚的输出位，否则会设置引脚的输出位。

表 190. GPIO 端口字引脚寄存器 （W，地址 0x400F 5000 (W0) 至 0x400F 13FC (W255)）位描述

位	符号	描述	复位值	访问类型
31:0	PWORD	读 0：引脚 GPIOn[m] 为低电平。 写 0：清除输出位。 读 0xFFFF FFFF: 引脚为高电平。 写 0x0000 0001 至 0xFFFF FFFF 之间的任何值：置位输出位。 **注：**仅可读取 0 或 0xFFFF FFFF。写 0 以外的任何值时，将置位输出位。	ext	R/W

17.5.3.3 GPIO 端口方向寄存器

每个 GPIO 端口 n（n = 0 至 7）都有一个方向寄存器，用于将端口引脚配置为输入或输出。

表 191. GPIO 端口方向寄存器（DIR，地址 0x400F 6000 (DIR0) 至 0x400F 601C (DIR7)）位描述

位	符号	描述	复位值	访问类型
31:0	DIR	选择引脚 GPIOn[m] 的引脚方向 （位 0 = GPIOn[0]，位 1 = GPIOn[1]， ...，位 31 = GPIOn[31]）。 0 = 输入。 1 = 输出。	0	R/W

17.5.3.4 GPIO 端口掩码寄存器

每个 GPIO 端口都有一个掩码寄存器。该掩码寄存器会影响读写 MPORT 寄存器。将这些寄存器设为 0，可以使能读写；设为 1 则禁用写入以及让对应位置读为 0。

表 192. GPIO 端口掩码寄存器（MASK，地址 0x400F 6080 (MASK0) 至 0x400F 609C (MASK7)）位描述

位	符号	描述	复位值	访问类型
31:0	MASK	控制 MPORT 寄存器中哪些对应 GPIOn[m] 的位是有效的（位 0 = GPIOn[0]，位 1 = GPIOn[1]， ...，位 31 = GPIOn[31]）。 0 = 读 MPORT：引脚状态；写 MPORT：加载输出位。 1 = 读 MPORT：0；写 MPORT：输出位不受影响。	0	R/W

17.5.3.5 GPIO 端口引脚寄存器

每个 GPIO 端口都有一个端口引脚寄存器。读取这些寄存器会返回所读引脚的当前状态，引脚的方向、屏蔽或可选功能都不会影响结果，除非引脚配置为模拟 I/O 时才会始终读为 0。写这些寄存器会加载所写引脚的输出位，无论是否为 MASK 寄存器。

表 193. GPIO 端口引脚寄存器 （PIN，地址 0x400F 6100 (PIN0) 至 0x400F 611C (PIN7)）位描述

位	符号	描述	复位值	访问类型
31:0	PORT	读取引脚状态或加载输出位 （位 0 = GPIOn[0]，位 1 = GPIOn[1]， ...，位 31 = GPIOn[31]）。 0 = 读。引脚为低电平；写入：清除输出位。 1 = 读：引脚为高电平；写入：置位输出位。	ext	R/W

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

17.5.3.6 GPIO 屏蔽端口引脚寄存器

每个 GPIO 端口都有一个屏蔽端口引脚寄存器。这些寄存器与 PORT 寄存器类似，不同之处在于通过和相应 MASK 寄存器中的反向内容进行 AND 操作，可以屏蔽读取某些值；同时，写其中一个寄存器仅影响在相应 MASK 寄存器中由 0 使能的输出寄存器位。

表 194. GPIO 屏蔽端口引脚寄存器（MPIN，地址 0x400F 6180 (MPIN0) 至 0x400F 619C (MPIN7)）位描述

位	符号	描述	复位值	访问类型
31:0	MPORT	掩码后端口寄存器（位 0 = GPIOn[0]，位 1 = GPIOn[1]，...，位 31 = GPIOn[31]）。 0 = 读。引脚为低电平和 / 或 MASK 寄存器中的对应位为 1；写入：当 MASK 寄存器中的对应位为 0 时，清除输出位。 1 = 读：引脚为高电平和 MASK 寄存器中的对应位为 0；写入：当 MASK 寄存器中的对应位为 0 时，置位输出位。	ext	R/W

17.5.3.7 GPIO 端口设置寄存器

每个 GPIO 端口都有一个端口设置寄存器。向这些寄存器写 1，可以置位输出位，无论其是否为 MASK 寄存器。读这些寄存器会返回端口的输出位，无论其引脚方向如何。

表 195. GPIO 端口设置寄存器（SET，地址 0x400F 6200 (SET0) 至 0x400F 621C (SET7)）位描述

位	符号	描述	复位值	访问类型
31:0	设置	读取或设置输出位（位 0 = GPIOn[0]、位 1 = GPIOn[1]、...、位 31 = GPIOn[31]）。 0 = 读。输出位；写入：无操作。 1 = 读：输出位；写入：置位输出位。	0	R/W

17.5.3.8 GPIO 端口清除寄存器

每个 GPIO 端口都有一个输出清除寄存器。向这些只写寄存器写 1，可以清除输出位，无论是否为 MASK 寄存器。

表 196. GPIO 端口清除寄存器（CLR，地址 0x400F 6280 (CLR0) 至 0x400F 629C (CLR7)）位描述

位	符号	描述	复位值	访问类型
31:0	CLR	清除输出位（位 0 = GPIOn[0]、位 1 = GPIOn[1]、...、位 31 = GPIOn[31]）： 0 = 无操作。 1 = 清除输出位。	不适用	WO

17.5.3.9 GPIO 端口切换寄存器

每个 GPIO 端口都有一个输出切换寄存器。向这些只写寄存器写 1，可以切换 / 反转 / 取补输出位，无论是否为 MASK 寄存器。

表 197. GPIO 端口切换寄存器（NOT，地址 0x400F 6300 (NOT0) 至 0x400F 632C (NOT7)）位描述

位	符号	描述	复位值	访问类型
31:0	NOTP0	切换输出位（位 0 = GPIOn[0]、位 1 = GPIOn[1]、...、位 31 = GPIOn[31]）： 0 = 无操作。 1 = 切换输出位。	不适用	WO

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

17.6 功能描述

17.6.1 读取引脚状态

软件可以读取所有 GPIO 引脚的状态，除在 I/O 配置逻辑中选为模拟输入或输出的引脚外。要读取某个引脚的状态，并不一定要在 I/O 配置中将其选为 GPIO。共有 4 种方式可以读取引脚状态：

- 可以从字节引脚寄存器中读取单个引脚的状态，为 7 个高位 0。
- 可以从字引脚寄存器中读取单个引脚的状态，为 1 个字节、半字或整字的所有位。
- 可以从 PORT 寄存器中读取多个引脚的状态，为 1 个字节、半字或整字。
- 可以从屏蔽端口 (MPORT) 寄存器中读取某个端口的所选引脚子集的状态。在端口的掩码寄存器中置 1 的位所对应的引脚，它们在 MPORT 寄存器中总是读为 0。

17.6.2 GPIO 输出

每个 GPIO 引脚在 GPIO 模块中都有一个输出位。这些输出位是写操作到引脚的目标位。如需将引脚的输出位驱动到引脚，必须符合两个条件：

1. 在 I/O 配置模块中，必须将该引脚选为 GPIO 操作；
2. 该引脚必须选为输出，方法为将其端口的 DIR 寄存器设为 1。

如果其中一个或两个条件未能满足，写入到引脚无效。

共有 7 种方式可以更改 GPIO 输出位：

- 写入到字节引脚寄存器会从最低有效位加载输出位。
- 写入到字引脚寄存器会通过所有写入字节的 OR 操作加载输出位。（该功能遵循程序设计语言中多位值为真的定义。）
- 写入到端口的 PORT 寄存器会加载所有写入引脚的输出位。
- 写入到端口的 MPORT 寄存器会加载由该端口 MASK 寄存器对应位置的 0 确定的引脚输出位。
- 向端口的 SET 寄存器写 1 会设置输出位。
- 向端口的 CLR 寄存器写 1 会清除输出位。
- 向端口的 NOT 寄存器写 1 会切换 / 取补 / 反转输出位。

可以从 SET 寄存器中读取某个端口的输出位状态。读 17.6.1 中描述的任何寄存器会返回引脚状态，无论其方向或可选功能如何。

17.6.3 I/O 的掩码

端口的 MASK 寄存器确定哪些引脚应当在 MPORT 寄存器中可访问。MASK 中的位为 0 时，可以使能从 MPORT 中读取和写入的对应引脚。MASK 中的位为 1 时，引脚会强制读为 0，其输出位将不会受写入到 MPORT 的影响。当端口的 MASK 寄存器中的位全为 0 时，其 PORT 和 MPORT 寄存器将以完全一致的方式进行读写。

对于使用具有类似 GPIO 模块的前代恩智浦设备的用户，应注意其不可兼容性：在 LPC43xx 上，写入到 SET、CLR 和 NOT 寄存器不会受 MASK 寄存器的影响。前代设备上这些寄存器也是要做掩码处理的。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

在一些应用程序中，如果在中断中执行掩码的 I/O 操作，或者任务中有执行掩码的 I/O 操作，这些应用程序必须将使用掩码寄存器的代码作为受保护 / 受限制的区域。中断禁用或使用信号量都可以实现这一点。

保护使用 MASK 寄存器的代码块的更简单方式是：设置 MASK 寄存器前禁用中断，并在使用 MPORT 或 MASK 寄存器的最后一次操作结束后重新使能中断。

更有效的是，软件可以为 MASK 寄存器指定专用信号量；在设置 MASK 寄存器前，设置 / 捕获控制 MASK 寄存器专用的信号量；并在使用 MPORT 或 MASK 寄存器的最后一次操作结束后释放信号量。

17.6.4 GPIO 中断

提供两种独立的 GPIO 中断机制。引脚中断：多达 8 个 GPIO 引脚可以拥有独立的向量、边沿或电平触发中断。

分组中断：每个端口的任一引脚子集都可以构成一个通用中断。可以使能任何引脚和端口中断，将器件从深度睡眠模式或掉电模式下唤醒。

17.6.4.1 引脚中断

在该中断机制中，引脚中断选择寄存器 (PINTSEL0-7) 将多达 8 个引脚确认为中断源。所有其他引脚中断寄存器中均包含 8 个位，对应 PINTSEL0-7 寄存器调用的各个引脚。PINTMODE 寄存器确定每个中断引脚是边沿敏感还是电平触发。 PINTRISE 和 PINTFALL 寄存器检测每个中断引脚的边沿，其写操作可以清除 （和设置）边沿检测。 PINTST 寄存器指示每个中断引脚当前是否在请求中断，写 PINTST 可以清除中断。

其他引脚中断寄存器对边沿敏感和电平触发引脚起到不同的作用，如表 198 中的描述。

表 198. 边沿和电平触发引脚的引脚中断寄存器

名称	边沿敏感功能	电平触发功能
PINTEN_R	使能上升沿中断。	使能中断。
PINTSEN_R	写入使能上升沿中断。	写入使能中断。
PINTCEN_R	写入禁用上升沿中断。	写入禁用中断。
PINTEN_F	使能下降沿中断。	选择有效电平。
PINTSEN_F	写入使能下降沿中断。	写入选择高电平有效。
PINTCEN_F	写入禁用下降沿中断。	写入选择低电平有效。

17.6.4.2 分组中断

在该中断机制中，根据选择的每个端口的任一引脚子集，都可以为每个端口请求中断。将端口的使能寄存器设为 1，选出构成每个端口中断的引脚，同时还可在端口的极性寄存器中选择每个引脚的中断极性。每个引脚的电平与其极性位进行异或操作，结果与其使能位进行 AND 操作；然后这些结果再与该端口的所有引脚进行或操作，从而创建了该端口的原始中断请求。

两个分组中断的原始中断请求发送到 NVIC，经过编程会将其作为电平或边沿敏感（参见表 21）。

17.6.5 推荐用法

下表列出了 GPIO 端口寄存器的若干推荐用法：

- 对于复位或重新初始化后的初始设置，写 PORT 寄存器。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- 如需更改某个引脚的状态，写字节引脚或字引脚寄存器。
- 如需一次性更改多个引脚的状态，写 SET 和 / 或 CLR 寄存器。
- 如需在严格控制的环境（如软件状态机）中更改多个引脚的状态，考虑使用 NOT 寄存器。这比 SET 和 CLR 需要的写操作更少。
- 如需读取某个引脚的状态，读字节引脚或字引脚寄存器。
- 如需根据多个引脚作出决定，读取并屏蔽 PORT 寄存器。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

18.1 本章导读

SGPIO 可用于所有 LPC43xx 器件。 12 位 ADC 不可用于器件 LPC4350/30/20/10。

18.2 基本配置

SGPIO 的配置如下：

- 参见表 199 以了解时钟与电源控制。
- SGPIO 通过 SGPIO_RST 复位 （复位 # 57）。
- SGPIO 中断连接到 ARM Cortex-M4 NVIC 中的中断插槽 # 31。
- SGPIO输出SGPIO3和SGPIO12通过GIMA与事件记录器、定时器以及SCT连接（参见表 138）。 SGPIO3 和 SGPIO12 还支持 GIMA 的 128 分频信号 （SGPIO3_DIV 和 SGPIO12_DIV）。
- SGPIO 输出 10 和 12 可启动 12 位 ADC。
- SGPIO 输出 14 和 15 可启动 GPDMA 请求。

表 199. SGPIO 时钟和电源控制

	基本时钟	分支时钟	工作频率	备注
SGPIO 外设时钟 (SGPIO_CLOCK)	BASE_PERIPH_CLK	CLK_PERIPH_SGPIO	高达 204 MHz	该时钟与主时钟不同步，可自由选择该时钟来创建所需的 SGPIO 数据速率。

18.3 功能要点

- 每个 SGPIO 输入 / 输出位串都可用于执行串行到并行或并行到串行数据转换。
- 16个SGPIO输入/输出位串（每个都带32位FIFO）可利用移位时钟的每个周期将来自引脚的输入值或对引脚的输出值移位。
- 每个位串都支持双缓冲。
- FIFO 已满、转换时钟或模式匹配时生成中断。
- 可将位串级联以增加缓冲区大小。
- 每个位串有一个 32 位模式匹配滤波器。

18.4 简介

串行 GPIO (SGPIO) 提供标准 GPIO 功能，并具备加速串行数据流处理的增强功能。单一 SGPIO 输入 / 输出或双路、四路和字节通道数据输入 / 输出上的数据流使用所谓的位串进行处理。最多可支持16个位串，所有16个位串的基本功能集相同，某些位串还具有其他功能。

- 位串：
  - 每个位串支持 32 位 FIFO，在每个移位时钟从某个引脚移入或移出至某个引脚。
  - 某些位串控制引脚输出使能。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- 位串为双缓冲； POS 计数器达到 0x0 时，位串与影子寄存器进行交换。
- 可将位串级联以增加缓冲区大小。如果位串n与位串m级联，则软件应当确保POS[n]和 POS[m] 相等且同相运行。
- POS 计数器达到 0x0 时产生中断。
- 支持并行到多数据流串行转换，反之亦然。转换支持交错并联；每次最多转换 8 位（n= 1、 2、 4 或 8 分别对应串行、双串行、四串行和字节并行 IO）。

• 时钟：
- 以 SGPIO_CLOCK 运行的 12 位计数器创建移位时钟来捕获输入或创建输出值。请注意，输入频率应当小于 SGPIO_CLOCK 频率的一半。
- 外部输入可用作移位时钟。有效边沿可以为上升沿或下降沿（但不能同时为两者）。
- 外部输入可用作移位时钟门控。有效电平可以是低电平或高电平。
- 内部信号可用作移位时钟门控。
- 以移位时钟运行的 POS 计数器使能双缓冲。

• 输入：
- 使用移位时钟捕获输入。
- 输入可在输入电平 （低电平或高电平）或跳变 （上升、下降上升或下降）时产生中断。中断可被掩码。
- 可在模式匹配时产生中断。模式最多为 32 位长并且可被掩码 （匹配值 = “不关注”）。 4 个位串 （A、 I、 H 和 P）支持掩码功能。
- 4 个输入可用作其他位串的移位时钟来捕获其他输入。

• 输出：
- 创建输出：有效电平 （低电平或高电平）或三态。注意，如果位串用于创建速率为 f_{out} 的输出时钟，则位串时钟的速率至少应为 2x f_{out}。这样每个位串的数据速率就被限制为 SGPIO_CLOCK 速率的一半。
- 输出使能可由其他位串控制。如果是多通道输出，则可独立于 MSB 线控制 LSB 通道输出使能。
- 从移位时钟计数器创建输出时钟。
- 输出时钟极性可反转。

• 接口：
- 寄存器存储器映射支持 ARM “存储多个”和 “加载多个”指令的使用。控制相同特性的位串功能在地址连续的寄存器中映射。
- 位顺序针对 MSB 优先进行优化。需要 LSB 优先的接口应当使用软件指令 (RBIT) 来反转位顺序。

18.4.1 中断

• 下列事件中将产生中断：
- COUNTx 等于零、出现移位时钟时提出移位时钟中断 （参见表 222 至表 227）。该中断在每个移位位生成。
- 发生位串交换时 （即 POSx 等于零、出现捕获时钟时），提出捕获时钟中断 （参见表 228 至表 233）。
- 输入数据等于掩码的模式时提出模式匹配中断 （参见表 234 至表 239）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- 输入位与 DATA_CAPTURE_MODE 中设置的条件相等时提出输入位匹配中断（表 240 至表 244）。

- 每个中断可由一组寄存器采用以下某种方法进行控制：
  - 通过寄存器 CLR_EN/SET_EN 禁用或使能中断。
  - 通过寄存器 STATUS 读取中断。
  - 通过寄存器 ENABLE 读取中断掩码。
  - 通过寄存器 SET_STAT 设置中断。
  - 通过寄存器 CTR_STAT 清除中断。

18.5 引脚说明

表 200. SGPIO 引脚描述

引脚功能	方向	描述
SGPIO[15:0]	I/O	串行 GPIO 输入 / 输出

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

18.6 寄存器描述

表 201. 寄存器简介：SGPIO （基址 0x4010 1000）

名称	访问类型	地址偏移	描述	复位值	参考
OUT_MUXCFG0 至 OUT_MUX_CFG15	R/W	0x0000 至 0x003C	引脚多路复用器配置寄存器。	0	表 202
SGPIO_MUX_CFG0 至 SGPIO_MUX_CFG15	R/W	0x0040 至 0x007C	SGPIO 多路复用器配置寄存器。	0	表 205
SLICE_MUX_CFG0 至 SLICE_MUX_CFG15	R/W	0x0080 至 0x00BC	位串多路复用器配置寄存器。	0	表 207
REG0 至 REG15	R/W	0x00C0 至 0x00FC	位串数据寄存器。每当 COUNT0 达到 0x0 时，寄存器移位一次，并把从 DIN(n) 捕获的数据加载到位 31，以及使用 REG(0) 驱动 DOUT(n)	0	表 208
REG_SS0 至 REG_SS15	R/W	0x0100 至 0x013C	位串数据影子寄存器。每当 POS 达到 0x0 时，REG_SS 的内容与 REG 的内容进行交换	0	表 209
PRESET0 至 PRESET15	R/W	0x0140 至 0x17C	COUNT0 的重新加载值，COUNT0 达到 0x0 时加载	0	表 210
COUNT0 至 COUNT15	R/W	0x0180 至 0x1BC	下行计数器，每个时钟周期向下计数。	0	表 211
POS0 至 POS15	R/W	0x01C0 至 0x01FC	每当 COUNT0 达到 0x0 时，POS 向下计数。	0	表 212
MASK_A	R/W	0x0200	位串 A 匹配功能的掩码	0	表 213
MASK_H	R/W	0x0204	位串 H 匹配功能的掩码	0	表 214
MASK_I	R/W	0x0208	位串 I 匹配功能的掩码	0	表 215
MASK_P	R/W	0x020C	位串 P 匹配功能的掩码	0	表 216
GPIO_INREG	R	0x0210	GPIO 输入状态寄存器	0	表 217
GPIO_OUTREG	R/W	0x0214	GPIO 输出控制寄存器	0	表 218
GPIO_OENREG	R/W	0x0218	GPIO OE 控制寄存器	0	表 219
CTRL_ENABLE	R/W	0x021C	使能位串 COUNT 计数器	0	表 220
CTRL_DISABLE	R/W	0x0220	禁用位串 COUNT 计数器	0	表 221
CLR_EN_0	W	0x0F00	移位时钟中断清除掩码	0	表 222
SET_EN_0	W	0x0F04	移位时钟中断设置掩码	0	表 223
ENABLE_0	R	0x0F08	移位时钟中断使能	0	表 224
STATUS_0	R	0x0F0C	移位时钟中断状态	0	表 225
CTR_STAT_0	W	0x0F10	移位时钟中断清除状态	0	表 226
SET_STAT_0	W	0x0F14	移位时钟中断设置状态	0	表 227
CLR_EN_1	W	0x0F20	捕获时钟中断清除掩码	0	表 228
SET_EN_1	W	0x0F24	捕获时钟中断设置掩码	0	表 229
ENABLE_1	R	0x0F28	捕获时钟中断使能	0	表 230
STATUS_1	R	0x0F2C	捕获时钟中断状态	0	表 231
CTR_STAT_1	W	0x0F30	捕获时钟中断清除状态	0	表 232
SET_STAT_1	W	0x0F34	捕获时钟中断设置状态	0	表 233
CLR_EN_2	W	0x0F40	模式匹配中断清除掩码	0	表 234
SET_EN_2	W	0x0F44	模式匹配中断设置掩码	0	表 235

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 201. 寄存器简介：SGPIO （基址 0x4010 1000）*（续）*

名称	访问类型	地址偏移	描述	复位值	参考
ENABLE_2	R	0x0F48	模式匹配中断使能	0	表 236
STATUS_2	R	0x0F4C	模式匹配中断状态	0	表 237
CTR_STAT_2	W	0x0F50	模式匹配中断清除状态	0	表 238
SET_STAT_2	W	0x0F54	模式匹配中断设置状态	0	表 239
CLR_EN_3	W	0x0F60	输入位中断清除掩码	0	表 240
SET_EN_3	W	0x0F64	输入位匹配中断设置掩码	0	表 241
ENABLE_3	R	0x0F68	输入位匹配中断使能	0	表 242
STATUS_3	R	0x0F6C	输入位匹配中断状态	0	表 243
CTR_STAT_3	W	0x0F70	输入位匹配中断清除状态	0	表 244
SET_STAT_3	W	0x0F74	输入位匹配中断设置状态	0	表 245

18.6.1 引脚多路复用器配置寄存器 （OUT_MUX_CFG0 至 15）

每个寄存器控制一个本地输入引脚多路复用器：OUT_MUX_CFG0 至 OUT_MUX_CTRL15 控制引脚 SGPIO0 至 SGPIO15。

注：引脚指的是 LPC43xx 上的引脚，通过引脚配置寄存器配置为 SGPIOn 引脚。

P_OUT_CFG 位控制位串生成数据时数据流的宽度或输出是用作 GPIO 还是时钟。并非所有模式都受所有位串的支持。表 203 显示了 SGPIOn 引脚如何连接至位串。例如， 8 位宽模式仅受位串 A/B、J/M 或 L/N 支持。使用位宽大于 1 的模式时，每个时钟的移位位数应设置为与并行模式的宽度相等 （参见表 207 中的位 PARALLEL_MODE）。

P_OE_CFG 位控制输出使能源。可使用寄存器 GPIO_OEREG 以静态方式或由其他位串以动态方式来进行控制。表 204 指示哪些位串控制哪些引脚的输出使能。请注意，对于位宽大于 1 的模式，输出使能由 2 位定义：一位用于 LSB，一位用于余下的全部 MSB。使用位宽大于 1 的模式 （P_OE_CFG = dout_oem2、 dout_oem4 或 dout_oem8）时，每个时钟的移位位数应设置为每个时钟 2 位。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 202. 引脚多路复用器配置寄存器（OUT_MUX_CFG0 至 15，地址 0x4010 1000 至 0x4010 103C）位描述

位	符号	值	描述	复位值	访问类型
3:0	P_OUT_CFG		输出 SGPIOn 的输出控制。保留所有其它值。	0	R/W
		0x0	dout_doutm1 （1 位模式）		
		0x1	dout_doutm2a （2 位模式 2a）		
		0x2	dout_doutm2b （2 位模式 2b）		
		0x3	dout_doutm2c （2 位模式 2c）		
		0x4	gpio_out （由 GPIO_OUTREG 设置的电平）		
		0x5	dout_doutm4a （4 位模式 4a）		
		0x6	dout_doutm4b （4 位模式 4b）		
		0x7	dout_doutm4c （4 位模式 4c）		
		0x8	clk_out		
		0x9	dout_doutm8a （8 位模式 8a）		
		0xA	dout_doutm8b （8 位模式 8b）		
		0xB	dout_doutm8c （8 位模式 8c）		
6:4	P_OE_CFG		输出使能源。保留所有其它值。	0	R/W
		0x0	gpio_oe（由 GPIO_OEREG 设置的状态）		
		0x4	dout_oem1 （1 位模式）		
		0x5	dout_oem2 （2 位模式）		
		0x6	dout_oem4 （4 位模式）		
		0x7	dout_oem8 （8 位模式）		
31:7	-		保留。	-	-

表 203. 输出引脚多路复用

SGPIO 引脚	输出模式 - 寄存器 OUT_MUX_CFG，位 P_OUT_CFG （参见表 202）											
	1011	1010	1001	0111	0110	0101	0011	0010	0001	0000	1000	0100
	8 位 8c	8 位 8b	8 位 8a	4 位 4c	4 位 4b	4 位 4a	2 位 2c	2 位 2b	2 位 2a	1 位	clk	gpio
0	L0	J0	A0	J0	I0	A0	J0	I0	A0	A0	Bck	A
1	L1	J1	A1	J1	I1	A1	J1	I1	A1	I0	Dck	B
2	L2	J2	A2	J2	I2	A2	I0	J0	E0	E0	Eck	C
3	L3	J3	A3	J3	I3	A3	I1	J1	E1	J0	Hck	D
4	L4	J4	A4	L0	K0	C0	L0	K0	C0	C0	Cck	E
5	L5	J5	A5	L1	K1	C1	L1	K1	C1	K0	Fck	F
6	L6	J6	A6	L2	K2	C2	K0	L0	F0	F0	Ock	G
7	L7	J7	A7	L3	K3	C3	K1	L1	F1	L0	Pck	H
8	N0	M0	B0	N0	M0	B0	N0	M0	B0	B0	Ack	I
9	N1	M1	B1	N1	M1	B1	N1	M1	B1	M0	Mck	J
10	N2	M2	B2	N2	M2	B2	M0	N0	G0	G0	Gck	K
11	N3	M3	B3	N3	M3	B3	M1	N1	G1	N0	Nck	L
12	N4	M4	B4	P0	O0	D0	P0	O0	D0	D0	Ick	M

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 203. 输出引脚多路复用 *（续）*

SGPIO 引脚	输出模式 - 寄存器 OUT_MUX_CFG，位 P_OUT_CFG （参见表 202）											
	1011	1010	1001	0111	0110	0101	0011	0010	0001	0000	1000	0100
	8 位 8c	8 位 8b	8 位 8a	4 位 4c	4 位 4b	4 位 4a	2 位 2c	2 位 2b	2 位 2a	1 位	clk	gpio
13	N5	M5	B5	P1	O1	D1	P1	O1	D1	O0	Jck	N
14	N6	M6	B6	P2	O2	D2	O0	P0	H0	H0	Kck	O
15	N7	M7	B7	P3	O3	D3	O1	P1	H1	P0	Lck	P

表 204. 输出使能控制

SGPIO 引脚	OE 控制 - 寄存器 OUT_MUX_CFG，位 P_OE_CFG				
	111	110	101	100	000
	8 位	4 位	2 位	1 位	gpio
0	H0	H0	H0	B0	A
1	H1	H1	H1	M0	B
2	H1	H1	D0	G0	C
3	H1	H1	D1	N0	D
4	H1	O0	G0	D0	E
5	H1	O1	G1	O0	F
6	H1	O1	O0	H0	G
7	H1	O1	O1	P0	H
8	P0	P0	P0	A0	I
9	P1	P1	P1	I0	J
10	P1	P1	B0	E0	K
11	P1	P1	B1	J0	L
12	P1	N0	N0	C0	M
13	P1	N1	N1	K0	N
14	P1	N1	M0	F0	O
15	P1	N1	M1	L0	P

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 30. SGPIO 本地输出引脚多路复用器配置

18.6.2 SGPIO 多路复用器配置寄存器 （SGPIO_MUX_CFG0 至 15）

每个寄存器控制一个 SGPIO 多路复用器来选择下列源：

- 外部时钟源 （未使用 COUNT 时）
- 外部时钟门控 （静态或由位串或外部引脚控制）
- 输入级联结构

SGPIO_MUX_CFG0 至 SGPIO_MUX_CFG15 控制位串 A（寄存器 0）至 P（寄存器 15）。

要避免振荡，可作为其他位串的时钟源的位串本身不能支持外部位串时钟 (CLK_SOURCE_SLICE_MODE)。这些位串不应将高于 SGPIO_CLOCK/2 的时钟输入到其他位串。

表 205. SGPIO 多路复用器配置寄存器（SGPIO_MUX_CFG0 至 15，地址 0x4010 0040 至 0x4010 007C）位描述

位	符号	值	描述	复位值	访问类型
0	EXT_CLK_ENABLE		选择时钟信号。	0	R/W
		0x0	内部时钟信号 （位串）		
		0x1	外部时钟信号 （引脚）		
2:1	CLK_SOURCE_PIN_MODE		选择源时钟引脚。	0	R/W
		0x0	SGPIO8		
		0x1	SGPIO9		
		0x2	SGPIO10		
		0x3	SGPIO11		

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 205. SGPIO 多路复用器配置寄存器（SGPIO_MUX_CFG0 至 15，地址 0x4010 0040 至 0x4010 007C）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
4:3	CLK_SOURCE_SLICE_MODE		选择时钟源位串。 请注意，位串D、H、O和P不支持该模式。	0	R/W
		0x0	位串 D		
		0x1	位串 H		
		0x2	位串 O		
		0x3	位串 P		
6:5	QUALIFIER_MODE		选择门控模式。	0	R/W
		0x0	使能		
		0x1	禁用		
		0x2	位串 （参见该寄存器的 QUALIFIER_SLICE_MODE 位）		
		0x3	外部 SGPIO 引脚 （8、 9、 10 或 11）		
8:7	QUALIFIER_PIN_MODE		选择门控引脚。	0	R/W
		0x0	SGPIO8		
		0x1	SGPIO9		
		0x2	SGPIO10		
		0x3	SGPIO11		
10:9	QUALIFIER_SLICE_MODE		选择门控位串。	0	R/W
		0x0	位串 A，但位串 A 使用的是位串 D。		
		0x1	位串 H，但位串 H 使用的是位串 O。		
		0x2	位串 I，但位串 I 使用的是位串 D。		
		0x3	位串 P，但位串 P 使用的是位串 O。		
11	CONCAT_ENABLE		使能级联。	0	R/W
		0x0	外部数据引脚		
		0x1	级联数据		
13:12	CONCAT_ORDER		选择级联顺序	0	R/W
		0x0	自环		
		0x1	2 个位串		
		0x2	4 个位串		
		0x3	8 个位串		
31:14	-		保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 206. SGPIO 多路复用器

位串	位串 Din					位串	时钟			
							位串 Din			
	CONCAT_ENABLE						CLK_SOURCE_ SLICE_MODE			
	0	1								
		CONCAT_ORDER								
		00	01	10	11		00	01	10	11
A	引脚 0	A	I	J	L	A	D	H	O	P
I	引脚 8	I	A	A	A	I	D	H	O	P
E	引脚 4	E	J	I	I	E	D	H	O	P
J	引脚 9	J	E	E	E	J	D	H	O	P
C	引脚 2	C	K	L	J	C	D	H	O	P
K	引脚 10	K	C	C	C	K	D	H	O	P
F	引脚 5	F	L	K	K	F	D	H	O	P
L	引脚 11	L	F	F	F	L	D	H	O	P
B	引脚 1	B	M	N	P	B	D	H	O	P
M	引脚 12	M	B	B	B	M	D	H	O	P
G	引脚 6	G	N	M	M	G	D	H	O	P
N	引脚 13	N	G	G	G	N	D	H	O	P
D	引脚 3	D	O	P	N	D	-	-	-	-
O	引脚 14	O	D	D	D	O	-	-	-	-
H	引脚 7	H	P	O	O	H	-	-	-	-
P	引脚 15	P	H	H	H	P	-	-	-	-

18.6.3 位串多路复用器配置寄存器 （SLICE_MUX_CFG0 至 15）

每个寄存器控制一个位串多路复用器移位时钟以及每个时钟移位位串 FIFO 的多少位。

寄存器 SLICE_MUX_CFG0 至 SLICE_MUX_CFG15 控制位串 A （寄存器 0）至 P （寄存器 15）。

位 MATCH_MODE 选择匹配滤波器是否有源或是否捕获了数据。仅位串 A、H、I 和 P 支持使用掩码的匹配 （寄存器 MASK_x）。对于其他位串，不对模式进行掩码。

位 CLKGEN_MODE 选择由位串计数器、外部引脚或其他位串生成的时钟作为移位时钟。

位 INV_OUT_CLK 可反转移位时钟。这应当仅用于来自引脚的外部时钟。

Data_capture_mode 用于定义哪种输入数据条件可生成中断。

有关将位串数据连接到引脚以便对位 PARALLEL_MODE 进行多种设置，请参见表 207。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 207. 位串多路复用器配置寄存器 （SLICE_MUX_CFG0 至 15，地址 0x4010 1080 至 0x4010 10BC）位描述

位	符号	值	描述	复位值	访问类型
0	MATCH_MODE		匹配模式	0	R/W
		0x0	不匹配数据。		
		0x1	匹配数据。		
1	CLK_CAPTURE_MODE		捕获时钟模式	0	R/W
		0x0	使用时钟上升沿。		
		0x1	使用时钟下降沿。		
2	CLKGEN_MODE		时钟生成模式	0	R/W
		0x0	使用 COUNTER 内部生成的时钟。		
		0x1	使用来自引脚或其他位串的外部时钟。		
3	INV_OUT_CLK		反转输出时钟	0	R/W
		0x0	普通时钟。		
		0x1	反向时钟。		
5:4	DATA_CAPTURE_MODE		输入位匹配中断的条件	0	R/W
		0x0	检测上升沿。		
		0x1	检测下降沿。		
		0x2	检测低电平。		
		0x3	检测高电平。		
7:6	PARALLEL_MODE		并行模式	0	R/W
		0x0	每个时钟移位 1 位。		
		0x1	每个时钟移位 2 位。		
		0x2	每个时钟移位 4 位。		
		0x3	每个时钟移位 1 个字节。		
8	INV_QUALIFIER		反转门控	0	R/W
		0x0	使用普通门控。		
		0x1	使用反向门控。		
31:9	-	-	保留。	-	-

18.6.4 位串数据寄存器 （REG0 至 15）

每个寄存器包含一个位串的数据：REG0 至 REG15 包含位串 A（寄存器 0）至位串 P（寄存器 15）的数据。

在有效移位时钟，数据向右移位；捕获的数据在位 31 移入，寄存器数据从位 0 移出。

表 208. 位串数据寄存器 （REG0 至 15，地址 0x4010 10C0 至 0x4010 10FC）位描述

位	符号	描述	复位值	访问类型
31:0	REG	在每个有效移位时钟，寄存器向右移位；用捕获自 DIN(n) 的数据加载 REG(31)，DOUT(n) 设置为 REG(0)。	0	R/W

18.6.5 位串数据影子寄存器 （REG_SS0 至 15）

每个位串数据影子寄存器包含一个位串的数据：REG_SS0 至 REG_SS15 包含位串 A（寄存器 0）至位串 P （寄存器 15）的数据。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

该寄存器控制影子寄存器 REG_SS 内容何时与主寄存器 REG 交换。POS 计数器达到 0x0 时发生交换。

表 209. 位串数据影子寄存器 （REG_SS0 至 15，地址 0x4010 110 至 0x4010 113C）位描述

位	符号	描述	复位值	访问类型
31:0	REG_SS	每当 POS 达到 0x0 时，REG_SS 的内容与 REG 的内容进行交换。	0	R/W

18.6.6 重新加载寄存器 （PRESET0 至 15）

每个寄存器包含一个位串的重新加载值：PRESET0 至 PRESET15 包含位串 A（寄存器 0）至位串 P （寄存器 15）的重新加载值。

该寄存器控制内部生成的位串移位时钟频率：
频率 (shift_clock) = 频率 (SGPIO_CLK) / (PRESET+1)。

表 210. 重新加载寄存器 （PRESET0 至 15，地址 0x4010 1140 至 0x4010 117C）位描述

位	符号	描述	复位值	访问类型
11:0	PRESET	计数器重新加载值； COUNT 达到 0x0 时加载。	0	R/W
31:12	-	保留。	-	-

18.6.7 下行计数器寄存器 （COUNT0 至 15）

该状态寄存器反映位串移位时钟计数器值。如果该计数器需要以已定义的相位启动，则在使用 CTRL_ENABLE 使能计数器前，应当将 COUNT 设置为所需的值。

表 211. 下行计数器寄存器 （COUNT0 至 15，地址 0x4010 1180 至 0x4010 11BC）位描述

位	符号	描述	复位值	访问类型
11:0	COUNT	下行计数器，每个移位时钟周期向下计数。0x0 后的下一计数为 PRESET。	0	R/W
31:12	-	保留。	-	-

18.6.8 位置寄存器 （POS0 至 15）

每个位置寄存器包含一个位串的位置计数器：POS0 至 POS15 包含位串 A （寄存器 0）至位串 P （寄存器 15）的计数器。

该寄存器控制影子寄存器 REG_SS 内容何时与主寄存器 REG 交换。

要每 k x 32 位交换内容， POS_PRESET 应当为 0x20 x k -1。 k 个位串级联时 （设置了 CONCAT_ENABLE），应当使用该设置。

注：位串启动前 （使用 CTRL_ENABLE）， POS 应设置为 POS_PRESET 值。

表 212. 位置寄存器 （POS0 至 15，地址 0x4010 11C0 至 0x4010 11FC）位描述

位	符号	描述	复位值	访问类型
7:0	POS	每当 COUNT 达到 0x0 时， POS 向下计数。	0	R/W
15:8	POS_RESET	POS 达到 0x0 后 POS 的重新加载值。	0	R/W
31:16	-	保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

18.6.9 位串 A 掩码寄存器 (MASK_A)

表 213. 位串 A 掩码寄存器 （MASK_A，地址 0x4010 1200）位描述

位	符号	描述	复位值	访问类型
31:0	MASK_A	位串 A 匹配功能的掩码 0 = 无效。 1 = 掩码该位。	0	R/W

18.6.10 位串 H 掩码寄存器 (MASK_H)

表 214. 位串 H 掩码寄存器 （MASK_H，地址 0x4010 1204）位描述

位	符号	描述	复位值	访问类型
31:0	MASK_H	位串 H 匹配功能的掩码 0 = 无效。 1 = 掩码该位。	0	R/W

18.6.11 位串 I 掩码寄存器 (MASK_I)

表 215. 位串 I 掩码寄存器 （MASK_I，地址 0x4010 1208）位描述

位	符号	描述	复位值	访问类型
31:0	MASK_I	位串 I 匹配功能的掩码 0 = 无效。 1 = 掩码该位。	0	R/W

18.6.12 位串 P 掩码寄存器 (MASK_P)

表 216. 位串 P 掩码寄存器 （MASK_P，地址 0x4010 120C）位描述

位	符号	描述	复位值	访问类型
31:0	MASK_P	位串 P 匹配功能的掩码 0 = 无效。 1 = 掩码该位。	0	R/W

18.6.13 GPIO 输入状态寄存器 (GPIO_INREG)

表 217. GPIO 输入状态寄存器 （GPIO_INREG，地址 0x4010 1210）位描述

位	符号	描述	复位值	访问类型
15:0	GPIO_INi	位 i 反映 SGPIO 引脚 i 的输入状态。 0 = 低电平 1 = 高电平	0	R
31:16	-	保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

18.6.14 GPIO 输出控制寄存器 (GPIO_OUTREG)

表 218. GPIO 输出控制寄存器 （GPIO_OUTREG，地址 0x4010 1214）位描述

位	符号	描述	复位值	访问类型
15:0	GPIO_OUT	GPIO 输出寄存器。位 i 设置 SGPIO 引脚 i 的输出。 0 = 低电平 1 = 高电平	0	R/W
31:16	-	保留。	-	-

18.6.15 GPIO 输出使能寄存器 (GPIO_OENREG)

表 219. GPIO 输出使能寄存器 （GPIO_OENREG，地址 0x4010 1218）位描述

位	符号	描述	复位值	访问类型
15:0	GPIO_OE	位 i 选择 SGPIO 引脚 i 的输出使能状态。 0 = GPIO 输出 i 为三态 1 = GPIO 输出 i 有效。	0	R/W
31:16	-	保留。	-	-

18.6.16 位串计数使能寄存器 (CTRL_ENABLED)

表 220. 位串计数使能寄存器 （CTRL_ENABLED，地址 0x4010 121C）位描述

位	符号	描述	复位值	访问类型
15:0	CTRL_ENABLED	位串计数使能。位 n 控制位串 n（0 = 位串 A、...、15 = 位串 P）。 0 = tbd。 1 = 使能 COUNTn。	0	R/W
31:16	-	保留。	-	-

18.6.17 位串计数禁用寄存器 (CTRL_DISABLED)

如果设置了该寄存器，则当POSi计数器达到零计数时，它会同步禁用位串计数器COUNTi。

启动 COUNTi（通过设置 CTRL_ENABLEi）时，该寄存器应当始终已清除。如果仅在 POSi 需要倒计数（仅应处理一个位串时），则应当在用寄存器 CTRL_ENABLEi 启动 COUNTi 后设置该寄存器。

表 221. 位串计数禁用寄存器 （CTRL_DISABLED，地址 0x4010 1220）位描述

位	符号	描述	复位值	访问类型
15:0	CTRL_DISABLEDn	位串计数禁用。位 n 控制位串 n （0 = 位串 A、...、 15 = 位串 P）。 0 = tbd 1 = 禁用 COUNTn。	0	R/W
31:16	-	保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

18.6.18 移位时钟中断清除掩码寄存器 (CLR_EN_0)

该寄存器清除某个位串的移位时钟中断掩码。

表 222. 移位时钟中断清除掩码寄存器

位	符号	描述	复位值	访问类型
15:0	CLR_SCI	1 = 位串 n 的移位时钟中断清除掩码。	0	W
31:16	-	保留。	-	-

18.6.19 移位时钟中断设置掩码寄存器 (SET_EN_0)

该寄存器掩码某个位串的移位时钟中断。

表 223. 移位时钟中断设置掩码寄存器 （SET_EN_0，地址 0x4010 1F04）位描述

位	符号	描述	复位值	访问类型
15:0	SET_SCI	1 = 位串 n 的移位时钟中断设置掩码。	0	W
31:16	-	保留。	-	-

18.6.20 移位时钟中断使能寄存器 (ENABLE_0)

该寄存器指示是否使能了某个位串的移位时钟中断。

表 224. 移位时钟中断使能寄存器 （ENABLE_0，地址 0x4010 1F08）位描述

位	符号	描述	复位值	访问类型
15:0	ENABLE_SCI	1 = 位串 n 的移位时钟中断使能。	0	R
31:16	-	保留。	-	-

18.6.21 移位时钟中断状态寄存器 (STATUS_0)

该寄存器指示某个位串的移位时钟中断状态。

表 225. 移位时钟中断状态寄存器 （STATUS_0，地址 0x4010 1F0C）位描述

位	符号	描述	复位值	访问类型
15:0	STATUS_SCI	位串 n 的移位时钟中断状态。	0	R
31:16	-	保留。	-	-

18.6.22 移位时钟中断清除状态寄存器 (CTR_STATUS_0)

该寄存器清除某个位串的移位时钟中断。

表 226. 移位时钟中断清除状态寄存器 （CTR_STATUS_0，地址 0x4010 1F10）位描述

位	符号	描述	复位值	访问类型
15:0	CTR_STATUS_SCI	位串 n 的移位时钟中断清除状态。	0	W
31:16	-	保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

18.6.23 移位时钟中断设置状态寄存器 (SET_STATUS_0)

该寄存器设置某个位串的移位时钟中断。

表 227. 移位时钟中断设置状态寄存器 （SET_STATUS_0，地址 0x4010 1F14）位描述

位	符号	描述	复位值	访问类型
15:0	CTR_STATUS_SCI	位串 n 的移位时钟中断设置状态。	0	W
31:16	-	保留。	-	-

18.6.24 捕获时钟中断清除掩码寄存器 (CLR_EN_1)

表 228. 捕获时钟中断清除掩码寄存器 （CLR_EN_1，地址 0x4010 1F20）位描述

位	符号	描述	复位值	访问类型
15:0	CLR_EN_CCI	1 = 位串 n 的捕获时钟中断清除掩码。	0	W
31:16	-	保留。	-	-

18.6.25 捕获时钟中断设置掩码寄存器 (SET_EN_1)

表 229. 捕获时钟中断设置掩码寄存器 （SET_EN_1，地址 0x4010 1F24）位描述

位	符号	描述	复位值	访问类型
15:0	SET_EN_CCI	1 = 位串 n 的捕获时钟中断设置掩码。	0	W
31:16	-	保留。	-	-

18.6.26 捕获时钟中断使能 (ENABLE_1)

表 230. 捕获时钟中断使能寄存器 （ENABLE_1，地址 0x4010 1F28）位描述

位	符号	描述	复位值	访问类型
15:0	ENABLE_CCI	位串 n 的捕获时钟中断使能。	0	R
31:16	-	保留。	-	-

18.6.27 捕获时钟中断状态寄存器 (STATUS_1)

表 231. 捕获时钟中断状态寄存器 （STATUS_1，地址 0x4010 1F2C）位描述

位	符号	描述	复位值	访问类型
15:0	STATUS_CCI	位串 n 的捕获时钟中断状态。	0	R
31:16	-	保留。	-	-

18.6.28 捕获时钟中断清除状态寄存器 (CTR_STATUS_1)

表 232. 捕获时钟中断清除状态寄存器 （CTR_STATUS_1，地址 0x4010 1F30）位描述

位	符号	描述	复位值	访问类型
15:0	CTR_STATUS_CCI	位串 n 的捕获时钟中断清除状态。	0	W
31:16	-	保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

18.6.29 捕获时钟中断设置状态寄存器 (SET_STATUS_1)

表 233. 捕获时钟中断设置状态寄存器 （SET_STATUS_1，地址 0x4010 1F34）位描述

位	符号	描述	复位值	访问类型
15:0	CTR_STATUS_CCI	位串 n 的捕获时钟中断设置状态。	0	W
31:16	-	保留。	-	-

18.6.30 模式匹配中断清除掩码寄存器 (CLR_EN_2)

表 234. 模式匹配中断清除掩码寄存器 （CLR_EN_2，地址 0x4010 1F40）位描述

位	符号	描述	复位值	访问类型
15:0	CLR_EN2_PMI	1 = 位串 n 的匹配中断清除掩码。	0	W
31:16	-	保留。	-	-

18.6.31 模式匹配中断设置掩码寄存器 (SET_EN_2)

表 235. 模式匹配中断设置掩码寄存器 （SET_EN_2，地址 0x4010 1F44）位描述

位	符号	描述	复位值	访问类型
15:0	SET_EN_PMI	1 = 位串 n 的匹配中断设置掩码。	0	W
31:16	-	保留。	-	-

18.6.32 模式匹配中断使能 (ENABLE_2)

表 236. 模式匹配中断使能寄存器 （ENABLE_2，地址 0x4010 1F48）位描述

位	符号	描述	复位值	访问类型
15:0	ENABLE_PMI	位串 n 的匹配中断使能。	0	R
31:16	-	保留。	-	-

18.6.33 模式匹配中断状态寄存器 (STATUS_2)

表 237. 模式匹配中断状态寄存器 （STATUS_2，地址 0x4010 1F4C）位描述

位	符号	描述	复位值	访问类型
15:0	STATUS_PMI	位串 n 的匹配中断状态。	0	R
31:16	-	保留。	-	-

18.6.34 模式匹配中断清除状态寄存器 (CTR_STATUS_2)

表 238. 模式匹配中断清除状态寄存器 （CTR_STATUS_2，地址 0x4010 1F50）位描述

位	符号	描述	复位值	访问类型
15:0	CTR_STATUS_PMI	位串 n 的匹配中断清除状态。	0	W
31:16	-	保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

18.6.35 模式匹配中断设置状态寄存器 (SET_STATUS_2)

表 239. 模式匹配中断设置状态寄存器 （SET_STATUS_2，地址 0x4010 1F54）位描述

位	符号	描述	复位值	访问类型
15:0	CTR_STATUS_PMI	位串 n 的匹配中断设置状态。	0	W
31:16	-	保留。	-	-

18.6.36 输入位中断清除掩码寄存器 (CLR_EN_3)

表 240. 输入位中断清除掩码寄存器 （CLR_EN_3，地址 0x4010 1F60）位描述

位	符号	描述	复位值	访问类型
15:0	CLR_EN_INPI	1 = 位串 n 的输入位中断清除掩码。	0	W
31:16	-	保留。	-	-

18.6.37 输入位匹配中断设置掩码寄存器 (SET_EN_3)

表 241. 输入中断设置掩码寄存器 （SET_EN_3，地址 0x4010 1F64）位描述

位	符号	描述	复位值	访问类型
15:0	SET_EN_INPI	1 = 位串 n 的输入中断设置掩码。	0	W
31:16	-	保留。	-	-

18.6.38 输入位匹配中断使能 (ENABLE_3)

表 242. 输入中断使能寄存器 （ENABLE_3，地址 0x4010 1F68）位描述

位	符号	描述	复位值	访问类型
15:0	ENABLE3_INPI	位串 n 的输入中断使能。	0	R
31:16	-	保留。	-	-

18.6.39 输入位匹配中断状态寄存器 (STATUS_3)

表 243. 输入中断状态寄存器 （STATUS_3，地址 0x4010 1F6C）位描述

位	符号	描述	复位值	访问类型
15:0	STATUS_INPI	位串 n 的输入中断状态。	0	R
31:16	-	保留。	-	-

18.6.40 输入位匹配中断清除状态寄存器 (CTR_STATUS_3)

表 244. 输入位中断清除状态寄存器 （CTR_STATUS_3，地址 0x4010 1F70）位描述

位	符号	描述	复位值	访问类型
15:0	CTR_STATUS_INPI	位串 n 的输入位中断清除状态。	0	W
31:16	-	保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

18.6.41 输入位匹配中断设置状态寄存器 (SET_STATUS_3)

表 245. 移位时钟中断设置状态寄存器 （SET_STATUS_3，地址 0x4010 1F74）位描述

位	符号	描述	复位值	访问类型
15:0	CTR_STATUS_INPI	位串 n 的移位中断设置状态。	0	W
31:16	-	保留。	-	-

18.7 功能说明

串行 GPIO (SGPIO) 提供标准 GPIO 功能，并具备加速串行数据流处理的增强功能。增强型功能使用所谓的位串构成。所有 16 个位串具有相同的基本功能集。某些位串还具有附加功能，可用于模式匹配和处理 2 位、 4 位或 8 位宽的数据流。

图 31. SGPIO 框图

位串执行并行到串行数据转换（反之亦然）。一个位串包含 32 个以链条形式连接在一起的 1 位寄存器 (REG)。数据通过链条右移。输入数据在 MSB 处移入，从 LSB 移出。

输入数据被同步；这就产生了一个 SGPIO_CLOCK 周期的延迟。更改引脚方向时，尤其是在模拟高速双向总线时，应考虑该延迟。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 32. 一个位串的基本操作

移位时钟控制数据通过链条移位的速率。位的移位增量由寄存器 parallel_mode 设置为 1 位、 2 位、 4 位或 8 位。多通道模式中使用大于 1 的移位。

移位时钟可来自外部源 （参见 <tbd>）或从本地生成。移位时钟可由外部引脚或其他位串来限定。本地时钟由以 SGPIO_CLOCK 运行的 12 位下行计数器 COUNT 构成。计数持续时间可通过寄存器 PRESET 预设。 COUNT 达到零时用 PRESET 加载。

移位时钟频率因此与频率 (shift_clock) = 频率 (SGPIO_CLK) / (PRESET+1) 相等。

每当 COUNT 达到零时寄存器右移；用捕获自 DIN 的数据加载位 REG[31]，用位 REG[0] 加载 DOUT。因此，COUNT 控制串行数据速率。当几个位串用于创建接口端口时，不同位串间的相位可通过使用不同的 CONUT 初始值来控制。

另外一个下行计数器 (POS) 控制并行数据加载到链条和从链条存储的时间。POS 计数器由 COUNT 等于零时会递减的 8 位计数器组成。 POS 达到零时，用 PRESET_POS 值加载。位串数据使用双缓冲方案进行交换。 POS 达到零时，位串寄存器 REG 的内容与位串缓冲区 REG_SS 的内容交换。这使得 CPU 有更多时间来处理 REG_SS 缓冲区的内容。PRESET_POS 可用于指示并非 32 位的倍数的字的字边界。如果 PRESET_POS 值为零，则在每个移位时钟后 REG 会与 REG_SS 交换。

18.7.1 级联

可将位串级联以增加缓冲区大小，使其超出 REG_SS。借助该功能，还可通过执行反向级联来创建 PWM 流。

级联由寄存器 SGPIO_MUX_CFG 设置。通过字段 CONCAT_ENABLE 使能该功能。多个位串级联时，该字段的设置对所有相关位串应相同。字段 CONCAT_ORDER 设置级联的大小。对于大小为零的自环模式，位串输出会传回到位串输入。该模式可用于创建重复数据流。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

下面的图 33 显示了哪些位串可级联。位串排列在 4 个组中，以位串 A、B、C、D 开始。每个组由 4 个级联位串组成。例如，位串 A 输入可使用 3 个额外位串 I、E 和 J 来扩展。如果是输出模式，则位串 I 可由位串 A、I 和 E 扩展。位串进一步排列在四个子组中，以 E、F、G、 H 开始。这些子组支持两个位串的级联。位串 A 和 B 最多支持 7 个位串的级联。

如果多个位串级联，则这些位串的 concat_order 应当一致。就上例而言，输入位串 A 应设置为输入 (concat_enable=0)。如果未设置 concat_enable，则 concat_order 值为 “不关注”值。对于位串 I、E 和 J，设置 concat_enable=1 并使用 concat_order =10（4 个位串）。

级联位串应当具有一致的移位时钟和位移位设置（SLICE_MUX_CFG 中）。K 个位串级联时， PRESET_POS 应设置为 0x20 x k - 1。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

E = 外部数据输入模式

8 = 8 个位串级联模式

4 = 4 个位串级联模式

2 = 2 个位串级联模式

0 = 0 个位串级联（自环）模式

图 33. 级联互连

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

18.7.2 模式匹配

所有位串均具有模式匹配功能。该功能可用于检测启动代码等。要使用该功能，则必须用需匹配的模式来对 REG_SS 编程 （请注意， POS 达到零时 REG_SS 不会与 REG 交换！） MATCH_MODE 位必须设为 1。输入数据现在与已编程的模式相比较。发现匹配时会提出模式匹配中断。

四个位串 （A、H、I 和 P）同样支持对模式的掩码；必须设置 MASK_x 以便对模式位进行比较 （‘1’表示比较）。例如，寻找模式 0x1234.xxxx.5678.9ABC 时， REG 应设置为 0x1234.xxxx.5678.9ABC， MASK 设置为 0xFFFF.0000.FFFF.FFFF。

18.7.3 引脚复用

位串输入和输出通过本地引脚多路复用器连接到全局引脚多路复用器 （第 15 章）。

针对普通GPIO功能（输入、输出、输出使能），每个引脚同样可使用寄存器GPIO_INREG、GPIO_OUTREG 和 GPIO_OEREG 来控制。

表 203 显示了输出多路复用方案。这些设置由寄存器 OUT_MUX_CFG 控制。表 244 显示了输入多路复用方案。这些设置由寄存器 SGPIO_MUX_CFG 控制。

16 个位串以 A 到 P 表示。后缀表示与引脚连接的位串位，例如， 8 位并行输入模式下，引脚 SGPIO0 至 7 连接到位串 A 位 31 至 24。

并非所有功能都可用于所有引脚。例如，仅引脚 SGPIO9 至 11 可用作时钟源输入或门控输入或门控输出。

18.7.4 位串多路复用器

位串多路复用器选择外部位串时钟和位串时钟门控。它由寄存器 SGPIO_MUX_CFG 配置。

时钟源可以来自引脚 8、9、10 或 11，也可以来自位串 D、H、O 或 P。请注意，可用作时钟源的位串不能以自身或其他位串作为时钟源。外部时钟与内部 SGPIO 时钟同步。为防止混叠，时钟源频率应小于 SGPIO 频率的一半。

门控信号来自引脚 8、9、10、11 或位串 A、H、I、P。由于位串无法将门控反馈给自身，因此还使用其他一些位串。位串 D 可作为位串 A 和位串 I 的门控，位串 O 可作为位串 H 和位串 P 的门控。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 34. SGPIO_MUX_CFG 位串多路复用器设置

表 246. 位串 I/O 多路复用

x = 外部； cl = 时钟； q = 门控

SGPIO 引脚	输入模式 并行模式					
	8 位	**4 位**	**2 位**	**1 位**	**时钟**	**Q**
0	A24	A28	A30	A31		
1	A25	A29	A31	I31		
2	A26	A30	E30	E31		
3	A27	A31	E31	J31		
4	A28	C28	C30	C31		
5	A29	C29	C31	K31		
6	A30	C30	F30	F31		
7	A31	C31	F31	L31		
8	B31	B28	B30	B31	xcl	xq
9	B30	B29	B31	M31	xcl	xq

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 246. 位串 I/O 多路复用 *（续）*
x = 外部； cl = 时钟； q = 门控

SGPIO 引脚	输入模式 并行模式 8 位	4 位	2 位	1 位	时钟	Q
10	B29	B30	G30	G31	xcl	xq
11	B28	B31	G31	N31	xcl	xq
12	B27	D28	D30	D31		
13	B26	D29	D31	O31		
14	B25	D30	H30	H31		
15	B24	D31	H31	P31		

18.7.5 内部连接

SGPIO 引脚 10 和 12 可启动 12 位 ADC。

SGPIO 引脚 14 和 15 可启动 GPDMA 复位。

SGPIO 引脚 3 和 12 连接至 SCT 和定时器捕获输入。对于 I2S 应用，SGPIO3 和 SGPIO12 引脚同样支持 128 分频的信号 （SGPIO3_DIV 和 SGPIO12_DIV）。

启动信号可由相关的位串创建（例如，引脚 15 由 1 位输出模式下的位串 P 控制），或使用 GPIO 模式下的这些引脚创建。

18.8 示例

表 247. LPC43xx 上的 SGPIO 应用

示例	描述
逻辑分析器	尽可能快地捕获多达 16 行
模式生成器	尽可能快地移出多达 16 行，数据输入环回 （如果需要）
PWM	与模式生成器相似，现在形成环路。位串可级联来支持更长的模式。 将更多位设置为高位同样可实现固定频率的变化占空比。
SPIFI	使用数据的 4 位移位模式。使用 oe 来改变数据总线方向。 GPIO 用于 CS，一个位串用于时钟。
I2C	将 oe 路径用作数据，将输出数据连接到 0。
I2S	每个立体声通道使用一个位串。主机或从机模式下，使用本地或外部时钟。位时钟和主机时钟以数据时钟的整数倍运行。
SPDIF	每个 SPDIF 音频接口使用一个位串。
TDM	每个 TDM 音频接口使用一个位串。
UART	使用匹配功能来检测启动位，使用门控 （如果需要）捕获首个数据。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

18.8.1 多通道 I2S

18.8.1.1 I2S 位串选择

主机模式下的 5.1 通道 I2S 输出接口需要 3 个数据输出 (SD[2:0])、1 个字选择输出 (WS) 和 1 个时钟输出 (SCK)。从机模式下 SCK 成为输入。这表示输出 SCK 应同样支持时钟输入。这由在串行输出模式下映射到位串 B、M、G 或 N 的引脚 8-10 支持，我们使用的是位串 B。因此，这些位串不应当用于 SD 或 WS 信号。

如果需要过采样从机时钟 (MCK)，则使用能为其他位串 （D、 H、 O 或 P）创建时钟的位串，例如使用位串 D。从机模式下， MCK 为输入。MCK 经过分频后用来创建数据、WS 和 SCK 的移位时钟。主机模式下， MCK 为输出。

输出音频数据速率为 6 x Fs。如果 Fs = 192 kHz，这会变成 1.152 MWps，因此对于 100+ MHz 的 CPU 频率来说相对较低。可使用单位串 （无级联），例如，位串 A、 I 和 E。 WS 由位串 J 构成。这就产生了下列映射：

表 248. I2S 5.1 的 SGPIO 位串映射

位串：功能	A、I、E：SD[2:0]	J：WS	B：SCK	D：MCK

图 35. 5.1 通道 I2S 输出映射至 SGPIO 位串

图 36. I2S 配置

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

18.8.1.2 I2S 位串配置

使用 FS = 192 kHz 和 32 位音频采样可提供下列参数：

- WS=FS=192 kHz
- SCK = 2 x sample_width x FS = 2 x 32 x 192 kHz = 12.288 MHz。位串移位时钟应当是该速率的两倍。
- MCK = oversampling_rate x SCK= 4 x SCK = 49.152 MHz。

将 SGPIO IP 时钟 (SGPIO_CLOCK) 设置为 2 x MCK = 98.304 MHz。

输出宽度、 1 位串行以及输出使能由 GPIO 静态控制，其设置如表 249 中所示。

表 249. I2S 5.1 的 SGPIO 设置， OUT_MUX_CFG 寄存器

OUT_MUX_CFGi	A,I,E (i=0,8,4)	J (i=9)	B (i=1)	D (i=3)
P_out_cfg	0000: dout_doutm1	dout_doutm1	dout_doutm1	dout_doutm1
P_oe_cfg	000: gpio_oe	gpio_oe	gpio_oe	gpio_oe
GPIO_OUTREG	1	1	1	1

主机模式下，移位时钟由本地位串 COUNTER 生成。 WS 和 CK 位串包含重复模式，在自环模式下级联。

表 250. I2S 5.1 的 SGPIO 设置， SGPIO_MUX_CFG 寄存器

SGPIO_MUX_CFGi	A,I,E (i=0,8,4)	J (i=9)	B (i=1)	D (i=3)
ext_clk_enable	x	x	x	x
qualifier_mode	00: 使能	00: 使能	00: 使能	00: 使能
concat_enable	0； 否	1； 是	1； 是	1； 是
concat_order	x	00； 自环	00； 自环	00； 自环

数据宽度为 1 位，所有位串的移位量均为每个时钟 1 位。

表 251. I2S 5.1 的 SGPIO 设置， SLICE_MUX_CFG 寄存器

SLICE_MUX_CFGi	A,I,E (i=0,8,4)	J (i=9)	B (i=1)	D (i=3)
match_mode	0: 否	0: 否	0: 否	0: 否
clk_gen_mode	0: 使用COUNTER clk	0: 使用COUNTER clk	0: 使用COUNTER clk	0: 使用COUNTER clk
parallel_mode	00: 1 位 / 时钟	00: 1 位 / 时钟	00: 1 位 / 时钟	00: 1 位 / 时钟

所有位串计数器在相同相位以值 0 启动。

所有数据位串将位串寄存器长度和音频采样宽度用于 POS。对于 32 位、16 位或 8 位采样，POS 设置为 width-1：0x1F、 0x0F 或 0x07。对于 16 位或 8 位采样， POS 同样可设置为 0x1F 并处理 2 个或 4 个采样。自环位串应始终使用完整的 32 位大小。

表 252. I2S 5.1 的 SGPIO 设置，位串

位串	A,I,E (i=0,8,4)	J (i=9)	B (i=1)	D (i=3)
PRESETi	7: f=clk/8	7: f=clk/8	3: f=clk/4	0: f=clk
COUNTi	0	0	0*[1]	0
POS_PRESETi	0x1F/0x0F/0x07	0x1F	0x1F	0x1F

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

[1] 值 1 或 2 可用于更改相对于数据和 WS 的 SCK 相位。

I2S 从机模式下使用外部时钟输入。有两种情况：

1. SCK 主机模式：SCK 在引脚 8 处提供，该时钟下降沿用作所有位串的移位时钟。不使用任何 MCK。
2. MCK 主机模式：MCK 在引脚 12 处提供给位串 D，在该处，过采样率 (=4) 对该时钟进行分频。位串 D 的输出用作其他位串的移位时钟。

SCK 主机模式下，SCK 输入为 SD 和 WS 信号的移位时钟。不使用 MCK。SCK 从机模式 (1) 下，在引脚 8 处提供的不同位串设置如表 253 所示：

表 253. I2S 5.1 的 SGPIO 设置 （主机模式，引脚 8）

OUT_MUX_CFGi	A,I,E (i=0,8,4)	J (i=9)	B (i=1)	D (i=3)
P_out_cfg	0000: dout_doutm1	dout_doutm1	未使用位串	未使用位串
P_oe_cfg	000: gpio_oe	gpio_oe	未使用位串	未使用位串
GPIO_OUTREG	1	1	0	0
SGPIO_MUX_CFGi				
ext_clk_enable	1: 引脚	1: 引脚	未使用位串	未使用位串
clk_source_pin	00: 引脚 8	00: 引脚 8	未使用位串	未使用位串
qualifier_mode	00: 使能	00: 使能	未使用位串	未使用位串
SLICE_MUX_CFGi				
clk_gen_mode	1: 使用外部时钟	1: 使用外部时钟	未使用位串	未使用位串
inv_out_clk	1: 反向时钟	1: 反向时钟	未使用位串	未使用位串
PRESETi	未使用计数器	未使用计数器	未使用计数器	未使用计数器
COUNTi	未使用计数器	未使用计数器	未使用计数器	未使用计数器
POS_PRESETi	0x1F/0x0F/0x07	0x1F	未使用位串	未使用位串

MCK 主机模式下，MCK 输入经过分频后生成 SCK 并将 SD 和 WS 信号移位。4x 过采样时钟 MCK 从机模式 (1) 下，在引脚 9 处提供的不同位串设置如下所示：

表 254. I2S 5.1 的 SGPIO 设置 （主机模式，引脚 9）

OUT_MUX_CFGi	A,I,E (i=0,8,4)	J (i=9)	B (i=1)	D (i=3)
P_out_cfg	0000: dout_doutm1	0000: dout_doutm1	1000: clk	x
P_oe_cfg	000: gpio_oe	000: gpio_oe	000: gpio_oe	x
GPIO_OUTREG	1	1	1	0
SGPIO_MUX_CFGi				
ext_clk_enable	0: 内部时钟	0: 内部时钟	0: 内部时钟	1: 引脚
clk_source_pin	x	x	x	01: 引脚 9
clk_source_slice	00: 位串 D	位串 D	位串 D	x
qualifier_mode	00: 使能	00: 使能	00: 使能	00: 使能
SLICE_MUX_CFGi				
clk_gen_mode	1: 使用外部时钟	1: 使用外部时钟	1: 使用外部时钟	1: 使用外部时钟

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 254. I2S 5.1 的 SGPIO 设置 （主机模式，引脚 9）*（续）*

clk_capture_mode	1: 使用下降时钟	1: 使用下降时钟	未使用	x
PRESETi	未使用	未使用	未使用	3: f=MCK/4
COUNTi	未使用	未使用	未使用	0
POS_PRESETi	0x1F/0x0F/0x07	0x1F	未使用	0x1F

18.8.1.3 I2S 位串编程

配置后，数据模式在 REGi 和 REG_SSi 中加载。

对于 SD，音频采样在 REGi 和 REG_SSi 中加载。数据从 LSB 开始移出。某个采样处理后， POS 达到倒计数并且 REG_SSi 与 REGi 进行交换以加载下一个采样。在当前采样完成时并进行下一个 POS 倒计数之前， CPU 应当使用新的采样来更新 REG_SSi。

WS 模式每 64 位会重复，存储在 REG9 和 REG_SS9 中。要创建图 35 中所示的 WS 模式（32 位数据宽度），则设置 REG9 = 0x0000.0001 和 REG_SS9=0xFFFF.FFFE。

SCK 模式为静态，存储在 REG3 和 REG_SS3 中。要创建图 35 中所示的 SCK 模式，则设置 REG1 = 0x5555.5555 和 REG_SS1 = 0x5555.5555。要反转时钟相位，则改用模式 0xAAAA.AAAA。

MCK 与其他 I2S 信号的相位不对齐。要切换输出，则设置 REG3=0x5555.5555 和 REG_SS3 = 0x5555.5555。从机模式下， MCK 应当 4 分频来创建 SCK 以及 D 和 WS 移位时钟，这就需要模式 11001100，因此 REG3 = REG_SS3=0xCCCC.CCCC。

写入 CTRL_ENABLE = 0x031B 后使能 COUNTER 可启动所有位串。

18.8.2 相机接口示例

相机接口使用下列输入信号：

- DIN[0:7] 数据输入
- HSYNC 水平同步输入
- VSYNC 垂直同步输入
- PIXCLX 像素时钟输入

图 37. SGPIO 相机接口配置

DIN 输入在 8 位输入上捕获。较宽输入 （10 位或 12 位）需要附加的 2 位或 4 位输入。将 8 位和 2 位 （或 4 位）数据组合到 10 位或 12 位单字的操作必须在软件内进行。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

从支持 8 位输入模式的位串 A 和 B 中，选择位串 A。要使 CPU 实时负载最小，则级联这 8 个位串：位串 A、 I、 E、 J、 C、 K、 F 和 L。

输入在 PIXCLK 下降沿上捕获。从可用作时钟输入的引脚 8-11 中，选择引脚 8。 HSCYNC 用作输入数据的门控。引脚 8-11 可用作门控；引脚 8 已使用，因此使用引脚 9。引脚 9 同样是位串 M 的输入。 VSYNC 使用某个剩余的空闲位串，例如 G。

输出 SGPIO15（位串 P）作为内部信号用于在某个数据模块传输到本地 SRAM 后请求 DMA 传输。

表 255. 相机接口的 SGPIO 位串映射

位串：功能	A、I、E、J、C、K、F、L：DIN[7:0]	引脚 8：PIXCLK	引脚 9/M：HSCYNC	G：VSYNC	P：DMA_REQ

18.8.2.1 相机接口位串配置

所有接口信号仅为输入。输出配置为 “不关注”。

表 256. 相机接口的 SGPIO 设置 （OUT_MUX_CFG 寄存器）

OUT_MUX_CFGi	A...L	引脚 8	M (i=12)	G (i=6)
P_out_cfg	x	x	x	x
P_oe_cfg	x	x	x	x
GPIO_OUTREG	0	0	0	0

使用 HSYNC （引脚 9）作为门控在 PIXCLK （引脚 8）处将数据移位。

表 257. 相机接口的 SGPIO 设置 （SGPIO_MUX_CFG 寄存器）

SGPIO_MUX_CFGi	A...L	M (i=12)	G (i=6)
ext_clk_enable	1= 引脚	1= 引脚	1= 引脚
clk_source_pin_mode	00= 引脚 8	00= 引脚 8	00= 引脚 8
qualifier_mode	11: 外部引脚	00: 使能	00: 使能
qualifier_pin_mode	01: 引脚 9	x	x
concat_enable	1: 是	0: 否	0:; 否
concat_order	11: 8 个位串 concat	x	x

数据宽度为 8 位，所有数据位串的移位量均为每个时钟 8 位。

由于外部时钟用作移位时钟，因此不使用 PRESET 和 COUNT。

HSYNC 和 VSYNC 被检测下降沿来产生中断并指示行或帧启动。 HSYNC 为低电平时，捕获 DIN。软件应当丢弃不需要的数据行。

或者检测嵌入式代码 0xFF.00.00.YY （行启动 / 结束，帧启动 / 结束），位串 A 设置为 match_mode， YY 包含实际代码。请注意， match_mode 下不使用影子缓冲区。要使用影子缓冲区，一旦检测到嵌入式代码，就应当禁用 match_mode。

将 8 个位串级联，产生 8 x 32/8 = 32 个移位，直到 REG 需要影子，因此 POS = 0x1F。

表 258. 相机接口的 SGPIO 设置 （SLICE_MUX_CFG 寄存器）

SLICE_MUX_CFGi	A...L	M (i=12)	G (i=6)
match_mode	0: 否	0: 否	0: 否
clk_capture_mode	1: 下降沿	1: 下降沿	1: 下降沿

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 258. 相机接口的 SGPIO 设置 （SLICE_MUX_CFG 寄存器） *（续）*

clkgen_mode	1: 使用引脚时钟	1: 使用引脚时钟	1: 使用引脚时钟
data_capture_mode	x	01: 检测下降沿	01: 检测下降沿
parallel_mode	11: 1 字节 /clk	x	x
inv_qualifier	1: 反向门控	x	x
PRESETi	x	x	x
COUNTi	x	x	x
POS_PRESETi	0x1F	x	x

写入 CTRL_ENABLE=0x1F3F 后启动位串 A、I、E、J、C、K、F、L 和 M 可启动接口。

HSYNC 为低电平时数据在 PIXCLK 下降沿处捕获。在 POS 中断处，32 数据字从 REG_SS 读取并写入数据 SRAM。然后 SGPIO15 被切换以请求从数据 SRAM 到最终目标的 32 字 GP-DMA 传输。如果每行 （或每帧）的 DIN 并非 32 数据字的倍数，则软件应当在行 （或帧）结束时读取 POS 计数器以确定是否所有数据都已在 REG 中捕获。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

19.1 本章导读

GPDMA 在所有 LPC43xx 器件上可用。

注：VADC 在器件 LPC4350/30/20/10 上不可用。

19.2 基本配置

GPDMA 配置如下：

- 参见表 259 以了解时钟与电源控制。
- GPDMA 通过 DMA_RST （复位号 19）复位。
- CREG模块中的DMAMUX寄存器（参见表41）为每条GPDMA至外设线路在多达3个外设之间选择。
- GPIO 模块、 WWDT 以及定时器可由 GPDMA 作为存储器到存储器传输访问。

表 259. GPDMA 时钟和电源控制

	基本时钟	分支时钟	工作频率
GPDMA	BASE_M4_CLK	CLK_M4_DMA	204 MHz

19.3 功能要点

- 八个 DMA 通道。每个通道可支持一个单向传输。
- 16 条 DMA 请求线。
- 单发DMA和连发DMA请求信号。每个连接到DMA控制器的外围设备可以发出一个连发 DMA 请求或一个单发 DMA 请求。 DMA 连发大小通过编程 DMA 控制器进行设置。
- 支持存储器到存储器、存储器到外围设备、外围设备到存储器和外围设备到外围设备的传输。
- GPIO 模块、 WWDT 以及定时器可由 GPDMA 作为存储器到存储器传输访问。
- 通过使用链表可支持分散或收集 DMA。这意味着源区和目标区不一定要占用连续的存储区。
- 硬件 DMA 通道的优先级。
- AHB从机DMA编程接口。通过AHB从机接口对DMA控制寄存器写入，从而对DMA控制器进行编程。
- 两个用于传输数据的AHB总线主机。这些接口在DMA请求有效时传输数据。主机1可以访问存储器和外围设备，主机 0 仅可访问存储器。
- 32 位 AHB 主机总线宽度。
- 来源和目标的递增或非递增寻址。
- 可编程的 DMA 连发大小。编程 DMA 连发大小可以提高传输数据的效率。
- 每个通道的内部四字 FIFO。
- 支持 8、 16 和 32 位宽的传送。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- 支持大端和小端。复位时， DMA 控制器默认为小端模式。
- 在 DMA 完成后或当 DMA 发生错误时，可中断处理器。
- 原始中断状态。掩码前，可以读取 DMA 错误和 DMA 计数的原始中断状态。

19.4 简介

DMA 控制器允许外围设备到存储器、存储器到外围设备、外围设备到外围设备和存储器到存储器之间的传输。每个 DMA 流都可以为单个源和目标提供单向串行 DMA 传输。例如，一个双向端口就需要一个发送流和一个接收流。对主机 1 而言，源和目标区既可以是存储区，也可以是外设；主机 0 仅可访问存储器 （参见第 3.5 节）。

图 38. DMA 控制器框图

19.5 DMA 系统连接

DMA 控制器与所支持外围设备的连接如表 260 中所示。对于将外围设备连接至 DMA 的每个通道，LPC43xx 支持多种复用选项。CREG 模块的 DMAMUX 寄存器控制所要使用的选项 （参见表 41）。

表 260. 外围设备至 DMA 控制器的连接以及匹配流控制信号

外围设备编号	DMA 复用选项（参见表 41）	SREQ	BREQ
0	0x0	SPIFI	SPIFI
	0x1	-	SCT 匹配 2
	0x2	-	SGPIO14
	0x3	-	定时器 3 匹配 1
1	0x0	-	定时器 0 匹配 0
	0x1	-	USART0 发送
	0x2	保留	保留
	0x3	保留	保留

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 260. 外围设备至 DMA 控制器的连接以及匹配流控制信号 *（续）*

外围设备编号	DMA 复用选项（参见表 41）	SREQ	BREQ
2	0x0	-	定时器 0 匹配 1
	0x1	-	USART0 接收
	0x2	保留	保留
	0x3	保留	保留
3	0x0	-	定时器 1 匹配 0
	0x1	-	UART1 发送
	0x2	-	I2S1 DMA 请求 0
	0x3	SSP1 发送	SSP1 发送
4	0x0	-	定时器 1 匹配 1
	0x1	-	UART1 接收
	0x2	-	I2S1 DMA 请求 1
	0x3	SSP1 接收	SSP1 接收
5	0x0	-	定时器 2 匹配 0
	0x1	-	USART2 发送
	0x2	SSP1 发送	SSP1 发送
	0x3	-	SGPIO15
6	0x0	-	定时器 2 匹配 1
	0x1	-	USART2 接收
	0x2	SSP1 接收	SSP1 接收
	0x3	-	SGPIO14
7	0x0	-	定时器 3 匹配 0
	0x1	-	USART3 发送
	0x2	-	SCT DMA 请求 0
	0x3	保留	VADC 写
8	0x0	-	定时器 3 匹配 1
	0x1	-	USART3 接收
	0x2	-	SCT DMA 请求 1
	0x3	保留	VADC 读
9	0x0	SSP0 接收	SSP0 接收
	0x1	-	I2S0 DMA 请求 0
	0x2	-	SCT DMA 请求 1
	0x3	保留	保留
10	0x0	SSP0 发送	SSP0 发送
	0x1	-	I2S0 DMA 请求 1
	0x2	-	SCT 匹配 0
	0x3	保留	保留

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 260. 外围设备至 DMA 控制器的连接以及匹配流控制信号 （续）

外围设备编号	DMA 复用选项（参见表 41）	SREQ	BREQ
11	0x0	SSP1 接收	SSP1 接收
	0x1	-	SGPIO14
	0x2	-	USART0 发送
	0x3	保留	保留
12	0x0	SSP1 发送	SSP1 发送
	0x1	-	SGPIO15
	0x2	-	USART0 接收
	0x3	保留	保留
13	0x0	-	ADC0
	0x1	保留	保留
	0x2	SSP1 接收	SSP1 接收
	0x3	-	USART3 接收
14	0x0	-	ADC1
	0x1	保留	保留
	0x2	SSP1 发送	SSP1 发送
	0x3	-	USART3 发送
15	0x0	-	DAC
	0x1	-	SCT 匹配 3
	0x2	-	SGPIO15
	0x3	-	定时器 3 匹配 0

除表 260 中列出的外设之外，GPIO、WWDT 和定时器还可以由 GPDMA 作为无流控制的存储器到存储器事务来访问。

19.5.1 DMA 请求信号

外围设备利用 DMA 请求信号进行数据传输请求。 DMA 请求信号表明需要的是一个单次数据传输还是连发数据传输，以及该传输是否是数据包中的最后一次。可用的 DMA 请求信号有：

BREQ[15:0] — 连发请求信号。这些信号可以让已编程连发数量的数据传输。

SREQ[15:0] — 单一传输请求信号。这些信号可以让单次数据传输。 DMA 控制器实现与外围设备之间的单次传输。

LBREQ[15:0] — 最后一个连发请求信号。

LSREQ[15:0] — 最后一个单次传输请求信号。

请注意，大多数外围设备不支持所有请求类型。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

19.5.2 DMA 响应信号

DMA 响应信号指示 DMA 请求信号启动的传输是否已经结束。响应信号也可以用来指示一个完整的数据包是否已经完成传输。 DMA 控制器的响应信号有：

CLR[15:0] — DMA 清除或应答信号。 DMA 控制器利用 CLR 信号来响应外围设备的 DMA 请求。

TC[15:0] — DMA 终结计数信号。 DMA 控制器使用 TC 信号指示外围设备 DMA 传输已经结束。

19.6 寄存器描述

DMA 控制器支持 8 个通道。每个通道都有通道操作所对应的寄存器。其他寄存器控制源外围设备与 DMA 控制器的关联。还有全局 DMA 控制和状态寄存器。

表 261. 寄存器简介：GPDMA （基址 0x4000 2000）

名称	访问类型	地址偏移	描述	复位值	参考
通用寄存器					
INTSTAT	RO	0x000	DMA 中断状态寄存器	0x0000 0000	表 262
INTTCSTAT	RO	0x004	DMA 中断终结计数请求状态寄存器	0x0000 0000	表 263
INTTCCLEAR	WO	0x008	DMA 中断终结计数请求清除寄存器	-	表 264
INTERRSTAT	RO	0x00C	DMA 中断错误状态寄存器	0x0000 0000	表 265
INTERRCLR	WO	0x010	DMA 中断错误清除寄存器	-	表 266
RAWINTTCSTAT	RO	0x014	DMA 原始中断终结计数状态寄存器	0x0000 0000	表 267
RAWINTERR STAT	RO	0x018	DMA 原始错误中断状态寄存器	0x0000 0000	表 268
ENBLDCHNS	RO	0x01C	DMA 使能通道寄存器	0x0000 0000	表 269
SOFTBREQ	R/W	0x020	DMA 软件连发传送请求寄存器	0x0000 0000	表 270
SOFTSREQ	R/W	0x024	DMA 软件单次请求寄存器	0x0000 0000	表 271
SOFTLBREQ	R/W	0x028	DMA 软件最后一个连发请求寄存器	0x0000 0000	表 272
SOFTLSREQ	R/W	0x02C	DMA 软件最后一个单次请求寄存器	0x0000 0000	表 273
CONFIG	R/W	0x030	DMA 配置寄存器	0x0000 0000	表 274
SYNC	R/W	0x034	DMA 同步寄存器	0x0000 0000	表 275
通道 0 寄存器					
C0SRCADDR	R/W	0x100	DMA 通道 0 源地址寄存器	0x0000 0000	表 276
C0DESTADDR	R/W	0x104	DMA 通道 0 目标地址寄存器	0x0000 0000	表 277
C0LLI	R/W	0x108	DMA 通道 0 链接列表项寄存器	0x0000 0000	表 278
C0CONTROL	R/W	0x10C	DMA 通道 0 控制寄存器	0x0000 0000	表 279
C0CONFIG	R/W	0x110	DMA 通道 0 配置寄存器	0x0000 0000[1]	表 280
通道 1 寄存器					
C1SRCADDR	R/W	0x120	DMA 通道 1 源地址寄存器	0x0000 0000	表 276
C1DESTADDR	R/W	0x124	DMA 通道 1 目标地址寄存器	0x0000 0000	表 277
C1LLI	R/W	0x128	DMA 通道 1 链接列表项寄存器	0x0000 0000	表 278
C1CONTROL	R/W	0x12C	DMA 通道 1 控制寄存器	0x0000 0000	表 279

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 261. 寄存器简介：GPDMA （基址 0x4000 2000） *（续）*

名称	访问类型	地址偏移	描述	复位值	参考
C1CONFIG	R/W	0x130	DMA 通道 1 配置寄存器	0x0000 0000[1]	表 280
通道 2 寄存器					
C2SRCADDR	R/W	0x140	DMA 通道 2 源地址寄存器	0x0000 0000	表 276
C2DESTADDR	R/W	0x144	DMA 通道 2 目标地址寄存器	0x0000 0000	表 277
C2LLI	R/W	0x148	DMA 通道 2 链接列表项寄存器	0x0000 0000	表 278
C2CONTROL	R/W	0x14C	DMA 通道 2 控制寄存器	0x0000 0000	表 279
C2CONFIG	R/W	0x150	DMA 通道 2 配置寄存器	0x0000 0000[1]	表 280
通道 3 寄存器					
C3SRCADDR	R/W	0x160	DMA 通道 3 源地址寄存器	0x0000 0000	表 276
C3DESTADDR	R/W	0x164	DMA 通道 3 目标地址寄存器	0x0000 0000	表 277
C3LLI	R/W	0x168	DMA 通道 3 链接列表项寄存器	0x0000 0000	表 278
C3CONTROL	R/W	0x16C	DMA 通道 3 控制寄存器	0x0000 0000	表 279
C3CONFIG	R/W	0x170	DMA 通道 3 配置寄存器	0x0000 0000[1]	表 280
通道 4 寄存器					
C4SRCADDR	R/W	0x180	DMA 通道 4 源地址寄存器	0x0000 0000	表 276
C4DESTADDR	R/W	0x184	DMA 通道 4 目标地址寄存器	0x0000 0000	表 277
C4LLI	R/W	0x188	DMA 通道 4 链接列表项寄存器	0x0000 0000	表 278
C4CONTROL	R/W	0x18C	DMA 通道 4 控制寄存器	0x0000 0000	表 279
C4CONFIG	R/W	0x190	DMA 通道 4 配置寄存器	0x0000 0000[1]	表 280
通道 5 寄存器					
C5SRCADDR	R/W	0x1A0	DMA 通道 5 源地址寄存器	0x0000 0000	表 276
C5DESTADDR	R/W	0x1A4	DMA 通道 5 目标地址寄存器	0x0000 0000	表 277
C5LLI	R/W	0x1A8	DMA 通道 5 链接列表项寄存器	0x0000 0000	表 278
C5CONTROL	R/W	0x1AC	DMA 通道 5 控制寄存器	0x0000 0000	表 279
C5CONFIG	R/W	0x1B0	DMA 通道 5 配置寄存器	0x0000 0000[1]	表 280
通道 6 寄存器					
C6SRCADDR	R/W	0x1C0	DMA 通道 6 源地址寄存器	0x0000 0000	表 276
C6DESTADDR	R/W	0x1C4	DMA 通道 6 目标地址寄存器	0x0000 0000	表 277
C6LLI	R/W	0x1C8	DMA 通道 6 链接列表项寄存器	0x0000 0000	表 278
C6CONTROL	R/W	01CC	DMA 通道 6 控制寄存器	0x0000 0000	表 279
C6CONFIG	R/W	0x1D0	DMA 通道 6 配置寄存器	0x0000 0000[1]	表 280
通道 7 寄存器					
C7SRCADDR	R/W	0x1E0	DMA 通道 7 源地址寄存器	0x0000 0000	表 276
C7DESTADDR	R/W	0x1E4	DMA 通道 7 目标地址寄存器	0x0000 0000	表 277

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 261. 寄存器简介：GPDMA （基址 0x4000 2000） *（续）*

名称	访问类型	地址偏移	描述	复位值	参考
C7LLI	R/W	0x1E8	DMA 通道 7 链接列表项寄存器	0x0000 0000	表 278
C7CONTROL	R/W	0x1EC	DMA 通道 7 控制寄存器	0x0000 0000	表 279
C7CONFIG	R/W	0x1F0	DMA 通道 7 配置寄存器	0x0000 0000[1]	表 280

[1] 该寄存器的位 17 是一个只读状态标志位。

19.6.1 DMA 中断状态寄存器

IntStat 是一个只读寄存器，显示掩码后的中断状态。高电平位表明某个特定的 DMA 通道中断请求有效。中断请求可以由错误产生，也可以由终结计数中断请求产生。

表 262. DMA 中断状态寄存器 （INTSTAT，地址 0x4000 2000）位描述

位	符号	描述	复位值	访问类型
7:0	INTSTAT	掩码后 DMA 通道中断的状态。每个位代表一个通道： 0 - 相应的通道没有有效的中断请求。 1 - 相应的通道有一个有效的中断请求。	0x00	RO
31:8	-	保留。未定义读取。	-	-

19.6.2 DMA 中断终结计数请求状态寄存器

INTTCSTAT 是一个只读寄存器，指示掩码后终结计数的状态。

表 263. DMA 中断终结计数请求状态寄存器 （INTTCCLEAR，地址 0x4000 2004）位描述

位	符号	描述	复位值	访问类型
7:0	INTTCSTAT	DMA 通道终结计数中断请求的状态。每个位代表一个通道： 0 - 相应的通道没有有效的终结计数中断请求。 1 - 相应的通道有一个有效的终结计数中断请求。	0x00	RO
31:8	-	保留。未定义读取。	-	-

19.6.3 DMA 中断终结计数请求清除寄存器

INTTCCLEAR 是一个只写寄存器，清除一个或多个终结计数中断请求。写该寄存器时，每个设置为高电平的数据位可以清除状态寄存器(IntTCStat)中的相应位。低电平数据位无效。

表 264. DMA 中断终结计数请求清除寄存器 （INTTCCLEAR，地址 0x4000 2008）位描述

位	符号	描述	复位值	访问类型
7:0	INTTCCLEAR	允许清除 DMA 通道的终结计数中断请求 (IntTCStat)。每个位代表一个通道： 0 - 写 0 无效。 1 - 清除相应的通道终结计数中断。	0x00	WO
31:8	-	保留。未定义读取。写入保留位为 0。	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

19.6.4 DMA 中断错误状态寄存器

INTERRSTAT 是一个只读寄存器，指示掩码后错误请求的状态。

表 265. DMA 中断错误状态寄存器 （INTERRSTAT，地址 0x4000 200C）位描述

位	符号	描述	复位值	访问类型
7:0	INTERRSTAT	DMA 通道的中断错误状态。每个位代表一个通道： 0 - 相应的通道没有有效的错误中断请求。 1 - 相应的通道有一个有效的错误中断请求。	0x00	RO
31:8	-	保留。未定义读取。	-	-

19.6.5 DMA 中断错误清除寄存器

INTERRCLR 是一个只写寄存器，清除错误中断请求。当写该寄存器时，每个高电平数据位可以清除状态寄存器中的相应位。低电平数据位对寄存器中的相应位无效。

表 266. DMA 中断错误清除寄存器 （INTERRCLR，地址 0x4000 2010）位描述

位	符号	描述	复位值	访问类型
7:0	INTERRCLR	写 1 会清除 DMA 通道的错误中断请求 （IntErrStat）。每个位代表一个通道： 0 – 写 0 无效。 1 – 清除相应的通道错误中断。	0x00	WO
31:8	-	保留。未定义读取。写入保留位为 0。	-	-

19.6.6 DMA 原始中断终结计数状态寄存器

RAWINTTCSTAT 是一个只读寄存器，指示在掩码前哪个 DMA 通道正在请求传输结束（终结计数中断）。（注：掩码后， IntTCStat 寄存器仍包含相同的信息。）高电平位表明在掩码前终结计数中断请求有效。

表 267. DMA 原始中断终结计数状态寄存器 （RAWINTTCSTAT，地址 0x4000 2014）位描述

位	符号	描述	复位值	访问类型
7:0	RAWINTTCSTAT	掩码前 DMA 通道的终结计数中断的状态。每个位代表一个通道： 0 - 相应的通道没有有效的终结计数中断请求。 1 - 相应的通道有一个有效的终结计数中断请求。	0x00	RO
31:8	-	保留。未定义读取。	-	-

19.6.7 DMA 原始错误中断状态寄存器

RAWINTERRSTAT 是一个只读寄存器，指示在掩码前哪个 DMA 通道正在请求错误中断。（注：掩码后，IntErrStat 寄存器仍包含相同的信息。）高电平位表明在掩码前错误中断请求有效。

表 268. DMA 原始错误中断状态寄存器 （RAWINTERRSTAT，地址 0x4000 2018）位描述

位	符号	描述	复位值	访问类型
7:0	RAWINTERRSTAT	掩码前 DMA 通道的错误中断的状态。每个位代表一个通道： 0 - 相应的通道没有有效的错误中断请求。 1 - 相应的通道有一个有效的错误中断请求。	0x00	RO
31:8	-	保留。未定义读取。	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

19.6.8 DMA 使能通道寄存器

ENBLDCHNS 是一个只读寄存器，指示哪个 DMA 通道被使能 （正如 CCONFIG 寄存器的使能位所指示的一样）。高电平位表明DMA通道被使能。DMA传输结束时相应的位被清除。

表 269. DMA 使能通道寄存器 （ENBLDCHNS，地址 0x4000 201C）位描述

位	符号	描述	复位值	访问类型
7:0	ENABLEDCHANNELS	DMA 通道的使能状态。每个位代表一个通道： 0 - DMA 通道被禁用。 1 - DMA 通道被使能。	0x00	RO
31:8	-	保留。未定义读取。	-	-

19.6.9 DMA 软件连发传送请求寄存器

SOFTBREQ 是一个可读 / 写寄存器，使软件可以产生 DMA 连发请求。每个源的 DMA 请求可以通过向相应的寄存器位写 1 来产生。传输结束时，寄存器位被清除。读取寄存器会指示哪个源正在请求 DMA 连发传输。可由外围设备或软件请求寄存器产生请求。相关传输结束时，每个位被清除。

表 270. DMA 软件连发传送请求寄存器 （SOFTBREQ，地址 0x4000 2020）位描述

位	符号	描述	复位值	访问类型
15:0	SOFTBREQ	16 个可能的源的软件连发请求标志。每个位代表一个 DMA 请求线路或外设功能 （有关外设硬件到 DMA 控制器的连接，请参见表 260）： 0 - 写 0 无效。 1 - 写 1 产生相应请求线的 DMA 连发请求。	0x00	R/W
31:16	-	保留。未定义读取。写入保留位为 0。	-	-

注：建议不要同时使用软件和硬件外设请求。

19.6.10 DMA 软件单次请求寄存器

SOFTSREQ 是一个可读 / 写寄存器，使能软件产生 DMA 单次传输请求。每个源的 DMA 请求可以通过向相应的寄存器位写 1 来产生。传输结束时，寄存器位被清除。读取寄存器会指示哪个源正在请求单发 DMA 传输。可由外围设备或软件请求寄存器产生请求。

表 271. DMA 软件单次请求寄存器 （SOFTSREQ，地址 0x4000 2024）位描述

位	符号	描述	复位值	访问类型
15:0	SOFTSREQ	16 个可能的源的软件单次传输请求标志。每个位代表一个 DMA 请求线或外围设备功能： 0 - 写 0 无效。 1 - 写 1 产生相应请求线的 DMA 单次传输请求。	0x00	R/W
31:16	-	保留。未定义读取。写入保留位为 0。	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

19.6.11 DMA 软件最后一个连发请求寄存器

SOFTLBREQ 是一个可读 / 写寄存器，使能软件产生 DMA 最后一个连发请求。每个源的 DMA 请求可以通过向相应的寄存器位写 1 来产生。传输结束时，寄存器位被清除。读取寄存器会指示哪个源正在请求最后一个连发 DMA 传输。可由外围设备或软件请求寄存器产生请求。

表 272. DMA 软件最后一个连发请求寄存器 （SOFTBREQ，地址 0x4000 2028）位描述

位	符号	描述	复位值	访问类型
15:0	SOFTLBREQ	16 个可能的源的软件最后一个连发请求标志。每个位代表一个 DMA 请求线或外围设备功能： 0 - 写 0 无效。 1 - 写 1 产生相应请求线的 DMA 最后一个连发请求。	0x00	R/W
31:16	-	保留。未定义读取。写入保留位为 0。	-	-

19.6.12 DMA 软件最后一个单次请求寄存器

SOFTLSREQ 是一个可读 / 写寄存器，使能软件产生 DMA 最后一个单次请求。每个源的 DMA 请求可以通过向相应的寄存器位写 1 来产生。传输结束时，寄存器位被清除。读取寄存器会指示哪个源正在请求最后一个单发 DMA 传输。可由外围设备或软件请求寄存器产生请求。

表 273. DMA 软件最后一个单次请求寄存器 （SOFTLSREQ，地址 0x4000 202C）位描述

位	符号	描述	复位值	访问类型
15:0	SOFTLSREQ	16 个可能的源的软件最后一个单次传输请求标志。每个位代表一个 DMA 请求线或外围设备功能： 0 - 写 0 无效。 1 - 写 1 产生相应请求线的 DMA 最后一个单次传输请求。	0x00	R/W
31:16	-	保留。未定义读取。写入保留位为 0。	-	-

19.6.13 DMA 配置寄存器

CONFIG 是一个可读 / 写寄存器，配置 DMA 控制器的操作。AHB 主机接口的字节顺序通过写该寄存器的 M 位来改变。复位时， AHB 主机接口被设置成小端模式。

表 274. DMA 配置寄存器 （CONFIG，地址 0x4000 2030）位描述

位	符号	值	描述	复位值	访问类型
0	E		DMA 控制器使能：	0x00	R/W
		0	禁用 （默认）。禁用 DMA 控制器来降低功耗。		
		1	使能		
1	M0		AHB 主机 0 字节顺序配置：	0x00	R/W
		0	小端模式 （默认）。		
		1	大端模式。		
2	M1		AHB 主机 1 字节顺序配置：	0x00	R/W
		0	小端模式 （默认）。		
		1	大端模式。		
31:3	-		保留。未定义读取。写入保留位为 0。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

19.6.14 DMA 同步寄存器

Sync 是一个可读 / 写的寄存器，使能或禁用 DMA 请求信号的同步逻辑。DMA 请求信号由 BREQ[15:0]、SREQ[15:0]、LBREQ[15:0] 和 LSREQ[15:0] 组成。将某个位设为 0，可以使能特定组 DMA 请求的同步逻辑。将某个位设为 1，可以禁用特定组 DMA 请求的同步逻辑。该寄存器复位到 0，同步逻辑使能。

表 275. DMA 同步寄存器 （SYNC，地址 0x4000 2034）位描述

位	符号	描述	复位值	访问类型
15:0	DMACSYNC	控制 DMA 请求信号的同步逻辑。每个位代表一组 DMA 请求线 （如前文所述）： 0 - 对应的 DMA 请求信号的同步逻辑被禁用。 1 - 对应的响应请求线信号的同步逻辑被使能。	0x00	R/W
31:16	-	保留。未定义读取。写入保留位为 0。	-	-

19.6.15 DMA 通道寄存器

通道寄存器用于编程 8 个 DMA 通道。这些寄存器的组成如下：

- 8 个 CSRCADDR 寄存器。
- 8 个 CDESTADDR 寄存器。
- 8 个 CLLI 寄存器。
- 8 个 CCONTROL 寄存器。
- 8 个 CCONFIG 寄存器。

执行分散 / 聚集 DMA 时，其中前四个寄存器会自动更新。

19.6.16 DMA 通道源地址寄存器

8 个可读 / 写的 CSRCADDR （C0SRCADDR 至 C7SRCADDR）寄存器包含当前要传输数据的源地址 （字节对齐） 。每个寄存器通过软件进行直接编程后，相应的通道才会使能。DMA 通道使能后，该寄存器会进行更新：

- 随源地址增加而更新。
- 在一个完整的数据包传输结束后紧跟在链接列表之后更新。

通道有效时，读该寄存器不能提供有用的信息。这是因为当软件处理了读回的值后，地址可能已进行更新。只有在停止通道后才读该寄存器，寄存器会显示读取的最后一项的源地址。

注：源和目标地址必须与源和目标宽度对齐。

表 276. DMA 通道源地址寄存器 （CSRCADDR， 0x4000 2100 (C0SRCADDR) 至 0x4000 21E0 (C7SRCADDR)）位描述

位	符号	描述	复位值	访问类型
31:0	SRCADDR	DMA 源地址。读该寄存器会返回当前的源地址。	0x0000 0000	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

19.6.17 DMA 通道目标地址寄存器

8 个可读 / 写的 CDESTADDR （C0DESTADDR 至 C7DESTADDR）寄存器包含当前要传输数据的目标地址（字节对齐）。每个寄存器通过软件进行直接编程后，通道才会使能。当 DMA 通道使能时，该寄存器会随着目标地址的增加而更新，以及在一个完整的数据包传输结束后紧跟在链接列表之后更新。通道有效时，读该寄存器不能提供有用的信息。这是因为当软件处理了读回的值后，地址可能已进行更新。只有在停止通道后才读该寄存器，寄存器显示读取的最后一项的目标地址。

表 277. DMA通道目标地址寄存器 （CDESTADDR，0x4000 2104 (C0DESTADDR)至0x4000 21E4 (C7DESTADDR)）位描述

位	符号	描述	复位值	访问类型
31:0	DESTADDR	DMA 目标地址。读该寄存器会返回当前的目标地址。	0x0000 0000	R/W

19.6.18 DMA 通道链接列表项寄存器

8 个可读 / 写的 CLLI 寄存器 （C0LLI 至 C7LLI）包含下一个链接列表项 (LLI) 的字对齐地址。如果 LLI 为 0，则当前的 LLI 是链的最后一项；当与之相关的所有 DMA 传输结束后，DMA 通道被禁用。 DMA 通道使能时，编程该寄存器可能会有不可预知的副作用。

表 278. DMA 通道链接列表项寄存器 （CLLI， 0x4000 2108 (C0LLI) 至 0x4000 21E8 (C7LLI)）位描述

位	符号	值	描述	复位值	访问类型
0	LM		加载下一个 LLI 的 AHB 主机选择：	0	R/W
		0	AHB 主机 0。		
		1	AHB 主机 1。		
1	R		保留，必须写入为 0，读取时掩码。	0	R/W
31:2	LLI		链接列表项。下一个 LLI 地址的位 [31:2]。地址位 [1:0] 为 0。	0x0000 0000	R/W

19.6.19 DMA 通道控制寄存器

8 个可读 / 写的 CCONTROL 寄存器 （C0CONTROL 至 C7CONTROL）包含传输长度、连发大小和传输宽度等 DMA 通道的控制信息。每个寄存器通过软件进行直接编程后， DMA 通道才会使能。通道使能时，该寄存器会在一个完整的数据包传输结束后紧跟在链接列表之后更新。通道有效时，读该寄存器不能提供有用的信息。这是因为当软件处理了读回的值后，通道可能已进行更新。只有在停止通道后，才读该寄存器。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 279. DMA 通道控制寄存器 （CCONTROL， 0x4000 210C (C0CONTROL) 至 0x4000 21EC (C7CONTROL)）位描述

位	符号	值	描述	复位值	访问类型
11:0	TRANSFERSIZE		传输长度，即传输的数量。当 DMA 控制器为流控制器时，写该字段可以设置传输的大小。传输长度值必须在使能通道前设置。数据传输结束时，传输长度进行更新。 读该字段可获得在目标总线上已完成的传输数量。通道有效时，读该寄存器不能提供有用信息。这是因为当软件处理了读回的值后，该通道可能已更新。应只有在通道先使能后禁用的情况下才使用。 如果 DMA 控制器非流控制器，则传输长度值不可用。	0x0	R/W
14:12	SBSIZE		源连发大小。表示形成源连发的传输数量。该值必须设为源外设的连发大小，如果源是存储器，该值就设为存储器边界的大小（参见图 3）。当源外设的 BREQ 信号有效时，连发大小为传输的数据量。	0x0	R/W
		0x0	源连发大小 = 1		
		0x1	源连发大小 = 4		
		0x2	源连发大小 = 8		
		0x3	源连发大小 = 16		
		0x4	源连发大小 = 32		
		0x5	源连发大小 = 64		
		0x6	源连发大小 = 128		
		0x7	源连发大小 = 256		
17:15	DBSIZE		目标连发大小。表示形成目标连发传输请求的传输数量。该值必须设为目标外围设备的连发大小，如果目标是存储器，该值就设为存储器边界大小。当目标外围设备的 BREQ 信号有效时，连发大小为传输的数据量。	0x0	R/W
		0x0	目标连发大小 = 1		
		0x1	目标连发大小 = 4		
		0x2	目标连发大小 = 8		
		0x3	目标连发大小 = 16		
		0x4	目标连发大小 = 32		
		0x5	目标连发大小 = 64		
		0x6	目标连发大小 = 128		
		0x7	目标连发大小 = 256		
20:18	SWIDTH		源传输宽度。不允许传输宽度大于 AHB 主机总线宽度的数据。源和目标宽度可以不同。硬件会根据需要自动打包和拆分数据。0x3 至 0x7 – 保留。	0x0	R/W
		0x0	字节 （8 位）		
		0x1	半字 （16 位）		
		0x2	字 （32 位）		
23:21	DWIDTH		目标传输宽度。不允许传输宽度大于 AHB 主机总线宽度的数据。源和目标宽度可以不同。硬件会根据需要自动打包和拆分数据。0x3 至 0x7 – 保留。	0x0	R/W
		0x0	字节 （8 位）		
		0x1	半字 （16 位）		
		0x2	字 （32 位）		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 279. DMA 通道控制寄存器（CCONTROL，0x4000 210C (C0CONTROL) 至 0x4000 21EC (C7CONTROL)）位描述（续）

位	符号	值	描述	复位值	访问类型
24	S		源 AHB 主机选择：	0	R/W
		0	源传输选择 AHB 主机 0。		
		1	源传输选择 AHB 主机 1。		
25	D		目标 AHB 主机选择： **注：**仅主机 1 可以访问外设。主机 0 仅可访问存储器。	0	R/W
		0	目标传输所选的 AHB 主机 0。		
		1	目标传输所选的 AHB 主机 1。		
26	SI		源自增：	0	R/W
		0	每次传输后源地址不自增。		
		1	每次传输后源地址自增。		
27	DI		目标自增：	0	R/W
		0	每次传输后目标地址不自增。		
		1	每次传输后目标地址自增。		
28	PROT1		指明是在用户模式下还是特权模式下访问：	0	R/W
		0	用户模式下访问		
		1	特权模式下访问。		
29	PROT2		指明访问是否可以缓冲：	0	R/W
		0	访问不可以缓冲。		
		1	访问可以缓冲。		
30	PROT3		指明访问是否可以缓存：	0	R/W
		0	访问不可以缓存。		
		1	访问可以缓存。		
31	I		终结计数中断使能位。	0	R/W
		0	终结计数中断禁用。		
		1	终结计数中断使能。		

19.6.19.1 保护和访问信息

传输发生时，AHB 访问信息会提供给源和目标外围设备。通过编程 DMA 通道（CCONTROL 寄存器的 Prot 位以及 CCONFIG 寄存器的 Lock 位），提供传输信息。这些位通过软件编程。必要时，外围设备可以使用这些信息。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

19.6.20 通道配置寄存器

8 个 CCONFIG 寄存器 （C0CONFIG 至 C7CONFIG）为可读 / 写，但位 [17] 除外，为只读。使用这些寄存器配置 DMA 通道。请求新的 LLI 时，寄存器不会更新。

表 280. DMA 通道配置寄存器 （CCONFIG， 0x4000 2110 (C0CONFIG) 至 0x4000 21F0 (C7CONFIG)）位描述

位	符号	值	描述	复位值	访问类型
0	E		通道使能。读该位可以获得当前通道的使能状态： 通道使能位状态也可通过读取 ENBLDCHNS 寄存器找到。 清除使能位可以禁用通道。这可以让当前 AHB 传输（如果有一个传输正在进行）结束并禁用通道。相应通道 FIFO 中的所有数据都会丢失。如果通过置位通道使能位来重启通道，就会产生不可预知的影响，必须对通道重新进行完全初始化。 最后一个 LLI 到达时、 DMA 传输结束时，或碰到一个通道错误时，通道也会被禁用，并且清除通道使能位。 如果某个通道必须被禁用且不丢失 FIFO 中的数据，则 Halt 位必须置位，以忽略后续的 DMA 请求。然后查询 Active 位直至其为 0，表明 FIFO 中没有留下数据。最后，清除通道使能位。	0	R/W
		0	通道禁用。		
		1	通道使能。		
5:1	SRC 外围设备		源外围设备。该值选择 DMA 源请求外围设备。如果传输源是存储器，则忽略该字段。参见表 260 以了解详细信息。		R/W
		0x0	SPIFI/SCT 匹配 3/SGPIO14/ 定时器 3 匹配 1		
		0x1	定时器 0 匹配 0/USART0 发送		
		0x2	定时器 0 匹配 1/USART0 接收		
		0x3	定时器 1 匹配 0/UART1 发送 / I2S1 DMA 请求 0/SSP1 发送		
		0x4	定时器 1 匹配 1/UART1 接收 / I2S1 DMA 请求 1/SSP1 接收		
		0x5	定时器 2 匹配 0/USART2 发送 /SSP1 发送 /SGPIO15		
		‘	定时器 2 匹配 1/USART2 接收 /SSP1 接收 /SGPIO14		
		0x7	定时器 3 匹配 0/UART3 发送 /SCT 匹配 0/VADC 写入		
		0x8	定时器 3 匹配 1/UART3 接收 /SCT 匹配 1/VADC 读取		
		0x9	SSP0 接收 /I2S0 DMA 请求 0/SCT 匹配 1		
		0xA	SSP0 发送 /I2S DMA 请求 1/SCT 匹配 0		
		0xB	SSP1 接收 /SGPIO14/USART0 发送		
		0xC	SSP1 发送 /SGPIO15/USART0 接收		
		0xD	ADC0/SSP1 接收 /USART3 接收		
		0xE	ADC1/SSP1 发送 /USART3 发送		
		0xF	DAC/SCT 匹配 3/SGPIO15/ 定时器 3 匹配 0		

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 280. DMA 通道配置寄存器 （CCONFIG， 0x4000 2110 (C0CONFIG) 至 0x4000 21F0 (C7CONFIG)）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
10:6	DEST 外围设备		目标外围设备。该值选择 DMA 目标请求外围设备。如果传输目标是存储器，则忽略该字段。参见表 260 以了解详细信息。		R/W
		0x0	SPIFI/SCT 匹配 3/SGPIO14/ 定时器 3 匹配 1		
		0x1	定时器 0 匹配 0/USART0 发送		
		0x2	定时器 0 匹配 1/USART0 接收		
		0x3	定时器 1 匹配 0/UART1 发送 / I2S1 DMA 请求 0/SSP1 发送		
		0x4	定时器 1 匹配 1/UART1 接收 / I2S1 DMA 请求 1/SSP1 接收		
		0x5	定时器 2 匹配 0/USART2 发送 /SSP1 发送 /SGPIO15		
		0x6	定时器 2 匹配 1/USART2 接收 /SSP1 接收 /SGPIO14		
		0x7	定时器 3 匹配 0/UART3 发送 /SCT 匹配 0/VADC 写入		
		0x8	定时器 3 匹配 1/UART3 接收 /SCT 匹配 1/VADC 读取		
		0x9	SSP0 接收 /I2S0 DMA 请求 0/SCT 匹配 1		
		0xA	SSP0 发送 /I2S DMA 请求 1/SCT 匹配 0		
		0xB	SSP1 接收 /SGPIO14/USART0 发送		
		0xC	SSP1 发送 /SGPIO15/USART0 接收		
		0xD	ADC0/SSP1 接收 /USART3 接收		
		0xE	ADC1/SSP1 发送 /USART3 发送		
		0xF	DAC/SCT 匹配 3/SGPIO15/ 定时器 3 匹配 0		
13:11	FLOWCNTRL		流控制和传输类型。该值指明流控制器和传输类型。流控制器可以是 DMA 控制器、源外围设备或目标外围设备。 传输类型可以是存储器到存储器、存储器到外围设备、外围设备到存储器或外围设备到外围设备。 有关该字段的编码，请参见表 281。		R/W
		0x0	存储器到存储器 （DMA 控制）		
		0x1	存储器到外围设备 （DMA 控制）		
		0x2	外围设备到存储器 （DMA 控制）		
		0x3	源外围设备到目标外围设备 （DMA 控制）		
		0x4	源外围设备到目标外围设备 （目标控制）		
		0x5	存储器到外围设备 （外围设备控制）		
		0x6	外围设备到存储器 （外围设备控制）		
		0x7	源外围设备到目标外围设备 （源控制）		
14	IE		中断错误掩码。清除时，该位会掩码相关通道的错误中断。		R/W
15	ITC		终结计数中断掩码。清除时，该位会掩码相关通道的终结计数中断。		R/W
16	L		锁定。置位时，该位会使能锁定的传输。		R/W
17	A		活动： 0 = 通道 FIFO 中没有数据。 1 = 通道 FIFO 中有数据。 该值可以和 Halt 以及通道使能位一起使用，彻底禁用一个 DMA 通道。这是一个只读位。		RO

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 280. DMA 通道配置寄存器 （CCONFIG， 0x4000 2110 (C0CONFIG) 至 0x4000 21F0 (C7CONFIG)）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
18	H		Halt： 0 = 使能 DMA 请求。 1 = 忽略后续的源 DMA 请求。 通道 FIFO 的内容被输出。 该值可以和 Active 以及通道使能位一起使用，彻底禁用一个 DMA 通道。		R/W
		0	使能 DMA 请求。		
		1	忽略后续的源 DMA 请求。		
31:19	-		保留，不修改，读取时掩码。		-

19.6.20.1 锁定控制

向 CCONFIG 寄存器的位 16 写 1 可以设置锁定控制。连发传输发生时， AHB 仲裁器不会在连发传输过程中剥夺主机的授权，直至锁定被取消。在单次连发传输 （例如：一个长源读取连发传输或一个长目标输出连发传输）过程中， DMA 控制器会被锁定。一般情况下，DMA 控制器不会连续锁定在源读取连发传输和目标输出连发传输中。

也存在 DMA 控制器先后在源传输和目标传输中被锁定的情况。当 DMA 控制器的内部条件允许时，才可能在源读取后继续执行目标输出。

19.6.20.2 流控制和传输类型

表 281 列出了表 280 所确定的三种流控制和传输类型位的位值。

表 281. 流控制和传输类型位

位值	传输类型	控制器
000	存储器到存储器	DMA
001	存储器到外围设备	DMA
010	外围设备到存储器	DMA
011	源外围设备到目标外围设备	DMA
100	源外围设备到目标外围设备	目标外围设备
101	存储器到外围设备	外设
110	外围设备到存储器	外设
111	源外围设备到目标外围设备	源外围设备

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

19.7 功能说明

19.7.1 DMA 控制器功能描述

DMA 控制器允许外围设备到存储器、存储器到外围设备、外围设备到外围设备和存储器到存储器之间的传输。每个 DMA 流都可以为单个源和目标提供单向串行 DMA 传输。例如，一个双向端口就需要一个发送流和一个接收流。源和目标区既可以是存储区，也可以是外设，并且可通过 AHB 主机进行访问。图 38 显示了 DMA 控制器的框图。

在接下来的小节中，我们将对 DMA 控制器的功能进行详细介绍。

19.7.1.1 AHB 从机接口

通过 AHB 从机接口对 DMA 控制器的所有传输都是 32 位宽度。不支持 8 位和 16 位访问，这会导致异常出现。

19.7.1.2 控制逻辑和寄存器组

该寄存器模块存储通过 AHB 接口写入或读取的数据。

19.7.1.3 DMA 请求和响应接口

有关 DMA 请求和响应接口的信息，请参见 DMA 接口的描述。

19.7.1.4 通道逻辑和通道寄存器组

通道逻辑和通道寄存器组包含了每个 DMA 通道所需的寄存器和逻辑。

19.7.1.5 中断请求

中断请求向 ARM 处理器产生中断。

19.7.1.6 AHB 主机接口

DMA 控制器包含 2 个 AHB 主机接口。每个 AHB 主机都能够处理下列类型的 AHB 传输：

- 拆分、重试，以及来自从机的报错响应。如果一个外围设备执行一次分离或重试传输，DMA 控制器停止并等待传输结束。
- 锁定每个流的源和目标传输。
- 为每个流的传输设置保护位。

19.7.1.6.1 总线和传输宽度

AHB 总线的物理宽度是 32 位。源和目标传输的宽度可以不同，也可以相同，还可以比物理总线宽度更窄。 DMA 控制器根据需要打包或拆分数据。

19.7.1.6.2 字节顺序特性

DMA 控制器可处理小端和大端寻址。软件可以分别设置每个 AHB 主机的字节顺序。

在内部，DMA 控制将所有数据当作一个字节流来处理，而不是 16 位或 32 位的数据量。这就意味着，当执行源和目标传输字节顺序不同的混合字节时，可以观察到 32 位数据总线内的字节交换。

注：如果不需要字节交换，则在源和目标地址之间应避免使用不同的字节顺序。表 282 显示了不同的源和目标组合的字节顺序特性。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 282. 字节顺序特性

源字节顺序	目标字节顺序	源宽度	目标宽度	源传输编号 / 字节通道	源数据	目标传输编号 / 字节通道	目标数据
低位优先	低位优先	8	8	1/[7:0]	21	1/[7:0]	21212121
				2/[15:8]	43	2/[15:8]	43434343
				3/[23:16]	65	3/[23:16]	65656565
				4/[31:24]	87	4/[31:24]	87878787
低位优先	低位优先	8	16	1/[7:0]	21	1/[15:0]	43214321
				2/[15:8]	43	2/[31:16]	87658765
				3/[23:16]	65		
				4/[31:24]	87		
低位优先	低位优先	8	32	1/[7:0]	21	1/[31:0]	87654321
				2/[15:8]	43		
				3/[23:16]	65		
				4/[31:24]	87		
低位优先	低位优先	16	8	1/[7:0]	21	1/[7:0]	21212121
				1/[15:8]	43	2/[15:8]	43434343
				2/[23:16]	65	3/[23:16]	65656565
				2/[31:24]	87	4/[31:24]	87878787
低位优先	低位优先	16	16	1/[7:0]	21	1/[15:0]	43214321
				1/[15:8]	43	2/[31:16]	87658765
				2/[23:16]	65		
				2/[31:24]	87		
低位优先	低位优先	16	32	1/[7:0]	21	1/[31:0]	87654321
				1/[15:8]	43		
				2/[23:16]	65		
				2/[31:24]	87		
低位优先	低位优先	32	8	1/[7:0]	21	1/[7:0]	21212121
				1/[15:8]	43	2/[15:8]	43434343
				1/[23:16]	65	3/[23:16]	65656565
				1/[31:24]	87	4/[31:24]	87878787
低位优先	低位优先	32	16	1/[7:0]	21	1/[15:0]	43214321
				1/[15:8]	43	2/[31:16]	87658765
				1/[23:16]	65		
				1/[31:24]	87		
低位优先	低位优先	32	32	1/[7:0]	21	1/[31:0]	87654321
				1/[15:8]	43		
				1/[23:16]	65		
				1/[31:24]	87		
高位优先	高位优先	8	8	1/[31:24]	12	1/[31:24]	12121212
				2/[23:16]	34	2/[23:16]	34343434
				3/[15:8]	56	3/[15:8]	56565656
				4/[7:0]	78	4/[7:0]	78787878

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 282. 字节顺序特性 *（续）*

源字节顺序	目标字节顺序	源宽度	目标宽度	源传输编号 / 字节通道	源数据	目标传输编号 / 字节通道	目标数据
高位优先	高位优先	8	16	1/[31:24]	12	1/[15:0]	12341234
				2/[23:16]	34	2/[31:16]	56785678
				3/[15:8]	56		
				4/[7:0]	78		
高位优先	高位优先	8	32	1/[31:24]	12	1/[31:0]	12345678
				2/[23:16]	34		
				3/[15:8]	56		
				4/[7:0]	78		
高位优先	高位优先	16	8	1/[31:24]	12	1/[31:24]	12121212
				1/[23:16]	34	2/[23:16]	34343434
				2/[15:8]	56	3/[15:8]	56565656
				2/[7:0]	78	4/[7:0]	78787878
高位优先	高位优先	16	16	1/[31:24]	12	1/[15:0]	12341234
				1/[23:16]	34	2/[31:16]	56785678
				2/[15:8]	56		
				2/[7:0]	78		
高位优先	高位优先	16	32	1/[31:24]	12	1/[31:0]	12345678
				1/[23:16]	34		
				2/[15:8]	56		
				2/[7:0]	78		
高位优先	高位优先	32	8	1/[31:24]	12	1/[31:24]	12121212
				1/[23:16]	34	2/[23:16]	34343434
				1/[15:8]	56	3/[15:8]	56565656
				1/[7:0]	78	4/[7:0]	78787878
高位优先	高位优先	32	16	1/[31:24]	12	1/[15:0]	12341234
				1/[23:16]	34	2/[31:16]	56785678
				1/[15:8]	56		
				1/[7:0]	78		
高位优先	高位优先	32	32	1/[31:24]	12	1/[31:0]	12345678
				1/[23:16]	34		
				1/[15:8]	56		
				1/[7:0]	78		

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

19.7.1.6.3 错误条件

DMA 传输过程中的错误是由外围设备直接标记的，在传输过程中，外围设备在 AHB 总线上产生一个错误响应。当前传输结束后，DMA 控制器自动禁用 DMA 流，还可以向 CPU 发送一个错误中断。该错误中断可以被掩码。

19.7.1.7 通道硬件

每个流都受到一个专用的硬件通道支持，包括源和目标控制器，以及 FIFO。与只带有一个硬件通道（该通道由多个 DMA 流共享）的 DMA 控制器相比，这就减少了等待时间并简化了控制逻辑。

19.7.1.8 DMA 请求优先级

DMA 通道的优先级固定。 DMA 通道 0 的优先级最高， DMA 通道 7 的优先级最低。

当 DMA 控制器传输低优先级通道的数据时，高优先级通道变得有效，则先完成低优先级通道指定给主机接口的传输数，然后才切换到传输高优先级通道的数据。最差情况下，高优先级通道需要等待 4 倍字长的时间。

建议：存储器到存储器的传输使用优先级最低的通道。否则，在 DMA 控制器的存储器到存储器传输过程中，其他 AHB 总线主机会受到阻止，而无法访问总线。

19.7.1.9 中断生成

DMA 控制器的各个中断请求通过“OR”功能产生共用的中断输出，并连接到中断控制器。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

19.8 使用 DMA 控制器

19.8.1 编程 DMA 控制器

所有 DMA 控制器的内部寄存器必须使用字读取 （32 位）和字写入来访问。

19.8.1.1 使能 DMA 控制器

如需使能 DMA 控制器，置位 CONFIG 寄存器的使能位。

19.8.1.2 禁用 DMA 控制器

如需禁用 DMA 控制器：

- 读取 ENBLDCHNS 寄存器，确保所有 DMA 通道都已被禁用。如果仍有通道有效，请参见 “禁用 DMA 通道”。
- 向 CONFIG 寄存器的 DMA 使能位写入 0 来禁用 DMA 控制器。

19.8.1.3 使能 DMA 通道

如需使能 DMA 通道，置位相关 DMA 通道配置寄存器中的通道使能位。请注意，通道在使能前必须完全初始化。

19.8.1.4 禁用 DMA 通道

有 3 种方式可以禁用 DMA 通道：

- 直接写入到通道使能位。如果使用该方法，则 FIFO 中所有未处理的数据都会丢失。
- 将有效位和停止位与通道使能位搭配使用。
- 等到传输完成。通道会自动清除。

禁用 DMA 通道且丢失 FIFO 中的数据

清除相关通道配置寄存器中的相关通道使能位。当前 AHB 传输 （如果有一个传输正在进行）会结束，该通道会被禁用。 FIFO 中的所有数据会丢失。

禁用 DMA 通道且不丢失 FIFO 中的数据

- 设置相关通道配置寄存器中的停止位。这会导致忽略之后的任何 DMA 请求。
- 查询相关通道配置寄存器的有效位，直至其为 0。该位表示通道中是否存在要传输的数据。
- 清除相关通道配置寄存器的通道使能位。

19.8.1.5 设置新的 DMA 传输

如需设置新的 DMA 传输：

如果该通道不能留出来供 DMA 传输使用：

1. 读 ENBLDCHNS 控制器寄存器，找出非活动通道。
2. 选择一个具有所需优先级的非活动通道。
3. 编程 DMA 控制器。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

19.8.1.6 中止 DMA 通道

置位相关 DMA 通道配置寄存器的 Halt 位。处理当前源请求。后续的源 DMA 请求均被忽略，直至 Halt 位清除。

19.8.1.7 编程 DMA 通道

1. 选择一个具有所需优先级的空闲 DMA 通道。DMA 通道 0 的优先级最高，DMA 通道 7 的优先级最低。
2. 通过写 IntTCClear 和 INTERRCLEAR 寄存器清除所使用通道上的任一挂起中断。之前的通道操作可能会使中断继续有效。
3. 向 CSRCADDR 寄存器写入源地址。
4. 向 CDESTADDR 寄存器写入目标地址。
5. 向CLLI寄存器写入下一个LLI的地址。如果只传输单个数据包，则必须向该寄存器写0。
6. 向 CCONTROL 寄存器写入控制信息。
7. 向CCONFIG寄存器写入通道配置信息。如果使能位被置位，则DMA通道将自动使能。

19.8.2 流控制

控制数据包长度的外围设备被称为流控制器。流控制器通常是指 DMA 控制器，在 DMA 通道使能前，软件会编程好数据包的长度。如果在 DMA 通道使能时数据包长度未知，则源外围设备或目标外围设备均可作为流控制器。

对于已知数据包长度的简单或低性能外围设备（即当外围设备为流控制器时），表明事件已完成的一种简单方式是外围设备产生一个中断并使能处理器对 DMA 通道进行重新编程。

如果外围设备配置为流控制器，则忽略 （CCONTROL 寄存器中的）传输长度值。

DMA 传输结束时：

1. DMA 控制器向外围设备产生一个应答，以指示传输已完成。
2. 产生一个 TC 中断 （如果被使能）。
3. DMA 控制器继续到下一个 LLI。

在接下来的小节中，我们将介绍 4 种允许的传输类型的 DMA 控制器数据流序列：

- 存储器到外设 （仅主机 1）。
- 外设到存储器 （仅主机 1）。
- 存储器到存储器。
- 外设到外设 （仅主机 1）。

每种传输类型可使用外围设备或DMA控制器作为流控制器，这样共有8种可能的控制情况。

表 283 所示为每种传输类型所使用的请求信号。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 283. DMA 请求信号使用情形

传输方向	请求发生器	流控制器
存储器到外围设备	外设	DMA 控制器
存储器到外围设备	外设	外设
外围设备到存储器	外设	DMA 控制器
外围设备到存储器	外设	外设
存储器到存储器	DMA 控制器	DMA 控制器
源外围设备到目标外围设备	源外围设备和目标外围设备	源外围设备
源外围设备到目标外围设备	源外围设备和目标外围设备	目标外围设备
源外围设备到目标外围设备	源外围设备和目标外围设备	DMA 控制器

19.8.2.1 外围设备到存储器或存储器到外围设备 DMA 流

对于外围设备到存储器或存储器到外围设备的 DMA 流，会出现以下序列：

1. 编程和使能 DMA 通道。
2. 等待 DMA 请求。
3. 当下列情况出现时， DMA 控制器开始传输数据：
   - DMA 请求变得有效。
   - DMA 流具有最高挂起优先级。
   - DMA 控制器是 AHB 总线的总线主机。
4. 如果在传输数据时出错，则产生一个错误中断，禁用 DMA 流，数据流终止。
5. 如果 DMA 控制器正在执行流控制，减少传输计数。
6. 如果传输已经结束（当 DMA 控制器正在执行流控制时，通过传输计数到达 0 来指示；或当外围设备正在执行流控制时，通过外围设备发送 DMA 请求来指示）：
   - DMA 控制器用一个 DMA 应答来响应。
   - 产生终结计数中断 （该中断可以被掩码）。
   - 如果 CLLI 寄存器不为 0，则重新加载 CSRCADDR、 CDESTADDR、 CLLI 和 CCONTROL 寄存器，并返回到步骤 2。但是，如果 CLLI 为 0，则 DMA 流被禁用且流序列终止。

19.8.2.2 外围设备到外围设备 DMA 流

对于外围设备到外围设备 DMA 流，会出现以下序列：

1. 编程和使能 DMA 通道。
2. 等待源 DMA 请求。
3. 当下列情况出现时， DMA 控制器开始传输数据：
   - DMA 请求变得有效。
   - DMA 流具有最高挂起优先级。
   - DMA 控制器是 AHB 总线的总线主机。
4. 如果在传输数据时出错，则产生一个错误中断，禁用 DMA 流，数据流终止。
5. 如果 DMA 控制器正在执行流控制，减少传输计数。
6. 如果传输已经结束（当 DMA 控制器正在执行流控制时，通过传输计数到达 0 来指示；或当外围设备正在执行流控制时，通过外围设备发送 DMA 请求来指示）：

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- DMA 控制器用一个对源外围设备的 DMA 应答来响应。
- 后续的源 DMA 请求被忽略。

7. 当目标 DMA 请求变得有效，并且 DMA 控制器 FIFO 中有数据时，将数据传输到目标外围设备。
8. 如果在传输数据时出错，则产生一个错误中断，禁用 DMA 流，数据流终止。
9. 如果传输已经结束，当 DMA 控制器正在执行流控制时，通过传输计数到达 0 来指示；或当外围设备正在执行流控制时，通过发送 DMA 请求来指示：以下情况发生：
   - DMA 控制器用一个对目标外围设备的 DMA 应答来响应。
   - 产生终结计数中断 （该中断可以被掩码）。
   - 如果 CLLI 寄存器不为 0，则重新加载 CSRCADDR、CDESTADDR、CLLI 和 CCONTROL 寄存器，并返回到步骤 2。但是，如果 CLLI 为 0，则 DMA 流被禁用且流序列终止。

19.8.2.3 存储器到存储器 DMA 流

对于存储器到存储器 DMA 流，会出现以下序列：

1. 编程和使能 DMA 通道。
2. 每当 DMA 通道具有最高的待处理优先级，并且 DMA 控制器获得 AHB 总线的主接口资格时，传输数据。
3. 如果在传输数据时出错，则产生一个错误中断，禁用 DMA 流。
4. 减少传输计数。
5. 如果计数已达到 0：
   - 产生一个终结计数中断 （该中断可以被掩码）。
   - 如果 CLLI 寄存器不为 0，则重新加载 CSRCADDR、CDESTADDR、CLLI 和 CCONTROL 寄存器，并返回到步骤 2。但是，如果 CLLI 为 0，则 DMA 流被禁用且流序列终止。

注：存储器到存储器的传输应使用低通道优先级进行编程，否则在存储器到存储器的传输结束前，其他 DMA 通道都无法访问总线，或者其他 AHB 主机无法执行任何事务。

19.8.3 中断请求

当前 LLI 对应的所有数据被传输到目标之后，如果遇到一个 AHB 错误或传输结束时（终结计数），可以产生中断请求。可以通过编程相关 CCONTROL 和 CCONFIG 通道寄存器中的位来掩码中断。 DMA 还提供了中断状态寄存器，将中断掩码前 （RAWINTTCSTAT 和 RAWINTERRSTAT）以及中断掩码后 （INTTCSTAT 和 INTERRSTAT）的所有 DMA 通道的中断请求组合起来。INTSTAT 寄存器将 INTTCSTAT 和 INTERRSTAT 请求组合到单个寄存器中，以便于迅速找到中断源。写 INTTCCLEAR 或 INTERRCLR 寄存器并将某个位设置为高电平，可以选择性地清除中断。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

19.8.3.1 硬件中断序列流

当 DMA 中断请求发生时，中断服务例程需要：

1. 读取 INTTCSTAT 寄存器，以判断中断是否因为传输的结束而产生 （中断计数）。高电平位表明传输结束。如果多个请求有效，建议先检查最高优先级的通道。
2. 读取 INTERRSTAT 寄存器，以确定中断是否是因为错误的出现而产生。高电平位表明出现错误。
3. 处理中断请求。
4. 对于终结计数中断，可以向 INTTCCLR 寄存器的相关位写 1。对于错误中断，可以向 INTERRCLR 寄存器的相关位写 1，清除中断请求。

19.8.4 地址产生

地址产生可以是递增，也可以是非递增的 （不支持地址换行）。

特别像存储器这种器件，在连发访问时不允许跨过指定的地址边界。 DMA 控制器假设这是在任何源或目标区都配置为递增寻址的情况下。而边界则假设为与指定的连发大小对齐。例如，如果将 16 传输连发的通道设置为 32 位宽的设备，则该边界为 64 字节对齐 （即地址位 [5:0] 等于 0）。如果 DMA 连发超出其中一个边界，则一个连发传输就会分成几个独立的 AHB 传输。

注：当数据传输与 SDRAM 相关时， SDRAM 访问必须始终编程为 32 位访问。 SDRAM 内存控制器不支持使用半字或字节传输长度的 AHB-INCR4 或 INCR8 连发。 SDRAM 的起始地址应当始终与连发边界地址对齐。

19.8.4.1 跨边界的字对齐传输

配置一个用于 16 次连发传输的通道，每次向目标 （使能递增寻址）传输 32 位数据。当前连发的起始地址为 0x0C000024，下一个边界为 0x0C000040 （由连发大小和传输宽度计算而来）。

传输将分成 2 个 AHB 传输：

- 起始地址为 0x0C000024 的 7 传输连发
- 起始地址为 0x0C000040 的 9 传输连发。

19.8.5 分散 / 聚集

通过使用链接列表来支持分散 / 聚集。这意味着源区和目标区不一定要占用连续的存储区。不需要分散 / 聚集时， CLLI 寄存器必须设为 0。

源和目标数据区由一连串的链接列表来定义。每个链接列表项 (LLI) 控制着一个数据块的传输，然后可以选择并加载另一个 LLI 来继续 DMA 操作或停止 DMA 流。第一个 LLI 被编程到 DMA 控制器中。

LLI 描述的传输数据 （指数据包）通常需要进行一次或多次 DMA 连发传输 （到每个源和目标）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

19.8.5.1 链接列表项

链接列表项 (LLI) 由 4 个字组成。这些字按照以下顺序排列：

1. CSRCADDR
2. CDESTADDR
3. CLLI
4. CCONTROL

注：CCONFIG DMA 通道配置寄存器不属于链接列表项的一部分。

19.8.5.1.1 编程 DMA 控制器，用于分散 / 聚集 DMA

如需编程 DMA 控制器以分散 / 聚集 DMA：

1. 向存储器中写入整个 DMA 传输的 LLI。每个链接列表项包含 4 个字：
   - 源地址。
   - 目标地址。
   - 指向下一个 LLI 的指针。
   - 控制字。

   最后一个 LLI 的链接列表项字指针需设为 0。
2. 选择一个具有所需优先级的空闲 DMA 通道。DMA 通道 0 的优先级最高，DMA 通道 7 的优先级最低。
3. 将第一个先前写入存储器的链接列表项写入 DMA 控制器的相关通道。
4. 向通道配置寄存器写入通道配置信息，并置位通道使能位。随着每个链接列表项的加载， DMA 控制器会传输第一个，然后按顺序传输数据包。
5. 每个 LLI 结束时可以产生中断，由 CCONTROL 寄存器的终结计数位决定。如果该位被置位，相关 LLI 结束时会产生中断。然后处理中断请求， INTTCCLEAR 寄存器中的相关位必须置位，以清除该中断。

19.8.5.1.2 分散 / 聚集 DMA 示例

对于 LLI 的示例，请参见图 39。需要将某个存储区的内容传输到一个外围设备。图左侧为每个 LLI 的地址，以十六进制的形式表示。图的右侧显示了包含要传输的数据的存储器。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 39. LLI 示例

第一个 LLI 存储在 0x2000 0000，定义了要传输的第一个数据块，这个数据块是存储在地址 0x2000 A200 和 0x2000 AE00 之间的数据：

- 源起始地址 0x2000 A200。
- 目标地址设为目标外设地址。
- 传输宽度，字 （32 位）。
- 传输长度， 3072 字节 (0xC00)。
- 源和目标连发大小， 16 次传输。
- 下一个 LLI 地址：0x2000 0010。

第二个 LLI 存储在 0x2000 0010 上，描述要传输的下一个数据块：

- 源起始地址 0x2000 B200。
- 目标地址设为目标外设地址。
- 传输宽度，字 （32 位）。
- 传输长度， 3072 字节 (0xC00)。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

- 源和目标连发大小， 16 次传输。
- 下一个 LLI 地址：0x2000 0020。

这样，一个描述符链就形成了，其中每个描述符都指向链接列表中的下一项。为了初始化 DMA 流，第一个 LLI (0x2000 0000) 编入到 DMA 控制器中。当第一个数据包被传输后，下一个 LLI 就自动加载。

最后一个 LLI 存储在 0x2000 0070 上，其中包含：

- 源起始地址 0x2000 1200。
- 目标地址设为目标外设地址。
- 传输宽度，字 （32 位）。
- 传输长度， 3072 字节 (0xC00)。
- 源和目标连发大小， 16 次传输。
- 下一个 LLI 地址：0x0。

由于下一个 LLI 地址设为 0，因此，这是最后一个描述符；而且， DMA 通道在传输完最后一个数据项后被禁用。此时，通道也可能设置成产生中断，以向 ARM 处理器指明通道可以重新编程。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

20.1 本章导读

SD/MMC 卡接口可在所有 LPC43xx 器件上使用。

20.2 基本配置

表 284. SDIO 的时钟与电源控制

	基本时钟	分支时钟	工作频率
SDIO 寄存器接口	BASE_M4_CLK	CLK_M4_SDIO	最大 204 MHz
SDIO 比特率时钟	BASE_SDIO_CLK	CLK_SDIO	MMC-Ver4.0 SD 卡的最大工作频率是 25 MHz、 26 MHz 或 52 MHz。

SDIO 通过 SD_RST 复位 （复位 # 20）。

20.3 功能要点

SD/MMC 卡接口支持以下模式：

- 安全数字存储器 （SD 3.0 版）
- 安全数字 I/O （SDIO 2.0 版）
- 消费类电子产品先进的运输架构 （CE-ATA 1.1 版）
- 多媒体卡 （MMC 4.4 版）

20.4 简介

SD/MMC 安全接口包括以下主要功能块：

- 总线接口单元 (BIU) - 为寄存器和数据读取 / 写入提供 AHB 和 DMA 接口。
- 卡接口单元 (CIU) - 处理卡协议和提供时钟管理。
- SD/MMC DMA - 集成 DMA 控制器

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

注：卡检测和写保护信号来自 SD/MMC 卡插槽，而非 SD/MMC 卡。

图 40. SD/MMC 功能框图

20.5 引脚说明

表 285. SDIO 引脚说明

引脚功能	方向	描述
SD_CLK	O	SD/SDIO/MMC 时钟。
SD_CD	I	单插槽 SDIO 卡检测。
SD_WP	I	SDIO 卡写保护。
SD_LED	O	“LED 开”信号。该信号提醒用户在访问 SD 卡时不要移除该卡。
SD_CMD	I/O	指令输入 / 输出。
SD_D[7:0]	I/O	数据线 DAT[7:0] 的数据输入 / 输出。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 285. SDIO 引脚说明 （续）

引脚功能	方向	描述
SD_VOLT[2:0]	O	SD/MMC 总线电压选择输出 2:0。
SD_RST	O	MMC4.4 卡 SD/MMC 复位信号。
SD_POW	O	< 待定 >

< 待定 >

20.6 寄存器描述

表 286. 寄存器简介：SDMMC （基址：0x4000 4000）

名称	访问类型	地址偏移	描述	复位值	参考
CTRL	R/W	0x000	控制寄存器	0	表 287
PWREN	R/W	0x004	电源使能寄存器	0	表 288
CLKDIV	R/W	0x008	时钟分频器寄存器	0	表 289
CLKSRC	R/W	0x00C	SD 时钟源寄存器	0	表 290
CLKENA	R/W	0x010	时钟使能寄存器	0	表 291
TMOUT	R/W	0x014	超时寄存器		表 292
CTYPE	R/W	0x018	卡类型寄存器	0	表 293
BLKSIZ	R/W	0x01C	块大小寄存器	0x200	表 294
BYTCNT	R/W	0x020	字节计数寄存器	0x200	表 295
INTMASK	R/W	0x024	中断掩码寄存器		表 296
CMDARG	R/W	0x028	指令参数寄存器	0x00000000	表 297
CMD	R/W	0x02C	指令寄存器	0x00000000	表 298
RESP0	R	0x030	响应寄存器 0	0x00000000	表 299
RESP1	R	0x034	响应寄存器 1	0x00000000	表 300
RESP2	R	0x038	响应寄存器 2	0	表 301
RESP3	R	0x03C	响应寄存器 3	0	表 302
MINTSTS	r	0x040	掩码中断状态寄存器	复位值	表 303
RINTSTS	R/W	0x044	原始中断状态寄存器	0	表 304
STATUS	R	0x048	状态寄存器		表 305
FIFOTH	R/W	0x04C	FIFO 阈值水印寄存器		表 306
CDETECT	R	0x050	卡检测寄存器		表 307
WRTPRT	R	0x054	写保护寄存器		表 308
-	-	0x058	保留	-	-
TCBCNT	R	0x05C	CIU 卡已传输字节计数寄存器	0x00000000	表 309
TBBCNT	R	0x060	主机至 BIU-FIFO 已传输字节计数寄存器	0	表 310
DEBNCE	R/W	0x064	去抖动计数寄存器		表 311
-	-	0x068	保留	-	-
-	-	0x06C	保留	-	-
UHS_REG	R/W	0x074	UHS-1 寄存器	0x00000000	表 312
RST_N	R/W	0x078	硬件复位		表 313
-	-	0x07C	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 286. 寄存器简介：SDMMC （基址：0x4000 4000）*（续）*

名称	访问类型	地址偏移	描述	复位值	参考
BMOD	R/W	0x080	总线模式寄存器	0x00000000	表 314
PLDMND	W	0x084	查询需求寄存器	0x00000000	表 315
DBADDR	R/W	0x088	描述符表基地址寄存器	0x00000000	表 316
IDSTS	R/W	0x08C	内部 DMAC 状态寄存器	0x00000000	表 317
IDINTEN	R/W	0x090	内部 DMAC 中断使能寄存器	0x00000000	表 318
DSCADDR	R	0x094	当前主机描述符地址寄存器	0x00000000	表 319
BUFADDR	R	0x098	当前缓冲描述符地址寄存器	0x00000000	表 320
数据	R/W	? 0x100	数据 FIFO 读取 / 写入；如果地址等于或大于 0x100，则只要选择设备，就选择 FIFO。0x100 及以上地址将映射到数据 FIFO。多个地址映射到数据 FIFO，以便可通过连发访问 FIFO。		-

20.6.1 控制寄存器 (CTRL)

表 287. 控制寄存器 （CTRL，地址 0x4000 4000）位描述

位	符号	值	描述	复位值
0	CONTROLLER_RESET		控制器复位。如需复位控制器，固件应当将位设为 1。在两个 AHB 和两个 clk_in 时钟周期后，该位会自动清除。这会复位： - BIU/CIU 接口 - CIU 和状态机 - 控制寄存器的 abort_read_data、send_irq_response 和 read_wait 位 - 指令寄存器的 start_cmd 位不会影响任何寄存器或 DMA 接口、FIFO 或主机中断	0
		0	无变化	
		1	复位 SD/MMC 控制器	
1	FIFO_RESET		FIFO 复位。如需复位 FIFO，固件应当将位设为 1。完成复位操作后，该位会自动清除。两个 AHB 时钟周期后自动清除。	0
		0	无变化	
		1	复位至数据 FIFO 复位 FIFO 指针	
2	DMA_RESET		DMA 复位。如需复位 DMA 接口，固件应当将位设为 1。两个 AHB 时钟周期后，该位会自动清除。	0
		0	无变化	
		1	复位内部 DMA 接口控制逻辑	
3	-		保留	-
4	INT_ENABLE		全局中断使能 / 禁用位。仅当该位为 1，并且一个或多个未掩蔽中断已置位时，中断端口才为 1。	0
		0	禁用中断	
		1	使能中断	
5			保留。只可向此位写入 0	
6	READ_WAIT		读 / 等待。用于将读等待发送至 SDIO 卡。	0
		0	清除读等待	
		1	发送读等待	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 287. 控制寄存器 （CTRL，地址 0x4000 4000）位描述 *（续）*

位	符号	值	描述	复位值
7	SEND_IRQ_RESPONSE		发送 IRQ 响应。发送响应后，位自动清除。如需等待 MMC 卡中断，主机发送 CMD40，SD/MMC 等待来自 MMC 卡的中断响应。同时，如果主机想让 SD/MMC 退出中断等待状态，可以置位该位，而 SD/MMC 接口指令状态机会发送总线 CMD40 响应并返回至空闲状态。	0
		0	无变化	
		1	发送自动 IRQ 响应	
8	ABORT_READ_DATA		中止读取数据。用于 SDIO 卡挂起序列。	0
		0	无变化	
		1	在读传输过程中，挂起指令发出后，软件会对卡进行查询，找出发生挂起的时间。挂起发生后，软件进行置位，复位正在等待下一个数据块的数据状态机。数据状态机复位至空闲状态后，位自动清除。	
9	SEND_CCSD		发送 CCSD。置位时， SD/MMC 向 CE-ATA 设备发送 CCSD。仅在当前指令需要 CCS（即 RW_BLK）且 CE-ATA 设备中使能中断时，软件置位该位。 CCSD 模式发送至设备后， SD/MMC 接口自动清除 send_ccsd 位。同时，也置位 RINTSTS 寄存器中的指令完成 (CD) 位，如果指令完成中断未掩码，则向主机产生中断。 注：send_ccsd 位置位后，需要两个卡时钟周期来驱动 CMD 线的 CCSD。因此，在边界状态下，CCSD 可能会发送至 CE-ATA 设备，即使该设备发送了 CCS。	0
		0	如果 SD/MMC 没有复位该位，则清除该位。	
		1	向 CE-ATA 设备发送指令完成信号禁用 (CCSD)。	
10	SEND_AUTO_STOP_CCSD		发送自动停止 CCSD。注：始终将 send_auto_stop_ccsd 和 send_ccsd 位同时置位； send_auto_stop_ccsd 不应当独立于 send_ccsd 置位。设置后， SD/MMC 接口自动向 CE-ATA 设备发送内部产生的 STOP 指令 (CMD12)。发送内部产生的 STOP 指令后，RINTSTS 的自动指令完成 (ACD) 位将置位，如果自动指令完成中断未掩码，则向主机产生中断。发送 CCSD 后，SD/MMC 接口自动清除 send_auto_stop_ccsd 位。	0
		0	如果 SD/MMC 没有复位该位，则清除该位。	
		1	向 CE-ATA 设备发送 CCSD 后，发送内部发生的 STOP 指令。	
11	CEATA_DEVICE_INTERRUPT _STATUS		CEATA 设备中断状态。重启后或 CE-ATA 设备的任何其他复位后，软件应当相应地写该位。复位后， CE-ATA 设备中断一般被禁用 (nIEN = 1)。如果主机使能 CE-ATA 设备中断，则软件应当置位该位。	0
		0	CE-ATA 设备中未使能中断 （ATA 控制寄存器中 nIEN = 1）	
		1	CE-ATA 设备中使能中断 （ATA 控制寄存器中 nIEN = 0）	
15:12	-		保留	-
19:16	CARD_VOLTAGE_A		< 待定 >	0
23:20	CARD_VOLTAGE_B		< 待定 >	0
24	ENABLE_OD_PULLUP		< 待定 >	1
		0	禁用	
		1	使能	
25	USE_INTERNAL_DMAC		SD/MMC DMA 使用。	0
		0	主机通过从机接口行进数据传输	
		1	使用内部 DMA 进行数据传输	
31:26	-		保留	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

20.6.2 电源使能寄存器 (PWREN)

表 288. 电源使能寄存器 （PWREN，地址 0x4000 4004）位描述

位	符号	描述	复位值
0	POWER_ENABLE	用于卡的电源开关；开启电源后，固件在尝试初始化卡前，应当等待调压器 / 开关爬坡时间。 0 - 断电 1 - 通电 <tbd>	0
31:1	-	保留	-

20.6.3 时钟分频器寄存器 (CLKDIV)

表 289. 时钟分频器寄存器 （CLKDIV，地址 0x4000 4008）位描述

位	符号	描述	复位值
7:0	CLK_DIVIDER0	时钟分频器 -0 的值。时钟分频为 2*n。例如，值为 0 表示根据 2*0 = 0 进行分频（不分频，绕过），值为 1 表示根据 2*1 = 2 进行分频，值为 ff 表示根据 2*255 = 510 进行分频，等等。	0
15:8	CLK_DIVIDER1	时钟分频器 -1 的值。时钟分频为 2*n。例如，值为 0 表示根据 2*0 = 0 进行分频（不分频，绕过），值为 1 表示根据 2*1 = 2 进行分频，值为 ff 表示根据 2*255 = 510 进行分频，等等。在仅 MMC-Ver3.3 模式下，由于仅支持一个时钟分频器，因此不实施位。	0
23:16	CLK_DIVIDER2	时钟分频器 -2 的值。时钟分频为 2*n。例如，值为 0 表示根据 2*0 = 0 进行分频（不分频，绕过），值为 1 表示根据 2*1 = 2 进行分频，值为 ff 表示根据 2*255 = 510 进行分频，等等。在仅 MMC-Ver3.3 模式下，由于仅支持一个时钟分频器，因此不实施位。	0
31:24	CLK_DIVIDER3	时钟分频器 -3 的值。时钟分频为 2*n。例如，值为 0 表示根据 2*0 = 0 进行分频（不分频，绕过），值为 1 表示根据 2*1 = 2 进行分频，值为 ff 表示根据 2*255 = 510 进行分频，等等。在仅 MMC-Ver3.3 模式下，由于仅支持一个时钟分频器，因此不实施位。值为 0 表示根据 2*0 = 0 进行分频 （不分频，绕过），值为 1 表示根据 2*1 = 2 进行分频，值为 ff 表示根据 2*255 = 510 进行分频，等等。在仅 MMC-Ver3.3 模式下，由于仅支持一个时钟分频器，因此不实施位。	0

20.6.4 SD 时钟源寄存器 (CLKSRC)

表 290. SD 时钟源寄存器 （CLKSRC，地址 0x4000 400C）位描述

位	符号	描述	复位值
1:0	CLK_SOURCE	SD 卡的时钟分频器源。 00 – 时钟分频器 0 01 – 时钟分频器 1 10 – 时钟分频器 2 11 – 时钟分频器 3 在仅 MMC-Ver3.3 控制器中，仅支持一个时钟分频器。cclk_out 始终来自时钟分频器 0，且该寄存器未实施。	0
31:1	-	保留	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

20.6.5 时钟使能寄存器 (CLKENA)

表 291. 时钟使能寄存器 （CLKENA，地址 0x4000 4010）位描述

位	符号	描述	复位值
0	CCLK_ENABLE	SD 卡时钟的低电源控制。支持一个 MMC 卡时钟。 0 – 非低功耗模式 1 - 低功耗模式； 当卡处于空闲状态时，停止时钟 （通常应仅对 MMC 和 SD 存储卡置位；对于 SDIO 卡，如果必须检测中断，则不应当停止时钟）。对于仅 MMC-Ver3.3 模式： 0 – 时钟禁用 1 – 时钟使能	0
15:1	-	保留	-
16	CCLK_LOW_POWER	SD 卡时钟的时钟使能控制。支持一个 MMC 卡时钟。 0 – 时钟禁用 1 – 时钟使能 在仅 MMC-Ver3.3 模式中： 0 – 非低功耗模式 1 - 低功耗模式； 当卡处于空闲状态时，停止时钟 （通常应仅对 MMC 和 SD 存储卡置位；对于 SDIO 卡，如果必须检测中断，则不应当停止时钟）。	0
31:17	-	保留	-

20.6.6 超时寄存器 (TMOUT)

表 292. 超时寄存器 （TMOUT，地址 0x4000 4014）位描述

位	符号	描述	复位值
7:0	RESPONSE_TIMEOUT	响应超时值。值为卡输出时钟数 - cclk_out。	0x40
31:8	DATA_TIMEOUT	卡数据读取超时值；相同值还用于“由主机造成的数据不足”超时。值为卡输出时钟数 - 所选卡的 cclk_out。“由主机造成的数据不足”超时。值为卡输出时钟数 - 所选卡的 cclk_out。	0xFFFFFF

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

20.6.7 卡类型寄存器 (CTYPE)

表 293. 卡类型寄存器 （CTYPE，地址 0x4000 4018）位描述

位	符号	描述	复位值
0	CARD_WIDTH0	指示卡为 1 位或 4 位： 0 – 1 位模式 1 - 4-bit mode	0
15:1	-	保留	-
16	CARD_WIDTH1	指示卡为 8 位： 0 – 非 8 位模式 1 – 8 位模式	0
31:17	-	保留	-

20.6.8 块大小寄存器 (BLKSIZ)

表 294. 块大小寄存器 （BLKSIZ，地址 0x4000 401C）位描述

位	符号	描述	复位值
15:0	BLOCK_SIZE	块大小	0x200
31:16	-	保留	-

20.6.9 字节计数寄存器 (BYTCNT)

表 295. 字节计数寄存器 （BYTCNT，地址 0x4000 4020）位描述

位	符号	描述	复位值
31:0	BYTE_COUNT	要传输的字节数量；应当等于整数乘以进行模块传输的块大小。对于未定义数量的字节传输，字节计数应当设为 0。当字节计数设为 0 时，主机应当明确发送停止 / 中止指令，结束数据传输。	0x200

20.6.10 中断掩码寄存器 (INTMASK)

表 296. 中断掩码寄存器 （INTMASK，地址 0x4000 4024）位描述

位	符号	描述	复位值
0	CDET	卡检测。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
1	RE	响应错误。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
2	CDONE	指令完成。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
3	DTO	数据传输结束。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
4	TXDR	发送 FIFO 数据请求。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
5	RXDR	接收 FIFO 数据请求。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
6	RCRC	响应 CRC 错误。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
7	DCRC	数据 CRC 错误。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 296. 中断掩码寄存器 （INTMASK，地址 0x4000 4024）位描述 *（续）*

位	符号	描述	复位值
8	RTO	响应超时。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
9	DRTO	数据读取超时。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
10	HTO	“主机造成的数据不足”超时 (HTO)/Volt_switch_int。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
11	FRUN	FIFO 上溢 / 下溢错误。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
12	HLE	硬件锁定写错误。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
13	SBE	起始位错误。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
14	ACD	自动指令完成。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
15	EBE	结束位错误（读）/ 未写 CRC。用于掩码不需要的中断。值为 0 时掩码中断；值为 1 时使能中断。	0
16	SDIO_INT_MASK	掩码 SDIO 中断。掩码时，该卡的 SDIO 中断检测被禁用。值为 0 时掩码中断，值为 1 时使能中断。在仅 MMC-Ver3.3 模式下，这些位始终为 0。	0
31:17	-	保留。	-

20.6.11 指令参数寄存器 (CMDARG)

表 297. 指令参数寄存器 （CMDARG，地址 0x4000 4028）位描述

位	符号	描述	复位值
31:0	CMD_ARG	值表明传给卡的指令参数。	0

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

20.6.12 指令寄存器 (CMD)

表 298. 指令寄存器 （CMD，地址 0x4000 402C）位描述

位	符号	值	描述	复位值
5:0	CMD_INDEX		指令索引	0
6	RESPONSE_EXPECT		响应期望	0
		0	期望卡无响应	
		1	期望卡有响应	
7	RESPONSE_ LENGTH		响应长度	0
		0	期望卡有短响应	
		1	期望卡有长响应	
8	CHECK_RESPONSE_CRC		检查响应 crc。一些指令响应未返回有效的 CRC 位。如需禁用控制器的 CRC 检查，软件应当禁用这些指令的 CRC 检查。	0
		0	不检查响应 CRC	
		1	检查响应 CRC	
9	DATA_EXPECTED		数据期望	0
		0	期望无数据传输 （R/W）	
		1	期望有数据传输 （R/W）	
10	READ_WRITE		R/W。如果无需卡上的数据，则无需关注。	0
		0	读卡	
		1	期望有数据传输 （R/W）	
11	TRANSFER_MODE		传输模式。如果期望无数据，则无需关注。	0
		0	模块数据传输指令	
		1	流数据传输指令	
12	SEND_AUTO_STOP		发送自动停止。设置时，SD/MMC 接口在数据传输结束时向 SD_MMC_CEATA 卡发送停止指令。参见表 321，确定： - 何时应当置位 send_auto_stop 位，因为一些数据传输不需要明确的停止指令 - 软件应当明确发送停止指令的开放式传输 另外，何时发送恢复指令，以恢复挂起的 SD-Combo 卡的内存访问。如果挂起的数据传输需要 send_auto_stop，则应该当正确置位。如果无需卡上的数据，则无需关注。	0
		0	数据传输结束时不发送停止指令	
		1	数据传输结束时发送停止指令	
13	WAIT_PRVDATA_COMPLETE		等待 prvdata 完成。wait_prvdata_complete = 0 选项通常用于在数据传输过程中查询卡状态或停止当前数据传输； card_number 应当与之前指令中的相同。	0
		0	立刻发送指令，即使之前的数据传输还未完成。	
		1	等待之前的数据传输完成，然后才发送指令。	
14	STOP_ABORT_CMD		停止中止指令。如果开放式或预定义数据传输正在进行，主机发送停止或中止指令以停止数据传输，则应当置位该位，这样 CIU 的指令 / 数据状态机可以正确返回空闲状态。这同样也适用于启动模式传输。为了中止启动模式，该位应当与 CMD[26] = disable_boot 同时置位。	0
		0	无正在执行的用于停止当前数据传输的停止或中止指令。如果将中止指令发送到当前所选的功能 - 编号或不处于数据传输模式，则该位应当设为 0。	
		1	有正在执行的用于停止当前数据传输的停止或中止指令。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 298. 指令寄存器 （CMD，地址 0x4000 402C）位描述 *（续）*

位	符号	值	描述	复位值
15	SEND_INITIALIZATION		发送初始化指令。开启电源后，必须先向卡发送 80 个时钟，然后才向卡发送初始化指令。向卡发送第一个指令时设置该位，这样控制器将在向卡发送指令前初始化时钟。该位不应当设置为任何一种启动模式（可选或强制）。	0
		0	发送该指令前，不发送初始化序列 （1 的 80 个时钟）。	
		1	发送该指令前，发送初始化序列。	
20:16	-		保留	0
21	UPDATE_CLOC_REGISTERS_ONLY		仅更新时钟寄存器。按照发送到卡时钟域中的寄存器值：CLKDIV、CLRSRC、 CLKENA。更改卡时钟 （更改频率、截断关闭或开启和设置低频率模式）；旨在更改时钟频率或停止时钟，但无需向卡发送指令。正常指令序列期间，当 update_clock_registers_only = 0，下列控制寄存器从 BIU 发送到 CIU：CMD、CMDARG、TMOUT、CTYPE、BLKSIZ、BYTCNT。向卡发送新指令序列时，CIU 使用新的寄存器值。置位时，无指令完成中断，这是因为未向 SD_MMC_CEATA 卡发送指令。 registers_only。	0
		0	正常指令序列	
		1	不发送指令，仅向卡时钟域更新时钟寄存器值	
22	READ_CEATA_DEVICE		读取 ceata 设备。软件应置位该位，表明 CE-ATA 设备正在进行读传输访问。该位用于在执行 CE-ATA 读传输时禁用读取数据超时指示。 I/O 传输延迟的最大值不少于 10 秒。在等待 CE-ATA 设备的数据时，SD/MMC 接口不应当指示读取数据超时。	0
		0	主机不在执行 CE-ATA 设备的读访问 （RW_REG 或 RW_BLK）。	
		1	主机正在执行 CE-ATA 设备的读访问 （RW_REG 或 RW_BLK）。	
23	CCS_EXPECTED		CCS 期望。如果指令需要 CE-ATA 设备的指令完成信号 (CCS)，则软件应将该控制位置位。SD/MMC 置位 RINTSTS 寄存器的数据传输结束 (DTO) 位，当数据传输结束中断未掩码时，则向主机发送中断。	0
		0	CE-ATA 设备中的中断未使能（ATA 控制寄存器中 nIEN = 1），或指令无需设备的 CCS。	
		1	CE-ATA 设备中的中断使能 (nIEN = 0)，RW_BLK 指令需要 CE-ATA 设备的指令完成信号。	
24	ENABLE_BOOT		使能启动 – 置位该位仅用于强制启动模式。当软件置位该位和 start_cmd 时， CIU 通过将 CMD 线确认为低电平，开始相应卡的启动序列。切勿将 disable_boot 和 enable_boot 同时置位。	0
25	EXPECT_BOOT_ACK		期望启动应答。当软件和置位该位和 enable_boot 时，CIU 期望获得所选卡的 0-1-0 启动应答开始模式。	0
26	DISABLE_BOOT		禁用启动。当软件置位该位和 start_cmd 时，CIU 结束启动操作。切勿将 disable_boot 和 enable_boot 同时置位。	0
27	BOOT_MODE		启动模式	0
		0	强制启动操作	
		1	可选启动操作	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 298. 指令寄存器 （CMD，地址 0x4000 402C）位描述 *（续）*

位	符号	值	描述	复位值
28	VOLT_SWITCH		电压切换位	0
		0	无电压切换	
		1	电压切换使能；必须仅为 CMD11 置位	
30:29	-		保留	
31	START_CMD		开始指令。CIU 执行指令后，位清除。置位后，主机不应当尝试写任何指令寄存器。如果尝试写入，原始中断寄存器中的硬件锁错误会被置位。发送指令并接收到 SD_MMC_CEATA 卡的响应后，原始中断寄存器中的指令完成位被置位。	

20.6.13 响应寄存器 0 (RESP0)

表 299. 响应寄存器 0 （RESP0，地址 0x4000 4030）位描述

位	符号	描述	复位值
31:0	RESPONSE0	响应的位 [31:0]	0

20.6.14 响应寄存器 1 (RESP1)

表 300. 响应寄存器 1 （RESP1，地址 0x4000 4034）位描述

位	符号	描述	复位值
31:0	RESPONSE1	寄存器代表长响应的位 [63:32]。如果 CIU 发送自动停止指令，将响应存放到寄存器中。主机之前发送的指令的响应仍保存在响应 0 寄存器中。仅针对数据传输指令发出额外的自动停止指令，而这类响应一般都较短。有关 CIU 发送自动停止指令的信息，请参见 “自动停止 < 待定 >”。	0

20.6.15 响应寄存器 2 (RESP2)

表 301. 响应寄存器 2 （RESP2，地址 0x4000 4038）位描述

位	符号	描述	复位值
31:0	RESPONSE2	长响应的位 [95:64]	0

20.6.16 响应寄存器 3 (RESP3)

表 302. 响应寄存器 3 （RESP3，地址 0x4000 403C）位描述

位	符号	描述	复位值
31:0	RESPONSE3	长响应的位 [127:96]	0

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

20.6.17 掩码中断状态寄存器 (MINTSTS)

表 303. 掩码中断状态寄存器 （MINTSTS，地址 0x4000 4040）位描述

位	符号	描述	复位值
0	CDET	卡检测。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
1	RE	响应错误。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
2	CDONE	指令完成。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
3	DTO	数据传输结束。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
4	TXDR	发送 FIFO 数据请求。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
5	RXDR	接收 FIFO 数据请求。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
6	RCRC	响应 CRC 错误。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
7	DCRC	数据 CRC 错误。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
8	RTO	响应超时。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
9	DRTO	数据读取超时。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
10	HTO	“主机造成的数据不足”超时 (HTO)。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
11	FRUN	FIFO 上溢 / 下溢错误。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
12	HLE	硬件锁定写错误。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
13	SBE	起始位错误。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
14	ACD	自动指令完成。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
15	EBE	结束位错误（读）/ 未写 CRC。仅当中断掩码寄存器中的对应位置位时，使能中断。	0
16	SDIO_INTERRUPT	SDIO 卡中断。仅当中断掩码寄存器中相应的 sdio_int_mask 位置位时，使能 SDIO 卡中断 （mask 位为 1 时使能中断；为 0 时掩码中断）。 0 – 无 SDIO 卡中断 1 – 有 SDIO 卡中断 在仅 MMC-Ver3.3 模式下，这些位始终为 0。	-
31:17	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

20.6.18 原始中断状态寄存器 (RINTSTS)

表 304. 原始中断状态寄存器 （RINTSTS，地址 0x4000 4044）位描述

位	符号	描述	复位值
0	CDET	卡检测。写位清除状态位。值为 1 时清除状态位，值为 0 时保持位不变。无论是否是中断掩码状态，都会记录位。	0
1	RE	响应错误。写位清除状态位。值为 1 时清除状态位，值为 0 时保持位不变。无论是否是中断掩码状态，都会记录位。	0
2	CDONE	指令完成。写位清除状态位。值为 1 时清除状态位，值为 0 时保持位不变。无论是否是中断掩码状态，都会记录位。	0
3	DTO	数据传输结束。写位清除状态位。值为 1 时清除状态位，值为0时保持位不变。无论是否是中断掩码状态，都会记录位。	0
4	TXDR	发送 FIFO 数据请求。写位清除状态位。值为 1 时清除状态位，值为 0 时保持位不变。无论是否是中断掩码状态，都会记录位。	0
5	RXDR	接收 FIFO 数据请求。写位清除状态位。值为 1 时清除状态位，值为 0 时保持位不变。无论是否是中断掩码状态，都会记录位。	0
6	RCRC	响应 CRC 错误。写位清除状态位。值为 1 时清除状态位，值为 0 时保持位不变。无论是否是中断掩码状态，都会记录位。	0
7	DCRC	数据 CRC 错误。写位清除状态位。值为 1 时清除状态位，值为 0 时保持位不变。无论是否是中断掩码状态，都会记录位。	0
8	RTO_BAR	响应超时 (RTO)/ 启动接收的应答 (BAR)。写位清除状态位。值为 1 时清除状态位，值为 0 时保持位不变。无论是否是中断掩码状态，都会记录位。	0
9	DRTO_BDS	数据读取超时 (DRTO)/ 启动数据开始 (BDS)。写位清除状态位。值为 1 时清除状态位，值为 0 时保持位不变。无论是否是中断掩码状态，都会记录位。	0
10	HTO	“主机造成的数据不足”超时 (HTO)。写位清除状态位。值为 1 时清除状态位，值为 0 时保持位不变。无论是否是中断掩码状态，都会记录位。 /Volt_switch_int	0
11	FRUN	FIFO 上溢 / 下溢错误。写位清除状态位。值为 1 时清除状态位，值为 0 时保持位不变。无论是否是中断掩码状态，都会记录位。	0
12	HLE	硬件锁定写错误。写位清除状态位。值为 1 时清除状态位，值为 0 时保持位不变。无论是否是中断掩码状态，都会记录位。	0
13	SBE	起始位错误。写位清除状态位。值为 1 时清除状态位，值为 0 时保持位不变。无论是否是中断掩码状态，都会记录位。	0
14	ACD	自动指令完成。写位清除状态位。值为 1 时清除状态位，值为0时保持位不变。无论是否是中断掩码状态，都会记录位。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 304. 原始中断状态寄存器 （RINTSTS，地址 0x4000 4044）位描述 *（续）*

位	符号	描述	复位值
15	EBE	结束位错误（读）/ 未写 CRC。写位清除状态位。值为 1 时清除状态位，值为 0 时保持位不变。无论是否是中断掩码状态，都会记录位。	0
16	SDIO_INTERRUPT	SDIO 卡中断。写位可以清除位。值为 1 时清除位，值为 0 时保持不变。 0 – 无 SDIO 卡中断 1 – 有 SDIO 卡中断 在仅 MMC Ver3.3 模式下，位始终为 0。无论是否是中断掩码状态，都会记录位。	0
31:17	-	保留。	-

20.6.19 状态寄存器 (STATUS)

表 305. 状态寄存器 （STATUS，地址 0x4000 4048）位描述

位	符号	描述	复位值
0	FIFO_RX_WATERMARK	FIFO 达到接收水印级；不适用于数据。	0
1	FIFO_TX_WATERMARK	FIFO 达到发送水印级；不适用于数据传输。	1
2	FIFO_EMPTY	FIFO 状态为空。	1
3	FIFO_FULL	FIFO 状态为满。	0
7:4	CMDFSMSTATES	指令 FSM 状态： 0 – 空闲 1 – 发送 init 序列 2 - Tx cmd 起始位 3 - Tx cmd tx 位 4 - Tx cmd index + arg 5 - Tx cmd crc7 6 - Tx cmd 结束位 7 - Rx resp 起始位 8 - Rx resp IRQ 响应 9 - Rx resp tx 位 10 - Rx resp cmd idx 11 - Rx resp 数据 12 - Rx resp crc7 13 - Rx resp 结束位 14 – Cmd 路径等待 NCC 15 – 等待； CMD 到响应周转期 注：指令 FSM 状态用 19 个位表示。STATUS 寄存器 (7:4) 有 4 个位可以表示指令 FSM 状态。使用这 4 个位，仅可表示 16 种状态。因此，在 STATUS(7:4) 寄存器中有 3 种状态无法表示。在 STATUS 寄存器 (7:4) 中无法表示的 3 种状态是： - 位 16 – 等待 CCS - 位 17 – 发送 CCSD - 位 18 – 启动模式 正因为如此，当指令 FSM 处于等待 CCS 状态或发送 CCSD 或启动模式时，状态寄存器会将位字段 7:4 的状态指示为 0。	0
8	DATA_3_STATUS	原始选择的 card_data[3]；检查卡是否存在 0 – 卡不存在 1 – 卡存在	
9	DATA_BUSY	原始选择的 card_data[0] 的反向 0 – 卡数据不忙 1 – 卡数据忙	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 305. 状态寄存器 （STATUS，地址 0x4000 4048）位描述 *（续）*

位	符号	描述	复位值
10	DATA_STATE_MC_BUSY	数据发送或接收状态机忙。	1
16:11	RESPONSE_INDEX	前个响应的索引，包括内核发送的任何自动停止指令。	0
29:17	FIFO_COUNT	FIFO 计数 – FIFO 中的填充位置数。	0
30	DMA_ACK	DMA 应答信号状态：dw_dma_ack 或 ge_dma_ack，取决于 DW-DMA 或通用 DMA 的选择。	0
31	DMA_REQ	DMA 请求信号状态； dw_dma_req 或 ge_dma_req，取决于 DW-DMA 或通用 DMA 的选择。	0

20.6.20 FIFO 阈值水印寄存器 (FIFOTH)

表 306. FIFO 阈值水印寄存器 （FIFOTH，地址 0x4000 404C）位描述

位	符号	值	描述	复位值
11:0	TX_WMARK		向卡发送数据时的 FIFO 阈值水印级。 FIFO 数据计数少于或等于该数值时，提出 DMA/FIFO 请求。如果使能中断，则发生中断。在数据包末尾时，产生请求或中断，无论是否为编程的阈值。在非 DMA 模式下，当发送 FIFO 阈值 (TXDR) 中断使能，则产生中断，而不是 DMA 请求。在数据包末尾的最后一次中断时，主机应使用仅要求的剩余字节填充 FIFO （并非在 FIFO 满之前或 CIU 完成数据传输后，因为 FIFO 可能不为空）。在 DMA 模式下，数据包末尾时，如果最后一个传输小于连发大小，DMA 控制器进行单个周期，直至所需字节传输完成。12 个位 – 比 13 个位的状态寄存器的 FIFO 计数少 1 个位。限制：TX_WMark >= 1 ；建议：FIFO_DEPTH/2 ；（表示少于或等于 FIFO_DEPTH/2）。	
15:12	-		保留。	
27:16	RX_WMARK		向卡接收数据时的 FIFO 阈值水印级。 FIFO 数据计数大于该数值时，提出 DMA/FIFO 请求。数据包末尾时，无法是否为编程的阈值，产生请求并完成剩余的任何数据。在非 DMA 模式下，当接收 FIFO 阈值 (RXDR) 中断使能，则产生中断，而不是 DMA 请求。数据包末尾时，如果编程的阈值大于任何剩余的数据，则不产生中断。主机发现数据传输完成中断时，应当读取剩余字节。在 DMA 模式下，数据包末尾时，即使剩余字节小于阈值，在数据传输完成中断置位前， DMA 请求只进行单次传输，以找出所有剩余字节。12 个位 – 比 13 个位的状态寄存器的 FIFO 计数少 1 个位。限制：RX_Wmark 小于 FIFO_DEPTH-2 建议：(FIFO_DEPTH/2) – 1 ；（表示大于 (FIFO_DEPTH/2) – 1）。注：在 DMA 模式下且 CCS 超时期间，数据包末尾时 DMA 不产生请求，即使剩余字节小于阈值。这种情况下，会有一些数据遗留在 FIFO 中。应用程序应在 CCS 超时后复位 FIFO。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 306. FIFO 阈值水印寄存器 （FIFOTH，地址 0x4000 404C）位描述 *（续）*

位	符号	值	描述	复位值
30:28	DW_DMA_MUTIPL E _ TRANSACTION_ SIZE		多个传输的连发大小；应当编程为 DW-DMA 控制器多传输大小 SRC/DEST_MSIZE。传输单位为 H_DATA_WIDTH 的参数。单次传输（在非 DW DMA 接口的情况下， dw_dma_single 发出）是根据该值发出信号的。值应该是 (RX_WMark + 1)* (F_DATA_WIDTH/H_DATA_WIDTH) 和 (FIFO_DEPTH - TX_WMark)* (F_DATA_WIDTH/ H_DATA_WIDTH) 的公约数 例如，如果 FIFO_DEPTH = 16，FDATA_WIDTH == H_DATA_WIDTH MSize 和 TX_WMark 的可行组合有： MSize = 1， TX_WMARK = 1-15 MSize = 4， TX_WMark = 8 MSize = 4， TX_WMark = 4 MSize = 4， TX_WMark = 12 MSize = 8， TX_WMark = 8 MSize = 8， TX_WMark = 4。 MSize 和 RX_WMark 的可行组合有： MSize = 1， RX_WMARK = 0-14 MSize = 4， RX_WMark = 3 MSize = 4， RX_WMark = 7 MSize = 4， RX_WMark = 11 MSize = 8， RX_WMark = 7 MSize = 8， RX_WMark = 11 建议：MSize = 8， TX_WMark = 8， RX_WMark = 7。	0
		0x0	1 次传输	
		0x1	4 次传输	
		0x2	8 次传输	
		0x3	16 次传输	
		0x4	32 次传输	
		0x5	64 次传输	
		0x6	128 次传输	
		0x7	256 次传输	
31	-		保留。	

20.6.21 卡检测寄存器 (CDETECT)

表 307. 卡检测寄存器 （CDETECT，地址 0x4000 4050）位描述

位	符号	描述	复位值
0	CARD_DETECT	卡检测。 0 代表有卡。	0
31:1	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

20.6.22 写保护寄存器 (WRTPRT)

表 308. 写保护寄存器 （WRTPRT，地址 0x4000 4054）位描述

位	符号	描述	复位值
0	WRITE_PROTECT	写保护。 1 代表写保护。	0
31:1	-	保留。	-

20.6.23 CIU 卡已传输字节计数寄存器 (TCBCNT)

表 309. CIU 卡已传输字节计数寄存器 （TCBCNT，地址 0x4000 405C）位描述

位	符号	描述	复位值
31:0	TRANS_CARD_BYTE_COUNT	CIU 单元传输到卡中的字节数。 当 AREA_OPTIMIZED 参数为 1 时，仅在数据传输结束后才可读取寄存器；在数据传输期间，寄存器返回 0。	0

20.6.24 主机至 BIU-FIFO 已传输字节计数寄存器 (TBBCNT)

表 310. 主机至 BIU-FIFO 已传输字节计数寄存器 （TBBCNT，地址 0x4000 4060）位描述

位	符号	描述	复位值
31:0	TRANS_FIFO_BYTE_COUNT	主机 /DMA 存储器和 BIU FIFO 之间传输的字节数。	0

20.6.25 去抖动计数寄存器 (DEBNCE)

表 311. 去抖动计数寄存器 （DEBNCE，地址 0x4000 4064）位描述

位	符号	描述	复位值
23:0	DEBOUNCE_COUNT	去抖动过滤逻辑使用的主机时钟 (clk) 数；通常，去抖动时间为 5-25 ms。	0xFFFFFF
31:24	-	保留。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

20.6.26 UHS-1 寄存器 (UHS_REG)

表 312. UHS-1 寄存器 （UHS_REG，地址 0x4000 4074）位描述

位	符号	描述	复位值
0	VOLT_REG	高压模式：确定由外部电压调节器提供给缓冲的电压。 0 - 向缓冲提供 3.3V Vdd 1 - 向缓冲提供 1.8V Vdd 这些位可作为主机控制器的输出，馈入外部电压调节器。电压调节器必须根据寄存器中确定的值，将某张特定卡的缓冲电压切换至 3.3V 或 1.8V。对于卡号 0， VOLT_REG[0] 应设为 1，这样就可以 1.8V 运行。	0
15:1	-	保留。	-
16	DDR_REG	DDR 模式。确定由外部电压调节器提供给缓冲的电压。 0 – 非 DDR 模式 1 – DDR 模式 应当设置 UHS_REG [16]。	0
31:17	-	保留。	-

20.6.27 硬件复位 (RST_N)

表 313. 硬件复位 （RST_N，地址 0x4000 4078）位描述

位	符号	描述	复位值
0	CARD_RESET	硬件复位。 1 – 有效模式 0 - 复位 这些位会让卡进入预空闲状态，需要重新进行初始化。	1
31:1	-	保留。	

20.6.28 总线模式寄存器 (BMOD)

表 314. 总线模式寄存器 （BMOD，地址 0x4000 4080）位描述

位	符号	值	描述	复位值
0	SWR		软件复位。置位后， DMA 控制器复位所有的内部寄存器。 SWR 为可 R/W。 1 个时钟周期后会自动清除。	0
1	FB		固定连发。控制 AHB 主机接口是否执行固定连发传输。置位后，AHB 将在正常连发传输的开始阶段仅使用 SINGLE、INCR4、INCR8 或 INCR16。复位后， AHB 将使用 SINGLE 和 INCR 连发传输操作。 FB 为可 R/W。	0
6:2	DSL		描述符跳过长度。指定两个未连接的描述符之间跳过的半字 / 字 / 双字数。仅适用双缓冲结构。 DSL 为可 R/W。	0
7	DE		SD/MMC DMA 使能。设置后， SD/MMC DMA 使能。 DE 为可 R/W。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 314. 总线模式寄存器 （BMOD，地址 0x4000 4080）位描述 *（续）*

位	符号	值	描述	复位值
10:8	PBL		可编程的连发长度。这些位指示某次 SD/MMC DMA 事务中要执行的最大拍数。每次在主机总线上开始一个连发传输时，SD/MMC DMA 都会尝试 PBL 中指定的连发。允许的值为 1、4、8、16、32、64、128 和 256。该值是 FIFOTH 寄存器中 MSIZE 的映像。如需更改该值，向 FIFOTH 寄存器中写入所需的值。这是一个编码值，如下所示。根据HDATA_WIDTH，传输单位可以是16、32或64位。PBL是一个只读值。	0
		0x0	1 次传输	
		0x1	4 次传输	
		0x2	8 次传输	
		0x3	16 次传输	
		0x4	32 次传输	
		0x5	64 次传输	
		0x6	128 次传输	
		0x7	256 次传输	
31:11	-		保留。	

20.6.29 查询需求寄存器 (PLDMND)

表 315. 查询需求寄存器 （PLDMND，地址 0x4000 4084）位描述

位	符号	描述	复位值
31:0	PD	查询需求。如果某个指示符的 OWN 位未置位， FSM 变为挂起状态。主机需要向该寄存器写入任何值，以便 SD/MMC DMA 状态机恢复正常描述符获取操作。这是一个只写寄存器。PD 位为只写。	

20.6.30 描述符表基地址寄存器 (DBADDR)

表 316. 描述符表基地址寄存器 （DBADDR，地址 0x4000 4088）位描述

位	符号	描述	复位值
31:0	SDL	指示符表的起始。包含第一个指示符的基址。16/32/64 位总线宽度的 LSB 位 [0/1/2:0] 被忽略，SD/MMC DMA 内部会将其全视为 0。因此，这些 LSB 位为只读。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

20.6.31 内部 DMAC 状态寄存器 (IDSTS)

表 317. 内部 DMAC 状态寄存器 （IDSTS，地址 0x4000 408C）位描述

位	符号	描述	复位值
0	TI	发送中断。指示某个指示符的数据传输完成。写 1 可以清除该位。	0
1	RI	接收中断。指示某个指示符的数据接收完成。写 1 可以清除该位。	0
2	FBE	严重总线错误中断。指示发生总线错误(IDSTS[12:10])。该位置位后，DMA 禁用所有总线访问。写 1 可以清除该位。	0
3	-	保留。	
4	DU	指示符不可用中断。由于 OWN 位 = 0 (DES0[31] =0)，指示符不可用，置位该位。写 1 可以清除该位。	0
5	CES	卡错误汇总。指示发送自 / 至卡的传输状态； RINTSTS 中也存在。指示下列位的逻辑或： EBE - 结束位错误 RTO - 响应超时 / 启动应答超时 RCRC - 应答 CRC SBE - 起始位错误 DRTO - 数据读取超时 /BDS 超时 DCRC - 用于接收的数据 CRC RE - 响应错误 写 1 可以清除该位。	0
7:6	-	保留。	
8	NIS	正常中断汇总。下列各项的逻辑或：IDSTS[0] – 发送中断 IDSTS[1] – 接收中断 仅未掩码的位会影响该位。这是一个粘滞位，每次引起 NIS 置位的对应位清除时，该位也必须清除。写 1 可以清除该位。	0
9	AIS	异常中断汇总。下列各项的逻辑或：IDSTS[2] – 严重总线中断 IDSTS[4] – DU 位中断 IDSTS[5] – 卡错误汇总中断 仅未掩码的位会影响该位。这是一个粘滞位，每次引起 AIS 置位的对应位清除时，该位也必须清除。写 1 可以清除该位。	0
12:10	EB	错误位。指示引起总线错误的错误类型。仅当严重总线错误位 (IDSTS[2]) 置位时有效。该字段不产生中断。001 – 发送时收到主机中止 010 – 接收时收到主机中止 其他：保留的 EB 为只读。	0
16:13	FSM	DMAC 状态机表示状态。 0 - DMA_IDLE 1 - DMA_SUSPEND 2 - DESC_RD 3 - DESC_CHK 4 - DMA_RD_REQ_WAIT 5 - DMA_WR_REQ_WAIT 6 - DMA_RD 7 - DMA_WR 8 - DESC_CLOSE 该位为只读位。	0
31:17	-	保留。	

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

20.6.32 内部 DMAC 中断使能寄存器 (IDINTEN)

表 318. 内部 DMAC 中断使能寄存器 (IDINTEN，地址 0x4000 4090）位描述

位	符号	描述	复位值
0	TI	发送中断使能。置位且正常中断汇总使能时，使能发送中断。复位时，禁用发送中断。	0
1	RI	接收中断使能。置位且正常中断汇总使能时，使能接收中断。复位时，禁用接收中断。	0
2	FBE	严重总线错误使能。置位且异常中断汇总使能时，使能严重总线错误中断。复位时，禁用严重总线错误使能中断。	0
3	-	保留。	
4	DU	指示符不可用中断。置位且异常中断汇总使能时，使能 DU 中断。	0
5	CES	卡错误汇总中断使能。置位时，使能卡中断汇总。	0
7:6	-	保留。	
8	NIS	正常中断汇总使能。置位时，使能正常中断。复位时，禁用正常中断。该位使能以下位：IDINTEN[0] – 发送中断 IDINTEN[1] – 接收中断。	0
9	AIS	异常中断汇总使能。置位时，使能异常中断。该位使能以下位：IDINTEN[2] – 严重总线错误中断 IDINTEN[4] – DU 中断 IDINTEN[5] – 卡错误汇总中断。	0
31:10	-	保留。	

20.6.33 当前主机描述符地址寄存器 (DSCADDR)

表 319. 当前主机描述符地址寄存器 （DSCADDR，地址 0x4000 4094）位描述

位	符号	描述	复位值
31:0	HDA	主机描述符地址指针。复位时清除。操作期间， IDMAC 更新指针。该寄存器指向 SD/MMC DMA 读取的当前指示符的起始地址。	0

20.6.34 当前缓冲描述符地址寄存器 (BUFADDR)

表 320. 当前缓冲描述符地址寄存器 （BUFADDR，地址 0x4000 4098）位描述

位	符号	描述	复位值
31:0	HBA	主机缓冲地址指针。复位时清除。操作期间， IDMAC 更新指针。该寄存器指向 SD/MMC DMA 正在访问的当前数据缓冲地址。	0

20.7 功能说明

20.7.1 自动停止。

自动停止指令有助于使用流读取或写入为 MMC 发送准确的数据字节数，对于 SD 卡则利用针对 SD 内存传输的多块读取或写入。该模块内部生成停止指令，并在指令寄存器中设置 SEND_AUTO_STOP 位时加载到指令路径中。

软件应根据下表中列出的详细信息设置 SEND_AUTO_STOP 位：

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 321. SEND_AUTO_STOP 位

卡类型	传输类型	字节计数	SEND_AUTO_STOP 位设置	注释
MMC	流读取	0	无	开放式流
MMC	流读取	>0	有	所有字节传输后自动停止
MMC	流读取	0	无	开放式流
MMC	流读取	>0	有	所有字节传输后自动停止
MMC	单块读取	>0	无	字节数 = 0 非法
MMC	单块写入	>0	无	字节数 = 0 非法
MMC	多块读取	0	无	开放式多块
MMC	多块读取	>0	是 [1]	预定义多块
MMC	多块写入	0	无	开放式多块
MMC	多块写入	>0	是 [1]	预定义多块
SDMEM	单块读取	>0	无	字节数 = 0 非法
SDMEM	单块写入	>0	无	字节数 = 0 非法
SDMEM	多块读取	0	无	开放式多块
SDMEM	多块读取	>0	有	所有字节传输后自动停止
SDMEM	多块写入	0	无	开放式多块
SDMEM	多块写入	>0	有	所有字节传输后自动停止
SDIO	单块读取	>0	无	字节数 = 0 非法
SDIO	单块写入	>0	无	字节数 = 0 非法
SDIO	多块读取	0	无	开放式多块
SDIO	多块读取	>0	无	预定义多块
SDIO	多块写入	0	无	开放式多块
SDIO	多块写入	>0	无	预定义多块

[1] 对于 MMC 和 SD 卡，传输模式设置为块传输和 byte_count 等于块大小的条件将视为单块数据传输指令。如果 byte_count = n × block_size (n = 2, 3, ?，该条件将视为预设的多块数据传输指令。对于 MMC 卡， CPU 软件可以两种方式执行预设的数据传输：1) 在发布 CMD18/CMD25 指令前，向卡发布 CMD23 指令 – 在此情况下，发布 CMD18/CMD25 指令，不设置 send_auto_stop 位。 2) 在设置 send_auto_stop 位的情况下，向卡发布 CMD18/CMD25 指令，不发布 CMD23 指令。在此情况下，多块数据传输在编程的字节计数后，将由内部生成的自动停止指令而终止。

下表列出了自动停止指令的条件。

- MMC 卡的流读取，字节计数大于 0 - 模块生成内部停止指令并将其加载到指令路径，以便在从卡中读取数据的最后一个字节后发送停止指令的结束位，而不再接收额外的数据字节。如果字节计数小于 6（48 位），在发送停止指令的结束位前会再从卡接收几个额外的数据字节。
- MMC 卡的流写入，字节计数大于 0 - 模块生成内部停止指令并将其加载到指令路径，以便在将数据的最后一个字节传输到卡总线后发送停止指令的结束位，而不再发送额外的数据字节。如果字节计数小于 6（48 位），数据路径将发送最后的数据以便满足上述条件。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

- SD 卡的多块读取内存，字节计数大于 0 - 如果块大小小于 4（单位数据总线）、16（4 位数据总线）或 32（8 位数据总线），将在读取所有字节后将自动停止指令加载到指令路径中。否则，在指令路径中加载停止指令，以便在收到最后一个数据块后发送停止指令的结束位。
- SD 卡的多块写入内存，字节计数大于 0 - 如果块大小小于 3（单位数据总线）、12（4 位数据总线）或 24（8 位数据总线），将在发送所有数据块后将自动停止指令加载到指令路径中。否则，在指令路径中加载第停止指令，以便在收到 CRC 状态的结束位发送停止指令的结束位。
- 自动停止期间CPU软件预防措施 - 无论何时发布自动停止指令，在模块发送自动停止且数据传输完成之前，CPU 软件都不应向模块发布新的指令。如果 CPU 在自动停止指令进行数据传输期间发布了新的指令，可能会在新指令发送并收到其响应后才发送自动停止指令，由此可能导致停止指令发送延迟而传输额外的数据字节。对于流写入，额外的数据字节属于可能损坏卡数据的错误数据。如果 CPU 希望在数据传输完成前终止数据传输，可以发布停止或中止指令，在这种情况下模块不会生成自动停止指令。

20.7.2 软件 / 硬件限制

一次只应发布一个数据传输指令。对于CE-ATA设备，如果CE-ATA设备中断使能(nIEN=0)，只应发布一个 RW_MULTIPLE_BLOCK 指令 (RW_BLK)；在输出传输之前，不应再发布任何其他指令（包括新的 RW_BLK）。对未执行的 RW_BLK 设置结束状态。

发布新的数据传输指令之前，软件应确保卡未被任何之前的数据传输指令占用。更改卡时钟频率之前，软件必须确保没有正在进行的数据或指令传输。

为避免卡时钟输出 (cclk_out) 中出现干扰，软件应采用以下步骤来更改卡时钟频率：

1. 更新“时钟使能”寄存器，禁用所有时钟。为确保所有之前的指令在更新前均已完成，通过进行以下设置向 CIU 发送指令以更新时钟寄存器：
   - start_cmd 位
   - “仅更新时钟寄存器”位
   - “wait_previous 数据完成”位

   通过将 start_cmd 位设置为 0，等待 CIU 执行指令。
2. 设置 start_cmd 位以更新时钟分频器和 / 或时钟源寄存器，向 CIU 发送指令以更新时钟寄存器，等待 CIU 执行指令。
3. 设置 start_cmd 更新时钟使能寄存器以使能所需的时钟，向 CIU 发送指令以更新时钟寄存器，等待 CIU 执行指令。

在非DMA模式下从卡中读取时，会在卡中数据传输结束后很快出现数据传输结束(RINTSTS[3])中断。如果 FIFO 中可能还留有一些数据，RX_WMark 中断可能会出现，也可能不会，具体取决于 FIFO 中剩余的字节数。在看到数据传输结束 (DTO) 中断后，软件应立即读取任何剩余的字节。在 DMA 模式下从卡中读取时，仅在所有 FIFO 数据都通过 DMA 接口单元涌入内存后才会出现 DTO 中断。

在 DMA 模式下向卡中写入时，如果通过将字节计数寄存器设置为 0 而选择了长度未定义的传输，则很可能 DMA 逻辑请求的数据会比向卡中发送的数据多，因为它无法了解软件会在何时停止传输。 DMA 会在 CIU 设置 DTO 后立即请求停止。

如果软件通过将控制寄存器位 [0] 设置为 1 发布了 controller_reset 指令，则所有 CIU 状态机会复位，不清除 FIFO。 DMA 会将剩余的所有字节发送到 CPU。除卡复位外，如果还发布了 FIFO 复位，则会：

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- 正确完成总线上所有待定的 DMA 传输
- 忽略 DMA 数据读取
- 写入数据未知 (x)

此外，如果还发布了 dma_reset，所有待定的 DMA 传输会突然终止。还应对 DMA 控制器通道复位和重新编程。

如果有任何之前的数据指令未正确终止，软件应发布 FIFO 复位指令以便删除 FIFO 中的任何残余数据 （如有）。断言 FIFO 复位后，应等待此位清除完成。

FIFO 和 CPU 之间数据传输的一个要求是，传输量应是 FIFO 数据宽度 (F_DATA_WIDTH)（即 32）的倍数。因此，如果您只希望向 SD/MMC/CE-ATA 卡 (BYTCNT) 写入 15 字节，则 CPU 应向 DMA 写入 16 字节或对 DMA 编程以执行 16 字节的传输 （如果 DMA 模式使能）。软件仍然可以将字节计数寄存器仅设置为 15，此时将只向卡中传输 15 字节。同理，当从卡中读取 15 字节时， CPU 也会从 FIFO 中读取全部 16 字节。

建议您不要在数据传输过程中更改 FIFO 阈值寄存器。

20.7.3 编程序列

20.7.3.1 初始化

在功耗和时钟稳定后，为 clk 或 cclk_in（选较慢者）断言的 reset_n 应至少为两个时钟。复位将初始化寄存器、端口、FIFO 指示器、DMA 接口控制和设计中的状态机。开启复位后，软件应执行以下操作：

1. 开启复位后，使用 Syscon 模块中的 SFSP 寄存器配置 SD/MMC 引脚 (表 117)。
2. 通过清除中断掩码寄存器 @0x024 中相应的位设置中断掩码。设置控制寄存器 @0x00 的全局 int_enable 位。建议您向原始中断寄存器 @0x044 写入 0xffff_ffff，以便在设置 int_enable 位前清除任何待定中断。
3. 枚举卡协议栈 - 根据卡类型枚举各个卡；有关详细信息，请参阅 “枚举的卡协议栈”。对于枚举，应根据 SD_MMC/CE-ATA 标准将时钟频率限制为 400 KHz。
4. 更改时钟。卡运行的最大频率为 26 MHz （高速模式下最大频率为 52 MHz）。
5. 将其他IP参数（通常对于各个指令无需更改）设置为典型值，例如根据SD_MMC/CE-ATA 规格设置 cclk_out 中的超时值。

   ResponseTimeOut = 0x40

   DataTimeOut = 以下设置中的最大值：

   - (10 × ((TAAC × Fop) + (100 × NSAC))
   - FIFO 空 / 满时的 CPU FIFO 读取 / 写入延迟

   FIFOTH 寄存器 @0x04C 中的 FIFO 阈值 （以字节为单位）通常，该阈值可以设置为 FIFO 深度 (=32) 的一半，即：

   - RX_WMark = (FIFO_DEPTH/2) - 1 ；
   - TX_WMark = FIFO_DEPTH/2
6. 如果软件决定处理 IP 核提供的中断，您应创建另一个线程来处理中断。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

20.7.3.2 枚举的卡协议栈

卡协议栈执行以下操作：

- 枚举所有连接的卡
- 为连接的卡设置 RCA
- 读取卡特定的信息
- 本地存储卡特定的信息

Enumerate_Card_Stack - 枚举连接到模块上的卡。卡的类型包括 MMC、 CE-ATA、 SD 或 SDIO。支持所有 SDIO 卡类型，即 SDIO_IO_ONLY、SDIO_MEM_ONLY 和 SDIO_COMBO 卡。枚举序列包括以下步骤：

1. 检查卡是否已连接。
2. 清除 card_type 寄存器中的位。清除寄存器位以获取 1 位、 4 位或 8 位总线宽度。
3. 识别卡类型，即 SD、 MMC 或 SDIO。
   - 首先发送 CMD5。如果收到响应，则卡为 SDIO。
   - 如果没有响应，发送 ACMD41 ；如果收到响应，则卡为 SD。
   - 否则，卡为 MMC 或 CE-ATA。
4. 根据卡类型枚举卡。

   使用时钟源频率 = 按需传真 （即 400 KHz）和使用以下枚举指令序列：
   - SD 卡 - 发送 CMD0、 ACMD41、 CMD2、 CMD3。
   - SDHC 卡 - 发送 CMD0、 SDCMD8、 ACMD41、 CMD2、 CMD3。
   - SDIO - 发送 CMD5 ；如果功能计数有效，发送 CMD3。有关 SDIO 内存部分，请执行与 SD 卡相同的指令。
   - MMC - 发送 CMD0、 CMD1、 CMD2、 CMD3。
5. 识别 MMC/CE-ATA 设备。
   - 为 CE-ATA 设备选择 ATA 模式。
   - CPU 将通过发送 CMD8 查询 EXT_CSD 寄存器的字节 504 (S_CMD_SET)。如果位 4 设置为 1，则该设备支持 ATA 模式。
   - 如果支持 ATA 模式，则 CPU 可通过设置 EXT_CSD 寄存器扇区 191(CMD_SET) 的 ATA 位 （位 4）选择 ATA 模式，激活 ATA 指令集以供使用。 CPU 使用 SWITCH (CMD6) 指令选择指令集。
   - 当前所选的显示在 EXT_CSD 寄存器的字节 191 中。
   - 如果该设备不支持 ATA 模式，则该设备可能是 MMC 设备或 CE-ATA v1.0 设备。
   - 发送 RW_REG ；如果收到响应且响应数据包含 CE-ATA 签名，则该设备为 CE-ATA 设备。
   - 否则，该设备为 MMC 卡。
6. 枚举后，可以更改卡时钟频率。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

20.7.3.3 时钟编程

时钟编程必须在 CGU 中完成。 cclk_in 必须等于 cclk_out。因此，支持此设置的寄存器必须为：

- CLKDIV @0x08 = 0x0 （时钟分频器旁路）。
- CLKSRC @0x0C = 0x0。
- CLKENA @0x10 =0x0 or 0x1。此寄存器使能或禁用卡的时钟并使能低功耗模式，该模式在卡闲置超过 8 个时钟时将自动停止卡的时钟。

仅在 CMD 寄存器的 start_cmd 位和 Update_clk_regs_only 位设置的情况下，该模块才能加载每个寄存器。成功加载指令后，模块将清除此位，除非模块在序列中已拥有其他指令，在此情况下将出现 HLE （硬件锁定错误）；有关 HLE 的详细信息，请参见 “错误处理”。

软件将查询 start_cmd 和 Update_clk_regs_only 位，还应设置 wait_prvdata_complete 位以确保时钟参数在数据传输期间不会发生变化。请注意，即使设置了 start_cmd 来更新时钟寄存器，该模块在指令完成后也不会发出 command_done 信号。

20.7.3.4 带或不带响应序列的无数据指令

要发送任何无数据指令，软件需要使用相应的参数对 CMD 寄存器 @0x2C 和 CMDARG 寄存器 @0x28 进行编程。利用上述两个寄存器，模块可构建该指令并将其发送到指令总线。模块将通过 RINTSTS 寄存器的错误位反映指令响应中的错误。

收到响应时，无论错误还是有效，模块都会在 RINTSTS 寄存器中设置 command_done 位。短的响应将复制到响应寄存器 0 中，而长的响应将复制到所有四个响应寄存器 @0x30、0x34、0x38和0x3C中。响应3寄存器位31代表MSB，而响应0寄存器位0代表长响应的LSB。

对于基本指令或无数据指令，请执行以下步骤：

1. 利用相应的指令参数对指令寄存器 @0x28 编程。
2. 使用表 322 中的设置对命令寄存器 @0x2C 进行编程。
3. 等待 CPU 接受指令。将指令加载到模块中后，会发生以下情况：
   - 模块接受执行和清除 CMD 寄存器中 start_cmd 位的指令，除非正在执行一项指令，在此情况下，模块可将其加载到缓冲区并将其排为第二个指令。
   - 如果模块无法加载指令，即正在执行一项指令，缓冲区中存在第二个指令，并且正在尝试第三个指令，则会生成 HLE （硬件锁定错误）。
   - 检查是否出现 HLE。
   - 等待指令执行完成。收到卡的响应或响应超时后，模块将设置 RINTSTS 寄存器中的 command_done 位。软件可以等待此位或响应生成的中断。
   - 检查是否已设置 response_timeout 错误、response_CRC 错误或响应错误。通过响应上述错误引发的中断或查询 RINTSTS 寄存器 @0x44 的位 1、位 6 和位 8，可以解决上述问题。如果未收到响应错误，则响应有效。如果需要，软件可以从响应寄存器 @0x30-0x3C 中复制响应。

在指令执行过程中，软件不应修改时钟参数。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 322. CMD 寄存器的无数据指令设置

名称	值	备注
start_cmd	1	
update_clock_ registers_only	0	无时钟参数更新指令。
card_number	0	正在使用的卡数。由于支持一张卡，因此仅有 0 为可能的值。
Data_expected	0	无数据指令。
Send_initialization	0	可为 1，但仅适用于卡复位指令，如 CMD0。
stop_abort_cmd	0	对于停止数据传输的指令，如CMD12，可为1。
Cmd_index	指令索引	
Response_length	0	对于 R2 （长）响应可为 1。
Response_expect	1	对于无响应的指令可为 0 ；例如 CMD0、CMD4、 CMD15 等等。
用户可选		
Wait_prvdata_complete	1	发送命令行的指令之前， CPU 应等待 任何进行中的数据指令完成 （若有） （除非当前指令是查询状态或在传输过程中停止数据传输，否则建议始终设置此位）
Check_response_crc	1	0 - 不检查响应 CRC 1 - 检查响应 CRC 一些指令响应未返回有效的 CRC 位。如需禁用控制器的 CRC 检查，软件应当禁用这些指令的 CRC 检查。

20.7.3.5 数据传输指令

数据传输命令在存储卡和模块间传输数据。要发送数据指令，模块需要指令参数、总数据大小和块大小。软件可以通过 FIFO 接收或发送数据。

发送数据传输指令前，软件应确认卡未被占用并处于传输状态，可分别使用 CMD13 和 CMD7 指令进行上述确认。

对于数据传输指令，务必在卡类型寄存器 @0x18 中对卡中编程的总线宽度设置相同的值。因此，要更改总线宽度，应使用下面提供的适用于卡类型的 API：

- Set_SD_Mode() - SD/SDIO 卡
- Set_HSmodeSettings() - HSMMC 卡

数据传输过程中，模块会针对不同的条件生成中断，并在 RINTSTS 寄存器 @0x44 中作出以下反应：

1. Data_Transfer_Over (bit 3) - 当数据传输结束或终止时。如果出现响应超时错误，模块则不会尝试任何数据传输，并且不会设置 “数据传输结束”位。
2. Transmit_FIFO_Data_request （位 4） - 达到 FIFO 发送数据的阈值；软件 （如果可用）将在 FIFO 中写入数据。
3. Receive_FIFO_Data_request（位 5）- 达到 FIFO 接收数据的阈值；软件将从 FIFO 中读取数据。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

4. CPU 超时造成的数据不足（位 10）- FIFO 在传输过程中为空或在接收过程中已满。软件只有在空的条件下写入数据或在满的条件下读取数据，才能继续进行数据传输。卡的时钟已停止。
5. 数据读取超时错误 （位 9） - 卡在超时期间内未发送数据。
6. 数据 CRC 错误 （位 7） - 数据接收过程中出现 CRC 错误。
7. 起始位错误 （位 13） - 数据接收过程中未收到起始位。
8. 结束位错误 （位 15） - 数据接收或写入操作过程中未收到结束位；卡指示 CRC 错误。

条件 6、7 和 8 表示接收的数据可能存在错误。如果出现响应超时，则未进行数据传输。

20.7.3.6 单块或多块读取

单块或多块读取操作的步骤为：

1. 在 BYTCNT 寄存器 @0x20 中写入数据大小 （以字节为单位）。
2. 在 BLKSIZ 寄存器 @0x1C 中写入块大小 （以字节为单位）。模块希望卡中每个数据块的大小为 BLKSIZ。
3. 使用数据读取一开始的数据地址为 CMDARG 寄存器 @0x28 编程。使用表 323 中列出的参数为命令寄存器编程。对于 SD 和 MMC 卡，单模块读取时，使用 CMD17 ；多模块读取时，使用 CMD18。对于 SDIO 卡，将 CMD53 同时用于单块和多块传输。

   写入 CMD 寄存器后，模块将开始执行指令；指令发送到总线后，将生成 command_done 中断。
4. 软件将查询数据错误中断，即 RINTSTS 寄存器的位 7、9、13 和 15。如果需要，软件可以通过发送 STOP 指令终止数据传输。
5. 软件应查询 Receive_FIFO_Data_request 和 / 或 CPU 超时造成的数据不足条件。在这两种情况下，软件将从 FIFO 中读取数据，留出 FIFO 中的空间以接收更多数据。
6. 收到 Data_Transfer_Over 中断后，软件将从 FIFO 中读取剩余的数据。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 323. 单块或多块读取操作的 CMD 寄存器设置

名称	值	备注
start_cmd	1	
update_clock_ registers_only	0	无时钟参数更新指令。
card_number	0	正在使用的卡数。由于支持一张卡，因此仅有 0 为可能的值。
Data_expected	1	
Send_initialization	0	可为 1，但仅适用于卡复位指令，如 CMD0。
stop_abort_cmd	0	对于停止数据传输的指令，如 CMD12，可为 1。
Send_auto_stop	0/1	根据 < 待定 > 进行设置
Transfer_mode	0	模块传输
Read_write	0	读卡
Cmd_index	指令索引	
Response_length	0	对于 R2 （长）响应可为 1。
Response_expect	1	对于无响应的指令可为0；例如CMD0、CMD4、 CMD15 等等。
用户可选		
Wait_prvdata_complete	1	发送命令行的指令之前， CPU 应等待任何进行中的数据指令完成 （若有）（除非当前指令是查询状态或在传输过程中停止数据传输，否则建议始终设置此位）
Check_response_crc	1	0 - 不检查响应 CRC 1 - 检查响应 CRC 一些指令响应未返回有效的 CRC 位。如需禁用控制器的 CRC 检查，软件应当禁用这些指令的 CRC 检查。

20.7.3.7 单块或多块写入

单块或多块写入操作的步骤为：

1. 在 BYTCNT 寄存器 @0x20 中写入数据大小 （以字节为单位）。
2. 在 BLKSIZ 寄存器 @0x1C 中写入块大小（以字节为单位）；模块发送的每个数据块大小为 BLKSIZ。
3. 在 CMDARG 寄存器 @0x28 中编程数据要写入的数据地址。
4. 在 FIFO 中写入数据；通常最好向 FIFO 的完整深度填充数据。
5. 使用表 324 中列出的参数为命令寄存器编程。对于 SD 和 MMC 卡，单模块写入时，使用 CMD24 ；多模块写入时，使用 CMD25。对于 SDIO 卡，将 CMD53 同时用于单块和多块传输。

   写入 CMD 寄存器后，模块将开始执行指令；指令发送到总线后，将生成 command_done 中断。
6. 软件将查询数据错误中断，即 RINTSTS 寄存器的位 7、9 和 15。如果需要，软件可以通过发送 STOP 指令终止数据传输。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

7. 软件应查询 Transmit_FIFO_Data_request 和 / 或 CPU 数据不足造成超时条件。在上述两种情况下，软件都将向 FIFO 写入数据。
8. 收到 Data_Transfer_Over 中断后，数据指令即结束。对于开放式块传输，如果字节计数为 0，则软件必须发送 STOP 指令。如果字节计数不是 0，在完成特定字节数的传输后，模块将根据需要发送STOP指令。AUTO-STOP指令完成可以通过Auto_command_done 中断 （即 RINTSTS 寄存器的位 14）体现。对 AUTO_STOP 的响应将存储在 RESP1 @0x34 中。

表 324. 单块或多块写入操作的 CMD 寄存器设置

名称	值	注释
start_cmd	1	
update_clock_ registers_only	0	无时钟参数更新指令。
card_number	0	正在使用的卡数。由于支持一张卡，因此仅有 0 为可能的值。
Data_expected	1	
Send_initialization	0	可为 1，但仅适用于卡复位指令，如 CMD0。
stop_abort_cmd	0	对于停止数据传输的指令，如 CMD12，可为 1。
Send_auto_stop	0/1	根据 XXXXXXXXX 进行设置
Transfer_mode	0	模块传输
Read_write	1	写入卡
Cmd_index	指令索引	
Response_length	0	对于 R2 （长）响应可为 1。
Response_expect	1	对于无响应的指令可为0；例如CMD0、CMD4、 CMD15 等等。
用户可选		
Wait_prvdata_complete	1	发送命令行的指令之前， CPU 应等待任何进行中的数据指令完成 （若有） （除非当前指令是查询状态或在传输过程中停止数据传输，否则建议始终设置此位）
Check_response_crc	1	0 - 不检查响应 CRC 1 - 检查响应 CRC 一些指令响应未返回有效的 CRC 位。如需禁用控制器的 CRC 检查，软件应当禁用这些指令的 CRC 检查。

20.7.3.8 流读取

流读取与 “单块或多块读取”中提到的块读取操作类似，但命令寄存器中的以下位除外：

transfer_mode = 1; // 流传输

cmd_index = CMD20 ；

仅单个位总线宽度允许流传输。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

20.7.3.9 流写入

流写入与 “单块或多块写入”中提到的块写入操作极其类似，但指令寄存器中的以下位除外：

- transfer_mode = 1; // 流传输
- cmd_index = CMD11 ;

在流传输中，如果字节计数为 0，则软件必须发送 STOP 指令。如果字节计数不是 0，当特定的字节数完成传输后，模块将发送STOP指令。AUTO_STOP指令完成可通过Auto_command_done中断体现。对 AUTO_STOP 的响应将存储在 RESP1 寄存器 @0x34 中。

仅单个位总线宽度允许流传输。

20.7.3.10 传输过程中发送 Stop 或 Abort

STOP指令可以终止内存卡和模块之间的数据传输，而ABORT指令仅可以终止SDIO_IOONLY和 SDIO_COMBO 卡的 I/O 数据传输。

- 发送STOP 命令 - 可在数据传输进行过程中通过命令行发送；该命令可在数据传输期间的任何时间发送。有关发送该指令的信息，请参阅“带或不带响应序列的无数据命令”。

  您还可使用该命令的额外设置，将命令寄存器位 (5-0) 设为 CMD12，并将位 14 (stop_abort_cmd) 设为 1。如果未将 stop_abort_cmd 设为 1，则用户停止了数据传输。将命令寄存器的位 13(wait_prvdata_complete) 复位为 0，使模块即使在数据传输过程中也可以立刻发送指令。

  - 发送 ABORT 指令 - 仅可用于 SDIO_IOONLY 或 SDIO_COMBO 卡。要中止正在传输数据的函数，可使用 CMD52 对 ASx 位中的函数编号进行编程（卡的 CCCR 寄存器，地址 0x06，位 (0-2)）。

    这是一个非数据命令。有关发送该指令的信息，请参阅 “带或不带响应序列的无数据命令”。

    使用表 325 中列出的适当命令参数对 CMDARG 寄存器 @0x28 进行编程。
  - 使用指令索引 CMD52 对指令寄存器进行编程。与 STOP 指令相同，将指令寄存器的位 14(stop_abort_cmd) 设置为 1，只有这样才能通知模块用户已中止数据传输。将指令寄存器的位 13(wait_prvdata_complete) 复位为 0，使模块即使在数据传输过程中也可以立刻发送指令。
  - 等待 command_transfer_over。
  - 检查响应 (R5) 错误。
- 在开放式卡写入操作中，如果卡时钟由于FIFO为空而停止，软件必须首先为FIFO填充数据，启动卡时钟，然后再向卡发布停止 / 中止命令。

表 325. CMDARG 寄存器的参数

位	内容	值
31	R/W 标志	1
30-28	功能编号	0，用于 CCCR 访问
27	RAW 标志	1 （如果写入后需要读取）
26	不必介意	-
25-9	寄存器地址	0x06
8	不必介意	-
7-0	写入数据	要终止的功能编号

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

20.7.4 挂起或恢复序列

对于 SDIO 卡，可以使用 SUSPEND 指令暂时停止 I/O 功能和模块之间的数据传输；在执行另一函数的高优先级数据传输时可能需要这样做。如果需要，使用 RESUME 指令可以恢复数据传输。

通过对 SDIO 卡的 CCCR 寄存器（函数 0）中的相应位编程，可以执行下列函数。要从 CCCR 寄存器中读取或写入 CCCR 寄存器，请使用 CMD52 指令。

1. SUSPEND 数据传输 - 无数据指令。
   - 检查 SDIO 卡是否支持 SUSPEND/RESUME 协议；通过卡的 CCCR 寄存器 @0x08 的 SBS 位可以执行上述操作。

     检查是否正在执行所需函数编号的数据传输；当前活动的函数编号反映在 CCCR 寄存器 @0x0D 的位 0-3。请注意，如果 BS 位（地址 0xc:: 位 0）为 1，则只有 FSx 位指定的函数编号有效。

     要挂起传输，请设置 CCCR 寄存器 @0x0C 的 BR（位 2）。

     等待查看 CCCR @0x0C 的位 BR（位 1）和 BS（位 0）的状态。当正确选定的函数使用数据总线时，BS（总线状态）位为 1；在总线释放完成前，BR（总线释放）位将保持为 1。当 BR 和 BS 位为 0 时，表示所选函数的数据传输被挂起。

     在读取数据传输过程中，模块可以等待卡中的数据。如果数据传输是从卡中读取数据，则在成功完成 SUSPEND 指令后必须通知模块。之后，模块即可复位数据状态机并退出等待状态。要完成此操作，请设置控制寄存器中的 abort_read_data（位 8）。

     等待数据完成。通过读取 TCBCNT 寄存器 @0x5C，使等待的字节进行传输。
2. RESUME 数据传输 - 这是数据指令。
   - 检查卡是否处于传输状态，从而确认总线中是否有数据传输。

     如果卡处于未连接状态，请使用 CMD7 选择它。可以通过检索卡状态来响应 CMD52/CMD53 指令。

     检查要恢复的函数是否准备好进行数据传输；通过读取 CCCR @0x0F 中的 RFx 标志可以进行确认。如果 RF = 1，则该函数已准备好进行数据传输。

     要恢复传输，应使用 CMD52 在 CCCR 寄存器 @0x0D 的 FSx 位 (0-3) 写入函数编号。形成 CMD52 命令参数并将其写入 CMDARG @0x28；表 326 中列出了位值。
   - 在 BLKSIZ 寄存器 @0x1C 中写入块大小，数据将以该块大小为单位传输。

     在 BYTCNT 寄存器 @0x20 中写入字节计数。这是数据的总大小，即还需要传输剩余的字节。这些数据由软件负责处理。

     对指令寄存器进行编程，具体与块传输相同。有关详细信息，请参阅“单块或多块读取”或“单块或多块写入”。

     对指令寄存器编程后，将发送指令并且函数恢复数据传输。读取 DF 标志（恢复数据标志）。如果值为 1，则函数具有传输的数据，将在该函数或内存恢复后立即开始数据传输。如果值为 0，则该函数没有要传输的数据。

     如果 DF 标志为 0，则在读取时，模块将等待数据。数据超时期间过后，将出现数据超时错误。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 326. CMDARG 寄存器的参数

位	内容	值
31	R/W 标志	1
30-28	功能编号	0，用于 CCCR 访问
27	RAW 标志	1，写入后读取
26	不必介意	-
25-9	寄存器地址	0x0D
8	不必介意	-
7-0	写入数据	要终止的功能编号

20.7.4.1 Read_Wait 序列

Read_wait 仅用于 SDIO 卡，可以暂时停止来自函数或内存的数据传输，并允许 CPU 向 SDIO 设备的任何函数发送指令。 CPU 可以根据需要停止此类传输。模块将向设备提供信号表示停止向卡传输。执行此操作的步骤是：

1. 检查卡是否支持 read_wait 设备；读取 CCCR 寄存器 @0x08 的 SRW （位 2）。如果此位为 1，则卡中的所有函数均支持 read_wait 设备。使用 CMD52 读取此位。
2. 如果卡支持 read_wait 信号，则通过设置 CTRL 寄存器 @0x00 中的 read_wait（位 6）对其断言。
3. 清除 CTRL 寄存器中的 read_wait 位。

20.7.4.2 CE-ATA 数据传输指令

本节介绍 CE-ATA 数据传输指令。有关基本设置和不同条件下生成中断的信息，请参阅“数据传输指令”。

20.7.4.2.1 复位和设备恢复

开始 CE-ATA 操作前， CPU 应执行 MMC 复位和初始化程序。在设备输入 MMC TRAN 状态前， CPU 和设备应协商 MMC TRAN 状态（根据多媒体卡系统规格定义）。CPU 应遵循现有的 MMC 卡枚举程序来协商 MMC。

TRAN 状态。完成正常的 MMC 复位和初始化程序后，CPU 将使用 RW_REG/CMD39 来查询初始 ATA 任务文件值。

默认情况下， MMC 块大小为 512 字节，由 CE-ATA 设备中 srcControl 寄存器的位 1:0 指示。 CPU 可协商使用 1KB 还是 4KB 的 MMC 块大小。设备指示可通过 srcCapabilities 寄存器支持的 MMC 块大小； CPU 将读取此寄存器以协商 MMC 块大小。当 CPU 控制器将 MMC 块大小写入设备的 srcControl 寄存器位 1:0 后，即完成协商。

20.7.4.2.2 ATA 任务文件传输

ATA 任务文件寄存器将映射到 MMC 寄存器空间的地址 0x00h-0x10h。RW_REG 用于发布 ATA 指令，并且 ATA 任务文件将在单一 RW_REG MMC 指令序列中发送。

设置 CMDARG 和 CMD 寄存器前， CPU 软件协议栈会将任务文件映像写入 FIFO。然后，在设置 CMD（偏移 0x2C）寄存器位前，CPU 将在 BIU 寄存器中的 CMDARG（偏移 0x28）设置地址和字节计数。

对于 RW_REG， CE-ATA 设备不会发送指令完成信号。

使用 RW_MULTIPLE_REGISTER (RW_REG) 执行 ATA 任务文件传输。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

该指令包括 CE-ATA 设备和模块间的数据传输。要发送数据指令，模块需要指令参数、总数据大小和块大小。软件可以通过 FIFO 接收或发送数据。

ATA 任务文件传输 （读取或写入）操作的步骤为：

1. 在 BYTCNT 寄存器 @0x20 中写入数据大小 （以字节为单位）。
2. 在 BLKSIZ 寄存器 @0x1C 中写入块大小 （以字节为单位）；模块希望执行单块传输。
3. 在 CMDARG 寄存器 @0x28 中编程开始的寄存器地址。

您应根据下表对 CMDARG、 CMD、 BLKSIZ 和 BYTCNT 寄存器编程。

- 如下所示对指令参数 (CMDARG) 寄存器编程。

表 327. CMDARG 寄存器的参数

位	内容	值
31	R/W 标志	1 （写）或 0 （读）
30-24	保留	0
23:18	读取 / 写入的起始寄存器地址；双字对齐	0
17:16	寄存器地址；双字对齐	0
15-8	保留，CPU 将位清除为 0	0
7:2	要读取 / 写入的字节数；双字整数	16
1:0	双字整数中的字节计数	0

- 如下所示对指令 (CMD) 寄存器编程。

表 328. CMD 寄存器设置

名称	值	备注
start_cmd	1	
Css_expect	0	不发送指令完成信号。
Read_ceata_device	0/1	1 – 如果 RW_BLK 或 RW_REG 读取。
update_clock_ registers_only	0	无时钟参数更新指令。
card_number	0	正在使用的卡数。由于支持一张卡，因此仅有 0 为可能的值。
Data_expected	1	
Send_initialization	0	可为1，但仅适用于卡复位指令，如CMD0。
stop_abort_cmd	0	
Send_auto_stop	0	
Transfer_mode	0	模块传输
Read_write	0/1	0 从卡读取； 1 - 写入卡
Cmd_index	指令索引	
Response_length	0	
Response_expect	1	

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 328. CMD 寄存器设置 （续）

名称	值	备注
用户可选		
Wait_prvdata_complete	1	0 - 立刻发送指令 1 - 之前的数据传输完成后发送指令
Check_response_crc	1	0 - 不检查响应 CRC 1 - 检查响应 CRC

- 如下所示对块大小 (BLKSIZ) 寄存器编程。

表 329. BLKSIZ 寄存器

位	值	备注
31:16	0	保留位为 0 (0)
15:0	16	用于访问整个任务文件 （16、 8 位寄存器）； 16 字节块大小

- 如下所示对字节计数 (BYTCNT) 寄存器编程。

表 330. BYTCNT 寄存器

位	值	备注
31:0	16	用于访问整个任务文件 （16、 8 位寄存器）；字节计数值为 16，用于表示将块大小设置为 16

20.7.4.2.3 使用 RW_MULTIPLE_BLOCK (RW_BLK) 执行 ATA 有效载荷传输

该指令包括 CE-ATA 设备和模块间的数据传输。要发送数据指令，模块需要指令参数、总数据大小和块大小。软件可以通过 FIFO 接收或发送数据。

ATA 有效载荷传输 （读取或写入）操作的步骤为：

1. 在 BYTCNT 寄存器 @0x20 中写入数据大小 （以字节为单位）。
2. 在 BLKSIZ 寄存器 @0x1C 中写入块大小 （以字节为单位）。模块预期执行单块 / 多块传输。
3. 对 CMDARG 寄存器 @0x28 进行编程以指示数据计数单元。

   您应根据下表对 CMDARG、 CMD、 BLKSIZ 和 BYTCNT 寄存器编程。

   - 如下所示对指令参数 (CMDARG) 寄存器编程。

表 331. CMDARG 寄存器的参数

位	内容	值
31	R/W 标志	1 （写）或 0 （读）
30-24	保留	0
23:16	保留	0
15:8	数据计数单元 [15:8]	数据计数
1:0	数据计数单元 [7:0]	数据计数

- 如下所示对指令 (CMD) 寄存器编程。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 332. CMD 寄存器设置

名称	值	备注
start_cmd	1	-
Css_expect	1	预期接收指令完成信号；如果在 CE-ATA 设备中使能了中断 (nIEN = 0)，则设置 RW_BLK。
Read_ceata_device	0/1	1 – 如果 RW_BLK 或 RW_REG 读取。
update_clock_ registers_only	0	无时钟参数更新指令。
card_number	0	正在使用的卡数。由于支持一张卡，因此仅有 0 为可能的值。
Data_expected	1	
Send_initialization	0	可为 1，但仅适用于卡复位指令，如 CMD0。
stop_abort_cmd	0	
Send_auto_stop	0	
Transfer_mode	0	模块传输
Read_write	0/1	0 从卡读取； 1 - 写入卡
Cmd_index	指令索引	
Response_length	0	
Response_expect	1	
用户可选		
Wait_prvdata_complete	1	0 - 立刻发送指令 1 - 之前的数据传输完成后发送指令
Check_response_crc	1	0 - 不检查响应 CRC 1 - 检查响应 CRC

- 如下所示对块大小 (BLKSIZ) 寄存器编程。

表 333. BLKSIZ 寄存器

位	值	备注
31:16	0	保留位为 0 (0)
15:0	512, 1024, 4096	根据 CPU 协商， MMC 块大小可以使 512、1024 或 4096 字节。

- 如下所示对字节计数 (BYTCNT) 寄存器编程。

表 334. BYTCNT 寄存器

位	值	备注
31:0	N ≤ block_size	byte_count应是块大小的整数倍数；对于ATA 媒体访问指令，字节计数应为 4KB 的倍数。（N × block_size = X × 4KB，其中 N 和 X 是整数）

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

20.7.4.2.4 发送指令完成信号禁用

如果未执行的 RW_BLK 需要等待指令完成信号 (CCS)，则 CPU 可以发送指令完成信号禁用 (CCSD)。

- 发送CCSD - 如果CTRL寄存器中设置了send_ccsd位，则模块可以向CE-ATA设备发送CCSD；仅在 RW_BLK 收到响应后，才设置此位。
- 发送内部停止命令 - CCSD 模式发送后，发送内部生成的 STOP (CMD12) 命令。如果在编程控制器以发送 CCSD 模式时还设置了 send_auto_stop_ccsd 位，模块将发送 CMD 行上内部生成的 STOP 命令。发送停止指令后，模块设置 RINTSTS 寄存器中的自动指令完成位。

20.7.4.2.5 指令完成信号超时后进行恢复

如果在等待指令完成信号(CCS)时出现超时，则CPU需要发送指令完成信号禁用(CCSD)，然后再发送 STOP 指令来中止待定的 ATA 指令。 CPU 可以对模块编程，先发送 CCSD 模式，再发送内部生成的 STOP 指令

- 发送 CCSD - 设置 CTRL 寄存器中的 send_ccsd 位。
- 将命令寄存器的位13(wait_prvdata_complete)复位为0，使模块即使在数据传输过程中也可以立刻发送指令。
- 发送内部STOP命令 - 设置CTRL寄存器的send_auto_stop_ccsd位，编程CPU控制器以发送内部生成的 STOP 命令。发送停止指令后，模块设置 RINTSTS 寄存器中的自动指令完成位。

20.7.4.2.6 减少 ATA 指令集

CE-ATA 设备有必要支持减少的 ATA 指令子集。下面将详细讨论此减少的指令集。

- IDENTIFY DEVICE - 向 CPU 返回 512 字节数据结构，描述设备特定的信息和功能。仅在 MMC 块大小设置为 512 字节时， CPU 才会发布 IDENTIFY DEVICE 指令；任何其他 MMC 块大小均具有不确定的结果。

CPU 对 ATA 指令发出 RW_REG，并将通过 RW_BLK 来检索数据。

CPU 控制器在针对 IDENTIFY DEVICE ATA 指令发送 RW_REG 时使用以下设置。下面列出了主位

- CMD 寄存器设置 – data_expected 字段设为 0
- CMDARG 寄存器设置：
  - 位 [31] 设为 0
  - 位 [7:2] 设为 128。
- 任务文件设置：
  - ATA 任务文件的指令字段设为 ECh
  - 任务文件的保留字段清零
- BLKSIZ 寄存器位 [15:0] 和 BYTCNT 寄存器 – 设为 16

  CPU 控制器采用以下设置进行数据检索 (RW_BLK):\
- CMD 寄存器设置：
  - ccs_expect 设为 1
  - data_expected 设为 1

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- CMDARG 寄存器设置：
  - 位 [31] 设为 0 （读取操作）

    数据计数设为 1[16'h0001]
- BLKSIZ 寄存器位 [15:0] 和 BYTCNT 寄存器 – 设为 512，IDENTIFY DEVICE 可能因为 CPU 发出 CMD12 而中止。
  - READ DMA EXT - 使用数据输入数据传输协议从设备读取多个逻辑数据模块。CPU 使用 RW_REG 发出 ATA 指令，使用 RW_BLK 进行数据传输。
  - WRITE DMA EXT - 使用数据输出数据传输协议将多个逻辑数据模块写入设备。CPU 使用 RW_REG 发出 ATA 指令，使用 RW_BLK 进行数据传输。
  - STANDBY IMMEDIATE - 此 ATA 指令预期会发出无数据传输 (RW_BLK)，促使设备立即输入仍然保留内部设备上下文的最激进的功率管理模式。
- CMD 寄存器设置 – data_expected 字段设为 0

  CMDARG 寄存器设置：
  - 位 [31] 设为 1
  - 位 [7:2] 设为 4
- 任务文件设置：
  - ATA 任务文件的指令字段设为 E0h
  - 任务文件的保留字段清零
- BLKSIZ 寄存器位 [15:0] 和 BYTCNT 寄存器 – 设为 16
  - FLUSH CACHE EXT - 此 ATA 指令预期会发出无数据传输。对于缓冲 / 缓存写入数据的设备，FLUSH CACHE EXT 指令可确保将缓冲的数据写入设备媒体。对于不缓冲写入数据的设备，FLUSH CACHE EXT 将返回成功状态。CPU 将针对 ATA 指令发出 RW_REG，并可通过 CMD39/RW_REG 检索状态； ATA 指令可能会出现错误状态，在这种情况下 ATA 任务文件状态之外的其他字段仍然有效。
- CPU 在针对 STANDBY IMMEDIATE ATA 指令发送 RW_REG 时使用以下设置：
- CMD 寄存器设置 – data_expected 字段设为 0
- CMDARG 寄存器设置：
- 位 [31] 设为 1
- 位 [7:2] 设为 4
- 任务文件设置：
- ATA 任务文件的指令字段设为 EAh
- 任务文件的保留字段清零
- BLKSIZ 寄存器位 [15:0] 和 BYTCNT 寄存器 – 设为 16

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

20.7.4.3 控制器 /DMA/FIFO 复位使用

与卡进行通信涉及到以下操作：

- 控制器 - 控制模块的所有功能。
- FIFO - 保存要发送或接收的数据。
- DMA - 如果 DMA 传输模式使能，则在系统内存和 FIFO 之间传输数据。
- 控制器复位 - 通过设置CTRL寄存器的controller_reset位（位0）复位控制器；这将复位 CIU 和状态机，还将复位 BIU 至 CIU 接口。由于该复位位会自动清除，发出复位后，应等到该位清除。
- FIFO复位 - 通过设置CTRL寄存器的fifo_reset位（位1）复位FIFO；这将复位FIFO指针和 FIFO 计数器。由于该复位位会自动清除，发出复位后，应等到该位清除。

  DMA 复位 - 通过设置 CTRL 寄存器的 dma_reset（位 2）复位内部 DMA 控制器逻辑，这将会突然终止进行中的任何 DMA 传输。由于该复位位会自动清除，发出复位后，应等到该位清除。

建议采用以下方法发出复位指令：

- 无DMA传输模式 - 同时设置controller_reset和fifo_reset；使用其他写入清除RAWINTS 寄存器 @0x44，以便清除由此产生的所有中断。
- 通用DMA模式 - 同时设置controller_reset、fifo_reset和dma_reset；使用其他写入清除 RAWINTS 寄存器 @0x44，以便清除由此产生的所有中断。如果需要 “宽限”完成 DMA，则建议在复位 DMA 界面控制和发出其他 FIFO 复位指令前等待状态寄存器，以查看 DMA 请求是否为 0。
- 在DMA传输模式下，即使对FIFO指示器进行了复位，如果有DMA传输正在进行，也可能会向 FIFO 置入或从中弹出数据； DMA 本身正确完成。为了清除 FIFO，软件应发出其他 FIFO 复位指令并清除由于复位 FIFO 导致 RAWINTS 寄存器中产生的所有 FIFO 欠载或溢出错误。

20.7.4.4 错误处理

模块实施错误检查；错误将反映在 RAWINTS 寄存器 @0x44 中并通过中断发送至软件，或者软件可轮询这些位。上电后，中断禁用 （CTRL 寄存器中的 int_enable 为 0），并掩码所有中断 （INTMASK 寄存器的位 0-31，默认为 0）。错误处理：

- 响应和数据超时错误 - 对于响应超时，软件可以重试该指令。对于数据超时，如果模块在超时期间内未收到数据起始位（第一个块或中间的块），软件可以再次重试整个数据传输或从指定的块开始重试。通过稍后读取 TCBCNT 的内容，软件可以确定还需要复制多少字节。
- 响应错误 - 接收响应期间收到错误时设置。在此情况下，响应寄存器中复制的响应无效。软件可以重试指令。
- 数据错误 - 数据接收中发现错误时设置；例如，数据 CRC、未找到起始位、未找到结束位等等。可以对任何块设置这些错误，无论是第一个块、中间块还是最后一个块。收到错误时，软件可以发出 STOP 或 ABORT 指令并对所有数据或部分数据重试指令。
- 硬件锁定错误 - 当模块无法加载软件发出的指令时设置。在软件设置 CMD 寄存器的 start_cmd 位后，模块将尝试加载指令。如果指令缓冲区中已存在指令，将发生此错误。然后，软件将重新加载该指令。
- FIFO欠载/溢出错误 - 如果FIFO已满且软件尝试向FIFO写入数据，则设置溢出错误。相反，如果 FIFO 为空且软件尝试从 FIFO 读取数据，则设置欠载错误。在 FIFO 中读取或写入数据前，软件应读取状态寄存器中的 fifo_empty 或 fifo_full 位。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- CPU 超时造成的数据不足 - 当模块等待软件干预以向 FIFO 或从其向外传输数据时，如果软件在规定的超时期间内未进行传输，则会出现此错误。在此条件下以及正在进行读取传输时，软件
- 应从 FIFO 读取数据并创建空间以供接收更多数据。在进行发送操作时，软件应向 FIFO 中填充数据以便开始向卡传输数据。
- 指令CRC错误 - 如果检测到指令出现CRC错误，CE-ATA设备将不会发送响应，模块则会出现响应超时。系统将会通知 ATA 层， MMC 传输层出现错误。
- 写入操作 - 设备已知的任何 MMC 传输层错误都将导致未执行的 ATA 指令被终止。设置 ATA 状态寄存器的 ERR 位，并向 ATA 错误寄存器发送相应的错误代码。
- 如果 nIEN=0，则向 CPU 发送指令完成信号 (CCS)。

如果设备中断未使能 (nIEN=1)，在 CPU 控制器未中止正在进行的传输的情况下，设备将完成整个数据计数单元。

在多块数据传输中，如果从设备收到的是负 CRC 状态，则数据路径将通过设置 RINTSTS 寄存器的数据 CRC 错误位向 BIU 发送 CRC 错误。然后，将继续后续的数据发送，直到所有字节发送完成。

- 读取操作 - 如果CPU控制器检测到MMC传输层错误，CPU将在错误状态下完成ATA指令。

CPU 控制器可以先发出指令完成信号禁用 (CCSD)，再发出 STOP TRANSMISSION (CMD12) 指令来中止读取传输。此外，CPU 也可以在不中止数据传输的情况下执行完整的数据计数单元字节传输。

20.7.5 DMA 描述符

SD/MMC DMA 控制器采用以下描述符结构：

- 双缓冲结构 – 两个描述符之间的距离由在总线模式寄存器 (BMOD) 的描述符跳过长度 (DSL) 字段中编程的跳过长度值确定。
- 链状结构 – 每个描述符指向唯一的缓冲和下一个描述符。

图 41. 双缓冲描述符结构

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 42. 链状描述符结构

20.7.5.1 SD/MMC DMA 描述符

20.7.5.1.1 SD/MMC DMA 描述符 DESC0

DES0 描述符包含控制和状态信息。

表 335. SD/MMC DMA DESC0 描述符

位	符号	描述
0	-	保留
1	DIC	禁用完成时中断 设置时，此位将阻止对此描述符指向的缓冲区中结束的数据设置 IDMAC 状态寄存器 (IDSTS) 的 TI/RI 位。
2	LD	最后描述符 设置时，此位表示此描述符指向的缓冲区是数据的最后缓冲区。
3	FS	第一个描述符 设置时，此位表示此描述符包含数据的第一个缓冲区。如果第一个缓冲区的大小为 0，则下一描述符包含数据的起始端。
4	CH	第二链接地址 设置时，此位表示描述符中的第二地址是下一描述符地址，而非第二缓冲区地址。设置此位时， BS2 (DES1[25:13]) 应全部为 0。
5	ER	环端 设置时，此位表示描述符列表到达其最后的描述符。 IDMAC 返回列表的基础地址，创建描述符环。此设置仅适用于双缓冲描述符结构。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 335. SD/MMC DMA DESC0 描述符 *（续）*

位	符号	描述
29:6	-	保留
30	CES	卡错误汇总 这些错误位指示卡与外设备之间事务的状态。此外，这些位还存在于 RINTSTS 中，指示下列位的逻辑 OR： • EBE：结束位错误 • RTO：响应超时 • RCRC：响应 CRC • SBE：起始位错误 • DRTO：数据读取超时 • DCRC：用于接收的数据 CRC • RE：响应错误
31	OWN	设置时，该位表示描述符为 SD/MMC DMA 所有。此位复位时，它表示描述符为主机所有。完成数据传输后， SD/MMC DMA 将清除此位。

20.7.5.1.2 SD/MMC DMA 描述符 DESC1

DES1 描述符包含缓冲区大小。

表 336. SD/MMC DMA DESC1 描述符

位	符号	描述
12:0	BS1	缓冲区 1 大小 表示数据缓冲区字节大小，必须是 4 字节的倍数。如果缓冲区大小不是 4 的倍数，将产生未定义的行为。如果此字段为 0，DMA 将忽略此缓冲区并继续下一描述符（对于链状结构）或下一缓冲区 （对于双缓冲结构）。 **注：**如果仅编程了一个描述符和一个缓冲区，则仅使用缓冲区 1，而不是缓冲区 2。
25:13	BS2	缓冲区 2 大小 这些位表示第二个数据缓冲区的字节大小。缓冲区大小必须是 4 的倍数。否则，产生的行为未定义。 DES0[4] 设置时，此字段无效。
31:26	-	保留

20.7.5.1.3 SD/MMC DMA 描述符 DESC2

DES2 描述符包含指向数据缓冲区的地址指针：

表 337. SD/MMC DMA DESC2 描述符

位	符号	描述
31:0	BAP1	缓冲区地址指针 1 这些位表示第一个数据缓冲区的物理地址。 SD/MMC DMA 将忽略 DES2 [2/1/0:0]，它从内部与总线宽度 64/32/16 对应。

20.7.5.1.4 SD/MMC DMA 描述符 DESC3

如果现有的描述符不是链状描述符结构中的最后一个描述符或双缓冲结构的第二个缓冲地址，则 DES3 描述符包含下一描述符的地址指针。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 338. SD/MMC DMA DESC3 描述符

位	符号	描述
31:0	BAP2	缓冲区地址指针 2/ 下一描述符地址 这些位表示第二个缓冲区 （如果使用的是双缓冲结构）的物理地址。如果第二链接地址 (DES0[4]) 位已设置，则此地址包含指向有下一描述符的物理内存的指针。 如果这不是最后的描述符，则下一描述符地址指针必须与总线宽度对齐（DES3[2/1/0:0] = 0，内部将忽略 LSB）。

20.7.5.2 初始化

有关 SD/MMC DMA 初始化，请遵循下列步骤：

1. 写入总线模式寄存器 (BMOD)，设置主机总线访问参数。
2. 写入中断使能寄存器 (IDINTEN)，以掩码不必要的中断原因。
3. 软件驱动程序建立发送和接收描述符列表。然后，写入描述符表基地址寄存器 (DBADDR)，向 IDMAC 提供列表的起始地址。
4. SD/MMC DMA 引擎尝试从描述符列表获取描述符。

20.7.5.3 主机总线连发访问

如果已使用 IDMAC 总线模式寄存器的 FB 位配置，SD/MMC DMA 将尝试在 AHB 主机接口执行固定长度连发传输。最大连发长度由 PBL 字段表示和限制。始终以要读取的 16 字节的最大可能连发量存取描述符：16*8/ 总线宽度。

仅当 FIFO 或直到数据尾端的字节数 （当小于配置的连发长度时）中有足够空间容纳已配置连发时， SD/MMC DMA 才启动数据传输。 SD/MMC DMA 指出 AHB 主机接口要求的起始地址和传输数量。当 AHB 接口被配置为固定长度连发时，它使用 INCR4/8/16 和 SINGLE 事务的最佳组合来传输数据。否则 （没有固定长度的连发），它使用 INCR （未定义长度）和 SINGLE 事务来传输数据。

20.7.5.4 主机数据缓冲区对齐

主机内存中的收发数据缓冲区必须为 32 位对齐。

20.7.5.5 缓冲区大小计算

驱动器知道要收发的数据量。向卡发送时， IDMAC 将向 FIFO 传输 DES1 的缓冲区大小字段指示的准确字节数。

如果描述符未标记为最后一位 （DES0 的 LS 位），则描述符对应的缓冲区已满，缓冲区容量字段将准确标出缓冲区中的有效数据量。如果描述符被标记为最后一位，则缓冲区可能未满 （如 DES1 中的缓冲区容量所示）。在此情况下，驱动器知道有效的位置数。

20.7.5.6 传输

SD/MMC 发送的过程如下：

1. 主机建立发送描述符 (DES0-DES3) 并设置 OWN 位 (DES0[31])。此外，主机还要准备数据缓冲区。
2. 主机在 BIU 的 CMD 寄存器中编程写入数据指令。
3. 主机还要对所需的发送阈值级别进行编程 （FIFOTH 寄存器的 TX_WMark 字段）。
4. SD/MMC DMA 根据步骤 2 的结果，确定需要执行写入数据传输。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

5. SD/MMC DMA 引擎获取描述符并检查 OWN 位。如果 OWN 位未设置，则表示主机拥有描述符。这种情况下，SD/MMC DMA 进入挂起状态并在 SD/MMC DMA 状态寄存器 (IDSTS) 中断言描述符禁用中断。这样，主机需要向轮询请求寄存器写入任意值以释放 SD/MMC DMA。
6. 然后等待指令完成 (CD) 位，并且 BIU 没有错误，表明可以进行传输。
7. 现在，SD/MMC DMA 引擎将等待 BIU 的 DMA 接口请求。此请求将基于编程的发送阈值生成。对于无法使用连发访问的数据最后字节，将在 AHB 主接口执行 SINGLE 传输。
8. SD/MMC DMA 从主机内存的数据缓冲区获取发送数据，并向 FIFO 传输以发送到卡中。
9. 当数据横跨多个描述符时，SD/MMC DMA 将获取下一描述符并继续对下一描述符的操作。描述符中的最后描述符位表示数据是否横跨多个描述符。
10. 数据传输完成后，通过设置发送中断（如果已使能）在 SD/MMC DMA 状态寄存器 (IDSTS)中更新状态信息。此外，SD/MMC DMA通过向DES0执行写事务清除OWN位。

20.7.5.7 接收

SD/MMC 接收的过程如下：

1. 主机建立接收描述符 (DES0-DES3) 并设置 OWN 位 (DES0[31])。
2. 主机在 BIU 的 CMD 寄存器中编程读取数据指令。
3. 主机还要对所需的接收阈值级别进行编程（FIFOTH 寄存器的 RX_WMark 字段）。
4. SD/MMC DMA 根据步骤 2 的结果，确定需要执行读取数据传输。
5. SD/MMC DMA 引擎获取描述符并检查 OWN 位。如果 OWN 位未设置，则表示主机拥有描述符。这种情况下， DMA 进入挂起状态并在 SD/MMC DMA 状态寄存器 (IDSTS) 中断言描述符禁用中断。这样，主机需要向轮询请求寄存器写入任意值以释放 SD/MMC DMA。
6. 然后等待指令完成 (CD) 位，并且 BIU 没有错误，表明可以进行传输。
7. 现在， SD/MMC DMA 引擎将等待 BIU 的 DMA 接口请求 (dw_dma_req)。此请求将基于编程的接收阈值生成。对于无法使用连发访问的数据最后字节，将在 AHB 上执行 SINGLE 传输。
8. SD/MMC DMA 从 FIFO 获取数据并向主机内存传输。
9. 当数据横跨多个描述符时，SD/MMC DMA 将获取下一描述符并继续对下一描述符的操作。描述符中的最后描述符位表示数据是否横跨多个描述符。
10. 数据接收完成后，通过设置接收中断（如果已使能）在 SD/MMC DMA 状态寄存器 (IDSTS)中更新状态信息。此外，SD/MMC DMA通过向DES0执行写事务清除OWN位。

20.7.5.8 中断

中断作为各种事件的结果而生成。SD/MMC DMA 状态寄存器 (IDSTS) 包含所有可能引起中断的位。SD/MMC DMA 中断使能寄存器 (IDINTEN) 包含针对可能引发中断的每个事件的使能位。

如状态寄存器 (IDSTS) 中所述，有两组汇总中断，正常中断和异常中断。通过向相应位的位置写 1 可清零中断。当一个组内所有已使能中断清零时，相应摘要位也将清零。当两个汇总位都被清除，中断信号被取消。

中断不排队且如果中断事件发生在驱动器响应其之前，不会生成额外的中断。例如，接收中断 (IDSTS[1]) 表示向主机缓冲区传输了一个或多个数据。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

对于同时发生的多个事件，只生成一次中断。驱动器必须对 SD/MMC DMA 状态寄存器扫描，以了解中断原因。

注：创建的最终中断 (int) 信号是来自 BIU 和 SD/MMC DMA 的中断逻辑 OR。

20.7.5.9 中止

当数据正在卡数据线上传输时，主机发出 CMD12，此时 FSM 将在完成数据传输后到断言 DTO 中断前，关闭当前描述符。发出中止指令后， DMA 将执行单连发传输：

1. 当数据正在卡数据线上传输时，主机发出 CMD12，此时 FSM 将在完成数据传输后到断言 DTO 中断前，关闭当前描述符。发出中止指令后， DMAC 将执行单连发传输。
2. 对于卡读取，在生成 DTO 中断前，SD/MMC DMA 将保持从 FIFO 弹出数据并写入主机内存。这是必需的要求，因为在清空所有 FIFO 数据前不会生成 DTO 中断。

注：下列情况下将关闭描述符：

- 对于 FBE，当前描述符和剩余的未读取的描述符不是由 SD/MMC DMA 关闭。
- 对于写入中止，仅发生中止时的当前描述符由 SD/MMC DMA 关闭。剩余的未读取的描述符不是由 IDMAC 关闭。
- 对于读取中止，SD/MMC DMA会将数据弹出FIFO并将它们写入对应的描述符数据缓冲区。不关闭剩余的未读取的描述符。

20.7.5.10 FBE 案例

由于 AHB 总线出现 AHB 响应错误，发生 FBE。这是系统错误，所以软件驱动程序不应再对 SD/MMC 执行任何编程。对此情形，唯一的恢复方法是执行下列操作之一：

- 通过设置 reset_n 信号发出硬复位指令。
- 通过写入 CTRL[0] 寄存器，执行程序控制器复位。

20.7.5.11 FIFO 上溢和下溢

在正常数据传输条件下，不会发生 FIFO 上溢和下溢。但是，如果出现编程错误，则可能会产生 FIFO 上溢 / 下溢。例如，请考虑以下情形。

对于发送：PBL=4， Tx 水印 = 1。对于这些编程值，如果 FIFO 只有一个位置为空，它将向 DMA 状态机发出 dw_dma_req。由于 PBL 值 =4， DMA 将向 FIFO 执行 4 次推入。这样将会导致 FIFO 上溢中断。

对于接收：PBL=4， Rx 水印 = 1。对于这些编程值，如果 FIFO 只填充了一个位置，它将向 DMA 状态机发出 dw_dma_req。由于 PBL 值 =4， DMA 将对 FIFO 执行 4 次弹出。这样将会导致 FIFO 下溢中断。

驱动器应确保描述符指示的传输字节数为 4 字节的倍数。例如，如果 BYTCNT = 13，则描述符中指示的字节数应为 16。

20.7.5.12 PBL 和水印级编程

SD/MMC DMA 根据已编程 PBL 和阈值执行数据传输。表 339 列出了允许的编程值。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 339. PBL 和水印级

PBL （传输数）	收发水印值
1	大于或等于 1
4	大于或等于 4
8	大于或等于 8
16	大于或等于 16
32	大于或等于 32
64	大于或等于 64
128	大于或等于 128
264	大于或等于 256

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

21.1 本章导读

EMC 在所有 LPC43xx 器件中可用。

存储器和地址总线宽度取决于封装尺寸 （参见表 340）。

表 340. 不同封装的 EMC 引脚

功能	LBGA256	TFBGA180	TFBGA100	LQFP208	LQFP144	LQFP100
A	EMC_A[22:0]	EMC_A[22:0]	EMC_A[13:0]	EMC_A[22:0]	EMC_A[15:0]	EMC_A[13:0]
D	EMC_D[31:0]	EMC_D[15:0]	EMC_D[7:0]	EMC_D[29:0]	EMC_D[15:0]	EMC_D[7:0]
$\overline{\text{BLS}}$	$\overline{\text{EMC_BLS[3:0]}}$	$\overline{\text{EMC_BLS[3:0]}}$	BLS0	$\overline{\text{EMC_BLS[3:0]}}$	$\overline{\text{EMC_BLS[1:0]}}$	BLS0
$\overline{\text{CS}}$	$\overline{\text{EMC_CS[3:0]}}$	$\overline{\text{EMC_CS[3:0]}}$	CS0	$\overline{\text{EMC_CS[3:0]}}$	$\overline{\text{EMC_CS[1:0]}}$	CS0
$\overline{\text{OE}}$	$\overline{\text{EMC_OE}}$	$\overline{\text{EMC_OE}}$	$\overline{\text{EMC_OE}}$	$\overline{\text{EMC_OE}}$	$\overline{\text{EMC_OE}}$	$\overline{\text{EMC_OE}}$
$\overline{\text{WE}}$	$\overline{\text{EMC_WE}}$	$\overline{\text{EMC_WE}}$	$\overline{\text{EMC_WE}}$	$\overline{\text{EMC_WE}}$	$\overline{\text{EMC_WE}}$	$\overline{\text{EMC_WE}}$
$\overline{\text{CKEOUT}}$	$\overline{\text{EMC_CKEOUT[3:0]}}$	$\overline{\text{EMC_CKEOUT[1:0]}}$	$\overline{\text{EMC_CKEOUT[1:0]}}$	$\overline{\text{EMC_CKEOUT[1:0]}}$	$\overline{\text{EMC_CKEOUT[1:0]}}$	EMC_CKEOUT0
CLK	EMC_CLK[3:0]；EMC_CLK12、EMC_CLK23	EMC_CLK0、EMC_CLK3；EMC_CLK12、EMC_CLK23	EMC_CLK0、EMC_CLK3；EMC_CLK12、EMC_CLK23	EMC_CLK0、EMC_CLK3；EMC_CLK12、EMC_CLK23	EMC_CLK0、EMC_CLK3；EMC_CLK12、EMC_CLK23	EMC_CLK0、EMC_CLK3；EMC_CLK12、EMC_CLK23
$\overline{\text{DQMOUT}}$	$\overline{\text{EMC_DQMOUT[3:0]}}$	$\overline{\text{EMC_DQMOUT[1:0]}}$	-	$\overline{\text{EMC_DQMOUT[1:0]}}$	$\overline{\text{EMC_DQMOUT[1:0]}}$	-
$\overline{\text{DYCS}}$	$\overline{\text{EMC_DYCS[3:0]}}$	$\overline{\text{EMC_DYCS[2:0]}}$	$\overline{\text{EMC_DYCS[1:0]}}$	$\overline{\text{EMC_DYCS[2:0]}}$	$\overline{\text{EMC_DYCS[1:0]}}$	$\overline{\text{EMC_DYCS0}}$
$\overline{\text{CAS}}$	$\overline{\text{EMC_CAS}}$	$\overline{\text{EMC_CAS}}$	$\overline{\text{EMC_CAS}}$	$\overline{\text{EMC_CAS}}$	$\overline{\text{EMC_CAS}}$	$\overline{\text{EMC_CAS}}$
$\overline{\text{RAS}}$	$\overline{\text{EMC_RAS}}$	$\overline{\text{EMC_RAS}}$	$\overline{\text{EMC_RAS}}$	$\overline{\text{EMC_RAS}}$	$\overline{\text{EMC_RAS}}$	$\overline{\text{EMC_RAS}}$

21.2 基本配置

外部存储控制器配置如下：

- 参见表 341 以了解时钟与电源控制。支持两个时钟：BASE_M3_CLK 和 1/2 ≤ BASE_M3_CLK。
- 如果为1/2 ≤ BASE_M3_CLK配置了EMC CCLK，则必须对CLK_M3_EMC_DIV分支时钟进行配置，以便在 CREG6 寄存器 （表 43）和 CCU1 CLK_EMCDIV_CFG 寄存器（表 97）中进行半频时钟的相关操作。
- 要使用 16 位宽和 32 位宽 SDRAM 接口，应选择 EMC_CLK 功能并在 SCU 的全部四个 SFSCLKn 寄存器中使能输入缓冲 (EZI = 1)。
- EMC 通过 EMC_RST 复位 （复位 # 21）。
- 在SCU模块的EMCDELAYCLK寄存器中，为EMC_CLKn线路编程SDRAM延迟值。（参见第 15.4.9 节）。为典型条件下以超过 96 MHz 的频率运行的多数 SDRAM 设备添加 SDRAM 延迟。为任意频率添加 SDRAM 延迟，以补偿不同温度下的差异。有关详情，请参见 *LPC4350_30_20_10 数据手册*。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 341. EMC 时钟和电源控制

	基本时钟	分支时钟	工作频率	备注
EMC 寄存器和 EMC CCLK	BASE_M4_CLK	CLK_M4_EMC	高达204 MHz	最大操作频率取决于温度和电压设置，对于 SDRAM 设备通常为 120 MHz。有关详情，请参见 *LPC4350_30_20_10 数据手册*。
EMC CCLK（分频时钟）	BASE_M4_CLK	CLK_M4_EMC_DIV	高达204 MHz	这是 CCLK 的备选时钟选项。该时钟能以与 BASE_M4_CLK 相同的频率或者 BASE_M4_CLK 频率的一半运行。

21.3 功能要点

- 支持 8 位、 16 位和 32 位宽的静态存储器，多达四个片选。
- 支持包括RAM、ROM和NOR闪存在内的异步静态存储设备，带有或不带有异步分页模式。
- 静态存储器功能要点包括：
  - 异步页面模式读取
  - 可编程的等待状态
  - 总线周转延迟
  - 输出使能和写使能延迟
  - 延长等待
- 支持 16 位和 32 位宽片选 SDRAM 存储器，多达四个片选和 256 Mb 数据。
- 控制器支持2 kbit、4 kbit和8 kbit的行地址同步存储器件。这是典型的512 Mb、256 Mb和 128 Mb 器件，每个设备有 4、 8、 16 或 32 个数据位。
- 动态存储器接口支持，包括单一数据速率 SDRAM。
- 软件控制动态存储器的自动刷新模式；
- 掉电模式动态控制 SDRAM 的 EMC_CKEOUT 和 EMC_CLK 至 SDRAM。
- 低事务延迟。
- 读和写缓冲区用来降低延迟并提高性能。
- 如果需要，独立的复位域允许通过芯片复位进行自动刷新。
- 可编程延迟元素允许对 EMC 定时进行微调。

注：不支持同步静态存储设备 （同步连发模式）。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

21.4 简介

LPC43xx 外部存储器 (EMC) 是一款 ARM PrimeCell MultiPort 存储控制器外设，支持异步静态存储设备（例如 RAM、ROM 和闪存）以及动态存储器（例如单一数据速率 SDRAM）。

对SYSCON模块的EMCDELAYCLK寄存器中的EMC_CLKn延迟进行编程（表134），调整不同SDRAM频率和操作条件的延迟。

图 43. EMC 框图 (SDRAM)

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 44. EMC 框图 (SRAM)

21.5 存储组别选择

支持 8 个可独立配置的存储器片选：

- 引脚 $\overline{\text{EMC_CS3}}$ 至 $\overline{\text{EMC_CS0}}$ 用于选择静态存储设备。
- 引脚 $\overline{\text{EMC_DYCS3}}$ 至 $\overline{\text{EMC_DYCS0}}$ 用于选择动态存储设备。

每个静态存储器片选范围为 16 Mb，而每个动态存储器片选覆盖范围则为 256 Mb。表 342 显示了芯片选择的地址范围。

表 342. 存储组别选择

片选引脚	地址范围	存储器类型	范围大小
$\overline{\text{EMC_CS0}}$	0x1C00 0000 - 0x1CFF FFFF	静态	16 Mb
$\overline{\text{EMC_CS1}}$	0x1D00 0000 - 01DFF FFFF	静态	16 Mb
$\overline{\text{EMC_CS2}}$	0x1E00 0000 - 0x1EFF FFFF	静态	16 Mb
$\overline{\text{EMC_CS3}}$	0x1F00 0000 - 0x1FFF FFFF	静态	16 Mb
$\overline{\text{EMC_DYCS0}}$	0x2800 0000 - 0x2FFF FFFF	动态	128 Mb
$\overline{\text{EMC_DYCS1}}$	0x3000 0000 - 0x3FFF FFFF	动态	256 Mb
$\overline{\text{EMC_DYCS2}}$	0x6000 0000 - 0x6FFF FFFF	动态	256 Mb
$\overline{\text{EMC_DYCS3}}$	0x7000 0000 - 0x7FFF FFFF	动态	256 Mb

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

21.6 引脚说明

表 343. EMC 引脚描述

函数名称	方向	描述
EMC_A[22:0]	O	地址总线
EMC_D[31:0]	I/O	数据总线
$\overline{\text{EMC_BLS[3:0]}}$	O	字节通道选择
$\overline{\text{EMC_CS[3:0]}}$	O	静态 RAM 存储组别选择
$\overline{\text{EMC_OE}}$	O	输出使能
$\overline{\text{EMC_WE}}$	O	写使能
EMC_CKEOUT[3:0]	O	SDRAM 时钟使能信号
EMC_CLK[3:0]	O	SDRAM 时钟信号
$\overline{\text{EMC_DQMOUT[3:0]}}$	O	至 SDRAM 存储组别的数据掩码输出
$\overline{\text{EMC_DYCS[3:0]}}$	O	SDRAM 存储组别选择
$\overline{\text{EMC_CAS}}$	O	列地址选通
$\overline{\text{EMC_RAS}}$	O	行地址选通

21.7 寄存器描述

本章描述了 EMC 寄存器，并提供了编程微控制器所需的详细信息。 EMC 寄存器如表 344 中所示。
复位值仅反映存储在所使用的位中的数据。不包含保留位中的内容。

表 344. 寄存器简介：外部存储控制器 （基址 0x4000 5000）

名称	访问类型	地址偏移	描述	复位值	启动后的复位值	参考
控制	R/W	0x000	控制存储控制器的操作。	0x3	[1]0x1	表 345
STATUS	RO	0x004	提供 EMC 状态信息。	0x5	0x5	表 346
CONFIG	R/W	0x008	配置存储控制器的操作。	0	0	表 347
-	-	0x00C - 0x01C	保留。	-	-	-
DYNAMICCONTROL	R/W	0x020	控制动态存储器操作。	0x6	0x6	表 348
DYNAMICREFRESH	R/W	0x024	配置动态存储器刷新操作。	0	0	表 349
DYNAMICREADCONFIG	R/W	0x028	配置动态存储器读策略。	0	0	表 350
-	-	0x02C	保留。	-	-	-
DYNAMICRP	R/W	0x030	选择预充电命令周期。	0xF	0xF	表 351
DYNAMICRAS	R/W	0x034	选择有效到预充电命令周期。	0xF	0xF	表 352
DYNAMICSREX	R/W	0x038	选择自刷新退出时间。	0xF	0xF	表 353
DYNAMICAPR	R/W	0x03C	选择最后一个数据输出到有效命令的时间。	0xF	0xF	表 354
DYNAMICDAL	R/W	0x040	选择数据输入到有效命令的时间。	0xF	0xF	表 355
DYNAMICWR	R/W	0x044	选择写恢复时间。	0xF	0xF	表 356
DYNAMICRC	R/W	0x048	选择有效到有效命令周期。	0x1F	0x1F	表 357
DYNAMICRFC	R/W	0x04C	选择自动刷新周期。	0x1F	0x1F	表 358
DYNAMICXSR	R/W	0x050	选择退出自刷新到有效命令的时间。	0x1F	0x1F	表 359

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 344. 寄存器简介：外部存储控制器 （基址 0x4000 5000）*（续）*

名称	访问类型	地址偏移	描述	复位值	启动后的复位值	参考
DYNAMICRRD	R/W	0x054	选择有效组A到有效组B的等待时间。	0xF	0xF	表 360
DYNAMICMRD	R/W	0x058	选择加载模式寄存器到有效命令的时间。	0xF	0xF	表 361
-	R/W	0x05C - 0x07C	保留。	-	-	-
STATICEXTENDEDWAIT	R/W	0x080	选择静态存储器长时间读写转换的时间。	0	0	表 362
-	R/W	-	保留。	-	-	-
DYNAMICCONFIG0	R/W	0x100	选择动态存储器片选 0 的配置信息。	0	0	表 363
DYNAMICRASCAS0	R/W	0x104	选择动态存储器片选 0 的 RAS 和 CAS 等待时间。	0x303	0x303	表 365
-		0x108 - 0x11C	保留。	-	-	-
DYNAMICCONFIG1	R/W	0x120	选择动态存储器片选 1 的配置信息。	0	0	表 363
DYNAMICRASCAS1	R/W	0x124	选择动态存储器片选 1 的 RAS 和 CAS 等待时间。	0x303	0x303	表 365
-	-	0x128 - 0x13C	保留。	-	-	-
DYNAMICCONFIG2	R/W	0x140	选择动态存储器片选 2 的配置信息。	0	0	表 363
DYNAMICRASCAS2	R/W	0x144	选择动态存储器片选 2 的 RAS 和 CAS 等待时间。	0x303	0x303	表 365
-	-	0x148 - 0x15C	保留。	-	-	-
DYNAMICCONFIG3	R/W	0x160	选择动态存储器片选 3 的配置信息。	0	0	表 363
DYNAMICRASCAS3	R/W	0x164	选择动态存储器片选 3 的 RAS 和 CAS 等待时间。	0x303	0x303	表 365
-	-	0x168 - 0x1FC	保留。	-	-	-
STATICCONFIG0	R/W	0x200	选择静态片选 0 的存储器配置。	0	0x81	表 366
STATICWAITWEN0	R/W	0x204	选择从片选 0 到写使能的延迟。	0	0	表 367
STATICWAITOEN0	R/W	0x208	选择从片选 0 或地址变更 （较晚出现的一个）到输出使能的延迟。	0	0	表 368
STATICWAITRD0	R/W	0x20C	选择从片选 0 到读访问的延迟。	0x1F	[2] 0xE	表 369
STATICWAITPAGE0	R/W	0x210	选择片选 0 的异步页面模式连续访问的延迟。	0x1F	0x1F	表 370
STATICWAITWR0	R/W	0x214	选择从片选 0 到写访问的延迟。	0x1F	0x1F	表 371
STATICWAITTURN0	R/W	0x218	选择片选 0 总线周转期的数量。	0xF	0xF	表 372
-	-	0x21C	保留。	-	-	-
STATICCONFIG1	R/W	0x220	选择静态片选 1 的存储器配置。	0	0	表 366
STATICWAITWEN1	R/W	0x224	选择从片选 1 到写使能的延迟。	0	0	表 367
STATICWAITOEN1	R/W	0x228	选择从片选 1 或地址变更 （较晚出现的一个）到输出使能的延迟。	0	0	表 368
STATICWAITRD1	R/W	0x22C	选择从片选 1 到读访问的延迟。	0x1F	0x1F	表 369

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 344. 寄存器简介：外部存储控制器 （基址 0x4000 5000）*（续）*

名称	访问类型	地址偏移	描述	复位值	启动后的复位值	参考
STATICWAITPAGE1	R/W	0x230	选择片选 1 的异步页面模式连续访问的延迟。	0x1F	0x1F	表 370
STATICWAITWR1	R/W	0x234	选择从片选 1 到写访问的延迟。	0x1F	0x1F	表 371
STATICWAITTURN1	R/W	0x238	选择片选 1 总线周转期的数量。	0xF	0xF	表 372
-	-	0x23C	保留。	-	-	-
STATICCONFIG2	R/W	0x240	选择静态片选 2 的存储器配置。	0	0	表 366
STATICWAITWEN2	R/W	0x244	选择从片选 2 到写使能的延迟。	0	0	表 367
STATICWAITOEN2	R/W	0x248	选择从片选 2 或地址变更 （较晚出现的一个）到输出使能的延迟。	0	0	表 368
STATICWAITRD2	R/W	0x24C	选择从片选 2 到读访问的延迟。	0x1F	0x1F	表 369
STATICWAITPAGE2	R/W	0x250	选择片选 2 的异步页面模式连续访问的延迟。	0x1F	0x1F	表 370
STATICWAITWR2	R/W	0x254	选择从片选 2 到写访问的延迟。	0x1F	0x1F	表 371
STATICWAITTURN2	R/W	0x258	选择片选 2 总线周转期的数量。	0xF	0xF	表 372
-	-	0x25C	保留。	-	-	-
STATICCONFIG3	R/W	0x260	选择静态片选 3 的存储器配置。	0	0	表 366
STATICWAITWEN3	R/W	0x264	选择从片选 3 到写使能的延迟。	0	0	表 367
STATICWAITOEN3	R/W	0x268	选择从片选 3 或地址变更 （较晚出现的一个）到输出使能的延迟。	0	0	表 368
STATICWAITRD3	R/W	0x26C	选择从片选 3 到读访问的延迟。	0x1F	0x1F	表 369
STATICWAITPAGE3	R/W	0x270	选择片选 3 的异步页面模式连续访问的延迟。	0x1F	0x1F	表 370
STATICWAITWR3	R/W	0x274	选择从片选 3 到写访问的延迟。	0x1F	0x1F	表 371
STATICWAITTURN3	R/W	0x278	选择片选 3 总线周转期的数量。	0xF	0xF	表 372

[1] CONTROL 寄存器热复位后的复位值为 0x0000 0001。

[2] 如果从 EMC 启动，请参见第 5.3.4.3 节。

21.7.1 EMC 控制寄存器

控制 （Control）寄存器是控制存储控制器操作的读 / 写寄存器。在正常操作过程中可以改变控制位。

表 345. EMC 控制寄存器 （CONTROL – 地址 0x4000 5000）位描述

位	符号	值	描述	复位值
0	E		EMC 使能。指示 EMC 为使能还是停用状态。停用 EMC 可降低功耗。如果禁用了存储控制器，则不会刷新存储器。设置使能位或通过复位可以使能存储控制器。仅当 EMC 处于空闲状态时，才能对该位进行修改。[1]	1
		0	已禁用	
		1	使能 （POR 和热复位值）。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 345. EMC 控制寄存器 （CONTROL – 地址 0x4000 5000）位描述

位	符号	值	描述	复位值
1	M		地址镜像。表示正常或复位存储器映射。上电复位时，CS1 被镜像到 CS0 和 DYCS0 存储器区域。清零 M 位使 CS0 和 DYCS0 存储器可被访问。	1
		0	正常存储器映射。	
		1	复位存储器映射。静态存储器 CS1 被镜像到 CS0 和 DYCS0（POR 复位值）。	
2	L		低功耗模式。表示正常模式或低功耗模式。进入低功耗模式以减少存储控制器的功耗。必要时刷新动态存储器。通过清零低功耗模式位 （L）或 POR 可使存储控制器返回正常功能模式。 仅当 EMC 处于空闲状态时，才能对该位进行修改。[1]	0
		0	正常模式 （热复位值）。	
		1	低功耗模式。	
31:3	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 在低功耗或禁用的状态下，不能访问外部存储器。如果执行存储器访问，会生成 AHB 错误响应。在低功耗和 / 或禁用的状态下，可以对 EMC 寄存器进行编程。

21.7.2 EMC 状态寄存器

只读的 Status 寄存器提供 EMC 状态信息。

表 346. EMC 状态寄存器 （STATUS – 地址 0x4000 5004）位描述

位	符号	值	描述	复位值
0	B		忙。 该位用于确保存储控制器能顺利进入低功耗或停用模式，具体则是通过确定存储控制器是否忙碌而实现的：	1
		0	EMC 处于空闲状态 （热复位值）。	
		1	EMC 忙于执行存储器事务、命令、自动刷新周期，或处于自刷新模式 （POR 复位值）。	
1	S		写缓冲区状态。该位可使 EMC 进入低功耗模式或禁用模式：	0
		0	写缓冲区为空 （POR 复位值）	
		1	写缓冲区包含数据。	
2	SA		自刷新应答。该位指示 EMC 的操作模式：	1
		0	正常模式	
		1	自刷新模式 （POR 复位值）。	
31:3	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

21.7.3 EMC 配置寄存器

配置（Config）寄存器配置存储控制器的操作。建议在系统初始化时对该寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。该寄存器以一个等待状态访问。

表 347. EMC 配置寄存器 （CONFIG – 地址 0x4000 5008）位描述

位	符号	值	描述	复位值
0	EM		端模式。	0
		0	小端模式 （POR 复位值）。	
		1	大端模式。	
			上电复位时，端位值为 0。在小端与大端模式之间进行切换之前，必须清除 EMC 中的所有数据。	
7:1	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
8	CR		时钟比率。 CCLK:CLKOUT[1:0] 比率：	0
		0	1:1 （POR 复位值）	
		1	1:2	
			该位必须包含 0，以便进行正确的 EMC 操作。	
31:9	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

21.7.4 动态存储器控制寄存器

动态控制 （DynamicControl）寄存器控制动态存储器操作。在正常操作过程中可以改变控制位。

表 348. 动态控制寄存器 （DYNAMICCONTROL – 地址 0x4000 5020）位描述

位	符号	值	描述	复位值
0	CE		动态存储器时钟使能。	0
		0	使空闲设备的时钟使能失效，以节省功耗 （POR 复位值）。	
		1	所有时钟使能被持续驱动为高电平。[1]	
1	CS		动态存储器时钟控制。如果时钟控制为低电平，则在无 SDRAM 事务时，输出时钟 CLKOUT 会停止。自刷新模式下该时钟也会停止。	1
		0	当所有 SDRAM 在自刷新模式下并处于空闲状态时，CLKOUT 停止。	
		1	CLKOUT 持续运行 （POR 复位值）。	
2	SR		自刷新请求， EMCSREFREQ。通过向该位写 1，可以在软件控制下进入自刷新模式。向该位写 0，则会使 EMC 返回正常模式。 必须轮询状态寄存器中的自刷新确认位，以发现 EMC 的当前运行模式。[2]	1
		0	正常模式。	
		1	进入自刷新模式 （POR 复位值）。	
4:3	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
5	MMC		存储器时钟控制。	0
		0	使能 CLKOUT （POR 复位值）。	
		1	禁用 CLKOUT。[3]	
6	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 348. 动态控制寄存器 （DYNAMICCONTROL – 地址 0x4000 5020）位描述 *（续）*

位	符号	值	描述	复位值
8:7	I		SDRAM 初始化。	00
		0x0	发出 SDRAM NORMAL 操作命令 （POR 复位值）。	
		0x1	发出 SDRAM MODE 命令。	
		0x2	发出 SDRAM PALL （预充电全部）命令。	
		0x3	发出 SDRAM NOP （无操作）命令。	
12:9	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
13	DP		低功耗 SDRAM 深度睡眠模式。	0
		0	正常操作 （POR 复位值）。	
		1	进入深度睡眠模式。	
31:14	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 在 SDRAM 初始化中，时钟使能必须为高电平。

[2] 存储控制器退出上电复位，其自刷新位为高电平。如果要进入正常功能模式，则可将该位设为低电平。

[3] 如果没有 SDRAM 存储器事务，则可执行停用 CLKOUT。该位使能后，可与动态存储器时钟控制 （CS）字段组合使用。

注：可以通过将深度睡眠模式 (DP) 位、动态存储器时钟使能位 (CE) 和动态时钟控制位 (CS) 设为 1 来进入深度睡眠模式。然后，器件会进入低功耗模式；在该模式下，器件掉电且不再刷新。存储器中所有数据丢失。

21.7.5 动态存储器刷新定时器寄存器

动态刷新 （DynamicRefresh）寄存器配置动态存储器操作。建议在系统初始化时对该寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。但是，如有必要，则在正常操作过程中也可以改变这些控制位。该寄存器以一个等待状态访问。

注：该寄存器用于全部 4 个动态存储器片选。因此，必须对所有片选在最差情况下的值进行编程。

表 349. 动态存储器刷新定时器寄存器 （DYNAMICREFRESH – 地址 0x4000 5024）位描述

位	符号	描述	复位值
10:0	REFRESH	刷新定时器。 指示 SDRAM 刷新周期之间 16 CCLK 的倍数。 0x0 = 刷新禁用 （POR 复位值）。 0x1 - 0x7FF = n x16 = SDRAM 刷新周期之间有多个 （16 的 n 倍） CCLK。 例如： 0x1 = 1 x 16 = SDRAM 刷新周期之间有 16 个 CCLK。 0x8 = 8 x 16 = SDRAM 刷新周期之间有 128 个 CCLK。	0
31:11	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

例如，对于 16 µs 的刷新周期，如果 CCLK 频率为 50 MHz，则必须将以下值编入该寄存器：

$(16 \times 10^{-6} \times 50 \times 10^{6}) / 16 = 50$ 或 0x32

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

如果要求通过热复位自动刷新（置位 EMC_Reset_Disable 位），那么当时钟速率在唤醒复位周期的过程中下降时，必须调整自动刷新的时序以允许一个足够高的刷新速率。在该周期内， EMC（以及芯片中被计时的所有其他部分）从 12 MHz 的 IRC 振荡器运行。因此，如果要求通过热复位自动刷新，则必须将 IRC 振荡器的频率用作刷新计算的 CCLK 速率。

注：刷新周期是均匀分布的。但是，当发出自动刷新命令时，时序可能会稍有变化，这取决于存储控制器的状态。

21.7.6 动态存储器读配置寄存器

动态读取配置 （DynamicReadConfig）寄存器配置动态存储器的读策略。该寄存器必须在系统初始化时使用位值 RD ≥ 1 进行修改。该寄存器以一个等待状态访问。

注：该寄存器用于全部 4 个动态存储器片选。因此，必须对所有片选在最差情况下的值进行编程。

注：为 SDRAM 操作选择指令延迟策略 (RD = 0x1)。

关于编程 EMC_CLKn 延迟的延迟值，请参见第 15.4.9 节。

表 350. 动态存储器读配置寄存器 （DYNAMICREADCONFIG – 地址 0x4000 5028）位描述

位	符号	值	描述	复位值
1:0	RD		读数据策略。	0x0
		0x0	不使用。 POR 复位值。	
		0x1	指令延迟 1/2CCLK。	
		0x2	指令延迟 1/2 CCLK 加上 1 个时钟周期。	
		0x3	指令延迟 1/2 CCLK 加上 2 个时钟周期，	
31:2	-	-	保留，用户软件不应向保留位写入1。未定义从保留位读取的值。	-

21.7.7 动态存储器预充电命令周期寄存器

可以通过 DynamicTRP 寄存器编程预充电命令周期 tRP。该寄存器只能在系统初始化时进行修改。 SDRAM 数据手册中通常将该值表示为 tRP。该寄存器以一个等待状态访问。

注：该寄存器用于全部 4 个动态存储器片选。因此，必须对所有片选在最差情况下的值进行编程。

表 351. 动态存储器预充电命令周期寄存器 （DYNAMICRP – 地址 0x4000 5030）位描述

位	符号	描述	复位值
3:0	tRP	预充电命令周期。 0x0 - 0xE = n + 1 个时钟周期。该延迟以 CCLK 周期为单位。 0xF = 16 个时钟周期 （POR 复位值）。	0xF
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

21.7.8 动态存储器有效到预充电命令周期寄存器

可以通过 DynamicTRAS 寄存器编程动态存储器有效到预充电命令周期 tRAS。建议在系统初始化时对该寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。SDRAM 数据手册中通常将该值表示为 tRAS。该寄存器以一个等待状态访问。

注：该寄存器用于全部 4 个动态存储器片选。因此，必须对所有片选在最差情况下的值进行编程。

表 352. 动态存储器有效到预充电命令周期寄存器 （DYNAMICRAS – 地址 0x4000 5034）位描述

位	符号	描述	复位值
3:0	tRAS	有效到预充电命令周期。 0x0 - 0xE = n + 1 个时钟周期。该延迟以 CCLK 周期为单位。 0xF = 16 个时钟周期 （POR 复位值）。	0xF
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

21.7.9 动态存储器自刷新退出时间寄存器

可以通过 DynamicTSREX 寄存器编程自刷新退出的时间 tSREX。建议在系统初始化时对该寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。SDRAM 数据手册中通常将该值表示为 tSREX，对于没有此参数的设备，可以使用与 tXSR 相同的值。该寄存器以一个等待状态访问。

注：该寄存器用于全部 4 个动态存储器片选。因此，必须对所有片选在最差情况下的值进行编程。

表 353. 动态存储器自刷新退出时间寄存器 （DYNAMICSREX – 地址 0x4000 5038）位描述

位	符号	描述	复位值
3:0	tSREX	自刷新退出时间。 0x0 - 0xE = n + 1 个时钟周期。该延迟以 CCLK 周期为单位。 0xF = 16 个时钟周期 （POR 复位值）。	0xF
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

21.7.10 动态存储器最后一个数据输出到有效时间寄存器

可以通过 DynamicTAPR 寄存器编程最后一个数据输出到有效命令的时间 tAPR。建议在系统初始化时对该寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。SDRAM 数据手册中通常将该值表示为 tAPR。该寄存器以一个等待状态访问。

注：该寄存器用于全部 4 个动态存储器片选。因此，必须对所有片选在最差情况下的值进行编程。

表 354. 动态存储器最后一个数据输出到有效时间寄存器 （DYNAMICAPR – 地址 0x4000 503C）位描述

位	符号	描述	复位值
3:0	tAPR	最后一个数据输出到活动命令时间。 0x0 - 0xE = n + 1 个时钟周期。该延迟以 CCLK 周期为单位。 0xF = 16 个时钟周期 （POR 复位值）。	0xF
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

21.7.11 动态存储器数据输入到有效命令时间寄存器

可以通过 DynamicTDAL 寄存器编程数据输入到有效命令的时间 tDAL。建议在系统初始化时对该寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。SDRAM 数据手册中通常将该值表示为 tDAL 或 tAPW。该寄存器以一个等待状态访问。

注：该寄存器用于全部 4 个动态存储器片选。因此，必须对所有片选在最差情况下的值进行编程。

表 355. 动态存储器数据输入到有效命令时间寄存器（DYNAMICDAL – 地址 0x4000 5040）位描述

位	符号	描述	复位值
3:0	tDAL	数据输入到活动命令。 0x0 - 0xE = n 个时钟周期。该延迟以 CCLK 周期为单位。 0xF = 15 个时钟周期 （POR 复位值）。	0xF
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

21.7.12 动态存储器写恢复时间寄存器

可以通过 DynamicTWR 寄存器编程写恢复时间 tWR。建议在系统初始化时对该寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到EMC 变为空闲状态，然后进入低功耗或禁用模式。SDRAM 数据手册中通常将该值表示为 tWR、tDPL、 tRWL 或 tRDL。该寄存器以一个等待状态访问。

注：该寄存器用于全部 4 个动态存储器片选。因此，必须对所有片选在最差情况下的值进行编程。

表 356. 动态存储器写入恢复时间寄存器 （DYNAMICWR - 地址 0x4000 5044）位描述

位	符号	描述	复位值
3:0	tWR	写恢复时间。 0x0 - 0xE = n + 1 个时钟周期。该延迟以 CCLK 周期为单位。 0xF = 16 个时钟周期 （POR 复位值）。	0xF
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

21.7.13 动态存储器有效到有效命令周期寄存器

可以通过 DynamicTRC 寄存器编程有效到有效命令周期 tRC。建议在系统初始化时对该寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。SDRAM 数据手册中通常将该值表示为 tRC。该寄存器以一个等待状态访问。

注：该寄存器用于全部 4 个动态存储器片选。因此，必须对所有片选在最差情况下的值进行编程。

表 357. 动态存储器有效到有效命令周期寄存器 （DYNAMICRC – 地址 0x4000 5048）位描述

位	符号	描述	复位值
4:0	tRC	有效到有效命令周期。 0x0 - 0xE = n + 1 个时钟周期。该延迟以 CCLK 周期为单位。 0x1F = 32 个时钟周期 （POR 复位值）。	0x1F
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

21.7.14 动态存储器自动刷新周期寄存器

可以通过 DynamicTRFC 寄存器编程自动刷新周期以及自动刷新到有效命令周期 tRFC。建议在系统初始化时对该寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。 SDRAM 数据手册中通常将该值表示为 tRFC，有时也表示为 tRC。该寄存器以一个等待状态访问。

注：该寄存器用于全部 4 个动态存储器片选。因此，必须对所有片选在最差情况下的值进行编程。

表 358. 动态存储器自动刷新周期寄存器 （DYNAMICRFC – 地址 0x4000 504C）位描述

位	符号	描述	复位值
4:0	tRFC	自动刷新周期和自动刷新到有效命令周期。 0x0 - 0xE = n + 1 个时钟周期。该延迟以 CCLK 周期为单位。 0x1F = 32 个时钟周期 （POR 复位值）。	0x1F
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

21.7.15 动态存储器退出自刷新寄存器

可以通过 DynamicTXSR 寄存器编程退出自刷新到有效命令的时间 tXSR。建议在系统初始化时对该寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。 SDRAM 数据手册中通常将该值表示为 tXSR。该寄存器以一个等待状态访问。

注：该寄存器用于全部 4 个动态存储器片选。因此，必须对所有片选在最差情况下的值进行编程。

表 359. 动态存储器退出自刷新寄存器 （DYNAMICXSR – 地址 0x4000 5050）位描述

位	符号	描述	复位值
4:0	tXSR	退出自刷新到活动命令时间。 0x0 - 0xE = n + 1 个时钟周期。该延迟以 CCLK 周期为单位。 0x1F = 32 个时钟周期 （POR 复位值）。	0x1F
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

21.7.16 动态存储器有效组 A 到有效组 B 时间寄存器

可以通过 DynamicTRRD 寄存器编程有效组 A 到有效组 B 的等待时间 tRRD。建议在系统初始化时对该寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。 SDRAM 数据手册中通常将该值表示为 tRRD。该寄存器以一个等待状态访问。

注：该寄存器用于全部 4 个动态存储器片选。因此，必须对所有片选在最差情况下的值进行编程。

表 360. 动态存储器有效组 A 至有效组 B 时间寄存器（DYNAMICRRD - 地址 0x4000 5054）位描述

位	符号	描述	复位值
3:0	tRRD	有效组 A 到有效组 B 延迟 0x0 - 0xE = n + 1 个时钟周期。该延迟以 CCLK 周期为单位。 0xF = 16 个时钟周期 （POR 复位值）。	0xF
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

21.7.17 动态存储器加载模式寄存器到有效命令时间

可以通过 DynamicTMRD 寄存器编程加载模式寄存器到有效命令的时间 tMRD。建议在系统初始化时对该寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。SDRAM 数据手册中通常将该值表示为 tMRD 或 tRSA。该寄存器以一个等待状态访问。

注：该寄存器用于全部 4 个动态存储器片选。因此，必须对所有片选在最差情况下的值进行编程。

表 361. 动态存储器加载模式寄存器到有效命令时间（DYNAMICMRD – 地址0x4000 5058）位描述

位	符号	描述	复位值
3:0	tMRD	加载模式寄存器到有效命令时间。 0x0 - 0xE = n + 1 个时钟周期。该延迟以 CCLK 周期为单位。 0xF = 16 个时钟周期 （POR 复位值）。	0xF
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

21.7.18 静态存储器延长等待寄存器

置位静态配置 (StaticConfig) 寄存器中的延长等待 (EW) 位。建议在系统初始化时对该寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。但是，如有必要，在正常操作过程中也可以改变这些控制位。该寄存器以一个等待状态访问。

表 362. 静态存储器延长等待寄存器 （STATICEXTENDEDWAIT – 地址 0x4000 5080）位描述

位	符号	描述	复位值
9:0	EXTENDEDWAIT	延长等待超时。 16 个时钟周期（POR 复位值）。该延迟以 CCLK 周期为单位。 0x0 = 16 个时钟周期。 0x1 - 0x3FF = （n+1）x16 个时钟周期。	0x0
31:10	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

例如，在为期 16 µs 的静态存储器读取 / 写入传输时间内，并且 CCLK 频率为 50 MHz，则必须将下列值编入该寄存器中：
$(16 \times 10^{-6} \times 50 \times 10^{6}) / 16 - 1 = 49$

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

21.7.19 动态存储器配置寄存器

可以通过 DynamicConfig 寄存器编程相关动态存储器片选的配置信息。这些寄存器通常只能在系统初始化时进行修改。这些寄存器以一个等待状态访问。

表 363. 动态存储器配置寄存器 （DYNAMICCONFIG，地址 0x4000 5100 (DYNAMICCONFIG0)、0x4000 5120 (DYNAMICCONFIG1)、0x4000 5140 (DYNAMICCONFIG2)、0x4000 5160 (DYNAMICCONFIG3)）位描述

位	符号	值	描述	复位值
2:0	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
4:3	MD		存储器件。	00
		0x0	SDRAM （POR 复位值）。	
		0x1	保留。	
		0x2	保留。	
		0x3	保留。	
6:5	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
12:7	AM0		地址映射。 见 表 364。 000000 = 复位值。[1]	0
13	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
14	AM1		地址映射 见 表 364。 0 = 复位值。	0
18:15	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
19	B		缓冲使能。	
		0	禁用缓冲区，以便访问该片选 （POR 复位值）。	
		1	使能缓冲区，以便访问该片选。动态存储器配置后，必须使能缓冲区才能正常操作。[2]	
20	P		写保护。	0
		0	无写保护 （POR 复位值）。	
		1	有写保护。	
31:21	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] SDRAM 列宽和行宽以及组的数量均从地址映射中自动计算。

[2] SDRAM 和 SyncFlash 初始化时，必须禁用缓冲区。执行 SyncFlash 命令时也必须禁用。而在正常操作时，则必须使能缓冲区。

表 364 中未列出的地址映射保留。

表 364. 地址映射

14	12	11:9	8:7	描述
16 位外部总线高性能地址映射 （行、组、列）				
0	0	000	00	16 Mb (2Mx8)， 2 组，行长度 = 11，列长度 = 9
0	0	000	01	16 Mb (1Mx16)， 2 组，行长度 = 11，列长度 = 8
0	0	001	00	64 Mb (8Mx8)， 4 组，行长度 = 12，列长度 = 9
0	0	001	01	64 Mb (4Mx16)， 4 组，行长度 = 12，列长度 = 8
0	0	010	00	128 Mb (16Mx8)， 4 组，行长度 = 12，列长度 = 10
0	0	010	01	128 Mb (8Mx16)， 4 组，行长度 = 12，列长度 = 9
0	0	011	00	256 Mb (32Mx8)， 4 组，行长度 = 13，列长度 = 10
0	0	011	01	256 Mb (16Mx16)、 4 组、行长度 = 13、列长度 = 9

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 364. 地址映射 *（续）*

14	12	11:9	8:7	描述
0	0	100	00	512 Mb (64Mx8)、 4 组、行长度 = 13、列长度 = 11
0	0	100	01	512 Mb (32Mx16)、 4 组、行长度 = 13、列长度 = 10
16 位外部总线低功耗 SDRAM 地址映射 （组、行、列）				
0	1	000	00	16 Mb (2Mx8)， 2 组，行长度 = 11，列长度 = 9
0	1	000	01	16 Mb (1Mx16)， 2 组，行长度 = 11，列长度 = 8
0	1	001	00	64 Mb (8Mx8)， 4 组，行长度 = 12，列长度 = 9
0	1	001	01	64 Mb (4Mx16)， 4 组，行长度 = 12，列长度 = 8
0	1	010	00	128 Mb (16Mx8)， 4 组，行长度 = 12，列长度 = 10
0	1	010	01	128 Mb (8Mx16)， 4 组，行长度 = 12，列长度 = 9
0	1	011	00	256 Mb (32Mx8)， 4 组，行长度 = 13，列长度 = 10
0	1	011	01	256 Mb (16Mx16)、 4 组、行长度 = 13、列长度 = 9
0	1	100	00	512 Mb (64Mx8)、 4 组、行长度 = 13、列长度 = 11
0	1	100	01	512 Mb (32Mx16)、 4 组、行长度 = 13、列长度 = 10
32 位外部总线高性能地址映射 （行、组、列）				
1	0	000	00	16 Mb (2Mx8)， 2 组，行长度 = 11，列长度 = 9
1	0	000	01	16 Mb (1Mx16)， 2 组，行长度 = 11，列长度 = 8
1	0	001	00	64 Mb (8Mx8)， 4 组，行长度 = 12，列长度 = 9
1	0	001	01	64 Mb (4Mx16)， 4 组，行长度 = 12，列长度 = 8
1	0	001	10	64 Mb (2Mx32)， 4 组，行长度 = 11，列长度 = 8
1	0	010	00	128 Mb (16Mx8)， 4 组，行长度 = 12，列长度 = 10
1	0	010	01	128 Mb (8Mx16)， 4 组，行长度 = 12，列长度 = 9
1	0	010	10	128 Mb (4Mx32)， 4 组，行长度 = 12，列长度 = 8
1	0	011	00	256 Mb (32Mx8)， 4 组，行长度 = 13，列长度 = 10
1	0	011	01	256 Mb (16Mx16)、 4 组、行长度 = 13、列长度 = 9
1	0	011	10	256 Mb (8Mx32)、 4 组、行长度 = 13、列长度 = 8
1	0	100	00	512 Mb (64Mx8)、 4 组、行长度 = 13、列长度 = 11
1	0	100	01	512 Mb (32Mx16)、 4 组、行长度 = 13、列长度 = 10
32 位外部总线低功耗 SDRAM 地址映射 （组、行、列）				
1	1	000	00	16 Mb (2Mx8)， 2 组，行长度 = 11，列长度 = 9
1	1	000	01	16 Mb (1Mx16)， 2 组，行长度 = 11，列长度 = 8
1	1	001	00	64 Mb (8Mx8)， 4 组，行长度 = 12，列长度 = 9
1	1	001	01	64 Mb (4Mx16)， 4 组，行长度 = 12，列长度 = 8
1	1	001	10	64 Mb (2Mx32)， 4 组，行长度 = 11，列长度 = 8
1	1	010	00	128 Mb (16Mx8)， 4 组，行长度 = 12，列长度 = 10
1	1	010	01	128 Mb (8Mx16)， 4 组，行长度 = 12，列长度 = 9
1	1	010	10	128 Mb (4Mx32)， 4 组，行长度 = 12，列长度 = 8
1	1	011	00	256 Mb (32Mx8)， 4 组，行长度 = 13，列长度 = 10
1	1	011	01	256 Mb (16Mx16)、 4 组、行长度 = 13、列长度 = 9
1	1	011	10	256 Mb (8Mx32)、 4 组、行长度 = 13、列长度 = 8
1	1	100	00	512 Mb (64Mx8)、 4 组、行长度 = 13、列长度 = 11
1	1	100	01	512 Mb (32Mx16)、 4 组、行长度 = 13、列长度 = 10

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

片选可以连接至单个存储器件，在这种情况下，片选数据总线宽度与器件宽度相同。或者，片选也可连接至多个外部器件。在这种情况下，片选数据总线宽度是存储器件数据总线宽度的总和。

例如，对于片选连接至：

- 32 位宽的存储设备，选择 32 位宽的地址映射。
- 16 位宽的存储设备，选择 16 位宽的地址映射。
- 4 个 8 位宽的存储设备，选择 32 位宽的地址映射。
- 2 个 8 位宽的存储设备，选择 16 位宽的地址映射。

SDRAM 组选择引脚 BA1 和 BA0 分别连接至地址行 A14 和 A13。

21.7.20 动态存储器 RAS 和 CAS 延迟寄存器

可以通过 DynamicRasCas0:3 编程相关动态存储器的 RAS 和 CAS 等待时间。建议在系统初始化时对这些寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。这些寄存器以一个等待状态访问。

注：在这些寄存器中编入的值必须与 SDRAM 存储器件初始化使用的值保持一致。

表 365. 动态存储器 RASCAS 延迟寄存器 （DYNAMICRASCAS，地址 0x4000 5104 (DYNAMICRASCAS0)、 0x4000 5124 (DYNAMICRASCAS1)、 0x4000 5144 (DYNAMICRASCAS2)、 0x4000 5164 (DYNAMICRASCAS3)）位描述

位	符号	值	描述	复位值
1:0	RAS		RAS 等待时间 （有效到读 / 写延迟）。	11
		0x0	保留。	
		0x1	一个 CCLK 周期。	
		0x2	两个 CCLK 周期。	
		0x3	三个时钟周期 （POR 复位值）。	
7:2	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
9:8	CAS		CAS 等待时间。	11
		0x0	保留。	
		0x1	一个 CCLK 周期。	
		0x2	两个 CCLK 周期。	
		0x3	三个时钟周期 （POR 复位值）。	
31:10	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

21.7.21 静态存储器配置寄存器

静态配置寄存器配置静态存储器。建议在系统初始化时对这些寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。这些寄存器以一个等待状态访问。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 366. 静态存储器配置寄存器 （STATICCONFIG，地址 0x4000 5200 (STATICCONFIG0)、0x4000 5220 (STATICCONFIG1)、 0x4000 5240 (STATICCONFIG2)、 0x4000 5260 (STATICCONFIG3)）位描述

位	符号	值	描述	复位值
1:0	MW		存储器宽度。	0
		0x0	8 位 （POR 复位值）。	
		0x1	16 位。	
		0x2	32 位。	
		0x3	保留。	
2	-	-	保留，用户软件不应向保留位写入1。未定义从保留位读取的值。	-
3	PM		页面模式。 在页面模式中， EMC 最多可以突然发起 4 个外部访问。因此，支持异步页面模式突然发起长度为 4 或更高的器件。不支持异步页面模式可控脉冲长度为 2 的器件，这些器件必须正常访问。	0
		0	禁用 （POR 复位值）。	
		1	异步页面模式已使能 （页面长度四）。	
5:4	-	-	保留，用户软件不应向保留位写入1。未定义从保留位读取的值。	-
6	PC		片选极性。 上电复位时的片选极性值为 0。	0
		0	片选驱动为低电平有效。	
		1	片选驱动为高电平有效。	
7	PB		字节通道状态。 通过字节通道状态位 PB 可连接到不同类型的存储器。对于字节宽静态存储器，来自 EMC 的 BLSn[3:0] 信号通常连接到 WE（写使能）。在这种情况下，对于读操作，所有的 BLSn[3:0] 位必须为高电平。这就意味着字节通道状态 (PB) 位必须为低电平。 16 位宽的静态存储设备通常使 BLSn[3:0] 信号连接到静态存储器中的 UBn 和 LBn （高字节和低字节）信号。在该情形中，对特定字节的写入必须将相应的 UBn 或 LBn 信号断定为 LOW。对于读取，所有的 UB 和 LB 信号必须断定为低电平，以便总线可被驱动。在这种情况下，字节通道状态 (PB) 位必须为高电平。 **注：**当 PB 设为 0 时， WE 信号未定义或为 0。必须将 PB 设为 1，才能使用 WE 信号。	0
		0	对于读操作，BLSn[3:0] 中的所有位必须为高电平。对于写操作，BLSn[3:0] 中相应的有效位为低电平 （POR 复位值）。	
		1	对于读操作，BLSn[3:0] 中相应的有效位为低电平。对于写操作，BLSn[3:0] 中相应的有效位为低电平。	
8	EW		延长等待。 延长等待 (EW) 使用 StaticExtendedWait 寄存器来为读写传输计时，而不是 StaticWaitRd 和 StaticWaitWr 寄存器。这样可以使能更长的事务。[1]	0
		0	延长等待禁用 （POR 复位值）。	
		1	延长等待使能。	
18:9	-	-	保留，用户软件不应向保留位写入1。未定义从保留位读取的值。	-
19	B		缓冲使能 [2]。	0
		0	缓冲已禁用 （POR 复位值）。	
		1	缓冲已使能。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 366. 静态存储器配置寄存器 （STATICCONFIG，地址 0x4000 5200 (STATICCONFIG0)、0x4000 5220 (STATICCONFIG1)、 0x4000 5240 (STATICCONFIG2)、 0x4000 5260 (STATICCONFIG3)）位描述 *（续）*

位	符号	值	描述	复位值
20	P		写保护。	0
		0	无写保护 （POR 复位值）。	
		1	写保护。	
31:21	-	-	保留，用户软件不应向保留位写入1。未定义从保留位读取的值。	-

[1] 等待延长和页面模式不能同时选择。

[2] 即使缓冲区使能位清除， EMC 还是可能执行连发读访问。

21.7.22 静态存储器写使能延迟寄存器

可以通过 StaticWaitWen 寄存器编程从片选到写使能的延迟。建议在系统初始化时对这些寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。这些寄存器以一个等待状态访问。

表 367. 静态存储器写使能延迟寄存器 （STATICWAITWEN，地址 0x4000 5204 (STATICWAITWEN0)、 0x4000 5224 (STATICWAITWEN1)、 0x4000 5244 (STATICWAITWEN2)、 0x4000 5264 (STATICWAITWEN3)）位描述

位	符号	描述	复位值
3:0	WAITWEN	等待写入使能。 自芯片选择断定到写入使能的延迟。 0x0 = 片选有效与写使能之间有 1 个 CCLK 周期延迟（POR 复位值）。 0x1 - 0xF = (n + 1) 个 CCLK 周期延迟。延迟为 (WAITWEN +1) x tCCLK。	0x0
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

21.7.23 静态存储器输出使能延迟寄存器

可以通过 StaticWaitOen 寄存器编程从片选或地址更改 （较晚出现的一个）到输出使能的延迟。建议在系统初始化时对这些寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。这些寄存器以一个等待状态访问。

表 368. 静态存储器输出使能延迟寄存器 （STATICWAITOEN，地址 0x4000 5208 (STATICWAITOEN0)、 0x4000 5228 (STATICWAITOEN1)、 0x4000 5248 (STATICWAITOEN2)、 0x4000 5268 (STATICWAITOEN3)）位描述

位	符号	描述	复位值
3:0	WAITOEN	等待输出使能。 从片选断定到输出使能的延迟。 0x0 = 无延迟 （POR 复位值）。 0x1 - 0xF = n 个周期延迟。延迟为 WAITOEN x tCCLK。	0x0
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

21.7.24 静态存储器读延迟寄存器

可以通过 StaticWaitRd 寄存器编程从片选到读访问的延迟。建议在系统初始化时对这些寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。如果在静态配置寄存器中使能延长等待，则不使用该寄存器。这些寄存器以一个等待状态访问。

表 369. 静态存储器读延迟寄存器 （STATICWAITRD，地址 0x4000 520C (STATICWAITRD0)、0x4000 522C (STATICWAITRD1)、 0x4000 524C (STATICWAITRD2)、 0x4000 526C (STATICWAITRD3)）位描述

位	符号	描述	复位值
4:0	WAITRD	非页面模式读取等待状态，或异步页面模式先读取访问等待状态。 非页面模式读取或异步页面模式读取，先只读： 0x0 - 0x1E = (n + 1) 个 CCLK 周期用于读取访问。对于非顺序读取，其等待状态时间为 (WAITRD + 1) x tCCLK。 0x1F = 32 个 CCLK 周期用于读取访问 （POR 复位值）。	0xB [1]
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 仅 STATICWAITRD0 寄存器的复位值为 0x0B。

21.7.25 静态存储器页面模式读延迟寄存器

可以通过 StaticWaitPage 寄存器编程从异步页面模式连续访问的延迟。建议在系统初始化时对这些寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。该寄存器以一个等待状态访问。

表 370. 静态存储器页面模式读延迟寄存器 （STATICWAITPAGE，地址 0x4000 5210 (STATICWAITPAGE0)、 0x4000 5230 (STATICWAITPAGE1)、 0x4000 5250 (STATICWAITPAGE2)、 0x4000 5270 (STATICWAITPAGE3)）位描述

位	符号	描述	复位值
4:0	WAITPAGE	第一次读等待状态后的异步页面模式读操作。 第一次读操作后的异步页面模式读访问的等待状态次数： 0x0 - 0x1E = （n + 1）个 CCLK 周期的读访问时间。对于顺序访问的异步页面模式读取，第一次读取后页面模式访问的等待状态时间为 (WAITPAGE + 1) x tCCLK。 0x1F = 32 个 CCLK 周期的读访问时间 （POR 复位值）。	0x1F
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

21.7.26 静态存储器写延迟寄存器

可以通过 StaticWaitWr 寄存器编程从片选到写访问的延迟。建议在系统初始化时对这些寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。如果在静态配置寄存器中使能等待延长，则不使用这些寄存器。这些寄存器以一个等待状态访问。

表 371. 静态存储器写延迟寄存器 （STATICWAITWR，地址 0x4000 5214 (STATICWAITWR0)、0x4000 5234 (STATICWAITWR1)、 0x4000 5254 (STATICWAITWR2)、 0x4000 5274 (STATICWAITWR3)）位描述

位	符号	描述	复位值
4:0	WAITWR	写等待状态。 第一次读操作后的写访问 SRAM 等待状态时间： 0x0 - 0x1E = （n + 2）个 CCLK 周期的写访问时间。第一次读操作后的写访问等待状态时间为 WAITWR (n + 2) x tCCLK。 0x1F = 33 个 CCLK 周期的写访问时间 （POR 复位值）。	0x1F
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

21.7.27 静态存储器周转期延迟寄存器

可以通过 StaticWaitTurn 寄存器编程总线周转期的数量。建议在系统初始化时对这些寄存器进行修改，或者在没有当前事务或未完成事务时进行修改。如需确保该操作，可以等待直到 EMC 变为空闲状态，然后进入低功耗或禁用模式。这些寄存器以一个等待状态访问。

表 372. 静态存储器周转期延迟寄存器 （STATICWAITTURN，地址 0x4000 5218 (STATICWAITTURN0)、 0x4000 5238 (STATICWAITTURN1)、 0x4000 5258 (STATICWAITTURN2)、 0x4000 5278 (STATICWAITTURN3)）位描述

位	符号	描述	复位值
3:0	WAITTURN	总线周转期。 0x0 - 0xE = n + 1 个时钟周期。总线周转时间为 (WAITTURN + 1) x tCCLK。 0xF = 16 个时钟周期 （POR 复位值）。	0xF
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

为了防止外部存储器数据总线上的总线争用，WAITTURN 字段控制静态存储器读写访问之间所添加的总线周转期数量。WAITTURN 字段还控制静态存储器和动态存储器访问之间的周转期数量。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

21.8 功能说明

图 43 显示了 EMC 的框图。

在接下来的章节中，我们将对 EMC 模块的功能进行详细介绍。

- AHB 从机寄存器接口。
- AHB 从机存储器接口。
- 数据缓冲区。
- 存储控制器状态机。
- 管脚接口。

注：对于 32 位宽片选，动态存储器输入 / 输出数据采用 SDRAM 的 4 位连发传输。对于 16 位宽片选，采用 SDRAM 的 8 位连发传输。

21.8.1 AHB 从机寄存器接口

通过 AHB 从机寄存器接口模块可使能需要进行编程的 EMC 寄存器。该模块还包含大量寄存器，并执行绝大多数的寄存器地址译码。

如果要避免字节顺序问题的产生，则 EMC 寄存器的所有数据传输必须为 32 位宽。

注：如果某个访问试图使用非字码 （32 位）的大小，则会导致对 AHB 总线的 ERROR 响应，而且传输也被终止。

21.8.2 AHB 从机存储器接口

通过 AHB 从机存储器接口允许外部存储器访问。

21.8.2.1 存储器事务字节序

外部存储器的所有数据传输的字节顺序由配置寄存器的端模式 (N) 位决定。

注：字节顺序变更之前，存储控制器必须处于空闲状态 （查看状态寄存器的 “忙碌” 字段），才能保证数据正确传输。

21.8.2.2 存储器传输大小

存储器事务可以为 8 位、 16 位或 32 位宽。对于任何所尝试的访问，如果其大小超过一个字 （32 位） , 则会引起对 AHB 总线的 ERROR 响应，进而终止传输。

21.8.2.3 写保护存储器区域

向写保护的存储器区域的写事务会对 AHB 总线产生 ERROR 响应，并终止传输。

21.8.3 管脚接口

管脚接口模块提供连接到管脚的接口。

对于 16 位存储器和 32 位存储器， EMC 动态存储器接口要求选择 CLKn 引脚上的所有 EMC_CLK 信号。

对于静态存储器， STATICWAIT 寄存器会以一个 EMC 时钟周期的步幅定义更大延迟 （参见第 21.7.22 节至第 21.7.27 节）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

21.8.4 数据缓冲区

AHB 接口通过缓冲区进行读写操作，以此来提高存储器的带宽并减少事务等待时间。EMC 含有 4 个 16 字的缓冲区。缓冲区可用作读缓冲区、写缓冲区或读写缓冲区。可自动对缓冲区进行分配。

SDRAM 和 SyncFlash 初始化时，必须禁用缓冲区。执行 SyncFlash 命令时也必须禁用。而在正常操作时，则必须使能缓冲区。

对于静态存储器，可以使用静态配置寄存器来使能或禁用缓冲区。

21.8.4.1 写缓冲区

写缓冲区用于：

- 合并写事务，最大程度减少外部事务数量。

  缓冲数据，直至 EMC 能够完成写事务，从而改善 AHB 写等待时间。

  在外部存储器接口上将所有动态存储器写事务转换成四倍字长的可控脉冲。这样可以提高动态存储器的传输效率。
- 减少外部存储器数据流量。这样可以提高存储器带宽并降低功耗。

写缓冲区操作：

- 如果使能缓冲区，则AHB写操作将写入使用率近来最低(LRU)的缓冲区（如果该缓冲区为空）。

  如果 LRU 缓冲区不为空，则缓冲内容会被刷新到存储器，为 AHB 写数据腾出空间。
- 如果缓冲区含有标为经修改的写数据，则在该缓冲区重新分配前将其内容写入存储器。

出现下列情况时刷新写缓冲区：

- 存储控制器状态机未忙于执行外部存储器访问。

  存储控制器状态机未忙于执行外部存储器访问， AHB 接口正在向另一个不同缓冲区写入。

注：对于动态存储器，最小的缓冲区刷新量是四倍字长的数据。而静态存储器的最小缓冲区刷新量是一个字节的数据。

21.8.4.2 读缓冲区

读缓冲区用于：

- 缓冲来自存储器的读请求。之后达到缓冲区的读请求从缓冲区 （并非存储器）来读取数据，从而减少了事务等待时间。

  在外部存储器接口上将所有读事务转换成四倍字长的可控脉冲。这样可以提高动态存储器的传输效率。
- 减少外部存储器数据流量。这样可以提高存储器带宽并降低功耗。

读缓冲区操作：

- 如果使能缓冲区，并且其中一个缓冲区包含读数据，则直接从该缓冲区提供读取数据。
- 如果没有缓冲区包含该读数据，则选用 LRU 缓冲区。如果该缓冲区是经过修改的 （包含写数据），则写数据会刷新到存储器。当一个空缓冲区可用时，读命令提交到存储器。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

通过执行存储器读操作来填充的缓冲区被标记为未经修改（不含有写数据），除非后续的 AHB 传输执行请求所到达的缓冲区的写操作，否则其内容不会被刷新回存储控制器。

21.9 低功耗

在很多系统中，必须通过低功耗睡眠模式来保留存储器系统的内容。EMC 提供一种机制将动态存储器放置于自刷新模式中。

通过软件设置动态控制寄存器中的 SREFREQ 位和轮询 Status 寄存器中的 SREFACK 位可进入自刷新模式。

当存储控制器处于自刷新模式时，对存储器产生的任何操作都被拒绝，并且对 AHB 总线产生错误响应。清除动态控制寄存器中的 SREFREQ 位可使存储器恢复正常操作。刷新要求参见存储器数据手册。

注：当 SDRAM 存储器处于自刷新模式时，静态存储器可以正常访问。

21.9.1 低功耗 SDRAM 深度睡眠模式

EMC 支持 JEDEC 低功耗 SDRAM 深度睡眠模式。如果要进入深度睡眠模式，则可在动态控制寄存器中设置深度睡眠模式 (DP) 位、动态存储器时钟使能位 (CE) 和动态时钟控制位 (CS)。然后，器件会进入低功耗模式；在该模式下，器件掉电且不再刷新。存储器中所有数据丢失。

21.9.2 低功耗 SDRAM 部分阵列刷新

EMC支持JEDEC低功耗SDRAM部分阵列刷新。部分阵列刷新可以通过初始化相应SDRAM存储器件进行编程。当存储器件进入自刷新模式时，仅刷新指定的存储组别。不刷新的存储组别丢失数据内容。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

21.10 外部静态存储器接口

外部存储器连接取决于组宽 （在相应静态配置寄存器中通过 MW 位选择 32 位、 16 位或 8 位）。

如果存储组别配置为 32 位宽，则地址行 A0 和 A1 可用作非地址行。如果存储组别配置为 16 位宽，则 A0 不是必需的。不过， 8 位宽的存储组别要求所有地址行下行到 A0。使用 SYSCON 寄存器可以配置 A1 和 / 或 A0 行，从而提供地址或非地址功能。

下列图片中的 “a_b” 符号指的是数据总线中的最高位地址行。“a_m” 符号指的是外部存储器接口中使用的存储器芯片的最高位地址行。

当外部存储器用作无闪存器件的外部启动存储器时，关于如何连接 EMC，请参见第 5.2 节。存储组别 1 和 2 的存储组别宽由 BOOT 引脚的设置决定。

21.10.1 32 位宽存储组别连接

a. 连接到 4 个 8 位存储器芯片的 32 位宽存储组别

b. 连接到两个 16 位存储器芯片的 32 位宽存储组别

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

c. 连接到 1 个 8 位存储器芯片的 32 位宽存储组别

图 45. 32 位存储组别外部存储器接口 （位 MW = 10）

21.10.2 16 位宽存储组别连接

a. 连接到两个 8 位存储器芯片的 16 位宽存储组别

b. 连接到 1 个 16 位存储器芯片的 16 位宽存储组别

图 46. 16 位存储组别外部存储器接口 （位 MW = 01）

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

21.10.3 8 位宽存储组别连接

图 47. 8 位存储组别外部存储器接口 （位 MW = 00）

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

22.1 本章导读

SPIFI 在所有 LPC43xx 器件上可用。

22.2 基本配置

SPIFI 的配置如下：

- 参见表 373 以了解时钟与电源控制。
- SPIFI 可由 SPIFI_RST 复位 （复位编号 53）。

表 373. SPIFI 时钟和电源控制

	基本时钟	分支时钟	工作频率
SPIFI AHB 寄存器时钟	BASE_M4_CLK	CLK_M4_SPIFI	高达 204 MHz
SPIFI 串行时钟输入	BASE_SPIFI_CLK	SPIFI_CLK	120 MHz

注：所有器件都可使用 SPIFI 启动。参见第 5.3.4.5 节。

22.3 功能要点

- 与外部闪存的四通道 SPI 闪存接口 (SPIFI)。
- 传输速率达 SPIFI_CLK/2 字节 / 秒。
- 外部闪存直接进行存储器映射以便快速访问。
- 支持 1 位、 2 位和 4 位双向串行协议。
- 半双工协议与不同的供应商和设备兼容。
- 可通过 GPDMA 访问 SPIFI 存储器。
- 关于软件驱动程序库，请登陆 LPCware 网站。

22.4 简介

SPI 闪存接口 (SPIFI) 支持低成本串行闪存连接到 Cortex-M4 处理器，相对于引脚数量更多的并行闪存设备而言，性能损失并不大。

许多串行闪存设备使用半双工指令驱动的 SPI 协议进行系统设置和初始化。四通道设备则使用半双工指令驱动的 4 位协议进行正常操作。不同的串行闪存供应商和产品需要不同的指令和指令格式。SPIFI 为此提供了充分的支持，可兼容大部分闪存，并预留了扩展功能，确保兼容今后的产品。

当软件读取存储器映射中两个只读串行闪存区域的任意一个时，串行闪存设备会对软件发送的命令或由 SPIFI 自动发送的命令做出响应 （参见表 374）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 374. SPIFI 闪存映射

存储器	地址
SPIFI 数据	0x1400 0000 至 0x17FF FFFF （使用该存储器区域来调试代码）。 0x8000 0000 至 0x87FF FFFF （程序计数器位于该存储器区域时，调试将不起作用）。 **注：**这些是 LPC43xx 中分配给 SPIFI 的空间。相同的数据出现在第一区和第二区的前半部分。这些区域最多允许 64 MB 和 128 MB 的 SPI 闪存（分别）映射到 Cortex-M4 的存储器空间。在实际应用中，可用空间上限为所连接设备的大小

22.5 引脚说明

表 375. SPIFI 引脚描述

引脚	方向	描述
SPIFI_SCK	O	闪存串行时钟，仅在 MOSI/IO0、 MISO/IO1 以及 IO3:2 线的有效位期间开启。
$\overline{\text{SPIFI_CS}}$	O	闪存芯片选择，当指令正在进行时则为低电平，无指令进行时为高电平。在单个串行从机的典型情况下，该信号可直接连接至设备。如果有多个串行从机连接，软件和片外硬件应当结合使用通用 I/O 信号和该信号，以产生各个从机的芯片选择信号。
SPIFI_MOSI 或 IO0	I/O	这是除了四路 / 双路输入数据字段以外的输出。四路 / 双路输入数据字段后，它会在 $\overline{\text{CS}}$ 变为高电平后的 1 个串行时钟期周期再变为输出。
SPIFI_MISO 或 IO1	I/O	这是四路 / 双路操作代码、地址、中间和输出数据字段的输出，以及 SPI 模式和四路 / 双路输入数据字段的输入。四路 / 双路模式的输入数据字段后，它会在 $\overline{\text{CS}}$ 变为高电平后的 1 个串行时钟期周期再变为输出。
SPIFI_SIO[3:2]	I/O	这些是四路操作代码、地址、中间和输出数据字段的输出，以及四路输入数据字段的输入。如果闪存不具备四路功能，这些引脚可以指定给 GPIO 或其他功能。

22.6 支持的 QSPI 设备

表 224 显示了经认证支持 SPIFI API 的供应商 QSPI 设备列表。可使用其他设备，这些设备将在基本的单一 SPI 模式下低速运行。

注：所有 QSPI 设备都已在 3.3 V 工作电压下经过测试。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 376. 支持的 QSPI 设备

制造商	设备名称
AMIC	A25L512、A25L010、A25L020、A25L040、A25L080、A25L016、A25L032、A25LQ032
Atmel	AT25F512B、 AT25DF021、 AT25DF041A、 AT25DF081A、 AT25DF161、 AT25DQ161、 AT25DF321A、 AT25DF641
Chingis	Pm25LD256、 Pm25LD512、 Pm25LD010、 Pm25LD020、 Pm25LD040、 Pm25LQ032
Elite (ESMT)	F25L08P、 F25L16P、 F25L32P、 F25L32Q
Eon	EN25F10、 EN25F20、 EN25F40、 EN25Q40、 EN25F80、 EN25Q80、 EN25QH16、 EN25Q32、 EN25Q64、 EN25Q128
Gigadevice	GD25Q512、 GD25Q10、 GD25Q20、 GD25Q40、 GD25Q80、 GD25Q16、 GD25Q32、 GD25Q64
Macronix	MX25L8006、 MX25L8035、 MX25L8036、 MX25U8035[1]、 MX25L1606、 MX25L1633、 MX25L1635、 MX25L1636、 MX25U1635[1]、 MX25L3206、 MX25L3235、 MX25L3236、 MX25U3235[1]、 MX25L6436、 MX25L6445、 MX25L6465、MX25L12836、MX25L12845、MX25L12865、MX25L25635、MX25L25735
Numonyx	M25P10、M25P20、M25P40、M25P80、M25PX80、M25P16、M25PX16、M25P32、M25PX32、M25P64、M25PX64、N25Q032、N25Q064、N25Q128
Spansion	S25FL004K、 S25FL008K、 S25FL016K、 S25FL032K、 S25FL032P、 S25FL064K、 S25FL064P、 S25FL129P
SST	SST26VF016、 SST26VF032、 SST25VF064
Winbond	W25Q40、 W25Q80、 W25Q16、 W25Q32、 W25Q64

[1] 这些器件需要电平转换电路（可能会影响性能）。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

23.1 本章导读

USB0 主机 / 设备 /OTG 控制器在器件 LPC4350、 LPC4330 和 LPC4320 上可用。

23.2 基本配置

USB0 主机 / 设备 /OTG 控制器配置如下：

- 参见表 377 以了解时钟与电源控制。
- USB0 可由 USB0_RST 复位 （复位编号 17）。
- USB0 OTG 中断连接到 NVIC 中的中断插槽 #8，唤醒请求指示符连接到事件路由器中的插槽 #9。
- 通过CREG模块控制USB0片上PHY的电源（参见表38）。除非已选择USB0引导模式，片上 PHY 默认为掉电。要使用 USB0 控制器，需使能 CREG0 寄存器位 5 的 PHY。
- SOF/VF 指示器可连接到定时器 3 或者通过 GIMA 连接到 SCT （参见第 23.7.7 节和表 138）。

表 377. USB0 时钟和电源控制

	基本时钟	分支时钟	工作频率	备注
USB0 时钟	BASE_USB0_CLK	CLK_USB0	480 MHz	使用USB0专用的PLL0USB。在所有 USB 模式中 （低速、全速和高速模式），必须将 CLK_USB0 设置为 480 MHz 时钟。
USB0 寄存器接口时钟	BASE_M4_CLK	CLK_M4_USB0	高达 204 MHz	

23.3 功能要点

- 包含符合片上高速 UTMI+ 标准的收发器 (PHY)。
- 支持所有符合高速、全速和低速 USB 的外设。
- 符合通用串行总线规范 2.0。
- 符合 USB On-The-Go 补充规范。
- 符合增强型主机控制器接口 (EHCI) 规范。
- 支持 USB 2.0 自动模式发现。
- 支持 OTG 外设的 HNP 和 SRP 软件。
- 支持功率管理。
- 支持六个逻辑端点 （包括一个控制端点）组成总共 12 个物理端点。
- 这个模块有其自己集成的 DMA 引擎。
- 可以和音频 PLL 一起用于 USB 流应用。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

23.4 简介

通用串行总线 （USB）是一种标准协议，用于实现多种类型设备之间的互连，以便交换数据或另作它用。许多便携式设备的优势在于，可在不受主机电脑干预的情况下，通过 USB 接口来彼此交换数据。在 USB 上添加 OTG 功能，则可在实现这种功能的同时保留标准 USB 协议的优势。USB 设备的示例包括：电脑、鼠标、键盘、MP3 播放器、数码相机、USB 存储设备 （USB 棒）。

23.4.1 功能框图

图 48. 高速 USB OTG 框图

23.4.2 关于 USB OTG

USB OTG 模块可在设备模式和主机模式中使用。这就意味着，可以连接到电脑以交换数据，并且可连接到其他 USB 设备，如数码相机或 MP3 播放器。

23.4.3 USB 首字母缩略词和缩写

表 378. USB 相关的首字母缩略词

首字母缩略词	描述
ATX	模拟收发器
DCD	设备控制器驱动器
dQH	设备端点队列头
dTD	设备传输描述符
EOP	数据包结束
EP	端点
FS	全速
HCD	主机控制器驱动器
HS	高速
LS	低速
MPS	最大数据包大小

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 378. USB 相关的首字母缩略词 *（续）*

首字母缩略词	描述
NAK	否定应答
OTG	On-The-Go
PID	数据包标识
QH	队列头
SE0	单端 0
SOF	帧起始
TT	事务转译器
USB	通用串行总线

23.4.4 发送和接收缓冲区

USB OTG 控制器包含一个发送缓冲区 （用于存储 USB 上所要发送的数据）以及一个接收缓冲区 （用于存储从 USB 收到的数据）。接收缓冲区包含 256 个字。发送缓冲区的各个端点在设备模式下包含 128 个字，在主机模式下包含 512 个字。

23.4.5 固定的端点配置

表 379 列出了所支持的端点配置。最大数据包大小 （参见表 380）取决于端点类型和设备配置 （低速、全速或高速）。

表 379. 固定的端点配置

逻辑端点	物理端点	端点类型	方向
0	0	控制	输出
0	1	控制	输入
1	2	中断 / 批量 / 同步	输出
1	3	中断 / 批量 / 同步	输入
2	4	中断 / 批量 / 同步	输出
2	5	中断 / 批量 / 同步	输入
3	6	中断 / 批量 / 同步	输出
3	7	中断 / 批量 / 同步	输入
4	8	中断 / 批量 / 同步	输出
4	9	中断 / 批量 / 同步	输入
5	10	中断 / 批量 / 同步	输出
5	11	中断 / 批量 / 同步	输入

表 380. USB 数据包大小

端点类型	速度	数据包大小 （字节）
控制	低速	8
	全速	8、 16、 32 或 64
	高速	64
同步	低速	不适用
	全速	高达 1023
	高速	高达 1024

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 380. USB 数据包大小 *（续）*

端点类型	速度	数据包大小 （字节）
中断	低速	高达 8
	全速	高达 64
	高速	高达 1024
批量	低速	不适用
	全速	8、 16、 32 或 64
	高速	8、 16、 32、 64 或 512

23.5 引脚说明

表 381. USB0 引脚描述

引脚功能	方向	描述
USB0_IND0	O	端口 LED 指示灯控制输出。
USB0_IND1	O	端口 LED 指示灯控制输出。
USB0_PWR_FAULT	O	指示过流状况的端口电源故障信号，该信号监控 USB 总线上的过流状况 （检测过电流状况需要外部电路）。
USB0_PPWR	O	VBUS 驱动信号（发送至外部充电泵或电源管理单元）；指示 VBUS 必须被驱动（有效高电平）。 复位时，添加一个下拉电阻以禁用电源开关。和其他恩智浦 LPC 器件使用的 USB_PPWR 相比，该信号具有相反的极性。
USB0_DP	I/O	USB0 双向 D+ 线。D+ 线具有一个内部 1.5 kΩ 上拉电阻。软件设置 USBCMD 寄存器的 RS 位 （位 0）时，使能上拉电阻，并且 USB0 控制器会看到 VBUS 引脚上的有效 VBUS 电压电平 （高于 ~1.8V）。
USB0_DM	I/O	USB0 双向 D– 线路。D- 线具有一个内部 1.5 kΩ 上拉电阻。软件设置 USBCMD 寄存器的 RS 位 （位 0）时，使能上拉电阻，并且 USB0 控制器会看到 VBUS 引脚上的有效 VBUS 电压电平 （高于 ~1.8V）。
USB0_VBUS	I	VBUS 引脚 （USB 线缆电源）。该引脚包括一个 64 kOhm （典型）± 16 kOhm 的内部下拉电阻。对于最大负载 C_L = 6.5 uF 和最大下拉电阻 R_{pd} = 80 kOhm，当 VBUS 信号不再被驱动时，从 VBUS = 5 V 下降至 VBUS = 0.2 V 约需 2 秒。
USB0_ID	I	向收发器指明是连接到器件 A （USB0_ID 低）还是器件 B （USB0_ID 高）。对于 OTG，该引脚具有一个内部上拉电阻。
USB0_RREF		用于电流基准的 12.0 kΩ （精度 1%）板载接地电阻。
USB0_VDDA3V3_ DRIVER		驱动器独立模拟 3.3 V 电源。
USB0_VDDA3V3		USB 3.3 V 独立电源电压。
USB0_VSSA_TERM		用于终结电阻独立基准的专用模拟接地。
USB0_VSSA_REF		基准电流和电压产生的专用独立模拟接地。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

23.6 寄存器描述

表 382. 寄存器访问缩写

缩写	描述
R/W	读 / 写
R/WC	读 / 写 1 清零
R/WO	读 / 写一次
RO	只读
WO	只写

表 383. 寄存器简介：USB0 OTG 控制器 （寄存器基址 0x4000 6000）

名称	访问类型	地址偏移	描述	复位值	USB0 启动后的复位值	参考
-	-	0x000 - 0x0FF	保留	-	-	
设备 / 主机功能寄存器						
CAPLENGTH	RO	0x100	功能寄存器长度	0x0100 0040	0x0100 0040	表 384
HCSPARAMS	RO	0x104	主机控制器结构参数	0x0001 0011	0x0001 0011	表 385
HCCPARAMS	RO	0x108	主机控制器功能参数	0x0000 0006	0x0000 0006	表 386
DCIVERSION	RO	0x120	设备接口版本号	0x0000 0001	0x0000 0001	表 387
DCCPARAMS	RO	0x124	设备控制器功能参数	0x0000 0186	0x0000 0186	表 388
-	-	0x128 - 0x13C	保留			-
设备 / 主机操作寄存器						
USBCMD_D	R/W	0x140	USB 命令 （设备模式）	0x0008 0000	0	表 389
USBCMD_H	R/W	0x140	USB 命令 （主机模式）	0x0008 0000	-	表 390
USBSTS_D	R/W	0x144	USB 状态 （设备模式）	0	0x0000 0080	表 392
USBSTS_H	R/W	0x144	USB 状态 （主机模式）	0	-	表 393
USBINTR_D	R/W	0x148	USB 中断使能 （设备模式）	0	0x0001 0147	表 394
USBINTR_H	R/W	0x148	USB 中断使能 （主机模式）	0	-	表 395
FRINDEX_D	R/W	0x14C	USB 帧索引 （设备模式）	0	0x0000 1F20	表 396
FRINDEX_H	R/W	0x14C	USB 帧索引 （主机模式）	0	-	表 397
-	-	0x150	保留			-
DEVICEADDR	R/W	0x154	USB 设备地址 （设备模式）	0	0x0400 0000	表 399
PERIODICLISTBASE	R/W	0x154	帧列表基准地址 （主机模式）	0	-	表 400

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 383. 寄存器简介：USB0 OTG 控制器 （寄存器基址 0x4000 6000）*（续）*

名称	访问类型	地址偏移	描述	复位值	USB0 启动后的复位值	参考
ENDPOINTLISTADDR	R/W	0x158	存储器中端点列表地址	0	0x2000 0000	表 401
ASYNCLISTADDR	R/W	0x158	异步列表地址	0	0	表 402
TTCTRL	R/W	0x15C	嵌入式 TT 的异步缓冲状态 （主机模式）	0	0	表 403
BURSTSIZE	R/W	0x160	可编程的突发流量	0	0	表 404
TXFILLTUNING	R/W	0x164	主机发送预缓冲数据包调谐 （主机模式）	0	0	表 405
-	-	0x168 - 0x170	保留	-	-	
BINTERVAL	R/W	0x174	虚拟帧长度	0	0	表 406
ENDPTNAK	R/W	0x178	端点 NAK （设备模式）	0	0	表 407
ENDPTNAKEN	R/W	0x17C	端点 NAK 使能 （设备模式）	0	0	表 408
-	-	0x180	保留	-	-	
PORTSC1_D	R/W	0x184	端口 1 状态 / 控制 （设备模式）	0	0x3C00 0004	表 409
PORTSC1_H	R/W	0x184	端口 1 状态 / 控制 （主机模式）	0	-	表 410
-	-	0x188 - 0x1A0		-	-	-
OTGSC	R/W	0x1A4	OTG 状态和控制	0	0x0020 0D09	表 412
USBMODE_D	R/W	0x1A8	USB 设备模式 （设备模式）	0	0xA	表 413
USBMODE_H	R/W	0x1A8	USB 设备模式 （主机模式）	0	-	表 414
设备端点寄存器						
ENDPTSETUPSTAT	R/W	0x1AC	端点设置状态	0	0	表 415
ENDPTPRIME	R/W	0x1B0	端点初始化	0	0	表 416
ENDPTFLUSH	R/W	0x1B4	端点取消初始化	0	0	表 417
ENDPTSTAT	RO	0x1B8	端点状态	0	0	表 418
ENDPTCOMPLETE	R/W	0x1BC	端点完成	0	0	表 419
ENDPTCTRL0	R/W	0x1C0	端点控制 0	0	0	表 420
ENDPTCTRL1	R/W	0x1C4	端点控制 1	0	0	表 421
ENDPTCTRL2	R/W	0x1C8	端点控制 2	0	0	表 421
ENDPTCTRL3	R/W	0x1CC	端点控制 3	0	0	表 421
ENDPTCTRL4	R/W	0x1D0	端点控制 4	0	0	表 421
ENDPTCTRL5	R/W	0x1D4	端点控制 5	0	0	表 421

23.6.1 寄存器使用

寄存器接口具有针对设备模式的位功能和针对主机模式的位功能。但是，在 OTG 操作期间，必须执行与控制器模式无关的任务。

使控制器脱离主机模式或设备模式的唯一方法是设置控制器的复位位。因此，还必须执行与控制器复位和控制器模式无关的 OTG 任务。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 49. USB 控制器模式

以下寄存器和寄存器位将用于 OTG 操作。这些寄存器位的值与控制器模式无关，且不会受到 USBCMD 寄存器中复位位写操作的影响。

- 所有标识寄存器
- 所有设备 / 主机功能寄存器
- OTGSC 寄存器的所有位 （第 23.6.16 节）
- PORTSC 寄存器的下列位 （第 23.6.15 节）：
  - PTS （并行接口选择）
  - STS （串行收发器选择）
  - PTW （并行收发器宽度）
  - PHCD （PHY 低功耗挂起）
  - WKOC、 WKDC、 WKCN （唤醒信号）
  - PIC[1:0] （端口指示器）
  - PP （端口电源）

23.6.2 设备 / 主机功能寄存器

表 384. CAPLENGTH 寄存器 （CAPLENGTH – 地址 0x4000 6100）位描述

位	符号	描述	复位值	访问类型
7:0	CAPLENGTH	指示添加到操作寄存器开始处寄存器基址的偏移量	0x40	RO
23:8	HCIVERSION	该主机控制器支持的 EHCI 修订版本号的 BCD 编码。	0x100	RO
31:24	-	这些位将保留且应设为 0。	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 385. HCSPARAMS 寄存器 （HCSPARAMS – 地址 0x4000 6104） 位描述

位	符号	描述	复位值	访问类型
3:0	N_PORTS	下行端口数量。该字段指定该主机控制器上提供的物理下行端口数量。	0x1	RO
4	PPC	端口电源控制。该字段指示主机控制器实施是否包括端口电源控制。	0x1	RO
7:5	-	这些位将保留且应设为 0。	-	-
11:8	N_PCC	每个辅助控制器的端口数量。该字段指示每个内部辅助控制器支持的端口数量。	0x0	RO
15:12	N_CC	辅助控制器的数量。该字段指示与该 USB2.0 主机控制器关联的辅助控制器的数量。	0x0	RO
16	PI	端口指示器。该位指示端口是否支持端口指示器控制。	0x1	RO
19:17	-	这些位将保留且应设为 0。	-	-
23:20	N_PTT	每个事务转译器的端口数量。该字段指示 USB2.0 主机控制器内分配给每个事务转译器的端口数量。	0x0	RO
27:24	N_TT	事务转译器的数量。该字段指示与 USB2.0 主机控制器关联的嵌入式事务转译器的数量。	0x0	RO
31:28	-	这些位将保留且应设为 0。	-	-

表 386. HCCPARAMS 寄存器 （HCCPARAMS - 地址 0x4000 6108）位描述

位	符号	描述	复位值	访问类型
0	ADC	64 位寻址能力。如果为零，则不支持 64 位寻址能力。	0	RO
1	PFL	可编程的帧列表标志。如果设为 1，则系统软件可以指定和使用小型帧列表，并通过 USBCMD 寄存器帧列表大小字段配置主机控制器。帧列表必须始终在 4K 边界上对齐。该要求可确保帧列表始终在物理上连通。	1	RO
2	ASP	异步调度表暂停功能。该位设为 1 时，主机控制器支持异步调度表中的高速队列头暂停功能。通过使用 USBCMD 寄存器的异步调度表暂停模式使能和异步调度表暂停模式计数字段，可以禁用或使能该功能和设置到特定级别。	1	RO
7:4	IST	同步调度阈值。该字段指示软件可以更新的、较执行主机控制器当前位置更为可靠的同步调度。	0	RO
15:8	EECP	EHCI 扩展能力指针。该可选字段指示是否存在能力列表。	0	RO
31:16	-	这些位将保留且应设为 0。	-	-

表 387. DCIVERSION 寄存器 （DCIVERSION - 地址 0x4000 6120）位描述

位	符号	描述	复位值	访问类型
15:0	DCIVERSION	设备控制器接口符合此寄存器中接口版本号的双字节 BCD 编码。	0x1	RO

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 388. DCCPARAMS （地址 0x4000 6124）

位	符号	描述	复位值	访问类型
4:0	DEN	设备端点号。	0x4	RO
6:5	-	这些位将保留且应设为 0。	-	-
7	DC	正常设备。	0x1	RO
8	HC	正常主机。	0x1	RO
31:9	-	这些位将保留且应设为 0。	-	-

23.6.3 USB 指令寄存器 (USBCMD)

主机 / 设备控制器执行该寄存器指示的命令。

23.6.3.1 设备模式

表 389. 设备模式中的 USB 指令寄存器 （USBCMD_D - 地址 0x4000 6140）位描述

位	符号	值	描述	访问类型	复位值
0	RS		运行 / 停止	R/W	0
		0	向该位写入 0 将导致分离事件。		
		1	向该位写入 1 将导致设备控制器在 USB_DP 上使能上拉，并启动连接事件。该控制位不会直接连接到上拉启动，因为跳变至高速模式后将禁用上拉。在设备控制器正确初始化之前，软件应使用该位防止发生连接事件。		
1	RST		控制器复位。 软件使用该位来复位控制器。复位过程完成时，该位将由主机 / 设备控制器设为 0。软件无法通过向该寄存器写入 0 来提前终止复位过程。	R/W	0
		0	当复位过程完成时，由硬件设为 0。		
		1	当软件向该位写入 1 时，设备控制器会将其内部管道、定时器、计数器和状态机等复位为初始值。建议在设备处于连接状态时，不要向该位写入 1，因为无法定义对连接主机的影响。为了确保在启动设备控制器复位前，设备未处于连接状态下，应刷新所有已启动的端点，且 USBCMD Run/Stop 位应设为 0。		
3:2	-	-	不适用于设备模式。	-	0
4	-	-	不适用于设备模式。	-	0
5	-	-	不适用于设备模式。	-	0
6	-	-	不适用于设备模式。在选择设备模式后向该位写入1，将产生未定义的结果。	-	-
7	-	-	保留。这些位应设为 0。	-	-
9:8	-	-	不适用于设备模式。	-	-
10	-		保留。这些位应设为 0。	-	0
11	-	-	不适用于设备模式。	-	
12	-		保留。这些位应设为 0。	-	0
13	SUTW		设置绊网 在处理设置数据包期间，此位用作信号灯，以确保 DCD 从 QH 获得的 8 字节设置数据负载未损坏。如果设置锁定模式关闭（见 USBMODE 寄存器），在 DCD 从 QH 复制上一个设置数据包的设置数据负载时，新的设置数据也同时到达，则会存在危险。该位的设置和清零由软件来进行，如果存在危险，则由硬件来清零。（参见第 23.10 节）。	R/W	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 389. 设备模式中的 USB 指令寄存器 （USBCMD_D - 地址 0x4000 6140）位描述 *（续）*

位	符号	值	描述	访问类型	复位值
14	ATDTW		添加 dTD 绊网 该位用作信号量，可确保新的 dTD 正确添加到工作 （已启动）端点的链接列表。该位在添加新 dTD 的过程中由软件来设置和清零。另请参见第 23.10 节。 当其状态机处于危险区域时，将 dTD 添加到已启动的端点可能无法识别，该位还将由硬件清零。	R/W	0
15	-		不适用于设备模式。	-	-
23:16	ITC		中断阈值控制。 系统软件使用该字段来设置主机 / 设备控制器发出中断的最大速率。 ITC 包含最大的中断时间间隔 （以微帧为单位）。有效值如下所示。保留所有其它值。 0x0 = 立即 （无阈值） 0x1 = 1 微帧。 0x2 = 2 微帧。 0x8 = 8 微帧。 0x10 = 16 微帧。 0x20 = 32 微帧。 0x40 = 64 微帧。	R/W	0x8
31:24	-		保留		0

23.6.3.2 主机模式

表 390. 主机模式中的 USB 指令寄存器 （USBCMD_H - 地址 0x4000 6140）位描述 – 主机模式

位	符号	值	描述	访问类型	复位值
0	RS		运行 / 停止	R/W	0
		0	如果将该位设为 0，则主机控制器会完成 USB 上的当前事务，然后停止。状态寄存器中的 HC Halted 位指示主机控制器何时完成事务以及何时进入停止状态。除非主机控制器处于停止状态（即 USBSTS 寄存器中的 HCHalted 为 1），否则软件不会向该字段写入 1。		
		1	在设为 1 时，主机控制器会继续执行调度。只要将该位设为 1，主机控制器就会继续执行。		
1	RST		控制器复位。 软件使用此位复位控制器。复位过程完成时，该位将由主机 / 设备控制器设为 0。软件无法通过向该寄存器写入 0 来提前终止复位过程。	R/W	0
		0	复位过程完成时，该位将由硬件设为 0。		
		1	当软件向该位写入 1 时，主机控制器会将其内部管道、定时器、计数器和状态机等复位为初始值。当前正在 USB 上处理的任何事务将立即终止。无法在下行端口驱动 USB 复位。当 USBSTS 寄存器中的 HCHalted 位为 0 时，软件不会将该位设为 1。尝试复位主动运行的主机控制器将导致未定义的行为。		
2	FS0		帧列表大小位的位 0。参见表 391。 该字段指定帧列表大小，而帧列表控制帧索引寄存器中用于帧列表当前索引的位。需要注意的是，该字段由 USBCMD 位 15、3 和 2 组成。		0
3	FS1		帧列表大小位的位 1。参见表 391。		0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 390. 主机模式中的 USB 指令寄存器 （USBCMD_H - 地址 0x4000 6140）位描述 – 主机模式 *（续）*

位	符号	值	描述	访问类型	复位值
4	PSE		该位控制主机控制器是否跳过处理定期调度。	R/W	0
		0	不处理定期调度。		
		1	使用 PERIODICLISTBASE 寄存器访问定期调度。		
5	ASE		该位控制主机控制器是否跳过处理异步调度。	R/W	0
		0	不处理异步调度。		
		1	使用 ASYNCLISTADDR 访问异步调度。		
6	IAA		该位由软件用作提示信息，告知主机控制器在下次超前进行异步调度时发出中断	R/W	0
		0	主机控制器将 USBSTS 寄存器中的异步推进时中断状态位设为 1 后，便会将该位设为 0。		
		1	软件必须向该位写入 1，以启动提示信息。 当主机控制器退出所有相关缓存的调度状态时，便会设置 USBSTS 寄存器中的异步推进时中断状态位。如果 USBINTR 寄存器中的异步推进使能时中断位为 1，则主机控制器将在下一个中断阈值中确认中断。 在异步调度不工作时，软件不会向该位写入 1。否则，将会产生未定义的结果。		
7	-	-	保留		0
9:8	ASP1_0		异步调度表暂停模式 包含在继续异步调度表前主机控制器可以从异步调度表高速队列头执行的连续事务数。有效值为 0x1 至 0x3。 **注：**当暂停模式使能位为 1 时，软件不得向此位写入 00，因为这将导致未定义的行为。	R/W	11
10	-	-	保留。	-	0
11	ASPE		异步调度表暂停模式使能	R/W	1
		0	暂停模式禁用。		
		1	暂停模式使能。		
12	-	-	保留。	-	0
13	-	-	不适用于主机模式。	-	
14	-	-	保留。	-	0
15	FS2		帧列表大小位的位 2。参见表 391。	-	0
23:16	ITC		中断阈值控制。 系统软件使用该字段来设置主机 / 设备控制器发出中断的最大速率。ITC 包含最大的中断时间间隔（以微帧为单位）。有效值如下所示。保留所有其它值。 0x0 = 立即 （无阈值） 0x1 = 1 微帧。 0x2 = 2 微帧。 0x8 = 8 微帧。 0x10 = 16 微帧。 0x20 = 32 微帧。 0x40 = 64 微帧。	R/W	0x8
31:24	-		保留		0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 391. 帧列表大小值

USBCMD 第 15 位	USBCMD 第 3 位	USBCMD 第 2 位	帧列表大小
0	0	0	1024 元素 （4096 字节） - 默认值
0	0	1	512 元素 （2048 字节）
0	1	0	256 元素 （1024 字节）
0	1	1	128 元素 （512 字节）
1	0	0	64 元素 （256 字节）
1	0	1	32 元素 （128 字节）
1	1	0	16 元素 （64 字节）
1	1	1	8 元素 （32 字节）

23.6.4 USB 状态寄存器 (USBSTS)

该寄存器指示主机 / 设备控制器的各种状态及任何挂起的中断。软件向该寄存器写入 1，即可将该位设为 0。

注：该寄存器不会指示因串行总线上的事务所产生的状态。

23.6.4.1 设备模式

表 392. 设备模式的 USB 状态寄存器 （USBSTS_D - 地址 0x4000 6144）寄存器位描述

位	符号	值	描述	复位值	访问类型
0	UI		USB 中断	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	在传输描述符 （TD）设置了完成时中断 （IOC）位的情况下，如果中断是由 USB 事务完成所致，则该位将由主机 / 设备控制器进行设置。 当检测到短数据包时，该位也将由主机 / 设备控制器设置。短数据包指收到的实际字节数少于预期字节数。		
1	UEI		USB 错误中断。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	如果 USB 事务的完成导致错误条件的出现，则该位将由主机 / 设备控制器设置。如果发生错误中断的 TD 也设置了其完成时中断 （IOC）位，则该位将和 USBINT 位一并设置。设备控制器仅检测恢复信号 （参见第 23.10.11.6 节）。		
2	PCI		端口更改检测。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	当端口控制器进入全速或高速工作状态时，设备控制器会将该位设为 1。当端口控制器因复位或挂起事件而退出全速或高速工作状态时，通知机制分别为收到 USB 复位位 （URI）和 DCSuspend 位 （SLI）。		
3	-		不适用于设备模式。		-
4	-		保留。	0	-
5	AAI		不适用于设备模式。	0	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 392. 设备模式的 USB 状态寄存器 （USBSTS_D - 地址 0x4000 6144）寄存器位描述 *（续）*

位	符号	值	描述	复位值	访问类型
6	URI		收到 USB 复位	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	当设备控制器检测到USB复位并进入默认状态时，该位将设为 1。		
7	SRI		收到 SOF	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	当设备控制器检测到（微）帧起始时，该位将设为 1。当帧起始 （SOF）太迟时，设备控制器将自动设置该位，以指示预期的 SOF。因此，在设备 FS 模式下，将大约每 1 ms 对该位进行设置，而在 HS 模式下，则将每 125 µs 对其进行设置，并将与收到的实际 SOF 同步。由于设备控制器在连接前会初始化为 FS，因此该位在预载期间，将按 1ms 的时间间隔设置，以便连接和线性调频。		
8	SLI		DCSuspend	0	R/WC
		0	设备控制器退出挂起状态时，便会将该位清零。软件向该位写入 1，即可将该位清零。		
		1	当设备控制器从活动状态进入挂起状态时，该位将设为 1。		
11:9	-	-	保留。软件只会向保留位写入 0。		
12	-	-	不适用于设备模式。	0	
13	-	-	不适用于设备模式。	0	
14	-	-	不适用于设备模式。	0	
15	-	-	不适用于设备模式。	0	
16	NAKI		NAK 中断位	0	RO
		0	如果所有使能的 TX/RX 端点 NAK 位都被清零，则该位将由硬件自动清零。		
		1	在特定端点的 TX/RX 端点 NAK 位和相应的 TX/RX 端点 NAK 使能位设置时，将由硬件设置该位。		
17	-	-	保留。软件只会向保留位写入 0。	0	-
18	-		不适用于设备模式。	0	-
19	-		不适用于设备模式。	0	-
31:20	-	-	保留。软件只会向保留位写入 0。		-

23.6.4.2 主机模式

表 393. 主机模式的 USB 状态寄存器 （USBSTS_H - 地址 0x4000 6144）寄存器位描述

位	符号	值	描述	复位值	访问类型
0	UI		USB 中断 （USBINT）	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	在传输描述符 （TD）设置了完成时中断 （IOC）位的情况下，如果中断是由 USB 事务完成所致，则该位将由主机 / 设备控制器进行设置。 当检测到短数据包时，该位也将由主机 / 设备控制器设置。短数据包指收到的实际字节数少于预期字节数。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 393. 主机模式的 USB 状态寄存器 （USBSTS_H - 地址 0x4000 6144）寄存器位描述 *（续）*

位	符号	值	描述	复位值	访问类型
1	UEI		USB 错误中断 （USBERRINT）	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	如果 USB 事务的完成导致错误条件的出现，则该位将由主机 / 设备控制器设置。如果发生错误中断的 TD 也设置了其完成时中断 （IOC）位，则该位将和 USBINT 位一并设置。		
2	PCI		端口更改检测。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	当任何端口发生连接状态、发生端口使能 / 禁用更改或因挂起端口上 J-K 跳变而设置强制端口恢复位时，主机控制器会将该位设为 1。		
3	FRI		帧列表滚动更新	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	当帧列表索引从其最大值滚动更新为 0 时，主机控制器会将该位设为 1。滚动更新的确切数值取决于帧列表大小。例如，如果帧列表大小（如 USBCMD 寄存器中帧列表大小字段所设定的）为 1024，则每次 FRINDEX 位 13 切换时帧索引寄存器滚动。同理，如果大小为 512，则每次 FRINDEX 位 12 切换时，主机控制器将该位设为 1 （参见第 23.6.6 节）。		
4	-		保留。		
5	AAI		异步推进时中断	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	系统软件可在下次主机控制器超前进行异步调度时，通过向 USBCMD 寄存器的异步预先提示信息位写入 1，强制主机控制器发出中断。该状态位指示确认该中断源。		
6	-	-	主机控制器未使用。	0	R/WC
7	SRI		收到 SOF	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	在主机模式下，将每 125 µs 对该位进行设置，并且该位可以作为时间基准以供主机控制驱动器使用。		
8	-	-	主机控制器未使用。	-	-
11:9	-	-	保留。		
12	HCH		HCHalted	1	RO
		0	USBCMD 中的 RS 位设为 0。由主机控制器设置。		
		1	由于运行 / 停止位设为 0，主机控制器停止执行后，将通过软件或主机控制器硬件将该位设为 1 （例如，由于内部错误）。		
13	RCL		收回	0	RO
		0	未检测到空的异步调度。		
		1	检测到空的异步调度。由主机控制器设置。		
14	PS		定期调度状态 该位报告定期调度的当前实际状态。当软件切换 USBCMD 寄存器中的定期调度使能位时，主机控制器无需立即禁用或使能定期调度。当该位和定期调度使能位具有相同值时，定期调度将使能 （如果两者都为 1）或禁用 （如果两者都为 0）。	0	RO
		0	定期调度状态禁用。		
		1	定期调度状态使能。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 393. 主机模式的 USB 状态寄存器 （USBSTS_H - 地址 0x4000 6144）寄存器位描述 *（续）*

位	符号	值	描述	复位值	访问类型
15	AS		异步调度状态 该位报告异步调度的当前实际状态。当软件切换 USBCMD 寄存器中的异步调度使能位时，主机控制器无需立即禁用或使能异步调度。当该位和异步调度使能位具有相同值时，异步调度将使能（如果两者都为 1）或禁用（如果两者都为 0）。	0	
		0	异步调度状态禁用。		
		1	异步调度状态使能。		
16	-		不适用于主机模式。	0	-
17	-		保留。		
18	UAI		USB 主机异步中断 （USBHSTASYNCINT）	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	在传输描述符 (TD) 设置了完成时中断 (IOC) 位的情况下，**且** TD 来自异步调度表，如果中断是由 USB 事务完成所致，则该位将由主机控制器进行设置。当检测到短数据包**且**该数据包位于异步调度中时，该位也将由主机设置。短数据包指收到的实际字节数少于预期字节数。		
19	UPI		USB 主机定期中断 （USBHSTPERINT）	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	在传输描述符 (TD) 设置了完成时中断 (IOC) 位的情况下，**且** TD 来自周期性调度表，如果中断是由 USB 事务完成所致，则该位将由主机控制器进行设置。当检测到短数据包**且**该数据包位于定期调度中时，该位也将由主机控制器设置。短数据包指收到的实际字节数少于预期字节数。		
31:20	-				

23.6.5 USB 中断寄存器 (USBINTR)

该寄存器可使能软件中断。当该位设置且相应中断处于工作状态时，便产生中断。 USB 状态寄存器 （USBSTS）即使通过 USBINTR 寄存器禁用，仍会显示中断源，以便允许软件轮询中断事件。所有中断必须由软件将 USBSTS 寄存器中相应的位清零 （即写入 1）来进行确认。

23.6.5.1 设备模式

表 394. 设备模式中的 USB 中断寄存器 （USBINTR_D - 地址 0x4000 6148）位描述

位	符号	描述	复位值	访问类型
0	UE	USB 中断使能 当该位为 1，且 USBSTS 寄存器中的 USBINT 位为 1 时，主机 / 设备控制器将在下一个中断阈值中发出中断。该中断由软件将 USBSTS 中的 USBINT 位清零来进行确认。	0	R/W
1	UEE	USB 错误中断使能 当该位为 1，且 USBSTS 寄存器中的 USBERRINT 位为 1 时，主机 / 设备控制器将在下一个中断阈值中发出中断。该中断由软件将 USBSTS 寄存器中的 USBERRINT 位清零来进行确认。	0	R/W
2	PCE	端口更改检测使能。 当该位为 1，且 USBSTS 寄存器中的端口更改检测位为 1 时，主机 / 设备控制器将发出中断。该中断由软件将 USBSTS 中的端口更改检测位清零来进行确认。	0	R/W
3	-	设备控制器未使用。		
4	-	保留	0	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 394. 设备模式中的 USB 中断寄存器 （USBINTR_D - 地址 0x4000 6148）位描述 *（续）*

位	符号	描述	复位值	访问类型
5	-	设备控制器未使用。		
6	URE	USB 复位使能 当该位为 1，且 USBSTS 寄存器中的收到 USB 复位位为 1 时，主机 / 设备控制器将发出中断。该中断由软件将收到 USB 复位位清零来进行确认。	0	R/W
7	SRE	收到 SOF 使能 当该位为 1，且 USBSTS 寄存器中的收到 SOF 位为 1 时，主机 / 设备控制器将发出中断。该中断由软件将收到 SOF 位清零来进行确认。	0	R/W
8	SLE	睡眠使能 当该位为 1 时，且 USBSTS 寄存器中的 DCSuspend 位切换时，设备控制器将发出中断。该中断由软件向 DCSuspend 位写入 1 来进行确认。	0	R/W
15:9	-	保留	-	-
16	NAKE	NAK 中断使能 如果要使能 NAK 中断位的硬件中断，则该位将由软件设置。如果已经设置了该位和相应的 NAK 中断，则将产生硬件中断。	0	R/W
17	-	保留		
18	-	设备控制器未使用。		
19	-	设备控制器未使用。		
31:20	-	保留		

23.6.5.2 主机模式

表 395. 主机模式中的 USB 中断寄存器 （USBINTR_H - 地址 0x4000 6148）位描述

位	符号	描述	访问类型	复位值
0	UE	USB 中断使能 当该位为 1，且 USBSTS 寄存器中的 USBINT 位为 1 时，主机 / 设备控制器将在下一个中断阈值中发出中断。该中断由软件将 USBSTS 中的 USBINT 位清零来进行确认。	R/W	0
1	UEE	USB 错误中断使能 当该位为 1，且 USBSTS 寄存器中的 USBERRINT 位为 1 时，主机 / 设备控制器将在下一个中断阈值中发出中断。该中断由软件将 USBSTS 寄存器中的 USBERRINT 位清零来进行确认。	R/W	0
2	PCE	端口更改检测使能。 当该位为 1，且 USBSTS 寄存器中的端口更改检测位为 1 时，主机 / 设备控制器将发出中断。该中断由软件将 USBSTS 中的端口更改检测位清零来进行确认。	R/W	0
3	FRE	帧列表滚动更新使能 当该位为 1，且 USBSTS 寄存器中的帧列表滚动更新位为 1 时，主机控制器将发出中断。该中断由软件将帧列表滚动更新位清零来进行确认。		
4	-	保留	-	0
5	AAE	异步推进时中断使能 当该位为 1，且 USBSTS 寄存器中的异步推进时中断位为 1 时，主机控制器将在下一个中断阈值中发出中断。该中断由软件将异步推进时中断位清零来进行确认。	R/W	0
6	-	主机控制器未使用。	-	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 395. 主机模式中的 USB 中断寄存器 （USBINTR_H - 地址 0x4000 6148）位描述 *（续）*

位	符号	描述	访问类型	复位值
7	SRE	如果该位为 1，且 USBSTS 寄存器中的 SRI 位为 1，则主机控制器将发出中断。在主机模式下， SRI 位将每 125 µs 设置一次，并可以由主机控制器用作时基。该中断由软件将 USBSTS 寄存器中的 SRI 位清零来进行确认。	-	0
8	-	主机控制器未使用。	-	0
15:9	-	保留		
16	-	主机控制器未使用。	R/W	0
17	-	保留		
18	UAIE	USB 主机异步中断使能 当该位为 1，且 USBSTS 寄存器中的 USBHSTASYNCINT 位为 1 时，主机控制器将在下一个中断阈值中发出中断。该中断由软件将USBHSTASYNCINT位清零来进行确认。	R/W	0
19	UPIA	USB 主机定期中断使能 当该位为 1，且 USBSTS 寄存器中的 USBHSTPERINT 位为 1 时，主机控制器将在下一个中断阈值中发出中断。该中断由软件将 USBHSTPERINT 位清零来进行确认。	R/W	0
31:20	-	保留		

23.6.6 帧索引寄存器 (FRINDEX)

23.6.6.1 设备模式

在设备模式下，该寄存器为只读寄存器，且设备控制器会从 SOF 标记指示的帧数更新 FRINDEX[13:3] 寄存器。只要 USB 总线收到 SOF，系统便会依据 SOF 时标检查 FRINDEX[13:3]。如果 FRINDEX[13:3] 与 SOF 时标不同，则 FRINDEX[13:3] 将设为 SOF 值，而 FRINDEX[2:0] 将设为 0（1 ms 帧的 SOF）。如果 FRINDEX [13:3] 与 SOF 值相等，则 FRINDEX[2:0] 将由硬件进行数值递增 （即 125 µs 微帧的 SOF）。

表 396. 设备模式中的 USB 帧索引寄存器 （FRINDEX_D - 地址 0x4000 614C）位描述

位	符号	描述	复位值	访问类型
2:0	FRINDEX2_0	当前微帧数	不适用	RO
13:3	FRINDEX13_3	发送的最后一帧的当前帧数	不适用	RO
31:14	-	保留	不适用	

23.6.6.2 主机模式

该寄存器由主机控制器用于对定期帧列表编制索引。该寄存器每 125 µs （每微帧一次）更新一次。位 [N:3] 用于在定期调度执行期间选择定期帧列表中的特定项。用于索引的位数取决于系统软件在 USBCMD 寄存器的 “帧列表大小”字段中设置的帧列表大小。

该寄存器必须以双字方式写入。写入字节将产生未定义的结果。除非主机控制器处于 USBSTS 寄存器中 HCHalted 位指示的 “停止”状态 （主机模式），否则无法写入该寄存器。当运行 / 停止位设为 1 时，向该寄存器写入会产生未定义的结果。对该寄存器进行写操作还会影响 SOF 值。

表 397. 主机状态的 USB 帧索引寄存器 （FRINDEX_H - 地址 0x4000 614C）位描述

位	符号	描述	复位值	访问类型
2:0	FRINDEX2_0	当前微帧数	不适用	R/W
12:3	FRINDEX12_3	帧列表当前索引。	不适用	R/W
31:13	-	保留	不适用	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 398. 用于帧列表索引的位数

USBCMD 第 15 位	USBCMD 第 3 位	USBCMD 第 2 位	帧列表大小	N
0	0	0	1024 元素 （4096 字节）。默认值。	12
0	0	1	512 元素 （2048 字节）	11
0	1	0	256 元素 （1024 字节）	10
0	1	1	128 元素 （512 字节）	9
1	0	0	64 元素 （256 字节）	8
1	0	1	32 元素 （128 字节）	7
1	1	0	16 元素 （64 字节）	6
1	1	1	8 元素 （32 字节）	5

23.6.7 设备地址 （DEVICEADDR – 设备）和周期性列表基址 （PERIODICLISTBASE – 主机）寄存器

23.6.7.1 设备模式

该寄存器的高 7 位代表设备地址。任何控制器复位或 USB 复位后，设备地址便会设为默认地址 （0）。默认地址将匹配所有传入的地址。软件收到 SET_ADDRESS 描述符后，将对地址重新编程。

USBADRA 位用于通过允许 DCD 在 SET_ADDRESS 描述符的状态阶段前预设 USBADR 寄存器位，以加速 SET_ADDRESS 序列。

表 399. 设备模式中的 USB 设备地址寄存器 （DEVICEADDR - 地址 0x4000 6154）位描述

位	符号	值	描述	复位值	访问类型
23:0	-		保留	0	-
24	USBADRA		设备地址推进		
		0	USBADR 的任何写入都为即时写入。		
		1	如果用户在写入 USBADR 的同时或之前向该位写入 1，则 USBADR 字段写入将暂存并保留在隐式寄存器中。在端点 0 发生输入并获得应答后，USBADR 将从保持寄存器载入。 在以下条件下，硬件会自动将该位清零： • 确认向端点 0 输入。 USBADR 从暂存寄存器更新。 • 端点 0 上发生 OUT/SETUP 事件， USBADR 未更新。 • 设备复位。 USBADR 设为 0。 **注：** 在 SET_ADDRESS 描述符的状态阶段之后，DCD 有 2 ms 时间来编程 USBADR 字段。该机制将确保当 DCD 在 SET_ADDRESS 状态阶段 2 ms 内无法写入设备地址时，可符合此规范。如果 DCD 在 SET_ADDRESS 数据阶段后 （状态阶段开始前）通过 USBADRA=1 写入 USBADR，则 USBADR 将在正确时间即时编程并满足 2 ms USB 要求。		
31:25	USBADR		USB 设备地址	0	R/W

23.6.7.2 主机模式

该 32 位寄存器包含系统存储器中定期帧列表的开始地址。主机控制器驱动器 （HCD）将在主机控制器执行调度前加载该寄存器。该物理存储器指针参考的存储器结构假设与 4 kB 对齐。该寄存器的内容与帧索引寄存器 （FRINDEX）组合，可使主机控制器在序列中单步调试定期帧列表。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 400. 主机模式中的 USB 周期性列表基址寄存器 （PERIODICLISTBASE - 地址 0x4000 6154）位描述

位	符号	描述	复位值	访问类型
11:0	-	保留	-	-
31:12	PERBASE31_12	基址 （低） 这些位对应存储器地址信号 31:12。	-	R/W

23.6.8 端点列表地址寄存器 （端点 LISTADDR – 设备）和异步列表地址 （ASYNCLISTADDR – 主机）寄存器

23.6.8.1 设备模式

在设备模式下，该寄存器包含系统存储器中端点列表顶端的地址。该寄存器位 [10:0] 无法由系统软件修改，且在读取时将始终返回 0。该物理存储器指针参考的存储器结构假设与 64 字节对齐。

表 401. 设备模式中的 USB 端点列表地址寄存器 （ENDPOINTLISTADDR - 地址 0x4000 6158）位描述

位	符号	描述	复位值	访问类型
10:0	-	保留	0	-
31:11	EPBASE31_11	端点列表指针 （低） 这些位分别对应内存地址信号 31:11。该字段将参考多达 4 个队列头 （QH）的列表。（即每个端点和方向一个队列头。）	-	R/W

23.6.8.2 主机模式

该 32 位寄存器包含主机将执行的下一个异步队列头的地址。该寄存器位 [4:0] 无法由系统软件修改，且在读取时将始终返回 0。

表 402. 主机模式中的 USB 异步列表地址寄存器 （ASYNCLISTADDR - 地址 0x4000 6158）位描述

位	符号	描述	复位值	访问类型
4:0	-	保留	0	-
31:5	ASYBASE31_5	链接指针 （低） LPL 这些位分别对应内存地址信号 31:5。该字段只能参考一个队列头 （OH）。	-	R/W

23.6.9 TT 控制寄存器 (TTCTRL)

23.6.9.1 设备模式

该寄存器不适用于设备模式。

23.6.9.2 主机模式

此寄存器包含内部 TT 操作所需的参数。该寄存器仅供主机控制器使用。该寄存器必须以双字方式写入。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 403. 主机模式中的 USB TT 控制寄存器 （TTCTRL - 地址 0x4000 615C）位描述

位	符号	描述	复位值	访问类型
23:0	-	保留。	0	-
30:24	TTHA	直接连接 FS 设备或 LS 设备时的集线器地址。	不适用	R/W
31	-	保留。	0	

23.6.10 突发大小寄存器 (BURSTSIZE)

该寄存器用于控制及动态更改USB DMA控制器主接口上数据移动期间所使用的突发流量。该寄存器必须以双字方式写入。

RX 和 TX DMA 数据传输的 32 位字的默认突发长度各为 16 个字。

表 404. USB 连发大小寄存器 （BURSTSIZE – 地址 0x4000 6160）位描述 - 设备 / 主机模式

位	符号	描述	复位值	访问类型
7:0	RXPBURST	可编程的 RX 突发长度 该寄存器代表将数据从 USB 总线移至系统存储器时 32 位字的最大突发长度。	0x10	R/W
15:8	TXPBURST	可编程的 TX 突发长度 该寄存器代表将数据从系统存储器移至 USB 总线时 32 位字的最大突发长度。	0x10	R/W
31:16	-	保留。	-	-

23.6.11 传输缓冲区填充调谐寄存器 (TXFILLTUNING)

23.6.11.1 设备控制器

该寄存器不适用于设备模式。

23.6.11.2 主机控制器

此寄存器中的各字段控制与主机控制器在向 USB 总线移动数据前向 TX 延迟 FIFO 发送数据的方式相关的性能调谐。性能的具体方面包括向 FIFO 中发送的数据量以及该操作在目标系统中耗时的估计值。

定义：

T_0 = 标准数据包开销

T_1 = 发送数据负载的时间

T_{ff} = 获取数据至指定水平的 TX FIFO 的时间

T_s = 数据包传输总时间 （仅发送数据包）； $T_s = T_0 + T_1$

T_p = 数据包传输总时间 （获取和发送数据包）； $T_p = T_{ff} + T_0 + T_1$

在数据结构中发现发送 (OUT/SETUP) 数据包后，主机控制器将检查以确定 $T_{p\ 在}$（微）帧结束前一直保留。如果 Tp 在（微）帧结束前，则继续预填充 TX FIFO。如果在预填充操作过程中的任何时间，[微] 帧剩余时间小于 < T_s 则，数据包将尝试停止，并在稍后再试。尽管这不是错误条件，且主机控制器将最终恢复，但调度程序运行状况计数器将做出标记，以注明发生回退事件。如果检测到回退事件，可能需要从延迟缓冲区中丢弃部分已获取的数据包，以为在下一个 SOF 后开始的定期通信腾出空间。回退事件过多可能会浪费系统总线上的宽带和功率，因此应尽量减少 （但没有必要清除）。通过下述 TSCHHEALTH (T_{ff}) 可以最大限度地减少回退事件。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 405. 主机模式中的 USB 传输缓冲区填充调谐寄存器 （TXFILLTUNING - 地址 0x4000 6164）位描述

位	符号	描述	复位值	访问类型
7:0	TXSCHOH	FIFO 突发阈值 此寄存器控制在开始向总线发送数据包之前以主机模式向 TX 延迟 FIFO 发送的数据突发数量。其最小值为 2，该值应尽可能小，以便最大限度发挥 USB 性能。在延迟不可预知和 / 或带宽不足时，系统可使用更大的值，因为从系统存储器补充数据之前，数据已从延迟 FIFO 传输至 USB，FIFO 可能不足。如果已设置了 USBMODE 寄存器中的流禁用位，则将忽略该值。	0x2	R/W
12:8	TXSCHEATLTH	调度程序运行状况计数器 如果主机控制器在发送下一个帧起始前的数据包超时前，未能将 TX 延迟 FIFO 填充至 TXFIFOTHRES 编程的电平，则该寄存器将增加。 该运行状况计数器将计量该事件发生的次数，以提供反馈来选择适当的 TXSCHOH。对该寄存器执行写操作时，该计数器将清零。最大值为 31。	0x0	R/W
15:13	-	保留	-	-
21:16	TXFIFOTHRES	调度程序开销 该寄存器会将额外的固定偏移量添加到调度时间估测器，上文描述为 T_{ff}。据粗略估算，在总线利用率高的情况下，为此寄存器选用的值应将 TXSCHHEALTH 中的回退事件数量限制到每秒低于 10 次。不建议为此寄存器选用过高的值，这样会不必要地降低 USB 利用率。 如果在 OTG 和 SPH 的高速模式下连接设备，则该寄存器中使用的时间单位为 1.267 µs。 如果在 OTG 和 SPH 的低速 / 全速模式下连接设备，则该寄存器中采用的时间单位为 6.333 µs。	0x0	R/W
31:22	-	保留		

23.6.12 BINTERVAL 寄存器

该寄存器确定 bInterval 值，该值决定虚拟帧的长度 （参见第 23.7.7 节）。 BINTERVAL 寄存器以 BINT 值分频 SOF 信号。

注：BINTERVAL 寄存器和 USB 规范中的 bInterval 端点描述符字段无关。

表 406. USB BINTERVAL 寄存器 （BINTERVAL - 地址 0x4000 6174）位描述

位	符号	描述	复位值	访问类型
3:0	BINT	bInterval 值 （参见第 23.7.7 节）	0x00	R/W
31:4	-	保留	-	-

23.6.13 USB 端点 NAK 寄存器 (ENDPTNAK)

23.6.13.1 设备模式

该寄存器指示设备何时发送端点上的 NAK 信号握手。每个 Tx 和 Rx 端点分别在 EPTN 和 EPRN 字段中有一个对应的位。

向该寄存器中的位写入 1，即可将该位清零。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 407. USB 端点 NAK 寄存器 （ENDPTNAK - 地址 0x4000 6178）位描述

位	符号	描述	复位值	访问类型
5:0	EPRN	Rx 端点 NAK 每个 RX 端点在此字段均有一个位。设备在相应端点接收到的 OUT 或 PING 标记上发送 NAK 反馈检验时，该位会被设置。 位 5 对应于端点 5。 ... 位 1 对应于端点 1。 位 0 对应于端点 0。	0x00	R/WC
15:6	-	保留	-	-
21:16	EPTN	Tx 端点 NAK 每个 TX 端点在该字段中都有一个对应的位。当设备发送相应端点收到的 IN 令牌上的 NAK 信号握手时，便会设置该位。 位 5 对应于端点 5。 ... 位 1 对应于端点 1。 位 0 对应于端点 0。	0x00	R/WC
31:22	-	保留	-	-

23.6.13.2 主机模式

该寄存器不适用于主机模式。

23.6.14 USB 端点 NAK 使能寄存器 (ENDPTNAKEN)

23.6.14.1 设备模式

该寄存器中的每个位会使能 ENDPTNAK 寄存器中相应的位。每个 Tx 和 Rx 端点分别在 EPTNE 和 EPRNE 字段中有一个对应的位。

表 408. USB 端点 NAK 使能寄存器 （ENDPTNAKEN - 地址 0x4000 617C）位描述

位	符号	描述	复位值	访问类型
5:0	EPRNE	Rx 端点 NAK 使能 每个位可使能对应的 RX NAK 位。如果设置该位和 RX 端点 NAK 位，则会设置 NAK 中断位。 位 5 对应于端点 5。 ... 位 1 对应于端点 1。 位 0 对应于端点 0。	0x00	R/W

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 408. USB 端点 NAK 使能寄存器 （ENDPTNAKEN - 地址 0x4000 617C）位描述 *（续）*

位	符号	描述	复位值	访问类型
15:6	-	保留	-	-
21:16	EPTNE	Tx 端点 NAK 每个位可使能对应的 TX NAK 位。如果设置该位和 TX 端点 NAK 位，则会设置 NAK 中断位。 位 5 对应于端点 5。 ... 位 1 对应于端点 1。 位 0 对应于端点 0。	0x00	R/W
31:22	-	保留	-	-

23.6.14.2 主机模式

该寄存器不适用于主机模式。

23.6.15 端口状态和控制寄存器 (PORTSC1)

23.6.15.1 设备模式

设备控制器提供一个端口寄存器，且不支持电源控制。设备模式下的端口控制用于状态端口复位、挂起和当前连接状态。它还可用于启动测试模式或强制发出信号。通过该寄存器，软件可以使 PHY 进入低功耗挂起模式及禁用 PHY 时钟。

表 409. 设备模式中的端口状态和控制寄存器 （PORTSC1_D - 地址 0x4000 6184）位描述

位	符号	值	描述	复位值	访问类型
0	CCS		当前连接状态	0	RO
		0	设备未连接 0 表示设备连接失败，或因软件向 USBCMD 寄存器中的运行位写入 0 而强制断开。该位不会指明设备已断开或挂起。		
		1	设备已连接。 1 表示设备成功连接并在高速模式或全速模式下运行，由此寄存器中的 “高速端口” 位指示。		
1	-	-	不适用于设备模式	0	-
2	PE		端口使能。 该位始终为 1。设备端口始终使能。	1	RO
3	PEC		端口使能 / 禁用更改 该位始终为 0。设备端口始终使能。	0	RO
5:4	-	-	保留	0	RO
6	FPR		强制端口恢复 在设备处于挂起状态 5ms 或更长时间之后，软件必须将此位设置为 1，以便在清除之前驱动恢复信号。如果端口处于挂起状态时检测到 J 至 K 跳变，则设备控制器会将该位设为 1。当设备恢复正常工作时，该位便会清零。如果此位因为检测到 J 至 K 转换而转换至 1，则 USBSTS 寄存器中的 “端口改变检测” 位也设置为 1。	0	R/W
		0	不恢复 （K 状态）端口检测 / 驱动。		
		1	恢复端口检测 / 驱动。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 409. 设备模式中的端口状态和控制寄存器 （PORTSC1_D - 地址 0x4000 6184）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
7	SUSP		挂起 在设备模式下，这是一个只读状态位。	0	RO
		0	端口未处于挂起状态		
		1	端口处于挂起状态		
8	PR		端口复位 在设备模式下，这是一个只读状态位。从 USB 总线复位的设备也会在 USBSTS 寄存器中指出。	0	RO
		0	端口未处于复位状态。		
		1	端口处于复位状态。		
9	HSP		高速状态 **注：**此位与此寄存器中的位 27:26 (PSPD) 冗余。因兼容性原因而采用。	0	RO
		0	连接到端口的主机 / 设备未处于高速模式下。		
		1	连接到端口的主机 / 设备处于高速模式下。		
11:10	-	-	不适用于设备模式。		
12	-	-	不适用于设备模式。		
13	-	-	保留	-	-
15:14	PIC1_0		端口指示器控制 向此字段执行写操作将影响 USB0_IND[1:0] 引脚的值。	00	R/W
		0x0	端口指示器关闭。		
		0x1	黄色		
		0x2	绿色		
		0x3	未定义		
19:16	PTC3_0		端口检测控制 除 0000 以外的任何值表示端口正在检测模式下运行。 FORCE_ENABLE_FS 和 FORCE ENABLE_LS 是 EHCI 规范中指定的测试模式支持的扩展功能。向 PTC 字段写入任何 FORCE_ENABLE_HS/FS/LS 值将强制端口以所选速度进入连接和使能状态。通过将 PTC 字段写回 TEST_MODE_DISABLE，端口状态机可以从该点正常运行。值 0111 至 1111 无效。	0000	R/W
		0x0	TEST_MODE_DISABLE		
		0x1	J_STATE		
		0x2	K_STATE		
		0x3	SE0 （主机） /NAK （设备）		
		0x4	数据包		
		0x5	FORCE_ENABLE_HS		
		0x6	FORCE_ENABLE_FS		
20	-	-	不适用于设备模式。此位在设备模式下始终为 0。	0	-
21	-	-	不适用于设备模式。此位在设备模式下始终为 0。	0	-
22	-		不适用于设备模式。此位在设备模式下始终为 0。	0	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 409. 设备模式中的端口状态和控制寄存器 （PORTSC1_D - 地址 0x4000 6184）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
23	PHCD		PHY 低功耗挂起 - 时钟禁用 （PLPSCD） 在设备模式下，当设备未运行 (USBCMD Run/Stop = 0) 或主机已发出挂起信号 (PORTSC SUSPEND = 1) 时， PHY 可处于低功耗挂起 - 时钟禁用状态。当主机发出恢复信号时，低功耗挂起将自动清零。在强制从设备恢复前，设备控制器驱动器必须将该位清零。	0	R/W
		0	如果写入 0，则将使能 PHY 时钟。如果读取 0，则将指示 PHY 时钟的状态（使能）。		
		1	如果写入 1，则将禁用 PHY 时钟。如果读取 1，则将指示 PHY 时钟的状态（禁用）。		
24	PFSC		端口强制全速连接	0	R/W
		0	端口以任何速度连接。		
		1	如果向该位写入 1，则将强制端口仅以全速连接。它将禁用脉冲组序列，后者可使端口将其本身标识为高速。该功能对于使用 HS 主机、集线器或设备测试 FS 配置非常有效。		
25	-	-	保留		
27:26	PSPD		端口速度 该寄存器字段指示端口运行的速度。	0	RO
		0x0	全速		
		0x1	在设备模式中无效		
		0x2	高速		
31:28	-	-	保留	-	-

23.6.15.2 主机模式

主机控制器使用一个端口。只有在初次上电或对控制器复位作出响应时，寄存器才会复位。端口的初始状况为：

- 设备未连接
- 端口禁用

如果端口具有电源控制功能，则该状态将保持，直到软件通过在 PORTSC 寄存器中将端口电源设为 1 对端口上电为止。

表 410. 主机模式中的端口状态和控制寄存器 （PORTSC1_H - 地址 0x4000 6184）位描述

位	符号	值	描述	复位值	访问类型
0	CCS		当前连接状态 该值反映端口的当前状态，且可能不会与导致 CSC 位设置的事件直接对应。 如果 PP （端口电源位）为 0，则该位为 0。 软件向该位写入 1，即可将该位清零。	0	R/WC
		0	不存在设备。		
		1	端口上存在设备。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 410. 主机模式中的端口状态和控制寄存器 （PORTSC1_H - 地址 0x4000 6184）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
1	CSC		连接状态更改 指示端口的当前连接状态已发生变化。即使系统软件未将现有连接状态更改清零，主机 / 设备控制器也会针对端口设备连接状态的所有更改设置该位。例如，系统软件清除变更情况前插入状态变化两次，集线器硬件将设置已设置好的位 （即该位保持设置状态）。软件向该位写入 1，即可将该位清零。 如果 PP （端口电源位）为 0，则该位为 0	0	R/WC
		0	当前状态未更改。		
		1	当前状态更改。		
2	PE		端口使能。 作为复位和使能的一部分，端口只能通过主机控制器使能。软件无法通过向该字段写入 1 来使能端口。端口可以因故障条件 （断开事件或其他故障条件）或由主机软件来禁用。需要注意的是，在端口状态实际更改前，位状态不会更改。由于其他主机控制器和总线事件的关系，禁用或使能端口时可能存在延迟。 如果禁用端口，除非进行复位，否则数据的下行传播将被阻断。 如果 PP （端口电源位）为 0，则该位为 0。	0	R/W
		0	端口禁用。		
		1	端口使能。		
3	PEC		端口禁用 / 使能更改 对于根集线器，此位仅在端口由于断开或由于在 EOF2 点存在适当条件而被禁用时设置为 1（参见 *USB 规范的第 11 章*）。软件通过在此写入 1 来清除。 如果 PP （端口电源位）为 0，则该位为 0，	0	R/WC
		0	无变化。		
		1	端口使能 / 禁用状态已更改。		
4	OCA		过电流激活 过电流条件被去除后，此位自动从 1 转换至 0。	0	RO
		0	端口没有过流条件。		
		1	端口当前具有过流条件。		
5	OCC		过流更改 当过流有效更改时，该位将设为 1。软件向该位位置写入 1，即可将该位清零。	0	R/WC
6	FPR		强制端口恢复 软件将该位设为 1，即可驱动恢复信号。如果端口处于挂起状态时检测到 J 至 K 跳变，则主机控制器会将该位设为 1。如果由于检测到 J 至 K 跳变，该位跳变为 1，则 USBSTS 寄存器中的端口更改检测位也会设为 1。在恢复序列完成后，该位将自动更改为 0。该行为与 EHCI 不同，后者是在驱动器中对恢复持续时间计时后，需要主机控制器将该位设为 0。 注意，如果主机控制器拥有端口，则恢复序列遵守 USB 规范修订版 2.0 中定义的序列。只要该位保持为 1，该恢复信号发送 （全速 K）在端口上驱动。在端口切换至高速空闲之前，该位将保持为 1。由于端口控制器将对恢复操作计时，因此写入 0 并无影响，将该位清零后，端口控制状态将切换至 HS 或 FS 空闲状态。 如果 PP （端口电源位）为 0，则该位为 0。	0	R/W
		0	不恢复 （K 状态）端口检测 / 驱动。		
		1	恢复端口检测 / 驱动。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 410. 主机模式中的端口状态和控制寄存器 （PORTSC1_H - 地址 0x4000 6184）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
7	SUSP		挂起 该位与 PE （端口使能位）一样，都是描述端口状态，参见表 411。 当软件将强制端口恢复位设为 0 时，主机控制器将无条件将该位设为 0。主机控制器将忽略向该位写入 0。 在端口未使能 （即端口端口位为 0）时，如果主机软件将该位设为 1，则将产生未定义的结果。 如果 PP （端口电源位）为 0，则该位为 0。	0	R/W
		0	端口未处于挂起状态		
		1	端口处于挂起状态 当处于挂起状态时，除非对端口进行复位，否则端口上数据的下行传输将被阻断。如果某个事务正在进行时向该位写 1，则在当前事务结束时会发生阻断。当处于挂起状态时，端口对恢复检测非常敏感。需要注意的是，在端口处于挂起状态前，位状态不会更改；而如果 USB 上当前正在处理事务，则端口挂起可能出现延迟。		
8	PR		端口复位 当软件向该位写入 1 时，便会使能 USB 规范 2.0 版中定义的总线复位序列。在复位序列完成后，该位将自动更改为 0。该行为与 EHCI 不同，后者是在驱动器中对复位持续时间计时后，需要主机控制器将该位设为 0。 如果 PP （端口电源位）为 0，则该位为 0。	0	R/W
		0	端口未处于复位状态。		
		1	端口处于复位状态。		
9	HSP		高速状态	0	RO
		0	连接到端口的主机 / 设备未处于高速模式下。		
		1	连接到端口的主机 / 设备处于高速模式下。		
11:10	LS		线路状态 这些位反映 USB_DP 和 USB_DM 信号线的当前逻辑电平。USB_DP 对应于位 11，而 USB_DM 对应于位 10。 在主机模式下，由于控制器硬件管理 LS 和 FS 的连接，主机控制器驱动器无需将线路状态用于该控制器 （与 EHCI 不同）。	0x3	RO
		0x0	SE0 （USB_DP 和 USB_DM 低电平）		
		0x1	J 状态 （USB_DP 高电平和 USB_DM 低电平）		
		0x2	K 状态 （USB_DP 低电平和 USB_DM 高电平）		
		0x3	未定义		
12	PP	-	端口电源控制 主机 /OTG 控制器需要端口电源控制开关。该位表示开关的当前设置 （0= 关， 1= 开）。当端口上电源不可用 （即 PP=0）时，端口将不能正常运行且不会报告连接、分离等状况。 如果在上电端口检测到过流条件，且 PPC 为 1，每个受影响端口中的 PP 位可能由主机控制器驱动器从 1 跳变为 0 （移除端口的电源）。	0	R/W
		0	端口电源关闭。		
		1	端口上电。		
13	-	-	保留	0	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 410. 主机模式中的端口状态和控制寄存器 （PORTSC1_H - 地址 0x4000 6184）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
15:14	PIC1_0		端口指示器控制 向此字段执行写操作将影响引脚 USB0_IND1 和 USB0_IND0 的值。	00	R/W
		0x0	端口指示器关闭。		
		0x1	黄色		
		0x2	绿色		
		0x3	未定义		
19:16	PTC3_0		端口检测控制 除 0000 以外的任何值表示端口正在检测模式下运行。 FORCE_ENABLE_FS 和 FORCE ENABLE_LS 是 EHCI 规范中指定的测试模式支持的扩展功能。向 PTC 写入任何 FORCE_ENABLE_{HS/FS/LS} 值将强制端口以所选速度进入连接和使能状态。通过将 PTC 字段写回 TEST_MODE_DISABLE，端口状态机可以从该点正常运行。值 0x8 至 0xF 保留。	0000	R/W
		0x0	TEST_MODE_DISABLE		
		0x1	J_STATE		
		0x2	K_STATE		
		0x3	SE0 （主机） /NAK （设备）		
		0x4	数据包		
		0x5	FORCE_ENABLE_HS		
		0x6	FORCE_ENABLE_FS		
		0x7	FORCE_ENABLE_LS		
20	WKCN		连接使能时唤醒 (WKCNNT_E) 如果 PP （端口电源位）为 0，则该位为 0	0	R/W
		0	禁止端口在设备连接时唤醒。		
		1	如果向该位写入 1，则端口将对作为唤醒事件的设备连接非常敏感。		
21	WKDC		断开时唤醒使能 （WKDSCNNT_E） 如果 PP （端口电源位）为 0，则该位为 0。	0	R/W
		0	禁止端口在设备断开时唤醒。		
		1	如果向该位写入 1，则端口将对作为唤醒事件的设备断开非常敏感。		
22	WKOC		过流时唤醒使能 （WKOC_E）	0	R/W
		0	禁止端口在过流事件时唤醒。		
		1	如果向该位写入 1，则端口将对作为唤醒事件的过流条件非常敏感。		
23	PHCD		PHY 低功耗挂起 - 时钟禁用 （PLPSCD） 在主机模式下，当下行设备处于挂起模式下或未连接下行设备时，PHY 可处于低功耗挂起 - 时钟禁用状态。低功耗挂起可完全由软件控制。	0	R/W
		0	如果写入 0，则将使能 PHY 时钟。如果读取 0，则将指示 PHY 时钟的状态（使能）。		
		1	如果写入 1，则将禁用 PHY 时钟。如果读取 1，则将指示 PHY 时钟的状态（禁用）。		

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 410. 主机模式中的端口状态和控制寄存器 （PORTSC1_H - 地址 0x4000 6184）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
24	PFSC		端口强制全速连接	0	R/W
		0	端口以任何速度连接。		
		1	如果向该位写入 1，则将强制端口仅以全速连接。它将禁用脉冲组序列，后者可让端口将其本身标识为高速。该功能对于使用 HS 主机、集线器或设备测试 FS 配置非常有效。		
25	-	-	保留		
27:26	PSPD		端口速度 该寄存器字段指示端口运行的速度。如果在主机控制器中的 HS 模式下工作及在设备控制器中的HS/FS模式下工作，则端口线路可控制传输至协议引擎的数据。如果在主机控制器中的 FS 和 LS 模式下工作，则端口线路可通过嵌入式事务转译器来控制传输至协议引擎的数据。	0	RO
		0x0	全速		
		0x1	低速		
		0x2	高速		
31:28	-		保留	-	-

表 411. PORTSC1 寄存器中 PE 和 SUSP 位指定的端口状态

PE 位	SUSP 位	端口状态
0	0 或 1	禁用。
1	0	使能
1	1	挂起

23.6.16 OTG 状态和控制寄存器 (OTGSC)

OTG 寄存器分为四个部分：

- OTG 中断使能 （R/W）
- OTG 中断使能 （R/WC）
- OTG 状态输入 （RO）
- OTG 控制 （R/W）

状态输入使用 1 毫秒时间常数防反跳。状态输入中的值保持时间不超过 1 毫秒时，将不会导致状态输入寄存器更新或导致 OTG 中断。

表 412. OTG 状态和控制寄存器 （OTGSC – 地址 0x4000 61A4）位描述

位	符号	值	描述	复位值	访问类型
0	VD		VBUS 放电 如果将该位设为 1，则将导致 VBUS 通过电阻放电。	0	R/W
1	VC		VBUS_ 充电 如果将该位设为 1，则将导致 VBUS 线路充电。该功能用于在 SRP 期间的 VBUS 脉冲调制。	0	R/W
2	HAAR		硬件辅助自动复位	0	R/W
		0	已禁用		
		1	在主机端口上连接后，使能自动复位。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 412. OTG 状态和控制寄存器 （OTGSC – 地址 0x4000 61A4）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
3	OT		OTG 终止 如果 OTG 控制器处于设备模式，则该位必须设为 1。该位控制 USB_DM 上的下拉。	0	R/W
4	DP		数据脉冲调制 如果将该位设为 1，则将导致 USB_DP 上拉以用于 SRP 期间的数据脉冲调制。	0	R/W
5	IDPU		ID 上拉。 该位可控制上拉电阻。	1	R/W
		0	上拉关闭。 ID 位不会采样。		
		1	上拉开启。		
6	HADP		硬件辅助数据脉冲 写入 1 可启动数据脉冲序列。	0	R/W
7	HABA		硬件辅助 B 断开至 A 连接	0	R/W
		0	禁能。		
		1	启动自动 B 断开至 A 连接序列。		
8	ID		USB ID	0	RO
		0	A 设备		
		1	B 设备		
9	AVV		A-VBUS 有效 读取 1 表示 VBUS 在 A-VBUS 的有效阈值以上。	0	RO
10	ASV		A 会话有效 读取 1 表示 VBUS 在 A 会话的有效阈值以上。	0	RO
11	BSV		B 会话有效 读取 1 表示 VBUS 在 B 会话的有效阈值以上。	0	RO
12	BSE		B 会话结束 读取 1 表示 VBUS 在 B 会话的结束阈值以下。	0	RO
13	MS1T		1 毫秒定时器切换 该位每毫秒切换一次。	0	RO
14	DPS		数据总线脉冲状态 读取 1 表示端口上检测到数据总线脉冲。	0	RO
15	-	-	保留	0	
16	IDIS		USB ID 中断状态 当检测到 ID 输入上发生变化时，便会设置该位、。 软件必须向该位写入 1，才能将它清零。	0	R/WC
17	AVVIS		A-VBUS 有效中断状态 设置此位，则 VBUS 升高（或下降）到 A-VBUS 有效阈值（A 设备为 4.4 V）以上 （或以下）。 软件必须向该位写入 1，才能将它清零。	0	R/WC
18	ASVIS		A 会话有效中断状态 当 VBUS 超过或低于 A 会话有效阈值 (0.8 V) 时，便会设置该位。 软件必须向该位写入 1，才能将它清零。	0	R/WC

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 412. OTG 状态和控制寄存器 （OTGSC – 地址 0x4000 61A4）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
19	BSVIS		B 会话有效中断状态 当 VBUS 超过或低于 B 会话有效阈值 (0.8 V) 时，便会设置该位。 软件必须向该位写入 1，才能将它清零。	0	R/WC
20	BSEIS		B 会话结束中断状态 当 VBUS 低于 B 会话结束阈值时，便会设置该位。 软件必须向该位写入 1，才能将它清零。	0	R/WC
21	ms1S		1 毫秒定时器中断状态 该位每毫秒设置一次。 软件必须向该位写入 1，才能将它清零。	0	R/WC
22	DPIS		数据脉冲中断状态 当DP或DM上发生数据总线脉冲调制时，便会设置该位。只有当USBMODE 中的 CM 位 = 主机 （11）且 PORTSC 中的端口电源位 = 关 （0）时，才会检测到数据总线脉冲。 软件必须向该位写入 1，才能将它清零。	0	R/WC
23	-	-	保留	0	
24	IDIE		USB ID 中断使能 如果设置该位，则将使能中断。如果写入 0，则将禁用中断。	0	R/W
25	AVVIE		A-VBUS 有效中断使能 如果设置该位，则将使能 A-VBUS 有效中断。如果写入 0，则将禁用中断。	0	R/W
26	ASVIE		A 会话有效中断使能 如果设置该位，则将使能 A 会话有效中断。如果写入 0，则将禁用中断	0	R/W
27	BSVIE		B 会话有效中断使能 如果设置该位，则将使能 B 会话有效中断。如果写入 0，则将禁用中断。	0	R/W
28	BSEIE		B 会话结束中断使能 如果设置该位，则将使能 B 会话结束中断。如果写入 0，则将禁用中断。	0	R/W
29	MS1E		1 毫秒定时器中断使能 如果设置该位，则将使能 1 毫秒定时器中断。如果写入 0，则将禁用中断。	0	R/W
30	DPIE		数据脉冲中断使能 如果设置该位，则将使能数据脉冲中断。如果写入 0，则将禁用中断	0	R/W
31	-	-	保留	0	-

23.6.17 USB 模式寄存器 (USBMODE)

USBMODE 寄存器可设置 OTG 控制器的 USB 模式。OTG 工作的可能模式包括设备、主机和空闲模式。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

23.6.17.1 设备模式

表 413. 设备模式中的 USB 模式寄存器 （USBMODE_D – 地址 0x4000 61A8）位描述

位	符号	值	描述	复位值	访问类型
1:0	CM1_0		控制器模式 控制器默认设为空闲状态，且在复位后需要初始化才能进入所需的工作模式。该寄存器在复位后只能写入一次。如果需要切换模式，则软件必须在重新编程该寄存器前，对 USBCMD 寄存器中的复位位执行写入操作才能复位控制器。	00	R/WO
		0x0	空闲		
		0x1	保留		
		0x2	设备控制器		
		0x3	主机控制器		
2	ES		字节序选择 该位可更改传输缓冲区的字节顺序，以匹配主机微处理器总线架构。微处理器接口和 DMA 数据结构 （包括设备 QH 中的设备缓冲区）中的位字段是以 32 位字为基础，因此不会受该位值的影响。	0	R/W
		0	小端：32 位字的最低有效位参考第一个字节。		
		1	大端：32 位字的最高有效位参考第一个字节。		
3	SLOM		设置锁定模式 在设备模式中，此位控制设置锁定机制的行为。参见第 23.10.8 节。	0	R/W
		0	设置锁定开		
		1	设置锁定关 （DCD 需要在 USBCMD 中使用设置缓冲区绊网）		
4	SDIS		流禁用模式 **注：**使用此功能会严重限制可达到的 USB 整体性能。	0	R/W
		0	未禁用		
		1	禁能。 如果将该位设为 1，将禁用低带宽系统中 RX 和 TX 的双启动。该模式可确保当 RX 和 TX 缓冲区足以容纳整个数据包时，禁用标准双缓冲机制，以防止带宽有限系统中出现溢出 / 不足。注：处于高速模式时，如果流禁用有效，则将通过 NYET 信号握手对收到的所有数据包作出响应。		
5	-		不适用于设备模式。	0	-
31:6	-	-	保留		

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

23.6.17.2 主机模式

表 414. 主机模式中的 USB 模式寄存器 （USBMODE_H – 地址 0x4000 61A8）位描述

位	符号	值	描述	复位值	访问类型
1:0	CM		控制器模式 控制器默认设为空闲状态，且在复位后需要初始化才能进入所需的工作模式。该寄存器在复位后只能写入一次。如果需要切换模式，则软件必须在重新编程该寄存器前，对 USBCMD 寄存器中的复位位执行写入操作才能复位控制器。	00	R/WO
		0x0	空闲		
		0x1	保留		
		0x2	设备控制器		
		0x3	主机控制器		
2	ES		字节序选择 该位可更改传输缓冲区的字节顺序。微处理器接口和 DMA 数据结构 （包括设备 QH 中的设备缓冲区）中的位字段是以 32 位字为基础，因此不会受该位值的影响。	0	R/W
		0	小端：32 位字的最低有效位参考第一个字节。		
		1	大端：32 位字的最高有效位参考第一个字节。		
3	-		不适用于主机模式	0	-
4	SDIS		流禁用模式 **注：**使用此功能会严重限制可达到的 USB 整体性能。	0	R/W
		0	未禁用		
		1	禁能。 对于 RX 和 TX 缓冲区足以包含整个数据包的低带宽系统，设置为 1 可确保消除延期 FIFO 的溢出 / 欠载。在 USB 上启动数据包前，使能流禁用也可确保 TX 延迟填充到容量。 注：当流禁用有效时，预填充 FIFO 的持续时间非常重要。请参见 TXFILLTUNING，以确定使用该功能时调度器所需的调整。		
5	VBPS		VBUS 电源选择	0	R/WO
		0	vbus_pwr_select 设为低电平。		
		1	vbus_pwr_select 设为高电平		
31:6	-	-	保留	-	-

23.6.18 USB 端点设置状态寄存器 (ENDPSETUPSTAT)

表 415. USB 端点设置状态寄存器 （ENDPTSETUPSTAT - 地址 0x4000 61AC）位描述

位	符号	描述	复位值	访问类型
5:0	ENDPTSETUPSTAT	逻辑端点 0 至 5 的设置端点状态。 对于收到的每个设置事务，该寄存器中相应的位将设为 1。软件必须清除或确认设置传输，方法是在从队列头读取设置数据后向相应位写入 1。在设置锁定机制使能时，按操作顺序和总响应时间对设置数据包做出的响应对于限制总线时间超时非常重要。	0	R/WC
31:6	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

23.6.19 USB 端点启动寄存器 (ENDPTPRIME)

对于每个端点，只要向端点传递新的传输描述符，软件即应向相应位写入 1。硬件将自动使用该位开始分析来自队列头的新传输描述符，并准备接收缓冲区。当相关端点成功启动时，硬件便会将该位清零。

注：在 dTD 退出和 dQH 更新时，硬件端点重新启动操作期间，由硬件立即对这些位进行设置。

表 416. USB 端点启动寄存器 （ENDPTPRIME - 地址 0x4000 61B0）位描述

位	符号	描述	复位值	访问类型
5:0	PERB	物理 OUT 端点 5 至 0 的启动端点接收缓冲区。 对于每个 OUT 端点，软件都会将相应的位设置为 1，在 USB 主机启动 USB OUT 事务时，请求为接收操作准备缓冲区。只要向端点传递新的传输描述符，软件即应向相应位写入 1。硬件将自动使用该位开始分析来自队列头的新传输描述符，并准备接收缓冲区。当相关端点成功启动时，硬件便会将该位清零。 PERB0 = 端点 0 ... PERB5 = 端点 5	0	R/WS
15:6	-	保留	-	-
21:16	PETB	物理 IN 端点 5 至 0 的启动端点发送缓冲区。 对于每个 IN 端点，相应位将由软件设为 1，请求为发送操作准备缓冲区，以便对 USB IN/INTERRUPT 事务做出响应。当向端点传递新的传输描述符，软件即应向相应位写入 1。硬件将自动使用该位来开始分析来自队列头的新传输描述符，并准备发送缓冲区。当相关端点成功启动时，硬件便会将该位清零。 PETB0 = 端点 0 ... PETB5 = 端点 5	0	R/WS
31:22	-	保留	-	-

23.6.20 USB 端点刷新寄存器 (ENDPTFLUSH)

如果向该寄存器中的位写入 1，将导致相关端点将任何已启动的缓冲区清零。如果数据包正在其中一个相关端点上处理，则该传输将继续执行，直到完成为止。在端点刷新操作成功后，硬件便会将该寄存器清零。

表 417. USB 端点刷新寄存器 （ENDPTFLUSH – 地址 0x4000 61B4）位描述

位	符号	描述	复位值	访问类型
5:0	FERB	物理 OUT 端点 5 至 0 的刷新端点接收缓冲区。 如果向相关位写入 1，将使任何已启动的缓冲区清零。 FERB0 = 端点 0 ... FERB5 = 端点 5	0	R/WS

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 417. USB 端点刷新寄存器 （ENDPTFLUSH – 地址 0x4000 61B4）位描述 *（续）*

位	符号	描述	复位值	访问类型
15:6	-	保留	-	-
21:16	FETB	物理 IN 端点 5 至 0 的刷新端点发送缓冲区。 如果向相关位写入 1，将使任何已启动的缓冲区清零。 FETB0 = 端点 0 ... FETB5 = 端点 5	0	R/WS
31:22	-	保留	-	-

23.6.21 USB 端点状态寄存器 (ENDPTSTAT)

每个端点的一个位指示相应端点缓冲区的状态。该位由硬件设置，作为对收到 ENDPTPRIME 寄存器中相应位发出的命令的响应。设置 ENDPTPRIME 寄存器中的位与端点指示就绪之间将始终存在延迟。该延迟时间取决于当前的 USB 流量和 ENDPTPRIME 寄存器中所设置的位数。缓冲区就绪将通过 USB 复位、USB DMA 系统或 ENDPTFLUSH 寄存器清零。

注：在 dTD 退出和 dQH 更新时，硬件端点重新启动操作期间，由硬件立即对这些位进行清除。

表 418. USB 端点状态寄存器 （ENDPTSTAT – 地址 0x4000 61B8）位描述

位	符号	描述	复位值	访问类型
5:0	ERBR	物理 OUT 端点 5 至 0 的端点接收缓冲区就绪。 该位由硬件设为 1，作为对收到 ENDPTPRIME 寄存器中相应位发出的命令的响应。 ERBR0 = 端点 0 ... ERBR5 = 端点 5	0	RO
15:6	-	保留	-	-
21:16	ETBR	物理 IN 端点 3 至 0 的端点发送缓冲区就绪。 该位由硬件设为 1，作为对收到 ENDPTPRIME 寄存器中相应位发出的命令的响应。 ETBR0 = 端点 0 ... ETBR5 = 端点 5	0	RO
31:22	-	保留	-	-

23.6.22 USB 端点完成寄存器 (ENDPTCOMPLETE)

该寄存器中的每个位指示是否发生收到 / 发送事件，且软件应读取相应的端点队列以确定传输状态。如果在传输描述符中设置相应的 IOC 位，则该位将与 USBINT 同时设置。

如果写入 1，便会将该寄存器中相应的位清零。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 419. USB 端点完成寄存器 （ENDPTCOMPLETE – 地址 0x4000 61BC）位描述

位	符号	描述	复位值	访问类型
5:0	ERCE	物理 OUT 端点 5 至 0 的端点接收完成事件。 如果发生接收事件 （OUT/SETUP），则该位将由硬件设为 1。 ERCE0 = 端点 0 ... ERCE5 = 端点 5	0	R/WC
15:6	-	保留	-	-
21:16	ETCE	物理 IN 端点 5 至 0 的端点发送完成事件。 如果发生发送事件 （IN/INTERRUPT），则该位将由硬件设为 1。 ETCE0 = 端点 0 ... ETCE5 = 端点 5	0	R/WC
31:22	-	保留	-	-

23.6.23 USB 端点 0 控制寄存器 (ENDPTCTRL0)

该寄存器可初始化用于控制传输的端点 0。端点 0 始终为控制端点。

表 420. USB 端点 0 控制寄存器 （ENDPTCTRL0 - 地址 0x4000 61C0）位描述

位	符号	值	描述	复位值	访问类型
0	RXS		Rx 端点停止	0	R/W
		0	端点正常。		
		1	端点停止 软件可以向该位写入 1，以强制端点向主机返回 STALL 信号握手。端点将继续返回 STALL，直到该位由软件清零，或在收到新的设置请求后自动清零为止。 收到 SETUP 请求后，此位将继续由硬件清除，直到相关的 ENDSETUPSTAT 位清除为止。[1]		
1	-	-	保留		
3:2	RXT1_0		端点类型 端点 0 始终为控制端点。	00	R/W
6:4	-	-	保留	-	-
7	RXE		Rx 端点使能 端点使能。控制端点 0 始终处于使能状态。该位始终为 1。	1	RO
15:8	-	-	保留	-	-
16	TXS		Tx 端点停止		R/W
		0	端点正常。		
		1	端点停止 软件可以向该位写入 1，以强制端点向主机返回 STALL 信号握手。端点将继续返回 STALL，直到该位由软件清零，或在收到新的设置请求后自动清零为止。 收到 SETUP 请求后，此位将继续由硬件清除，直到相关的 ENDSETUPSTAT 位清除为止。[1]		
17	-	-	保留		
19:18	TXT1_0		端点类型 端点 0 始终为控制端点。	00	RO

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 420. USB 端点 0 控制寄存器 （ENDPTCTRL0 - 地址 0x4000 61C0）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
22:20	-	-	保留		
23	TXE		Tx 端点使能 端点使能。控制端点 0 始终处于使能状态。该位始终为 1。	1	RO
31:24	-	-	保留		

[1] ENPTSETUPSTAT 清零与硬件继续将该位清零之间会稍有延迟（最多 50 个时钟）。在大多数系统中，DCD 软件不大可能会注意到该延迟。但是，如果 DCD 注意到向 STALL 位写入 1 后未设置该位，软件将继续写入该 STALL 位，直到该位设置为止，或直到通过检查相关的 ENDPTSETUPSTAT 位已收到新设置为止。

23.6.24 端点 1 到 5 控制寄存器

每个并非控制端点的端点都有其自身的寄存器，以便设置端点类型及使能或禁用端点。

注：所有端点类型的复位值均为控制端点。如果一个端点方向使能，而相反方向的配对端点禁用，则未使用方向的端点类型必须由控制类型更改为任何其他类型（如批量）。退出未配置的端点控制将导致活动端点上数据 PID 追踪出现未定义的行为。

表 421. USB端点1到5控制寄存器（ENDPTCTRL – 地址0x4000 61C4 (ENDPTCTRL1)到0x4000 61D4 (ENDPTCTRL5)）位描述

位	符号	值	描述	复位值	访问类型
0	RXS		Rx 端点停止	0	R/W
		0	端点正常。 如果该端口配置为控制端点，则该位在收到设置请求后将自动清零；且在相关的 ENDPTSETUPSTAT 位清零之前，该位将继续由硬件清零。		
		1	端点停止 软件可以向该位写入 1，以强制端点向主机返回 STALL 信号握手。端点将继续返回 STALL，直到该位由软件清零，或在收到新的设置请求后自动清零为止。		
1	-		保留	0	R/W
3:2	RXT		端点类型	00	R/W
		0x0	控制		
		0x1	同步		
		0x2	批量		
		0x3	中断		
4	-	-	保留		
5	RXI		Rx 数据切换禁止 该位仅用于测试，且应始终写入 0。如果向该位写入 1，将导致该端点忽略数据切换序列并始终接受数据包，而无论其数据 PID 如何。	0	R/W
		0	已禁用		
		1	使能		
6	RXR		Rx 数据切换复位 写入 1 可复位 PID 序列。 只要收到该端点的配置事件，软件便必须向该位写入 1，以便同步主机与设备之间的数据 PID。	0	WS

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 421. USB端点1到5控制寄存器（ENDPTCTRL – 地址0x4000 61C4 (ENDPTCTRL1)到0x4000 61D4 (ENDPTCTRL5)）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
7	RXE		Rx 端点使能 **注：**端点应仅在配置后使能。	0	R/W
		0	端点禁用。		
		1	端点使能。		
15:8	-	-	保留		
16	TXS		Tx 端点停止	0	R/W
		0	端点正常。 如果该端口配置为控制端点，则该位在收到设置请求后将自动清零；且在相关的 ENDPTSETUPSTAT 位清零之前，该位将继续由硬件清零。		
		1	端点停止 软件可以向该位写入 1，以强制端点向主机返回 STALL 信号握手。端点将继续返回 STALL，直到该位由软件清零，或在收到新的设置请求后自动清零为止。		
17	-	-	保留	0	-
19:18	TXT1_0		Tx 端点类型	00	R/W
		0x0	控制		
		0x1	同步		
		0x2	批量		
		0x3	中断		
20	-	-	保留		
21	TXI		Tx 数据切换禁止 该位仅用于测试，且应始终写入 0。如果向该位写入 1，将导致该端点忽略数据切换序列并始终接受数据包，而无论其数据 PID 如何。	0	R/W
		0	使能		
		1	已禁用		
22	TXR		Tx 数据切换复位 写入 1 可复位 PID 序列。 只要收到该端点的配置事件，软件便必须向该位写入 1，以便同步主机与设备之间的数据 PID。	1	WS
23	TXE		Tx 端点使能 **注：**端点应仅在配置后使能	0	R/W
		0	端点禁用。		
		1	端点使能。		
31:24	-	-	保留	0	

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

23.7 功能说明

23.7.1 OTG 核

OTG 核构成了 USB-OTG 的主要数字部分。有关该内核的详情，请参见 *USB EHCI 规范*。

23.7.2 主机数据结构

请参见*通用串行总线 1.0 的 USB EHCI 规范*的第 4 章。

23.7.3 主机操作模式

请参见*通用串行总线 1.0 的 USB EHCI 规范*的第 3 章。

23.7.4 ATX_RGEN 模块

关于 ATX 收发器的复位信号的要求如下：

- 时钟必须运行才能正确复位。
- ATX 必须看到复位的上升沿，以便正确复位时钟产生模块。
- 复位持续时间必须至少 133 ns （时钟周期 4 × 30 MHz）才能正确复位所有逻辑。

无论 AHB 复位如何，ATX_RGEN 模块都会产生关于 ATX 的复位信号且满足上述 3 个要求。

23.7.5 ATX 收发器

USB-OTG 具有 USB 收发器和 UTMI+ 接口。它具有所需的收发器 OTG 功能，其中包括：

- VBUS 感测以产生会话有效及 VBUS 有效信号。
- USB_ID 输入采样以检测 A 设备或 B 设备连接。
- VBUS 充电和放电以开始及结束 B 设备的会话。

23.7.6 操作模式

通常，可在主机模式或设备模式下操作 USB-OTG。软件必须通过设置 USBMODE.CM 字段 （11 用于主机模式， 10 用于设备模式）将核置于适当模式下。

USBMODE.CM 字段也可等于 00，这表示核处于闲置模式（既非主机模式也非设备模式）。这发生在以下事件后：

- 硬件复位。
- 通过USBCMD.RST位进行软件复位。当从主机模式切换至设备模式作为HNP协议的一部分 （反之亦然）时，软件必须发出软件复位，将核设为闲置状态，而发生的时间将由软件决定。

23.7.7 SOF/VF 指示器

有关将 SOF 信号连接至 LPC43xx 上的其他外围设备的信息，请参见第 23.2 节。

USB-OTG 会产生 SOF/VF 指示器信号，可供特定外部逻辑使用。

在 FS 模式下， SOF/VF 指示器信号的频率等于帧频率，约 1 kHz。该信号在半个帧周期为高电平，而在另半个帧周期为低电平。正沿与帧起始 （= SOF）对齐。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

在 HS 模式下，SOF/VF 指示器信号的频率等于虚拟帧频率。正沿表示虚拟帧的起始，并与 USB 上发生的 SOF 标记对齐

虚拟帧的长度定义如下：VF = 微帧 × $2^{bInterval}$

bInterval 在 4 位可编程的 BINTERVAL.BINT 寄存器字段中指定。 bInterval 的最小值为 0，而最大值为 15。

在挂起模式下， SOF/VF 指示器信号关闭 （即仍为低电平）。

23.7.8 硬件辅助

如果存在重大中断延迟响应时间，则硬件辅助可提供自动响应和序列，而这些功能在软件中无法实现。该额外电路可选择使用，并可以通过设置 OTGSC 寄存器中的相应位，用于辅助以下三种状态跳变：

- 自动复位 （设置位 HAAR）。
- 数据脉冲 （设置位 HADP）。
- B 断开至 A 连接 （设置位 HABA）。

23.7.8.1 自动复位

当 OTGSC 寄存器中的 HAAR 设为 1 时，主机将在连接事件后自动启动复位。这样会缩短一般流程，一般流程是软件收到连接事件的通知，然后开始复位。软件仍将收到连接事件（PORTSC 寄存器中的 CCS 位）的通知，但在设置 HAAR 时不会在 USBCMD 寄存器中写入复位位。通过导致端点更改中断的 PORTSC 寄存器中的使能更改位完成复位后，软件将再次收到通知。

此辅助功能可确保满足 OTG 参数 TB_ACON_BSE0_MAX = 1 ms（有关 OTG 计时要求的说明，请参见 *OTG 规范*）。

23.7.8.2 数据脉冲

如果向 OTGSC 寄存器中的 HADP 写入 1，则将启动持续时间约为 7 ms 的数据脉冲，然后自动停止数据脉冲。在数据脉冲期间，将设置 DP 位，然后再将其清零。该自动流程使得软件无需对数据脉冲的持续时间进行精确控制。在数据脉冲期间，HCD通过轮询可确定HADP 和 DP 位是否返回低电平来识别完成，或 HCD 只要启动数据脉冲并等待，即可确定在 A 端对总线供电时，是否发生 VBUS 有效中断。

该辅助功能将确保数据脉冲满足 > 5 ms 且 < 10 ms 的 OTG 要求。

23.7.8.3 B 断开到 A 连接 （跳变到 A 外设状态）

在 HNP 期间，从 OTG A 挂起状态跳变为 B 断开，且在 3 ms 内， A 设备必须在 A 外围设备状态下使能 DP 线上的上拉。若要开始硬件辅助，则必须满足以下条件：

- HABA 已设置。
- 主机控制器处于挂起模式。
- 设备正在断开。

硬件辅助包括以下步骤：

1. 硬件复位 OTG 控制器 （向 USBCMD 中的 RST 位写入 1）。
2. 硬件选择设备模式 （向 USBMODE 中的 CM[1:0] 位写入 10）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

3. 硬件设置 USBCMD 中的 RS 位，并使能必要的中断：
   - USB 复位使能 (URE) – 使能 USB 总线复位到设备中断。
   - 睡眠使能 (SLE) – 使能设备挂起中断。
   - 端口更改检测使能 (PCE) – 使能设备连接中断。

如果软件已使能该硬件辅助，则不得在跳变期间干预，且在收到来自设备控制器的中断，指明已发生复位中断，或确认内核已进入设备模式之前，不应对 OTG 内核中的任何寄存器执行写入操作。在任何时候，HCD/DCD 都不得激活内核软复位，因为该操作由硬件执行。在跳变期间，软件可能会看到因断开中断和 / 或其他杂散中断 （即 SOF/ 等），这些中断可能会或可能不会级联，并可以通过软复位清零，具体取决于软件响应时间。

内核在硬件辅助的协助下进入设备模式后，DCD 必须确保在主机发送设置数据包之前正确对 ENDPTLISTADDR 进行编程。由于复位持续时间（连接后可在数微秒内迅速启动）结束将至少需要 50 ms， DCD 在收到已检测到复位或 OTG 处于设备模式 （以先发生者为准）的通知后，必须在该时间内准备接收设置数据包。

如果控制器进入设备模式并启动 D+ 上拉后，A 外围设备未能发生复位，则设备控制器将中断 DCD，以指示已发生挂起。该辅助功能将确保满足参数 TA_BDIS_ACON_MAX = 3ms 的条件。

23.8 与 EHCI 标准的偏差

就支持 OTG 应用的双作用主机 / 设备控制器而言，无需遵循 EHCI 规范。EHCI 中未规定设备操作和 OTG 操作，因此此核支持的实施是 LPC43xx 特定的。该核的主机模式操作几乎是 EHCI 兼容的，只有本节说明的少数几处轻微差别。

各方面的差异详情汇总如下：

- 嵌入式事务转换器 – 在主机模式下，无需辅助控制器就可直接连接 FS 和 LS 设备。
- 设备操作 - 在主机模式下，设备操作寄存器通常处于禁用状态，因此在该模式下，设备模式通常透明。但是，以下章节列出了一些例外情况。
- OTG 操作 - 该设计包括 OTG 控制器。

23.8.1 嵌入式事务转换器功能

USB-HS OTG 控制器具有 USB 2.0 高速集线器事务转译器的功能，可直接连接全速和低速设备，而无需辅助控制器。尽管系统中没有独立的事务转译器模块，但已在 DMA 和协议引擎模块中实现通常与高速集线器相关的事务转译器功能。嵌入式事务转换器功能是 EHCI 接口的扩展功能，
但采用标准数据结构和 EHCI 规范中存在的操作模型，可支持全速和低速设备。

23.8.1.1 功能寄存器

为了支持嵌入式事务转换器功能，已将下例项目添加到功能寄存器：

- N_TT 位添加到 HCSPARAMS – 主机控制结构性参数 （参见 表 385）。
- N_PTT 添加到 HCSPARAMS – 主机控制结构性参数 （参见 表 385）。

23.8.1.2 操作寄存器

以下项目已添加至操作寄存器，以支持嵌入式事务转译器：

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- 新寄存器 TTCTRL （参见第 23.6.9 节）。
- 两位端口速度 (PSPD) 位添加到 PORTSC1 寄存器 （参见第 23.6.15 节）。

23.8.1.3 探索

在标准 EHCI 控制器设计中，EHCI 主机控制器驱动器通过注释端口复位操作后是否设置端口使能位，来检测全速 (FS) 或低速 (LS) 设备。只有在端口复位操作后及当主机和设备协商高速连接（即脉冲组成功完成）时，才会在标准的 EHCI 控制器实施方案中设置端口使能。由于此控制器有嵌入式事务转换器，端口使能将始终在端口复位操作后设置，无论主机设备线性调频的结果如何，且生成的端口速度将由 PORTSC1 中的 PSPD 字段指示 （参见第 23.6.15 节）。

表 422. 直接连接全速和低速设备的处理

标准 EHCI 模型	带嵌入式事务转译器的 EHCI
在按照连接和复位顺序对端口使能位进行设置后，假设设备 / 集线器为 HS。	在按照连接和复位顺序设置端口使能位后，从 PORTSC1 中记录设备 / 集线器速度。
由于假设 FS 设备和 LS 设备是来自 HS 集线器的下行设备，因此，通过最近集线器的集线器类别可执行所有端口级控制。	FS 设备和 LS 设备既可以是来自 HS 集线器的下行设备，也可以是直接连接的设备。如果 FS/LS 设备是来自 HS 集线器的下行设备，则通过最近集线器的集线器类别可执行端口级控制。如果是直接连接的 FS/LS 设备，则通过 PORTSC1 可实现端口级控制。
FS 和 LS 设备被假定为 HS 集线器 (HubAddr=X) 的下行设备，其中 HubAddr > 0 且 HubAddr 是从 HS 到 FS/LS 的总线跳变处的集线器 （即拆分目标集线器）的地址。	FS 和 LS 设备可以是 HS 集线器 （HubAddr = X [HubAddr > 0]）的下行设备，也可直接连接，其中HubAddr = TTHA（TTHA可编程且默认为0），且 HubAddr 是从 HS 到 FS/LS 的总线跳变处的根集线器的地址（即拆分目标集线器是根集线器）。

23.8.1.4 数据结构

HS 集线器中 FS/LS 事务所使用的相同数据结构还用于通过根集线器（带 sm 嵌入式事务转译器）的事务。下文介绍如何针对直接连接的 FS/LS 设备和集线器设置集线器地址和端点速度字段：

1. QH （用于直接连接 FS/LS） – 异步 （批量 / 控制端点）定期 （中断）
   - Hub 地址 = TTHA （默认 TTHA = 0）
   - 直接连接设备 / 集线器的事务：QH.EPS = 端口速度
   - 直接连接型 FS 集线器下行设备的事务：QH.EPS = 下行设备速度

     注：当 QH.EPS = 01 （LS）且 PORTSCx.PSPD = 00 （FS）时，将在发送 LS 流量之前发送 LS-pre-pid。

     最大数据包大小必须小于或等于 64，否则可能产生未定义的行为。
2. siTD （用于直接连接 FS） – 定期 （ISO 端点）

   所有 FS ISO 事务：

   集线器地址 = （默认情况下， TTHA = 0）

   siTD.EPS = 00 （全速）

   最大数据包大小必须小于或等于 1023，否则可能产生未定义的行为。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

23.8.1.5 操作模式

对于事务转译器行为（参见 USB 2.0 规范）以及在系统存储器和 USB-HS 集线器之间移动数据包的 EHCI 控制器来说，它们的操作模式已得到明确定义。由于嵌入式“事务转换器”存在于主机控制器内，因此 EHCI 主机控制器驱动器和 USB FS/LS 总线之间没有实际的物理总线。这些章节将简要说明 EHCI 和 “事务转换器”操作模式是如何在没有中间物理总线的情况下结合的。以下各节假设读者熟悉 EHCI 和 USB 2.0 事务转译器操作模式。

23.8.1.5.1 微帧管道

EHCI 操作模式采用 H 帧和 B 帧的概念来描述主机（H）与总线（B）之间的管道。嵌入式事务转译器使用 USB 2.0 规范中针对基于集线器的事务转译器所指定的相同管道算法。

值得注意的是，当在 EHCI 数据结构中对 S 掩模和 C 掩模进行编程以调度嵌入式事务转译器的定期传输时，EHCI 主机控制器驱动器必须遵循 EHCI 针对基于集线器的下行事务转译器对 S 掩模和 C 掩模编程所指定的相同规则。一旦定期传输用完，将移动任何存储的异步传输。异步传输得以实现，是因为只要可能便会执行异步传输，且其操作不会受 H 帧和 B 帧范围的限制，但异步传输无法通过 SOF （B 帧 0 起始）串音。

23.8.1.6 拆分状态机

开始和完成拆分操作模式与 EHCI 稍有不同，因为 EHCI 控制器与嵌入式事务转译器之间没有总线介质。如果因向 HS 集线器请求拆分而发生开始或完成拆分操作，则开始 / 完成拆分操作只是嵌入式事务转译器的内部操作。下表总结了从内部状态模拟信号握手而不是 HS 拆分总线流量的实际信号握手的条件。

表 423. 拆分状态机属性

	环境	模拟事务转译器响应
开始拆分	所有异步缓冲区满。	NAK
	所有定期缓冲区满。	ERR
	异步事务启动成功。	ACK
	启动定期事务。	无信号握手 （正常）
完成拆分	队列中找不到事务。	总线超时
	队列中的事务忙。	NYET
	队列中的事务完成。	[来自 LS/FS 设备的实际信号握手]

23.8.1.7 异步事务计划和缓冲区管理

以下 USB 2.0 规范项目应用于嵌入式事务转译器：

1. *USB 2.0 规范，章节 11.17.3*：提供序列且数据包长度模拟器确保全速 / 低速数据包不会在 SOF 时间串音。
2. *USB 2.0 规范，章节 11.17.4*：2 个数据管道的事务追踪。
3. *USB 2.0 规范，章节 11.17.5*：通过使用 TTCTRL 寄存器提供清除事务转译器缓冲区的能力。

23.8.1.8 定期事务计划和缓冲区管理

以下 USB 2.0 规范项目应用于嵌入式事务转译器：

1. *USB 2.0 规范，章节 11.18.6.[1-2]*：
   - 中止未决开始拆分：

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

EOF （且在微帧 6 时未开始）

空闲超过 4 微帧

- 中止未决完成拆分：

  EOF

  空闲超过 4 微帧

2. *USB 2.0 规范，章节 11.18.6.[7-8]*：
   - 最高达 16 条数据管道的事务追踪：

     一些应用可能不需要最多 16 个定期数据管道的事务追踪。将配置常数 VUSB_HS_TT_PERIODIC_CONTEXTS 更改为 4，即可选择将追踪限于仅 4 个定期数据管道。该限制可为内核节省大量的门数。

     注：如果将嵌入式事务转译器中的追踪管道数量限制为四 (4) 个，则将施加以下限制：无法通过每帧嵌入式 TT 调度超过 4 个定期事务 （中断 / 同步）。在 USB 规范中，选择的是 16 个，因为这足以确保高速至全速定期管道可以保持满的状态。如果使管道保持满的状态，则不会限制帧内可调度的定期事务数量，只是对总线上数据包的传输时间有所限制。

   - 完成拆分事务搜索：

     除微帧管道外，这些事务并无数据调度机制。嵌入式事务转译器假设，帧内调度的数据包数量不会超过帧持续时间 （1 ms），否则可会导致未定义行为。

23.8.1.9 多事务转译器

当前支持的嵌入式事务转换器的最大数目为 1，由 HCSPARAMS － 主机控制结构参数寄存器中的 N_TT 字段指定。

23.8.2 设备操作

除本节所述之外，主机控制器内设备操作控制器的共存对主机操作的 EHCI 兼容性影响不大。

23.8.2.1 USBMODE 寄存器

由于双作用控制器并非在主机或设备模式下初始化，因此，必须针对主机操作对 USBMODE 寄存器进行编程后， EHCI 主机控制器驱动器才能开始 EHCI 主机操作。

23.8.2.2 寄存器文件的非零字段

功能寄存器和操作寄存器中的部分保留字段和保留地址用于设备模式时，必须遵循以下规则：

- 通过操作寄存器向所有 EHCI 保留字段（其中一些为设备字段）执行的写入操作应始终写入 0。这是必须遵循的设备控制器驱动器的 EHCI 要求。
- 主机控制器执行的读取操作必须适当掩码 EHCI 保留字段 （其中一些为设备字段），因为主机模式中未定义专用于设备的字段。

23.8.2.3 SOF 中断

用于设备模式的此 SOF 中端被作为主机模式的自由运行 125us 中断。EHCI 没有指定该中断，但为了方便起见，已添加该中断来作为可能的软件时间基准。参见 USBSTS（第 23.6.4 节）和 USBINTR （第 23.6.5 节）寄存器。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

23.8.3 EHCI 其它变型

23.8.3.1 探索

23.8.3.1.1 端口复位

EHCI 指定的端口连接方法要求在 PORTSCx 寄存器中设置端口复位位，持续时间 10 ms。由于支持非高速设备连接所需的复杂性，在设计中已包含了计数器，它可计数 10ms 的复位脉冲，以减轻对软件测量此持续时间的要求。因此，基本连接汇总如下：

- [端口更改中断] 发生端口连接更改，以通知主机控制器驱动器设备已连接。
- 软件应写入 1 来复位设备。
- 软件应写入 0 在 10 ms 后复位设备。

  此步骤是标准 EHCI 设计所必要的，对于此实施可忽略。如果在复位进行过程中，EHCI 主机控制器驱动器试图将 0 写入复位位，则该写入操作会被忽略，且复位将继续进行直到完成。
- [端口更改中断]发生端口使能更改，以通知主机控制器设备目前正在运行，而此时端口速度已确定。

23.8.3.1.2 端口速度检测

端口更改中断指示端口已使能后， EHCI 协议栈应确定端口速度。 EHCI 实施方案将为未以高速连接的任何设备重新分配端口所有者，而该主机控制器可直接连接非高速设备。因此，以下差异对于端口速度检测至关重要：

- 端口所有者为只读且始终为 0。
- 2 位端口速度指示器已添加至 PORTSC 中，可向主机控制器驱动器提供端口的当前运行速度。
- 已将一个1位高速指示器添加到PORTSC，以指示该端口处于高速还是全速/低速。对于上述 2 位端口速度指示器，此信息是冗余的。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

23.9 设备数据结构

本节定义用于设备控制器驱动器 （DCD）软件与设备控制器之间传达控制、状态和数据的接口数据结构。本章中的数据结构定义支持 32 位存储器缓冲区地址空间。

注：软件必须确保设备控制器可访问的接口数据结构不会跨越 4k 页面边界

本章中定义的数据结构 （从设备控制器的角度看）是只读和可读 / 写字段的混合。设备控制器必须在所有数据结构写入时保留只读字段。

图 50. 端点队列头组织

设备队列头在 ENDPOINTLISTADD 指针指向的存储器连续区域内以阵列排列。列表中的偶数设备队列头支持接收端点 （OUT/SETUP），而列表中的奇数列队头用于发送端点（IN/INTERRUPT）。设备控制器将基于从 USB 总线接收的端点号编索引到此数组中。响应所有已启动传输事务必需的所有必要信息都包含在此列表中，所以设备控制器可立即响应进入请求而无需遍历链接列表。

注：端点队列头列表必须对齐到 2 kB 边界。

23.9.1 端点队列头 (dQH)

设备端点队列头 （dQH）指管理所有传输的位置。 dQH 为 48 字节数据结构，但必须在 64 字节边界上对齐。在端点启动期间， dTD （设备传输描述符）会复制到 dQH 的重叠区，它从 nextTD 指针双字处开始，直到缓冲区指针双字的结尾。传输完成后，dTD 状态双字将在当前 TD 指针指向的 dTD 中更新。当数据包正在处理中时，dQH 的重叠区将用作 dTD 的暂存区，以便设备控制器可以极低的延迟访问所需的信息。

23.9.1.1 端点性能和特点

该双字指定端点的静态信息，换言之，该信息不会随着端点的使用时间而更改。在相应端点使能时，设备控制器软件不会尝试修改此信息。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 424. 端点性能和特点

访问类型	位	名称	描述
RO	31:30	MULT	每个事务描述符执行的数据包数量 00 - 执行 USB 变量长度协议所示的 N 个事务，其中，N 使用 dTD 中的 Max_packet_length 和 Total_bytes 字段来计算。 01 - 执行一个事务 10 - 执行两个事务 11 - 执行三个事务 **注：**非同步端点必须设置 MULT = 00。 **注：**同步端点必须根据需要设置 MULT = 01、 10 或 11。
RO	29	ZLT	零长度终止选择 总传输长度是倍数时，此位用于非同步端点，表示收到零长度数据包时终止传输。 0 - 使能零长度数据包以终止长度等于 Max_packet_length （默认值）倍数的传输。 1 - 禁用长度等于 Max_packet_length 倍数的传输中的零长度数据包。
RO	28:27	-	保留
RO	26:16	Max_packet_length	相关端点的最大数据包大小 (< 1024)
RO	15	IOS	设置时中断 该位用于控制类型端点，以指示是否设置 USBINT 以响应收到的设置。
RO	14:0	-	保留

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 51. 端点队列头数据结构

23.9.1.2 传输重叠

重叠区的七个双字代表设备控制器的事务工作区。通常采用的操作模式是，设备控制器可以检测重叠区是否包含有效传输的描述符。如果不包含有效传输，则将不会读取相关的端点。

读取端点后， dTD 将由设备控制器复制到该队列头重叠区。在传输过期之前，软件不得写入队列头重叠区或相关传输描述符。在传输完成后，设备控制器便会将结果写回原始传输描述符，并推进队列。有关重叠字段的描述，请参见 dTD。

23.9.1.3 当前 dTD 指针

设备控制器使用当前 dTD 指针查找正在执行的传输。该字仅供设备控制器 （硬件）使用，且 DCD 软件无法修改。

表 425. 当前 dTD 指针

访问类型	位	名称	描述
R/W（仅限硬件）	31:5	Current_TD_pointer	当前 dTD 指针 该字段为传输重叠区代表的 dTD 的指针。在端点启动或队列推进期间，该字段将由设备控制器修改为下一个dTD 指针。
-	4:0	-	保留

23.9.1.4 设置缓冲区

设置缓冲区是跟随设置 PID 的 8 字节数据的专用存储区。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

注: 每个端点都有一个 TX 和一个 RX dQH 与其关联，且仅 RX 队列头用于接收设置数据包。

表 426. 设置缓冲区

双字	访问类型	位	名称	描述
1	R/W	31:0	BUF0	设置缓冲区 0 该缓冲区包含传入设置缓冲区数据包的字节 3 至 0，并由设备控制器写入而由软件读取。
2	R/W	31:0	BUF1	设置缓冲区 1 该缓冲区包含传入设置缓冲区数据包的字节 7 至 4，并由设备控制器写入而由软件读取。

23.9.2 端点传输描述符 (dTD)

dTD 向设备控制器描述要为给定传输发送 / 接收的数据的位置和数量。 DCD 不应尝试修改活动 dTD 中除下一个链接指针外的任何字段，且该指针仅应按照第 23.10.11 节中所述进行修改。

表 427. 下一个 dTD 指针

访问类型	位	名称	描述
RO	31:5	Next_link_pointer	下一个链接指针 该字段包含将处理的下一个 dTD 的物理存储器地址。该字段分别对应于存储器地址信号 [31:5]。
	4:1	-	保留
	0	T	终止 当队列中没有有效项时，该位向设备控制器发出指示。 1 - 指针无效 0 - 指针有效，即指针指向有效的传输元素描述符。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 428. dTD 标记

访问类型	位	名称	描述
-	31	-	保留
R/W	30:16	Total_bytes	总字节 该字段指定通过该传输描述符移动的总字节数。该字段按事务期间实际移动的字节数递减，且只有在事务成功完成时，才会递减。 软件可以向该字段写入的最大值为 0x5000 (5 x 4 kB)，即五页指针可访问的最大字节数。尽管可以创建多达 20 kB 的传输，但该处假设第一页的首个偏移量为零。如果无法预先确定偏移量，则将总字节限制为 16 kB 可保证跨越第五页。因此，建议的最大总字节数 = 16 kB （0x4000）。 如果主机控制器获取该传输描述符时， Total_bytes = 0 且该 dTD 的状态字段中已设置有效位，则设备控制器将执行零长度事务，并停用 dTD。 **注：**对于 IN 传输， Total_bytes 无需为 Max_packet_length 的偶数倍。如果软件构建该类 dTD，则最后一个事务将始终小于 Max_packet_length。
RO	15	IOC	完成时中断 该位用于指示当设备控制器完成该dTD时，是否设置USBINT。 1 - USBINT 已设置。 0 - USBINT 未设置。
-	14:12	-	保留
RO	11:10	MultO	乘数覆盖 （参见第 23.9.2.1 节示例） 该字段可用于发送 ISO，以替代 dQH 中的 MULT 字段。对于非发送 ISO 的所有数据包类型，该字段必须为 0。 00 - 执行 USB 变量长度协议所示的 N 个事务，其中， N 使用 dTD 中的 Max_packet_length 和 Total_bytes 字段来计算。 01 - 执行一个事务 10 - 执行两个事务 11 - 执行三个事务 **注：**非 ISO 和非 TX 端点必须设置 MultO=00。
	9:8	-	保留
R/W	7:0	状态	状态 设备控制器使用该字段将个别执行状态传回软件。该字段包含该 dTD 上执行的最后一个事务的状态。 位 7 = 1 - 状态：活动 位 6 = 1 - 状态：停止 位 5 = 1 - 状态：缓冲区错误 位 4 - 保留 位 3 = 1 - 状态：事务错误 位 2 - 保留 位 1 - 保留 位 0 - 保留

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 429. dTD 缓冲页指针列表

访问类型	位	名称	描述
RO	31:12	BUFF_P	选择数据包缓冲区存储器中的页面偏移量。非虚拟存储器系统通常会将缓冲区指针设为一系列递增整数。
	第 0 页：11:0	CURR_OFFS	数据包开始所在的 4 kB 缓冲区的偏移量。
	第 1 页：10:0	FRAME_N	由设备控制器写入，指示数据包完成的帧数。该位通常用于将同步端点上数据包的完成时间互相关连。

23.9.2.1 确定同步 IN 端点数据包的数目

以下示例说明如何使用 dQH 中的 MULT 字段及 dTD 中的 MultO 字段来控制同步端点上输入事务中发送的数据包数量。

示例 1

MULT = 3； Max_packet_size = 8； Total_bytes = 15； MultO = 0 （默认值）

在该例中，将发送三个数据包：数据 2（8 个字节）、数据 1（7 个字节）、数据 0（0 个字节）。

示例 2

MULT = 3； Max_packet_size = 8； Total_bytes = 15； MultO = 2

在该例中，将发送两个数据包：数据 1 （8 个字节）、数据 0 （7 个字节）。

为了优化 IN 传输的效率，软件应计算 MultO = （Total_bytes/Max_packet_size）的最大整数。如果 Total_bytes = 0，则 MultO 应为 1。

23.10 设备操作模型

设备操作的功能是将内存图像中的请求传输到通用串行总线以及从中传出。设备控制器将使用一组链接列表传输描述符 （由队列头指向）执行数据传输。下面各节说明了从设备控制器驱动器 (DCD) 角度说明设备控制器的用途，并进一步说明特定 USB 总线事件如何与设备控制器编程器的接口中的状态更改相关。

23.10.1 设备控制器初始化

在硬件复位后，便会禁用设备，直到“运行 / 停止”位设为 1 为止。在禁用状态下，USB_DM 的上拉无效，这可防止发生连接事件。在发生设备连接之前，至少必需设置端点零的队列头。设备使能后不久，设置数据将到达端点 0，然后便会发生 USB 复位。队列头必须做好准备，以便设备控制器可以存储传入的设置数据包。

为了初始化设备，软件应执行以下步骤：

1. 在 USBMODE 寄存器中将控制器模式设为设备模式。

   注：从主机模式跳变到设备模式需要在修改 USBMODE 之前复位设备控制器。

2. 在系统存储器中分配并初始化设备队列头 （参见第 23.9 节）。

   最小：初始化设备队列头 0 Tx 和 0 Rx。

   注：所有与控制端点关联的设备队列头必须在控制端点使能之前初始化。在端点使用之前，必须初始化非控制设备队列头，而端点使能之前则没有必要。

3. 配置 ENDPOINTLISTADDR 指针 （参见第 23.6.8 节）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

4. 使能与 USB-HS 内核相关的微处理器中断。

   建议：使能所有设备中断，包括：USBINT、 USBERRINT、端口更改检测、 USB 复位接收、 DCSuspend （参见表 394）。

5. 将运行 / 停止位设为运行模式。

   设置运行位后，便会发生设备复位。 DCD 必须监控复位事件并调整软件状态，如下面“端口状态和控制”一节的 “总线复位”部分中所述。

   注：端点 0 仅设计为控制端点，且无需使用 ENDPTCTRL0 寄存器来配置。

因为收到的第一个数据包始终为设置数据包，因此也无需首先启动端点 0。第一个设置数据包的内容将需要根据 USB 设备框架命令集作出响应 （参见 *USB 规范 2.0 版*的*第 9 章*）。

23.10.2 端口状态和控制

设备控制器从芯片或系统复位进入加电状态。如果运行 / 停止位设为 1，便会从上电状态跳变至连接状态。在接收总线复位后，根据 *USB 规范修订版 2.0 附录 C.2* 中所述，端口将进入 defaultFS 或 defaultHS 状态。下面的状态图描述了 USB 2.0 设备的状态。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 52. 设备状态图

在设备控制器中可实现上电、连接、默认 FS/HS 和挂起 FS/HS 状态，并使用以下状态位发送至 DCD：

- DCSuspend – 参见表 392。
- USB 复位接收 – 参见表 392。
- 端口更改检测 - 参见表 392。
- 高速端口 - 参见表 409。

DCD 负责维护状态变量以区分 DefaultFS/HS 状态和 “地址 / 配置” 状态。将状态从默认更改为地址和已配置状态是 *USB 2.0 规范设备框架部分*中所述的枚举过程的一部分。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

进入地址状态后，必须由 DCD 对设备地址寄存器 （DEVICEADDR）进行编程。

进入已配置状态表示，设备运行中使用的所有端点已通过编程ENDPTCTRLx寄存器及初始化相关队列头正确初始化。

23.10.3 总线复位

主机通过总线复位来初始化下行设备。当检测到总线复位时，设备控制器将重新协商其连接速度、将设备地址复位为 0 以及通知 DCD 发生中断 （假设 USB 复位中断使能已设置）。收到复位后，将禁用所有端点（端点 0 除外），且设备控制器将取消任何已启动的事务。下面将明确启动的概念，但收到复位时， DCD 必须执行以下任务：

- 通过读取 ENDPTSETUPSTAT 寄存器并将相同的值写回 ENDPTSETUPSTAT 寄存器，清除所有设置令牌信号量。
- 通过读取ENDPTCOMPLETE寄存器并将相同的值写回ENDPTCOMPLETE寄存器，清除所有端点完成状态位。
- 等到ENDPTPRIME中的所有位为0，然后将0xFFFFFFFF写入ENDPTFLUSH后，取消所有已启动的状态。
- 读取PORTSCx寄存器中的复位位，并确保该位仍处于活动状态。在最少3 ms内将发生USB 复位，而 DCD 必须在复位结束前到达复位清除中的该点；否则，建议进行设备控制器硬件复位 （极少情况下）。

  注：可通过将 1 写入 USBCMD 复位中设备控制器的复位位中执行硬件复位。注：硬件复位通过将运行 / 停止位清零，将导致设备从总线分离。因此，在硬件复位后，DCD 必须完全重新初始化设备控制器。
- 释放所有分配的dTD，因为它们将不再由设备控制器执行。如果这是DCD首次处理USB复位事件，则可能不会分配 dTD。此时， DCD 可能向 OS 释放控制，因为在指示端口更改检测前，不允许对设备控制器做出进一步的更改。
- 在端口更改检测之后，设备达到默认状态，DCD可读取PORTSCx来确定设备是在FS还是 HS 模式下操作。此时，设备控制器进入正常工作模式，且根据 *USB2.0 规范第 9 章 - 设备框架*， DCD 可以开始枚举。

  注：设备 DCD 可使用 FS/HS 模式信息来确定设备的带宽模式。

在某些应用中，可能无法在 FS 模式下使能一个或多个管道。除数据速率问题外，FS 与 HS 模式之间的 DCD 操作并无差异。

23.10.4 挂起 / 恢复

23.10.4.1 挂起

为了降低功耗，当设备注意到指定期间无总线流量时， USB 设备将自动进入挂起状态。在挂起时， USB 设备会保持任何内部状态，包括其地址和配置。已连接设备必须准备在上电的任何时候挂起，无论它们是否获分配非默认地址、配置或上述任何一种情况。总线活动可能因主机进入其自身的挂起模式而停止。此外，当USB设备连接的集线器端口禁用时，USB设备也将进入挂起状态。

如果存在总线活动，USB 设备便会退出挂起模式。通过电子信号指示远程唤醒，USB 设备也可以请求主机退出挂起模式。设备发出远程唤醒信号为可选功能。如果 USB 设备能够发出远程唤醒信号，则该设备必须支持主机使能和禁用此功能。当设备复位时，远程唤醒信号必须禁用。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

23.10.4.1.1 操作模式

当检测到挂起信号或在上游端口无活动的时间超出指定时段时，设备控制器转变到挂起状态。设备控制器进入挂起状态后， DCD 便会收到中断通知 （假设 DC 挂起中断已使能）。当 PORTSCx 中的 DCSuspend 位设置为 1 时，设备控制器挂起。

设备控制器挂起时的 DCD 响应是应用程序特定的，且可能涉及切换到低功率操作。有关挂起状态下总线功率限制的信息，请参见 *USB 2.0 规范*。

23.10.4.2 恢复

如果设备控制器挂起，在面向上行的端口上收到任何非空闲信号时，便会恢复其操作。此外，设备可以强制向上行端口发出恢复信号，以向系统发出恢复操作的信号。在设备处于挂起状态时，恢复信号通过将 1 写入 PORTSCx 中的恢复位向上游发送。将恢复信号发送到上游端口将导致主机发出恢复信号并将挂起的总线段（一个以上的设备）带回到激活状态。

注：在可以使用恢复信号之前，主机必须使用 *USB 2.0 规范的设备框架 （第 9 章）*中定义的设置功能命令使能它。

23.10.5 管理端点

*USB 2.0 规范*将一个端点，也称为设备端点或地址端点，定义为 USB 设备的唯一可寻址部分，它可在主机和设备之间的通信通道中收发数据。端点地址通过结合端点号和端点方向进行指定。

主机和特定设备上端点之间的通道代表一个数据管道。设备的端点 0 始终为用于设备发现和枚举的控制类型数据通道。 USB 支持的其他端点类型包括批量、中断和同步。每种端点类型都有与数据包响应和错误处理相关的特定行为。有关端点操作的更多详情，请参见 *USB 2.0 规范*。

LPC43xx 支持多达六个端点。

每个端点方向实质上都独立，且可以通过每个方向不同的行为进行配置。例如， DCD 可以将端点 1 - IN 配置为批量端点，而将端点 1 - OUT 配置为同步端点。这有助于降低设备操作所需的端点总数。唯一例外是控制端点必须使用单个端点号上的两个方向，才能作为控制端点使用。例如，端点 0 始终为控制端点，且使用两个方向。

每个端点方向需要在存储器中分配一个队列头。如果最多 4 个端点号，每个端点方向的一个队列头将供设备控制器使用，则需要 8 个队列头。端点的操作和队列头的使用将在本文档的后面部分介绍。

23.10.5.1 端点初始化

硬件复位后，除了端点零之外的所有端点都未进行初始化并被禁用。 DCD 必须通过写入 ENDPTCTRLx 寄存器中的配置位配置并使能每个端点 （参见表 421）。每个 32 位 ENDPTCTRLx都分为上半部分和下半部分。ENDPTCTRLx的下半部分用于配置接收或OUT 端点，而上半部分同样地用于配置相应的发送或 IN 端点。控制端点必须在 ENDPTCTRLx 寄存器的上半部分和下半部分中配置相同，否则会产生未定义的行为。下表显示如何创建端点初始化的配置字。

表 430. 设备控制器端点初始化

域	值
数据切换复位	1
数据切换禁止	0
端点类型	00 – 控制

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 430. 设备控制器端点初始化 *（续）*

域	值
	01 - 同步
	10 - 批量
	11 - 中断
端点停止	0

23.10.5.2 停止

两种情况下，设备控制器可能需要返回主机一个 STALL：

1. 第一种情况是**功能失速**，如 *USB 2.0 设备框架（第 9 章）*所述，这是由 DCD 设置的条件。功能 STALL 仅用于非控制端点，并在 ENDPTCTRLx 寄存器中设置与指定端点和指定方向相关的端点 STALL 位，即可在设备控制器中使能。在功能失速状态下，设备控制器会继续将 STALL 响应返回到各端点和方向上发生的所有事务，直到端点失速位被 DCD 清除。
2. **与功能 STALL 不同的是，**协议 STALL（protocol stall）用于控制端点，并在新控制事务开始时（设置阶段）由设备控制器自动清除。在使能协议 STALL 时，DCD 应成对使能 STALL 位 （两个方向）。到 ENDPTCTRLx 寄存器的单次写入可确保同时设置两个失速位。

注：在操作模式下对 ENDPTCTRLx 寄存器执行任何写入操作时，必须保留端点类型字段（即执行读取 - 修改 - 写入）。

表 431. 设备控制器失速响应矩阵

USB 数据包	端点 STALL 位	STALL 位的影响	USB 响应
非控制端点收到的 SETUP 数据包。	不适用	无	STALL
非控制端点接收的 IN/OUT/PING 数据包。	1	无	STALL
非控制端点接收的 IN/OUT/PING 数据包。	0	无	ACK/NAK/NYET
控制端点收到的 SETUP 数据包。	不适用	已清除	ACK
控制端点接收的 IN/OUT/PING 数据包。	1	无	STALL
控制端点接收的 IN/OUT/PING 数据包。	0	无	ACK/NAK/NYET

23.10.5.3 数据切换

数据切换是一种维护主机和设备之间任何给定数据管道的数据一致性的机制。有关数据切换的更多信息，请参见 *USB 2.0 规范*。

23.10.5.3.1 数据切换复位

DCD 可复位数据切换状态位，并通过将 1 写入 ENDPTCTRLx 寄存器的数据切换复位位中，使数据切换序列在设备控制器中复位。只有在配置 / 初始化端点或从 STALL 状态返回时，才需要执行复位。

23.10.5.3.2 数据切换禁止

注：此功能仅用于测试目的，且绝不能在正常设备控制器操作期间使用。

将数据切换禁止位设置为激活 (1) 会导致设备控制器忽略正常发送的数据切换模式，并接受所有进入的数据包，无论数据切换状态如何。在正常操作模式下，设备控制器会根据数据切换来检查 DATA0/DATA1 位，以确定数据包是否有效。如果数据 PID 与设备控制器为该端点所保持的数据切换状态位不匹配，则数据切换将被视为无效。如果数据切换无效，则设备

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

控制器将假设已收到数据包，并丢弃有关数据包（不会向 DCD 报告该情况）。为阻止主机控制器重新发送同一数据包，设备控制器将通过使用 ACK 或 NYET 响应对错误数据包进行确认，对其进行响应。

23.10.6 数据包传输操作模式

USB 总线上的所有事务将由主机启动，而设备必须在 USB 2.0 规范规定的周转时间内对主机的任何请求作出响应。在 USB 1.1 全速率和低速率情况下，此周转时间很重要，因此 USB 1.1 设备控制器设计为使其可以访问主存储器或者中断主机协议处理器，以响应 USB 1.1 事务。USB 2.0 设备控制器的架构必须有所不同，这是因为，如果只增加时钟频率，相同的方法将无法满足 USB 2.0 高速周转时间的要求。

USB 主机将以单个管道无法精确预测的顺序向设备控制器发送请求，因此无法准备单个数据包供设备控制器执行。但是，如果考虑到端点号和方向，则就能预测数据包请求的顺序。例如，如果将端点 3（发送方向）配置为批量管道，则就可以预测主机将向该端点发送 IN 请求。经过这样的设计后，该设备控制器可以预期主机请求，从而为每个端点 / 方向准备数据包。准备设备控制器来发送或接收数据以响应总线上主机启动的事务的过程称为启动端点。该术语在以下文档中用于描述设备控制器操作，以便 DCD 在经过适当设计后可使用启动。另外需注意的是，“刷新”一词用于描述清除位于执行队列中的数据包的操作。

23.10.6.1 启动发送端点

如果启动发送端点，则将导致设备控制器获取由设备队列头（dQH）所指向的事务的设备传输描述符（dTD）。获取 dTD 后， dTD 将存储在 dQH 中，直到设备控制器完成 dTD 描述的传输为止。通过将 dTD 存储在 dQH 中，设备控制器可获取处理主机请求所需的操作上下文，且无需遵循链接列表，只需在收到主机请求时从 dQH 开始。设备加载了 dTD 后，数据包中的主要数据将被存储在设备控制器的 FIFO 中。该 FIFO 会分成多个虚拟通道，以便可以存储任何端点（最多四个端点）的前导数据。

启动请求完成后， ENDPTSTATUS 寄存器中将指示已启动的端点状态。对于已启动的发送端点，设备控制器可以响应主机的 IN 请求，并满足高速 USB 严格的总线周转时间。由于只有前导数据存储在设备控制器 FIFO 中，因此，在事务开始后，设备控制器需要开始填充前导数据后的数据。 FIFO 必须足够大，要考虑系统存储器母线上发生的最大延迟。在 LPC43xx 系列中，各个 IN 端口都使用 128 x 36 位的双端口存储器 FIFO。

23.10.6.2 启动接收端点

从 DCD 的角度而言，启动接收端点与启动发送端点相同。在设备控制器上，操作模式的主要差异在于，由于仅从主机接收前导数据包数据，因此该数据并不存在数据移动。需要注意的是，作为架构的一部分，接收端点的 FIFO 不会象发送 FIFO 那样被划分为多个通道。因此， RX FIFO 的大小不会根据端点数量而调整。

23.10.7 中断 / 批量端点操作模式

设备控制器用于中断和用于批量端点时的特性是相同的。除非端点已启动，否则送向批量管道的所有有效的IN和OUT事务都与NAK进行信号握手。一旦端点启动，便会开始数据传送。

当传输描述符描述的数据包完成时，设备控制器将停用 dTD。每个 dTD 根据 USB 可变长度传输协议来描述要传输的 N 个数据包。下页的公式和表格说明了设备控制器如何根据总字节数和最大数据包长度来计算 USB 发送 / 接收的数据包数量和长度。

零长度终止（ZLT）= 0

N = INT(Number Of Bytes/Max. Packet Length) + 1

零长度终止（ZLT）= 1

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

N = MAXINT(Number Of Bytes/Max. Packet Length)

表 432. 可变长度传输协议示例 (ZLT = 0)

字节 （dTD）	最大数据包长度 (dQH)	N	P1	P2	P3
511	256	2	256	255	-
512	256	3	256	256	0
512	512	2	512	0	-

表 433. 可变长度传输协议示例 (ZLT = 1)

字节 （dTD）	最大数据包长度 (dQH)	N	P1	P2	P3
511	256	2	256	255	-
512	256	2	256	256	-
512	512	1	512	-	-

注：对于批量、中断和控制端点， dQH 中的 MULT 字段必须设置为 00。

当dTD描述的所有数据包成功发送后，TX-dTD便完成。完成后，dTD中的总字节数将为零。

RX-dTD 完成仅当：

- dTD 中描述的所有数据包成功接收。成功接收后， dTD 中的总字节数将为零。
- 收到短数据包（字节数 < 最大数据包长度）。传输完成成功； DCD必须检查dTD中的总字节数，以确定剩余的字节数。根据 dTD 中剩余的总字节数， DCD 可以计算实际收到的字节数。
- 收到长数据包（字节数 > 最大数据包尺寸）或（收到的总字节数 > 指定的总字节数）。这是错误情况。设备控制器将丢弃剩余的数据包，并在 dTD 中设置缓冲区错误位。此外，端点将刷新，而 USBERR 中断位将变为有效。

成功完成 dTD 描述的单个 （多个）数据包后， dTD 中的有效位将清零，而在终止位清零时，将使用下一个指针。设置终止位时，设备控制器将刷新端点 / 方向，并停止该端点 / 方向的操作。如果未成功完成数据包 （参见上述长数据包）， dQH 将仍指向出错的 dTD。若要从该错误条件中恢复，则 DCD 必须将有效位清零以正确初始化 dQH，并在尝试重新启动端点前更新 nextTD 指针。

注：设备控制器会自动重试所有数据包级别的错误，例如缺失的信号握手或 CRC 错误。

处理该类错误无需与 DCD 通信。

23.10.7.1 中断 / 批量端点总线响应矩阵

表 434. 中断 / 批量端点总线响应矩阵

令牌类型	STALL	未启动	启动	下溢	溢出
设置	忽略	忽略	忽略	不适用	不适用
输入	STALL	NAK	发送	BS 错误	不适用
输出	STALL	NAK	收到并 NYET/ACK	不适用	NAK
Ping	STALL	NAK	ACK	不适用	不适用
无效	忽略	忽略	忽略	忽略	忽略

[1] BS 错误 = 强制位填充错误

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

[2] NYET/ACK – NYET 根据 USB 变量长度协议，除非传输描述符有剩余的数据包，否则应答。

[3] SYSERR – 如果正确调整延迟 FIFO 且 DCD 可以响应，则不会发生系统错误。

23.10.8 控制端点操作模式

23.10.8.1 设置阶段

控制端点的所有请求将从设置阶段开始，然后是可选的数据阶段和所需的状态阶段。除非使能设置锁定，否则设备控制器将始终接受设置阶段。

将使能设置锁定，这样就可忽略之后的设置数据包。将设置数据包锁定可确保软件在读取队列头中所存储的设置数据包时，不会写入该数据，因为读取可能导致设置数据包无效。

在硬件中，可以禁用设置锁定机制，而新的绊网式信号量将确保从队列头中获取设置数据包负载，且不会因传入的设置数据包而损坏。由于因软件中断延迟时间长而忽略了重复的设置数据包是合规性问题，因此该行为为首选行为。

23.10.8.1.1 使用设置锁定机制处理设置数据包

在收到中断并检查 USBMODE 以确定是否收到特定管道上的设置数据包后：

1. 将 dQH.SsetupBuffer 的内容复制到本地软件字节数组中。
2. 写入 “1” 可将相应的 ENDPTSETUPSTAT 位清零，并可因此禁用设置锁定 （即只要设置一到达，设置锁定就会激活。通过对 ENDPTSETUPSTAT 执行写入操作，设备控制器将接收新的设置数据包。）。
3. 使用本地软件字节数组副本和执行状态 / 信号握手阶段处理设置数据包。

   注：接收新的设置数据包后，状态和 / 或信号握手阶段可能仍然在先前的控制序列中待定。在链接最新设置数据包的新状态和/或信号握手dTD之前，应刷新并释放这些阶段。
4. 启动各状态 / 信号握手阶段前要确保 ENDPTSETUPSTAT 为 0。启动各状态 / 信号握手阶段前要确保 ENDPTSETUPSTAT 为 0。从向 ENDPTSETUPSTAT 写入 1 到读回 0 的时间可能根据主线上流量类型的不同而变化，最长约为 1 ms。但是必须保证在步骤 1 后、启动各状态 / 信号阶段前使 ENDPTSETUPSTAT 变为 0。

注：若要限制设置数据包对设置锁定机制（如使用）的影响，则 DCD 应指定响应设置数据包的优先级高于响应其他数据包完成的优先级

23.10.8.1.2 使用绊网机制处理设置数据包

- 通过对USBMODE的设置锁定模式(SLOM)写入1来禁用设置锁定。（在初始化后）。如下所述，使用绊网时无需设置锁定。

  注：将设置锁定模式保留为 0 将导致前面 2.3 所述的硬件特性。
- 收到中断并检查 ENDPTSETUPSTAT 以确定是否收到特定管道上的设置数据包后：

  a. 写入 1 清除相应的 ENDPTSETUPSTAT 位。

  b. 将 dQH.SetupBuffer 的内容复制到本地软件字节数组中。

  c. 向 USBCMD 寄存器中的设置绊网 (SUTW) 写入 1。

  d. 在 USBCMD 寄存器中读取设置绊网 (SUTW)。（如果已被设置，则继续；如果已被清零，则转到 b）。

  e. 写入 0 清除 USBCMD 寄存器中的设置绊网 (SUTW)。

  f. 使用本地软件字节数组副本和执行状态 / 信号握手阶段处理设置数据包。

  g. 启动各状态 / 信号握手阶段前要确保 ENDPTSETUPSTAT 为 0。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- 在上述步骤 a) 完成后、启动各状态 / 信号握手阶段前，应使用轮询循环等待，直到 ENDPTSETUPSTAT 变为 0 为止。
- 从向ENDPTSETUPSTAT写入1到读回0的时间非常短(~1-2 us)，因此DCD中的轮询循环没有危害。

注：接收新的设置数据包后，状态和 / 或信号握手阶段可能仍然在先前的控制序列中待定。在链接最新设置数据包的新状态和 / 或信号握手 dTD 之前，应刷新并释放这些阶段。

23.10.8.2 数据阶段

在设置阶段后， DCD 必须为数据阶段创建设备传输描述符并启动传输。

在启动数据包后，DCD 必须在验证启动已完成后立即读取 ENDPTSETUPSTAT 寄存器，以验证是否尚未收到新的设置数据包。当 ENDPTPRIME 寄存器中的相关位为 0，且 ENDPTSTATUS 寄存器中的相关位为 1 时，启动将完成。如果启动失败，即 ENDPTPRIME 位为 0，而 ENDPTSTATUS 位未设置，则启动失败。启动失败仅可能因 dQH 或 dTD 设置不当或启动操作期间设置到达所致。如果 ENDPTPRIME 位清零后，指示收到新的设置数据包，则可能释放传输描述符，而 DCD 必须重新中断设置数据包。

如果数据阶段启动后设置到达，则设备控制器将自动清除启动状态 （ENDPTSTATUS），以强制数据与设置数据包一致。

注：对于批量、中断和控制端点， dQH 中的 MULT 字段必须设置为 00。

注：数据阶段的数据包错误处理和先前描述的批量数据包的处理相同。

23.10.8.3 状态阶段

与数据阶段相似， DCD 必须创建传输描述符 （字节长度为零），并启动状态阶段的端点。而 DCD 也必须根据上述内容，在数据阶段中对 ENDPTSETUPSTAT 执行相同的检查。

注：对于批量、中断和控制端点， dQH 中的 MULT 字段必须设置为 00。

注：数据阶段的数据包错误处理和先前描述的批量数据包的处理相同。

23.10.8.4 控制端点母线响应矩阵

下表列出了设备控制器根据设备控制器状态对控制端点上数据包的响应。

表 435. 控制端点母线响应矩阵

令牌类型	端点状态					设置锁定
	STALL	未启动	启动	下溢	溢出	
设置	ACK	ACK	ACK	不适用	SYSERR	-
输入	STALL	NAK	发送	BS 错误	不适用	不适用
输出	STALL	NAK	收到并 NYET/ACK	不适用	NAK	不适用
Ping	STALL	NAK	ACK	不适用	不适用	不适用
无效	忽略	忽略	忽略	忽略	忽略	忽略

[1] BS 错误 = 强制位填充错误

[2] NYET/ACK – NYET 根据 USB 变量长度协议，除非传输描述符有剩余的数据包，否则应答。

[3] SYSERR – 如果正确调整延迟 FIFO 且 DCD 可以响应，则不会发生系统错误。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

23.10.9 同步端点操作模式

同步端点用于实时调度传送数据，而其操作模式与主机限制的批量、中断和控制数据管道显著不同。主机控制器通过以下方式完成实时传送：

- 每（微）帧确切发送 / 收到 MULT 数据包。注：MULT 指设备队列头中的两位字段。变量长度数据包协议不适用于同步端点。
- 不会使用 NAK 响应。相反，会发送零长度数据包，以响应未启动端点的 IN 请求。对于未启动的 RX 端点，对 OUT 事务作出的响应是忽略设备控制器中的数据包。
- 启动请求始终会在下一个（微）帧内调度 dTD 中所描述的传输。如果 ISO-dTD 在该帧后仍有效，则 ISO-dTD 将保持就绪状态，直到 DCD 执行或取消为止。

兼容 EHCI 的主机控制器使用定期帧列表来调度至同步端点的数据交换。设备模式的操作模型不会使用该类数据结构。用于控制 / 批量 / 中断端点的相同 dTD 也将用于同步端点。差别在于 dTD 的处理方式。

批量端点与 ISO 端点之间的第一项差异在于，启动 ISO 端点为延迟操作，因此只有在收到 SOF 后，端点才会启动。在 DCD 写入启动位后，启动位将照常清零，以向软件指示设备控制器已为传输完成启动 dTD。就设计内部而言，设备控制器硬件会掩码该启动开始，直到下一个帧边界为止。对于 DCD 来说，尽管该行为是隐藏的，但是会发生，因此设备控制器可以将 dTD 与特定 （微）帧匹配。

同步端点的另一项差异在于，事务必须在一个 （微）帧内全部完成。一旦 ISO 事务在一个（微）帧内开始，则在MULT事务发生或设备控制器找到履行状况时，便会停止相应的dTD。在状态字段中设置的事务错误位将指示履行错误状况。如果发生履行错误，则在传输未能全部完成的帧内，设备控制器将强制停止 ISO-dTD 并移至下一个 ISO-dTD。

值得注意的是，履行错误只会因部分完成的数据包导致。如果已启动的 ISO-dTD 未发生活动，则事务将无限期地启动。这意味着，一切取决于因主机未能移动数据而堆积的软件丢弃发送 ISO-dTD。最后一项差异在于， ISO 数据包采用数据级错误处理方式。当收到的数据包中发生 CRC 错误时，数据包不会象批量和控制端点一样停用。通过设置事务错误位即可记录 CRC，且数据将照常存储以供应用软件整理。

TX 数据包停用

- MULT 计数器为零。
- 履行错误 [事务错误位已设置]。
- 产生 # 个数据包 > 产生的 0 和 # 个数据包 < MULT。

注：对于 TX-ISO，可以加载 MULT 计数器，且 dTD 乘数替代字段中使用更小的值。如果乘数替代为零，则 MULT 计数器将初始化为 QH 中的乘数。

RX 数据包停用

- MULT 计数器为零。
- 收到非 MDATA 数据 PID。

  注：退出标准仅在 2.3 或更高版本的硬件中有效。在 2.3 版硬件之前，与 MULT 字段不完全匹配的任何 PID 序列将标记为因 PID 不匹配导致的事务错误或履行错误。
- 溢出错误：
  - 收到的数据包 > 最大数据包长度。 [缓冲区错误位被设置]。
  - 接受的数据包超出了 dTD 中分配的总字节数。 [缓冲区错误位被设置]。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- 履行错误 [已设置事务错误位]：
  产生 # 个数据包 > 产生的 0 和 # 个数据包 < MULT。
- CRC 错误 [已设置事务错误位]：

注：对于 ISO，如果 dTD 停用，将在下一个帧内启动下一个 dTD。对于连接的 （微）帧至（微）帧操作， DCD 应确保 dTD 链接列表先于设备控制器至少两 （微）帧。

23.10.9.1 同步管道的同步化

如果同步数据管道需要与主机同步，则可以使用 （微）帧数 （FRINDEX 寄存器）作为时标。若要在特定（微）帧数 [N] 时发生数据包传输，则 DCD 应在帧 N-1 期间中断 SOF。当 FRINDEX=N-1 时，DCD 必须写入启动位。设备控制器将在（微）帧 N-1 中启动同步端点，以便设备控制器将在 （微）帧 N 期间执行传输。

注：如果在 （微）帧 N-1 结束时启动端点，则将无法保证在 （微）帧 N 内传送。如果在收到数据包 N 的 SOF 之前，设备控制器没有足够的时间完成启动，则实际可能在 （微）帧 N+1 内进行传送。

23.10.9.2 同步端点母线响应矩阵

表 436. 同步端点母线响应矩阵

令牌类型	STALL	未启动	启动	下溢	溢出
设置	STALL	STALL	STALL	不适用	不适用
输入	NULL 数据包	NULL 数据包	发送	BS 错误	不适用
输出	忽略	忽略	接收	不适用	丢弃数据包
Ping	忽略	忽略	忽略	忽略	忽略
无效	忽略	忽略	忽略	忽略	忽略

[1] BS 错误 = 强制位填充错误

[2] NULL 数据包 = 零长度数据包。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

23.10.10 管理队列头

图 53. 端点队列头图

设备队列头 (dQH) 指向传输任务的链接列表，每个设备队列头都由设备传输描述符 (dTD) 描述。ENDPOINTLISTADDR 指向的存储器区域在顺序表中包含所有 dQH 的组（如 图 53 所示）。dQH 列表中的偶数元素用于接收端点 (OUT/SETUP)，而奇数元素则用于发送端点 (IN/INTERRUPT)。设备传输描述符将头链接到尾，从队列头开始，并在终止位结束。一旦 dTD 停用，便不再属于队列头链接列表的一部分。因此， dTD 被停用后，队列头中指针将不复存在，因此要求软件追踪所有的传输描述符 （参见第 23.10.11.1 节）。

除当前和后续指针及“数据包传输的操作模型”一节中检查的 dTD 重叠外，dQH 还包含相关端点的以下参数：乘数、最大数据包长度、设置时中断。下节介绍完全初始化包括这些字段的 dQH。

23.10.10.1 队列头初始化

每个活动的端点必须初始化一对设备队列头。若要初始化设备队列头：

- 按要求，依照 *USB 第 9 章*或应用特定协议对 wMaxPacketSize 字段进行写操作。
- 对于控制端点、批量端点和中断端点，向乘数字段写入 0。对于 ISO 端点，根据带宽和 *USB 第 9 章协议*的要求将乘数设为 1、 2 或 3。注：在 FS 模式下， ISO 端点的乘数字段只能设置为 1。
- 向下一个 dTD 终止位字段写入 1。
- 向状态字段中的有效位写入 0。
- 向状态字段中的停止位写入 0。

注：如果相关端点未启动且没有未处理的 dTD，则 DCD 仅需修改 dQH。

23.10.10.2 设置传输的操作模式

如 “控制端点操作模式”（第 23.10.8 节）所述，设置传输要求通过 DCD 进行特殊处理。设置传输不会使用 dTD，而是将来自设置数据包的传入数据存储在 dQH 的 8 字节缓冲区内。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

在收到设置数据包的通知后， DCD 应按照如下所述来处理设置传输：

1. 将设置缓冲区内容从 dQH - RX 复制到软件缓冲区。
2. 通过向 ENDPTSETUPSTAT 中的相应位写入 1 确认设置备份。

   注：必须在继续处理设置数据包前进行应答。

   注：确认以后，DCD 不得试图访问 dQH – RX 中的设置缓冲区。应仅检查本地软件副本。
3. 检查之前控制传输的挂起数据或状态 dTD，如果存在任何挂起数据或状态 dTD 则按照“刷新 / 取消启动端点”一节所述进行刷新。

   注：在之前的控制传输完成前，设备控制器可以收到设置数据包。必须刷新当前正在处理的控制数据包，才能完成新的控制数据包。
4. 按照 *USB 规范第 9 章*或应用程序特定协议的要求，解码设置数据包，然后准备数据阶段 [可选] 和状态阶段传输。

23.10.11 使用传输描述符管理传输

23.10.11.1 软件链接指针

DCD 软件需要保留每个相应队列头中 dTD 链接列表的头指针和尾指针。这点非常必要，因为dQH 只会保留当前工作 dTD 的指针和将执行的下一个 dTD 的指针。下节中关于管理 dTD 所述的操作将假设 DCD 可以参考 dTD 链接列表的头和尾。

注：为了节约存储器，dQH 结尾的保留字段可以用于存储头指针和尾指针，但仍由 DCD 负责保留指针。

图 54. 软件链接指针

23.10.11.2 建立传输描述符

在可以从链接列表执行传输之前，必须构建 dTD 以描述传输。使用以下流程来构建 dTD：

分配存储器的 8 个双字 dTD 模块，与 8 个双字边界对齐。示例：位地址 4:0 将等于 00000。

写入以下字段：

1. 将前 7 个双字初始化为 0。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

2. 将 Terminate 位设置为 1。
3. 使用传输长度填入总字节数。
4. 需要时，设置完成时中断。
5. 初始化状态字段，有效位设置为 1，其他状态位设置为 0。
6. 填入缓冲区指针页面 0 和当前偏移量，以指向数据缓冲区的开始处。
7. 将缓冲区指针页面 1 至页面 4 初始化为比每个上一个缓冲区指针大 1。

23.10.11.3 执行传输描述符

若要安全添加 dTD，则 DCD 必须遵循处理事件的该项流程，即在设备控制器到达 dTD 列表的同时，新的 dTD 将添加到列表的结尾。

确定链接列表是否为空：检查 DCD 驱动器，以确定管道是否为空（链接列表的内部表示应指出是否存在未处理的任何数据包）。

链接列表为空

1. 向 dQH 的下一个指针 AND dQH 终止位写入 0，作为单个双字操作。
2. 在 dQH 中将有效位和停止位清零 （以防从上一个错误中设置）。
3. 通过向 ENDPTPRIME 的正确位位置写入 1 来启动端点。

链接列表不为空

1. 将 dTD 添加到链接列表结尾处。
2. 如果 1 完成，读取 ENDPTPRIME 中的正确启动位。
3. 将 USBCMD 寄存器中的 ATDTW 位设置为 1。
4. 读取 ENDPTSTAT 中正确的状态位。（存储在临时变量中供随后使用）。
5. 读取 USBCMD 寄存器中的 ATDTW 位。
   - 如果为 0，则转到步骤 3。
   - 如果为 1，则继续步骤 6。
6. 向 USBCMD 寄存器中的 ATDTW 位写入 0。
7. 如果步骤 4 中读取的状态位（ENDPSTAT reg）指示端点启动已完成（相应的 ERBRx 或 ETBRx 为 1）：完成。
8. 如果步骤 4 中读取的状态位为 0，则转到链接列表为空：步骤 1。

23.10.11.4 传输完成

在 dTD 初始化且相关端点启动后，设备控制器将在主机启动的请求后执行传输。如果完成时已设置中断位，则将通过 USB 中断通知 DCD，或者 DCD 可以轮询端点完成寄存器以查找何时已执行 dTD。在执行 dTD 后， DCD 可以检查状态位，以确定成功还是失败。

注：单个端点完成通知中可以完成多个 dTD。清除通知后， DCD 必须搜索 dTD 链接列表，并停用所有已完成 （有效位清零）的 dTD。

通过读取已完成 dTD 的状态字段，DCD 可以确定是否成功完成传输。通过以下状态位组合可确定成功完成：

有效位 = 0

停止位 = 0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

事务错误 = 0

数据缓冲区错误 = 0

如果存在除上述组合以外的任何组合，则 DCD 必须采取适当的措施。传输失败机制在“设备错误矩阵”中显示 （参见表 437）。

除检查状态位外，DCD 必须读取传输字节字段以确定实际已传输的字节数。当传输完成时，已传输的总字节数将减少实际已传输的字节数。对于发送数据包，只有在实际字节数为 0 时，数据包才完成；而对于接收数据包，根据 USB 变量长度数据包协议，主机可能在传输中发送更少的字节数。

23.10.11.5 刷新 / 取消启动端点

在 USB 设备复位时或断开的控制传输期间，需要刷新 DCD 以取消启动另一个端点。应用特定要求还可能需要停止正在处理的传输。DCD 可以使用以下流程来停止正在处理的传输：

1. 向 ENDPTFLUSH 中的相应位写入 1。
2. 等待直到 ENDPTFLUSH 中的所有位都为 0。

   注：软件注意事项：该操作可能需要大量时间，具体取决于 USB 总线活动。最好不要在中断服务例程中使用该等待循环。
3. 读取 ENDPTSTAT 以确保对于所有命令刷新的端点，相应位现在都为 0。如果步骤 #2 完成后，相应位是 1，则刷新失败，如下所述：

   在极少数情况下，当使用 ENDPTFLUSH 命令刷新时，数据包正在特定端点中处理。应备妥预防措施以拒绝刷新，确保正在处理的数据包成功完成。 DCD 可能需要重复步骤 1-3，反复刷新任何刷新失败的端点，直到每个端点成功刷新为止。

23.10.11.6 设备错误矩阵

表 437 总结了不能由设备控制器自动处理的数据包错误。

可能发生以下错误：

溢出：收到的字节数超过最大数据包大小或缓冲区总长度。该错误还将设置 dQH 中的停止位，且如果端点的链接列表中有剩余的 dTD，则不会执行这些 dTD。

ISO 数据包错误：收到的 ISO 数据包发生 CRC 错误。无法保证内容正确。

ISO 履行错误：主机未能在指定 （微）帧内完成 dQH mult 字段中定义的数据包数量。对于调度数据传送， DCD 可能需要重新调整数据队列，因为履行错误将导致设备控制器停止管道上的数据传输 1 （微）帧。在死 （微）帧期间，设备控制器报告管道上的错误并启动下一帧。

表 437. 设备错误矩阵

错误	方向	数据包类型	数据缓冲区错误位	事务错误位
溢出	Rx	任意	1	0
ISO 数据包错误	Rx	ISO	0	1
ISO 履行错误	双向	ISO	0	1

23.10.12 服务中断

中断服务例程必须相应地考虑到所存在的高频操作、低频操作、错误操作和顺序。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

23.10.12.1 高频中断

尤其是高频中断应按以下顺序进行处理。最重要的列在最前面，因为 DCD 必须以尽可能最快的方式应答设置缓冲区。

表 438. 高频中断事件

执行顺序	中断	操作
1a	USB 中断：ENDPTSETUPSTATUS [1]	复制设置缓冲区的内容并确认设置数据包（如第 23.10.10 节中所示）。根据 *USB 2.0 第 9 章*或应用特定协议的要求处理设置数据包。
1b	USB 中断：ENDPTCOMPLETE[1]	按第 23.10.10 节中所示处理 dTD。
2	SOF 中断	应用认为必要的操作。该中断可能不会用于所有应用。

[1] 在中断服务例程期间，可能多个中断层叠调用中断服务例程 AND。

23.10.12.2 低频中断

低频事件包括以下中断。与高频中断相比，这些中断不常发生，因此可以按任意顺序处理。

表 439. 低频中断事件

中断	操作
端口更改	更改软件状态信息。
睡眠使能（挂起）	更改软件状态信息。必要时低功耗处理。
已收到复位	更改软件状态信息。中止挂起的传输。

23.10.12.3 错误中断

错误中断发生的频率最低，应放置在中断服务例程的最后。

表 440. 错误中断事件

中断	操作
USB 错误中断。	该错误属于冗余，因为它整合了 USB 中断和 dTD 中的错误状态。 DCD 在收到 USB 中断（带 ENDPTCOMPLETE）后，将检查 dTD 状态字段，从而更适当地处理数据包级错误。
系统错误	不可恢复的错误。内核立即复位；正在释放传输缓冲区及重新启动 DCD。

23.11 USB 功率优化

USB-HS 内核采用完全同步的静态设计。设计使用的功率依赖于内核的应用程序使用情况。应用所传输的数据越多，或发送的数据包越多，则其功耗也将越高。

可通过禁用 USB 时钟和高速 PHY 来控制 USB 功耗（参见第 23.1 节）。

通过与 USB 断开，设备可以自动挂起运行；或为响应挂起信号，USB 使其转为挂起状态。主机可以自动挂起运行，或可以向部分或整个 USB 发出命令跳变至挂起状态。

23.11.1 USB 功率状态

USB 提供了一种将部分或全部 USB 置于低功率挂起状态的机制。 USB 总线供电设备需要进入挂起状态，以对 USB 总线 3ms 不活动做出响应。在 USB-HS 模式下，将通过 PORTSC 寄存器中的跳变，通知内核软件挂起状态。可以选择产生中断，该中断由 USBINTR 控制寄

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

存器中的端口更改检测使能位控制。然后，软件可以在 7 ms 内将总线供电设备跳变至挂起状态。在挂起状态下， USB 设备最大的 USB 总线功率预算为 500μA。总的来说，为了实现该等级的功率节约，大部分的设备回路都需要关闭，或是时钟工作在极限低频。这可以通过挂起时钟来实现。

USB-HS 内核低功耗状态的实现取决于设备角色 （主机或外围设备）的用途、设备是否由总线供电以及内核的所选时钟架构。

USB 规范要求总线供电的外围设备支持低功耗挂起状态。自供电的外围设备和主机可根据其系统级别要求设置自身的电源管理策略。应谨慎考虑所选的时钟架构，因为它可决定在跳变至低功耗状态时，设计的哪些部分仍将处于活动状态。

在系统时钟挂起或设置的频率低于 USB-HS 内核的工作频率之前，内核必须从工作状态转变为低功耗状态。 USB-HS 内核采用的功耗策略设计可应对最具挑战的情况，即自供电设备由收发器时钟驱动。

23.11.2 设备功率状态

总线供电的外围设备必须按主机指示切换电源状态。可以选择性地执行自动指示的低功耗状态。

图 55. 设备功率状态图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

在工作状态下，收发器时钟和系统时钟都在运行。软件与主机断开并进入断开状态，即可自动启动低功耗模式。处于该状态后，软件可以设置挂起位以关闭收发器时钟，从而使系统进入断开 - 挂起状态。由于软件无法依赖时钟将挂起位清零，因此必须定义唤醒事件以将挂起位清零并使收发器时钟恢复。

设备还可能因主机发出的挂起命令而进入挂起模式。挂起信号以总线上 3 ms 空闲时间的方式在总线上发出。这将对软件产生挂起中断，这时软件必须准备进入挂起，然后设置挂起位。一旦设置挂起位，收发器时钟便可以关闭，而设备将处于挂起状态。设备可以通过两种方式退出挂起状态。

1. 如果使能远程唤醒，则可以定义唤醒事件以对挂起位清零。然后，软件在端口控制器中设置恢复位，并等待端口更改中断指示端口处于工作状态，便会启动恢复。
2. 如果主机在总线上发出恢复信号，则将对挂起位清零，并在恢复完成时产生端口更改中断。

23.11.3 主机功率状态

图 56. 主机 /OTG 电源状态图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

在工作状态下，如果主机收到低功耗请求，则必须在端口控制器中设置挂起位。该操作将使总线处于空闲状态、阻断通过端口的所有流量以及关闭收发器时钟。主机控制器可以通过两种方式退出挂起状态。如果已使能远程唤醒，则总线上的 K 状态将开启收发器时钟并产生中断。然后，软件将必须等待 20 ms，恢复才会完成，端口才会返回活动状态。另外，外部事件可以将挂起位清零并使收发器时钟重新开始运行。然后，软件在端口控制器中设置恢复位，即可启动恢复；或在端口控制器中设置复位位，即可强制重新连接。

如果所有设备已与主机断开，则通过软件设置挂起位，主机便可以进入低功耗模式。在断开 - 挂起状态下，连接事件将启动收发器时钟并中断软件。然后，软件将需要设置复位位以启动连接过程。

23.11.4 Susp_CTRL 模块

SUSP_CTRL 模块执行 USB-OTG 的电源管理逻辑。该模块可控制收发器的挂起输入。如果该挂起信号使收发器进入挂起模式，则 30 MHz 和 60 MHz 时钟将关闭。

之前连接高电平且防止收发器进入全挂起模式的收发器挂起控制输入 (otg_on) 已连接到 <tbd>。该位默认情况下为低电平，且只有在 OTG 主机模式运行时，才需要设为高电平。

在挂起模式下，收发器将发出输出信号，指示产生 480 MHz 时钟的 PLL 可以关闭。

SUSP_CTRL 模块还会产生输出信号，指示是否需要 AHB 时钟。如果为“0”，则 AHB 时钟可以关闭或降低频率以便降低功耗。

在以下情况下，内核将进入低功耗状态：

- 软件设置 PORTSC.PHCD 位。

如果在主机模式下运行，内核将在以下其中一种情况下退出低功耗状态：

- 软件将 PORTSC.PHCD 位清零
- 连接设备并设置 PORTSC.WKCN 位
- 断开设备并设置 PORTSC.WKDC 位
- 发生过流状况并设置 PORTSC.WKOC 位
- 发生从连接的设备远程唤醒 （USB 总线处于挂起状态）
- vbusvalid 发生更改 （= 超过 4.4 V 时的 VBUS 阈值）
- bvalid 发生更改 （= 超过 4.0 V 时的 VBUS 阈值）。

如果在设备模式下运行，内核将在以下其中一种情况下退出低功耗状态：

- 软件将 PORTSC.PHCD 位清零。
- USB 数据线路 （dp/dm）发生更改。
- vbusvalid 发生更改 （= 超过 4.4 V 时的 VBUS 阈值）。
- bvalid 发生更改 （= 超过 4.0 V 时的 VBUS 阈值）。

来自收发器的 vbusvalid 和 bvalid 信号不会在 SUSP_CTRL 模块中过滤。这些信号的任何更改将导致唤醒事件。

输入信号 “host_wakeup_n”和 “dev_wakeup_n”是额外的外部唤醒信号 （分别针对主机模式和设备模式）。但是，所有 USB 相关的唤醒事件检测已在 SUSP_CTRL 模式中处理。因此，在正常情况下，这些信号可能连接高电平 （即无效）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

24.1 本章导读

USB1 主机 / 设备控制器在器件 LPC4350 和 LPC4330 上可用。

24.2 基本配置

USB1 控制器配置如下：

- 参见表 441 以了解时钟与电源控制。
- USB1 由 USB1_RST 复位 （复位 # 18）。
- USB1 <tbd> 中断连接到 NVIC 中的中断槽 # 9。USB 唤醒中断连接至事件路由器中的 # 10 插槽。
- 必须将 SFSUSB 寄存器的 USB_ESEA 位设为 1，以便 USB1 正常工作（参见表 126）。

表 441. USB1 时钟和电源控制

	基本时钟	分支时钟	工作频率	备注
USB1 时钟	BASE_USB1_CLK	CLK_USB1	60 MHz	使用 PLL1。USB1 以低速、全速模式运转时，CLK_USB1 必须为60 MHz。在高速模式下，时钟由 ULPI PHY 提供。
USB1 寄存器接口时钟	BASE_M4_CLK	CLK_M4_USB1	高达 204 MHz	

24.2.1 全速模式，无外部 PHY

在全速模式下，利用 CLK_USB1 为 USB1 接口生成时钟。

24.2.2 高速模式，有 ULPI 接口

在高速模式下，外部 PHY 可为 USB1 接口生成时钟，但必须通过系统配置模块中各自的引脚配置寄存器使能 PC_0 或 P8_8 上的 USB1_ULPI_CLK。必须禁用 USB1 分支时钟 CLK_USB1。

24.3 功能要点

- 如果连接到外部 ULPI PHY，则支持符合 USB 标准的所有高速外围设备。
- 支持符合 USB 标准的所有全速外围设备。
- 符合*通用串行总线规范 2.0*。
- 符合*增强型主机控制器接口规范*。
- 支持 USB 2.0 自动模式发现。
- 支持三个逻辑端点加一个控制端点组成总共 8 个物理端点。
- 这个模块有其自己集成的 DMA 引擎。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

24.4 简介

USB1 控制器是一个嵌入式应用外设，包含数字电路，可提供 USB2.0 OTG 功能。

USB2.0 为外设与主机之间提供即插即用连接，支持三种数据传输速度：高速 480Mbps、全速 12Mbps 和低速 1.5Mbps。许多便携式设备的优势在于，可在不受主机电脑干预的情况下，通过 USB 接口来彼此交换数据。在 USB 上添加 OTG 功能，则可在实现这种功能的同时保留标准 USB 协议的优势。

要支持高速数据传输速率，就需要安装一个外部 USB HS OTG PHY，通过 ULPI 接口连接到 USB 控制器。片内全速 PHY 可支持全速或低速数据传输。

图 57. 带 ULPI 的 USB1 框图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 58. 带内部全速 PHY 的 USB1 框图

24.5 引脚说明

表 442. USB1 引脚描述

函数名称	方向	描述
USB1_DP	I/O	USB1 双向 D+ 线。D+ 线具有一个内部 1.5 kΩ 上拉电阻。软件设置 USBCMD 寄存器的 RS 位（位 0）时，使能此上拉电阻。USB1 控制器检查 USB1_VBUS 引脚是否被拉高（无电压监控）。对于将 USB1_VBUS 引脚用作 GPIO 的应用，软件可通过 SFSUSB 寄存器 (0x4008 6C80) 中的位 5 监控 VBUS。
USB1_DM	I/O	USB1 双向 D– 线路。D- 线具有一个内部 1.5 kΩ 上拉电阻。软件设置 USBCMD 寄存器的 RS 位（位 0）时，使能此上拉电阻。USB1 控制器检查 USB1_VBUS 引脚是否被拉高（无电压监控）。对于将 USB1_VBUS 引脚用作 GPIO 的应用，软件可通过 SFSUSB 寄存器 (0x4008 6C80) 中的位 5 监控 VBUS。
USB1_VBUS	I	VBUS 引脚（USB 电缆上电）。如果 USB1_VBUS 功能未连接至引脚 P2_5，使用 SFSUSB 寄存器表 126 中的 USB_VBUS 位向 USB1 控制器指示 VBUS 状态。
USB1_PPWR	O	VBUS 驱动信号（发送至外部充电泵或电源管理单元）；指示 VBUS 必须被驱动（有效高电平）。 复位时，添加一个下拉电阻以禁用电源开关。和其他恩智浦 LPC 器件使用的 USB_PPWR 相比，该信号具有相反的极性。
USB1_IND0	O	端口 LED 指示灯控制输出 0。
USB1_IND1	O	端口 LED 指示灯控制输出 1。
USB1_PWR_FAULT	I	指示过流状况的端口电源故障信号；此信号监控 USB 总线上的过流状况（外部电路需要检测过流状况）。
ULPI 引脚		
ULPI_DATA[7:0]	I/O	在上升时钟沿上计时的 ULPI 连接 8 位双向数据总线
ULPI_STP	O	ULPI 链接 STP 信号。断言结束或中断到 PHY 的传输。
ULPI_NXT	I	ULPI 链接 NXT 信号。来自 PHY 的数据流控制信号。
ULPI_DIR	I	ULPI 链接 DIR 信号。控制数据总线方向。
ULPI_CLK	I	ULPI 链接 CLK 信号。 PHY 生成的 60 MHz 时钟。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

24.6 寄存器描述

注：如果要通过片内全速 PHY 实现全速操作，需配置 PHY 管脚。有关这些管脚的配置，请参见表 126。

注：如果要利用通过 ULPI 接口连接的外部 PHY 实现操作，需在 PORTSC1 寄存器的 PTS 各位选中该接口（第 24.6.15 节）。

表 443. 寄存器访问缩写

缩写	描述
R/W	读 / 写
R/WC	读 / 写 1 清零
R/WO	读 / 写一次
RO	只读
WO	只写

表 444. 寄存器简介：USB1 主机 / 设备控制器 （寄存器基址 0x4000 7000）

名称	访问类型	地址偏移	描述	复位值	参考
-	-	0x000 - 0x0FF	保留		
设备 / 主机功能寄存器					
CAPLENGTH	RO	0x100	功能寄存器长度	0x0001 0040	表 445
HCSPARAMS	RO	0x104	主机控制器结构参数	0x0001 0011	表 446
HCCPARAMS	RO	0x108	主机控制器功能参数	0x0000 0005	表 447
DCIVERSION	RO	0x120	设备接口版本号	0x0000 0001	表 448
DCCPARAMS	RO	0x124	设备控制器功能参数	0x0000 0184	表 449
-	-	0x128 - 0x13C	保留	-	-
设备 / 主机操作寄存器					
USBCMD_D	R/W	0x140	USB 命令 （设备模式）	0x0004 0000	表 450
USBCMD_H	R/W	0x140	USB 命令 （主机模式）	0x0004 00B0	表 451
USBSTS_D	R/W	0x144	USB 状态 （设备模式）	0x0000 0000	表 453
USBSTS_H	R/W	0x144	USB 状态 （主机模式）	0x0000 1000	表 454
USBINTR_D	R/W	0x148	USB 中断使能 （设备模式）	0x0000 0000	表 455
USBINTR_H	R/W	0x148	USB 中断使能 （主机模式）	0x0000 0000	表 456
FRINDEX_D	RO	0x14C	USB 帧索引 （设备模式）	0x0000 0000	表 457
FRINDEX_H	R/W	0x14C	USB 帧索引 （主机模式）	0x0000 0000	表 458
-	-	0x150	保留		-
DEVICEADDR	R/W	0x154	USB 设备地址	0x0000 0000	表 460
PERIODICLISTBASE	R/W	0x154	帧列表基址	0x0000 0000	表 461
ENDPOINTLISTADDR	R/W	0x158	存储器中端点列表地址	0x0000 0000	表 462
ASYNCLISTADDR	R/W	0x158	异步列表地址	0x0000 0000	表 463
TTCTRL	R/W	0x15C	嵌入式 TT 的异步缓冲状态（主机模式）	0x0000 0000	表 464
BURSTSIZE	R/W	0x160	可编程的突发流量	0x0000 0000	表 465
TXFILLTUNING	R/W	0x164	主机发送预缓冲数据包调谐（主机模式）	0x0000 0000	表 466

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 444. 寄存器简介：USB1 主机 / 设备控制器 （寄存器基址 0x4000 7000） *（续）*

名称	访问类型	地址偏移	描述	复位值	参考
-	-	0x168 - 0x16C	保留	-	-
ULPIVIEWPORT	R/W	0x170	ULPI 视口	0x0000 0000	表 467
BINTERVAL	R/W	0x174	虚拟帧长度	0x0000 0000	表 468
ENDPTNAK	R/W	0x178	端点 NAK （设备模式）	0x0000 0000	表 469
ENDPTNAKEN	R/W	0x17C	端点 NAK 使能 （设备模式）	0x0000 0000	表 470
-	-	0x180	-	-	-
PORTSC1_D	R/W	0x184	端口 1 状态 / 控制 （设备模式）	0x0000 0000	表 471
PORTSC1_H	R/W	0x184	端口 1 状态 / 控制 （主机模式）	0x0000 0000	表 472
-	-	0x188 - 0x1A0	-	-	-
-	-	0x1A4	-	-	-
USBMODE_D	R/W	0x1A8	USB 模式 （设备模式）	0x0000 0000	表 474
USBMODE_H	R/W	0x1A8	USB 模式 （主机模式）	0x0000 0000	表 475
设备端点寄存器					
ENDPTSETUPSTAT	R/W	0x1AC	端点设置状态	0x0000 0000	表 476
ENDPTPRIME	R/W	0x1B0	端点初始化	0x0000 0000	表 477
ENDPTFLUSH	R/W	0x1B4	端点取消初始化	0x0000 0000	表 478
ENDPTSTAT	RO	0x1B8	端点状态	0x0000 0000	表 479
ENDPTCOMPLETE	R/W	0x1BC	端点完成	0x0000 0000	表 480
ENDPTCTRL0	R/W	0x1C0	端点控制 0	0x0000 0000	表 481
ENDPTCTRL1	R/W	0x1C4	端点控制 1	0x0000 0000	表 482
ENDPTCTRL2	R/W	0x1C8	端点控制 2	0x0000 0000	表 482
ENDPTCTRL3	R/W	0x1CC	端点控制 3	0x0000 0000	表 482

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

24.6.1 设备 / 主机功能寄存器

表 445. CAPLENGTH 寄存器 （CAPLENGTH – 地址 0x4000 7100）位描述

位	符号	描述	复位值	访问类型
7:0	CAPLENGTH	指示添加到操作寄存器开始处寄存器基址的偏移量。	0x40	RO
23:8	HCIVERSION	该主机控制器支持的EHCI修订版本号的BCD编码。	0x100	RO
31:24	-	这些位将保留且应设为 0。	-	-

表 446. HCSPARAMS 寄存器 （HCSPARAMS – 地址 0x4000 7104） 位描述

位	符号	描述	复位值	访问类型
3:0	N_PORTS	下行端口数量。该字段指定该主机控制器上提供的物理下行端口数量。	0x1	RO
4	PPC	端口电源控制。该字段指示主机控制器实施是否包括端口电源控制。	0x1	RO
7:5	-	这些位将保留且应设为 0。	-	-
11:8	N_PCC	每个辅助控制器的端口数量。该字段指示每个内部辅助控制器支持的端口数量。	0x0	RO
15:12	N_CC	辅助控制器的数量。该字段指示与该 USB2.0 主机控制器关联的辅助控制器的数量。	0x0	RO
16	PI	端口指示器。该位指示端口是否支持端口指示器控制。	0x1	RO
19:17	-	这些位将保留且应设为 0。	-	-
23:20	N_PTT	每个事务转译器的端口数量。该字段指示 USB2.0 主机控制器内分配给每个事务转译器的端口数量。	0x0	RO
27:24	N_TT	事务转译器的数量。该字段指示与USB2.0主机控制器关联的嵌入式事务转译器的数量。	0x0	RO
31:28	-	这些位将保留且应设为 0。	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 447. HCCPARAMS 寄存器 （HCCPARAMS - 地址 0x4000 7108）位描述

位	符号	描述	复位值	访问类型
0	ADC	64 位寻址能力。如果为零，则不支持 64 位寻址能力。	0	RO
1	PFL	可编程的帧列表标志。如果设为 1，则系统软件可以指定和使用小型帧列表，并通过 USBCMD 寄存器帧列表大小字段配置主机控制器。帧列表必须始终在 4K 边界上对齐。该要求可确保帧列表始终在物理上连通。	1	RO
2	ASP	异步调度表暂停功能。该位设为 1 时，主机控制器支持异步调度表中的高速队列头暂停功能。通过使用 USBCMD 寄存器的异步调度表暂停模式使能和异步调度表暂停模式计数字段，可以禁用或使能该功能和设置到特定级别。	1	RO
7:4	IST	同步调度阈值。该字段指示软件可以更新的、较执行主机控制器当前位置更为可靠的同步调度。	0	RO
15:8	EECP	EHCI 扩展能力指针。该可选字段指示是否存在能力列表。	0	RO
31:16	-	这些位将保留且应设为 0。	-	-

表 448. DCIVERSION 寄存器 （DCIVERSION - 地址 0x4000 7120）位描述

位	符号	描述	复位值	访问类型
15:0	DCIVERSION	设备控制器接口符合此寄存器中接口版本号的双字节 BCD 编码。	0x1	RO
31:16	-	这些位将保留且应设为 0。	-	-

表 449. DCCPARAMS （地址 0x4000 7124）

位	符号	描述	复位值	访问类型
4:0	DEN	设备端点号。	0x4	RO
6:5	-	这些位将保留且应设为 0。	-	-
7	DC	正常设备。	0x1	RO
8	HC	正常主机。	0x1	RO
31:9	-	这些位将保留且应设为 0。	-	-

24.6.2 USB 指令寄存器 (USBCMD)

主机 / 设备控制器执行该寄存器指示的命令。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

24.6.2.1 设备模式

表 450. 设备模式中的 USB 指令寄存器 （USBCMD_D - 地址 0x4000 7140）位描述

位	符号	值	描述	复位值	访问类型
0	RS		运行 / 停止。	0	R/W
		0	向该位写入 0 将导致分离事件。		
		1	向该位写入 1 将导致设备控制器在 USB_DP 上使能上拉，并启动连接事件。该控制位不会直接连接到上拉启动，因为跳变至高速模式后将禁用上拉。在设备控制器正确初始化之前，软件应使用该位防止发生连接事件。		
1	RST		控制器复位。 软件使用该位来复位控制器。复位过程完成时，该位将由主机 / 设备控制器设为 0。软件无法通过向该寄存器写入 0 来提前终止复位过程。	0	R/W
		0	当复位过程完成时，由硬件设为 0。		
		1	当软件向该位写入 1 时，设备控制器会将其内部管道、定时器、计数器和状态机等复位为初始值。建议在设备处于连接状态时，不要向该位写入 1，因为无法定义对连接主机的影响。为了确保在启动设备控制器复位前，设备未处于连接状态下，应刷新所有已启动的端点，且 USBCMD Run/Stop 位应设为 0。		
3:2	-		不适用于设备模式。	0	-
4	-		不适用于设备模式。	0	-
5	-		不适用于设备模式。	0	-
6	-		不适用于设备模式。在选择设备模式后向该位写入1，将产生未定义的结果。	-	-
7	-	-	保留。这些位应设为 0。	-	-
9:8	-	-	不适用于设备模式。	-	-
10	-		保留。这些位应设为 0。	0	-
11	-	-	不适用于设备模式。		-
12	-		保留。这些位应设为 0。	0	-
13	SUTW		设置绊网。 在处理设置数据包期间，此位用作信号灯，以确保 DCD 从 QH 获得的 8 字节设置数据负载未损坏。如果设置锁定模式关闭（见 USBMODE 寄存器），在 DCD 从 QH 复制上一个设置数据包的设置数据负载时，新的设置数据也同时到达，则会存在危险。该位的设置和清零由软件来进行，如果存在危险，则由硬件来清零。（参见第 23.10 节）。	0	R/W
14	ATDTW		添加 dTD 绊网。 该位用作信号量，可确保新的 dTD 正确添加到工作 （已启动）端点的链接列表。该位在添加新 dTD 的过程中由软件来设置和清零。另请参见第 23.10 节。 当其状态机处于危险区域时，将 dTD 添加到已启动的端点可能无法识别，该位还将由硬件清零。	0	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 450. 设备模式中的 USB 指令寄存器 （USBCMD_D - 地址 0x4000 7140）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
15	FS2		不适用于设备模式。	-	-
23:16	ITC		中断阈值控制。 系统软件使用该字段来设置主机 / 设备控制器发出中断的最大速率。ITC 包含最大的中断时间间隔 （以微帧为单位）。有效值如下所示。保留所有其它值。 0x0 = 立即 （无阈值）。 0x1 = 1 微帧。 0x2 = 2 微帧。 0x8 = 8 微帧。 0x10 = 16 微帧。 0x20 = 32 微帧。 0x40 = 64 微帧。	0x8	R/W
31:24	-		保留。	0	

24.6.2.2 主机模式

表 451. 主机模式中的 USB 指令寄存器 （USBCMD_H - 地址 0x4000 7140）位描述

位	符号	值	描述	复位值	访问类型
0	RS		运行 / 停止。	0	R/W
		0	如果将该位设为 0，则主机控制器会完成 USB 上的当前事务，然后停止。状态寄存器中的 HC Halted 位指示主机控制器何时完成事务以及何时进入停止状态。除非主机控制器处于停止状态（即 USBSTS 寄存器中的 HCHalted 为 1），否则软件不会向该字段写入 1。		
		1	在设为 1 时，主机控制器会继续执行调度。只要将该位设为 1，主机控制器就会继续执行。		
1	RST		控制器复位。 软件使用此位复位控制器。复位过程完成时，该位将由主机 / 设备控制器设为 0。软件无法通过向该寄存器写入 0 来提前终止复位过程。	0	R/W
		0	复位过程完成时，该位将由硬件设为 0。		
		1	当软件向该位写入 1 时，主机控制器会将其内部管道、定时器、计数器和状态机等复位为初始值。当前正在 USB 上处理的任何事务将立即终止。无法在下行端口驱动 USB 复位。当 USBSTS 寄存器中的 HCHalted 位为 0 时，软件不会将该位设为 1。尝试复位主动运行的主机控制器将导致未定义的行为。		
2	FS0		帧列表大小位的位 0。参见表 452。 该字段指定帧列表大小，而帧列表控制帧索引寄存器中用于帧列表当前索引的位。需要注意的是，该字段由 USBCMD 位 15、3 和 2 组成。	0	
3	FS1		帧列表大小位的位 1。见表 452。	0	
4	PSE		该位控制主机控制器是否跳过处理定期调度。	0	R/W
		0	不处理定期调度。		
		1	使用 PERIODICLISTBASE 寄存器访问定期调度。		
5	ASE		该位控制主机控制器是否跳过处理异步调度。	0	R/W
		0	不处理异步调度。		
		1	使用 ASYNCLISTADDR 访问异步调度。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 451. 主机模式中的 USB 指令寄存器 （USBCMD_H - 地址 0x4000 7140）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
6	IAA		该位由软件用作提示信息，告知主机控制器在下次超前进行异步调度时发出中断。	0	R/W
		0	主机控制器将 USBSTS 寄存器中的异步推进时中断状态位设为 1 后，便会将该位设为 0。		
		1	软件必须向该位写入 1，以启动提示信息。 当主机控制器退出所有相关缓存的调度状态时，便会设置 USBSTS 寄存器中的异步推进时中断状态位。如果 USBINTR 寄存器中的异步推进使能时中断位为 1，则主机控制器将在下一个中断阈值中确认中断。 在异步调度不工作时，软件不会向该位写入 1。否则，将会产生未定义的结果。		
7	-	-	保留。	0	
9:8	ASP1_0		异步调度表暂停模式。 包含在继续异步调度表前主机控制器可以从异步调度表高速队列头执行的连续事务数。有效值为 0x1 至 0x3。 **注：**当暂停模式使能位为 1 时，软件不得向此位写入 00，因为这将导致未定义的行为。	11	R/W
10	-	-	保留。	0	-
11	ASPE		异步调度表暂停模式使能。	1	R/W
		0	暂停模式禁用。		
		1	暂停模式使能。		
12	-	-	保留。	0	-
13	-	-	不适用于主机模式。		-
14	-	-	保留。	0	-
15	FS2		帧列表大小位的位 2。参见表 452。	0	-
23:16	ITC		中断阈值控制。 系统软件使用该字段来设置主机 / 设备控制器发出中断的最大速率。ITC 包含最大的中断时间间隔（以微帧为单位）。有效值如下所示。保留所有其它值。 0x0 = 立即 （无阈值）。 0x1 = 1 微帧。 0x2 = 2 微帧。 0x8 = 8 微帧。 0x10 = 16 微帧。 0x20 = 32 微帧。 0x40 = 64 微帧。	0x8	R/W
31:24	-		保留。	0	

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 452. 帧列表大小值

USBCMD 第 15 位	USBCMD 第 3 位	USBCMD 第 2 位	帧列表大小
0	0	0	1024 元素 （4096 字节） - 默认值
0	0	1	512 元素 （2048 字节）
0	1	0	256 元素 （1024 字节）
0	1	1	128 元素 （512 字节）
1	0	0	64 元素 （256 字节）
1	0	1	32 元素 （128 字节）
1	1	0	16 元素 （64 字节）
1	1	1	8 元素 （32 字节）

24.6.3 USB 状态寄存器 (USBSTS)

该寄存器指示主机 / 设备控制器的各种状态及任何挂起的中断。软件向该寄存器写入 1，即可将该位设为 0。

注：该寄存器不会指示因串行总线上的事务所产生的状态。

24.6.3.1 设备模式

表 453. 设备模式的 USB 状态寄存器 （USBSTS_D - 地址 0x4000 7144）寄存器位描述

位	符号	值	描述	复位值	访问类型
0	UI		USB 中断。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	在传输描述符 （TD）设置了完成时中断 （IOC）位的情况下，如果中断是由 USB 事务完成所致，则该位将由主机 / 设备控制器进行设置。 当检测到短数据包时，该位也将由主机 / 设备控制器设置。短数据包指收到的实际字节数少于预期字节数。		
1	UEI		USB 错误中断。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	如果 USB 事务的完成导致错误条件的出现，则该位将由主机 / 设备控制器设置。如果发生错误中断的 TD 也设置了其完成时中断 （IOC）位，则该位将和 USBINT 位一并设置。设备控制器仅检测恢复信号 （参见第 23.10.11.6 节）。		
2	PCI		端口更改检测。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	当端口控制器进入全速或高速工作状态时，设备控制器会将该位设为1。当端口控制器因复位或挂起事件而退出全速或高速工作状态时，通知机制分别为收到 USB 复位位（URI）和 DCSuspend 位（SLI）。		
3	-		不适用于设备模式。		
4	-	0	保留。		
5	-		不适用于设备模式。	0	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 453. 设备模式的 USB 状态寄存器 （USBSTS_D - 地址 0x4000 7144）寄存器位描述 *（续）*

位	符号	值	描述	复位值	访问类型
6	URI		收到 USB 复位。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	当设备控制器检测到USB复位并进入默认状态时，该位将设为 1。		
7	SRI		收到 SOF。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	当设备控制器检测到（微）帧起始时，该位将设为 1。当帧起始 （SOF）太迟时，设备控制器将自动设置该位，以指示预期的 SOF。因此，在设备 FS 模式下，将大约每 1 ms 对该位进行设置，而在 HS 模式下，则将每 125 µs 对其进行设置，并将与收到的实际 SOF 同步。由于设备控制器在连接前会初始化为 FS，因此该位在预载期间，将按 1ms 的时间间隔设置，以便连接和线性调频。		
8	SLI		DCSuspend	0	R/WC
		0	设备控制器退出挂起状态时，便会将该位清零。软件向该位写入 1，即可将该位清零。		
		1	当设备控制器从活动状态进入挂起状态时，该位将设为 1。		
11:9	-	-	保留。软件只会向保留位写入 0。	0	
12	-	-	不适用于设备模式。	0	
13	-	-	不适用于设备模式。	0	
14	-	-	不适用于设备模式。	0	
15	-	-	不适用于设备模式。	0	
16	NAKI		NAK 中断位。	0	RO
		0	如果所有使能的 TX/RX 端点 NAK 位都被清零，则该位将由硬件自动清零。		
		1	在特定端点的 TX/RX 端点 NAK 位和相应的 TX/RX 端点 NAK 使能位设置时，将由硬件设置该位。		
17	-	-	保留。软件只会向保留位写入 0。	0	-
18	-		不适用于设备模式。	0	-
19	-		不适用于设备模式。	0	-
31:20	-	-	保留。软件只会向保留位写入 0。		-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

24.6.3.2 主机模式

表 454. 主机模式的 USB 状态寄存器 （USBSTS_H - 地址 0x4000 7144）寄存器位描述

位	符号	值	描述	复位值	访问类型
0	UI		USB 中断 （USBINT）。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	在传输描述符 （TD）设置了完成时中断 （IOC）位的情况下，如果中断是由 USB 事务完成所致，则该位将由主机 / 设备控制器进行设置。 当检测到短数据包时，该位也将由主机 / 设备控制器设置。短数据包指收到的实际字节数少于预期字节数。		
1	UEI		USB 错误中断 （USBERRINT）。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	如果 USB 事务的完成导致错误条件的出现，则该位将由主机 / 设备控制器设置。如果发生错误中断的 TD 也设置了其完成时中断 （IOC）位，则该位将和 USBINT 位一并设置。		
2	PCI		端口更改检测。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	当任何端口发生连接状态、发生端口使能 / 禁用更改或因挂起端口上 J-K 跳变而设置强制端口恢复位时，主机控制器会将该位设为 1。		
3	FRI		帧列表滚动更新。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	当帧列表索引从其最大值滚动更新为 0 时，主机控制器会将该位设为 1。滚动更新的确切数值取决于帧列表大小。例如，如果帧列表大小（在 USBCMD 寄存器的帧列表大小字段中编程）为 1024，则在每次 FRINDEX [13] 切换时，帧索引寄存器会滚动更新。同理，如果大小为 512，则每次 FRINDEX 位 12 切换时，主机控制器将该位设为 1 （参见第 24.6.5 节）。		
4	-	0	保留。		
5	AAI		异步推进时中断。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	系统软件可在下次主机控制器超前进行异步调度时，通过向 USBCMD 寄存器的异步预先提示信息位写入 1，强制主机控制器发出中断。该状态位指示确认该中断源。		
6	-	-	主机控制器未使用。	0	R/WC
7	SRI		收到 SOF。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	在主机模式下，将每 125 µs 对该位进行设置，并且该位可以作为时间基准以供主机控制驱动器使用。		
8	SLI	-	主机控制器未使用。	-	-
11:9	-	-	保留。		
12	HCH		HCHalted	1	RO
		0	USBCMD 中的 RS 位设为 0。由主机控制器设置。		
		1	由于运行 / 停止位设为 0，主机控制器停止执行后，将通过软件或主机控制器硬件将该位设为 1 （例如，由于内部错误）。		
13	RCL		收回。	0	RO
		0	未检测到空的异步调度。		
		1	检测到空的异步调度。由主机控制器设置。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 454. 主机模式的 USB 状态寄存器 （USBSTS_H - 地址 0x4000 7144）寄存器位描述 *（续）*

位	符号	值	描述	复位值	访问类型
14	PS		定期调度状态。 该位报告定期调度的当前实际状态。当软件切换 USBCMD 寄存器中的定期调度使能位时，主机控制器无需立即禁用或使能定期调度。当该位和定期调度使能位具有相同值时，定期调度将使能 （如果两者都为 1）或禁用 （如果两者都为 0）。	0	RO
		0	定期调度状态禁用。		
		1	定期调度状态使能。		
15	AS		异步调度状态。 该位报告异步调度的当前实际状态。当软件切换 USBCMD 寄存器中的异步调度使能位时，主机控制器无需立即禁用或使能异步调度。当该位和异步调度使能位具有相同值时，异步调度将使能 （如果两者都为 1）或禁用 （如果两者都为 0）。	0	
		0	异步调度状态禁用。		
		1	异步调度状态使能。		
16	-		不适用于主机模式。	0	-
17	-		保留。		
18	UAI		USB 主机异步中断 （USBHSTASYNCINT）。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	在传输描述符 (TD) 设置了完成时中断 (IOC) 位的情况下，**且** TD 来自异步调度表，如果中断是由 USB 事务完成所致，则该位将由主机控制器进行设置。当检测到短数据包**且**该数据包位于异步调度中时，该位也将由主机设置。短数据包指收到的实际字节数少于预期字节数。		
19	UPI		USB 主机定期中断 （USBHSTPERINT）。	0	R/WC
		0	软件向该位写入 1，即可将该位清零。		
		1	在传输描述符 (TD) 设置了完成时中断 (IOC) 位的情况下，**且** TD 来自周期性调度表，如果中断是由 USB 事务完成所致，则该位将由主机控制器进行设置。当检测到短数据包**且**该数据包位于定期调度中时，该位也将由主机控制器设置。短数据包指收到的实际字节数少于预期字节数。		
31:20	-		保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

24.6.4 USB 中断寄存器 (USBINTR)

该寄存器可使能软件中断。当该位设置且相应中断处于工作状态时，便产生中断。 USB 状态寄存器 （USBSTS）即使通过 USBINTR 寄存器禁用，仍会显示中断源，以便允许软件轮询中断事件。所有中断必须由软件将 USBSTS 寄存器中相应的位清零 （即写入 1）来进行确认。

24.6.4.1 设备模式

表 455. 设备模式中的 USB 中断寄存器 （USBINTR_D - 地址 0x4000 7148）位描述

位	符号	描述	复位值	访问类型
0	UE	USB 中断使能。 当该位为 1，且 USBSTS 寄存器中的 USBINT 位为 1 时，主机 / 设备控制器将在下一个中断阈值中发出中断。该中断由软件将 USBSTS 中的 USBINT 位清零来进行确认。	0	R/W
1	UEE	USB 错误中断使能。 当该位为 1，且 USBSTS 寄存器中的 USBERRINT 位为 1 时，主机 / 设备控制器将在下一个中断阈值中发出中断。该中断由软件将 USBSTS 寄存器中的 USBERRINT 位清零来进行确认。	0	R/W
2	PCE	端口更改检测使能。 当该位为 1，且 USBSTS 寄存器中的端口更改检测位为 1 时，主机 / 设备控制器将发出中断。该中断由软件将 USBSTS 中的端口更改检测位清零来进行确认。	0	R/W
3	-	设备控制器未使用。		
4	-	保留。	0	-
5	-	设备控制器未使用。		
6	URE	USB 复位使能。 当该位为 1，且 USBSTS 寄存器中的收到 USB 复位位为 1 时，主机 / 设备控制器将发出中断。该中断由软件将收到 USB 复位位清零来进行确认。	0	R/W
7	SRE	收到 SOF 使能。 当该位为 1，且 USBSTS 寄存器中的收到 SOF 位为 1 时，主机 / 设备控制器将发出中断。该中断由软件将收到 SOF 位清零来进行确认。	0	R/W
8	SLE	睡眠使能。 当该位为 1 时，且 USBSTS 寄存器中的 DCSuspend 位切换时，设备控制器将发出中断。该中断由软件向 DCSuspend 位写入 1 来进行确认。	0	R/W
15:9	-	保留。	-	-
16	NAKE	NAK 中断使能。 如果要使能 NAK 中断位的硬件中断，则该位将由软件设置。如果已经设置了该位和相应的 NAK 中断，则将产生硬件中断。	0	R/W
17	-	保留。		
18	UAIE	设备控制器未使用。		
19	UPIA	设备控制器未使用。		
31:20	-	保留。		

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

24.6.4.2 主机模式

表 456. 主机模式中的 USB 中断寄存器 （USBINTR_H - 地址 0x4000 7148）位描述

位	符号	描述	访问类型	复位值
0	UE	USB 中断使能。 当该位为 1，且 USBSTS 寄存器中的 USBINT 位为 1 时，主机 / 设备控制器将在下一个中断阈值中发出中断。该中断由软件将 USBSTS 中的 USBINT 位清零来进行确认。	R/W	0
1	UEE	USB 错误中断使能。 当该位为 1，且 USBSTS 寄存器中的 USBERRINT 位为 1 时，主机 / 设备控制器将在下一个中断阈值中发出中断。该中断由软件将 USBSTS 寄存器中的 USBERRINT 位清零来进行确认。	R/W	0
2	PCE	端口更改检测使能。 当该位为 1，且 USBSTS 寄存器中的端口更改检测位为 1 时，主机 / 设备控制器将发出中断。该中断由软件将 USBSTS 中的端口更改检测位清零来进行确认。	R/W	0
3	FRE	帧列表滚动更新使能。 当该位为 1，且 USBSTS 寄存器中的帧列表滚动更新位为 1 时，主机控制器将发出中断。该中断由软件将帧列表滚动更新位清零来进行确认。		
4	-	保留。	-	0
5	AAE	异步推进时中断使能。 当该位为 1，且 USBSTS 寄存器中的异步推进时中断位为 1 时，主机控制器将在下一个中断阈值中发出中断。该中断由软件将异步推进时中断位清零来进行确认。	R/W	0
6	-	主机控制器未使用。	-	0
7	SRE	如果该位为 1，且 USBSTS 寄存器中的 SRI 位为 1，则主机控制器将发出中断。在主机模式下， SRI 位将每 125 µs 设置一次，并可以由主机控制器用作时基。该中断由软件将 USBSTS 寄存器中的 SRI 位清零来进行确认。	-	0
8	-	主机控制器未使用。	-	0
15:9	-	保留。		
16	-	主机控制器未使用。	R/W	0
17	-	保留。		
18	UAIE	USB 主机异步中断使能。 当该位为 1，且 USBSTS 寄存器中的 USBHSTASYNCINT 位为 1 时，主机控制器将在下一个中断阈值中发出中断。该中断由软件将USBHSTASYNCINT位清零来进行确认。	R/W	0
19	UPIA	USB 主机定期中断使能。 当该位为 1，且 USBSTS 寄存器中的 USBHSTPERINT 位为 1 时，主机控制器将在下一个中断阈值中发出中断。该中断由软件将 USBHSTPERINT 位清零来进行确认。	R/W	0
31:20	-	保留。		

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

24.6.5 帧索引寄存器 (FRINDEX)

24.6.5.1 设备模式

在设备模式下，该寄存器为只读寄存器，且设备控制器会从 SOF 标记指示的帧数更新 FRINDEX[13:3]寄存器。只要USB总线收到SOF，系统便会依据SOF时标检查FRINDEX[13:3]。如果FRINDEX[13:3]与SOF时标不同，则FRINDEX[13:3]将设为SOF值，而FRINDEX[2:0]将设为 0 （1 ms 帧的 SOF）。如果 FRINDEX [13:3] 与 SOF 值相等，则 FRINDEX[2:0] 将由硬件进行数值递增 （即 125 µs 微帧的 SOF）。

表 457. 设备模式中的 USB 帧索引寄存器 （FRINDEX_D - 地址 0x4000 714C）位描述

位	符号	描述	复位值	访问类型
2:0	FRINDEX2_0	当前微帧数	-	RO
13:3	FRINDEX13_3	发送的最后一帧的当前帧数	-	RO
31:14	-	保留	-	

24.6.5.2 主机模式

该寄存器由主机控制器用于对定期帧列表编制索引。该寄存器每 125 µs （每微帧一次）更新一次。位 [N:3] 用于在定期调度执行期间选择定期帧列表中的特定项。用于索引的位数取决于系统软件在 USBCMD 寄存器的 “帧列表大小”字段中设置的帧列表大小。

该寄存器必须以双字方式写入。写入字节将产生未定义的结果。除非主机控制器处于 USBSTS 寄存器中 HCHalted 位指示的 “停止”状态 （主机模式），否则无法写入该寄存器。当运行 / 停止位设为 1 时，向该寄存器写入会产生未定义的结果。对该寄存器进行写操作还会影响 SOF 值。

表 458. 主机模式中的 USB 帧索引寄存器 （FRINDEX_H - 地址 0x4000 714C）位描述

位	符号	描述	复位值	访问类型
2:0	FRINDEX2_0	当前微帧数	-	R/W
12:3	FRINDEX12_3	1024 个元素的帧列表当前索引	-	R/W
31:13	-	保留	-	

表 459. 用于帧列表索引的位数

USBCMD 第 15 位	USBCMD 第 3 位	USBCMD 第 2 位	帧列表大小	FRINDEX12_3 位字段的大小
0	0	0	1024 元素 （4096 字节）。默认值。	12
0	0	1	512 元素 （2048 字节）。	11
0	1	0	256 元素 （1024 字节）。	10
0	1	1	128 元素 （512 字节）。	9
1	0	0	64 元素 （256 字节）。	8
1	0	1	32 元素 （128 字节）。	7
1	1	0	16 元素 （64 字节）。	6
1	1	1	8 元素 （32 字节）。	5

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

24.6.6 设备地址(DEVICEADDR)和周期性列表基址(PERIODICLISTBASE)寄存器

24.6.6.1 设备模式

该寄存器的高 7 位代表设备地址。任何控制器复位或 USB 复位后，设备地址便会设为默认地址 （0）。默认地址将匹配所有传入的地址。软件收到 SET_ADDRESS 描述符后，将对地址重新编程。

USBADRA 位用于通过允许 DCD 在 SET_ADDRESS 描述符的状态阶段前预设 USBADR 寄存器位，以加速 SET_ADDRESS 序列。

表 460. 设备模式中的 USB 设备地址寄存器 （DEVICEADDR - 地址 0x4000 7154）位描述

位	符号	值	描述	复位值	访问类型
23:0	-		保留	0	-
24	USBADRA		设备地址推进		
		0	USBADR 的任何写入都为即时写入。		
		1	如果用户在写入 USBADR 的同时或之前向该位写入 1，则 USBADR 字段写入将暂存并保留在隐式寄存器中。在端点 0 发生输入并获得应答后，USBADR 将从保持寄存器载入。 在以下条件下，硬件会自动将该位清零： • 确认向端点 0 输入。 USBADR 从暂存寄存器更新。 • 端点 0 上发生 OUT/SETUP 事件， USBADR 未更新。 • 设备复位。 USBADR 设为 0。 **注：**在 SET_ADDRESS 描述符的状态阶段之后，DCD 有 2 ms 时间来编程 USBADR 字段。该机制将确保当 DCD 在 SET_ADDRESS 状态阶段 2 ms 内无法写入设备地址时，可符合此规范。如果 DCD 在 SET_ADDRESS 数据阶段后（状态阶段开始前）通过 USBADRA=1 写入 USBADR，则 USBADR 将在正确时间即时编程并满足 2 ms USB 要求。		
31:25	USBADR		USB 设备地址	0	R/W

24.6.6.2 主机模式

该 32 位寄存器包含系统存储器中定期帧列表的开始地址。主机控制器驱动器 （HCD）将在主机控制器执行调度前加载该寄存器。该物理存储器指针参考的存储器结构假设与 4 kB 对齐。该寄存器的内容与帧索引寄存器 （FRINDEX）组合，可使主机控制器在序列中单步调试定期帧列表。

表 461. 主机模式中的 USB 周期性列表基址寄存器 （PERIODICLISTBASE - 地址 0x4000 7154）位描述

位	符号	描述	复位值	访问类型
11:0	-	保留。	不适用	-
31:12	PERBASE31_12	基址 （低）。 这些位对应于存储器地址信号 [31:12]。	不适用	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

24.6.7 端点列表地址寄存器 (ENDPOINTLISTADDR) 和异步列表地址 (ASYNCLISTADDR) 寄存器

24.6.7.1 设备模式

在设备模式下，该寄存器包含系统存储器中端点列表顶端的地址。该寄存器位 [10:0] 无法由系统软件修改，且在读取时将始终返回 0。该物理存储器指针参考的存储器结构假设与 64 字节对齐。

表 462. 设备模式中的 USB 端点列表地址寄存器 （ENDPOINTLISTADDR - 地址 0x4000 7158）位描述

位	符号	描述	复位值	访问类型
10:0	-	保留	0	-
31:11	EPBASE31_11	端点列表指针 （低） 这些位分别对应内存地址信号 31:11。该字段将参考多达 4 个队列头 （QH）的列表。（即每个端点和方向一个队列头。）	不适用	R/W

24.6.7.2 主机模式

该 32 位寄存器包含主机将执行的下一个异步队列头的地址。该寄存器位 [4:0] 无法由系统软件修改，且在读取时将始终返回 0。

表 463. 主机模式中的 USB 异步列表地址寄存器 （ASYNCLISTADDR - 地址 0x4000 7158）位描述

位	符号	描述	复位值	访问类型
4:0	-	保留	0	-
31:5	ASYBASE31_5	链接指针 （低） LPL 这些位分别对应内存地址信号 31:5。该字段只能参考一个队列头 （OH）。	-	R/W

24.6.8 TT 控制寄存器 (TTCTRL)

24.6.8.1 设备模式

该寄存器不适用于设备模式。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

24.6.8.2 主机模式

该寄存器包含内部 TT 运行所需的参数。该寄存器仅供主机控制器使用。该寄存器必须以双字方式写入。

表 464. 主机模式中的 USB TT 控制寄存器 （TTCTRL - 地址 0x4000 715C）位描述

位	符号	描述	复位值	访问类型
23:0	-	保留。	0	-
30:24	TTHA	直接连接 FS 设备或 LS 设备时的集线器地址。	不适用	R/W
31	-	保留。	0	

24.6.9 突发大小寄存器 (BURSTSIZE)

该寄存器用于控制及动态更改USB DMA控制器主接口上数据移动期间所使用的突发流量。该寄存器必须以双字方式写入。

RX 和 TX DMA 数据传输的 32 位字的默认突发长度各为 16 个字。

表 465. 设备 / 主机模式中的 USB 突发大小寄存器 （BURSTSIZE – 地址 0x4000 7160）位描述

位	符号	描述	复位值	访问类型
7:0	RXPBURST	可编程的 RX 突发长度。 该寄存器代表将数据从 USB 总线移至系统存储器时 32 位字的最大突发长度。	0x10	R/W
15:8	TXPBURST	可编程的 TX 突发长度。 该寄存器代表将数据从系统存储器移至 USB 总线时 32 位字的最大突发长度。	0x10	R/W
31:16	-	保留。	-	-

24.6.10 传输缓冲区填充调谐寄存器 (TXFILLTUNING)

24.6.10.1 设备控制器

该寄存器不适用于设备模式。

24.6.10.2 主机控制器

此寄存器中的各字段控制与主机控制器在向 USB 总线移动数据前向 TX 延迟 FIFO 发送数据的方式相关的性能调谐。性能的具体方面包括向 FIFO 中发送的数据量以及该操作在目标系统中耗时的估计值。

定义：

T_0 = 标准数据包开销

T_1 = 发送数据负载的时间

T_{ff} = 获取数据至指定水平的 TX FIFO 的时间

T_s = 数据包传输总时间 （仅发送数据包）； $T_s = T_0 + T_1$

T_p = 数据包传输总时间 （获取和发送数据包）； $T_p = T_{ff} + T_0 + T_1$

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

在数据结构中发现发送 (OUT/SETUP) 数据包后，主机控制器将检查以确定 T_p 在 （微）帧结束前一直保留。如果 T_p 在 （微）帧结束前，则继续预填充 TX FIFO。如果在预填充操作过程中的任何时间，[微] 帧剩余时间小于 < T_s 则，数据包将尝试停止，并在稍后再试。尽管这不是错误条件，且主机控制器将最终恢复，但调度程序运行状况计数器将做出标记，以注明发生回退事件。如果检测到回退事件，可能需要从延迟缓冲区中丢弃部分已获取的数据包，以为在下一个 SOF 后开始的定期通信腾出空间。回退事件过多可能会浪费系统总线上的宽带和功率，因此应尽量减少 （但没有必要清除）。通过下述 TSCHHEALTH (T_{ff}) 可以最大限度地减少回退事件。

表 466. 主机模式中的 USB 传输缓冲区填充调谐寄存器 （TXFILLTUNING - 地址 0x4000 7164）位描述

位	符号	描述	复位值	访问类型
7:0	TXSCHOH	FIFO 突发阈值 此寄存器控制在开始向总线发送数据包之前以主机模式向 TX 延迟 FIFO 发送的数据突发数量。其最小值为 2，该值应尽可能小，以便最大限度发挥 USB 性能。在延迟不可预知和 / 或带宽不足时，系统可使用更大的值，因为从系统存储器补充数据之前，数据已从延迟 FIFO 传输至 USB，FIFO 可能不足。如果已设置了 USBMODE 寄存器中的流禁用位，则将忽略该值。	0x2	R/W
12:8	TXSCHEATLTH	调度程序运行状况计数器 如果主机控制器在发送下一个帧起始前的数据包超时前，未能将 TX 延迟 FIFO 填充至 TXFIFOTHRES 编程的电平，则该寄存器将增加。 该运行状况计数器将计量该事件发生的次数，以提供反馈来选择适当的 TXSCHOH。对该寄存器执行写操作时，该计数器将清零。最大值为 31。	0x0	R/W
15:13	-	保留	-	-
21:16	TXFIFOTHRES	调度程序开销 该寄存器会将额外的固定偏移量添加到调度时间估测器，上文描述为 T_{ff}。据粗略估算，在总线利用率高的情况下，为此寄存器选用的值应将 TXSCHHEALTH 中的回退事件数量限制到每秒低于 10 次。不建议为此寄存器选用过高的值，这样会不必要地降低 USB 利用率。 如果在高速模式下连接设备，则此寄存器中使用的时间单位为 1.267 µs。 如果在低速 / 全速模式下连接设备，则此寄存器中使用的时间单位为 6.333 µs。	0x0	R/W
31:22	-	保留		

24.6.11 USB ULPI 视口寄存器 (ULPIVIEWPORT)

此寄存器提供对 ULPI PHY 寄存器设置的间接访问。虽然内核可执行对 ULPI PHY 寄存器设置的访问，但在特殊情况下，软件可能需要直接访问。

注：通过视口写入 ULPI 会对标准 USB 操作造成严重损害。目前，还没有定义使用模式来说明软件应需要在何处直接对 ULPI 执行写操作 – 参见下文有关可选功能的异常处理。

注：通过 ULPI 视口执行读取操作，应该不会对标准 USB 操作产生不良后果。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

可通过 ULPI 视口执行的操作有两种：唤醒与读 / 写操作。唤醒操作用于将 ULPI 接口置于正常操作模式中，并在必要时重新使能时钟。当 ULPI 接口在低功耗模式、串行模式或车载模式下运行时，访问寄存器前需要进行唤醒操作。 ULPI 状态可通过读取同步状态位 (ULPISS) 来决定。如果该位为 1，则 ULPI 接口在正常操作模式下运行，并可接受读 / 写操作。如果 ULPISS 指示 0，则将无法执行读 / 写操作。如果 ULPISS = 0，并执行读或写操作，则会导致未定义的行为。要执行唤醒操作，应写入 ULPI 视口的所有 32 位数据，其中 ULPIPORT 构造正确、ULPIWU 位为 1 且 ULPIRUN 位为 0。查询 ULPI 视口直至 ULPIWU 为零，以便完成操作。

要执行读 / 写操作，可写入 ULPI 视口的所有 32 位数据，其中 ULPIDATWR、ULPIADDR、ULPIPORT、 ULPIRW 构造正确，且 ULPIRUN 位为 1。查询 ULPI 视口，直至 ULPIRUN 为零，以便完成操作。当 ULPIRUN 为零时，如果是读操作，则 ULPIDATRD 将有效。

上述轮询方式也可替换，并使用 USBSTS 和 USBINTR 寄存器中定义的 ULPI 中断驱动中断。唤醒或读 / 写操作完成时，便会设置 ULPI 中断。

表 467. USB ULPI 视口寄存器 （ULPIVIEWPORT – 地址 0x4000 7170）位描述

位	符号	值	描述	访问类型	复位值
7:0	ULPIDATWR		当命令进行写操作时，要发送的数据将被写入该字段。	R/W	0
15:8	ULPIDATRD		在读操作完成后，结果将被置于该字段。	R	0
23:16	ULPIADDR		当命令进行读或写操作时，操作的地址将被写入该字段。	R/W	0
26:24	ULPIPORT		对于将要执行的唤醒或读 / 写操作，该值必须写为 0。	R/W	000
27	ULPISS		ULPI 同步状态。该位表示 ULPI 接口的状态。	R	0
		0	处于另一种状态中 （即车载、串行、低功耗）。		
		1	正常同步。状态。		
28	-	-	保留。	-	-
29	ULPIRW		ULPI 读 / 写控制。该位在运行读或写操作之间进行选择。	R/W	0
		0	读。		
		1	写。		
30	ULPIRUN		ULPI 读 / 写运行。 向此位写入 1，将开始读 / 写操作。读 / 写操作完成后，该位将自动跳变至 0。一旦设置此位，驱动器将无法将其重新设置为 0。 **注：**驱动器不能同时执行唤醒和读 / 写操作。	R/W	-
31	ULPIWU		ULPI 唤醒。 向此位写入 1，将开始唤醒操作。唤醒完成后，此位将自动变为 0。一旦设置此位，驱动器将无法将其重新设置为 0。 **注：**驱动器不能同时执行唤醒和读 / 写操作。	R/W	0

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

24.6.12 BINTERVAL 寄存器

该寄存器确定 bInterval 值，该值决定虚拟帧的长度 （参见第 23.7.7 节）。

注：BINTERVAL 寄存器和 USB 规范中的 bInterval 端点描述符字段无关。

表 468. 设备 / 主机模式中的 USB BINTERVAL 寄存器 （BINTERVAL – 地址 0x4000 7174）位描述

位	符号	描述	复位值	访问类型
3:0	BINT	bInterval 值	0x00	R/W
31:4	-	保留	-	-

24.6.13 USB 端点 NAK 寄存器 (ENDPTNAK)

24.6.13.1 设备模式

该寄存器指示设备何时发送端点上的 NAK 信号握手。每个 Tx 和 Rx 端点分别在 EPTN 和 EPRN 字段中有一个对应的位。

向该寄存器中的位写入 1，即可将该位清零。

表 469. 设备模式中的 USB 端点 NAK 寄存器 （ENDPTNAK – 地址 0x4000 7178）位描述

位	符号	描述	复位值	访问类型
3:0	EPRN	Rx 端点 NAK 每个 RX 端点在此字段均有一个位。设备在相应端点接收到的 OUT 或 PING 标记上发送 NAK 反馈检验时，该位会被设置。 位 3 对应于端点 3。 ... 位 1 对应于端点 1。 位 0 对应于端点 0。	0x00	R/WC
15:6	-	保留	-	-
19:16	EPTN	Tx 端点 NAK 每个 TX 端点在该字段中都有一个对应的位。当设备发送相应端点收到的 IN 令牌上的 NAK 信号握手时，便会设置该位。 位 3 对应于端点 3。 ... 位 1 对应于端点 1。 位 0 对应于端点 0。	0x00	R/WC
31:20	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

24.6.13.2 主机模式

该寄存器不适用于主机模式。

24.6.14 USB 端点 NAK 使能寄存器 (ENDPTNAKEN)

24.6.14.1 设备模式

该寄存器中的每个位会使能 ENDPTNAK 寄存器中相应的位。每个 Tx 和 Rx 端点分别在 EPTNE 和 EPRNE 字段中有一个对应的位。

表 470. 设备模式中的 USB 端点 NAK 使能寄存器 （ENDPTNAKEN – 地址 0x4000 717C）位描述

位	符号	描述	复位值	访问类型
3:0	EPRNE	Rx 端点 NAK 使能 每个位可使能对应的 RX NAK 位。如果设置该位和 RX 端点 NAK 位，则会设置 NAK 中断位。 位 3 对应于端点 3。 ... 位 1 对应于端点 1。 位 0 对应于端点 0。	0x00	R/W
15:4	-	保留	-	-
19:16	EPTNE	Tx 端点 NAK 每个位可使能对应的 TX NAK 位。如果设置该位和 TX 端点 NAK 位，则会设置 NAK 中断位。 位 3 对应于端点 3。 ... 位 1 对应于端点 1。 位 0 对应于端点 0。	0x00	R/W
31:20	-	保留	-	-

24.6.14.2 主机模式

该寄存器不适用于主机模式。

24.6.15 端口状态和控制寄存器 (PORTSC1)

24.6.15.1 设备模式

设备控制器提供一个端口寄存器，且不支持电源控制。设备模式下的端口控制用于状态端口复位、挂起和当前连接状态。它还可用于启动测试模式或强制发出信号。通过该寄存器，软件可以使 PHY 进入低功耗挂起模式及禁用 PHY 时钟。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 471. 设备模式中的端口状态和控制寄存器 （PORTSC1_D - 地址 0x4000 7184）位描述

位	符号	值	描述	复位值	访问类型
0	CCS		当前连接状态	0	RO
		0	设备未连接 0 表示设备连接失败，或因软件向 USBCMD 寄存器中的运行位写入 0 而强制断开。该位不会指明设备已断开或挂起。		
		1	设备已连接。 1 表示设备成功连接并在高速模式或全速模式下运行，由此寄存器中的 “高速端口” 位指示。		
1	CSC	-	不适用于设备模式	0	-
2	PE	1	端口使能。 该位始终为 1。设备端口始终使能。	1	RO
3	PEC	0	端口使能 / 禁用更改 该位始终为 0。设备端口始终使能。	0	RO
5:4	-	-	保留	0	RO
6	FPR		强制端口恢复 在设备处于挂起状态 5ms 或更长时间之后，软件必须将此位设置为 1，以便在清除之前驱动恢复信号。如果端口处于挂起状态时检测到 J 至 K 跳变，则设备控制器会将该位设为 1。当设备恢复正常工作时，该位便会清零。如果此位因为检测到 J 至 K 转换而转换至 1，则 USBSTS 寄存器中的 “端口改变检测” 位也设置为 1。	0	R/W
		0	不恢复 （K 状态）端口检测 / 驱动。		
		1	恢复端口检测 / 驱动。		
7	SUSP		挂起 在设备模式下，这是一个只读状态位。	0	RO
		0	端口未处于挂起状态		
		1	端口处于挂起状态		
8	PR		端口复位 在设备模式下，这是一个只读状态位。从 USB 总线复位的设备也会在 USBSTS 寄存器中指出。	0	RO
		0	端口未处于复位状态。		
		1	端口处于复位状态。		
9	HSP		高速状态 **注：**此位与此寄存器中的位 27:26 (PSPD) 冗余。因兼容性原因而采用。	0	RO
		0	连接到端口的主机 / 设备未处于高速模式下。		
		1	连接到端口的主机 / 设备处于高速模式下。		
11:10	LS	-	不适用于设备模式。		
12	PP	-	不适用于设备模式。		
13	-	-	保留	-	-
15:14	PIC1_0		端口指示器控制 向此字段执行写操作将影响 USB1_IND1:0 引脚的值。	00	R/W
		0x0	端口指示器关闭。		
		0x1	黄色		
		0x2	绿色		
		0x3	未定义		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 471. 设备模式中的端口状态和控制寄存器 （PORTSC1_D - 地址 0x4000 7184）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
19:16	PTC3_0		端口检测控制 除 0000 以外的任何值表示端口正在检测模式下运行。 FORCE_ENABLE_FS 和 FORCE ENABLE_LS 是 EHCI 规范中指定的测试模式支持的扩展功能。向 PTC 字段写入任何 FORCE_ENABLE_HS/FS/LS 值将强制端口以所选速度进入连接和使能状态。通过将 PTC 字段写回 TEST_MODE_DISABLE，端口状态机可以从该点正常运行。值 0x7 到 0xF 保留。	0000	R/W
		0x0	TEST_MODE_DISABLE		
		0x1	J_STATE		
		0x2	K_STATE		
		0x3	SE0 （主机） /NAK （设备）		
		0x4	数据包		
		0x5	FORCE_ENABLE_HS		
		0x6	FORCE_ENABLE_FS		
20	-	-	不适用于设备模式。此位在设备模式下始终为 0。	0	-
21	-	-	不适用于设备模式。此位在设备模式下始终为 0。	0	-
22	-		不适用于设备模式。此位在设备模式下始终为 0。	0	-
23	PHCD		PHY 低功耗挂起 - 时钟禁用 （PLPSCD） 在设备模式下，当设备未运行 (USBCMD Run/Stop = 0) 或主机已发出挂起信号 (PORTSC SUSPEND = 1) 时， PHY 可处于低功耗挂起 - 时钟禁用状态。当主机发出恢复信号时，低功耗挂起将自动清零。在强制从设备恢复前，设备控制器驱动器必须将该位清零。	0	R/W
		0	如果写入 0，则将使能 PHY 时钟。如果读取 0，则将指示 PHY 时钟的状态（使能）。		
		1	如果写入 1，则将禁用 PHY 时钟。如果读取 1，则将指示 PHY 时钟的状态（禁用）。		
24	PFSC		端口强制全速连接	0	R/W
		0	端口以任何速度连接。		
		1	如果向该位写入 1，则将强制端口仅以全速连接。它将禁用脉冲组序列，后者可使端口将其本身标识为高速。该功能对于使用 HS 主机、集线器或设备测试 FS 配置非常有效。		
25	-	-	保留		
27:26	PSPD		端口速度 该寄存器字段指示端口运行的速度。	0	RO
		0x1	全速		
		0x2	在设备模式中无效		
		0x3	高速		
29:28	-	-	保留	-	-
31:30	PTS		并行收发器选择。保留所有其它值。	< 待定 >	R/W
		0x2	ULPI		
		0x3	串行 /1.1 PHY （仅全速）		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

24.6.15.2 主机模式

主机控制器使用一个端口。只有在初次上电或对控制器复位作出响应时，寄存器才会复位。端口的初始状况为：

- 设备未连接
- 端口禁用

如果端口具有电源控制功能，则该状态将保持，直到软件通过在 PORTSC 寄存器中将端口电源设为 1 对端口上电为止。

表 472. 主机模式中的端口状态和控制寄存器 （PORTSC1_H - 地址 0x4000 7184）位描述

位	符号	值	描述	复位值	访问类型
0	CCS		当前连接状态 该值反映端口的当前状态，且可能不会与导致 CSC 位设置的事件直接对应。 如果 PP （端口电源位）为 0，则该位为 0。 软件向该位写入 1，即可将该位清零。	0	R/WC
		0	不存在设备。		
		1	端口上存在设备。		
1	CSC		连接状态更改 指示端口的当前连接状态已发生变化。即使系统软件未将现有连接状态更改清零，主机 / 设备控制器也会针对端口设备连接状态的所有更改设置该位。例如，系统软件清除变更情况前插入状态变化两次，集线器硬件将设置已设置好的位 （即该位保持设置状态）。软件向该位写入 1，即可将该位清零。 如果 PP （端口电源位）为 0，则该位为 0。	0	R/WC
		0	当前状态未更改。		
		1	当前状态更改。		
2	PE		端口使能。 作为复位和使能的一部分，端口只能通过主机控制器使能。软件无法通过向该字段写入 1 来使能端口。端口可以因故障条件 （断开事件或其他故障条件）或由主机软件来禁用。需要注意的是，在端口状态实际更改前，位状态不会更改。由于其他主机控制器和总线事件的关系，禁用或使能端口时可能存在延迟。 如果禁用端口，除非进行复位，否则数据的下行传播将被阻断。 如果 PP （端口电源位）为 0，则该位为 0。	0	R/W
		0	端口禁用。		
		1	端口使能。		
3	PEC	0	端口禁用 / 使能更改 对于根集线器，此位仅在端口由于断开或由于在 EOF2 点存在适当条件而被禁用时设置为 1（参见 *USB 规范的第 11 章*）。软件通过在此写入 1 来清除。 如果 PP （端口电源位）为 0，则该位为 0。	0	R/WC
		0	无变化。		
		1	端口使能 / 禁用状态已更改。		
4	OCA		过电流激活 过电流条件被去除后，此位自动从 1 转换至 0。	0	RO
		0	端口没有过流条件。		
		1	端口当前具有过流条件。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 472. 主机模式中的端口状态和控制寄存器 （PORTSC1_H - 地址 0x4000 7184）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
5	OCC		过流更改 当过流有效更改时，该位将设为 1。软件向该位位置写入 1，即可将该位清零。	0	R/WC
6	FPR		强制端口恢复 软件将该位设为 1，即可驱动恢复信号。如果端口处于挂起状态时检测到 J 至 K 跳变，则主机控制器会将该位设为 1。如果由于检测到 J 至 K 跳变，该位跳变为 1，则 USBSTS 寄存器中的端口更改检测位也会设为 1。在恢复序列完成后，该位将自动更改为 0。该行为与 EHCI 不同，后者是在驱动器中对恢复持续时间计时后，需要主机控制器将该位设为 0。 注意，如果主机控制器拥有端口，则恢复序列遵守 USB 规范修订版 2.0 中定义的序列。只要该位保持为 1，该恢复信号发送 （全速 K）在端口上驱动。在端口切换至高速空闲之前，该位将保持为 1。由于端口控制器将对恢复操作计时，因此写入 0 并无影响，将该位清零后，端口控制状态将切换至 HS 或 FS 空闲状态。 如果 PP （端口电源位）为 0，则该位为 0。	0	R/W
		0	不恢复 （K 状态）端口检测 / 驱动。		
		1	恢复端口检测 / 驱动。		
7	SUSP		挂起 该位与 PE（端口使能位）一样，都是描述端口状态，参见 表 473“PORTSC1 寄存器中 PE 和 SUSP 位指定的端口状态”。 当软件将强制端口恢复位设为 0 时，主机控制器将无条件将该位设为 0。主机控制器将忽略向该位写入 0。 在端口未使能 （即端口端口位为 0）时，如果主机软件将该位设为 1，则将产生未定义的结果。 如果 PP （端口电源位）为 0，则该位为 0。	0	R/W
		0	端口未处于挂起状态		
		1	端口处于挂起状态 当处于挂起状态时，除非对端口进行复位，否则端口上数据的下行传输将被阻断。如果某个事务正在进行时向该位写 1，则在当前事务结束时会发生阻断。当处于挂起状态时，端口对恢复检测非常敏感。需要注意的是，在端口处于挂起状态前，位状态不会更改；而如果 USB 上当前正在处理事务，则端口挂起可能出现延迟。		
8	PR		端口复位 当软件向该位写入 1 时，便会使能 USB 规范 2.0 版中定义的总线复位序列。在复位序列完成后，该位将自动更改为 0。该行为与 EHCI 不同，后者是在驱动器中对复位持续时间计时后，需要主机控制器将该位设为 0。 如果 PP （端口电源位）为 0，则该位为 0。	0	R/W
		0	端口未处于复位状态。		
		1	端口处于复位状态。		
9	HSP		高速状态	0	RO
		0	连接到端口的主机 / 设备未处于高速模式下。		
		1	连接到端口的主机 / 设备处于高速模式下。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 472. 主机模式中的端口状态和控制寄存器 （PORTSC1_H - 地址 0x4000 7184）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
11:10	LS		线路状态 这些位反映 USB_DP 和 USB_DM 信号线的当前逻辑电平。USB_DP 对应于位 11，而 USB_DM 对应于位 10。 在主机模式下，由于控制器硬件管理 LS 和 FS 的连接，主机控制器驱动器无需将线路状态用于该控制器 （与 EHCI 不同）。	0x3	RO
		0x0	SE0 （USB_DP 和 USB_DM 低电平）		
		0x1	J 状态 （USB_DP 高电平和 USB_DM 低电平）		
		0x2	K 状态 （USB_DP 低电平和 USB_DM 高电平）		
		0x3	未定义		
12	PP	-	端口电源控制 主机控制器需要端口电源控制开关。该位表示开关的当前设置 （0= 关， 1= 开）。当端口上电源不可用 （即 PP=0）时，端口将不能正常运行且不会报告连接、分离等状况。 如果在上电端口检测到过流条件，且 PPC 为 1，每个受影响端口中的 PP 位可能由主机控制器驱动器从 1 跳变为 0 （移除端口的电源）。	0	R/W
		0	端口电源关闭。		
		1	端口上电。		
13	-	-	保留	0	-
15:14	PIC1_0		端口指示器控制 向此字段执行写操作将控制引脚 USB1_IND1 和 USB1_IND0 的值。	00	R/W
		0x0	端口指示器关闭。		
		0x1	黄色		
		0x2	绿色		
		0x3	未定义		
19:16	PTC3_0		端口检测控制 除 0000 以外的任何值表示端口正在检测模式下运行。 FORCE_ENABLE_FS 和 FORCE ENABLE_LS 是 EHCI 规范中指定的测试模式支持的扩展功能。向 PTC 写入任何 FORCE_ENABLE_{HS/FS/LS} 值将强制端口以所选速度进入连接和使能状态。通过将 PTC 字段写回 TEST_MODE_DISABLE，端口状态机可以从该点正常运行。值 0x8 至 0xF 保留。	0000	R/W
		0x0	TEST_MODE_DISABLE		
		0x1	J_STATE		
		0x2	K_STATE		
		0x3	SE0 （主机） /NAK （设备）		
		0x4	数据包		
		0x5	FORCE_ENABLE_HS		
		0x6	FORCE_ENABLE_FS		
		0x7	FORCE_ENABLE_LS		
20	WKCN		连接使能时唤醒 (WKCNNT_E) 如果 PP （端口电源位）为 0，则该位为 0。	0	R/W
		0	禁止端口在设备连接时唤醒。		
		1	如果向该位写入 1，则端口将对作为唤醒事件的设备连接非常敏感。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 472. 主机模式中的端口状态和控制寄存器 （PORTSC1_H - 地址 0x4000 7184）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
21	WKDC		断开时唤醒使能 （WKDSCNNT_E） 如果 PP （端口电源位）为 0，则该位为 0。	0	R/W
		0	禁止端口在设备断开时唤醒。		
		1	如果向该位写入 1，则端口将对作为唤醒事件的设备断开非常敏感。		
22	WKOC		过流时唤醒使能 （WKOC_E）	0	R/W
		0	禁止端口在过流事件时唤醒。		
		1	如果向该位写入 1，则端口将对作为唤醒事件的过流条件非常敏感。		
23	PHCD		PHY 低功耗挂起 - 时钟禁用 （PLPSCD） 在主机模式下，当下行设备处于挂起模式下或未连接下行设备时，PHY 可处于低功耗挂起 - 时钟禁用状态。低功耗挂起可完全由软件控制。	0	R/W
		0	如果写入 0，则将使能 PHY 时钟。如果读取 0，则将指示 PHY 时钟的状态（使能）。		
		1	如果写入 1，则将禁用 PHY 时钟。如果读取 1，则将指示 PHY 时钟的状态（禁用）。		
24	PFSC		端口强制全速连接	0	R/W
		0	端口以任何速度连接。		
		1	如果向该位写入 1，则将强制端口仅以全速连接。它将禁用脉冲组序列，后者可让端口将其本身标识为高速。该功能对于使用 HS 主机、集线器或设备测试 FS 配置非常有效。		
25	-	-	保留		
27:26	PSPD		端口速度 该寄存器字段指示端口运行的速度。如果在主机控制器中的 HS 模式下工作及在设备控制器中的HS/FS模式下工作，则端口线路可控制传输至协议引擎的数据。如果在主机控制器中的 FS 和 LS 模式下工作，则端口线路可通过嵌入式事务转译器来控制传输至协议引擎的数据。	0	RO
		0x0	全速		
		0x1	低速		
		0x2	高速		
29:28	-	-	保留	-	-
31:30	PTS		并行收发器选择。保留所有其它值。	< 待定 >	R/W
		0x2	ULPI		
		0x3	串行 /1.1 PHY （仅全速）		

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 473. PORTSC1 寄存器中 PE 和 SUSP 位指定的端口状态

PE 位	SUSP 位	端口状态
0	0 或 1	禁用
1	0	使能
1	1	挂起

24.6.16 USB 模式寄存器 (USBMODE)

USBMODE 寄存器可设置 USB 控制器的 USB 模式。可能模式包括设备、主机和空闲模式。

24.6.16.1 设备模式

表 474. 设备模式中的 USB 模式寄存器 （USBMODE_D – 地址 0x4000 71A8）位描述

位	符号	值	描述	复位值	访问类型
1:0	CM1_0		控制器模式 控制器默认设为空闲状态，且在复位后需要初始化才能进入所需的工作模式。该寄存器在复位后只能写入一次。如果需要切换模式，则软件必须在重新编程该寄存器前，对 USBCMD 寄存器中的复位位执行写入操作才能复位控制器。	00	R/WO
		0x0	空闲		
		0x1	保留		
		0x2	设备控制器		
		0x3	主机控制器		
2	ES		字节序选择 该位可更改传输缓冲区的字节顺序，以匹配主机微处理器总线架构。微处理器接口和 DMA 数据结构 （包括设备 QH 中的设备缓冲区）中的位字段是以 32 位字为基础，因此不会受该位值的影响。	0	R/W
		0	小端：32 位字的最低有效位参考第一个字节。		
		1	大端：32 位字的最高有效位参考第一个字节。		
3	SLOM		设置锁定模式 在设备模式中，此位控制设置锁定机制的行为。参见第 23.10.8 节。	0	R/W
		0	设置锁定开		
		1	设置锁定关 （DCD 需要在 USBCMD 中使用设置缓冲区绊网）		
4	SDIS		流禁用模式 **注：**使用此功能会严重限制可达到的 USB 整体性能。	0	R/W
		0	未禁用		
		1	禁能。 如果将该位设为 1，将禁用低带宽系统中 RX 和 TX 的双启动。该模式可确保当 RX 和 TX 缓冲区足以容纳整个数据包时，禁用标准双缓冲机制，以防止带宽有限系统中出现溢出 / 不足。注：处于高速模式时，如果流禁用有效，则将通过 NYET 信号握手对收到的所有数据包作出响应。		
5	-		不适用于设备模式。	0	-
31:6	-	-	保留		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

24.6.16.2 主机模式

表 475. 主机模式中的 USB 模式寄存器 （USBMODE_H – 地址 0x4000 71A8）位描述

位	符号	值	描述	复位值	访问类型
1:0	CM1_0		控制器模式 控制器默认设为空闲状态，且在复位后需要初始化才能进入所需的工作模式。该寄存器在复位后只能写入一次。如果需要切换模式，则软件必须在重新编程该寄存器前，对 USBCMD 寄存器中的复位位执行写入操作才能复位控制器。	00	R/WO
		0x0	空闲		
		0x1	保留		
		0x2	设备控制器		
		0x3	主机控制器		
2	ES		字节序选择 该位可更改传输缓冲区的字节顺序。微处理器接口和 DMA 数据结构 （包括设备 QH 中的设备缓冲区）中的位字段是以 32 位字为基础，因此不会受该位值的影响。	0	R/W
		0	小端：32 位字的最低有效位参考第一个字节。		
		1	大端：32 位字的最高有效位参考第一个字节。		
3	-		不适用于主机模式	0	-
4	SDIS		流禁用模式 **注：**使用此功能会严重限制可达到的 USB 整体性能。	0	R/W
		0	未禁用		
		1	禁能。 对于 RX 和 TX 缓冲区足以包含整个数据包的低带宽系统，设置为 1 可确保消除延期 FIFO 的溢出 / 欠载。在 USB 上启动数据包前，使能流禁用也可确保 TX 延迟填充到容量。 注: 当流禁用有效时，预填充FIFO的持续时间非常重要。请参见TXFILLTUNING，以确定使用该功能时调度器所需的调整。		
5	VBPS		VBUS 电源选择	0	R/WO
		0	vbus_pwr_select 设为低电平。		
		1	vbus_pwr_select 设为高电平。		
31:6	-	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

24.6.17 USB 端点设置状态寄存器 (ENDPSETUPSTAT)

表 476. USB 端点设置状态寄存器 （ENDPTSETUPSTAT - 地址 0x4000 71AC）位描述

位	符号	描述	复位值	访问类型
3:0	ENDPT SETUP STAT	逻辑端点的设置端点状态。 对于收到的每个设置事务，该寄存器中相应的位将设为 1。软件必须清除或确认设置传输，方法是在从队列头读取设置数据后向相应位写入 1。在设置锁定机制使能时，按操作顺序和总响应时间对设置数据包做出的响应对于限制总线时间超时非常重要。	0	R/WC
31:4	-	保留。	-	-

24.6.18 USB 端点启动寄存器 (ENDPTPRIME)

对于每个端点，只要向端点传递新的传输描述符，软件即应向相应位写入 1。硬件将自动使用该位开始分析来自队列头的新传输描述符，并准备接收缓冲区。当相关端点成功启动时，硬件便会将该位清零。

注：在 dTD 退出和 dQH 更新时，硬件端点重新启动操作期间，由硬件立即对这些位进行设置。

表 477. USB 端点启动寄存器 （ENDPTPRIME - 地址 0x4000 71B0）位描述

位	符号	描述	复位值	访问类型
3:0	PERB	物理 OUT 端点的启动端点接收缓冲区。 对于每个 OUT 端点，软件都会将相应的位设置为 1，在 USB 主机启动 USB OUT 事务时，请求为接收操作准备缓冲区。只要向端点传递新的传输描述符，软件即应向相应位写入 1。硬件将自动使用该位开始分析来自队列头的新传输描述符，并准备接收缓冲区。当相关端点成功启动时，硬件便会将该位清零。 PERB0 = 端点 0 ... PERB3 = 端点 3	0	R/WS
15:4	-	保留	-	-
19:16	PETB	物理 IN 端点的启动端点发送缓冲区。 对于每个 IN 端点，相应位将由软件设为 1，请求为发送操作准备缓冲区，以便对 USB IN/INTERRUPT 事务做出响应。当向端点传递新的传输描述符，软件即应向相应位写入 1。硬件将自动使用该位来开始分析来自队列头的新传输描述符，并准备发送缓冲区。当相关端点成功启动时，硬件便会将该位清零。 PETB0 = 端点 0 ... PETB3 = 端点 3	0	R/WS
31:20	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

24.6.19 USB 端点刷新寄存器 (ENDPTFLUSH)

如果向该寄存器中的位写入 1，将导致相关端点将任何已启动的缓冲区清零。如果数据包正在其中一个相关端点上处理，则该传输将继续执行，直到完成为止。在端点刷新操作成功后，硬件便会将该寄存器清零。

表 478. USB 端点刷新寄存器 （ENDPTFLUSH – 地址 0x4000 71B4）位描述

位	符号	描述	复位值	访问类型
3:0	FERB	物理 OUT 端点的刷新端点接收缓冲区。 如果向相关位写入 1，将使任何已启动的缓冲区清零。 FERB0 = 端点 0 ... FERB3 = 端点 3	0	R/WS
15:4	-	保留	-	-
19:16	FETB	物理 IN 端点的刷新端点发送缓冲区。 如果向相关位写入 1，将使任何已启动的缓冲区清零。 FETB0 = 端点 0 ... FETB3 = 端点 3	0	R/WS
31:20	-	保留	-	-

24.6.20 USB 端点状态寄存器 (ENDPTSTAT)

每个端点的一个位指示相应端点缓冲区的状态。该位由硬件设置，作为对收到 ENDPTPRIME 寄存器中相应位发出的命令的响应。设置 ENDPTPRIME 寄存器中的位与端点指示就绪之间将始终存在延迟。该延迟时间取决于当前的 USB 流量和 ENDPTPRIME 寄存器中所设置的位数。缓冲区就绪将通过 USB 复位、USB DMA 系统或 ENDPTFLUSH 寄存器清零。

注：在 dTD 退出和 dQH 更新时，硬件端点重新启动操作期间，由硬件立即对这些位进行清除。

表 479. USB 端点状态寄存器 （ENDPTSTAT – 地址 0x4000 71B8）位描述

位	符号	描述	复位值	访问类型
3:0	ERBR	物理 OUT 端点的端点接收缓冲区就绪。 该位由硬件设为 1，作为对收到 ENDPTPRIME 寄存器中相应位发出的命令的响应。 ERBR0 = 端点 0 ... ERBR3 = 端点 3	0	RO
15:4	-	保留	-	-
19:16	ETBR	物理 IN 端点 3 至 0 的端点发送缓冲区就绪。 该位由硬件设为 1，作为对收到 ENDPTPRIME 寄存器中相应位发出的命令的响应。 ETBR0 = 端点 0 ... ETBR3 = 端点 3	0	RO
31:20	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

24.6.21 USB 端点完成寄存器 (ENDPTCOMPLETE)

该寄存器中的每个位指示是否发生收到 / 发送事件，且软件应读取相应的端点队列以确定传输状态。如果在传输描述符中设置相应的 IOC 位，则该位将与 USBINT 同时设置。

如果写入 1，便会将该寄存器中相应的位清零。

表 480. USB 端点完成寄存器 （ENDPTCOMPLETE – 地址 0x4000 71BC）位描述

位	符号	描述	复位值	访问类型
3:0	ERCE	物理 OUT 端点的端点接收完成事件。 如果发生接收事件 （OUT/SETUP），则该位将由硬件设为 1。 ERCE0 = 端点 0 ... ERCE3 = 端点 3	0	R/WC
15:4	-	保留	-	-
19:16	ETCE	物理 IN 端点的端点发送完成事件。 如果发生发送事件 （IN/INTERRUPT），则该位将由硬件设为 1。 ETCE0 = 端点 0 ... ETCE3 = 端点 3	0	R/WC
31:20	-	保留	-	-

24.6.22 USB 端点 0 控制寄存器 (ENDPTCTRL0)

该寄存器可初始化用于控制传输的端点 0。端点 0 始终为控制端点。

表 481. USB 端点 0 控制寄存器 （ENDPTCTRL0 - 地址 0x4000 71C0）位描述

位	符号	值	描述	复位值	访问类型
0	RXS		Rx 端点停止	0	R/W
		0	端点正常。		
		1	端点停止 软件可以向该位写入 1，以强制端点向主机返回 STALL 信号握手。端点将继续返回 STALL，直到该位由软件清零，或在收到新的设置请求后自动清零为止。 收到 SETUP 请求后，此位将继续由硬件清除，直到相关的 ENDSETUPSTAT 位清除为止。[1]		
1	-	-	保留		
3:2	RXT	0x0	端点类型 端点 0 始终为控制端点。	0	R/W
6:4	-	-	保留	-	-
7	RXE	1	Rx 端点使能 端点使能。控制端点 0 始终处于使能状态。该位始终为 1。	1	RO
15:8	-	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 481. USB 端点 0 控制寄存器 （ENDPTCTRL0 - 地址 0x4000 71C0）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
16	TXS		Tx 端点停止		R/W
		0	端点正常。		
		1	端点停止 软件可以向该位写入 1，以强制端点向主机返回 STALL 信号握手。端点将继续返回 STALL，直到该位由软件清零，或在收到新的设置请求后自动清零为止。 收到SETUP请求后，此位将继续由硬件清除，直到相关的ENDSETUPSTAT位清除为止。[1]		
17	-	-	保留		
19:18	TXT	0x0	端点类型 端点 0 始终为控制端点。	0	RO
22:20	-	-	保留		
23	TXE	1	Tx 端点使能 端点使能。控制端点 0 始终处于使能状态。该位始终为 1。	1	RO
31:24	-	-	保留		

[1] ENPTSETUPSTAT 清零与硬件继续将该位清零之间会稍有延迟（最多 50 个时钟）。在大多数系统中，DCD 软件不大可能会注意到该延迟。但是，如果 DCD 注意到向 STALL 位写入 1 后未设置该位，软件将继续写入该 STALL 位，直到该位设置为止，或直到通过检查相关的 ENDPTSETUPSTAT 位已收到新设置为止。

24.6.23 端点 1 到 3 控制寄存器

每个并非控制端点的端点都有其自身的寄存器，以便设置端点类型及使能或禁用端点。

注：所有端点类型的复位值均为控制端点。如果一个端点方向使能，而相反方向的配对端点禁用，则未使用方向的端点类型必须由控制类型更改为任何其他类型（如批量）。退出未配置的端点控制将导致活动端点上数据 PID 追踪出现未定义的行为。

表 482. USB端点1到3控制寄存器（ENDPTCTRL – 地址0x4000 71C4 (ENDPTCTRL1)到0x4000 71CC (ENDPTCTRL3)）位描述

位	符号	值	描述	复位值	访问类型
0	RXS		Rx 端点停止	0	R/W
		0	端点正常。 如果该端口配置为控制端点，则该位在收到设置请求后将自动清零；且在相关的ENDPTSETUPSTAT位清零之前，该位将继续由硬件清零。		
		1	端点停止 软件可以向该位写入 1，以强制端点向主机返回 STALL 信号握手。它将继续返回 STALL，直至软件清除此位，或接收到新的 SETUP 请求时自动将其清除。[1]		
1	-		保留	0	R/W
3:2	RXT		端点类型	00	R/W
		0x0	控制		
		0x1	同步		
		0x2	批量		
		0x3	中断		
4	-	-	保留		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 482. USB端点1到3控制寄存器（ENDPTCTRL – 地址0x4000 71C4 (ENDPTCTRL1)到0x4000 71CC (ENDPTCTRL3)）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
5	RXI		Rx 数据切换禁止 该位仅用于测试，且应始终写入 0。如果向该位写入 1，将导致该端点忽略数据切换序列并始终接受数据包，而无论其数据 PID 如何。	0	R/W
		0	已禁用		
		1	使能		
6	RXR		Rx 数据切换复位 写入 1 可复位 PID 序列。 只要收到该端点的配置事件，软件便必须向该位写入 1，以便同步主机与设备之间的数据 PID。	0	WS
7	RXE		Rx 端点使能 **注：**端点应仅在配置后使能。	0	R/W
		0	端点禁用。		
		1	端点使能。		
15:8	-	-	保留		
16	TXS		Tx 端点停止	0	R/W
		0	端点正常。 如果该端口配置为控制端点，则该位在收到设置请求后将自动清零；且在相关的 ENDPTSETUPSTAT 位清零之前，该位将继续由硬件清零。		
		1	端点停止。 软件可以向该位写入 1，以强制端点向主机返回 STALL 信号握手。它将继续返回 STALL，直至软件清除此位，或接收到新的 SETUP 请求时自动将其清除。[1]		
17	-	-	保留	0	-
19:18	TXT		Tx 端点类型	00	R/W
		0x0	控制		
		0x1	同步		
		0x2	批量		
		0x3	中断		
20	-	-	保留		
21	TXI		Tx 数据切换禁止 该位仅用于测试，且应始终写入 0。如果向该位写入 1，将导致该端点忽略数据切换序列并始终接受数据包，而无论其数据 PID 如何。	0	R/W
		0	使能		
		1	已禁用		
22	TXR		Tx 数据切换复位 写入 1 可复位 PID 序列。 只要收到该端点的配置事件，软件便必须向该位写入 1，以便同步主机与设备之间的数据 PID。	1	WS

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 482. USB端点1到3控制寄存器（ENDPTCTRL – 地址0x4000 71C4 (ENDPTCTRL1)到0x4000 71CC (ENDPTCTRL3)）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
23	TXE		Tx 端点使能 **注：**端点应仅在配置后使能。	0	R/W
		0	端点禁用。		
		1	端点使能。		
31:24	-	-	保留	0	

[1] 仅用于控制端点：ENPTSETUPSTAT 清零与硬件继续将该位清零之间会稍有延迟（最多 50 个时钟）。在大多数系统中， DCD 软件不大可能会注意到该延迟。但是，如果 DCD 注意到向 STALL 位写入 1 后未设置该位，软件将继续写入该 STALL 位，直到该位设置为止，或直到通过检查相关的 ENDPTSETUPSTAT 位已收到新设置为止。

24.7 功能说明

有关设备数据结构的详细信息，请参见第 23.9 节。关于设备操作模型，请参见第 23.10 节。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

25.1 本章导读

USB ROM API 在器件 LPC4350/30/20 上可用。

25.2 简介

Boot ROM 包含一个用于简化 USB 应用开发的 USB 驱动器。该 USB 驱动器实施通信设备类 (CDC)、人机接口设备 (HID) 和海量存储设备 (MSC) 设备类。应用软件只能使用一个设备函数。

25.3 USB 驱动器函数

USB 设备驱动器 ROM API 包含以下模块：

- 通信设备类 (CDC) 函数驱动器
  - 通信设备类函数驱动器初始化参数数据结构 （表 510 “USBD_CDC_INIT_PARAM 类结构”）。
  - CDC 类 API 函数结构。该模块公开直接与 USB 设备控制器硬件交互的函数（表 509 “USBD_CDC_API 类结构”）。
- USB 内核层
  - struct (表 506 “ _WB_T 类结构”)
  - union (表 483 “ __WORD_BYTE 类结构”)
  - struct (表 484 “ _BM_T 类结构”)
  - struct (表 497 “ _REQUEST_TYPE 类结构”)
  - struct (表 504 “ _USB_SETUP_PACKET 类结构”)
  - struct (表 500 “ _USB_DEVICE_QUALIFIER_DESCRIPTOR 类结构”)
  - struct <tbd>USB 设备描述符
  - struct (表 500 “ _USB_DEVICE_QUALIFIER_DESCRIPTOR 类结构”)
  - struct <tbd>USB 配置描述符
  - struct (表 502 “ _USB_INTERFACE_DESCRIPTOR 类结构”)
  - struct <tbd>USB 端点描述符
  - struct (表 505 “ _USB_STRING_DESCRIPTOR 类结构”)
  - struct (表 498 “ _USB_COMMON_DESCRIPTOR 类结构”)
  - struct (表 503 “ _USB_OTHER_SPEED_CONFIGURATION 类结构”)
  - USB 描述符数据结构 （表 499 “ _USB_CORE_DESCS_T 类结构”）
  - USB 设备协议栈初始化参数数据结构 （表 508 “USBD_API_INIT_PARAM 类结构”）。
  - USB 设备协议栈内核 API 函数结构 （表 511 “USBD_CORE_API 类结构”）。
- 设备固件升级 (DFU) 类函数驱动器
  - DFU 描述符数据结构 （表 513 “USBD_DFU_INIT_PARAM 类结构”）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- DFU 类 API 函数结构。该模块公开直接与 USB 设备控制器硬件交互的函数（表 512 “USBD_DFU_API 类结构”）。

- HID 类函数驱动器
  - struct (表 492 “_HID_DESCRIPTOR 类结构”).
  - struct (表 494 “_HID_REPORT_T 类结构”).
  - USB 描述符数据结构 （表 515 “USBD_HID_INIT_PARAM 类结构”）。
  - HID 类 API 函数结构。该结构包含指向 HID 函数驱动器模块所公开的所有函数的指针 （表 516 “USBD_HW_API 类结构”）。
- USB 设备控制器驱动器
  - 硬件 API 函数结构。该模块公开直接与 USB 设备控制器硬件交互的函数 （表 516 “USBD_HW_API 类结构”）。
- 海量存储类 (MSC) 函数驱动器
  - 海量存储类函数驱动器初始化参数数据结构 （表 518）。
  - MSC类API函数结构。该模块公开直接与USB设备控制器硬件交互的函数（表517）。

25.4 调用 USB 设备驱动器

ROM 中的一个固定位置 （即， 0x1FFF 1FF8）包含指向 ROM 驱动器表的指针。 ROM 驱动器表包含一个指向 USB 驱动器表的指针。指向各种 USB 驱动器函数的指针都存储在该表中。可使用 C 结构调用 USB 驱动器函数。图 59 说明了用于访问片内 USB 驱动器的指针机制。

```
typedef struct USBD_API

{

const USBD_HW_API_T* hw;

const USBD_CORE_API_T* core;

const USBD_MSC_API_T* msc;

const USBD_DFU_API_T* dfu;

const USBD_HID_API_T* hid;

const USBD_CDC_API_T* cdc;

const uint32_t* reserved6;

const uint32_t version;

} USBD_API_T;
```


本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 59. USB 设备驱动器指针结构

25.5 USB API

25.5.1 __WORD_BYTE

表 483. __WORD_BYTE 类结构

成员	描述
W	uint16_tuint16_t __WORD_BYTE::W 进行 16 位访问的数据成员
WB	WB_TWB_T __WORD_BYTE::WB 进行 8 位访问的数据成员

25.5.2 _BM_T

表 484. _BM_T 类结构

成员	描述
Recipient	uint8_tuint8_t _BM_T::Recipient 收件人类型。
类型	uint8_tuint8_t _BM_T::Type 请求类型。
方向	uint8_tuint8_t _BM_T::Dir 方向类型。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

25.5.3 _CDC_ABSTRACT_CONTROL_MANAGEMENT_DESCRIPTOR

表 485. _CDC_ABSTRACT_CONTROL_MANAGEMENT_DESCRIPTOR 类结构

成员	描述
bFunctionLength	uint8_tuint8_t _CDC_ABSTRACT_CONTROL_MANAGEMENT_DESCRIPTOR::bFunctionLength
bDescriptorType	uint8_tuint8_t _CDC_ABSTRACT_CONTROL_MANAGEMENT_DESCRIPTOR::bDescriptorType
bDescriptorSubtype	uint8_tuint8_t _CDC_ABSTRACT_CONTROL_MANAGEMENT_DESCRIPTOR::bDescriptorSubtype
bmCapabilities	uint8_tuint8_t _CDC_ABSTRACT_CONTROL_MANAGEMENT_DESCRIPTOR::bmCapabilities

25.5.4 _CDC_CALL_MANAGEMENT_DESCRIPTOR

表 486. _CDC_CALL_MANAGEMENT_DESCRIPTOR 类结构

成员	描述
bFunctionLength	uint8_tuint8_t _CDC_CALL_MANAGEMENT_DESCRIPTOR::bFunctionLength
bDescriptorType	uint8_tuint8_t _CDC_CALL_MANAGEMENT_DESCRIPTOR::bDescriptorType
bDescriptorSubtype	uint8_tuint8_t _CDC_CALL_MANAGEMENT_DESCRIPTOR::bDescriptorSubtype
bmCapabilities	uint8_tuint8_t _CDC_CALL_MANAGEMENT_DESCRIPTOR::bmCapabilities
bDataInterface	uint8_tuint8_t _CDC_CALL_MANAGEMENT_DESCRIPTOR::bDataInterface

25.5.5 _CDC_HEADER_DESCRIPTOR

表 487. _CDC_HEADER_DESCRIPTOR 类结构

成员	描述
bFunctionLength	uint8_tuint8_t _CDC_HEADER_DESCRIPTOR::bFunctionLength
bDescriptorType	uint8_tuint8_t _CDC_HEADER_DESCRIPTOR::bDescriptorType
bDescriptorSubtype	uint8_tuint8_t _CDC_HEADER_DESCRIPTOR::bDescriptorSubtype
bcdCDC	uint16_tuint16_t _CDC_HEADER_DESCRIPTOR::bcdCDC

25.5.6 _CDC_LINE_CODING

表 488. _CDC_LINE_CODING 类结构

成员	描述
dwDTERate	uint32_tuint32_t _CDC_LINE_CODING::dwDTERate
bCharFormat	uint8_tuint8_t _CDC_LINE_CODING::bCharFormat
bParityType	uint8_tuint8_t _CDC_LINE_CODING::bParityType
bDataBits	uint8_tuint8_t _CDC_LINE_CODING::bDataBits

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

25.5.7 _CDC_UNION_1SLAVE_DESCRIPTOR

表 489. _CDC_UNION_1SLAVE_DESCRIPTOR 类结构

成员	描述
sUnion	CDC_UNION_DESCRIPTORCDC_UNION_DESCRIPTOR _CDC_UNION_1SLAVE_DESCRIPTOR::sUnion
bSlaveInterfaces	uint8_tuint8_t _CDC_UNION_1SLAVE_DESCRIPTOR::bSlaveInterfaces[1][1]

25.5.8 _CDC_UNION_DESCRIPTOR

表 490. _CDC_UNION_DESCRIPTOR 类结构

成员	描述
bFunctionLength	uint8_tuint8_t _CDC_UNION_DESCRIPTOR::bFunctionLength
bDescriptorType	uint8_tuint8_t _CDC_UNION_DESCRIPTOR::bDescriptorType
bDescriptorSubtype	uint8_tuint8_t _CDC_UNION_DESCRIPTOR::bDescriptorSubtype
bMasterInterface	uint8_tuint8_t _CDC_UNION_DESCRIPTOR::bMasterInterface

25.5.9 _DFU_STATUS

表 491. _DFU_STATUS 类结构

成员	描述
bStatus	uint8_tuint8_t _DFU_STATUS::bStatus
bwPollTimeout	uint8_tuint8_t _DFU_STATUS::bwPollTimeout[3][3]
bState	uint8_tuint8_t _DFU_STATUS::bState
iString	uint8_tuint8_t _DFU_STATUS::iString

25.5.10 _HID_DESCRIPTOR

HID 特定类 HID 描述符。

表 492. _HID_DESCRIPTOR 类结构

成员	描述
bLength	uint8_tuint8_t _HID_DESCRIPTOR::bLength 描述符的大小（字节）。
bDescriptorType	uint8_tuint8_t _HID_DESCRIPTOR::bDescriptorType HID 描述符的类型。
bcdHID	uint16_tuint16_t _HID_DESCRIPTOR::bcdHID HID 描述符和设备符合的 BCD 编码版。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 492. _HID_DESCRIPTOR 类结构 *（续）*

成员	描述
bCountryCode	uint8_tuint8_t _HID_DESCRIPTOR::bCountryCode 本地化设备的国家 / 地区代码，如果通用，则为 0。
bNumDescriptors	uint8_tuint8_t _HID_DESCRIPTOR::bNumDescriptors 接口的 HID 报告描述符总数。
DescriptorList	PRE_PACK struct POST_PACK _HID_DESCRIPTOR::_HID_DESCRIPTOR_LISTPRE_PACK struct POST_PACK _HID_DESCRIPTOR::_HID_DESCRIPTOR_LIST _HID_DESCRIPTOR::DescriptorList[1][1] 一个或多个描述符的数组

25.5.11 _HID_DESCRIPTOR::_HID_DESCRIPTOR_LIST

表 493. _HID_DESCRIPTOR::_HID_DESCRIPTOR_LIST 类结构

成员	描述
bDescriptorType	uint8_tuint8_t _HID_DESCRIPTOR::_HID_DESCRIPTOR_LIST::bDescriptorType HID 报告的类型。
wDescriptorLength	uint16_tuint16_t _HID_DESCRIPTOR::_HID_DESCRIPTOR_LIST::wDescriptorLength 相关 HID 报告描述符的长度 （字节）。

25.5.12 _HID_REPORT_T

HID 报告描述符数据结构。

表 494. _HID_REPORT_T 类结构

成员	描述
len	uint16_tuint16_t _HID_REPORT_T::len 报告描述符的大小 （字节）。
idle_time	uint8_tuint8_t _HID_REPORT_T::idle_time 协议栈使用该值响应指定报告 ID 的 Set_Idle 和 GET_Idle 请求。该字段的值指定了为指定报告 ID 生成两个报告的速率。例如，带两个输入报告的设备可指定报告 ID 1 的闲置率为 20 毫秒，报告 ID 2 的闲置率为 500 毫秒。
__pad	uint8_tuint8_t _HID_REPORT_T::__pad 填充空间。
desc	uint8_t *uint8_t* _HID_REPORT_T::desc 报告描述符。

25.5.13 _MSC_CBW

表 495. _MSC_CBW 类结构

成员	描述
dSignature	uint32_tuint32_t _MSC_CBW::dSignature
dTag	uint32_tuint32_t _MSC_CBW::dTag
dDataLength	uint32_tuint32_t _MSC_CBW::dDataLength
bmFlags	uint8_tuint8_t _MSC_CBW::bmFlags

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 495. _MSC_CBW 类结构 *（续）*

成员	描述
bLUN	uint8_tuint8_t _MSC_CBW::bLUN
bCBLength	uint8_tuint8_t _MSC_CBW::bCBLength
CB	uint8_tuint8_t _MSC_CBW::CB[16][16]

25.5.14 _MSC_CSW

表 496. _MSC_CSW 类结构

成员	描述
dSignature	uint32_tuint32_t _MSC_CSW::dSignature
dTag	uint32_tuint32_t _MSC_CSW::dTag
dDataResidue	uint32_tuint32_t _MSC_CSW::dDataResidue
bStatus	uint8_tuint8_t _MSC_CSW::bStatus

25.5.15 _REQUEST_TYPE

表 497. _REQUEST_TYPE 类结构

成员	描述
B	uint8_tuint8_t _REQUEST_TYPE::B 字节宽访问成员
BM	BM_TBM_T _REQUEST_TYPE::BM 位字段结构访问成员

25.5.16 _USB_COMMON_DESCRIPTOR

表 498. _USB_COMMON_DESCRIPTOR 类结构

成员	描述
bLength	uint8_tuint8_t _USB_COMMON_DESCRIPTOR::bLength 该描述符的大小 （字节）
bDescriptorType	uint8_tuint8_t _USB_COMMON_DESCRIPTOR::bDescriptorType 描述符类型

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

25.5.17 _USB_CORE_DESCS_T

USB 描述符数据结构。

表 499. _USB_CORE_DESCS_T 类结构

成员	描述
device_desc	uint8_t *uint8_t* _USB_CORE_DESCS_T::device_desc 指向 USB 设备描述符的指针
string_desc	uint8_t *uint8_t* _USB_CORE_DESCS_T::string_desc 指向 USB 字符串描述符数组的指针
full_speed_desc	uint8_t *uint8_t* _USB_CORE_DESCS_T::full_speed_desc 设备以全速模式运行时，指向 USB 设备配置描述符的指针。
high_speed_desc	uint8_t *uint8_t* _USB_CORE_DESCS_T::high_speed_desc 设备以高速模式运行时，指向 USB 设备配置描述符的指针。对于仅限于全速的实施，该指针应和 full_speed_desc 相同。
device_qualifier	uint8_t *uint8_t* _USB_CORE_DESCS_T::device_qualifier 指向 USB 设备门控描述符的指针。对于仅限于全速的实施，应将该指针设置为空 (0)。

25.5.18 _USB_DEVICE_QUALIFIER_DESCRIPTOR

表 500. _USB_DEVICE_QUALIFIER_DESCRIPTOR 类结构

成员	描述
bLength	uint8_tuint8_t _USB_DEVICE_QUALIFIER_DESCRIPTOR::bLength 描述符大小
bDescriptorType	uint8_tuint8_t _USB_DEVICE_QUALIFIER_DESCRIPTOR::bDescriptorType 设备门控类型
bcdUSB	uint16_tuint16_t _USB_DEVICE_QUALIFIER_DESCRIPTOR::bcdUSB USB 规范版本号 （如 V2.00 为 0x0200H）
bDeviceClass	uint8_tuint8_t _USB_DEVICE_QUALIFIER_DESCRIPTOR::bDeviceClass 类代码
bDeviceSubClass	uint8_tuint8_t _USB_DEVICE_QUALIFIER_DESCRIPTOR::bDeviceSubClass 子类代码
bDeviceProtocol	uint8_tuint8_t _USB_DEVICE_QUALIFIER_DESCRIPTOR::bDeviceProtocol 协议代码
bMaxPacketSize0	uint8_tuint8_t _USB_DEVICE_QUALIFIER_DESCRIPTOR::bMaxPacketSize0 其他速度下的最大数据包大小
bNumConfigurations	uint8_tuint8_t _USB_DEVICE_QUALIFIER_DESCRIPTOR::bNumConfigurations 其他速度配置数
bReserved	uint8_tuint8_t _USB_DEVICE_QUALIFIER_DESCRIPTOR::bReserved 留待将来使用，必须为 0

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

25.5.19 _USB_DFU_FUNC_DESCRIPTOR

表 501. _USB_DFU_FUNC_DESCRIPTOR 类结构

成员	描述
bLength	uint8_tuint8_t _USB_DFU_FUNC_DESCRIPTOR::bLength
bDescriptorType	uint8_tuint8_t _USB_DFU_FUNC_DESCRIPTOR::bDescriptorType
bmAttributes	uint8_tuint8_t _USB_DFU_FUNC_DESCRIPTOR::bmAttributes
wDetachTimeOut	uint16_tuint16_t _USB_DFU_FUNC_DESCRIPTOR::wDetachTimeOut
wTransferSize	uint16_tuint16_t _USB_DFU_FUNC_DESCRIPTOR::wTransferSize
bcdDFUVersion	uint16_tuint16_t _USB_DFU_FUNC_DESCRIPTOR::bcdDFUVersion

25.5.20 _USB_INTERFACE_DESCRIPTOR

表 502. _USB_INTERFACE_DESCRIPTOR 类结构

成员	描述
bLength	uint8_tuint8_t _USB_INTERFACE_DESCRIPTOR::bLength 该描述符的大小 （字节）
bDescriptorType	uint8_tuint8_t _USB_INTERFACE_DESCRIPTOR::bDescriptorType INTERFACE 描述符类型
bInterfaceNumber	uint8_tuint8_t _USB_INTERFACE_DESCRIPTOR::bInterfaceNumber 该接口的编号。以零为基准的值，用于标识该配置所支持的并行接口数组的索引。
bAlternateSetting	uint8_tuint8_t _USB_INTERFACE_DESCRIPTOR::bAlternateSetting 用于为在先前字段标识的接口选择备用设置的值
bNumEndpoints	uint8_tuint8_t _USB_INTERFACE_DESCRIPTOR::bNumEndpoints 该接口使用的端点数量 （不包括端点 0）。如果该值为 0，则该接口仅使用默认控制管道。
bInterfaceClass	uint8_tuint8_t _USB_INTERFACE_DESCRIPTOR::bInterfaceClass 类代码 （由 USB-IF 分配）。
bInterfaceSubClass	uint8_tuint8_t _USB_INTERFACE_DESCRIPTOR::bInterfaceSubClass 子类代码 （由 USB-IF 分配）。
bInterfaceProtocol	uint8_tuint8_t _USB_INTERFACE_DESCRIPTOR::bInterfaceProtocol 协议代码 （由 USB-IF 分配）。
iInterface	uint8_tuint8_t _USB_INTERFACE_DESCRIPTOR::iInterface 描述该接口的字符串描述符的索引

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

25.5.21 _USB_OTHER_SPEED_CONFIGURATION

表 503. _USB_OTHER_SPEED_CONFIGURATION 类结构

成员	描述
bLength	uint8_tuint8_t _USB_OTHER_SPEED_CONFIGURATION::bLength 描述符大小
bDescriptorType	uint8_tuint8_t _USB_OTHER_SPEED_CONFIGURATION::bDescriptorType Other_speed_Configuration Type
wTotalLength	uint16_tuint16_t _USB_OTHER_SPEED_CONFIGURATION::wTotalLength 返回数据的总长度
bNumInterfaces	uint8_tuint8_t _USB_OTHER_SPEED_CONFIGURATION::bNumInterfaces 该速度配置支持的接口数量
bConfigurationValue	uint8_tuint8_t _USB_OTHER_SPEED_CONFIGURATION::bConfigurationValue 用于选择配置的值
IConfiguration	uint8_tuint8_t _USB_OTHER_SPEED_CONFIGURATION::IConfiguration 字符串描述符索引
bmAttributes	uint8_tuint8_t _USB_OTHER_SPEED_CONFIGURATION::bmAttributes 与配置描述符相同
bMaxPower	uint8_tuint8_t _USB_OTHER_SPEED_CONFIGURATION::bMaxPower 与配置描述符相同

25.5.22 _USB_SETUP_PACKET

表 504. _USB_SETUP_PACKET 类结构

成员	描述
bmRequestType	REQUEST_TYPEREQUEST_TYPE _USB_SETUP_PACKET::bmRequestType 该位映射字段可标识特定请求的特性。 _BM_T.
bRequest	uint8_tuint8_t _USB_SETUP_PACKET::bRequest 该字段可指定特定请求。 bmRequestType 字段的类型位可修改该字段的含义。 USBD_REQUEST.
wValue	WORD_BYTEWORD_BYTE _USB_SETUP_PACKET::wValue 用于将参数传递给设备，特定于请求。
wIndex	WORD_BYTEWORD_BYTE _USB_SETUP_PACKET::wIndex 用于将参数传递给设备，特定于请求。 wIndex 字段常在请求中用于指定端点或接口。
wLength	uint16_tuint16_t _USB_SETUP_PACKET::wLength 该字段指定在控制传输的第二个阶段传输的数据长度。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

25.5.23 _USB_STRING_DESCRIPTOR

表 505. _USB_STRING_DESCRIPTOR 类结构

成员	描述
bLength	uint8_tuint8_t _USB_STRING_DESCRIPTOR::bLength 该描述符的大小（字节）
bDescriptorType	uint8_tuint8_t _USB_STRING_DESCRIPTOR::bDescriptorType STRING 描述符类型
bString	uint16_tuint16_t _USB_STRING_DESCRIPTOR::bString UNICODE 编码的字符串

25.5.24 _WB_T

表 506. _WB_T 类结构

成员	描述
L	uint8_tuint8_t _WB_T::L 低字节
H	uint8_tuint8_t _WB_T::H 高字节

25.5.25 USBD_API

主要的 USBD API 函数结构。该结构包含指向各种 USB 设备协议栈子模块函数表的指针。该结构作为主入口点，用于访问基于 ROM 的 USB 设备协议栈公开的各种方法（按子模块分组）。

表 507. USBD_API 类结构

成员	描述
hw	const USBD_HW_API_T *const USBD_HW_API_T* USBD_API::hw 指向函数表的指针，该函数表公开直接与 USB 设备协议栈内核层交互的函数。
core	const USBD_CORE_API_T *const USBD_CORE_API_T* USBD_API::core 指向函数表的指针，该函数表公开直接与 USB 设备控制器硬件交互的函数。
msc	const USBD_MSC_API_T *const USBD_MSC_API_T* USBD_API::msc 指向函数表的指针，该函数表公开 MSC 函数驱动器模块提供的函数。
dfu	const USBD_DFU_API_T *const USBD_DFU_API_T* USBD_API::dfu 指向函数表的指针，该函数表公开 DFU 函数驱动器模块提供的函数。
hid	const USBD_HID_API_T *const USBD_HID_API_T* USBD_API::hid 指向函数表的指针，该函数表公开 HID 函数驱动器模块提供的函数。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 507. USBD_API 类结构 *（续）*

成员	描述
cdc	const USBD_CDC_API_T *const USBD_CDC_API_T* USBD_API::cdc 指向函数表的指针，该函数表公开 CDC-ACM 函数驱动器模块提供的函数。
reserved6	const uint32_t *const uint32_t* USBD_API::reserved6 保留用于未来的函数驱动器模块。
version	const uint32_tconst uint32_t USBD_API::version USB ROM 协议栈的版本标识符。版本定义为 0x0CHDMhCC，其中每个半字节代表相应元件的版本号。CC - 7:0 - 8 位内核版本号 h - 11:8 - 4 位硬件接口版本号 M - 15:12 - 4 位 MSC 类模块版本号 D - 19:16 - 4 位 DFU 类模块版本号 H - 23:20 - 4 位 HID 类模块版本号 C - 27:24 - 4 位 CDC 类模块版本号 H - 31:28 - 4 位保留

25.5.26 USBD_API_INIT_PARAM

USB 设备协议栈初始化参数数据结构。

表 508. USBD_API_INIT_PARAM 类结构

成员	描述
usb_reg_base	uint32_tuint32_t USBD_API_INIT_PARAM::usb_reg_base USB 设备控制器的寄存器基址。
mem_base	uint32_tuint32_t USBD_API_INIT_PARAM::mem_base 协议栈从中分配数据和缓冲的基本存储器位置。 **注：**此字段中设置的存储器地址应可由 USB DMA 控制器访问。该值还应在 2048 字节边界对齐。
mem_size	uint32_tuint32_t USBD_API_INIT_PARAM::mem_size 协议栈可使用的存储器缓冲区大小。 **注：**mem_size 应大于 USBD_HW_API::GetMemSize() 例程返回的大小。
max_num_ep	uint8_tuint8_t USBD_API_INIT_PARAM::max_num_ep USB 设备控制器实例支持的最大端点数 （由 指定
pad0	uint8_tuint8_t USBD_API_INIT_PARAM::pad0[3][3]
USB_Reset_Event	USB_CB_TUSB_CB_T USBD_API_INIT_PARAM::USB_Reset_Event 用于 USB 接口复位的事件。当 USB 主机请求设备复位其接口时，启动该事件。组别自动配置控制端点后，启动该事件。 **注：**该事件从 USB_ISR 上下文调用，因此时间非常关键。如果此回调中具有延迟，将使设备无法正确枚举或正常工作。
USB_Suspend_Event	USB_CB_TUSB_CB_T USBD_API_INIT_PARAM::USB_Suspend_Event 用于 USB 挂起的事件。当 USB 主机通过中止将帧起始脉冲发送到设备的操作而挂起设备时，启动该事件。通常情况下，挂接该事件的目的是为了将设备移动到低功率状态，直到主机唤醒设备为止。 **注：**该事件从 USB_ISR 上下文调用，因此时间非常关键。如果此调用中具有延迟，会引发其他系统问题。
USB_Resume_Event	USB_CB_TUSB_CB_T USBD_API_INIT_PARAM::USB_Resume_Event 用于 USB 唤醒或恢复的事件。当 USB 设备接口挂起，且主机通过提供帧起始脉冲唤醒设备时，启动该事件。通常情况下，挂接该事件的目的是为了让用户应用脱离低功率状态，并返回到正常操作模式。 **注：**该事件从 USB_ISR 上下文调用，因此时间非常关键。如果此调用中具有延迟，会引发其他系统问题。
reserved_sbz	USB_CB_TUSB_CB_T USBD_API_INIT_PARAM::reserved_sbz 保留参数应设为 0。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 508. USBD_API_INIT_PARAM 类结构 *（续）*

成员	描述
USB_SOF_Event	USB_CB_TUSB_CB_T USBD_API_INIT_PARAM::USB_SOF_Event 用于 USB 帧起始检测的事件（使能时）。每个 USB 帧起始时启动该事件（全速模式下，每毫秒一次；高速模式下，每 125 毫秒一次），且和 USB 总线同步。 该事件的处理时间要求严格，每毫秒运行一次（全速模式下），因此，长处理器将明显降低设备性能。仅应在需要时使能该事件以减少设备唤醒次数。 通常情况下，该事件未激活 - 必须手动使能并通过 USB 中断寄存器禁用。 **注：**通常情况下，该事件未激活 - 必须手动使能并通过 USB 中断寄存器禁用。
USB_WakeUpCfg	USB_PARAM_CB_TUSB_PARAM_CB_T USBD_API_INIT_PARAM::USB_WakeUpCfg 用于远程唤醒配置的事件（使能时）。当 USB 主机请求设备为其自己配置远程唤醒功能时，启动该事件。进入低功率状态前， USB 主机将该请求发送至设备（在设备描述符中报告远程唤醒功能）。如果设备拥有可启动远程唤醒事件的特殊板上电路，应用层应实施该回调。此外，应用还可使用本回调区分后续挂起事件是由电缆拔出还是主机挂起请求引起的。仅在收到此回调，且主机使能远程唤醒功能后，设备才可唤醒主机。要发送远程唤醒信号，设备必须通过调用 usapi.hw->WakeUp() 例程在主机上生成恢复信号。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. param1 = 为 0 时 - 清除唤醒配置，为 1 时 - 使能唤醒配置。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。
USB_Power_Event	USB_PARAM_CB_TUSB_PARAM_CB_T USBD_API_INIT_PARAM::USB_Power_Event 保留参数应设为 0。
USB_Error_Event	USB_PARAM_CB_TUSB_PARAM_CB_T USBD_API_INIT_PARAM::USB_Error_Event 用于错误条件的事件。当 USB 设备控制器检测到系统内的错误条件时，启动该事件。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. param1 = USB 设备中断状态寄存器。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。
USB_Configure_Event	USB_CB_TUSB_CB_T USBD_API_INIT_PARAM::USB_Configure_Event 用于已更改的 USB 配置编号的事件。当 USB 主机更改选定配置编号时，启动该事件。从主机接收到配置更改请求后，协议栈在调用该回调函数前，使能 / 配置新配置所需的端点。 **注：**该事件从 USB_ISR 上下文调用，因此时间非常关键。如果此回调中具有延迟，将使设备无法正确枚举或正常工作。
USB_Interface_Event	USB_CB_TUSB_CB_T USBD_API_INIT_PARAM::USB_Interface_Event 用于已更改的 USB 接口设置的事件。当 USB 主机将接口设置更改为备用接口设置之一时，启动该事件。从主机接收到接口更改请求后，协议栈在调用该回调函数前，使能 / 配置新备用接口设置所需的端点。 **注：**该事件从 USB_ISR 上下文调用，因此时间非常关键。如果此回调中具有延迟，将使设备无法正确枚举或正常工作。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 508. USBD_API_INIT_PARAM 类结构 *（续）*

成员	描述
USB_Feature_Event	USB_CB_TUSB_CB_T USBD_API_INIT_PARAM::USB_Feature_Event 用于已更改的 USB 功能的事件。当 USB 主机发送设置 / 清除功能请求时，启动该事件。协议栈仅处理针对 USB_FEATURE_REMOTE_WAKEUP、 USB_FEATURE_TEST_MODE 和 USB_FEATURE_ENDPOINT_STALL 功能的此类请求。从主机接收到功能请求后，协议栈将相应地处理请求，然后调用此回调函数。 **注：**该事件从 USB_ISR 上下文调用，因此时间非常关键。如果此回调中具有延迟，将使设备无法正确枚举或正常工作。
virt_to_phys	uint32_t(*uint32_t(* USBD_API_INIT_PARAM::virt_to_phys)(void *vaddr))(void *vaddr) 留待将来使用的参数。应设为 0。
cache_flush	void(*void(* USBD_API_INIT_PARAM::cache_flush)(uint32_t *start_adr, uint32_t *end_adr))(uint32_t *start_adr, uint32_t *end_adr) 留待将来使用的参数。应设为 0。

25.5.27 USBD_CDC_API

CDC 类 API 函数结构。该模块公开直接与 USB 设备控制器硬件交互的函数。

表 509. USBD_CDC_API 类结构

成员	描述
GetMemSize	uint32_t(*uint32_t USBD_CDC_API::GetMemSize)(USBD_CDC_INIT_PARAM_T *param) 用于确定 CDC 函数驱动器模块所要求的存储器的函数。 在调用 pUsbApi->CDC->Init() 前，应用层调用该函数以分配 CDC 函数驱动器模块所使用的存储器。应用程序应分配可由 USB 控制器 /DMA 控制器访问的存储器。 **注：**一些存储器区域无法由所有总线主机访问。 参数： 1. param = 含 CDC 函数驱动器模块初始化参数的结构。 返回： 返回需要的存储器大小 （字节）。
init	ErrorCode_t(*ErrorCode_t USBD_CDC_API::init)(USBD_HANDLE_T hUsb, USBD_CDC_INIT_PARAM_T *param, USBD_HANDLE_T *phCDC) 初始化 CDC 函数驱动器模块的函数。 应用层调用该函数以初始化 CDC 函数驱动器模块。 hUsb：USB 设备协议栈的句柄。 param：含 CDC 函数驱动器模块初始化参数的结构。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. param = 含 CDC 函数驱动器模块初始化参数的结构。 返回： 返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功 2. ERR_USBD_BAD_MEM_BUF = 传递的存储器缓冲区不以 4 字节对齐或小于要求。 3. ERR_API_INVALID_PARAM2 = CDC_Write()、 CDC_Read() 或 CDC_Verify() 回调未定义。 4. ERR_USBD_BAD_INTF_DESC = 传递错误的接口描述符。 5. ERR_USBD_BAD_EP_DESC = 传递错误的端点描述符。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 509. USBD_CDC_API 类结构 *（续）*

成员	描述
SendNotification	ErrorCode_t(*ErrorCode_t USBD_CDC_API::SendNotification)(USBD_HANDLE_T hCdc, uint8_t bNotification, uint16_t data) 初始化 CDC 函数驱动器模块的函数。 应用层调用该函数以初始化 CDC 函数驱动器模块。 hUsb：USB 设备协议栈的句柄。 param：含 CDC 函数驱动器模块初始化参数的结构。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. param = 含 CDC 函数驱动器模块初始化参数的结构。 返回： 返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功 2. ERR_USBD_BAD_MEM_BUF = 传递的存储器缓冲区不以 4 字节对齐或小于要求。 3. ERR_API_INVALID_PARAM2 = CDC_Write()、 CDC_Read() 或 CDC_Verify() 回调未定义。 4. ERR_USBD_BAD_INTF_DESC = 传递错误的接口描述符。 5. ERR_USBD_BAD_EP_DESC = 传递错误的端点描述符。

25.5.28 USBD_CDC_INIT_PARAM

通信设备类函数驱动器初始化参数数据结构。

表 510. USBD_CDC_INIT_PARAM 类结构

成员	描述
mem_base	uint32_tuint32_t USBD_CDC_INIT_PARAM::mem_base 协议栈从中分配数据和缓冲的基本存储器位置。 **注：**此字段中设置的存储器地址应可由 USB DMA 控制器访问。该值还应在 4 字节边界对齐。
mem_size	uint32_tuint32_t USBD_CDC_INIT_PARAM::mem_size 协议栈可使用的存储器缓冲区大小。 **注：**mem_size 应大于 USBD_CDC_API::GetMemSize() 例程返回的大小。
cif_intf_desc	uint8_t *uint8_t* USBD_CDC_INIT_PARAM::cif_intf_desc 指向描述符数组内的控制接口描述符的指针 （
dif_intf_desc	uint8_t *uint8_t* USBD_CDC_INIT_PARAM::dif_intf_desc 指向描述符数组内的数据接口描述符的指针 （

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 510. USBD_CDC_INIT_PARAM 类结构 *（续）*

成员	描述
CIC_GetRequest	ErrorCode_t(*ErrorCode_t(* USBD_CDC_INIT_PARAM::CIC_GetRequest)(USBD_HANDLE_T hHid, USB_SETUP_PACKET *pSetup, uint8_t **pBuffer, uint16_t *length))(USBD_HANDLE_T hHid, USB_SETUP_PACKET *pSetup, uint8_t **pBuffer, uint16_t *length) 通信接口类特定获取请求回调函数。 该函数由应用软件提供。当主机发送 CIC 管理元素获取请求时，调用该函数。设置数据包 (hCdc：CDC 函数驱动器的句柄。 pSetup：指向从主机接收的设置数据包的指针。 pBuffer：指向含请求数据的数据缓冲区指针的指针。指向指针的指针用于实施 0 复制缓冲区。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。 length：发回主机的数据量。 参数： 1. hCdc = CDC 函数驱动器的句柄。 2. pSetup = 指向从主机接收的设置数据包的指针。 3. pBuffer = 指向含请求数据的数据缓冲区指针的指针。指向指针的指针用于实施 0 复制缓冲区。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。 4. length = 发回主机的数据量。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。
CIC_SetRequest	ErrorCode_t(*ErrorCode_t(* USBD_CDC_INIT_PARAM::CIC_SetRequest)(USBD_HANDLE_T hCdc, USB_SETUP_PACKET *pSetup, uint8_t **pBuffer, uint16_t length))(USBD_HANDLE_T hCdc, USB_SETUP_PACKET *pSetup, uint8_t **pBuffer, uint16_t length) 通信接口类特定设置请求回调函数。 该函数由应用软件提供。当主机发送 CIC 管理元素请求时，调用该函数。设置数据包 (hCdc：CDC 函数驱动器的句柄。 pSetup：指向从主机接收的设置数据包的指针。 pBuffer：指向含请求数据的数据缓冲区指针的指针。指向指针的指针用于实施 0 复制缓冲区。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。 length：复制到目标缓冲区的数据量。 参数： 1. hCdc = CDC 函数驱动器的句柄。 2. pSetup = 指向从主机接收的设置数据包的指针。 3. pBuffer = 指向含请求数据的数据缓冲区指针的指针。指向指针的指针用于实施 0 复制缓冲区。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。 4. length = 复制到目标缓冲区的数据量。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 510. USBD_CDC_INIT_PARAM 类结构 *（续）*

成员	描述
CDC_BulkIN_Hdlr	ErrorCode_t(*ErrorCode_t(* USBD_CDC_INIT_PARAM::CDC_BulkIN_Hdlr)(USBD_HANDLE_T hUsb, void *data, uint32_t event))(USBD_HANDLE_T hUsb, void *data, uint32_t event) 通信设备类特定 BULK IN 端点处理器。 应用软件应提供 BULK IN 端点处理器。应用程序应根据描述符中设置的通信协议类型传输数据。 **注：** 参数： 1. hUsb = USB 设备协议栈的句柄。 2. data = 指向协议栈调用回调函数时所传递数据的指针。 3. event = 端点事件的类型。更多详情请参见 USBD_EVENT_T。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。
CDC_BulkOUT_Hdlr	ErrorCode_t(*ErrorCode_t(* USBD_CDC_INIT_PARAM::CDC_BulkOUT_Hdlr)(USBD_HANDLE_T hUsb, void *data, uint32_t event))(USBD_HANDLE_T hUsb, void *data, uint32_t event) 通信设备类特定 BULK OUT 端点处理器。 应用软件应提供 BULK OUT 端点处理器。应用程序应根据描述符中设置的通信协议类型传输数据。 **注：** 参数： 1. hUsb = USB 设备协议栈的句柄。 2. data = 指向协议栈调用回调函数时所传递数据的指针。 3. event = 端点事件的类型。更多详情请参见 USBD_EVENT_T。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。
SendEncpsCmd	ErrorCode_t(*ErrorCode_t(* USBD_CDC_INIT_PARAM::SendEncpsCmd)(USBD_HANDLE_T hCDC, uint8_t *buffer, uint16_t len))(USBD_HANDLE_T hCDC, uint8_t *buffer, uint16_t len)
GetEncpsResp	ErrorCode_t(*ErrorCode_t(* USBD_CDC_INIT_PARAM::GetEncpsResp)(USBD_HANDLE_T hCDC, uint8_t **buffer, uint16_t *len))(USBD_HANDLE_T hCDC, uint8_t **buffer, uint16_t *len)
SetCommFeature	ErrorCode_t(*ErrorCode_t(* USBD_CDC_INIT_PARAM::SetCommFeature)(USBD_HANDLE_T hCDC, uint16_t feature, uint8_t *buffer, uint16_t len))(USBD_HANDLE_T hCDC, uint16_t feature, uint8_t *buffer, uint16_t len)
GetCommFeature	ErrorCode_t(*ErrorCode_t(* USBD_CDC_INIT_PARAM::GetCommFeature)(USBD_HANDLE_T hCDC, uint16_t feature, uint8_t **pBuffer, uint16_t *len))(USBD_HANDLE_T hCDC, uint16_t feature, uint8_t **pBuffer, uint16_t *len)
ClrCommFeature	ErrorCode_t(*ErrorCode_t(* USBD_CDC_INIT_PARAM::ClrCommFeature)(USBD_HANDLE_T hCDC, uint16_t feature))(USBD_HANDLE_T hCDC, uint16_t feature)
SetCtrlLineState	ErrorCode_t(*ErrorCode_t(* USBD_CDC_INIT_PARAM::SetCtrlLineState)(USBD_HANDLE_T hCDC, uint16_t state))(USBD_HANDLE_T hCDC, uint16_t state)

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 510. USBD_CDC_INIT_PARAM 类结构 *（续）*

成员	描述
SendBreak	ErrorCode_t(*ErrorCode_t(* USBD_CDC_INIT_PARAM::SendBreak)(USBD_HANDLE_T hCDC, uint16_t mstime))(USBD_HANDLE_T hCDC, uint16_t mstime)
SetLineCode	ErrorCode_t(*ErrorCode_t(* USBD_CDC_INIT_PARAM::SetLineCode)(USBD_HANDLE_T hCDC, CDC_LINE_CODING *line_coding))(USBD_HANDLE_T hCDC, CDC_LINE_CODING *line_coding)
CDC_InterruptEP_Hdlr	ErrorCode_t(*ErrorCode_t(* USBD_CDC_INIT_PARAM::CDC_InterruptEP_Hdlr)(USBD_HANDLE_T hUsb, void *data, uint32_t event))(USBD_HANDLE_T hUsb, void *data, uint32_t event) 可选通信设备类特定 INTERRUPT IN 端点处理器。 应用软件应提供 INT IN 端点处理器。应用程序应根据描述符中设置的通信协议类型传输数据。 **注：** 参数： 1. hUsb = USB 设备协议栈的句柄。 2. data = 指向协议栈调用回调函数时所传递数据的指针。 3. event = 端点事件的类型。更多详情请参见 USBD_EVENT_T。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。
CDC_Ep0_Hdlr	ErrorCode_t(*ErrorCode_t(* USBD_CDC_INIT_PARAM::CDC_Ep0_Hdlr)(USBD_HANDLE_T hUsb, void *data, uint32_t event))(USBD_HANDLE_T hUsb, void *data, uint32_t event) 代替默认 CDC 类处理器的可选用户可覆盖函数。 应用软件提供处理器函数地址作为参数结构的此数据成员，从而用自己的处理器覆盖默认 EP0 类处理器。首选默认处理器的应用应在调用 USBD_CDC_API::Init() 前将该数据成员置零。 **注：** 参数： 1. hUsb = USB 设备协议栈的句柄。 2. data = 指向协议栈调用回调函数时所传递数据的指针。 3. event = 端点事件的类型。更多详情请参见 USBD_EVENT_T。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

25.5.29 USBD_CORE_API

USBD 协议栈内核 API 函数结构。

表 511. USBD_CORE_API 类结构

成员	描述
RegisterClassHandler	ErrorCode_t(*ErrorCode_t USBD_CORE_API::RegisterClassHandler)(USBD_HANDLE_T hUsb, USB_EP_HANDLER_T pfn, void *data) 向 USB 设备协议栈注册类特定 EP0 事件处理器的函数。 应用层在必须注册自定义类 EP0 处理器时，将使用该函数。该协议栈在进行事件的默认处理前对任何 EP0 事件调用所有注册的类处理器。这就提供了类处理器来实施特定类的请求处理器，并覆盖面向接口的特定事件的默认协议栈处理。有关如何实施回调函数的更多详情，请参见 USB_EP_HANDLER_T。应用层还可使用该函数注册响应供应商特定请求的 EP0 处理器。 hUsb：USB 设备协议栈的句柄。 pfn：特定类的 EP0 处理器函数。 data：指向协议栈调用回调函数时所传递数据的指针。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. pfn = 特定类的 EP0 处理器函数。 3. data = 指向协议栈调用回调函数时所传递数据的指针。 返回： 返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功 2. ERR_USBD_TOO_MANY_CLASS_HDLR(0x0004000c) = 注册的类处理器数量大于协议栈允许的处理器数量。
RegisterEpHandler	ErrorCode_t(*ErrorCode_t USBD_CORE_API::RegisterEpHandler)(USBD_HANDLE_T hUsb, uint32_t ep_index, USB_EP_HANDLER_T pfn, void *data) 向 USB 设备协议栈注册所请求端点的中断 / 事件处理器的函数。 应用层使用该函数注册自定义类 EP0 处理器。该协议栈在进行事件的默认处理前对任何 EP0 事件调用所有注册的类处理器。这就提供了类处理器来实施特定类的请求处理器，并覆盖面向接口的特定事件的默认协议栈处理。有关如何实施回调函数的更多详情，请参见 USB_EP_HANDLER_T。 hUsb：USB 设备协议栈的句柄。 ep_index：特定类的 EP0 处理器函数。 pfn：特定类的 EP0 处理器函数。 data：指向协议栈调用回调函数时所传递数据的指针。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. ep_index = 特定类的 EP0 处理器函数。 3. pfn = 特定类的 EP0 处理器函数。 4. data = 指向协议栈调用回调函数时所传递数据的指针。 返回： 返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功 2. ERR_USBD_TOO_MANY_CLASS_HDLR(0x0004000c) = 端点处理器太多。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 511. USBD_CORE_API 类结构 *（续）*

成员	描述
SetupStage	void(*void USBD_CORE_API::SetupStage)(USBD_HANDLE_T hUsb) 在设置状态设置 EP0 状态机的函数。 该函数由 USB 协议栈和应用层调用，以在设置状态设置 EP0 状态机。该函数会将从 USB 主机接收的设置包读入协议栈缓存。 **注：**该接口提供给用户以在 （当前协议栈未处理的）其他情况下调用此函数。在多数用户应用中，该函数未直接调用。该函数还可由用户用来通过协议栈公开的回调接口，选择性地修改 USB 设备协议栈的标准处理器。 参数： 1. hUsb = USB 设备协议栈的句柄。 返回： 无任何内容。
DataInStage	void(*void USBD_CORE_API::DataInStage)(USBD_HANDLE_T hUsb) 在 data_in 状态设置 EP0 状态机的函数。 该函数由 USB 协议栈或应用层调用，以在 data_in 状态设置 EP0 状态机。该函数会将 EP0Data 缓冲区中存在的数据写入 EP0 FIFO，以发送至主机。 **注：**该接口提供给用户以在 （当前协议栈未处理的）其他情况下调用此函数。在多数用户应用中，该函数未直接调用。该函数还可由用户用来通过协议栈公开的回调接口，选择性地修改 USB 设备协议栈的标准处理器。 参数： 1. hUsb = USB 设备协议栈的句柄。 返回： 无任何内容。
DataOutStage	void(*void USBD_CORE_API::DataOutStage)(USBD_HANDLE_T hUsb) 在 data_out 状态设置 EP0 状态机的函数。 该函数由 USB 协议栈或应用层调用，以在 data_out 状态设置 EP0 状态机。该函数会将从 USB 主机接收的控制数据 （EP0 OUT 数据包）读入 EP0Data 缓冲区。 **注：**该接口提供给用户以在 （当前协议栈未处理的）其他情况下调用此函数。在多数用户应用中，该函数未直接调用。该函数还可由用户用来通过协议栈公开的回调接口，选择性地修改 USB 设备协议栈的标准处理器。 参数： 1. hUsb = USB 设备协议栈的句柄。 返回： 无任何内容。
StatusInStage	void(*void USBD_CORE_API::StatusInStage)(USBD_HANDLE_T hUsb) 在 status_in 状态设置 EP0 状态机的函数。 该函数由 USB 协议栈或应用层调用，以在 status_in 状态设置 EP0 状态机。该函数会将 EP0 上的零长度 IN 数据包发送至主机，表示正状态。 **注：**该接口提供给用户以在 （当前协议栈未处理的）其他情况下调用此函数。在多数用户应用中，该函数未直接调用。该函数还可由用户用来通过协议栈公开的回调接口，选择性地修改 USB 设备协议栈的标准处理器。 参数： 1. hUsb = USB 设备协议栈的句柄。 返回： 无任何内容。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 511. USBD_CORE_API 类结构 *（续）*

成员	描述
StatusOutStage	void(*void USBD_CORE_API::StatusOutStage)(USBD_HANDLE_T hUsb) 在 status_out 状态设置 EP0 状态机的函数。 该函数由 USB 协议栈或应用层调用，以在 status_out 状态设置 EP0 状态机。该函数会将从 USB 主机接收的零长度 OUT 数据包读到 EP0。 **注：**该接口提供给用户以在 （当前协议栈未处理的）其他情况下调用此函数。在多数用户应用中，该函数未直接调用。该函数还可由用户用来通过协议栈公开的回调接口，选择性地修改 USB 设备协议栈的标准处理器。 参数： 1. hUsb = USB 设备协议栈的句柄。 返回： 无任何内容。
StallEp0	void(*void USBD_CORE_API::StallEp0)(USBD_HANDLE_T hUsb) 用于在停止状态设置 EP0 状态机的函数。 该函数由 USB 协议栈和应用层调用，以在 EP0 端点产生 STALL 信号。该函数还将复位 EP0Data 缓冲区。 **注：**该接口提供给用户以在 （当前协议栈未处理的）其他情况下调用此函数。在多数用户应用中，该函数未直接调用。该函数还可由用户用来通过协议栈公开的回调接口，选择性地修改 USB 设备协议栈的标准处理器。 参数： 1. hUsb = USB 设备协议栈的句柄。 返回： 无任何内容。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

25.5.30 USBD_DFU_API

DFU 类 API 函数结构。该模块公开直接与 USB 设备控制器硬件交互的函数。

表 512. USBD_DFU_API 类结构

成员	描述
GetMemSize	uint32_t(*uint32_t USBD_DFU_API::GetMemSize)(USBD_DFU_INIT_PARAM_T *param) 用于确定 DFU 函数驱动器模块所要求的存储器的函数。 在调用 pUsbApi->dfu->Init() 前，应用层调用该函数以分配 DFU 函数驱动器模块所使用的存储器。应用程序应分配可由 USB 控制器 /DMA 控制器访问的存储器。 **注：**一些存储器区域无法由所有总线主机访问。 参数： 1. param = 含 DFU 函数驱动器模块初始化参数的结构。 返回： 返回需要的存储器大小 （字节）。
init	ErrorCode_t(*ErrorCode_t USBD_DFU_API::init)(USBD_HANDLE_T hUsb, USBD_DFU_INIT_PARAM_T *param, uint32_t init_state) 初始化 DFU 函数驱动器模块的函数。 应用层调用该函数以初始化 DFU 函数驱动器模块。 hUsb：USB 设备协议栈的句柄。 param：含 DFU 函数驱动器模块初始化参数的结构。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. param = 含 DFU 函数驱动器模块初始化参数的结构。 返回： 返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功 2. ERR_USBD_BAD_MEM_BUF = 传递的存储器缓冲区不以 4 字节对齐或小于要求。 3. ERR_API_INVALID_PARAM2 = DFU_Write()、 DFU_Done() 或 DFU_Read() 回调未定义。 4. ERR_USBD_BAD_DESC = USB_DFU_DESCRIPTOR_TYPE 未紧接在接口描述符之后定义。描述符中的 wTransferSize 与 param->wTransferSize 中传递的值不匹配。在 DFU 描述符中设置了 USB_DFU_WILL_DETACH 时，未定义 DFU_Detach()。 5. ERR_USBD_BAD_INTF_DESC = 传递错误的接口描述符。

25.5.31 USBD_DFU_INIT_PARAM

USB 描述符数据结构。

表 513. USBD_DFU_INIT_PARAM 类结构

成员	描述
mem_base	uint32_tuint32_t USBD_DFU_INIT_PARAM::mem_base 协议栈从中分配数据和缓冲的基本存储器位置。 **注：**此字段中设置的存储器地址应可由 USB DMA 控制器访问。该值还应在 4 字节边界对齐。
mem_size	uint32_tuint32_t USBD_DFU_INIT_PARAM::mem_size 协议栈可使用的存储器缓冲区大小。 **注：**mem_size 应大于 USBD_DFU_API::GetMemSize() 例程返回的大小。
wTransferSize	uint16_tuint16_t USBD_DFU_INIT_PARAM::wTransferSize DFU 传输块大小 （字节数）。该值应匹配作为描述符数组一部分的 DFU 描述符中设置的值 （

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 513. USBD_DFU_INIT_PARAM 类结构 *（续）*

成员	描述
管脚	uint16_tuint16_t USBD_DFU_INIT_PARAM::pad
intf_desc	uint8_t *uint8_t* USBD_DFU_INIT_PARAM::intf_desc 指向描述符数组内的 DFU 接口描述符的指针 （
DFU_Write	uint8_t(*uint8_t(* USBD_DFU_INIT_PARAM::DFU_Write)(uint32_t block_num, uint8_t **src, uint32_t length, uint8_t *bwPollTimeout))(uint32_t block_num, uint8_t **src, uint32_t length, uint8_t *bwPollTimeout) DFU 写回调函数。 该函数由应用软件提供。主机发送写命令时调用此函数。对于使用 0 复制缓冲机制的应用，该函数首次采用 调用 block_num：目标起始地址。 src：指向数据源指针的指针。指向指针的指针用于实施 0 复制缓冲区。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。bwPollTimeout：指向 3 字节缓冲区的指针，回调实施者应在其中填充主机发送后续 DFU_GETSTATUS 请求前应等待的最短时间（毫秒）。length：要写入的字节数。 参数： 1. block_num = 目标起始地址。 2. src = 指向数据源指针的指针。指向指针的指针用于实施 0 复制缓冲区。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。 3. bwPollTimeout = 指向 3 字节缓冲区的指针，回调实施者应在其中填充主机在发送后续 DFU_GETSTATUS 请求前应等待的最短时间 （毫秒）。 4. length = 要写入的字节数。 返回： 返回 mw_usbd_dfu.h 中定义的 DFU_STATUS_ 值。
DFU_Read	uint32_t(*uint32_t(* USBD_DFU_INIT_PARAM::DFU_Read)(uint32_t block_num, uint8_t **dst, uint32_t length))(uint32_t block_num, uint8_t **dst, uint32_t length) DFU 读回调函数。 该函数由应用软件提供。主机发送读命令时调用此函数。 block_num：目标起始地址。 dst：指向数据源指针的指针。指向指针的指针用于实施 0 复制缓冲区。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。 length：复制到目标缓冲区的数据量。 参数： 1. block_num = 目标起始地址。 2. dst = 指向数据源指针的指针。指向指针的指针用于实施 0 复制缓冲区。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。 3. length = 复制到目标缓冲区的数据量。 返回： 返回 mw_usbd_dfu.h 中定义的 DFU_STATUS_ 值。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 513. USBD_DFU_INIT_PARAM 类结构 *（续）*

成员	描述
DFU_Done	void(*void(* USBD_DFU_INIT_PARAM::DFU_Done)(void))(void) DFU 完成回调函数。 该函数由应用软件提供。下载完成后调用该函数。 无任何内容。 返回： 无任何内容。
DFU_Detach	void(*void(* USBD_DFU_INIT_PARAM::DFU_Detach)(USBD_HANDLE_T hUsb))(USBD_HANDLE_T hUsb) DFU 分离回调函数。 该函数由应用软件提供。接收 USB_REQ_DFU_DETACH 后调用该函数。在 DFU 描述符中设置 USB_DFU_WILL_DETACH 位的应用应定义该函数。作为该函数的一部分，应用可调用 Connect() 例程以断开连接并和主机重新连接。由于 USB 复位只能由 Windows 主机上的内核驱动器调用，依靠基于 WinUSB 的主机的应用应使用该功能。实施该功能后，主机不需要发出重设，而是由设备通过连接和断开连接程序自动执行。 hUsbHandle DFU 控制结构。 参数： 1. hUsb = 处理 DFU 控制结构。 返回： 无任何内容。
DFU_Ep0_Hdlr	ErrorCode_t(*ErrorCode_t(* USBD_DFU_INIT_PARAM::DFU_Ep0_Hdlr)(USBD_HANDLE_T hUsb, void *data, uint32_t event))(USBD_HANDLE_T hUsb, void *data, uint32_t event) 代替默认 DFU 类处理器的可选用户可覆盖函数。 应用软件提供处理器函数地址作为参数结构的此数据成员，从而用自己的处理器覆盖默认 EP0 类处理器。首选默认处理器的应用应在调用 USBD_DFU_API::Init() 前将该数据成员置零。 **注：** 参数： 1. hUsb = USB 设备协议栈的句柄。 2. data = 指向协议栈调用回调函数时所传递数据的指针。 3. event = 端点事件的类型。更多详情请参见 USBD_EVENT_T。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

25.5.32 USBD_HID_API

HID 类 API 函数结构。该结构包含指向 HID 函数驱动器模块所公开的所有函数的指针。

表 514. USBD_HID_API 类结构

成员	描述
GetMemSize	uint32_t(*uint32_t USBD_HID_API::GetMemSize)(USBD_HID_INIT_PARAM_T *param) 用于确定 HID 函数驱动器模块所要求的存储器的函数。 在调用 pUsbApi->hid->Init() 前，应用层调用该函数以分配 HID 函数驱动器模块所使用的存储器。应用程序应分配可由 USB 控制器 /DMA 控制器访问的存储器。 **注：**一些存储器区域无法由所有总线主机访问。 参数： 1. param = 含 HID 函数驱动器模块初始化参数的结构。 返回： 返回需要的存储器大小 （字节）。
init	ErrorCode_t(*ErrorCode_t USBD_HID_API::init)(USBD_HANDLE_T hUsb, USBD_HID_INIT_PARAM_T *param) 初始化 HID 函数驱动器模块的函数。 应用层调用该函数以初始化 HID 函数驱动器模块。成功初始化后，该函数将返回已传递 param 结构中 HID 函数驱动器模块的一个句柄。 hUsb：USB 设备协议栈的句柄。 param：含 HID 函数驱动器模块初始化参数的结构。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. param = 含 HID 函数驱动器模块初始化参数的结构。 返回： 返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功 2. ERR_USBD_BAD_MEM_BUF = 传递的存储器缓冲区不以 4 字节对齐或小于要求。 3. ERR_API_INVALID_PARAM2 = HID_GetReport() 或 HID_SetReport() 回调未定义。 4. ERR_USBD_BAD_DESC = HID_HID_DESCRIPTOR_TYPE 未紧接在接口描述符之后定义。 5. ERR_USBD_BAD_INTF_DESC = 传递错误的接口描述符。 6. ERR_USBD_BAD_EP_DESC = 传递错误的端点描述符。

25.5.33 USBD_HID_INIT_PARAM

USB 描述符数据结构。

表 515. USBD_HID_INIT_PARAM 类结构

成员	描述
mem_base	uint32_tuint32_t USBD_HID_INIT_PARAM::mem_base 协议栈从中分配数据和缓冲的基本存储器位置。 **注：**此字段中设置的存储器地址应可由 USB DMA 控制器访问。该值还应在 4 字节边界对齐。
mem_size	uint32_tuint32_t USBD_HID_INIT_PARAM::mem_size 协议栈可使用的存储器缓冲区大小。 **注：**mem_size 应大于 USBD_HID_API::GetMemSize() 例程返回的大小。
max_reports	uint8_tuint8_t USBD_HID_INIT_PARAM::max_reports 该 HID 类驱动器实例支持的 HID 报告数量。
管脚	uint8_tuint8_t USBD_HID_INIT_PARAM::pad[3][3]

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 515. USBD_HID_INIT_PARAM 类结构 *（续）*

成员	描述
intf_desc	uint8_t *uint8_t* USBD_HID_INIT_PARAM::intf_desc 指向描述符数组内的 HID 接口描述符的指针 （
report_data	USB_HID_REPORT_T *USB_HID_REPORT_T* USBD_HID_INIT_PARAM::report_data 指向 HID 报告描述符数据结构数组的指针 （ **注：**该数组应属于全局范围。
HID_GetReport	ErrorCode_t(*ErrorCode_t(* USBD_HID_INIT_PARAM::HID_GetReport)(USBD_HANDLE_T hHid, USB_SETUP_PACKET *pSetup, uint8_t **pBuffer, uint16_t *length))(USBD_HANDLE_T hHid, USB_SETUP_PACKET *pSetup, uint8_t **pBuffer, uint16_t *length) HID 获取报告回调函数。 该函数由应用软件提供。主机发送 HID_REQUEST_GET_REPORT 请求时调用此函数。设置数据包 (**注：**还会通过中断 IN 端点发送 HID 报告。仅当在控制端点上接收报告请求时，才调用该函数。应用应实施 HID_EpIn_Hdlr 以通过中断 IN 端点将报告发送至主机。 参数： 1. hHid = HID 函数驱动器的句柄。 2. pSetup = 指向从主机接收的设置数据包的指针。 3. pBuffer = 指向含报告数据的数据缓冲区指针的指针。指向指针的指针用于实施 0 复制缓冲区。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。 4. length = 复制到目标缓冲区的数据量。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。
HID_SetReport	ErrorCode_t(*ErrorCode_t(* USBD_HID_INIT_PARAM::HID_SetReport)(USBD_HANDLE_T hHid, USB_SETUP_PACKET *pSetup, uint8_t **pBuffer, uint16_t length))(USBD_HANDLE_T hHid, USB_SETUP_PACKET *pSetup, uint8_t **pBuffer, uint16_t length) HID 设置报告回调函数。 该函数由应用软件提供。主机发送 HID_REQUEST_SET_REPORT 请求时调用此函数。设置数据包 (hHid：HID 函数驱动器的句柄。pSetup：指向从主机接收的设置数据包的指针。pBuffer：指向含报告数据的数据缓冲区指针的指针。指向指针的指针用于实施 0 复制缓冲区。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。 length：复制到目标缓冲区的数据量。 参数： 1. hHid = HID 函数驱动器的句柄。 2. pSetup = 指向从主机接收的设置数据包的指针。 3. pBuffer = 指向含报告数据的数据缓冲区指针的指针。指向指针的指针用于实施 0 复制缓冲区。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。 4. length = 复制到目标缓冲区的数据量。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 515. USBD_HID_INIT_PARAM 类结构 *（续）*

成员	描述
HID_GetPhysDesc	ErrorCode_t(*ErrorCode_t(* USBD_HID_INIT_PARAM::HID_GetPhysDesc)(USBD_HANDLE_T hHid, USB_SETUP_PACKET *pSetup, uint8_t **pBuf, uint16_t *length))(USBD_HANDLE_T hHid, USB_SETUP_PACKET *pSetup, uint8_t **pBuf, uint16_t *length) 用于处理 HID_GetPhysDesc 请求的可选回调函数。 应用软件可提供此回调 HID_GetPhysDesc 处理器以处理主机发送的获取物理描述符请求。当主机请求将物理描述符置零时，应用应返回一个可标识描述符集数量及其大小的特殊描述符。物理索引等于 1 的 Get_Descriptor 请求应返回第一个物理描述符集。一个设备的项目可能有备用的用法。这些可以通过发出后续 Get_Descriptor 请求，同时递增描述符索引来枚举。设备返回的最后一个描述符集应针对索引大于 HID 描述符中所定义的最后编号的请求。 **注：**不含物理描述符的应用应在调用 USBD_HID_API::Init() 前将该数据成员置零。 参数： 1. hHid = HID 函数驱动器的句柄。 2. pSetup = 指向从主机接收的设置数据包的指针。 3. pBuf = 指向含物理描述符数据的数据缓冲区指针的指针。如果物理描述符位于 USB 可访问存储器区域中，应用可更新指针，否则应将描述符复制到该指针指向的地址。 4. length = 复制到目标缓冲区的数据量或描述符长度。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。
HID_SetIdle	ErrorCode_t(*ErrorCode_t(* USBD_HID_INIT_PARAM::HID_SetIdle)(USBD_HANDLE_T hHid, USB_SETUP_PACKET *pSetup, uint8_t idleTime))(USBD_HANDLE_T hHid, USB_SETUP_PACKET *pSetup, uint8_t idleTime) 用于处理 HID_REQUEST_SET_IDLE 请求的可选回调函数。 应用软件可提供此回调以处理主机发送的 HID_REQUEST_SET_IDLE 请求。该回调提供给应用，以调整与通过中断端点发送给主机的各种报告关联的定时器。设置数据包 (**注：**未在中断端点上发送报告或者在两个报告之间没有空闲时间的应用应在调用 USBD_HID_API::Init() 前将该数据成员置零。 参数： 1. hHid = HID 函数驱动器的句柄。 2. pSetup = 指向从主机接收的设置数据包的指针。 3. idleTime = 要为指定报告设置的空闲时间。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 515. USBD_HID_INIT_PARAM 类结构 *（续）*

成员	描述
HID_SetProtocol	ErrorCode_t(*ErrorCode_t(* USBD_HID_INIT_PARAM::HID_SetProtocol)(USBD_HANDLE_T hHid, USB_SETUP_PACKET *pSetup, uint8_t protocol))(USBD_HANDLE_T hHid, USB_SETUP_PACKET *pSetup, uint8_t protocol) 用于处理 HID_REQUEST_SET_PROTOCOL 请求的可选回调函数。 应用软件可提供此回调以处理主机发送的 HID_REQUEST_SET_PROTOCOL 请求。该回调提供给应用，以调整启动模式和报告模式之间代码的模式。 **注：**不支持协议模式的应用应在调用 USBD_HID_API::Init() 前将该数据成员置零。 参数： 1. hHid = HID 函数驱动器的句柄。 2. pSetup = 指向从主机接收的设置数据包的指针。 3. protocol = 协议模式。 0 = 使能协议 1 = 报告协议 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。
HID_EpIn_Hdlr	ErrorCode_t(*ErrorCode_t(* USBD_HID_INIT_PARAM::HID_EpIn_Hdlr)(USBD_HANDLE_T hUsb, void *data, uint32_t event))(USBD_HANDLE_T hUsb, void *data, uint32_t event) 可选中断 IN 端点事件处理器。 应用软件可提供中断 IN 端点事件处理器。在中断端点上将报告发送至主机的应用应通过该数据成员提供端点事件处理器。如果接口描述符，忽略此数据成员 hUsb：USB 设备协议栈的句柄。 data：HID 函数驱动器的句柄。 event：端点事件的类型。更多详情请参见 USBD_EVENT_T。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. data = HID 函数驱动器的句柄。 3. event = 端点事件的类型。更多详情请参见 USBD_EVENT_T。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 515. USBD_HID_INIT_PARAM 类结构 *（续）*

成员	描述
HID_EpOut_Hdlr	ErrorCode_t(*ErrorCode_t(* USBD_HID_INIT_PARAM::HID_EpOut_Hdlr)(USBD_HANDLE_T hUsb, void *data, uint32_t event))(USBD_HANDLE_T hUsb, void *data, uint32_t event) 可选中断 OUT 端点事件处理器。 应用软件可提供中断 OUT 端点事件处理器。在中断端点上从主机接收报告的应用应通过该数据成员提供端点事件处理器。如果接口描述符，忽略此数据成员 hUsb：USB 设备协议栈的句柄。 data：HID 函数驱动器的句柄。 event：端点事件的类型。更多详情请参见 USBD_EVENT_T。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. data = HID 函数驱动器的句柄。 3. event = 端点事件的类型。更多详情请参见 USBD_EVENT_T。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。
HID_GetReportDesc	ErrorCode_t(*ErrorCode_t(* USBD_HID_INIT_PARAM::HID_GetReportDesc)(USBD_HANDLE_T hHid, USB_SETUP_PACKET *pSetup, uint8_t **pBuf, uint16_t *length))(USBD_HANDLE_T hHid, USB_SETUP_PACKET *pSetup, uint8_t **pBuf, uint16_t *length) 代替默认 HID_GetReportDesc 处理器的可选用户可覆盖函数。 应用软件提供处理器函数地址作为参数结构的此数据成员，从而用自己的处理器覆盖默认 HID_GetReportDesc 处理器。首选默认处理器的应用应在调用 USBD_HID_API::Init() 前将该数据成员置零，并提供报告数据阵列 **注：** 参数： 1. hUsb = USB 设备协议栈的句柄。 2. data = 指向协议栈调用回调函数时所传递数据的指针。 3. event = 端点事件的类型。更多详情请参见 USBD_EVENT_T。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 515. USBD_HID_INIT_PARAM 类结构 *（续）*

成员	描述
HID_Ep0_Hdlr	ErrorCode_t(*ErrorCode_t(* USBD_HID_INIT_PARAM::HID_Ep0_Hdlr)(USBD_HANDLE_T hUsb, void *data, uint32_t event))(USBD_HANDLE_T hUsb, void *data, uint32_t event) 代替默认 HID 类处理器的可选用户可覆盖函数。 应用软件提供处理器函数地址作为参数结构的此数据成员，从而用自己的处理器覆盖默认 EP0 类处理器。首选默认处理器的应用应在调用 USBD_HID_API::Init() 前将该数据成员置零。 **注：** 参数： 1. hUsb = USB 设备协议栈的句柄。 2. data = 指向协议栈调用回调函数时所传递数据的指针。 3. event = 端点事件的类型。更多详情请参见 USBD_EVENT_T。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

25.5.34 USBD_HW_API

硬件 API 函数结构。该模块公开直接与 USB 设备控制器硬件交互的函数。

表 516. USBD_HW_API 类结构

成员	描述
GetMemSize	uint32_t(*uint32_t USBD_HW_API::GetMemSize)(USBD_API_INIT_PARAM_T *param) 用于确定 USB 设备协议栈的 DCD 和内核层所要求的存储器的函数。 应用层在调用 pUsbApi->hw-> 前调用该函数。 **注：**一些存储器区域无法由所有总线主机访问。 参数： 1. param = 含 USB 设备协议栈初始化参数的结构。 返回： 返回需要的存储器大小 （字节）。
Init	ErrorCode_t(*ErrorCode_t USBD_HW_API::Init)(USBD_HANDLE_T *phUsb, USB_CORE_DESCS_T *pDesc, USBD_API_INIT_PARAM_T *param) 用于初始化 USB 设备协议栈的 DCD 和内核层的函数。 应用层调用该函数以初始化 USB 硬件和内核层。成功初始化后，该函数将返回应传递给剩余函数的 USB 设备协议栈的一个句柄。 phUsb：指向 USBD_HANDLE_T 类型的 USB 设备协议栈句柄的指针。 param：含 USB 设备协议栈初始化参数的结构。 参数： 1. phUsb = 指向 USBD_HANDLE_T 类型的 USB 设备协议栈句柄的指针。 2. param = 含 USB 设备协议栈初始化参数的结构。 返回： 返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK(0) = 成功 2. ERR_USBD_BAD_MEM_BUF(0x0004000b) = 当未传递足够的存储器缓冲或者存储器未在 2048 边界上对齐时。
Connect	void(*void USBD_HW_API::Connect)(USBD_HANDLE_T hUsb, uint32_t con) 使 USB 设备在 USB 总线上可见 / 不可见的函数。 USB 初始化后，调用该函数。该函数使用软连接功能使设备在 USB 总线上可见。仅在应用准备好处理 USB 数据后，调用该函数。枚举过程在设备检测后由主机启动。驱动器根据在 USB 初始化函数中传递的 USB 描述符处理枚举过程。 hUsb：USB 设备协议栈的句柄。 con：连接 (1) 或断开连接 (0) 状态。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. con = 连接 (1) 或断开连接 (0) 状态。 返回： 无任何内容。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 516. USBD_HW_API 类结构 *（续）*

成员	描述
ISR	void(*void USBD_HW_API::ISR)(USBD_HANDLE_T hUsb) USB 设备控制器中断事件函数。 激活用户应用时，中断处理器映射到用户闪存空间。用户应用必须为 USB 中断提供中断处理器，并在中断处理器例程中调用该函数。驱动器中断处理器根据 USB 总线上接收的数据采取适当操作。 hUsb：USB 设备协议栈的句柄。 参数： 1. hUsb = USB 设备协议栈的句柄。 返回： 无任何内容。
复位	void(*void USBD_HW_API::Reset)(USBD_HANDLE_T hUsb) 用于复位 USB 设备协议栈和硬件控制器的函数。 复位 USB 设备协议栈和硬件控制器。禁用除 EP0 外的所有端点。清除所有挂起中断并复位端点传输队列。在内部通过 pUsbApi->hw->init() 并从复位事件调用该函数。 hUsb：USB 设备协议栈的句柄。 参数： 1. hUsb = USB 设备协议栈的句柄。 返回： 无任何内容。
ForceFullSpeed	void(*void USBD_HW_API::ForceFullSpeed)(USBD_HANDLE_T hUsb, uint32_t cfg) 强制高速 USB 设备以全速模式运行的函数。 连接到仅限全速的主机时，该函数对检测当前设备的行为很有用。 hUsb：USB 设备协议栈的句柄。 cfg：为 1 时 - 设置强制全速或者为 0 时 - 清除强制全速。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. cfg = 为 1 时 - 设置强制全速或者为 0 时 - 清除强制全速。 返回： 无任何内容。
WakeUpCfg	void(*void USBD_HW_API::WakeUpCfg)(USBD_HANDLE_T hUsb, uint32_t cfg) 用于配置 USB 设备控制器以基于远程事件唤醒主机的函数。 应用层调用该函数以配置 USB 设备控制器以基于远程事件唤醒。建议从注册到协议栈的用户 USB_WakeUpCfg() 回调例程调用此函数。 **注：**调用 pUsbApi->hw->Init() 例程前，通过设置 USBD_API_INIT_PARAM_T 结构的 USB_WakeUpCfg 成员将用户的 USB_WakeUpCfg() 注册到协议栈。某些 USB 设备控制器需要保持某些时钟始终运行以通过 pUsbApi->hw->WakeUp() 产生恢复信号。该挂接可支持此类控制器。在多数控制器情况中，这是空例程。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. cfg = 为 1 时 - 配置控制器以基于远程事件唤醒或者为 0 时 - 配置控制器以不基于远程事件唤醒。 返回： 无任何内容。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 516. USBD_HW_API 类结构 *（续）*

成员	描述
SetAddress	void(*void USBD_HW_API::SetAddress)(USBD_HANDLE_T hUsb, uint32_t adr) 用于设置设备控制器硬件中的主机分配的 USB 地址的函数。 协议栈从 USB 主机接收 USB_REQUEST_SET_ADDRESS 请求后会自动调用该函数。该接口提供给用户以在 （当前协议栈未处理的）其他情况下调用此函数。在多数用户应用中，该函数未直接调用。该函数还可由用户用来通过协议栈公开的回调接口，选择性地修改 USB 设备协议栈的标准处理器。 hUsb：USB 设备协议栈的句柄。 adr：设备控制器应响应的 USB 总线地址。通常由 USB 主机分配。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. adr = 设备控制器应响应的 USB 总线地址。通常由 USB 主机分配。 返回： 无任何内容。
Configure	void(*void USBD_HW_API::Configure)(USBD_HANDLE_T hUsb, uint32_t cfg) 用于将设备控制器硬件配置为选定配置的函数。 协议栈从 USB 主机接收 USB_REQUEST_SET_CONFIGURATION 请求后会自动调用该函数。该接口提供给用户以在 （当前协议栈未处理的）其他情况下调用此函数。在多数用户应用中，该函数未直接调用。该函数还可由用户用来通过协议栈公开的回调接口，选择性地修改 USB 设备协议栈的标准处理器。 hUsb：USB 设备协议栈的句柄。 cfg：配置索引。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. cfg = 配置索引。 返回： 无任何内容。
ConfigEP	void(*void USBD_HW_API::ConfigEP)(USBD_HANDLE_T hUsb, USB_ENDPOINT_DESCRIPTOR *pEPD) 用于根据描述符配置 USB 端点的函数。 协议栈从 USB 主机接收 USB_REQUEST_SET_CONFIGURATION 请求后会自动调用该函数。配置所有与选定配置相关的端点。该接口提供给用户以在 （当前协议栈未处理的）其他情况下调用此函数。在多数用户应用中，该函数未直接调用。该函数还可由用户用来通过协议栈公开的回调接口，选择性地修改 USB 设备协议栈的标准处理器。 hUsb：USB 设备协议栈的句柄。 pEPD：USB 2.0 规范中定义的端点描述符结构。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. pEPD = USB 2.0 规范中定义的端点描述符结构。 返回： 无任何内容。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 516. USBD_HW_API 类结构 *（续）*

成员	描述
DirCtrlEP	void(*void USBD_HW_API::DirCtrlEP)(USBD_HANDLE_T hUsb, uint32_t dir) 用于设置 USB 控制端点 EP0 的方向的函数。 该函数由协议栈按需自动调用。该接口提供给用户以在（当前协议栈未处理的）其他情况下调用此函数。在多数用户应用中，该函数未直接调用。该函数还可由用户用来通过协议栈公开的回调接口，选择性地修改 USB 设备协议栈的标准处理器。 hUsb：USB 设备协议栈的句柄。 cfg：为 1 时 - 在 IN 传输模式中设置 EP0，为 0 时 - 在 OUT 传输模式中设置 EP0 参数： 1. hUsb = USB 设备协议栈的句柄。 2. cfg = 为 1 时 - 在 IN 传输模式中设置 EP0，为 0 时 - 在 OUT 传输模式中设置 EP0 返回： 无任何内容。
EnableEP	void(*void USBD_HW_API::EnableEP)(USBD_HANDLE_T hUsb, uint32_t EPNum) 用于使能选定 USB 端点的函数。 该函数可在所选端点使能中断。 hUsb：USB 设备协议栈的句柄。EPNum：符合 USB 规范的端点号。例如，EP1_IN 由 0x81 编号表示。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. EPNum = 符合 USB 规范的端点号。例如， EP1_IN 由 0x81 编号表示。 返回： 无任何内容。 该函数可在所选端点使能中断。 hUsb：USB 设备协议栈的句柄。 EPNum：符合 USB 规范的与事件对应的端点号。例如， EP1_IN 由 0x81 编号表示。对于设备事件，将此 param 设为 0x0。 event：端点事件的类型。更多详情请参见 USBD_EVENT_T。 enable：1 - 使能事件， 0 - 禁用事件。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. EPNum = 符合 USB 规范的与事件对应的端点号。例如， EP1_IN 由 0x81 编号表示。对于设备事件，将此 param 设为 0x0。 3. event = 端点事件的类型。更多详情请参见 USBD_EVENT_T。 4. enable = 1 - 使能事件， 0 - 禁用事件。 返回： 返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK(0) = - 成功 2. ERR_USBD_INVALID_REQ(0x00040001) = - 无效事件类型。
DisableEP	void(*void USBD_HW_API::DisableEP)(USBD_HANDLE_T hUsb, uint32_t EPNum) 用于禁用选定 USB 端点的函数。 该函数可在所选端点禁用中断。 hUsb：USB 设备协议栈的句柄。EPNum：符合 USB 规范的端点号。例如，EP1_IN 由 0x81 编号表示。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. EPNum = 符合 USB 规范的端点号。例如， EP1_IN 由 0x81 编号表示。 返回： 无任何内容。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 516. USBD_HW_API 类结构 *（续）*

成员	描述
ResetEP	void(*void USBD_HW_API::ResetEP)(USBD_HANDLE_T hUsb, uint32_t EPNum) 用于复位选定 USB 端点的函数。 该函数可刷新端点缓冲区并复位数据切换逻辑。 hUsb：USB 设备协议栈的句柄。EPNum：符合 USB 规范的端点号。例如，EP1_IN 由 0x81 编号表示。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. EPNum = 符合 USB 规范的端点号。例如， EP1_IN 由 0x81 编号表示。 返回： 无任何内容。
SetStallEP	void(*void USBD_HW_API::SetStallEP)(USBD_HANDLE_T hUsb, uint32_t EPNum) 用于停止选定 USB 端点的函数。 为请求的端点产生停止信号。 hUsb：USB 设备协议栈的句柄。EPNum：符合 USB 规范的端点号。例如，EP1_IN 由 0x81 编号表示。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. EPNum = 符合 USB 规范的端点号。例如， EP1_IN 由 0x81 编号表示。 返回： 无任何内容。
ClrStallEP	void(*void USBD_HW_API::ClrStallEP)(USBD_HANDLE_T hUsb, uint32_t EPNum) 用于为请求的端点清除停止状态的函数。 该函数可为请求的端点清除停止状态。 hUsb：USB 设备协议栈的句柄。EPNum：符合 USB 规范的端点号。例如，EP1_IN 由 0x81 编号表示。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. EPNum = 符合 USB 规范的端点号。例如， EP1_IN 由 0x81 编号表示。 返回： 无任何内容。
SetTestMode	ErrorCode_t(*ErrorCode_t USBD_HW_API::SetTestMode)(USBD_HANDLE_T hUsb, uint8_t mode) 用于在请求的检测模式中设置高速 USB 设备控制器的函数。 USB-IF 要求将高速设备置于各种检测模式进行电气检测。每次接收到针对 USB_FEATURE_TEST_MODE 的 USB_REQUEST_CLEAR_FEATURE 请求时，该 USB 设备协议栈均会调用此函数。用户可以通过直接调用此函数将设备置于检测模式。设备控制器仅以全速运行时，返回 ERR_USBD_INVALID_REQ。 hUsb：USB 设备协议栈的句柄。 mode：USB 2.0 电气检测规范中定义的检测模式。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. mode = USB 2.0 电气检测规范中定义的检测模式。 返回： 返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK(0) = - 成功 2. ERR_USBD_INVALID_REQ(0x00040001) = - 无效的检测模式或设备控制器仅以全速运行。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 516. USBD_HW_API 类结构 *（续）*

成员	描述
ReadEP	uint32_t(*uint32_t USBD_HW_API::ReadEP)(USBD_HANDLE_T hUsb, uint32_t EPNum, uint8_t *pData) 用于读取在请求端点接收到的数据的函数。 该函数由 USB 协议栈和应用层调用，以读取在请求的端点接收到的数据。 hUsb：USB 设备协议栈的句柄。EPNum：符合 USB 规范的端点号。例如，EP1_IN 由 0x81 编号表示。pData：指向要在其中复制数据的数据缓冲区的指针。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. EPNum = 符合 USB 规范的端点号。例如， EP1_IN 由 0x81 编号表示。 3. pData = 指向要在其中复制数据的数据缓冲区的指针。 返回： 返回复制到缓冲区的字节数。
ReadReqEP	uint32_t(*uint32_t USBD_HW_API::ReadReqEP)(USBD_HANDLE_T hUsb, uint32_t EPNum, uint8_t *pData, uint32_t len) 用于为指定端点的读请求排队的函数。 该函数由 USB 协议栈和应用层调用，为指定端点的读请求排队。 hUsb：USB 设备协议栈的句柄。EPNum：符合 USB 规范的端点号。例如，EP1_IN 由 0x81 编号表示。pData：指向要在其中复制数据的数据缓冲区的指针。该缓冲地址应可由 USB DMA 主机访问。 len：已传递缓冲区的长度。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. EPNum = 符合 USB 规范的端点号。例如， EP1_IN 由 0x81 编号表示。 3. pData = 指向要在其中复制数据的数据缓冲区的指针。该缓冲区地址应可由 USB DMA 主机访问。 4. len = 已传递缓冲区的长度。 返回： 返回所请求的缓冲区的长度。
ReadSetupPkt	uint32_t(*uint32_t USBD_HW_API::ReadSetupPkt)(USBD_HANDLE_T hUsb, uint32_t EPNum, uint32_t *pData) 用于读取在请求端点接收到的设置数据包数据的函数。 该函数由 USB 协议栈和应用层调用，以读取在请求的端点接收到的设置数据包数据。 hUsb：USB 设备协议栈的句柄。EPNum：符合 USB 规范的端点号。例如，EP0_IN 由 0x80 编号表示。pData：指向要在其中复制数据的数据缓冲区的指针。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. EPNum = 符合 USB 规范的端点号。例如， EP0_IN 由 0x80 编号表示。 3. pData = 指向要在其中复制数据的数据缓冲区的指针。 返回： 返回复制到缓冲区的字节数。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 516. USBD_HW_API 类结构 *（续）*

成员	描述
WriteEP	uint32_t(*uint32_t USBD_HW_API::WriteEP)(USBD_HANDLE_T hUsb, uint32_t EPNum, uint8_t *pData, uint32_t cnt) 用于将要发送的数据写到请求端点的函数。 该函数由 USB 协议栈和应用层调用，以发送数据到请求端点。 hUsb：USB 设备协议栈的句柄。EPNum：符合 USB 规范的端点号。例如，EP1_IN 由 0x81 编号表示。pData：指向要在其中复制数据的数据缓冲区的指针。 cnt：要写入的字节数。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. EPNum = 符合 USB 规范的端点号。例如， EP1_IN 由 0x81 编号表示。 3. pData = 指向要从其中复制数据的数据缓冲区的指针。 4. cnt = 要写入的字节数。 返回： 返回写入的字节数。
WakeUp	void(*void USBD_HW_API::WakeUp)(USBD_HANDLE_T hUsb) 用于在主机上产生用于远程主机唤醒的恢复信号的函数。 该函数由应用层调用，在系统处于挂起状态时远程唤醒主机控制器。应用应在配置描述符的bmAttributes 中设置 USB_CONFIG_REMOTE_WAKEUP，指示该远程唤醒功能。并且，仅当主机在挂起总线前通过发送 SET_FEATURE 请求使能了 USB_FEATURE_REMOTE_WAKEUP 时，该例程才会产生恢复信号。 hUsb：USB 设备协议栈的句柄。 参数： 1. hUsb = USB 设备协议栈的句柄。 返回： 无任何内容。
EnableEvent	ErrorCode_t(*ErrorCode_t(* USBD_HW_API::EnableEvent)(USBD_HANDLE_T hUsb, uint32_t EPNum, uint32_t event_type, uint32_t enable))(USBD_HANDLE_T hUsb, uint32_t EPNum, uint32_t event_type, uint32_t enable)

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

25.5.35 USBD_MSC_API

MSC 类 API 函数结构。该模块公开直接与 USB 设备控制器硬件交互的函数。

表 517. USBD_MSC_API 类结构

成员	描述
GetMemSize	uint32_t(*uint32_t USBD_MSC_API::GetMemSize)(USBD_MSC_INIT_PARAM_T *param) 用于确定 MSC 函数驱动器模块所要求的存储器的函数。 在调用 pUsbApi->msc->Init() 前，应用层调用该函数以分配 MSC 函数驱动器模块所使用的存储器。应用程序应分配可由 USB 控制器 /DMA 控制器访问的存储器。 **注：**一些存储器区域无法由所有总线主机访问。 参数： 1. param = 含 MSC 函数驱动器模块初始化参数的结构。 返回： 返回需要的存储器大小 （字节）。
init	ErrorCode_t(*ErrorCode_t USBD_MSC_API::init)(USBD_HANDLE_T hUsb, USBD_MSC_INIT_PARAM_T *param) 初始化 MSC 函数驱动器模块的函数。 应用层调用该函数以初始化 MSC 函数驱动器模块。 hUsb：USB 设备协议栈的句柄。 param：含 MSC 函数驱动器模块初始化参数的结构。 参数： 1. hUsb = USB 设备协议栈的句柄。 2. param = 含 MSC 函数驱动器模块初始化参数的结构。 返回： 返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功 2. ERR_USBD_BAD_MEM_BUF = 传递的存储器缓冲区不以 4 字节对齐或小于要求。 3. ERR_API_INVALID_PARAM2 = Either MSC_Write() or MSC_Read() or MSC_Verify() callbacks are not defined. 4. ERR_USBD_BAD_INTF_DESC = 传递错误的接口描述符。 5. ERR_USBD_BAD_EP_DESC = 传递错误的端点描述符。

25.5.36 USBD_MSC_INIT_PARAM

海量存储类函数驱动器初始化参数数据结构。

表 518. USBD_MSC_INIT_PARAM 类结构

成员	描述
mem_base	uint32_tuint32_t USBD_MSC_INIT_PARAM::mem_base 协议栈从中分配数据和缓冲的基本存储器位置。 **注：**此字段中设置的存储器地址应可由 USB DMA 控制器访问。该值还应在 4 字节边界对齐。
mem_size	uint32_tuint32_t USBD_MSC_INIT_PARAM::mem_size 协议栈可使用的存储器缓冲区大小。 **注：**mem_size 应大于 USBD_MSC_API::GetMemSize() 例程返回的大小。
InquiryStr	uint8_t *uint8_t* USBD_MSC_INIT_PARAM::InquiryStr 指向 28 字符的字符串的指针。发送该字符串以响应 SCSI 查询命令。 **注：**指针指向的数据应属于全局范围。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 518. USBD_MSC_INIT_PARAM 类结构 *（续）*

成员	描述
BlockCount	uint32_tuint32_t USBD_MSC_INIT_PARAM::BlockCount 海量存储设备上存在的模块数量。
BlockSize	uint32_tuint32_t USBD_MSC_INIT_PARAM::BlockSize 块大小 （字节数）
MemorySize	uint32_tuint32_t USBD_MSC_INIT_PARAM::MemorySize 存储器大小 （字节数）
intf_desc	uint8_t *uint8_t* USBD_MSC_INIT_PARAM::intf_desc 指向描述符数组内的接口描述符的指针 （
MSC_Write	void(*void(* USBD_MSC_INIT_PARAM::MSC_Write)(uint32_t offset, uint8_t **src, uint32_t length))(uint32_t offset, uint8_t **src, uint32_t length) MSC 写回调函数。 该函数由应用软件提供。主机发送写命令时调用此函数。 offset：目标起始地址。 src：指向数据源指针的指针。指向指针的指针用于实施 0 复制缓冲区。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。 length：要写入的字节数。 参数： 1. offset = 目标起始地址。 2. src = 指向数据源指针的指针。指向指针的指针用于实施 0 复制缓冲区。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。 3. length = 要写入的字节数。 返回： 无任何内容。
MSC_Read	void(*void(* USBD_MSC_INIT_PARAM::MSC_Read)(uint32_t offset, uint8_t **dst, uint32_t length))(uint32_t offset, uint8_t **dst, uint32_t length) MSC 读回调函数。 该函数由应用软件提供。主机发送读命令时调用此函数。 offset：源起始地址。 dst：指向数据源指针的指针。协议栈中实施的 MSC 函数驱动器以 0 复制模型写入。表示协议栈从 USB 硬件 FIFO 写入 / 读取数据前，无需另外复制缓冲区。因此，参数是指向含地址缓冲区的指针的指针 (uint8_t** dst)。因此，用户应用可更新缓冲区指针，而不是将数据复制到参数指向的地址。/note 更新后的缓冲区地址应可由 USB DMA 主机访问。如果用户不想使用 0 复制模型，还应将数据复制到已传递的缓冲区指针参数指向的地址，并不得改变地址值。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。 length：要读取的字节数。 参数： 1. offset = 源起始地址。 2. dst = 指向数据源指针的指针。协议栈中实施的 MSC 函数驱动器以 0 复制模型写入。表示协议栈从 USB 硬件 FIFO 写入 / 读取数据前，无需另外复制缓冲区。因此，参数是指向含地址缓冲区的指针的指针 (uint8_t** dst)。因此，用户应用可更新缓冲区指针，而不是将数据复制到参数指向的地址。 /note 更新后的缓冲区地址应可由 USB DMA 主机访问。如果用户不想使用 0 复制模型，还应将数据复制到已传递的缓冲区指针参数指向的地址，并不得改变地址值。有关 0 复制概念的更多详情，请参见 0 复制数据传输模型。 3. length = 要读取的字节数。 返回： 无任何内容。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 518. USBD_MSC_INIT_PARAM 类结构 *（续）*

成员	描述
MSC_Verify	ErrorCode_t(*ErrorCode_t(* USBD_MSC_INIT_PARAM::MSC_Verify)(uint32_t offset, uint8_t buf[], uint32_t length))(uint32_t offset, uint8_t buf[], uint32_t length) MSC 验证回调函数。 该函数由应用软件提供。主机发送验证命令时调用此函数。回调函数应将缓冲区与所请求偏移量的目标存储器进行比较 offset：目标起始地址。 buf：含主机发送数据的缓冲区。 length：要验证的字节数。 参数： 1. offset = 目标起始地址。 2. buf = 含主机发送数据的缓冲区。 3. length = 要验证的字节数。 返回： 返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 如果缓冲区的数据匹配目标数据 2. ERR_FAILED = 至少有一个字节不同。
MSC_GetWriteBuf	void(*void(* USBD_MSC_INIT_PARAM::MSC_GetWriteBuf)(uint32_t offset, uint8_t **buff_adr, uint32_t length))(uint32_t offset, uint8_t **buff_adr, uint32_t length) 用于优化 MSC_Write 缓冲传输的可选回调函数。 该函数由应用软件提供。主机发送 SCSI_WRITE10/SCSI_WRITE12 命令时调用此函数。回调函数应更新 offset：目标起始地址。 buf：含主机发送数据的缓冲区。 length：要写入的字节数。 参数： 1. offset = 目标起始地址。 2. buf = 含主机发送数据的缓冲区。 3. length = 要写入的字节数。 返回： 无任何内容。
MSC_Ep0_Hdlr	ErrorCode_t(*ErrorCode_t(* USBD_MSC_INIT_PARAM::MSC_Ep0_Hdlr)(USBD_HANDLE_T hUsb, void *data, uint32_t event))(USBD_HANDLE_T hUsb, void *data, uint32_t event) 代替默认 MSC 类处理器可选用户可覆盖函数。 应用软件提供处理器函数地址作为参数结构的此数据成员，从而用自己的处理器覆盖默认 EP0 类处理器。首选默认处理器的应用应在调用 USBD_MSC_API::Init() 前将该数据成员置零。 **注：** 参数： 1. hUsb = USB 设备协议栈的句柄。 2. data = 指向协议栈调用回调函数时所传递数据的指针。 3. event = 端点事件的类型。更多详情请参见 USBD_EVENT_T。 返回： 回调应返回 ErrorCode_t 类型，以指示成功或错误状态。 返回值： 1. LPC_OK = 成功。 2. ERR_USBD_UNHANDLED = 事件未处理，因此将事件传递到行中的下一个。 3. ERR_USBD_xxx = 针对其他错误状态。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.1 本章导读

以太网控制器在器件 LPC4350 和 LPC4330 上可用。

26.2 基本配置

以太网控制器配置如下：

- 参见表 519 以了解时钟与电源控制。
- 以太网通过 ETHERNET_RST （复位号 22）复位。
- 以太网中断连接到 NVIC 中的中断插槽 # 5。
- 以太网唤醒包指示器连接到事件路由器中的插槽 #8。
- 在 CREG 模块 CREG6 寄存器中将以太网模式设置为 RMII 或 MII （参见表 43）。

表 519. 以太网时钟和电源控制

	基本时钟	分支时钟	工作频率	备注
以太网寄存器接口时钟	BASE_M4_CLK	CLK_M4_ETHERNET	高达 204 MHz	-
以太网 PHY 时钟	BASE_PHY_RX_CLK	-	75 MHz	选择时钟引脚 ENET_RX_CLK 作为 CGU 的 OUTCLK_7_CTRL 寄存器中此基础时钟的时钟源。
以太网 PHY 时钟	BASE_PHY_TX_CLK	-	75 MHz	选择时钟引脚 ENET_TX_CLK 作为 CGU 的 OUTCLK_8_CTRL 寄存器中此基础时钟的时钟源。

26.3 功能要点

- 10/100 Mbit/s
- TCP/IP 硬件校验和
- IP 校验和
- DMA 支持
- IEEE 1588 时间戳模块
- IEEE 1588 高级时间戳支持 (IEEE 1588-2008 v2)
- 电源管理远程唤醒帧和魔术包检测
- 同时支持全双工和半双工操作
  - 支持半双工操作的 CSMA/CD 协议。
  - 支持全双工操作的 IEEE 802.3x 流控制。
  - 在全双工操作中，可选择将接收到的暂停控制帧向前发送到用户应用程序。
  - 用于半双工操作的背压支持。
  - 在全双工操作中取消流控制输入时，自动传输零时间片暂停帧。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.4 简介

以太网模块使主机能够通过符合 IEEE 802.3-2005 标准的以太网发送和接收数据。以太网接口包括一个通过使用 DMA 硬件加速来提供优化性能的全功能 10 Mbps 或 100 Mbps 以太网 MAC （媒体访问控制器）。

功能包括一整套控制寄存器、半双工操作或全双工操作、流控制、控制帧、发送重试的硬件加速、对 LAN 活动的接收包过滤和唤醒、支持唤醒和魔法包帧。帧通过分散 - 聚集 DMA 进行自动发送和接收，可让 CPU 免去不少的操作。

IEEE 1588 时间戳 (IEEE 1588-2002) 和 IEEE 高级时间戳支持 (IEEE 1588-2008 v2) 以及 IP（IPv4或IPv6）、TCP、UDP和ICMP的硬件校验和等附加功能丰富了所支持的功能列表。

以太网模块是连接至 AHB 多层矩阵的 AHB 主机，可访问连接至外部存储控制器的内部 SRAM 和存储器的以太网数据、控制以及状态信息。LPC43xx 中的其他 AHB 流量可使用其他主机将以太网活动与系统其他部分有效分开来产生。

以太网模块与使用 MII （媒体独立接口）或 RMII （简化型 MII）协议的片外以太网 PHY 以及片内 MIIM （媒体独立接口管理）串行总线连接。

图 60. 以太网框图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.5 引脚说明

表 520. 以太网引脚描述

引脚功能	方向	描述
MIIM 接口		
ENET_MDIO	I/O	以太网 MIIM 数据输入和输出。
ENET_MDC	O	以太网 MIIM 时钟。
RMII 接口		
ENET_RXD[1:0]	I	以太网接收数据。
ENET_TXD[1:0]	O	以太网发送数据。
ENET_RX_DV	I	以太网接收数据有效。
ENET_REF_CLK	I	以太网参考时钟。
ENET_TX_EN	O	以太网发送数据使能。
MII 接口		
ENET_RXD[3:0]	I	以太网接收数据。
ENET_TXD[3:0]	O	以太网发送数据。
ENET_COL	I	以太网冲突检测。
ENET_CRS	I	以太网载波监听。
ENET_TX_ER	O	以太网发送错误。
ENET_TX_CLK	I	以太网发送时钟。
ENET_RX_CLK	I	以太网接收时钟。
ENET_RX_ER	I	以太网接收错误。
ENET_TX_EN	O	以太网发送使能。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.6 寄存器描述

表 521. 寄存器简介：以太网 MAC 和 DMA （基址 0x4001 0000）

名称	访问类型	地址偏移	描述	复位值	参考
MAC_CONFIG	R/W	0x0000	MAC 配置寄存器	0x0000 8000	表 522
MAC_FRAME_FILTER	R/W	0x0004	MAC 帧滤波器	0x0000 0000	表 523
MAC_HASHTABLE_HIGH	R/W	0x0008	Hash 表高位寄存器	0x0000 0000	表 524
MAC_HASHTABLE_LOW	R/W	0x000C	Hash 表低位寄存器	0x0000 0000	表 525
MAC_MII_ADDR	R/W	0x0010	MII 地址寄存器	0x0000 0000	表 526
MAC_MII_DATA	R/W	0x0014	MII 数据寄存器	0x0000 0000	表 528
MAC_FLOW_CTRL	R/W	0x0018	流控制寄存器	0x0000 0000	表 529
MAC_VLAN_TAG	R/W	0x001C	虚拟局域网 (VLAN) 标志寄存器	0x0000 0000	表 530
-	-	0x0020	保留	-	-
MAC_DEBUG	RO	0x0024	调试寄存器	0x0000 0000	表 531
MAC_RWAKE_FRFLT	R/W	0x0028	远程唤醒帧滤波器	0x0000 0000	表 532
MAC_PMT_CTRL_STAT	R/W	0x002C	PMT 控制和状态	0x0000 0000	表 533
-	-	0x0030 - 0x0034	保留	-	-
MAC_INTR	RO	0x0038	中断状态寄存器	0x0000 0000	表 534
MAC_INTR_MASK	R/W	0x003C	中断掩码寄存器	0x0000 0000	表 535
MAC_ADDR0_HIGH	R/W	0x0040	MAC 地址 0 高位寄存器	0x8000 FFFF	表 536
MAC_ADDR0_LOW	R/W	0x0044	MAC 地址 0 低位寄存器	0xFFFF FFFF	表 537
-	-	0x0048 - 0x06FC	保留	-	-
MAC_TIMESTP_CTRL	R/W	0x0700	时间戳控制寄存器	0x0000 2000	表 538
SUBSECOND_INCR	R/W	0x0704	亚秒递增寄存器	0x0000 0000	表 540
SECONDS	RO	0x0708	系统时间秒寄存器	0x0000 0000	表 541
NANOSECONDS	RO	0x070C	系统时间纳秒寄存器	0x0000 0000	表 542
SECONDSUPDATE	R/W	0x0710	系统时间秒更新寄存器	0x0000 0000	表 543
NANOSECONDSUPDATE	R/W	0x0714	系统时间纳秒更新寄存器	0x0000 0000	表 544
ADDEND	R/W	0x0718	时间戳加数寄存器	0x0000 0000	表 545
TARGETSECONDS	R/W	0x071C	目标时间秒寄存器	0x0000 0000	表 546
TARGETNANOSECONDS	R/W	0x0720	目标时间纳秒寄存器	0x0000 0000	表 547
HIGHWORD	R/W	0x0724	系统时间高字秒寄存器	0x0000 0000	表 548
TIMESTAMPSTAT	RO	0x0728	时间戳状态寄存器	0x0000 0000	表 549
PPSCTRL	R/W	0x072C	PPS 控制寄存器	0x0000 0000	表 550
AUXNANOSECONDS	RO	0x0730	辅助时间戳纳秒寄存器	0x0000 0000	表 551
AUXSECONDS	RO	0x0734	辅助时间戳秒寄存器	0x0000 0000	表 552
-		0x0738 - 0x0FFC	保留		-
DMA_BUS_MODE	R/W	0x1000	总线模式寄存器	0x0002 0100	表 553
DMA_TRANS_POLL_DEMAND	R/W	0x1004	发送轮询请求寄存器	0x0000 0000	表 555
DMA_REC_POLL_DEMAND	R/W	0x1008	接收轮询请求寄存器	0x0000 0000	表 556

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 521. 寄存器简介：以太网 MAC 和 DMA （基址 0x4001 0000）*（续）*

名称	访问类型	地址偏移	描述	复位值	参考
DMA_REC_DES_ADDR	R/W	0x100C	接收描述符列表地址寄存器	0x0000 0000	表 557
DMA_TRANS_DES_ADDR	R/W	0x1010	发送描述符列表地址寄存器	0x0000 0000	表 558
DMA_STAT	R/W	0x1014	状态寄存器	0x0000 0000	表 559
DMA_OP_MODE	R/W	0x1018	操作模式寄存器	0x0000 0000	表 560
DMA_INT_EN	R/W	0x101C	中断使能寄存器	0x0000 0000	表 561
DMA_MFRM_BUFOF	RO	0x1020	遗漏帧和缓冲区溢出寄存器	0x0000 0000	表 562
DMA_REC_INT_WDT	R/W	0x1024	接收中断看门狗定时器寄存器	0x0000 0000	表 563
-	-	0x1028 - 0x1044	保留	-	-
DMA_CURHOST_TRANS_DES	RO	0x1048	当前主机发送描述符寄存器	0x0000 0000	表 564
DMA_CURHOST_REC_DES	RO	0x104C	当前主机接收描述符寄存器	0x0000 0000	表 565
DMA_CURHOST_TRANS_BUF	RO	0x1050	当前主机发送缓冲区地址寄存器	0x0000 0000	表 566
DMA_CURHOST_REC_BUF	RO	0x1054	当前主机接收缓冲区地址寄存器	0x0000 0000	表 567
-		0x1058	-	-	-

26.6.1 MAC 配置寄存器

MAC 配置寄存器可建立接收和发送操作模式。

表 522. MAC 配置寄存器 （MAC_CONFIG，地址 0x4001 0000）位描述

位	符号	描述	复位值	访问类型
1:0	-	保留	00	RO
2	RE	接收器使能 此位设置时，MAC 的接收器状态机使能，用于接收来自 MII 的帧。此位复位时，MAC 接收状态机在完成当前帧的接收之后禁用，不再接收来自 MII 的任何帧。	0	R/W
3	TE	发送器使能 此位设置时， MAC 的发送状态机使能，用于在 MII 上发送。此位复位时， MAC 发送状态机在完成当前帧的发送之后被禁用，不再发送任何帧。	0	R/W
4	DF	延迟检查 此位设置时，MAC 中的延期检查功能使能。如果发送状态机在 10/100-Mbps 模式下延期超过 24,288 位时间，则 MAC 将发布 “帧终止” 状态，且在发送帧状态中设置超延期错误位。如果将内核配置为在 1000 Mbps 传输率下运行，或者如果在 10/100-Mbps 模式下使能巨型帧模式，则延迟阈值为 155,680 位时间。延期在发送器准备好进行发送时开始，但是会被 MII 上的活动 CRS （载波侦听）阻止。延迟时间不累计。如果发送器延迟 10,000 位时间，然后发送、碰撞、退回，则退回完成后势必再次延迟，延迟定时器复位至 0 后重新启动。 此位复位时，延期检查功能被禁用，且 MAC 延期至 CRS 信号变为无效。该位仅适用于半双工模式，且仅保留 (RO) 在全双工配置中。	0	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 522. MAC 配置寄存器 （MAC_CONFIG，地址 0x4001 0000）位描述 *（续）*

位	符号	描述	复位值	访问类型
6:5	BL	退回限制 后退限值决定插槽时间延迟（传输率为 1000 Mbps 时，延迟 4096 位时间；传输率为 10/100 Mbps 时，延迟 512 位时间）的随机整数 (r)，而插槽时间延迟即发生冲突后重试时，MAC 在重新安排发送尝试之前等待的时间。该位仅适用于半双工模式，且仅保留 (RO) 在全双工配置中。 • 00: k = 最小值 (n, 10) • 01: k = 最小值 (n, 8) • 10: k = 最小值 (n, 4) • 11: k = 最小值 (n, 1) 其中， n = 再发送尝试。随机整数 r 的取值范围为 $0 \le r \le 2^k$。	0	R/W
7	ACS	自动 Pad/CRC 剥离 此位设置时，仅在长度的字段值小于或等于 1,500 字节时，MAC 才会剥离输入帧上的 Pad/FCS 字段。已接收的所有长度字段大于或等于 1,501 字节的帧都将传递到未剥离 Pad/FCS 字段的应用。 此位复位时， MAC 将把所有输入帧以未修改状态传至主机。	0	R/W
8	-	上行链路 / 下行链路 表示链路在 SMII 接口配置发送过程中是向上还是向下： 0 = 下行链路 1 = 上行链路	0	R/W
9	DR	禁用重试 此位设置时， MAC 仅尝试一次发送。如果 MII 上发生冲突， MAC 将忽略当前帧发送并在发送帧状态中报告具有过多冲突错误的 “帧终止”。 此位复位时，MAC 将基于 BL 的设置尝试重试。此位仅适用于半双工模式，在仅全双工配置中保留 （RO 带默认值）。	0	R/W
10	IPC	校验和卸载 此位设置时，MAC 计算所有接收到的以太网帧有效载荷的 16 位二进制反码的和。它还会检查 IPv4 首部校验和 （假设已接收以太网的帧为 25 至 26 或 29 至 30 字节（VLAN 标志））对于已接收帧来说是否正确，并指定接收状态字的状态。 MAC 内核也附带 16 位校验和计算，用于 IP 首部数据报有效载荷 （IPv4 文件头之后的字节），并将之附于传输至应用程序的以太网帧（如果已取消选定类型 2 COE）。当该位复位时，该功能禁用。 如果选择了类型 2 COE，则此位设置时，将使能 IPv4 校验和检查，用于接收帧的有效载荷的 TCP/UDP/ICMP 首部。此位复位时，接收器中的 COE 功能被禁用，且相应的 PCE 和 IP HCE 状态位始终被清除。	0	R/W
11	DM	双工模式 此位设置时， MAC 以全双工模式操作，可同时发送和接收。	0	R/W
12	LM	环回模式 设置此位时， MAC 在 MII 上以环回模式操作。环回正确操作需要 (G)MII 接收时钟输入，因为发送时钟不在内部环回。	0	R/W
13	DO	禁用接收自有数据 此位设置时， MAC 禁用在半双工模式下接收帧。 此位复位时， MAC 在发送时接收 PHY 给出的所有数据包。 如果 MAC 以全双工模式操作，此位不适用。	0	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 522. MAC 配置寄存器 （MAC_CONFIG，地址 0x4001 0000）位描述 *（续）*

位	符号	描述	复位值	访问类型
14	FES	速度 表示高速以太网 (MII) 模式下的速度： 0 = 10 Mbps 1 = 100 Mbps	0	
15	PS	端口选择 1 = MII (100 Mbp) - 这是唯一允许的值。	1	RO
16	DCRS	发送过程中禁用载波监听 当设置为高位时，在半双工模式下，该位在帧发送过程中会使 MAC 发送器忽略 (G)MII CRS 信号。由于在此类发送过程中载波丢失或无载波，该请求不会发生错误。当该位为低位时， MAC 发送器会因载波监听而发生此类错误，甚至会中止发送。	0	R/W
19:17	IFG	帧间隙 这些位可控制发送过程中帧之间的最小 IFG。 000 = 96 位时间 001 = 88 位时间 010 = 80 位时间 ... 000 = 40 位时间 请注意，在半双工模式下，最小 IFG 只能配置为 64 位时间 (IFG = 100)。下限值将不予考虑	000	R//W
20	JE	巨型帧使能 此位设置时，MAC 允许 9,018 字节的巨型帧（对于 VLAN 标签帧为 9,022 字节），在接收帧状态中不会报告巨型帧错误。	0	R/W
21	-	保留。	0	RO
22	JD	Jabber 禁用 此位设置时， MAC 禁用发送器中的 jabber 定时器，并可传输高达 16,384 字节的帧。 此位复位时，如果应用程序在发送期间发出超过 2,048 字节的数据 （如果 JE 设为高位，则为 10,240），则 MAC 关闭发送器。	0	R/W
23	WD	看门狗禁用 此位设置时， MAC 禁用接收器中的看门狗定时器，可接收高达 16,384 字节的帧。 此位复位时， MAC 允许接收不超过 2,048 字节 （如果 JE 设为高位，则为 10,240）的帧，并切除其后接收的任何字节。	0	R/W
31:24	-	保留。	0x00	RO

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.6.2 MAC 帧滤波寄存器

MAC 帧滤波寄存器包含接收帧的滤波器控制。该寄存器的一些控制转到了 MAC 的地址检查块，执行第一级地址滤波。根据其他控制 （例如通过坏帧和通过控制帧）对输入帧执行第二级滤波。

表 523. MAC 帧滤波寄存器 （MAC_FRAME_FILTER，地址 0x4001 0004）位描述

位	符号	描述	复位值	访问类型
0	PR	混杂模式 当对该位进行设置时，地址滤波器模块会通过所有输入帧，不管其目标地址和源地址。当对 PR 进行设置时，接收状态字的 SA/DA 滤波失败状态位将始终被清除。	0	R/W
1	-	保留	0	RO
2	-	保留	0	RO
3	DAIF	DA 反向滤波 当对该位进行设置时，地址检查块在反向滤波模式下运行，以对单播和多播帧进行 DA 地址比较。 复位时，对帧进行正常滤波。	0	R/W
4	PM	通过全部多播 设置时，此位表示所有接收的带有多播目标地址 （目标地址字段的第一位为 “1” ）的帧均已通过。 复位时，多播帧的滤波操作取决于 HMC 位。	0	R/W
5	DBF	禁用广播帧 当对该位进行设置时， AFM 模块会对所有输入广播帧进行滤波操作。 当该位复位时， AFM 模块通过所有接收到的广播帧。	0	R/W
7:6	PCF	通过控制帧 这些位可控制所有控制帧 （包括单播和多播暂停帧）的转发。请注意，暂停控制帧的处理只取决于流控制寄存器 [2] 的 RFE。 00 = MAC 会对所有传入应用程序的控制帧进行滤波操作。 01 = MAC 将所有控制帧 （除了暂停控制帧之外）转发至应用程序，即使其地址滤波失败。 10 = MAC 将所有控制帧转发至应用程序，即使其地址滤波失败。 11 = MAC 转发通过地址滤波器的控制帧。	00	R/W
8	SAIF	SA 反向滤波 当对该位进行设置时，地址检查块在反向滤波模式下运行，以进行 SA 地址比较。其 SA 与 SA 寄存器匹配的帧将被标记为 SA 地址滤波失败。 当该位复位时，其 SA 与 SA 寄存器不匹配的帧将被标记为 SA 地址滤波失败。	0	R/W
9	SAF	源地址滤波器使能 MAC 内核将接收帧的 SA 字段与使能的 SA 寄存器中的编程值进行比较。如果比较结果匹配，则 RxStatus 状态字的 SAMatch 位将被设置为高位。将该位设置为高位且 SA 滤波失败时， MAC 会丢弃该帧。 此位复位时，MAC 内核会将已接收到的帧转发至应用程序，并根据 SA 地址比较结果决定是否转发 RxStatus 更新的 SA 匹配位。	0	R/W
30:10	-	保留	0	RO
31	RA	全部接收 此位设置时，MAC 接收器模块会将所有已接收到的帧传递到应用程序，无论其是否通过地址滤波器。 SA/DA 滤波的结果 （通过或失败）将更新在接收状态字的相应位中。当该位复位时，接收器模块只会将那些通过 SA/DA 地址滤波器的帧传递到应用程序。	0	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.6.3 MAC Hash 表高位寄存器

64 位 Hash 表用于群地址滤波。就 Hash 滤波而言，输入帧中目标地址的内容通过 CRC 逻辑，而 CRC 寄存器的高 6 位用于为 Hash 表的内容编索引。最高有效位决定要使用的寄存器（Hash 表高位 /Hash 表低位），而其他 5 位决定位于寄存器中的位。00000 的 Hash 值选择选定寄存器的位 0，而 11111 的 Hash 值选择选定寄存器的位 31。

例如，如果接收的进入帧 DA 是 0x1F52419CB6AF（0x1F 是 MII 接口上接收的首字节），则内部计算得到的 6 位 Hash 值为 0x2C，并且会检查 HTH 寄存器位 [12] 以便滤波。如果接收的进入帧 DA 为 0xA00A98000045，则计算得到的 6 位 Hash 值为 0x07，并且会检查 HTL 寄存器位 [7] 以便滤波。

如果寄存器的相应位值为 1，则帧被接受。否则将遭到拒绝。如果在 MAC_CONFIG 寄存器中对 PM（通过全部多播）位进行设置，则所有多播帧都将被接受，无论多播 Hash 值是多少。

如果将 Hash 表寄存器配置为对 (G)MII 时钟域进行双同步，则只有在对 Hash 表高位 / 低位寄存器的位 [31:24]（在小端模式下）或位 [7:0]（在大端模式下）进行写操作时，才能启动同步。请注意，只有在双同步使能时，在目标时钟域中经过至少 4 个时钟周期后，才能对这些寄存器连续进行写操作。

Hash 表高位寄存器包含 Hash 表的高 32 位。

表 524. MAC Hash 表高位寄存器（MAC_HASHTABLE_HIGH，地址 0x4001 0008）位描述

位	符号	描述	复位值	访问类型
31:0	HTH	Hash 表高位 该字段包含 Hash 表的高 32 位。	0	R/W

26.6.4 MAC Hash 表低位寄存器

Hash 表低位寄存器包含 Hash 表的低 32 位。

表 525. MAC Hash 表低位寄存器（MAC_HASHTABLE_LOW，地址 0x4001 0008）位描述

位	符号	描述	复位值	访问类型
31:0	HTL	Hash 表低位 该字段包含 Hash 表的高 32 位。	0	R/W

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.6.5 MAC MII 地址寄存器

MII 地址寄存器通过管理接口控制外部 PHY 的管理周期。

表 526. MAC MII 地址寄存器 （MAC_MII_ADDR，地址 0x4001 0010）位描述

位	符号	描述	复位值	访问类型
0	GB	MII 忙 该寄存器字段可由应用程序读取 （读操作），可由应用程序通过某个寄存器写入 1 设为 1 （写设置），并由内核清零 （自清除）。应用程序无法清除此类字段，寄存器向该位写入 0 对该字段无任何作用。 该位应在对该寄存器和 MAC_MII_DATA 寄存器进行写操作之前读逻辑 0。另外，在对该寄存器进行写操作的过程中，必须将该位设置为 0。在 PHY 寄存器访问过程中，该位将由应用程序设置为 1，以表示读取或写入访问正在进行。MAC_MII_DATA 寄存器应保持有效，直到 MAC 在 PHY 写操作过程中将该位清除。只有 MAC 在 PHY 读操作过程中清除了该位之后， MAC_MII_DATA 寄存器才有效。在该位清除之前，不能对该寄存器进行写操作。	0	R/W
1	W	MII 写操作 如果对该位进行了设置，那么 PHY 通过该位会明白这是一次使用 MII 数据寄存器的写操作。如果未对该位进行设置，则为将数据置入 MII 数据寄存器中的读操作。	0	R/W
5:2	CR	CSR 时钟范围 CSR 时钟范围选择决定 MDC 时钟的频率。适用于下面每个值 （当 Bit[5] = 0 时）的 CLK_M4_ETHERNET 频率的建议范围保证了 MDC 时钟大约在 1.0 MHz - 2.5 MHz 频率范围之间。 当对位 5 进行设置时，您获得的 MDC 时钟频率将高于 IEEE 802.3 中规定的 2.5 MHz 的频率极限，并对下限值时钟分频器进行编程。例如，当 CLK_M4_ETHERNET 频率为 100 Mhz 且将这些位编程为 1010，那么由此产生的 MDC 时钟频率将为 12.5 Mhz，这超出了 IEEE 802.3 规定范围的限制。仅在接口芯片支持快速 MDC 时钟的情况下，对以下给定值进行编程。 有关位值的信息，请参见表 527。	0	R/W
10:6	GR	MII 寄存器 这些位可在选定的 PHY 设备中选择所需的 MII 寄存器。	0	R/W
15:11	PA	物理层地址 该字段可指明可能正在访问的 32 个 PHY 设备。	0	R/W
31:16	-	保留	0	RO

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 527. CSR 时钟范围值

位 5:2	CLK_M4_ETHERNET	MDC 时钟
0000	60 - 100 MHz	CLK_M4_ETHERNET/42
0001	100 - 150 MHz	CLK_M4_ETHERNET/62
0010	20 - 35 MHz	CLK_M4_ETHERNET/16
0011	35 - 60 MHz	CLK_M4_ETHERNET/26
0100	150 - 250 MHz	CLK_M4_ETHERNET/102
0101	250 - 300 MHz	CLK_M4_ETHERNET/124
0110, 0111	保留	-
1000	-	CLK_M4_ETHERNET/42
1001	-	CLK_M4_ETHERNET/62
1010	-	CLK_M4_ETHERNET/16
1011	-	CLK_M4_ETHERNET/26
1100	-	CLK_M4_ETHERNET/102
1101	-	CLK_M4_ETHERNET/124
1110	-	CLK_M4_ETHERNET/42
1111	-	CLK_M4_ETHERNET/62

26.6.6 MAC MII 数据寄存器

MII 数据寄存器可以存储将要写入 PHY 寄存器 （位于 MAC_MII_ADDR 寄存器中指定的地址）的写数据。该寄存器还可以存储 PHY 寄存器（位于 MAC_MII_ADDR 寄存器指定的地址）的读数据。

表 528. MII 数据寄存器 （MAC_MII_DATA，地址 0x4001 0014）位描述

位	符号	描述	复位值	访问类型
15:0	GD	MII 数据 这包含在管理读操作之后从 PHY 读取的 16 位数据值，或者在管理写操作之前将要写入 PHY 的 16 位数据值。	0	R/W
31:16	-	保留	0	RO

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.6.7 MAC 流控制寄存器

流控制寄存器通过 MAC 的流控制模块来控制 “控制”（暂停命令）帧的生成和接收。将 Busy 位设置为 1 后，对寄存器进行写操作可启动流控制模块生成暂停控制帧。根据 802.3x 规格中的规定选择控制帧的字段，而在控制帧的暂停时间字段中可使用该寄存器的暂停时间值。 Busy 位仍然处于设置状态，直到将控制帧传输到电缆为止。主机必须确保在对寄存器进行写操作之前将 Busy 位清除。

表 529. MAC 流控制寄存器 （MAC_FLOW_CTRL，地址 0x4001 0018）位描述

位	符号	描述	复位值	访问类型
0	FCB	流控制忙 / 背压激活 该寄存器字段可由应用程序读取（读操作），可由应用程序通过某个寄存器写入 1 设为 1 （写设置），并由内核清零 （自清除）。应用程序无法清除此类字段，寄存器向该位写入 0 对该字段无任何作用。 该位可在全双工模式下启动暂停控制帧。 在全双工模式下，该位在对流控制寄存器进行写操作之前应读为 0。若要启动暂停控制帧，则应用程序必须将该位设置为 1。在控制帧的传输过程中，该位将继续设置为表示正在发送帧的状态。暂停控制帧发送完成之后， MAC 将该位复位为 0。在该位清除之前，不能对流控制寄存器进行写操作。 在半双工模式下对该位进行设置 （并对 TFE 进行设置）时，背压通过 MAC 内核断言。MAC 在背压期间接收到新帧时，发送器开始发送 JAM 模式，结果产生冲突。该控制寄存器位可通过 mti_flowctrl_i 输入信号的逻辑 OR 操作，实现背压功能。当 MAC 配置为全双工模式时， BPA 将自动禁用。	0	R/W
1	TFE	发送流控制使能 在全双工模式下，如果设置了该位，则 MAC 会使能流控制操作，以发送暂停帧。当此位复位时， MAC 中的流控制操作被禁用，且 MAC 将不会发送任何暂停帧。 在半双工模式下，如果设置了该位， MAC 会使能背压操作。当该位复位时，背压功能被禁用。	0	R/W
2	RFE	接收流控制使能 此位设置时， MAC 将解码收到的暂停帧并在指定的 （暂停时间）时间内禁用其发送器。当该位复位时，暂停帧的解码功能被禁用。	0	R/W
3	UP	单播暂停帧检测 此位设置时，除了检测具有独特的多播地址的暂停帧以外，MAC 还将检测具有 MAC Address0 高位寄存器和 MAC Address0 低位寄存器中指定的站单播地址的暂停帧。此位复位时， MAC 将只检测 802.3x 标准中规定的带有唯一多播地址的暂停帧。	0	R/W
5:4	PLT	暂停低阈值 该字段配置暂停定时器的阈值，输入流控制在该阈值处受检查以用于暂停帧的自动再发送。阈值应始终小于位 [31:16] 中配置的暂停时间。例如，在 PT = Ox100（256 插槽时间）且 PLT = 01 的情况下，如果在发送第一个暂停帧后，流控制信号在 228 (256-28) 插槽时间被断言，则将自动发送另一个暂停帧。	00	R/W
6	-	保留	0x000	RO
7	DZPQ	禁用零量子暂停 如果设置了该位，该位会在来自 FIFO 层的流控制信号取消时禁用零时间片控制帧的自动生成。此位复位时，会使能使用自动零时间片暂停控制帧生成的正常操作。	0	R/W
15:8	-	保留	0	RO
31:16	PT	暂停时间 该字段包含将要在发送控制帧的暂停时间字段中使用的值。如果将暂停时间位配置为对 (G)MII 时钟域进行双同步，则只有在目标时钟域中经过至少 4 个时钟周期后，才能对该寄存器连续进行写操作。	0x0000	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.6.8 MAC VLAN 标签寄存器

VLAN 标志寄存器包含可识别 VLAN 帧的 IEEE 802.1Q VLAN 标志。MAC 将接收帧的第 13 个和第 14 个字节 （长度 / 类型）与 0x8100 进行比较，并将随后的 2 个字节与 VLAN 标志进行比较；如果匹配，它将把接收到的 VLAN 位设置为处于接收帧状态。帧的法定长度从 1518 字节增加到 1522 字节。

如果将 VLAN 标志寄存器配置为对 MII 时钟域进行双同步，则只有在目标时钟域中经过至少 4 个时钟周期后，才能对这些寄存器连续进行写操作。

表 530. MAC VLAN 标签寄存器 （MAC_VLAN_TAG，地址 0x4001 01C）位描述

位	符号	描述	复位值	访问类型
15:0	VL	VLAN 标志标识符，用于接收帧 这包含可识别 VLAN 帧的 802.1Q VLAN 标志，并与为 VLAN 帧正在接收的帧的第 15 个和第 16 个字节进行比较。位 [15:13] 为用户优先级，位 [12] 为规范格式指示符 (CFI)，而位 [11:0] 为 VLAN 标志的 VLAN 标识符 (VID) 字段。当对 ETV 位进行设置时，只有 VID （位 [11:0]）用于比较。 如果 VL （如果对 ETV 进行设置，则为 VL[11:0]）全为 0，则 MAC 不会为 VLAN 标志比较而检查第 15 个和第 16 个字节，并表明类型字段值为 0x8100 的所有帧均为 VLAN 帧。	0x0000	R/W
16	ETV	使能 12 位 VLAN 标志比较 当对该位进行设置时， 12 位 VLAN 识别符 （而不是全部 16 位 VLAN 标志）用于比较和滤波。将VLAN标志的位[11:0]与接收到的VLAN标志帧中的相应字段进行比较。 当该位复位时，接收到的所有 16 位 VLAN 帧的第 15 个和第 16 个字节均用于比较。	0	R/W
31:17	-	保留	0x0000	RO

26.6.9 MAC 调试寄存器

该调试寄存器可指定发送和接收数据路径和FIFO的所有主模块的状态。全零状态表示MAC内核处于空闲状态 （且 FIFO 为空），数据路径中没有正在进行的活动。

注：仅当以太网模块的时钟在复位操作期间出现，此寄存器中的复位值才有效。

表 531. MAC 调试寄存器 （MAC_DEBUG，地址 0x4001 0024）位描述

位	符号	描述	复位值	访问类型
0	RXIDLESTAT	当处于高位时，表示 MAC MII 接收协议引擎正在主动接收数据，且并未处于空闲状态。	0	RO
2:1	FIFOSTAT0	当处于高位时，表示 MAC 接收帧控制器模块的小型 FIFO 读和写控制器各自的活动状态。	0	RO
3	-	保留	-	RO
4	RXFIFOSTAT1	当处于高位时，表示 MTL RxFIFO 写控制器有效，且正在将接收到的帧传输到 FIFO。	0	RO
6:5	RXFIFOSTAT	RxFIFO 读控制器的状态： 00 = 空闲状态 01 = 正在读取帧数据 10 = 正在读取帧状态 （或时间戳） 11 = 正在刷新帧数据和状态	0	RO
7	-	保留	-	RO

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 531. MAC 调试寄存器 （MAC_DEBUG，地址 0x4001 0024）位描述 *（续）*

位	符号	描述	复位值	访问类型
9:8	RXFIFOLVL	RxFIFO 填充水平的状态 00 = RxFIFO 为空 01 = RxFIFO 填充水平低于流控制取消激活阈值 10 = RxFIFO 填充水平高于流控制激活阈值 11 = RxFIFO 为满	0	RO
15:10	-	保留	-	RO
16	TXIDLESTAT	当处于高位时，表示 MAC MII 发送协议引擎正在主动发送数据，且并未处于空闲状态。	0	RO
18:17	TXSTAT	MAC 发送帧控制器模块的状态： 00 = 空闲 01 = 等待前一帧的状态或 IFG/ 退回期结束 10 = 生成和发送暂停控制帧 （全双工模式下） 11 = 传输输入帧，以进行发送	0	RO
19	PAUSE	当处于高位时，表示 MAC 发送器处于暂停状态 （仅在全双工模式下），因此也不会安排发送任何帧。	0	RO
21:20	TXFIFOSTAT	TxFIFO 读控制器的状态 00 = 空闲状态 01 = 读状态 （将数据传输到 MAC 发送器） 10 = 等待来自 MAC 发送器的 TxStatus 11 = 写入接收到的 TxStatus 或刷新 TxFIFO	0	RO
22	TXFIFOSTAT1	当处于高位时，表示 MTL TxFIFO 写控制器有效，且正在将数据传输到 TxFIFO。	0	RO
23	-	保留	0	RO
24	TXFIFOLVL	当处于高位时，表示 MTL TxFIFO 不为空，且还有一些要发送的数据。	0	RO
25	TXFIFOFULL	当处于高位时，表示 MTL TxStatus FIFO 为满，因此 MTL 将不再接受要发送的帧。	0	RO
31:26	-	保留	-	RO

26.6.10 MAC 远程唤醒帧滤波寄存器

这是应用程序通过它对远程唤醒帧滤波寄存器 (WKUPFMFILTER) 进行写 / 读操作的地址。WKUPFMFILTER 实际上是 8 个 （不透明）此类 WKUPFMFILTER 寄存器的指针。 8 次按序写入该地址 (0x028) 就写入了所有 WKUPFMFILTER 寄存器。8 次按序从该地址 (0x028) 读取就读取了所有 WKUPFMFILTER 寄存器。参见第 26.7.1.1 节以了解详细信息。

注：请勿将位带机制用于该寄存器。

表 532. MAC 远程唤醒帧滤波寄存器 （MAC_RWAKE_FRFLT，地址 0x4001 0028）位描述

位	符号	描述	复位值	访问类型
31:0	ADDR	WKUPFMFILTER 地址	-	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.6.11 MAC PMT 控制和状态寄存器

PMT 控制和状态寄存器可对请求唤醒事件进行编程，并监控唤醒事件。参见第 26.7.1 节以了解详细信息。

表 533. MAC PMT 控制和状态寄存器 （MAC_PMT_CTRL_STAT，地址 0x4001 002C）位描述

位	符号	描述	复位值	访问类型
0	PD	掉电 该寄存器字段可由应用程序读取 （读操作），可由应用程序通过某个寄存器写入 1 设为 1 （写设置），并由内核清零 （自清除）。应用程序无法清除此类字段，寄存器向该位写入 0 对该字段无任何作用。 设置时，所有接收到的帧将被丢弃。当接收到魔法包或唤醒帧时，该位自动清零，且掉电模式被禁用。该位清零后接收到的帧将被转发到应用程序。只有在将魔法包使能位或唤醒帧使能位设置为高位时，才能对该位进行设置。	0	R/W
1	MPE	魔法包使能 设置时，因魔法包接收而使电源管理事件生成。	0	R/W
2	WFE	唤醒帧使能 设置时，因唤醒帧接收而使电源管理事件生成。	0	R/W
4:3	-	保留	00	RO
5	MPR	魔法包已接收 该寄存器字段可由应用程序读取 （读操作），可由某个内部事件上的以太网内核设为 1（自设置），并在寄存器读取时自动清零。寄存器向该位写入 0 对该字段无任何作用。 设置时，该位表示电源管理事件通过接收魔法包生成。该位通过读取到该寄存器清零。	0	RO
6	WFR	唤醒帧已接收 该寄存器字段可由应用程序读取 （读操作），可由某个内部事件上的以太网内核设为 1（自设置），并在寄存器读取时自动清零。寄存器向该位写入 0 对该字段无任何作用。 如果设置了该位，该位会指示电源管理事件由唤醒帧的接收生成。该位通过读取到该寄存器清零。	0	RO
8:7	-	保留	0	RO
9	GU	全局单播 设置时，所有由 MAC (DAF) 地址识别进行滤波操作的单播数据包都成为唤醒帧。	0	R/W
30:10	-	保留	0x00 0000	RO
31	WFFRPR	唤醒帧滤波寄存器指针复位 该寄存器字段可由应用程序读取 （读操作），可由应用程序通过某个寄存器写入 1 设为 1 （写设置），并由内核清零 （自清除）。应用程序无法清除此类字段，寄存器向该位写入 0 对该字段无任何作用。 设置时，将远程唤醒帧滤波寄存器指针复位至 000。它在 1 个时钟周期后自动清除。	0	R/W

26.6.12 MAC 中断状态寄存器

中断状态寄存器内容可识别 MAC-CORE 中可生成中断的事件。

表 534. MAC 中断状态寄存器 （MAC_INTR，地址 0x4001 0038）位描述

位	符号	描述	复位值	访问类型
31:0	-	保留	0	RO

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.6.13 MAC 中断掩码寄存器

由于中断状态寄存器中的相应事件，中断掩码寄存器的位使用户能够掩码中断信号。

表 535. MAC 中断掩码寄存器 （MAC_INTR_MASK，地址 0x4001 003C）位描述

位	符号	描述	复位值	访问类型
2:0	-	保留	0	RO
3	PMTMSK	PMT 中断掩码 如果设置了该位，则会禁用因表 534 中 PMT 中断状态位的设置导致的中断信号的断言。	0	R/W
31:4		保留	0	R/W

26.6.14 MAC 地址 0 高位寄存器

MAC 地址 0 高位寄存器包含 6 字节首 MAC 站地址的高 16 位。请注意，在 (G)MII 接口上接收到的第一个 DA 字节对应于 MAC 地址低位寄存器的 LS 字节 （位 [7:0]）。例如，如果在 (G)MII 上接收 0x112233445566（0x11 为第一个字节）作为目标地址，则会将 MacAddress0 寄存器 [47:0] 与 0x665544332211 进行比较。

如果将 MAC 地址寄存器配置为对 (G)MII 时钟域进行双同步，则只有在对 MAC 地址低位寄存器 （寄存器 17）的位 [31:24] （在小端模式下）或位 [7:0] （在大端模式下）进行写操作时，才能启动同步。请注意，只有在目标时钟域中经过至少 4 个时钟周期后，才能对此地址低位寄存器连续进行写操作，以进行适当的同步更新。

表 536. MAC 地址 0 高位寄存器 （MAC_ADDR0_HIGH，地址 0x4001 0040）位描述

位	符号	描述	复位值	访问类型
15:0	A47_32	MAC 地址 0 [47:32] 该字段包含 6 字节首 MAC 地址的高 16 位 (47:32)。 MAC 将该字段用于对接收到的帧进行滤波操作，并用于将 MAC 地址插入发送流控制 （暂停）帧。	0xFFFF	R/W
30:16	-	保留	0x0000	RO
31	MO	始终为 1	1	RO

26.6.15 MAC 地址 0 低位寄存器

MAC 地址 0 低位寄存器包含 6 字节首 MAC 站地址的低 32 位。

表 537. MAC 地址 0 低位寄存器 （MAC_ADDR0_LOW，地址 0x4001 0044）位描述

位	符号	描述	复位值	访问类型
31:0	A31_0	MAC 地址 0 [31:0] 该字段包含 6 字节首 MAC 地址的低 32 位。MAC 将该字段用于对接收到的帧进行滤波操作，并用于将 MAC 地址插入发送流控制 （暂停）帧。	0xFFFF FFFF	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.6.16 MAC IEEE1588 时间戳控制寄存器

该寄存器可控制系统时间生成器的操作以及接收器中时间戳 PTP 数据包的探测。

表 538. MAC IEEE1588 时间戳控制寄存器 （MAC_TIMESTP_CTRL，地址 0x4001 0700）位描述

位	符号	描述	复位值	访问类型
0	TSENA	时间戳使能 当对该位进行设置时，时间戳被使能，可发送和接收帧。禁用时，不会添加时间戳用于发送和接收帧，时间戳生成器也暂停。使能该模式后，用户要始终初始化时间戳 （系统时间）。	0	R/W
1	TSCFUPDT	时间戳微调或粗调更新 设置时，表示将使用微调更新方式更新系统时间。复位时，表示将使用粗调方式更新系统时间戳。如果微调校正选项未使能，则该位保留。	0	R/W
2	TSINIT	时间戳初始化 该寄存器字段可由应用程序读取和写入（读操作和写操作），并由以太网内核清零 （自清除）。 设置时， 系统时间利用时间戳高位更新和时间戳低位更新寄存器中指定的值进行初始化 （重写）。该寄存器位应先读零，再更新。初始化完成后，该位复位。	0	R/W
3	TSUPDT	时间戳更新 该寄存器字段可由应用程序读取和写入（读操作和写操作），并由以太网内核清零 （自清除）。 设置时， 系统时间利用时间戳高位更新和时间戳低位更新寄存器中指定的值进行更新 （添加 / 减少）。该寄存器位应先读零，再更新。更新在硬件中完成后，该位复位。	0	R/W
4	TSTRIG	时间戳中断启动使能 该寄存器字段可由应用程序读取和写入（读操作和写操作），并由以太网内核清零 （自清除）。 设置时，时间戳中断在系统时间大于写入目标时间寄存器的值时生成。时间戳启动中断生成后，该位复位。	0	R/W
5	TSADDREG	加数寄存器更新 设置时，时间戳加数寄存器的内容在 PTP 模块中更新，以进行微调校正。更新完成时，该位清零。该寄存器位在设置前应为零。当只选择粗调校正选项时，该位为保留位。		
7:6	-	保留		
8	TSENALL	为所有帧使能时间戳 设置时，将为所有内核接收到的帧使能时间戳快照。	0	R/W
9	TSCTRLSSR	时间戳数字或二进制翻转控制 设置时，时间戳低位寄存器在 0x3B9A_C9FF 值 （即 1 纳秒精度）后翻转并递增时间戳（高位）秒数。复位时，亚秒级寄存器的翻转值为 0x7FFF_FFFF。亚秒递增必须根据 PTP 参考时钟频率和此位值正确编程。	0	R/W
10	TSVER2ENA	使能版本 2 格式的 PTP 数据包探测 设置时，使用 1588 版本 2 格式或版本 1 格式探寻 PTP 数据包。	0	R/W
11	TSIPENA	使能遍布以太网帧的 PTP 时间戳快照 设置时，可为那些在以太网帧（PTP 遍布以太网）中具有 PTP 消息的帧拍摄时间戳快照。根据默认设置，可为 UDP-IP-Ethernet PTP 数据包拍摄快照。	0	R/W
12	TSIPV6ENA	使能 IPv6 帧的时间戳快照 设置时，可为 IPv6 帧拍摄时间戳快照。	0	R/W
13	TSIPV4ENA	使能 IPv4 帧的时间戳快照 设置时，可为 IPv4 帧拍摄时间戳快照。	1	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 538. MAC IEEE1588 时间戳控制寄存器 （MAC_TIMESTP_CTRL，地址 0x4001 0700）位描述 *（续）*

位	符号	描述	复位值	访问类型
14	TSEVNTENA	使能事件消息的时间戳快照 设置时，仅获取事件消息的时间戳快照。复位时，获取除了通告、管理和信令外的所有其他消息的快照。	0	R/W
15	TSMSTRENA	使能主机相关消息的快照 设置时，只能为主节点相关消息拍摄快照，否则为从节点相关消息拍摄快照。这仅对普通时钟和边界时钟节点有效。	0	R/W
17:16	TSCLKTYPE	选择时钟节点的类型 选择时钟节点类型的选项如下： 00 = 普通时钟 01 = 边界时钟 10 = 端对端透明时钟 11 = 点对点透明时钟	00	R/W
18	TSENMACADDR	使能 PTP 帧滤波的 MAC 地址 设置时，使用 DA MAC 地址 （与除了默认 MAC 地址 0 之外的所有 MAC 地址寄存器都匹配）在直接通过以太网发送 PTP 时对 PTP 帧进行滤波操作。	0	R/W
31:19				

表 539 表明，根据时钟获取快照的消息会使能主机，并使能进行事件消息寄存器设置的快照。

表 539. 时间戳快照对寄存器位的依存性

TSCLKTYPE	TSMSTRENA	TSEVNTENA	拍摄快照消息
00 或 01	x	0	SYNC、 Follow_Up、 Delay_Req、 Delay_Resp
00 或 01	1	1	Delay_req
00 或 01	0	1	SYNC
10	不适用	0	SYNC、 Follow_Up、 Delay_Req、 Delay_Resp
10	不适用	1	SYNC、 Follow_Up
11	不适用	0	SYNC、Follow_Up、Delay_Req、Delay_Resp、Pdelay_Req、 Pdelay_Resp
11	不适用	1	SYNC、 Pdelay_Req、 Pdelay_Resp

26.6.17 亚秒递增寄存器

该寄存器包含亚秒寄存器按其递增的 8 位值。

“粗略更新”模式下（表 538 中的 TSCFUPDT 位），该寄存器中的值会添加到每时钟周期系统时间中。在 “微调更新”模式下，一旦累加器溢出，则该寄存器中的值会添加系统时间中。

表 540. 亚秒递增寄存器 （SUBSECOND_INCR，地址 0x4001 0704）位描述

位	符号	描述	复位值	访问类型
7:0	SSINC	亚秒递增值。 该寄存器中编程的值与亚秒寄存器的内容一起累加。例如，如要精度达到 20 ns，则要编程的值为 20。（如果选择 1 ns 精度，则为带 50 MHz 参考时钟的 0x14。）	0	R/W
31:8	-	保留。	0	RO

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.6.18 系统时间秒寄存器

该寄存器包含系统时间秒字段的低 32 位。

系统时间 - 秒寄存器以及系统时间 - 纳秒寄存器，表示内核保持的系统时间的当前值。尽管持续更新，但由于时钟域传输延迟，与实际时间相比仍有延迟。

表 541. 系统时间秒寄存器 （SECONDS，地址 0x4001 0708）位描述

位	符号	描述	复位值	访问类型
31:0	TSS	时间戳秒 该字段内的值表示由内核保持的系统时间的当前值 （单位为秒）。	0	R/W

26.6.19 系统时间纳秒寄存器

该寄存器包含系统时间纳秒字段的 32 位。

表 542. 系统时间纳秒寄存器 （NANOSECONDS，地址 0x4001 070C）位描述

位	符号	描述	复位值	访问类型
30:0	TSSS	时间戳亚秒 该字段内的值以亚秒方式表示时间，精度为 0.46 纳秒。（如果设置了 MAC_TIMESTAMP_CTRL 寄存器中的 TSCTRLSSR，则每个位代表 1 ns 并且最大值会是 0x3B9A_C9FF，在这之后该值会翻转至零）。	0	RO
31	PSNT	正时间或负时间 该位表示的是正时间值或负时间值。如果该位得到复位，则表示时间表示为正；如果进行了设置，则它表示时间值为负。（使能高级时间戳功能时，该位表示纳秒值的第 32 位）。	0	RO

26.6.20 系统时间秒更新寄存器

该寄存器包含了要写入、加入系统时间值或从该值中减去的秒字段的低 32 位。

系统时间 – 秒更新寄存器以及系统时间 – 纳秒更新寄存器，初始化或更新由内核保持的系统时间。在设置时间戳控制寄存器中的 TSINIT 或 TSUPDT 位之前，必须对这两个寄存器进行写操作。

表 543. 系统时间秒更新寄存器 （SECONDSUPDATE，地址 0x4001 0710）位描述

位	符号	描述	复位值	访问类型
31:0	TSS	时间戳秒 该字段内的值表示要初始化或添加至系统时间的时间（单位为秒）。	0	R/W

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.6.21 系统时间纳秒更新寄存器

该寄存器包含了要写入、加入系统时间值或从该值中减去的纳秒字段的 32 位。

表 544. 系统时间纳秒更新寄存器 （NANOSECONDSUPDATE，地址 0x4001 0714）位描述

位	符号	描述	复位值	访问类型
30:0	TSSS	时间戳亚秒 该字段内的值以亚秒方式表示时间，精度为 0.46 纳秒。（如果在时间戳控制寄存器中设置了TSCTRLSSR，则每个位代表 1 ns 并且编程值不应超过 0x3B9A_C9FF。）	0	R/W
31	ADDSUB	加入或减去时间 如果设置了该位，则根据更新寄存器的内容来减去时间值。如果该位得到复位，则根据更新寄存器的内容来加入时间值。	0	R/W

26.6.22 时间戳加数寄存器

软件使用该寄存器线性重新调整时钟频率以匹配主时钟频率。

仅当系统时间配置为“微调更新”模式时才使用该寄存器值 （表 538 中的 TSCFUPDT 位）。在每个时钟周期中，该寄存器内容会添加至 32 位累加器，而一旦累加器溢出，则会更新系统时间。

表 545. 时间戳加数寄存器 （ADDEND，地址 0x4001 0718）位描述

位	符号	描述	复位值	访问类型
31:0	TSAR	时间戳加数 该寄存器表示添加至累加器寄存器以实现时间同步的 32 位时间值。	0	R/W

26.6.23 目标时间秒寄存器

该寄存器包含与系统时间相比较的时间的高 32 位 （用于生成中断事件）。

系统时间超过目标时间秒寄存器和目标时间纳秒寄存器的编程值时，这些寄存器用于调度中断事件（使能高级时间戳记时为表549中的TSTARGT位，否则为表534中的TS中断位）。

表 546. 目标时间秒寄存器 （TARGETSECONDS，地址 0x4001 071C）位描述

位	符号	描述	复位值	访问类型
31:0	TSTR	目标时间秒寄存器 该寄存器存储以秒为单位的时间。如果时间戳值匹配或超出两个目标时间戳寄存器，那么一旦使能了 MAC，就会生成中断。	0	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.6.24 目标时间纳秒寄存器

该寄存器包含与系统时间相比较的时间的高 32 位 （用于生成中断事件）。

表 547. 目标时间纳秒寄存器 （TARGETNANOSECONDS，地址 0x4001 0720）位描述

位	符号	描述	复位值	访问类型
30:0	TSTR	目标时间戳低位 该寄存器存储以 （有符号）纳秒为单位的时间。如果时间戳的值与两个目标时间戳寄存器匹配，那么一旦使能了 MAC，就会生成中断。（在时间戳控制寄存器中设置了 TSCTRLSSR 时，该值不应超过 0x3B9A_C9FF。）	0	RO
31	-	保留。	-	-

26.6.25 系统时间高字秒寄存器

该寄存器包含时间戳秒值的最高有效 16 位。

表 548. 系统时间高字秒寄存器 （HIGHWORD，地址 0x4001 0724）位描述

位	符号	描述	复位值	访问类型
15:0	TSHWR	时间戳高字 包含时间戳秒值的最高有效 16 位。直接对该寄存器进行写操作以将值初始化。如果系统时间 - 秒寄存器的 32 位溢出，则该寄存器会递增。	0	R/W
31:16	-	保留。	-	-

26.6.26 时间戳状态寄存器

该寄存器包含 PTP 状态。在主机读取该寄存器后，除了位 [27:25] 之外的所有位都会清除。

寄存器字段可由应用程序读取 （读操作），可由某个内部事件的内核设为 1 （自设置），并在寄存器读取时自动清零。寄存器向该位写入 0 对该字段无任何作用。

表 549. 时间戳状态寄存器 （TIMESTAMPSTAT，地址 0x4001 0728）位描述

位	符号	描述	复位值	访问类型
0	TSSOVF	时间戳秒溢出 设置时，表示时间戳的秒值（支持版本 2 格式时）已经溢出 0xFFFF_FFFF 之外。	0	R/W
1	TSTARGT	达到时间戳目标 设置时，表示系统时间值大于或等于目标时间高位和低位寄存器中指定的值。	0	R/W
2	AUXSS	辅助时间戳启动快照 辅助快照写入 FIFO 时，该位设为高位。	0	R/W
23:3	-	保留。	-	-
24	ATSSTM	辅助时间戳快照启动丢失 辅助时间戳快照 FIFO 为满状态且设置了外部启动时，设置该位。这表示在 FIFO 中未存储最新的快照。	0	R/W
27:25	ATSNS	辅助时间戳的快照数 这些位表示 FIFO 中可用的快照数。如果值为 4 (100)，则表示辅助快照 FIFO 为满状态。如果设置了辅助快照 FIFO 清除位，这些位会清除 （为 000）。	0	R/W
31:28	-	保留。	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.6.27 PPS 控制寄存器

该寄存器用来控制 PPS 信号输出的间隔，这样就可在脉冲间获得小于 1 秒的持续时间。

表 550. PPS 控制寄存器 （PPSCTRL，地址 0x4001 072C）位描述

位	符号	描述	复位值	访问类型
3:0	PPSCTRL	控制 PPS 信号输出的两个脉冲之间的持续时间 该位控制 PPS 输出信号的两个脉冲之间的持续时间。默认情况下，脉冲间的控制设为 1 秒。有效设置的脉冲持续时间在下文中提及。 0000 = 1.0 秒 （二进制和数字翻转） 0001 = 0.5 秒 （二进制翻转）， 0.536 秒 （数字翻转） 0010 = 0.25 秒 （二进制翻转）， 0.26 秒 （数字翻转） 0011 = 0.125 秒 （二进制翻转）， 0.13 秒 （数字翻转） ... 1111 = 15.28 μ 秒（二进制翻转），32.77? 秒（数字翻转）	0	R/W
31:4	-	保留。	-	-

26.6.28 辅助时间戳纳秒寄存器

该寄存器包含辅助时间戳寄存器的低 32 位 （纳秒字段）。

该寄存器与辅助时间戳秒寄存器 （表 552）使 64 位时间戳以辅助快照的形式存储。两个寄存器共同形成 4 深度 64 位宽 FIFO 的读取端口。在该 FIFO 中可存储多个快照，并且该 FIFO 的填充水平由时间戳状态寄存器中的 ATSNS 位表示。仅当读取表 552 的最后一个字节时，才移除 FIFO 的最高字节。在小端模式下，这表示在大端模式下读取位 [31:24] 时，它与读取表 552 的位 [7:0] 相对应。

表 551. 辅助时间戳纳秒寄存器 （AUXNANOSECONDS，地址 0x4001 0730）位描述

位	符号	描述	复位值	访问类型
31:0	AUXNS	辅助时间戳的低 32 位 （纳秒字段）。	0	RO

26.6.29 辅助时间戳秒寄存器

该寄存器包含辅助时间戳寄存器秒字段的低 32 位。

表 552. 辅助时间戳秒寄存器 （AUXSECONDS，地址 0x4001 0734）位描述

位	符号	描述	复位值	访问类型
31:0	AUXS	辅助时间戳秒字段的低 32 位。	0	RO

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.6.30 DMA 总线模式寄存器

总线模式寄存器可为 DMA 建立总线操作模式。

表 553. DMA 总线模式寄存器 （DMA_BUS_MODE，地址 0x4001 1000）位描述

位	符号	描述	复位值	访问类型
0	SWR	软件复位 该寄存器字段可由应用程序读取 （读操作），可由应用程序通过某个寄存器写入 1 设为 1 （写设置），并由内核清零 （自清除）。应用程序无法清除此类字段，寄存器向该位写入 0 对该字段无任何作用。 此位设置时， MAC DMA 控制器复位所有的 MAC 子系统内部寄存器和逻辑。复位操作在所有内核时钟域中完成后，该位自动清零。在对任何内核寄存器进行重新编程之前，读取该位的 0 值。 **注：**仅在所有有效时钟域的所有复位都取消时，复位操作才能完成。因此，所有 PHY 输入时钟 （适用于所选的 PHY 接口）都必须存在，以完成软件复位，这很重要。	0	R/W
1	DA	DMA 仲裁方案 0 = 位 [15:14] 中指定的 Rx:Tx 循环优先 1 = Rx 优先于 Tx	0	R/W
6:2	DSL	描述符跳过长度 该位规定了在两个未解开链接的描述符之间跳过的字/双字/长字数（取决于32/64/128位总线）。跳址是从当前描述符的末端开始，跳到下一个描述符的始端。如果 DSL 的值等于零，则在环状模式下， DMA 将描述符表视为相邻。	0	R/W
7	ATDS	交替描述符大小 设置时，备用描述符 （参见第 26.10.3 节）的大小会增加到 32 字节 （8 个双字）。在接收器中使能高级时间戳功能或满 IPC 卸载引擎时，这个是必需的。 复位时，描述符大小还原成 4 双字 （16 字节）。	0	R/W
13:8	PBL	可编程连发长度 这些位表示将要在一次 DMA 事务处理中传输的最大节拍数。这将是用于单块读 / 写操作的最大值。每次启动主机总线上的连发传输时， DMA 将始终尝试按照 PBL 的规定进行连发。PBL 可用允许值 1、2、4、8、16 和 32 编程。任何其它值将导致未定义的行为。当将 USP 设置为高位时，该 PBL 值仅适用于 TxDMA 事务处理。 PBL 值具有以下限制。 最大可能节拍数 (PBL) 受到 MTL 层中 Tx FIFO 和 Rx FIFO 的大小以及 DMA 上数据总线宽度的限制。FIFO 具有一个限制因素，即所支持的最大节拍是 FIFO 深度的二分之一，除指定外 （如下所述）。针对不同的数据总线宽度和 FIFO 大小，下表提供了有效的 PBL 范围 （包括 x8 模式）。如果 PBL 为发送和接收 DMA 通用值，则必需考虑 Rx FIFO 和 Tx FIFO 的最小深度。切勿对超出范围的 PBL 值进行编程，因为系统可能无法正常运行。	1	R/W
15:14	PR	Rx 比 Tx 的优先级比率 根据以下比率，RxDMA 请求的优先级高于 TxDMA 请求。这只在 DA 位复位时才有效。 00 = 1:1 01 = 2:1 10 = 3:1 11 = 4:1	00	R/W
16	FB	固定连发 该位可控制 AHB 主接口是否执行固定连发传输。置位后， AHB 将在正常连发传输的开始阶段仅使用 SINGLE、INCR4、INCR8 或 INCR16。复位后，AHB 将使用 SINGLE 和 INCR 连发传输操作。	0	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 553. DMA 总线模式寄存器 （DMA_BUS_MODE，地址 0x4001 1000）位描述 *（续）*

位	符号	描述	复位值	访问类型
22:17	RPBL	RxDMA PBL 这些位表示将要在一次 RxDMA 事务处理中传输的最大节拍数。这将是用于单块读 / 写操作的最大值。每次启动主机总线上的连发传输时， RxDMA 将始终尝试按照 RPBL 的规定进行连发。RPBL 可用允许值 1、2、4、8、16 和 32 编程。任何其它值将导致未定义的行为。仅在将 USP 设置为高位时，这些位才有效且适用。	1	R/W
23	USP	使用不同的 PBL 当设置为高位时，它将 RxDMA 配置为将位 [22:17] 中配置的值用作 PBL 值，而位 [13:8] 中的 PBL 值仅适用于 TxDMA 操作。当复位至低位时，位 [13:8] 中的 PBL 值适用于两个 DMA 引擎。	0	R/W
24	PBL8X	8 x PBL 模式 当设置为高位时，该位可使已编程的 PBL 值 （位 [22:17] 和位 [13:8]）增长 8 倍。这样 DMA 将根据 PBL 值将数据传输到最大节拍 8、 16、 32、 64、 128 和 256 中。 **注：**该位的功能不能向下兼容。在 3.50a 之前的版本中该位是 4xPBL。	0	R/W
25	AAL	地址对齐节拍 当此位被设置为高电平且 FB 位等于 1，AHB 接口生成的所有连发都与起始地址 LS 位对齐。如果 FB 位等于 0，则第一次连发 （访问数据缓冲区的起始地址）无法对齐，但后续连发将对齐到地址。	0	R/W
26	MB	混合连发 当把该位设置为高位，且 FB 位为低位时，AHB 主接口将启动含 INCR（未定义连发）的长度超过 16 的所有连发；反之，它将恢复为连发长度为 16 或 16 以下的固定连发传输 （INCRx 和 SINGLE）。	0	R/W
27	TXPR	当其被设置，此位表明在系统侧总线仲裁期间，发送 DMA 具有比接受 DMA 更高的优先级。	0	R/W
31:28	-	保留	0	RO

表 554. 可编程连发长度设置

数据总线宽度	FIFO 深度	全双工模式下的有效 PBL 范围
32 位	128 字节	8 或更小
	256 字节	32 或更小
	512 字节	64 或更小
	1 kB	128 或更小
	2 kB 及 2 kB 以上	所有

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.6.31 DMA 发送轮询请求寄存器

发送轮询请求寄存器使发送 DMA 能够检查当前描述符是否由 DMA 所有。如果处于暂停模式，则发出发送轮询请求命令即可唤醒 TxDMA。由于已发送的帧中出现下溢错误，或由于发送 DMA 所拥有的描述符不可用，所以 TxDMA 可能进入暂停模式。您可以随时发出该命令，而且一旦它开始从主机存储器重新提取当前描述符， TxDMA 就会将该命令复位。

表 555. DMA 发送轮询请求寄存器（DMA_TRANS_POLL_DEMAND，地址 0x4001 1004）位描述

位	符号	描述	复位值	访问类型
31:0	TPD	发送轮询请求 该寄存器字段可由应用程序读取，采用任何数据值进行写操作时，会启动某个事件。 当这些位被写入任何值， DMA 都读取 “当前主机发送描述符”寄存器指向的当前描述符 （第 26.6.40 节）。如果该描述符无效（被主机所拥有），则发送返回挂起状态且 DMA_STAT 寄存器中的位 2 被断言。如果描述符可用，发送恢复。	0	R/W

26.6.32 DMA 接收轮询请求寄存器

接收轮询请求寄存器使接收 DMA 能够检查新描述符。发出该命令可以从暂停状态唤醒 RxDMA。 RxDMA 可能进入暂停状态，仅因为它所拥有的描述符不可用。

表 556. DMA 接收轮询请求寄存器 （DMA_REC_POLL_DEMAND，地址 0x4001 1008）位描述

位	符号	描述	复位值	访问类型
31:0	RPD	接收轮询请求 该寄存器字段可由应用程序读取，采用任何数据值进行写操作时，会启动某个事件。 当这些位被写入任何值， DMA 都会读取 “当前主机接收描述符”寄存器指向的当前描述符 (第 26.6.41 节)。如果改描述符无效（被主机所拥有），则接收返回挂起状态且 DMA_STAT 寄存器中的位 7 被断言。如果描述符有效，则接收 DMA 返回有效状态。	0	R/W

26.6.33 DMA 接收描述符列表地址寄存器

接收描述符列表地址寄存器指向接收描述符列表的起始位置。描述符列表位于主机的物理内存空间，而且一定是字 / 双字 / 长字对齐（适用于 32/64/128 位数据总线）。DMA 通过使相应的 LS 位变为低位，从内部将其转换为总线宽度对齐的地址。只有在接收停止时，才允许对该寄存器进行写操作。停止时，必须在发出接收开始命令之前对该寄存器进行写操作。

表 557. DMA 接收描述符列表地址寄存器（DMA_REC_DES_ADDR，地址 0x4001 100C）位描述

位	符号	描述	复位值	访问类型
31:0	SRL	接收列表的起始位置 此字段包含 “接收描述符列表”中首个描述符的基址。LSB 位 [1/2/3:0]（对于 32/64/128 位总线宽度）被忽略，DMA 内部会将其视为全 0。因此，这些 LSB 位为只读。	0	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.6.34 DMA 发送描述符列表地址寄存器

发送描述符列表地址寄存器指向发送描述符列表的起始位置。描述符列表位于主机的物理内存空间，而且一定是字 / 双字 / 长字对齐（适用于 32/64/128 位数据总线）。DMA 通过使相应的 LSB 变为低位，从内部将其转换为总线宽度对齐的地址。只有在发送停止时，才允许对该寄存器进行写操作。停止时，可以在发出发送开始命令之前对该寄存器进行写操作。

表 558. DMA 发送描述符列表地址寄存器（DMA_TRANS_DES_ADDR，地址 0x4001 1010）位描述

位	符号	描述	复位值	访问类型
31:0	SRL	发送列表的起始位置 此字段包含“发送描述符列表”中首个描述符的基址。LSB 位 [1/2/3:0]（对于 32/64/128 位总线宽度）被忽略，DMA 内部会将其视为全 0。因此，这些 LSB 位为只读。	0	R/W

26.6.35 DMA 状态寄存器

状态寄存器包含 DMA 向主机报告的所有状态位。在中断服务例程或轮询中，通常由软件驱动程序对该寄存器进行读操作。该寄存器中的大多数字段都会导致主机中断。该寄存器中的位在进行读操作时不会清零。向该寄存器 （位 [16:0]）中的 （非保留）位写 1 清零，写 0 无效。通过掩码 DMA_INT_EN 寄存器中相应的位，可以掩码每个字段 （位 [16:0]）。

该寄存器的该字段可由应用程序读取 （读操作），可由某个内部事件上的以太网内核设为 1 （自设置），并可由应用程序通过某个寄存器写入 1 清零（写清除）。向寄存器写入 0 对该字段无任何作用。

表 559. DMA 状态寄存器 （DMA_STAT，地址 0x4001 1014）位描述

位	符号	描述	复位值	访问类型
0	TI	发送中断 该位表示帧发送结束， TDES1[31] 已在第一个描述符中进行设置。	0	R/W
1	TPS	发送过程停止 该位在发送停止时进行设置。	0	RW
2	TU	发送缓冲区无效 该位表示发送列表中的下一个描述符为主机所有，且 DMA 无法获取。发送暂停。位 [22:20] 说明了发送过程状态转换。若要继续处理发送描述符，主机应更改描述符位的所有权，然后发出发送轮询请求命令。	0	R/W
3	TJT	发送 Jabber 超时 该位表示发送 Jabber 定时器失效，意味着发送器过度活跃。发送过程被中止并置于停止状态。这使得发送 Jabber 超时 TDES0[14] 标志生效。	0	R/W
4	OVF	接收溢出 该位表示接收缓冲区在帧接收过程中溢出。如果将部分帧传输到应用程序，则要在 RDES0[11] 中设置溢出状态。	0	R/W
5	UNF	发送下溢 该位表示发送缓冲区在帧发送过程中下溢。发送暂停，且要对下溢错误 TDES0[1] 进行设置。	0	R/W
6	RI	接收中断 该位表示帧接收完成。特定帧的状态消息已在描述符中列出。接收仍处于运行状态。	0	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 559. DMA 状态寄存器 （DMA_STAT，地址 0x4001 1014）位描述 *（续）*

位	符号	描述	复位值	访问类型
7	RU	接收缓冲区无效 该位表示接收列表中的下一个描述符为主机所有，且 DMA 无法获取。接收过程暂停。若要继续处理接收描述符，主机应更改描述符的所有权，然后发出接收轮询请求命令。如果未发出接收轮询请求，则接收过程在接收到下一个可识别输入帧时恢复。仅在前一个接收描述符为 DMA 所有时，才对该位进行设置。	0	R/W
8	RPS	接收过程停止 接收过程进入停止状态时，该位生效。	0	R/W
9	RWT	接收看门狗超时 所接收帧的长度大于 2,048 字节（巨型帧模式使能时为 10,240 字节）时，该位生效。	0	R/W
10	ETI	超前发送中断 该位表示将要发送的帧已全部传输到 MTL 发送 FIFO。	0	R/W
12:11	-	保留	0	RO
13	FBI	致命总线错误中断 该位表示发生总线错误，详见位 [25:23]。当对该位进行设置时，相应的 DMA 引擎将禁用其所有总线访问。	0	R/W
14	ERI	超前接收中断 此位表明 DMA 已填充数据包的首个数据缓冲区。此寄存器中的接收中断位 6 自动清除此位。	0	R/W
15	AIE	异常中断摘要 当在 DMA_INT_EN 寄存器中使能相应的中断位时，异常中断摘要位值就为以下各项的逻辑或： DMA_STAT 寄存器，位 1：发送过程停止 DMA_STAT 寄存器，位 3：发送 Jabber 超时 DMA_STAT 寄存器，位 4：接收溢出 DMA_STAT 寄存器，位 5：发送下溢 DMA_STAT 寄存器，位 7：接收器缓冲区不可用 DMA_STAT 寄存器，位 8：接收进程停止 DMA_STAT 寄存器，位 9：接收看门狗超时 DMA_STAT 寄存器，位 10：超前发送中断 DMA_STAT 寄存器，位 13：致命总线错误 只有非掩码位会影响异常中断摘要位。 这是一个粘滞位，每次引起 AIS 置位的对应位清除时，该位也必须清除。	0	R/W
16	NIS	正常中断摘要 当在 DMA_INT_EN 寄存器中使能相应的中断位时，正常中断摘要位值就为以下各项的逻辑或： DMA_STAT 寄存器，位 0：发送中断 DMA_STAT 寄存器，位 2：发送缓冲区无效 DMA_STAT 寄存器，位 6：接收中断 DMA_STAT 寄存器，位 14：超前接收中断 只有非掩码位会影响正常中断摘要位。 这是一个粘滞位，每次引起 NIS 设置的对应位清除时，该位也必须清除（通过向此位写入 1）。	0	R/W
31:17	-	保留	0	RO

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.6.36 DMA 操作模式寄存器

操作模式寄存器可建立发送和接收操作模式及命令。此寄存器作为 DMA 初始化的一部分，应该是最后要写入的 CSR。

表 560. DMA 操作模式寄存器 （DMA_OP_MODE，地址 0x4001 1018）位描述

位	符号	描述	复位值	访问类型
0	-	保留	0	RO
1	SR	开始 / 停止接收 此位设置时，接收进程被置于“运行”状态。DMA 试图从接收列表获取描述符，并处理输入帧。描述符获取操作试图从列表的当前位置 （即 DMA_REC_DES_ADDR 寄存器设置的地址）进行，或从先前接收过程停止时保留的位置进行。如果 DMA 不拥有描述符，则接收挂起且 “接收缓冲区不可用”位 （DMA_STAT 寄存器中的位 7）被设置。开始接收命令仅在接收停止时有效。如果该命令在设置 DMA_REC_DES_ADDR 前发出，则 DMA 的特性不可预测。	0	R/W
2	OSF	第二帧的操作 当对该位进行设置时，该位甚至在保留第一帧的状态之前就命令 DMA 处理发送数据的另一帧。	0	R/W
4:3	RTC	接收阈值控制 这两个位可控制 MTL 接收 FIFO 的阈值电平。当 MTL 接收 FIFO 中的帧尺寸大于阈值时，对 DMA 的传输 （请求）开始。此外，长度小于阈值的全帧会自动传输。请注意，如果已配置的接收 FIFO 的大小为 128 字节，则值 11 不适用。这些位仅在 RSF 位为 0 时才有效，当 RSF 位被设置为 1 时，将被忽略。 00 = 64 01 = 32 10 = 96 11 = 128	0	R/W
5	-	保留	0	RO
6	FUF	转发尺寸不足的好帧 设置时， Rx FIFO 将转发尺寸不足的帧 （无错误的和长度小于 64 字节的帧），包括填充字节和 CRC。 复位时，Rx FIFO 将丢弃所有小于 64 字节的帧，除非它已经因接收阈值的下限值（例如， RTC = 01）而传输。	0	R/W
7	FEF	转发错误帧 当该位复位时， Rx FIFO 将丢弃有错误状态 （CRC 错误、冲突错误、 GMII_ER、巨型帧、看门狗超时、溢出）的帧。但是，如果帧的起始字节 （写）指针已被传输到读取控制器侧 （在阈值模式下），则不丢弃帧。如果设置了 FEF，则除了超短错误帧之外的所有帧都被转发到 DMA。但如果 RxFIFO 在写入部分帧时溢出，则此类帧将被丢弃，即使已经对 FEF 进行设置。	0	R/W
12:8	-	保留	0	RO
13	ST	开始 / 停止发送命令 当对该位进行设置时，发送被置于运行状态， DMA 检查当前位置将要发送的帧发送列表。描述符获取操作试图从列表的当前位置（即 DMA_TRANS_DES_ADDR 寄存器设置的发送列表基址）进行，或从先前发送停止时保留的位置进行。如果当前描述符并非为 DMA 所有，则发送进入暂停状态，那么要对发送缓冲区无效（DMA_STAT 寄存器，位 2）进行设置。开始发送命令仅在发送停止时有效。如果在设置 DMA_TRANS_DES_ADDR 寄存器之前发出命令，则 DMA 的行为不可预测。 当该位复位时，发送过程在完成当前帧的发送之后将被置于暂停状态。发送列表中下一个描述符的位置被保存，并成为发送重新开始时的当前位置。停止发送命令仅在当前帧的发送完成后或发送处于暂停状态时有效。	0	R/W

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 560. DMA 操作模式寄存器 （DMA_OP_MODE，地址 0x4001 1018）位描述 *（续）*

位	符号	描述	复位值	访问类型
16:14	TTC	发送阈值控制 这三个位可控制 MTL 发送 FIFO 的阈值电平。当 MTL 发送 FIFO 中的帧尺寸大于阈值时，发送开始。此外，长度小于阈值的全帧也会被发送。这些位仅在 TSF 位 （位 21）复位时才使用。 000 = 64 001 = 128 010 = 192 011 = 256 100 = 40 101 = 32 110 = 24 111 = 16	0	R/W
19:17	-	保留	0	RO
20	FTF	刷新发送 FIFO 该寄存器字段可由应用程序读取 （读操作），可由应用程序通过某个寄存器写入 1 设为 1 （写设置），并由以太网内核清零 （自清除）。应用程序无法清除此类字段，寄存器向该位写入 0 对该字段无任何作用。 当对该位进行设置时，发送 FIFO 控制器逻辑复位为其默认值，因此 Tx FIFO 中的所有数据丢失 / 刷新。当刷新操作全部完成后，该位从内部清零。在该位清零之前，不能对操作模式寄存器进行写操作。MAC 发送器已经接受的数据不会刷新。它将按预定计划进行发送，可能发生下溢和超短帧发送。 **注：**刷新操作只有在清空其内容的 TxFIFO 后才算完成，而主机将接受已发送帧的所有挂起发送状态。为了完成该刷新操作，要求激活 PHY 发送时钟。	0	R/W
21	TSF	发送存储和转发 当对该位进行设置时，发送将在全帧位于 MTL 发送 FIFO 时开始。当对该位进行设置时，该寄存器 （位 [16:14]）中指定的 TTC 值被忽略。该位仅在发送停止时才应进行更改。	0	R/W
23:22	-	保留	0	RO
24	DFF	禁止刷新已接收的帧 当对该位进行设置时，由于接收描述符 / 缓冲区不可用，RxDMA 不会刷新任何帧；当该位复位时，如正常情况一样进行操作。（参见）。	0	R/W
25	RSF	接受存储和转发 当对该位进行设置时， MTL 仅在完整帧写入之后从 Rx FIFO 读取帧，忽略 RTC 位。当该位复位时， Rx FIFO 在直通模式下操作，受 RTC 位指定阈值的影响。	0	R/W
26	DT	禁止丢弃 TCP/IP 校验和错误帧 当对该位进行设置时，内核不会丢弃仅具有接受校验和卸载引擎检测到的错误的帧。此类帧在 MAC 接收到的以太网帧中没有任何错误（包括 FCS 错误），但仅在封装的有效负载中有错误。当该位复位时，如果 FEF 位复位，则将丢弃所有错误帧。	0	R/W
31:27	-	保留	0	RO

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.6.37 DMA 中断使能寄存器

中断使能寄存器可使能 DMA_STAT 寄存器报告的中断。将一个位设置为 1 可使能相应中断。硬件或软件复位后，所有中断禁用。

表 561. DMA 中断使能寄存器 （DMA_INT_EN，地址 0x4001 101C）位描述

位	符号	描述	复位值	访问类型
0	TIE	发送中断使能 当此位与 “正常中断汇总使能”（此寄存器中的位 16）一同设置时，“发送中断”使能。此位复位时，“发送中断”禁用。	0	R/W
1	TSE	发送停止使能 当此位与“异常中断汇总使能”（此寄存器中的位 15）一同设置时，“发送停止中断”使能。此位复位时，“发送停止中断”禁用。	0	R/W
2	TUE	发送缓冲区无效使能 当此位与 “正常中断汇总使能”（此寄存器中的位 16）一同设置时，“发送缓冲区不可用中断”使能。此位复位时，“发送缓冲区不可用中断”被禁用。	0	R/W
3	TJE	发送 Jabber 超时使能 当此位与“异常中断汇总使能”（此寄存器中的位 15）一同设置时，“发送 Jabber 超时中断”使能。此位复位时，“发送 Jabber 超时中断”禁用。	0	R/W
4	OVE	溢出中断使能 当此位与“异常中断汇总使能”（此寄存器中的位 15）一同设置时，“接收溢出中断”使能。此位复位时，“溢出中断”禁用。	0	R/W
5	UNE	下溢中断使能 当此位与“异常中断汇总使能”（此寄存器中的位 15）一同设置时，“发送下溢中断”使能。此位复位时，“下溢中断”禁用。	0	R/W
6	RIE	接收中断使能 当此位与 “正常中断汇总使能”（此寄存器中的位 16）一同设置时，“接收中断”使能。此位复位时，“接收中断”禁用。	0	R/W
7	RUE	接收缓冲区无效使能 当此位与 “异常中断汇总使能”（此寄存器中的位 15）一同设置时，“接收缓冲区不可用中断”使能。此位复位时，“接收缓冲区不可用中断”禁用。	0	R/W
8	RSE	接收停止使能 当此位与“异常中断汇总使能”（此寄存器中的位 15）一同设置时，“接收停止中断”使能。此位复位时，“接收停止中断”禁用。	0	R/W
9	RWE	接收看门狗超时使能 当此位与 “异常中断汇总使能”（此寄存器中的位 15）一同设置时，“接收看门狗超时中断”使能。此位复位时，“接收看门狗超时中断”禁用。	0	R/W
10	ETE	超前发送中断使能 当此位与“异常中断汇总使能”（此寄存器中的位 15）一同设置时，“早期发送中断”使能。此位复位时，“早期发送中断”禁用。	0	R/W
12:11	-	保留	0	RO
13	FBE	致命总线错误使能 当此位与 “异常中断汇总使能”（此寄存器中的位 15）一同设置时，“致命总线错误中断”使能。此位复位时，“致命总线错误使能中断”禁用。	0	R/W
14	ERE	超前接收中断使能 当此位与“正常中断汇总使能”（此寄存器中的位 16）一同设置时，“早期接收中断”使能。此位复位时，“早期接收中断”禁用。	0	R/W

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 561. DMA 中断使能寄存器 （DMA_INT_EN，地址 0x4001 101C）位描述 *（续）*

位	符号	描述	复位值	访问类型
15	AIE	异常中断摘要使能 当对该位进行设置时，异常中断被使能。当该位复位时，异常中断被禁用。该位可使能下列位 DMA_STAT 寄存器，位 1：发送过程停止 DMA_STAT 寄存器，位 3：发送 Jabber 超时 DMA_STAT 寄存器，位 4：接收溢出 DMA_STAT 寄存器，位 5：发送下溢 DMA_STAT 寄存器，位 7：接收器缓冲区不可用 DMA_STAT 寄存器，位 8：接收进程停止 DMA_STAT 寄存器，位 9：接收看门狗超时 DMA_STAT 寄存器，位 10：超前发送中断 DMA_STAT 寄存器，位 13：致命总线错误	0	R/W
16	NIE	正常中断摘要使能 当对该位进行设置时，正常中断被使能。当该位复位时，正常中断被禁用。该位使能以下位： DMA_STAT 寄存器，位 0：发送中断 DMA_STAT 寄存器，位 2：发送缓冲区无效 DMA_STAT 寄存器，位 6：接收中断 DMA_STAT 寄存器，位 14：超前接收中断	0	R/W
31:17	-	保留	0	RO

中断 (sbd_intr_o_interrupt) 如图 61 所示生成。当 NIS/AIS 状态位被断言且相应的中断使能位 (NIE/AIE) 被使能时，它被断言。

图 61. 中断生成

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.6.38 DMA 遗漏帧和缓冲区溢出计数寄存器

DMA 支持两个计数器，以跟踪接收过程中遗漏的帧数。该寄存器可报告计数器的当前值。计数器用于诊断用途。位 [15:0] 表示由于主机缓冲区无效而遗漏的帧。位 [27:17] 表示由于缓冲区溢出条件（MTL和MAC）而遗漏的帧以及MTL丢弃的超短帧（小于64字节的好帧）。

表 562. DMA遗漏帧和缓冲区溢出计数寄存器（DMA_MFRM_BUFOF，地址0x4001 1020）位描述

位	符号	描述	复位值	访问类型
15:0	FMC	遗漏帧数 该寄存器字段可由应用程序读取（读操作），可由某个内部事件上的以太网内核设为 1（自设置），并在寄存器读取时自动清零。寄存器向该位写入 0 对该字段无任何作用。 表示控制器由于主机接收缓冲区无效而遗漏的帧数。每次 DMA 丢弃输入帧时，计数器就会增大。当使用 对该寄存器进行读操作时，计数器清零。	0	RO
16	OC	遗漏帧计数器的溢出位 该寄存器字段可由应用程序读取（读操作），可由某个内部事件上的以太网内核设为 1（自设置），并在寄存器读取时自动清零。寄存器向该位写入 0 对该字段无任何作用。	0	RO
27:17	FMA	应用程序遗漏的帧数 该寄存器字段可由应用程序读取（读操作），可由某个内部事件上的以太网内核设为 1（自设置），并在寄存器读取时自动清零。寄存器向该位写入 0 对该字段无任何作用。 表示应用程序遗漏的帧数。每次 MTL 断言旁带信号 mtl_rxoverflow_o，此计数器递增一次。使用 <tbd> 读取此寄存器时，计数器被清除。	0	RO
28	OF	FIFO 溢出计数器的溢出位 该寄存器字段可由应用程序读取（读操作），可由某个内部事件上的以太网内核设为 1（自设置），并在寄存器读取时自动清零。寄存器向该位写入 0 对该字段无任何作用。	0	RO
31:29	-	保留	0	RO

26.6.39 DMA 接收中断看门狗定时器寄存器

当此寄存器被写入非零值时，将为 RI 使能看门狗定时器 （DMA_STAT 寄存器的位 6）。

表 563. DMA 接收中断看门狗定时器寄存器（DMA_REC_INT_WDT，地址 0x4001 1024）位描述

位	符号	描述	复位值	访问类型
7:0	RIWT	RI 看门狗超时 表示为看门狗定时器设置的乘以 256 的系统时钟周期数。在 RxDMA 传输完因相应描述符 RDES1[31] 中的设置而未设置 RI 状态位的帧之后，使用编程值可启动看门狗定时器。当看门狗定时器失效时，RI 位被设置，定时器停止。根据所有已接收帧的 RDES1[31]，由于 RI 的自动设置，当将 RI 位设置为高位时，看门狗定时器复位。	0	R/W
31:8	-	保留	0	RO

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.6.40 DMA 当前主机发送描述符寄存器

“当前主机发送描述符”寄存器指向 DMA 读取的当前发送描述符的起始地址。

表 564. DMA 当前主机发送描述符寄存器（DMA_CURHOST_TRANS_DES，地址 0x4001 1048）位描述

位	符号	描述	复位值	访问类型
31:0	HTD	主机发送描述符地址指针 复位时清除。指针在操作过程中由 DMA 更新。	0	RO

26.6.41 DMA 当前主机接收描述符寄存器

当前主机接收描述符寄存器指向 DMA 读取的当前接收描述符的起始地址。

表 565. DMA 当前主机接收描述符寄存器（DMA_CURHOST_REC_DES，地址 0x4001 104C）位描述

位	符号	描述	复位值	访问类型
31:0	HRD	主机接收描述符地址指针。 复位时清除。指针在操作过程中由 DMA 更新。	0	RO

26.6.42 DMA 当前主机发送缓冲区地址寄存器

“当前主机发送缓冲区地址”寄存器指向 DMA 正在读取的当前发送缓冲区地址。

表 566. DMA 当前主机发送缓冲区地址寄存器（DMA_CURHOST_TRANS_BUF，地址 0x4001 1050）位描述

位	符号	描述	复位值	访问类型
31:0	HTB	主机发送缓冲区地址指针 复位时清除。指针在操作过程中由 DMA 更新。	0	RO

26.6.43 DMA 当前主机接收缓冲区地址寄存器

当前主机接收缓冲区地址寄存器指向 DMA 正在读取的当前接收缓冲区地址。

表 567. DMA 当前主机接收缓冲区地址寄存器（DMA_CURHOST_TRANS_BUF，地址 0x4001 1054）位描述

位	符号	描述	复位值	访问类型
31:0	HRB	主机接收缓冲区地址指针 复位时清除。指针在操作过程中由 DMA 更新。	0	RO

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.7 功能说明

26.7.1 电源管理块

此节描述 MAC 支持的电源管理 (PMT) 机制。PMT 支持接收网络（远程）唤醒帧和魔法包帧。PMT 不执行时钟门控功能，但是为被 MAC 接收的唤醒帧和魔法包生成中断。PMT 块位于在 MAC 的接收器路径上，且被远程唤醒帧使能和魔法包使能而使能。这些使能位于 PMT 控制和状态寄存器中，并由应用程序进行编程。

在 PMT 中使能掉电模式后，所有接收帧都被内核丢弃且不会被转发到应用程序中。仅当接收到魔法包或远程唤醒帧，且相应检测使能时，内核才会退出掉电模式。

26.7.1.1 远程唤醒帧寄存器

寄存器 wkupfmfilter_reg [地址 (0x028)] 可加载唤醒帧滤波寄存器。若要在唤醒帧滤波寄存器中加载值，则必须对整个寄存器 (WKUPFMFILTER_REG) 进行写操作。通过在地址 (0x028) 中分别为 WKUPFMFILTER_REG0、WKUPFMFILTER_REG1、...WKUPFMFILTER_REG7 按序加载 8 个寄存器的值来加载 WKUPFMFILTER_REG 寄存器。以相同的方式对 WKUPFMFILTER_REG 进行读操作。

注：当 CPU 访问通道 3（或大端的通道 0）时，访问相应 WKUPFMFILTER_REG 的内部计数器递增。在字节或半字模式下访问这些寄存器时，应记住这一点。

WKUPFMFILTER0	滤波器 0 字节掩码							
WKUPFMFILTER1	滤波器 1 字节掩码							
WKUPFMFILTER2	滤波器 2 字节掩码							
WKUPFMFILTER3	滤波器 3 字节掩码							
WKUPFMFILTER4	RSVD	滤波器 3 命令	RSVD	滤波器 2 命令	RSVD	滤波器 1 命令	RSVD	滤波器 0 命令
WKUPFMFILTER5	滤波器 3 偏移		滤波器 2 偏移		滤波器 1 偏移		滤波器 0 偏移	
WKUPFMFILTER6	滤波器 1 CRC - 16				滤波器 0 CRC - 16			
WKUPFMFILTER7	滤波器 3 CRC - 16				滤波器 2 CRC - 16			

图 62. 唤醒帧滤波寄存器

滤波器 i 字节掩码

该寄存器可定义帧的哪些字节要由滤波器 i（0、1、2 和 3）来进行检查，以确定该帧是否为唤醒帧。MSB（第 31 位）必须为零。位 j [30:0] 为字节掩码。如果对字节掩码的位 j（字节数）进行设置，则 CRC 模块会处理滤波器 i 的偏移量 + 输入帧的 j；否则滤波器 i 的偏移量 + j 将被忽略。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

滤波器 i 命令

该 4 位命令可控制滤波器 i 的操作。位 3 规定了地址类型，定义模式目标地址的类型。当对该位进行设置时，此模式只适用于多播帧；当该位复位时，此模式只适用于单播帧。位 2 和位 1 为保留位。位 0 为滤波器 i 使能；如果位 0 未设置，则滤波器 i 被禁用。

滤波器 i 偏移量

此寄存器定义偏移 （帧内），滤波器 i 从此偏移开始检查帧。此 8 位模式偏移是滤波器 i 要检查的首字节的偏移。允许的最小值为 12，该值指帧的第 13 个字节。偏移值 0 指帧的首字节。

滤波器 i CRC-16

该寄存器包含从该模式计算出的CRC_16的值，以及编程到唤醒滤波寄存器块的字节掩码。

26.7.1.2 远程唤醒探测

如果 MAC 处于睡眠模式且在 PMT 控制与状态寄存器 (0x002C) 中使能了远程唤醒位，则在接收远程唤醒帧后恢复正常操作。应用程序通过按序写入该地址 (0x0028) 对全部 8 个唤醒滤波寄存器进行写操作。应用程序通过向 PMT 控制与状态寄存器的位 2 写入 1 来使能远程唤醒。

PMT 支持 4 个允许支持接收不同帧结构的可编程滤波器。如果输入帧通过了滤波器命令的地址滤波，且如果滤波器 CRC-16 与输入检查模式匹配，则接收唤醒帧。

Filter_offset （最小值 12，指的是帧的第 13 个字节）可确定将要检查的帧的偏移量。滤波器字节掩码可确定必须进行检查的帧的字节。第 31 个字节掩码位必须设置为 0。

远程唤醒 CRC 块决定与过滤器 CRC-16 比较的 CRC 值。只检查唤醒帧的长度错误、FCS 错误、dribble 位错误、MII 错误和碰撞，也以此确保它不是超短帧。即使唤醒帧的长度超过 512 字节，但如果帧具备有效的 CRC 值，那么它也被认为是有效的。在 PMT 控制和状态寄存器中，为每个接收到的远程唤醒帧更新唤醒帧检测。应用程序的 PMT 中断可启动对 PMT 控制和状态寄存器的读操作，以确定唤醒帧的接收。

26.7.1.3 魔法包检测

魔法包的帧是以使用 “高级微型设备魔法包”技术使网络上的睡眠设备通电的方法为基础的。 MAC 接收特定的信息数据包，称为魔法包，寻址到网络上的节点。

只会对提交到设备或广播地址的魔法包进行检查，以确定他们是否满足唤醒要求。通过地址过滤（单播或广播）的魔法包将被检查以确定它们是否满足远程 LAN 上唤醒数据格式，该数据格式为 6 字节，全部为 1，后跟出现 16 次的 MAC 地址。

应用程序通过向 PMT 控制和状态寄存器的位 1 写 1 使能魔法包唤醒。 PMT 模块持续监控提交到特定魔法包模式节点的每一帧。接收到的每一帧都会检查接着目标和源地址字段的 0xFFFF FFFF FFFF 模式。然后 PMT 模块检查帧以确定 MAC 地址是否有 16 次没有任何中止和中断的重复。如果 16 个重复的地址产生间隔，将会在输入帧中再次扫描 0xFFFF FFFF FFFF模式。这16个重复可存在于帧内的任何位置，但必须位于同步流(0xFFFF FFFF FFFF) 之后。只要检测到 MAC 地址的 16 次重复，设备也将接受多播帧。

如果某节点的 MAC 地址为 0x0011 2233 4455，那么 MAC 扫描数据序列：

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
...CRC

在 PMT 控制和状态寄存器中，为接收到的魔法包更新魔法包检测。应用程序的 PMT 中断可启动对 PMT CSR 的读操作，以确定是否已接收到魔法包帧。

26.7.1.4 掉电期间系统注意事项

掉电模式使能后， MAC 既不门控也不中止时钟。时钟门控节能必须由应用程序在内核外部进行。在掉电模式期间，接收数据路径必须用 ENET_RX_CLK 进行时钟门控，因为它涉及到魔法包 /LAN 上唤醒帧检测。但是，在掉电模式期间，可断开发送路径和应用程序路径的时钟。

PMT 中断在接收到有效的唤醒帧时被断言。该信号在接收时钟域中生成

推荐的掉电和唤醒序列如下所示。

1. 禁用发送 “DMA” 并等待任何先前的帧发送完成。接收到发送中断（参见 DMA_STAT 寄存器位 NIS ；表 559）时，可检测到这些发送。
2. 通过将 MAC 配置寄存器中的相应位清零来禁用 MAC 发送器和 MAC 接收器。
3. 等待直到接收 “DMA” 清空来自 Rx FIFO 的所有帧 （可能需要软件定时器）。
4. 通过适当配置 PMT 寄存器来使能掉电模式。
5. 使能 MAC 接收器，并进入掉电模式。
6. 开启应用程序并将时钟输入发送到内核（和系统中的其他相关时钟），以降低功耗，进入睡眠模式。
7. 接收到有效的唤醒帧时， MAC PMT 中断信号并退出掉电模式。
8. 当接收到中断时，系统必须使能应用程序，并将时钟输入发送到内核。
9. 对 PMT 状态寄存器进行读操作可清零中断，然后使能系统中的其他模块并恢复正常操作。

注：

26.7.2 DMA 仲裁程序功能

如果已使能某通道的发送 (Tx) DMA 和接收 (Rx) DMA，则可以指定当通道获得总线控制时哪个 DMA 将获得总线。可以使用 DMA 总线模式寄存器的位 27 （TXPR：发送优先级）设置对应 Tx DMA 和 Rx DMA 之间的优先级。对于轮询仲裁，可使用总线模式寄存器的位 [15:14] （PR：优先级比率）来指定 Tx DMA 和 Rx DMA 之间的加权优先级。表 568 提供了 Tx DMA 和 Rx DMA 之间优先级方案的信息。

表 568. 发送和接收 DMA 的优先级方案

位 27	位 15	位 14	位 1	优先级方案
0	x	x	x	Rx 优先级始终高于 Tx
0	0	0	0	Tx 和 Rx 优先级相同。出现同步请求时，Rx 首先获得访问权限。
0	0	1	0	比率 2:1 下， Rx 优先级高于 Tx。
0	1	0	0	比率 3:1 下， Rx 优先级高于 Tx。
0	1	1	0	比率 4:1 下， Rx 优先级高于 Tx。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 568. 发送和接收 DMA 的优先级方案

位 27	位 15	位 14	位 1	优先级方案
1	x	x	1	Tx 优先级始终高于 Rx。
1	0	0	0	Tx 和 Rx 优先级相同。出现同步请求时， Tx 首先获得访问权限。
1	0	1	0	比率 2:1 下， Tx 优先级高于 Rx。
1	1	0	0	比率 3:1 下， Tx 优先级高于 Rx。
1	1	1	0	比率 4:1 下， Tx 优先级高于 Rx。

26.7.3 IPC 接收校验和卸载引擎

在此模式下，为了数据的完整性，会检测和处理已接收以太网帧中的 IPv4 和 IPv6 帧。可通过设置 MAC 配置寄存器中的位 10 (IPC) 使能此模块（第 26.6.1 节）。MAC 接收器通过分别检查已接收以太网帧的值 0x0800 或 0x86DD 来识别 IPv4 或 IPv6 帧。类型字段。此识别方法同样适用于 VLAN 标记帧。

接收校验和卸载引擎计算 IPv4 头文件校验和，并检查它们是否匹配已接收的 IPv4 头文件校验和。此操作的结果（通过或失败）被送给 RFC 模块以插入接收状态字。如果指示的有效载荷类型 （以太网类型字段）与 IP 头文件版本之间有任何不匹配，或者当接收的帧没有 IPv4 header.s 长度字段中指定的足够字节 （或者 IPv4 或 IPv6 头文件中可用字节数少于 20），则 “IP 头文件错误位被设置”。

此引擎也识别接收 IP 数据报（IPv4 或 IPv6）中的 TCP、UDP 和 ICMP 有效载荷，并按照 TCP、 UDP 和 ICMP 规范的定义，正确计算此类有效载荷的校验和。此引擎包括用于校验和计算的 TCP/UDP/ICMPv6 伪头文件字节，并检查已接收的校验和字段是否与计算值匹配。此操作的结果作为接收状态字中的 “有效载荷校验和错误” 位给出。如果 TCP、 UDP 和 ICMP 有效载荷的长度与 IP 头文件给出的预期有效载荷长度不匹配，此状态位也被设置。

此引擎绕过分段 IP 数据报、带安全功能的 IP 数据报和 IPv6 路由选择头文件的有效载荷，以及 TCP、 UDP 或 ICMP 外的其它有效载荷。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.8 IEEE 1588-2002 时间戳

IEEE 1588-2002 标准定义了精密时间协议 (PTP)，该协议使测量和控制系统（运用网络通信、本地计算以及分布式对象等技术进行实施）的时钟能够精密同步。 PTP 适用于通过支持多播消息的局域网 （包括但不限于以太网）进行通信的系统。该协议能使时钟的固有精度、分辨率和稳定性各不相同的异构系统进行同步。该协议以最小网络和本地时钟计算资源在亚微秒范围内支持全系统同步精度。

PTP 通过 UDP/IP 进行传输。系统或网络分为主机节点和从机节点，用于分发定时 / 时钟信息。图 63 显示了 PTP 通过交换 PTP 消息将从机节点同步到主机节点所使用的过程。

图 63. 网络时间同步

如图 63 所示， PTP 使用下列过程：

1. 主机将 PTP 同步消息广播至它的所有节点。同步消息包含主机的基准时间信息。该消息离开主机系统的时间为 t1。必须在 GMII/MII 为以太网端口捕获该时间。
2. 从机使用其时间基准接收同步消息并捕获精确时间 t2。
3. 主机发送 Follow_up 消息到从机，其中包含了供以后使用的 t1 信息。
4. 从机发送 Delay_Req 消息到主机，指出该帧离开 GMII/MII 的精确时间 t3。
5. 主机接收消息，捕获其进入系统的精确时间 t4。
6. 主机在 Delay_Resp 消息中发送 t4 信息到从机。
7. 从机使用 t1、 t2、 t3 和 t4 四个值将其本地时间基准同步到主机的时间基准。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

大多数 PTP 在 UDP 层上方的软件中完成实施。但是，需要硬件支持来捕获特定 PTP 数据包进入或离开GMII/MII处以太网端口的精确时间。必须捕获此定时信息并将其返回给软件，以便以高精度正确实施 PTP。

26.8.1 基准时间源

要获得时间的快照，MAC 需要 IEEE 1588 规范中定义的 64 位格式基准时间。以太网 MAC 使用内部定时来提供基准时间源，也就是通过基准时钟输入在内部生成基准时间 （也称为系统时间）并捕获时间戳。系统时间的生成、更新和修改如第 26.8.2 节中所述。

26.8.2 系统时间寄存器模块

64 位时间在该模块中保持并通过输入基准时钟进行更新。
该时间是为 GMII 处发送或接收的以太网帧生成快照 （时间戳）的源。

系统时间计数器可通过粗调校正方法来初始化或校正。在该方法中，初始值或偏移值写入时间戳更新寄存器（表 538）。对于初始化，通过时间戳更新寄存器中的值对系统时间计数器进行写操作；对于系统时间校正，则会向系统时间加入或从中减去偏移值。在微调校正方法中，与主机时钟有关的从机时钟频率漂移量（如 IEEE 1588 中定义）会在一段时间内 （而非像粗调校正方法中，在一个时钟内）得到校正。这样就保持了线性时间，并且不会使 PTP 同步消息间隔之间的基准时间发生显著变化（或强烈抖动）。在该方法中，累加器会计算加数寄存器内容的总和，如图 4-2 所示。累加器生成的算术进位用作使系统时间计数器递增的脉冲。累加器和加数为 32 位寄存器。此处的累加器相当于一个高精度倍频器或分频器。

该算法如图 64 中所示：

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 64. 使用微调方式的系统更新

系统时间更新逻辑需要 50-MHz 的时钟频率来达到 20-ns 的精度。分频是基准时钟频率与所需时钟频率之比。因此，如果基准时钟为 66 MHz，则该比率的计算方法为：66 MHz / 50 MHz = 1.32。因此，要在寄存器中设置的默认加数值为 2^{32} / 1.32， 0xC1F07C1F。

如果基准时钟漂移量降低到 65 MHz，则比率为 65 / 50 或 1.3，加数寄存器中要设置的值为 2^{32} / 1.30 或 0xC4EC4EC4。如果时钟漂移量升高到 67 MHz，则必须将加数寄存器设置为 0xBF0B7672。当时钟漂移量为零时，必须对 0xC1F07C1F (2^{32} / 1.32) 的默认加数值进行编程。

在图 64 中，用于累加亚秒级寄存器的常数值为 43（十进制），系统时间达到的精度为 20 ns（也就是说，它按 20 ns 的步幅进行递增）。

软件必须基于同步消息计算频率漂移量，并相应地更新加数寄存器。

最初，按照加数寄存器中的 FreqCompensationValue0 来设置从机时钟。该值如下所示：

FreqCompensationValue0 = 232 / FreqDivisionRatio

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

如果最初假设 MasterToSlaveDelay 对于连续同步消息是相同的，则必须采用下文所述算法。在几个同步周期后，发生频率锁定。随后，从机时钟可确定精确的 MasterToSlaveDelay 值并使用新的值与主机重新进行同步。

该算法如下所示：

- 在 $MasterSyncTime_n$ 这一时，主机会发送同步消息给从机时钟。从机的本地时钟为 $SlaveClockTime_n$ 时，从机接收该消息并计算 $MasterClockTime_n$，方法为：
  $MasterClockTime_n = MasterSyncTime_n + MasterToSlaveDelay_n$
- 当前同步周期的主机时钟计数 $MasterClockCount_n$ 的给定方法为：
  $MasterClockCount_n = MasterClockTime_n .MasterClockTime_{n-1}$ （假设 MasterToSlaveDelay 对于同步周期 n 和 n - 1 是相同的）
- 当前同步周期的从机时钟计数 $SlaveClockCount_n$ 的给定方法为：
  $SlaveClockCount_n = SlaveClockTimen_n - SlaveClockTime_{n-1}$
- 当前同步周期的主机和从机时钟计数之间的差值 $ClockDiffCount_n$ 的给定方法为：
  $ClockDiffCount_n = MasterClockCount_n - SlaveClockCount_n$
- 从机时钟的频率缩放系数 $FreqScaleFactor_n$ 的给定方法为：
  $FreqScaleFactor_n = (MasterClockCount_n + ClockDiffCount_n) / SlaveClockCount_n$
- 加数寄存器的频率补偿值 $FreqCompensationValue_n$ 的给定方法为：
  $FreqCompensationValue_n = FreqScaleFactor_n * FreqCompensationValue_{n-1}$

从理论上来说，该算法在一个同步周期实现了锁定；但是，由于变化的网络传播延迟以及操作条件，可能需要几个周期才能实现。

该算法会自校正：如果出于某种原因从机时钟最初被设置为来自主机的某个不正确的值，那么算法会花费更多同步周期来进行校正。

26.8.3 发送路径功能

帧的帧首定界符 (SFD) 在 GMII/MII 上发送时，MAC 会捕获时间戳。需捕获时间戳的帧逐个可控。也就是说，可对每个发送帧进行标记来表示是否应当为该帧捕获时间戳。

MAC 不通过处理已发送帧来识别 PTP 帧。需指定要捕获时间戳的帧。

使用发送描述符中的控制位 （参见第 26.10.3.1 节）。 MAC 将时间戳返回给对应发送描述符内的软件，从而自动将时间戳与特定的 PTP 帧连接起来。64 位时间戳信息写入至 TDES6 和 TDES7 字段。 TDES7 字段保持时间戳的 32 位最低有效位。

26.8.4 接收路径功能

MAC 捕获 GMII/MII 上所有已经接收帧的时间戳。在默认模式下，也就是在未选择高级时间戳功能时， MAC 不通过处理已接收帧来识别 PTP 帧。

除 EOF 数据外， MAC 还给出 MAC 接收接口 (MRI) 上的时间戳及其对应状态。在传输完 EOF 数据后，MTL 提供数据总线上的时间戳。MTL 发送单独的信号来验证时间戳并指示时间戳是否可用。一旦传输了时间戳，则只要在数据总线上有状态数据，状态有效信号就会被断言。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

DMA 将时间戳返回至对应接收描述符中的软件。64 位时间戳信息写回至 RDES6 和 RDES7 字段。RDES2 保持时间戳的 32 位最低有效位，第 26.10.3.2 节中所述除外。时间戳只写入最后描述符状态字段已经设为 1 的接收描述符 （EOF 标记）。时间戳不可用时 （比如，由于 RxFIFO 溢出），则全 1 模式写入描述符（RDES6 和 RDES7），表示时间戳不正确。如果软件使用控制寄存器位来禁用时间戳，则 DMA 不会更改 RDES6 或 RDES7。

26.8.5 时间戳误差容限

根据 IEEE 1588 规范，必须在 GMII/MII 接口的已发送帧和已接收帧的 SFD 处捕获时间戳。由于基准时间源 （PTP 时钟）不同于 GMII/MII 时钟，因此产生了较小误差容限 （在异步时钟域之间传输信息的缘故）。

在发送路径中，已捕获和已报告的时间戳具有两个 PTP 时钟的最大误差容限。这就表示，在 GMII 上发送 SFD 后，已捕获的时间戳具有两个时钟内所给定的基准时间源值。

类似地，在接收路径中，误差容限为三个 GMII/MII 时钟加上最多两个 PTP 时钟。如果假设在 SFD 数据到达 GMAC 的 GMII/MII 接口前，系统中存在三个 GMII/MII 时钟的常数延迟，该误差容限可忽略。

26.8.6 基准时间时钟的频率范围

时间戳信息在异步时钟域之间传输 （从 MAC 时钟域到应用时钟域）。因此，在两个连续的时间戳捕获之间需要最小延迟。该延迟为 GMII/MII 的四个时钟周期和 PTP 时钟的三个时钟周期。如果时间戳捕获之间的延迟小于该延迟，则 MAC 不会生成第二帧的时间戳快照。

最大 PTP 时钟频率受基准时间最大分辨率 （1 ns 导致 1 GHz）以及 PTP 时钟上运行逻辑可实现的定时约束的限制。此外，基准时间源的分辨率或粒度决定同步的精度。因此，具有较高 PTP 时钟频率的系统，其性能更佳。

最小 PTP 时钟频率取决于两个连续的 SFD 字节之间所需的时间。由于 GMII/MII 时钟频率由 IEEE 规范确定，因此正确运行所需的最小 PTP 时钟频率取决于 MAC 的运行模式以及运行速度，如表 4-1 所示。

表 569. 最小 PTP 时钟频率周期

模式	两个 SFD 的最小间隙。	最小 PTP 频率
100-Mbps 全双工操作	168 个 MII 时钟 （64 字节帧的 128 个时钟 + 最小 IFG 的 24 个时钟 + 前同步码的 16 个时钟）	(3 * PTP) + (4 * MII) ?168 * MII ，即： ~0.5 MHz（(168 - 4) * 40 ns ÷ 3 = 2,180 周期）

26.9 IEEE 1588-2008 高级时间戳

除基本时间戳功能 （参见第 26.8 节）之外，以太网控制器支持 IEEE 1588-2008 标准中定义的下列高级时间戳功能：

- 支持 IEEE 1588-2008 （版本 2）时间戳格式。
- 提供选项以生成所有帧或仅 PTP 型帧的快照。
- 提供选项以仅生成事件消息的快照。
- 提供选项以生成基于时钟类型 （普通、边界、端对端和点对点）的快照。
- 提供选项以选择将作为普通时钟和边界时钟主机或从机的节点。
- 识别 PTP 消息类型、版本以及通过以太网直接发送的帧内 PTP 有效负载，并发送状态。
- 提供选项以采用数字或二进制格式测量亚秒时间。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.9.1 点对点 PTP 透明时钟 (P2P TC) 消息支持

IEEE 1588-2008 版本不仅支持 SYNC、延迟请求、跟进以及延迟响应消息，还支持点对点 PTP (Pdelay) 消息。图 65 显示的方法用于计算支持点对点路径校正的时钟的传播延迟。

图 65. 支持点对点路径校正的时钟的传播延迟计算

如图 65 中所示，传播延迟的计算方法如下：

1. 端口 1 发出 Pdelay_Req 消息并生成 Pdelay_Req 消息的时间戳 t1。
2. 端口 2 接收 Pdelay_Req 消息并生成该消息的时间戳 t2。
3. 端口 2 返回 Pdelay_Resp 消息并生成该消息的时间戳 t3。

   在收到 Pdelay_Req 消息后，端口 2 尽可能快地返回 Pdelay_Resp 消息，以在最大程度上减少因两个端口间任何频率偏移而引起的误差。端口 2 返回下列任意一项：

   - Pdelay_Resp 消息中时间戳 t2 和 t3 之间的差值。
   - Pdelay_Resp_Follow_Up 消息中时间戳 t2 和 t3 之间的差值。
   - Pdelay_Resp 消息和 Pdelay_Resp_Follow_Up 消息中各自的时间戳 t2 和 t3。
4. 端口 1 一接收到 Pdelay_Resp 消息就生成时间戳 t4。
5. 端口 1 使用所有四个时间戳来计算平均链路延迟。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.9.2 时钟类型

以太网控制器支持 IEEE 1588-2008 标准中定义的下列时钟类型：

- 普通时钟
- 边界时钟
- 端对端透明时钟
- 点对点透明时钟

26.9.2.1 普通时钟

域中的普通时钟支持协议的单个副本。普通时钟具有单一的 PTP 状态和单一物理端口。在典型的工业自动化应用中，普通时钟与应用设备 （如传感器或执行机构）相关联。在电信应用中，普通时钟可与定时分界设备相关联。

普通时钟可以是主时钟，也可以是从机时钟。普通时钟支持下列功能：

- 发送和接收 PTP 消息。可按表 538 中所述，控制时间戳快照。
- 保持数据集，如时间戳值。

表 570. 普通时钟：快照的 PTP 消息

主机	从机
Delay_Req	SYNC

对于普通时钟，则可生成下列任一 PTP 消息类型的快照：版本 1 或版本 2。两种 PTP 消息类型的快照都可生成。可通过设置控制位 (TSVER2ENA) 并选择表 538 中的快照模式来生成快照。

26.9.2.2 边界时钟

通常，边界时钟有几个物理端口可与网络通信。与同步、主机 - 从机体系以及信令有关的消息在边界时钟的协议引擎内终止，并且不转发。由内核给定的 PTP 消息类型状态（参见第 26.8.4 节）有助于识别消息类型并采取适当的操作。

除下列功能外，边界时钟与普通时钟相似：

- 边界时钟的所有端口共用时钟数据集。
- 边界时钟的所有端口共用本地时钟。

因此，普通时钟的功能也适用于边界时钟。

26.9.2.3 端对端透明时钟

端对端透明时钟支持从机时钟和主机时钟间的端对端延迟测量机制。端对端透明时钟转发类似普通桥、路由器或中继器的所有消息。 PTP 数据包的驻留时间是 PTP 数据包从 “入口”端口到 “出口”端口所需的时间。

在发送前，端对端透明时钟内 SYNC 数据包的驻留时间在相关 Follow_Up PTP 数据包的校正字段中更新。类似地，在发送前，端对端透明时钟内 Delay_Req 数据包的驻留时间在相关 Delay_Resp PTP 数据包的校正字段中更新。因此，只需要对表 571 中提及的消息在“入口”端口和 “出口”端口处都生成快照。

将快照选择位 (SNAPTYPSEL) 设为表 538 中的 10 可生成快照。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 571. 端对端透明时钟：快照的 PTP 消息

PTP 消息
SYNC
Delay_Req

26.9.2.4 PTP 处理和控制

表 4-5 显示了 PTP 消息常用的消息标头。该格式取自 IEEE 标准 1588-2008 （IEEE 标准 1588-2002 的修订版本）。

表 572. IEEE 1588-2008 中定义的消息格式

位									
7	6	5	4	3	2	1	0	八位字节	偏移
transportSpecific				messageType				1	0
保留				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
保留								1	5
flagField								2	6
correctionField								8	8
保留								4	16
sourcePortIdentity								10	20
sequenceId								2	30
controlField ([1])								1	32
logMessageInterva								1	33

[1] 字段在版本 1 中使用。在版本 2 中， messageType 字段用于检测不同的消息类型。

可使用以太网有效负载中的某些字段来检测 PTP 数据包类型并控制要生成的快照。对于下列 PTP 帧，这些字段是不同的：

- IPv4 上的 PTP 帧
- IPv6 上的 PTP 帧
- 以太网上的 PTP 帧

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.9.2.4.1 IPv4 上的 PTP 帧

表 4-6 提供了经过匹配后的字段的相关信息来控制通过 UDP 和 IPv4 发送的 PTP 数据包快照（针对 IEEE 1588 版本 1 和版本 2）。带标签帧的八位字节位置的偏移量为 4。这基于 IEEE 1588-2008 标准的附录 D 以及表 571 中定义的消息格式。

表 573. 控制和状态所需的 IPv4-UDP PTP 帧字段

匹配的字段	八位字节位置	匹配值	描述
MAC 帧类型	12, 13	0x0800	IPv4 数据报
IP 版本和首部长度	14	0x45	IP 版本为 IPv4
第 4 层协议	23	0x11	UDP
IP 多播地址（IEEE 1588 版本 1）	30, 31, 32, 33	0 x E 0 、 0 x 0 0 、0x01、0x81（0x82 或 0x83 或 0x84）	允许的多播 IPv4 地址。 224.0.1.129 224.0.1.130 224.0.1.131 224.0.1.132
IP 多播地址（IEEE 1588 版本 2）	30, 31, 32, 33	0xE0、 0x00、0x01、 0x81 （十六进制） 0xE0、0x00、 0x00、0x6B （十六进制）	PTP- 主多播地址：224.0.1.129 PTP-Pdelay 多播地址：224.0.0.107
UDP 目标端口	36, 37	0x013F、 0x0140	0x013F – PTP 事件消息 ([1]) 0x0140 – PTP 通用消息
PTP 控制字段（IEEE 版本 1）	74	0x00/0x01/0x02/0x03/0x04	0x00 – SYNC, 0x01 – Delay_Req 0x02 – Follow_Up 0x03 – Delay_Resp 0x04 – Management
PTP 消息类字段（IEEE 版本 2）	42 （半字节）	0x0/0x1/0x2/0x3/0x8/0x9/0xB/0xC/0xD	0x0 – SYNC 0x1 – Delay_Req 0x2 – Pdelay_Req 0x3 – Pdelay_Resp 0x8 – Follow_Up 0x9 – Delay_Resp 0xA – Pdelay_Resp_Follow_Up 0xB – 通告 0xC – 信令 0xD – 管理
PTP 版本	43 （半字节）	0x1 或 0x2	0x1 – 支持 PTP 版本 1 0x2 – 支持 PTP 版本 2

[1] PTP 事件消息为 SYNC、 Delay_Req （IEEE 1588 版本 1 和版本 2）或 Pdelay_Req、Pdelay_Resp （仅 IEEE 1588 版本 2）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.9.2.4.2 IPv6 上的 PTP 帧

表 574 提供了经过匹配后的字段的相关信息来控制通过 UDP 和 IPv6 发送的 PTP 数据包快照（针对 IEEE 1588 版本 1 和版本 2）。带标签帧的八位字节位置的偏移量为 4。这基于 IEEE 1588-2008 标准的附录 D 以及表 571 中定义的消息格式。

表 574. 控制和状态所需的 IPv6-UDP PTP 帧字段

匹配的字段	八位字节位置	匹配值	描述
MAC 帧类型	12, 13	0x86DD	IP 数据报
IP 版本	14 （位 [7:4]）	0x6	IP 版本为 IPv6
第 4 层协议	20 ([1])	0x11	UDP
PTP 多播地址	38 – 53	FF0x:0:0:0:0:0:0:181 (Hex) FF02:0:0:0:0:0:0:6B (Hex)	PTP – 主多播地址： FF0x:0:0:0:0:0:0:0:0:181 (Hex) PTP – Pdelay 多播地址： FF02:0:0:0:0:0:0:0:0:6B （十六进制）
UDP 目标端口	56, 57 ([1])	0x013F、 0x140	0x013F – PTP 事件消息 0x0140 – PTP 通用消息
PTP 控制字段（IEEE 版本 1）	93 ([1])	0x00/0x01/0x02/0x03/0x04	0x00 – SYNC 0x01 – Delay_Req 0x02 – Follow_Up 0x03 – Delay_Resp 0x04 – 管理 （版本 1）
PTP 消息类字段（IEEE 版本 2）	74 ([1]) （半字节）	0x0/0x1/0x2/0x3/0x8/0x9/0xB/0xC/0xD	0x0 – SYNC 0x1 – Delay_Req 0x2 – Pdelay_Req 0x3 – Pdelay_Resp 0x8 – Follow_Up 0x9 – Delay_Resp 0xA – Pdelay_Resp_Follow_Up 0xB – 通告 0xC – 信令 0xD – 管理
PTP 版本	75 （半字节）	0x1 或 0x2	0x1 – 支持 PTP 版本 1 0x2 – 支持 PTP 版本 2

[1] 没有为 PTP 数据包定义扩展首部。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.9.2.4.3 以太网上的 PTP 帧

表 4-8 提供了经过匹配后的字段的相关信息来控制通过以太网 （针对 IEEE 1588 版本 1 和版本 2）发送的 PTP 数据包快照。带标签帧的八位字节位置的偏移量为 4。这基于 IEEE 1588-2008 标准的附录 D 以及表 572 中定义的消息格式。

表 575. 控制和状态所需的以太网 PTP 帧字段

匹配的字段	八位字节位置	匹配值	描述
MAC 目标多播地址 [1]	0-5	01-1B-19-00-00-00 01-80-C2-00-00-0E	所有 PTP 消息都可使用下列任何多播地址 [2]：01-1B-19-00-00-00 01-80-C2-00-00-0E[3]
MAC 帧类型	12, 13	0x88F7	PTP 以太网帧
PTP 控制字段（IEEE 版本 1）	45	0x00/0x01/0x02/ 0x03/0x04	0x00 – SYNC 0x01 – Delay_Req 0x02 – Follow_Up 0x03 – Delay_Resp 0x04 – 管理
PTP 消息类字段（IEEE 版本 2）	14 （半字节）	0x0/0x1/0x2/0x3/ 0x8/0x9/0xB /0xC/0xD	0x0 – SYNC 0x1 – Delay_Req 0x2 – Pdelay_Req 0x3 – Pdelay_Resp 0x8 – Follow_Up 0x9 – Delay_Resp 0xA – Pdelay_Resp_Follow_Up 0xB – 通告 0xC – 信令 0xD – 管理
PTP 版本	14 （半字节）	0x1 或 0x2	0x1 – 支持 PTP 版本 1 0x2 – 支持 PTP 版本 2

[1] 如果设置了时间戳控制寄存器的控制位 18（TSENMACADDR: 使能 PTP 帧滤波 MAC 地址），则会使用 MAC 地址 1 到 31 中编程的目标地址 (DA) 的地址匹配。

[2] IEEE 标准 1588-2008，附录 F

[3] 带对等体延迟多播地址 (01-80-C2-00-00- 0E) 的 PTP 版本 1 消息不被以太网控制器视为有效的 PTP 消息。

26.9.3 基准时间源

以太网控制器支持 IEEE 1588-2008 标准中所述的下列基准时间源：

- 48 位秒字段
- 每秒脉冲输出
- 带外部事件的辅助快照

26.9.3.1 48 位秒字段

以太网控制器支持 80 位时间戳。时间戳具有下列字段：

- UInteger48 secondsField

  秒字段是时间戳的整数部分 （以秒为单位）， 48 位宽。例如， 2.000000001 秒的表示方法为 secondsField = 0x0000_0000_0002。

- UInteger32 nanosecondsField

  纳秒字段是时间戳的小数部分（以纳秒为单位）。例如，2.000000001 纳秒的表示方法为 nanoSeconds = 0x0000_0001。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

纳秒字段支持下列两种模式：

- 数字翻转模式：数字翻转模式下，纳秒字段的最大值为0x3B9A_C9FF，也就是(10e9-1)纳秒。
- 二进制翻转模式：二进制翻转模式下，纳秒字段在值 0x7FFF_FFFF 后翻转并使秒字段递增。精度约为每位 0.466 ns。

使用表 538 中的位 9 (TSCTRLSSR) 可设置这些模式。

选择高级时间戳功能时，内核中保持的时间戳仍为 64 位宽。秒寄存器的高 16 位溢出 130 年内发生一次。可从 CSR 寄存器读取秒字段高 16 位的值。

26.9.3.2 固定每秒脉冲输出

以太网控制器支持用来表示 1 秒间隔 （默认情况下）的每秒脉冲 (PPS) 输出。设置表 550 中的位 [3:0] 可更改 PPS 输出的频率。

26.9.3.3 灵活的每秒脉冲输出

以太网控制器还具备灵活性，可对起始时间或停止时间、宽度以及 PPS 输出时生成的脉冲的间隔进行编程。

以太网控制器提供下列功能，其中包含灵活的 PPS 输出选项：

- 支持对起始时间或停止时间进行编程 （按照高级系统时间）。
- 支持对单脉冲的起始点和脉冲群的起始点和停止点进行编程（按照 64 位系统时间）。现有的目标时间寄存器 （寄存器 455 和 456）用于对起始时间和停止时间进行编程。
- 支持提前对停止时间进行编程，也就是说，可在实际起始时间过去之前对停止时间进行编程。
- 支持对 PPS 信号输出上升沿和对应的下降沿之间的宽度进行编程 （按照亚秒递增寄存器 （表 540）中编程的亚秒递增值的单位数）。可对 1 到 2^{32}-1 个亚秒递增值单位的脉冲的宽度进行编程。
- 支持对 PPS 信号上升沿之间的间隔进行编程 （按照亚秒递增值的单位数）。可对 1 到 2^{32}-1 个亚秒递增值单位的脉冲的间隔进行编程。
- 提供取消编程 PPS 起始或停止请求的选项。
- 如果正在编程的起始或停止时间已过，会指示错误。

注：下文各章节提到的 PTP 基准时钟是系统时间更新时的时钟。当时间戳控制寄存器的位 1 （TSCFUPDT：时间戳微调或粗略更新）设为 0 时，该时钟与 <tbd> 时钟类似。在微调模式下，这是系统时间更新时的时钟节拍 (<tbd>)。

26.9.3.4 PPS 起始和停止时间

最初可对目标时间寄存器 （表 546 和表 547）的起始时间进行编程。如果需要，可对起始时间或停止时间再次进行编程，不过仅能在较早编程值同步至 PTP 时钟域后才能进行。目标时间纳秒寄存器（表 547）的位 31 表示同步完成。这就允许提前对起始时间或停止时间进行编程，甚至在较早的停止时间或起始时间过去之前。

应当根据高级系统时间来进行起始时间或停止时间的编程，以确保 PPS 信号正确输出。如果应用程序对已过的起始时间或停止时间进行编程，则以太网控制器会设置表示编程错误的错误状态位。如果使能，以太网控制器会设置已 “达到目标时间”中断事件。只有在对应的起始时间或停止时间未过去的情况下，应用程序才可取消起始或停止请求。如果时间已过，则取消命令起不到任何作用。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.9.3.5 PPS 宽度和间隔

PPS 宽度和间隔根据系统时间的粒度进行编程，也就是亚秒递增值的单位数。例如，要获得宽度为 40、间隔为 100 ns 的 PPS 脉冲 （PTP 基准时钟为 50 MHz），应当分别将宽度和间隔编程为值 2 和值 5。使用较快的 PTP 基准时钟可获得更小粒度。

在发出命令启动 PPS 输出上的脉冲或脉冲群前，应当对 PPS 信号输出的间隔和宽度进行编程或更新。

26.9.3.6 带外部事件的辅助快照

以太网控制器支持生成带外部事件的辅助时间戳快照。有关详情，请参见第 26.9.3.6 节。

26.9.4 发送路径功能

高级时间戳的发送路径功能无变化。

如果使能高级时间戳功能，则描述符结构会发生变化。高级时间戳功能仅通过备用 （增强型）描述符格式支持。该描述符为 32 字节长（8 个双字），时间戳的快照写入描述符 TDES6 和 TDES7。有关描述符的详情，请参见第 26.10.3 节。

26.9.5 接收路径功能

选择高级时间戳功能时，MAC 会处理已接收帧以识别有效的 PTP 帧。使用时间戳控制寄存器的下列选项可控制将要发送至应用程序的时间快照 （第 26.6.16 节）：

- 选择高级时间戳功能时，MAC 会处理已接收帧以识别有效的 PTP 帧。使用寄存器 448（时间戳控制寄存器）的下列选项可控制将要发送至应用程序的时间快照。
- 选择高级时间戳功能时，MAC 会处理已接收帧以识别有效的 PTP 帧。使用寄存器 448（时间戳控制寄存器）的下列选项可控制将要发送至应用程序的时间快照。
- 使能通过以太网或 UDP-IP 以太网直接发送的 PTP 帧的快照。
- 使能 IPv4 或 IPv6 已接收帧的时间戳快照。
- 仅为事件消息（SYNC、DELAY_REQ、PDELAY_REQ 或 PDELAY_RESP）使能时间戳快照。
- 使能要成为主机或从机的节点并选择快照类型。这样就控制了生成快照的消息类型。

注：以太网控制器还支持 VLAN 帧上的 PTP 消息。

除 EOF 外，MAC 还提供 MRI 上的 64 位总线的时间戳。附加信号验证接收帧是否有时间戳。

在传输完 EOF 数据后，MTL 提供数据总线上的时间戳。附加信号验证时间戳。该信号被断言以验证时间戳的可用性。一旦传输了时间戳， MTL 就会发送接收状态到应用程序。在 32 位数据宽度模式下，MTL 会在读取了正常状态后，提供与数据总线上的时间戳和 IP 校验和卸载 (IPC) 有关的附加状态。仅在设置并验证了正常状态的 0 位时提供附加状态。在 64 位和 128 位数据宽度模式下， MTL 使用正常状态的 55:32 位提供附加状态。

DMA 将时间戳返回至对应发送和接收描述符内的软件。高级时间戳功能仅在采用 32 字节长备用 （增强型）描述符的情况下受支持。扩展状态 （包含时间戳消息状态和 IPC 状态）写入描述符 RDES4，时间戳快照写入描述符 RDES6 和 RDES7。有关描述符的详情，请参见第 26.10.3 节。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.9.6 辅助快照

通过辅助快照功能可存储基于外部事件的系统时间快照。该事件被视为旁带信号 <tbd> 的上升沿。该功能与系统时间是在内部生成还是作为输入给出无关（基于 <tbd> 总线）。

MAC 将这些快照存储在 4 深度 FIFO 中。只有时间戳的 64 位存储在 FIFO 中。如果有的话，可从系统时间 – 高字秒寄存器（表 548）读取秒的高 16 位。存储快照时，MAC 会通过中断向主机表明这一情况。快照值通过 FIFO 寄存器访问读取。如果 FIFO 变为满状态且生成快照的外部启动被断言，

则快照启动丢失状态 (ATSSTM) 在时间戳状态寄存器（表 549）中设置。这表示最新的辅助时间戳快照并未存储在 FIFO 中。 FIFO 处于满状态时，最新的快照不写入 FIFO。

26.10 DMA 控制器说明

DMA 有独立的发送和接收引擎及 CSR 空间。发送引擎将数据从系统存储器传输到设备端口 (MTL)，而接收引擎将数据从设备端口传输到系统存储器。控制器使用描述符可在主机 CPU 干预最小的情况下将数据从源地址有效地移动到目标地址。DMA 专门设计用于面向数据包的数据传输，例如以太网中的帧。根据情况（例如帧发送和接收传输完成情况）及其他正常 / 错误条件，控制器可编程为中断主机 CPU。

DMA 和主机驱动器通过两种数据结构进行通信：

- 控制和状态寄存器 (CSR)。参见第 26.6 节。
- 描述符列表和数据缓冲区。参见第 26.10.3 节。

DMA 将内核接收到的数据帧传输到主机存储器中的接收缓冲区，并发送来自主机存储器发送缓冲区的发送数据帧。位于主机存储器的描述符可用作这些缓冲区的指针。有两个描述符列表；一个用于接收，另一个用于发送。每个列表的基址写入 DMA 寄存器表 557 和表 558。描述符列表向前链接（隐性或显性）。最后一个描述符可能指回第一个条目，以创建环状结构。描述符的显式链接可以通过设置链接着接收和发送描述符（RDES1[24] 和 TDES1[24]）的第二个地址来实现。描述符列表位于主机物理内存地址空间。每个描述符最多可指向两个缓冲区。这使得可以使用两个缓冲区，物理寻址，而不是存储器中的连续缓冲区。

数据缓冲区位于主机物理内存空间，包括整个帧或帧的一部分，但不能超过一个帧。缓冲区只包含数据，缓冲区状态保存在描述符中。数据链接指的是跨越多个数据缓冲区的帧。但是，一个描述符不能跨越多个帧。 DMA 在检测到帧末端时会跳到下一帧缓冲区。数据链接可以使能或禁用。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 66. 描述符环状和链状结构

26.10.1 初始化

按照以下步骤初始化以太网控制器：

1. 写入 DMA 寄存器表 553 来设置主机总线访问参数。
2. 写入 DMA 寄存器表 561 来掩码不必要的中断原因。
3. 软件驱动程序建立发送和接收描述符列表。然后它写入 DMA 寄存器表 557 和 DMA 寄存器表 558，向 DMA 提供每个列表的起始地址。
4. 写入 MAC 寄存器表 523、表 525 和表 524 以获得所需的滤波选项。
5. 写入 MAC 寄存器表 522 来配置操作模式并使能发送操作（位 3：发送器使能）。PS 和 DM 位根据自动协商结果 （从 PHY 中读取）来设置。
6. 写入 DMA 寄存器表 560 来设置用位 13 和 1 以启动发送和接收。
7. 写入 MAC 寄存器表 522 来使能接收操作 （位 2：接收器使能）。

发送和接收引擎进入运行状态，并试图从各自的描述符列表获取描述符。然后，发送和接收引擎开始处理接收和发送操作。发送和接收过程相互独立，可以分别开始或停止。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.10.1.1 主机总线连发访问

如果已进行相关配置 （DMA 寄存器 0 的 FB 位）， DMA 会试图在 AHB 主机接口上执行固定长度的连发传输。最大连发长度由 PBL 字段 （DMA 寄存器 0[13:8]）表示和限制。始终以要读取的 16 字节的最大可能（受限于 PBL 或 16 x 8/ 总线宽度）连发量对接收和发送描述符进行存取。

仅当 “MTL 发送 FIFO”或直到帧尾端的字节数 （当它小于配置的连发长度时）中有足够空间容纳已配置连发时，“发送 DMA”才启动数据传输。 DMA 指出 AHB 主机接口要求的起始地址和传输数量。当AHB接口被配置为固定长度连发时，它使用INCR4/8/16和SINGLE事务的最佳组合来传输数据。否则 （没有固定长度的连发），它使用 INCR （未定义长度）和 SINGLE 事务来传输数据。

仅当 “MTL 接收 FIFO”中有足够数据来容纳已配置连发时，或者在接收 FIFO 中探测到帧的尾端 （当它小于配置的连发长度）时，“接收 DMA”才启动数据传输。 DMA 指出 AHB 主机接口要求的起始地址和传输数量。当 AHB 接口被配置为固定长度连发时，它使用 INCR4/8/16 和 SINGLE 事务的最佳组合来传输数据。如果在 AHB 接口上的固定连发结束前到达帧的尾端，那么为了完成固定连发，将执行虚拟传输。否则 （复位 DMA 寄存器表 553 的 FB 位），它使用 INCR （未定义长度）和 SINGLE 事务处理来传输数据。

当 AHB 接口被配置为地址对齐节拍时，两个 DMA 引擎都确保 AHB 启动的首个连发传输小于等于已配置 PBL 的大小。因此，所有后续节拍在对齐到已配置 PBL 的地址处开始。DMA 最多只能对齐大小达 16 （PBL > 16）的节拍的地址，因为 AHB 接口不支持大于 INCR16 的大小。

26.10.1.2 主机数据缓冲区对齐

发送和接收数据缓冲区没有任何起始地址对齐限制。例如，在配备 32 位存储器的系统中，缓冲区的起始地址可以对齐到任何 4 个字节。但是， DMA 通常使用对齐到不需要字节通道的虚拟数据总线宽度的地址启动传输。这一般发生在以太网帧传输开始或结束期间。

示例：缓冲区读操作

如果发送缓冲区地址为 0x00000FF2 （用于 32 位数据总线），且有 15 字节需要传输，则 DMA 将从地址 0x00000FF0 读取 5 个全字，但将数据传输到 MTL 发送 FIFO 时，将会丢弃或忽略额外字节（前两个字节）。同样地，最后一次传输的最后 3 个字节也将被忽略。DMA 一般都可确保将 32 位数据全部传输到 MTL 发送 FIFO，除非是帧末端。

示例：缓冲区写操作

如果接收缓冲区地址是 0x0000FF2 （对于 64 位数据总线）且需要传输接收帧的 16 个字节，那么 DMA 从地址 0x00000FF0 写入 3 个完整的字。但是第一个传输的前两个字节和第三个传输的最后 6 个字节有虚拟数据。

26.10.1.3 缓冲区大小计算

DMA 不会更新发送和接收描述符中的容量字段。 DMA 仅更新描述符的状态字段 （RDES 和 TDES）。驱动器必须要进行容量计算。

发送 “DMA”将确切的字节数 （由 TDES1 缓冲区大小指示）向 MAC 内核传输。如果描述符被标记为第一位 （已对 TDES1 的 FS 位进行设置），则 DMA 将缓冲区的第一次传输标记为帧的起始端。如果描述符被标记为最后一位（TDES1 的 LS 位），则 DMA 将该数据缓冲区的最后一次传输标记为 MTL 的帧末端。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

接收 DMA 将数据传输到缓冲区，直到缓冲区已满或从 MTL 接收到帧末端为止。如果描述符未被标记为最后一位 （RDES0 的 LS 位），则描述符的对应缓冲区已满；对该描述符的 FS 位进行设置时，缓冲区中的有效数据量由其减去数据缓冲区指针偏移量的缓冲区大小字段进行精确表示。当数据缓冲区指针对齐到数据总线宽度时，偏移量为零。如果描述符被标记为最后一位，则缓冲区可能未满（如 RDES1 中的缓冲区容量所示）。若要计算最终缓冲区的有效数据量，驱动器必须读取帧长度 （RDES0[29:16] 的 FL 位），并减去该帧中上一缓冲区的缓冲区容量之和。接收 DMA 通常使用新的描述符传输下一帧的起始端。

注：即使接收缓冲区的起始地址与系统总线的数据宽度不对齐，系统也应分配大小与系统总线宽度对齐的接收缓冲区。例如，如果系统分配一个从地址 0x1000 开始的 1,024 字节的 (1 KB) 的接收缓冲区，则软件可对接收描述符中的缓冲区起始地址进行编程，以具有 0x1002 的偏移量。接收 DMA 将帧写到前两个位置 （0x1000 和 0x1001）中含有虚拟数据的缓冲区。实际的帧从位置 0x1002 开始写入。因此，由于起始地址的偏移，虽然缓冲区大小被设置为 1024 个字节，此缓冲区中实际有用的空间为 1022 个字节。

26.10.1.4 DMA 仲裁程序

DMA 模块内的仲裁器执行 AHB 主接口发送和接收通道访问之间的仲裁。两种类型的仲裁都有可能：循环优先和固定优先。

选定循环仲裁时（复位寄存器表 553（总线模式寄存器）的 DA 位），如果发送和接收 DMA 同时请求访问，则仲裁程序按照 DMA 寄存器表 553 PR 位设置的比率分配数据总线。默认情况下，当对 DA 位进行设置时，接收 DMA 通常优先于发送 DMA 进行数据访问。如果还设置了 TXPR 位（DMA 寄存器表 553 的位 27），则发送 DMA 获得高于接收 DMA 的优先级。

26.10.2 传输

26.10.2.1 TxDMA 操作：默认 （非 OSF）模式

在默认模式下，发送 DMA 引擎的操作步骤如下：

1. 主机设置发送描述符 (TDES0-TDES3)，并在设置含以太网帧数据的相应数据缓冲区之后设置拥有位 (TDES0[31])。
2. 设置完 ST 位 （DMA 寄存器）后， DMA 进入运行状态。
3. 处于运行状态时， DMA 轮询 “发送描述符列表”以获得要求发送的帧。轮询开始后，将按照顺序描述符的环状顺序或链状顺序继续进行。如果 DMA 检测到标记为主机拥有的描述符，或发生错误情况，则将挂起发送且设置 “发送缓冲区不可用”（DMA 寄存器表 559）和 “正常中断汇总”（DMA 寄存器表 559）位。发送引擎进行步骤 9。
4. 如果获取的描述符标记为 DMA (TDES0[31] = 1) 所有，则 DMA 将对获取描述符的发送数据缓冲区地址进行解码。
5. DMA 从主机存储器中提取发送数据，并将数据传输到 MTL，以进行发送。
6. 如果以太网帧存储在多个描述符的数据缓冲区上，则 DMA 将关闭中间描述符并提取下一个描述符。第 3 步、第 4 步和第 5 步将重复进行，直到以太网帧末端数据传输到 MTL 为止。
7. 帧发送完成后，如果 IEEE 1588 时间戳为帧使能 （如发送状态所示），则从 MTL 获得的时间戳值将被写入包含帧末端缓冲区的发送描述符 （TDES2 和 TDES3）。然后，状态信息也将被写入该发送描述符 (TDES0)。因为拥有位在该步骤中清零，所以现在主机拥有该描述符。如果没有为该帧使能时间戳，则 DMA 不会更改 TDES2 和 TDES3 的内容。
8. 在最后描述符中设置了 “完成时中断” (TDES1[31]) 的帧传输完成后，会设置 “发送中断”（DMA 寄存器表 559）。然后 DMA 引擎返回步骤 3。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

9. 在暂停状态中，当 DMA 接收到发送轮询请求，且下溢中断状态位清零时，DMA 尝试重新获取描述符 （因此返回第 3 步）。

默认模式下的 TxDMA 传输流如图 67 所示。

图 67. 默认模式下的 TxDMA 操作

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.10.2.2 TxDMA 操作：OSF 模式

当处于运行状态，发送进程可在不关闭第一个状态描述符的情况下，同时获得两帧 （如果 DMA 操作模式寄存器的 OSF 位被设置，位 2）。当发送过程完成第一帧的传输之后，它将立即轮询第二帧的发送描述符列表。如果第二帧有效，则发送过程将在写第一帧的状态信息之前传输该帧。

在 OSF 模式下，运行状态发送 DMA 按照以下顺序进行操作：

1. DMA 按照 TxDMA （默认模式）的第 1 步到第 6 步中所描述的一样进行操作。
2. 在没有关闭前一帧的最后一个描述符的情况下， DMA 提取了下一个描述符。
3. 如果 DMA 拥有获取的描述符，则 DMA 将对该描述符中的发送缓冲区地址进行解码。如果 DMA 没有描述符，则 DMA 将进入暂停模式并转到第 7 步。
4. DMA 从主机存储器中提取发送帧，并将帧传输到 MTL，直到传输帧末端数据为止；如果该帧分散到多个描述符中，则关闭中间描述符。
5. DMA 等待前一帧的帧发送状态和时间戳。一旦状态可用，DMA 则将时间戳写到 TDES2 和 TDES3，如果获取了此类时间戳的话（如状态位表示）。然后， DMA 使用清零的拥有位将状态写到相应的 TDES0，从而关闭描述符。如果没有为前一帧使能时间戳，则 DMA 不会更改 TDES2 和 TDES3 的内容。
6. 如果已使能，则发送中断将被设置， DMA 将提取下一个描述符，然后转到第 3 步（状态正常时）。如果前一发送状态发生下溢错误，则 DMA 进入暂停模式 （第 7 步）。
7. 在挂起模式下，如果从 MTL 接收到待定状态和时间戳，则 DMA 将时间戳 （如果已为当前帧使能）写入 TDES2 和 TDES3，然后将状态写入相应的 TDES0。然后它设置相关中断并返回挂起状态。
8. 仅在接收到 “发送轮询”要求 （DMA 发送轮询要求寄存器）后， DMA 才可退出挂起模式并进入运行状态 （根据待定状态进入步骤 1 或步骤 2）。

注：由于 DMA 在关闭当前描述符前，提前获得下一描述符，描述符链应该有超过 2 个的不同描述符，才能正确合理运行。

基本流如图 68 所述。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 68. OSF 模式下的 TxDMA 操作

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.10.2.3 发送帧处理

发送 DMA 期望数据缓冲区包含完整的以太网帧，不包含前序编码、填充字节和 FCS 字段。DA、SA 和 Type/Len 字段包含有效数据。如果发送描述符要求 MAC 内核必须禁用 CRC 或 PAD 插入，则缓冲区必须具有完整的以太网帧 （不包括前序编码），包括 CRC 字节。

帧可以由数据链接，也可跨越多个缓冲区。帧必须由第一个描述符 (TDES1[29]) 和最后一个描述符 (TDES1[30]) 分别进行限定。

发送开始时，第一个描述符 (TDES1[29]) 必须已经设置。发生这种情况时，帧数据将从主机缓冲区传输到 MTL 发送 FIFO。同时，如果当前帧将最后一个描述符 (TDES1[30]) 清零，则发送过程试图获取下一个描述符。发送过程期望该描述符将 TDES1[29] 清零。如果 TDES1[30] 清零，则它表示中间缓冲区。如果 TDES1[30] 被设置，则它表示帧的最后一个缓冲区。

在发送帧的最后一个缓冲区之后，DMA将最终状态信息回写到在发送描述符1 (TDES1[30]) 中设置最后一段的描述符的发送描述符 0 (TDES0) 字。此时，如果 “完成时中断” (TDES1[31]) 已被设置，则 “发送中断”（DMA 状态寄存器，位 0）被设置，下一描述符被获取，然后重复过程。

实际的帧传输在 “MTL 发送 FIFO”达到可编程的发送阈值 （DMA 操作模式寄存器，位 [16:14]）或完整的帧被包含在 FIFO 中后开始。也有存储和转发模式的选项 （DMA 操作模式寄存器，位 [21]）。当 DMA 完成帧的传输时，描述符被释放（拥有位 TDES0[31] 清零）。

注：为了保证帧和下一帧的正确传输，必须为设置了最后描述符 (TDES1[30]) 的发送描述符指定非零缓冲区大小。

26.10.2.4 发送轮询暂停

发送轮询可通过以下任一条件暂停：

- DMA 检测到为主机所有的描述符 (TDES0[31]=0)。若要恢复，则驱动器必须将描述符所有权让与 DMA，并且发出轮询请求命令。
- 由于检测到下溢而发生发送错误时，帧发送中止。相应的发送描述符 0 (TDES0) 位被设置。

如果发生第二种情况，则会设置 “异常中断汇总”（DMA 状态寄存器表 559）和 “发送下溢”位（DMA 状态寄存器表 559），且信息写入发送描述符 0，造成挂起。如果 DMA 由于第一种情况而进入 SUSPEND 状态，则会设置“正常中断操作汇总”（DMA 状态寄存器表 559）和 “发送缓冲区不可用”（DMA 状态寄存器表 559）。

在这两种情况下，发送列表中的位置将被保留。保留的位置是接着 DMA 关闭的最后一个描述符的描述符位置。

在纠正暂停原因之后，驱动器必须明确地发出发送轮询请求命令。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.10.2.5 接收

接收 DMA 引擎的接收顺序如图 69 所示，按下列方式进行操作：

1. 主机建立接收描述符 (RDES0-RDES3) 并设置 Own 位 (RDES0[31])。
2. 一旦设置了 SR（DMA 操作模式寄存器表 560）位，则 DMA 会进入运行状态。当处于运行状态时， DMA 将轮询接收描述符列表，试图获取自由描述符。如果提取的描述符不是自由的 （为主机所有），则 DMA 进入暂停状态，并跳转到第 9 步。
3. DMA 对所获取描述符的接收数据缓冲区地址进行解码。
4. 输入帧被处理并置于所获取描述符的数据缓冲区。
5. 当缓冲区已满或帧传输完成时，接收引擎提取下一个描述符。
6. 如果当前帧传输完成，则 DMA 将转到第 7 步。如果 DMA 没有下一个提取的描述符，且帧传输没有完成（EOF 尚未传输），则 DMA 将在 RDES0 中设置描述符错误位（除非刷新禁用）。DMA 关闭当前描述符（清零拥有位），并通过在 RDES0 值中将最后一段 (LS) 位清零将其标记为中间描述符 （如果刷新未禁用，则将其标记为最后一个描述符），然后转到第 8 步。如果 DMA 确实拥有下一个描述符，但当前帧传输未完成，则 DMA 将当前描述符作为中间描述符关闭，并转到第 4 步。
7. 如果使能 IEEE 1588 时间戳记，则 DMA 会将时间戳 （如果可用）写入当前描述符 RDES2 和 RDES3。然后，它从 MTL 中获取接收帧状态，并将状态字写入当前描述符的 RDES0， Own 位被清除，最末段位被设置。
8. 接收引擎检查最后描述符的 Own 位。如果主机拥有描述符 （Own 位为 0），则会设置“接收缓冲区不可用”位 （DMA 状态寄存器表 559）且 DMA 接收引擎进入挂起状态（步骤 9）。如果 DMA 拥有描述符，则引擎返回步骤 4 并等待下一帧。
9. 在接收引擎进入挂起模式前，会刷新接收 FIFO 的部分帧 （可使用 DMA 操作模式寄存器表 560 的位 24 来控制刷新）。
10. 当发出接收轮询请求或 MTL 接收 FIFO 可提供下一帧的起始位置时，接收 DMA 退出暂停状态。引擎转到第 2 步，并重新提取下一个描述符。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 69. 接收 DMA 操作

DMA 不会对接受 MTL 的状态作出反应，直到它完成时间戳回写操作并准备将状态回写到描述符为止。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

如果软件已通过 CSR 使能时间戳，当有效地时间戳值对帧不可用时（例如，在时间戳可以写入接收 FIFO 以前，它已经是满的），DMA 将 RDES2 和 RDES3 全部写入 1。否则（也就是，时间戳没有使能），RDES2 和 RDES3 保持不变。

26.10.2.6 接收描述符获得

接收引擎总是试图获取额外描述符，期望输入帧。如果满足以下任一条件，就可以尝试获取描述符：

- 接收“起始 / 停止”位（DMA 操作模式寄存器表 560）已在被置于运行状态后被立即设置。
- 当前描述符的数据缓冲区在帧结束当前传输之前已满。
- 控制器已完成帧接收，但当前接收描述符尚未关闭。
- 由于主机所拥有的缓冲区 (RDES0[31] = 0)，所以接收过程被暂停，且接收到新帧。
- 接收轮询请求已经发出。

26.10.2.7 接收帧处理

仅当帧通过地址过滤器且帧的大小大于等于为 MTL 的接收 FIFO 配置的阈值字节时，或当完整的帧被写入“存储和转发”模式下的 FIFO 时，MAC 将接收到的帧传输到主机存储器。

如果帧未通过地址过滤，则会在 MAC 块本身将其丢弃（除非已设置 MAC 帧滤波寄存器中的“接收所有”位 31；表 523）。由于冲突和提前终止，可以从 MTL 接收 FIFO 中清除短于 64 字节的帧。

在接收到 64（可配置阈值）字节之后，MTL 模块请求 DMA 模块开始将帧数据传输到当前描述符指向的接收缓冲区。在 DMA 主机接口（AHB 或 MDC）准备好接收数据传输后（如果 DMA 不从主机获取发送数据），DMA 设置第一个描述符 (RDES0[9]) 以限制帧。由于数据缓冲区已满，或由于帧的最后一段被传输到接收缓冲区，当拥有 (RDES[31]) 位复位为 0 时，描述符被释放。如果帧包含在单个描述符中，则最后一个描述符 (RDES[8]) 和第一个描述符 (RDES[9]) 都将被设置。

DMA 获得下一描述符，设置“最后描述符”(RDES[8]) 位，并释放前一帧描述符中的 RDES0 状态位。然后，DMA 设置接收中断（寄存器 5[6]）。如果 DMA 没有遇到标记为主机所有的描述符，则重复相同步骤。如果该情况发生，则接收进程会设置“接收缓冲区不可用”（DMA 状态寄存器表 559），然后进入挂起模式。接收列表中的位置被保留。

26.10.2.8 接收进程挂起

如果新的接收帧在接收过程处于暂停状态时到达，则 DMA 重新提取主机存储器中的当前描述符。如果描述符现在为 DMA 所有，则接收过程重新进入运行状态，并开始帧接收操作。如果描述符仍为主机所有，则默认情况下，DMA 将在 MTL Rx FIFO 顶部丢弃当前帧，并增大遗漏帧计数器。如果 MTL Rx FIFO 中存储的帧不止一个，则该步骤将重复。

通过设置表 560 中的操作模式寄存器位 24 (DFF)，可以避免在 MTL Rx FIFO 顶部丢弃帧或对其进行刷新操作。在这种情况下，接收进程将设置“接收缓冲区不可用”状态，然后返回到挂起状态。

26.10.2.9 中断

中断作为各种事件的结果而生成。DMA 状态寄存器（表 559）包含所有可能引起中断的位。表 561 包含每个可能引起中断的事件的使能位。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

如 DMA 状态寄存器（表 559）中所述，有两组中断，正常中断和异常中断。通过向相应位的位置写 1 可清零中断。当一个组内所有已使能中断清零时，相应摘要位也将清零。当两个汇总位都被清除，中断信号被取消。如果 MAC 内核是导致中断断言的原因，则将 DMA 状态寄存器 （表 559）的 GLI、 GMI 或 GPI 位设置为高电平。

注：DMA 状态寄存器 （表 559）为 （中断）状态寄存器。仅当设置了 DMA 中断使能寄存器（表 561）的对应中断使能位时，中断引脚才会因为该状态寄存器中的任何事件被断言。

中断不排队且如果中断事件发生在驱动器响应其之前，不会生成额外的中断。例如，接收中断 （DMA 状态寄存器 （表 559）的位 6）表示一个或多个帧传输到主机缓冲区。驱动器必须扫描所有的描述符，从最后记录的位置到 DMA 拥有的第一个位置。

对于同时发生的多个事件，只生成一次中断。驱动器必须扫描 DMA 状态寄存器 （表 559）以确定中断的原因。驱动器清除 DMA 状态寄存器中的相应位以后，除非有新的中断事件发生，否则中断不会再次生成。例如，控制器生成 DMA 接收中断 （DMA 状态寄存器的位 6），且驱动器开始读取 DMA 状态寄存器。然后，“接收缓冲区不可用”（DMA 状态寄存器（状态寄存器）的位 7）发生。驱动器将接收中断清零。即使这样，因为有效的或挂起的“接收缓冲区不可用”中断， sbd_intr_o 信号没有被断言。

提供中断定时器 RIWT （接收中断看门狗定时器寄存器 （表 563）的位 7:0）以便对接收中断进行灵活控制。当使用非零值对该中断定时器进行编程时，RxDMA 将接收到的帧传输到系统存储器一完成，就会激活中断定时器，而无需断言接收中断，因为它并未在相应的接收寄存器 (RDES1[31]) 中使能。如果该定时器超限（根据预设的值），则在使能了 DMA 中断使能寄存器 （表 561）的对应 RI 的情况下，会设置 RI 位并且中断会被断言。当帧被传输到存储器中且 RI 被设置时 （因为已为该描述符使能它），定时器在超限前被禁用。

26.10.2.10 DMA 错误响应

对于由 DMA 通道发起的任何数据传输，如果从机回复错误响应，则该 DMA 会停止所有操作并更新 DMA 状态寄存器（表 559）中的错误位和致命总线错误位。只有在软复位或硬复位内核并重新初始化 DMA 之后，该 DMA 控制器才可恢复操作。此 DMA 特性适用于接收错误响应的非 AHB 接口 DMA。

26.10.3 以太网描述符

描述符结构支持多达 8 个双字（32 字节）和 IEEE 1588-2008 高级时间戳功能或 AV 功能。描述符结构的特点如下：

- 根据 DMA 总线模式寄存器（表 553）的 ATDS 位设置，描述符大小可达 4 个双字（16 字节）或 8 个双字 （32 字节）
- 支持多达 8 KB （用于巨型帧）的缓冲区。
- 选择高级时间戳时，发送描述符会将时间戳存储在 TDES6 和 TDES7 中。
- 选择高级时间戳功能或IPC全卸载时，此接收描述符结构也用于存储扩展状态 (RDES4) 和时间戳 （RDES6 和 RDES7）。
- 选择增强描述符模式时，会使能时间戳功能，软件需要为每个描述符分配 32 字节（8 个双字）的存储器。如果未使能 “时间戳”或 “接收 IPC 全卸载”引擎，则不需扩展描述符， SW 可使用默认大小为 16 字节的备用描述符。内核也需为此变化进行配置，方法为使用 DMA 寄存器的位 7 （ATDS：DMA 总线模式寄存器 （表 553）的备用描述符大小。
- 如果选择的描述符无时间戳或全IPC卸载功能，则描述符大小始终是4个双字(DES0-DES3)。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

26.10.3.1 发送描述符

发送描述符结构如图 70 所示。应用软件必须在描述符初始化期间对控制位 TDES0[31:20] 进行编程。当 DMA 更新描述符时，它将写回除 OWN 位（被清除）之外的所有控制位并更新状态位 [19:0]。发送器描述符字 0 (TDES0) 至字 3 (TDES3) 的内容分别在表 576 至表 579 中给出。

在高级时间戳支持下，可通过设置位 TTSE 为给定帧使能要获取的时间戳的快照。发送时间戳使能。（TDES0 的位 25）。如果描述符关闭 （即， OWN 位被清除），则时间戳写入 TDES6 和 TDES7。这通过状态位 TTSS 指示：发送时间戳状态。（TDES0 的位 17）。如图 70 所示。 TDES6 和 TDES7 的内容如表 580 至表 581 所述。

当高级时间戳或 IPC 卸载 （类型 2）功能使能时， SW 应设置 DMA 总线模式寄存器 [7]，以便 DMA 使用扩展的描述符大小进行操作。当此控制位复位时，TDES4-TDES7 描述符空间无效。

图 70. 发送器描述符字段

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

DMA 始终从系统存储器读取或获取描述符的 4 个双字，以获得缓冲区和控制信息（如图 71 所示）。使能高级时间戳功能支持时， TDES0 具有通道 1 和通道 2 的额外控制位 [6:3]。对于通道 0，则忽略位 6:3。位 6:3 如表 576 所述。

图 71. 发送描述符获取 （读取）

表 576. 发送描述符字 0 (TDES0)

位	符号	描述
0	DB	延期位 设置时，该位表示 MAC 因发送之前出现载波而延期。此位仅在半双工模式下有效。
1	UF	下溢错误 设置时，该位表示 MAC 因从主机存储器到达的数据延迟而终止帧。下溢错误表示在发送帧时 DMA 遇到空白发送缓冲区。发送过程进入挂起状态，并设置“发送下溢”（寄存器 5[5]）和 “发送中断”（寄存器 5[0]）。
2	ED	超延期 设置时，此位表示，如果 MAC 控制寄存器中的 “延期检查 (DC)”位设置为高电平，发送因超延期超过 24,288 位时间 （在 1,000-Mbps 模式下或使能巨型帧后的 155,680 位时间）已经结束。
6:3	CC/ SLOTNUM	CC：冲突计数 （状态字段） 这些状态位表示在发送帧之前所发生的冲突数。此计数在 “过多冲突”位 (TDES0[8]) 设置时无效。内核仅在半双工模式下更新此状态字段。 SLOTNUM：AV 模式下的插槽编号控制位 这些位表示从由 TDES2 或 TDES3 寻址的对应缓冲区获取的数据的插槽间隔。获取发送描述符后， DMA 将此字段中的插槽编号值与内核中保存的插槽间隔进行比较 （寄存器 11xx）。它仅在数值匹配时才从缓冲区获取数据。这些位仅对 AV 通道 （非通道 0）有效。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 576. 发送描述符字 0 (TDES0) *（续）*

位	符号	描述
7	VF	VLAN 帧 设置时，该位表示已发送帧是 VLAN 类型帧。
8	EC	过多冲突 设置时，此位表示在试图发送当前帧时发生了 16 个连续冲突，发送终止。如果设置了 MAC 配置寄存器中的 DR （禁用重试）位，则此位在首次冲突后设置，且帧发送终止。
9	LC	延迟冲突 设置时，此位表示由于冲突窗口（MII 模式下， 64 字节时间，包括前同步码；MII 模式下， 512 字节时间，包括前同步码和载波扩展）之后发生冲突，帧发送被终止。设置了 “下溢错误”位时，此位无效。
10	NC	无载波 设置时，此位表示在发送期间来自 PHY 的载波感应信号未被断言。
11	LC	载波丢失 设置时，此位表示在帧发送期间发生载波损失（即，帧发送期间 gmii_crs_i 信号在一个或多个发送时钟周期内处于未激活状态）。这仅限 MAC 在半双工模式下操作且发送帧无冲突时有效。
12	IPE	IP 有效载荷错误 设置时，此位表示 MAC 发送器检测到 TCP、 UDP 或 ICMP IP 数据报有效载荷中出现错误。 发送器根据从应用程序接收的 TCP、 UDP 或 ICMP 数据包字节的实际数量，来检查在IPv4或IPv6首部接收的有效载荷长度，并在不匹配时发出错误状态。
13	FF	帧刷新 设置时，此位表示 DMA/MTL 根据 CPU 发出的软件刷新命令刷新帧。
14	JT	Jabber 超时 设置时，此位表示 MAC 经历了 jabber 超时。此位仅在未设置 MAC 配置寄存器的 JD 位时设置。
15	ES	错误汇总 表示下列位的逻辑 OR： • TDES0[14]: Jabber 超时 • TDES0[13]: 帧刷新 • TDES0[11]: 载波丢失 • TDES0[10]: 无载波 • TDES0[9]: 延迟冲突 • TDES0[8]: 过多冲突 • TDES0[2]: 超延期 • TDES0[1]: 下溢错误 • TDES0[16]: IP 首部错误 • TDES0[12]: IP 有效载荷错误
16	IHE	IP 首部错误 设置时，此位表示 MAC 发送器在 IP 数据报首部中发现错误。发送器根据从应用程序中接收到的首部字节数检查 IPv4 数据包的首部长度，并在不匹配时指示错误状态。对于 IPv6 帧来说，如果主首部长度不是 40 字节，便会报告首部错误。另外， IPv4 或 IPv6 帧的以太网长度 / 类型字段值必须与同数据包一起接收到的 IP 首部版本匹配。对于 IPv4 帧，如果首部长度字段值小于 0x5，也指示错误状态。
17	TTSS	发送时间戳状态 此字段仅在描述符的 “最末段”控制位 (TDES0[29]) 设置时有效。此位设置时， TDES2 和 TDES3 具有为发送帧捕获的时间戳值。仅在描述符的 “最末段”控制位 (TDES0[29]) 设置时，该字段才有效。
19:18	-	保留

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 576. 发送描述符字 0 (TDES0) *（续）*

位	符号	描述
20	TCH	第二链接地址 设置时，此位表示描述符中的第二地址是下一描述符地址，而非第二缓冲区地址。TDES0[20] 设置时，TBS2 (TDES1[28:16]) 为“不关心”值。TDES0[21] 优先于 TDES0[20]。
21	TER	发送环端 设置时，此位表示描述符列表到达其最后的描述符。 DMA 返回列表的基址，创建描述符环。
23:22	CIC	校验和插入控制 这些位控制校验和计算和插入。位编码如下所示。 • 00：校验和插入禁用。 • 01：仅使能 IP 首部校验和计算与插入。 • 10：使能 IP 首部校验和及有效载荷校验和计算与插入，但在硬件中不计算伪首部校验和。 • 11：使能 IP 首部校验和及有效载荷校验和计算与插入，并在硬件中计算伪首部校验和。 如果未选择 IPC_FULL_OFFLOAD 配置参数，则此字段保留。
24	-	保留
25	TTSE	发送时间戳使能 设置时，该位使能 IEEE1588 硬件时间戳，用于描述符参考的发送帧。仅在“首段”控制位 (TDES0[28]) 设置时，此字段才有效。
26	DP	禁用间隔 设置时， MAC 对小于 64 字节的帧不自动添加间隔。此位复位时， DMA 会自动将间隔和 CRC 添加到短于 64 字节的帧，而且不管 DC (TDES0[27]) 位的状态如何都会添加 CRC 字段。这仅在首段 (TDES0[28]) 被设置时才有效。
27	DC	禁用 CRC 此位设置时， MAC 不在发送帧的末端附加循环冗余检查 (CRC)。这仅在首段 (TDES0[28]) 被设置时才有效。
28	FS	首段 设置时，此位表示缓冲区包含帧的首段。
29	LS	最末段 设置时，此位表示缓冲区包含帧的最末段。此位设置时，TDES1 中的 TBS1：发送缓冲区 1 容量或 TBS2：发送缓冲区 2 容量字段应含有非零值。
30	IC	完成时中断 设置时，此位在当前帧发送后设置 “发送中断”（寄存器 5[0]）。
31	OWN	Own 位 设置时，此位表示描述符为 DMA 所有。此位复位时，它表示描述符为主机所有。当它完成帧发送时，或当描述符中分配的缓冲区读取完成时， DMA 将清除该位。必须在属于同一帧的所有后续描述符都设置后，才设置该帧的首个描述符的所有权位。这样就避免了获取描述符操作和驱动程序设置所有权位之间可能发生的竞争。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 577. 发送描述符字 1(TDES1)

位	符号	描述
12:0	TBS1	发送缓冲区 1 大小 这些位表示第一个数据缓冲区的字节大小（以字节为单位）。如果该字段为 0，则 DMA 将忽略该缓冲区并使用缓冲区 2 或下一个描述符，取决于 TCH (TDES0[20]) 的值。
15:13	-	保留
28:16	TBS2	这些位表示第二个数据缓冲区的大小，以字节为单位。TDES0[20] 设置时，此字段无效。参见第 26.10.1.3 节。
31:29	-	保留

表 578. 发送描述符字 2 (TDES2)

位	符号	描述
31:0	B1ADD	缓冲区 1 地址指针 这些位表示缓冲区 1 的物理地址。对缓冲区地址对齐无限制。有关缓冲区地址对齐的详细内容，请参见第 26.10.1.2 节。

表 579. 发送描述符字 3 (TDES3)

位	符号	描述
31:0	B2ADD	缓冲区 2 地址指针 （下一描述符地址） 表示使用描述符环形结构时缓冲区 2 的物理地址。如果链接的第二个地址 (TDES1[24]) 位被设置，则此地址包含指向有下一描述符的物理内存的指针。仅在 TDES1[24] 设置时，缓冲区地址指针必须与总线宽度对齐。（LSBs 在内部忽略。）

表 580. 发送描述符字 6 (TDES6)

位	符号	描述
31:0	TTSL	发送帧时间戳低位 此字段由 DMA 通过为对应发送帧捕获的时间戳的最低有效 32 位来更新。仅在设置了描述符的最末段位 (LS) 和时间戳状态位 (TTSS) 位时，此字段才有时间戳。

表 581. 发送描述符字 7 (TDES7)

位	符号	描述
31:0	TTSH	发送帧时间戳高位 此字段由 DMA 通过为对应接收帧捕获的时间戳的最高有效 32 位来更新。仅在设置了描述符的最末段位 (LS) 和时间戳状态位 (TTSS) 位时，此字段才有时间戳。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.10.3.2 接收描述符

已接收描述符的结构如图 72 所示。如果选择了高级时间戳或 IPC 全卸载功能，就可含有 32 字节 （8 个双字）的描述符数据。

注：使能任意一种功能时， SW 应设置 DMA 总线模式寄存器 [7]，以便 DMA 以扩展的描述符大小进行操作。此控制位复位时，始终清除 RDES0[7] 和 RDES0[0]，且 RDES4-RDES7 描述符空间无效。

图 72. 接收描述符字段 （备用配置）

RDES0 的内容如表 582 所示。 RDES1 至 RDES3 的内容在表 583 至表 585 中显示。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 582. 接收描述符字段 0 (RDES0)

位	符号	描述
0	ESA	扩展状态可用 /Rx MAC 地址 如果有 “高级时间戳”或 “IP 校验和卸载”（类型 2），则设置时，此位表示在描述符字 4 (RDES4) 中可使用扩展状态。此字段仅在设置了 “最后描述符” 位 (RDES0[8]) 时有效。如果未选择 “高级时间戳功能”或 “IPC 全卸载”，则此位表示 Rx MAC 地址状态。设置时，此位表示与帧的 DA 字段匹配的 Rx MAC 地址寄存器值 （1 至 31）。复位时，此位表示与 DA 字段匹配的 Rx MAC 地址寄存器 0 值。
1	CE	CRC 错误 设置时，此位表示在接收帧上出现循环冗余检查 (CRC) 错误。此字段仅在设置了 “最后描述符” (RDES0[8]) 时有效。
2	DE	Dribble 位错误 设置时，此位表示已接收帧有非整数多字节 （奇半字节）。此位仅在 MII 模式下有效。
3	RE	接收错误 设置时，此位表示帧接收期间如果 gmii_rxdv_i 被断言，则 gmii_rxer_i 被断言。此错误也包括 MII 和半双工模式下的载波扩展错误。错误可能是扩展不足 / 无扩展、或扩展期间的错误 (rxd ≠ 0f)。
4	RWT	接收看门狗超时 设置时，此位表示接收看门狗定时器在接收当前帧时过期，且当前帧在看门狗超时后截断。
5	FT	帧类型 设置时，此位表示接收帧是以太网类型帧 （LT 字段不小于 0x0600）。此位复位时，表示接收帧是 IEEE802.3 帧。如果 Runt 帧少于 14 字节，则此位无效。
6	LC	延迟冲突 设置时，此位表示在半双工模式下接收帧时出现延迟冲突。
7	TSA	时间戳可用 /IP 校验和错误 （类型 1） / 巨型帧 如果启用了高级时间戳功能，则设置时，此位表示时间戳的快照写入描述符字 6 (RDES6) 和 7 (RDES7)。此字段仅在设置了 “最后描述符”位 (RDES0[8]) 时有效。如果选择了 IP 校验和引擎 （类型 1），则此位设置时，表示内核计算出的 16 位 IPv4 首部校验和与接收到的校验和字节不匹配。否则，此位设置时，表示巨型帧状态。巨型帧是指大于 1,518 字节 （或对于 VLAN 为 1,522 字节）的普通帧，或在使能了 Jumbo 帧处理时大于 9,018 字节（对于 VLAN 为 9,022 字节）的帧。
8	LS	最后描述符 设置时，此位表示该描述符指向帧的最后缓冲区。
9	FS	第一个描述符 设置时，此位表示该描述符包含帧的首个缓冲区。如果第一个缓冲区的大小为 0，则第二个缓冲区包含帧的起始端。如果第二个缓冲区的大小仍为 0，则下一描述符包含帧的起始端。
10	VLAN	VLAN 标志 设置时，此位表示该描述符指向的帧是由 MAC 内核标记的 VLAN 帧。
11	OE	上溢错误 设置时，此位表示接收帧因 MTL 中的缓冲区溢出而被破坏。
12	LE	长度错误 设置时，此位表示接收帧的实际长度与长度 / 类型字段不匹配。此位仅在复位了 “帧类型” (RDES0[5]) 位时有效。
13	SAF	源地址过滤失败 设置时，此位表示帧的 SA 字段未能通过 MAC 内核中的 SA 滤波器。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 582. 接收描述符字段 0 (RDES0) *（续）*

位	符号	描述
14	DE	描述符错误 设置时，此位表示由于帧不适合当前描述符缓冲区而造成帧截断，且 DMA 不拥有 “下一描述符”。帧被截断。此字段仅在设置了 “最后描述符” (RDES0[8]) 时有效。
15	ES	ES: 错误汇总 表示下列位的逻辑 OR： • RDES0[1]: CRC 错误 • RDES0[3]: 接收错误 • RDES0[4]: 看门狗超时 • RDES0[6]: 延迟冲突 • RDES0[7]: 巨型帧 • RDES4[4:3]: IP 首部 / 有效载荷错误 • RDES0[11]: 上溢错误 • RDES0[14]: 描述符错误 此字段仅在设置了 “最后描述符” (RDES0[8]) 时有效。
29:16	FL	帧长度 这些位表示传输到主机存储器（包括 CRC）的已接收帧的字节长度。设置了最后描述符 (RDES0[8])，且 “描述符错误” (RDES0[14]) 或 “上溢错误” 位复位时，此字段有效。当使能了 IP 校验和计算 （类型 1）且接收帧不是 MAC 控制帧时，帧长度也包括附加到以太网帧的两个字节。此字段在设置了 “最后描述符” 位 (RDES0[8]) 时有效。如果未设置 “最后描述符” 和 “错误汇总” 位，则此字段表示当前帧已传输的累加字节数。
30	AFM	目标地址过滤失败 设置时，此位表示帧未通过 MAC 内核中的 DA 过滤器。
31	OWN	Own 位 设置时，此位表示描述符为 MAC 子系统的 DMA 所有。此位复位时，它表示描述符为主机所有。当帧接收完成时，或与该描述符关联的缓冲区已满时， DMA 将清除该位。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 583. 接收描述符字段 1 (RDES1)

位	符号	描述
12:0	RBS1	接收缓冲区 1 大小 表示第一个数据缓冲区的大小（以字节为单位）。缓冲区大小必须是 4、8 或 16 的倍数，取决于总线宽度（32、64 或 128），即使 RDES2（缓冲区 1 地址指针）的值没有对齐。当缓冲区大小不是 4、8 或 16 的倍数时，可能产生未定义的行为。如果此字段为 0，则 DMA 将忽略此缓冲区并使用缓冲区 2 或下一描述符，取决于 RCH（位 14）的值。有关计算缓冲区大小的详细内容，请参见第 26.10.1.3 节。
13	-	保留
14	RCH	第二链接地址 设置时，此位表示描述符中的第二地址是下一描述符地址，而非第二缓冲区地址。此位设置时，RBS2 (RDES1[28:16]) 为“不关心”值。RDES1[15] 优先于 RDES1[14]。
15	RER	接收环端 设置时，此位表示描述符列表到达其最后的描述符。 DMA 返回列表的基址，创建描述符环。
28:16	RBS2	接收缓冲区 2 大小 这些位表示第二个数据缓冲区的大小（以字节为单位）。缓冲区大小必须是 4、8 或 16 的倍数，取决于总线宽度（分别为 32、64 或 128），即使 RDES3（缓冲区 2 地址指针）的值没有与总线宽度对齐。当缓冲区大小不是 4、8 或 16 的适当倍数时，可能产生未定义的行为。RDES1[14] 设置时，此字段无效。有关计算缓冲区大小的详细内容，请参见第 26.10.1.3 节。

表 584. 接收描述符字段 2 (RDES2)

位	符号	描述
31:0	B1ADD	地址指针 这些位表示缓冲区 1 的物理地址。对缓冲区地址对齐无限制，但以下情况除外：当 RDES2 值用于存储帧的起始端时， DMA 使用已配置值生成缓冲区地址。注意，在传输帧起始期间 DMA 对 RDES2[3/2/1:0] 位写入 0，但帧数据根据实际缓冲区地址指针偏移。如果地址指针指向存储帧的中间或最后部分的缓冲区，则 DMA 忽略 RDES2[3/2/1:0]（相应于总线宽度 128/64/32）。有关缓冲区地址对齐的详细内容，请参见第 26.10.1.2 节。

表 585. 接收描述符字段 3 (RDES3)

位	符号	描述
31:0	B2ADD	缓冲区 2 地址指针（下一描述符地址） 这些位表示使用描述符环形结构时缓冲区 2 的物理地址。如果第二链接地址 (RDES1[24]) 位已设置，则此地址包含指向出现下一描述符的物理内存的指针。如果 RDES1[24] 已设置，则缓冲区（下一描述符）地址指针必须与总线宽度对齐（RDES3[3、2 或 1:0] = 0，对应于总线宽度 128、64 或 32。LSB 被内部忽略。）但是，如果复位 RDES1[24]，则 RDES3 值无限制，下列情况除外：使用 RDES3 值存储帧的起始时， DMA 使用已配置值生成其缓冲区地址。如果地址指针指向存储帧的中间或最后部分的缓冲区，则 DMA 忽略 RDES3[3、2 或 1:0]（相应于总线宽度 128、64 或 32）。

写入的扩展状态如表 586 所示。扩展状态仅在与 IPC 或时间戳相关的状态可用时写入。扩展状态的可用性由 RDES0 的位 0 表示。此状态仅在选择了高级时间戳或 IPC 全卸载功能时可用。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 586. 接收描述符字段 4 (RDES4)

位	符号	描述
2:0	IPPL	IP 有效载荷类型 这些位表示 IP 数据报中封装的、由接收校验和卸载引擎 (COE) 处理的有效载荷类型。如果 COE 由于 IP 首部错误或 IP 不完整而不处理 IP 数据报的有效载荷，它还会将这些位设置为 00。 • 000: 未知或不处理 IP 有效载荷 • 001: UDP • 010: TCP • 011: ICMP • 1xx: 保留
3	IPHE	IP 首部错误 设置时，此位表示内核计算的 16 位 IPv4 首部校验和与接收到的校验和字节不匹配，或者 IP 数据报版本与以太网类型值不一致。
4	IPPLE	IP 有效载荷错误 设置时，此位表示内核计算出的 16 位 IP 有效载荷校验和 （即：TCP、 UDP 或 ICMP 校验和）与已接收片段中的相应校验和字段不匹配。如果 TCP、UDP 或 ICMP 片段长度与 IP 头文件字段中的有效载荷长度值不匹配，此位也将设置。
5	IPCSB	IP 校验和绕过 设置时，此位表示校验和卸载引擎被绕过。
6	IPv4	接收到 IPv4 数据包 设置时，此位表示接收的数据包是 IPv4 数据包。
7	IPv6	接收到 IPv6 数据包 设置时，此位表示接收的数据包是 IPv6 数据包。
11:8	MT	信息类型 编码这些位以提供已接收信息的类型。 • 0000：未接收到 PTP 信息 • 0001: SYNC （所有时钟类型） • 0010: Follow_Up （所有时钟类型） • 0011: Delay_Req （所有时钟类型） • 0100: Delay_Resp （所有时钟类型） • 0101: Pdelay_Req （在点对点透明时钟中） • 0110: Pdelay_Resp （在点对点透明时钟中） • 0111: Pdelay_Resp_Follow_Up （在点对点透明时钟中） • 1000: 通告 • 1001: 管理 • 1010: 信令 • 1011-1110: 保留 • 1111: 带 “保留”消息类型的 PTP 数据包 这些位仅在选择高级时间戳功能时有效。

RDES6 和 RDES7 包含时间戳快照。RDES6 和 RDES7 中时间戳的快照的可用性由 RDES0 描述符的位 7 指示。 RDES6 和 RDES7 的内容在表 587 和表 588 中显示。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 587. 接收描述符字段 6 (RDES6)

位	符号	描述
31:0	RTSL	接收帧时间戳低位 此字段由 DMA 通过为对应接收帧捕获的时间戳的最低有效 32 位来更新。此字段由 DMA 更新，仅用于接收帧的最后描述符，通过最后描述符状态位 (RDES0[8]) 表示。

表 588. 接收描述符字段 7 (RDES7)

位	符号	描述
31:0	RTSH	接收帧时间戳高位 该字段由 DMA 通过为相应接收帧捕获的时间戳的最高有效 32 位来更新。此字段由 DMA 更新，仅用于接收帧的最后描述符，通过最后描述符状态位 (RDES0[8]) 表示。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.1 本章导读

LCD 控制器在 LPC4350 器件中可用。

27.2 基本配置

LCD 控制器将按如下方式进行配置：

- 参见表 589 以了解时钟与电源控制。
- LCD 将通过 LCD_RST (reset # 16) 来复位。
- LCD 中断将连接至 NVIC 中编号为 7 的中断插槽。

表 589. LCD 时钟和电源控制

	基本时钟	分支时钟	工作频率
LCD 寄存器接口时钟	BASE_M4_CLK	CLK_M4_LCD	高达 204 MHz

27.3 功能要点

- 提供了用于访问帧缓冲区的 AHB 总线主机接口。
- 通过单独的 AHB 从机接口来进行设置和控制。
- 提供了双 16 深度可编程 64 位宽 FIFO，为传入的显示数据提供缓冲支持。
- 支持 4 位或 8 位接口的单面板和双面板单色超扭曲向列 (STN) 显示器。
- 支持单面板和双面板彩色 STN 显示器。
- 支持薄膜晶体管 (TFT) 彩色显示器。
- 提供了可编程显示分辨率，包括但不限于：320x200、320x240、640x200、640x240、640x480、 800x600 和 1024x768。
- 为单面板显示器提供硬件游标支持。
- 提供 15 灰阶单色、 3375 彩色 STN 以及 32K 彩色调色板式 TFT 支持。
- 为单色 STN 提供每像素 1 位、 2 位或 4 位 (bpp) 调色板式显示。
- 为彩色 STN 和 TFT 提供 1 bpp、 2 bpp、 4 bpp 或 8 bpp 调色板式彩色显示。
- 为彩色 STN 和 TFT 提供 16 bpp 真彩色非调色板式显示。
- 为彩色 TFT 提供 24 bpp 真彩色非调色板式显示。
- 为不同的显示板提供可编程定时。
- 将 256 个条目、 16 位调色板 RAM 作为 128x32 位 RAM 来使用。
- 提供帧、行和像素时钟信号。
- 为 STN 提供交流偏置信号，为 TFT 面板提供数据使能信号。
- 支持大小端格式以及 Windows CE 数据格式。
- LCD 面板时钟可以通过外设时钟或时钟输入引脚来生成。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

27.4 简介

27.4.1 可编程参数

可编程的显示器和控制器的主要参数如下所示：

- 水平前后边缘
- 水平同步脉冲宽度
- 每行的像素数
- 垂直前后边缘
- 垂直同步脉冲宽度
- 每一面板的行数
- 每行的像素时钟数
- 硬件游标控制。
- 信号极性、有源高电平或低电平
- 交流面板偏置
- 面板时钟频率
- 每个像素的位数
- 显示类型：STN 单色、 STN 彩色或 TFT
- STN 4 位或 8 位接口模式
- STN 双面板或单面板模式
- 大小端模式或 Windows CE 模式
- 中断生成事件

27.4.2 硬件游标支持

使用硬件游标功能，可减少在维护 LCD 帧缓冲区中游标图像时所需的软件开销。

如果没有该功能，则软件需要：

- 将区域图像保存在下一个游标位置的下方。
- 使用游标图像来更新该区域。
- 使用之前保存的某个图像来修复最后一个游标位置。

此外， LCD 驱动程序必须检查图形操作是否已覆盖游标，并对其进行更正。由于游标的大小为 64x64 并且使用了 24 位色彩显示，因此每个游标的移动都会读取并写入大约 75 kB 的数据。

硬件游标是通过为游标提供完全独立的图像缓冲区，并在当前游标 (X,Y) 坐标上将游标图像叠加在 LCD 输出流上，来避免这种管理的必要性。

需要移动硬件游标时，软件驱动程序会提供新的游标坐标。因此，无需对帧缓冲区作任何修改。这就极大地减少了软件开销。

游标图像保留在内部 256x32 位缓冲存储器的 LCD 控制器中。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.4.3 所支持的 LCD 面板

LCD 控制器支持以下类型的 LCD 面板：

- 有源矩阵 TFT 面板，支持 24 位总线接口。
- 单面板单色 STN 面板，支持 4 位和 8 位总线接口。
- 双面板单色 STN 面板，每个面板支持 4 位和 8 位总线接口。
- 单面板彩色 STN 面板，支持 8 位总线接口。
- 双面板彩色 STN 面板，每个面板支持 8 位总线接口。

27.4.3.1 TFT 面板

TFT 面板支持下列某种或多种彩色模式：

- 1 bpp，调色板式， 2 种颜色 (从可用颜色中选取)。
- 2 bpp，调色板式， 4 种颜色 (从可用颜色中选取)。
- 4 bpp，调色板式， 16 种颜色 (从可用颜色中选取)。
- 8 bpp，调色板式， 256 种颜色 (从可用颜色中选取)。
- 12 bpp、直接 4:4:4 RGB。
- 16 bpp、直接 5:5:5 RGB，有 1 bpp 通常不使用。该像素仍为输出，并可作为亮度位连接至 6:6:6 TFT 面板的 RGB 组件的最低有效位 (LSB)。
- 16 bpp、直接 5:6:5 RGB。
- 24 bpp、直接 8:8:8 RGB，提供 1600 多万种颜色。

每个 16 位调色板条目都是由 5 bpp (RGB) 和通用强度位组成的。与完整的 6 bpp 结构相比，它提供了更有效的存储器使用率以及性能。如果使用强度位并将其同时应用于所有三个颜色组件，则系统支持的总颜色数会翻一倍，即从最初的 32K 增至 64K。

或者，可以使用 16 个信号来驱动 5:6:5 面板，使额外的位仅用于绿色通道。

27.4.3.2 彩色 STN 面板

彩色 STN 面板支持下列某种或多种彩色模式：

- 1 bpp，调色板式， 2 种颜色 (从 3375 中选取)。
- 2 bpp，调色板式， 4 种颜色 (从 3375 中选取)。
- 4 bpp，调色板式， 16 种颜色 (从 3375 中选取)。
- 8 bpp，调色板式， 256 种颜色 (从 3375 中选取)。
- 16 bpp，直接 4:4:4 RGB，有 4 bpp 不使用。

27.4.3.3 单色 STN 面板

单色 STN 面板支持下列某种或多种模式：

- 1 bpp，调色板式， 2 种灰阶 (从 15 中选取)。
- 2 bpp，调色板式， 4 种灰阶 (从 15 中选取)。
- 4 bpp，调色板式， 16 种灰阶 (从 15 中选取)。

虽然为单色面板进行 4 bpp 以上的编程可行，但由于显示所支持的最大灰阶数为 15，因此使用这些模式没有任何益处。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

27.5 引脚说明

根据配置， LCD 控制器最多可使用 31 个引脚。对于单色 STN 面板而言，许多变体仅使用 10 个引脚。引脚是根据所选的配置按组分配的。所有 LCD 功能均与其他芯片功能进行交互。在表 590 中，仅显示引脚名称中与 LCD 相关的部分。

表 590. LCD 控制器引脚

引脚名称	类型	功能
LCDPWR	输出	LCD 面板电源使能。
LCDCLK	输出	LCD 面板时钟。
LCDENAB/LCDM (LCDAC)	输出	STN 交流偏置驱动或 TFT 数据使能输出。
LCDFP	输出	帧脉冲 (STN)。垂直同步脉冲 (TFT)
LCDLE	输出	行结束信号
LCDLP	输出	行同步脉冲 (STN)。水平同步脉冲 (TFT)
LCDVD[23:0]	输出	LCD 面板数据。使用的位数取决于面板配置。
GP_CLKIN	输入	通用 CGU 输入时钟。可用作 LCD 外部时钟 LCDCLKIN。

27.5.1 信号使用

以下章节将对各种显示类型所使用的信号进行说明。

27.5.1.1 单面板 STN 显示器所使用的信号

表 591 显示了单面板 STN 显示器所使用的信号。 UD 指上面板数据。

表 591. 单面板 STN 显示器所使用的引脚

引脚名称	4 位单色 （10 个引脚）	8 位单色 （14 个引脚）	色彩 （14 个引脚）
LCDPWR	Y	Y	Y
LCDDCLK	Y	Y	Y
LCDENAB/ LCDM	Y	Y	Y
LCDFP	Y	Y	Y
LCDLE	Y	Y	Y
LCDLP	Y	Y	Y
LCDVD[3:0]	UD[3:0]	UD[3:0]	UD[3:0]
LCDVD[7:4]	-	UD[7:4]	UD[7:4]
LCDVD[23:8]	-	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.5.1.2 双面板 STN 显示器所使用的信号

表 592 显示了双面板 STN 显示器所使用的信号。 UD 指上面板数据， LD 指下面板数据。

表 592. 双面板 STN 显示器所使用的引脚

引脚名称	4 位单色 （14 个引脚）	8 位单色 （22 个引脚）	色彩 （22 个引脚）
LCDPWR	Y	Y	Y
LCDDCLK	Y	Y	Y
LCDENAB/ LCDM	Y	Y	Y
LCDFP	Y	Y	Y
LCDLE	Y	Y	Y
LCDLP	Y	Y	Y
LCDVD[3:0]	UD[3:0]	UD[3:0]	UD[3:0]
LCDVD[7:4]	-	UD[7:4]	UD[7:4]
LCDVD[11:8]	LD[3:0]	LD[3:0]	LD[3:0]
LCDVD[15:12:00]	-	LD[7:4]	LD[7:4]
LCDVD[23:16:00]	-	-	-

27.5.1.3 TFT 显示器所使用的信号

表 593 显示了 TFT 显示器所使用的信号。

表 593. TFT 显示器所使用的引脚

引脚名称	12 位、 4:4:4 模式 （18 个引脚）	16 位、 5:6:5 模式 （22 个引脚）	16 位、1:5:5:5 模式 （24 个引脚）	24 位 （30 个引脚）
LCDPWR	Y	Y	Y	Y
LCDDCLK	Y	Y	Y	Y
LCDENAB/ LCDM	Y	Y	Y	Y
LCDFP	Y	Y	Y	Y
LCDLE	Y	Y	Y	Y
LCDLP	Y	Y	Y	Y
LCDVD[1:0]	-	-	-	RED[1:0]
LCDVD[2]	-	-	强度	RED[2]
LCDVD[3]	-	RED[0]	RED[0]	RED[3]
LCDVD[7:4]	RED[3:0]	RED[4:1]	RED[4:1]	RED[7:4]
LCDVD[9:8]	-	-	-	GREEN[1:0]
LCDVD[10]	-	GREEN[0]	强度	GREEN[2]
LCDVD[11]	-	GREEN[1]	GREEN[0]	GREEN[3]
LCDVD[15:12:00]	GREEN[3:0]	GREEN[5:2]	GREEN[4:1]	GREEN[7:4]
LCDVD[17:16:00]	-	-	-	BLUE[1:0]
LCDVD[18]	-	-	强度	BLUE[2]
LCDVD[19]	-	BLUE[0]	BLUE[0]	BLUE[3]
LCDVD[23:20]	BLUE[3:0]	BLUE[4:1]	BLUE[4:1]	BLUE[7:4]

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

27.6 寄存器描述

表 594 显示了与 LCD 控制器相关的寄存器，并对其功能进行了概述。表格后是各个寄存器的详细信息。

表 594. 寄存器简介：LCD 控制器 （基址：0x4000 8000)

名称	访问类型	地址偏移	描述	复位值 [1]	参考
TIMH	R/W	0x000	水平定时控制寄存器	0x0	表 595
TIMV	R/W	0x004	垂直定时控制寄存器	0x0	表 596
POL	R/W	0x008	时钟和信号极性控制寄存器	0x0	表 597
LE	R/W	0x00C	行结束控制寄存器	0x0	表 598
UPBASE	R/W	0x010	上面板帧基址寄存器	0x0	表 599
LPBASE	R/W	0x014	下面板帧基址寄存器	0x0	表 600
CTRL	R/W	0x018	LCD 控制寄存器	0x0	表 601
INTMSK	R/W	0x01C	中断掩码寄存器	0x0	表 602
INTRAW	RO	0x020	原始中断状态寄存器	0x0	表 603
INTSTAT	RO	0x024	掩码中断状态寄存器	0x0	表 604
INTCLR	WO	0x028	中断清除寄存器	0x0	表 605
UPCURR	RO	0x02C	上面板当前地址值寄存器	0x0	表 606
LPCURR	RO	0x030	下面板当前地址值寄存器	0x0	表 607
-	-	0x034 至 0x1FC	保留	-	-
PAL	R/W	0x200 至 0x3FC	256x16 位颜色调色板寄存器	0x0	表 608
-	-	0x400 至 0x7FC	保留	-	-
CRSR_IMG	R/W	0x800 至 0xBFC	游标图像寄存器	0x0	表 609
CRSR_CTRL	R/W	0xC00	游标控制寄存器	0x0	表 610
CRSR_CFG	R/W	0xC04	游标配置寄存器	0x0	表 611
CRSR_PAL0	R/W	0xC08	游标调色板寄存器 0	0x0	表 612
CRSR_PAL1	R/W	0xC0C	游标调色板寄存器 1	0x0	表 613
CRSR_XY	R/W	0xC10	游标 XY 位置寄存器	0x0	表 614
CRSR_CLIP	R/W	0xC14	游标剪切位置寄存器	0x0	表 615
CRSR_INTMSK	R/W	0xC20	游标中断掩码寄存器	0x0	表 616
CRSR_INTCLR	WO	0xC24	游标中断清除寄存器	0x0	表 617
CRSR_INTRAW	RO	0xC28	游标原始中断状态寄存器	0x0	表 618
CRSR_INTSTAT	RO	0xC2C	游标掩码中断状态寄存器	0x0	表 619

[1] 复位值仅反映已使用位中保存的数据，不包括保留位的内容。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.6.1 水平定时寄存器

TIMH 寄存器用于控制水平同步脉冲宽度 (HSW)、水平前边缘 (HFP) 周期、水平后边缘 (HBP) 周期以及每行像素数 (PPL)。

表 595. 水平定时寄存器 （TIMH，地址 0x4000 8000）位描述

位	符号	描述	复位值
1:0	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
7:2	PPL	每行像素数。 PPL 位字段用于指定屏幕每行 （列）的像素数。 PPL 是 6 位值，其所表示的每行像素数范围为16至1024。PPL会计算在应用HFP之前所出现的像素时钟数。 将 PPL 编写为所需的值除以 16 再减 1。实际每行的像素数为 16 * (PPL + 1)。例如，如果要获取每行 320 个像素，则将 PPL 编写为 (320/16) -1 = 19。	0x0
15:8	HSW	水平同步脉冲宽度。 8 位 HSW 字段用于在无源模式中指定行时钟的脉冲宽度，或在有源模式中指定水平同步脉冲。将 HBP 编写为所需的值减 1。	0x0
23:16	HFP	水平前边缘。 在建立 LCD 行时钟脉冲之前，8 位 HFP 字段会设置每行像素传输结束时的像素时钟间隔数。当整行像素发送至 LCD 驱动程序时，HFP 中的值会计算在使行时钟变为有效前需等待的像素时钟数。HFP 会生成 1-256 个像素时钟周期。将 HBP 编写为所需的值减 1。	0x0
31:24	HBP	水平后边缘。 8 位 HBP 字段用于指定在每行像素开始传输时插入的像素时钟周期数。在使前一行的行时钟变为无效后， HBP 中的值会计算下一个显示行出现之前需等待的像素时钟数。HBP 会生成 1-256 个像素时钟周期延迟。将 HBP 编写为所需的值减 1。	0x0

27.6.1.1 水平定时限制

DMA 会在水平显示行开始传输时请求新数据。必须有一段时间容许 DMA 传输，并使数据能够沿着 LCD 接口中的 FIFO 路径传播。数据路径延迟会对 STN 模式中水平边缘宽度的可用最小值强制实施某些限制。可用的最小值为 HSW = 2 和 HBP = 2。

单面板模式：

- HSW = 3 个像素时钟周期
- HBP = 5 个像素时钟周期
- HFP = 5 个像素时钟周期
- 面板时钟分频器 (PCD) = 1 (LCDCLK/3)

双面板模式：

- HSW = 3 个像素时钟周期
- HBP = 5 个像素时钟周期
- HFP = 5 个像素时钟周期
- PCD = 5 (LCDCLK/7)

如果在行传输开始时所提供的时间足够，比如 HSW = 6 和 HBP = 10 设置，则在 PCD 等于最小值 4 时不会损坏数据。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.6.2 垂直定时寄存器

TIMV寄存器用于控制垂直同步脉冲宽度(VSW)、垂直前边缘(VFP)周期、垂直后边缘(VBP)周期以及每一面板的行数 (LPP)。

表 596. 垂直定时寄存器 （TIMV，地址 0x4000 8004）位描述

位	符号	描述	复位值
9:0	LPP	每一面板的行数。 这是每个屏幕的有效行数。LPP 字段用于指定 LCD 面板上被控制的总行数。 LPP 是 10 位值，所允许的行数值范围为 1 到 1024。将该寄存器编写为每一 LCD 面板的行数减 1。对于双面板显示器来说，将该寄存器编写为每个上面板和下面板的行数。	0x0
15:10	VSW	垂直同步脉冲宽度。 这是水平同步行数。6 位 VSW 字段用于指定垂直同步脉冲的宽度。将寄存器编写为所需的行数减一。 对于无源 STN LCD 显示器来说，必须使用很少的水平同步行数 （例如，将其编写为零）。值越大， STN LCD 显示器上的对比度越差。	0x0
23:16	VFP	垂直前边缘。 这是在垂直同步周期之前帧传输结束时的无效行数。 8 位 VFP 字段用于指定每个帧传输结束时所插入的行时钟数。当整个像素帧发送至 LCD 显示器时，会使用 VFP 中的值来计算需等待的行时钟周期数。 在计数完成后，垂直同步信号LCDFP会在有源模式中被断言，或VSW位字段指定的其他行时钟会在无源模式中插入。 VFP 会生成 0-255 行时钟周期。可以将无源显示器的同步行数编写为零来改进对比度的显示。	0x0
31:24	VBP	垂直后边缘。 这是垂直同步周期之后帧开始传输时的无效行数。 8 位 VBP 字段用于指定每个帧开始传输时所插入的行时钟数。在使有源模式中前一帧的垂直同步信号无效，或 VSW 位字段指定的其他行时钟插入到无源模式后，VBP 计数会立即开始。在计数完成后，VBP 中的计数值会设置在下一帧开始传输之前所要插入的行时钟周期数。 VBP 会生成 0-255 个额外行时钟周期。可以将无源显示器的同步行数编写为零来改进对比度的显示。	0x0

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.6.3 时钟和信号极性寄存器

POL 寄存器用于控制时钟定时和信号极性的各种详细信息。

表 597. 时钟和信号极性寄存器 （POL，地址 0x4000 8008）位描述

位	符号	描述	复位值
4:0	PCD_LO	面板时钟分频器的较低五个位。 十位 PCD 字段是由 PCD_HI（该寄存器的 31:27 位）和 PCD_LO 组成的，它通过输入时钟来获取 LCD 面板时钟频率 LCDDCLK，即 LCDDCLK = LCDCLK/(PCD+2)。 对于具有 4 位或 8 位接口的单色 STN 显示器来说，面板时钟频率要比实际的单个像素时钟频率低 4 倍或 8 倍。对于彩色 STN 显示器而言， 22/3 个像素是每个 LCDDCLK 周期的输出，因此面板时钟频率是像素频率的 0.375 倍。 对于 TFT 显示器而言，可以在该寄存器中设置 BCD 位而忽略对像素时钟分频器的使用。 **注：**数据路径延迟会对 STN 模式中面板时钟分频器的可用最小值强制实施某些限制： 单面板彩色模式， PCD = 1 (LCDDCLK = LCDCLK/3)。 双面板彩色模式， PCD = 4 (LCDDCLK = LCDCLK/6)。 单面板单色 4 位接口模式， PCD = 2(LCDDCLK = LCDCLK/4)。 双面板单色 4 位接口模式和单面板单色 8 位接口模式， PCD = 6(LCDDCLK = LCDCLK/8)。 双面板单色 8 位接口模式，PCD = 14(LCDDCLK = LCDCLK/16)。	0x0
5	CLKSEL	时钟选择。 该位用于控制 LCDCLK 源的选择。 0 表示 LCD 模块的时钟源为 CCLK。 1 表示 LCD 模块的时钟源为 LCDCLKIN（LVD 的外部时钟输入）。	0x0
10:6	ACB	交流偏置引脚频率。 交流偏置引脚频率仅适用于 STN 显示器。这些显示器需要可以定期逆转的像素电压极性，从而可防止直流电荷聚集造成的损坏。将该字段编写为所需的值减一，从而在交流偏置引脚 LCDENAB 的每次切换之间应用行时钟数。当 LCDENAB 引脚用作数据使能信号时，如果在 TFT 模式中使用 LCD 显示器，则该字段将无效。	0x0
11	IVS	反向垂直同步。 IVS 位会反向改变 LCDFP 信号的极性。 0 表示 LCDFP 引脚处于高电平有效和低电平无效状态。 1 表示 LCDFP 引脚处于低电平有效和高电平无效状态。	0x0
12	IHS	反向水平同步。 IHS 位会反向改变 LCDLP 信号的极性。 0 表示 LCDLP 引脚处于高电平有效和低电平无效状态。 1 表示 LCDLP 引脚处于低电平有效和高电平无效状态。	0x0
13	IPC	反向面板时钟。 IPC 位用于选择面板时钟边缘，像素数据将导出到该边缘的 LCD 数据线上。 0 表示数据会导入到 LCDDCLK 上升沿的 LCD 数据线上。 1 表示数据会导入到 LCDDCLK 下降沿的 LCD 数据线上。	0x0

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 597. 时钟和信号极性寄存器 （POL，地址 0x4000 8008）位描述 *（续）*

位	符号	描述	复位值
14	IOE	反向输出使能。 该位用于选择 TFT 模式中的输出使能信号的有源极性。在该模式中，LCDENAB 引脚将用作使能功能，以指示 LCD 面板何时可以使用有效显示数据。在有源显示模式中，当 LCDENAB 处于激活状态时，数据会导入到 LCDDCLK 编程边缘的 LCD 数据线上。 0 表示 LCDENAB 输出引脚在 TFT 模式中处于高电平有效状态。 1 表示 LCDENAB 输出引脚在 TFT 模式中处于低电平有效状态。	0x0
15	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
25:16	CPL	每行时钟数。 该字段用于指定每行 LCD 面板的实际 LCDDCLK 时钟数。这是 PPL 数值除以 1（TFT 模式），除以 4 或 8（单色无源模式），除以 2 2/3（彩色无源模式），再减 1。如果要使 LCD 显示器正常工作，除了 TIMH 寄存器中的 PPL 位以外，还必须对 CPL 进行编程。	0x0
26	BCD	旁路像素时钟分频器。 将其设为 1，会忽略使用像素时钟分频器逻辑。它主要用于 TFT 显示器。	0x0
31:27	PCD_HI	面板时钟分频器的较高五个位。 请参见 PCD_LO 说明，它位于该寄存器的 [4:0] 位中。	0x0

27.6.4 行结束控制寄存器

LE 寄存器将控制是否使能行结束信号 LCDLE。在使能行结束信号时，正脉冲 （四个 LCDCLK 周期宽度）是继可编程延迟 LED 之后的 LCDLE 上的输出，它来自每个显示行的最后一个像素。如果禁用行结束信号，则它会永久处于低电平状态。

表 598. 行结束控制寄存器 （LE，地址 0x4000 800C）位描述

位	符号	描述	复位值
6:0	LED	行结束延迟。 用于控制最后一个面板时钟 LCDDCLK 上升沿中的行结束信号延迟。将 LED 编写为 LCDCLK 时钟周期数减 1。	0x0
15:7	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
16	LEE	LCD 行结束使能。 0 表示 LCDLE 已禁用 （处于低电平状态）。 1 表示 LCDLE 信号处于激活状态。	0x0
31:17	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

27.6.5 上面板帧基址寄存器

UPBASE 寄存器是彩色 LCD 上面板 DMA 基址寄存器，它用于编写上面板的帧缓冲区基址。在使能 LCD 控制器之前，必须对 LCDUPBase （双面板为 LCDLPBase）进行初始化。基址必须双字对齐。

（可选）该值可能是已更改的中间帧，用于创建双缓冲视频显示。这些寄存器会在每次 LCD 垂直同步时复制到相应的当前寄存器中。该事件会生成 LNBU 位以及可选的中断。在生成双缓冲视频时，可以使用该中断重新编写基址。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 599. 上面板帧基址寄存器 （UPBASE，地址 0x4000 8010）位描述

位	符号	描述	复位值
2:0	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
31:3	LCDUPBASE	LCD 上面板基址。 这是存储器中的上面板帧数据的起始地址，是双字对齐的地址。	0x0

27.6.6 下面板帧基址寄存器

LPBASE 寄存器是彩色 LCD 下面板 DMA 基址寄存器，它用于编写下面板的帧缓冲区的基址。在使能 LCD 控制器之前必须对 LCDLPBase 进行初始化。基址必须双字对齐。

（可选）该值可能是已更改的中间帧，用于创建双缓冲视频显示。这些寄存器会在每次 LCD 垂直同步时复制到相应的当前寄存器中。该事件会生成 LNBU 位以及可选的中断。在生成双缓冲视频时，可以使用该中断重新编写基址。

表 600. 下面板帧基址寄存器 （LPBASE，地址 0x4000 8014）位描述

位	符号	描述	复位值
2:0	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
31:3	LCDLPBASE	LCD 下面板基址。 这是存储器中下面板帧数据的起始地址，且是双字对齐地址。	0x0

27.6.7 LCD 控制寄存器

CTRL 寄存器用于控制 LCD 操作模式以及面板像素参数。

表 601. LCD 控制寄存器 （CTRL，地址 0x4000 8018）位描述

位	符号	描述	复位值
0	LCDEN	LCD 使能控制位。 0 表示 LCD 显示器已禁用。LCDLP、LCDDCLK、LCDFP、LCDENAB 和 LCDLE 信号处于低电平状态。 1 表示 LCD 显示器已使能。LCDLP、LCDDCLK、LCDFP、LCDENAB 和 LCDLE 信号处于高电平状态。 有关 LCD 电源序列的详细信息，请参见 LCD 加电和掉电序列。	0x0
3:1	LCDBPP	每个像素的 LCD 位数： 选择每个 LCD 像素的位数： 000 = 1 bpp。 001 = 2 bpp。 010 = 4 bpp。 011 = 8 bpp。 100 = 16 bpp。 101 = 24 bpp （仅限 TFT 面板）。 110 = 16 bpp、5:6:5 模式。 111 = 12 bpp、4:4:4 模式。	0x0
4	LCDBW	STN LCD 单色 / 彩色选择。 0 表示 STN LCD 显示器为彩色显示器。 1 表示 STN LCD 显示器为单色显示器。 该位在 TFT 模式中没有任何意义。	0x0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 601. LCD 控制寄存器 （CTRL，地址 0x4000 8018）位描述 *（续）*

位	符号	描述	复位值
5	LCDTFT	LCD 面板 TFT 类型选择。 0 表示 LCD 显示器为 STN 显示器。使用灰阶器。 1 表示 LCD 显示器为 TFT 显示器。不使用灰阶器。	0x0
6	LCDMONO8	单色 LCD 接口宽度。 该位会控制单色 STN LCD 显示器使用 4 位还是 8 位并行接口。它在其他模式中没有任何意义，因此必须将其编写为零。 0 表示单色 LCD 显示器使用 4 位接口。 1 表示单色 LCD 显示器使用 8 位接口。	0x0
7	LCDDUAL	单或双 LCD 面板选择。 STN LCD 接口为： 0 = 单面板。 1 = 双面板。	0x0
8	BGR	颜色格式选择。 0 = RGB：正常输出。 1 = BGR：红色和蓝色交换。	0x0
9	BEBO	大端字节顺序。 控制存储器中的字节顺序： 0 = 小端字节顺序。 1 = 大端字节顺序。	0x0
10	BEPO	大端像素顺序。 控制字节内的像素顺序： 0 = 字节内的小端顺序。 1 = 字节内的大端像素顺序。 对于 1、2 和 4 bpp 显示模式，BEPO 位可在小端和大端像素打包之间进行选择，它对 8 或 16 bpp 像素格式无效。 有关数据格式的更多信息，请参阅像素串行器。	0x0
11	LCDPWR	LCD 电源使能。 0 = 电源未门控到 LCD 面板且 LCDV[23:0] 信号禁用，（处于低电平状态）。 1 = 电源门控到 LCD 面板且 LCDV[23:0] 信号使能，（激活状态）。 有关 LCD 电源序列的详细信息，请参见 LCD 加电和掉电序列。	0x0
13:12	LCDVCOMP	LCD 垂直比较中断。 VComp 中断在以下时间生成： 00 = 垂直同步起始时。 01 = 后沿起始时。 10 = 有效视频起始时。 11 = 前沿起始时。	0x0

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 601. LCD 控制寄存器 （CTRL，地址 0x4000 8018）位描述 *（续）*

位	符号	描述	复位值
15:14	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
16	WATERMARK	LCD DMA FIFO 水印级。 控制生成 DMA 请求的时间： 0 = 当任意一个 DMA FIFO 拥有 4 个或 4 个以上空位置时，生成 LCD DMA 请求。 1 = 当任意一个 DMA FIFO 拥有 8 个或 8 个以上空位置时，生成 LCD DMA 请求。	0x0
31:17	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

27.6.8 中断掩码寄存器

INTMSK 寄存器控制是否发生各种 LCD 中断。设置此寄存器中的位可使相应原始中断 INTRAW 状态位值能够传递到 INTSTAT 寄存器，以作为中断处理。

表 602. 中断掩码寄存器 （INTMSK，地址 0x4000 801C）位描述

位	函数	描述	复位值
0	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
1	FUFIM	FIFO 下溢中断使能。 0: FIFO 下溢中断禁用。 1: 将在 FIFO 下溢时生成中断。	0x0
2	LNBUIM	LCD 下一基址更新中断使能。 0: 基址更新中断已禁用。 1: 当 LCD 基址寄存器已从下一地址寄存器更新时，中断生成。	0x0
3	VCOMPIM	垂直比较中断使能。 0: 垂直比较时间中断禁用。 1: 在到达垂直比较时间 （由 CTRL 寄存器中的 LcdVComp 字段定义）时生成中断。	0x0
4	BERIM	AHB 主机错误中断使能。 0: AHB 主机错误中断禁用。 1: 发生 AHB 主机错误时生成中断。	0x0
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.6.9 原始中断状态寄存器

INTRAW 寄存器包含了各种 LCD 控制器事件的状态标志。当 INTMSK 寄存器中的掩码位使能了这些标志时，这些标志会生成中断。

表 603. 原始中断状态寄存器 （INTRAW，地址 0x4000 8020）位描述

位	函数	描述	复位值
0	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
1	FUFRIS	FIFO 下溢原始中断状态。 在空位造成下溢问题时已读取访问上或下 DMA FIFO 会设置该位。 如果在 INTMSK 寄存器中设置了 FUFIM 位，则会生成中断。	
2	LNBURIS	LCD 下一个地址基更新原始中断状态。 与模式相关。在下一个地址寄存器成功更新了当前基址寄存器时设置该位。表示在使用双缓冲时可以加载下一个新地址。 如果在 INTMSK 寄存器中设置了 LNBUIM 位，则会生成中断。	0x0
3	VCOMPRIS	垂直比较原始中断状态。 在到达四个垂直区域（CTRL 寄存器中的 LcdVComp 位选取）之一时设置该位。 如果在 INTMSK 寄存器中设置了 VCompIM 位，则会生成中断。	0x0
4	BERRAW	AHB 主机总线错误原始中断状态。 在 AHB 主机接口接收从机中的总线错误响应时设置该位。 在 INTMSK 寄存器中设置 BERIM 位时会生成中断。	0x0
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

27.6.10 掩码中断状态寄存器

INTSTAT 寄存器为只读寄存器，它包含了 INTRAW 寄存器与 INTMASK 寄存器的逐位逻辑与运算。并且为系统中断控制器提供了所有中断的逻辑或运算。

表 604. 掩码中断状态寄存器 （INTSTAT，地址 0x4000 8024）位描述

位	函数	描述	复位值
0	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
1	FUFMIS	FIFO 下溢掩码中断状态。 在 INTRAW 寄存器中设置 FUFRIS 位并在 INTMSK 寄存器中设置 FUFIM 位后设置该位。	0x0
2	LNBUMIS	LCD 下一个地址基更新掩码中断状态。 在 INTRAW 寄存器中设置 LNBURIS 位并在 INTMSK 寄存器中设置 LNBUIM 位后设置该位。	0x0
3	VCOMPMIS	垂直比较掩码中断状态。 在 INTRAW 寄存器中设置 VCompRIS 位并在 INTMSK 寄存器中设置 VCompIM 位后设置该位。	0x0
4	BERMIS	AHB 主机总线错误掩码中断状态。 在 INTRAW 寄存器中设置 BERRAW 位并在 INTMSK 寄存器中设置 BERIM 位后设置该位。	0x0
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.6.11 中断清除寄存器

INTCLR 寄存器为只写寄存器。将逻辑 1 写入到相关的位会清除相应的中断。

表 605. 中断清除寄存器 （INTCLR，地址 0x4000 8028）位描述

位	函数	描述	复位值
0	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
1	FUFIC	FIFO 下溢中断清除。 向此位写 1 可清除 FIFO 下溢中断。	0x0
2	LNBUIC	LCD 下一个地址基更新中断清除。 向此位写入 1 可清除 LCD 下一个地址基更新中断。	0x0
3	VCOMPIC	垂直比较中断清除。 向此位写 1 可清除垂直比较中断。	0x0
4	BERIC	AHB 主机错误中断清除。 向此位写 1 可清除 AHB 主机错误中断。	0x0
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

27.6.12 上面板当前地址寄存器

UPCURR 寄存器为只读寄存器，它包含了要读取的上面板数据 DMA 地址的近似值。

注：该寄存器会随时发生更改，因此它只能粗略地表示显示位置。

如需了解 UPCURR 寄存器的相关内容，请参见表 606。

表 606. 上面板当前地址寄存器 （UPCURR，地址 0x4000 802C）位描述

位	函数	描述	复位值
31:0	LCDUPCURR	LCD 上面板当前地址。 它包含了当前 LCD 上面板数据 DMA 地址。	0x0

27.6.13 下面板当前地址寄存器

LPCURR 寄存器为只读寄存器，它包含了要读取的下面板数据 DMA 地址的近似值。

注：该寄存器会随时发生更改，因此它只能粗略地表示显示位置。

表 607. 下面板当前地址寄存器 （LPCURR，地址 0x4000 8030）位描述

位	函数	描述	复位值
31:0	LCDLPCURR	LCD 下面板当前地址。 它包含了当前 LCD 下面板数据 DMA 地址。	0x0

27.6.14 颜色调色板寄存器

PAL 寄存器包含了 256 个调色板条目，经过组织后的这些条目，每个字有两个条目共 128 个字位置。

每个字位置包含了两个调色板条目。这就意味着该调色板使用 128 个字位置。在进行小端字节顺序配置时， [15:0] 位是较低编号的调色板条目，而 [31:16] 位是较大编号的调色板条目。在进行大端字节顺序配置时，其字节顺序正好与上述顺序相反， [31:16] 位是较低编号的调色板条目，而 [15:0] 是较大编号条目。

注：仅 TFT 显示器使用所有调色板条目位。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

如需了解 PAL 寄存器的相关内容，请参见表 608。

表 608. 颜色调色板寄存器（PAL，地址从 0x4000 8200 (PAL0) 至 0x4000 83FC (PAL255)）位描述

位	函数	描述	复位值
4:0	R04_0	红色调色板数据。 对于 STN 显示器，仅使用四个 MSB，即 [4:1] 位。对于单色显示器，仅使用红色调色板数据。所有调色板寄存器具有相同的位字段。	0x0
9:5	G04_0	绿色调色板数据。	0x0
14:10	B04_0	蓝色调色板数据。	0x0
15	I0	强度 / 未使用的位。 可用作 6:6:6 TFT 显示器的 R、 G 和 B 输入的 LSB，使颜色数增加了一倍，达到 64K，其中每种颜色都具有两个不同的强度。	0x0
20:16	R14_0	红色调色板数据。 对于 STN 显示器，仅使用四个 MSB，即 [4:1] 位。对于单色显示器，仅使用红色调色板数据。所有调色板寄存器具有相同的位字段。	0x0
25:21	G14_0	绿色调色板数据。	0x0
30:26	B14_0	蓝色调色板数据。	0x0
31	I1	强度 / 未使用的位。 可用作 6:6:6 TFT 显示器的 R、 G 和 B 输入的 LSB，使颜色数增加了一倍，达到 64K，其中每种颜色都具有两个不同的强度。	0x0

27.6.15 游标图像寄存器

CRSR_IMG 寄存器区域包含了 256 字宽值，硬件游标机制将使用这些值来定义在显示器上重叠的某个图像或多个图像。图像必须始终以 LBBP 模式 （小端字节、大端像素）存储，如第 27.7.5.6 节所述。两个位用于对游标中每个像素的颜色和透明度进行编码。

游标图像 RAM 可能包含四个 32x32 游标图像或一个 64x64 游标图像，具体要取决于 CRSR_CFG 寄存器 （请参见游标配置寄存器描述）中的位 0 状态。

为游标定义的两种颜色会分别映射到 CRSR_PAL0 和 CRSR_PAL1 寄存器（请参见游标调色板寄存器描述）的对应值中。

如需了解 CRSR_IMG 寄存器的相关内容，请参见表 609。

表 609. 游标图像寄存器 （CRSR_IMG，地址 0x4000 8800 (CRSR_IMG0) 到 0x4000 8BFC (CRSR_IMG1)）位描述

位	函数	描述	复位值
31:0	CRSR_IMG	游标图像数据。 游标图像寄存器的 256 个字可定义一个 64x64 游标或 4 个 32x32 游标的外观。	0x0

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.6.16 游标控制寄存器

CRSR_CTRL 寄存器提供了常用游标功能的访问，如游标的显示开 / 关控制以及游标的编号。

如果选择的是 32x32 游标，则可以使能四个 32x32 游标之一。这四个图像中的每个图像都占用图像存储器的四分之一内存，其中 Cursor0 位于 location 0 上，然后是 Cursor1 位于地址 0x100 上， Cursor2 在 0x200 上， Cursor3 在 0x300 上。如果仅选择 64x64 游标，则在图像缓冲区中只能包含一个游标，就无法进行其他选择。

类似的帧同步规则即可应用于游标编号中，也可应用于游标坐标中。如果 CrsrFramesync 为 1，则仅在垂直帧消隐期更改所显示的游标图像。如果 CrsrFrameSync 为 0，会立即更改游标图像索引，即使当前正在扫描该游标也如此。

如需了解 CRSR_CTRL 寄存器的相关内容，请参见表 610。

表 610. 游标控制寄存器 （CRSR_CTRL，地址 0x4000 8C00）位描述

位	函数	描述	复位值
0	CrsrOn	游标使能。 0 表示游标未显示。 1 表示游标已显示。	0x0
3:1	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	0x0
5:4	CRSRNUM1_0	游标图像编号。 如果所选的游标大小为 6x64，则该字段将无效。如果所选的游标大小为 32x32： 00 = Cursor0。 01 = Cursor1。 10 = Cursor2。 11 = Cursor3。	0x0
31:6	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	0x0

27.6.17 游标配置寄存器

CRSR_CFG 寄存器为硬件游标提供了整体配置信息。

如需了解 CRSR_CFG 寄存器的相关内容，请参见表 611。

表 611. 游标配置寄存器 （CRSR_CFG，地址 0x4000 8C04）位描述

位	函数	描述	复位值
0	CrsrSize	游标大小选择。 0 表示 32x32 像素游标。允许使用 4 个定义的游标。 1 表示 64x64 像素游标。	0x0
1	FRAMESYNC	游标帧同步类型。 0 表示游标坐标是异步的。 1 表示游标坐标与帧同步脉冲同步。	0x0
31:2	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.6.18 游标调色板寄存器 0

游标调色板寄存器为游标的可视色提供了颜色调色板信息。Color0 将映射至 CRSR_PAL0。

该寄存器提供了根据 LCD 面板的功能来显示的 24 位 RGB 值，其显示方式与帧缓冲调色板输出的显示方式相同。

在 STN 单色模式中，仅使用红色字段的较高 4 个位。在 STN 彩色模式中，将使用红、蓝和绿字段的较高 4 个位。在每像素 24 位模式中，调色板寄存器的所有 24 个位都具有很重要的作用。

如需了解 CRSR_PAL0 寄存器的相关内容，请参见表 612。

表 612. 游标调色板寄存器 0 （CRSR_PAL0，地址 0x4000 8C08）位描述

位	函数	描述	复位值
7:0	RED	红色组件	0x0
15:8	GREEN	绿色组件	0x0
23:16	BLUE	蓝色组件。	0x0
31:24	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

27.6.19 游标调色板寄存器 1

游标调色板寄存器为游标的可视色提供了颜色调色板信息。Color1 将映射至 CRSR_PAL1。

该寄存器提供了根据 LCD 面板的功能来显示的 24 位 RGB 值，其显示方式与帧缓冲调色板输出的显示方式相同。

在 STN 单色模式中，仅使用红色字段的较高 4 个位。在 STN 彩色模式中，将使用红、蓝和绿字段的较高 4 个位。在每像素 24 位模式中，调色板寄存器的所有 24 个位都具有很重要的作用。

如需了解 CRSR_PAL1 寄存器的相关内容，请参见表 613。

表 613. 游标调色板寄存器 1 （CRSR_PAL1，地址 0x4000 8C0C）位描述

位	函数	描述	复位值
7:0	RED	红色组件	0x0
15:8	GREEN	绿色组件	0x0
23:16	BLUE	蓝色组件。	0x0
31:24	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

27.6.20 游标 XY 位置寄存器

CRSR_XY 寄存器用于定义游标左上角与游标覆盖左上角之间的距离。有关更多详细信息，请参阅 “游标剪切” 一节。

如果 CRSR_CFG 寄存器中的 FrameSync 位为 0，则游标位置将会立即发生改变，即使当前正在扫描该游标也如此。如果 Framesync 为 1，则仅在下一个垂直帧消隐期更改游标位置。

如需了解 CRSR_XY 寄存器的相关内容，请参见表 614。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 614. 游标 XY 位置寄存器 （CRSR_XY，地址 0x4000 8C10）位描述

位	函数	描述	复位值
9:0	CRSRX	游标原点的 X 坐标是以像素为测量单位的。 当值为 0 时，游标的左边缘位于显示器的左侧。	0x0
15:10	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
25:16	CRSRY	游标原点的 Y 坐标是以像素为测量单位的。 当值为 0 时，游标的上边缘位于显示器的顶部。	0x0
31:26	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

27.6.21 游标剪切位置寄存器

CRSR_CLIP 寄存器用于定义游标图像左上角与游标图像中的第一个被显示像素之间的距离。

不同的同步规则可应用于游标剪切寄存器中，但无法应用到游标坐标中。如果 CRSR_CFG 寄存器中的 FrameSync 位为 0，则游标剪切点将会立即发生改变，即使当前正在扫描该游标也如此。

如果 CRSR_CFG 寄存器中的 Framesync 位为 1，则仅在垂直帧消隐期更改所显示的游标图像，但前提条件是在编写剪切寄存器后游标位置得到了更新。在编程时，必须先编写剪切寄存器，然后再编写位置寄存器 (ClcdCrsrXY)，这样才能确保在指定的帧中，剪切和位置信息保持一致。

如需了解 CRSR_CLIP 寄存器的相关内容，请参见表 615。

表 615. 游标剪切位置寄存器 （CRSR_CLIP，地址 0x4000 8C14）位描述

位	函数	描述	复位值
5:0	CRSRCLIPX	X 方向的游标剪切位置。 游标图像的左边缘与游标中的第一个被显示像素之间的距离。 当值为 0 时，将显示游标行的第一个像素。	0x0
7:6	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
13:8	CRSRCLIPY	Y 方向的游标剪切位置。 游标图像顶端与游标中的第一个被显示像素之间的距离。 当值为 0 时，第一个被显示像素将位于游标图像的第一行。	0x0
31:14	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

27.6.22 游标中断掩码寄存器

CRSR_INTMSK 寄存器用于使能或禁用游标来中断处理器操作。

如需了解 CRSR_INTMSK 寄存器的相关内容，请参见表 616。

表 616. 游标中断掩码寄存器 （CRSR_INTMSK，地址 0x4000 8C20）位描述

位	功能	描述	复位值
0	CRSRIM	游标中断掩码。 在取消该位时，游标将不会中断处理器操作。 在设置该位时，游标会在处理器读取游标图像的最后一个字后立即中断处理器的操作。	0x0
31:1	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

27.6.23 游标中断清除寄存器

软件将使用 CRSR_INTCLR 寄存器来清除处理器处理的游标中断状态以及游标中断信号。

如需了解 CRSR_INTCLR 寄存器的相关内容，请参见表 617。

表 617. 游标中断清除寄存器 （CRSR_INTCLR，地址 0x4000 8C24）位描述

位	函数	描述	复位值
0	CRSRIC	游标中断清除。 在将 0 写入到该位时，将无效。 在将 1 写入到该位时，会使游标中断状态被清除。	0x0
31:1	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

27.6.24 游标原始中断状态寄存器

CRSR_INTRAW 寄存器用于指示游标中断。当通过 CRSR_INTMSK 寄存器中的 CrsrIM 位使能该寄存器时，它将为系统中断控制器提供中断操作。

如需了解 CRSR_INTRAW 寄存器的相关内容，请参见表 618。

表 618. 游标原始中断状态寄存器 （CRSR_INTRAW，地址 0x4000 8C28）位描述

位	函数	描述	复位值
0	CRSRRIS	游标原始中断状态。 在系统从游标图像中读取当前帧的最后一个数据后会立即设置游标中断状态。 将 0 写入到 CRSR_INTCLR 寄存器的 CrsrIC 位中时会清除该位。	0x0
31:1	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

27.6.25 游标掩码中断状态寄存器

CRSR_INTSTAT 寄存器用于指示游标中断，但前提条件是该中断未在 CRSR_INTMSK 寄存器中掩码。

如需了解 CRSR_INTSTAT 寄存器的相关内容，请参见表 619。

表 619. 游标掩码中断状态寄存器 （CRSR_INTSTAT，地址 0x4000 8C2C）位描述

位	函数	描述	复位值
0	CRSRMIS	游标掩码中断状态。 在系统从游标图像中读取当前帧的最后一个数据后会立即设置游标中断状态，但前提条件是 CRSR_INTMSK 寄存器中的相应位已设置。 如果 CRSR_INTMSK 寄存器被清除，则该位也为清除状态。 将 0 写入到 CRSR_INTCLR 寄存器中会清除该位。	0x0
31:1	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.7 LCD 控制器功能描述

LCD 控制器可将像素编码数据转换为所需的格式和时间，从而推动了各种单面板或双面板单色和彩色 LCD 显示器的发展。

首先将使用 AHB 接口为两个独立的、可编程、 32 位宽的 DMA FIFO （作为输入数据流缓冲区）提供像素编码数据包。

然后使用像素串行器对缓冲的像素编码数据进行解压缩。

根据 LCD 类型和模式，解压缩数据可以表示为：

- 真实显示灰色或彩色值。
- 256x16 位宽调色板 RAM 灰色或彩色值的地址。

就STN显示器而言，可以从已寻址的调色板位置获取值，也可以将真值传递到灰阶生成器。硬件编码灰阶算法逻辑将对编写的若干个帧的已寻址像素活动进行排序，以提供有效的显示外观。

对于 TFT 显示器，会直接将已寻址的调色板值或真彩色值传递到输出显示驱动程序中，从而绕过灰阶算法逻辑处理。

除了数据格式设置外， LCD 控制器还提供了一系列可编程显示控制信号，其中包括：

- LCD 面板电源使能
- 像素时钟
- 水平和垂直同步脉冲
- 显示偏置

LCD 控制器将在下列情况下生成单个中断：

- 上面板或下面板 DMA FIFO 下溢
- 基址更新意含
- 垂直比较
- 总线错误

还提供了在任意单个中断变为激活状态时，断言的单个共用中断。

图 73 显示了简化的 LCD 控制器框图。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 73. LCD 控制器框图

27.7.1 AHB 接口

LCD 控制器包括两个独立的 AHB 接口。一个是 AHB 从机接口， CPU 主要使用它来访问 LCD 控制器中的控制和数据寄存器。另一个是 AHB 主机接口， LCD 控制器使用它对存储在系统存储器其他位置中的显示数据进行 DMA 访问。 LCD DMA 控制器可以访问 AHB 上的 SRAM 和外部存储器。

27.7.1.1 AMBA AHB 从机接口

AHB 从机接口会将 LCD 控制器连接至 AHB 总线，并提供对寄存器和调色板 RAM 的 CPU 访问。

27.7.1.2 AMBA AHB 主机接口

AHB 主机接口会将显示数据从所选的从机 （存储器）传输到 LCD 控制器 DMA FIFO 中。该主机接口在经过配置后，可从 AHB 上的任何片内 SRAM、各种类型的片外静态存储器或片外 SDRAM 中获取数据。

在双面板模式中，通过单发 DMA 请求，可采用交替的方式来填充 DMA FIFO。在单面板模式中，通过单发 DMA 请求，可按序列来填充 DMA FIFO。

内置的 AHB 主机接口状态机会执行下列功能：

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

- 在识别新帧后，将上面板基址加载到 AHB 地址增量器中。
- 监控上下DMA FIFO级别，并使DMA请求被断言，以请求存储器中的显示数据，在已编写的水印上方填充显示数据，在任一 FIFO （双面板模式）中至少具有四个位置时会重新使 DMA 请求变为有效。
- 在固定长度连发期间检查是否具有1 KB的边界，如果该类情况发生，则应当适当调整该地址。
- 为固定长度连发和未定义的连发生成地址序列。
- 控制存储器与DMA FIFO之间的握手。如果在FIFO尚未完成其同步和更新序列时，该状态机会插入繁忙周期。
- 在双面板模式中，会通过单发 DMA 请求以交替方式来填充 DMA FIFO。
- 如果在活动连发期间出现错误，则会使总线错误中断被断言。
- 通过重新启动失败的访问来响应重试命令。这会在其重新同步时产生某些繁忙周期。

27.7.2 双 DMA FIFO 和相关的控制逻辑

从存储器访问的像素数据会缓冲到两个 DMA FIFO 中，单独控制这两个 DMA FIFO 就可以包含单面板和双面板两种类型的 LCD。每个 FIFO 的大小为 16 字深 64 位宽，可以在单面板模式中将其串联在一起形成有效的 32 双字节深度的 FIFO。

同步逻辑会将像素数据从 AHB 时钟域传输到 LCD 控制器时钟域中。并且会在每个 FIFO 中设置水面标记，以便在至少有四个位置可用时每个 FIFO 都可以请求数据。

如果两个 DMA FIFO 为空 （产生下溢问题），则在尝试读取任意一个 DMA FIFO 时会使中断信号被断言。

27.7.3 像素串行器

该模块会从 DMA FIFO 输出端口中读取 32 位宽的 LCD 数据，并会提取 24 bpp、16 bpp、8 bpp、 4 bpp、 2 bpp 或 1 bpp 数据，具体取决于当前的操作模式。 LCD 控制器支持大小端格式以及 Windows CE 数据格式。

根据操作模式，提取的数据可用于指向调色板 RAM 中的彩色值或灰阶值，也可作为直接应用于 LCD 面板输入的真正彩色值。

表 620 至表 622 显示了与字节顺序和 bpp 组合相对应的每个 DMA FIFO 字中的数据结构。对于每个支持的数据格式 （共三个）而言，必须从数据字中提取每个面板显示器像素所需的数据。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 620. 小端字节、小端像素顺序的 FIFO 位

<table>
<tr><th>FIFO 位</th><th>1 bpp</th><th>2 bpp</th><th>4 bpp</th><th>8 bpp</th><th>16 bpp</th><th>24 bpp</th></tr>
<tr><td>31</td><td>p31</td><td rowspan="2">p15</td><td rowspan="4">p7</td><td rowspan="6">p3</td><td rowspan="10">p1</td><td rowspan="6"></td></tr>
<tr><td>30</td><td>p30</td></tr>
<tr><td>29</td><td>p29</td><td rowspan="2">p14</td></tr>
<tr><td>28</td><td>p28</td></tr>
<tr><td>27</td><td>p27</td><td rowspan="2">p13</td><td rowspan="4">p6</td></tr>
<tr><td>26</td><td>p26</td></tr>
<tr><td>25</td><td>p25</td><td rowspan="2">p12</td></tr>
<tr><td>24</td><td>p24</td></tr>
<tr><td>23</td><td>p23</td><td rowspan="2">p11</td><td rowspan="4">p5</td><td rowspan="8">p2</td><td rowspan="14">p0</td></tr>
<tr><td>22</td><td>p22</td></tr>
<tr><td>21</td><td>p21</td><td rowspan="2">p10</td></tr>
<tr><td>20</td><td>p20</td></tr>
<tr><td>19</td><td>p19</td><td rowspan="2">p9</td><td rowspan="4">p4</td></tr>
<tr><td>18</td><td>p18</td></tr>
<tr><td>17</td><td>p17</td><td rowspan="2">p8</td></tr>
<tr><td>16</td><td>p16</td></tr>
<tr><td>15</td><td>p15</td><td rowspan="2">p7</td><td rowspan="4">p3</td><td rowspan="8">p1</td><td rowspan="12">p0</td></tr>
<tr><td>14</td><td>p14</td></tr>
<tr><td>13</td><td>p13</td><td rowspan="2">p6</td></tr>
<tr><td>12</td><td>p12</td></tr>
<tr><td>11</td><td>p11</td><td rowspan="2">p5</td><td rowspan="4">p2</td></tr>
<tr><td>10</td><td>p10</td></tr>
<tr><td>9</td><td>p9</td><td rowspan="2">p4</td></tr>
<tr><td>8</td><td>p8</td></tr>
<tr><td>7</td><td>p7</td><td rowspan="2">p3</td><td rowspan="4">p1</td><td rowspan="6">p0</td></tr>
<tr><td>6</td><td>p6</td></tr>
<tr><td>5</td><td>p5</td><td rowspan="2">p2</td></tr>
<tr><td>4</td><td>p4</td></tr>
<tr><td>3</td><td>p3</td><td rowspan="2">p1</td><td rowspan="4">p0</td></tr>
<tr><td>2</td><td>p2</td></tr>
<tr><td>1</td><td>p1</td><td rowspan="2">p0</td></tr>
<tr><td>0</td><td>p0</td></tr>
</table>

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 621. 大端字节、大端像素顺序的 FIFO 位

<table>
<tr><th>FIFO 位</th><th>1 bpp</th><th>2 bpp</th><th>4 bpp</th><th>8 bpp</th><th>16 bpp</th><th>24 bpp</th></tr>
<tr><td>31</td><td>p0</td><td rowspan="2">p0</td><td rowspan="4">p0</td><td rowspan="8">p0</td><td rowspan="16">p0</td><td rowspan="8"></td></tr>
<tr><td>30</td><td>p1</td></tr>
<tr><td>29</td><td>p2</td><td rowspan="2">p1</td></tr>
<tr><td>28</td><td>p3</td></tr>
<tr><td>27</td><td>p4</td><td rowspan="2">p2</td><td rowspan="4">p1</td></tr>
<tr><td>26</td><td>p5</td></tr>
<tr><td>25</td><td>p6</td><td rowspan="2">p3</td></tr>
<tr><td>24</td><td>p7</td></tr>
<tr><td>23</td><td>p8</td><td rowspan="2">p4</td><td rowspan="4">p2</td><td rowspan="8">p1</td><td rowspan="24">p0</td></tr>
<tr><td>22</td><td>p9</td></tr>
<tr><td>21</td><td>p10</td><td rowspan="2">p5</td></tr>
<tr><td>20</td><td>p11</td></tr>
<tr><td>19</td><td>p12</td><td rowspan="2">p6</td><td rowspan="4">p3</td></tr>
<tr><td>18</td><td>p13</td></tr>
<tr><td>17</td><td>p14</td><td rowspan="2">p7</td></tr>
<tr><td>16</td><td>p15</td></tr>
<tr><td>15</td><td>p16</td><td rowspan="2">p8</td><td rowspan="4">p4</td><td rowspan="8">p2</td><td rowspan="16">p1</td></tr>
<tr><td>14</td><td>p17</td></tr>
<tr><td>13</td><td>p18</td><td rowspan="2">p9</td></tr>
<tr><td>12</td><td>p19</td></tr>
<tr><td>11</td><td>p20</td><td rowspan="2">p10</td><td rowspan="4">p5</td></tr>
<tr><td>10</td><td>p21</td></tr>
<tr><td>9</td><td>p22</td><td rowspan="2">p11</td></tr>
<tr><td>8</td><td>p23</td></tr>
<tr><td>7</td><td>p24</td><td rowspan="2">p12</td><td rowspan="4">p6</td><td rowspan="8">p3</td></tr>
<tr><td>6</td><td>p25</td></tr>
<tr><td>5</td><td>p26</td><td rowspan="2">p13</td></tr>
<tr><td>4</td><td>p27</td></tr>
<tr><td>3</td><td>p28</td><td rowspan="2">p14</td><td rowspan="4">p7</td></tr>
<tr><td>2</td><td>p29</td></tr>
<tr><td>1</td><td>p30</td><td rowspan="2">p15</td></tr>
<tr><td>0</td><td>p31</td></tr>
</table>

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 622. 小端字节、大端像素顺序的 FIFO 位

<table>
<tr><th>FIFO 位</th><th>1 bpp</th><th>2 bpp</th><th>4 bpp</th><th>8 bpp</th><th>16 bpp</th><th>24 bpp</th></tr>
<tr><td>31</td><td>p24</td><td rowspan="2">p12</td><td rowspan="4">p6</td><td rowspan="8">p3</td><td rowspan="12">p1</td><td rowspan="8"></td></tr>
<tr><td>30</td><td>p25</td></tr>
<tr><td>29</td><td>p26</td><td rowspan="2">p13</td></tr>
<tr><td>28</td><td>p27</td></tr>
<tr><td>27</td><td>p28</td><td rowspan="2">p14</td><td rowspan="4">p7</td></tr>
<tr><td>26</td><td>p29</td></tr>
<tr><td>25</td><td>p30</td><td rowspan="2">p15</td></tr>
<tr><td>24</td><td>p31</td></tr>
<tr><td>23</td><td>p16</td><td rowspan="2">p8</td><td rowspan="4">p4</td><td rowspan="8">p2</td><td rowspan="16">p0</td></tr>
<tr><td>22</td><td>p17</td></tr>
<tr><td>21</td><td>p18</td><td rowspan="2">p9</td></tr>
<tr><td>20</td><td>p19</td></tr>
<tr><td>19</td><td>p20</td><td rowspan="2">p10</td><td rowspan="4">p5</td></tr>
<tr><td>18</td><td>p21</td></tr>
<tr><td>17</td><td>p22</td><td rowspan="2">p11</td></tr>
<tr><td>16</td><td>p23</td></tr>
<tr><td>15</td><td>p8</td><td rowspan="2">p4</td><td rowspan="4">p2</td><td rowspan="8">p1</td><td rowspan="12">p0</td></tr>
<tr><td>14</td><td>p9</td></tr>
<tr><td>13</td><td>p10</td><td rowspan="2">p5</td></tr>
<tr><td>12</td><td>p11</td></tr>
<tr><td>11</td><td>p12</td><td rowspan="2">p6</td><td rowspan="4">p3</td></tr>
<tr><td>10</td><td>p13</td></tr>
<tr><td>9</td><td>p14</td><td rowspan="2">p7</td></tr>
<tr><td>8</td><td>p15</td></tr>
<tr><td>7</td><td>p0</td><td rowspan="2">p0</td><td rowspan="4">p0</td><td rowspan="8">p0</td></tr>
<tr><td>6</td><td>p1</td></tr>
<tr><td>5</td><td>p2</td><td rowspan="2">p1</td></tr>
<tr><td>4</td><td>p3</td></tr>
<tr><td>3</td><td>p4</td><td rowspan="2">p2</td><td rowspan="4">p1</td></tr>
<tr><td>2</td><td>p5</td></tr>
<tr><td>1</td><td>p6</td><td rowspan="2">p3</td></tr>
<tr><td>0</td><td>p7</td></tr>
</table>

表 623 显示了 RGB 模式中的每个 DMA FIFO 字的数据结构。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 623. RGB 模式数据格式

FIFO 数据	24 位 RGB	16 位 (1:5:5:5 RGB)	16 位 (5:6:5 RGB)	16 位 (4:4:4 RGB)
31	-	p1 强度位	p1，蓝色 4	-
30	-	p1，蓝色 4	p1，蓝色 3	-
29	-	p1，蓝色 3	p1，蓝色 2	-
28	-	p1，蓝色 2	p1，蓝色 1	-
27	-	p1，蓝色 1	p1，蓝色 0	p1，蓝色 3
26	-	p1，蓝色 0	p1，绿色 5	p1，蓝色 2
25	-	p1，绿色 4	p1，绿色 4	p1，蓝色 1
24	-	p1，绿色 3	p1，绿色 3	p1，蓝色 0
23	p0，蓝色 7	p1，绿色 2	p1，绿色 2	p1，绿色 3
22	p0，蓝色 6	p1，绿色 1	p1，绿色 1	p1，绿色 2
21	p0，蓝色 5	p1，绿色 0	p1，绿色 0	p1，绿色 1
20	p0，蓝色 4	p1，红色 4	p1，红色 4	p1，绿色 0
19	p0，蓝色 3	p1，红色 3	p1，红色 3	p1，红色 3
18	p0，蓝色 2	p1，红色 2	p1，红色 2	p1，红色 2
17	p0，蓝色 1	p1，红色 1	p1，红色 1	p1，红色 1
16	p0，蓝色 0	p1，红色 0	p1，红色 0	p1，红色 0
15	p0、绿色 7	p0 强度位	p0，蓝色 4	-
14	p0、绿色 6	p0，蓝色 4	p0，蓝色 3	-
13	p0、绿色 5	p0，蓝色 3	p0，蓝色 2	-
12	p0、绿色 4	p0，蓝色 2	p0，蓝色 1	-
11	p0、绿色 3	p0，蓝色 1	p0，蓝色 0	p0，蓝色 3
10	p0、绿色 2	p0，蓝色 0	p0、绿色 5	p0，蓝色 2
9	p0、绿色 1	p0、绿色 4	p0、绿色 4	p0，蓝色 1
8	p0、绿色 0	p0、绿色 3	p0、绿色 3	p0，蓝色 0
7	p0，红色 7	p0、绿色 2	p0、绿色 2	p0、绿色 3
6	p0，红色 6	p0、绿色 1	p0、绿色 1	p0、绿色 2
5	p0，红色 5	p0、绿色 0	p0、绿色 0	p0、绿色 1
4	p0，红色 4	p0，红色 4	p0，红色 4	p0、绿色 0
3	p0，红色 3	p0，红色 3	p0，红色 3	p0，红色 3
2	p0，红色 2	p0，红色 2	p0，红色 2	p0，红色 2
1	p0，红色 1	p0，红色 1	p0，红色 1	p0，红色 1
0	p0，红色 0	p0，红色 0	p0，红色 0	p0，红色 0

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

27.7.4 RAM 调色板

基于 RAM 的调色板为 256 x 16 位双端口 RAM，其物理结构为 128 x 32 位。通过单字写入访问操作，可在调色板中写入两个条目。串行像素数据的最低有效位 (LSB) 会选择调色板 RAM 的上半部分或下半部分。所选的 RAM 部分将由字节顺序模式来确定。在小端模式中，会将 LSB 设置为选择上半部分，但在大端模式中，会选择调色板的下半部分。

通过 AHB 从机接口可以写入和验证像素数据值。有关所支持的颜色的信息，请参考本章之前所述的相关面板类型一节。

调色板 RAM 是双端口 RAM，其中每个端口都具有独立的控件和地址。端口 1 将用作读 / 写端口，它会连接至 AHB 从机接口。通过该端口可以写入并验证调色板条目。 Port2 用作只读端口，并连接到解包器和灰阶定标器。对于少于 16 bpp 的彩色模式，调色板会使每个像素值映射到 16 位彩色：

- 对于 TFT 显示器，会将 16 位值直接传递至像素串行器。
- 对于 STN 显示器，将首先使用灰阶器来转换 16 位值。

表 624 显示了以位方式表示的调色板数据。调色板 16 位输出使用 TFT 1:5:5:5 数据格式。在16 bpp和24 bpp TFT模式中，会忽略调色板，而将像素串行器的输出作为TFT面板数据。

表 624. TFT 模式的调色板数据存储。

位	名称（RGB 格式）	描述（RGB 格式）	名称（BGR 格式）	描述（BGR 格式）
31	I	强度 / 未使用	I	强度 / 未使用
30:26	B[4:0]	蓝色调色板数据	R[4:0]	红色调色板数据
25:21	G[4:0]	绿色调色板数据	G[4:0]	绿色调色板数据
20:16	R[4:0]	红色调色板数据	B[4:0]	蓝色调色板数据
15	I	强度 / 未使用	I	强度 / 未使用
14:10	B[4:0]	蓝色调色板数据	R[4:0]	红色调色板数据
9:5	G[4:0]	绿色调色板数据	G[4:0]	绿色调色板数据
4:0	R[4:0]	红色调色板数据	B[4:0]	蓝色调色板数据

可以使用控制寄存器位 （位 8 = BGR）进行红色和蓝色像素数据交换，以提供 BGR 数据格式支持。有关更多信息，请参见 CTRL 寄存器描述。表 625 显示了以位方式表示的 STN 彩色模式的调色板数据。

表 625. STN 彩色模式的调色板数据存储。

位	名称（RGB 格式）	描述（RGB 格式）	名称（BGR 格式）	描述（BGR 格式）
31	-	未使用	-	未使用
30:27	B[3:0]	蓝色调色板数据	R[3:0]	红色调色板数据
26	-	未使用	-	未使用
25:22	G[3:0]	绿色调色板数据	G[3:0]	绿色调色板数据
21	-	未使用	-	未使用
20:17	R[3:0]	红色调色板数据	B[3:0]	蓝色调色板数据
16	-	未使用	-	未使用
15	I	未使用	I	未使用
14:11	B[4:1]	蓝色调色板数据	R[4:1]	红色调色板数据

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 625. STN 彩色模式的调色板数据存储。（续）

位	名称（RGB 格式）	描述（RGB 格式）	名称（BGR 格式）	描述（BGR 格式）
10	B[0]	未使用	R[0]	未使用
9:6	G[4:1]	绿色调色板数据	G[4:1]	绿色调色板数据
5	G[0]	未使用	G[0]	未使用
4:1	R[4:1]	红色调色板数据	B[4:1]	蓝色调色板数据
0	R[0]	未使用	B[0]	未使用

对于 STN 单色模式，仅使用红色调色板字段位 [4:1]。但是，在 STN 彩色模式中，也使用绿色和蓝色 [4:1]。每种颜色仅使用 4 个位，因为灰阶器为每种颜色仅提供 16 种不同渐变。

表 626 显示了以位方式表示的 STN 单色模式的调色板数据。

表 626. STN 单色模式的调色板数据存储。

位	名称	描述
31	-	未使用
30:27	-	未使用
26	-	未使用
25:22	-	未使用
21	-	未使用
20:17	Y[3:0]	强度数据
16	-	未使用
15	-	未使用
14:11	-	未使用
10	-	未使用
9:6	-	未使用
5	-	未使用
4:1	Y[3:0]	强度数据
0	-	未使用

27.7.5 硬件游标

硬件游标是 LCD 控制器的完整组成部分。它使用 LCD 定时模块来表示当前扫描位置坐标，并且会截取调色板逻辑和灰阶 / 输出多路复用器之间的像素流。

所有游标编程寄存器都可以通过 LCD 从机接口进行访问。同时还为游标图像 RAM 提供了读 / 写端口。

27.7.5.1 游标操作

双端口 RAM 中包含了硬件游标。软件将通过 AHB 从机接口对其进行编程。并且通过 AHB 从机接口还可对硬件游标控制寄存器进行访问。使用这些寄存器可以修改游标的位置，并执行各种其他功能。

在使能游标操作时，硬件游标将使用水平同步信号和垂直同步信号，以及像素时钟使能和各种显示参数来计算当前的扫描坐标。

当显示点位于光标图像的边界内时，光标使用光标像素替代帧缓冲器像素。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

在显示最后一个游标像素时，系统会生成中断，软件可以使用该中断来表示修改游标图像是安全的。这样，在帧同步游标没有闪烁的情况下，就可以执行软件控制的动画。

27.7.5.2 游标大小

系统支持两种游标大小，如表 627 所示。

表 627. STN 单色模式的调色板数据存储。

X 像素数	Y 像素数	每像素位数	每行的字数	游标图像中的字数
32	32	2	2	64
64	64	2	4	256

27.7.5.3 游标移动

以下描述假设屏幕和游标原点均位于可视屏幕的左上角 （每一帧中被扫描的第一个可视像素）。图 74 显示了每个像素坐标是如何呈现在像素左上角的。

图 74. 游标移动

27.7.5.4 游标 XY 定位

CRSR_XY 寄存器用于控制游标在游标覆盖上的位置（请参见游标 XY 位置寄存器）。这就为 X 和 Y 坐标提供了单独的字段。

CRSR_CFG 寄存器（请参见游标配置寄存器）提供了用于控制游标可视行为的 FrameSync 位。

当 FrameSync 处于禁用状态时，游标会立即对已编程 CRSR_XY 值中的任意更改作出响应。如果在 LCD 扫描行上移动游标，可能会看见某些瞬时的拖尾效应。

当 FrameSync 处于激活状态时，游标仅在出现垂直同步后更新其位置。这样，游标在移动时就会比较顺畅，但游标位置仅针对一个帧更新一次。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.7.5.5 游标剪切

将 CRSR_XY 寄存器 （请参见游标 XY 位置寄存器）编写为正的二进制数值，使游标图像位于可视屏幕图像上的任意位置。当游标图像的大小超过屏幕图像，扩展至右侧或底部边缘（请参见图 75 中的 X1,Y1）时，系统会在屏幕限制范围内自动剪切游标图像。方格图案用于显示游标的可视部分。

由于 CRSR_XY 寄存器值是正整数，因此为了模拟屏幕左侧和顶部的游标剪切，系统提供了剪切位置寄存器，即 CRSR_CLIP。这将控制哪个游标图像点会位于 CRSR_CLIP 坐标上。如果要在 Y 轴上使用剪切功能，则 CRSR_XY(X) 必须为零，并且必须编写 Clip(X) 来提供到游标图像的偏移 （X2 和 X3）。使用相同的功能还可以在显示器顶部的 X 轴上执行剪切操作 (Y2)。

如果不在 X=0 或 Y=0 行上剪切游标，则必须将剪切位置寄存器 X 字段和 Y 字段编写为零，以正确显示该游标。有关错误编程造成的影响，请参见 Clip(X4,Y4)。

图 75. 游标剪切

© NXP B.V. 2012. 保留所有权利。

27.7.5.6 游标图像格式

LCD 帧缓冲区提供了三种包装格式，但简化了硬件游标图像要求，它仅支持 LBBP。这是为 Windows CE 模式提供的小端字节和大端像素。

如本章中的寄存器描述所示，图像 RAM 的起始地址相对于 LCD 基址偏移了 0x800。

所显示的游标坐标系统将通过 (X,Y) 来表示。64 x 64 是 32 x 32 格式的扩展，如图 76 所示。

图 76. 游标图像格式

32 x 32 像素格式

四个游标均保留在存储器中，其中每个游标具有相同的像素格式。表 628 列出了四个游标的基址。

表 628. 32 x 32 游标的地址

地址	描述
0x4000 8800	游标 0 起始地址。
0x4000 8900	游标 1 起始地址。
0x4000 8A00	游标 2 起始地址。
0x4000 8B00	游标 3 起始地址。

表 629 显示了游标 0 的像素映射缓冲。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 629. 32 x 32 像素游标格式的像素映射缓冲

数据位	到游标存储器的偏移					
	0	4	(8 * y)	(8 * y) +4	F8	FC
31:30	(12, 0)	(28, 0)	(12, y)	(28, y)	(12, 31)	(28,31)
29:28	(13, 0)	(29, 0)	(13, y)	(29, y)	(13, 31)	(29, 31)
27:26	(14, 0)	(30, 0)	(14, y)	(30, y)	(14, 31)	(30, 31)
25:24	(15, 0)	(31, 0)	(15, y)	(31, y)	(15, 31)	(31, 31)
23:22	(8, 0)	(24, 0)	(8, y)	(24, y)	(8, 31)	(24, 31)
21:20	(9, 0)	(25, 0)	(9, y)	(25, y)	(9, 31)	(25, 31)
19:18	(10, 0)	(26, 0)	(10, y)	(26, y)	(10, 31)	(26, 31)
17:16	(11, 0)	(27, 0)	(11, y)	(27, y)	(11, 31)	(27, 31)
15:14	(4, 0)	(20, 0)	(4, y)	(20, y)	(4, 31)	(20, 31)
13:12	(5, 0)	(21, 0)	(5, y)	(21, y)	(5, 31)	(21, 31)
11:10	(6, 0)	(22, 0)	(6, y)	(22, y)	(6, 31)	(22, 31)
9:8	(7, 0)	(23, 0)	(7, y)	(23, y)	(7, 31)	(23, 31)
7:6	(0, 0)	(16, 0)	(0, y)	(16, y)	(0, 31)	(16, 31)
5:4	(1, 0)	(17, 0)	(1, y)	(17, y)	(1, 31)	(17, 31)
3:2	(2, 0)	(18, 0)	(2, y)	(18, y)	(2, 31)	(18, 31)
1:0	(3, 0)	(19, 0)	(3, y)	(19, y)	(3, 31)	(19, 31)

64 x 64 像素格式

在64 x 64模式中，只有一个游标可以包含在存储器空间中。表630显示了64 x 64游标格式。

表 630. 64 x 64 像素游标格式的像素映射缓冲

数据位	到游标存储器的偏移								
	0	4	8	12	(16 * y)	(16 * y) +4	(16 * y) + 8	(16 * y) + 12	FC
31:30	(12, 0)	(28, 0)	(44, 0)	(60, 0)	(12, y)	(28, y)	(44, y)	(60, y)	(60, 63)
29:28	(13, 0)	(29, 0)	(45, 0)	(61, 0)	(13, y)	(29, y)	(45, y)	(61, y)	(61, 63)
27:26	(14, 0)	(30, 0)	(46, 0)	(62, 0)	(14, y)	(30, y)	(46, y)	(62, y)	(62, 63)
25:24	(15, 0)	(31, 0)	(47, 0)	(63, 0)	(15, y)	(31, y)	(47, y)	(63, y)	(63, 63)
23:22	(8, 0)	(24, 0)	(40, 0)	(56, 0)	(8, y)	(24, y)	(40, y)	(56, y)	(56, 63)
21:20	(9, 0)	(25, 0)	(41, 0)	(57, 0)	(9, y)	(25, y)	(41, y)	(57, y)	(57, 63)
19:18	(10, 0)	(26, 0)	(42, 0)	(58, 0)	(10, y)	(26, y)	(42, y)	(58, y)	(58, 63)
17:16	(11, 0)	(27, 0)	(43, 0)	(59, 0)	(11, y)	(27, y)	(43, y)	(59, y)	(59, 63)
15:14	(4, 0)	(20, 0)	(36, 0)	(52, 0)	(4, y)	(20, y)	(36, y)	(52, y)	(52, 63)
13:12	(5, 0)	(21, 0)	(37, 0)	(53, 0)	(5, y)	(21, y)	(37, y)	(53, y)	(53, 63)
11:10	(6, 0)	(22, 0)	(38, 0)	(54, 0)	(6, y)	(22, y)	(38, y)	(54, y)	(54, 63)
9:8	(7, 0)	(23, 0)	(39, 0)	(55, 0)	(7, y)	(23, y)	(39, y)	(55, y)	(55, 63)
7:6	(0, 0)	(16, 0)	(32, 0)	(48, 0)	(0, y)	(16, y)	(32, y)	(48, y)	(48, 63)
5:4	(1, 0)	(17, 0)	(33, 0)	(49, 0)	(1, y)	(17, y)	(33, y)	(49, y)	(49, 63)
3:2	(2, 0)	(18, 0)	(34, 0)	(50, 0)	(2, y)	(18, y)	(34, y)	(50, y)	(50, 63)
1:0	(3, 0)	(19, 0)	(35, 0)	(51, 0)	(3, y)	(19, y)	(35, y)	(51, y)	(51, 63)

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

游标像素编码

游标的每个像素都需要两位信息。它们被解释为 Color0、 Color1、透明、透明反转。

在编码机制中，位 1 用于选择颜色与透明 （与掩码），位 0 用于选择变体 （异或掩码）。

表 631 显示了像素编码位的分配。

表 631. 像素编码

值	描述
00	Color0。 该游标颜色是根据编写到 CRSR_PAL0 寄存器中的红绿蓝 (RGB) 值来显示的。
01	Color1。 该游标颜色是根据编写到 CRSR_PAL1 寄存器中的 RGB 值来显示的。
10	透明。 游标像素是透明的，因此其显示将保持不变。这会使可视游标呈现为非方形图形。
11	透明反转。 游标像素会采用已显示帧像素的互补色。这可以确保无论帧缓冲图像是何种颜色，游标都是可见的。

27.7.6 灰阶器

已获专利的灰阶算法推动了单色和彩色 STN 面板的发展。它为单色显示器提供了 15 种灰阶。对于 STN 彩色显示器，会同时提供三种颜色组件 (RGB) 的灰阶，从而可以呈现 3375 (15x15x15) 种颜色。灰阶器会将每个 4 位灰色值转换为若干个帧的每个像素活动序列，这在某种程度上取决于表示灰阶和颜色的显示特性。

27.7.7 上面板和下面板格式器。

在 STN 模式中，会使用格式器将灰阶器输出转换为显示器所需的并行格式。对于单色显示器，它为 4 位宽或 8 位宽，而对于彩色显示器，则为 8 位宽。表 632 显示了一个由重复序列中的 2 2/3 像素数据驱动的彩色显示。

表 632. 由 2 2/3 像素数据驱动的彩色显示

位	CLD[7]	CLD[6]	CLD[5]	CLD[4]	CLD[3]	CLD[2]	CLD[1]	CLD[0]
0	P2[绿色]	P2[红色]	P1[蓝色]	P1[绿色]	P1[红色]	P0[蓝色]	P0[绿色]	P0[红色]
1	P5[红色]	P4q[蓝色]	P4[绿色]	P4[红色]	P3[蓝色]	P3[绿色]	P3[红色]	P2[蓝色]
2	P7[蓝色]	P7[绿色]	P7[红色]	P6[蓝色]	P6[绿色]	P6[红色]	P5[蓝色]	P5[绿色]

每个格式器是由三个 3 位 (RGB) 左移位寄存器组成的。灰阶器中的 RGB 像素数据位值将同时移入相应的寄存器中。在有足够的数据可用时，会将寄存的数据多路传输到正确的位置来构造字节，从而满足 LCD 面板的 RGB 数据模式。该字节将传输到 3 字节 FIFO 中，它具有足够的空间来存储八个颜色像素。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.7.8 面板时钟生成器

面板时钟生成器模块的输出为面板时钟，即引脚 LCDDCLK。面板时钟可以基于 LCD 模块的外设时钟，也可以基于 LCD 显示器的外部时钟输入，即引脚 LCDCLKIN。无论选择哪种源，都可以进行分频以产生内部 LCD 时钟，即 LCDCLK。

面板时钟生成器经过编写后，可在 LCDCLK/2 至 LCDCLK/1025 范围内输出 LCD 面板时钟，从而与正在使用的 LCD 面板的 bpp 数据速率相符。

POL 寄存器中的 CLKSEL 位可确定所使用的基准时钟是 CCLK 还是 LCDCLKIN 引脚。

27.7.9 定时控制器

定时控制器模块的主要功能是生成水平和垂直定时面板信号。它还提供面板偏置和使能信号。所有这些定时均为寄存器可编程定时。

27.7.10 STN 和 TFT 数据选择

系统提供了对各类无源超扭曲向列 (STN) 和有源薄膜晶体管 (TFT)LCD 显示器的支持：

27.7.10.1 STN 显示器

STN 显示板需要生成算法像素图形才能为单色显示器提供伪灰阶，或为彩色显示器生成彩色图形。

27.7.10.2 TFT 显示器

TFT 显示板需要将每个像素的数字颜色值应用到显示器数据输入中。

27.7.11 中断生成

LCD 控制器将生成四个中断，以及一个共用中断。这四个中断分别为：

- 主机总线错误中断。
- 垂直比较中断。
- 下一个基址更新中断。
- FIFO 下溢中断。

通过更改 INT_MSK 寄存器中的掩码位可以使能或禁用任何一个可掩码中断（共四个）。这些中断还可组合成一个整体中断，当任何独立中断被断言且又未屏蔽时，该共用中断被断言。除了共用中断输出外，您还可以通过单个输出来使用全局中断服务程序或模块化设备驱动程序来处理中断。

从 INTRAW 寄存器中可以读取单个中断源的状态。

27.7.11.1 主机总线错误中断

在处理从机事务期间主机接口收到错误响应时，系统会使主机总线错误中断被断言。在遇到这种错误时，主机接口会进入错误状态，直至对其发出了清除该错误的信号为止。在完成相应的中断服务程序后，可以将 1 写入到 INTCLR 寄存器的 BERIC 位中来清除主机总线错误中断。该操作可以使主机接口从错误状态变为帧起始状态，并且可以启动数据显示的新帧。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

27.7.11.2 垂直比较中断

当到达 CTRL 寄存器所选的四个垂直显示区域之一时，系统会使垂直比较中断被断言。可以在以下情况开始时执行中断操作：

- 垂直同步。
- 后边缘。
- 有效视频。
- 前边缘。

将 1 写入到 INTCLR 寄存器的 VcompIC 位中可以清除该中断。

27.7.11.2.1 下一基址更新中断

当 LCDUPBASE 或 LCDLPBASE 值分别传输到 LCDUPCURR 或 LCDLPCURR 增量器时，系统会使 LCD 下一个基址更新中断被断言。它会向系统发出信号，表示可以安全使用新帧基址来更新 LCDUPBASE 或 LCDLPBASE 寄存器 （如有必要）。

将 1 写入到 INTCLR 寄存器的 LNBUIC 位中可以清除该中断。

27.7.11.2.2 FIFO 下溢中断

当从空的 DMA FIFO 中请求内部数据时，系统会使 FIFO 下溢中断被断言。在内部会生成上面板和下面板 DMA FIFO 下溢中断信号。

将 1 写入到 INTCLR 寄存器的 FUFIC 位中可以清除该中断。

27.7.12 LCD 加电和掉电序列

LCD 控制器需要执行下列加电序列：

1. 在加电后，下列信号将处于低电平状态：

- LCDLP
- LCDDCLK
- LCDFP
- LCDENAB/ LCDM
- LCDVD[23:0]
- LCDLE

2. 当 LCD 电源趋于稳定后，系统会将 1 写入到 CTRL 寄存器的 LcdEn 位中。这就会使下列信号处于激活状态：

- LCDLP
- LCDDCLK
- LCDFP
- LCDENAB/ LCDM
- LCDLE

LCDV[23:0] 信号仍处于失效状态。

3. 第2步中的信号趋于稳定后，非LCD控制器控制或提供的比较电压会应用于LCD面板中。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

4. 在必要时，可以使用软件或硬件定时器在面板显示的控制信号和电源应用之间提供特定于显示的最小延迟时间。在达到该时间间隔后，将 1 写入到 CTRL 寄存器的 LcdPwr 位中进行面板加电，进而将 LCDPWR 信号设置为高电平，并使 LCDV[23:0] 信号进入激活状态。LCDPWR 信号旨在为 LCD 面板提供电源。

掉电序列正好与上述四个步骤相反，必须严格遵循该序列，此时应将相应的寄存器位写为 0。

图 77 显示了加电和掉电序列。

图 77. 加电和掉电序列

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.8 LCD 时序图

(1) 有源数据线会因 STN 面板的类型 （4 位、 8 位、彩色和单色）以及单帧或双帧而异。

(2) LCD 控制器会选择 LCD 面板时钟并进行定阶，用于产生 LCDCLK。

(3) LCDLP 信号的持续时间是由 TIMH 寄存器的 HSW 字段来控制的。

(4) LCDLP 信号的极性是由 POL 寄存器的 IHS 位来确定的。

图 78. STN 显示器的水平时序

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

(1) 信号极性会因某些显示器而异。

图 79. STN 显示器的垂直时序

(1) 有源数据线会因 TFT 面板类型而异。

(2) LCD 控制器会选择 LCD 面板时钟并进行定阶，用于产生 LCDCLK。

(3) LCDLP 的持续时间是由 TIMH 寄存器的 HSW 字段来控制的。

(4) LCDLP 信号的极性是由 POL 寄存器的 IHS 位来确定的。

图 80. TFT 显示器的水平时序

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

(1) 极性会因某些显示器而异。

图 81. TFT 显示器的垂直时序

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.9 LCD 面板信号的使用

表 633. STN 单面板模式的 LCD 面板连接

外部引脚	4 位单色 STN 单面板		8 位单色 STN 单面板		彩色 STN 单面板	
	用到的 LPC43xx 引脚	LCD 功能	用到的 LPC43xx 引脚	LCD 功能	用到的 LPC43xx 引脚	LCD 功能
LCDVD23	-	-	-	-	-	-
LCDVD22	-	-	-	-	-	-
LCDVD21	-	-	-	-	-	-
LCDVD20	-	-	-	-	-	-
LCDVD19	-	-	-	-	-	-
LCDVD18	-	-	-	-	-	-
LCDVD17	-	-	-	-	-	-
LCDVD16	-	-	-	-	-	-
LCDVD15	-	-	-	-	-	-
LCDVD14	-	-	-	-	-	-
LCDVD13	-	-	-	-	-	-
LCDVD12	-	-	-	-	-	-
LCDVD11	-	-	-	-	-	-
LCDVD10	-	-	-	-	-	-
LCDVD9	-	-	-	-	-	-
LCDVD8	-	-	-	-	-	-
LCDVD7	-	-	P8_4	UD[7]	P8_4	UD[7]
LCDVD6	-	-	P8_5	UD[6]	P8_5	UD[6]
LCDVD5	-	-	P8_6	UD[5]	P8_6	UD[5]
LCDVD4	-	-	P8_7	UD[4]	P8_7	UD[4]
LCDVD3	P4_2	UD[3]	P4_2	UD[3]	P4_2	UD[3]
LCDVD2	P4_3	UD[2]	P4_3	UD[2]	P4_3	UD[2]
LCDVD1	P4_4	UD[1]	P4_4	UD[1]	P4_4	UD[1]
LCDVD0	P4_1	UD[0]	P4_1	UD[0]	P4_1	UD[0]
LCDLP	P7_6	LCDLP	P7_6	LCDLP	P7_6	LCDLP
LCDENAB/ LCDM	P4_6	LCDENAB/ LCDM	P4_6	LCDENAB/ LCDM	P4_6	LCDENAB/ LCDM
LCDFP	P4_5	LCDFP	P4_5	LCDFP	P4_5	LCDFP
LCDDCLK	P4_7	LCDDCLK	P4_7	LCDDCLK	P4_7	LCDDCLK
LCDLE	P7_0	LCDLE	P7_0	LCDLE	P7_0	LCDLE
LCDPWR	P7_7	CDPWR	P7_7	LCDPWR	P7_7	LCDPWR
GP_CLKIN	PF_4	LCDCLKIN	PF_4	LCDCLKIN	PF_4	LCDCLKIN

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 634. STN 双面板模式的 LCD 面板连接

外部引脚	4 位单色 STN 双面板		8 位单色 STN 双面板		彩色 STN 双面板	
	用到的 LPC43xx 引脚	LCD 功能	用到的 LPC43xx 引脚	LCD 功能	用到的 LPC43xx 引脚	LCD 功能
LCDVD23	-	-	-	-	-	-
LCDVD22	-	-	-	-	-	-
LCDVD21	-	-	-	-	-	-
LCDVD20	-	-	-	-	-	-
LCDVD19	-	-	-	-	-	-
LCDVD18	-	-	-	-	-	-
LCDVD17	-	-	-	-	-	-
LCDVD16	-	-	-	-	-	-
LCDVD15	-	-	PB_4	LD[7]	PB_4	LD[7]
LCDVD14	-	-	PB_5	LD[6]	PB_5	LD[6]
LCDVD13	-	-	PB_6	LD[5]	PB_6	LD[5]
LCDVD12	-	-	P8_3	LD[4]	P8_3	LD[4]
LCDVD11	P4_9	LD[3]	P4_9	LD[3]	P4_9	LD[3]
LCDVD10	P4_10	LD[2]	P4_10	LD[2]	P4_10	LD[2]
LCDVD9	P4_8	LD[1]	P4_8	LD[1]	P4_8	LD[1]
LCDVD8	P7_5	LD[0]	P7_5	LD[0]	P7_5	LD[0]
LCDVD7	-	-		UD[7]	P8_4	UD[7]
LCDVD6	-	-	P8_5	UD[6]	P8_5	UD[6]
LCDVD5	-	-	P8_6	UD[5]	P8_6	UD[5]
LCDVD4	-	-	P8_7	UD[4]	P8_7	UD[4]
LCDVD3	P4_2	UD[3]	P4_2	UD[3]	P4_2	UD[3]
LCDVD2	P4_3	UD[2]	P4_3	UD[2]	P4_3	UD[2]
LCDVD1	P4_4	UD[1]	P4_4	UD[1]	P4_4	UD[1]
LCDVD0	P4_1	UD[0]	P4_1	UD[0]	P4_1	UD[0]
LCDLP	P7_6	LCDLP	P7_6	LCDLP	P7_6	LCDLP
LCDENAB/ LCDM	P4_6	LCDENAB/ LCDM	P4_6	LCDENAB/ LCDM	P4_6	LCDENAB/ LCDM
LCDFP	P4_5	LCDFP	P4_5	LCDFP	P4_5	LCDFP
LCDDCLK	P4_7	LCDDCLK	P4_7	LCDDCLK	P4_7	LCDDCLK
LCDLE	P7_0	LCDLE	P7_0	LCDLE	P7_0	LCDLE
LCDPWR	P7_7	LCDPWR	P7_7	LCDPWR	P7_7	LCDPWR
GP_CLKIN	PF_4	LCDCLKIN	PF_4	LCDCLKIN	PF_4	LCDCLKIN

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 635. TFT 面板的 LCD 面板连接

外部引脚	TFT 12 位（4:4:4 模式）		TFT 16 位 （5:6:5 模式）		TFT 16 位（1:5:5:5 模式）		TFT 24 位	
	用到的 LPC43xx 引脚	LCD 功能	用到的 LPC43xx 引脚	LCD 功能	用到的 LPC43xx 引脚	LCD 功能	用到的 LPC43xx 引脚	LCD 功能
LCDVD23	PB_0	蓝色 3	PB_0	蓝色 4	PB_0	蓝色 4		蓝色 7
LCDVD22	PB_1	蓝色 2	PB_1	蓝色 3	PB_1	蓝色 3		蓝色 6
LCDVD21	PB_2	蓝色 1	PB_2	蓝色 2	PB_2	蓝色 2		蓝色 5
LCDVD20	PB_3	蓝色 0	PB_3	蓝色 1	PB_3	蓝色 1		蓝色 4
LCDVD19	-	-	P7_1	蓝色 0	P7_1	蓝色 0		蓝色 3
LCDVD18	-	-	-	-	P7_2	强度		蓝色 2
LCDVD17	-	-	-	-	-	-	P7_3	蓝色 1
LCDVD16	-	-	-	-	-	-	P7_4	蓝色 0
LCDVD15	PB_4	绿色 3	PB_4	绿色 5	PB_4	绿色 4	PB_4	绿色 7
LCDVD14	PB_5	绿色 2	PB_5	绿色 4	PB_5	绿色 3	PB_5	绿色 6
LCDVD13	PB_6	绿色 1	PB_6	绿色 3	PB_6	绿色 2	PB_6	绿色 5
LCDVD12	P8_3	绿色 0	P8_3	绿色 2	P8_3	绿色 1	P8_3	绿色 4
LCDVD11	-	-	P4_9	绿色 1	P4_9	绿色 0	P4_9	绿色 3
LCDVD10	-	-	P4_10	绿色 0	P4_10	强度	P4_10	绿色 2
LCDVD9	-	-	-	-	-	-	P4_8	绿色 1
LCDVD8	-	-	-	-	-	-	P7_5	绿色 0
LCDVD7	P8_4	红色 3	P8_4	红色 4	P8_4	红色 4	P8_4	红色 7
LCDVD6	P8_5	红色 2	P8_5	红色 3	P8_5	红色 3	P8_5	红色 6
LCDVD5	P8_6	红色 1	P8_6	红色 2	P8_6	红色 2	P8_6	红色 5
LCDVD4	P8_7	红色 0	P8_7	红色 1	P8_7	红色 1	P8_7	红色 4
LCDVD3	-	-	P4_2	红色 0	P4_2	红色 0	P4_2	红色 3
LCDVD2	-	-	-	-	P4_3	强度	P4_3	红色 2
LCDVD1	-	-	-	-	-	-	P4_4	红色 1
LCDVD0	-	-	-	-	-	-	P4_1	红色 0
LCDLP	P7_6	LCDLP	P7_6	LCDLP	P7_6	LCDLP	P7_6	LCDLP
LCDENAB/ LCDM	P4_6	LCDENAB/ LCDM	P4_6	LCDENAB/ LCDM	P4_6	LCDENAB/ LCDM	P4_6	LCDENAB/ LCDM
LCDFP	P4_5	LCDFP	P4_5	LCDFP	P4_5	LCDFP	P4_5	LCDFP
LCDDCLK	P4_7	LCDDCLK	P4_7	LCDDCLK	P4_7	LCDDCLK	P4_7	LCDDCLK
LCDLE	P7_0	LCDLE	P7_0	LCDLE	P7_0	LCDLE	P7_0	LCDLE
LCDPWR	P7_7	LCDPWR	P7_7	LCDPWR	P7_7	LCDPWR	P7_7	LCDPWR
GP_CLKIN	PF_4	LCDCLKIN	PF_4	LCDCLKIN	PF_4	LCDCLKIN	PF_4	LCDCLKIN

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

28.1 本章导读

SCT 在所有 LPC43xx 器件中可用。

28.2 基本配置

SCT 将按如下方式进行配置：

- 参见表 636 以了解时钟与电源控制。
- SCT 将通过 SCT_RST (reset #37) 来复位。
- 将通过 GIMA 连接 SCT 的输入和输出 （参见第 16 章）。
- SCT 共用中断将连接至 NVIC 的编号为 10 的插槽中。将通过定时器匹配通道对 SCT 2、6、14 输出执行或运算，并连接至事件路由器的第 13、14 和 16 插槽中（参见表 24）。
- 如果要将 SCT 输出 0 和 1 连接至 GPDMA，请使用 CREG 模块中的 DMAMUX 寄存器（参见表 41），并在 DMA 通道配置寄存器 （第 19.6.20 节）中使能 GPDMA 通道。

表 636. SCT 时钟和电源控制

	基本时钟	分支时钟	工作频率
SCT	BASE_M4_CLK	CLK_M4_SCT	最大 204 MHz

28.3 功能要点

- 两个 16 位计数器或一个 32 位计数器。
- 由总线时钟或所选输入计时的计数器。
- 正计数或倒计数。
- 状态变量可以跨多个计数周期进行定序。
- 事件可以通过计数器匹配条件、输入（或输出）条件，以及指定状态中的匹配和 / 或输入 / 输出条件组合来定义。
- 事件控制输出、中断和 DMA 请求。
- 所选事件可以限制、终止、启动或停止计数器操作。
- 提供如下支持：
  - 8 个输入
  - 16 个输出
  - 16 个匹配 / 捕获寄存器
  - 16 个事件
  - 32 个状态

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

28.4 简介

状态可配置定时器 (SCT) 可以执行各种定时、计数、输出调制和输入捕获操作。

最基本的一个用户可编程选项是 SCT 究竟是作为两个 16 位计数器，还是一个统一的 32 位计数器来使用。在使用两个计数器时，除了计数器值外，下列操作要素对于每个计数器来说都是独立的：

- 状态变量
- 限制、终止、停止和启动条件
- 匹配 / 捕获寄存器的值，以及重新载入或捕获控制值

在使用两个计数器时，下列操作要素对于 SCT 是通用的，但事件、输出、中断和 DMA 请求可以使用任一计数器中的匹配条件：

- 时钟选择
- 输入
- 事件
- 输出
- 中断
- DMA 请求

注：本文使用 “总线错误”一词来表示使处理器接受异常的 SCT 响应。

图 82. SCT 框图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 83. SCT 计数器和选择逻辑

28.5 引脚说明

SCT输入可从外部引脚或几个内部源产生。每个SCT输入都连接到一个定义输入源的GIMA寄存器。

在 CREG6 中（参见表 43）将 CTOUCTRL 位设为 0 时，SCT 输出连接到 CTOUT_n 引脚并与定时器匹配输出进行或操作。一些 SCT 输出可一次连接至多个目标，例如，一个外部引脚和事件路由器。

表 637. SCT 输入和输出

描述	引脚功能	内部信号	默认（参见 GIMA，表 138）	CTOUTCTRL 位（参见表 43）
SCT 输入				
SCT 输入 0	CTIN_0	-	是	-
SCT 输入 1	CTIN_1	-	是	-
	-	USART2 TX 有效	否	-
SCT 输入 2	CTIN_2	-	是	-
SCT 输入 3	CTIN_3	-	是	-
	-	USART0 TX 有效	否	-
	-	I2S1_RX_MWS	否	-
	-	I2S1_TX_MWS	否	-
SCT 输入 4	CTIN4	-	是	-
		USART0 RX 有效	否	-
		I2S1_RX_MWS	否	-
		I2S1_TX_MWS	否	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 637. SCT 输入和输出 *（续）*

描述	引脚功能	内部信号	默认（参见 GIMA，表 138）	CTOUTCTRL 位（参见表 43）
SCT 输入 5	CTIN_5	-	是	-
	-	USART2 TX 有效	否	-
SCT 输入 6	CTIN_6	-	是	-
	-	USART3 TX 有效	否	-
	-	I2S0_RX_MWS	否	-
	-	I2S0_TX_MWS	否	-
SCT 输入 7	CTIN_7	-	是	-
	-	USART3 RX 有效	否	-
	-	SOF0	否	-
	-	SOF1	否	-
SCT 输出				
SCT 输出 0 或者定时器 0 匹配输出 0	CTOUT_0	-	-	0
SCT 输出 0	CTOUT_0	-	-	1
SCT 输出 1 或者定时器 0 匹配输出 1	CTOUT_1	-	-	0
SCT 输出 1	CTOUT_1	-	-	1
SCT 输出 2 或者定时器 0 匹配输出 2	CTOUT_2	事件路由器输入 13	-	0
SCT 输出 2	CTOUT_2	事件路由器输入 13	-	1
SCT 输出 3 或者定时器 0 匹配输出 3	CTOUT_3	T1 捕获通道 3	-	0
SCT 输出 3	CTOUT_3	T1 捕获通道 3	-	1
SCT 输出 4 或者定时器 1 匹配输出 0	CTOUT_4	-	-	0
SCT 输出 4	CTOUT_4	-		1
SCT 输出 5 或者定时器 1 匹配输出 1	CTOUT_5	-	-	0
SCT 输出 5	CTOUT_5	-		1
SCT 输出 6 或者定时器 1 匹配输出 2	CTOUT_6	事件路由器输入 14	-	0
SCT 输出 6	CYOUT_6	事件路由器输入 14		1
SCT 输出 7 或者定时器 1 匹配输出 3	CTOUT_7	T2 捕获通道 3	-	0
SCT 输出 7	CTOUT_7	T2 捕获通道 3		1
SCT 输出 8 或者定时器 2 匹配输出 0	CTOUT_8	ADC 启动 1 输入（ADC CR 寄存器位 START = 0x3）	-	0
SCT 输出 8	CTOUT_8	ADC 启动 1 输入（ADC CR 寄存器位 START = 0x3）		1
SCT 输出 9 或者定时器 2 匹配输出 1	CTOUT_9	-	-	0
SCT 输出 9	CTOUT_9	-		1
SCT 输出 10 或者定时器 2 匹配输出 2	CTOUT_10	-	-	0
SCT 输出 10	CTOUT_10	-		1
SCT 输出 11 或者定时器 2 匹配输出 3	CTOUT_11	T3 捕获通道 3	-	0
SCT 输出 11	CTOUT_11	T3 捕获通道 3	-	1
SCT 输出 12 或者定时器 3 匹配输出 0	CTOUT_12	-	-	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 637. SCT 输入和输出 （续）

描述	引脚功能	内部信号	默认 （参见 GIMA，表 138）	CTOUTCTRL 位 （参见表 43）
SCT 输出 12	CTOUT_12	-	-	1
SCT 输出 13 或者定时器 3 匹配输出 1	CTOUT_13	-	-	0
SCT 输出 13	CTOUT_13	-	-	1
SCT 输出 14 或者定时器 3 匹配输出 2	CTOUT_14	事件路由器输入 16	-	0
SCT 输出 14	CTOUT_14	事件路由器输入 16	-	1
SCT 输出 15 或者定时器 3 匹配输出 3	CTOUT_15	T0 捕获通道 3/ADC 启动 0 输入 （ADC CR 寄存器 START 位 = 0x2）	-	0
SCT 输出 15	CTOUT_15	T0 捕获通道 3/ADC 启动 0 输入 （ADC CR 寄存器 START 位 = 0x2）	-	1

28.6 寄存器描述

状态可配置定时器的寄存器地址如表 638 所示。对于大部分 SCT 寄存器而言，其功能将取决于特定的其他寄存器位的设置：

1. CONFIG 寄存器中的 UNIFY 位可确定将 SCT 用作一个 32 位寄存器 （作为一个 32 位计数器 / 定时器使用）还是两个名为 L 和 H 的 16 位计数器 / 定时器。UNIFY 位的设置会体现在寄存器映射中：
   - UNIFY = 1：仅使用一个寄存器 （作为一个 32 位计数器 / 定时器使用）。
   - UNIFY = 0：通过 32 位读或写操作访问 L 寄存器和 H 寄存器，也可以分别对 L 寄存器和 H 寄存器执行读或写操作 （作为两个 16 位计数器 / 定时器使用）。

   通常，在访问任何其他寄存器之前，对 CONFIG 寄存器执行写操作来配置 UNIFY 位。
2. REGMODE 寄存器中的 REGMODEn 位可确定每组匹配 / 捕获寄存器是使用匹配功能还是捕获功能：
   - REGMODEn = 1：寄存器作为匹配和重新载入寄存器使用。
   - REGMODEn = 0：寄存器作为捕获和捕获控制寄存器使用。

表 638. 寄存器简介：状态可配置定时器 （基址 0x4000 0000）

名称	访问类型	地址偏移	描述	复位值	参考
CONFIG	R/W	0x000	SCT 配置寄存器	0x0000 7E00	表 639
CTRL	R/W	0x004	SCT 控制寄存器	0x0004 0004	表 640
CTRL_L	R/W	0x004	SCT 控制寄存器低速计数器 16 位	0x0004 0004	表 640
CTRL_H	R/W	0x006	SCT 控制寄存器高速计数器 16 位	0x0004 0004	表 640
LIMIT	R/W	0x008	SCT 限值寄存器	0x0000 0000	表 641
LIMIT_L	R/W	0x008	SCT 限值寄存器低速计数器 16 位	0x0000 0000	表 641
LIMIT_H	R/W	0x00A	SCT 限值寄存器高速计数器 16 位	0x0000 0000	表 641
HALT	R/W	0x00C	SCT 终止条件寄存器	0x0000 0000	表 642

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 638. 寄存器简介：状态可配置定时器 （基址 0x4000 0000）*（续）*

名称	访问类型	地址偏移	描述	复位值	参考
HALT_L	R/W	0x00C	SCT 终止条件寄存器低速计数器 16 位	0x0000 0000	表 642
HALT_H	R/W	0x00E	SCT 终止条件寄存器高速计数器 16 位	0x0000 0000	表 642
STOP	R/W	0x010	SCT 停止条件寄存器	0x0000 0000	表 643
STOP_L	R/W	0x010	SCT 停止条件寄存器低速计数器 16 位	0x0000 0000	表 643
STOP_H	R/W	0x012	SCT 停止条件寄存器高速计数器 16 位	0x0000 0000	表 643
START	R/W	0x014	SCT 启动条件寄存器	0x0000 0000	表 644
START_L	R/W	0x014	SCT 启动条件寄存器低速计数器 16 位	0x0000 0000	表 644
START_H	R/W	0x016	SCT 启动条件寄存器高速计数器 16 位	0x0000 0000	表 644
-	-	0x018 - 0x03C	保留		-
COUNT	R/W	0x040	SCT 计数器寄存器	0x0000 0000	表 645
COUNT_L	R/W	0x040	SCT 计数器寄存器低速计数器 16 位	0x0000 0000	表 645
COUNT_H	R/W	0x042	SCT 计数器寄存器高速计数器 16 位	0x0000 0000	表 645
STATE	R/W	0x044	SCT 状态寄存器	0x0000 0000	表 646
STATE_L	R/W	0x044	SCT 状态寄存器低速计数器 16 位	0x0000 0000	表 646
STATE_H	R/W	0x046	SCT 状态寄存器高速计数器 16 位	0x0000 0000	表 646
INPUT	RO	0x048	SCT 输入寄存器	0x0000 0000	表 647
REGMODE	R/W	0x04C	SCT 匹配 / 捕获寄存器模式寄存器	0x0000 0000	表 648
REGMODE_L	R/W	0x04C	SCT 匹配 / 捕获寄存器模式寄存器低速计数器 16 位	0x0000 0000	表 648
REGMODE_H	R/W	0x04E	SCT 匹配 / 捕获寄存器模式寄存器高速计数器 16 位	0x0000 0000	表 648
OUTPUT	R/W	0x050	SCT 输出寄存器	0x0000 0000	表 649
OUTPUTDIRCTRL	R/W	0x054	SCT 输出计数器方向控制寄存器	0x0000 0000	表 650
RES	R/W	0x058	SCT 解决冲突寄存器	0x0000 0000	表 651
DMAREQ0	R/W	0x05C	SCT DMA 请求 0 寄存器	0x0000 0000	表 652
DMAREQ1	R/W	0x060	SCT DMA 请求 1 寄存器	0x0000 0000	表 653
-	-	0x064 - 0x0EC	保留	-	-
EVEN	R/W	0x0F0	SCT 事件使能寄存器	0x0000 0000	表 654
EVFLAG	R/W	0x0F4	SCT 事件标志寄存器	0x0000 0000	表 655
CONEN	R/W	0x0F8	SCT 冲突使能寄存器	0x0000 0000	表 656
CONFLAG	R/W	0x0FC	SCT 冲突标志寄存器	0x0000 0000	表 657
MATCH0 至 MATCH15	R/W	0x100 至 0x13C	SCT 匹配值寄存器的匹配通道为 0 至 15；从 REGMOD0 至 REGMODE15 等于 0	0x0000 0000	表 657
MATCH0_L 至 MATCH15_L	R/W	0x100 至 0x13C	SCT 匹配值寄存器的匹配通道为 0 至 15；低速计数器 16 位；从 REGMOD0_L 至 REGMODE15_L 等于 0	0x0000 0000	表 657
MATCH0_H 至 MATCH15_H	R/W	0x102 至 0x13E	SCT 匹配值寄存器的匹配通道为 0 至 15；高速计数器 16 位；从 REGMOD0_H 至 REGMODE15_H 等于 0	0x0000 0000	表 657
CAP0 至 CAP15		0x100 至 0x13C	SCT 捕获寄存器的捕获通道为 0 至 15；从 REGMOD0 至 REGMODE15 等于 1	0x0000 0000	表 659

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 638. 寄存器简介：状态可配置定时器 （基址 0x4000 0000）*（续）*

名称	访问类型	地址偏移	描述	复位值	参考
CAP0_L 至 CAP15_L		0x100 至 0x13C	SCT 捕获寄存器的捕获通道为 0 至 15；低速计数器 16 位；从 REGMOD0_L 至 REGMODE15_L 等于 1	0x0000 0000	表 659
CAP0_H 至 CAP15_H		0x102 至 0x13E	SCT 捕获寄存器的捕获通道为 0 至 15；高速计数器 16 位；从 REGMOD0_H 至 REGMODE15_H 等于 1	0x0000 0000	表 659
MATCHREL0 至 MATCHREL15	R/W	0x200 至 0x23C	SCT 匹配重新载入值寄存器 0 至 15；REGMOD0 = 0 至 REGMODE15 = 0	0x0000 0000	表 660
MATCHREL0_L 至 MATCHREL15_L	R/W	0x200 至 0x23C	SCT 匹配重新载入值寄存器 0 至 15；低速计数器 16 位； REGMOD0_L = 0 至 REGMODE15_L = 0	0x0000 0000	表 660
MATCHREL0_H 至 MATCHREL15_H	R/W	0x202 至 0x23E	SCT 匹配重新载入值寄存器 0 至 15；高速计数器 16 位； REGMOD0_H = 0 至 REGMODE15_H = 0	0x0000 0000	表 660
CAPCTRL0 至 CAPCTRL15		0x200 至 0x23C	SCT 捕获控制寄存器 0 至 15； REGMOD0 = 1 至 REGMODE15 = 1	0x0000 0000	表 661
CAPCTRL0_L 至 CAPCTRL15_L		0x200 至 0x23C	SCT 捕获控制寄存器 0 至 15；低速计数器 16 位；REGMOD0_L = 1 至 REGMODE15_L = 1	0x0000 0000	表 661
CAPCTRL0 至 CAPCTRL15		0x202 至 0x23E	SCT 捕获控制寄存器 0 至 15；高速计数器 16 位；REGMOD0 = 1 至 REGMODE15 = 1	0x0000 0000	表 661
EVSTATEMSK0	R/W	0x300	SCT 事件状态寄存器 0	0x0000 0000	表 662
EVCTRL0	R/W	0x304	SCT 事件控制寄存器 0	0x0000 0000	表 663
EVSTATEMSK1	R/W	0x308	SCT 事件状态寄存器 1	0x0000 0000	表 662
EVCTRL1	R/W	0x30C	SCT 事件控制寄存器 1	0x0000 0000	表 663
EVSTATEMSK2	R/W	0x310	SCT 事件状态寄存器 2	0x0000 0000	表 662
EVCTRL2	R/W	0x314	SCT 事件控制寄存器 2	0x0000 0000	表 663
EVSTATEMSK3	R/W	0x318	SCT 事件状态寄存器 3	0x0000 0000	表 662
EVCTRL3	R/W	0x31C	SCT 事件控制寄存器 3	0x0000 0000	表 663
EVSTATEMSK4	R/W	0x320	SCT 事件状态寄存器 4	0x0000 0000	表 662
EVCTRL4	R/W	0x324	SCT 事件控制寄存器 4	0x0000 0000	表 663
EVSTATEMSK5	R/W	0x328	SCT 事件状态寄存器 5	0x0000 0000	表 662
EVCTRL5	R/W	0x32C	SCT 事件控制寄存器 5	0x0000 0000	表 663
EVSTATEMSK6	R/W	0x330	SCT 事件状态寄存器 6	0x0000 0000	表 662
EVCTRL6	R/W	0x334	SCT 事件控制寄存器 6	0x0000 0000	表 663
EVSTATEMSK7	R/W	0x338	SCT 事件状态寄存器 7	0x0000 0000	表 662
EVCTRL7	R/W	0x33C	SCT 事件控制寄存器 7	0x0000 0000	表 663
EVSTATEMSK8	R/W	0x340	SCT 事件状态寄存器 8	0x0000 0000	表 662
EVCTRL8	R/W	0x344	SCT 事件控制寄存器 8	0x0000 0000	表 663
EVSTATEMSK9	R/W	0x348	SCT 事件状态寄存器 9	0x0000 0000	表 662
EVCTRL9	R/W	0x34C	SCT 事件控制寄存器 9	0x0000 0000	表 663
EVSTATEMSK10	R/W	0x350	SCT 事件状态寄存器 10	0x0000 0000	表 662
EVCTRL10	R/W	0x354	SCT 事件控制寄存器 10	0x0000 0000	表 663
EVSTATEMSK11	R/W	0x358	SCT 事件状态寄存器 11	0x0000 0000	表 662
EVCTRL11	R/W	0x35C	SCT 事件控制寄存器 11	0x0000 0000	表 663
EVSTATEMSK12	R/W	0x360	SCT 事件状态寄存器 12	0x0000 0000	表 662

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 638. 寄存器简介：状态可配置定时器 （基址 0x4000 0000）*（续）*

名称	访问类型	地址偏移	描述	复位值	参考
EVCTRL12	R/W	0x364	SCT 事件控制寄存器 12	0x0000 0000	表 663
EVSTATEMSK13	R/W	0x368	SCT 事件状态寄存器 13	0x0000 0000	表 662
EVCTRL13	R/W	0x36C	SCT 事件控制寄存器 13	0x0000 0000	表 663
EVSTATEMSK14	R/W	0x370	SCT 事件状态寄存器 14	0x0000 0000	表 662
EVCTRL14	R/W	0x374	SCT 事件控制寄存器 14	0x0000 0000	表 663
EVSTATEMSK15	R/W	0x378	SCT 事件状态寄存器 15	0x0000 0000	表 662
EVCTRL15	R/W	0x37C	SCT 事件控制寄存器 15	0x0000 0000	表 663
OUTPUTSET0	R/W	0x500	SCT 输出 0 设置寄存器	0x0000 0000	表 664
OUTPUTCL0	R/W	0x504	SCT 输出 0 清除寄存器	0x0000 0000	表 665
OUTPUTSET1	R/W	0x508	SCT 输出 1 设置寄存器	0x0000 0000	表 664
OUTPUTCL1	R/W	0x50C	SCT 输出 1 清除寄存器	0x0000 0000	表 665
OUTPUTSET2	R/W	0x510	SCT 输出 2 设置寄存器	0x0000 0000	表 664
OUTPUTCL2	R/W	0x514	SCT 输出 2 清除寄存器	0x0000 0000	表 665
OUTPUTSET3	R/W	0x518	SCT 输出 3 设置寄存器	0x0000 0000	表 664
OUTPUTCL3	R/W	0x51C	SCT 输出 3 清除寄存器	0x0000 0000	表 665
OUTPUTSET4	R/W	0x520	SCT 输出 4 设置寄存器	0x0000 0000	表 664
OUTPUTCL4	R/W	0x524	SCT 输出 4 清除寄存器	0x0000 0000	表 665
OUTPUTSET5	R/W	0x528	SCT 输出 5 设置寄存器	0x0000 0000	表 664
OUTPUTCL5	R/W	0x52C	SCT 输出 5 清除寄存器	0x0000 0000	表 665
OUTPUTSET6	R/W	0x530	SCT 输出 6 设置寄存器	0x0000 0000	表 664
OUTPUTCL6	R/W	0x534	SCT 输出 6 清除寄存器	0x0000 0000	表 665
OUTPUTSET7	R/W	0x538	SCT 输出 7 设置寄存器	0x0000 0000	表 664
OUTPUTCL7	R/W	0x53C	SCT 输出 7 清除寄存器	0x0000 0000	表 665
OUTPUTSET8	R/W	0x540	SCT 输出 8 设置寄存器	0x0000 0000	表 664
OUTPUTCL8	R/W	0x544	SCT 输出 8 清除寄存器	0x0000 0000	表 665
OUTPUTSET9	R/W	0x548	SCT 输出 9 设置寄存器	0x0000 0000	表 664
OUTPUTCL9	R/W	0x54C	SCT 输出 9 清除寄存器	0x0000 0000	表 665
OUTPUTSET10	R/W	0x550	SCT 输出 10 设置寄存器	0x0000 0000	表 664
OUTPUTCL10	R/W	0x554	SCT 输出 10 清除寄存器	0x0000 0000	表 665
OUTPUTSET11	R/W	0x558	SCT 输出 11 设置寄存器	0x0000 0000	表 664
OUTPUTCL11	R/W	0x55C	SCT 输出 11 清除寄存器	0x0000 0000	表 665
OUTPUTSET12	R/W	0x560	SCT 输出 12 设置寄存器	0x0000 0000	表 664
OUTPUTCL12	R/W	0x564	SCT 输出 12 清除寄存器	0x0000 0000	表 665
OUTPUTSET13	R/W	0x568	SCT 输出 13 设置寄存器	0x0000 0000	表 664
OUTPUTCL13	R/W	0x56C	SCT 输出 13 清除寄存器	0x0000 0000	表 665
OUTPUTSET14	R/W	0x570	SCT 输出 14 设置寄存器	0x0000 0000	表 664
OUTPUTCL14	R/W	0x574	SCT 输出 14 清除寄存器	0x0000 0000	表 665
OUTPUTSET15	R/W	0x578	SCT 输出 15 设置寄存器	0x0000 0000	表 664
OUTPUTCL15	R/W	0x57C	SCT 输出 15 清除寄存器	0x0000 0000	表 665

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

28.6.1 SCT 配置寄存器

该寄存器将会配置 SCT 的整体操作。在写入任何其他寄存器之前写入该寄存器。

表 639. SCT 配置寄存器 （CONFIG - 地址 0x4000 0000）位描述

位	符号	值	描述	复位值
0	UNIFY		SCT 操作	0
		0	SCT 将作为两个名为 L 和 H 的 16 位计数器来使用。	
		1	SCT 将作为统一的 32 位计数器来使用。	
2:1	CLKMODE		SCT 时钟模式	00
		0x0	总线时钟为 SCT 和预分频器计时。	
		0x1	SCT 时钟是总线时钟，但仅在 CKSEL 字段选择的输入的采样找到所选的边沿时才使能预分频器来计数。时钟输入的最小脉冲宽度为 1 个总线时钟周期。该模式是高性能的采样时钟模式。	
		0x2	CKSEL 选定的输入为 SCT 和预分频器计时。该输入同步到总线时钟，也可能反向同步。时钟输入的最小脉冲宽度为 1 个总线时钟周期。该模式是低功耗的采样时钟模式。	
		0x3	CKSEL 字段选定的输入边沿为 SCT 和预分频器计时。在该模式中下列情况是成立的： 大部分 SCT 使用输入 （所选的极性）来计时。 输出会同步切换到输入时钟。 输入时钟速率必须至少为总线时钟速率的一半，并且其速度可以比总线时钟快。	
6:3	CLKSEL		SCT 时钟选择	0000
		0x0	输入 0 的上升沿。	
		0x1	输入 0 的下降沿。	
		0x2	输入 1 的上升沿。	
		0x3	输入 1 的下降沿。	
		0x4	输入 2 的上升沿。	
		0x5	输入 2 的下降沿。	
		0x6	输入 3 的上升沿。	
		0x7	输入 3 的下降沿。	
		0x8	输入 4 的上升沿。	
		0x9	输入 4 的下降沿。	
		0xA	输入 5 的上升沿。	
		0xB	输入 5 的下降沿。	
		0xC	输入 6 的上升沿。	
		0xD	输入 6 的下降沿。	
		0xE	输入 7 的上升沿。	
		0xF	输入 7 的下降沿。	
7	NORELAODL_ NORELOADU	-	该位中的数值 1 会防止较低匹配寄存器从其对应的重新载入寄存器中重新载入。在任何时候都可以编写软件来设置或清除该位。在设置 UNIFY 位后，该位可用于较高和较低寄存器中。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 639. SCT 配置寄存器 （CONFIG - 地址 0x4000 0000）位描述 *（续）*

位	符号	值	描述	复位值
8	NORELOADH	-	该位中的数值 1 会防止较高匹配寄存器从其对应的重新载入寄存器中重新载入。在任何时候都可以编写软件来设置或清除该位。在设置 UNIFY 位时，将不使用该位。	0
16:9	INSYNCn	-	输入 n （位 9 = 输入 0、位 10 = 输入 1... 位 16 = 输入 7）同步。在使用该位创建事件之前，这些位中的任意一个位中的数值 1 会使相应的输入同步到 SCT 时钟中。如果输入同步到 SCT 时钟，请保留其位 0 以获取更快的响应。 当 CKMODE 字段为 1x 时，将不使用该字段中的位 （与 CKSEL 字段所选的输入对应）。	1
31:17	-		保留	-

28.6.2 SCT 控制寄存器

如果 CONFIG 寄存器中的 UNIFY 等于 1，则仅使用 _L 位。

如果 CONFIG 寄存器中的 UNIFY 等于 0，则会将该寄存器作为两个寄存器 CTRL_L（地址 0x4000 4004）和 CTRL_H （地址 0x4000 4006）来写入。 L 和 H 寄存器都可单独执行读或写操作，也可在单一的 32 位读或写操作中读写。

在计数器停止或终止操作时，可以写入该寄存器中的所有位。在计数器运行时，只能写入 STOP 或 HALT 位。（将 HALT 设为 1 后，可以在后续写操作中写入其他位。）

表 640. SCT 控制寄存器 （CTRL - 地址 0x4000 0004）位描述

位	符号	值	描述	复位值
0	DOWN_L	-	当 L 或统一计数器向下计数时，此位为 1。当达到计数器的限值且 BIDIR 为 1 时，此位由硬件设置。当计数器达到 0 时由硬件清除此位。	0
1	STOP_L	-	当该位为 1、HALT 为 0 时，将不运行 L 计数器或统一计数器，但会发生与计数器相关的 I/O 事件。如果该类事件与启动寄存器中的掩码相匹配，则会清除该位并重新开始计数。	0
2	HALT_L	-	当该位为 1 时，将不运行 L 计数器或统一计数器，也不会发生任何事件。通过复位设置此位。 **注：**此位设置后，只能由软件清除此位来恢复计数器操作。	1
3	CLRCTR_L	-	在将 1 写入到该位时会清除 L 计数器或统一计数器。该位始终作为 0 来读取。	0
4	BIDIR_L		L 计数器或统一计数器方向选择。	0
		0	计数器会一直向上计数至其所限范围，然后清零。	
		1	计数器会一直向上计数至其所限范围，然后向下计数至 0。	
12:5	PRE_L	-	指定对 SCT 时钟进行预分频以产生 L 计数器或统一计数器时钟的系数。计数器时钟通过 PRE_L+1 分频的 SCT 时钟速率来计时。 **注：**无论何时更改 PRE 值，都会清除计数器 （通过将 1 写入 CLRCTR 位）。	0
15:13	-		保留	
16	DOWN_H	-	当 H 计数器向下计数时，该位为 1。当达到计数器的限值且 BIDIR 为 1 时，此位由硬件设置。当计数器达到 0 时由硬件清除此位。	0
17	STOP_H	-	当该位为 1、HALT 为 0 时，将不运行 H 计数器，但会发生与计数器相关的 I/O 事件。如果该类事件与启动寄存器中的掩码相匹配，则会清除该位并重新开始计数。	0
18	HALT_H	-	当该位为 1 时，将不运行 H 计数器，也不会发生任何事件。通过复位设置此位。 **注：**此位设置后，只能由软件清除此位来恢复计数器操作。	1
19	CLRCTR_H	-	在将 1 写入到该位时会清除 H 计数器。该位始终作为 0 来读取。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 640. SCT 控制寄存器 （CTRL - 地址 0x4000 0004）位描述 *（续）*

位	符号	值	描述	复位值
20	BIDIR_H		方向选择	0
		0	H 计数器会一直向上计数至其所限范围，然后清零。	
		1	H 计数器会一直向上计数至其所限范围，然后向下计数至 0。	
28:21	PRE_H	-	指定对 SCT 时钟进行预分频以产生 H 计数器时钟的系数。计数器时钟通过 PRELH+1 分频的 SCT 时钟速率来计时。 **注：**无论何时更改 PRE 值，都会清除计数器 （通过将 1 写入 CLRCTR 位）。	0
31:29	-		保留	

28.6.3 SCT 限值寄存器

如果 CONFIG 寄存器中的 UNIFY 等于 1，则仅使用 _L 位。

如果 CONFIG 寄存器中的 UNIFY 等于 0，则会将该寄存器作为两个寄存器 LIMIT_L（地址 0x4000 4008）和 LIMIT_H （地址 0x4000 400A）来写入。 L 和 H 寄存器都可单独执行读或写操作，也可在单一的 32 位读或写操作中读写。

该寄存器中的位会设置将哪个事件作为计数器限值。在发生限值事件时，计数器将在单向模式中清零，或开始在双向模式中向下计数。当计数器计数完毕后，该状态将始终被视为限值事件，并且计数器会在单向模式中清零，或在双向模式中，开始在下一个时钟边缘向下计数，即使已发生 SCT 限值计数器定义的限值事件。

表 641. SCT 限值寄存器 （LIMIT - 地址 0x4000 0008）位描述

位	符号	描述	复位值
15:0	LIMMSK_L	如果位 n 等于 1，则会将事件 n 作为 L 计数器或统一计数器的计数器限值（事件 0 = 位 0、事件 1 = 位 1 及事件 15 = 位 15）。	0
31:16	LIMMSK_H	如果位 n 等于 1，则会将事件 n 作为 H 计数器的计数器限值（事件 0 = 位 16、事件 1 = 位 17 及事件 15 = 位 31）。	0

28.6.4 SCT 终止条件寄存器

如果 CONFIG 寄存器中的 UNIFY 等于 1，则仅使用 _L 位。

如果 CONFIG 寄存器中的 UNIFY 等于 0，则会将该寄存器作为两个寄存器 HALT_L（地址 0x4000 400C）和 HALT_H （地址 0x4000 400E）来写入。 L 和 H 寄存器都可单独执行读或写操作，也可在单一的 32 位读或写操作中读写。

注：终止计数器的任何事件都会禁用其操作，直至软件清除了 CTRL 寄存器 （表 640）中的 HALT 位为止。

表 642. SCT 终止条件寄存器 （HALT - 地址 0x4000 000C）位描述

位	符号	描述	复位值
15:0	HALTMSK_L	如果位 n 等于 1，则事件 n 会在 CTRL 寄存器中设置 HALT_L 位（事件 0 = 位 0、事件 1 = 位 1 及事件 15 = 位 15）。	0
31:16	HALTMSK_H	如果位 n 等于 1，则事件 n 会在 CTRL 寄存器中设置 HALT_H 位（事件 0 = 位 16、事件 1 = 位 17 及事件 15 = 位 31）。	0

28.6.5 SCT 停止条件寄存器

如果 CONFIG 寄存器中的 UNIFY 等于 1，则仅使用 _L 位。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

如果 CONFIG 寄存器中的 UNIFY 等于 0，则会将该寄存器作为两个寄存器 STOPT_L （地址 0x4000 4010）和 STOP_H （地址 0x4000 4012）来写入。 L 和 H 寄存器都可单独执行读或写操作，也可在单一的 32 位读或写操作中读写。

表 643. SCT 停止条件寄存器 （STOP - 地址 0x4000 0010）位描述

位	符号	描述	复位值
15:0	STOPMSK_L	如果位 n 等于 1，则事件 n 会在 CTRL 寄存器中设置 STOP_L 位（事件 0 = 位 0、事件 1 = 位 1 及事件 15 = 位 15）。	0
31:16	STOPMSK_H	如果位 n 等于 1，则事件 n 会在 CTRL 寄存器中设置 STOP_H 位（事件 0 = 位 16、事件 1 = 位 17 及事件 15 = 位 31）。	0

28.6.6 SCT 启动条件寄存器

如果 CONFIG 寄存器中的 UNIFY 等于 1，则仅使用 _L 位。

如果 CONFIG 寄存器中的 UNIFY 等于 0，则会将该寄存器作为两个寄存器 START_L （地址 0x4000 4014）和 START_H （地址 0x4000 4016）来写入。 L 和 H 寄存器都可单独执行读或写操作，也可在单一的 32 位读或写操作中读写。

该寄存器中的位会选择使用哪些事件 （如果有）来清除控制寄存器中的 STOP 位。（由于在 HALT 为 1 时不会发生任何事件，因此软件只能通过写入控制寄存器来清除 HALT 位）。

表 644. SCT 启动条件寄存器 （START - 地址 0x4000 0014）位描述

位	符号	描述	复位值
15:0	STARTMSK_L	如果位 n 等于 1，则事件 n 会清除 CTRL 寄存器中的 STOP_L 位（事件 0 = 位 0、事件 1 = 位 1 及事件 15 = 位 15）。	0
31:16	STARTMSK_H	如果位 n 等于 1，则事件 n 会在 CTRL 寄存器中清除 STOP_H 位 （事件 0 = 位 16、事件 1 = 位 17 及事件 15 = 位 31）。	0

28.6.7 SCT 计数器寄存器

如果 CONFIG 寄存器中的 UNIFY 等于 1，则计数器将为统一的 32 位寄存器，并且会同时使用 _L 和 _H 位。

如果 CONFIG 寄存器中的 UNIFY 等于 0，则会将该寄存器作为两个寄存器 COUNT_L（地址 0x4000 4040）和 COUNT_H （地址 0x4000 4042）来写入。 L 和 H 寄存器都可单独执行读或写操作，也可在单一的 32 位读或写操作中读写。在本例中，L 寄存器和 H 寄存器会在其他寄存器的控制下进行单独计数。
在计数器运行时尝试写入计数器，将不会影响计数器，但会产生总线错误。软件可以随时读取计数器寄存器。

表 645. SCT 计数器寄存器 （COUNT - 地址 0x4000 0040）位描述

位	符号	描述	复位值
15:0	CTR_L	当 UNIFY = 0 时，读取或写入 16 位 L 计数器值。当 UNIFY = 1 时，读取或写入 32 位统一计数器的较低 16 位。	0
31:16	CTR_H	当 UNIFY = 0 时，读取或写入 16 位 H 计数器值。当 UNIFY = 1 时，读取或写入 32 位统一计数器的较高 16 位。	0

28.6.8 SCT 状态寄存器

如果 CONFIG 寄存器中的 UNIFY 等于 1，则仅使用 _L 位。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

如果 CONFIG 寄存器中的 UNIFY 等于 0，则会将该寄存器作为两个寄存器 STATE_L （地址 0x4000 4044）和 STATE_H（地址 0x4000 4046）来写入。L 和 H 寄存器都可单独执行读或写操作，也可在单一的 32 位读或写操作中读写。

软件可以随时读取与计数器相关的状态。当计数器的 HALT 位为 1 时才允许写入该状态；当 HALT 为 0 时，写入尝试将不会更改该状态，但会导致总线错误。

状态变量是区分 SCT 与其他计数器 / 定时器 /PWM 模块的主要方式。仅在特定状态下才产生事件。而事件可以执行下列操作：

- 设置和清除输出
- 限制、停止和启动计数器
- 原因中断和 DMA 请求
- 修改状态变量

状态变量值完全在应用程序的控制之下。如果应用程序不使用状态，则状态变量值将保留为零，这是系统默认值。

使用状态变量可以跟踪和控制任何所需操作序列中的相关计数器的多个周期。状态变量在逻辑上与代表 SCT 配置的状态机图相关。有关状态与事件之间关系的更多信息，参见第 28.6.23 节和 28.6.24。

状态变量的所有可能值均由包含所有定义事件的事件控制寄存器中的 STATELD/STADEV 字段设置。在多个计数器周期期间，状态变量的更改反映了相关状态机是如何从一种状态变为另一种状态的。

表 646. SCT 状态寄存器 （STATE - 地址 0x4000 0044）位描述

位	符号	描述	复位值
4:0	STATE_L	状态变量。	0
15:5	-	保留。	-
20:16	STATE_H	状态变量。	0
31:21	-	保留。	

28.6.9 SCT 输入寄存器

软件可以在只读寄存器中以稍微不同的两种格式来读取 SCT 输入的状态。如果要获得完全不同的值，则唯一的条件是 CONFIG 寄存器中的 CLKMODE 等于 2。

表 647. SCT 输入寄存器 （INPUT - 地址 0x4000 0048）位描述

位	符号	描述	复位值
0	AIN0	输入 0 的实时状态。	引脚
1	AIN1	输入 1 的实时状态。	引脚
2	AIN2	输入 2 的实时状态。	引脚
3	AIN3	输入 3 的实时状态。	引脚
4	AIN4	输入 4 的实时状态。	引脚
5	AIN5	输入 5 的实时状态。	引脚
6	AIN6	输入 6 的实时状态。	引脚
7	AIN7	输入 7 的实时状态。	引脚
15:8	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 647. SCT 输入寄存器 （INPUT - 地址 0x4000 0048）位描述 *（续）*

位	符号	描述	复位值
16	SIN0	同步到 SCT 时钟的输入 0 状态。	-
17	SIN1	同步到 SCT 时钟的输入 1 状态。	-
18	SIN2	同步到 SCT 时钟的输入 2 状态。	-
19	SIN3	同步到 SCT 时钟的输入 3 状态。	-
20	SIN4	同步到 SCT 时钟的输入 4 状态。	-
21	SIN5	同步到 SCT 时钟的输入 5 状态。	-
22	SIN6	同步到 SCT 时钟的输入 6 状态。	-
23	SIN7	同步到 SCT 时钟的输入 7 状态。	-
31:24	-	保留	-

28.6.10 SCT 匹配 / 捕获寄存器模式寄存器

如果 CONFIG 寄存器中的 UNIFY 等于 1，则仅使用该寄存器中的 _L 位。 L 位控制是否将每组匹配 / 捕获寄存器作为统一的 32 位匹配 / 捕获寄存器。

如果 CONFIG 寄存器中的 UNIFY 等于 0，则会将该寄存器作为两个寄存器 REGMODE_L（地址 0x4000 404C）和 REGMODE_H （地址 0x4000 404E）来写入。 L 和 H 寄存器都可单独执行读或写操作，也可在单一的 32 位读或写操作中读写。 _L 位 / 寄存器用于控制 L 匹配 / 捕获寄存器，而 _H 位 / 寄存器用于控制 H 匹配 / 捕获寄存器。

SCT 包含了 16 个匹配 / 捕获寄存器对。寄存器模式寄存器用于选择将每个寄存器对作为匹配寄存器 （参见第 28.6.19 节）还是捕获寄存器 （参见第 28.6.20 节）。每个匹配 / 捕获寄存器都有一个随附寄存器，在将寄存器用作匹配寄存器 （第 28.6.21 节）时被用作重新载入寄存器，或者在将寄存器用作捕获寄存器（第 28.6.22 节）时被用作捕获控制寄存器。仅在 UNIFY 位为 0 时才会使用 REGMODE_H。

所有匹配 / 捕获和重新载入 / 捕获控制寄存器均可使用交替寻址模式，从而在 UNIFY 等于 0 时可以对半字寄存器执行 DMA 访问。该模式如第 28.7.9 节所述。

表 648. SCT 匹配 / 捕获寄存器模式寄存器 （REGMODE - 地址 0x4000 004C）位描述

位	符号	描述	复位值
15:0	REGMOD_L	每个位控制一对匹配 / 捕获寄存器 （寄存器 0 = 位 0、寄存器 1 = 位 1、 ...、寄存器 15 = 位 15）。 0 表示寄存器作为匹配寄存器来使用。 1 表示寄存器作为捕获寄存器来使用。	0
31:16	REGMOD_H	每个位控制一对匹配 / 捕获寄存器 （寄存器 0 = 位 16、寄存器 1 = 位 17、 ...、寄存器 15 = 位 31）。 0 表示寄存器作为匹配寄存器来使用。 1 表示寄存器作为捕获寄存器来使用。	0

28.6.11 SCT 输出寄存器

SCT 支持 16 个输出，其中每个输出在该寄存器中都有相应的位。
当终止两个计数器来直接控制输出时，软件可以对任何一个输出寄存器执行写入操作。在任何一个计数器停止或运行时，对该寄存器执行写入操作将不会影响输出，但会导致总线错误。

软件可以随时对该寄存器执行读取操作来了解输出的状态。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 649. SCT 输出寄存器 （OUTPUT - 地址 0x4000 0050）位描述

位	符号	描述	复位值
15:0	OUT	将 1 写入到位 n 中，会使相应的输出处于高电平状态。写入 0，会使相应的输出处于低电平状态（输出 0= 位 0、输出 1= 位 1、...、输出 15= 位 15）。	0
31:16	-	保留	

28.6.12 SCT 双向输出控制寄存器

此寄存器用于指定（为每个输出）计数方向对输出的设置和清除操作含义（参见第 28.6.25 节和第 28.6.26 节）的影响。

表 650. SCT 双向输出控制寄存器 （OUTPUTDIRCTRL - 地址 0x4000 0054）位描述

位	符号	值	描述	复位值
1:0	SETCLR0		输出 0 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	
3:2	SETCLR1		输出 1 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	
5:4	SETCLR2		输出 2 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	
7:6	SETCLR3		输出 3 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	
9:8	SETCLR4		输出 4 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	
11:10	SETCLR5		输出 5 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 650. SCT 双向输出控制寄存器 （OUTPUTDIRCTRL - 地址 0x4000 0054）位描述 *（续）*

位	符号	值	描述	复位值
13:12	SETCLR6		输出 6 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	
15:14	SETCLR7		输出 7 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	
17:16	SETCLR8		输出 8 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	
19:18	SETCLR9		输出 9 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	
21:20	SETCLR10		输出 5 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	
23:22	SETCLR11		输出 11 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	
25:24	SETCLR12		输出 12 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	
27:26	SETCLR13		输出 13 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 650. SCT 双向输出控制寄存器 （OUTPUTDIRCTRL - 地址 0x4000 0054）位描述 *（续）*

位	符号	值	描述	复位值
29:28	SETCLR14		输出 14 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	
31:30	SETCLR15		输出 15 的设置 / 清除操作。值 0x3 是保留值。请不要编写该值。	0
		0x0	设置操作和清除操作与任何计数器无关。	
		0x1	当计数器 L 或统一计数器向下计数时，会反向执行设置操作和清除操作。	
		0x2	当计数器 H 向下计数时，会反向执行设置操作和清除操作。如果 UNIFY = 1，将不使用该位。	

28.6.13 SCT 解决冲突寄存器

使用寄存器 OUTPUTSETn （第 28.6.25 节）和 OUTPUTCLn （第 28.6.26 节）可以对相同时钟周期中的输出（甚至是相同事件）执行设置和清除操作。该 SCT 解决冲突寄存器可解决该冲突。

如果要通过事件来切换输出，则在该寄存器中将 OnRES 值设置为 0x3，并在设置寄存器和清除寄存器中设置事件位。

表 651. SCT 冲突解决寄存器 （RES - 地址 0x4000 0058）位描述

位	符号	值	描述	复位值
1:0	O0RES		输出 0 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR0 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR0 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	
3:2	O1RES		输出 1 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR1 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR1 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	
5:4	O2RES		输出 2 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR2 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR2 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	
7:6	O3RES		输出 3 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR3 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR3 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 651. SCT 冲突解决寄存器 （RES - 地址 0x4000 0058）位描述 *（续）*

位	符号	值	描述	复位值
9:8	O4RES		输出 4 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR4 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR4 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	
11:10	O5RES		输出 5 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR5 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR5 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	
13:12	O6RES		输出 6 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR6 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR6 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	
15:14	O7RES		输出 7 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR7 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR7 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	
17:16	O8RES		输出 8 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR8 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR8 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	
19:18	O9RES		输出 9 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR9 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR9 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	
21:20	O10RES		输出 10 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR10 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR10 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	
23:22	O11RES		输出 11 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR11 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR11 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 651. SCT 冲突解决寄存器 （RES - 地址 0x4000 0058）位描述 *（续）*

位	符号	值	描述	复位值
25:24	O12RES		输出 12 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR12 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR12 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	
27:26	O13RES		输出 13 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR13 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR13 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	
29:28	O14RES		输出 14 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR14 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR14 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	
31:30	O15RES		输出 15 的同步设置和清除效应。	0
		0x0	无变化。	
		0x1	基于 SETCLR15 字段的设置输出 （或清除输出）。	
		0x2	基于 SETCLR15 字段的清除输出 （或设置输出）。	
		0x3	切换输出。	

28.6.14 SCT DMA 请求 0 和请求 1 寄存器

SCT 包含了两个 DMA 请求输出。这些寄存器会在发生特定事件，或计数器的匹配寄存器从其重新载入寄存器中加载时启动 DMA 请求。

表 652. SCT DMA 0 请求寄存器 （DMAREQ0 - 地址 0x4000 005C）位描述

位	符号	描述	复位值
15:0	DEV_0	如果位 n 等于 0，则事件 n 会设置 DMA 请求 1 （事件 0 = 位 0、事件 1 = 位 1、 ...、事件 15 = 位 15）。	0
29:16	-	保留	-
30	DRL0	当此位为 1 时， SCT 在从 “重新载入 _L/ 统一” 寄存器中加载 “匹配 _L/ 统一” 寄存器时，设置 DMA 请求 0。	
31	DRQ0	该只读位表示 DMA 请求 0 的状态。	

表 653. SCT DMA 1 请求寄存器 （DMAREQ1 - 地址 0x4000 0060）位描述

位	符号	描述	复位值
15:0	DEV_1	如果位 n 等于 1，则事件 n 会设置 DMA 请求 1 （事件 0 = 位 0、事件 1 = 位 1、 ...、事件 15 = 位 15）。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 653. SCT DMA 1 请求寄存器 （DMAREQ1 - 地址 0x4000 0060）位描述 *（续）*

位	符号	描述	复位值
29:16	-	保留	-
30	DRL1	如果该位为 1，则 SCT 从重新载入 L 寄存器 / 统一寄存器中加载匹配 L 寄存器 / 统一寄存器时，会设置 DMA 请求 1。	
31	DRQ1	该只读位表示 DMA 请求 1 的状态。	

28.6.15 SCT 标志使能寄存器

如果还在 SCT 事件标志寄存器 （第 28.6.16 节）中设置了 FLAGn 位，则此寄存器使能请求中断的标志。

表 654. SCT 标志使能寄存器 （EVEN - 地址 0x4000 00F0）位描述

位	符号	描述	复位值
15:0	IEN	当该寄存器和事件标志寄存器的位 n 均等于 1 时， SCT 会请求中断 （事件 0 = 位 0、事件 1 = 位 1、 ...、事件 15 = 位 15）。	0
31:16	-	保留	

28.6.16 SCT 事件标志寄存器

该寄存器会记录各种事件。在将 1 写入到该寄存器时，会清除相应的标志，且当所有使能的标志位为零时，会取消 SCT 中断请求。

表 655. SCT 事件标志寄存器 （EVFLAG - 地址 0x4000 00F4）位描述

位	符号	描述	复位值
15:0	FLAG	在进行复位后或将 1 最后写入到该位后，如果发生了事件 n，则位 n 为 1 （事件 0 = 位 0、事件 1 = 位 1、 ...、事件 15 = 位 15）。	0
31:16	-	保留	-

28.6.17 SCT 冲突使能寄存器

该寄存器能够使 SCT 冲突解决寄存器中所指定的 “无变化冲突” 事件进行 IRQ 请求。

表 656. SCT 冲突使能寄存器 （CONEN - 地址 0x4000 00F8）位描述

位	符号	描述	复位值
15:0	NCEN	当该寄存器和 SCT 冲突标志寄存器的位 n 均等于 1 时， SCT 会请求中断 （输出 0= 位 0、输出 1= 位 1、 ...、输出 15= 位 15）。	0
31:16	-	保留	

28.6.18 SCT 冲突标志寄存器

该寄存器会记录中断使能的无变化冲突事件，并提供详细的总线错误信息。将 1 写入 NCFLAG位会清除相应的读取位，且当所有已使能标志位为零时，将取消SCT的中断请求。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 657. SCT 冲突标志寄存器 （CONFLAG - 地址 0x4000 00FC）位描述

位	符号	描述	复位值
15:0	NCFLAG	在进行复位后或将 1 最后写入到该位后，如果输出 n 发生了无变化冲突事件，则位 n 为 1 （输出 0= 位 0、输出 1= 位 1、 ...、输出 15= 位 15）。	0
29:16	-	保留。	-
30	BUSERRL	该 SCT 中出现的最新总线错误，与在 L/U 计数器未停止运行时对 CTR L/ 统一 （CTR L/Unified）寄存器、状态 L/ 统一 （STATE L/Unified）寄存器、匹配 L/ 统一 （MATCH L/Unified）寄存器或输出（Output）寄存器执行写入操作有关。对特定 L 和 H 寄存器执行字写入操作，其成功概率只占一半。	0
31	BUSERRH	该 SCT 中所出现的最新总线错误，与在 H 计数器未停止运行时对 CTR H、STATE H、MATCH H 或输出寄存器执行写入操作有关。	0

28.6.19 SCT 匹配寄存器 0 至 15 （REGMODEn 位 =0）

匹配寄存器与计数器进行比较来帮助创建事件。当 UNIFY 位为 0 时， L 寄存器和 H 寄存器将分别与 L 计数器和 H 计数器进行比较。当 UNIFY 为 1 时， L 寄存器和 H 寄存器将保留 32 位值与统一计数器进行比较。匹配事件仅发生在计数器运行 （STOP 和 HALT 均为 0）所在的时钟内。

在任何时候都可读取匹配寄存器。在相关计数器运行时对匹配寄存器执行写操作不会影响匹配寄存器，但会造成总线错误。匹配事件发生在 SCT 时钟内，在该时钟内，计数器的计数递增至下一个值 （或将递增至下一个值）。当匹配事件限制了其计数器 （如第 28.6.3 节所述）时，匹配寄存器中的值在其清零之前将为该计数器的最后一个值 （或，在 BIDIR 为 1 时递减）。

无论是重新载入寄存器，还是匹配寄存器，都不能采用“透写”机制。在启动计数器之前，软件可以将值 1 写入到在计数器第一个周期中所使用的匹配寄存器中，并且将另一个值写入到在计数器第二个周期中所使用的对应匹配重新载入寄存器中。

表 658. SCT 匹配寄存器 0 至 15 （MATCH – 地址 0x4000 0100 (MATCH0) 至 0x4000 4013C (MATCH15)）位描述 （REGMODEn 位 = 0）

位	符号	描述	复位值
15:0	MATCHn_L	当 UNIFY = 0 时，将读取或写入与 L 计数器进行比较的 16 位值。当 UNIFY = 1 时，将读取或写入与统一计数器进行比较的 32 位值的较低 16 位。	0
31:16	MATCHn_H	当 UNIFY = 0 时，将读取或写入与 H 计数器进行比较的 16 位值。当 UNIFY = 1 时，将读取或写入与统一计数器进行比较的 32 位值的较高 16 位。	0

28.6.20 SCT 捕获寄存器 0 至 15 （REGMODEn 位 = 1）

这些寄存器允许软件读取相应捕获控制寄存器选取的事件发生时的计数器值。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 659. SCT 捕获寄存器 0 至 15（CAP - 地址 0x4000 0100 (CAP0) 至 0x4000 013C (CAP15)）位描述 （REGMODEn 位 = 1）

位	符号	描述	复位值
15:0	CAPn_L	当 UNIFY = 0 时，将读取最后一次捕获该寄存器时的 16 位计数器值。当 UNIFY = 1 时，将读取最后一次捕获该寄存器时的 32 位值的较低 16 位。	0
31:16	CAPn_H	当 UNIFY = 0 时，将读取最后一次捕获该寄存器时的 16 位计数器值。当 UNIFY = 1 时，将读取最后一次捕获该寄存器时的 32 位值的较高 16 位。	0

28.6.21 SCT 匹配重新载入寄存器 0 至 15 （REGMODEn 位 = 0）

当 BIDIR 为 0 且计数器的值到达其限值范围，或 BIDIR 为 1 且计数器的值到达 0 时，将从相应的重新载入寄存器中加载匹配寄存器 （L 寄存器、 H 寄存器或统一 32 位寄存器）。

表 660. SCT 匹配重新载入寄存器 0 至 15（MATCHREL – 地址 0x4000 0200 (MATCHRELOAD0) 至 0x4000 023C (MATCHRELOAD15)）位描述 （REGMODEn 位 = 0）

位	符号	描述	复位值
15:0	RELOADn_L	当 UNIFY = 0 时，将读取或写入加载到 SCTMATCHn_L 寄存器中的 16 位值。当 UNIFY = 1 时，将读取或写入加载到 MATCHn 寄存器中的 32 位值的较低 16 位。	0
31:16	RELOADn_H	当 UNIFY = 0 时，将读取或写入加载到 MATCHn_H 寄存器中的 16 位值。当 UNIFY = 1 时，将读取或写入加载到 MATCHn 寄存器中的 32 位值的较高 16 位。	0

28.6.22 SCT 捕获控制寄存器 0 至 15 （REGMODEn 位 = 1）

如果 CONFIG 寄存器中的 UNIFY 等于 1，则仅使用 _L 位。

如果 CONFIG 寄存器中的 UNIFY 等于 0，则该寄存器会作为两个寄存器 CAPCTRLn_L（地址从 0x4000 4100 至 0x4000 413C）和 CAPCTRLn_H （地址从 0x4000 4102 至 0x4000 413E）来写入。L 和 H 寄存器都可单独执行读或写操作，也可在单一的 32 位读或写操作中读写。

每个捕获控制寄存器（L 寄存器、H 寄存器或统一 32 位寄存器）将控制使用哪个事件从计数器中加载相应的捕获寄存器。

表 661. SCT 捕获控制寄存器 0 至 15 （CAPCTRL – 地址 0x4000 0200 (CAPCTRL0) 至 0x4000 023C (CAPCTRL15)）位描述 （REGMODEn 位 = 1）

位	符号	描述	复位值
15:0	CAPCONn_L	如果位 m 等于 1，则事件 m 会加载 CAPn_L (UNIFY = 0) 或 CAPn (UNIFY = 1) 寄存器（事件 0 = 位 0、事件 1 = 位 1、...、事件 15 = 位 15）。	0
31:16	CAPCONn_H	如果位 m 等于 1，则事件 m 会加载 CAPn_H (UNIFY = 0) 寄存器（事件 0 = 位 16、事件 1 = 位 17、 ...、事件 15 = 位 31）。	0

28.6.23 SCT 事件状态掩码寄存器 0 至 15

每个事件都有一个相关的 SCT 事件状态掩码寄存器，它允许该事件发生在 HEVENT 位（位于相应的 EVCTRLn 寄存器中）所选的计数器的一个或多个状态下。

当其 EVSTATEMSKn 寄存器始终为零时，会禁用事件 n，因为无论当前为何种状态事件 n 均会被掩码。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

在不使用状态的简单应用中，将 0x01 写入此寄存器可使能事件。由于该状态值始终为其复位值 0，因此写入 0x01 可以长期有效地通过状态来使能此事件。

表 662. SCT 事件状态掩码寄存器 0 至 15 （EVSTATEMSK - 地址 0x4000 0300 (EVSTATEMSK0) 至 0x4000 0378 (EVSTATEMSK15)）位描述

位	符号	描述	复位值
31:0	STATEMSKn	如果位 m 等于 1，则事件 n （n= 0 至 15）将发生在 HEVENT 位所选的计数器的 m 状态 （m = 状态号；状态 0 = 位 0，状态 1 = 位 1， ...，状态 31 = 位 31）。	0

28.6.24 SCT 事件控制寄存器 0 至 15

该寄存器定义的是事件 n 发生的条件，而并非状态掩码寄存器所定义的状态变量。大部分事件都与特定的计数器 （高速计数器、低速计数器或统一计数器）有关，因此，事件将取决于是否与该寄存器相匹配。事件的其他组成部分可能为所选定的输入信号或输出信号。

当 UNIFY 位为 0 时，每个事件都与其事件控制寄存器中 HEVENT 位所选的特定计数器有关。当事件相关的计数器停止运行或当前的状态无法启动其事件掩码寄存器中所指定的事件时，则不会发生事件。当其事件状态掩码寄存器始终为 0 时，将会永久禁用事件。

根据所选的输入 （输出）边缘或输入 （输出）电平以及 / 或者根据与所选的匹配寄存器相匹配的计数器值，可对已使能的事件编写来使其发生。

每个事件都可以修改其计数器的 STATE 值。如果在指定的时钟周期内发生了与相同计数器相关的多个事件，则仅会发生为最大编号事件指定的状态更改。任何同步发生的事件所指示的其他操作也会发生。

表 663. SCT 事件控制寄存器 0 至 15 （EVCTRL - 地址 0x4000 0304 (EVCTRL0) 至 0x4000 037C (EVCTRL15)）位描述

位	符号	值	描述	复位值
3:0	MATCHSEL	-	选择与该事件 （如果存在）相关的匹配寄存器。仅在 HEVENT 位所选的计数器运行时，才会发生匹配事件。	0
4	HEVENT		选择 L/H 计数器。如果 UNIFY = 1，将不设置该位。	0
		0	选择由 MATCHSEL 所选的 L 状态寄存器和 L 匹配寄存器。	
		1	选择由 MATCHSEL 所选的 H 状态寄存器和 H 匹配寄存器。	
5	OUTSEL		输入 / 输出选择	0
		0	选择 IOSEL 所选的输出。	
		1	选择 IOSEL 所选的输入。	
9:6	IOSEL	-	选择与该事件 （如果存在）相关的输入或输出信号。如果 CKMODE 为 1x，请不要在此寄存器中选择输入。在这种情况下，时钟输入是每个事件的隐式部分。	0
11:10	IOCOND		选择事件 n 的 I/O 条件。（对输出边沿的检测操作会使切换输出的条件延迟一个 SCT 时钟）。要确保进行正确的边沿 / 状态检测，输入必须具有至少一个 SCT 时钟周期的最小脉冲宽度。	0
		0x0	低	
		0x1	上升	
		0x2	下降	
		0x3	高	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 663. SCT 事件控制寄存器 0 至 15（EVCTRL - 地址 0x4000 0304 (EVCTRL0) 至 0x4000 037C (EVCTRL15)）位描述*（续）*

位	符号	值	描述	复位值
13:12	COMBMODE		选择指定的匹配和 I/O 条件的使用方法和合并方法。	
		0x0	或条件。无论是出现指定的匹配，还是出现 I/O 条件，都将发生该事件。	
		0x1	匹配条件。仅使用指定的匹配条件。	
		0x2	IO。仅使用指定的 I/O 条件。	
		0x3	与条件。在同时出现指定的匹配与 I/O 条件时将发生该事件。	
14	STATELD		如果该事件是在该状态下所发生的最大编号事件，则该位将会控制如何使用STATEV值来修改 HEVENT 所选的状态。	
		0	STATEV 值将添加到 STATE 中 （将忽略进位输出）。	
		1	STATEV 值将加载到 STATE 中。	
19:15	STATEV		如果该事件是在该状态下所发生的最大编号事件，则该值将加载或添加到 HEVENT 所选的状态中，具体取决于 STATELD。如果 STATELD 和 STATEV 均为零，则 STATE 值不会发生改变。	
31:20	-		保留	

28.6.25 SCT 输出设置寄存器 0 至 15

每个输出 n 都有一个设置寄存器，可控制事件对每个输出的影响方式。对输出执行设置操作还是清除操作，要取决于 SCTOUTPUTDIRCTRL 寄存器中的 SETCLRn 字段设置。

表 664. SCT 输出设置寄存器 0 至 15 （OUTPUTSET – 地址 0x4000 0500 (OUTPUTSET0) 至 0x4000 0578 (OUTPUTSET15)）位描述

位	符号	描述	复位值
15:0	设置	当位 m 为 1 时，会选择事件 m 来设置输出 n，而在 SETCLRn = 0x1 或 0x2 时，会清除输出 n （事件 0= 位 0、事件 1= 位 1、 ...、事件 15= 位 15）。	0
31:16	-	保留	

28.6.26 SCT 输出清除寄存器 0 至 15

每个输出 n 都有一个清除寄存器，可控制事件对每个输出的影响方式。对输出执行设置操作还是清除操作，要取决于 OUTPUTDIRCTRL 寄存器中的 SETCLRn 字段设置。

表 665. SCT 输出清除寄存器 0 至 15（OUTPUTCL - 地址 0x4000 0504 (OUTPUTCL0) 至 0x4000 057C (OUTPUTCL15)）位描述

位	符号	描述	复位值
15:0	CLR	当位 m 为 1 时，会选择事件 m 来清除输出 n，而在 SETCLRn = 0x1 或 0x2 时，会设置输出 n （事件 0= 位 0、事件 1= 位 1、 ...、事件 15= 位 15）。	0
31:16	-	保留	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

28.7 功能说明

28.7.1 匹配逻辑

图 84. 匹配逻辑

28.7.2 捕获逻辑

图 85. 捕获逻辑

28.7.3 事件选择

状态变量可控制跨计数器多个周期的 SCT。计数器匹配、输入 / 输出边沿和状态值将组合为一组通用的事件，可以切换输出、请求中断并更改状态值。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 86. 事件选择

28.7.4 输出生成

图 87 显示了 SCT 的一个输出限幅。

图 87. 输出限幅 i

28.7.5 中断生成

SCT 会为 NVIC 生成一个中断。

图 88. SCT 中断生成

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

28.7.6 清除预分频器

当控制寄存器中的非零 PRE 字段使能了预分频器后，该预分频器将作为计数器的时钟分频器，就像计数器值的小数部分。当计数器因下列任何原因被清除或加载时，则会清除预分频器：

- 硬件复位
- 软件对计数器寄存器执行写入操作
- 软件将 1 写入到控制器寄存器的 CLRCTR 位中
- BIDIR 等于 0 时，计数器的限值寄存器中的 1 选择了事件

当 BIDIR 为 0 时，I/O 信号所产生的限值事件会清除非零的预分频器。但匹配产生的限值事件仅会在一种特殊情况下清除非零预分频器，如第 28.7.7 节所述。

当 BIDIR 为 1 时，限值事件将不会清除预分频器，而是会清除控制寄存器中的 DOWN 位，并且如果在该时钟内使能了计数器，则会在相同的时钟上使计数器的计数递减。

28.7.7 匹配事件与 I/O 事件

预分频器以及 SCT 时钟为总线时钟的时钟模式 01 会使计数器操作复杂化，但仅在时钟输入中检测到所选的边沿时会使能预分频器和计数器来计数。

- 如果时钟模式非 01 或检测到 CLKSEL 字段所选的输入边缘时，会使能预分频器。
- 在使能预分频器，以及 PRELIM=0 或预分频器等于 PRELIM 中的值时，将会使能计数器。

当其计数器的 HALT 位为 0 时，会在任何 SCT 时钟内出现事件的 I/O 组件。在一般情况下，当其计数器的 HALT 和 STOP 位均为 0 且计数器已使能时，在 UT 时钟内仅会出现事件的匹配组件。

表 666 显示了各种事件在何种情况下发生。

表 666. 事件条件

COMBMODE	IOMODE	在时钟内发生事件：
IO	任意	当 HALT = 0 （类型 A）时会发生事件。
MATCH	任意	当 HALT = 0、STOP = 0 且使能了计数器（类型 C）时会发生事件。
OR	任意	就 IO 组件而言：当 HALT = 0 （类型 A）时会发生事件。 就匹配组件而言：当 HALT = 0、STOP = 0 且使能了计数器（类型 C）时会发生事件。
AND	低或高	当 HALT = 0、STOP = 0 且使能了计数器（类型 C）时会发生事件。
AND	上升或下降	当 HALT = 0 （类型 A）时会发生事件。

28.7.8 DMA 操作

通常在计数器周期开始时，可以使用 DMA 控制器对一个或多个重新载入寄存器执行写入操作或对一个或多个捕获寄存器执行读取操作。如果要对多个重新载入寄存器或捕获寄存器执行 DMA 存取，则这些寄存器应当为地址连续的寄存器。（SCT 中的任何其他设置将不会约束这些寄存器的分配和使用方式。）

如第 28.6.14 节所述，可以通过事件来设置 DMA 请求，或设置计数器匹配寄存器何时从其重新载入寄存器中加载。当 UNIFY 为 0 时，可以使用 SCT 的两个请求对两个计数器执行相同的寄存器存取，或可以使用一个请求对重新载入寄存器执行写操作，而使用另一个请求读取捕获寄存器。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

由于 SCT 不知道为每个请求完成了多少次寄存器传输，因此它无法相应控制其 DMA 请求。

这两个 DMA 请求将分别连接至 DMABREQ7 和 DMABREQ8。在请求所连接的 DMA 通道的通道控制寄存器的 TransferSize 字段中，写入要为每个请求传输的寄存器数。如果使用链接列表功能，则在每个链接列表项中会有一个 TransferSize 值。在完成这些传输之后，GPDMA 会使 DMACCLR 信号变为有效，使 SCT 清除该请求。

28.7.9 匹配 / 捕获寄存器的替换地址

匹配寄存器、重新载入寄存器、捕获寄存器和捕获控制寄存器将作为连续的字来排列，并且会将每个字标准分隔为两个半字。当 UNIFY 位为零时，这两个半字将与 L 计数器和 H 计数器相关联。软件可以最初写入整个字以同时设置 SCT 的两个半部分，也可以写入半个字单独设置这两个半部分。
应用程序可以使用 DMA 控制器对重新载入寄存器执行写入操作或对捕获寄存器执行读取操作。但是，当 UNIFY 为 0 时，半字寄存器的寻址与许多 DMA 控制器使用连续地址进行序列地址操作的要求不相符。表 667 显示了每种类型的寄存器所占范围的第二部分，是如何包含用于半字存取同一寄存器（与使用连续地址操作的 DMA 控制器的要求相符）的替换地址映射的。当 UNIFY 为 1 时，使用标准偏移来完成 DMA 字存取。

表 667. DMA 半字存取的替换地址映射

匹配寄存器	捕获寄存器	标准偏移	DMA 半字偏移
MATCH0_L	CAP0_L	0x100	0x180
MATCH0_H	CAP0_H	0x102	0x1C0
MATCH1_L	CAP1_L	0x104	0x182
MATCH1_H	CAP1_H	0x106	0x1C2
...	...	...	...
MATCHREL0_L	CAPCTRL0_L	0x200	0x280
MATCHREL0_H	CAPCTRL0_H	0x202	0x2C0
MATCHREL1_L	CAPCTRL1_L	0x204	0x282
MATCHREL1_H	CAPCTRL1_H	0x206	0x2C2
...	...	...	...

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

28.7.10 SCT 操作

在其最简单的、单一状态配置中， SCT 用作事件控制的单向或双向计数器。可以将事件配置为计数器匹配事件、输入或输出电平、输入或输出引脚的跃迁或者匹配与输入 / 输出行为的组合。在响应事件时，SCT 输出可以跃迁或 SCT 可以执行其他操作，如创建中断或者起动、停止或复位计数器。每个事件都可以包含多个同步操作。并且，任意多个事件可以启动 SCT 的一个特定操作。

SCT 的一个或任意多个操作唯一性定义一个事件。状态是通过启动计数器任意阶段中的某个 SCT 操作或多个操作所使能的事件来定义的。系统会忽略该状态未选择的事件。

在多状态配置中，状态会随着事件而发生更改。状态更改是在事件发生时 SCT 可以执行的另一种操作。在将事件配置为更改状态时，新的状态会定义一组用于产生其他 SCT 操作的新事件。在计数器的多个周期内，事件可以多次更改状态，并因此创建 SCT 输出和 / 或中断的大量事件控制的跃迁。

在完成配置后， SCT 可以在无软件介入的情况下连续运行，并且可以生成多个完全受事件所控制的输出模式。

- 要配置 SCT，参见第 28.7.10.1 节。
- 如果要对 SCT 执行启动、运行和停止操作，请参见第 28.7.10.2 节。
- 要将 SCT 配置为简单的事件控制计数器 / 定时器，参见第 28.7.10.3 节。

28.7.10.1 配置 SCT

要为多个事件和多个状态设置 SCT，请执行下列配置步骤：

28.7.10.1.1 配置计数器

1. 在 CONFIG 寄存器中配置 L 计数器和 H 计数器，方法是在 UNIFY 字段中选择两个独立的 16 位计数器 （L 计数器和 H 计数器）或一个组合的 32 位计数器。
2. 从任意输入或内部时钟中为 CONFIG 寄存器（CLKMODE 字段和 CLKSEL 字段）选择 SCT 时钟源。

28.7.10.1.2 配置匹配寄存器和捕获寄存器

1. 选择应用程序使用多少个匹配寄存器和捕获寄存器 （总共最多 16 个）：
   - 在 REGMODE 寄存器中，针对 16 个匹配 / 捕获寄存器对，选择将该寄存器用作匹配寄存器还是捕获寄存器。
2. 为每个所选的匹配寄存器定义匹配条件：
   - 如果使用 32 位计数器，则每个匹配寄存器 MATCH 将设置一个匹配值，如果使用 L 16 位计数器和 H 16 位计数器，则设置两个匹配值。
   - 计数器值达到其限值范围或值 0 时，每个匹配重新载入寄存器 MATCHRELOAD 将对加载到匹配寄存器中的重新载入值进行设置。

28.7.10.1.3 配置事件和事件响应

1. 在 EVCTRL 寄存器 （共 16 个寄存器，每个事件对应一个寄存器）中定义每个事件将在下列何种情况下发生：
   - 在 COMBMODE 字段中选择是在输入或输出发生变化时，还是输入电平或输出电平、计数器的匹配条件或匹配和输入 / 输出条件的组合下发生该事件。
   - 对于匹配条件：

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

选择包含事件发生的匹配条件的匹配寄存器。在 MATCHSEL 字段中输入所选匹配寄存器的数量。

如果使用 L 计数器和 H 计数器，则定义在 HEVENT 字段中进行 L 计数器匹配或 H 计数器匹配时是否发生该事件。

- 对于 SCT 输入或输出电平或跃迁：

  在字段 IOSEL 和 OUTSEL 中选择与该事件相关的输入编号或输出编号。

  在 IOCOND 字段中定义所选的输入或输出将如何启动该事件（与边缘或电平相关）。

2. 在 OUTPUTSET 或 OUTPUTCLR 寄存器中定义每个事件对 SCT 输出（共 16 个输出，每个输出对应一个寄存器）的作用：
   - 对于每个 SCT 输出，选择使用哪些事件来设置或清除该输出。可以使用多个事件来更改输出，并且每个事件可以更改多个输出。
3. 定义每个事件对计数器的影响：
   - 在 LIMIT 寄存器中设置相应的事件位，以通过该事件来设置计数器的上限值。

     当在单向模式中发生限值事件时，计数器将会清零，并从下一个时钟沿开始向上计数。

     当在双向模式中发生限值事件时，计数器将从下一个时钟沿的当前值开始向下计数。
   - 在 HALT 寄存器中设置相应的事件位，以通过该事件来终止计数器运行。如果该计数器终止运行，则它会停止计数，并且不会发生任何新事件。只有清除了 CTRL 计数器中的 HALT_L 和 / 或 HALT_H 位，才能恢复计数器操作。
   - 在 STOP 寄存器中设置相应的事件位，以通过该事件来停止计数器运行。如果该计数器停止运行，它会停止计数。但是通过配置为输入 / 输出跃迁的事件可以重新启动该计数器。
   - 在 START 寄存器中设置相应的事件位，以通过该事件来重新启动计数。只能使用输入更改所定义的事件来重新启动计数器。
4. 定义哪些事件可以启动 SCT 中断：
   - 在 EVEN 和 EVFLAG 寄存器中设置相应的事件位，以通过该事件来启动 SCT 中断。
5. 定义是否通过事件来启动 DMA 请求。
   - 在 DMAREQ0/1 寄存器中设置相应的事件位，以通过该事件来启动 DMA 请求 0 或 1。

28.7.10.1.4 配置多个状态

1. 在每个事件（共 16 个事件，每个事件对应一个寄存器）的 EVSTATEMASK 寄存器中，选择发生该事件所处于的某个状态或多个状态 （共 31 个）。所选的每个状态可用于多个事件。
2. 确定事件对系统状态的影响：

   在 EVCTRL 寄存器 （共 16 个事件，每个事件对应一个寄存器）中，在 STATEV 字段中为该事件设置新的状态值。如果该事件是当前状态中的最大编号事件，则该值会添加到现有状态值中或替换现有的状态值，具体取决于 STATELD 字段。

   注：如果在当前状态中还具有其他更大编号的事件，则该事件将无法更改状态。

   如果 STATEV 和 STATELD 值设置为零，则状态将不会发生改变。

28.7.10.1.5 其他选项

- 捕获寄存器的数量是特定的（可选）。可以将每个捕获寄存器编写为在发生一个或多个事件时捕获计数器内容。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- 如果计数器处于双向模式中，则可以建立输出的设置和清除功能，具体取决于计数器是向上计数还是向下计数 （通过对 OUTPUTDIRCTRL 计数器执行写入操作来确定）。
- < 待定 >

28.7.10.2 操作 SCT

1. 配置 SCT （参见第 28.7.10.1 节 “配置 SCT”）。
   a. 配置计数器 （参见第 28.7.10.1.1 节）。
   b. 配置匹配和捕获寄存器 （参见第 28.7.10.1.2 节）。
   c. 配置事件和事件响应 （参见第 28.7.10.1.3 节）。
   d. 配置多个状态 （第 28.7.10.1.4 节）。
2. 对状态寄存器执行写入操作来定义初始状态。默认初始状态为状态 0。
3. 如果要启动 SCT，则请对 CTRL 寄存器执行写入操作：
   - 清除计数器。
   - 清除或设置 STOP_L 和 / 或 STOP_H 位。

     注：计数器将在清除 STOP 位后立即开始计数。如果设置了 STOP 位，则 SCT 将等待配置为启动计数器的事件发生。
   - 对于每个计数器，可以选择单向计数或双向计数模式（字段BIDIR_L和/或BIDIR_H）。
   - 为计数器时钟 （CTRL 寄存器）选择预分频系数。
   - 清除 HALT_L 和 / 或 HALT_H 位。默认情况下，将会终止计数器运行，并且不会发生任何事件。
4. 如果要通过软件随时停止计数器运行，则请在 CTRL 寄存器中写入 STOP_L 和 / 或 STOP_H 位或 HALT_L 和 / 或 HALT_H 位来停止或终止计数器运行。
   - 当计数器停止运行时，可以将事件配置为清除STOP位或通过软件将零写入到STOP位，重新启动计数器。
   - 在计数器终止运行时，只需通过软件写入来清除 HALT 位便可重新启动计数器，并且不会发生任何事件。
   - 在计数器终止运行时，软件可以对 OUT 寄存器执行写入操作，以直接将任何 SCT 输出设置为高电平或低电平。

通过读取状态寄存器，即可随时读取当前的状态。

如果要通过软件来更改当前的状态（它与任何事件的发生无关），则应设置 HALT 位，并对状态寄存器执行写入操作来更改状态值。当计数器终止运行（设置了HALT_L和/或HALT_H位）时，仅允许对状态寄存器执行写入操作，并且此时不会发生任何事件。

28.7.10.3 在不使用状态的情况下配置 SCT

SCT 可用作带有外部捕获输入和匹配输出的标准计数器 / 定时器，但它不使用状态逻辑。如果要在无任何状态的情况下使用 SCT，则请按如下方式配置 SCT：

- 将零写入到状态寄存器中 （默认值是 0）。
- 将零写入到每个事件的 EVCTRL 寄存器的 STATELD 和 STATEV 字段中。
- 将 0x1 写入到每个事件的 EVSTATEMASK 寄存器中。写入 0x1 会使能该事件。

  实际上，事件可以在计数器运行的同时，在无变化的单一状态下发生。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

28.7.10.4 示例

图 89 显示了 SCT 使用两组匹配事件 （EV0/1 和 EV3/4）来设置 / 清除 SCT 输出 0 的简单应用。第三个匹配事件 (EV2) 用于对计数器执行复位 （无论当前状态如何）。

在初始状态 0 中，匹配事件 EV0 会将输出 0 设置为高电平，匹配事件 EV1 会清除输出 0。系统会监控 SCT 输入 0：如果输入从高电平跃迁至低电平 (EV2)，则该状态将变为状态 1，并且会使能 EV3/4，从而创建由不同的匹配值启动的相同输出。如果输入 0 从低电平跃迁至高电平，则相关事件 (EV5) 会使该状态改回至状态 0。在状态 0 中，将会使能 EV0 和 EV1。

本示例使用了下列 SCT 配置：

- 1 个输入
- 1 个输出
- 5 个匹配寄存器
- 7 个事件
- 2 个状态

图 89. SCT 配置示例

该 SCT 应用将使用下列配置 （已将表 668 中未列出的所有寄存器值设置为其默认值）：

表 668. SCT 配置示例

配置	寄存器	设置
计数器	CONFIG	使用一个计数器 （UNIFY = 1）。
	CTRL	使用单向计数器 （BIDIR_L = 0）。
时钟基准	CONFIG	使用默认值作为时钟配置。
匹配 / 捕获寄存器	REGMODE	为每个匹配事件配置一个匹配寄存器，即将 REGMODE_L 位 0、 1、 2、 4 和 5 设置为 0。这是默认设置。
定义匹配值	MATCH0/1/2/4/5	在每个寄存器中设置匹配值 MATCH0/1/2/4/5_L。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 668. SCT 配置示例 *（续）*

配置	寄存器	设置
定义匹配重新载入值	MATCHREL0/1/2/4/5	在每个寄存器中设置匹配重新载入值 RELOAD0/1/2/4/5_L （与该示例中的匹配值相同）。
在发生事件 0 时定义	EVCTRL0	• 将 COMBMODE 设置为 0x1。事件 0 仅使用匹配条件。 • 将 MATCHSEL 设置为 0。选择匹配寄存器 0 的匹配值。
在发生事件 1 时定义	EVCTRL1	• 将 COMBMODE 设置为 0x1。事件 1 仅使用匹配条件。 • 将 MATCHSEL 设置为 1。选择匹配寄存器 1 的匹配值。
在发生事件 2 时定义	EVCTRL2	• 将 COMBMODE 设置为 0x1。事件 2 仅使用匹配条件。 • 将 MATCHSEL 设置为 2。选择匹配寄存器 2 的匹配值。
在发生事件 3 时定义	EVCTRL3	• 将 COMBMODE 设置为 0x2。事件 3 仅使用 I/O 条件。 • 将 IOSEL 设置为 0。选择输入 0。 • 将 IOCOND 设置为 0x2。输入 0 将变为低电平。
定义事件 3 更改状态的方式	EVCTRL3	将 STATEV 位设置为 1 并将 STATED 位设置为 1。事件 3 会将该状态改为状态 1。
在发生事件 4 时定义	EVCTRL4	• 将 COMBMODE 设置为 0x1。事件 4 仅使用匹配条件。 • 将 MATCHSEL 设置为 0x3。选择匹配寄存器 4 的匹配值。
在发生事件 5 时定义	EVCTRL5	• 将 COMBMODE 设置为 0x1。事件 5 仅使用匹配条件。 • 将 MATCHSEL 设置为 0x3。选择匹配寄存器 5 的匹配值。
在发生事件 6 时定义	EVCTRL6	• 将 COMBMODE 设置为 0x2。事件 6 仅使用 I/O 条件。 • 将 IOSEL 设置为 0。选择输入 0。 • 将 IOCOND 设置为 0x1。输入 0 将转为高电平。
定义事件 6 更改状态的方式	EVCTRL6	将 STATEV 位设置为 0 并将 STATED 位设置为 1。事件 6 会将该状态改为状态 0。
定义将如何设置事件输出 0	OUTPUTSET0	将 SET0 位 0（对于事件 0）和 4（对于事件 4）设置为 1，以便在发生事件 0 和 4 时设置该事件输出。
定义将如何清除事件输出 0	OUTPUTCLR0	将 CLR0 位 1 （对于事件 1）和 5 （对于事件 5）设置为 1，以便在发生事件 1 和 5 时清除该事件输出。
定义将使用哪个事件对计数器进行复位	LIMIT	将 LIMMASK_L 位 2 设置为 1 （通过事件 2 来限制计数器）。将所有其他位设置为零。
配置在状态中使能事件 0	EVSTATEMSK0	将 STATEMSK0 位 0 设置为 1。将所有其他位设置为 0。将在状态 0 中使能事件 0。
配置在状态中使能事件 1	EVSTATEMSK1	将 STATEMSK1 位 0 设置为 1。将所有其他位设置为 0。将在状态 0 中使能事件 1。
配置在状态中使能事件 2	EVSTATEMSK2	将 STATEMSK2 位 0 设置为 1，并将位 1 设置为 1。将所有其他位设置为 0。将在状态 0 和状态 1 中使能事件 2。
配置在状态中使能事件 3	EVSTATEMSK3	将 STATEMSK3 位 0 设置为 1。将所有其他位设置为 0。将在状态 0 中使能事件 3。
配置在状态中使能事件 4	EVSTATEMSK4	将 STATEMSK4 位 1 设置为 1。将所有其他位设置为 0。将在状态 1 中使能事件 4。
配置在状态中使能事件 5	EVSTATEMSK5	将 STATEMSK5 位 1 设置为 1。将所有其他位设置为 0。将在状态 1 中使能事件 5。
配置在状态中使能事件 6	EVSTATEMSK6	将 STATEMSK6 位 1 设置为 1。将所有其他位设置为 0。将在状态 1 中使能事件 6。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

29.1 本章导读

定时器可用于所有 LPC43xx 器件中。

29.2 基本配置

定时器将按如下方式进行配置：

- 参见 表 669 以了解时钟与电源控制。
- 定时器 0/1/2/3 将通过 TIMER0/1/2/3_RST （复位 #32/33/34/35）来复位。
- 定时器0/1/2/3中断将连接至NVIC中的编号为12/13/14/15的插槽中。定时器0/1/3的匹配通道 2 将连接至事件路由器的第 13、14 和 16 插槽中。（可使用 SCT 输出 2、6 和 14 来对这些输出进行 “OR”操作）
- 如果要将定时器 0/1/2/3 的匹配通道 0 和 1 连接至 GPDMA，则请在 CREG 模块中使用 DMAMUX 寄存器 （参见表 41），并在 DMA 通道配置寄存器 （参见第 19.6.20 节）中使能 GPDMA 通道。
- 定时器寄存器还可由 GPDMA 作为存储器到存储器传输来访问。
- 定时器捕获输入和匹配输出将通过 GIMA 来配置 （参见第 16.3 节）。
- 所有定时器捕获输入还可连接至专用的外部引脚中（参见第 16.3 节和第 15.3.11 节）。

表 669. 定时器 0/1/2/3 时钟和电源控制

	基本时钟	分支时钟	工作频率
定时器 0 寄存器接口时钟和定时器 0 外设时钟 PCLK。	BASE_M4_CLK	CLK_M4_TIMER0	高达 204 MHz
定时器 1 寄存器接口时钟和定时器 1 外设时钟 PCLK。	BASE_M4_CLK	CLK_M4_TIMER1	高达 204 MHz
定时器 2 寄存器接口时钟和定时器 2 外设时钟 PCLK。	BASE_M4_CLK	CLK_M4_TIMER2	高达 204 MHz
定时器 3 寄存器接口时钟和定时器 3 外设时钟 PCLK。	BASE_M4_CLK	CLK_M4_TIMER3	高达 204 MHz

29.3 功能要点

- 使用 32 位 可编程预分频器的 32 位定时器 / 计数器。
- 计数器或定时器操作
- 每个定时器拥有四个32位捕获通道，可以在输入信号跃迁时生成定时器值快照。捕获事件还可以有选择性地生成中断。
- 四个 32 位匹配寄存器可以：

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- 连续操作，可选择在匹配时产生中断。
- 在与可选中断生成相匹配时停止定时器运行。
- 在与可选中断生成相匹配时进行定时器复位。

• 匹配寄存器拥有四个外部输出，它们具有如下功能：
- 匹配时设置为低电平。
- 匹配时设置为高电平。
- 匹配时切换。
- 匹配时不执行任何操作。

29.4 简介

定时器 / 计数器旨在计算外设时钟 (PCLK) 或外部提供的时钟的周期，并且可以有选择性地生成中断或基于四个匹配寄存器的指定定时器值来执行其他操作。它还包括四个捕获输入用于捕获在输入信号跃迁时的定时器值，并且可以有选择性地生成中断。

表 670 简要总结了每个定时器 / 计数器相关的功能。

表 670. 定时器 / 计数器功能描述

引脚	类型	描述
CAP0_[3:0] CAP1_[3:0] CAP2_[3:0] CAP3_[3:0]	输入	捕获信号 - 可以配置捕获输入上的跳变，使某个捕获寄存器载入定时器计数器中的值，并且可以有选择性地生成中断。捕获功能可从多个引脚中进行选择。 定时器 / 计数器模块可以选择捕获信号，将其作为时钟源，而不是 PCLK 衍生的时钟。有关更多详情，请参阅第 29.7.11 节。
MAT0_[3:0] MAT1_[3:0] MAT2_[3:0] MAT3_[3:0]	输出	外部匹配输出 - 当匹配寄存器 (MR3:0) 等于定时器计数器 (TC) 时，该输出可以进行切换、转入低电平、转入高电平或不执行任何操作。外部匹配寄存器 (EMR) 用于控制该输出的功能。可以为多个并行引脚选择匹配输出功能。

29.5 引脚说明

每个定时器捕获通道的输入信号可从外部引脚或几个其他内部源产生。GIMA（参见表138）和（对于每个定时器的捕获通道 3）CREG6 的 CTOUTCTRL 位确定由定时器捕获的信号。

匹配输出连接至 Tn_MATm 引脚功能。此外，在 CREG6 寄存器中将 CTOUTCTRL 位设为 0 （默认）时，匹配输出或 SCT 输出可在 CTOUT 引脚上进行监控 （参见表 43）。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 671. 定时器 0/1/2/3 输入和输出

描述	引脚功能	内部信号	默认（参见 GIMA，表 138）	CTOUTCTRL 位（参见表 43）
定时器 0 输入				
定时器 0，捕获通道 0 的输入	CTIN_0	-	是	-
	T0_CAP0	-	否	-
定时器 0，捕获通道 1 的输入	CTIN_1	-	是	-
	T0_CAP1	-	否	-
	-	USART2 TX 有效	否	-
定时器 0，捕获通道 2 的输入	CTIN_2	-	是	-
	T0_CAP2	-	否	-
定时器 0，捕获通道 3 的输入	-	SCT 输出 15 或 T3 匹配通道 3	是	0
	-	SCT 输出 15	是	1
	T0_CAP3	-	否	-
	-	T3 匹配通道 3	否	-
定时器 0 输出				
定时器 0；匹配输出通道 3 到 0	T0_MAT[3:0]	-	-	-
定时器 0；匹配输出通道 0	-	ADC 启动 0 输入（ADC CR 寄存器 START 位 = 0x2）		1
定时器 0；匹配输出通道 2	-	事件路由器输入 13	否	1
定时器 0；匹配输出通道 3 至 0 或者 SCT 输出 3 至 0。	CTOUT_[3:0]	-		0
定时器 1 输入				
定时器 1，捕获通道 0 的输入	CTIN_0	-	是	-
	T1_CAP0	-	否	-
定时器 1，捕获通道 1 的输入	CTIN_3	-	是	-
	T1_CAP1	-	否	-
	-	USART0 TX 有效	否	-
定时器 1，捕获通道 2 的输入	CTIN_4	-	是	-
	T1_CAP2	-	否	-
	-	USART0 RX 有效	否	-
定时器 1，捕获通道 3 的输入	-	SCT 输出 3 或 T0 匹配通道 3	是	0
	-	SCT 输出 3	是	1
	-	T0 匹配通道 3	否	-
	T1_CAP3	-	否	-
定时器 1 输出				
定时器 1；匹配输出通道 3 到 0	T1_MAT[3:0]	-	-	-
定时器 1；匹配输出通道 2	-	事件路由器输入 14	否	1
定时器 1；匹配输出通道 3 至 0 或者 SCT 输出 7 至 4。	CTOUT_[7:4]	-	-	0
定时器1；匹配输出通道2或者SCT输出6	-	事件路由器输入 14	否	0

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 671. 定时器 0/1/2/3 输入和输出 *（续）*

描述	引脚功能	内部信号	默认（参见 GIMA，表 138）	CTOUTCTRL 位（参见表 43）
定时器 2 输入				
定时器 2，捕获通道 0 的输入	CTIN_0	-	是	-
	T2_CAP0	-	否	-
定时器 2，捕获通道 1 的输入	CTIN_1	-	是	-
	T2_CAP1	-	否	-
	-	USART2 TX 有效	否	-
	-	I2S1_RX_MWS	否	-
定时器 2，捕获通道 2 的输入	CTIN_5	-	是	-
	T2_CAP2	-	否	-
	-	USART2 RX 有效	否	-
	-	I2S1_TX_MWS	否	-
定时器 2，捕获通道 3 的输入	-	SCT 输出 7 或 T1 匹配通道 3	是	0
定时器 2，捕获通道 3 的输入	-	SCT 输出 7	是	1
	-	T1 匹配通道 3	否	-
	T2_CAP3	-	否	-
定时器 2 输出				
定时器 2；匹配输出通道 3 到 0	T2_MAT[3:0]	-	-	-
定时器 2；匹配输出通道 0	-	ADC 启动 1 输入（ADC CR 寄存器位 START = 0x3）	否	1
定时器 2；匹配输出通道 3 至 0 或者 SCT 输出 11 至 8。	CTOUT_[11:8]	-	-	0
定时器2；匹配输出通道0或者SCT输出8	-	ADC 启动 1 输入（ADC CR 寄存器位 START = 0x3）	否	0
定时器 3 输入				
定时器 3，捕获通道 0 的输入	CTIN_0		是	-
	-	I2S0_RX_MWS	否	-
	T3_CAP0	-	否	-
定时器 3，捕获通道 1 的输入	CTIN_6	-	是	-
	T3_CAP1	-	否	-
	-	USART3 TX 有效	否	-
	-	I2S0_TX_MWS	否	-
定时器 3，捕获通道 2 的输入	CTIN_7	-	是	-
	T3_CAP2	-	否	-
	-	USART3 RX 有效	否	-
	-	SOF0	否	-
定时器 3，捕获通道 3 的输入	T3_CAP3		否	-
	-	SCT 输出 11 或 T2 匹配通道 3	是	0
	-	SCT 输出 11	是	1
	-	T2 匹配通道 3	否	-
	-	SOF1	否	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 671. 定时器 0/1/2/3 输入和输出 *（续）*

描述	引脚功能	内部信号	默认（参见 GIMA，表 138）	CTOUTCTRL 位（参见表 43）
定时器 3 输出				
定时器 3；匹配输出通道 3 到 0	T3_MAT[3:0]	-	-	-
定时器 3；匹配输出通道 2	-	事件路由器输入 16	否	1
定时器 3；匹配输出通道 3 至 0 或者 SCT 输出 15 至 12。	CTOUT_[15:12]	-	-	0
定时器 3；匹配输出通道 2 或者 SCT 输出 14	-	事件路由器输入 16	否	0

29.6 DMA 连接

< 待定 >

29.7 寄存器描述

每个定时器 / 计数器都包含寄存器，如表 672 所示。

表 672. 寄存器简介：定时器 0/1/2/3（寄存器基址 0x4008 4000 (TIMER0)、0x4008 5000 (TIMER1)、0x400C 3000 (TIMER2) 和 0x400C 4000 (TIMER3)）

名称	访问类型	地址偏移	描述	复位值 [1]	参考
IR	R/W	0x000	中断寄存器。可以对 IR 执行写入操作来清除中断。可以对 IR 执行读取操作，以确定可能使用哪个中断源（共 8 个）。	0	表 673
TCR	R/W	0x004	定时器控制寄存器。TCR 用于控制定时器计数器功能。通过 TCR 可以对定时器计数器执行禁用或复位操作。	0	表 674
TC	R/W	0x008	定时器计数器。32 位 TC 会在 PCLK 的每个 PR+1 周期时递增计数。 TC 将通过 TCR 来控制。	0	表 675
PR	R/W	0x00C	预分频寄存器。当预分频计数器 (PC) 与该值相等时，下个时钟 TC 加 1， PC 清除。	0	表 676
PC	R/W	0x010	预分频计数器。 32 位 PC 是一个计数器，它会增加到与 PR 中存放的值相等。当计数到达 PR 中的值时， TC 将递增计数，并且会清除 PC 值。通过总线接口可以观察和控制 PC。	0	表 677
MCR	R/W	0x014	匹配控制寄存器。 MCR 用于控制在发生匹配时是否生成中断，以及是否进行 TC 复位。	0	表 678
MR0	R/W	0x018	匹配寄存器 0。通过 MCR 可以使能 MR0，以便在每次 MR0 与 TC 相匹配时执行 TC 复位、停止 TC 和 PC 运行和 / 或生成中断。	0	表 679
MR1	R/W	0x01C	匹配寄存器 1。请参见 MR0 描述。	0	表 679
MR2	R/W	0x020	匹配寄存器 2。请参见 MR0 描述。	0	表 679
MR3	R/W	0x024	匹配寄存器 3。请参见 MR0 描述。	0	表 679
CCR	R/W	0x028	捕获控制寄存器。CCR用于控制将使用哪些捕获输入边缘来加载捕获寄存器，并且在出现捕获时是否生成中断。	0	表 680
CR0	RO	0x02C	捕获寄存器 0。当发生 CAPn.0（分别为 CAP0.0 或 CAP1.0）输入事件时， CR0 将载入 TC 值。	0	表 681
CR1	RO	0x030	捕获寄存器 1。请参见 CR0 描述。	0	表 681

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 672. 寄存器简介：定时器 0/1/2/3（寄存器基址 0x4008 4000 (TIMER0)、0x4008 5000 (TIMER1)、0x400C 3000 (TIMER2) 和 0x400C 4000 (TIMER3)）*（续）*

名称	访问类型	地址偏移	描述	复位值 [1]	参考
CR2	RO	0x034	捕获寄存器 2。请参见 CR0 描述。	0	表 681
CR3	RO	0x038	捕获寄存器 3。请参见 CR0 描述。	0	表 681
EMR	R/W	0x03C	外部匹配寄存器。 EMR 用于控制外部匹配引脚 MATn.0-3（分别为 MAT0.0-3 和 MAT1.0-3）。	0	表 682
CTCR	R/W	0x070	计数控制寄存器。 CTCR 用于选择定时器模式和计数器模式。并且在计数器模式中，会选择要进行计数的信号和边缘。	0	表 684

[1] 复位值仅反映存储在所用位中的数据，不包括保留位的内容。

29.7.1 定时器中断寄存器

中断寄存器由匹配中断的四个位以及捕获中断的四个位组成。如果生成中断，则 IR 中的相应位将为高位，否则将为低位。将逻辑 1 写入到相应的 IR 位中，将执行中断复位。写入 0 无效。在执行清除定时器匹配中断操作时，也会清除任何相应的 DMA 请求。

表 673. 定时器中断寄存器IR（IR - 地址0x4008 4000 (TIMER0)、0x4008 5000 (TIMER1)、0x400C 3000 (TIMER3) 和 0x400C 4000 (TIMER4)）位描述

位	符号	描述	复位值
0	MR0INT	匹配通道 0 的中断标志。	0
1	MR1INT	匹配通道 1 的中断标志。	0
2	MR2INT	匹配通道 2 的中断标志。	0
3	MR3INT	匹配通道 3 的中断标志。	0
4	CR0INT	捕获通道 0 事件的中断标志。	0
5	CR1INT	捕获通道 1 事件的中断标志。	0
6	CR2INT	捕获通道 2 事件的中断标志。	0
7	CR3INT	捕获通道 3 事件的中断标志。	0
31:8	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

29.7.2 定时器控制寄存器

定时器控制寄存器 (TCR) 用于控制定时器 / 计数器的操作。

表 674. 定时器控制寄存器 TCR（TCR - 地址 0x4008 4004 (TIMER0)、0x4008 5004 (TIMER1)、0x400C 3003 (TIMER2) 和 0x400C 4004 (TIMER3)）位描述

位	符号	描述	复位值
0	CEN	当值为 1 时，会使能定时器计数器和预分频计数器来计数。当值为 0 时，会禁用计数器。	0
1	CRST	当值为 1 时，会在 PCLK 的下一个正边缘上对定时器计数器和预分频计数器同步执行复位。计数器将保留为复位状态，直至 TCR[1] 归零为止。	0
31:2	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

29.7.3 定时器计数器寄存器

当预分频计数器的值达到其最终计数时， 32 位定时器计数器寄存器会递增计数。除非该计数器在达到其上限值之前复位，否则定时器计数器将向上计数至 0xFFFF FFFF 值，然后再返回至 0x0000 0000 值。该事件将不会产生中断，但是在需要时可使用匹配寄存器来检测溢出。

表 675. 定时器计数器寄存器 TC（TC - 地址 0x4008 4008 (TIMER0)、0x4008 5008 (TIMER1)、0x400C 3008 (TIMER2) 和 0x400C 4008 (TIMER3)）位描述

位	符号	描述	复位值
31:0	TC	定时器计数器值。	0

29.7.4 定时器预分频寄存器

32 位定时器预分频寄存器用于指定预分频计数器的最大值。

表 676. 定时器预分频寄存器 PR（PR - 地址 0x4008 400C (TIMER0)、0x4008 500C (TIMER1)、0x400C 300C (TIMER2) 和 0x400C 400C (TIMER3)）位描述

位	符号	描述	复位值
31:0	PM	预分频计数器最大值。	0

29.7.5 定时器预分频计数器寄存器

32 位预分频计数器将在 PCLK 应用于定时器计数器之前，使用某一常数值对 PCLK 执行分频控制。它所控制的是定时器分辨率与最大时间之间的关系，从而能防止定时器溢流。预分频计数器会在每个 PCLK 时钟上递增计数。当预分频计数器的计数到达预分频寄存器中存储的值时，定时器计数器将递增计数，并且在下一个 PCLK 时钟上进行预分频计数器复位。这将使得定时器计数器当 PR = 0 时在每个 PCLK 上递增计数，当 PR = 1 时，在每 2 个 PCLK 上递增计数，依次类推。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 677. 定时器预分频计数器寄存器 PC（PC - 地址 0x4008 4010 (TIMER0)、0x4008 5010 (TIMER1)、0x400C 3010 (TIMER2) 和 0x400C 4010 (TIMER3)）位描述

位	符号	描述	复位值
31:0	PC	预分频计数器值。	0

29.7.6 定时器匹配控制寄存器

匹配控制寄存器用于控制在某个匹配寄存器与定时器计数器相匹配时将执行的操作。匹配控制寄存器各位的功能如表 678 所示。

表 678. 定时器匹配控制寄存器MCR（MCR - 地址0x4008 4014 (TIMER0), 0x4008 5014 (TIMER1), 0x400C 3014 (TIMER2), 0x400C 4014 (TIMER3)）位描述

位	符号	值	描述	复位值
0	MR0I		MR0 的中断操作。	0
		1	当 MR0 与 TC 中的值相匹配时会生成中断。	
		0	中断处于禁用状态。	
1	MR0R		MR0 的复位操作。	0
		1	如果 MR0 与 TC 相匹配，则会执行 TC 复位。	
		0	该功能处于禁用状态。	
2	MR0S	1	MR0 的停止操作。	0
		1	如果 MR0 与 TC 相匹配，则 TC 和 PC 将停止，且 TCR[0] 将被设置为 0。	
		0	该功能处于禁用状态。	
3	MR1I		MR1 的中断操作。	0
		1	当 MR1 与 TC 中的值相匹配时会生成中断。	
		0	中断处于禁用状态。	
4	MR1R		MR1 的复位操作。	0
		1	如果 MR1 与 TC 相匹配，则会执行 TC 复位。	
		0	该功能处于禁用状态。	
5	MR1S		MR1 的停止操作。	0
		1	如果 MR1 与 TC 相匹配，则 TC 和 PC 将停止运行，并且 TCR[0] 将设置为 0。	
		0	该功能处于禁用状态。	
6	MR2I		MR2 的中断操作。	0
		1	当 MR2 与 TC 中的值相匹配时会生成中断。	
		0	中断处于禁用状态。	
7	MR2R		MR2 的复位操作。	0
		1	如果 MR2 与 TC 相匹配，则会执行 TC 复位。	
		0	该功能处于禁用状态。	
8	MR2S		MR2 的停止操作。	0
		1	如果 MR2 与 TC 相匹配，则 TC 和 PC 将停止运行，并且 TCR[0] 将设置为 0。	
		0	该功能处于禁用状态。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 678. 定时器匹配控制寄存器MCR（MCR - 地址0x4008 4014 (TIMER0), 0x4008 5014 (TIMER1), 0x400C 3014 (TIMER2), 0x400C 4014 (TIMER3)）位描述 *（续）*

位	符号	值	描述	复位值
9	MR3I		MR3 的中断操作。	0
		1	当 MR3 与 TC 中的值相匹配时会生成中断。	
		0	该中断处于禁用状态。	
10	MR3R		MR3 的复位操作。	0
		1	如果 MR3 与 TC 相匹配时，则会执行 TC 复位。	
		0	该功能处于禁用状态。	
11	MR3S		MR3 的停止操作。	0
		1	如果 MR3 与 TC 相匹配，则 TC 和 PC 将停止运行，并且 TCR[0] 将设置为 0。	
		0	该功能处于禁用状态。	
31:12	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

29.7.7 定时器匹配寄存器 (MR0 - MR3)

匹配寄存器值将不断与定时器计数器值进行比较。直至两个值相等时，会自动启动各种操作。可能产生的操作有生成中断、定时器计数器复位或停止定时器运行。这些操作将由 MCR 寄存器中的设置来控制。

表 679. 定时器匹配寄存器 MR0 至 3（MR，地址 0x4008 4018 (MR0) 至 0x4008 4024 (M3) (TIMER0)、0x4008 5018 (MR0) 至 0x4008 5024 (MR3) (TIMER1)、0x400C 3018 (MR0) 至 0x400C 8024 (MR3) (TIMER2)、 0x400C 4018 (MR0) 至 0x400C 4024 (MR3) (TIMER3)）位描述

位	符号	描述	复位值
31:0	MATCH	定时器计数器匹配值。	0

29.7.8 定时器捕获控制寄存器

捕获控制寄存器用于控制在发生捕获事件时，四个捕获寄存器之一是否会载入定时器计数器中的值，并且控制是否通过捕获事件来生成中断。同时设置上升位和下降位是一种有效的配置，这会在两个边缘产生捕获事件。在下列描述中， n 代表定时器编号。

注：如果在 CTCR 中为特定的 CAP 输入选择了计数器模式，则应在该寄存器中将该输入的 3 个位编写为 000，而为其他 3 个 CAP 输入选择捕获和 / 或中断。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 680. 定时器捕获控制寄存器（CCR - 地址 0x4008 4028 (TIMER0), 0x4008 5020 (TIMER1), 0x400C 3028 (TIMER2), 0x400C 4028 (TIMER3)）位描述

位	符号	值	描述	复位值
0	CAP0RE		CAPn.0 上升沿上的捕获操作。	0
		1	CAPn.0 上的从 0 至 1 的序列，将使 CR0 载入 TC 内容。	
		0	该功能处于禁用状态。	
1	CAP0FE		CAPn.0 下降沿上的捕获操作。	0
		1	CAPn.0 上的从 1 至 0 的序列，将使 CR0 载入 TC 内容。	
		0	该功能处于禁用状态。	
2	CAP0I		CAPn.0 事件的中断操作。	0
		1	由于 CAPn.0 事件而加载 CR0，将导致中断生成。	
		0	该功能处于禁用状态。	
3	CAP1RE		CAPn.1 上升沿上的捕获操作。	0
		1	CAPn.1 上的从 0 至 1 的序列，将使 CR1 载入 TC 内容。	
		0	该功能处于禁用状态。	
4	CAP1FE		CAPn.1 下降沿上的捕获操作。	0
		1	CAPn.1 上的从 1 至 0 的序列，将使 CR1 载入 TC 内容。	
		0	该功能处于禁用状态。	
5	CAP1I		CAPn.1 事件的中断操作。	0
		1	由于 CAPn.1 事件而加载 CR1，将导致中断生成。	
		0	该功能处于禁用状态。	
6	CAP2RE		CAPn.2 上升沿上的捕获操作。	0
		1	CAPn.2 上的从 0 至 1 的序列，将使 CR2 载入 TC 内容。	
		0	该功能处于禁用状态。	
7	CAP2FE		CAPn.2 下降沿上的捕获操作。	0
		1	CAPn.2 上的从 1 至 0 的序列，将使 CR2 载入 TC 内容。	
		0	该功能处于禁用状态。	
8	CAP2I		CAPn.2 事件的中断操作。	0
		1	由于 CAPn.2 事件而加载 CR2，将导致中断生成。	
		0	该功能处于禁用状态。	
9	CAP3RE		CAPn.3 上升沿上的捕获操作。	0
		1	CAPn.3 上的从 0 至 1 的序列，将使 CR3 载入 TC 内容。	
		0	该功能处于禁用状态。	
10	CAP3FE		CAPn.3 下降沿上的捕获操作。	0
		1	CAPn.3 上的从 1 至 0 的序列，将使 CR3 载入 TC 内容。	
		0	该功能处于禁用状态。	
11	CAP3I		CAPn.3 事件的中断操作。	0
		1	由于 CAPn.3 事件而加载 CR3，将导致中断生成。	
		0	该功能处于禁用状态。	
31:12	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

29.7.9 定时器捕获寄存器 (CR0 - CR3)

每个捕获寄存器都与设备引脚相关，并且在该引脚上发生指定的事件时会载入定时器计数器值。捕获控制寄存器中的设置用于确定是否使能捕获功能，以及是在相关引脚的上升沿，还是下降沿或两者之上发生捕获事件。

表 681. 定时器捕获寄存器CR0至3（CR，地址0x4008 402C (CR0)至0x4008 4038 (CR3) (TIMER0)、0x4008 502C (CR0) 至 0x4008 5038 (CR3)(TIMER1)、 0x400C 302C (CR0) 至 0x400C 3038 (CR3) (TIMER2) 和 0x400C 402C (CR0) 至 0x400C 4038 (CR3) (TIMER3)）位描述

位	符号	描述	复位值
31:0	CAP	定时器计数器捕获值。	0

29.7.10 定时器外部匹配寄存器

外部匹配寄存器为外部匹配引脚提供控制和状态。在下列描述中，“n”代表定时器编号 0 或 1，“m”代表匹配编号 0 至 3。

每个定时器中的匹配 0 和匹配 1 的匹配事件会产生 DMA 请求，请参见第 29.7.12 节。

表 682. 定时器外部匹配寄存器（EMR - 地址 0x4008 403C (TIMER0)、0x4008 503C (TIMER1)、0x400C 303C (TIMER2) 和 0x400C 403C (TIMER3)）位描述

位	符号	值	描述	复位值
0	EM0		外部匹配 0。当 TC 与 MR0 相匹配时，该位可以进行切换、转入低电平、转入高电平或不执行任何操作，具体取决于该寄存器的位 5:4。可通过正逻辑的方式将该位导入到 MATn.0 引脚中（0 表示低电平， 1 表示高电平）。	0
1	EM1		外部匹配 1。当 TC 与 MR1 相匹配时，该位可以进行切换、转入低电平、转入高电平或不执行任何操作，具体取决于该寄存器的位 7:6。可通过正逻辑的方式将该位导入到 MATn.1 引脚中（0 表示低电平， 1 表示高电平）。	0
2	EM2		外部匹配 2。当 TC 与 MR2 相匹配时，该位可以进行切换、转入低电平、转入高电平或不执行任何操作，具体取决于该寄存器的位 9:8。可通过正逻辑的方式将该位导入到 MATn.0 引脚中（0 表示低电平， 1 表示高电平）。	0
3	EM3		外部匹配 3。当 TC 与 MR3 相匹配时，该位可以进行切换、转入低电平、转入高电平或不执行任何操作，具体取决于该寄存器的位 11:10。可通过正逻辑的方式将该位导入到 MATn.0 引脚中（0 表示低电平， 1 表示高电平）。	0
5:4	EMC0		外部匹配控制 0。用于确定外部匹配 0 的功能。	00
		0x0	不执行任何操作。	
		0x1	将相应的外部匹配位 / 输出清 0 （如果引脚处于输出状态，则 MATn.m 引脚为低电平）。	
		0x2	将相应的外部匹配位 / 输出设置为 1 （如果引脚处于输出状态，则 MATn.m 引脚为高电平）。	
		0x3	切换相应的外部匹配位 / 输出。	
7:6	EMC1		外部匹配控制 1。用于确定外部匹配 1 的功能。	00
		0x0	不执行任何操作。	
		0x1	将相应的外部匹配位 / 输出清 0 （如果引脚处于输出状态，则 MATn.m 引脚为低电平）。	
		0x2	将相应的外部匹配位 / 输出设置为 1 （如果引脚处于输出状态，则 MATn.m 引脚为高电平）。	
		0x3	切换相应的外部匹配位 / 输出。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 682. 定时器外部匹配寄存器（EMR - 地址 0x4008 403C (TIMER0)、0x4008 503C (TIMER1)、0x400C 303C (TIMER2) 和 0x400C 403C (TIMER3)）位描述 *（续）*

位	符号	值	描述	复位值
9:8	EMC2		外部匹配控制 2。用于确定外部匹配 2 的功能。	00
		0x0	不执行任何操作。	
		0x1	将相应的外部匹配位 / 输出清 0 （如果引脚处于输出状态，则 MATn.m 引脚为低电平）。	
		0x2	将相应的外部匹配位 / 输出设置为 1 （如果引脚处于输出状态，则 MATn.m 引脚为高电平）。	
		0x3	切换相应的外部匹配位 / 输出。	
11:10	EMC3		外部匹配控制 3。用于确定外部匹配 3 的功能。	00
		0x0	不执行任何操作。	
		0x1	将相应的外部匹配位 / 输出清 0 （如果引脚处于输出状态，则 MATn.m 引脚为低电平）。	
		0x2	将相应的外部匹配位 / 输出设置为 1 （如果引脚处于输出状态，则 MATn.m 引脚为高电平）。	
		0x3	切换相应的外部匹配位 / 输出。	
15:12	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

表 683. 外部匹配控制

EMR[11:10], EMR[9:8], EMR[7:6], or EMR[5:4]	功能
00	不执行任何操作。
01	将相应的外部匹配位 / 输出清 0 （如果引脚处于输出状态，则 MATn.m 引脚为低电平）。
10	将相应的外部匹配位/输出设置为1（如果引脚处于输出状态，则MATn.m 引脚为高电平）。
11	切换相应的外部匹配位 / 输出。

29.7.11 定时器计数控制寄存器

计数控制寄存器 （CTCR）用于选择定时器模式和计数器模式。并且在计数器模式中，会选择要进行计数的引脚和边缘。

如果将计数器模式选为操作模式，则在 PCLK 时钟的每个上升沿上会对 （CTCR 位 3:2 所选的）CAP 输入进行采样。在比较该 CAP 输入的两个连续样本后，将会确认下列四个事件之一：所选 CAP 输入电平处于上升沿、下降沿，或上升下降沿或无变化。仅当所标识的事件与 CTCR 寄存器中的位 1:0 所选的内容相匹配时，定时器计数器寄存器才会递增计数。

计数器的外部提供时钟的处理能力具有一定的局限性，因此不能执行有效的处理。由于 PCLK 时钟的两个连续上升沿仅用于标识 CAP 所选输入的一个边缘，因此 CAP 输入的频率不能超过 PCLK 时钟的四分之一。因此在这种情况下，相同的 CAP 输入上的高电平 / 低电平的持续时间不能少于 1/(2 PCLK)。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 684. 定时器计数控制寄存器 CTCR（CTCR - 地址 0x4008 4070 (TIMER0), 0x4008 5070 (TIMER1), 0x400C 3070 (TIMER2), 0x400C 4070 (TIMER3)）位描述

位	符号	值	描述	复位值
1:0	CTMODE		计数器 / 定时器模式 该字段用于选择可使用哪个 PCLK 上升沿，使定时器预分频计数器 (PC) 递增计数，或清除 PC 并使定时器计数器 (TC) 递增计数。 定时器模式：当预分频计数器与预分频寄存器相匹配时，TC 将递增计数。	00
		0x0	定时器模式：每个 PCLK 的上升沿。	
		0x1	计数器模式：将在位 3:2 所选的 CAP 输入的上升沿上执行 TC 递增计数。	
		0x2	计数器模式：将在位 3:2 所选的 CAP 输入的下降沿上执行 TC 递增计数。	
		0x3	计数器模式：将在位 3:2 所选的 CAP 输入的上升沿和下降沿上执行 TC 递增计数。	
3:2	CINSEL		计数输入选择 当此寄存器中的位 1:0 不为 00 时，这些位将选择采样哪个 CAP 引脚来提供时钟：	00
		0x0	CAPn.0 用于 TIMERn	
		0x1	CAPn.1 用于 TIMERn	
		0x2	CAPn.2 用于 TIMERn	
		0x3	CAPn.3 用于 TIMERn **注：**如果在 TnCTCR 中为特定的 CAPn 输入选择了计数器模式，则应在捕获控制寄存器 (TnCCR) 中将该输入的 3 个位编写为 000，而在相同定时器中为其他 3 个 CAPn 输入选择捕获和 / 或中断。	
31:4	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

29.7.12 DMA 操作

每个定时器的外部匹配 0 位和 1 位在 0 到 1 的跃迁过程中，会生成 DMA 请求。如果要产生某种效果，则必须配置 GPDMA，并且必须通过 CREG 模块将相关的定时器 DMA 请求选为 DMA 源，如表 41 所示。

当定时器最初设置为生成 DMA 请求时，可能会在匹配条件产生之前使该请求变为有效。通过软件可以将 1 写入到中断标志位置来避免最初的 DMA 请求，就像清除定时器中断一样。参见第 29.7.1 节。当使用 GPDMA 控制器来执行中断时，会自动清除 DMA 请求。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

29.8 定时器操作示例

如图 90 所示，定时器配置为在匹配时复位计数并产生中断。将预分频器设置为 2，匹配寄存器设置为 6。在发生匹配的定时器周期结束时，会进行定时器计数复位。这就为匹配值提供了一个完整周期。当定时器计数到达匹配值后，将在下一个时钟内生成中断以指示出现了匹配。

如图 91 所示，定时器配置为在匹配时停止计数并产生中断。将预分频器再次设置为 2，匹配寄存器设置为 6。当定时器计数到达匹配值后，将在下一个时钟内清除 TCR 中的定时器使能位，并且会生成中断以指示出现了匹配。

图 90. 定时器周期，其中 PR = 2、 MRx = 6，并且使能了在匹配时同时执行中断和复位操作

图 91. 定时器周期，其中 PR = 2、 MRx = 6，并且使能了在匹配时同时执行中断和停止操作

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

29.9 架构

图 92 显示了 TIMER/COUNTER0 和 TIMER/COUNTER1 的框图。

图 92. 定时器框图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

30.1 本章导读

电机控制 PWM 可用于所有 LPC43xx 器件中。

30.2 基本配置

PWM 将按如下方式进行配置：

- 参见表 685 以了解时钟与电源控制。
- PWM 通过 MOTOCONPWM_RST （复位号 38）复位。
- PWM 中断将连接至 NVIC 的编号为 16 的插槽中。

表 685. PWM 时钟和电源控制

	基本时钟	分支时钟	工作频率
PWM 电机控制模块时钟和 PWM 电机控制外设时钟。	BASE_APB1_CLK	CLK_APB1_MOTOCON	高达 204 MHz

30.3 简介

电动机控制 PWM (MCPWM) 是专为三相交流和直流电动机控制应用而优化的器件，但它也可以在定时、计数、捕获和比较等许多其他应用中使用。

30.4 功能要点

MCPWM 包含了三个独立的通道，并且每个通道包含：

- 一个 32 位定时器 / 计数器 (TC)
- 一个 32 位限值寄存器 (LIM)
- 一个 32 位匹配寄存器 (MAT)
- 一个 10 位死区寄存器 (DT) 和一个相关的 10 位死区计数器
- 一个 32 位捕获寄存器 (CAP)
- 具有相反极性的两个调制输出 （MCOA 和 MCOB）
- 一个周期中断、一个脉冲宽度中断和一个捕获中断

输入引脚 MCI0-2 可以启动 TC 捕获或实现通道 TC 计数递增。全局终止输入会强制所有通道进入 “无源” 状态，并产生中断。

30.5 简介

第 30.8 节提供了各种 MCPWM 操作模式的详细说明，此处的快速预览为后面的寄存器说明提供了背景信息。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

MCPWM 包括 3 个通道，其中每个通道控制一对输出，这些输出进而控制某些片外操作，如电动机中的一组线圈。每个通道都配有一个定时器 / 计数器 （TC）寄存器，它通过处理器时钟 （定时器模式）或输入引脚 （计数器模式）进行递增计数。

每个通道都有一个限值寄存器与 TC 值进行比较，当出现匹配时， TC 将通过以下两种方式之一进行“再循环”。在边缘对齐模式中，TC 将复位为 0，而在中心对齐模式中，TC 将在匹配时切换到如下状态：即在每个处理器时钟或输入引脚跳变时 TC 进行递减计数，直至到达 0 为止，然后 TC 又重新开始向上计数。

每个通道还配有一个匹配寄存器，用于保留小于限值寄存器的值。在边缘对齐模式中，只要 TC 与匹配寄存器或限值寄存器相匹配，通道输出就会进行切换；而在中心对齐模式中，只有当 TC 与匹配寄存器相匹配时，通道输出才会切换。

因此限值寄存器用于控制输出周期，而匹配寄存器用于控制在每种状态下输出在每一周期所花费的时间。如果输出被积分为电压，则限值寄存器中的较小值将使波动最小化，使 MCPWM 能够控制以高速运行的设备。

在限值寄存器中使用较小值的不利因素就在于它们会减少匹配寄存器控制的占空比的分辨率。如果限值寄存器中的值为 8，则匹配寄存器只能选择 0%、 12.5%、 25%、…、 87.5% 或 100% 的占空比。通常，匹配值中的每一步的分辨率是 1 除以限值。

分辨率与周期 / 频率之间的制衡关系是脉冲宽度调制器设计所固有的一种关系。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

30.5.1 框图

图 93. MCPWM 框图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

30.6 引脚说明

表 686 列出了 MCPWM 引脚。

表 686. MOTOCON PWM 引脚描述

函数名称	类型	描述
MCOA0/1/2	O	通道 0、 1 和 2 的输出 A
MCOB0/1/2	O	通道 0、 1 和 2 的输出 B
$\overline{\text{MCABORT}}$	I	低电平有效紧急停机
MCI0/1/2	I	通道 0、 1 和 2 的输入

30.7 寄存器描述

控制寄存器和中断寄存器具有单独的读取、设置和清除地址。读取这类寄存器的读取地址（如 MCCON）会产生寄存器位状态。在将 1 写入到设置地址 （如 MCCON_SET）时会设置寄存器位，在将 1 写入到清除地址 （如 MCCON_CLR）会清除寄存器位。

捕获寄存器 （MCCAP）是只读寄存器，并且只写 MCCAP_CLR 地址可用于清除一个或多个位。所有其他 MCPWM 寄存器（MCTIM、MCPER、MCPW、MCDEADTIME 和 MCCP）均为普通的读写寄存器。

表 687. 寄存器简介：电动机控制脉冲宽度调制器 (MCPWM) （基址 0x400A 0000）

名称	访问类型	地址偏移	描述	复位值	参考
CON	RO	0x000	PWM 控制读取地址	0	表 688
CON_SET	WO	0x004	PWM 控制设置地址	-	表 689
CON_CLR	WO	0x008	PWM 控制清除地址	-	表 690
CAPCON	RO	0x00C	捕获控制读取地址	0	表 691
CAPCON_SET	WO	0x010	捕获控制设置地址	-	表 692
CAPCON_CLR	WO	0x014	事件控制清除地址	-	表 693
TC0	R/W	0x018	定时器计数器寄存器，通道 0	0	表 694
TC1	R/W	0x01C	定时器计数器寄存器，通道 1	0	表 694
TC2	R/W	0x020	定时器计数器寄存器，通道 2	0	表 694
LIM0	R/W	0x024	限值寄存器，通道 0	0xFFFF FFFF	表 695
LIM1	R/W	0x028	限值寄存器，通道 1	0xFFFF FFFF	表 695
LIM2	R/W	0x02C	限值寄存器，通道 2	0xFFFF FFFF	表 695
MAT0	R/W	0x030	匹配寄存器，通道 0	0xFFFF FFFF	表 696
MAT1	R/W	0x034	匹配寄存器，通道 1	0xFFFF FFFF	表 696
MAT2	R/W	0x038	匹配寄存器，通道 2	0xFFFF FFFF	表 696
DT	R/W	0x03C	死区寄存器	0x3FFF FFFF	表 697
MCCP	R/W	0x040	通信模式寄存器	0	表 698
CAP0	RO	0x044	捕获寄存器，通道 0	0	表 699
CAP1	RO	0x048	捕获寄存器，通道 1	0	表 699
CAP2	RO	0x04C	捕获寄存器，通道 2	0	表 699
INTEN	RO	0x050	中断使能读取地址	0	表 701
INTEN_SET	WO	0x054	中断使能设置地址	-	表 702

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 687. 寄存器简介：电动机控制脉冲宽度调制器 (MCPWM) （基址 0x400A 0000）*（续）*

名称	访问类型	地址偏移	描述	复位值	参考
INTEN_CLR	WO	0x058	中断使能清除地址	-	表 703
CNTCON	RO	0x05C	计数控制读取地址	0	表 704
CNTCON_SET	WO	0x060	计数控制设置地址	-	表 705
CNTCON_CLR	WO	0x064	计数控制清除地址	-	表 706
INTF	RO	0x068	中断标志读取地址	0	表 707
INTF_SET	WO	0x06C	中断标志设置地址	-	表 708
INTF_CLR	WO	0x070	中断标志清除地址	-	表 709
CAP_CLR	WO	0x074	捕获清除地址	-	表 710

30.7.1 MCPWM 控制寄存器

30.7.1.1 MCPWM 控制读取地址

CON 寄存器用于控制 PWM 所有通道的操作。该地址为只读地址，但通过对地址 CON_SET 和 CON_CLR 执行写入操作可修改该基础寄存器。

表 688. MCPWM 控制读取地址 (CON - 0x400A 0000) 位描述

位	符号	值	描述	复位值
0	RUN0		停止 / 启动定时器通道 0。	0
		0	停止。	
		1	运行。	
1	CENTER0		通道 0 的边缘 / 中心对齐操作。	0
		0	边缘对齐。	
		1	中心对齐。	
2	POLA0		选择 MCOA0 和 MCOB0 引脚的极性。	0
		0	失效状态为低电平，激活状态为高电平。	
		1	失效状态为高电平，激活状态为低电平。	
3	DTE0		控制通道 0 的死区功能。	0
		0	死区处于禁用状态。	
		1	死区处于使能状态。	
4	DISUP0		使能 / 禁用通道 0 的功能寄存器更新 （参见第 30.8.2 节）。	0
		0	将在每个 PWM 周期结束时通过写入寄存器来更新功能寄存器。	
		1	只要定时器在运行，功能寄存器将保持不变。	
7:5	-	-	保留。	
8	RUN1		停止 / 启动定时器通道 1。	0
		0	停止。	
		1	运行。	
9	CENTER1		通道 1 的边缘 / 中心对齐操作。	0
		0	边缘对齐。	
		1	中心对齐。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 688. MCPWM 控制读取地址 (CON - 0x400A 0000) 位描述 *（续）*

位	符号	值	描述	复位值
10	POLA1		选择 MCOA1 和 MCOB1 引脚的极性。	0
		0	失效状态为低电平，激活状态为高电平。	
		1	失效状态为高电平，激活状态为低电平。	
11	DTE1		控制通道 1 的死区功能。	0
		0	死区处于禁用状态。	
		1	死区处于使能状态。	
12	DISUP1		使能 / 禁用通道 1 的功能寄存器更新 （参见第 30.8.2 节）。	0
		0	将在每个 PWM 周期结束时通过写入寄存器来更新功能寄存器。	
		1	只要定时器在运行，功能寄存器将保持不变。	
15:13	-	-	保留。	0
16	RUN2		停止 / 启动定时器通道 2。	0
		0	停止。	
		1	运行。	
17	CENTER2		通道 2 的边缘 / 中心对齐操作。	0
		0	边缘对齐。	
		1	中心对齐。	
18	POLA2		选择 MCOA2 和 MCOB2 引脚的极性。	0
		0	失效状态为低电平，激活状态为高电平。	
		1	失效状态为高电平，激活状态为低电平。	
19	DTE2		控制通道 1 的死区功能。	0
		0	死区处于禁用状态。	
		1	死区处于使能状态。	
20	DISUP2		使能 / 禁用通道 2 的功能寄存器更新 （参见第 30.8.2 节）。	0
		0	将在每个 PWM 周期结束时通过写入寄存器来更新功能寄存器。	
		1	只要定时器在运行，功能寄存器将保持不变。	
28:21	-	-	保留。	
29	INVBDC		控制所有 3 个通道的 MCOB 输出的极性。该位通常仅在 3 相直流模式下设置为 1。	
		0	MCOB 输出与 MCOA 输出具有相反的极性 （死区除外）。	
		1	MCOB 输出与 MCOA 输出具有基本相同的极性。（参见第 30.8.6 节）	
30	ACMODE		3 相交流模式选择 （参见第 30.8.7 节）。	0
		0	3 相交流模式关闭：每个 PWM 通道使用其自己的定时器计数器和周期寄存器。	
		1	3 相交流模式开启：所有 PWM 通道使用通道为 0 的定时器寄存器和周期寄存器。	
31	DCMODE		3 相直流模式选择 （参见第 30.8.6 节）。	0
		0	3 相直流模式关闭：PWM 通道是相互独立的通道 （除非位 ACMODE 等于 1）	
		1	3 相直流模式开启：内部 MCOA0 输出将通过 CP 寄存器（即掩码）来传送到所有六个 PWM 输出。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

30.7.1.2 MCPWM 控制设置地址

将 1 写入到该只写地址中，会在 MCCON 中设置相应的位。

表 689. MCPWM 控制设置地址 (CON_SET - 0x400A 0004) 位描述

位	符号	描述	复位值
0	RUN0_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
1	CENTER0_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
2	POLA0_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
3	DTE0_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
4	DISUP0_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
7:5	-	写入 1，会在 CON 寄存器中设置相应的位。	-
8	RUN1_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
9	CENTER1_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
10	POLA1_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
11	DTE1_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
12	DISUP1_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
15:13	-	写入 1，会在 CON 寄存器中设置相应的位。	-
16	RUN2_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
17	CENTER2_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
18	POLA2_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
19	DTE2_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
20	DISUP2_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
28:21	-	写入 1，会在 CON 寄存器中设置相应的位。	-
29	INVBDC_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
30	ACMODE_SET	写入 1，会在 CON 寄存器中设置相应的位。	-
31	DCMODE_SET	写入 1，会在 CON 寄存器中设置相应的位。	-

30.7.1.3 MCPWM 控制清除地址

将 1 写入到该只写地址中，会在 CON 中清除相应的位。

表 690. MCPWM 控制清除地址 (CON_CLR - 0x400A 0008) 位描述

位	符号	描述	复位值
0	RUN0_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
1	CENTER0_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
2	POLA0_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
3	DTE0_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
4	DISUP0_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
7:5	-	写入 1，会清除 CON 寄存器中的相应位。	-
8	RUN1_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
9	CENTER1_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
10	POLA1_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
11	DTE1_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
12	DISUP1_CLR	写入 1，会清除 CON 寄存器中的相应位。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 690. MCPWM 控制清除地址 (CON_CLR - 0x400A 0008) 位描述 *（续）*

位	符号	描述	复位值
15:13	-	写入 1，会清除 CON 寄存器中的相应位。	-
16	RUN2_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
17	CENTER2_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
18	POLA2_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
19	DTE2_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
20	DISUP2_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
28:21	-	写入 1，会清除 CON 寄存器中的相应位。	-
29	INVBDC_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
30	ACMOD_CLR	写入 1，会清除 CON 寄存器中的相应位。	-
31	DCMODE_CLR	写入 1，会清除 CON 寄存器中的相应位。	

30.7.2 PWM 捕获控制寄存器

30.7.2.1 MCPWM 捕获控制读取地址

MCCAPCON 寄存器用于控制对所有 MCPWM 通道的 MCI0-2 输入的事件检测。使用三个 MCI 输入中的任何一个，可以在任意通道或所有三个通道中启动捕获事件。该地址为只读地址，但通过对地址 CAPCON_SET 和 CAPCON_CLR 执行写入操作可以修改该基础寄存器。

表 691. MCPWM 捕获控制读取地址 (CAPCON - 0x400A 000C) 位描述

位	符号	描述	复位值
0	CAP0MCI0_RE	当该位为 1 时，会在 MCI0 的上升沿上使能通道 0 捕获事件。	0
1	CAP0MCI0_FE	当该位为 1 时，会在 MCI0 的下降沿上使能通道 0 捕获事件。	0
2	CAP0MCI1_RE	当该位为 1 时，会在 MCI1 的上升沿上使能通道 0 捕获事件。	0
3	CAP0MCI1_FE	当该位为 1 时，会在 MCI1 的下降沿上使能通道 0 捕获事件。	0
4	CAP0MCI2_RE	当该位为 1 时，会在 MCI2 的上升沿上使能通道 0 捕获事件。	0
5	CAP0MCI2_FE	当该位为 1 时，会在 MCI2 的下降沿上使能通道 0 捕获事件。	0
6	CAP1MCI0_RE	当该位为 1 时，会在 MCI0 的上升沿上使能通道 1 捕获事件。	0
7	CAP1MCI0_FE	当该位为 1 时，会在 MCI0 的下降沿上使能通道 1 捕获事件。	0
8	CAP1MCI1_RE	当该位为 1 时，会在 MCI1 的上升沿上使能通道 1 捕获事件。	0
9	CAP1MCI1_FE	当该位为 1 时，会在 MCI1 的下降沿上使能通道 1 捕获事件。	0
10	CAP1MCI2_RE	当该位为 1 时，会在 MCI2 的上升沿上使能通道 1 捕获事件。	0
11	CAP1MCI2_FE	当该位为 1 时，会在 MCI2 的下降沿上使能通道 1 捕获事件。	0
12	CAP2MCI0_RE	当该位为 1 时，会在 MCI0 的上升沿上使能通道 2 捕获事件。	0
13	CAP2MCI0_FE	当该位为 1 时，会在 MCI0 的下降沿上使能通道 2 捕获事件。	0
14	CAP2MCI1_RE	当该位为 1 时，会在 MCI1 的上升沿上使能通道 2 捕获事件。	0
15	CAP2MCI1_FE	当该位为 1 时，会在 MCI1 的下降沿上使能通道 2 捕获事件。	0
16	CAP2MCI2_RE	当该位为 1 时，会在 MCI2 的上升沿上使能通道 2 捕获事件。	0
17	CAP2MCI2_FE	当该位为 1 时，会在 MCI2 的下降沿上使能通道 2 捕获事件。	0
18	RT0	如果该位为 1， TC0 将通过通道 0 捕获事件来复位。	0
19	RT1	如果该位为 1， TC1 将通过通道 1 捕获事件来复位。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 691. MCPWM 捕获控制读取地址 (CAPCON - 0x400A 000C) 位描述 *（续）*

位	符号	描述	复位值
20	RT2	如果该位为 1， TC2 将通过通道 2 捕获事件来复位。	0
21	HNFCAP0	硬件噪音滤波器：如果该位为 1，通道 0 捕获事件将会出现延迟，如第 30.8.4 节所述。	0
22	HNFCAP1	硬件噪音滤波器：如果该位为 1，通道 1 捕获事件将会出现延迟，如第 30.8.4 节所述。	0
23	HNFCAP2	硬件噪音滤波器：如果该位为 1，通道 2 捕获事件将会出现延迟，如第 30.8.4 节所述。	0
31:24	-	保留。	-

30.7.2.2 MCPWM 捕获控制设置地址

将 1 写入到该只写地址中，会在 CAPCON 中设置相应的位。

表 692. MCPWM 捕获控制设置地址 (CAPCON_SET - 0x400A 0010) 位描述

位	符号	描述	复位值
0	CAP0MCI0_RE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
1	CAP0MCI0_FE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
2	CAP0MCI1_RE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
3	CAP0MCI1_FE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
4	CAP0MCI2_RE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
5	CAP0MCI2_FE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
6	CAP1MCI0_RE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
7	CAP1MCI0_FE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
8	CAP1MCI1_RE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
9	CAP1MCI1_FE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
10	CAP1MCI2_RE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
11	CAP1MCI2_FE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
12	CAP2MCI0_RE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
13	CAP2MCI0_FE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
14	CAP2MCI1_RE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
15	CAP2MCI1_FE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
16	CAP2MCI2_RE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
17	CAP2MCI2_FE_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
18	RT0_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
19	RT1_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
20	RT2_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
21	HNFCAP0_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
22	HNFCAP1_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
23	HNFCAP2_SET	写入 1，会在 CAPCON 寄存器中设置相应的位。	-
31:24	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

30.7.2.3 MCPWM 捕获控制清除地址

将 1 写入到该只写地址中，会清除 MCCAPCON 中的相应位。

表 693. MCPWM 捕获控制清除寄存器 （CAPCON_CLR - 地址 0x400A 0014）位描述

位	符号	描述	复位值
0	CAP0MCI0_RE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
1	CAP0MCI0_FE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
2	CAP0MCI1_RE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
3	CAP0MCI1_FE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
4	CAP0MCI2_RE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
5	CAP0MCI2_FE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
6	CAP1MCI0_RE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
7	CAP1MCI0_FE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
8	CAP1MCI1_RE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
9	CAP1MCI1_FE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
10	CAP1MCI2_RE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
11	CAP1MCI2_FE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
12	CAP2MCI0_RE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
13	CAP2MCI0_FE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
14	CAP2MCI1_RE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
15	CAP2MCI1_FE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
16	CAP2MCI2_RE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
17	CAP2MCI2_FE_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
18	RT0_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
19	RT1_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
20	RT2_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
21	HNFCAP0_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
22	HNFCAP1_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
23	HNFCAP2_CLR	写入 1，会清除 CAPCON 寄存器中的相应位。	-
31:24	-	保留。	-

30.7.3 MCPWM 定时器 / 计数器 0-2 寄存器

这些寄存器将保留通道 0-2 的 32 位计数器 / 定时器的当前值。每一值会在每个 PCLK 时钟或 CNTCON 所选的 MCI0-2 引脚边缘上递增。定时器 / 计数器会从 0 开始向上计数，直至到达其相应 PER 寄存器中的值为止（或对 CON_CLR 执行写入操作来停止该定时器 / 计数器运行）。

任何时候都可读取 TC 寄存器。如果要对 TC 寄存器执行写入操作，则必须停止其通道运行。如果无法停止，则不能执行写入操作，并且也不会产生异常。

表 694. MCPWM 定时器 / 计数器 0 至 2 寄存器（TC - 0x400A 0018 (TC0)、0x400A 001C (TC1)、0x400A 0020) (TC2)）位描述

位	符号	描述	复位值
31:0	MCTC	定时器 / 计数器值。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

30.7.4 MCPWM 限值 0-2 寄存器

这些寄存器将保留定时器 / 计数器 0-2 的限值。当定时器 / 计数器到达其相应的限值时：1) 在边缘对齐模式中，它会复位，从 0 开始计数； 2) 在中心对齐模式中，它开始向下计数直至到 0 为止，然后又开始向上计数。

如果 CON 中的通道 CENTER 位为 0，系统会选择边缘对齐模式，并且在 TC 与 LIM 相匹配时，会将通道 A 输出从有源切换至失效状态。如果 CON 中的通道 CENTER 位和 DTE 位均为 0，则该匹配会同时将通道 B 输出从无源切换至激活状态。

如果通道的 CENTER 位为 0，但 DTE 位为 1，则该匹配会启动通道死区计数器开始计数，当死区计数器结束计数时，通道 B 输出会从无源切换至激活状态。

在中心对齐模式中，当通道的 TC 与 LIM 寄存器相匹配时，不会对其 A 和 B 输出产生任何效应。

在对限值寄存器或匹配寄存器 (30.7.5) 执行写操作时会加载写入寄存器，并且如果该通道已停止运行，该寄存器还会加载操作寄存器以与 TC 进行比较。如果该通道在运行，并且 CON 中的禁用更新位为 0，则会从写入寄存器中加载操作寄存器：1) 在边缘对齐模式中，当 TC 与操作限值寄存器相匹配时； 2) 在中心对齐模式中，当 TC 向下计数至 0 时。如果该通道在运行，并且禁用更新位为 1，则不会从写入寄存器中加载操作寄存器直至软件终止了通道运行。

读取 LIM 地址时，始终会返回操作值。

表 695. MCPWM 限值 0 至 2 寄存器 （LIM - 0x400A 0024 (LIM0)、 0x400A 0028 (LIM1) 和 0x400A 002C (LIM2)）位描述

位	符号	描述	复位值
31:0	MCLIM	限值。	0xFFFF FFFF

注：在定时器模式中，通道的调制 MCO 输出的周期将由其限值寄存器确定，并且周期开始时的脉冲宽度由其匹配寄存器确定。如果这符合您的思维方式，请考虑将限值寄存器作为周期寄存器，将匹配寄存器作为脉冲宽度寄存器。

30.7.5 MCPWM 匹配 0-2 寄存器

这些寄存器还为限值寄存器提供了写入和操作版本（如上所述），并且操作寄存器还将与通道的 TC 进行比较。有关读取和写入限值寄存器和匹配寄存器的详细信息，请参见上述 30.7.4。

匹配寄存器和限值寄存器用于控制 MCO0-2 输出。如果匹配寄存器要对其通道操作产生任何作用，它必须具有小于相应限值寄存器的值。

表 696. MCPWM 匹配 0 至 2 寄存器（MAT - 地址 0x400A 0030 (MAT0)、0x400A 0034 (MAT1) 和 0x400A 0038 (MAT2)）位描述

位	符号	描述	复位值
31:0	MCMAT	匹配值	0xFFFF FFFF

30.7.5.1 边缘对齐模式中的匹配寄存器

如果 CON 中的通道 CENTER 位为 0，系统会选择边缘对齐模式，并且在 TC 与 MAT 相匹配时，会将通道 A 输出从有源切换至失效状态。如果 CON 中的通道 CENTER 位和 DTE 位均为 0，则该匹配会同时将通道 B 输出从无源切换至激活状态。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

如果通道的 CENTER 位为 0，但 DTE 位为 1，则该匹配会启动通道死区计数器开始计数，当死区计数器结束计数时，通道 B 输出会从无源切换至激活状态。

30.7.5.2 中心对齐模式中的匹配寄存器

如果 CON 中的通道 CENTER 位为 1，系统会选择中心对齐模式。在 TC 进行递增计数时，TC 与 MAT 之间进行的匹配会将通道 B 输出从有源切换至失效状态，而在 TC 进行递减计数时， TC 与 MAT 之间的匹配则会将通道 A 输出从有源切换至失效状态。如果 CON 中的通道 CENTER 位为 1，但 DTE 位为 0 时，则匹配会同时按相反的方向来切换通道中的其他输出。

如果通道的 CENTER 位和 DTE 位均为 1，则在 TC 与 MAT 相匹配时会启动通道死区计数器开始计数，当死区计数器结束计数时，如果 TC 在匹配时开始向上计数则通道 B 输出将从无源切换至激活状态，如果 TC 在匹配时开始向下计数则通道 A 输出将从无源切换至激活状态。

30.7.5.3 0 和 100% 的占空比

如果要在“B 有源、A 无源”状态下锁定通道的 MCO 输出，则必须在其匹配寄存器中写入一个高于其限值寄存器的值。这样就永远不会出现匹配了。

如果要在“A 有源、B 无源”的相反状态下锁定通道的 MCO 输出，只需在其匹配寄存器中写入 0 即可。

30.7.6 MCPWM 死区寄存器

该寄存器将为三个通道保留死区值。如果 CON 中的通道 DTE 位为 1，则会使能其死区计数器，当其通道的一个输出从有源变为失效状态时，该计数器将从该值开始向下计数。当死区计数器计数至 0 时，该通道会将其他输出从无源变为激活状态。

使用死区功能的目的是使功率晶体管（如电动机控制应用中的 A 和 B 输出驱动的功率晶体管）完全关闭的时间比其开启的时间长。如果同时开启 A 和 B 晶体管电源，则浪费的、有损坏性的电流将通过晶体管在电源轨之间流动。在这类应用中，应对死区寄存器进行编程，使用的 PCLK 周期数为大于或等于晶体管的最大关闭时间减去其最小的开启时间。

表 697. MCPWM 死区寄存器 （DT - 地址 0x400A 003C）位描述

位	符号	描述	复位值
9:0	DT0	通道 0 的死区。[1]	0x3FF
19:10	DT1	通道 1 的死区。[2]	0x3FF
29:20	DT2	通道 2 的死区。[2]	0x3FF
31:30	-	保留	

[1] 如果 ACMODE 为 1，则系统会选择交流模式，该字段用于控制所有三个通道的死区。

[2] 如果 ACMODE 为 0。

30.7.7 MCPWM 通信模式寄存器

该寄存器仅在直流模式下使用。内部 MCOA0 信号将在该寄存器中的位控制下传送到所有六个输出引脚或任何一个引脚中。与匹配寄存器和限值寄存器一样，此寄存器具有写入和操作版本。有关该主题的更多信息，请参见 30.7.4 和 30.8.2。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 698. MCPWM 通信模式寄存器 （CP - 地址 0x400A 0040）位描述

位	符号	值	描述	复位值
0	CCPA0		通信模式输出 A，通道 0。	0
		0	MCOA0 失效状态。	
		1	内部 MCOA0。	
1	CCPB0		通信模式输出 B，通道 0。	0
		0	MCOB0 失效状态。	
		1	MCOB0 将跟踪内部 MCOA0。	
2	CCPA1		通信模式输出 A，通道 1。	0
		0	MCOA1 失效状态。	
		1	MCOA1 将跟踪内部 MCOA0。	
3	CCPB1		通信模式输出 B，通道 1。	0
		0	MCOB1 失效状态。	
		1	MCOB1 将跟踪内部 MCOA0。	
4	CCPA2		通信模式输出 A，通道 2。	0
		0	MCOA2 失效状态。	
		1	MCOA2 将跟踪内部 MCOA0。	
5	CCPB2		通信模式输出 B，通道 2。	0
		0	MCOB2 失效状态。	
		1	MCOB2 将跟踪内部 MCOA0。	
31:6	-		保留。	

30.7.8 MCPWM 捕获读取地址

CAPCON 寄存器 (表 691) 允许软件选择任何 MCI0-2 输入上的任意边缘作为每个通道的捕获事件。当发生通道捕获事件时，该通道的当前 TC 值会存储在其只读捕获寄存器中。这些地址为只读地址，但对 CAP_CLR 地址执行写入操作可以清除该基础寄存器。

表 699. MCPWM 捕获读取地址（CAP - 0x400A 0044 (CAP0)、0x400A 0048 (CAP1) 和 0x400A 004C 9CAP2)）位描述

位	符号	描述	复位值
31:0	CAP	捕获事件下的当前 TC 值。	0x0000 00 00

30.7.9 MCPWM 中断寄存器

电动机控制 PWM 模块包含了下列中断源：

表 700. 电机控制 PWM 中断

符号	描述
ILIM0/1/2	通道 0、 1 和 2 的限制中断。
IMAT0/1/2	通道 0、 1 和 2 的匹配中断。
ICAP0/1/2	通道 0、 1 和 2 的捕获中断。
ABORT	紧急停机中断

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

30.7.9.1 MCPWM 中断使能读取地址

INTEN 寄存器用于控制将使能哪些 MCPWM 中断。该地址为只读地址，但对地址 INTEN_SET 和 INTEN_CLR 执行写入操作可以修改该基础寄存器。

表 701. MCPWM 中断使能读取地址 (INTEN - 0x400A 0050) 位描述

位	符号	值	描述	复位值
0	ILIM0		通道 0 的限制中断。	0
		0	中断处于禁用状态。	
		1	中断处于使能状态。	
1	IMAT0		通道 0 的匹配中断。	0
		0	中断处于禁用状态。	
		1	中断处于使能状态。	
2	ICAP0		通道 0 的捕获中断。	0
		0	中断处于禁用状态。	
		1	中断处于使能状态。	
3	-		保留。	-
4	ILIM1		通道 1 的限制中断。	0
		0	中断处于禁用状态。	
		1	中断处于使能状态。	
5	IMAT1		通道 1 的匹配中断。	0
		0	中断处于禁用状态。	
		1	中断处于使能状态。	
6	ICAP1		通道 1 的捕获中断。	0
		0	中断处于禁用状态。	
		1	中断处于使能状态。	
7	-		保留。	-
8	ILIM2		通道 2 的限制中断。	0
		0	中断处于禁用状态。	
		1	中断处于使能状态。	
9	IMAT2		通道 2 的匹配中断。	0
		0	中断处于禁用状态。	
		1	中断处于使能状态。	
10	ICAP2		通道 2 的捕获中断。	0
		0	中断处于禁用状态。	
		1	中断处于使能状态。	
14:11	-		保留。	-
15	ABORT		紧急停机中断。	0
		0	中断处于禁用状态。	
		1	中断处于使能状态。	
31:16	-		保留。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

30.7.9.2 MCPWM 中断使能设置地址

将 1 写入到该只写地址中，会在 INTEN 中设置相应的位，从而使能中断。

表 702. MCPWM 中断使能设置寄存器 （INTEN_SET - 地址 0x400A 0054）位描述

位	符号	描述	复位值
0	ILIM0_SET	写入 1，会在 INTEN 中设置相应的位，从而使能中断。	-
1	IMAT0_SET	写入 1，会在 INTEN 中设置相应的位，从而使能中断。	-
2	ICAP0_SET	写入 1，会在 INTEN 中设置相应的位，从而使能中断。	-
3	-	保留。	-
4	ILIM1_SET	写入 1，会在 INTEN 中设置相应的位，从而使能中断。	-
5	IMAT1_SET	写入 1，会在 INTEN 中设置相应的位，从而使能中断。	-
6	ICAP1_SET	写入 1，会在 INTEN 中设置相应的位，从而使能中断。	-
7	-	保留。	-
9	ILIM2_SET	写入 1，会在 INTEN 中设置相应的位，从而使能中断。	-
10	IMAT2_SET	写入 1，会在 INTEN 中设置相应的位，从而使能中断。	-
11	ICAP2_SET	写入 1，会在 INTEN 中设置相应的位，从而使能中断。	-
14:12	-	保留。	-
15	ABORT_SET	写入 1，会在 INTEN 中设置相应的位，从而使能中断。	-
31:16	-	保留。	-

30.7.9.3 MCPWM 中断使能清除地址

将 1 写入到该只写地址中，会清除 INTEN 中的相应位，从而禁用中断。

表 703. PWM 中断使能清除寄存器 （INTEN_CLR - 地址 0x400A 0058）位描述

位	符号	描述	复位值
0	ILIM0_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
1	IMAT0_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
2	ICAP0_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
3	-	保留。	-
4	ILIM1_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
5	IMAT1_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
6	ICAP1_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
7	-	保留。	-
8	ILIM2_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
9	IMAT2_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
10	ICAP2_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
14:11	-	保留。	-
15	ABORT_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
31:16	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

30.7.10 MCPWM 计数控制寄存器

30.7.10.1 MCPWM 计数控制读取地址

CNTCON 寄存器用于控制 MCPWM 通道是在定时器模式中还是在计数器模式中，并且在计数器模式中，计数器是在所有三个 MCI 输入或任意一个输入的上升沿、下降沿还是上升沿和下降沿上计数递增的。如果选择了定时器模式，则计数器将基于 PCLK 时钟进行计数递增。

该地址为只读地址。如要设置或清除寄存器位，则将 1 写入到 CNTCON_SET 地址或 CNTCON_CLR 地址中。

表 704. MCPWM 计数控制读取地址 (CNTCON - 0x400A 005C) 位描述

位	符号	值	描述	复位值
0	TC0MCI0_RE		计数器 0 上升沿模式，通道 0。	0
		0	MCI0 的上升沿不会影响计数器 0。	
		1	如果 MODE0 为 1，则计数器 0 会沿着 MCI0 的上升沿进行递增计数。	
1	TC0MCI0_FE		计数器 0 下降沿模式，通道 0。	0
		0	MCI0 的下降沿不会影响计数器 0。	
		1	如果 MODE0 为 1，则计数器 0 会沿着 MCI0 的下降沿进行递增计数。	
2	TC0MCI1_RE		计数器 0 上升沿模式，通道 1。	0
		0	MCI1 的上升沿不会影响计数器 0。	
		1	如果 MODE0 为 1，则计数器 0 会沿着 MCI1 的上升沿进行递增计数。	
3	TC0MCI1_FE		计数器 0 下降沿模式，通道 1。	0
		0	MCI1 的下降沿不会影响计数器 0。	
		1	如果 MODE0 为 1，则计数器 0 会沿着 MCI1 的下降沿进行递增计数。	
4	TC0MCI2_RE		计数器 0 上升沿模式，通道 2。	0
		0	MCI0 的上升沿不会影响计数器 0。	
		1	如果 MODE0 为 1，则计数器 0 会沿着 MCI2 的上升沿进行递增计数。	
5	TC0MCI2_FE		计数器 0 下降沿模式，通道 2。	0
		0	MCI0 的下降沿不会影响计数器 0。	
		1	如果 MODE0 为 1，则计数器 0 会沿着 MCI2 的下降沿进行递增计数。	
6	TC1MCI0_RE		计数器 1 上升沿模式，通道 0。	0
		0	MCI0 的上升沿不会影响计数器 1。	
		1	如果 MODE1 为 1，则计数器 1 会沿着 MCI0 的上升沿进行递增计数。	
7	TC1MCI0_FE		计数器 1 下降沿模式，通道 0。	0
		0	MCI0 的下降沿不会影响计数器 1。	
		1	如果 MODE1 为 1，则计数器 1 会沿着 MCI0 的下降沿进行递增计数。	
8	TC1MCI1_RE		计数器 1 上升沿模式，通道 1。	0
		0	MCI1 的上升沿不会影响计数器 1。	
		1	如果 MODE1 为 1，则计数器 1 会沿着 MCI1 的上升沿进行递增计数。	
9	TC1MCI1_FE		计数器 1 下降沿模式，通道 1。	0
		0	MCI0 的下降沿不会影响计数器 1。	
		1	如果 MODE1 为 1，则计数器 1 会沿着 MCI1 的下降沿进行递增计数。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 704. MCPWM 计数控制读取地址 (CNTCON - 0x400A 005C) 位描述 *（续）*

位	符号	值	描述	复位值
10	TC1MCI2_RE		计数器 1 上升沿模式，通道 2。	0
		0	MCI2 的上升沿不会影响计数器 1。	
		1	如果 MODE1 为 1，则计数器 1 会沿着 MCI2 的上升沿进行递增计数。	
11	TC1MCI2_FE		计数器 1 下降沿模式，通道 2。	0
		0	MCI2 的下降沿不会影响计数器 1。	
		1	如果 MODE1 为 1，则计数器 1 会沿着 MCI2 的下降沿进行递增计数。	
12	TC2MCI0_RE		计数器 2 上升沿模式，通道 0。	0
		0	MCI0 的上升沿不会影响计数器 2。	
		1	如果 MODE2 为 1，则计数器 2 会沿着 MCI0 的上升沿进行递增计数。	
13	TC2MCI0_FE		计数器 2 下降沿模式，通道 0。	0
		0	MCI0 的下降沿不会影响计数器 2。	
		1	如果 MODE2 为 1，则计数器 2 会沿着 MCI0 的下降沿进行递增计数。	
14	TC2MCI1_RE		计数器 2 上升沿模式，通道 1。	0
		0	MCI1 的上升沿不会影响计数器 2。	
		1	如果 MODE2 为 1，则计数器 2 会沿着 MCI1 的上升沿进行递增计数。	
15	TC2MCI1_FE		计数器 2 下降沿模式，通道 1。	0
		0	MCI1 的下降沿不会影响计数器 2。	
		1	如果 MODE2 为 1，则计数器 2 会沿着 MCI1 的下降沿进行递增计数。	
16	TC2MCI2_RE		计数器 2 上升沿模式，通道 2。	0
		0	MCI2 的上升沿不会影响计数器 2。	
		1	如果 MODE2 为 1，则计数器 2 会沿着 MCI2 的上升沿进行递增计数。	
17	TC2MCI2_FE		计数器 2 下降沿模式，通道 2。	0
		0	MCI2 的下降沿不会影响计数器 2。	
		1	如果 MODE2 为 1，则计数器 2 会沿着 MCI2 的下降沿进行递增计数。	
28:18	-	-	保留。	-
29	CNTR0		通道 0 计数器 / 定时器模式。	0
		0	通道 0 处于定时器模式中。	
		1	通道 0 处于计数器模式中。	
30	CNTR1		通道 1 计数器 / 定时器模式。	0
		0	通道 1 处于定时器模式中。	
		1	通道 1 处于计数器模式中。	
31	CNTR2		通道 2 计数器 / 定时器模式。	0
		0	通道 2 处于定时器模式中。	
		1	通道 2 处于计数器模式中。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

30.7.10.2 MCPWM 计数控制设置地址

将 1 写入到该只写地址中，会在 CNTCON 中设置相应的位。

表 705. MCPWM 捕获控制设置地址 (CNTCON_SET - 0x400A 0060) 位描述

位	符号	描述	复位值
0	TC0MCI0_RE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
1	TC0MCI0_FE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
2	TC0MCI1_RE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
3	TC0MCI1_FE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
4	TC0MCI2_RE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
5	TC0MCI2_FE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
6	TC1MCI0_RE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
7	TC1MCI0_FE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
8	TC1MCI1_RE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
9	TC1MCI1_FE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
10	TC1MCI2_RE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
11	TC1MCI2_FE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
12	TC2MCI0_RE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
13	TC2MCI0_FE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
14	TC2MCI1_RE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
15	TC2MCI1_FE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
16	TC2MCI2_RE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
17	TC2MCI2_FE_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
28:18	-	保留。	
29	CNTR0_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
30	CNTR1_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-
31	CNTR2_SET	写入 1，会在 CNTCON 寄存器中设置相应的位。	-

30.7.10.3 MCPWM 计数控制清除地址

将 1 写入到该只写地址中，会清除 CNTCON 中的相应位。

表 706. MCPWM 计数控制清除地址 (CNTCON_CLR - 0x400A 0064) 位描述

位	符号	描述	复位值
0	TC0MCI0_RE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
1	TC0MCI0_FE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
2	TC0MCI1_RE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
3	TC0MCI1_FE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
4	TC0MCI2_RE	写入 1，会清除 CNTCON 寄存器中的相应位。	-
5	TC0MCI2_FE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
6	TC1MCI0_RE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
7	TC1MCI0_FE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
8	TC1MCI1_RE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
9	TC1MCI1_FE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
10	TC1MCI2_RE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 706. MCPWM 计数控制清除地址 (CNTCON_CLR - 0x400A 0064) 位描述 *（续）*

位	符号	描述	复位值
11	TC1MCI2_FE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
12	TC2MCI0_RE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
13	TC2MCI0_FE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
14	TC2MCI1_RE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
15	TC2MCI1_FE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
16	TC2MCI2_RE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
17	TC2MCI2_FE_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
28:18	-	保留。	
29	CNTR0_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
30	CNTR1_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-
31	CNTR2_CLR	写入 1，会清除 CNTCON 寄存器中的相应位。	-

30.7.11 MCPWM 中断标志寄存器

30.7.11.1 MCPWM 中断标志读取地址

INTF 寄存器包含了所有 MCPWM 中断标志，会在发生相应的硬件事件或将 1 写入到 INTF_SET 地址时设置这些标志。当该寄存器和 INTEN 中的相应位均为 1 时，MCPWM 会使向中断控制器模块发送的中断请求变为有效。该地址为只读地址，但通过将 1 写入到地址 INTF_SET 和 INTF_CLR 中可以修改该基础寄存器中的位。

表 707. MCPWM 中断标志读取地址 (INTF - 0x400A 0068) 位描述

位	符号	值	描述	复位值
0	ILIM0_F		通道 0 的限制中断标志。	0
		0	该中断源不可用于 MCPWM 中断请求。	
		1	如果 INTEN 中的相应位为 1，则 MCPWM 模块会使向中断控制器发送的中断请求变为有效。	
1	IMAT0_F		通道 0 的匹配中断标志。	0
		0	该中断源不可用于 MCPWM 中断请求。	
		1	如果 INTEN 中的相应位为 1，则 MCPWM 模块会使向中断控制器发送的中断请求变为有效。	
2	ICAP0_F		通道 0 的捕获中断标志。	0
		0	该中断源不可用于 MCPWM 中断请求。	
		1	如果 INTEN 中的相应位为 1，则 MCPWM 模块会使向中断控制器发送的中断请求变为有效。	
3	-		保留。	-
4	ILIM1_F		通道 1 的限制中断标志。	0
		0	该中断源不可用于 MCPWM 中断请求。	
		1	如果 INTEN 中的相应位为 1，则 MCPWM 模块会使向中断控制器发送的中断请求变为有效。	
5	IMAT1_F		通道 1 的匹配中断标志。	0
		0	该中断源不可用于 MCPWM 中断请求。	
		1	如果 INTEN 中的相应位为 1，则 MCPWM 模块会使向中断控制器发送的中断请求变为有效。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 707. MCPWM 中断标志读取地址 (INTF - 0x400A 0068) 位描述 *（续）*

位	符号	值	描述	复位值
6	ICAP1_F		通道 1 的捕获中断标志。	0
		0	该中断源不可用于 MCPWM 中断请求。	
		1	如果 INTEN 中的相应位为 1，则 MCPWM 模块会使向中断控制器发送的中断请求变为有效。	
7	-		保留。	-
8	ILIM2_F		通道 2 的限制中断标志。	0
		0	该中断源不可用于 MCPWM 中断请求。	
		1	如果 INTEN 中的相应位为 1，则 MCPWM 模块会使向中断控制器发送的中断请求变为有效。	
9	IMAT2_F		通道 2 的匹配中断标志。	0
		0	该中断源不可用于 MCPWM 中断请求。	
		1	如果 INTEN 中的相应位为 1，则 MCPWM 模块会使向中断控制器发送的中断请求变为有效。	
10	ICAP2_F		通道 2 的捕获中断标志。	0
		0	该中断源不可用于 MCPWM 中断请求。	
		1	如果 INTEN 中的相应位为 1，则 MCPWM 模块会使向中断控制器发送的中断请求变为有效。	
14:11	-		保留。	-
15	ABORT_F		紧急停机中断标志。	0
		0	该中断源不可用于 MCPWM 中断请求。	
		1	如果 INTEN 中的相应位为 1，则 MCPWM 模块会使向中断控制器发送的中断请求变为有效。	
31:16	-		保留。	-

30.7.11.2 MCPWM 中断标志设置地址

将 1 写入到该只写地址中，会在 INTF 中设置相应的位，从而可能模拟硬件中断。

表 708. MCPWM 中断标志设置地址 (INTF_SET - 0x400A 006C) 位描述

位	符号	描述	复位值
0	ILIM0_F_SET	写入 1，会在 INTF 寄存器中设置相应的位，从而可能模拟硬件中断。	-
1	IMAT0_F_SET	写入 1，会在 INTF 寄存器中设置相应的位，从而可能模拟硬件中断。	-
2	ICAP0_F_SET	写入 1，会在 INTF 寄存器中设置相应的位，从而可能模拟硬件中断。	-
3	-	保留。	-
4	ILIM1_F_SET	写入 1，会在 INTF 寄存器中设置相应的位，从而可能模拟硬件中断。	-
5	IMAT1_F_SET	写入 1，会在 INTF 寄存器中设置相应的位，从而可能模拟硬件中断。	-
6	ICAP1_F_SET	写入 1，会在 INTF 寄存器中设置相应的位，从而可能模拟硬件中断。	-
7	-	保留。	-
8	ILIM2_F_SET	写入 1，会在 INTF 寄存器中设置相应的位，从而可能模拟硬件中断。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 708. MCPWM 中断标志设置地址 (INTF_SET - 0x400A 006C) 位描述 *（续）*

位	符号	描述	复位值
9	IMAT2_F_SET	写入 1，会在 INTF 寄存器中设置相应的位，从而可能模拟硬件中断。	-
10	ICAP2_F_SET	写入 1，会在 INTF 寄存器中设置相应的位，从而可能模拟硬件中断。	-
14:11	-	保留。	-
15	ABORT_F_SET	写入 1，会在 INTF 寄存器中设置相应的位，从而可能模拟硬件中断。	-
31:16	-	保留。	-

30.7.11.3 MCPWM 中断标志清除地址

将 1 写入到该只写地址中，会在 INTF 中设置相应的位，从而清除相应的中断请求。这通常会在中断服务程序中完成。

表 709. MCPWM 中断标志清除地址 (INTF_CLR - 0x400A 0070) 位描述

位	符号	描述	复位值
0	ILIM0_F_CLR	写入 1，会清除 INTF 寄存器中的相应位，从而清除相应的中断请求。	-
1	IMAT0_F_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
2	ICAP0_F_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
3	-	保留。	-
4	ILIM1_F_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
5	IMAT1_F_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
6	ICAP1_F_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
7	-	保留。	-
8	ILIM2_F_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
9	IMAT2_F_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
10	ICAP2_F_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
14:11	-	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
15	ABORT_F_CLR	写入 1，会清除 INTEN 中的相应位，从而禁用中断。	-
31:16	-	保留。	-

30.7.12 MCPWM 捕获清除地址

将 1 写入到该只写地址中，会清除所选的 CAP 寄存器。

表 710. MCPWM 捕获清除地址 (CAP_CLR - 0x400A 0074) 位描述

位	符号	描述
0	CAP_CLR0	在将 1 写入到该位时会清除 CAP0 寄存器。
1	CAP_CLR1	在将 1 写入到该位时会清除 CAP1 寄存器。
2	CAP_CLR2	在将 1 写入到该位时会清除 CAP2 寄存器。
31:3	-	保留

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

30.8 功能说明

30.8.1 脉冲宽度调制

MCPWM 的每个通道都具有两个输出 A 和 B，它们可驱动一对晶体管来切换两个电源轨之间的控制点。大部分情况下两个输出具有相反的极性，但可以使能死区功能 （基于每个通道）来延迟两个信号从失效状态到激活状态的跃迁，以便两个晶体管永远不会同时开启。更普遍的观点认为，可以将每个输出对视为高电平、低电平和悬浮或上、下以及偏中心状态。

每个通道从有源和无源到高电平和低电平的映射是可编程的。在复位后，三个 A 输出将为无源 / 低电平， B 输出为有源 / 高电平。

MCPWM 可以执行边缘对齐和中心对齐脉冲宽度调制。

注：在定时器模式中，通道的调制 MCO 输出的周期将由其限值寄存器确定，并且周期开始时的脉冲宽度由其匹配寄存器确定。如果这符合您的思维方式，请考虑将限值寄存器作为周期寄存器，将匹配寄存器作为脉冲宽度寄存器。

无死区的边缘对齐 PWM

在该模式中，定时器 TC 将从 0 开始向上计数至 LIM 寄存器中的值。如图 94 所示， MCO 状态为 A 失效状态，直至 TC 与匹配寄存器相匹配，此时 MCO 将变为 A 激活状态。当 TC 与限值寄存器相匹配时， MCO 状态会变回至 A 失效状态，并使 TC 复位，并重新开始向上计数。

图 94. 边沿对齐的 PWM 波形，无死区且 POLA = 0

无死区的中心对齐 PWM

在该模式中，定时器 TC 会从 0 开始向上计数至 LIM 寄存器中的值，然后向下计数至 0，并这样重复计数。如图 95 所示，当定时器向上计数时， MCO 状态为 A 无源，直至 TC 与匹配寄存器相匹配，此时 MCO 将变为 A 有源。当 TC 与限值寄存器相匹配时，它开始向下计数。当 TC 在向下计数时与匹配寄存器相匹配， MCO 状态会变回至 A 失效状态。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 95. 中心对齐的 PWM 波形，无死区且 POLA = 0

死区计数器

当在 CON 中设置通道 DTE 位时，死区计数器会延迟两个 MCO 输出从无源到有源的跃迁。当通道 A 或 B 输出从有源变为失效状态时，死区计数器将从通道 DT 值（在 DT 寄存器中）开始向下计数至 0。其他输出从无源到有源的跳变会延迟，直至死区计数器达到 0。在死区期间， MCOA 和 MCOB 的输出电平均为失效状态。图 96 显示了有死区时在边缘对齐模式中的操作，图 97 显示了有死区时的中心对齐操作。

图 96. 边缘对齐的 PWM 波形，有死区且 POLA = 0

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 97. 中心对齐的 PWM 波形，有死区且 POLA = 0

30.8.2 影子寄存器和同步更新

限值寄存器、匹配寄存器和通信模式寄存器（LIM、MAT 和 CP）可作为寄存器对来实现，每个寄存器对将由写入寄存器和操作寄存器来组成。软件会对写入寄存器执行写入操作。操作寄存器用于控制每个通道的实际操作，并当 TC 开始从 0 向上计数时，操作寄存器会载入写入寄存器中的当前值。

在 CON 寄存器中设置通道 DISUP 位可以禁用功能寄存器更新。如果设置了 DISUP 位，则不会更新功能寄存器，直至软件停止了该通道运行为止。

如果软件在对通道的 LIM 或 MAT 寄存器执行写入操作时通道未运行，则系统会立即更新功能寄存器。

仅在 TC 寄存器的通道停止运行时，软件才能对 TC 寄存器执行写入操作。

30.8.3 紧急停机 (ABORT)

MCPWM 具有一个外部输入 $\overline{\text{MCABORT}}$。当此输入转为低电平时，所有六个 MCO 输出都将采取 A 失效状态，并生成 “终止” 中断 （如已使能）。输出仍锁定在失效状态中，直至 ABORT 中断标志被清除或终止中断被禁用。在 $\overline{\text{MCABORT}}$ 输入转入高电平之前，无法清除 ABORT 标志。

如果要清除 ABORT 标志，则必须将 1 写入到 INTF_CLR 寄存器的位 15 中。这会删除中断请求。将 1 写入到 INTEN_CLR 寄存器的位 15 中也可以禁用该中断。

30.8.4 捕获事件

每个 PWM 通道都会在输入信号跃迁时生成其 TC 快照。任何通道都可以将所有 MCI0-2 输入或任一输入的上升沿、下降沿或它们的任意组合作为 CAPCON 寄存器控制下的捕获事件。将会同步检测有关 PCLK 的输入上升沿或下降沿。

如果将 CAPCON 寄存器中的通道 HNF 位设置为使能噪音滤波，则 MCI 引脚上的所选边缘将会启动该通道的死区计数器，并且会使下面所述的捕获事件操作延迟执行，直至死区计数器计数至 0 为止。该功能专用于对霍尔传感器执行三相无刷直流电机控制。

通道捕获事件 （可能已被 HNF 延迟）会引发如下操作：

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- TC 的当前值将存储在捕获寄存器 (CAP) 中。
- 如果使能了通道捕获事件中断 （参见 表 701），则会设置捕获事件中断标志。
- 如果在 CAPCON 寄存器中设置了通道 RT 位，则会对捕获事件进行复位，输入事件与使通道 TC 与其 LIM 寄存器相匹配所具有的效应相同。这包括在边缘对齐模式中复位 TC 以及切换 MCO 引脚，如 30.7.4 和 30.8.1 所述。

30.8.5 外部事件计数 （计数器模式）

如果 CNTCON 中的通道 MODE 位为 1，则 MCI0-2 输入而非 PCLK 的上升沿和 / 或下降沿（同步检测）会使其 TC 递增计数。这不会影响 PWM 功能和捕获功能。

30.8.6 三相直流模式

在 CON 寄存器中设置 DCMODE 位可以选择三相直流模式。

在该模式中，内部 MCOA0 信号可以传送到任何或所有 MCO 输出中。每个 MCO 输出都将通过电流换向模式寄存器 CP 中的位来掩码，如果 CP 寄存器中的位为 0，则其输出引脚为输出 MCOA0 的失效状态提供了逻辑电平。关闭状态的极性将由 POLA0 位来确定。

在 CP 寄存器中含有 1 位的所有 MCO 输出均由内部 MCOA0 信号来控制。

当 CON 寄存器中的 INVBDC 位为 1 时，会反转三个 MCOB 输出引脚。该功能包含了桥接驱动程序，它可以为低端交换机提供低电平有效输入。

CP 寄存器将作为影子寄存器对来实现，以便在新的 PWM 周期开始时进行有效的通信模式更改。有关对此类寄存器进行读写操作的更多信息，请参见 30.7.4 和 30.8.2。

图 98 显示了 MCO 输出在三相直流模式中的波形采样。已将 CP 寄存器中的位 1 和位 3（与输出 MCOB1 和 MCOB0 相对应）设置为 0，以便掩码这些输出，并使其处于关闭状态。其逻辑电平将由 POLA0 位 （在该处， POLA0 等于 0 以便失效状态为逻辑低电平）来确定。将 INVBDC 位设置为 0 （将不反转逻辑电平）以便 B 输出与 A 输出具有相同的极性。请注意，该模式与其他模式的区别在于 MCOB 输出与 MCOA 输出的极性不相反。

在图 98 所示的情况中， CP 寄存器中的位 0、 2、 4 和 5 均设置为 1。这就意味着 MCOA1 及其通道 2 的两个 MCO 输出均执行 MCOA0 信号指令。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 98. 三相直流模式采样波形

30.8.7 三相交流模式

在 CON 寄存器中设置 ACMODE 位可以选择三相交流模式。

在该模式中，通道 0 的 TC 值会传送到所有通道中以便与其 MAT 寄存器进行比较。（将不使用 LIM1-2 寄存器。）

每个通道会将其 MAT 值与 TC0 进行比较来控制其 MCO 输出。

图 99 显示了六个 MCO 输出在三相交流模式中的波形采样。将所有三个通道的 POLA 位设置为 0，以便所有 MCO 输出的有效电平为高电平，而无源电平为低电平。每个通道都具有一个不同的 MAT 值与 TC0 值进行比较。在该模式中，所有三个通道的周期值都相同，并且它由 LIM0 来确定，并且会禁用死区模式。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 99. 三相交流模式采样波形，边缘对齐的 PWM 模式

30.8.8 中断

MCPWM 包括 10 个可能的中断源：

- 当任何通道的 TC 与其匹配寄存器匹配时。
- 当任何通道的 TC 与其限值寄存器匹配时。
- 当任何通道捕获其TC值将其置于捕获寄存器时（由于所选的边缘会出现在任何MCI0-2输入中）。
- 当所有三个通道的输出因 $\overline{\text{MCABORT}}$ 引脚转入低电平而强制进入失效状态时。

第 30.7.9 节“MCPWM 中断寄存器”说明了如何使能这些中断，而第 30.7.2 节“PWM 捕获控制寄存器”介绍了如何将 MCI0-2 输入的边沿缘映射到三个通道的捕获事件上。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

31.1 本章导读

所有 LPC43xx 器件上均可使用 QEI。

31.2 基本配置

QEI 的配置如下：

- 参见表 711 以了解时钟与电源控制。
- QEI 通过 QEI_RST （复位号 39）复位。
- QEI 中断连接到 “事件”路由器中的插槽 # 15。

表 711. QEI 时钟和电源控制

	基本时钟	分支时钟	工作频率
QEI 寄存器接口时钟和 QEI 外设时钟。	BASE_M4_CLK	CLK_M4_QEI	高达 204 MHz

31.3 功能要点

正交编码器接口 (QEI) 功能如下：

- 跟踪编码器位置。
- 根据方向进行递增 / 递减计数。
- 可对 2X 或 4X 位置计数进行编程。
- 使用内置定时器来捕获速度。
- 速度比较功能，可产生 “小于”中断。
- 使用 32 位寄存器来保存位置和速度。
- 三个位置比较寄存器，可产生中断。
- 用于记录转数的索引计数器。
- 索引比较寄存器，可产生中断。
- 可结合索引和位置中断来产生整个位移或局部旋转位移的中断。
- 带可编程编码器输入信号延迟的数字过滤器。
- 能够接收解码信号输入 （时钟和方向）。

31.4 简介

正交编码器，又名双通道增量式编码器，把角位移转换成两个脉冲信号。通过监控脉冲的数量和两个信号的相对相位，可以追踪位置、旋转方向和速度。另外还有第三个通道，或索引信号，可用于复位位置计数器。该正交编码器接口模块解码来自正交编码器轮的数字脉冲，进行位置的时间积分并确定旋转方向。另外，它还可捕获编码器轮的速度。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 100. 编码器接口框图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

31.5 引脚说明

表 712. QEI 引脚描述

函数名称	I/O	描述
QEI_A	I	用作输入到 “正交编码器接口” 的 “相位 A” (PhA)。
QEI_B	I	用作输入到 “正交编码器接口” 的 “相位 B” (PhB)。
QEI_IDX	I	用作输入到 “正交编码器接口” 的 “索引” (IDX)。

31.6 寄存器描述

表 713. 寄存器简介：QEI （基址 0x400C 6000）

名称	访问类型	地址偏移	描述	复位值	参考
控制寄存器					
CON	WO	0x000	控制寄存器	0	表 714
STAT	RO	0x004	编码器状态寄存器	0	表 715
CONF	R/W	0x008	配置寄存器	0x000F 0000	表 716
位置、索引和定时器寄存器					
POS	RO	0x00C	位置寄存器	0	表 717
MAXPOS	R/W	0x010	最大位置寄存器	0	表 718
CMPOS0	R/W	0x014	位置比较寄存器 0	0xFFFF FFFF	表 719
CMPOS1	R/W	0x018	位置比较寄存器 1	0xFFFF FFFF	表 720
CMPOS2	R/W	0x01C	位置比较寄存器 2	0xFFFF FFFF	表 721
INXCNT	RO	0x020	索引计数寄存器	0	表 722
INXCMP0	R/W	0x024	索引比较寄存器 0	0xFFFF FFFF	表 723
LOAD	R/W	0x028	速度定时器重新载入寄存器	0xFFFF FFFF	表 724
TIME	RO	0x02C	速度定时器寄存器	0xFFFF FFFF	表 725
VEL	RO	0x030	速度计数器寄存器	0	表 726
CAP	RO	0x034	速度捕获寄存器	0xFFFF FFFF	表 727
VELCOMP	R/W	0x038	速度比较寄存器	0	表 728
FILTERPHA	R/W	0x03C	输入相位 A 的数字过滤器寄存器 (QEI_A)	0	表 729
FILTERPHB	R/W	0x040	输入相位 B 的数字过滤器寄存器 (QEI_B)	0	表 730
FILTERINX	R/W	0x044	输入索引的数字过滤器寄存器 (QEI_IDX)	0	表 731
WINDOW	R/W	0x048	索引接受窗口寄存器	0x0000 0000	表 732
INXCMP1	R/W	0x04C	索引比较寄存器 1	0xFFFF FFFF	表 733
INXCMP2	R/W	0x050	索引比较寄存器 2	0xFFFF FFFF	表 734
中断寄存器					
IEC	WO	0xFD8	中断使能清除寄存器	0	表 735
IES	WO	0xFDC	中断使能设置寄存器	0	表 736
INTSTAT	RO	0xFE0	中断状态寄存器	0	表 737

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 713. 寄存器简介：QEI （基址 0x400C 6000）*（续）*

名称	访问类型	地址偏移	描述	复位值	参考
IE	RO	0xFE4	中断使能寄存器	0	表 738
CLR	WO	0xFE8	中断状态清除寄存器	0	表 739
SET	WO	0xFEC	中断状态设置寄存器	0	表 740

31.6.1 控制寄存器

31.6.1.1 QEI 控制寄存器

此寄存器包含控制 QEI 模块的位置和速度计数器的操作的位。

表 714. QEI 控制寄存器 （CON - 地址 0x400C 6000）位描述

位	符号	描述	复位值
0	RESP	复位位置计数器。如果设置 = 1，将位置计数器复位为全零。在位置计数器清除时自动清除。	0
1	RESPI	复位索引的位置计数器。如果设置 = 1，在出现索引脉冲时，将位置计数器复位为全零。在位置计数器清除时自动清除。	0
2	RESV	复位速度。如果设置 = 1，将速度计数器复位为全零并重新载入速度定时器。在速度计数器清除时自动清除。	0
3	RESI	复位索引计数器。如果设置 = 1，将索引计数器复位为全零。在索引计数器清除时自动清除。	0
31:4	-	保留	0

31.6.1.2 QEI 状态寄存器

此寄存器提供编码器接口的状态。

表 715. QEI 中断状态寄存器 （STAT - 地址 0x400C 6004）位描述

位	符号	描述	复位值
0	DIR	方向位。与 DIRINV 位结合指示向前或相反方向。参见表 743。	
31:1	-	保留	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

31.6.1.3 QEI 配置寄存器

此寄存器包含 QEI 模块的配置。

表 716. QEI 配置寄存器 （CONF - 地址 0x400C 6008）位描述

位	符号	描述	复位值
0	DIRINV	反转方向。 = 1 时，补足 DIR 位。	0
1	SIGMODE	信号模式。= 0 时， PhA 和 PhB 用作正交编码器的输入。 = 1 时， PhA 用作方向信号， PhB 用作时钟信号。	0
2	CAPMODE	捕获模式。= 0 时，只计数 PhA 边值 (2X)。= 1 时，PhA 和 PhB 边都计数 (4X)，分辨率增加而范围减小。	0
3	INVINX	反转索引。设置时，反转所感测到的索引输入。	0
4	CRESPI	连续复位索引的位置计数器。设置 = 1 时，如果索引脉冲出现在下一位置增加出，则将位置计数器复位为全零 （重新校准）。在位置计数器清除时自动清除。	0
15:5	-	保留	0
19:16	INXGATE	索引门控配置： INXGATE(19)=1 时，当 Pha=0 且 Phb=0 时传递索引，否则阻止。 INXGATE(18)=1 时，当 Pha=0 且 Phb=1 时传递索引，否则阻止。 INXGATE(17)=1 时，当 Pha=1 且 Phb=1 时传递索引，否则阻止。 INXGATE(16)=1 时，当 Pha=1 且 Phb=0 时传递索引，否则阻止。	1111
31:20	-	保留	0

31.6.2 位置、索引和定时器寄存器

31.6.2.1 QEI 位置寄存器

此寄存器包含编码器位置的当前值。在发生编码器计数时递增或递减，取决于旋转方向。

表 717. QEI 位置寄存器 （POS - 地址 0x400C 600C）位描述

位	符号	描述	复位值
31:0	POS	当前位置值。	0

31.6.2.2 QEI 最大位置寄存器

此寄存器包含编码器位置的最大值。向前旋转时，在位置寄存器超过此值时，位置寄存器复位为零。反向旋转时，在位置寄存器从零开始递减时，位置寄存器复位为此值。

表 718. QEI 最大位置寄存器 （MAXPOS - 地址 0x400C 6010）位描述

位	符号	描述	复位值
31:0	MAXPOS	最大位置值。	0

31.6.2.3 QEI 位置比较寄存器 0

此寄存器包含位置比较值。此值与位置寄存器的当前值进行比较。当比较值小于、等于或大于位置寄存器的当前值时，可以使能中断。

表 719. QEI 位置比较寄存器 0（CMPOS0 - 地址 0x400C 6014）位描述

位	符号	描述	复位值
31:0	PCMP0	位置比较值 0	0xFFFF FFFF

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

31.6.2.4 QEI 位置比较寄存器 1

此寄存器包含位置比较值。此值与位置寄存器的当前值进行比较。当比较值小于、等于或大于位置寄存器的当前值时，可以使能中断。

表 720. QEI 位置比较寄存器 1（CMPOS1 - 地址 0x400C 6018）位描述

位	符号	描述	复位值
31:0	PCMP1	位置比较值 1	0xFFFF FFFF

31.6.2.5 QEI 位置比较寄存器 2

此寄存器包含位置比较值。此值与位置寄存器的当前值进行比较。当比较值小于、等于或大于位置寄存器的当前值时，可以使能中断。

表 721. QEI 位置比较寄存器 2 （CMPOS2 - 地址 0x400C 601C）位描述

位	符号	描述	复位值
31:0	PCMP2	位置比较值 2	0xFFFF FFFF

31.6.2.6 QEI 索引计数寄存器

此寄存器包含编码器位置的当前值。在发生编码器计数时递增或递减，取决于旋转方向。

表 722. QEI 索引计数寄存器 （INXCNT - 地址 0x400C 6020）位描述

位	符号	描述	复位值
31:0	ENCPOS	当前编码器位置值。	0

31.6.2.7 QEI 索引比较寄存器 0

此寄存器包含索引比较值，此值与索引计数寄存器的当前值进行比较。当比较值小于、等于或大于索引计数寄存器的当前值时，可以使能中断。

表 723. QEI 索引比较寄存器 0 （INXCMP0 - 地址 0x400C 6024）位描述

位	符号	描述	复位值
31:0	ICMP0	索引比较值。	0xFFFF FFFF

31.6.2.8 QEI 定时器重新载入寄存器

此寄存器包含速度定时器的重新载入值。如果定时器 (QEITIME) 溢出或 RESV 位被断言，则将此值载入定时器 (QEITIME)。

表 724. QEI 定时器载入寄存器 （LOAD - 地址 0x400C 6028）位描述

位	符号	描述	复位值
31:0	VELLOAD	当前速度定时器载入值。	0xFFFF FFFF

31.6.2.9 QEI 定时器寄存器

此寄存器包含速度定时器的当前值。此定时器溢出时，速度计数器 (QEIVEL) 的值存储于速度捕获寄存器 (QEICAP) 中，速度计数器复位为零。定时器重新载入存储于速度重新载入寄存器 (QEILOAD) 中的值，并进行速度中断 (TIM_Int) 断言。

表 725. QEI 定时器寄存器 （TIME - 地址 0x400C 602C）位描述

位	符号	描述	复位值
31:0	VELVAL	当前速度定时器值。	0xFFFF FFFF

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

31.6.2.10 QEI 速度寄存器

此寄存器包含当前时间段内的速度脉冲运行计数。速度定时器 (QEITIME) 溢出时，寄存器的内容被捕获在速度捕获寄存器 (QEICAP) 中。捕获之后，此寄存器设置为零。速度复位位 (RESV) 被断言时，则此寄存器也复位。

表 726. QEI 速度寄存器 （VEL - 地址 0x400C 6030）位描述

位	符号	描述	复位值
31:0	VELPC	当前速度脉冲计数。	0x0

31.6.2.11 QEI 速度捕获寄存器

此寄存器包含编码器的最近测量速度，这与前一速度定时器期间计数的速度脉冲的数量相应。如果速度定时器溢出，则把当前速度计数锁存到该寄存器中。

表 727. QEI 速度捕获寄存器 （CAP - 地址 0x400C 6034）位描述

位	符号	描述	复位值
31:0	VELCAP	速度捕获值。	0xFFFF FFFF

31.6.2.12 QEI 速度比较寄存器

此寄存器包含速度比较值，此值与速度捕获寄存器中的捕获速度进行比较。如果捕获速度小于此比较寄存器中的值，则将断言速度比较中断 (VELC_Int) （如已使能）。

表 728. QEI 速度比较寄存器 （VELCOMP – 地址 0x400C 6038）位描述

位	符号	描述	复位值
31:0	VELCMP	速度比较值。	0x0

31.6.2.13 相位 A 输入寄存器上的 QEI 数字过滤器

此寄存器包含用于数字过滤器的采样计数。零采样计数将绕过该过滤器。

表 729. 相位 A 输入寄存器上的 QEI 数字过滤器 （FILTERPHA - 0x400C 603C）位描述

位	符号	描述	复位值
31:0	FILTA	数字过滤器采样延迟	0x0

31.6.2.14 相位 B 输入寄存器上的 QEI 数字过滤器

此寄存器包含用于数字过滤器的采样计数。零采样计数将绕过该过滤器。

表 730. 相位 B 输入寄存器上的 QEI 数字过滤器 （FILTERPHB - 0x400C 6040）位描述

位	符号	描述	复位值
31:0	FILTB	数字过滤器采样延迟	0x0

31.6.2.15 索引输入寄存器上的 QEI 数字过滤器

此寄存器包含用于数字过滤器的采样计数。零采样计数将绕过该过滤器。

表 731. 索引输入寄存器上的 QEI 数字过滤器 （FILTERINX - 0x400C 6044）位描述

位	符号	描述	复位值
31:0	FITLINX	数字过滤器采样延迟	0x0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

31.6.2.16 QEI 索引接受窗口寄存器

如果索引和相位 / 时钟边沿几乎一起下降，则此寄存器包含索引接受窗口的宽度。激活的相位 / 时钟边沿在索引之前下降 （但在窗口范围内），则会对时钟 / 相位激活 （重新）校准。

表 732. QEI 索引接受窗口寄存器 (WINDOW - 0x400C 6048) 位描述

位	符号	描述	复位值
31:0	WINDOW	索引接受窗口宽度	0x0

31.6.2.17 QEI 索引比较寄存器 1

此寄存器包含索引比较值，此值与索引计数寄存器的当前值进行比较。当比较值小于、等于或大于索引计数寄存器的当前值时，可以使能中断。

表 733. QEI 索引比较寄存器 1 （INXCMP1 - 地址 0x400C 604C）位描述

位	符号	描述	复位值
31:0	ICMP1	索引比较值 1。	0xFFFF FFFF

31.6.2.18 QEI 索引比较寄存器 2

此寄存器包含索引比较值，此值与索引计数寄存器的当前值进行比较。当比较值小于、等于或大于索引计数寄存器的当前值时，可以使能中断。

表 734. QEI 索引比较寄存器 2 （INXCMP2 - 地址 0x400C 6050）位描述

位	符号	描述	复位值
31:0	ICMP2	索引比较值 2。	0xFFFF FFFF

31.6.3 中断寄存器

31.6.3.1 QEI 中断使能清除寄存器

向此寄存器中的某位写入 1 将清除 QEI 中断使能寄存器 (QEIIE) 中的相应位。

表 735. QEI 中断使能清除寄存器 （IEC – 地址 0x400C 6FD8）位描述

位	符号	描述	复位值
0	INX_EN	指示已检测到索引脉冲。	0
1	TIM_EN	指示已发生速度定时器溢出	0
2	VELC_EN	指示已捕获速度低于比较速度。	0
3	DIR_EN	指示已检测到方向改变。	0
4	ERR_EN	指示已检测到编码器相位错误。	0
5	ENCLK_EN	指示已检测到编码器时钟脉冲。	0
6	POS0_INT	指示位置 0 比较值等于当前位置。	0
7	POS1_INT	指示位置 1 比较值等于当前位置。	0
8	POS2_INT	指示位置 2 比较值等于当前位置。	0
9	REV0_INT	指示索引 0 比较值等于当前索引计数。	0
10	POS0REV_INT	把位置 0 和转数计数中断相结合。在 POS0_Int 位和 REV_0 都设置时设置。	0
11	POS1REV_INT	把位置 1 和转数计数中断相结合。在 POS1_Int 位和 REV_1 都设置时设置。	0
12	POS2REV_Int	把位置 2 和转数计数中断相结合。在 POS2_Int 位和 REV_2 都设置时设置。	0
13	REV1_INT	指示索引 1 比较值等于当前索引计数。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 735. QEI 中断使能清除寄存器 （IEC – 地址 0x400C 6FD8）位描述

位	符号	描述	复位值
14	REV2_INT	指示索引 2 比较值等于当前索引计数。	0
15	MAXPOS_INT	指示当前位置计数方向向前从 MAXPOS 值至零，或方向向后从零至 MAXPOS。	0
31:16	-	保留	0

31.6.3.2 QEI 中断使能设置寄存器

向此寄存器中的某位写入 1 将设置 QEI 中断使能寄存器 (QEIIE) 中的相应位。

表 736. QEI 中断使能设置寄存器 （IES – 地址 0x400C 6FDC）位描述

位	符号	描述	复位值
0	INX_EN	指示已检测到索引脉冲。	0
1	TIM_EN	指示已发生速度定时器溢出	0
2	VELC_EN	指示已捕获速度低于比较速度。	0
3	DIR_EN	指示已检测到方向改变。	0
4	ERR_EN	指示已检测到编码器相位错误。	0
5	ENCLK_EN	指示已检测到编码器时钟脉冲。	0
6	POS0_Int	指示位置 0 比较值等于当前位置。	0
7	POS1_Int	指示位置 1 比较值等于当前位置。	0
8	POS2_Int	指示位置 2 比较值等于当前位置。	0
9	REV_Int	指示索引比较值等于当前索引计数。	0
10	POS0REV_Int	把位置 0 和转数计数中断相结合。在 POS0_Int 位和 REV_Int 都设置时设置。	0
11	POS1REV_Int	把位置 1 和转数计数中断相结合。在 POS1_Int 位和 REV_Int 都设置时设置。	0
12	POS2REV_Int	把位置 2 和转数计数中断相结合。在 POS2_Int 位和 REV_Int 都设置时设置。	0
13	REV1_Int	指示索引 1 比较值等于当前索引计数。	0
14	REV2_Int	指示索引 2 比较值等于当前索引计数。	0
15	MAXPOS_Int	指示当前位置计数方向向前从 MAXPOS 值至零，或方向向后从零至 MAXPOS。	0
31:16	-	保留	0

31.6.3.3 QEI 中断状态寄存器

此寄存器提供编码器接口的状态以及断言到控制器的中断源的当前设置。设置为 1 的位表示已发生锁存事件；零位表示尚未发生所关注的事件。将 0 写入某位位置将清除相应的中断。

表 737. QEI 中断状态寄存器 （INTSTAT - 地址 0x400C 6FE0）位描述

位	符号	描述	复位值
0	INX_Int	指示已检测到索引脉冲。	0
1	TIM_Int	指示已发生速度定时器溢出	0
2	VELC_Int	指示已捕获速度低于比较速度。	0
3	DIR_Int	指示已检测到方向改变。	0
4	ERR_Int	指示已检测到编码器相位错误。	0
5	ENCLK_Int	指示已检测到编码器时钟脉冲。	0
6	POS0_Int	指示位置 0 比较值等于当前位置。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 737. QEI 中断状态寄存器 （INTSTAT - 地址 0x400C 6FE0）位描述 *（续）*

位	符号	描述	复位值
7	POS1_Int	指示位置 1 比较值等于当前位置。	0
8	POS2_Int	指示位置 2 比较值等于当前位置。	0
9	REV_Int	指示索引比较值等于当前索引计数。	0
10	POS0REV_Int	把位置 0 和转数计数中断相结合。在 POS0_Int 位和 REV_Int 都设置时设置。	0
11	POS1REV_Int	把位置 1 和转数计数中断相结合。在 POS1_Int 位和 REV_Int 都设置时设置。	0
12	POS2REV_Int	把位置 2 和转数计数中断相结合。在 POS2_Int 位和 REV_Int 都设置时设置。	0
13	REV1_Int	指示索引 1 比较值等于当前索引计数。	0
14	REV2_Int	指示索引 2 比较值等于当前索引计数。	0
15	MAXPOS_Int	指示当前位置计数方向向前从 MAXPOS 值至零，或方向向后从零至 MAXPOS。	0
31:16	-	保留	0

31.6.3.4 QEI 中断使能寄存器

此寄存器使能中断源。位设置为 1 会使能相应的中断； 0 位则禁用相应的中断。

表 738. QEI 中断使能寄存器 （IE - 地址 0x400C 6FE4）位描述

位	符号	描述	复位值
0	INX_Int	指示已检测到索引脉冲。	0
1	TIM_Int	指示已发生速度定时器溢出	0
2	VELC_Int	指示已捕获速度低于比较速度。	0
3	DIR_Int	指示已检测到方向改变。	0
4	ERR_Int	指示已检测到编码器相位错误。	0
5	ENCLK_Int	指示已检测到编码器时钟脉冲。	0
6	POS0_Int	指示位置 0 比较值等于当前位置。	0
7	POS1_Int	指示位置 1 比较值等于当前位置。	0
8	POS2_Int	指示位置 2 比较值等于当前位置。	0
9	REV_Int	指示索引比较值等于当前索引计数。	0
10	POS0REV_Int	把位置 0 和转数计数中断相结合。在 POS0_Int 位和 REV_Int 都设置时设置。	0
11	POS1REV_Int	把位置 1 和转数计数中断相结合。在 POS1_Int 位和 REV_Int 都设置时设置。	0
12	POS2REV_Int	把位置 2 和转数计数中断相结合。在 POS2_Int 位和 REV_Int 都设置时设置。	0
13	REV1_Int	指示索引 1 比较值等于当前索引计数。	0
14	REV2_Int	指示索引 2 比较值等于当前索引计数。	0
15	MAXPOS_Int	指示当前位置计数方向向前从 MAXPOS 值至零，或方向向后从零至 MAXPOS。	0
31:16	-	保留	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

31.6.3.5 QEI 中断状态清除寄存器

向此寄存器中的某位写入 1 将清除 QEI 中断状态寄存器 (STAT) 中的相应位。

表 739. QEI 中断状态清除寄存器 (CLR - 0x400C 6FE8) 位描述

位	符号	描述	复位值
0	INX_Int	指示已检测到索引脉冲。	0
1	TIM_Int	指示已发生速度定时器溢出	0
2	VELC_Int	指示已捕获速度低于比较速度。	0
3	DIR_Int	指示已检测到方向改变。	0
4	ERR_Int	指示已检测到编码器相位错误。	0
5	ENCLK_Int	指示已检测到编码器时钟脉冲。	0
6	POS0_Int	指示位置 0 比较值等于当前位置。	0
7	POS1_Int	指示位置 1 比较值等于当前位置。	0
8	POS2_Int	指示位置 2 比较值等于当前位置。	0
9	REV_Int	指示索引比较值等于当前索引计数。	0
10	POS0REV_Int	把位置 0 和转数计数中断相结合。在 POS0_Int 位和 REV_Int 都设置时设置。	0
11	POS1REV_Int	把位置 1 和转数计数中断相结合。在 POS1_Int 位和 REV_Int 都设置时设置。	0
13	REV1_Int	指示索引 1 比较值等于当前索引计数。	0
14	REV2_Int	指示索引 2 比较值等于当前索引计数。	0
15	MAXPOS_Int	指示当前位置计数方向向前从 MAXPOS 值至零，或方向向后从零至 MAXPOS。	0
31:16	-	保留	0

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

31.6.3.6 QEI 中断状态设置寄存器

向此寄存器中的某位写入 1 将设置 QEI 中断状态寄存器 (STAT) 中的相应位。

表 740. QEI 中断状态设置寄存器 （SET - 地址 0x400C 6FEC）位描述

位	符号	描述	复位值
0	INX_Int	指示已检测到索引脉冲。	0
1	TIM_Int	指示已发生速度定时器溢出	0
2	VELC_Int	指示已捕获速度低于比较速度。	0
3	DIR_Int	指示已检测到方向改变。	0
4	ERR_Int	指示已检测到编码器相位错误。	0
5	ENCLK_Int	指示已检测到编码器时钟脉冲。	
6	POS0_Int	指示位置 0 比较值等于当前位置。	0
7	POS1_Int	指示位置 1 比较值等于当前位置。	0
8	POS2_Int	指示位置 2 比较值等于当前位置。	0
9	REV_Int	指示索引比较值等于当前索引计数。	0
10	POS0REV_Int	把位置 0 和转数计数中断相结合。在 POS0_Int 位和 REV_Int 都设置时设置。	0
11	POS1REV_Int	把位置 1 和转数计数中断相结合。在 POS1_Int 位和 REV_Int 都设置时设置。	0
12	POS2REV_Int	把位置 2 和转数计数中断相结合。在 POS2_Int 位和 REV_Int 都设置时设置。	0
13	REV1_Int	指示索引 1 比较值等于当前索引计数。	0
14	REV2_Int	指示索引 2 比较值等于当前索引计数。	0
15	MAXPOS_Int	指示当前位置计数方向向前从 MAXPOS 值至零，或方向向后从零至 MAXPOS。	0
31:16	-	保留	0

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

31.7 功能说明

QEI 模块解析正交编码器轮产生的两位灰码，进行位置的时间积分并确定旋转方向。另外，它还可捕获编码器轮的速度。

31.7.1 输入信号

QEI 模块支持两种信号操作模式：正交相位模式和时钟 / 方向模式。在正交相位模式中，编码器产生两个彼此相位相差 90 度的时钟；边沿关系用于确定旋转方向。在时钟 / 方向模式中，编码器产生一个指示步骤的时钟信号，以及一个指示旋转方向的方向信号。

此模式由 QEI 控制 (CON) 寄存器的 SigMode 位决定 （参见表 714）。如果 SigMode 位 = 1，则绕过正交解码器，把 PhA 引脚函数用作计数器的方向信号， PhB 引脚函数作为计数器的时钟信号，等。如果 SigMode = 0，则使用正交解码器解码 PhA 引脚和 PhB 引脚。在此模式中，正交解码器产生用于计数器等的方向和时钟信号。在两个模式中，方向信号均受方向反转 (DIRINV) 位的影响。

31.7.1.1 正交输入信号

如果 PhA 边沿在 PhB 边沿前，则位置计数器递增。如果 PhB 边沿在 PhA 边沿前，则位置计数器递减。如果一个相位中出现一个上升和下降边沿对，而在另一个上则没出现任何边沿，则说明旋转方向已改变。

表 741. 编码器状态

相位 A	相位 B	状态
1	0	1
1	1	2
0	1	3
0	0	4

表 742. 编码器状态跳变 [1]

从状态	至状态	方向
1	2	正
2	3	
3	4	
4	1	
4	3	负
3	2	
2	1	
1	4	

[1] 所有其它状态跳变均为非法，应设置 ERR 位。

PhA 和 PhB 输入信号的互换通过补足 DIR 位进行补偿。设置 = 1 时，方向反转位 (DIRINV) 补足 DIR 位。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 743. 编码器方向

DIR 位	DIRINV 位	方向
0	0	向前
1	0	反转
0	1	反转
1	1	向前

图 101 显示了正交编码器信号如何与方向和计数对应。

图 101. 正交编码器基本操作

31.7.1.2 数字输入过滤

所有三个编码器输入 （PhA、 PhB 和索引）都要求数字过滤。采样时钟的数目可进行用户编程，从 1 至 4,294,967,295 (0xFFFF FFFF)。为了使跳变得以接受，对于已编程数目的采样时钟，输入信号必须保持在新状态。

31.7.2 位置捕获

位置积分器的捕获模式可设置为在 PhA 信号的每个边沿处更新位置计数器，或在所有 PhA 和 PhB 边沿处都更新。在每个 PhA 和 PhB 更新位置计数器可在牺牲少量位置计数器范围的条件下提供更高位置分辨率。

位置积分器和速度捕获均可独立使能。或者，相位信号可被解析为时钟和方向信号，作为某些编码器的输出。

位置计数器在三种条件下自动复位。递增超过最大位置值 (MAXPOS) 将复位位置计数器为零。如果在索引位 (RESPI) 设置了复位，则首次检测到索引脉冲时，将在下次位置增加（校正）后复位一次位置计数器至零。如果在索引位 (CRESPI) 设置了连续复位，则在检测到索引脉冲时，将在下次位置增加 （再次校正）后连续复位位置计数器至零。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

31.7.3 速度捕获

速度捕获具备可编程的定时器和捕获寄存器。它计数规定时间期限内的相位边沿数 （使用与位置积分器相同的配置）。如果速度定时器 (TIME) 溢出，则速度计数器 (VEL) 的内容传输至捕获 (CAP) 寄存器。速度计数器随即清除，速度重新载入寄存器 (LOAD) 的内容被加载到速度定时器中。最后，断言速度中断 (TIM_Int)。在给定时间期间计数的边沿数与编码器的速度成正比。

设置复位速度位 (RESV) 将清除速度计数器、复位速度捕获寄存器至 0xFFFF FFFF，并将速度重新载入寄存器 (LOAD) 的内容加载到速度定时器中。

下列公式将速度计数器值转换成 RPM 值：

RPM = （PCLK × 速度 × 60） / 载入 × PPR × 边沿数）

其中：

- PCLK 是 QEI 控制器时钟。
- PPR 是物理编码器每旋转一圈的脉冲数。
- 边沿数为2或4，根据CON寄存器中的捕获模式设置确定（在CapMode设置是0时为2，在 CapMode 设置是 1 时为 4）。

例如，假设电机转速是 600 rpm。电机上配有每转 2048 脉冲的正交编码器，每转可产生 8192 个相位边沿。PhA 和 PhB 边沿都应用时钟时，可产生每秒 81,920 脉冲（电机每秒转 10 次）。如果定时器时钟设为 10,000 Hz，载入值是 2,500 （¼ 秒），则可达到每次更新计数 20,480 脉冲。使用上述公式：

RPM = (10000 × 20480 × 60) / (2500 × 2048 × 4) = 600 RPM

现在，假设电机加速到 3000 RPM。结果为每秒 409,600 脉冲，或每 ¼ 秒 102,400 脉冲。同样，根据上述公式：

RPM = (10000 × 102400 × 60) / (2500 × 2048 × 4) = 3000 RPM

31.7.4 速度比较

除了速度捕获之外，速度测量系统还包括可编程的速度比较寄存器。在每次速度捕获事件后，速度捕获寄存器 (CAP) 的内容将与速度比较寄存器 (VELCOMP) 的内容进行比较。如果捕获速度低于比较值，则只要已设置速度比较中断使能位，就会断言一个中断。这可用于确定电机轴是失速还是运转过慢。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

32.1 本章导读

所有 LPC43xx 器件上均配有 RIT。

32.2 基本配置

RIT 的配置如下：

- 参见表 744 以了解时钟与电源控制。
- RIT 通过 RITIMER_RST （复位号 36）复位。
- RIT 中断连接到 NVIC 中的插槽 # 11。

表 744. RIT 时钟和电源控制

	基本时钟	分支时钟	工作频率
RI 定时器寄存器接口时钟和 RI 定时器外设时钟 (PCLK)。	BASE_M4_CLK	CLK_M4_RITIMER	高达 204 MHz

32.3 功能要点

- 从 BASE_M4_CLK 运转的 32 位计数器。计数器可自由运转，或通过生成的中断复位。
- 32 位比较值。
- 32 位比较掩码。计数器值等于比较值时，会在掩码后生成中断。这样可实现简单比较无法实现的组合。

32.4 简介

“重复性中断定时器” (RIT) 提供了以指定时间间隔生成中断的通用方法，无需使用标准定时器。其目的是重复与 “操作系统中断”不相关的中断。如果有不同的系统要求， RIT 也可用作 Cortex-M4 系统节拍定时器的替代装置。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 102. 重复性中断定时器 (RIT) 框图

32.5 寄存器描述

表 745. 寄存器简介：重复性中断定时器 (RIT) （基址 0x400C 0000）

名称	访问类型	地址	描述	复位值 [1]	参考
COMPVAL	R/W	0x000	比较寄存器	0xFFFF FFFF	表 746
MASK	R/W	0x004	掩码寄存器。此寄存器保存 32 位掩码值。在任何位写入 ‘1’ 都将强制计数器的相应位与比较寄存器进行比较。	0	表 747
CTRL	R/W	0x008	控制寄存器。	0xC	表 748
计数器	R/W	0x00C	32 位计数器	0	表 749

[1] 复位值仅反映已使用位中保存的数据，不包括保留位的内容。

32.5.1 RI 比较值寄存器

表 746. RI 比较值寄存器 （COMPVAL – 地址 0x400C 0000）位描述

位	符号	描述	复位值
31:0	RICOMP	比较寄存器。保存与计数器比较的比较值。	0xFFFF FFFF

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

32.5.2 RI 掩码寄存器

表 747. RI 掩码寄存器 （MASK - 地址 0x400C 0004）位描述

位	符号	描述	复位值
31:0	RIMASK	掩码寄存器。此寄存器保存 32 位掩码值。在任何位写入 1 将会覆盖计数器相应位和比较寄存器的比较结果 （导致寄存器位的比较始终为 True）。	0

32.5.3 RI 控制寄存器

表 748. RI 控制寄存器 （CTRL - 地址 0x400C 0008）位说明

位	符号	值	描述	复位值
0	RITINT		中断标志	0
		1	只要计数器值等于由 RICOMPVAL 和 RIMASK 的内容指定的已遮蔽值，此位就通过硬件设置为 1。 在此位写入 1 将清除至 0。写入 0 无效。	
		0	计数器值与已遮蔽的比较值不等。	
1	RITENCLR		定时器使能清除	
		1	只要计数器值等于由 RICOMPVAL 和 RIMASK 的内容指定的已遮蔽值，定时器就将被清除到 0。这将发生在设置中断标志的同一时钟上。	0
		0	定时器不能清除至 0。	
2	RITENBR		为调试使能定时器	1
		1	当处理器因调试而停止时，定时器停止。	
		0	调试对定时器操作无影响。	
3	RITEN		定时器使能。	1
		1	定时器已使能。 **注:** 如果已在位2使能，则此定时器可能因调试停止而被否决。	
		0	定时器已禁用。	
31:4	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

32.5.4 RI 计数器寄存器

表 749. RI 计数器寄存器 （COUNTER - 地址 0x400C 000C）位描述

位	符号	描述	复位值
31:0	RICOUNTER	32 位正计数。连续计数，除非 RICTRL 寄存器内的 RITEN 位被清除或输入了调试模式 （如果已被 RICTRL 内的 RITNEBR 使能）。可在软件中载入任何值。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

32.6 RI 定时器操作

复位后，计数器从 0x00000000 开始向上计数。无论何时计数器值等于编程到 RICOMPVAL 寄存器中的值，都将设置中断标志。通过向 RIMASK 寄存器中的相应位写入 ‘1’，可把任何位或位组合从此比较 （即，强制比较）中移除。如果 enable_clr 位低 （默认状态），有效比较仅导致设置中断标志。对计数顺序无影响，计数如常继续。计数器达到 0xFFFFFFFF 后，则在下一时钟翻转至 0x00000000，然后继续计数。如果 enable_clr 位设置为 ‘1’，则有效比较也将导致计数器复位为零。计数将在下一时钟边沿从此处恢复。

计数可在软件中停止，方法为向 Enable_Timer 位 - RICTRL(2) 写入 ‘0’ 。如果已设置 Enable_Break 位 – RICTRL(1)，则处理器因调试而停止时，计数也将停止。Enable_Timer 和 Enable_Break 位都在复位时进行设置。

可在软件中清除中断标志，方法为向中断位 – RICTRL(0) 写入 ‘1’。

通过写入 RICOUNTER，软件可在任何时间载入计数器任何值。

计数器 (RICOUNTER)、 RICOMPVAL 寄存器、 RIMASK 寄存器和 RICTRL 寄存器都可在任何时间通过软件读取。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

33.1 本章导读

所有 LPC43xx 器件上的报警定时器都是一样的。

33.2 基本配置

报警定时器的配置如下：

- 参见表750以了解时钟与电源控制。必须在CREG模块的CREG0寄存器中，使能32 kHz振荡器的 32 kHz 输出 （参见表 38）。
-
- 报警定时器中断连接到 “事件”路由器中的插槽 # 4。

表 750. 报警定时器时钟和电源控制

	基本时钟	分支时钟	工作频率
报警定时器寄存器接口时钟	BASE_M4_CLK	CLK_M4_BUS	最大 204 MHz
用于计数器 / 定时器时钟的 32 kHz 石英振荡器输出	-	-	1024 Hz （固定频率）

33.3 简介

报警定时器是 16 位定时器，从预定值向下计数。计数器达到 0x00 时启动状态位，如果使能的话产生中断。

报警定时器在 RTC 电源域操作，它由 16 位计数器 (DOWNCOUNTER) 组成，以 1024 Hz 时钟运行，该 1024 Hz 时钟源自 32 kHz 石英时钟。此时钟未激活时，报警定时器也未激活。

报警定时器从初始值PRESET开始向下计时。如果达到0x0且中断已使能（通过SET_EN），则启动 STATUS 位。计数器继续从 PRESET 开始向下计时。

STATUS 传播到中断输出，该中断连接到 “事件”路由器，并可用于把设备从低功耗模式唤醒。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

33.4 寄存器描述

表 751. 寄存器简介：报警定时器 （基址 0x4004 0000）

名称	访问类型	地址偏移	描述	复位值	参考
DOWNCOUNTER	R/W	0x000	向下计数器寄存器	0x000	表 752
PRESET	R/W	0x004	预设值寄存器	0x000	表 753
-	-	0x008 - 0xFD4	保留	-	-
CLR_EN	W	0xFD8	中断清除使能寄存器	0x0	表 754
SET_EN	W	0xFDC	中断设置使能寄存器	0x0	表 755
STATUS	R	0xFE0	状态寄存器	0x0	表 756
ENABLE	R	0xFE4	使能寄存器	0x0	表 757
CLR_STAT	W	0xFE8	清除寄存器	0x0	表 758
SET_STAT	W	0xFEC	设置寄存器	0x0	表 759

33.4.1 向下计数器寄存器

表 752. 向下计数器寄存器 (DOWNCOUNTER - 0x4004 0000) 位描述

位	符号	描述	复位值
15:0	CVAL	等于零时产生中断。 如果等于零，载入 PRESET 并继续计数。	0x0
31:16	-	保留。	-

33.4.2 预设值寄存器

表 753. 预设值寄存器 (PRESET - 0x4004 0004) 位描述

位	符号	描述	复位值
15:0	PRESETVAL	DOWNCOUNTER 等于零时，载入 DOWNCOUNTER 中的值。	
31:16	-	保留。	-

33.4.3 中断清除使能寄存器

表 754. 中断清除使能寄存器 (CLR_EN - 0x4004 0FD8) 位描述

位	符号	描述	复位值
0	CLR_EN	向此位写入 1 将清除 ENABLE 寄存器中的中断使能位。	-
31:1	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

33.4.4 中断设置使能寄存器

表 755. 中断设置使能寄存器 (SET_EN - 0x4004 0FDC) 位描述

位	符号	描述	复位值
0	SET_EN	向此位写入 1 将设置 ENABLE 寄存器中的中断使能位。	0
31:1	-	保留。	-

33.4.5 中断状态寄存器

表 756. 中断状态寄存器 (STATUS - 0x4004 0FE0) 位描述

位	符号	描述	复位值
0	STAT	此位为 1 表示 STATUS 中断已发生。	0
31:1	-	保留。	-

33.4.6 中断使能寄存器

表 757. 中断使能寄存器 (ENABLE - 0x4004 0FE4) 位描述

位	符号	描述	复位值
0	EN	此位为 1 表示已使能 STATUS 中断，且当 STATUS 寄存器中的 STAT = 1 时， STATUS 中断申请信号已被断言。	0
31:1	-	保留。	-

33.4.7 清除状态寄存器

表 758. 中断清除状态寄存器 (CLR_STAT - 0x4004 0FE8) 位描述

位	符号	描述	复位值
0	CSTAT	向此位写入 1 将清除 STATUS 寄存器中的 STATUS 中断位。	0
31:1	-	保留。	-

33.4.8 设置状态寄存器

表 759. 中断设置状态寄存器 (SET_STAT - 0x4004 0FEC) 位描述

位	符号	描述	复位值
0	SSTAT	向此位写入 1 将设置 STATUS 寄存器中的 STATUS 中断位。	0
31:1	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

34.1 本章导读

所有 LPC43xx 器件中的 WWDT 都是一样的。

34.2 基本配置

WWDT 的配置如下：

- 参见表 760 以了解时钟与电源控制。 WWDT 时钟 (WDCLK) 的唯一时钟源是 IRC。
- WWDT 不能用软件复位。
- WWDT 中断连接到事件路由器中的插槽 # 7 以及 M4 NVIC 中的插槽 # 49。
- WWDT 寄存器可由 GPDMA 作为存储器到存储器传输来访问。

表 760. WWDT 时钟和电源控制

	基本时钟	分支时钟	工作频率
WWDT 寄存器接口时钟 (PCLK)	BASE_M4_CLK	CLK_M4_WWDT	最大 204 MHz
看门狗时钟 (WDCLK)	BASE_SAFE_CLK	-	12 MHz（固定频率）

34.3 功能要点

- 如果不能在可编程超时期限内重新载入，则在内部复位芯片。
- 可选的窗口化操作，要求在最大和最小超时期限之间重新载入，两个时限都可编程。
- 可在看门狗超时之前的可编程时间生成可选的警报中断。
- 带内部固定预分频器的可编程 24 位定时器。
- 时间周期可选，从 1,024 个看门狗时钟 ($T_{WDCLK} \times 256 \times 4$) 到超过 6,700 万个看门狗时钟 ($T_{WDCLK} \times 2^{24} \times 4$)，取 4 个看门狗时钟的位数。
- 安全看门狗操作。如果使能，要求禁用软件复位或看门狗复位。
- 不当的输入序列会导致看门狗立即复位 （如已使能）。
- 可选择保护看门狗重新载入值，使其只能在 “警报中断” 时间后才能改变。
- 指示看门狗复位的标志。
- WWDT 使用 IRC 作为固定时钟源。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

34.4 应用

看门狗定时器的目的是，如果进入错误状态，可在合理时间内复位微控制器。当看门狗使能时，如果用户程序没有在预定时间内喂狗（或给看门狗定时器重载定时值），则将产生看门狗事件。看门狗事件将导致芯片复位，如果配置相应功能，

编程看门狗窗口时，早期的看门狗输入也被视为看门狗事件。这可防止有故障的系统仍然输入看门狗的情况。例如，应用程序代码可能在包含看门狗输入的中断服务中阻塞。将窗口设置为使这种情况导致早期输入，这将生成看门狗事件，允许系统恢复。

34.5 描述

看门狗包括一个固定的 4 分频预分频器和一个 24 位计数器，后者在每个时钟周期递减。计数器递减的最小起始值是 0xFF。设置低于 0xFF 的值将使 0xFF 被载入计数器。因此，看门狗的最小间隔为 ($T_{WDCLK} \times 256 \times 4$)，最大间隔为 ($T_{WDCLK} \times 2^{24} \times 4$)，两者都是 ($T_{WDCLK} \times 4$) 的倍数。看门狗应以下列方式使用：

- 在 TC 寄存器中设置看门狗超时值。
- 在 MOD 寄存器中设置看门狗定时器操作模式。
- 如果需要窗口化操作，在 WINDOW 寄存器中设置看门狗窗口时间的值。
- 如果需要警报中断，在 WARNINT 寄存器中为看门狗警报中断设置比较值。
- 向 FEED 寄存器中写入 0xAA，然后写入 0x55，使能看门狗。
- 在看门狗计数器达到零之前必须重新输入看门狗，以防止看门狗事件。如果窗口值是编程的，则还必须在看门狗计数器超过该值后才输入。

如果 “看门狗定时器”配置为看门狗事件导致复位，则当计数器达到零时， CPU 将复位，从向量表中载入协议栈指针和程序计数器，如同外部重启一样。可检查看门狗超时标志 (WDTOF)，以确定看门狗是否导致了复位条件。 WDTOF 标志必须通过软件清除。

如果看门狗定时器配置为生成警报中断，则当计数器与 WARNINT 寄存器定义的比较值匹配时，会发生中断。

34.5.1 调试模式中的 WWDT 特性

如果在 “调试”模式中停止执行编码，则 WWDT 停止计数直至编码执行恢复。

34.5.2 掉电模式中的 WWDT 特性

WWDT 在睡眠模式中运行。睡眠模式中看门狗启动的复位将复位和唤醒芯片。同样，如果 NVIC 中使能看门狗启动中断，则该中断将从睡眠模式唤醒芯片。

WWDT 在深度睡眠、掉电和深度掉电模式中不工作。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

34.6 时钟

看门狗定时器块使用两个时钟：PCLK 和 WDCLK。PCLK 用于 APB 访问看门狗寄存器，源自 BASE_M4_CLK。WDCLK 用于看门狗定时器计数，源自 IRC。时钟源 (IRC) 是固定的，以确保 WDT 始终有有效时钟。

这两个时钟域之间有一些同步逻辑。当 MOD 和 TC 寄存器通过 APB 操作更新时，新值将在 WDCLK 时钟域逻辑的三个 WDCLK 周期后生效。看门狗定时器计数 WDCLK 时钟周期时，同步逻辑将首先锁定 WDCLK 上计数器单独值，然后使其与 PCLK 同步，以便 CPU 将其作为 TV 寄存器读取。

34.7 寄存器描述

看门狗寄存器如表 761 中所示。

表 761. 寄存器简介：看门狗定时器 （基址 0x4008 0000）

名称	访问类型	地址偏移	描述	复位值 [1]	参考
MOD	R/W	0x000	看门狗模式寄存器，此寄存器包含 “看门狗定时器” 的基本模式和状态。	0	表 762
TC	R/W	0x004	看门狗定时器常数寄存器，此寄存器确定超时值。	0xFF	表 764
FEED	WO	0x008	看门狗输入序列寄存器，向此寄存器写入 0xAA，然后写入 0x55，将使用 WDTC 中包含的值重新载入看门狗定时器。	不适用	表 765
TV	RO	0x00C	看门狗定时器值寄存器，此寄存器读取看门狗定时器的当前值。	0xFF	表 766
-	-	0x010	保留。	-	-
WARNINT	R/W	0x014	看门狗警报中断寄存器，此寄存器包含看门狗警报中断比较值。	0	表 767
WINDOW	R/W	0x018	看门狗定时器窗口寄存器，此寄存器包含看门狗窗口值。	0xFF FFFF	表 768

[1] 复位值仅反映存储在所用位中的数据，不包括保留位的内容。

34.7.1 看门狗模式寄存器

WDMOD 寄存器根据 WDEN 和 RESET 位的组合控制看门狗的操作。注意，看门狗输入必须在 WDMOD 寄存器的任何改变生效之前执行。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 762. 看门狗模式寄存器 (MOD - 0x4008 0000) 位描述

位	符号	值	描述	复位值
0	WDEN		看门狗使能位，此位仅为 “设置”。	0
		0	看门狗定时器停止。	
		1	看门狗定时器运行。	
1	WDRESET		看门狗复位使能位。此位仅为 “设置”。	0
		0	看门狗超时不引起芯片复位。	
		1	看门狗超时引起芯片复位。	
2	WDTOF		看门狗超时标志。在看门狗定时器由于输入错误或与 WDPROTECT 关联的事件而超时时设置，使用软件清除。如果 WDRESET = 1，则导致芯片复位。此标志利用软件向此位写入 0 来清除。	0（仅在外部复位之后）
3	WDINT		看门狗中断标志。在定时器达到 WARNINT 寄存器中的值时设置。通过软件向此位写入 1 来清除。	0
4	WDPROTECT		看门狗更新模式。此位仅为 “设置”。	0
		0	看门狗超时值 (WDTC) 可随时改变。	
		1	看门狗超时值(WDTC)仅在计数器低于WDWARNINT 和 WDWINDOW 的值时才能改变。	
7:5	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

WDEN、 **WDPROTECT** 或 **WDRESET** 位设置后，不能用软件清除。两个标志都须利用外部复位或看门狗定时器复位来清除。

WDTOF 在看门狗超时、发生输入错误，或当 WDPROTECT =1 且试图写入 TC 寄存器时，设置该看门狗超时标志。此标志通过软件向此位写入 0 来清除。

WDINT 当看门狗计数器达到 WDWARNINT 规定的值时，设置该看门狗中断标志。此标志在发生复位时清除，并通过软件向此位写入 1 来清除。

在看门狗运行并具有操作时钟源的任何时候都可发生看门狗复位或中断。所有时钟源都以 “睡眠” 模式运行，且 IRC 以 “深度睡眠” 模式工作。如果看门狗中断发生在 “睡眠” 或 “深度睡眠” 模式，则会唤醒设备。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 763. 看门狗工作模式选择

WDEN	WDRESET	工作模式
0	X (0 或 1)	看门狗不运行时的调试 / 操作。
1	0	看门狗中断模式：将生成看门狗警报中断，但不会生成看门狗复位。 选择此模式后，在看门狗计数器达到 WDWARNINT 规定的值时将设置 WDINT 标志，并将生成看门狗中断请求。
1	1	看门狗复位模式：看门狗中断和看门狗复位都使能。 选择此模式后，在看门狗计数器达到 WDWARNINT 规定的值时将设置 WDINT 标志，并将生成看门狗中断请求。看门狗计数器达到零时将复位微控制器。 **注：**其它导致看门狗复位的原因有：在达到WDWINDOW值之前看门狗输入或改变 WDTC 值（如果在 MOD 寄存器中设置了 WDPROTECT 位）。

34.7.2 看门狗定时器常数寄存器

此 TC 寄存器确定超时值。每次发生输入序列时， TC 寄存器的内容就会重新载入看门狗定时器。复位时使用值 0x00 00FF 预加载。写入值低于 0xFF 将导致 0x00 00FF 被载入 TC 寄存器。因此，最小超时间隔为 $T_{WDCLK} \times 256 \times 4$。

如果 MOD 寄存器中的 WDPROTECT 位设置为 1，则在看门狗计数器低于 WDWARNINT 和 WDWINDOW 的值前试图改变 TC 的值，将导致看门狗复位并设置 WDTOF 标志。

表 764. 看门狗定时器常量寄存器 (TC - 0x4008 0004) 位描述

位	符号	描述	复位值
23:0	WDTC	看门狗超时值。	0x00 00FF
31:24	-	保留，用户软件不应向保留位写入1。未定义从保留位读取的值。	不适用

34.7.3 看门狗输入寄存器

向此寄存器写入 0xAA 然后写入 0x55，将用 TC 寄存器中的超时值重新加载看门狗定时器。如果看门狗是通过 MOD 寄存器使能的，此操作也将启动看门狗。设置 WDMOD 寄存器中的 WDEN 位不足以使能看门狗。设置 WDEN 后必须完成有效的输入序列，才能使看门狗能够生成复位。此时，看门狗将忽略输入错误。向 FEED 寄存器写入 0xAA 后，访问任何看门狗寄存器 （向 FEED 寄存器写入 0x55 除外）都将导致立即复位 / 中断 （如果看门狗已使能），并设置 WDTOF 标志。在输入序列期间，复位将在错误访问看门狗寄存器后的第二个 PCLK 期间生成。

在输入序列期间应禁用中断。如果在输入序列期间发生中断，会发生中止。

表 765. 看门狗输入寄存器 (FEED - 0x4008 0008) 位描述

位	符号	描述	复位值
7:0	Feed	输入值应是 0xAA，后跟 0x55。	不适用

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

34.7.4 看门狗定时器值寄存器

WDTV 寄存器用于读取看门狗定时器计数器的当前值。

在读取 24 位计数器的值时，锁定和同步步骤会占用 6 个 WDCLK 周期加 6 个 PCLK 周期，因此， WDTV 的值比 CPU 读取它时定时器的实际值早。

表 766. 看门狗定时器值寄存器 (TV - 0x4008 000C) 位描述

位	符号	描述	复位值
23:0	计数	计数器定时器值。	0x00 00FF
31:24	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

34.7.5 看门狗定时器报警中断寄存器

WDWARNINT 寄存器确定将生成看门狗中断的看门狗定时器计数器值。当看门狗定时器计数器与 WDWARNINT 规定的值匹配时，会在后续的 WDCLK 后生成中断。

如果计数器后 10 位与 WARNINT 的 10 位值相同，而计数器前面其余位的值都是 0，则看门狗定时器计数器与 WDWARNINT 匹配，这使得看门狗事件之前发生中断的最大时间是 1,023 看门狗定时器计数（4,096 个看门狗时钟）。如果 WDWARNINT 设置为 0，则在看门狗事件的同时发生中断。

表 767. 看门狗定时器警告中断寄存器 (WARNINT - 0x4008 0014) 位描述

位	符号	描述	复位值
9:0	WDWARNINT	看门狗警报中断比较值。	0
31:10	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

34.7.6 看门狗定时器窗口寄存器

WDWINDOW 寄存器确定在执行看门狗输入时允许的最大 WDTV 值。如果输入有效序列在 WDTV 达到 WDWINDOW 中的值之前完成，则将发生看门狗事件。

WDWINDOW 复位到可能的最大 WDTV 值，因此窗口操作无效。WDWINDOW 值低于 0x100 时不可能成功输入看门狗。

表 768. 看门狗定时器窗口寄存器 (WINDOW - 0x4008 0018) 位描述

位	符号	描述	复位值
23:0	WDWINDOW	看门狗窗口值。	0xFF FFFF
31:24	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

34.8 功能框图

看门狗的功能框图如下面的图 103 所示。功能框图中未显示同步逻辑 (PCLK - WDCLK)。

图 103. 看门狗功能框图

© NXP B.V. 2012. 保留所有权利。

34.9 看门狗定时示例

下图描绘了看门狗定时器工作 （如下面的图 104 所示）的几个方面。

图 104. 使能窗口模式时提前看门狗喂狗

图 105. 使能窗口模式时正确的看门狗喂狗

图 106. 看门狗警告中断

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

35.1 本章导读

所有 LPC43xx 器件上均配有 RTC。

35.2 基本配置

RTC 的配置如下：

- 参见表 769 以了解时钟与电源控制。必须在 CREG 模块的 CREG0 寄存器中，使能 1 kHz 振荡器的 32 kHz 输出 （参见表 38）。
- RTC 中断连接到 “事件” 路由器中的插槽 # 5。
- RTC_ALARM 引脚功能由 CREG0 寄存器控制（参见表 38）。默认情况下，可以在该引脚上监控 RTC 报警中断。
-

注：初始化 32 kHZ 振荡器后，等待 2 秒再写入 RTC 寄存器。

表 769. RTC 时钟和电源控制

	基本时钟	分支时钟	工作频率
RTC 寄存器接口的时钟	BASE_M4_CLK	CLK_M4_BUS	最大 204 MHz
用于 RTC 计数器 / 定时器时钟的 32 kHz 晶体振荡器输出	-	-	1024 Hz（固定频率）； RTC 接收内部 1 Hz 时钟。

35.3 功能要点

- 测量时间通道，以维护日历和时钟。提供秒、分钟、小时、月日期、月、年、周日期及年日期。
- 超低功耗设计，支持电池供电系统。使用 CPU 电源供电，如果有。
- 专用电池供电引脚。
- RTC 供电独立于芯片其它部分。
- 校准计数器允许优于 ±1 秒 / 天的调节，分辨率 1 秒。
- 时间寄存器的任何字段以及选定的分数秒值的递增均可生成定期中断。
- 可为具体日期 / 时间生成报警中断。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

35.4 简介

“实时时钟 (RTC)”是一组在系统电源开启时计量时间的计数器，电源关闭时也可使用。在 CPU 不访问其寄存器时耗电极低，特别是在低电模式下。在 LPC43xx 上， RTC 通过独立的 32 kHz 振荡器确定时钟，该振荡器生成 1 Hz 的内部时间基准。RTC 由自带的电源引脚 VBAT 供电。

图 107. RTC 功能框图

35.5 引脚说明

表 770. RTC 引脚描述

函数名称	方向	描述
RTC_ALARM	O	RTC 控制输出，这是一个 1.8 V 引脚，在生成 RTC 报警时达到“高”位。CREG0 位 ALARMCTRL 必须设为 0。（参见表 38）。 **注：**用于 JTAG 编程时，该引脚必须为低电平。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

35.6 寄存器描述

表 771. 寄存器简介：RTC （基址 0x4004 6000）

名称	访问类型	地址偏移	描述	复位值	参考
ILR	W	0x000	中断位置寄存器	0x0	表 773
-	-	0x004	保留	0x00	-
CCR	R/W	0x008	时钟控制寄存器	0x00	表 774
CIIR	R/W	0x00C	计数器递增中断寄存器	0x00	表 775
AMR	R/W	0x010	报警掩码寄存器	-[1]	表 776
CTIME0	R	0x014	完整时间寄存器 0	-[1]	表 777
CTIME1	R	0x018	完整时间寄存器 1	-[1]	表 778
CTIME2	R	0x01C	完整时间寄存器 2	-[1]	表 779
SEC	R/W	0x020	秒寄存器	-[1]	表 782
MIN	R/W	0x024	分钟寄存器	-[1]	表 783
HRS	R/W	0x028	小时寄存器	-[1]	表 784
DOM	R/W	0x02C	月日期寄存器	-[1]	表 785
DOW	R/W	0x030	周日期寄存器	-[1]	表 786
DOY	R/W	0x034	年日期寄存器	-[1]	表 787
MONTH	R/W	0x038	月份寄存器	-[1]	表 788
YEAR	R/W	0x03C	年份寄存器	-[1]	表 789
CALIBRATION	R/W	0x040	校准值寄存器	-[1]	表 790
-	-	0x044 - 0x05C		-	-
ASEC	R/W	0x060	报警秒寄存器	-[1]	表 792
AMIN	R/W	0x064	报警分钟寄存器	-[1]	表 793
AHRS	R/W	0x068	报警小时寄存器	-[1]	表 794
ADOM	R/W	0x6C	报警月日期寄存器	-[1]	表 795
ADOW	R/W	0x070	报警周日期寄存器	-[1]	表 796
ADOY	R/W	0x074	报警年日期寄存器	-[1]	表 797
AMON	R/W	0x078	报警月寄存器	-[1]	表 798
AYRS	R/W	0x07C	报警年寄存器	-[1]	表 799

[1] 此寄存器值不能通过复位改变。

主电源关闭时，除 RTC 寄存器之外，还提供有 64 个通用寄存器用于存储数据。通用寄存器驻留在 RTC 电源域，可以使用电池供电。

表 772. 寄存器简介：REGFILE （基址 0x4004 1000）

名称	访问类型	地址偏移	描述	复位值
REGFILE0	R/W	0x000	通用存储寄存器	0x0
至				
REGFILE63	R/W	0x0FC	通用存储寄存器	0x0

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

35.6.1 中断位置寄存器

中断位置寄存器是一种 2 位寄存器，它确定在哪些模块生成中断（参见表 773）。在相应位写入 1 将清除相应的中断。写入 0 无效。这使得编程人员可以读取此寄存器，并写回同样的值来仅清除读取操作检测到的中断。

表 773. 中断位置寄存器 （ILR – 地址 0x4004 6000）位描述

位	符号	描述	复位值
0	RTCCIF	为 1 时，计数器递增中断模块生成中断。向此位位置写入 1 将清除计数器递增中断。	0
1	RTCALF	为 1 时，报警寄存器生成中断。向此位位置写入 1 将清除警报中断。	0
31:2	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

35.6.2 时钟控制寄存器

时钟寄存器是一种 4 位寄存器，它控制时钟分频电路的操作。表 774 说明了时钟寄存器的每个位。在 RTC 首次启动时应初始化此寄存器中的 0、 1、 4 位。

表 774. 时钟控制寄存器 （CCR – 地址 0x4004 6008）位描述

位	符号	值	描述	复位值
0	CLKEN		时钟使能。	-[1]
		0	时间计数器被禁用以便可以初始化它们。	
		1	时间计数器被使能。	
1	CTCRST		CTC 复位。	0
		0	无效。	
		1	为 1 时，内部振荡器分频器的元件被复位，并保留复位直至 CCR[1] 变为零。该分频器从 32.768 kHz 石英生成 1 Hz 时钟，不能通过软件查看分频器的状态。	
3:2	-		内部测试模式控制。对于正常 RTC 操作，这些位必须为 0。	-[1]
4	CCALEN		校准计数器使能。	-[1]
		0	校准计数器使能并计数，使用 1 Hz 时钟。当校准计数器等于 CALIBRATION 寄存器的值时，计数器复位并重复计数至 CALIBRATION 寄存器的值。参见第 35.6.6.2 节和第 35.7.1 节。	
		1	校准计数器禁用并复位为零。	
31:5	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

[1] 此寄存器值不能通过复位改变。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

35.6.3 计数器递增中断寄存器

“计数器递增中断寄存器 (CIIR)”能够在计数器每次递增时生成中断。此中断保持有效直至向 “中断位置寄存器 (ILR[0])”的 0 位写入 1。

表 775. 计数器递增中断寄存器 （CIIR – 地址 0x4004 600C）位描述

位	符号	描述	复位值
0	IMSEC	为 1 时，秒值递增将生成中断。	0
1	IMMIN	为 1 时，分钟值递增将生成中断。	0
2	IMHOUR	为 1 时，小时值递增将生成中断。	0
3	IMDOM	为 1 时，月日期值递增将生成中断。	0
4	IMDOW	为 1 时，周日期值递增将生成中断。	0
5	IMDOY	为 1 时，年日期值递增将生成中断。	0
6	IMMON	为 1 时，月份值递增将生成中断。	0
7	IMYEAR	为 1 时，年份值递增将生成中断。	0
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

35.6.4 报警掩码寄存器

“报警掩码寄存器 (AMR)”允许用户掩码报警寄存器的任何内容。表 776 显示了 AMR 位与报警之间的关系。对于报警功能，每个未遮蔽报警寄存器都必须匹配相应的时间计数器，以便生成中断。只有在计数器比较首次从不匹配变为匹配时才生成中断。向 “中断位置寄存器 (ILR)”的适当位写入 1 可移除中断。如果设置了所有掩码位，则报警禁用。

表 776. 报警掩码寄存器 （AMR – 地址 0x4004 6010）位描述

位	符号	描述	复位值
0	AMRSEC	为 1 时，秒值不用于报警比较。	0
1	AMRMIN	为 1 时，分钟值不用于报警比较。	0
2	AMRHOUR	为 1 时，小时值不用于报警比较。	0
3	AMRDOM	为 1 时，月日期值不用于报警比较。	0
4	AMRDOW	为 1 时，周日期值不用于报警比较。	0
5	AMRDOY	为 1 时，年日期值不用于报警比较。	0
6	AMRMON	为 1 时，月份值不用于报警比较。	0
7	AMRYEAR	为 1 时，年份值不用于报警比较。	0
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

35.6.5 完整时间寄存器

也选择在完整格式中读取 “时间计数器” 的值，使程序员仅用三次读取操作就可读取所有时间计数器。各种寄存器汇总至 32 位值中，如表 777、表 778 和表 779 所示。每个寄存器的最低有效位在 0、 8、 16 或 24 位读回。

“完整时间寄存器” 是只读的。如需向 “时间计数器” 写入新值，则须使用 “时间计数器” 地址。

35.6.5.1 完整时间寄存器 0

“完整时间寄存器 0” 包含低阶时间值：秒、分钟、小时和周日期。

表 777. 完整时间寄存器 0 （CTIME0 – 地址 0x4004 6014）位描述

位	符号	描述	复位值
5:0	SECONDS	秒值，范围 0 至 59	-[1]
7:6	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
13:8	MINUTES	分钟值，范围 0 至 59	-[1]
15:14	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
20:16	HOURS	小时值，范围 0 至 23	-[1]
23:21	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-[1]
26:24	DOW	周日期值，范围 0 至 6	-[1]
31:27	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

35.6.5.2 完整时间寄存器 1

“完整时间寄存器 1” 包含月日期、月份和年份值。

表 778. 完整时间寄存器 1 （CTIME1 – 地址 0x4004 6018）位描述

位	符号	描述	复位值
4:0	DOM	月日期值，范围 1 至 28、 29、 30、 31 （取决于月份大小以及是否是闰年）。	-[1]
7:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
11:8	MONTH	月份值，范围 1 至 12。	-[1]
15:12	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
27:16	YEAR	年份值，范围 0 至 4095。	-[1]
31:28	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

35.6.5.3 完整时间寄存器 2

“完整时间寄存器 2”仅包含年日期值。

表 779. 完整时间寄存器 2 （CTIME2 – 地址 0x4004 601C）位描述

位	符号	描述	复位值
11:0	DOY	年日期值，范围 1 至 365 （闰年是 366）。	-[1]
31:12	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

35.6.6 时间计数器组

时间值由如表 780 和表 781 所示的八个计数器组成，这些计数器可在表 781 所示位置读写。

表 780. 时间计数器关系和值

计数器	Size	使能源	最小值	最大值
第二	6	1 Hz 时钟	0	59
分钟	6	第二	0	59
小时	5	分钟	0	23
月日期	5	小时	1	28、 29、 30 或 31
周日期	3	小时	0	6
年日期	9	小时	1	365 或 366 （闰年）
月份	4	月日期	1	12
年份	12	月份或年日期	0	4095

表 781. 时间计数器寄存器

名称	Size	描述	访问类型	地址
SEC	6	秒值，范围 0 至 59	R/W	0x4004 6020
MIN	6	分钟值，范围 0 至 59	R/W	0x4004 6024
HRS	5	小时值，范围 0 至 23	R/W	0x4004 6028
DOM	5	月日期值，范围 1 至 28、29、30、31（取决于月份大小以及是否是闰年）。[1]	R/W	0x4004 602C
DOW	3	周日期值，范围 0 至 6[1]	R/W	0x4004 6030
DOY	9	年日期值，范围 1 至 365 （闰年是 366）[1]	R/W	0x4004 6034
MONTH	4	月份值，范围 1 至 12	R/W	0x4004 6038
YEAR	12	年份值，范围 0 至 4095	R/W	0x4004 603C

[1] 这些值仅以适当的间隔递增，并在规定的溢出点复位。不用计算，并且必须正确初始化才有意义。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 782. 秒寄存器 （SEC – 地址 0x4004 6020）位描述

位	符号	描述	复位值
5:0	SECONDS	秒值，范围 0 至 59	-[1]
31:6	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

表 783. 分钟寄存器 （MIN – 地址 0x4004 6024）位描述

位	符号	描述	复位值
5:0	MINUTES	分钟值，范围 0 至 59	-[1]
31:6	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

表 784. 小时寄存器 （HRS – 地址 0x4004 6028）位描述

位	符号	描述	复位值
4:0	HOURS	小时值，范围 0 至 23	-[1]
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

表 785. 月日期寄存器 （DOM – 地址 0x4004 602C）位描述

位	符号	描述	复位值
4:0	DOM	月日期值，范围 1 至 28、29、30、31 （取决于月份大小以及是否是闰年）。	-[1]
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

表 786. 周日期寄存器 （DOW – 地址 0x4004 6030）位描述

位	符号	描述	复位值
2:0	DOW	周日期值，范围 0 至 6。	-[1]
31:3	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

表 787. 年日期寄存器 （DOY – 地址 0x4004 6034）位描述

位	符号	描述	复位值
8:0	DOY	年日期值，范围 1 至 365 （闰年是 366）。	-[1]
31:9	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 788. 月寄存器 （MONTH – 地址 0x4004 6038）位描述

位	符号	描述	复位值
3:0	MONTH	月份值，范围 1 至 12。	-[1]
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

表 789. 年寄存器 （YEAR – 地址 0x4004 603C）位描述

位	符号	描述	复位值
11:0	YEAR	年份值，范围 0 至 4095。	-[1]
31:12	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

35.6.6.1 闰年计算

RTC 进行简单位比较来检查年份计数器的两个最低阶位是否为零。如果是，则 RTC 即认为该年是闰年。RTC 认为能被 4 整除的所有年份都是闰年。此算法在 1901 年至 2099 年之间是正确的，但在 2100 年错误，该年并非闰年。闰年对 RTC 的唯一影响是更改该年二月的长度、二月的天数以及年份计数器。

35.6.6.2 校准寄存器

以下寄存器用于校准时间计数器，在此寄存器中的位不通过复位改变。

表 790. 校准寄存器 （CALIBRATION – 地址 0x4004 6040）位描述

位	符号	值	描述	复位值
16:0	CALVAL	-	使能后，校准计数器向上计数至此值。最大值是 131072，相当于约 36.4 小时。如果 CALVAL = 0，则禁用校准。	-[1]
17	CALDIR		校准方向	-[1]
		0	向前校准。CALVAL 等于校准计数器时，RTC 定时器将以 2 秒跳跃。	
		1	向后校准。 CALVAL 等于校准计数器时， RTC 定时器将停止递增 1 秒。	
31:18	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

35.6.7 报警寄存器组

报警寄存器如表 791 中所示。这些寄存器中的值与时间计数器比较。如果所有未掩码（参见 第 770 页上的章节 35.6.4 “报警掩码寄存器”）报警寄存器都与其相应的时间计数器匹配，则生成中断。向 “中断位置寄存器 (ILR[1])” 的 1 位写入 1 可清除中断。

表 791. 报警寄存器

名称	Size	描述	访问类型	地址
ALSEC	6	秒报警值	R/W	0x4004 6060
ALMIN	6	分钟报警值	R/W	0x4004 6064
ALHRS	5	小时报警值	R/W	0x4004 6068
ALDOM	5	月日期报警值	R/W	0x4004 606C
ALDOW	3	周日期报警值	R/W	0x4004 6070
ALDOY	9	年日期报警值	R/W	0x4004 6074
ALMON	4	月份报警值	R/W	0x4004 6078
ALYEAR	12	年份报警值	R/W	0x4004 607C

表 792. 报警秒寄存器 （ASEC – 地址 0x4004 6060）位描述

位	符号	描述	复位值
5:0	SECONDS	秒值，范围 0 至 59	-[1]
31:6	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

表 793. 报警分钟寄存器 （AMIN – 地址 0x4004 6064）位描述

位	符号	描述	复位值
5:0	MINUTES	分钟值，范围 0 至 59	-[1]
31:6	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

表 794. 报警小时寄存器 （AHRS – 地址 0x4004 6068）位描述

位	符号	描述	复位值
4:0	HOURS	小时值，范围 0 至 23	-[1]
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 795. 报警月日期寄存器 （ADOM – 地址 0x4004 606C）位描述

位	符号	描述	复位值
4:0	DOM	月日期值，范围 1 至 28、29、30、31（取决于月份大小以及是否是闰年）。	-[1]
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

表 796. 报警周日期寄存器 （ADOW – 地址 0x4004 6070）位描述

位	符号	描述	复位值
2:0	DOW	周日期值，范围 0 至 6。	-[1]
31:3	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

表 797. 报警年日期寄存器 （ADOY – 地址 0x4004 6074）位描述

位	符号	描述	复位值
8:0	DOY	年日期值，范围 1 至 365 （闰年是 366）。	-[1]
31:9	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

表 798. 报警月寄存器 （AMON – 地址 0x4004 6078）位描述

位	符号	描述	复位值
3:0	MONTH	月份值，范围 1 至 12。	-[1]
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

表 799. 报警年寄存器 （AYRS – 地址 0x4004 607C）位描述

位	符号	描述	复位值
11:0	YEAR	年份值，范围 0 至 4095。	-[1]
31:12	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

[1] 此寄存器值不能通过复位改变。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

35.7 功能说明

35.7.1 校准步骤

校准逻辑可通过不递增计数器或者按 2 而非 1 递增计数器来周期性地调节时间计数器，这样就可在某些特定电压和温度条件下校准 RTC 振荡器，无需外部微调 RTC 振荡器。

确定校准值的推荐方法是使用 CLKOUT 功能，无干扰地监视 RTC 振荡器在需要微调的条件下的频率，然后计算要减去一秒的时间前的时钟数。该值即用于确定 CALVAL。

确切的校准方法取决于 CALVAL 是奇数还是偶数。对于奇数值，硬件循序多次执行一种两次校准 （一次校准在 CALVAL+1 时，一次在 CALVAL-1 时），然后平均两个校准值。对于偶数 CALVAL 值，校准时间是准确的。

如果要对 RTC 振荡器进行外部微调，则无干扰监视 RTC 振荡器频率的方法对该过程可能有所帮助。

向后校准

在 CCR 寄存器中使能 RTC 定时器和校准（设置位 CLKEN = 1 及 CCALEN = 0）。在校准寄存器中，设置校准值 CALVAL ≥ 1 并选择 CALDIR = 1。

- 对于每个 1 Hz 时钟周期， SEC 定时器和校准计数器都向上计数。
- 校准计数器达到 CALVAL 时，发生校准匹配，且所有 RTC 定时器都将停止一个时钟周期，以便定时器在下一周期不会递增。
- 如果在校准匹配的同一周期发生警报匹配事件，则警报中断将延迟一个周期，以避免双警报中断。

向前校准

在 CCR 寄存器中使能 RTC 定时器和校准（设置位 CLKEN = 1 及 CCALEN = 0）。在校准寄存器中，设置校准值 CALVAL ≥ 1 并选择 CALDIR = 0。

- 对于每个 1 Hz 时钟周期， SEC 定时器和校准计数器都向上计数。
- 校准计数器达到 CALVAL 时，发生校准匹配，且 RTC 定时器以 2 递增。
- 发生校准事件时，ALSEC 寄存器的 LSB 被强制为 1，以便当跳跃一秒时警报中断不会被漏过。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

36.1 本章导读

USART0/2/3 控制器可在所有 LPC43xx 器件上使用。

36.2 基本配置

USART0/2/3 按如下方式进行配置：

- 参见表 800 以了解时钟与电源控制。
- USART0/2/3 由 UART0/2/3_RST 复位 （复位 #44/46/47）。
- USART0/2/3 中断连接到 NVIC 中的插槽 # 24/26/27。
- 要将USART0/2/3接收和发送线连接至GPDMA，请使用CREG模块中的DMAMUX寄存器 （参见表 41）并使能 DMA 通道配置寄存器 （第 19.6.20 节）中的 GPDMA 通道。
- 内部USART发送/接收有效信号(USARTx TX/RX)连接至GIMA（参见表 138）。这些信号在数据发送或接收时被驱动，可用于启动任何定时器或 SCT。

表 800. USART0/2/3 时钟和电源控制

	基本时钟	分支时钟	工作频率
到寄存器接口的 USART0 时钟	BASE_M4_CLK	CLK_M4_UART0	高达 204 MHz
USART0 外设时钟 (PCLK)	BASE_UART0_CLK	CLK_APB0_UART0	高达 204 MHz
到寄存器接口的 USART2 时钟	BASE_M4_CLK	CLK_M4_UART2	高达 204 MHz
USART2 外设时钟 (PCLK)	BASE_UART2_CLK	CLK_APB2_UART2	高达 204 MHz
到寄存器接口的 USART3 时钟	BASE_M4_CLK	CLK_M4_UART3	高达 204 MHz
USART3 外设时钟 (PCLK)	BASE_UART3_CLK	CLK_APB2_UART3	高达 204 MHz

36.3 功能要点

- 16 字节接收和发送 FIFO。
- 寄存器位置符合 550 工业标准。
- 接收器 FIFO 启动点在 1、 4、 8 和 14 字节。
- 内置波特率发生器。
- UART 模式中的 UART 可实现软件或硬件流控制。
- RS-485/EIA-485 9 位模式支持输出使能。
- 支持同步模式 UART (USART)。
- IrDA 接口 （仅限 USART3）。
- DMA 支持。
- 智能卡接口。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

36.4 简介

USART 的架构如以下框图所示。

APB 接口提供了 CPU 或主机与 USART 之间的通信链接。

USART 接收器模块 RX 监控串行输入线路， RXD 用于有效输入。 USART RX 移位寄存器 (RSR) 通过 RXD 接受有效字符。有效字符汇编到 RSR 中后，被传递到 USART RX 缓冲寄存器 FIFO，等待 CPU 或主机通过通用主机接口进行访问。

USART 发送器模块 TX 接受 CPU 或主机写入的数据，并将数据缓冲在 USART TX 保持寄存器 FIFO (THR) 中。USART TX 移位寄存器 (TSR) 读取 THR 中存储的数据，并汇编数据以通过串行输出引脚 TXD1 发送。

USART 波特率发生器模块 BRG 生成由 USART TX 模块使用的定时。 BRG 时钟输入源是 PCLK。主时钟被 DLL 和 DLM 寄存器中指定的除数分频。此分频后的时钟是 16 倍过采样时钟， NBAUDOUT。

中断接口包含寄存器 IER 和 IIR。中断接口接收从 TX 和 RX 块使能的多个一时钟宽度。

TX 和 RX 的状态信息存储在 LSR 中。 TX 和 RX 的控制信息存储在 LCR 中。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 108. USART 框图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

36.5 引脚说明

表 801. USART0/2/3 引脚描述

功能名称	方向	描述
USART0		
U0_RXD	I	**串行输入。**串行接收数据。
U0_TXD	O	**串行输出。**串行发送数据。
U0_DIR	I/O	RS-485/EIA-485 输出使能 / 方向控制。
U0_UCLK	I/O	同步模式下 USART0 的串行时钟输入 / 输出。
USART2		
U2_RXD	I	**串行输入。**串行接收数据。
U2_TXD	O	**串行输出。**串行发送数据。
U2_DIR	I/O	RS-485/EIA-485 输出使能 / 方向控制。
U2_UCLK	I/O	同步模式下 USART2 的串行时钟输入 / 输出。
USART3		
U3_RXD	I	**串行输入。**串行接收数据。
U3_TXD	O	**串行输出。**串行发送数据。
U3_DIR	I/O	RS-485/EIA-485 输出使能 / 方向控制。
U3_UCLK	I/O	同步模式下 USART3 的串行时钟输入 / 输出。
U3_BAUD	O	USART3 波特率输出。 U3_BAUD 是单时钟周期的有效低电平信号，在 UART 发送器部分的 16x 时钟信号的每个上升沿生成。时钟频率由小数分频器分频的 USART3 时钟频率确定，且除数在波特率生成器除数锁存器中指定。 U3_BAUD 可用作外部 IrDA 模块的输入。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

36.6 寄存器描述

USART 所包含的寄存器，其结构如表 802 所示。除数锁存器访问位 (DLAB) 包含在 LCR[7] 中，可实现对除数锁存器的访问。

复位值仅反映存储在所用位中的数据，不包含保留位中的内容。

表 802. 寄存器简介：USART0/2/3 （基址：0x4008 1000、 0x400C 1000、 0x400C 2000）

名称	访问类型	地址偏移	描述	复位值	参考
RBR	RO	0x000	接收缓冲寄存器。包含要读取的下一收到的字符 (DLAB = 0)。	不适用	表 803
THR	WO	0x000	发送保持寄存器。在此写入下一要发送的字符 (DLAB = 0)。	不适用	表 804
DLL	R/W	0x000	除数锁存 LSB。波特率除数值的最低有效字节。全除数用于从小数频率分频器生成波特率 (DLAB = 1)。	0x01	表 805
DLM	R/W	0x004	除数锁存 MSB。波特率除数值的最高有效字节。全除数用于从小数频率分频器生成波特率 (DLAB = 1)。	0x00	表 806
IER	R/W	0x004	中断使能寄存器。包含 7 个潜在 USART 中断的单独中断使能位 (DLAB = 0)。	0x00	表 807
IIR	RO	0x008	中断 ID 寄存器，识别要挂起的中断。	0x01	表 808
FCR	WO	0x008	FIFO 控制寄存器，控制 USART FIFO 的使用和模式。	0x00	表 810
LCR	R/W	0x00C	线路控制寄存器。包含对帧格式化和断点产生的控制。	0x00	表 811
-	-	0x010	保留。	-	-
LSR	RO	0x014	线路状态寄存器，包含发送和接收状态 （包括线路错误）的标志。	0x60	表 812
-	-	0x018	保留。	-	-
SCR	R/W	0x01C	暂存寄存器，软件的八位临时存储器。	0x00	表 813
ACR	R/W	0x020	自动波特率控制寄存器，包含自动波特率功能的控制。	0x00	表 814
ICR	R/W	0x024	IrDA 控制寄存器 （仅限 USART3）。	0x00	表 815
FDR	R/W	0x028	小数分频寄存器，为波特率分频器生成时钟输入。	0x10	表 817
OSR	R/W	0x02C	过采样寄存器。控制每个位时间内的过采样度。	0xF0	表 819
-	-	0x030 - 0x03C	保留。	-	-
HDEN	R/W	0x040	半双工使能寄存器		表 820
-	-	0x044	保留。	-	-
SCICTRL	R/W	0x048	智能卡接口控制寄存器	0x00	表 821
RS485CTRL	R/W	0x04C	RS-485/EIA-485 控制，包含对配置 RS-485/EIA-485 模式各个方面的控制。	0x00	表 822
RS485ADRMATCH	R/W	0x050	RS-485/EIA-485 地址匹配，包含 RS-485/EIA-485 模式的地址匹配值。	0x00	表 823
RS485DLY	R/W	0x054	RS-485/EIA-485 方向控制延迟。	0x00	表 824
SYNCCTRL	R/W	0x058	同步模式控制寄存器。	0x00	表 825
TER	R/W	0x05C	发送使能寄存器。关闭 USART 发送器，以使用软件流控制。	0x01	表 826

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

36.6.1 USART 接收缓冲寄存器

RBR 是 USART RX FIFO 的最高字节。RX FIFO 最高字节包含了接收到的最旧的字符，可以通过总线接口读取。LSB（位 0）代表接收到的最旧的数据位。如果接收到的字符少于 8 位，则未使用的 MSB 用 0 填充。

要访问 RBR， LCR 中的除数锁存访问位 (DLAB) 必须为零。 RBR 始终为 “只读”。

由于奇偶校验错误 (PE)、帧处理错误 (FE) 和中止中断 (BI) 位（参见表 812）与 RBR FIFO 顶部的字节（即下次从 RBR 读取时要读取的字节）相对应，因此，要正确提取有效的接收字节对及其状态位，应先读取 LSR 寄存器的内容，然后再读取 RBR 中的字节。

表 803. USART 接收缓冲寄存器在 DLAB = 0、只读时（RBR – 地址 0x4008 1000 (USART0)、0x400C 1000 (USART2)、 0x400C 2000 (USART3)）位描述

位	符号	描述	复位值
7:0	RBR	接收缓冲。 USART 接收缓冲寄存器包含在 USART RX FIFO 中最早收到的字节。	未定义
31:8	-	保留。	-

36.6.2 USART 发送器保持寄存器

THR 是 USART TX FIFO 的最高字节。最高字节是 TX FIFO 中最新的字符，可通过总线接口写入。 LSB 代表第一个要发送的位。

要访问 THR， LCR 中的除数锁存访问位 (DLAB) 必须为零。 THR 始终为 “只写”。

表 804. USART 发送器保持寄存器在 DLAB = 0、只写时（THR – 地址 0x4008 1000 (USART0)、0x400C 1000 (USART2)、 0x400C 2000 (USART3)）位描述

位	符号	描述	复位值
7:0	THR	发送保持寄存器。 写入 USART 发送保持寄存器的数据将存储在 USART 发送 FIFO 中。字节将在到达 FIFO 底部且发送器可用时发出。	不适用
31:8	-	保留。	-

36.6.3 USART 除数锁存 LSB 和 MSB 寄存器

USART 除数锁存是 USART 波特率发生器的一部分，持有所用的值以及小数分频器，以对 USART_PCLK 时钟进行分频，产生必须是 16x 所需波特率的波特率时钟。DLL 和 DLM 寄存器一起构成了一个 16 位除数，其中 DLL 包含了除数的低 8 位，而 DLM 包含了除数的高 8 位。0x0000 值按 0x0001 值处理，因为不允许被零除。LCR 中的除数锁存访问位 (DLAB) 必须为 1，才能访问 USART 除数锁存。有关如何为 DLL 和 DLM 选择正确值的详细信息，请参阅第 36.6.12 节。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 805. USART 除数锁存器 LSB 寄存器在 DLAB = 1 时 （DLL － 地址 0x4008 1000 (USART0)、0x400C 1000 (USART2)、 0x400C 2000 (USART3)）位描述

位	符号	描述	复位值
7:0	DLLSB	除数锁存 LSB。 USART 除数锁存器 LSB 寄存器与 DLM 寄存器共同决定 USART 的波特率。	0x01
31:8	-	保留。	-

表 806. USART 除数锁存器 MSB 寄存器在 DLAB = 1 时（DLM － 地址 0x4008 1004 (USART0)、0x400C 1004 (USART2)、 0x400C 2004 (USART3)）位描述

位	符号	描述	复位值
7:0	DLMSB	除数锁存 MSB。 USART 除数锁存器 MSB 寄存器与 DLL 寄存器共同决定 USART 的波特率。	0x00
31:8	-	保留。	-

36.6.4 USART 中断使能寄存器

IER 用于使能四个 USART 中断源。

表 807. USART 中断使能寄存器在 DLAB = 0 时（IER － 地址 0x4008 1004 (USART0)、0x400C 1004 (USART2)、 0x400C 2004 (USART3)）位描述

位	符号	值	描述	复位值
0	RBRIE		RBR 中断使能。 为 USART 使能接收数据可用中断。也控制字符接收超时中断。	0
		0	禁用 RDA 中断。	
		1	使能 RDA 中断。	
1	THREIE		THRE 中断使能。 为USART使能THRE中断。此中断的状态可从LSR[5]读取。	0
		0	禁用 THRE 中断。	
		1	使能 THRE 中断。	
2	RXIE		RX 线路中断使能。 使能USART RX线路状态中断。此中断的状态可从LSR[4:1]读取。	0
		0	禁用 RX 线路状态中断。	
		1	使能 RX 线路状态中断。	
3	-	-	保留。	-
6:4	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
7	-	-	保留。	0
8	ABEOINTEN		使能自动波特率结束中断。	0
		0	禁用自动波特率结束中断。	
		1	使能自动波特率结束中断。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 807. USART 中断使能寄存器在 DLAB = 0 时（IER – 地址 0x4008 1004 (USART0)、0x400C 1004 (USART2)、 0x400C 2004 (USART3)）位描述 *（续）*

位	符号	值	描述	复位值
9	ABTOINTEN		使能自动波特率超时中断。	0
		0	禁用自动波特率超时中断。	
		1	使能自动波特率超时中断。	
31:10	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

36.6.5 USART 中断识别寄存器

IIR 提供一个状态代码，表示挂起中断的优先级和源，这些中断在 IIR 访问期间被冻结。如果在 IIR 访问期间出现中断，则为下次 IIR 访问记录该中断。

表 808. USART 中断识别寄存器在只读时 （IIR – 地址 0x4008 1008 (USART0)、 0x400C 1008 (USART2)、 0x400C 2008 (USART3)）位描述

位	符号	值	描述	复位值
0	INTSTATUS		中断状态。 请注意， IIR[0] 低电平激活。挂起中断可通过评估 IIR 来确定 [3:1]。	1
		0	至少一个中断挂起。	
		1	没有中断挂起。	
3:1	INTID		中断识别。 IER[3:1] 识别与 USART Rx FIFO 对应的中断。以下没有列出的所有其他 IER[3:1] 组合都是保留的 (100,101,111)。	0
		0x3	优先级 1 （最高） - 接收线路状态 (RLS)。	
		0x2	优先级 2 - 接收数据可用 (RDA)。	
		0x6	优先级 2 – 字符超时指示器 (CTI)。	
		0x1	优先级 3 – THRE 中断。	
		0x0	优先级 4 （最低） – 保留。	
5:4	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
7:6	FIFOENABLE		FCR[0] 的副本。	0
8	ABEOINT		自动波特率结束中断。 自动波特率成功完成且中断使能时为真。	0
9	ABTOINT		自动波特率超时中断。 如果自动波特率超时且使能中断，则该值为真。	0
31:10	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

位 IIR[9:8] 由自动波特率功能设置，并发出超时信号或自动波特率条件结束信号。通过设置自动波特率控制寄存器中的相应 “清除” 位，清除自动波特率中断条件。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

如果 IntStatus 位为 1 且没有中断挂起，则 IntId 位将为零。如果 IntStatus 为 0，表示有一个非自动波特率中断被挂起，此时 IntId 位会却确定中断的类型和处理方式，如表 809 中所述。如果指定 IIR[3:0] 的状态，中断处理程序就能确定中断原因以及如何清除有效的中断。必须读取 IIR，才能在退出中断服务例程前清除中断。

USART RLS 中断 (IIR[3:1] = 011) 为最高优先级中断，每当在 USART RX 输入端发生下面 4 个错误状况中的任意一个时，都可以进行设置：溢出错误 (OE)、奇偶校验错误 (PE)、帧处理错误 (FE) 和中止中断 (BI)。设置中断的 USART Rx 错误状况可通过 LSR[4:1] 查看。该中断在读取 LSR 时被清除。

USART RDA 中断 (IIR[3:1] = 010) 与 CTI 中断 (IIR[3:1] = 110) 均为第二优先级。 RDA 在 USART Rx FIFO 达到 FCR7:6 中定义的启动级别时激活，并在 USART Rx FIFO 降低到启动级别以下时复位。 RDA 中断激活时， CPU 可读取启动级别定义的数据块。

CTI 中断 (IIR[3:1] = 110) 为第二优先级中断，在 USART RX FIFO 含有至少一个字符并且在接收到 3.5 到 4.5 个字符的时间内没有发生任何 USART RX FIFO 操作时，可对该中断进行设置。任何 USART Rx FIFO 操作 （读或写 USART RSR）都将清除中断。此中断的目的是在收到一条大小不是启动级别大小的倍数的消息后，刷新 USART RBR。例如，如果外设要发送一条 105 个字符的消息，并且启动级别为 10 个字符，则 CPU 将接收 10 次 RDA 中断以传输 100 个字符，还将接收 1 到 5 次 CTI 中断 （取决于服务例程）以传输剩余的 5 个字符。

表 809. USART 中断处理

IIR[3:0] 值[1]	优先级	中断类型	中断源	中断复位
0001	-	无	无	-
0110	最高	RX线路状态 / 错误	OE[2]、 PE[2]、 FE[2] 或 BI[2]	LSR 读操作[2]
0100	第二	RX数据可用	Rx 数据可用或启动级别达到 FIFO (FCR0=1)	RBR 读操作[3]或USART FIFO低于启动级别
1100	第二	字符超时指示	RX FIFO 中至少有一个字符且一段时间内无字符输入或者无字符删除，取决于 FIFO 中有多少字符和启动级别的设置 （3.5 至 4.5 字符时间）。 精确时间为： [（字长度）× 7 - 2] × 8 + [（启动电平 - 字符数）× 8 + 1] RCLKs	RBR 读操作[3]
0010	第三	THRE	THRE[2]	IIR 读操作[4]（如果是中断源）或THR写操作

[1] 值 0000、 0011、 010、 0111、 1000、 1001、 1010、 1011、 1101、 1110、 1111 是保留的。

[2] 有关详情，请参阅 第 36.6.8 节 “USART 线路状态寄存器”。

[3] 有关详情，请参阅 第 36.6.1 节 “USART 接收缓冲寄存器”。

[4] 有关详情，请参阅 第 36.6.5 节 “USART 中断识别寄存器” 和 第 36.6.2 节 “USART 发送器保持寄存器”。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

USART THRE 中断 (IIR[3:1] = 001) 为第三优先级中断，当 USART THR FIFO 为空，且满足特定的初始化条件时，该中断激活。这些初始化条件的目的是为 USART THR FIFO 提供填充数据的机会，以消除在系统启动时出现的许多 THRE 中断。当 THRE = 1 时，且在上次的 THRE = 1 事件后， THR 中没有出现至少两个字符时，这些初始化条件就会实现一个字符减去停止位的延时。提供此延迟的目的是让CPU有时间将数据写入THR，而无需THRE中断来进行解码和服务。如果 USART THR FIFO 曾一次持有两个或多个字符，并且 THR 当前被清空，则 THRE 中断立即设置。当发生 THR 写操作或 IIR 读操作，且 THRE 为最高优先级中断 (IIR[3:1] = 001) 时， THRE 中断复位。

36.6.6 USART FIFO 控制寄存器

FCR 控制 USART RX 和 TX FIFO 的操作。

表 810. USART FIFO 控制寄存器在只写时（FCR – 地址 0x4008 1008 (USART0)、0x400C 1008 (USART2)、 0x400C 2008 (USART3)）位描述

位	符号	值	描述	复位值
0	FIFOEN		FIFO 使能。	0
		0	USART FIFO 禁用，不得用于应用程序。	
		1	为 USART Rx 和 TX FIFO 以及 FCR[7:1] 访问使能有效高电平。必须设置此位， USART 才能正确操作。此位上的任何跳变都将自动清除 USART FIFO。	
1	RXFIFORES		RX FIFO 复位。	0
		0	对任一 USART FIFO 都没有影响。	
		1	将逻辑 1 写入 FCR[1] 将清除 USART Rx FIFO 中的所有字节，复位指针逻辑。此位是自清除的。	
2	TXFIFORES		TX FIFO 复位。	0
		0	对任一 USART FIFO 都没有影响。	
		1	将逻辑 1 写入 FCR[2] 将清除 USART TX FIFO 中的所有字节，复位指针逻辑。此位是自清除的。	
3	DMAMODE		DMA 模式选择。设置 FIFO 使能位（此寄存器的位 0）时，此位选择 DMA 模式。	0
5:4	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
7:6	RXTRIGLVL		RX 启动级别。 这两个位确定在激活中断前，必须写入多少接收器 USART FIFO 字符。	0
		0x0	启动级别 0 （1 个字符或 0x01）。	
		0x1	启动级别 1 （4 个字符或 0x04）。	
		0x2	启动级别 2 （8 个字符或 0x08）。	
		0x3	启动级别 3 （14 个字符或 0x0E）。	
31:8	-	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

36.6.6.1 DMA 操作

用户也可选择使用 DMA 来操作 USART 发送和 / 或接收。 DMA 模式由 FCR 寄存器中的 DMA 模式选择位确定。请注意，对于 DMA 操作，与 USART 的任何操作一样，都必须通过 FCR 寄存器中的 FIFO 使能位来使能 FIFO。

USART 接收器 DMA

在 DMA 模式下，当接收器 FIFO 级别等于或大于启动级别或出现字符超时时，接收器 DMA 请求被断言。请参见以上 RX 启动级别的说明。接收器 DMA 请求将被 DMA 控制器清除。

USART 发送器 DMA

在 DMA 模式下，发送器 DMA 请求在发送器 FIFO 跳变成不全满时被断言。发送器 DMA 请求将被 DMA 控制器清除。

36.6.7 USART 线路控制寄存器

LCR 确定要发送或接收的数据字符的格式。

表 811. USART线路控制寄存器（LCR － 地址0x4008 100C (USART0)、0x400C 100C (USART2)、0x400C 200C (USART3)）位描述

位	符号	值	描述	复位值
1:0	WLS		字长选择。	0
		0x0	5 位字符长度。	
		0x1	6 位字符长度。	
		0x2	7 位字符长度。	
		0x3	8 位字符长度。	
2	SBS		停止位选择。	0
		0	1 停止位。	
		1	2 停止位 （如果 LCR[1:0]=00，则为 1.5）。	
3	PE		奇偶校验使能。	0
		0	禁用奇偶校验生成和检查。	
		1	使能奇偶校验生成和检查。	
5:4	PS		奇偶校验选择。	0
		0x0	奇数校验。已发送字符中 1 的数量，附加的奇偶校验位将为奇数。	
		0x1	偶数校验。已发送字符中 1 的数量，附加的奇偶校验位将为偶数。	
		0x2	强制 "1" 粘附奇偶校验。	
		0x3	强制 "0" 粘附奇偶校验。	
6	BC		断点控制。	0
		0	禁用断点传输。	
		1	使能断点传输。LCR[6] 为高电平有效时，输出引脚 USART TXD 强制为逻辑 0。	
7	DLAB		除数锁存访问位。	0
		0	禁用对除数锁存的访问。	
		1	使能对除数锁存的访问。	
31:8	-	-	保留。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

36.6.8 USART 线路状态寄存器

LSR 是一个只读寄存器，提供有关 USART TX 和 RX 模块的状态信息。

表 812. USART 线路状态寄存器在只读时（LSR – 地址 0x4008 1014(USART0)、0x400C 1014 (USART2)、 0x400C 2014 (USART3)）位描述

位	符号	值	描述	复位值
0	RDR		接收器数据就绪。 当 RBR 包含未读字符时，LSR[0] 会被设置；当 USART RBR FIFO 为空时， LSR[0] 会被清除。	0
		0	RBR 被清空。	
		1	RBR 包含有效数据。	
1	OE		超限错误。 超限错误一发生，即会设置超限错误条件。 LSR 读取将清除 LSR[1]。在 USART RSR 汇编了新字符且 USART RBR FIFO 为满时，设置 LSR[1]。在此情况下，USART RBR FIFO 将不会被覆盖，且 USART RSR 中的字符将丢失。	0
		0	超限错误状态未激活。	
		1	超限错误状态激活。	
2	PE		奇偶错误。 当收到字符的奇偶位处于错误的状态时，将会产生一个奇偶错误。LSR读取将清除LSR[2]。奇偶校验错误检测的时间取决于FCR[0]。 **注：**奇偶检验错误与 USART RBR FIFO 顶部的字符相关。	0
		0	奇偶校验错误状态未激活。	
		1	奇偶校验错误状态激活。	
3	FE		帧处理错误。 当收到字符的停止位为逻辑 0 时，将产生一个帧处理错误。LSR 读取将清除 LSR[3]。帧处理错误检测的时间取决于 FCR0。当检测到有帧处理错误时， RX 会尝试与数据重新同步，并假设错误的停止位实际是一个超前的起始位。当然，不能假定下一个收到的字节是正确的，即使没有帧处理错误。 **注：**帧处理错误与 USART RBR FIFO 顶部的字符相关。	0
		0	帧处理错误状态未激活。	
		1	帧处理错误状态激活。	
4	BI		中止中断。 对于一个完整的字符发送 （开始、数据、优先级、停止）， RXD1 保持在间距状态（全部为零）时，发生中止中断。检测到断点条件后，接收器即进入空闲，直至 RXD1 进入标记状态（全部为 1）为止。 LSR 读取将清除此状态位。断点检测的时间取决于 FCR[0]。 **注：**中止中断与 USART RBR FIFO 顶部的字符相关。	0
		0	中止中断状态未激活。	
		1	中止中断状态激活。	
5	THRE		发送保持寄存器空。 THRE在检测到空USART THR时立即设置，并在THR写入时清除。	1
		0	THR 包含有效数据。	
		1	THR 为空。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 812. USART 线路状态寄存器在只读时（LSR – 地址 0x4008 1014(USART0)、0x400C 1014 (USART2)、 0x400C 2014 (USART3)）位描述 *（续）*

位	符号	值	描述	复位值
6	TEMT		发送器空。 TEMT 在 THR 和 TSR 都为空时设置； TEMT 在 TSR 或 THR 包含有效数据时被清除。	1
		0	THR 和 / 或 TSR 包含有效数据。	
		1	THR 和 TSR 为空。	
7	RXFE		RX FIFO 中的错误。 当一个含有 RX 错误 （例如帧处理错误、奇偶校验错误或中止中断）的字符被载入 RBR 时，将设置 LSR[7]。此位在 LSR 寄存器被读取且 USART FIFO 中没有后续错误时被清除。	0
		0	RBR 不包含 USART RX 错误或 FCR[0]=0。	
		1	USART RBR 包含至少一个 USART RX 错误。	
8	TXERR		被发送字符出错。 NACK 响应由处于智能卡 T=0 模式的接收器给出。此位在 LSR 寄存器被读取时被清除。	0
		0	无错误 （正常默认条件）。	
		1	在智能卡 T=0 操作期间收到 NACK 响应。	
31:9	-	-	保留。	-

36.6.9 USART 暂存寄存器

SCR 对 USART 操作没有影响。该寄存器可由用户决定写入和 / 或读取。中断接口中没有向主机指明已发生 SCR 读取或写入的规范。

表 813. USART 暂存寄存器 （SCR – 地址 0x4008 101C (USART0)、 0x400C 101C (USART2)、 0x400C 201C (USART3)）位描述

位	符号	描述	复位值
7:0	PAD	暂存。一个可读、可写的字节。	0x00
31:8	-	保留。	-

36.6.10 USART 自动波特率控制寄存器

USART 自动波特率控制寄存器 (ACR) 控制为波特率生成测量输入时钟 / 数据速率的过程，并且可由用户决定读取和写入。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 814. 自动波特率控制寄存器（ACR – 地址 0x4008 1020 (USART0)、0x400C 1020 (USART2)、0x400C 2020 (USART3)）位描述

位	符号	值	描述	复位值
0	START		起始位。 该位在自动波特率完成之后自动清除。	0
		0	自动波特率停止 （自动波特率不运行）。	
		1	自动波特率起动 （自动波特率运行） 。自动波特率运行位。自动波特率完成后此位自动清除。	
1	MODE		自动波特率模式选择位。	0
		0	模式 0。	
		1	模式 1。	
2	AUTORESTART		重启位。	0
		0	不重启。	
		1	如果超时则重新启动（计数器会在下一个 USART Rx 下降沿重新启动）。	
7:3	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	0
8	ABEOINTCLR		自动波特率结束中断清除位 （只写）。	0
		0	写入 0 没有影响。	
		1	写入 1 将清除 IIR 中的相应中断。	
9	ABTOINTCLR		自动波特率超时中断清除位 （只写）。	0
		0	写入 0 没有影响。	
		1	写入 1 将清除 IIR 中的相应中断。	
31:10	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	0

36.6.10.1 自动波特率

USART 自动波特率功能可用于基于 AT 协议 （Hayes 命令）测量输入的波特率。使能后，自动波特率功能将测量接收数据流的位时间，并相应设置除数锁存寄存器 DLM 和 DLL。

自动波特率通过设置 ACR 起始位来起动。可通过清除 ACR 起始位来停止自动波特率。自动波特率完成后起始位将清除，且读取位的操作将返回自动波特率的状态 （挂起 / 完成）。

有两种自动波特率测量模式可用，可通过 ACR 模式位选择。在模式 0 中，在 USART Rx 引脚的两个后续下降沿上测量波特率 （起始位的下降沿和最低有效位的下降沿） 。在模式 1 中，在 USART Rx 引脚的下降沿和后续上升沿之间测量波特率 （起始位的长度）。

出现超时时，ACR AutoRestart 位可用于自动重启波特率测量（速率测量计数器溢出）。如果设置了此位，则将在 USART Rx 引脚的下一下降沿重启速率测量。

自动波特率功能可产生两种中断。

- IIR ABTOInt 中断将在中断使能时设置 （IER ABToIntEn 设置，并且自动波特率测量计数器溢出）。
- IIR ABEOInt 中断将在中断使能时设置（IER ABEOIntEn 设置，并且自动波特率成功完成）。

必须通过设置相应的 ACR ABTOIntClr 和 ABEOIntEn 位来清除自动波特率中断。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

小数波特率发生器在自动波特率期间必须禁用 (DIVADDVAL = 0)。同样，使用自动波特率时，对 DLM 和 DLL 寄存器的任何写操作都必须在 ACR 寄存器写操作前完成。 USART 支持的最小和最大波特率是 USART_PCLK、数据位、停止位和奇偶校验位的数量的函数。

(6)

$$ratemin = \frac{2 \times PCLK}{16 \times 2^{15}} \leq UART_{baudrate} \leq \frac{PCLK}{16 \times (2 + databits + paritybits + stopbits)} = ratemax$$

36.6.10.2 自动波特率模式

软件要使用 AT 命令时，可用预期的字符格式配置 USART 并设置 ACR 起始位。无需考虑除数锁存器 DLM 和 DLM 中的初始值。由于 A 或者 a 的 ASCII 编码（“A" = 0x41, ” a" = 0x61），USART Rx 引脚检测到起始位，且预期字符的最低有效位由两个下降沿分隔。ACR 起始位设置后，自动波特率协议将执行以下几个阶段：

1. ACR 起始位设置时，波特率测量计数器被复位，USART RSR 也复位。RSR 波特率切换到最高速率。
2. USART Rx 引脚的下降沿启动起始位的开始。速率测量计数器将开始对 USART_PCLK 循环进行计数。
3. 接收起始位期间， RSR 波特率输入中产生 16 个脉冲，频率为 USART 输入时钟的频率，保证起始位存储在 RSR 中。
4. 在接收起始位的过程中（以及模式 = 0 下的字符 LSB），速率计数器将随着被预分频的 USART 输入时钟 (USART_PCLK) 递增。
5. 如果是模式 = 0，那么速率计数器将在 USART Rx 引脚的下一个下降沿停止。如果是模式 = 1，那么速率计数器将在 USART Rx 引脚的下一个上升沿停止。
6. 速率计数器被加载到 DLM/DLL 中，并且波特率将切换到正常操作。设置 DLM/DLL 后，自动波特率结束中断 IIR ABEOInt 将被设置（如果使能）。RSR 现在将继续接收 A/a 字符的剩余位。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

a. 模式 0（起始位和 LSB 均用于自动波特率）

b. 模式 1（仅起始位用于自动波特率）

图 109. 自动波特率 a) 模式 0 和 b) 模式 1 的波形图

36.6.11 IrDA 控制寄存器 (USART3)

IrDA 控制寄存器仅为 USART3 使能和配置 IrDA 模式。在发送和接收数据时 U3ICR 的值不能更改，否则会出现数据丢失或破坏。

注：IrDA 仅在 USART3 上可用。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 815. IrDA 控制寄存器 （ICR – 地址 0x4000 8024）位描述

位	符号	值	描述	复位值
0	IRDAEN		IrDA 模式使能。	0
		0	USART3上的IrDA模式禁用，USART3用作标准USART。	
		1	USART3 上的 IrDA 模式使能。	
1	IRDAINV		串行输入方向。	0
		0	串行输入不反向。	
		1	串行输入反向，这不会影响串行输出。	
2	FIXPULSEEN		IrDA 固定脉冲宽度模式。	0
		0	IrDA 固定脉冲宽度模式禁用。	
		1	IrDA 固定脉冲宽度模式使能。	
5:3	PULSEDIV		在 FixPulseEn = 1 时配置脉冲。有关详情，请参见表 816。	0
31:6	-	不适用	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	0

如果在 IrDA 模式下使用了固定脉冲宽度模式（IrDAEn = 1 且 FixPulseEn = 1），U3ICR 中的 PulseDiv 位用于选择脉冲宽度。用户必须对这些位的值进行设置，以便得到的脉冲宽度至少为 1.63 μs。表 816 显示了可能的脉冲宽度。

表 816. IrDA 脉冲宽度

FixPulseEn	PulseDiv	IrDA 发送器脉冲宽度 ((μs)
0	x	3 / (16 × 波特率)
1	0	$2 \times T_{PCLK}$
1	1	$4 \times T_{PCLK}$
1	2	$8 \times T_{PCLK}$
1	3	$16 \times T_{PCLK}$
1	4	$32 \times T_{PCLK}$
1	5	$64 \times T_{PCLK}$
1	6	$128 \times T_{PCLK}$
1	7	$256 \times T_{PCLK}$

36.6.12 USART 小数分频器寄存器 (U0FDR - 0x4000 8028)

USART 小数分频器寄存器 (FDR) 控制波特率生成时钟预分频器，可由用户决定读取或写入。预换算器采用 APB 时钟并根据指定的小数要求生成输出时钟。

注意事项：如果小数分频器有效 (DIVADDVAL > 0)，且 DLM = 0，则 DLL 寄存器的值必须大于或等于 3。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 817. USART 小数分频器寄存器（FDR – 地址 0x4008 1028 (USART0)、0x400C 1028 (USART2)、0x400C 2028 (USART3)）位描述

位	函数	描述	复位值
3:0	DIVADDVAL	波特率发生预换算器分频器的值。 如果此字段为0，则小数波特率发生器将不会影响USART波特率。	0
7:4	MULVAL	波特率预换算器乘数的值。 不管是否使用小数波特率生成器，该域值必须大于或等于 1 以保证 USART 正常运行。	1
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	0

此寄存器控制波特率发生的时钟预换算器。寄存器的复位值使 USART 的小数功能保持禁用，以确保 USART 在软件和硬件上，都与没有配备此功能的 USART 完全兼容。

USART 波特率的计算如下：

$$UART_{baudrate} = \frac{PCLK}{16 \times (256 \times DLM + DLL) \times \left(1 + \frac{DivAddVal}{MulVal}\right)} \quad (7)$$

其中， USART_PCLK 是外设时钟， DLM 和 DLL 是标准 USART 波特率分频器寄存器，DIVADDVAL 和 MULVAL 是 USART 小数波特率发生器的特定参数。

MULVAL 和 DIVADDVAL 的值应符合以下条件：

1. 1 ≤ MULVAL ≤ 15
2. 0 ≤ DIVADDVAL ≤ 14
3. DIVADDVAL< MULVAL

FDR 的值在发送 / 接收数据时不能修改，否则数据可能丢失或破坏。

如果 FDR 寄存器值不符合这两项要求，则小数分频器的输出是未定义的。如果 DIVADDVAL 为零，则小数分频器禁用，时钟不会被分频。

36.6.12.1 波特率计算

操作 USART 时可使用也可不使用小数分频器。在实际应用中，使用几个不同的小数分频器设置就可能获得所需的波特率。以下算法展示了一种查找一组 DLM、 DLL、 MULVAL 和 DIVADDVAL 值的方法。这组参数产生的波特率相对于所需波特率的误差小于 1.1%。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 110. 设置 USART 分频器的算法

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 818. 小数分频器设置查找表

FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal
1.000	0/1	1.250	1/4	1.500	1/2	1.750	3/4
1.067	1/15	1.267	4/15	1.533	8/15	1.769	10/13
1.071	1/14	1.273	3/11	1.538	7/13	1.778	7/9
1.077	1/13	1.286	2/7	1.545	6/11	1.786	11/14
1.083	1/12	1.300	3/10	1.556	5/9	1.800	4/5
1.091	1/11	1.308	4/13	1.571	4/7	1.818	9/11
1.100	1/10	1.333	1/3	1.583	7/12	1.833	5/6
1.111	1/9	1.357	5/14	1.600	3/5	1.846	11/13
1.125	1/8	1.364	4/11	1.615	8/13	1.857	6/7
1.133	2/15	1.375	3/8	1.625	5/8	1.867	13/15
1.143	1/7	1.385	5/13	1.636	7/11	1.875	7/8
1.154	2/13	1.400	2/5	1.643	9/14	1.889	8/9
1.167	1/6	1.417	5/12	1.667	2/3	1.900	9/10
1.182	2/11	1.429	3/7	1.692	9/13	1.909	10/11
1.200	1/5	1.444	4/9	1.700	7/10	1.917	11/12
1.214	3/14	1.455	5/11	1.714	5/7	1.923	12/13
1.222	2/9	1.462	6/13	1.727	8/11	1.929	13/14
1.231	3/13	1.467	7/15	1.733	11/15	1.933	14/15

36.6.12.1.1 示例 1：USART_PCLK = 14.7456 MHz、 BR = 9600

根据所提供的算法，DL_{est} = PCLK/(16 x BR) = 14.7456 MHz / (16 x 9600) = 96。因为这里的 DL_{est} 是一个整数，所以 DIVADDVAL = 0， MULVAL = 1， DLM = 0，且 DLL = 96。

36.6.12.1.2 示例 2：USART_PCLK = 12 MHz、 BR = 115200

根据所提供的算法，DL_{est} = PCLK/(16 x BR) = 12 MHz / (16 x 115200) = 6.51。此处 DL_{est} 不是整数，下一步可以估测 FR 参数。使用 FR_{est} = 1.5 进行首次估算，得到新的 DL_{est} = 4，然后使用 FR_{est} = 1.628 再进行计算。由于 FRest = 1.628 是在 1.1 到 1.9 的指定范围之内，因此 DIVADDVAL 和 MULVAL 的值可通过附带的查找表获得。

在查找表表 818 中，最接近 FRest = 1.628 的值为 FR = 1.625。也就是说 DIVADDVAL = 5 而 MULVAL = 8。

根据这些查到的值，建议的 USART 设置应为：DLM = 0, DLL = 4, DIVADDVAL = 5 和 MULVAL = 8. 根据方程 7，USART 的波特率为 115384。该速率与原来指定的 115200 之间存在 0.16% 的相对误差。

36.6.13 USART 过采样寄存器

在多数应用中，USART 在每个名义位时间对已接收数据进行 16 次采样，并发送 16 个输入时钟宽的位。该寄存器允许软件控制输入时钟和位时钟的比值。过采样是智能卡模式所必需的，并为其他模式的小数分频提供替代方式。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 819. USART 过采样寄存器（OSR － 地址 0x4008 102C (USART0)、0x400C 102C (USART2)、0x400C 20402C (USART3)）位描述

位	符号	描述	复位值
0	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
3:1	OSFRAC	过采样率的小数部分，以输入时钟周期的八分之一为单位。 (001 = 0.125, ..., 111 = 0.875)	0
7:4	OSINT	过采样率的整数部分，减去 1。复位值等于每个位时间 16 个输入时钟的正常操作模式。	0xF
14:8	FDINT	在智能卡模式中，这些位可以作为 OSint 字段的最有效扩展，支持 ISO7816-3 要求的最高 2048 的过采样率。在智能卡模式中，位 14:4 最初应设置为 371，产生 372 的过采样率。	0
31:15	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

示例：对于 24 MHz USART 时钟频率的 3.25 Mbps 波特率，理想的过采样率为 24/3.25 或 7.3846。为 7 时钟 / 位设置 OSINT 为 0110 （二进制），为 0.375 时钟 / 位设置 OSFrac 为 011 （二进制），得到 7.375 的过采样率。

在智能卡模式中， OSInt 由 FDINT 扩展。这将让可能的采样扩展到 2048 （符合支持 ISO 7816-3 的要求）。请注意，当 D<0 时，该值可能被超过，但不会得到 USART 支持。使能智能卡模式时，OSINT 和 FDINT 的初始值应编程为 00101110011（二进制）（372 减去 1）。

36.6.14 USART 半双工使能寄存器

注：在智能卡模式下，应禁用 HDEN 寄存器（默认情况下，智能卡在半双工模式下运行）。

复位后， USART 将处于全双工模式，意味着 TX 和 RX 都独立工作。设置 HDEN 位后，USART 将处于半双工模式。在此模式下， USART 确保接收器在空闲时锁定，或在接收完正在进行的完整字符后进入锁定状态。线路冲突必须在软件中处理。正在发送数据时，如果出现要接收的数据，则 USART 的行为是不可预测的。

因此， HDEN 寄存器的值在发送或接收数据时不能修改，否则数据可能丢失或破坏。

表 820. USART半双工使能寄存器（HDEN － 地址0x4008 1040 (USART0), 0x400C 1040 (USART2), 0x400C 2040 (USART3)）位描述

位	符号	值	描述	复位值
0	HDEN		半双工模式使能	0
		0	禁用半双工模式。	
		1	使能半双工模式。	
31:1	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

36.6.15 USART 智能卡接口控制寄存器

表 821. USART 智能卡接口控制寄存器（SCICTRL – 地址 0x4008 1048(USART0), 0x400C 1048 (USART2), 0x400C 2048 (USART3)）位描述

位	符号	值	描述	复位值
0	SCIEN		智能卡接口使能。	0
		0	智能卡接口禁用。	
		1	异步半双工智能卡接口使能。	
1	NACKDIS		NACK 响应禁用。仅在 T=0 时可用。	0
		0	NACK 响应使能。	
		1	NACK 响应禁止。	
2	PROTSEL		按 ISO7816-3 标准的定义进行协议选择。	0
		0	T = 0	
		1	T = 1	
7:5	TXRETRY		出现否定确认时的最大重发数（协议 T=0）。超过重试计数器时， USART 将被锁定，直至 FIFO 被清除。使能时生成一个 TX 错误中断。	-
15:8	GUARDTIME		额外保护时间。无额外保护时间 (0x0) 会导致 ISO 7816-3 所定义的标准保护时间，取决于协议类型。保护时间0xFF 表示为选定协议所定义的最小保护时间。	
31:16	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

复位后， USART 智能卡接口将被禁用。设置 SCIEN 位后， USART 将处于与 ISO 7816-3 兼容的异步智能卡模式 T=0。

NACKDIS 位用于禁止 T=0 期间的 NACK 响应 （在保护时间期间不会拉低 I/O 线路，以指示错误接收）。收到的字符将存储在 RX FIFO 中，但会生成一个奇偶校验错误。接收到的不正确字符由软件处理。

PROTSEL 位用于在两个支持的智能卡协议 T=0 和 T=1 之间进行选择。有关这两个协议的详细信息，可查看 ISO 7816-3 标准。

重试位字段指示收到 NACK 响应时的重发数，最多可以有 7 个后沿。超过此数时，将生成一个中断并锁定 USART，直至 FIFO 被清空，这可通过刷新 FIFO 来完成。没有 FIFO 可用或 FIFO 已清空时，软件可使用中断来确定下一操作。

保护时间位文件用于编程保护时间循环的额外数量，从而允许智能卡在发送响应前处理信息。额外保护时间可从 0 到 255 进行编程，其中 255 表示最小的可能字符长度。此值取决于选定的协议，对于协议 T=1 可以为 11 etu，对于协议 T=0 可以为 12 etu。

标准中定义的等待时间不能直接进行编程，但可使用定时器的 CAP1 和 CAP2 输入端来实现。 <tbd>

注：SCICTRL 寄存器在发送或接收数据时不能修改，否则数据可能丢失或破坏。

注：SCICTRL 寄存器不应与 SYNCCTRL 寄存器结合使能，因为仅支持异步智能卡。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

36.6.16 USART RS485 控制寄存器

RS485CTRL 寄存器控制 RS-485/EIA-485 模式下 USART 的配置。

表 822. USART RS485 控制寄存器（RS485CTRL – 地址 0x4008 104C (USART0), 0x400C 104C (USART2), 0x400C 204C (USART3)）位描述

位	符号	值	描述	复位值
0	NMMEN		NMM 使能。	0
		0	RS-485/EIA-485 正常多点模式 (NMM) 禁用。	
		1	RS-485/EIA-485 正常多点模式 (NMM) 使能。在该模式下，当收到的字节导致 USART 设置奇偶校验错误并生成中断时，检测到地址。	
1	RXDIS		接收器使能。	0
		0	接收器使能。	
		1	接收器禁用。	
2	AADEN		AAD 使能。	0
		0	自动地址检测 (AAD) 禁用。	
		1	自动地址检测 (AAD) 使能。	
3	-	-	保留。	-
4	DCTRL		DIR 引脚的方向控制。	0
		0	禁用自动方向控制。	
		1	使能自动方向控制。	
5	OINV		方向控制引脚极性。 此位反转 DIR 引脚上方向控制信号的极性。	0
		0	当发送器有数据要发送时，方向控制引脚会被驱动为逻辑 0。在最后一个数据位被发送后，该位就会被驱动为逻辑 1。	
		1	当发送器有数据要发送时，方向控制引脚会被驱动为逻辑 1。在最后一个数据位被发送后，该位就会被驱动为逻辑 0。	
31:6	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

复位后，RS485 模式将被禁用。RS485 功能可将 USART 配置为由单个 USART 控制的多个可寻址从机接收器之一。在 RS485 模式下， USART 通过第九个位来区分地址字符和数据字符。奇偶校验位用于执行此位，设置为 1 表示地址，设置为 0 表示数据。 RS485 模式通过设置 NMMEN 位使能。可以为 USART 从机接收器分配一个唯一的地址，并根据收到的地址手动或自动拒绝或接受数据。参见第 36.7.2 节以了解详细信息。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

36.6.17 USART RS485 地址匹配寄存器

RS485ADRMATCH 寄存器包含 RS-485/EIA-485 模式的地址匹配值。

表 823. USART RS485 地址匹配寄存器（RS485ADRMATCH － 地址 0x4008 1050 (USART0)、0x400C 1050 (USART2)、 0x400C 2050 (USART3)）位描述

位	符号	描述	复位值
7:0	ADRMATCH	包含地址匹配值。	0x00
31:8	-	保留。	-

ADRMATCH 位字段包含从机地址匹配值，用于与收到的地址值进行比较。在自动地址检测期间，此值用于接受或拒绝串行输入数据。

36.6.18 USART1 RS485 延迟值寄存器

用户可以对 8 位 RS485DLY 寄存器进行编程，在离开 TXFIFO 的最后停止位和 DIR 引脚取的消断言之间存在延迟，此延迟时间以波特率时钟为周期。可以编程 0 到 255 位倍数的任何延迟时间。

表 824. USART RS485 延迟值寄存器（RS485DLY － 地址 0x4008 1054 (USART0)、0x400C 1054 (USART2)、 0x400C 2054 (USART3)）位描述

位	符号	描述	复位值
7:0	DLY	包含方向控制延迟值。此寄存器与一个 8 位计数器共同工作。	0x00
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

36.6.19 USART 同步模式控制寄存器

SYNCCTRL 寄存器是一个控制同步模式的读 / 写寄存器。同步模式控制模块生成或接收具有串行输入 / 输出数据的同步时钟，并向发送和接收移位寄存器分发边沿检查样品。

表 825. USART同步模式控制寄存器（SYNCCTRL － 地址0x4008 1058 (USART0)、0x400C 1058 (USART2)、 0x400C 2058 (USART3)）位描述

位	符号	值	描述	复位值
0	SYNC		使能同步模式。	0
		0	已禁用。	
		1	使能。	
1	CSRC		时钟源选择。	0
		0	同步从模式 （SCLK 进）。	
		1	同步主模式 （SCLK 出）。	
2	FES		下降沿采样。	0
		0	RxD 在 SCLK 的上升沿采样。	
		1	RxD 在 SCLK 的下降沿采样。	
3	TSBYPASS		同步从机模式中的发送同步旁路。	0
		0	用于时钟边沿检测逻辑前，输入时钟同步。	
		1	用于时钟边沿检测逻辑前，输入时钟不同步。这将带来更高的输入时钟速率并影响潜在的亚稳定性。	
4	CSCEN		连续主时钟使能 （仅在 CSRC 为 1 时使用）	0
		0	SCLK 仅在字符正在 TxD 上发送时循环。	
		1	SCLK 连续运行 （RxD 上的字符接收可独立于 TxD 上的发送）。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 825. USART同步模式控制寄存器（SYNCCTRL － 地址0x4008 1058 (USART0)、0x400C 1058 (USART2)、 0x400C 2058 (USART3)）位描述 *（续）*

位	符号	值	描述	复位值
5	SSSDIS		启动 / 停止位。	0
		0	发送起动和停止位，如其他模式。	
		1	不发送起动 / 停止位。	
6	CCCLR		连续时钟清除。	0
		0	CSCEN 受软件控制。	
		1	收到每个字符后硬件清除 CSCEN。	
31:7	-		保留。未定义从保留位读取的值。	不适用

复位后，同步模式禁用。同步模式允许用户发送 （同步主模式）或接收 （同步从模式）具有串行输入和输出数据的时钟。通过设置 SYNC 位使能同步模式。 CSRC 位可用于在同步从模式 （逻辑 0）和同步主模式 （逻辑 1）之间进行切换。串行数据可在串行时钟的上升沿（默认）或下降沿上采样。STARTSTOPDISABLE 位设置时，FES 位被硬件覆盖到下降沿上的样本。

发送数据时，只需主时钟来生成时钟。在此情况下，数据只能在数据发送后接收。设置 CSCEN 位后，时钟将始终运行 （仅在同步主模式期间），使数据能连续接收。

请注意，此选项不应与 STARTSTOPDISABLE （全双工通信期间）结合使用。连续时钟在收到完整字符后可由硬件自动停止，可通过断言 CCCLR 位来完成。这在不能通过发送字符来生成时钟的半双工模式中非常有用。收到一个字符后，CSCEN 位由硬件自动清除。需要接收另一个字符时，应再次使能 CSCEN。

默认情况下，数据传输和接收的执行方式在异步模式和同步模式中相同。STARTSTOPDISABLE 位设置后，不发送起动和停止位 （也不接收它们）。这意味着所有发送或接收的位 （时钟在运行）都是数据位。

注：SYNCCTRL 寄存器的值在发送 / 接收数据时不能修改，否则数据可能丢失或破坏。

注：SYNCCTRL 寄存器不应与 SCICTRL 寄存器结合使能，因为仅支持异步智能卡。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

36.6.20 USART 发送使能寄存器

除配备全硬件流控制（上述的 auto-cts 和 auto-rts 机制）以外，TER 还可实现软件流控制。TxEn = 1 时，USART 发送器将连续发送数据，只要这些数据可用。TxEn 变为 0 时，USART 发送将立即停止。

表 826 描述了如何利用 TXEN 位来实现软件流控制。

表 826. USART 发送使能寄存器（TER – 地址 0x4008 1030 (USART0)、0x400C 1030 (USART2)、0x400C 205C (USART3)）位描述

位	符号	描述	复位值
0	TXEN	发送使能。 复位后，发送使能。 TXEN 位被取消时，将不发送数据，尽管数据可能在 TSR 或 THR 中挂起。	1
31:1	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

36.7 功能说明

36.7.1 同步模式

配置了同步接收器 / 发送器功能 (USART) 后，用串行输入和输出时钟以及可控制时钟 Pad 的输出，对串行接口进行扩展。

<tbd>

图 111. USART 串行接口协议

默认情况下，同步模式中的发送和接收操作使用的协议与异步模式中使用的协议相同。可使用同步模式控制寄存器来配置同步模式。此寄存器可控制：

- 串行时钟的方向，即同步从或主模式
- 串行时钟的采样边沿
- 发送期间输入串行时钟的两阶段或一阶段同步
- 同步主模式期间，在空闲或断点模式下，时钟可连续或禁用
- 起动和停止位的传输可以忽略。通过运行时钟来识别有效数据。始终在串行时钟的下降沿采样

在串行时钟的采样沿，数据在接收移位寄存器中移位。

36.7.1.1 同步从机模式

此模式通过将控制寄存器的 CSRC 位设为 0 使能。同步从机模式期间，需要一个计时串行输入和输出数据的外部时钟。请注意，串行时钟内部被视为数据信号。执行串行时钟边沿检测，以同步串行时钟和 USART 时钟域，因此没有寄存器用串行时钟计时。

接收

默认情况下，收到的字符类似于异步模式中的字符。串行数据流在没有数据可用时保持“高速”。此期间无需运行外部串行时钟。将收到的第一个位是起始位，此时必须运行外部串行时钟。起始位的起始端可与串行时钟的上升沿 （在下降沿上采样）或下降沿 （在上升沿上

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

采样）对齐，参见表 825 中的 FES 位。在上升沿上采样时，起始位的起始端不需要与时钟沿对齐 （时钟之前可能尚未运行）。此时，因起始位 （逻辑 1 至 0）而产生的串行输入数据的边沿用于确定字符的起始端 （参见 <tbd>）。

<tbd>

图 112. 以同步从模式发送数据

同步模式控制寄存器的 NOSTARTSTOPBITS 位允许用户禁用起始和停止位的发送 / 接收，从而提高 USART 的效率。由于字符不再由起始和停止位识别，因此串行时钟用于确定数据位。串行时钟运行时，采样的所有数据都被视为有效数据。

为了能够识别字符的起始，字符的开始必须与串行时钟的上升沿对齐。因此，同步模式控制寄存器的 FES 位在硬件中强制为 1。

采样最后一位后，字符直接存储在接收 FIFO 中。

传输

同步从模式期间，只能在外部串行时钟运行时发送数据。因此，没有发送起始和停止位时，仅在从主机接收数据时才能进行传输。发送起始和停止位时，外部时钟仅可在头半个收到的起始位后检测 （在外部串行时钟的上升沿采样）。使用收到的起始位 （逻辑 1 到 0）创建的边沿，可确保要由从机发送的字符起始位在外部从机时钟的上升沿前稳定。这样就能确保主机接收与其发送的位一样多的位。

当输入串行时钟的第一个采样沿在串行输入数据上采样到 1 时 （起始 - 停止位已发送，因此主机尚未启动事务），假定主机运行连续时钟 （而不是仅在发送数据字符时运行时钟）。USART 将不会等待来自主机的起始位，而是在数据可用时立即开始发送数据。请注意，在此情况下，主机发送的位数和从机发送的位数 （由主机接收）可能不对齐，假定有更高一级的协议确保在主机停止时钟时收到完整字符。

对于同步从模式期间传输数据而言，时间是最关键的。首先必须使用边沿检测逻辑检测外部串行输入时钟。然后，需要将数据移出并在外部串行时钟的采样沿前稳定下来。

注：在此模式下， u_clk 周期允许为串行时钟周期的 4 倍。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

36.7.1.2 同步主机模式

通过将 CSRC 寄存器位设置为 1 使能同步主机模式。在此模式下，外部时钟由波特率生成逻辑在内部生成，并用于为输入和输出串行数据提供时钟。波特率生成功能已在第 36.6.12.1 节中说明。同步模式期间不支持自动波特率。1x 波特率时钟用于移出串行输出数据并采样串行输入数据。

同步主机模式的行为类似于从机模式，除了串行输入数据不在接口寄存，而在串行时钟采样沿的 USART 时钟域计时以外。

同步主模式期间，发送起始和停止位时，用户可使用同步模式控制寄存器的 cscen 位连续使能外部时钟。这就能使所连接的从机发送数据，即使主机本身没有发送数据。

36.7.2 RS-485/EIA-485 模式的操作

RS-485/EIA-485 功能可将 USART 配置为可寻址从机。可寻址从机是由单个主机控制的多个从机之一。

USART 主机发送器将设置奇偶位（第 9 位）为 1 以标识一个地址字符。对于数据字符，将奇偶校验位设置为 0。

可为每个 USART 从机接收器分配一个唯一的地址。可以对从机进行编程，使其手动或自动拒绝不是其自身地址后的数据。

RS-485/EIA-485 正常多点模式 (NMM)

将 RS485CTRL 位设置为 0 可使能此模式。在该模式下，当收到的字节导致 USART 设置奇偶校验错误并生成中断时，检测到地址。

如果接收器被禁用（RS485CTRL 位 1=1），任何接收到的数据字节都将被忽略，且不会存入 RXFIFO。检测到地址字节 （奇偶校验位 =1）时，会将其放入 RXFIFO，并生成一个“Rx 数据就绪中断”。处理器然后即可读取地址字节，并决定是否使能接收器来接收随后的数据。

当接收器被使能 （RS485CTRL 位 1=0），所有接收到的字节将会被接受并存入 RXFIFO，不论是数据还是地址。收到地址字符时，将生成一个奇偶校验错误中断，然后处理器会决定是否禁用接收器。

RS-485/EIA-485 自动地址检测 (AAD) 模式

RS485CTRL 寄存器位 0（9 位模式使能）和 2（AAD 模式使能）都设置时， USART 处于自动地址检测模式。

在该模式下，接收器会将任何收到的地址字节（奇偶校验 = 1）与编程到 RS485ADRMATCH 寄存器中的 8 位值进行比较。

如果接收器被禁用 （RS485CTRL 位 1=1），任何接收到的与 RS485ADRMATCH 值不匹配的数据字节或者地址字节都将被丢弃。

检测到匹配的地址字符时，会将其连同奇偶校验位一起推入 RXFIFO，然后接收器将自动使能 （RS485CTRL 位 1 将由硬件清除）。接收器也将生成一个 “Rx 数据就绪中断”。

接收器使能时 （RS485CTRL 位 1 = 0），将接受收到的所有字节并存储在 RXFIFO 中，直至收到与 RS485ADRMATCH 值不匹配的地址字节。出现这种情况时，接收器将在硬件中自动禁用（RS485CTRL 位 1 将被设置），收到的不匹配地址字符将不会存储在 RXFIFO 中。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

RS-485/EIA-485 自动方向控制

RS485/EIA-485 模式包含允许发送器以方向控制输出信号的方式自动控制 DIR 引脚状态的选项。

设置 RS485CTRL 位 4=1 将使能该功能。

使能“自动方向控制”后，CPU 将数据写入 TXFIFO 时，选定的引脚将被断言（驱动低电平）。数据的最后一位发送后，引脚将被取消（驱动高电平）。请参见 RS485CTRL 寄存器中的位 4 和 5。

RS485CTRL 位 4 优先于所有其他控制方向控制引脚的机制。

RS485/EIA-485 驱动器延迟时间

驱动器延迟时间是最后一个停止位离开 TXFIFO 到 DIR 引脚取消之间的延迟。此延迟时间可在 8 位 RS485DLY 寄存器中编程，延迟时间以波特率时钟为周期。可以使用 0 到 255 位倍数的任何延迟时间。

RS485/EIA-485 输出反转

DIR 引脚上方向控制信号的极性可通过对 RS485CTRL 寄存器中的位 5 编程来反转。此位设置后，发送器有要发送的数据时，方向控制引脚将驱动到逻辑 1。数据的最后一位发送后，方向控制引脚将被驱动到逻辑 0。

36.7.3 智能卡模式

图 113 所示为一个典型的异步智能卡应用。

图 113. 典型智能卡应用

SCICTRL 寄存器中的 SCIEN 位（表 821）按如上所述设置后， USART 在开漏 TXD 引脚上提供双向串行数据。SCIEN 为 1 时不使用 RXD 引脚。USART UCLK 引脚将以数据比特率同步输出数据。软件必须使用定时器来实现字符和时钟等待时间（本器件不提供通过启动信号支持的硬件）。GPIO 引脚可用于控制智能卡复位和电源引脚。任何提供给卡的电源都必须采用外部开关，因为卡的电源要求通常超过了本器件上可能的电源电流。由于具体应用可能采用任何可用的 ISO 7816 的 A、B 或 C 类电源要求，因此使用与本器件不同的电源导轨进行通信或供应电源时，要注意逻辑电平公差和要求。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

36.7.3.1 智能卡设置步骤

T = 0 协议传输包含一个 8 位数据、一个偶数奇偶校验位和两个保护位，其中保护位允许特定传输的接收器通过 NACK 响应标记奇偶校验错误 （参见图 114）。也可根据卡的要求添加额外保护位。如果没有发送 NACK（假定接口在 SCICTRL 中接受它们），则下一个字节会在最后一个保护位后立即发送。如果已发送 NACK，则发送器将重试发送字节，直至被成功收到或达到 SCICTRL 重试限制。

图 114. 智能卡 T = 0 波形

设置智能卡时必须考虑以下事项：

1. 必要时，使 USART 脱离复位并对外设使能时钟。
2. 为双向传输设置可用的 USART TXD 引脚。
3. 在引脚配置寄存器中将 UCLK 引脚设为时钟源。大多数同步卡的默认时钟要求为 372 倍比特率。
4. 为波特率配置 DLL 和 DLM。可能没必要以特定标准波特率为目标，而是维护一部分前述时钟速率。例如，如果时钟速率设置为 4 MHz，波特率将为 10753。时钟速率为 3.5712 MHz 时，将需要波特率 9600。ISO 7816 PPS 交换可能要求在稍后更改波特率。
5. 配置 LCR 的字符大小和奇偶校验 （通常为 8 位和偶数奇偶校验）。
6. 使用所需的 NACK 响应、外部保护位和协议类型配置 SCICTRL。
7. 将 GPIO 输出信号置于不活动状态，在此状态下卡电源关闭、RST 低电平且 CLK 低电平并保持不变。

此后，软件应监控卡插入、处理激活、等待回答，以按 ISO7816-3 所述进行复位。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

37.1 本章导读

所有 LPC43xx 器件上都包含 UART1 控制器。

37.2 基本配置

UART1 配置如下：

- 参见表 827 以了解时钟与电源控制。
- UART1 通过 UART1_RST 复位 （复位 #45）。
- UART1 中断连接到 NVIC 中的插槽 # 25。
- 要将 UART1 接收和发送线连接至 GPDMA，请使用 CREG 模块中的 DMAMUX 寄存器（参见表 41）并使能 DMA 通道配置寄存器 （表 19.6.20）中的 GPDMA 通道。

表 827. UART1 时钟与电源控制

	基本时钟	分支时钟	工作频率
UART1 时钟至寄存器接口	BASE_M4_CLK	CLK_M4_UART0	高达 204 MHz
UART1 外设时钟 (PCLK)	BASE_UART1_CLK	CLK_APB0_UART1	高达 204 MHz

37.3 功能要点

- 全调制解调器控制信息交换可用。
- 数据大小 5、 6、 7、 8 位。
- 奇偶生成及校验：奇校验、偶校验、标识校验、空格校验或无校验。
- 一或两个停止位。
- 16 字节收发 FIFO。
- 内置波特率生成器，包括一个小数分频器以实现多功能性。
- 支持 DMA 发送和接收。
- 自动波特率功能。
- 中止产生及检测。
- 多处理器寻址模式。
- 支持 RS-485。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

37.4 引脚说明

表 828. UART1 引脚描述

函数名称	方向	描述
RXD1	输入	**串行输入。**串行接收数据。
TXD1	输出	**串行输出。**串行发送数据。
CTS1	输入	**准许发送。**有效低信号表示外部调制解调器是否准备好接受由 UART1 通过 TXD1 发送的数据。调制解调器接口正常运行时 (U1MCR[4] = 0)，该信号的补值存储在 U1MSR[4] 中。状态改变信息存储在 U1MSR[0] 中作为第 4 优先级中断的中断源 （如果使能的话 (U1IER[3] = 1)）。 准许发送。CTS1 是一个异步有效低调制解调器状态信号。通过读取调制解调器状态寄存器第 4 位值 (CTS) 可以检查其状况。调制解调器状态寄存器 (MSR) 的位 0(DCTS) 表示 CTS1 和上次从 MSR 读取值相比状态发生了改变。如果当 CTS1 电平改变时调制解调器状态中断已使能，且 auto-cts 模式未使能，将会产生一个中断。 CTS1 在 auto-cts 模式中也用来控制发送器。
DCD1	输入	**数据载波检测。**有效低信号表示外部调制解调器是否和 UART1 建立了一个通信链路，是否可以进行数据交换。调制解调器接口正常运行时 (U1MCR [4] = 0)，该信号的补值存储在 U1MSR [7] 中。状态改变信息存储在 U1MSR3 中作为第 4 优先级中断的中断源 （如果使能的话 (U1IER[3] = 1)）。
DSR1	输入	**数据设置就绪。**有效低信号表示外部调制解调器是否已准备好与 UART1 建立通信链路。调制解调器接口正常运行时 (U1MCR[4] = 0)，该信号的补值存储在 U1MSR[5] 中。状态改变信息存储在 U1MSR[1] 中作为第 4 优先级中断的中断源 （如果使能的话 (U1IER[3] = 1)）。
DTR1	输出	数据终结就绪。有效低信号表示 UART1 是否已准备好与外部调制解调器建立连接。该信号的补值存储在 U1MCR [0] 中。 DTR 引脚也可用作 RS-485/EIA-485 输出使能信号。
RI1	输入	**振铃指示器。**有效低信号表示调制解调器检测到了一个电话振铃信号。调制解调器接口正常运行时 (U1MCR[4] = 0)，该信号的补值存储在 U1MSR[6] 中。状态改变信息存储在 U1MSR[2] 中作为第 4 优先级中断的中断源 （如果使能的话 (U1IER[3] = 1)）。
RTS1	输出	**请求发送。**有效低信号表示 UART1 想要给外部调制解调器发送数据。该信号的补值存储在 U1MCR [1] 中。 在 auto-rts 模式中， RTS1 用于控制发送器 FIFO 阈值逻辑。 请求发送。RTS1 是一个有效低信号，通知调制解调器或者数据装置 UART 已准备好接收数据。设置 RTS 调制解调器控制寄存器位可以把 RTS1 设为有效 （低平）值，系统复位后、环回模式操作中或者清除 MCR 位 1 (RTS) 可将其设为无效 （高平）值。在 auto-rts 模式中， RTS1 受发送器 FIFO 阈值逻辑控制。 RTS 引脚也可用作 RS-485/EIA-485 输出使能信号。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

37.5 寄存器描述

UART1所包含的寄存器，其结构如表829所示。除数锁存器访问位(DLAB)包含在U1LCR[7]中，可实现对除数锁存器的访问。

复位值仅反映存储在所用位中的数据，不包含保留位中的内容。

表 829. 寄存器简介：UART1 （基址 0x40082000）

名称	访问类型	地址偏移	描述	复位值	参考
RBR	RO	0x000	接收缓冲寄存器。包含下一个待读的已接收字符。 (DLAB=0)	不适用	表 830
THR	WO	0x000	发送保持寄存器。下一个待发送字符写入在此。 (DLAB=0)	不适用	表 831
DLL	R/W	0x000	除数锁存 LSB。波特率除数值的最低有效字节。整除数用于从小数分频器产生波特率。 (DLAB=1)	0x01	表 832
DLM	R/W	0x004	除数锁存 MSB。波特率除数值的最高有效字节。整除数用于从小数分频器产生波特率。 (DLAB=1)	0x00	表 833
IER	R/W	0x004	中断使能寄存器。包含可能发生的 7 种 UART1 中断对应的 7 个不同中断使能位。 (DLAB=0)	0x00	表 834
IIR	RO	0x008	中断 ID 寄存器，识别要挂起的中断。	0x01	表 835
FCR	WO	0x008	FIFO 控制寄存器，控制 UART1 的 FIFO 使用和模式。	0x00	表 837
LCR	R/W	0x00C	线路控制寄存器。包含对帧格式化和断点产生的控制。	0x00	表 838
MCR	R/W	0x010	调制解调器控制寄存器。包括流量控制信息交换控制和环回模式控制。	0x00	表 839
LSR	RO	0x014	线路状态寄存器，包含发送和接收状态 （包括线路错误）的标志。	0x60	表 841
MSR	RO	0x018	调制解调器状态寄存器。包含信号握手信号状态标志。	0x00	表 842
SCR	R/W	0x01C	暂存寄存器，为软件设置的 8 位临时存储空间。	0x00	表 843
ACR	R/W	0x020	自动波特率控制寄存器，包含自动波特率功能的控制。	0x00	表 844
FDR	R/W	0x028	小数分频寄存器，为波特率分频器生成时钟输入。	0x10	表 845
TER	R/W	0x030	发送使能寄存器。关闭 UART 发送器，以使用软件流控制。	0x80	表 847
RS485CTRL	R/W	0x04C	RS-485/EIA-485 控制，包含对配置 RS-485/EIA-485 模式各个方面的控制。	0x00	表 848
RS485ADRMATCH	R/W	0x050	RS-485/EIA-485 地址匹配，包含 RS-485/EIA-485 模式的地址匹配值。	0x00	表 849
RS485DLY	R/W	0x054	RS-485/EIA-485 方向控制延迟。	0x00	表 850
FIFOLVL	RO	0x058	FIFO 电平寄存器。提供发送和接收 FIFO 的当前填充电平。	0x00	表 851

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

37.5.1 UART1 接收器缓冲寄存器 （当 DLAB = 0 时）

U1RBR 是 UART1 RX FIFO 的最高字节。RX FIFO 最高字节包含了接收到的最旧的字符，可以通过总线接口读取。LSB（位 0）代表接收到的最旧的数据位。如果接收到的字符少于 8 位，则未使用的 MSB 用 0 填充。

位于 U1LCR 中的除数锁存器访问位 (DLAB) 必须为 0 以实现对 U1RBR 的访问。 U1RBR 始终为只读模式。

由于 PE、 FE 和 BI 位与 RBR FIFO 中的最高字节相对应 （即下一次读取操作即将从 RBR 中读出的字节），获取收到字节及其状态位的有效对的正确方式是首先读取 U1LSR 寄存器中的内容，然后从 U1RBR 中读取一个字节。

表 830. UART1 接收缓冲寄存器 （当 DLAB = 0 (RBR - 地址 0x4008 2000）位描述

位	符号	描述	复位值
7:0	RBR	接收缓冲。 包含 UART1 RX FIFO 接收到的最旧字节。	未定义
31:8	-	保留，从保留位读取的值未定义。	不适用

37.5.2 UART1 发送器保持寄存器 （当 DLAB = 0 时）

U1THR 是 UART1 TX FIFO 的最高只写字节。最高字节是 TX FIFO 中最新的字符，可通过总线接口写入。 LSB 代表第一个要发送的位。

位于 U1LCR 中的除数锁存器访问位 (DLAB) 必须为 0 以实现对 U1THR 的访问。U1THR 为只写模式。

表 831. UART1 发送保持寄存器 （当 DLAB = 0 时）（THR - 地址 0x4008 2000）位描述

位	符号	描述	复位值
7:0	THR	发送保持寄存器。 写入 UART1 发送保持寄存器的数据将存储在 UART1 发送 FIFO 中。字节将在到达 FIFO 底部且发送器可用时发出。	不适用
31:8	-	保留，用户软件不应向保留位写入 1。	不适用

37.5.3 UART1 除数锁存器 LSB 和 MSB 寄存器 （当 DLAB = 1 时）

UART1 除数锁存器是 UART1 波特率生成器的一部分，保存使用过的值，与小数分频器一起分频 APB 时钟 (PCLK) 以产生波特率时钟。波特率时钟必须为要求波特率的 16 倍数。U1DLL 和 U1DLM 寄存器一起组成一个 16 位除数，其中 U1DLL 包含除数中的低 8 位内容，U1DLM 包含除数中的高 8 位内容。值 0x0000 被当作 0x0001 处理，因为 0 不能作为除数。U1LCR 中的除数锁存器访问位 (DLAB) 必须为 1 以实现对 UART1 除数锁存器的访问。选择 U1DLL、 U1DLM 正确值的方法详情，请参见本章稍后的第 37.5.16 节。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 832. UART1 除数锁存器 LSB 寄存器 （当 DLAB = 1 时）（DLL - 地址 0x4008 2000）位描述

位	符号	描述	复位值
7:0	DLLSB	除数锁存 LSB。 UART1 除数锁存器 LSB 寄存器与 U1DLM 寄存器共同决定 UART1 的波特率。	0x01
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

表 833. UART1 除数锁存器 MSB 寄存器 （当 DLAB = 1 时）（DLM - 地址 0x4008 2004）位描述

位	符号	描述	复位值
7:0	DLMSB	除数锁存 MSB。 UART1 除数锁存器 MSB 寄存器与 U1DLL 寄存器共同决定 UART1 的波特率。	0x00
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

37.5.4 UART1 中断使能寄存器 （当 DLAB = 0 时）

U1IER 用于使能 UART1 的四个中断源。

表 834. UART1 中断使能寄存器 （当 DLAB = 0 时）（IER - 地址 0x4008 2004）位描述

位	符号	值	描述	复位值
0	RBRIE		RBR 中断使能。使能 UART1 的收到数据可用中断。也控制字符接收超时中断。	0
		0	禁用 RDA 中断。	
		1	使能 RDA 中断。	
1	THREIE		THRE 中断使能。使能 UART1 的 THRE 中断。该中断的状态可以从 U1LSR [5] 读取。	0
		0	禁用 THRE 中断。	
		1	使能 THRE 中断。	
2	RXIE		RX 线路中断使能。使能 UART1 RX 线路状态中断。该中断的状态可以从 U1LSR [4:1] 读取。	0
		0	禁用 RX 线路状态中断。	
		1	使能 RX 线路状态中断。	
3	MSIE		调制解调器状态中断使能。使能调制解调器中断。该中断的状态可以从 U1MSR [3:0] 读取。	0
		0	禁用调制解调器中断。	
		1	使能调制解调器中断。	
6:4	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
7	CTSIE		CTS 中断使能。在 auto-cts 模式使能的情况下，该位使能 / 禁用在 CTS1 信号跳变时产生调制解调器状态中断。在非 auto-cts 模式禁用的模式下，如果设置调制解调器状态中断使能 (U1IER [3])， CTS1 信号跳变将会产生一个中断。 正常运行时， CTS1 信号跳变会产生一个调制解调器状态中断，除非清除 U1IER 寄存器中的 U1IER [3] 位以禁用中断。在 auto-cts 模式，仅当同时设置了 U1IER[3]、 U1IER[7] 位， CTS1 位跳变将会启动中断。	0
		0	禁用 CTS 中断。	
		1	使能 CTS 中断。	
8	ABEOIE		使能自动波特率结束中断。	0
		0	禁用自动波特率结束中断。	
		1	使能自动波特率结束中断。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 834. UART1 中断使能寄存器 （当 DLAB = 0 时）（IER - 地址 0x4008 2004）位描述 *（续）*

位	符号	值	描述	复位值
9	ABTOIE		使能自动波特率超时中断。	0
		0	禁用自动波特率超时中断。	
		1	使能自动波特率超时中断。	
31:10	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

37.5.5 UART1 中断识别寄存器

U1IIR 提供一个状态代码来表示待处理中断的优先级和中断源。中断在 U1IIR 访问期间将被冻结。如 U1IIR 访问期间发生了某一中断，其会被记录，用于下一次 U1IIR 访问。

表 835. UART1 中断识别寄存器 （IIR - 地址 0x4008 2008）位描述

位	符号	值	描述	复位值
0	INTSTATUS		中断状态。注意 U1IIR [0] 处于有效低位。等待处理的中断可以通过对 U1IIR [3:1] 求值决定。	1
		0	至少一个中断挂起。	
		1	没有中断挂起。	
3:1	INTID		中断识别。U1IER [3:1] 标示对应 UART1 RX 或者 TX FIFO 的一个中断。所有其他未在下面列出的 U1IER [3:1] 组合均为保留位 (100、 101、 111)。	0
		0x3	1 - 接收线路状态 (RLS)。	
		0x2	2a - 接收数据可用 (RDA)。	
		0x6	2b - 字符超时指示 (CTI)。	
		0x1	3 - THRE 中断。	
		0x0	4 - 调制解调器中断。	
5:4	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
7:6	FIFOENABLE		U1FCR [0] 的拷贝。	0
8	ABEOINT		自动波特率结束中断。自动波特率成功完成且中断使能时为真。	0
9	ABTOINT		自动波特率超时中断。自动波特率超时且中断使能时为真。	0
31:10	-		保留，从保留位读取的值未定义。	不适用

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

U1IIR [9:8] 由自动波特率功能设置，在超时或自动波特率结束条件时发出信号。通过设置自动波特率控制寄存器中的相应 “清除” 位，清除自动波特率中断条件。

如果 IntStatus 位为 1，则表示没有中断被挂起，且 IntId 位将为 0。如果 IntStatus 为 0，表示有一个非自动波特率中断被挂起，此时 IntId 位会却确定中断的类型和处理方式，如表 836 中所述。根据 U1IIR [3:0] 的状态，中断处理程序可以确定中断原因以及如何清除激活的中断。退出中断服务程序前必须读取 U1IIR 以清除中断。

UART1 RLS 中断 (U1IIR [3:1] = 011) 是最高优先级中断，当 UART1RX 输入发生四个错误条件中的任何一个时，将设置该中断：溢出错误 (OE)、奇偶校验错误 (PE)、帧处理错误 (FE) 和中止中断 (BI)。设置中断的 UART1 RX 错误条件可以通过 U1LSR [4:1] 观察。中断在发生 U1LSR 读取操作时清除。

UART1 RDA 中断 (U1IIR[3:1] = 010) 与 CTI 中断 (U1IIR[3:1] = 110) 均为第二优先级。当 UART1 的 RX FIFO 达到 U1FCR7:6 定义的启动水平时 RDA 将被激活，且在 UART1 的 RX FIFO 深度降至启动水平下时被复位。RDA 中断激活时，CPU 可读取启动级别定义的数据块。

CTI 中断 (U1IIR[3:1] = 110) 是第二级中断，当 UART1 的 RX FIFO 包含至少一个字符且在 3.5 至 4.5 字符时间无 UART1 RX FIFO 活动发生时被设置。任何 UART1 Rx FIFO 活动（读取或写入 UART1 RSR）将会清除该中断。该中断设计为当接收一条非启动级别大小倍数的信息以后刷新 UART1 RBR。例如，如果外设要发送一条 105 个字符的消息，并且启动级别为 10 个字符，则 CPU 将接收 10 次 RDA 中断以传输 100 个字符，还将接收 1 到 5 次 CTI 中断 （取决于服务例程）以传输剩余的 5 个字符。

表 836. UART1 中断处理

U1IIR[3:0] 值 [1]	优先级	中断类型	中断源	中断复位
0001	-	无	无	-
0110	最高	RX线路状态 / 错误	OE[2]、 PE[2]、 FE[2] 或 BI[2]	U1LSR 读操作 [2]
0100	第二	RX数据可用	Rx 数据可用或者已达到 FIFO 启动级别 (U1FCR0=1)	U1RBR 读操作 [3] 或 UART1 FIFO 低于启动电平
1100	第二	字符超时指示	RX FIFO 中至少有一个字符且一段时间内无字符输入或者无字符删除，取决于 FIFO 中有多少字符和启动级别的设置（3.5 至 4.5 字符时间）。 精确时间为： [（字长度）× 7 - 2] × 8 + [（启动电平 - 字符数）× 8 + 1] RCLKs	U1RBR 读操作 [3]
0010	第三	THRE	THRE[2]	U1IIR 读操作 [4]（如果是中断源）或 THR 写操作
0000	第四	调制解调器状态	CTS 或 DSR 或 RI 或 DCD	MSR 读取

[1] 保留的值为 "0000", “0011”, “0101”, “0111”, “1000”, “1001”, “1010”, “1011”, “1101”, “1110”, “1111”。

[2] 有关详情，请参阅第 37.5.10 节 “UART1 线路状态寄存器”。

[3] 有关详情，请参阅第 37.5.1 节 “UART1 接收器缓冲寄存器 （当 DLAB = 0 时）”。

[4] 有关详情，请参阅 第 37.5.5 节 “UART1 中断识别寄存器” 和 第 37.5.2 节 “UART1 发送器保持寄存器（当 DLAB = 0 时）”。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

UART1 THRE 中断 (U1IIR[3:1] = 001) 是一个三级中断，当 UART1 THR FIFO 为空且某些初始化条件满足的时候将激活该中断。上述初始化条件设计为启用 UART1 THR FIFO 填充数据以消除系统启动时发生的许多 THRE 中断。当 THRE = 1 时，且在上次的 THRE = 1 事件后，U1THR 中没有出现至少两个字符时，这些初始化条件就会实现一个字符减去停止位的延时。该延迟用来给 CPU 时间将数据写入 U1THR 而不以解码和服务的原因发生 THRE 中断。如果 UART1 THR FIFO 一次性保存了两个及以上字符且当前 U1THR 为空时，将立即设置 THRE 中断。当 U1THR 写入或者从 U1IIR 读取，且 THRE 为最高中断 (U1IIR[3:1] = 001) 时，将复位 THRE 中断。

该中断为最低优先级中断，当调制解调器输入引脚 DCD、 DSR 或 CTS 状态发生变化时被激活。除此之外，调制解调器输入 RI 从低至高的跳变将产生一个调制解调器中断。调制解调器中断源可以通过检查 U1MSR [3:0] 确定。 U1MSR 读取将清除调制解调器中断。

37.5.6 UART1 FIFO 控制寄存器

只写 U1FCR 控制 UART1 RX 和 TX FIFO 的运行。

表 837. UART1 FIFO 控制寄存器 （FCR - 地址 0x4008 2008）位描述

位	符号	值	描述	复位值
0	FIFOEN		FIFO 使能。	0
		0	不得用于应用程序。	
		1	有效高电平使能对 UART1 Rx、TX FIFO 和 U1FCR [7:1] 的访问。必须设置该位以实现正常的 UART1 运行。该位上的任何跳变将自动清除 UART1 的 FIFO。	
1	RXFIFORES		RX FIFO 复位。	0
		0	对 UART1 的 FIFO 无影响。	
		1	向 U1FCR[1] 写入逻辑 1，将清除 UART1 Rx FIFO 中所有字节、复位指针逻辑。此位是自清除的。	
2	TXFIFORES		TX FIFO 复位。	0
		0	对 UART1 的 FIFO 无影响。	
		1	向 U1FCR[2] 写入逻辑 1，将清除 UART1 TX FIFO 中所有字节、复位指针逻辑。此位是自清除的。	
3	DMAMODE		DMA 模式选择。设置 FIFO 使能位 （此寄存器的位 0）时，此位选择 DMA 模式。参见第 37.5.6.1 节。	0
5:4	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
7:6	RXTRIGLVL		RX启动级别。这两位决定必须写入多少接收器UART1 FIFO字符才会激活中断。	0
		0x0	启动级别 0 （1 个字符或 0x01）。	
		0x1	启动级别 1 （4 个字符或 0x04）。	
		0x2	启动级别 2 （8 个字符或 0x08）。	
		0x3	启动级别 3 （14 个字符或 0x0E）。	
31:8	-		保留，用户软件不应向保留位写入 1。	不适用

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

37.5.6.1 DMA 操作

用户也可选择使用 DMA 来操作 UART 发送和 / 或接收。DMA 模式由 FCR 寄存器中的 DMA 模式选择位确定。请注意，对于 DMA 操作，与 UART 的任何操作一样，都必须通过 FCR 寄存器中的 FIFO 使能位来使能 FIFO。

UART 接收器 DMA

在 DMA 模式下，当接收器 FIFO 级别变成等同或大于启动级别，或者发生字符超时时，接收器 DMA 请求变为有效。请参见以上 RX 启动级别的说明。接收器 DMA 请求将被 DMA 控制器清除。

UART 发送器 DMA

在 DMA 模式下，当发送器 FIFO 跳变为非满时，发送器 DMA 请求变为有效。发送器 DMA 请求将被 DMA 控制器清除。

37.5.7 UART1 线路控制寄存器

U1LCR 将决定即将发送或者接收的数据字符的格式。

表 838. UART1 线路控制寄存器 （LCR - 地址 0x4008 200C）位描述

位	符号	值	描述	复位值
1:0	WLS		字长选择。	0
		0x0	5 位字符长度。	
		0x1	6 位字符长度。	
		0x2	7 位字符长度。	
		0x3	8 位字符长度。	
2	SBS		停止位选择。	0
		0	1 停止位。	
		1	2 个停止位 （如果 U1LCR [1:0] =00，为 1.5 个）。	
3	PE		奇偶校验使能。	0
		0	禁用奇偶校验生成和检查。	
		1	使能奇偶校验生成和检查。	
5:4	PS		奇偶校验选择。	0
		0x0	奇数校验。已发送字符中 1 的数量，附加的奇偶校验位将为奇数。	
		0x1	偶数校验。已发送字符中 1 的数量，附加的奇偶校验位将为偶数。	
		0x2	强制 "1" 粘附奇偶校验。	
		0x3	强制 "0" 粘附奇偶校验。	
6	BC		断点控制。	0
		0	禁用断点传输。	
		1	使能断点传输。U1LCR[6] 为高电平有效时，输出引脚 UART1 TXD 强制设置为逻辑 0。	
7	DLAB		除数锁存器访问位 (DLAB)	0
		0	禁用对除数锁存的访问。	
		1	使能对除数锁存的访问。	
31:8	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

37.5.8 UART1 调制解调器控制寄存器

U1MCR 使能调制解调器环回模式并且控制调制解调器的输出信号。

表 839. UART1 调制解调器控制寄存器 （MCR - 地址 0x4008 2010）位描述

位	符号	值	描述	复位值
0	DTRCTRL	-	DTR 控制。 调制解调器输出引脚 DTR 的源。当调制解调器回送模式激活时，该位读为 0。	0
1	RTSCTRL	-	RTS 控制。 调制解调器输出引脚 RTS 的源。当调制解调器回送模式激活时，该位读为 0。	0
3:2	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	0
4	LMS		环回模式选择。 调制解调器环回模式提供了一个环回诊断检测机制。发送器中的串行数据在内部连接至接收器的串行输入。输入引脚 RXD1 对环回无影响，而输出引脚 TXD1 保持为标识状态。调制解调器的 4 种输入（CTS、DSR、RI 和 DCD）在外部断开。在外部，调制解调器输出 （RTS、 DTR）被设置为无效状态。在内部，调制解调器的 4 个输出与调制解调器的 4 个输入连接。上述连接使得 U1MSR 中的高 4 位将取决于 U1MCR 的低 4 位，而不是正常模式下的 4 个调制解调器的输入。这种情况下，环回模式中对 U1MCR 低四位写入值将产生调制解调器状态中断。	0
		0	禁用调制解调器环回模式。	
		1	使能调制解调器环回模式。	
5	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	0
6	RTSEN		RTS 使能。	0
		0	禁用 auto-rts 流量控制。	
		1	使能 auto-rts 流量控制。	
7	CTSEN		CTS 使能。	0
		0	禁用 auto-cts 流量控制。	
		1	使能 auto-cts 流量控制。	
31:8	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

37.5.9 自动流控制

如果使能 auto-RTS 模式， UART1 接收器 FIFO 硬件会控制 UART1 的 RTS1 输出。如果 auto-CTS 模式处于使能状态， UART1 的 U1TSR 硬件仅在 CTS1 输入信号被断言时开始发送。

37.5.9.1 自动 RTS

通过设置 RTSen 位可以使能自动 RTS 功能。 Auto-RTS 数据流量控制起于 U1RBR 模块，与编程的接收器 FIFO 启动级别相连。如果自动 RTS 使能，则按照以下方式对数据流进行控制：

当接收器 FIFO 级别达到编程的启动电平， RTS1 将解除有效状态（变为一个高值）。启动级别达到以后，发送 UART 可能会发出一个额外字节 （如果它有另外一个字节要发送），因为它可能在开始发送额外字节后才识别出 RTS1 解除了有效状态。一旦接收器 FIFO 达到之前的启动级别，RTS1 将自动重新变为有效状态（变为一个低值）。RTS1 重新变为有效状态将知会发送 UART 继续发送数据。

如果禁用 Auto-RTS 模式， RTSen 位控制 UART1 的 RTS1 输出。如果使能 Auto-RTS 模式，由硬件控制 RTS1 输出， RTS1 的实际值将被拷贝到 UART1 的 RTS 控制位。如果自动 RTS 使能，则只有软件可对 RTS 控制位的值进行只读操作。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

示例： 假设 UART1 以 550 模式运行，U1FCR 启动级别为 0x2，则如果使能 Auto-RTS，只要接收 FIFO 包含 8 个字节，UART1 将会立即取消 RTS1 输出 (第 815 的表 837)。一旦接收 FIFO 达到之前的 4 字节的启动级别，RTS1 输出将立即重新变为有效状态 4 字节。

图 115. 自动 RTS 功能时序

37.5.9.2 自动 CTS

通过设置 CTSen 位可以使能自动 CTS 功能。如果使能 Auto-CTS， U1TSR 模块中的发送器电路将在发出下一数据字节前检查 CTS1 输入。当 CTS1 为有效 （低电平），发送器发出下一字节。为了阻止发送器发送后面的字节，CTS1 必须在当前发送的最后一个停止位中间释放。在 Auto-CTS 模式下， CTS1 信号的变化不会启动调制解调器状态中断，除非对 CTS 中断使能位进行设置，不过将会设置 U1MSR 中的 Delta CTS 位。表 840 列出了生成调制解调器状态中断的条件。

表 840. 调制解调器状态中断生成

使能调制解调器状态中断 (U1ER[3])	CTSen (U1MCR[7])	CTS 中断使能 (U1IER[7])	Delta CTS (U1MSR [0])	Delta DCD 或后沿 RI 或 Delta DSR （U1MSR [3] 或 U1MSR [2] 或 U1MSR[1]）	调制解调器状态中断
0	x	x	x	x	无
1	0	x	0	0	无
1	0	x	1	x	有
1	0	x	x	1	有
1	1	0	x	0	无
1	1	0	x	1	有
1	1	1	0	0	无
1	1	1	1	x	有
1	1	1	x	1	有

自动 CTS 功能可减少主机系统的中断。当使能流量控制时， CTS1 状态变化不会启动主机中断，因为设备会自动控制其自身的发送器。若不使用自动 CTS，则发送器会将发送 FIFO 中存在的所有数据都发送出去，从而导致接收器发生溢出错误。图 116 展示了自动 CTS 功能时序。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 116. 自动 CTS 功能时序

当开始发送初始字符时，CTS1 信号变为有效。一旦等待处理中的发送完成了，发送过程将会立即停止。只要取消 CTS1（高电平），UART 将会继续发送 1 个比特。一旦 CTS1 被解除有效状态，发送将继续进行，在发送一个起始位后接着发送下一个字符的数据位。

37.5.10 UART1 线路状态寄存器

U1LSR 是一个只读寄存器，提供 UART1 TX 和 RX 模块的状态信息。

表 841. UART1 线路状态寄存器 （LSR - 地址 0x4008 2014）位描述

位	符号	值	描述	复位值
0	RDR		接收器数据就绪。 当 U1RBR 保存一个未读字符时 U1LSR [0] 被设置，当 UART1 RBR FIFO 为空时被清除。	0
		0	UART1 接收器 FIFO 为空。	
		1	UART1 接收器 FIFO 不为空。	
1	OE		超限错误。 超限错误一发生，即会设置超限错误条件。 U1LSR 读取操作将会清除 U1LSR [1]。当 UART1 RSR 有一个新的字符组成且 UART1 RBR FIFO 为满时，将设置 U1LSR [1]。在这种情况下， UART1 RBR FIFO 将不会被覆盖，但 UART1 RSR 中的字符会丢失。	0
		0	超限错误状态未激活。	
		1	超限错误状态激活。	
2	PE		奇偶校验错误。 当收到字符的奇偶位处于错误的状态时，将会产生一个奇偶校验错误。U1LSR 读取操作将会清除 U1LSR [2]。检测到奇偶校验错误的时间取决于 U1FCR [0]。 **注：**奇偶校验错误与 UART1 RBR FIFO 顶部的字符相关。	0
		0	奇偶校验错误状态未激活。	
		1	奇偶校验错误状态激活。	
3	FE		帧处理错误。 当收到字符的停止位为逻辑 0 时，将产生一个帧处理错误。U1LSR 读取操作将会清除 U1LSR [3]。检测出帧处理错误的时间取决于 U1FCR0。在检测到帧处理错误后，RX 将试图重新同步到数据并且将错误的停止位当成早到的开始位。当然，不能假定下一个收到的字节是正确的，即使没有帧处理错误。 **注：**帧处理错误与 UART1 RBR FIFO 顶部的字符相关。	0
		0	帧处理错误状态未激活。	
		1	帧处理错误状态激活。	

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 841. UART1 线路状态寄存器 （LSR - 地址 0x4008 2014）位描述 *（续）*

位	符号	值	描述	复位值
4	BI		中止中断。 当 RXD1 为一个全字符发送（开始、数据、奇偶、停止）保持为空格状态（全 0），中止中断将会产生。检测到断点条件后，接收器即进入空闲，直至 RXD1 进入标记状态（全部为 1）为止。U1LSR 读取操作将清除该状态位。检测到中止的时间取决于 U1FCR [0]。 **注：**中止中断与 UART1 RBR FIFO 顶部的字符相关。	0
		0	中止中断状态未激活。	
		1	中止中断状态激活。	
5	THRE		发送保持寄存器空。 THRE 在检测到一个空的 UART1 THR 后立即被设置，在 U1THR 写入操作发生时被清除。	1
		0	U1THR 包含有效数据。	
		1	U1THR 为空。	
6	TEMT		发送器空。 当 U1THR 和 U1TSR 都为空时，将设置 TEMT，当 U1TSR 或 U1THR 中的任何一个包含有效数据时，将清除 TEMT。	1
		0	U1THR 和 / 或 U1TSR 包含有效数据。	
		1	U1THR 和 U1TSR 皆为空。	
7	RXFE		RX FIFO 中的错误。 当一个含有 RX 错误 （例如帧处理错误、奇偶校验错误或中止中断）的字符被载入 U1RBR 时，将设置 U1LSR [7]。当 U1LSR 寄存器被读取且 UART1 FIFO 没有后续错误时，将清除该位。	0
		0	U1RBR 不包含 UART1 RX 错误或 U1FCR[0]=0。	
		1	UART1 RBR 包含至少一个 UART1 RX 错误。	
31:8	-		保留，从保留位读取的值未定义。	不适用

37.5.11 UART1 调制解调器状态寄存器

U1MSR为只读寄存器，提供调制解调器输入信号的状态信息。U1MSR [3:0]在发生U1MSR读取操作时被清除。注意调制解调器信号对于 UART1 运行没有直接影响，而只是方便软件实施调制解调器信号操作。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 842. UART1 调制解调器状态寄存器 （MSR - 地址 0x4008 2018）位描述

位	符号	值	描述	复位值
0	DCTS		Delta CTS。 输入 CTS 状态有变时，设置； U1MSR 读取时，清除。	0
		0	未检测到调制解调器输入 CTS 的变化。	
		1	检测到调制解调器输入 CTS 的变化。	
1	DDSR		Delta DSR。 发生 DSR 输入状态变化时被设置。 U1MSR 读取时，清除。	0
		0	未检测到调制解调器输入 DSR 的变化。	
		1	检测到调制解调器输入 DSR 的变化。	
2	TERI		后沿 RI。 在输入 RI 由低电平跳变为高电平时被设置。 U1MSR 读取时，清除。	0
		0	未检测到调制解调器输入 RI 的变化。	
		1	检测到 RI 上低电平至高电平的跳变。	
3	DDCD		Delta DCD。发生 DCD 输入状态变化时被设置。 U1MSR 读取时，清除。	0
		0	未检测到调制解调器输入 DCD 的变化。	
		1	检测到调制解调器输入 DCD 的变化。	
4	CTS	-	准许发送状态。输入信号 CTS 的补植。该位在调制解调器环回模式下连接到 U1MCR [1]。	0
5	DSR	-	数据集就绪状态。输入信号 DSR 的补值。该位在调制解调器环回模式下连接到 U1MCR [0]。	0
6	RI	-	振铃指示器状态。输入 RI 的补值。该位在调制解调器环回模式下连接到 U1MCR [2]。	0
7	DCD	-	数据载波检测状态。输入 DCD 的补值。该位在调制解调器环回模式下连接到 U1MCR [3]。	0
31:8	-	-	保留，从保留位读取的值未定义。	不适用

37.5.12 UART1 暂存寄存器

U1SCR 对 UART1 操作无影响。该寄存器可由用户决定写入和 / 或读取。没有提供中断接口向主机表示 U1SCR 是否发生了读取或者写入操作。

表 843. UART1 暂存寄存器 （SCR - 地址 0x4008 2014）位描述

位	符号	描述	复位值
7:0	Pad	暂存。一个可读、可写的字节。	0x00
31:8	-	保留，从保留位读取的值未定义。	不适用

37.5.13 UART1 自动波特率控制寄存器

UART1 自动波特率控制寄存器 (U1ACR) 控制为波特率生成测量输入时钟 / 数据速率的过程，并且可由用户决定读取和写入。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 844. 自动波特率控制寄存器 （ACR - 地址 0x4008 2020）位描述

位	符号	值	描述	复位值
0	START		自动波特率起始位。 该位在自动波特率完成之后自动清除。	0
		0	自动波特率停止 （自动波特率不运行）。	
		1	自动波特率起动（自动波特率运行）。自动波特率运行位。自动波特率完成后此位自动清除。	
1	MODE		自动波特率模式选择位。	0
		0	模式 0。	
		1	模式 1。	
2	AUTORESTART		自动波特率重启位。	0
		0	不重启	
		1	在超时的情况下重启 （计数器在下一个 UART1 Rx 下降沿重启）	
7:3	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	0
8	ABEOINTCLR		自动波特率结束中断清除位 （只写）。	0
		0	写入 0 没有影响。	
		1	写入 1 将清除 U1IIR 中相对应的中断。	
9	ABTOINTCLR		自动波特率超时中断清除位 （只写）。	0
		0	写入 0 没有影响。	
		1	写入 1 将清除 U1IIR 中相对应的中断。	
31:10	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	0

37.5.14 自动波特率

UART1 自动波特率功能可用于基于 AT 协议 （Hayes 指令）测量输入波特率。如果使能自动波特率功能，它将会测量接收数据流的位时间，并相应地设置除数锁存器寄存器 U1DLM 和 U1DLL。

注：在自动波特率运行期间小数分频器没有连接，因此当需要自动波特率功能时不应使用它。

设置 U1ACR 起始位可以启动自动波特率。清除 U1ACR 开始位可以中止自动波特率。自动波特率完成后起始位将清除，且读取位的操作将返回自动波特率的状态 （挂起 / 完成）。

自动波特率有两个测量模式可用，可通过 U1ACR 模式位进行选择。 0 模式下，在 UART1 Rx 引脚的两个后续下降沿 （开始位的下降沿和最低有效位下降沿）上测量波特率。 1 模式下，则是在 UART1 Rx 引脚的下降沿和后续上升沿之间 （起始位的长度）测量波特率。

U1ACR 的 AutoRestart 位可用来在发生超时的时候 （波特率测量计数器溢出）自动重启波特率测量。如果设置了该位，波特率测量将在 UART1 Rx 引脚的下一个下降沿重新启动。

自动波特率功能可产生两种中断。

- 如果使能中断 （设置了 U1IER ABToIntEn 且自动波特率测量计数器溢出），将设置 U1IIR ABTOInt 中断。
- 如果使能中断 （设置了 U1IER ABEOIntEn 且自动波特率成功结束），将设置 U1IIR ABEOInt 中断。

自动波特率中断必须通过设置相应的 U1ACR ABTOIntClr 和 ABEOIntEn 位来清除。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

在自动波特率期间，小数波特率发生器通常被禁用 （即 DIVADDVAL = 0）。然而，如果使能小数波特率生成器 (DIVADDVAL > 0)，将会影响对 UART1 Rx 引脚波特率的测量，但是 U1FDR 寄存器的值在波特率测量后不会被修改。另外，当使用自动波特率时，任何对 U1DLM 和 U1DLL 寄存器的写入都应该在 U1ACR 寄存器写入操作之前完成。UART1 支持的最小和最大波特率是 pclk、数据位数、停止位数和奇偶位数的函数。

(8)

$$ratemin = \frac{2 \times PCLK}{16 \times 2^{15}} \leq UART1_{baudrate} \leq \frac{PCLK}{16 \times (2 + databits + paritybits + stopbits)} = ratemax$$

37.5.15 自动波特率模式

软件预期会出现一条 AT 命令时，会使用预期字符格式配置 UART1，并设置 U1ACR 起始位。无需考虑除数锁存器 U1DLM 和 U1DLM 中的初始值。由于 A 或者 a 的 ASCII 编码（” A" = 0x41, “a” = 0x61），UART1 Rx 引脚检测到起始位，且预期字符的最低有效位由两个下降沿分隔。当 U1ACR 起始位被置位时，自动波特率协议将执行以下步骤：

1. 在 U1ACR 起始位设置时，将复位波特率测量计数器和 UART1 的 U1RSR。U1RSR 波特率调至最高速率。
2. UART1 Rx 引脚下降沿启动起始位的起点。速率测量计数器开始有选择地对使用小数波特率生成器进行预定标的 pclk 周期进行计数。
3. 在接收起始位期间，RSR 波特输入中产生 16 个脉冲，使用 UART1 输入时钟频率（小数波特率预定标），保证起始位存储在 U1RSR 中。
4. 接收起始位期间 （且字符最低有效位设为 0 = 模式），速率计数器继续随着预分频的 UART1 输入时钟 (pclk) 而增加。
5. 如果模式为 0，则速率计数器将在 UART1 Rx 引脚的下一个下降沿停止。如果模式为 1，则速率计数器将在 UART1 Rx 引脚的下一个上升沿停止。
6. 速率计数器的值被载入到 U1DLM/U1DLL 中，并且波特率将会切换到正常操作模式。设置完 U1DLM/U1DLL 后，将设置自动波特率中断 U1IIR ABEOInt 结束 （如果使能了的话）。 U1RSR 现在将继续接收 A/a 字符的剩余位。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

a. 模式 0（起始位和 LSB 均用于自动波特率）

b. 模式 1（仅起始位用于自动波特率）

图 117. 自动波特率 a) 模式 0 和 b) 模式 1 的波形图

37.5.16 UART1 小数分频器寄存器

UART1 小数分频器寄存器 (U1FDR) 控制波特率生成时钟预分频器，可由用户决定读取或写入。预换算器采用 APB 时钟并根据指定的小数要求生成输出时钟。

注意事项：如果小数分频器处于有效状态 (DIVADDVAL>0) 且 DLM = 0，DLL 寄存器的值必须大于 2。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 845. UART1 小数分频器寄存器 （FDR - 地址 0x4008 2028）位描述

位	函数	描述	复位值
3:0	DIVADDVAL	波特率发生器预分频器除数值。如果该字段为 0，则小数波特率生成器将不会影响 UARTn 的波特率。	0
7:4	MULVAL	波特率预分频器乘数值。不管是否使用小数波特率发生器，为了让 UARTn 正常运作，该字段必须大于或等于 1。	1
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	0

此寄存器控制波特率发生的时钟预换算器。该寄存器的复位值使 UART1 的小数能力处于禁用状态以保证 UART1 在软件和硬件上与未配备此特性的 UART 全面兼容。

UART1 波特率计算方式如下 (n=1)：

(9)

$$UART1_{baudrate} = \frac{PCLK}{16 \times (256 \times U1DLM + U1DLL) \times \left(1 + \frac{DivAddVal}{MulVal}\right)}$$

PCLK 为外设时钟时，U1DLM 和 U1DLL 为 UART1 的标准波特率分频寄存器，DIVADDVAL 和 MULVAL 为 UART1 小数波特率生成器的具体参数。

MULVAL 和 DIVADDVAL 的值应符合以下条件：

1. 1 ≤ MULVAL ≤ 15
2. 0 ≤ DIVADDVAL ≤ 14
3. DIVADDVAL< MULVAL

在发送 / 接收数据过程中，不能修改 U1FDR 的值，否则可能造成数据丢失或损坏。

如果U1FDR寄存器值不符合上述两个要求，则小数分频器的输出不明确。如果DIVADDVAL为零，则小数分频器禁用，时钟不会被分频。

37.5.16.1 波特率计算

无论是否使用小数分频器， UART1 都能正常运行。在实际应用中，使用几个不同的小数分频器设置就可能获得所需的波特率。以下算法展示了一种查找一组 DLM、 DLL、 MULVAL 和 DIVADDVAL 值的方法。这组参数产生的波特率相对于所需波特率的误差小于 1.1%。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 118. 设置 UART 分频器的算法

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 846. 小数分频器设置查找表

FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal
1.000	0/1	1.250	1/4	1.500	1/2	1.750	3/4
1.067	1/15	1.267	4/15	1.533	8/15	1.769	10/13
1.071	1/14	1.273	3/11	1.538	7/13	1.778	7/9
1.077	1/13	1.286	2/7	1.545	6/11	1.786	11/14
1.083	1/12	1.300	3/10	1.556	5/9	1.800	4/5
1.091	1/11	1.308	4/13	1.571	4/7	1.818	9/11
1.100	1/10	1.333	1/3	1.583	7/12	1.833	5/6
1.111	1/9	1.357	5/14	1.600	3/5	1.846	11/13
1.125	1/8	1.364	4/11	1.615	8/13	1.857	6/7
1.133	2/15	1.375	3/8	1.625	5/8	1.867	13/15
1.143	1/7	1.385	5/13	1.636	7/11	1.875	7/8
1.154	2/13	1.400	2/5	1.643	9/14	1.889	8/9
1.167	1/6	1.417	5/12	1.667	2/3	1.900	9/10
1.182	2/11	1.429	3/7	1.692	9/13	1.909	10/11
1.200	1/5	1.444	4/9	1.700	7/10	1.917	11/12
1.214	3/14	1.455	5/11	1.714	5/7	1.923	12/13
1.222	2/9	1.462	6/13	1.727	8/11	1.929	13/14
1.231	3/13	1.467	7/15	1.733	11/15	1.933	14/15

37.5.16.1.1 示例 1：PCLK = 14.7456 MHz、 BR = 9600

根据所提供的算法，DL_{est} = PCLK/(16 x BR) = 14.7456 MHz / (16 x 9600) = 96。因为这里的 DL_{est} 是一个整数，所以 DIVADDVAL = 0， MULVAL = 1， DLM = 0，且 DLL = 96。

37.5.16.1.2 示例 2：PCLK = 12 MHz、 BR = 115200

根据所提供的算法，DL_{est} = PCLK/(16 x BR) = 12 MHz / (16 x 115200) = 6.51。此处 DL_{est} 不是整数，下一步可以估测 FR 参数。使用 FR_{est} = 1.5 进行首次估算，得到新的 DL_{est} = 4，然后使用 FR_{est} = 1.628 再进行计算。由于 FRest = 1.628 是在 1.1 到 1.9 的指定范围之内，因此 DIVADDVAL 和 MULVAL 的值可通过附带的查找表获得。

在查找表表 846 中，最接近 FRest = 1.628 的值为 FR = 1.625。也就是说 DIVADDVAL = 5 而 MULVAL = 8。

根据这些查到的值，建议的 UART 设置应为：DLM = 0， DLL = 4， DIVADDVAL = 5 和 MULVAL = 8。根据方程 9，UART 的波特率为 115384。该速率与原来指定的 115200 之间存在 0.16% 的相对误差。

37.5.17 UART1 发送使能寄存器

除了配备全面的硬件流量控制（如上文所述的 auto-cts 和 auto-rts 机制）以外， U1TER 同时使能运用软件流量控制。当 TxEn=1，只要有数据需要发送，UART1 发送器将持续发送。一旦 TxEn 变为 0， UART1 将停止发送。

虽然表 847 描述了如何利用 TXEn 位来实现硬件流控制，但我们强烈建议用户采用 UART1 硬件所实现的自动流控制特性处理硬件流控制，并限制 TXEn 对软件流控制的范围。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

U1TER 使能运用软件和硬件流量控制。当 TXEn=1，只要有数据需要发送，UART1 发送器将持续发送。一旦 TXEn 变为 0， UART1 将停止发送。

表 847 描述了如何利用 TXEN 位来实现软件流控制。

表 847. UART1 发送使能寄存器 （TER - 地址 0x4008 2030）位描述

位	符号	描述	复位值
6:0	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
7	TXEN	发送使能位。 当此位为 1，例如复位后，一旦任何之前的数据已发送，写入 THR 的数据将输出至 TXD 引脚。如果字符发送时该位被清 0，将完成该字符的发送，但是在重新设置该位之前，将不会再发送字符。也就是说，该位的 0 值阻止了 THR 或者 TX FIFO 传送字符至发送移位寄存器。当检测到硬件信息交换 TX-permit 信号 (CTS) 变为假时，或者对于软件信息交换，接收到 XOFF 字符 (DC3) 时，软件可以将清除该位。当检测到 TX-permit 信号变为真时，或者在接收到 XON (DC1) 字符时，软件可以将该位重新置位。	1
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

37.5.18 UART1 RS485 控制寄存器

U1RS485CTRL 寄存器控制 UART 在 RS-485/EIA-485 模式下的配置。

表 848. UART1 RS485 控制寄存器 （RS485CTRL - 地址 0x4008 204C）位描述

位	符号	值	描述	复位值
0	NMMEN		多点模式选择。	0
		0	RS-485/EIA-485 正常多点模式 (NMM) 禁用。	
		1	RS-485/EIA-485 正常多点模式 (NMM) 使能。在此模式下，当收到的字节导致 UART 设置奇偶校验错误并生成中断时，检测到地址。	
1	RXDIS		接收使能。	0
		0	接收器使能。	
		1	接收器禁用。	
2	AADEN		自动地址检测使能。	0
		0	自动地址检测 (AAD) 禁用。	
		1	自动地址检测 (AAD) 使能。	
3	SEL		方向控制。	0
		0	如果方向控制已使能 （位 DCTRL = 1），则引脚 $\overline{\text{RTS}}$ 用于方向控制。	
		1	如果方向控制已使能 （位 DCTRL = 1），则引脚 DTR 用于方向控制。	
4	DCTRL		方向控制使能。	0
		0	禁用自动方向控制。	
		1	使能自动方向控制。	
5	OINV		极性。 该位反转了 RTS （或 DTR）引脚上方向控制信号的极性。	0
		0	当发送器有数据要发送时，方向控制引脚会被驱动为逻辑 0。在最后一个数据位被发送后，该位就会被驱动为逻辑 1。	
		1	当发送器有数据要发送时，方向控制引脚会被驱动为逻辑 1。在最后一个数据位被发送后，该位就会被驱动为逻辑 0。	
31:6	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

37.5.19 UART1 RS-485 地址匹配寄存器

U1RS485ADRMATCH 寄存器包含 RS-485/EIA-485 模式下的地址匹配值。

表 849. UART1 RS485 地址匹配寄存器 （RS485ADRMATCH – 地址 0x4008 2050）位描述

位	符号	描述	复位值
7:0	ADRMATCH	包含地址匹配值。	0x00
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

37.5.20 UART1 RS-485 延迟值寄存器

用户可通过对 8 位 RS485DLY 寄存器编程来设置：最后一个停止位离开 TXFIFO 到使 $\overline{\text{RTS}}$（或 $\overline{\text{DTR}}$）无效之间的延迟。此延迟时间以波特率时钟为周期。可以编程 0 到 255 位倍数的任何延迟时间。

表 850. UART1 RS485 延迟值寄存器 （RS485DLY – 地址 0x4008 2054）位描述

位	符号	描述	复位值
7:0	DLY	包含方向控制 （RTS 或 DTR）延迟值。此寄存器与一个 8 位计数器共同工作。	0x00
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

37.5.21 RS-485/EIA-485 模式的操作

RS-485/EIA-485 功能可将 UART 配置为可寻址从机。可寻址从机是由单个主机控制的多个从机之一。

UART 主机发送器将设置奇偶位 （第 9 位）为 1 以标识一个地址字符。对于数据字符，将奇偶校验位设置为 0。

可为每个 UART 从机接收器分配一个唯一的地址。可以对从机进行编程，使其手动或自动拒绝不是其自身地址后的数据。

RS-485/EIA-485 正常多点模式 (NMM)

将 RS485CTRL 位设置为 0 可使能此模式。在此模式下，当收到的字节导致 UART 设置奇偶校验错误并生成中断时，检测到地址。

如果接收器被禁用（RS485CTRL 位 1=1），任何接收到的数据字节都将被忽略，且不会存入 RXFIFO。检测到地址字节 （奇偶校验位 =1）时，会将其放入 RXFIFO，并生成一个“Rx 数据就绪中断”。处理器然后即可读取地址字节，并决定是否使能接收器来接收随后的数据。

当接收器被使能 （RS485CTRL 位 1=0），所有接收到的字节将会被接受并存入 RXFIFO，不论是数据还是地址。收到地址字符时，将生成一个奇偶校验错误中断，然后处理器会决定是否禁用接收器。

RS-485/EIA-485 自动地址检测 (AAD) 模式

RS485CTRL 寄存器位 0 （9 位模式使能）和 2 （AAD 模式使能）都设置时， UART 处于自动地址检测模式。

在该模式下，接收器会将任何收到的地址字节（奇偶校验 = 1）与编程到RS485ADRMATCH 寄存器中的 8 位值进行比较。

如果接收器被禁用 （RS485CTRL 位 1=1），任何接收到的与 RS485ADRMATCH 值不匹配的数据字节或者地址字节都将被丢弃。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

检测到匹配的地址字符时，会将其连同奇偶校验位一起推入 RXFIFO，然后接收器将自动使能 （RS485CTRL 位 1 将由硬件清除）。接收器也将产生 n Rx 数据就绪中断。

当接收器被使能 （RS485CTRL 位 1=0），所有接收到的字节将被接受并存入 RXFIFO，直到接收到一个与 RS485ADRMATCH 值不匹配的地址字节。出现这种情况时，接收器将在硬件中自动禁用 （RS485CTRL 位 1 将被设置），收到的不匹配地址字符将不会存储在 RXFIFO 中。

RS-485/EIA-485 自动方向控制

RS485/EIA-485 模式包含启用发送器自动控制 $\overline{\text{RTS}}$ 引脚或 $\overline{\text{DTR}}$ 引脚状态作为方向控制输出信号的选项。

设置 RS485CTRL 位 4=1 将使能该功能。

如果已使能方向控制，当 RS485CTRL 位 3=0 时会使用 $\overline{\text{RTS}}$ 引脚。当 RS485CTRL 位 3=1 时，将使用 $\overline{\text{DTR}}$ 引脚。

当使能自动方向控制时，在 CPU 写数据到 TXFIFO 时被选中的引脚会生效 （驱动为低电平）。一旦数据的最后一位发送完毕，引脚将变为无效 （驱动至高电平）。请参见 RS485CTRL 寄存器中的位 4 和 5。

RS485CTRL 位 4 比所有其他机制都优先控制 $\overline{\text{RTS}}$ （或 $\overline{\text{DTR}}$），环回模式例外。

RS485/EIA-485 驱动器延迟时间

驱动器延迟时间指最后一个停止位离开 TXFIFO 和解除 RTS（或 DTR）的有效状态之间的延迟。此延迟时间可在 8 位 RS485DLY 寄存器中编程，延迟时间以波特率时钟为周期。可以编程 0 到 255 位倍数的任何延迟时间。

RS485/EIA-485 输出反转

引脚 $\overline{\text{RTS}}$ （或 $\overline{\text{DTR}}$）方向控制信号的极性可通过对寄存器 U1RS485CTRL 第 5 位编程来反转。此位设置后，发送器有要发送的数据时，方向控制引脚将驱动到逻辑 1。数据的最后一位发送后，方向控制引脚将被驱动到逻辑 0。

37.5.22 UART1 FIFO 电平寄存器

U1FIFOLVL 寄存器为只读寄存器，启用软件读取当前的 FIFO 级别状态。发送和接收 FIFO 的电平均呈现在该寄存器中。

表 851. UART1 FIFO 电平寄存器 （FIFOLVL – 地址 0x4008 2058）位描述

位	符号	描述	复位值
3:0	RXFIFILVL	反映当前 UART1 接收器 FIFO 的级别。 0 = 空， 0xF = FIFO 为满状态。	0x00
7:4	-	保留。未定义从保留位读取的值。	不适用
11:8	TXFIFOLVL	反映当前 UART1 发送器 FIFO 的级别。 0 = 空， 0xF = FIFO 为满状态。	0x00
31:12	-	保留，从保留位读取的值未定义。	不适用

37.6 架构

下面的方框图显示了 UART1 的架构。

APB 接口提供 CPU 或主机与 UART1 之间的通信链路。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

UART1 接收模块 U1RX 监控串行输入线路 RXD1 上的有效输入。 UART1 RX 移位寄存器 (U1RSR) 通过 RXD1 接受有效字符。有效字符在 U1RSR 被组合以后将转移至 UART1 RX 缓存寄存器 FIFO，等待 CPU 或主机通过通用主机接口访问。

UART1 发送器模块 U1TX 接受 CPU 或主机写入的数据并将数据缓冲至 UART1 TX 保持寄存器 FIFO (U1THR)。UART1 TX 移位寄存器 (U1TSR) 读取 U1THR 中存储的数据并且组合此数据通过串行输出引脚 TXD1 发送。

UART1 波特率生成器模块 U1BRG，产生的时序使能用于 UART1 TX 模块。 U1BRG 时钟输入源为 APB 时钟 (PCLK)。主时钟依据 U1DLL 和 U1DLM 寄存器中指定的除数分频。此分频后的时钟是 16 倍过采样时钟， NBAUDOUT。

调制解调器接口包含 U1MCR 和 U1MSR 寄存器。该接口负责调制解调器外设和 UART1 之间的信息交换。

中断接口包含 U1IER 和 U1IIR 寄存器。中断接口从 U1TX 和 U1RX 模块接收几个一时钟周期宽的使能。

U1TX 和 U1RX 的状态信息存储在 U1LSR 中。U1TX 和 U1RX 的控制信息存储在 U1LCR 中。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 119. UART1 功能框图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

38.1 本章导读

所有 LPC43xx 器件上均有 SSP0/1 控制器。

38.2 基本配置

SSP0/1 配置如下：

- 参见表 852 以了解时钟与电源控制。
- SSP0/1 由 SSP0/1_RST 进行复位 （复位 #50/51）。
- SSP0/1 中断会连接到 NVIC 中 # 22/23 插槽。
- 要将 SSP0/1 接收和发送线连接至 GPDMA，请使用 CREG 模块中的 DMAMUX 寄存器（参见表 41）并使能 DMA 通道配置寄存器 （第 19.6.20 节）中的 GPDMA 通道。

表 852. SSP0/1 时钟和电源控制

	基本时钟	分支时钟	工作频率
SSP0 寄存器接口时钟	BASE_M4_CLK	CLK_M4_SSP0	高达 204 MHz
SSP0 外围设备时钟 (PCLK)	BASE_SSP0_CLK	CLK_APB0_SSP0	高达 204 MHz
SSP1 寄存器接口时钟	BASE_M4_CLK	CLK_M4_SSP1	高达 204 MHz
SSP1 外围设备时钟 (PCLK)	BASE_SSP1_CLK	CLK_APB2_SSP1	高达 204 MHz

38.3 功能要点

- 兼容摩托罗拉 SPI、 4 线德州仪器 SSI 和国家半导体 Microwire 总线。
- 同步串行通信。
- 支持主机或从机操作。
- 同时适用于发送与接收的 8 帧 FIFO。
- 4 位至 16 位帧。

38.4 简介

SSP 是同步串行端口 (SSP) 控制器，可控制 SPI、4 线 SSI 或 Microwire 总线的操作。它可以与总线上的多个主机和从机进行交互。在指定数据传输中，总线上只有一个主机和一个从机进行通信。数据传输原则上为全双工，主机和从机之间流动数据帧长为 4 至 16 位。实际上，通常情况下只有一个方向上的数据流包含有意义的数据。

LPC43xx 有两个同步串行端口控制器 -- SSP0 和 SSP1。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

38.5 引脚说明

表 853. SSP 引脚描述

引脚名称	方向	接口引脚名称 / 功能			引脚说明
		SPI	SSI	Microwire	
SCK0/1	I/O	SCK	CLK	SK	**串行时钟。**SCK/CLK/SK是用于同步数据传输的时钟信号。由主机驱动，从机接收。当使用 SPI 接口时，时钟可编程为高电平有效或低电平有效，否则，时钟总是高电平有效。SCK1 的状态只能在数据传送过程中改变。在其他时间里， SSPn 接口会使其保持无效状态或不驱动它 （使其处于高阻态）。
SSEL0/1	I/O	SSEL	FS	CS	**帧同步 / 从机选择。**当 SSPn 接口是总线主机时，它会在串行数据启动前驱动该信号，使之变为有效状态，并在串行数据发送后释放该信号，使之变为无效状态。该信号的有效状态可以是高电平或低电平，具体取决于所选的总线和模式。当 SSPn 是总线从机时，该信号会根据所使用的协议来限定主机中数据的状态。 当只有一个总线主机和一个总线从机时，来自主机的帧同步或从机选择信号可直接连接到从机的相应输入中。当总线上有多个从机时，通常必需进一步限制其帧选择 / 从机选择输入，以避免多个从机对传输作出响应。
MISO0/1	I/O	MISO	DR(M) DX(S)	SI(M) SO(S)	**主机输入从机输出。**MISO 信号将串行数据由从机传输到主机。当 SSPn 是从机时，串行数据从该信号输出。当 SSPn 是主机时，串行数据从该信号输入。当 SSPn 是从机且未被 FS/SSEL 选择时，它将不驱动该信号 （使其处于高阻态）。
MOSI0/1	I/O	MOSI	DX(M) DR(S)	SO(M) SI(S)	**主机输出从机输入。**MOSI 信号将串行数据从主机传输到从机。当 SSPn 是主机时，串行数据从该信号输出。当 SSPn 是从机时，串行数据从该信号输入。

38.6 寄存器描述

SSP 控制器的寄存器地址如表 854 和表 855 所示。

表 854. 寄存器简介：SSP0 （基址 0x4008 3000）

名称	访问类型	地址偏移	描述	复位值[1]	参考
CR0	R/W	0x000	控制寄存器 0。选择串行时钟速率、总线类型和数据大小。	0	表 856
CR1	R/W	0x004	控制寄存器 1。选择主机 / 从机和其他模式。	0	表 857
DR	R/W	0x008	数据寄存器。写满发送 FIFO，读空接收 FIFO。	0	表 858
SR	RO	0x00C	状态寄存器	0x0000 0003	表 859
CPSR	R/W	0x010	时钟预定标寄存器	0	表 860
IMSC	R/W	0x014	中断掩码设置和清除寄存器	0	表 861
RIS	RO	0x018	原始中断状态寄存器	0x0000 0008	表 862
MIS	RO	0x01C	掩码中断状态寄存器	0	表 863
ICR	WO	0x020	SSPICR 中断清除寄存器	-	表 864
DMACR	R/W	0x024	SSP0 DMA 控制寄存器	0	表 865

[1] 复位值仅反映存储在所用位中的数据，不包括保留位的内容。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 855. 寄存器简介：SSP1 （基址 0x400C 5000）

名称	访问类型	地址偏移	描述	复位值[1]	参考
CR0	R/W	0x000	控制寄存器 0。选择串行时钟速率、总线类型和数据大小。	0	表 856
CR1	R/W	0x004	控制寄存器 1。选择主机 / 从机和其他模式。	0	表 857
DR	R/W	0x008	数据寄存器。写满发送 FIFO，读空接收 FIFO。	0	表 858
SR	RO	0x00C	状态寄存器	0x0000 0003	表 859
CPSR	R/W	0x010	时钟预定标寄存器	0	表 860
IMSC	R/W	0x014	中断掩码设置和清除寄存器	0	表 861
RIS	RO	0x018	原始中断状态寄存器	0x0000 0008	表 862
MIS	RO	0x01C	掩码中断状态寄存器	0	表 863
ICR	R/W	0x020	SSPICR 中断清除寄存器	-	表 864
DMACR	R/W	0x024	SSP1 DMA 控制寄存器	0	表 865

[1] 复位值仅反映存储在所用位中的数据，不包括保留位的内容。

38.6.1 SSP 控制寄存器 0

该寄存器控制 SSP 控制器的基本运行。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 856. SSP 控制寄存器 0 （CR0 - 地址 0x4008 3000 (SSP0)、 0x400C 5000 (SSP1)）位描述

位	符号	值	描述	复位值
3:0	DSS		数据大小选择。该域控制每帧中传输的位数。不支持且不使用值 0000-0010。	0000
		0x3	4 位传输	
		0x4	5 位传输	
		0x5	6 位传输	
		0x6	7 位传输	
		0x7	8 位传输	
		0x8	9 位传输	
		0x9	10 位传输	
		0xA	11 位传输	
		0xB	12 位传输	
		0xC	13 位传输	
		0xD	14 位传输	
		0xE	15 位传输	
		0xF	16 位传输	
5:4	FRF		帧格式。	00
		0x0	SPI	
		0x1	TI	
		0x2	Microwire	
		0x3	此组合不受支持，因此不应使用。	
6	CPOL		时钟输出极性。该位只用于 SPI 模式。	0
		0	在帧与帧之间由 SSP 控制器将总线时钟维持在低电平状态。	
		1	在帧与帧之间由 SSP 控制器将总线时钟维持在高电平状态。	
7	CPHA		时钟输出相位。该位只用于 SPI 模式。	0
		0	SSP 控制器在帧传输的第一次时钟跳变时捕获串行数据，也就是说跳变**远离**时钟线的帧间状态。	
		1	SSP 控制器在帧传输的第二次时钟跳变时捕获串行数据，也就是说跳变**回到**时钟线的帧间状态。	
15:8	SCR		串行时钟速率。总线上的每位预定标器输出时钟数目，减去 1。假设 CPSDVSR 为预分频器，APB 时钟 PCLK 计时预分频器，则位频率为 PCLK / (CPSDVSR × [SCR+1])。	0x00
31:16	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

38.6.2 SSP 控制寄存器 1

该寄存器控制 SSP 控制器运行的某些方面。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 857. SSP 控制寄存器 1 （CR1 - 地址 0x4008 3004 (SSP0)、 0x400C 5004 (SSP1)）位描述

位	符号	值	描述	复位值
0	LBM		环回模式。	0
		0	正常操作。	
		1	环回模式。串行输入取自串行输出 （MOSI 或 MISO），而不是串行输入引脚 （分别为 MISO 或 MOSI）。	
1	SSE		SSP 允许。	0
		0	SSP 控制器被禁用。	
		1	SSP 控制器将在串行总线上与其他装置进行交互。在设置该位之前，软件应该向其他 SSP 寄存器和中断控制器寄存器写入合适的控制信息。	
2	MS		主机/从机模式。只有在SSE位为0时，才能对该位执行写入操作。	0
		0	SSP 控制器作为总线上的主机，驱动 SCLK、 MOSI 和 SSEL 线路，接收 MISO 线路。	
		1	SSP 控制器作为总线上的从机，驱动 MISO 线路，接收 SCLK、MOSI 和 SSEL 线路。	
3	SOD		从机输出禁用。只有在从机模式下才与此位有关 (MS = 1)。如果值为 1，则禁止此 SSP 控制器驱动发送数据线 (MISO)。	0
31:4	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

38.6.3 SSP 数据寄存器

软件可将需要发送的数据写入该寄存器，并从中读取接收到的数据。

表 858. SSP 数据寄存器 （DR - 地址 0x4008 3008 (SSP0)、 0x400C 5008 (SSP1)）位描述

位	符号	描述	复位值
15:0	数据	**写入：**只要状态寄存器中的 TNF 位为 1，表示 Tx FIFO 不处于满状态，软件就可将即将在未来帧中发送的数据写入该寄存器。如果 Tx FIFO 之前为空且总线上的 SSP 控制器不在忙碌状态，可以立即开始数据发送。否则写入该寄存器的数据将在之前所有数据发送 （接收）完毕后立即发送出去。如果数据长度小于 16 位，则软件必须使写入该寄存器的数据向右对齐。 **读：**只要状态寄存器的 RNE 位为 1，表示 Rx FIFO 不为空，软件就可从该寄存器中读取数据。当软件从该寄存器中读取数据时， SSP 控制器将返回 Rx FIFO 中最近帧中的数据。如果数据长度小于 16 位，数据将在该域被右对齐，在高阶位上填 0。	0x0000
31:16	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

38.6.4 SSP 状态寄存器

该只读寄存器反映 SSP 控制器的当前状态。

表 859. SSP 状态寄存器 （SR - 地址 0x4008 300C (SSP0)、 0x400C 500C (SSP1)）位描述

位	符号	描述	复位值
0	TFE	发送 FIFO 为空。当发送 FIFO 为空时该位为 1，否则为 0。	1
1	TNF	发送 FIFO 未满。如果 Tx FIFO 已满，则该位为 0 ；反之为 1。	1
2	RNE	接收 FIFO 未空。如果接收 FIFO 为空，则该位为 0 ；反之为 1。	0
3	RFF	接收 FIFO 满。如果接收 FIFO 已满，则该位为 1 ；反之为 0。	0
4	BSY	忙。如果 SSPn 控制器空闲，则该位为 0 ；或如果当前正在发送 / 接收一个帧和 / 或 Tx FIFO 未空，则该位为 1。	0
31:5	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

38.6.5 SSP 时钟预分频寄存器

该寄存器控制预定标器分频 SSP 外设时钟 PCLK 以产生预定标器时钟的系数，反过来，再在 SSPnCR0 中被 SCR 系数分频，决定位时钟。

表 860. SSP时钟预分频寄存器（CPSR - 地址0x4008 3010 (SSP0)、0x400C 5010 (SSP1)）位描述

位	符号	描述	复位值
7:0	CPSDVSR	这是 2-254 中的一个偶数值。它是 PCLK 的分频系数，通过分频后得到预分频器输出时钟。位 0 始终读取为 0。	0
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

注意事项：SSPnCPSR 值必须适当初始化，否则 SSP 控制器不能正确发送数据。

在从机模式下，由主机提供的 SSP 时钟速率不能超过 SSP 外设时钟的 1/12。SSPnCPSR 寄存器的内容与此无关。

在主机模式下， $CPSDVSR_{min}$ = 2 或更大的值 （只能为偶数）。

38.6.6 SSP 中断掩码设置 / 清除寄存器

该寄存器控制是否使能 SSP 控制器中 4 个可能的中断条件。请注意，ARM 使用的“掩码”一词与传统计算机术语含义相反，对于后者，“掩码”表示 “禁用”。 ARM 使用 “掩码”一词来表示 “使能”。为了避免混淆，我们将不会使用 “掩码”一词。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 861. SSP中断掩码设置/清除寄存器（IMSC - 地址0x4008 3014 (SSP0)、0x400C 5014 (SSP1)）位描述

位	符号	描述	复位值
0	RORIM	出现接收溢出 （即当 Rx FIFO 已满且另一个帧完全接收）时，软件应将此位设置为使能中断。ARM 规范指明，出现这种情况时，前面的帧数据会被新的帧数据覆盖。	0
1	RTIM	出现接收超时条件时，软件应将此位设置为使能中断。当 Rx FIFO 不为空且超过超时周期没有被读取时，将发生接收超时。对于主机和从机模式来说超时周期是相同的，由 SSP 比特率决定：在 PCLK / (CPSDVSR × [SCR+1]) 时为 32 位。	0
2	RXIM	当 Rx FIFO 为至少半满状态时，软件应该设置此位以允许中断。	0
3	TXIM	当 Tx FIFO 为至少半空状态时，软件应该设置此位以允许中断。	0
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

38.6.7 SSP 原始中断状态寄存器

每个中断条件变为有效时，该只读寄存器中都会包含一个 1，不论 SSPnIMSC 中的中断是否已使能。

表 862. SSP 原始中断状态寄存器（RIS - 地址 0x4008 3018 (SSP0)、RIS - 0x400C 5018 (SSP1)）位描述

位	符号	描述	复位值
0	RORRIS	如果在 Rx FIFO 已满时完整接收到另一个帧，则该位为 1。 ARM 规范指明，出现这种情况时，前面的帧数据会被新的帧数据覆盖。	0
1	RTRIS	当 Rx FIFO 不为空且超过超时周期没有被读取时，该位为 1。对于主机和从机模式来说超时周期是相同的，由 SSP 比特率决定：在 PCLK / (CPSDVSR × [SCR+1]) 时为 32 位。	0
2	RXRIS	如果 Rx FIFO 为至少半满状态该位为 1。	0
3	TXRIS	如果 Tx FIFO 为至少半空状态该位为 1。	1
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

38.6.8 SSP 掩码中断状态寄存器

当每个中断条件变为有效且在 SSPnIMSC 中被使能时，该只读寄存器中都会包含一个 1。当出现 SSP 中断时，中断服务例程会读取此寄存器以确定中断的原因。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 863. SSP掩码中断状态寄存器（MIS -地址0x4008 301C (SSP0)、0x400C 501C (SSP1)）位描述

位	符号	描述	复位值
0	RORMIS	如果当 RxFIFO 状态为满时另外一帧被完全接收了，该位为 1 且该中断被使能。	0
1	RTMIS	当 Rx FIFO 不为空且超过超时周期没有被读取时，该位为 1 且该中断被使能。对于主机和从机模式来说超时周期是相同的，由 SSP 比特率决定：在 PCLK / (CPSDVSR × [SCR+1]) 时为 32 位。	0
2	RXMIS	如果 Rx FIFO 为至少半满状态，该位为 1 且该中断被使能。	0
3	TXMIS	如果 Tx FIFO 为至少半空状态，该位为 1 且该中断被使能。	0
31:4	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

38.6.9 SSP 中断清除寄存器

软件可向该只写寄存器写入 1 个或多个 1 以清除 SSP 控制器中相应的中断条件。注意其他两个中断条件可以通过写入或者读取正确的 FIFO 来清除，或者通过清除 SSPnIMSCZ 中对应的位值来禁用。

表 864. SSP 中断清除寄存器 （ICR - 地址 0x4008 3020 (SSP0)、 0x400C 5020 (SSP1)）位描述

位	符号	描述	复位值
0	RORIC	对该位写入 1 将清除 “当 RxFIFO 为满状态时帧已接收”中断。	不适用
1	RTIC	对该位写入 1 将清除 “Rx FIFO 不为空且超过超时周期没有被读取”中断。对于主机和从机模式来说超时周期是相同的，由 SSP 比特率决定：在 PCLK / (CPSDVSR × [SCR+1]) 时为 32 位。	不适用
31:2	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

38.6.10 SSP DMA 控制寄存器

SSPnDMACR 寄存器是 DMA 控制寄存器。它是读 / 写寄存器。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 865. SSP DMA 控制寄存器（DMACR - 地址 0x4008 3024 (SSP0)、0x400C 5024 (SSP1)）位描述

位	符号	描述	复位值
0	RXDMAE	接收 DMA 使能。将该位设置为 1 时，接收 FIFO 的 DMA 被使能，否则接收 DMA 被禁用。	0
1	TXDMAE	发送 DMA 使能。当该位被置 1 时，发送 FIFO 的 DMA 被使能，否则发送 DMA 被禁用	0
31:2	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

38.7 功能说明

38.7.1 德州仪器同步串行帧格式

图 120 显示了 SSP 模块支持的 4 线德州仪器同步串行帧格式。

a. 单帧传输

b. 连续 / 背靠背帧传输

图 120. 德州仪器同步串行帧格式：a) 单帧和 b) 连续 / 背靠背 2 帧传输

对于在该模式下配置为主机的设备， CLK 和 FS 会被强制为低电平，并且只要 SSP 空闲，发送数据线 DX 就会为三态。一旦发送 FIFO 的底部入口包含数据， FS 将被脉冲启动为一个 CLK 周期的高电平。将要发送的值也被从发送 FIFO 传输到发送逻辑中的串行移位寄存器。在下一个 CLK 的上升沿， 4 至 16 位数据帧的 MSB 被移出至 DX 引脚。同样，接收数据的 MSB 由片外串行从设备传送到 DR 引脚。

在每个 CLK 的下降沿，SSP 和片外串行从设备将各个数据位放入其串行移位器。LSB 被锁存后，在 CLK 的第一个上升沿，接收的数据从串行移位器传输到接收 FIFO。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

38.7.2 SPI 帧格式

SPI 接口是 4 线接口，其中 SSEL 信号用作从机选择。 SPI 格式的主要特性是 SCK 信号的非工作状态和相位可通过对 SSPCR0 控制寄存器内的 CPOL 和 CPHA 位编程设定。

38.7.2.1 时钟极性 (CPOL) 及相位 (CPHA) 控制

当 CPOL 时钟极性控制位为 0 时，将会在 SCK 引脚上产生一个稳态低电平值。如果 CPOL 时钟极性控制位为 1，当不传输数据时 CLK 引脚上将产生一个稳态高电平值。

CPHA 控制位选择捕获数据及允许数据更改状态的时钟沿。通过是否在第一个数据捕获沿发生之前允许时钟跳变，该位能最大程度上影响第一个被发送的位。当 CPHA 相位控制位为 0 时，数据将在第一个时钟沿跳变被捕获。如果 CPHA 时钟相位控制位为 1，数据将在第二个时钟沿跳变被捕获。

38.7.2.2 CPOL=0， CPHA=0 时的 SPI 格式

CPOL = 0 且 CPHA = 0 时 SPI 格式的单帧和连续帧传输信号序列如图 121 所示。

a. CPOL=0 且 CPHA=0 时的单帧传输

b. CPOL=0 且 CPHA=0 时的连续帧传输

图 121. CPOL=0 且 CPHA=0 时的 SPI 帧格式 （a) 单帧和 b) 连续帧传输）

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

在这种配置下，空闲期间：

- CLK 信号被强制为低电平。
- SSEL 被强制为高电平。
- 发送 MOSI/MISO 垫片处于高阻态。

如果使能 SSP，且在发送 FIFO 中存在有效数据，则 SSEL 主机信号被驱动为低电平，指示数据传输开始。这将使得从机数据被连接至主机的 MISO 输入线上。使能主机的 MOSI。

1/2 个 SCK 周期后，有效的主机数据将被传输到 MOSI 引脚。由于主机和从机数据均已设定，因此再过 1/2 个 SCK 周期， SCK 主时钟引脚就会变为高电平。

此时，在 SCK 信号的上升沿捕获数据，并在 SCK 信号的下降沿传播数据。

在单字发送的情况下，当数据字的所有位都传输完毕以后，SSEL 线在最后一位被捕获后的一个 SCK 周期后回到空闲高电平状态。

但是，在进行连续的背靠背传输时，传输各数据字之间的 SSEL 信号必须为高电平。这是因为如果 CPHA 位为逻辑值 0，从机选择引脚会冻结其串行外设寄存器中的数据，并且不允许其被更改。因此主机装置必须在每个数据传输之间升高从机装置的 SSEL 引脚上的电平以使能串行外设数据写入。连续传输完成后，SSEL 引脚在捕获到最后一位后的一个 SCK 周期内将返回至空闲状态。

38.7.2.3 CPOL=0， CPHA=1 时的 SPI 格式

CPOL = 0 且 CPHA = 1 时 SPI 格式的传输信号序列如图 122 所示，包含单帧和连续帧传输。

图 122. CPOL=0 且 CPHA=1 时的 SPI 帧格式

在这种配置下，空闲期间：

- CLK 信号被强制为低电平。
- SSEL 被强制为高电平。
- 发送 MOSI/MISO 垫片处于高阻态。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

如果使能 SSP，且在发送 FIFO 中存在有效数据，则 SSEL 主机信号被驱动为低电平，指示数据传输开始。使能主机的 MOSI 引脚。在又过了半个 SCK 周期以后，主机和从机的有效数据都被连接至各自的发送线上。同时， SCK 在上升沿跳变时使能。

然后数据将在 SCK 信号下降沿捕获，并且在上升沿传播。

在单字传输的情况下，当所有位都传输完毕以后，SSEL 线在最后一位被捕获后的一个 SCK 周期后回到空闲高电平状态。

对于连续背靠背传输，SSEL 引脚在连续数据字之间保持为低电平，结束过程与单字传输相同。

38.7.2.4 CPOL = 1 且 CPHA = 0 时的 SPI 格式

CPOL=1 且 CPHA=0 时， SPI 格式的单帧和连续帧传输信号序列如图 123 所示。

a. CPOL=1 且 CPHA=0 时的单帧传输

b. CPOL=1 且 CPHA=0 时的连续帧传输

图 123. CPOL = 1 且 CPHA = 0 时的 SPI 帧格式 （a) 单帧和 b) 连续帧传输）

在这种配置下，空闲期间：

- CLK 信号被强制为高电平。
- SSEL 被强制为高电平。
- 发送 MOSI/MISO 垫片处于高阻态。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

如果使能 SSP，且在发送 FIFO 中存在有效数据，则 SSEL 主机信号被驱动为低电平，指示数据传输开始，这会使从机数据立即传输到主机的 MISO 线上。使能主机的 MOSI 引脚。

1/2 个 SCK 周期后，有效的主机数据将被传输到 MOSI 线。现在主机和从机的数据都被设置好了，再过半个 SCK 周期 SCK 主机时钟引脚转至低电平。这意味着，在 SCK 信号的下降沿捕获数据并在 SCK 信号的上升沿传播数据。

在单字发送的情况下，当数据字的所有位都传输完毕以后，SSEL 线在最后一位被捕获后的一个 SCK 周期后回到空闲高电平状态。

但是，在进行连续的背靠背传输时，传输各数据字之间的 SSEL 信号必须为高电平。这是因为如果 CPHA 位为逻辑值 0，从机选择引脚会冻结其串行外设寄存器中的数据，并且不允许其被更改。因此主机装置必须在每个数据传输之间升高从机装置的 SSEL 引脚上的电平以使能串行外设数据写入。连续传输完成后，SSEL 引脚在捕获到最后一位后的一个 SCK 周期内将返回至空闲状态。

38.7.2.5 CPOL = 1 且 CPHA = 1 时的 SPI 格式

CPOL = 1 且 CPHA = 1 时 SPI 格式的传输信号序列如图 124 所示，包含单帧和连续帧传输。

图 124. CPOL = 1 且 CPHA = 1 时的 SPI 帧格式

在这种配置下，空闲期间：

- CLK 信号被强制为高电平。
- SSEL 被强制为高电平。
- 发送 MOSI/MISO 垫片处于高阻态。

如果使能 SSP，且在发送 FIFO 中存在有效数据，则 SSEL 主机信号被驱动为低电平，指示数据传输开始。使能主机的 MOSI。在又过了半个 SCK 周期以后，主机和从机的数据都被连接至各自的发送线上。同时 SCK 被一个下降沿跳变使能。然后，在 SCK 信号的上升沿捕获数据，并在 SCK 信号的下降沿传播数据。

发送单个字时，在传输完所有位后，在捕获最后一位后的一个 SCK 周期内， SSEL 线返回到其空闲高电平状态。对于连续背靠背发送，SSEL 引脚保持有效低电平状态，直到最后一个字的最后一位被捕获，然后如上所述回到其空闲状态。通常，当进行连续的背靠背传输时，连续的数据字之间 SSEL 引脚保持为低电平，终止条件与传送单个字时相同。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

38.7.3 国家半导体 Microwire 帧格式

图 125 显示了单帧的 Microwire 帧格式。图 126 显示了进行背靠背帧发送时的 Microwire 帧格式。

图 125. Microwire 帧格式 （单帧传输）

Microwire 格式与 SPI 格式非常相似，都是使用主 - 从消息传递技术，但它采用的传输方式是半双工方式而不是全双工方式。每次串行传输均从一个 8 位控制字开始，由 SSP 发送到片外从设备。在此发送过程中， SSP 不会接收输入的数据。发送完消息后，片外从机对其进行解码，然后在 8 位控制消息的最后一位发送结束后再等待一个串行时钟，以所需的数据进行响应。返回的数据长度为 4 到 16 位，使整个帧长度范围为 13 到 25 位。

在这种配置下，空闲期间：

- SK 信号被强制为低电平。
- CS 被强制为高电平。
- 发送数据线 SO 被任意强制为低电平。

通过向发送 FIFO 写入控制字节来启动传送。 CS 下降沿会使包含在发送 FIFO 底端的值传输到发送逻辑的串行移位寄存器，并使 8 位控制帧的 MSB 输出到 SO 引脚。 CS 在帧传输期间保持低电平。 SI 引脚在此传输过程中保持三态。

片外串行从设备在每个 SK 的上升沿将各控制位锁存到串行移位器中。当从设备将最后一位锁存后，控制字节会在一个时钟等待状态期间被译码，而从机则通过将数据传回至 SSP 来进行响应。每个位在 SK 的下降沿被驱动到 SI 线。 SSP 依次在 SK 的上升沿锁存每个位。对于单帧传输，在帧的末端，在最后一位已锁存到接收串行移位器后的一个时钟周期内，CS 信号会被置为高电平，这会导致数据被传输到接收 FIFO。

注：LSB 被接收移位器锁存后或当 CS 引脚变为高电平时，片外从设备的接收线在 SK 下降沿呈现三态。

对于连续帧传输，数据传输开始和结束的方式与单帧传输相同。然而，CS 线持续有效（保持低电平），且数据以背靠背方式进行传输。当前数据帧的 LSB 被接收后，下一帧的控制字节立即发送。在帧数据的 LSB 被锁存到 SSP 后，每个收到的值将在 SK 的下降沿传输到接收移位器。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 126. Microwire 帧格式 （连续传送）

38.7.3.1 Microwire 模式下 CS 相对于 SK 的建立和保持时间要求

在 Microwire 模式下，CS 变为低电平后，SSP 从机在 SK 上升沿时对接收数据的第一位进行采样。主机驱动 SK 自由运行时必须确保 CS 信号相对于 SK 上升沿有充足的建立和保持时间。

图 127 描绘了这些建立和保持时间的要求。相对于 SK 上升沿 （SSP 从机在该上升沿对接收数据的第一位进行采样）， CS 的建立时间必须至少为 SK （SSP 在 SK 上运行）周期的 2 倍。相对于该边沿之前的 SK 上升沿， CS 必须保持至少一个 SK 周期。

图 127. Microwire 帧格式建立及保持时间

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

39.1 本章导读

所有 LPC43xx 器件上均有 SPI 控制器。

39.2 基本配置

SPI 配置如下：

- 参见表 866 以了解时钟与电源控制。
- SPI 由 SPI_RST 进行复位 （复位 # 58）。
- SPI 中断会被连接到 Cortex-M0 NVIC 中的 NVIV 插槽 # 20。

表 866. SPI 的时钟与电源控制

	基本时钟	分支时钟	工作频率
SPI 时钟；外围设备 SPI 时钟	BASE_SPI_CLK	CLK_SPI	高达 204 MHz
外设总线控制器的时钟	BASE_PERIPH_CLK	CLK_PERIPH_BUS	高达 204 MHz
外设内核控制器的时钟	BASE_PERIPH_CLK	BASE_PERIPH_CORE	高达 204 MHz

39.3 功能要点

- 兼容串行外围设备接口 (SPI) 规范。
- 同步、串行、全双工通信。
- SPI 主机或从机。
- 最大数据位速率为外围设备时钟速率的 1/8。
- 每次可以传送 8 至 16 位。

39.4 简介

SPI 是一种全双工串行接口，可处理多个连接到给定总线上的主机和从机。在给定数据传输过程中，接口上只能有一个主机和一个从机进行通信。在数据传送中，主机始终会向从机发送 8 至 16 位的数据，而从机也始终会向主机发送一个字节的数据。

SPI 接口中实施的 SPI 解决方案的框图如图 128 所示。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 128. SPI 功能框图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

39.5 引脚说明

表 867. SPI 引脚描述

引脚名称	类型	引脚说明
SCK	输入/输出	**串行时钟。**SPI 时钟信号 (SCK) 用于同步 SPI 接口间的数据传输。SPI 始终由主机驱动，并由从机接收。时钟可编程为高电平有效或低电平有效。 SPI 只有在数据传送过程中才有效。在其他时间里，它要么是无效状态，要么是三态。
SSEL	输入	**从机选择。**SPI 从机选择信号 (SSEL) 是低电平有效信号，用以指明目前选择哪个从机参与数据传输。每个从机都有其独特的从机选择信号输入。数据事务开始前， SSEL 必须处于低电平状态，并且在过程中通常应保持低电平状态。在数据传送过程中， SSEL 信号一旦变为高电平，则数据传送会被视为中止。在这种情况下，从机会恢复成空闲状态，并且接收到的任何数据均会被丢弃。没有其他迹象可用于表明存在此异常。主机不直接驱动该信号。它可能由一个在软件控制下的简单通用 I/O 进行驱动。 **注：**请注意，该引脚仅为输入引脚。SPI 在主机模式下不能驱动从机上的 CS 输入。主机模式下，可通过任何 GPIO 引脚为 SPI 选择芯片。
MISO	输入/输出	**主机输入从机输出。**SPI 主机输入 / 从机输出信号 (MISO) 是一个单向信号，用于将串行数据从 SPI 从机传输到 SPI 主机。当器件是从机时，串行数据在该引脚上输出。当器件是主机时，串行数据在该引脚上输入。当从器件未被选中时，从机将驱动该信号 （使其处于高阻态）。
MOSI	输入/输出	**主机输出从机输入。**SPI 主机输出 / 从机输入信号 (MOSI) 是一个单向信号，用于将串行数据从 SPI 主机传输到 SPI 从机。当器件是主机时，串行数据在该引脚上输出。当器件是从机时，串行数据在该引脚上输入。

39.6 寄存器描述

SPI 所包含的寄存器如表 868 所示。所有寄存器都是可进行字节、半字和字访问的。

表 868. 寄存器简介：SPI （基址 0x4010 0000）

名称	访问类型	地址偏移	描述	复位值 [1]
CR	R/W	0x000	SPI 控制寄存器。该寄存器控制着 SPI 的操作。	0x00
SR	RO	0x004	SPI 状态寄存器。该寄存器显示 SPI 的状态。	0x00
DR	R/W	0x008	SPI 数据寄存器。该双向寄存器为 SPI 提供发送和接收的数据。发送数据通过写该寄存器提供给 SPI0。 SPI0 接收的数据可从该寄存器中读出。	0x00
CCR	R/W	0x00C	SPI时钟计数器寄存器。该寄存器控制主机SCK0的频率。	0x00
TCR	R/W	0x010	SPI 测试控制寄存器。仅用于功能测试。	0x00
TSR	R/W	0x014	SPI 测试状态寄存器。仅用于功能测试。	0x00
-	R/W	0x018	保留。	-
INT	R/W	0x01C	SPI 中断标志该寄存器包含了 SPI 接口的中断标志。	0x00

[1] 复位值仅反映存储在所用位中的数据，不包括保留位的内容。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

39.6.1 SPI 控制寄存器

SPCR 寄存器通过表 869 中所示的配置位设置来控制 SPI 的操作。

表 869. SPI 控制寄存器 （CR - 地址 0x4010 0000）位描述

位	符号	值	描述	复位值
1:0	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
2	BITENABLE	0	SPI 控制器每次传送会发送和接收 8 位数据。	0
		1	SPI 控制器发送和接收的位数由位 11:8 选择。	
3	CPHA		时钟相位控制决定了 SPI 传送时数据和时钟之间的关系，并控制从机传送的开始和结束时间。	0
		0	数据在 SCK 的第一个时钟沿采样。传送从 SSEL 信号激活时开始，并在 SSEL 信号停用时结束。	
		1	数据在 SCK 的第二个时钟沿采样。当 SSEL 信号激活时，传送从第一个时钟沿开始，并在最后一个采样时钟沿结束。	
4	CPOL		时钟极性控制。	0
		0	SCK 为高电平有效。	
		1	SCK 为低电平有效。	
5	MSTR		主机模式选择	0
		0	SPI 处于从机模式。	
		1	SPI 处于主机模式。	
6	LSBF		LSB 优先用来控制传送的每个字节的移动方向。	0
		0	SPI 数据传送 MSB （位 7）优先。	
		1	SPI 数据传送 LSB （位 0）优先。	
7	SPIE		串行外围设备中断使能。	0
		0	SPI 中断被禁止。	
		1	每次激活 SPIF 或 MODF 位时都会产生硬件中断。	
11:8	BITS		当该寄存器的位 2 为 1 时，这个字段控制每次传送的位数：	0000
		0x8	每次传送 8 位	
		0x9	每次传送 9 位	
		0xA	每次传送 10 位	
		0xB	每次传送 11 位	
		0xC	每次传送 12 位	
		0xD	每次传送 13 位	
		0xE	每次传送 14 位	
		0xF	每次传送 15 位	
		0x0	每次传送 16 位	
31:12	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

39.6.2 SPI 状态寄存器

SPSR 寄存器根据表 870 中所示的配置位设置来控制 SPI 的操作。

表 870. SPI 状态寄存器 （SR - 地址 0x4010 0004）位描述

位	符号	描述	复位值
2:0	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
3	ABRT	从机中止。该位为 1 时，表示发生了从机中止。读取该寄存器时，该位清除。	0
4	MODF	模式错误。该位为 1 时，表示发生了模式错误。如果要清除该位，首先读取该寄存器，然后写 SPI0 控制寄存器。	0
5	ROVR	读溢出。该位为 1 时，表示发生了读溢出。读取该寄存器时，该位清除。	0
6	WCOL	写冲突。该位为 1 时，表示发生了写冲突。如果要清除该位，首先读取该寄存器，然后访问 SPI 数据寄存器。	0
7	SPIF	SPI 传送完成标志。该位为 1 时，表示一次 SPI 数据传送完成。在主机模式下，该位在传送的最后一个周期结束时置位。在从机模式下，该位在 SCK 的最后一个数据采样边沿置位。如果要清除该位，首先读取该寄存器，然后访问 SPI 数据寄存器。 **注：**这并不是 SPI 中断标志。该标志位于 SPINT 寄存器中。	0
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

39.6.3 SPI 数据寄存器

该双向数据寄存器为 SPI 提供发送和接收的数据。发送数据通过写该寄存器提供给 SPI。SPI 接收的数据可从该寄存器中读出。用作主机时，写该寄存器将启动 SPI 数据传送。启动数据传送时，或者 SPIF 状态位被置位，但是还没有读取 SPI 状态寄存器时，不能对该寄存器进行写操作。

表 871. SPI 数据寄存器 （DR - 地址 0x4010 0008）位描述

位	符号	描述	复位值
7:0	DATALOW	SPI 双向数据端口。	0x00
15:8	DATAHIGH	如果 SPCR 的位 2 为 1 且位 11:8 不是 1000，那么这些位的部分或全部含有其他的发送和接收位。当选择少于 16 位时，这些位中较高的位读为 0。	0x00
31:16	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

39.6.4 SPI 时钟计数器寄存器

该寄存器控制主机SCK的频率。寄存器显示了构成SPI时钟的SPI外围设备时钟周期个数。

在主机模式下，该寄存器的值必须是一个大于等于 8 的偶数。如果寄存器的值不符合上述条件，可能导致产生不可预测的动作。SPI SCK 速率可计算如下：PCLK_SPI / SPCCR 值。

在从机模式下，由主机提供的 SPI 时钟速率不能大于 SPI 外围设备时钟的 1/8。SPCCR 寄存器的内容是不相关的。

表 872. SPI 时钟计数器寄存器 （CCR - 地址 0x4010 0010）位描述

位	符号	描述	复位值
7:0	计数器	SPI0 时钟计数器设定。	0x00
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

39.6.5 SPI 测试控制寄存器

请注意，该寄存器中的位仅用于验证功能，在正常操作中不应使用该寄存器。

表 873. SPI 测试控制寄存器 （TCR - 地址 0x4010 0010）位描述

位	符号	描述	复位值
0	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
7:1	TEST	SPI 测试模式。该位为 0 时， SPI 正常操作。该位为 1 时， SCK 将始终打开，并且与主机模式选择和数据的可用性设置无关。	0
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

39.6.6 SPI 测试状态寄存器

注意：该寄存器中的位仅用于验证功能。在正常操作中不应使用该寄存器。

该寄存器是 SPI 状态寄存器的复制。这两个寄存器的不同之处在于：读该寄存器将不会启动清除这些状态位所必需的事件序列。如果相应位的写数据为 1，则写入该寄存器将会设置中断。

表 874. SPI 测试状态寄存器 （TSR - 地址 0x4010 0014）位描述

位	符号	描述	复位值
2:0	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
3	ABRT	从机中止。	0
4	MODF	模式错误。	0
5	ROVR	读溢出。	0
6	WCOL	写冲突。	0
7	SPIF	SPI 传送完成标志。	0
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

39.6.7 SPI 中断寄存器

该寄存器包含了 SPI0 接口的中断标志。

表 875. SPI 中断寄存器 （INT - 地址 0x4010 001C）位描述

位	符号	描述	复位值
0	SPIF	SPI 中断标志。由 SPI 接口置位以产生中断。向该位写入 1 可清除。 **注：**当 SPIE = 1 并且 SPIE 和 WCOL 至少有一个位为 1 时会设置该位。但是，只有当 SPI 中断位得到设置并且 SPI0 中断在 NVIC 中被使能时，SPI 中断才能由中断处理软件进行处理。	0
7:1	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
31:8	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

39.7 功能说明

39.7.1 SPI 数据传送

图 129 为时序图，显示了 SPI 端口可用的 4 种不同数据传输格式。该时序图描述的是一个 8 位数据传送。需要注意的是，该时序图按水平方向分成了 3 个部分。第一部分描述了 SCK 和 SSEL 信号；第二部分描述了 SPI 控制寄存器的时钟相位控制位 (CPHA) 为 0 时的 MOSI 和 MISO 信号；第三部分描述了 CPHA 变量为 1 时的 MOSI 和 MISO 信号。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

在时序图的第一部分需要注意两点：首先， SPI 是以 SPI 控制寄存器中的时钟极性控制位 (CPOL) 设置为 0 和 1 两种情况分别进行说明的；其次，需要注意 SSEL 信号的激活和未激活情况。当 CPHA = 0 时，SSEL 信号在两次数据传输之间将始终变为无效。但当 CPHA = 1 时，就无法保证这一点 （信号可以保持有效）。

图 129. SPI 数据传输格式 （CPHA = 0 和 CPHA = 1）

数据和时钟相位的关系在表 876 中进行了汇总。

表 876. SPI 数据和时钟相位的关系

CPOL 和 CPHA 设置	第一个数据位被驱动的时间	其他数据位被驱动的时间	数据的采样时间
CPOL = 0、 CPHA = 0	在第一个 SCK 上升沿之前	SCK 下降沿	SCK 上升沿
CPOL = 0、 CPHA = 1	第一个 SCK 上升沿	SCK 上升沿	SCK 下降沿
CPOL = 1、 CPHA = 0	在第一个 SCK 下降沿之前	SCK 上升沿	SCK 下降沿
CPOL = 1、 CPHA = 1	第一个 SCK 下降沿	SCK 下降沿	SCK 上升沿

传送的开始时间和结束时间的定义取决于器件是主机还是从机，以及 CPHA 变量的设置。

当器件是主机时，传送的开始时间由准备好发送一字节数据的主机来指明。此时，主机可激活时钟并开始传送。当传送的最后一个时钟周期结束时，传送结束。

当器件为从机并且 CPHA 被设置为 0 时，传送在 SSEL 信号被激活时开始，并且在 SSEL 变为无效时结束。当器件为从机并且 CPHA 被设置为 1 时，如果该从机被选中，则传送会从第一个时钟沿开始，并在数据采样的最后一个时钟沿结束。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

39.7.2 一般信息

SPI 端口有 5 个控制状态寄存器。这些寄存器在第 850 页上的章节 39.6 “寄存器描述”中有详细介绍。

SPI 控制寄存器 (S0SPCR) 包含一些可编程位用来控制 SPI 模块的功能。该寄存器的设置必须在开始传送指定的数据之前设定。

SPI 状态寄存器 (S0SPSR) 包含用于监控 SPI 接口状态的只读位，包括普通功能以及异常状况。该寄存器的主要用途是检测数据传送的完成，这可通过判断 S0SPINT 寄存器中的 SPI 中断标志 (SPIF) 来实现，该寄存器中的其他位用于指示异常状况。这些异常状况会在本小节的后续部分中进行描述。

SPI 数据寄存器 (S0SPDR) 用于提供发送和接收数据字节。串行数据实际的发送和接收是通过 SPI 模块逻辑中的内部移位寄存器来实现的。在发送时，数据会被写入 SPI 数据寄存器。数据寄存器和内部移位寄存器之间没有缓冲区，写数据寄存器会使数据直接进入内部移位寄存器，因此，数据只能在没有进行传送时才可写入该寄存器。读数据是带有缓冲区的，当传送结束时，接收到的数据会被转移到一个单字节的数据缓冲区，稍后会从中读取数据。读 SPI 数据寄存器将返回读数据缓冲区的值。

当 SPI 模块处于主机模式时， SPI 时钟计数器寄存器 (S0SPCCR) 用于控制时钟速率。当 SPI 模块为主机时，该寄存器必须在数据传送之前设定。而当 SPI 模块为从机时，该寄存器无效。

在使用之前，必须在 SPI 控制寄存器和 SPI 时钟计数器寄存器中设定主机 / 从机设置、时钟极性、时钟速率等 SPI 配置。

实施 SPI 所使用的 I/O 接口为标准的 CMOS I/O 接口。本设计中并没有提供开漏 SPI 选择。当器件被设置为从机时，其 I/O 接口只有在被有效的 SSEL 信号选中时才会被激活。

39.7.3 主机操作

SPI 首次用作主机之前，可遵循以下步骤对其进行设置。这通常是在程序初始化期间完成。

1. 设置 SPI 时钟计数器寄存器，得到所需的时钟速率。
2. 设置 SPI 控制寄存器，得到主机模式所需的设置。

下面的步骤描述了 SPI 模块被设置为主机时应如何处理数据传送。该处理假设上次的数据传送已经结束。

1. 或者，在启动传送之前验证 SPI 设置。
2. 将要发送的数据写入 SPI 数据寄存器，即启动 SPI 数据传送。
3. 等待 SPI 状态寄存器中的 SPIF 位被置位“1”。SPIF 位将会在 SPI 数据传送的最后一个周期之后置位。
4. 读 SPI 状态寄存器。
5. 从 SPI 数据寄存器中读取接收到的数据 （可选）。
6. 如果有更多的数据需要发送，则跳到第 2 步。

注：如果要将 SPIF 状态位清除，必须对 SPI 数据寄存器进行读 / 写操作。因此，如果没有对 SPI 数据寄存器进行读操作，则必须对该寄存器进行写操作来将 SPIF 状态位清除。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

39.7.4 从机操作

SPI 首次用作从机之前，可遵循以下步骤对其进行设置。这通常是在程序初始化期间完成。

1. 设置 SPI 控制寄存器，得到从机模式所需的设置。

下面的步骤描述了 SPI 模块被设置为从机时应如何处理数据传送。该处理假设上次的数据传送已经结束。要求驱动 SPI 逻辑的系统时钟速度至少是 SPI 的 8 倍。

1. 或者，在启动传送之前验证 SPI 设置。
2. 将要发送的数据写入 SPI 数据寄存器 （可选）。请注意，这只能在从机 SPI 传送没有进行时执行。
3. 等待 SPI 状态寄存器中的 SPIF 位被置位“1”。SPIF 位将在 SPI 数据传送的最后一个采样时钟沿后置位。
4. 读 SPI 状态寄存器。
5. 从 SPI 数据寄存器中读取接收到的数据 （可选）。
6. 如果有更多的数据需要传送，则跳到第 2 步。

注：如果要将 SPIF 状态位清除，必须对 SPI 数据寄存器进行读 / 写操作。因此，如果要将 SPIF 状态位清除，至少必须对 SPI 数据寄存器进行一次可选的读操作或写操作。

39.7.5 异常状况

读溢出

当 SPI 模块内部读缓冲区包含处理器尚未读取的数据时，又有一次新的传送已结束，就会发生读溢出。SPI 中断寄存器中的 SPIF 位被激活表示读缓冲区中包含了有效数据。当一次传送结束时，SPI 模块需要将接收到的数据移至读缓冲区中。如果 SPIF 位被激活（读缓冲区已满），则新接收的数据将会丢失，而 SPI 状态寄存器中的读溢出 (ROVR) 位将被激活。

写冲突

如前面所述，在 SPI 模块总线接口与内部移位寄存器之间没有写缓冲区。因此，当正在进行 SPI 数据传送时，不得向 SPI 数据寄存器写入数据。从启动传送到读取了 SPI 状态寄存器 （SPIF 状态已激活），这段时间内不能向 SPI 数据寄存器写入数据。如果在这段时间内写 SPI 数据寄存器，则写入的数据将会丢失，并且 SPI 状态寄存器中的写冲突 (WCOL) 位将被激活。

模式错误

当 SPI 模块为主机时，如果 SSEL 信号被激活，则表示有另一个主机将该器件选择为从机。这种情况称为模式错误。当检测到一个模式错误时，SPI 状态寄存器中的模式错误 (MODF) 位将被激活， SPI 信号驱动程序将被停用，而 SPI 模式将改为从机模式。

如果 SSEL 功能已分配给有关引脚功能选择寄存器中的相关引脚，则 SSEL 信号在 SPI 控制器为主机时必须始终处于无效状态。

从机中止

如果 SSEL 信号在传送结束前变为无效，则从机传送将被视为中止。此时，正在处理的发送或接收数据都将丢失，并且 SPI 状态寄存器中的从机中止 (ABRT) 位将被激活。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

40.1 本章导读

所有 LPC43xx 器件上均可使用 I^2S0/1 接口。

40.2 基本配置

I^2S 接口配置如下：

- 参见表 877 以了解时钟与电源控制。
- I2S0 通过 I2S0_RST 进行复位 （复位 52 号）。
- I2S1 通过 I2S1_RST 进行复位 （复位 53 号）。
- I2S0 中断连接至 NVIC 28 号时间槽。
- I2S1 中断连接至 NVIC 29 号时间槽。
- 要将 I2S 接收和发送线连接至 GPDMA，请使用 CREG 模块中的 DMAMUX 寄存器（参见表 41）并使能 DMA 通道配置寄存器 （第 19.6.20 节）中的 GPDMA 通道。
- 有关为音频 PLL 配置 I2S 时钟输入，请参见表 43。
- I2S0/1 MWS 信号 (I2S0_RX_MWS/I2S0_TX_MWS/IS1_RX_MWS/I2S1_TX_MWS) 可连接至定时器 3 或者通过 GIMA 连接至 SCT （参见表 138）。

表 877. I2S 的时钟与电源控制

	基础时钟	分支时钟	工作频率
输出至 I2S0 和 I2S1 寄存器接口及 I2S0/1 外设时钟。	BASE_APB1_CLK	CLK_APB1_I2S	高达 204 MHz

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

40.3 功能要点

I2S 总线为数字音频设备提供了标准通信接口。

I2S 总线规范定义了一种 3 线串行总线，包含一个数据选择信号、一个时钟选择信号和一个字选择信号。基本 I2S 连接包含一个主机 （它始终保持为主机）一个从机。 I2S 接口提供了独立的发送和接收通道，每个通道都可以作为主机或者从机运行。

- I2S 输入可以独立于 I2S 输出，在主机和从机模式下运行。
- I2S 输出可以独立于 I2S 输入，在主机和从机模式下运行。
- 能够处理 8 位、 16 位和 32 位字长。
- 支持单声道和立体声音频数据。
- 多功能时钟包括独立的发送和接收变速率发生器，能够在四线模式下使用单一的时钟输入或输出。
- 音频应用采样频率 (fs) 可介于（实际运用中）16 至 192 kHz（16、22.05、32、44.1、48、 96 或 192 kHz）之间。
- 发送通道和接收通道具有独立的主机时钟输出，支持高达512倍于I^2S采样频率的时钟。
- 主机模式下可配置字选择周期 （单独针对 I^2S 输入和 I^2S 输出）。
- 提供有两个 8 字 （32 字节） FIFO 数据缓冲区，一个用于发送，一个用于接收。
- 当缓存级别超过预编程界限时将生成中断要求。
- 两个 DMA 请求，由可编程的缓冲级别控制。它们连接至通用 DMA 块。
- 控制项目包括 I2S 输入和 I2S 输出的复位、停止和静音选项。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

40.4 简介

I2S 通过发送通道输出串行数据，通过接收通道接收串行数据。它们支持 Inter IC 音频格式的 8 位、16 位和 32 位音频数据，立体声道和单声道均可。通过 APB 寄存器设置来进行配置、数据访问和控制。 FIFO 用 8 个字长的深度来缓冲数据流。

I2S 接收和发送级在从机或者主机模式下都可独立运行。主机模式下，I2S 模块提供 SCK 和 WS 信号。从机模式下， SCK 和 WS 信号由外部主机提供。

- 在主机模式下，字选择由 9 位计数器内部产生。SCK 和 WS 信号的比例可在控制寄存器中进行编程。
- 在从机模式下，字选择是相关总线引脚的输入。
- 当 I2S 总线被激活时，字选择、接收时钟和发送时钟信号被总线主机持续发送，同时数据被发送器持续发送。
- 禁用 I2S 可通过分别停止或静音发送和接收的控制位来实现。
- 停止位将禁止发送通道或接收通道访问 FIFO，并将发送通道置于静音模式。
- 静音控制位将发送通道置于静音模式。在静音模式下，发送通道 FIFO 正常运行，但是输出被丢弃且被 0 替代。此位不影响接收通道，数据接收可运行正常。

40.4.1 I2S 连接方案

如果没有为 I2S1 时钟和数据线选择外部引脚， I2S1 将自动作为 I2S0 的从机。

MCLK 可由主机提供或被主机用来产生 I2S SCK。MCLK 也可由音频 PLL 通过 CREG 模块在内部产生 （参见表 40）。

40.4.2 GIMA 的 I2S 连接

I2S 模块在从机和主机模式下生成字选择 (WS) 信号以捕获或产生数据。 WS 信号被发送至 SCU 和 GIMA，在其中可被选作定时器 3 或者 SCT 的输入。 GIMA 输入进行 128 分频。

图 130. WS 信号连接

© NXP B.V. 2012. 保留所有权利。

40.5 引脚描述

表 878. 引脚描述

引脚功能	方向	描述
I2S0/1_RX_SCK	输入 / 输出	接收时钟。用于同步化接收通道上的数据传输的时钟信号。由主机驱动，从机接收。与 I2S 总线规范中的 SCK 信号对应。
I2S0/1_RX_WS	输入 / 输出	接收字选择。选择接收数据的通道。由主机驱动，从机接收。与 I2S 总线规范中的 WS 信号对应。 WS=0 表示数据由通道 1 接收 （左通道）。 WS=1 表示数据由通道 2 接收 （右通道）。
I2S0/1_RX_SDA	输入 / 输出	接收数据。串行数据，先接收 MSB。由发送器驱动，接收器读取。与 I2S 总线规范中的 SD 信号对应。
I2S0/1_RX_MCLK	输出	用于 I2S 接收功能的可选主机时钟。 I2S 模块处于从机模式时，该输出为 <tbd>。
I2S0/1_TX_SCK	输入 / 输出	发送时钟。用于同步化发送通道上的数据传输的时钟信号。由主机驱动，从机接收。与 I2S 总线规范中的 SCK 信号对应。
I2S0/1_TX_WS	输入 / 输出	发送字选择。选择发送数据的通道。由主机驱动，从机接收。与 I2S 总线规范中的 WS 信号对应。 WS=0 表示数据发送到通道 1 （左通道）。 WS=1 表示数据发送到通道 2 （右通道）。
I2S0/1_TX_SDA	输入 / 输出	发送数据。串行数据，先发送 MSB。由发送器驱动，接收器读取。与 I2S 总线规范中的 SD 信号对应。
IS0/1_TX_MCLK	输出	I2S 发送功能的可选主机时钟输出。 I2S 模块处于从机模式时，该输出为 <tbd>。

图 131. 简单 I2S 配置及总线时序

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

40.6 寄存器描述

表 879 列出了与 I2S 接口相关的寄存器及其功能概述。表格后是各个寄存器的详细信息。

复位值仅反映存储在所用位中的数据，不包括保留位的内容。

表 879. 寄存器简介：I2S0 （基址 0x400A 2000）

名称	访问类型	地址偏移	描述	复位值	参考
DAO	R/W	0x000	I2S 数字音频输出寄存器。包含 I2S 发送通道的控制位。	0x87E1	表 881
DAI	R/W	0x004	I2S 数字音频输入寄存器。包含 I2S 接收通道的控制位。	0x07E1	表 882
TXFIFO	WO	0x008	I2S 发送 FIFO。 8 x 32 位发送器 FIFO 的访问寄存器。	0	表 883
RXFIFO	RO	0x00C	I2S 接收 FIFO。 8 x 32 位接收器 FIFO 的访问寄存器。	0	表 884
STATE	RO	0x010	I2S 状态反馈寄存器。包含有关 I2S 接口的状态信息。	0x7	表 885
DMA1	R/W	0x014	I2S DMA 配置寄存器 1。包含 DMA 请求 1 的控制信息。	0	表 886
DMA2	R/W	0x018	I2S DMA 配置寄存器 2。包含 DMA 请求 2 的控制信息。	0	表 887
IRQ	R/W	0x01C	I2S 中断请求控制寄存器。包含控制 I2S 中断请求生成方式的位。	0	表 888
TXRATE	R/W	0x020	I2S 发送 MCLK 分频器。此寄存器通过指定产生 MCLK 所需的 PCLK 要除以的值，决定 I2S TX MCLK 速率。	0	表 889
RXRATE	R/W	0x024	I2S 接收 MCLK 分频器。此寄存器通过指定产生 MCLK 所需的 PCLK 要除以的值，决定 I2S RX MCLK 速率。	0	表 890
TXBITRATE	R/W	0x028	I2S 发送比特率分频器。此寄存器通过指定产生发送位时钟所需的 TX_MCLK 要除以的值，决定 I2S 发送位速率。	0	表 891
RXBITRATE	R/W	0x02C	I2S 接收比特率分频器。此寄存器通过指定产生发送位时钟所需的 RX_MCLK 要除以的值，决定 I2S 接收位速率。	0	表 892
TXMODE	R/W	0x030	I2S 发送模式控制。	0	表 893
RXMODE	R/W	0x034	I2S 接收模式控制。	0	表 894

表 880. 寄存器简介：I2S1 （基址 0x400A 3000）

名称	访问类型	地址偏移	描述	复位值	参考
DAO	R/W	0x000	I2S 数字音频输出寄存器。包含 I2S 发送通道的控制位。	0x87E1	表 881
DAI	R/W	0x004	I2S 数字音频输入寄存器。包含 I2S 接收通道的控制位。	0x07E1	表 882
TXFIFO	WO	0x008	I2S 发送 FIFO。 8 x 32 位发送器 FIFO 的访问寄存器。	0	表 883
RXFIFO	RO	0x00C	I2S 接收 FIFO。 8 x 32 位接收器 FIFO 的访问寄存器。	0	表 884
STATE	RO	0x010	I2S 状态反馈寄存器。包含有关 I2S 接口的状态信息。	0x7	表 885
DMA1	R/W	0x014	I2S DMA 配置寄存器 1。包含 DMA 请求 1 的控制信息。	0	表 886
DMA2	R/W	0x018	I2S DMA 配置寄存器 2。包含 DMA 请求 2 的控制信息。	0	表 887
IRQ	R/W	0x01C	I2S 中断请求控制寄存器。包含控制 I2S 中断请求生成方式的位。	0	表 888
TXRATE	R/W	0x020	I2S 发送 MCLK 分频器。此寄存器通过指定产生 MCLK 所需的 PCLK 要除以的值，决定 I2S TX MCLK 速率。	0	表 889
RXRATE	R/W	0x024	I2S 接收 MCLK 分频器。此寄存器通过指定产生 MCLK 所需的 PCLK 要除以的值，决定 I2S RX MCLK 速率。	0	表 890
TXBITRATE	R/W	0x028	I2S发送比特率分频器。此寄存器通过指定产生发送位时钟所需的 TX_MCLK 要除以的值，决定 I2S 发送位速率。	0	表 891

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 880. 寄存器简介：I2S1 （基址 0x400A 3000）*（续）*

名称	访问类型	地址偏移	描述	复位值	参考
RXBITRATE	R/W	0x02C	I2S接收比特率分频器。此寄存器通过指定产生发送位时钟所需的 RX_MCLK 要除以的值，决定 I2S 接收位速率。	0	表 892
TXMODE	R/W	0x030	I2S 发送模式控制。	0	表 893
RXMODE	R/W	0x034	I2S 接收模式控制。	0	表 894

40.6.1 I2S 数字音频输出寄存器

DAO 寄存器控制 I2S 发送通道的操作。 DAO 中各位的功能如表 881 所示。

表 881. I2S 数字音频输出寄存器 （DAO - 地址 0x400A 2000 (I2S0) 及 0x400A 3000 (I2S1)）位描述

位	符号	值	描述	复位值
1:0	WORDWIDTH		选择数据的字节数，如下所示：	01
		0x0	8 位数据	
		0x1	16 位数据	
		0x2	保留，不使用此设置	
		0x3	32 位数据	
2	MONO		为 1 时，数据为单声道格式。为 0 时，数据为立体声格式。	0
3	STOP		为 1 时，禁用对 FIFO 的访问，将发送通道置于静音模式。	0
4	RESET		为 1 时，异步复位发送通道和 FIFO。	0
5	WS_SEL		为 0 时，接口处于主机模式。为 1 时，接口处于从机模式。有关此位与 TXMODE 有用组合的概述，请参见第 40.7.2 节。	1
14:6	WS_HALFPERIOD		字选择半周期减 1，即 WS 64clk 周期 -> ws_halfperiod = 31。	0x1F
15	MUTE		为 1 时，发送通道仅发送 0。	1
31:16	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

40.6.2 I2S 数字音频输入寄存器

DAI 寄存器控制 I2S 接收通道的操作。 DAI 中各位的功能如表 882 所示。

表 882. I2S 数字音频输入寄存器 （DAI - 地址 0x400A 2004 (I2S0) 及 0x400A 3004 (I2S1)）位描述

位	符号	值	描述	复位值
1:0	WORDWIDTH		选择数据的字节数，如下所示：	01
		0x0	8 位数据	
		0x1	16 位数据	
		0x2	保留，不使用此设置	
		0x3	32 位数据	
2	MONO		为 1 时，数据为单声道格式。为 0 时，数据为立体声格式。	0
3	STOP		为 1 时，禁用对 FIFO 的访问，将发送通道置于静音模式。	0
4	RESET		为 1 时，异步复位发送通道和 FIFO。	0
5	WS_SEL		为0时，接口处于主机模式。为1时，接口处于从机模式。有关此位与RXMODE 有用组合的概述，请参见第 40.7.2 节。	1
14:6	WS_HALFPERIOD		字选择半周期减 1，即 WS 64clk 周期 -> ws_halfperiod = 31。	0x1F
31:15	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

40.6.3 I2S 发送 FIFO 寄存器

TXFIFO 寄存器提供对发送 FIFO 的访问。

表 883. 发送 FIFO 寄存器 （TXFIFO - 地址 0x400A 2008 (I2S0) 及 0x400A 3008 (I2S1)）位描述

位	符号	描述	复位值
31:0	I2STXFIFO	8 x 32 位发送 FIFO。	0

40.6.4 接收 FIFO 寄存器

I2SRXFIFO 寄存器提供对接收 FIFO 的访问。

表 884. I2S 接收 FIFO 寄存器 （RXFIFO - 地址 0x400A 200C (I2S0) 及 0x400A 300C (I2S1)）位描述

位	符号	描述	复位值
31:0	I2SRXFIFO	8 x 32 位发送 FIFO。	0

40.6.5 I2S 状态反馈寄存器

STATE 寄存器提供有关 I2S 接口的状态信息。

表 885. I2S 状态反馈寄存器 （STATE - 地址 0x400A 2010 (I2S0) 及 0x400A 3010 (I2S1)）位描述

位	符号	描述	复位值
0	IRQ	此位反映是否存在接收中断或发送中断。通过将当前 FIFO 电平与 IRQ 寄存器中的 rx_depth_irq 和 tx_depth_irq 字段进行比较来确定。	1
1	DMAREQ1	此位反映是否存在接收或发送 DMA 请求 1。通过将当前 FIFO 电平与 DMA1 寄存器中的 rx_depth_dma1 和 tx_depth_dma1 字段进行比较来确定。	1
2	DMAREQ2	此位反映是否存在接收或发送 DMA 请求 2。确定方法为将当前 FIFO 电平与 DMA2 寄存器中的 rx_depth_dma2 和 tx_depth_dma2 字段进行比较。	1
7:3	-	保留。	0
11:8	RX_LEVEL	反映接收 FIFO 的当前电平。	0
15:12	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
19:16	TX_LEVEL	反映发送 FIFO 的当前电平。	0
31:20	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

40.6.6 I2S DMA 配置寄存器 1

DMA1 寄存器控制 DMA 请求 1 的操作。DMA1 中位的功能如表 886 所示。关于 DMA 操作的详细信息，请参阅第 19 章。

此寄存器为 I²S 接收和发送通道使能 DMA 并设置 FIFO 电平。

注：FIFO 包含八个 32 位双字。因此，如果 I²S 控制器配置为 32 位模式 （参见表 881 和表 882），则允许的最大 FIFO 电平为 4。

表 886. I2S DMA 配置寄存器 1 （DMA1 - 地址 0x400A 2014 (I2S0) 及 0x400A 3014 (I2S1)）位描述

位	符号	描述	复位值
0	RX_DMA1_ENABLE	为 1 时，为 I2S 接收使能 DMA1。	0
1	TX_DMA1_ENABLE	为 1 时，为 I2S 发送使能 DMA1。	0
7:2	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	0
11:8	RX_DEPTH_DMA1	设置启动 DMA1 上接收 DMA 请求的 FIFO 电平。	0
15:12	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
19:16	TX_DEPTH_DMA1	设置启动 DMA1 上发送 DMA 请求的 FIFO 电平。	0
31:20	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

40.6.7 I2S DMA 配置寄存器 2

DMA2 寄存器控制 DMA 请求 2 的操作。 DMA2 中位的功能如表 881 所示。

此寄存器为 I²S 接收和发送通道使能 DMA 并设置 FIFO 电平。

注：FIFO 包含八个 32 位双字。因此，如果 I²S 控制器配置为 32 位模式 （参见表 881 和表 882），则允许的最大 FIFO 电平为 4。

表 887. I2S DMA 配置寄存器 2 （DMA2 - 地址 0x400A 2018 (I2S0) 及 0x400A 3018 (I2S1)）位描述

位	符号	描述	复位值
0	RX_DMA2_ENABLE	为 1 时，为 I2S 接收使能 DMA1。	0
1	TX_DMA2_ENABLE	为 1 时，为 I2S 发送使能 DMA1。	0
7:2	-	保留。	0
11:8	RX_DEPTH_DMA2	设置启动 DMA2 上接收 DMA 请求的 FIFO 电平。	0
15:12	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
19:16	TX_DEPTH_DMA2	设置启动 DMA2 上发送 DMA 请求的 FIFO 电平。	0
31:20	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

40.6.8 I2S 中断请求控制寄存器

IRQ 寄存器控制 I2S 中断请求的操作。 IRQ 中各位的功能如表 881 所示。

表 888. I2S 中断请求控制寄存器 （IRQ - 地址 0x400A 201C (I2S0) 及 0x400A 301C (I2S1)）位描述

位	符号	描述	复位值
0	RX_IRQ_ENABLE	为 1 时，使能 I2S 接收中断。	0
1	TX_IRQ_ENABLE	为 1 时，使能 I2S 发送中断。	0
7:2	-	保留。	0
11:8	RX_DEPTH_IRQ	设置要创建 irq 请求的 FIFO 电平。	0
15:12	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
19:16	TX_DEPTH_IRQ	设置要创建 irq 请求的 FIFO 电平。	0
31:20	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

40.6.9 I2S 发送时钟速率寄存器

I2S 发送器的 MCLK 频率由 TXRATE 寄存器中的值确定。要求的 TXRATE 设置取决于所需的音频采样率、所用的格式 （立体声 / 单声道）和数据大小。

使用变速率发生器生成发送器 MCLK 频率，对 PCLK = CLK_APB1_I2S 的频率进行分频。必须选择分子 (X) 和分母 (Y) 的值以产生发送器 MCLK 所需频率两倍的频率，它必须是发送器位时钟频率的整数倍。在选择设置时，用户应注意变速率发生器的某些事项，详细讨论见第 40.6.9.1 节。变速率发生器的方程为：

I2S_TX_MCLK = PCLK * (X/Y) /2

注：如果 X 或 Y 的值为 0，则不会产生时钟。另外， Y 的值必须大于或等于 X。

表 889. I2S 发送时钟速率寄存器 （TXRATE - 地址 0x400A 2020 (I2S0) 及 0x400A 3020 (I2S1)）位描述

位	符号	描述	复位值
7:0	Y_DIVIDER	I2S 发送 MCLK 频率分母。此值用于分频 PCLK，以生成发送 MCLK。八位分数分频支持范围广泛的可能性。值为 0 将停止时钟。	0
15:8	X_DIVIDER	I2S 发送 MCLK 频率分子。此值用于乘以 PCLK，以生成发送 MCLK。值为 0 将停止时钟。 八位分数分频支持范围广泛的可能性。注：生成的比值 X/Y 除以 2。	0
31:16	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

40.6.9.1 变速率发生器的说明

变速率发生器在某些除法设置中会产生输出抖动。这是因为变速率发生器是一个完全的数字功能，因此输出时钟转换与源时钟同步，而理论上完美的变速率时钟会具有与源时钟无关的边沿。因此，输出抖动将不会大于在两个连续时钟沿之间加上或减去一个源时钟。

例如，如果 X = 0x07、 Y = 0x11，则变速率发生器对每 17 （十六进制 11）个输入时钟将输出 7 个时钟，尽可能地均匀分布。在本例中，不能以一种完美均匀的方式来分布输出时钟，因此某些时钟将大于其他时钟。输出除以 2 以便解决此问题，也有助于缓解抖动。频率精确平均至 (7/17) / 2，但是某些时钟与邻近时钟的长度稍有不同。通过选择一个分数，使 X 能均匀地被 Y 除，如 2/4、 2/6、 3/9、 1/N 等，就有可能完全避免抖动。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

40.6.10 I2S 接收时钟速率寄存器

I2S 接收器的 MCLK 频率由 RXRATE 寄存器中的值确定。要求的 RXRATE 设置取决于外设时钟频率 (PCLK_I2S = CLK_APB1_I2S) 和所需的 MCLK 频率 （如 256 fs）。

接收器 MCLK 频率使用变速率发生器生成，对 PCLK_I2S 的频率进行分频。必须选择分子 (X) 和分母 (Y) 的值以产生接收器 MCLK 所需频率两倍的频率，它必须是接收器位时钟频率的整数倍。在选择设置时，用户应注意变速率发生器的某些事项，详细讨论见第 40.6.9.1 节。变速率发生器的方程为：

I2S_RX_MCLK = PCLK_I2S * (X/Y) /2

注：如果 X 或 Y 的值为 0，则不会产生时钟。另外， Y 的值必须大于或等于 X。

表 890. I2S 接收时钟速率寄存器 （RXRATE - 地址 0x400A 2024 (I2S0) 及 0x400A 3024 (I2S1)）位描述

位	符号	描述	复位值
7:0	Y_DIVIDER	I2S 接收 MCLK 频率分母。此值用于分频 PCLK，以生成接收 MCLK。八位分数分频支持范围广泛的可能性。值为 0 将停止时钟。	0
15:8	X_DIVIDER	I2S 接收 MCLK 频率分子。此值用于乘以 PCLK，以生成接收 MCLK。值为 0 将停止时钟。八位分数分频支持范围广泛的可能性。注：生成的比值 X/Y 除以 2。	0
31:16	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

40.6.11 I2S 发送时钟位速率寄存器

I2S 发送器的比特率由 TXBITRATE 寄存器中的值确定。值取决于所需的音频采样率、数据大小和所用的格式 （立体声 / 单声道）。例如：16 位立体声数据的 48 kHz 采样率要求比特率为 48 000 x 16 x 2 = 1.536 MHz。

表 891. I2S 发送时钟速率寄存器 （TXBITRATE - 地址 0x400A 2028 (I2S0) 及 0x400A 3028 (I2S1)）位描述

位	符号	描述	复位值
5:0	TX_BITRATE	I2S 发送比特率。使用此值加一去除 TX_MCLK，以生成发送比特率。	0
31:6	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

40.6.12 I2S 接收时钟位速率寄存器

I2S 接收器的比特率由 RXBITRATE 寄存器中的值确定。值取决于音频采样率、数据大小和所用的格式。计算方法与 TXBITRATE 相同。

表 892. I2S 接收时钟速率寄存器 （RXBITRATE - 地址 0x400A 202C (I2S0) 及 0x400A 302C (I2S1)）位描述

位	符号	描述	复位值
5:0	RX_BITRATE	I2S 接收比特率。该值加 1 用于分频 RX_MCLK 以产生接收位时钟。	0
31:6	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

40.6.13 I2S 发送模式控制寄存器

发送模式控制寄存器包含对发送时钟源、使能 4 引脚模式（I2S 发送和接收模块之间共享 SCK 和 WS 信号）和如何使用 MCLK 的附加控制。有关有用模式组合的概述，请参见第 40.7.2 节。

表 893. I2S 发送模式控制寄存器 （TXMODE - 地址 0x400A 2030 (I2S0) 及 0x400A 3030 (I2S1)）位描述

位	符号	值	描述	复位值
1:0	TXCLKSEL		发送位时钟分频器的时钟源选择。	0
		0x0	选择 TX 小数频率分频器时钟输出作为源。	
		0x1	保留	
		0x2	选择 RX_MCLK 信号作为 TX_MCLK 时钟源	
		0x3	保留	
2	TX4PIN		发送 4 引脚模式选择 （I2S 发送和接收模块之间共享 SCK 和 WS 信号）。为 1 时，使能 4 引脚模式。	0
3	TXMCENA		为 TX_MCLK 输出使能。为 0 时，TX_MCLK 的输出不使能。为 1 时，TX_MCLK 的输出使能。	0
31:4	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

40.6.14 I2S 接收模式控制寄存器

接收模式控制寄存器包含对接收时钟源、使能 4 引脚模式（I2S 发送和接收模块之间共享 SCK 和 WS 信号）和如何使用 MCLK 的附加控制。有关有用模式组合的概述，请参见第 40.7.2 节。

表 894. I2S 接收模式控制寄存器 （RXMODE - 地址 0x400A 2034 (I2S0) 及 0x400A 3034 (I2S1)）位描述

位	符号	值	描述	复位值
1:0	RXCLKSEL		接收位时钟分频器的时钟源选择。	0
		0x0	选择 RX 小数频率分频器时钟输出作为源。	
		0x1	保留	
		0x2	选择 TX_MCLK 信号作为 RX_MCLK 时钟源	
		0x3	保留	
2	RX4PIN		接收 4 引脚模式选择 （I2S 发送和接收模块之间共享 SCK 和 WS 信号）。为 1 时，使能 4 引脚模式。	0
3	RXMCENA		为 RX_MCLK 输出使能。为 0 时，RX_MCLK 的输出不使能。为 1 时，RX_MCLK 的输出使能。	0
31:4	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

40.7 功能描述

40.7.1 I²S 的发送和接收接口

I2S 接口可发送和接收 8 位、16 位或 32 位立体声或单声道音频信息。I2S 实施的一些细节如下：

- 发送FIFO包含的数据不足时，发送通道将重复发送最后一个数据，直至有新数据可用。当微处理器或 DMA 在某些时候无法足够快的提供新数据时会出现这种情况。由于新数据中的此延迟，因此需要通过连续发送上一个样本来填充空隙。数据将不会被静音，因为这将在声音中产生明显的不良影响。
- 静音为真时，数据值 0 被发送。
- 单声道为假时，两个连续的数据字分别为左数据和右数据。
- 数据字长度由配置寄存器中的 wordwidth 值确定。对于接收通道和发送通道，有单独的 wordwidth 值。
  - 0: 字可视为包含四个 8 位数据字。
  - 1: 字可视为包含两个 16 位数据字。
  - 3: 字可视为包含一个 32 位数据字。
- 发送通道和接收通道只处理 32 位对齐的字，数据块必须剪切或扩展到 32 位的倍数。

在数据宽度或模式之间切换时，必须通过控制寄存器中的复位位复位 I2S，以确保正确同步。建议同时设置停止位，直到发送 FIFO 中写入足够的数据。请注意，停止时数据输出为静音。

FIFO 的所有数据访问都是 32 位。图 146 所示为可能的数据序列。

数据在 WS 下降沿后从发送 FIFO 读取，并在 WS 的上升沿后被传输到发送时钟域。在下一 WS 下降沿，左声道数据将被加载入移位寄存器然后被发送，而右声道数据将在下一 WS 上升沿被加载和发送。

接收通道将在 WS 改变后开始接收数据。字选择变为低电平时，期望收到的数据为左声道数据，WS 为高电平时，期望收到的数据为右声道数据。位计数器达到 wordwidth 所设定的限值时，将停止接收。下次改变 WS 时，收到的数据将存储在相应的暂存寄存器中。整个数据可用时，会将其写入接收 FIFO 中。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

40.7.2 I²S 操作模式

I2S 接口的时钟和 WS 使用是可配置的。除了可为发送器和接收器独立配置的主机和从机模式外，还可使用几种不同的时钟源，包括在发送器和接收器之间共享时钟和 / 或 WS 的变化。最后一个选项可使用具有更少引脚 （通常四个）的 I2S。

许多配置不一定是有用的，以下各表和各图给出了最可能有用的配置。

表 895. I2S 发送模式

DAO 位 5	TXMODE 位 [3:0]	描述
0	0 0 0 0	典型的发送器主机模式。参见图 132。 I2S 发送功能按主机工作。 发送时钟源 (TX_MCLK) 使用小数分频器从 PCLK 派生。 使用的 WS 是内部生成的 TX_WS。 不为输出使能 TX_MCLK 引脚。
0	0 0 1 0	共享接收器参考时钟 (RX_MCLK) 的发送器主机模式。参见图 134。 I2S 发送功能按主机工作。 发送时钟源是 RX_MCLK。 使用的 WS 是内部生成的 TX_WS。 不为输出使能 TX_MCLK 引脚。
0	0 1 0 0	共享接收器位时钟和 WS 的 4 线发送器主机模式 （4 引脚模式）。参见图 135。 I2S 发送功能按主机工作。 发送时钟源是 RX 位时钟。 使用的 WS 是内部生成的 RX_WS。 不为输出使能 TX_MCLK 引脚。
0	1 0 0 0	带 TX_MCLK 输出的发送器主机模式。参见图 132。 I2S 发送功能按主机工作。 发送时钟源是小数频率分频器。 使用的 WS 是内部生成的 TX_WS。 TX_MCLK 引脚对输出使能。
1	0 0 0 0	典型的发送器从机模式。参见图 136。 I2S 发送功能按从机工作。 发送时钟源由外部主机在 TX_SCK 引脚提供。 使用的 WS 为 TX_WS 引脚。
1	0 0 1 0	共享接收器参考时钟 (RX_MCLK) 的发送器从机模式。参见图 137。 I2S 发送功能按从机工作。 发送时钟源是 RX_SCK。 使用的 WS 为 TX_WS 引脚。
1	0 1 0 0	共享接收器位时钟和 WS 的 4 线发送器从机模式。参见图 138。 I2S 发送功能按从机工作。 发送时钟源是 RX 位时钟。 使用的 WS 为 RX_WS ref。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 132. 典型的发送器主机模式 （有或没有 MCLK 输出）

图 133. 典型发送器主机模式，带或不带 MCLK 输出，使用音频 PLL

图 134. 发送器主机模式共享接收器的参考时钟

图 135. 4 线发送器主机模式共享接收器的位时钟和 WS

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 136. 典型的发送器从机模式

图 137. 发送器从机模式共享接收器的参考时钟

图 138. 4 线发送器从机模式共享接收器的位时钟和 WS

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 896. I2S 接收模式

DAI 位 5	RXMODE 位 [3:0]	描述
0	0 0 0 0	典型的接收器主机模式。参见图 139。 I2S 接收功能按主机工作。 接收时钟源 (RX_MCLK) 使用小数分频器从 PCLK 派生。 使用的 WS 是内部生成的 RX_WS。 不为输出使能 RX_MCLK 引脚。
0	0 0 1 0	共享发送器参考时钟 (TX_MCLK) 的接收器主机模式。参见图 141。 I2S 接收功能按主机工作。 接收时钟源是 TX_MCLK。 使用的 WS 是内部生成的 RX_WS。 不为输出使能 RX_MCLK 引脚。
0	0 1 0 0	共享发送器位时钟 (TX_SCK) 和 WS 的 4 线接收器主机模式。参见图 142。 I2S 接收功能按主机工作。 接收时钟源是 TX 位时钟 (TX_SCK)。 使用的 WS 是内部生成的 TX_WS。 不为输出使能 RX_MCLK 引脚。
0	1 0 0 0	带 RX_MCLK 输出的接收器主机模式。参见图 139。 I2S 接收功能按主机工作。 接收时钟源 (RX_MCLK) 使用小数分频器从 PCLK 派生。 使用的 WS 是内部生成的 RX_WS。 为输出使能 RX_MCLK 引脚。
1	0 0 0 0	典型的接收器从机模式。参见图 143。 I2S 接收功能按从机工作。 接收时钟源是 RX_SCK 引脚。 使用的 WS 是 RX_WS 引脚。
1	0 0 1 0	共享发送器参考时钟 (TX_MCLK) 的接收器从机模式。参见图 144。 I2S 接收功能按从机工作。 接收时钟源是 TX_MCLK。 使用的 WS 是 RX_WS 引脚。
1	0 1 0 0	共享发送器位时钟 (TX_SCK) 和 WS 的 4 线接收器从机模式。参见图 145。 I2S 接收功能按从机工作。 接收时钟源是 TX 位时钟 (TX_SCK)。 使用的 WS 是 TX_WS ref。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 139. 典型的接收器主机模式 （有或没有 MCLK 输出）

图 140. 典型接收器主机模式，带或不带 MCLK 输出，使用音频 PLL

图 141. 接收器主机模式共享发送器参考时钟

图 142. 4 线接收器主机模式共享发送器的位时钟和 WS

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 143. 典型的接收器从机模式

图 144. 接收器从机模式共享发送器的参考时钟

图 145. 4 线接收器从机模式共享发送器的位时钟和 WS

40.7.3 FIFO 控制器

发送和接收数据的处理是通过 FIFO 控制器完成的，该控制器可生成两个 DMA 请求和一个中断请求。控制器由一组比较器组成，将 FIFO 电平与寄存器中包含的深度设置进行比较。可在 APB 状态寄存器中看到电平比较器的当前状态。

表 897. FIFO 级别比较的条件

电平比较	环境
dmareq_tx_1	tx_depth_dma1 >= tx_level
dmareq_rx_1	rx_depth_dma1 <= rx_level
dmareq_tx_2	tx_depth_dma2 >= tx_level
dmareq_rx_2	rx_depth_dma2 <= rx_level
irq_tx	tx_depth_irq >= tx_level
irq_rx	rx_depth_irq <= rx_level

系统信号在电平检测为真且使能时出现。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 898. DMA 和中断请求产生

系统信号	环境
irq	(irq_rx & rx_irq_enable) | (irq_tx & tx_irq_enable)
dmareq[0]	(dmareq_tx_1 & tx_dma1_enable) | (dmareq_rx_1 & rx_dma1_enable)
dmareq[1]	(dmareq_tx_2 & tx_dma2_enable) | (dmareq_rx_2 & rx_dma2_enable)

表 899. STATE 寄存器中的状态反馈

状态反馈	状态
irq	irq_rx | irq_tx
dmareq1	(dmareq_tx_1 | dmareq_rx_1)
dmareq2	(dmareq_rx_2 | dmareq_tx_2)

图 146. 不同 I²S 模式的 FIFO 内容

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

41.1 本章导读

所有 LPC43xx 器件上都包含 C_CAN0/1 控制器。

41.2 基本配置

C_CAN 的配置如下：

- 参见表 900 以了解时钟与电源控制。
- C_CAN0 通过 CAN0_RST 进行复位 （复位 55 号）。
- C_CAN1 通过 CAN1_RST 进行复位 （复位 56 号）。
- ORed C_CAN0 和 C_CAN1 中断连接至事件路由器 12 号时间槽。
- C_CAN0 中断连接至 NVIC 51 号中断。
- C_CAN1 中断连接至 NVIC 43 号中断。

表 900. C_CAN 的时钟与电源控制

	基础时钟	分支时钟	工作频率
C_CAN0 寄存器接口的时钟和 C_CAN0 外围设备时钟 (PCLK)。	BASE_APB3_CLK	CLK_APB3_CAN0	高达 204 MHz
C_CAN1 寄存器接口的时钟和 C_CAN1 外围设备时钟 (PCLK)。	BASE_APB1_CLK	CLK_APB1_CAN1	高达 204 MHz

注：C_CAN0 和 C_CAN1 接口的时钟可分别进行设置。

41.3 功能要点

- 符合协议版本 2.0 部分 A 和 B。
- 支持最大 1 Mbit/s 的比特率。
- 支持 32 个消息对象。
- 每个消息对象都有自己的标识符掩码。
- 提供可编程 FIFO 模式 （消息对象的级联）。
- 提供可掩蔽中断。
- 对于时间启动的 CAN 应用，支持 “禁用自动重传 (DAR)” 模式。
- 为自检操作提供可编程环回模式。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

41.4 简介

控制器局域网络(CAN)定义为串行数据通信的高性能通信协议。C_CAN控制器可根据CAN规范版本 2.0B，完全执行 CAN 协议。C_CAN 控制器支持具有高度安全性的分布式实时控制，可构建功能强大、低成本多路布线的局域网络。

CAN 控制器由 CAN 核、消息 RAM、消息处理器、控制寄存器和 APB 接口组成。

对于 CAN 网络上的通信，配置了单独的消息对象。用于验收过滤已收到消息的消息对象和标识符掩码都存储在消息 RAM 中。

所有与处理消息有关的功能都在消息处理器中完成。这些功能为验收过滤、在 CAN 内核和消息 RAM 之间传输消息、处理发送请求以及生成模块中断。

CAN 控制器的寄存器组可通过 APB 总线由外部 CPU 直接访问。这些寄存器用于控制 / 配置 CAN 内核和消息处理器，并访问消息 RAM。

图 147. C_CAN 功能框图

41.5 引脚描述

表 901. C_CAN 引脚描述

函数名称	方向	描述
CAN0/1_RD	I	C_CAN 接收输入
CAN0/1_TD	O	C_CAN 发送输出

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

41.6 寄存器描述

复位时的寄存器值

硬件复位后，寄存器会保存表 902 中所述的值。此外，总线关闭状态还会被复位，并且输出 TD0、1 将被设置成隐性（高电平）。CAN 控制寄存器中的值 0x0001 (INIT= ‘1’) 使能软件初始化。 CAN 控制器在 CPU 将 INIT 复位为 0 前，不与 CAN 总线进行通信。

存储在消息 RAM 中的数据不受硬件复位的影响。加电后，消息 RAM 的内容未定义。

读取 / 写入操作的时序

注：读取任何 CAN 寄存器都需要从同一位置进行**两次**连续的读操作。只有**第二次**读操作所获取的数据才有效。

对 C_CAN 寄存器的连续读操作必须至少间隔 (CLKDIVVAL × 2 + 2) × PCLK，其中 CLKDIVVAL 为 CAN 时钟分频器值， PCLK 为外围时钟。

对C_CAN寄存器的连续写入操作必须至少间隔(CLKDIVVAL × 2) × PCLK，其中CLKDIVVAL 为 CAN 时钟分频器值， PCLK 为外设时钟。

表 902. 寄存器简介：C_CAN0 （基址 0x400E 2000）

名称	访问类型	地址偏移	描述	复位值	参考
CNTL	R/W	0x000	CAN 控制寄存器	0x0001	表 904
STAT	R/W	0x004	状态寄存器	0x0000	表 905
EC	RO	0x008	错误计数器寄存器	0x0000	表 906
BT		0x00C	位时序寄存器	0x2301	表 907
INT	RO	0x010	中断寄存器	0x0000	表 908
TEST	R/W	0x014	测试寄存器	-	表 909
BRPE	R/W	0x018	波特率预分频器扩展寄存器	0x0000	表 910
-	-	0x01C	保留	-	
IF1_CMDREQ	R/W	0x020	消息接口 1 命令请求	0x0001	表 913
IF1_CMDMSK_W	R/W	0x024	消息接口 1 命令掩码 （写方向）	0x0000	表 915
IF1_CMDMSK_R	R/W	0x024	消息接口 1 命令掩码 （读方向）	0x0000	表 917
IF1_MSK1	R/W	0x028	消息接口 1 掩码 1	0xFFFF	表 919
IF1_MSK2	R/W	0x02C	消息接口 1 掩码 2	0xFFFF	表 921
IF1_ARB1	R/W	0x030	消息接口 1 仲裁 1	0x0000	表 923
IF1_ARB2	R/W	0x034	消息接口 1 仲裁 2	0x0000	表 925
IF1_MCTRL	R/W	0x038	消息接口 1 消息控制	0x0000	表 927
IF1_DA1	R/W	0x03C	消息接口 1 数据 A1	0x0000	表 929
IF1_DA2	R/W	0x040	消息接口 1 数据 A2	0x0000	表 930
IF1_DB1	R/W	0x044	消息接口 1 数据 B1	0x0000	表 933
IF1_DB2	R/W	0x048	消息接口 1 数据 B2	0x0000	表 935
-		0x04C - 0x07C	保留	-	
IF2_CMDREQ	R/W	0x080	消息接口 2 命令请求	0x0001	表 914
IF2_CMDMSK_W	R/W	0x084	消息接口 2 命令掩码 （写方向）	0x0000	表 916
IF2_CMDMSK_R	R/W	0x084	消息接口 2 命令掩码 （读方向）	0x0000	表 918

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 902. 寄存器简介：C_CAN0 （基址 0x400E 2000）*（续）*

名称	访问类型	地址偏移	描述	复位值	参考
IF2_MSK1	R/W	0x088	消息接口 2 掩码 1	0xFFFF	表 920
IF2_MSK2	R/W	0x08C	消息接口 2 掩码 2	0xFFFF	表 922
IF2_ARB1	R/W	0x090	消息接口 2 仲裁 1	0x0000	表 924
IF2_ARB2	R/W	0x094	消息接口 2 仲裁 2	0x0000	表 926
IF2_MCTRL	R/W	0x098	消息接口 2 消息控制	0x0000	表 928
IF2_DA1	R/W	0x09C	消息接口 2 数据 A1	0x0000	表 930
IF2_DA2	R/W	0x0A0	消息接口 2 数据 A2	0x0000	表 931
IF2_DB1	R/W	0x0A4	消息接口 2 数据 B1	0x0000	表 934
IF2_DB2	R/W	0x0A8	消息接口 2 数据 B2	0x0000	表 936
-	-	0x0AC - 0x0FC			
TXREQ1	RO	0x100	发送请求 1	0x0000	表 937
TXREQ2	RO	0x104	发送请求 2	0x0000	表 938
-	-	0x108 - 0x11C	保留	-	
ND1	RO	0x120	新数据 1	0x0000	表 939
ND2	RO	0x124	新数据 2	0x0000	表 940
-	-	0x128 - 0x13C	保留	-	
IR1	RO	0x140	中断挂起 1	0x0000	表 941
IR2	RO	0x144	中断挂起 2	0x0000	表 942
-	-	0x148 - 0x15C	保留	-	
MSGV1	RO	0x160	消息有效 1	0x0000	表 943
MSGV2	RO	0x164	消息有效 2	0x0000	表 944
-	-	0x168 - 0x17C	保留	-	
CLKDIV	R/W	0x180	CAN 时钟分频器寄存器	0x0001	表 945

表 903. 寄存器简介：C_CAN1 （基址 0x400A 4000）

名称	访问类型	地址偏移	描述	复位值	参考
CNTL	R/W	0x000	CAN 控制	0x0001	表 904
STAT	R/W	0x004	状态寄存器	0x0000	表 905
EC	RO	0x008	错误计数器	0x0000	表 906
BT		0x00C	位时序寄存器	0x2301	表 907
INT	RO	0x010	中断寄存器	0x0000	表 908
TEST	R/W	0x014	测试寄存器	-	表 909
BRPE	R/W	0x018	波特率预分频器扩展寄存器	0x0000	表 910
-	-	0x01C	保留	-	
IF1_CMDREQ	R/W	0x020	消息接口 1 命令请求	0x0001	表 913
IF1_CMDMSK_W	R/W	0x024	消息接口 1 命令掩码 （写方向）	0x0000	表 915
IF1_CMDMSK_R	R/W	0x024	消息接口 1 命令掩码 （读方向）	0x0000	表 917

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 903. 寄存器简介：C_CAN1 （基址 0x400A 4000）*（续）*

名称	访问类型	地址偏移	描述	复位值	参考
IF1_MSK1	R/W	0x028	消息接口 1 掩码 1	0xFFFF	表 919
IF1_MSK2	R/W	0x02C	消息接口 1 掩码 2	0xFFFF	表 921
IF1_ARB1	R/W	0x030	消息接口 1 仲裁 1	0x0000	表 923
IF1_ARB2	R/W	0x034	消息接口 1 仲裁 2	0x0000	表 925
IF1_MCTRL	R/W	0x038	消息接口 1 消息控制	0x0000	表 927
IF1_DA1	R/W	0x03C	消息接口 1 数据 A1	0x0000	表 929
IF1_DA2	R/W	0x040	消息接口 1 数据 A2	0x0000	表 930
IF1_DB1	R/W	0x044	消息接口 1 数据 B1	0x0000	表 933
IF1_DB2	R/W	0x048	消息接口 1 数据 B2	0x0000	表 935
-		0x04C - 0x07C	保留	-	
IF2_CMDREQ	R/W	0x080	消息接口 2 命令请求	0x0001	表 914
IF2_CMDMSK_W	R/W	0x084	消息接口 2 命令掩码 （写方向）	0x0000	表 916
IF2_CMDMSK_R	R/W	0x084	消息接口 2 命令掩码 （读方向）	0x0000	表 918
IF2_MSK1	R/W	0x088	消息接口 2 掩码 1	0xFFFF	表 920
IF2_MSK2	R/W	0x08C	消息接口 2 掩码 2	0xFFFF	表 922
IF2_ARB1	R/W	0x090	消息接口 2 仲裁 1	0x0000	表 924
IF2_ARB2	R/W	0x094	消息接口 2 仲裁 2	0x0000	表 926
IF2_MCTRL	R/W	0x098	消息接口 2 消息控制	0x0000	表 928
IF2_DA1	R/W	0x09C	消息接口 2 数据 A1	0x0000	表 930
IF2_DA2	R/W	0x0A0	消息接口 2 数据 A2	0x0000	表 931
IF2_DB1	R/W	0x0A4	消息接口 2 数据 B1	0x0000	表 934
IF2_DB2	R/W	0x0A8	消息接口 2 数据 B2	0x0000	表 936
-	-	0x0AC - 0x0FC			
TXREQ1	RO	0x100	发送请求 1	0x0000	表 937
TXREQ2	RO	0x104	发送请求 2	0x0000	表 938
-	-	0x108 - 0x11C	保留	-	
ND1	RO	0x120	新数据 1	0x0000	表 939
ND2	RO	0x124	新数据 2	0x0000	表 940
-	-	0x128 - 0x13C	保留	-	
IR1	RO	0x140	中断挂起 1	0x0000	表 941
IR2	RO	0x144	中断挂起 2	0x0000	表 942
-	-	0x148 - 0x15C	保留	-	
MSGV1	RO	0x160	消息有效 1	0x0000	表 943
MSGV2	RO	0x164	消息有效 2	0x0000	表 944
-	-	0x168 - 0x17C	保留	-	
CLKDIV	R/W	0x180	CAN 时钟分频器寄存器	0x0001	表 945

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

41.6.1 CAN 协议寄存器

41.6.1.1 CAN 控制寄存器

硬件复位后，C_CAN 控制器的寄存器会保存表 902 中所述的值。此外，总线关闭状态还会被复位，并且 TD0/1 输出将被设置成高电平。CANCTRL 寄存器的复位值 0x0001 使能由软件执行的初始化 (INIT = 1)。在 CPU 将 INIT 位复位为 0 前，C_CAN 不会影响 CAN 总线。

表 904. CAN 控制寄存器 （CNTL，地址 0x400E 2000 (C_CAN0) 和 0x400A 4000 (C_CAN1)）位描述

位	符号	值	描述	复位值	访问类型
0	INIT		初始化	1	R/W
		0	正常操作。		
		1	初始化启动。复位时，软件需要初始化 CAN 控制器。		
1	IE		模块中断使能	0	R/W
		0	禁用 CAN 中断。中断线始终为 “高电平”。		
		1	使能 CAN 中断。中断线设置为 “低电平” 并保持 “低电平”，直至所有挂起中断被清除。		
2	SIE		状态更改中断使能	0	R/W
		0	禁用状态更改中断。将不会产生状态更改中断。		
		1	使能状态更改中断。消息传输成功完成或检测到 CAN 总线错误时，将产生状态更改中断。		
3	EIE		错误中断使能	0	R/W
		0	禁用错误中断。将不会产生错误状态中断。		
		1	使能错误中断。CANSTAT 寄存器中的位 BOFF 或 EWARN 更改时将产生一个中断。		
4	-	-	保留	0	-
5	DAR		禁用自动重传	0	R/W
		0	受干扰消息的自动重传已使能。		
		1	自动重传已禁用。		
6	CCE		配置更改使能	0	R/W
		0	CPU 没有对位时序寄存器的写访问权限。		
		1	INIT 位为 1 时， CPU 有对 CANBT 寄存器的写访问权限。		
7	TEST		测试模式使能	0	R/W
		0	正常操作。		
		1	测试模式。		
31:8	-		保留	-	-

注：总线关闭恢复序列 （参见 *CAN 规范 2.0 版*）不能通过置位或复位 INIT 位进行缩短。如果设备进入 busoff 状态，则会设置 INIT，停止所有总线活动。 CPU 清除 INIT 后，设备在恢复正常操作前将等待 129 个 “总线空闲” 事件（129 × 11 连续 “高电平” / 隐性位）。在 busoff 恢复序列结束时，“错误管理计数器” 将被复位。

在复位 INIT 后的等待时间内，每次监控到一序列 11 “高电平” / 隐性位时，都会将 Bit0Error 代码写入状态寄存器 CANSTAT，使 CPU 能监控 busoff 恢复序列的进程，然后确定 CAN 总线是停滞在 “低电平” / 显性，还是被连续干扰。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

41.6.1.2 CAN 状态寄存器

表 905. CAN 状态寄存器 （STAT，地址 0x400E 2004 (C_CAN0) 和 0x400A 4004 (C_CAN1)）位描述

位	符号	值	描述	复位值	访问类型
2:0	LEC		上一错误代码 CAN 总线上出现的上一错误的类型。 LEC 字段持有指示 CAN 总线上出现的上一错误的类型代码。 消息无错误传输（接收或发送）后，将此字段清为 0。 CPU 可写入未使用的代码 111 来检查是否有更新。	000	R/W
		0x0	**无错误**。		
		0x1	**填充错误**：序列中超过 5 个相等的位出现在部分已接收消息中，这是不允许的。		
		0x2	**格式错误**：已接收帧的固定格式部分有格式错误。		
		0x3	**应答错误**：此 CAN 内核发送的消息未被确认。		
		0x4	**位 1 错误**：消息发送期间（仲裁字段例外），设备要发送一个“高”/ 隐性电平 （逻辑值 1 的位），但是监控到的总线值为 “低” / 显性电平。		
		0x5	**位 0 错误**：消息 （或确认位，或活动错误标志或过载标志）发送期间，设备要发送一个“低”/ 显性电平（数据或标识符位逻辑值为 0），但是监控到的总线值为“高”/ 隐性电平。在总线关闭状态恢复期间，每次监控了一个由 11 个高电平 / 隐性位组成的序列后，该状态会被置位。这将使 CPU 能监控 busoff 恢复序列的进程 （指示总线没有停滞在 “低电平” / 显性或者连续受干扰）。		
		0x6	**CRC 错误**：收到的消息中的 CRC 校验和不正确。		
		0x7	**未使用**：未检测到 CAN 总线事件 （由 CPU 写入）。		
3	TXOK		成功发送一条消息 此位由 CPU 复位。从不由 CAN 控制器复位。	0	R/W
		0	由于此位被 CPU 复位，因此没有消息成功发送。		
		1	由于此位最后由 CPU 复位，因此消息已成功发送（没有错误且由至少一个其他节点确认）。		
4	RXOK		成功接收一条消息 此位由 CPU 复位。从不由 CAN 控制器复位。	0	R/W
		0	由于此位最后由 CPU 复位，因此没有消息成功发送。		
		1	由于此位最后由 CPU 设置为 0，因此已独立于验收过滤的结果成功接收消息。		
5	EPASS		错误被动	0	RO
		0	CAN 控制器处于错误主动状态。		
		1	CAN 控制器处于 *CAN 2.0 规范*所定义的错误被动状态。		
6	EWARN		警告状态	0	RO
		0	两个错误计数器都低于错误警告限值 96。		
		1	EML 中至少有一个错误计数器达到错误警告限值 96。		
7	BOFF		Busoff 状态	0	RO
		0	CAN 模块不为总线关闭状态。		
		1	CAN 控制器处于 busoff 状态。		
31:8	-	-	保留		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

状态中断由位 BOFF、EWARN、RXOK、TXOK 或 LEC 生成。BOFF 和 EWARN 生成错误中断，且如果在 CANCTRL 寄存器中 EIE 和 SIE 分别被设置为使能，则 RXOK、TXOK 和 LEC 生成状态更改中断。

位 EPASS 的更改以及向 RXOK、 TXOK 或 LEC 的写操作都不会产生状态中断。

读取 CANSTAT 寄存器将清除 CANIR 寄存器中的状态中断值。

41.6.1.3 CAN 错误计数器

表 906. CAN 错误计数器 （EC，地址 0x400E 2008 (C_CAN0) 和 0x400A 4008 (C_CAN1)）位描述

位	符号	值	描述	复位值	访问类型
7:0	TEC_7_0		发送错误计数器 发送错误计数器的当前值 （最大值 127）	0	RO
14:8	REC_6_0		接收错误计数器 接收错误计数器的当前值 （最大值 255）。	0	RO
15	RP		接收错误被动	0	RO
		0	接收计数器低于错误被动级别。		
		1	接收计数器已达到 *CAN2.0 规范*中定义的错误被动级别。		
31:16	-	-	保留	-	-

41.6.1.4 CAN 位时序寄存器

表 907. CAN 位计时寄存器 （BT，地址 0x400E 200C (C_CAN0) 和 0x400A 400C (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
5:0	BRP	波特率预分频器 振荡器频率除以该值以生成位时间量子。位时间从此量子倍数开始累积。波特率预分频器的有效值为 0-63[1]。有效的编程值为 0x01 - 0x3F[1]。	1	R/W
7:6	SJW	（重新）同步跳跃宽度 有效的编程值为 0 至 3[1]。	0	R/w
11:8	TSEG1	采样点后的时段 有效值为 0 至 7[1]。	0011	R/W
14:12	TSEG2	采样点前的时段 有效值为 1 至 15[1]。	010	R/W
31:15	-	保留	-	-

[1] 硬件将编程写入这些位的值解析为位值 + 1。

注：对于 8 MHz 的模块时钟 CAN_CLK，复位值 0x2301 将 C_CAN 的比特率配置为 250 kBit/s。如果配置更改在 CANCTRL 中使能，且控制器由软件初始化 （CAN 控制寄存器中的位 CCE 和 INIT 已设置），则这些寄存器仅可写。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

41.6.1.5 CAN 中断寄存器

表 908. CAN 中断寄存器（INT，地址 0x400E 2010 (C_CAN0) 和 0x400A 4010 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
15:0	INTID15_0	0x0000 = 没有中断挂起 0x0001 至 0x0020 = 导致中断的消息对象编号。 0x0021 至 0x7FFF = 未使用 0x8000 = 状态中断 0x8001 至 0xFFFF = 未使用	0	R
31:16	-	保留	-	-

如果有多个中断挂起，则 CAN 中断寄存器将指向具有最高优先级的挂起中断，而不考虑它们的时间顺序。中断保持挂起，直至 CPU 将其清除。如果 INTID 不同于 0x0000 并且设置了 IE，则到 CPU 的中断线激活。中断线保持激活直至 INTID 返回值 0x0000（原因是中断复位）或直至 IE 复位。

状态中断具有最高的优先级。在消息中断中间，消息对象的中断优先级随着消息编号的增加而降低。

通过清除消息对象的 INTPND 位，清除消息中断。通过读取状态寄存器来清除状态中断。

41.6.1.6 CAN 测试寄存器

对测试寄存器的写访问通过在 CAN 控制寄存器中设置位 Test 来使能。

不同的测试功能可以组合，但是选定 TX[1:0] ≠ “00”时，消息传输会受到干扰。

表 909. CAN检测寄存器（TEST，地址0x400E 2014 (C_CAN0)和0x400A 4014 (C_CAN1)）位描述

位	符号	值	描述	复位值	访问类型
1:0	-	-			-
2	BASIC		基本模式	0	R/W
		0	基本模式禁用。		
		1	IF1 寄存器用作 TX 缓冲区， IF2 寄存器用作 RX 缓冲区。		
3	SILENT		静默模式	0	R/W
		0	正常操作。		
		1	模块处于静默模式。		
4	LBACK		环回模式	0	R/W
		0	环回模式已禁用。		
		1	环回模式已使能。		
6:5	TX1_0		TD 引脚控制	00	R/W
		0x0	TD 引脚上的电平由 CAN 控制器控制。这是复位时的值。		
		0x1	可在 TD 引脚监控采样点。		
		0x2	TD 引脚是驱动 “低电平” / 显性。		
		0x3	TD 引脚是驱动 “高电平” / 隐性。		
7	RX		监控 RD 引脚的实际值。	0	R
		0	CAN 总线显性 (RD = 0)。		
		1	CAN 总线隐性 (RD = 1)。		
31:8	-		保留		-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

41.6.1.7 CAN 波特率预分频器扩展寄存器

表 910. CAN 波特率预分频器扩展寄存器（BRPE，地址 0x400E 2018 (C_CAN0) 和 0x400A 4018 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
3:0	BRPE	波特率预分频器扩展 通过编程 BRPE，波特率预分频器可扩展到最高值 1023。硬件将该值解析为 BRPE (MSB) 和 BRP (LSB) 的值加 1。 允许的值为 0x00 到 0x0F	0x0000	R/W
31:4	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

41.6.2 消息接口寄存器

有两组接口寄存器用于控制 CPU 对消息 RAM 的访问。接口寄存器通过缓冲要传输的数据，避免在 CPU 对消息 RAM 的访问和 CAN 消息接收和发送之间出现冲突。可在消息 RAM 和 IFx 消息缓冲寄存器之间一次传输完整的消息对象（参见第 41.6.2.1 节）或部分消息对象。

两个接口寄存器组的功能是相同的（“基本”测试模式除外）。一组寄存器可用于将数据传输到消息 RAM，而另一组寄存器可用于传输来自消息 RAM 的数据，使两个过程可以相互中断。

每组接口寄存器由自身命令寄存器控制的消息缓冲寄存器组成。指令掩码寄存器指定数据传输的方向，以及将传输消息对象的哪一部分。命令请求寄存器用于在消息 RAM 中选择消息对象，作为传输目标或源，并起动指令掩码寄存器中指定的操作。

图 148. 消息对象传送的功能框图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 911. 消息接口寄存器

IF1 寄存器名称	IF1 寄存器组	IF2 寄存器名称	IF2 寄存器组
IF1_CMDREQ	IF1 命令请求	IF2_CMDREQ	IF2 命令请求
IF1_CMDMASK	IF1 命令掩码	IF2_CMDMASK	IF2 命令掩码
IF1_MASK1	IF1 掩码 1	IF2_MSK1	IF2 掩码 1
IF1_MASK2	IF1 掩码 2	IF2_MSK2	IF2 掩码 2
IF1_ARB1	IF1 仲裁 1	IF2_ARB1	IF2 仲裁 1
IF1_ARB2	IF1 仲裁 2	IF2_ARB2	IF2 仲裁 2
IF1_MCTRL	IF1 消息控制	IF2_MCTRL	IF2 消息控制
IF1_DA1	IF1 数据 A1	IF2_DA1	IF2 数据 A1
IF1_DA2	IF1 数据 A2	IF2_DA2	IF2 数据 A2
CIF1_DB1	IF1 数据 B1	IF2_DB1	IF2 数据 B1
IF1_DB2	IF1 数据 B2	IF2_DB2	IF2 数据 B2

在消息 RAM 中有 32 个消息对象。为避免 CPU 对消息 RAM 的访问与 CAN 消息接收和发送之间出现冲突， CPU 不直接访问消息对象，通过 IFx 接口寄存器访问消息对象。

41.6.2.1 消息对象

消息对象包含来自消息接口寄存器中各个位的信息。下面的表 912 显示了消息对象结构的原理图。表中显示了消息对象的位以及分别设置或清除此位的接口寄存器。有关位功能，请参见相应的接口寄存器。

表 912. 消息 RAM 中消息对象的结构

UMASK	MSK[28:0]	MXTD	MDIR	EOB	NEWDAT	MSGLST	RXIE	TXIE	INTPND
IF1/2_MCTRL	IF1/2_MSK1/2			IF1/2_MCTRL					

RMTEN	TXRQST	MSGVAL	ID[28:0]	XTD	DIR	DLC3	DLC2	DLC1	DLC0
IF1/2_MCTRL		IF1/2_ARB1/2				IF1/2_MCTRL			

DATA0	DATA1	DATA2	DATA3	DATA4	DATA5	DATA6	DATA7
IF1/2_DA1		IF1/2_DA2		IF1/2_DB1		IF1/2_DB2	

41.6.2.2 CAN 消息接口命令请求寄存器

CPU 将消息编号写入命令请求寄存器后，消息传输立即起动。通过此写操作， BUSY 位自动设置为 1，信号 CAN_WAIT_B 被拉低，通知 CPU 正在进行传输。 3 到 6 个 CAN_CLK 周期的等待时间后，接口寄存器和消息 RAM 之间的传输完成。BUSY 位设置回零，并且信号 CAN_WAIT_B 设置回原值。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 913. CAN 信息接口指令请求寄存器（IF1_CMDREQ，地址 0x400E 2020 (C_CAN0) 和 0x400A 4020 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
5:0	消息编号	消息编号 0x01 到 0x20 = 有效的消息编号 消息 RAM 中的消息对象被选定进行数据传输。 0x00 = 不是有效的消息编号。此值被解析为 0x20。[1] 0x21 到 0x3F = 不是有效的消息编号。此值被解析为 0x01 - 0x1F。[1]	0x01	R/W
14:6	-	保留		
15	BUSY	忙标志 写入此命令请求寄存器时，由硬件设置为 1。 此命令请求寄存器的读 / 写操作完成时，由硬件设置为 0。	0	R
31:16	-	保留	-	-

[1] 向指令请求寄存器中写入无效的消息编号时，该消息编号将被转换成一个有效的值，并且将传送该消息对象。

表 914. CAN 信息接口指令请求寄存器（IF2_CMDREQ，地址 0x400E 2080 (C_CAN0) 和 0x400A 4080 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
5:0	消息编号	消息编号 0x01 到 0x20 = 有效的消息编号 消息 RAM 中的消息对象被选定进行数据传输。 0x00 = 不是有效的消息编号。此值被解析为 0x20。[1] 0x21 到 0x3F = 不是有效的消息编号。此值被解析为 0x01 - 0x1F。[1]	0x01	R/W
14:6	-	保留		
15	BUSY	忙标志 写入此命令请求寄存器时，由硬件设置为 1。 此命令请求寄存器的读 / 写操作完成时，由硬件设置为 0。	0	R
31:16	-	保留	-	-

[1] 向指令请求寄存器中写入无效的消息编号时，该消息编号将被转换成一个有效的值，并且将传送该消息对象。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

41.6.2.3 CAN 消息接口指令掩码寄存器

IFx 指令掩码寄存器的控制位指定传输方向并选择哪个 IFx 消息缓冲寄存器是数据传输的源或目标。寄存器位的功能取决于在此指令掩码寄存器的 WR/RD 位（位 7）中选定的传输方向 （读或写）。

将 WR/RD 选择为

1 表示写传输方向 （写入消息 RAM）

0 表示读传送方向 （从消息 RAM 中读取）

写传输方向

表 915. CAN 信息接口指令掩码寄存器写入方向（IF1_CMDMSK_W，地址 0x400E 2024 (C_CAN0) 和 0x400A 4024 (C_CAN1)）位描述

位	符号	值	描述	复位值	访问类型
0	DATA_B		访问数据字节 4-7	0	R/W
		0	数据字节 4-7 不变。		
		1	将数据字节 4-7 传输到消息对象。		
1	DATA_A		访问数据字节 0-3	0	R/W
		0	数据字节 0-3 不变。		
		1	将数据字节 0-3 传输到消息对象。		
2	TXRQST		访问发送请求位	0	R/W
		0	无发送请求。 TXRQSRT 位在 IF1/2_MCTRL 中未改变。 **注：**如果通过编程此位来请求发送，则 CANIFn_MCTRL 寄存器中的 TXRQST 位被忽略。		
		1	请求发送。设置 TXRQST 位 IF1/2_MCTRL。		
3	CLRINTPND	-	此位在写方向被忽略。	0	R/W
4	CTRL		访问控制位	0	R/W
		0	控制位未改变。		
		1	将控制位传输到消息对象		
5	ARB		访问仲裁位	0	R/W
		0	仲裁位未改变。		
		1	将标识符、 DIR、 XTD 和 MSGVAL 位传输到消息对象。		
6	MASK		访问掩码位	0	R/W
		0	掩码位未改变。		
		1	将标识符 MASK + MDIR + MXTD 传输到消息对象。		
7	WR_RD	1	写传输 从选定的消息缓冲寄存器将数据传输到命令请求寄存器 CANIFn_CMDREQ 寻址的消息对象。	0	R/W
31:8	-	-	保留	0	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 916. CAN 信息接口指令掩码寄存器写入方向（IF2_CMDMSK_W，地址 0x400E 2084 (C_CAN0) 和 0x400A 4084 (C_CAN1)）位描述

位	符号	值	描述	复位值	访问类型
0	DATA_B		访问数据字节 4-7	0	R/W
		0	数据字节 4-7 不变。		
		1	将数据字节 4-7 传输到消息对象。		
1	DATA_A		访问数据字节 0-3	0	R/W
		0	数据字节 0-3 不变。		
		1	将数据字节 0-3 传输到消息对象。		
2	TXRQST		访问发送请求位	0	R/W
		0	无发送请求。 TXRQSRT 位在 IF1/2_MCTRL 中未改变。 **注：**如果通过编程此位来请求发送，则 CANIFn_MCTRL 寄存器中的 TXRQST 位被忽略。		
		1	请求发送。设置 TXRQST 位 IF1/2_MCTRL。		
3	CLRINTPND	-	此位在写方向被忽略。	0	R/W
4	CTRL		访问控制位	0	R/W
		0	控制位未改变。		
		1	将控制位传输到消息对象		
5	ARB		访问仲裁位	0	R/W
		0	仲裁位未改变。		
		1	将标识符、 DIR、 XTD 和 MSGVAL 位传输到消息对象。		
6	MASK		访问掩码位	0	R/W
		0	掩码位未改变。		
		1	将标识符 MASK + MDIR + MXTD 传输到消息对象。		
7	WR_RD	1	写传输 从选定的消息缓冲寄存器将数据传输到命令请求寄存器 CANIFn_CMDREQ 寻址的消息对象。	0	R/W
31:8	-	-	保留	0	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

读传输方向

表 917. CAN 信息接口指令掩码寄存器读取方向（IF1_CMDMSK_R，地址 0x400E 2024 (C_CAN0) 和 0x400A 4024 (C_CAN1)）位描述

位	符号	值	描述	复位值	访问类型
0	DATA_B		访问数据字节 4-7	0	R/W
		0	数据字节 4-7 不变。		
		1	将数据字节 4-7 传输到 IFx 消息缓冲寄存器。		
1	DATA_A		访问数据字节 0-3	0	R/W
		0	数据字节 0-3 不变。		
		1	将数据字节 0-3 传输到 IFx 消息缓冲区。		
2	NEWDAT		访问新数据位	0	R/W
		0	NEWDAT 位保持不变。 **注：**对消息对象的读访问可与 IF1/2_MCTRL 中控制位 INTPND 和 NEWDAT 的复位相结合。传输到 IFx 消息控制寄存器的这些位的值始终反映复位这些位之前的状态。		
		1	清除消息对象中的 NEWDAT 位。		
3	CLRINTPND		清除中断挂起位。	0	R/W
		0	INTPND 位保持不变。		
		1	清除消息对象中的 INTPND 位。		
4	CTRL		访问控制位	0	R/W
		0	控制位未改变。		
		1	将控制位传输到 IFx 消息缓冲。		
5	ARB		访问仲裁位	0	R/W
		0	仲裁位未改变。		
		1	将标识符、DIR、XTD 和 MSGVAL 位传输到 IFx 消息缓冲寄存器。		
6	MASK		访问掩码位	0	R/W
		0	掩码位未改变。		
		1	将标识符 MASK + MDIR + MXTD 传输到 IFx 消息缓冲寄存器。		
7	WR_RD	0	读传输 将数据从命令请求寄存器寻址的消息对象传输到选定的消息缓冲寄存器 CANIFn_CMDREQ。	0	R/W
31:8	-	-	保留	0	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 918. CAN 信息接口指令掩码寄存器读取方向（IF2_CMDMSK_R，地址 0x400E 2084 (C_CAN0) 和 0x400A 4084 (C_CAN1)）位描述

位	符号	值	描述	复位值	访问类型
0	DATA_B		访问数据字节 4-7	0	R/W
		0	数据字节 4-7 不变。		
		1	将数据字节 4-7 传输到 IFx 消息缓冲寄存器。		
1	DATA_A		访问数据字节 0-3	0	R/W
		0	数据字节 0-3 不变。		
		1	将数据字节 0-3 传输到 IFx 消息缓冲区。		
2	NEWDAT		访问新数据位	0	R/W
		0	NEWDAT 位保持不变。 **注：**对消息对象的读访问可与 IF1/2_MCTRL 中控制位 INTPND 和 NEWDAT 的复位相结合。传输到 IFx 消息控制寄存器的这些位的值始终反映复位这些位之前的状态。		
		1	清除消息对象中的 NEWDAT 位。		
3	CLRINTPND		清除中断挂起位。	0	R/W
		0	INTPND 位保持不变。		
		1	清除消息对象中的 INTPND 位。		
4	CTRL		访问控制位	0	R/W
		0	控制位未改变。		
		1	将控制位传输到 IFx 消息缓冲。		
5	ARB		访问仲裁位	0	R/W
		0	仲裁位未改变。		
		1	将标识符、DIR、XTD 和 MSGVAL 位传输到 IFx 消息缓冲寄存器。		
6	MASK		访问掩码位	0	R/W
		0	掩码位未改变。		
		1	将标识符 MASK + MDIR + MXTD 传输到 IFx 消息缓冲寄存器。		
7	WR_RD	0	读传输 将数据从命令请求寄存器寻址的消息对象传输到选定的消息缓冲寄存器 CANIFn_CMDREQ。	0	R/W
31:8	-	-	保留	0	-

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

41.6.2.4 IF1 和 IF2 消息缓冲寄存器

消息缓冲寄存器的位镜像消息 RAM 中的消息对象。

41.6.2.4.1 CAN 消息接口命令掩码 1 寄存器

表 919. CAN 信息接口指令掩码 1 寄存器 （IF1_MSK1，地址 0x400E 2028 (C_CAN0) 和 0x400A 4028 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
15:0	MSK15_0	标识符掩码 0 = 消息标识符中相应的位不会禁止在验收滤波中匹配。 1 = 相应的标识符位用于验收滤波。	0xFFFF	R/W
31:16	-	保留	0	-

表 920. CAN 信息接口指令掩码 1 寄存器 （IF2_MSK1，地址 0x400E 2088 (C_CAN0) 和 0x400A 4088 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
15:0	MSK15_0	标识符掩码 0 = 消息标识符中相应的位不会禁止在验收滤波中匹配。 1 = 相应的标识符位用于验收滤波。	0xFFFF	R/W
31:16	-	保留	0	-

41.6.2.4.2 CAN 消息接口命令掩码 2 寄存器

表 921. CAN 信息接口指令掩码 2 寄存器 （IF1_MSK2，地址 0x400E 202C (C_CAN0) 和 0x400A 402C (C_CAN1)）位描述

位	符号	值	描述	复位值	访问类型
12:0	MSK28_16		标识符掩码 0 = 消息标识符中相应的位不会禁止在验收滤波中匹配。 1 = 相应的标识符位用于验收滤波。	0xFFF	R/W
13	-		保留	1	-
14	MDIR		掩码消息方向	1	R/W
		0	消息方向位 (DIR) 对验收过滤没有影响。		
		1	消息方向位 (DIR) 用于验收过滤。		
15	MXTD		掩码扩展标识符	1	R/W
		0	扩展标识符位 (IDE) 对验收过滤没有影响。		
		1	扩展标识符位 (IDE) 用于验收过滤。		
31:16	-	-	保留	0	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 922. CAN 信息接口指令掩码 2 寄存器（IF2_MSK2，0x400E 208C (C_CAN0) 和 0x400A 408C (C_CAN1)）位描述

位	符号	值	描述	复位值	访问类型
12:0	MSK28_16		标识符掩码 0 = 消息标识符中相应的位不会禁止在验收滤波中匹配。 1 = 相应的标识符位用于验收滤波。	0xFFF	R/W
13	-		保留	1	-
14	MDIR		掩码消息方向	1	R/W
		0	消息方向位 (DIR) 对验收过滤没有影响。		
		1	消息方向位 (DIR) 用于验收过滤。		
15	MXTD		掩码扩展标识符	1	R/W
		0	扩展标识符位 (IDE) 对验收过滤没有影响。		
		1	扩展标识符位 (IDE) 用于验收过滤。		
31:16	-	-	保留	0	-

41.6.2.4.3 CAN 消息接口命令仲裁 1 寄存器

表 923. CAN 信息接口指令仲裁 1 寄存器 （IF1_ARB1，地址 0x400E 2030 (C_CAN0) 和 0x400A 4030 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
15:0	ID15_0	消息标识符 29 位标识符 （扩展帧） 11 位标识符 （标准帧）	0x00	R/W
31:16	-	保留	0	-

表 924. CAN 信息接口指令仲裁 1 寄存器 （IF2_ARB1，地址 0x400E 2090 (C_CAN0) 和 0x400A 4090 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
15:0	ID15_0	消息标识符 29 位标识符 （扩展帧） 11 位标识符 （标准帧）	0x00	R/W
31:16	-	保留	0	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

41.6.2.4.4 CAN 消息接口命令仲裁 2 寄存器

表 925. CAN 信息接口指令仲裁 2 寄存器 （IF1_ARB2，地址 0x400E 2034 (C_CAN0) 和 0x400A 4034 (C_CAN1)）位描述

位	符号	值	描述	复位值	访问类型
12:0	ID28_16		消息标识符 29 位标识符 （扩展帧） 11 位标识符 （标准帧）	0x00	R/W
13	DIR		消息方向	0x00	R/W
		0	方向 = 接收。 在 TXRQST 上，发送具有此消息对象标识符的远程帧。在接收具有匹配标识符的数据帧时，该消息存储在此消息对象中。		
		1	方向 = 发送。 在 TXRQST 上，各消息对象作为数据帧发送。在接收具有匹配标识符的远程帧时，设置此消息对象的 TXRQST 位 （如果 RMTEN = 1）。		
14	XTD		扩展标识符	0x00	R/W
		0	11 位标准标识符将用于此消息对象。		
		1	29 位扩展标识符将用于此消息对象。		
15	MSGVAL		消息有效 **注：**在 CAN 控制寄存器中复位位 INIT 前，所有未使用消息对象的 MSGVAL 位在初始化期间复位。修改标识符 ID28:0、控制位 XTD、 DIR 或数据长度代码 DLC3:0 前，或不再要求消息对象时，此位必须设置为 0。	0	R/W
		0	消息对象被消息处理器忽略。		
		1	消息对象的配置应由消息处理器处理。		
31:16	-	-	保留	0	-

表 926. AN 信息接口指令仲裁 2 寄存器 （IF2_ARB2，地址 0x400E 2094 (C_CAN0) 和 0x400A 4094 (C_CAN1)）位描述

位	符号	值	描述	复位值	访问类型
12:0	ID28_16		消息标识符 29 位标识符 （扩展帧） 11 位标识符 （标准帧）	0x00	R/W
13	DIR		消息方向	0x00	R/W
		0	方向 = 接收。 在 TXRQST 上，发送具有此消息对象标识符的远程帧。在接收具有匹配标识符的数据帧时，该消息存储在此消息对象中。		
		1	方向 = 发送。 在 TXRQST 上，各消息对象作为数据帧发送。在接收具有匹配标识符的远程帧时，设置此消息对象的 TXRQST 位 （如果 RMTEN = 1）。		
14	XTD		扩展标识符	0x00	R/W
		0	11 位标准标识符将用于此消息对象。		
		1	29 位扩展标识符将用于此消息对象。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 926. AN 信息接口指令仲裁 2 寄存器 （IF2_ARB2，地址 0x400E 2094 (C_CAN0) 和 0x400A 4094 (C_CAN1)）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
15	MSGVAL		消息有效 **注：**在 CAN 控制寄存器中复位位 INIT 前，所有未使用消息对象的 MSGVAL 位在初始化期间复位。修改标识符 ID28:0、控制位 XTD、 DIR 或数据长度代码 DLC3:0 前，或不再要求消息对象时，此位必须设置为 0。	0	R/W
		0	消息对象被消息处理器忽略。		
		1	消息对象的配置应由消息处理器处理。		
31:16	-	-	保留	0	-

41.6.2.4.5 CAN 消息接口消息控制寄存器

表 927. CAN 信息接口信息控制寄存器 （IF1_MCTRL，地址 0x400E 2038 (C_CAN0) 和 0x400A 4038 (C_CAN1)）位描述

位	符号	值	描述	复位值	访问类型
3:0	DLC3_0		数据长度代码 **注：**消息对象的数据长度代码必须定义为与其他节点上具有相同标识符的所有相应对象相同。消息处理器存储数据帧时，会将 DLC 写为已收到消息给定的值。 0000 至 1000 = 数据帧有 0 - 8 个数据字节。 1001 至 1111 = 数据帧有 8 个数据字节。	0000	R/W
6:4	-		保留	-	-
7	EOB		缓冲区结束	0	R/W
		0	消息对象属于 FIFO 缓冲区，并且不是该 FIFO 缓冲区的最后一个消息对象。		
		1	FIFO 缓冲区的单个消息对象或最后一个消息对象。		
8	TXRQST		发送请求	0	R/W
		0	此消息对象不在等待发送。		
		1	已请求发送此消息对象但尚未完成。		
9	RMTEN		远程使能	0	R/W
		0	收到远程帧后， TXRQST 保持不变。		
		1	收到远程帧后，设置 TXRQST。		
10	RXIE		接收中断使能	0	R/W
		0	成功接收帧后 INTPND 将保持不变。		
		1	成功接收帧后将设置 INTPND。		
11	TXIE		发送中断使能	0	R/W
		0	成功接收帧后 INTPND 位将保持不变。		
		1	成功接收帧后将设置 INTPND。		

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 927. CAN 信息接口信息控制寄存器 （IF1_MCTRL，地址 0x400E 2038 (C_CAN0) 和 0x400A 4038 (C_CAN1)）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
12	UMASK		使用验收掩码 **注：**如果 UMASK 设置为 1，则在将 MAGVAL 设置为 1 前，消息对象的掩码位必须在消息对象初始化期间进行编程。	0	R/W
		0	掩码被忽略。		
		1	使用掩码 （MSK[28:0]、 MXTD 的和 MDIR）进行验收过滤。		
13	INTPND		中断挂起	0	R/W
		0	此消息对象不是中断源。		
		1	此消息对象是中断源。如果没有其他更高优先级的中断，则中断寄存器中的中断标识符将指向此消息对象。		
14	MSGLST		消息丢失 （仅对方向接收中的消息对象有效）。	0	R/W
		0	无消息丢失，因为此位最后由 CPU 复位。		
		1	在 NEWDAT 仍设置时，消息处理器将一条新消息存储到此对象中，则 CPU 丢失一条消息。		
15	NEWDAT		新数据	0	R/W
		0	无新数据被消息处理器写入此消息对象的数据部分，因为此标志最后一次是被 CPU 清除的。		
		1	消息处理器或CPU已将新数据写入此消息对象的数据部分。		
31:16	-	-	保留	0	-

表 928. CAN 信息接口信息控制寄存器 （IF2_MCTRL，地址 0x400E 2098 (C_CAN0) 和 0x400A 4098 (C_CAN1)）位描述

位	符号	值	描述	复位值	访问类型
3:0	DLC3_0		数据长度代码 **注：**消息对象的数据长度代码必须定义为与其他节点上具有相同标识符的所有相应对象相同。消息处理器存储数据帧时，会将 DLC 写为已收到消息给定的值。 0000 至 1000 = 数据帧有 0 - 8 个数据字节。 1001 至 1111 = 数据帧有 8 个数据字节。	0000	R/W
6:4	-		保留	-	-
7	EOB		缓冲区结束	0	R/W
		0	消息对象属于 FIFO 缓冲区，并且不是该 FIFO 缓冲区的最后一个消息对象。		
		1	FIFO 缓冲区的单个消息对象或最后一个消息对象。		
8	TXRQST		发送请求	0	R/W
		0	此消息对象不在等待发送。		
		1	已请求发送此消息对象但尚未完成。		
9	RMTEN		远程使能	0	R/W
		0	收到远程帧后， TXRQST 保持不变。		
		1	收到远程帧后，设置 TXRQST。		

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 928. CAN 信息接口信息控制寄存器 （IF2_MCTRL，地址 0x400E 2098 (C_CAN0) 和 0x400A 4098 (C_CAN1)）位描述 *（续）*

位	符号	值	描述	复位值	访问类型
10	RXIE		接收中断使能	0	R/W
		0	成功接收帧后 INTPND 将保持不变。		
		1	成功接收帧后将设置 INTPND。		
11	TXIE		发送中断使能	0	R/W
		0	成功接收帧后 INTPND 位将保持不变。		
		1	成功接收帧后将设置 INTPND。		
12	UMASK		使用验收掩码 **注：**如果 UMASK 设置为 1，则在将 MAGVAL 设置为 1 前，消息对象的掩码位必须在消息对象初始化期间进行编程。	0	R/W
		0	掩码被忽略。		
		1	使用掩码 （MSK[28:0]、 MXTD 的和 MDIR）进行验收过滤。		
13	INTPND		中断挂起	0	R/W
		0	此消息对象不是中断源。		
		1	此消息对象是中断源。如果没有其他更高优先级的中断，则中断寄存器中的中断标识符将指向此消息对象。		
14	MSGLST		消息丢失 （仅对方向接收中的消息对象有效）。	0	R/W
		0	无消息丢失，因为此位最后由 CPU 复位。		
		1	在 NEWDAT 仍设置时，消息处理器将一条新消息存储到此对象中，则 CPU 丢失一条消息。		
15	NEWDAT		新数据	0	R/W
		0	无新数据被消息处理器写入此消息对象的数据部分，因为此标志最后一次是被 CPU 清除的。		
		1	消息处理器或CPU已将新数据写入此消息对象的数据部分。		
31:16	-	-	保留	0	-

41.6.2.4.6 CAN 消息接口数据 A1 寄存器

在 CAN 数据帧中，DATA0 是第一个、DATA7（在 CAN_IF1B2 和 CAN_IF2B2 中）是最后一个要发送或接收的字节。在 CAN 的串行位流中，每个字节的 MSB 将首先被发送。

注：接收期间，字节 DATA0 是第一个移位到 CAN 内核的移位寄存器的数据字节，字节 DATA7 是最后一个。消息处理器存储数据帧时，会将所有八个数据字节写入消息对象中。如果数据长度代码小于 8，则消息对象的其余字节将被非指定的值覆盖。

表 929. CAN 信息接口数据 A1 寄存器（IF1_DA1，地址 0x400E 203C (C_CAN0) 和 0x400A 403C (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
7:0	DATA0	数据字节 0	0x00	R/W
15:8	DATA1	数据字节 1	0x00	R/W
31:16	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 930. CAN 信息接口数据 A1 寄存器（IF2_DA1，地址 0x400E 209C (C_CAN0) 和 0x400A 409C (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
7:0	DATA0	数据字节 0	0x00	R/W
15:8	DATA1	数据字节 1	0x00	R/W
31:16	-	保留	-	-

41.6.2.4.7 CAN 消息接口数据 A2 寄存器

表 931. CAN 信息接口数据 A2 寄存器（IF1_DA2，地址 0x400E 2040 (C_CAN0) 和 0x400A 4040 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
7:0	DATA2	数据字节 2	0x00	R/W
15:8	DATA3	数据字节 3	0x00	R/W
31:16	-	保留	-	-

表 932. CAN 信息接口数据 A2 寄存器（IF2_DA2，地址 0x400E 20A0 (C_CAN0) 和 0x400A 40A0 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
7:0	DATA2	数据字节 2	0x00	R/W
15:8	DATA3	数据字节 3	0x00	R/W
31:16	-	保留	-	-

41.6.2.4.8 CAN 消息接口数据 B1 寄存器

表 933. CAN 信息接口数据 B1 寄存器（IF1_DB1，地址 0x400E 2044 (C_CAN0) 和 0x400A 4044 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
7:0	DATA4	数据字节 4	0x00	R/W
15:8	DATA5	数据字节 5	0x00	R/W
31:16	-	保留	-	-

表 934. CAN 信息接口数据 B1 寄存器（IF2_DB1，地址 0x400E 20A4 (C_CAN0) 和 0x400A 40A4 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
7:0	DATA4	数据字节 4	0x00	R/W
15:8	DATA5	数据字节 5	0x00	R/W
31:16	-	保留	-	-

41.6.2.4.9 CAN 消息接口数据 B2 寄存器

表 935. CAN 信息接口数据 B2 寄存器（IF1_DB2，地址 0x400E 2048 (C_CAN0) 和 0x400A 4048 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
7:0	DATA6	数据字节 6	0x00	R/W
15:8	DATA7	数据字节 7	0x00	R/W
31:16	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 936. CAN 信息接口数据 B2 寄存器（IF2_DB2，地址 0x400E 20A8 (C_CAN0) 和 0x400A 40A8 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
7:0	DATA6	数据字节 6	0x00	R/W
15:8	DATA7	数据字节 7	0x00	R/W
31:16	-	保留	-	-

41.6.3 消息处理器寄存器

所有消息处理器寄存器都是只读的。其内容（每个消息对象的 TXRQST、NEWDAT、INTPND 和 MSGVAL 位以及中断标识符）是消息处理器 FSM 提供的状态信息。

41.6.3.1 CAN 发送请求 1 寄存器

此寄存器包含消息对象 1 到 16 的 TXRQST 位。通过读取 TXRQST 位，CPU 可检查哪个消息对象的发送请求挂起。特定消息对象的 TXRQST 位可在收到远程帧或成功发送后，由 CPU 通过 IFx 消息接口寄存器或消息处理器进行设置 / 复位。

表 937. CAN发送请求1寄存器（TXREQ1，地址0x400E 2100 (C_CAN0)和0x400A 4100 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
15:0	TXRQST16_1	消息对象 16 到 1 的发送请求位。 0 = 此消息对象不在等待发送。 1 = 已请求发送此消息对象但尚未完成。	0x00	R
31:16	-	保留	-	-

41.6.3.2 CAN 发送请求 2 寄存器

此寄存器包含消息对象 32 到 17 的 TXRQST 位。通过读取 TXRQST 位，CPU 可检查哪个消息对象的发送请求挂起。特定消息对象的 TXRQST 位可在收到远程帧或成功发送后，由 CPU 通过 IFx 消息接口寄存器或消息处理器进行设置 / 复位。

表 938. CAN发送请求2寄存器（TXREQ2，地址0x400E 2104 (C_CAN0)和0x400A 4104 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
15:0	TXRQST32_17	消息对象 32 到 17 的发送请求位。 0 = 此消息对象不在等待发送。 1 = 已请求发送此消息对象但尚未完成。	0x00	R
31:16	-	保留	-	-

41.6.3.3 CAN 新数据 1 寄存器

此寄存器包含消息对象 16 到 1 的 NEWDAT 位。通过读取 NEWDAT 位，CPU 可检查哪个消息对象的数据部分被更新。特定消息对象的 NEWDAT 位可在收到数据帧或成功传输后，由 CPU 通过 IFx 消息接口寄存器或消息处理器进行设置 / 复位。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 939. CAN 新数据 1 寄存器（ND1，地址 0x400E 2120 (C_CAN0) 和 0x400A 4120 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
15:0	NEWDAT16_1	消息对象 16 到 1 的新数据位。 0 = 无新数据被消息处理器写入此消息对象的数据部分，因为此标志最后一次是被 CPU 清除的。 1 = 消息处理器或 CPU 已将新数据写入此消息对象的数据部分。	0x00	R
31:16	-	保留	-	-

41.6.3.4 CAN 新数据 2 寄存器

此寄存器包含消息对象 32 到 17 的 NEWDAT 位。通过读取 NEWDAT 位， CPU 可检查哪个消息对象的数据部分被更新。特定消息对象的 NEWDAT 位可在收到数据帧或成功传输后，由 CPU 通过 IFx 消息接口寄存器或消息处理器进行设置 / 复位。

表 940. CAN 新数据 2 寄存器（ND2，地址 0x400E 2124 (C_CAN0) 和 0x400A 4124 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
15:0	NEWDAT32_17	消息对象 32 到 17 的新数据位。 0 = 无新数据被消息处理器写入此消息对象的数据部分，因为此标志最后一次是被 CPU 清除的。 1 = 消息处理器或 CPU 已将新数据写入此消息对象的数据部分。	0x00	R
31:16	-	保留	-	-

41.6.3.5 CAN 中断挂起 1 寄存器

此寄存器包含消息对象 16 到 1 的 INTPND 位。通过读取 INTPND 位， CPU 可检查哪个消息对象的中断挂起。特定消息对象的 INTPND 位在收到或成功发送帧后，可由 CPU 通过 IFx 消息接口寄存器或消息处理器进行设置 / 复位。这也会影响中断寄存器中 INTPND 的值。

表 941. CAN 中断挂起 1 寄存器（IR1，地址 0x400E 2140 (C_CAN0) 和 0x400A 4140 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
15:0	INTPND16_1	消息对象 16 到 1 的新数据位。 0 = 此消息对象被消息处理器忽略。 1 = 此消息对象是中断源。	0x00	R
31:16	-	保留	-	-

41.6.3.6 CAN 中断挂起 2 寄存器

此寄存器包含消息对象 32 到 17 的 INTPND 位。通过读取 INTPND 位， CPU 可检查哪个消息对象的中断挂起。特定消息对象的 INTPND 位在收到或成功发送帧后，可由 CPU 通过 IFx 消息接口寄存器或消息处理器进行设置 / 复位。这也会影响中断寄存器中 INTPND 的值。

表 942. CAN 中断挂起 2 寄存器（IR2，地址 0x400E 2144 (C_CAN0) 和 0x400A 4144 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
15:0	INTPND32_17	消息对象 32 到 17 的新数据位。 0 = 此消息对象被消息处理器忽略。 1 = 此消息对象是中断源。	0x00	R
31:16	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

41.6.3.7 CAN 消息有效 1 寄存器

此寄存器包含消息对象 16 到 1 的 MSGVAL 位。通过读取 MSGVAL 位， CPU 可检查哪个消息对象有效。特定消息对象的 MSGVAL 位可由 CPU 通过 IFx 消息接口寄存器进行设置 / 复位。

表 943. CAN 信息有效 1 寄存器（MSGV1，地址 0x400E 2160 (C_CAN0) 和 0x400A 4160 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
15:0	MSGVAL16_1	消息对象 16 到 1 的消息有效位。 0 = 此消息对象被消息处理器忽略。 1 = 此消息对象已配置且消息处理器应加以考虑。	0x00	R
31:16	-	保留	-	-

41.6.3.8 CAN 消息有效 2 寄存器

此寄存器包含消息对象 32 到 17 的 MSGVAL 位。通过读取 MSGVAL 位，CPU 可检查哪个消息对象有效。特定消息对象的 MSGVAL 位可由 CPU 通过 IFx 消息接口寄存器进行设置 / 复位。

表 944. CAN 信息有效 2 寄存器（MSGV2，地址 0x400E 2164 (C_CAN0) 和 0x400A 4164 (C_CAN1)）位描述

位	符号	描述	访问类型	复位值
15:0	MSGVAL32_17	消息对象 32 到 17 的消息有效位。 0 = 此消息对象被消息处理器忽略。 1 = 此消息对象已配置且消息处理器应加以考虑。	R	0x00
31:16	-	保留	-	-

41.6.4 CAN 定时寄存器

41.6.4.1 CAN 时钟分频器寄存器

此寄存器确定 CAN 时钟信号。 CAN_CLK 由外设时钟 PCLK 除以此寄存器中的值得出。

表 945. CAN时钟分频器寄存器（CLKDIV，地址0x400E 2180 (C_CAN0)和0x400A 4180 (C_CAN1)）位描述

位	符号	描述	复位值	访问类型
3:0	CLKDIVVAL	时钟分频器值 CAN_CLK = PCLK/($2^{CLKDIVVAL -1}$+1) 0000：CAN_CLK = PCLK 除以 1。 0001: CAN_CLK = PCLK 除以 2。 0010: CAN_CLK = PCLK 除以 3。 0010: CAN_CLK = PCLK 除以 4。 0011：CAN_CLK = PCLK 除以 5。 0100：CAN_CLK = PCLK 除以 9。 0101：CAN_CLK = PCLK 除以 17。 ... 1111: CAN_CLK = PCLK 除以 16385。	0001	R/W
31:4	-	保留	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

41.7 功能描述

41.7.1 复位后 C_CAN 控制器状态

硬件复位后，寄存器会保存表 902 中所述的值。此外，总线关闭状态还会被复位，并且输出 CAN_TXD 将被设置成隐性 （高电平）。 CAN 控制寄存器中的值 0x0001 (INIT= ‘1’) 使能软件初始化。 CAN 控制器在 CPU 将 INIT 复位为 0 前，不与 CAN 总线进行通信。

存储在消息 RAM 中的数据不受硬件复位的影响。加电后，消息 RAM 的内容未定义。

41.7.2 C_CAN 操作模式

41.7.2.1 软件初始化

通过软件、硬件复位或进入 busoff 状态来设置 CAN 控制寄存器中的位 INIT，从而启动软件初始化。

软件初始化期间 （INIT 位设置），出现以下情况：

- 与 CAN 总线来往的所有消息都被停止。
- CAN 输出 CAN_TXD 的状态为隐性 （高电平）。
- EML 计数器保持不变。
- 配置寄存器保持不变。
- 如果也设置了 CAN 控制寄存器中的 CCE 位，则使能对位时序寄存器和 BRP 扩展寄存器的访问。

要初始化CAN控制器，软件必须设置位时序寄存器和每个消息对象。如果不需要消息对象，则将 MSGVAL 位设置为无效就足够了。否则，整个消息对象都必须初始化。

复位 INIT 位即可完成软件初始化。然后，通过等待出现一序列 11 连续隐性位（总线空闲），位流处理器将自身与 CAN 总线上传输的数据同步，才能参与总线活动并开始发送消息。

注：消息对象的初始化与 INIT 无关，并且也可动态完成，但是在 BSP 开始传输消息前，消息对象都应在初始化期间配置为特定的标识符或设置为无效。要在正常操作期间更改消息对象的配置，必须通过将 MSGVAL 位设置为无效来启动 CPU。配置完成后，再次将 MSAGVAL 设置为有效。

41.7.2.2 CAN 消息传输

CAN 控制器已初始化并且 INIT 复位为零后，CAN 内核将自身与 CAN 总线同步并开始消息传输。

收到的消息如果通过了消息处理器的验收滤波，则会存储在相应的消息对象中。整个消息包括仲裁位、 DLC 和存储到消息对象中的八个数据字节。如果使用了标识符掩码，则掩码为不关心的仲裁位会在消息对象中被覆盖。

CPU 可通过接口寄存器随时读取或写入每条消息。消息处理器在出现并发访问时，可确保数据的一致性。

要发送的消息由 CPU 更新。如果消息存在永久性消息对象（仲裁和扩展位在配置时设置），则仅更新数据字节，然后设置 TXRQUT 位和 NEWDAT 位，以启动发送。如果多条发送消息被分配给同一消息对象（消息对象的数量不足时），则在请求发送此消息前必须配置整个消息对象。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

可同时请求发送任何数量的消息对象，并根据其内部优先级按顺序发送。消息可随时更新或设置为无效，甚至是在其请求的发送仍挂起时。如果在消息的挂起发送起动前更新它，则旧数据将被丢弃。

可通过接收具有匹配标识符的远程帧自主请求消息发送，取决于消息对象的配置。

41.7.2.3 禁用自动重传 (DAR)

根据 *CAN 规范（ISO11898，6.3.3 恢复管理）*，CAN 控制器提供了一种方式，自动再发送仲裁丢失的帧或在发送期间受错误干扰的帧。在发送成功完成前，帧发送服务将不会向用户确认。默认情况下，仲裁丢失的帧或错误帧的自动重传是使能的。可以禁用它，使 CAN 控制器在 “时间启动 CAN(TTCAN)”（参见 ISO11898-1）环境中工作。

通过将 CAN 控制寄存器中的位 DARr 编程为 1，使能 “禁用自动重传模式”。在此操作模式下，程序员必须考虑消息缓冲区的控制寄存器中位 TXRQST 和 NEWDAT 的不同特性：

- 发送开始时，接收消息缓冲区的位 TXRQST 复位，而位 NEWDAT 保持被设置。
- 发送成功完成时，位 NEWDAT 复位。
- 发送失败时（仲裁丢失或错误），位 NEWDAT 保持被设置。要重新启动发送，CPU 必须将 TXRQST 设置回 1。

41.7.2.4 测试模式

通过将 CAN 控制寄存器中的位 TEST 设置为 1，进入测试模式。在测试模式下，测试寄存器中的位 TX1、TX0、LBACK、SILENT 和 BASIC 都是可写的。位 RX 监控引脚 RD0,1 的状态，因此只是可读的。位 TEST 复位为 0 时，所有测试寄存器功能都被禁用。

41.7.2.4.1 静默模式

通过将测试寄存器位 SILENT 编程为 1，可将 CAN 内核设置为静默模式。

在静默模式下，CAN 控制器能够接收有效数据帧和有效远程帧，但仅发送 CAN 总线上的隐性位并且不能起动发送。如果需要 CAN 内核来发送显性位 （ACK 位、过载标志、活动错误标志），则该位被内部重新路由，使 CAN 内核监控此显性位，尽管 CAN 总线可能保留在隐性状态。静默模式可用于通过发送显性位 （确认位、错误帧），分析 CAN 总线上的流量而不对其产生影响。

图 149. 静默模式下的 CAN 内核

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

41.7.2.4.2 环回模式

通过将测试寄存器位 LBACK 编程为 1，可将 CAN 内核设置为环回模式。在环回模式下，CAN 内核将自身发送的消息视为收到的消息，并将其存储到接收缓冲区中（如果它们通过验收过滤）。

此模式用于自检功能。为了独立于外来激发，CAN 内核在环回模式中忽略确认错误（在数据 / 远程帧的确认插槽中采样的隐性位）。在此模式下， CAN 内核从其 CAN_TXD 输出到其 CAN_RXD 输入执行内部反馈。 CAN 内核不考虑 CAN_RXD 输入引脚的实际值。可在 CAN_TXD 引脚监控发送的消息。

图 150. 环回模式下的 CAN 内核

41.7.2.4.3 结合静默模式的环回模式

也可通过同时将位 LBACK 和 SILENT 编程为 1，将环回模式和静默模式结合在一起。该模式可用于“热自检”，意味着可检测 C_CAN，而不影响连接到引脚 CAN_TXD 和 CAN_RXD 的 CAN 系统的运行。在此模式下，从 CAN 内核断开 CAN_RXD 引脚的连接， CAN_TXD 引脚保持隐性，

图 151. 结合了静默模式的环回模式下的 CAN 内核

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

41.7.2.4.4 基本模式

通过将测试寄存器位 BASIC 编程为 1，可在基本模式中设置 CAN 内核。在此模式下，CAN 控制器在没有消息 RAM 的条件下运行。

IF1 寄存器用作发送缓冲区。通过将 IF1 命令请求寄存器的 BUSY 位写为 1，请求发送 IF1 寄存器的内容。 IF1 寄存器在 BUSY 位设置时锁定。 BUSY 位表示发送挂起。

一旦 CAN 总线空闲，IF1 寄存器就加载到 CAN 内核的移位寄存器中并启动发送。发送完成后， BUSY 位复位，锁定的 IF1 寄存器被释放。

在 IF1 寄存器锁定时，可通过复位 IF1 命令请求寄存器中的 BUSY 位，随时中止挂起的发送。如果 CPU 已复位 BUSY 位，则禁用仲裁丢失或错误时的可能重发。

IF2 寄存器用作接收缓冲区。接收消息后，移位寄存器的内容存储到 IF2 寄存器中，没有任何验收过滤操作。

另外，在消息发送期间还可监控移位寄存器的实际内容。每次通过向 IF2 命令请求寄存器的 BUSY 位写入 1 开始读取消息对象时，移位寄存器的内容都存储到 IF2 寄存器中。

在基本模式下，对所有消息对象相关控制和状态位以及 IFx 指令掩码寄存器控制位的评估都会关闭。命令请求寄存器的消息编号不评估。IF2 消息控制寄存器的 NEWDAT 和 MSGLST 位保留其功能， DLC3-0 将显示收到的 DLC，其他扩展位将被读为 0。

在基本模式中，就绪输出 CAN_WAIT_B 禁用 （始终为 1）。

41.7.2.4.5 引脚 CAN_TXD 的软件控制

CAN 发送引脚 CAN_TXD 有四个输出功能可用：

1. 串行数据输出 （默认）。
2. 驱动 CAN 采样点信号来监控 CAN 控制器的时序。
3. 驱动隐性常数值。
4. 驱动显性常数值。

最后两种功能，与可读 CAN 接收引脚 CAN_RXD 结合，可用于检查 CAN 总线的物理层。

如第 41.6.1.6 节中所述，通过对检测寄存器位 TX1 和 TX0 进行编程，可选择引脚 CAN_TXD 的输出模式。

注：引脚 CAN_TXD 接口的三种检测功能会影响所有的 CAN 协议功能。发送 CAN 消息时或选定了任何一个测试模式、环回模式、静默模式或基本模式时，CAN_TXD 引脚必须保持其默认功能。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

41.7.3 CAN 消息处理器

信息处理器控制 CAN 内核中 Rx/Tx 转移寄存器、消息 RAM 和 IFx 寄存器之间的数据传输，参见图 148。

信息处理机控制以下功能：

- IFx 寄存器和消息 RAM 之间的数据传输
- 从移位寄存器到消息 RAM 的数据传输
- 从消息 RAM 到移位寄存器的数据传输
- 从移位寄存器到验收过滤单元的数据传输
- 为匹配消息对象而扫描消息 RAM
- 处理 TXRQST 标志
- 处理中断

图 152. 消息对象传送的功能框图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

41.7.3.1 管理消息对象

消息 RAM 中消息对象的配置（位 MSGVAL、NEWDAT、INTPND 和 TXRQST 除外）不受复位芯片的影响。所有消息对象都必须由 CPU 初始化，或必须设置为无效 (MSGVAL = 0)。位时序必须在 CPU 清除 CAN 控制寄存器中的 INIT 位前配置。

配置消息对象的方法为，将掩码、仲裁、两个接口寄存器组之一的控制和数据字段编程到所需的值。通过写入相应的 IFx 命令请求寄存器，将 IFx 消息缓冲寄存器加载到消息 RAM 的寻址消息对象中。

CAN 控制寄存器中的 INIT 位被清除时，CAN 内核的 CAN 协议控制器状态机和消息处理器状态机控制 CAN 控制器的内部数据流。收到的通过验收滤波的消息被存储到消息 RAM 中，具有挂起发送请求的消息被加载到 CAN 内核的移位寄存器中，并通过 CAN 总线发送。

CPU 读取收到的消息并更新要通过 IFx 接口寄存器发送的消息。 CPU 在出现某些 CAN 消息和 CAN 错误事件时中断，取决于配置。

41.7.3.2 IFx 寄存器和消息 RAM 之间的数据传输

CPU 在 IFx 寄存器和消息 RAM 之间启动数据传输时，消息处理器将相关命令寄存器中的 BUSY 位设置为 1。传输完成后，将 BUSY 位设置回 0。

指令掩码寄存器指定是发送整个消息对象还是仅发送部分消息对象。由于消息 RAM 的结构，不可能对消息对象的单个位 / 字节执行写操作。软件必须始终将整个消息对象写入消息 RAM 中。因此从 IFx 寄存器到消息 RAM 的数据传输要求一个 “读 - 修改 - 写” 周期：

1. 使用指令掩码寄存器从消息 RAM 读取无需更改的消息对象部分。
   - 读取部分消息对象后，指令掩码寄存器中未选定的消息缓冲寄存器将保持不变。
2. 将消息缓冲寄存器的完整内容写入消息对象中。
   - 写入部分消息对象后，指令掩码寄存器中未选定的消息缓冲寄存器将设置为选定消息对象的实际内容。

41.7.3.3 在 CAN 内核中的移位寄存器和消息缓冲区之间发送消息

如果 CAN 内核单元的移位寄存器已准备好加载，并且如果 IFx 寄存器和消息 RAM 之间没有数据发送，则将评估消息有效寄存器中的 MSGVAL 位和发送请求寄存器中的 TXRQST 位。由消息处理器将具有最高优先级挂起发送请求的有效消息对象加载到移位寄存器中，然后启动发送。消息对象的 NEWDAT 位复位。

成功发送后，如果发送开始后没有新数据写入消息对象 (NEWDAT = 0)，则 TXRQST 位将被复位。如果 TXIE 设置，则在成功发送后设置 INTPND。如果 CAN 控制器失去仲裁或在发送期间出现错误，则将在 CAN 总线再次空闲时立即重发消息。如果同时请求发送具有更高优先级的消息，则按优先级顺序发送消息。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

41.7.3.4 收到消息的验收过滤

传入消息的仲裁和控制字段（标识符 + IDE + RTR + DLC）完全移位到 CAN 内核的 Rx/Tx 移位寄存器中后，消息处理器状态机开始为匹配有效消息对象扫描消息 RAM。

要为匹配有效消息对象扫描消息 RAM，应从 CAN 内核移位寄存器给验收过滤单元加载仲裁位。然后将消息对象 1 的仲裁和掩码字段（包括 MSGVAL、UMASK、NEWDAT 和 EOB）加载到验收过滤单元中，并与来自移位寄存器的仲裁字段进行比较。为后续的每个消息对象重复此操作，直至找到匹配的消息对象或到达消息 RAM 的结尾。

如果出现匹配，则扫描停止，消息处理器状态机继续操作，取决于所收到帧 （数据帧或远程帧）的类型。

41.7.3.4.1 接收数据帧

消息处理器状态机将来自 CAN 内核移位寄存器的消息存储到消息 RAM 中各自的消息对象中。数据字节、所有仲裁位和数据长度代码都存储到相应的消息对象中。执行此操作的目的是保持数据字节与标识符连接，即使已使用了仲裁掩码寄存器。

设置 NEWDAT 位的目的是指示新数据 （CPU 尚未看到）已经收到。 CPU/ 软件应在读取消息对象时复位 NEWDAT。如果在接收时 NEWDAT 位已经设置，则设置 MSGLST 的目的是指示前一数据 （假设 CPU 看不到）丢失。如果 RxIE 位设置，则 INTPND 位也设置，会导致中断寄存器指向此消息对象。

此消息对象的 TXRQST 位复位，以防止在请求的数据帧刚刚收到时发送远程帧。

41.7.3.4.2 接收远程帧

收到远程帧时，必须考虑匹配消息对象的三种不同配置：

1. DIR = 1 （方向 = 发送）， RMTEN = 1， UMASK = 1 或 0

   接收到匹配远程帧时，该信息对象的 TXRQST 位被设置。其他信息对象保持不变。
2. DIR = 1 （方向 = 发送）， RMTEN = 0， UMASK = 0

   接收匹配远程帧时，此消息对象的 TXRQST 位保持不变；远程帧被忽略。
3. DIR = 1 （方向 = 发送）， RMTEN = 0， UMASK = 1

   接收匹配远程帧时，此消息对象的 TXRQST 位复位。来自移位寄存器的仲裁和控制字段（标识符 + IDE + RTR + DLC）被存储到消息 RAM 的消息对象中，且此消息对象的 NEWDAT 位被设置。消息对象的数据字段保持不变；远程帧的处理方法与收到的数据帧类似。

41.7.3.5 接收 / 发送优先级

消息对象的接收 / 发送优先级被附加到消息编号上。消息对象 1 具有最高优先级，而消息对象 32 具有最低的优先级。如果有多个发送请求挂起，则根据相应消息对象的优先级进行处理。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

41.7.3.6 配置发送对象

表 946 显示了软件应如何初始化发送对象 （另请参见表 912）：

表 946. 初始化发送对象

MSGVAL	仲裁位	数据位	掩码位	EOB	DIR	NEWDAT
1	应用相关	应用相关	应用相关	1	1	0
MSGLST	**RXIE**	**TXIE**	**INTPND**	**RMTEN**	**TXRQST**	
0	0	应用相关	0	应用相关	0	

仲裁寄存器（ID28:0 和 XTD 位）由应用给定，它们定义发出消息的标识符和类型。如果使用了11位标识符（标准帧），则其被编程为ID28。在此情况下，可以忽略ID18、ID17到ID0。

如果 TXIE 位设置，则 INTPND 位将在成功发送消息对象后设置。

如果 RMTEN 位设置，则收到匹配的远程帧将导致 TXRQST 位被设置，并且远程帧将被数据帧自主应答。

数据寄存器(DLC3:0, Data0:7)由应用给定。TXRQST和RMTEN在数据有效前可以不设置。

掩码寄存器（Msk28-0、UMASK、MXTD 和 MDIR 位）可用于 (UMASK=1) 允许具有类似标识符的远程帧组来设置 TXRQST 位。有关详情，请参阅第 41.7.3.4.2 节。 DIR 位不能被掩码。

41.7.3.7 更新发送对象

CPU 可通过 IFx 接口寄存器在任何时候更新发送对象的数据字节。更新前， MSGVAL 和 TXRQST 都不必复位。

即使只有一部分数据字节要更新，在将该寄存器的内容传输到消息对象前，相应 IFx 数据 A 寄存器或 IFx 数据 B 寄存器的所有四个字节都必须有效。在 CPU 写入新的数据字节前，CPU 必须将所有四个字节写入 IFx 数据寄存器，或者消息对象被传输到 IFx 数据寄存器中。

仅更新 （八个）数据字节时，第一个 0x0087 被写入指令掩码寄存器。然后将消息对象的编号写入命令请求寄存器，同时更新数据字节并设置 TXRQST。

在数据更新时，为防止可能已经正在进行的发送结束时 TXRQST 复位， NEWDAT 必须与 TXRQST 一起设置。有关详情，请参阅第 41.7.3.3 节。

NEWDAT 与 TXRQST 一起设置时，一旦新发送启动， NEWDAT 将被复位。

41.7.3.8 配置接收对象

表 947 显示了软件应如何初始化接收对象 （另请参见表 912）

表 947. 初始化接收对象

MSGVAL	仲裁位	数据位	掩码位	EOB	DIR	NEWDAT
1	应用相关	应用相关	应用相关	1	0	0
MSGLST	**RXIE**	**TXIE**	**INTPND**	**RMTEN**	**TXRQST**	
0	应用相关	0	0	0	0	

仲裁寄存器 （ID28-0 和 XTD 位）由应用给定，它们定义所接受的收到消息的标识符和类型。如果使用了 11 位标识符（标准帧），则其被编程为 ID28 到 ID18。这时可忽略 ID17 到 ID0。收到具有 11 位标识符的数据帧时， ID17 到 ID0 将被设置为 0。

如果 RxIE 位设置，则当收到的数据帧被接受并存储到消息对象中时， INTPND 位被设置。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

数据长度代码 (DLC[3:0]) 由应用给定。消息处理器在消息对象中存储数据帧时，将存储收到的数据长度代码和八个数据字节。如果数据长度代码小于 8，则消息对象的其余字节将被非指定的值覆盖。

掩码寄存器 （Msk[28:0]、 UMASK、 MXTD 和 MDIR 位）可用于 (UMASK=1) 允许接受具有类似标识符的数据帧组。有关详情，请参见第 41.7.3.4.1 节。在典型应用中 DIR 位不应掩码。

41.7.3.9 处理收到的消息

CPU 可通过 IFx 接口寄存器随时读取收到的消息。数据一致性由消息处理器状态机保证。

要将收到的整个消息从消息 RAM 发送到消息缓冲区中，软件必须将第一个 0x007F 写入指令掩码寄存器，然后将消息对象的编号写入命令请求寄存器。此外，位NEWDAT和INTPND在消息 RAM 中 （而非在消息缓冲区中）清除。

如果消息对象使用验收过滤掩码，则仲裁位显示已接收哪些匹配消息。

NEWDAT 的实际值显示自上次读取此消息对象后是否收到一条新消息。MSGLST 的实际值显示自从上次读取此消息对象后是否收到多条新消息。 MSGLST 将不会自动复位。

使用远程帧，CPU 可请求另一 CAN 节点为接收对象提供新数据。设置接收对象的 TXRQST 位，将导致发送具有接收对象标识符的远程帧。此远程帧启动其他 CAN 节点，起动匹配数据帧的发送。如果在远程帧可发送前已收到匹配数据帧，则 TXRQST 位自动复位。

41.7.3.10 FIFO 缓冲区配置

除 EOB 位外，属于 FIFO 缓冲的接收对象的设置与 （单个）接收对象的设置相同，参见第 41.7.3.8 节。

要将两个或多个消息对象级联到 FIFO 缓冲区中，这些消息对象的标识符和掩码 （如果使用）都必须编程为匹配值。由于消息对象的隐式优先级，具有最低编号的消息对象将是 FIFO 缓冲区的第一个消息对象。FIFO 缓冲区中所有消息对象（最后一个除外）的 EOB 位都必须编程为 0。FIFO 缓冲区最后一个消息对象的 EOB 位被设置为 1，将其配置为“块末尾”。

41.7.3.10.1 使用 FIFO 缓冲区接收消息

标识符与 FIFO 缓冲区匹配的收到消息被存储到此 FIFO 缓冲区的消息对象中，从具有最低消息编号的消息对象开始。

将消息存储到 FIFO 缓冲区的消息对象中时，此消息对象的 NEWDAT 位设置。在 EOB 为 0 时设置 NEWDAT，会锁定消息对象，以便消息处理器进行进一步写访问，直至 CPU 将 NEWDAT 位写回 0。

消息存储到FIFO缓冲区中，直至此FIFO缓冲区的最后一个消息对象到达。如果将NEWDAT写为 0 没有释放前置消息对象之一，则此 FIFO 缓冲区的所有进一步消息都将被写入 FIFO 缓冲区的最后一个消息对象中，从而覆盖先前的消息。

41.7.3.10.2 从 FIFO 缓冲区读取

当 CPU 通过将消息对象的编号写入 IFx 命令请求寄存器，将消息对象的内容传输到 IFx 消息缓冲寄存器时，相应指令掩码寄存器中的位 NEWDAT 和 INTPND 应复位为 0 （TXRQST/NEWDAT = 1 且 ClrINTPND = 1）。消息控制寄存器中这些位的值始终反映位复位之前的状态。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

为确保 FIFO 缓冲区的正确功能，CPU 应从具有最低消息编号的 FIFO 对象开始读取消息对象。

图 153. 将消息从 FIFO 缓冲区读取到消息缓冲区

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

41.7.4 中断处理

如果有多个中断挂起，则 CAN 中断寄存器将指向具有最高优先级的挂起中断，而不考虑它们的时间顺序。中断保持挂起，直至 CPU 将其清除。

状态中断具有最高的优先级。在消息中断中间，消息对象的中断优先级随着消息编号的增加而降低。

通过清除消息对象的 INTPND 位，清除消息中断。通过读取状态寄存器来清除状态中断。

中断寄存器中的中断标识符 INTID 指示中断的原因。没有中断挂起时，该寄存器将保持值为 0。如果中断寄存器的值不为 0，则存在一个中断挂起，并且如果 IE 设置，则到 CPU、IRQ_B 的中断线激活。中断线保持激活直至中断寄存器返回值 0 （原因是中断复位）或直至 IE 复位。

值 0x8000 表示中断挂起的原因是 CAN 内核已更新（不一定是更改）状态寄存器（错误中断或状态中断）。此中断具有最高优先级。 CPU 可更新 （复位）状态位 RXOK、 TXOK 和 LEC，但是 CPU 对状态寄存器的写访问从来不会产生或复位中断。

其他所有值表示中断源是某个消息对象，此时 INTID 指向具有最高中断优先级的挂起消息中断。

CPU 控制状态寄存器的更改是否会导致中断（CAN 控制寄存器中的位 EIE 和 SIE），以及中断寄存器不为 0 时中断线是否激活 （CAN 控制寄存器中的 IE 位）。即使在 IE 复位时中断寄存器也将更新。

CPU 可采用两种方式来跟踪消息中断源：

- 软件跟踪中断寄存器中的 INTID。
- 软件轮询中断挂起寄存器。

读取消息 （该消息是中断源）的中断服务例程可读取消息并同时复位消息对象的 INTPND （指令掩码寄存器中的位 ClrINTPND）。INTPND 被清除时，中断寄存器将指向下一个具有挂起中断的消息对象。

41.7.5 位时序

即使 CAN 位时序配置中出现的较小错误不会立即导致故障， CAN 网络的性能也会显著降低。在很多情况下，CAN 位同步会将 CAN 位时序错误的配置修正到只是偶尔产生错误帧的程度。但是在仲裁的情况下，两个或多个 CAN 节点同时尝试发送一个帧时，错放的采样点会导致发送器之一变成错误被动。

分析此偶发错误要求详细了解CAN节点内的CAN位同步以及CAN总线上CAN节点的交互。

41.7.5.1 位时序和比特率

CAN 支持从小于 1 kBit/s 到最高 1000 kBit/s 范围内的比特率。CAN 网络的每个成员都有自己的时钟发生器，通常是晶体振荡器。可为每个 CAN 节点单独配置位时间（即比特率的倒数）的时序参数，从而创建一个公共比特率，即使 CAN 节点的振荡周期 (f_{osc}) 可能不同。

这些振荡器的频率不是绝对稳定的，因为温度或电压的变化以及器件性能的退化会导致小变化。只要变化保持在指定的振荡器容差范围 (df) 内， CAN 节点就能够通过重新同步到位流来补偿不同的比特率。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

根据 CAN 规范，位时间分为四段（图 154）。同步段、传播时间段、相缓冲段 1 和相缓冲段 2。各个段由特定的可编程数量的时间量子组成（参见表 948）。作为位时间基本单元的时间量子 (t_q) 的长度，由 CAN 控制器的系统时钟 f 和波特率预分频器 (BRP) 定义：t_q = BRP / f_{sys}。 C_CAN 的系统时钟 f_{sys} 是 C_CAN 外设时钟的频率。

同步段 Sync_Seg 是预计要出现 CAN 总线电平边沿的位时间部分，在 Sync_Seg 外侧出现的边沿和 Sync_Seg 之间的距离称为该边沿的相位误差。传播时间段 Prop_Seg 用来补偿 CAN 网络中的物理延迟时间。相缓冲段 Phase_Seg1 和 Phase_Seg2 包围采样点。（重新）“同步跳跃宽度 (SJW)”定义重新同步可将采样点在相缓冲段定义的限值内移动多远，才能对边沿相位误差进行补偿。

表 948 说明了 CAN 协议所要求的最小可编程范围。位时间参数通过 BT 寄存器编程，表 947。有关位时序和示例的详情，参见 *C_CAN 用户手册，版本 1.2*。

表 948. C_CAN 位时间参数

参数	范围	功能
BRP	(1...32)	时间量子 t_q 的长度。
SYNC_SEG	$1t_q$	同步段。固定长度。同步总线输入和系统时钟。
PROP_SEG	(1...8) ≤ t_q	传播时间段。补偿物理延迟时间。此参数由 C_CAN 网络中的系统延迟时间确定。
TSEG1	(1...8) ≤ t_q	相缓冲段 1。可通过同步临时延长。
TSEG2	(1...8) ≤ t_q	相缓冲段 2。可通过同步临时缩短。
SJW	(1...4) ≤ t_q	（重新）同步跳跃宽度。不能大于任一相缓冲段。

图 154. 位时序

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

42.1 本章导读

所有 LPC43xx 器件上均可使用 I2C 总线接口 I2C0 和 I2C1。

42.2 基本配置

I2C0/1 的配置如下：

- 参见表 949 以了解时钟与电源控制。
- I2C0/1 由 I2C0/1_RST (reset # 48/49) 复位。
- I2C0/1 中断连接到嵌套向量中断控制器 (NVIC) 中的插槽 # 18/19。
- 通过 SYSCON 模块中的 SFSI2C0 寄存器为超快速模式、快速模式或标准模式配置 I2C0 引脚 （参见表 127）。

表 949. I2C0/1 时钟及电源控制

	基本时钟	分支时钟	工作频率
I2C0 寄存器接口时钟和 I2C0 外设时钟。	BASE_APB1_CLK	CLK_APB1_I2C0	高达 204 MHz
I2C1 寄存器接口时钟和 I2C1 外设时钟。	BASE_APB3_CLK	CLK_APB3_I2C1	高达 204 MHz

42.3 功能要点

- I2C0 总线接口使用真正的开漏引脚并支持超快速模式 （比特率高达 1Mbit/s）、快速模式或标准模式。
- I2C1 总线接口使用标准端口引脚，支持高达 400 kbit/s 的比特率。
- 标准 I^2C 兼容总线接口，可配置为主机、从机或主 / 从机。
- 在同时发送的主接口之间处理仲裁，不会破坏总线上的串行数据。
- 可编程时钟允许调整 I^2C 传输速率。
- 主从接口之间的数据传输是双向的。
- 串行时钟同步允许具有不同位率的设备通过一个串行总线通信。
- 串行时钟同步用作信号握手机制，以挂起并恢复串行传输。
- 可以选择识别多达四个不同的从属地址。
- 监控模式可观察所有的 I^2C 总线流量，而不用考虑从属地址。
- I^2C 总线可用于测试和诊断。
- I^2C 总线包含一个带有 2 个引脚的标准 I^2C 兼容总线接口。

42.4 应用

与外部 I^2C 标准器件相连接，如串行 RAM、 LCD、音频发生器和其它微控制器等。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

42.5 简介

典型的 I²C 总线配置如图 155 所示。根据方向位的状态 (R/W)，I²C 总线上可能存在以下两种类型的数据传输方式：

- 从主机发送器到从机接收器的数据传输。主机发送的第一个字节是从属地址，后面跟着许多数据字节。从机在接收到每个字节后返回一个确认位。
- 从从机发送器到主机接收器的数据传输。第一个字节（从属地址）由主机发送，从机随后返回确认位。后面跟随从机发送到主机的数据字节。主机在接收到所有字节而非最后一个字节后返回确认位。在最后接收的字节末尾返回一个“未确认”。主设备生成所有串行时钟脉冲以及“起动”和“停止”条件。传输在“停止”条件或在“重复起动”条件下结束。由于重复起始条件也是下一次串行传输的开始，因此不释放 I²C 总线。

I²C 接口是字节导向型，有 4 个操作模式：主机发送器模式、主机接收器模式、从机发送器模式和从机接收器模式。

I²C 接口遵循整个 I²C 规范，支持在不影响同一 I²C 总线上其它器件的情况下关闭处理器。

图 155. I²C 总线配置

42.5.1 I²C 超快速模式

超快速模式支持以 1 Mbit/ 秒的传输速率与恩智浦半导体现在所提供的 I²C 总线产品通信。

要使用超快速模式，就必须在 SYSCON 模块的 SFSI2C0 寄存器正确配置 I²C 引脚（参见表 127）。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

42.6 引脚说明

表 950. I²C 总线引脚描述

函数名称	类型	描述
SDA0	输入 / 输出	I²C 数据输入 / 输出。开漏输出 （符合 I²C 总线规范）。
SCL0	输入 / 输出	I²C 时钟输入 / 输出。开漏输出 （符合 I²C 总线规范）。
SDA1	输入 / 输出	I²C 串行数据。使用标准 I/O 引脚 （仅快速模式）。
SCL1	输入 / 输出	I²C 串行时钟。使用标准 I/O 引脚 （仅快速模式）。

对于标准 / 快速模式或超快速模式， I²C 总线引脚必须通过 SYSCON 寄存器配置。

42.7 寄存器描述

表 951. 寄存器简介：I²C0 （基址 0x400A 1000）

名称	访问类型	地址偏移	描述	复位值[1]	参考
CONSET	R/W	0x000	**I2C 控制置位寄存器。**当向该寄存器的位写 1 时，I²C 控制寄存器中的相应位置位。写 0 时对 I²C 控制寄存器的相应位没有影响。	0x00	表 953
STAT	RO	0x004	**I2C 状态寄存器。**在 I²C 工作期间，该寄存器提供详细的状态码，允许软件决定需要执行的下一步操作。	0xF8	表 954
DAT	R/W	0x008	**I2C 数据寄存器。**在主 / 从发送模式期间，要发送的数据写入该寄存器。在主机或从机接收模式期间，可从此寄存器读取已经接收的数据。	0x00	表 955
ADR0	R/W	0x00C	**I2C 从属地址寄存器 0。**包含 7 位从属地址，用于从机模式下 I²C 接口操作，不用于主机模式下。最低有效位确定从机是否响应通用调用地址。	0x00	表 956
SCLH	R/W	0x010	**SCH 占空比寄存器高半字。**决定 I²C 时钟的高电平时间。	0x04	表 957
SCLL	R/W	0x014	**SCL 占空比寄存器低半字。**决定 I²C 时钟低电平时间。SCLL 和 SCLH 共同决定 I²C 主机产生的时钟频率及从机模式下所用的时间。	0x04	表 958
CONCLR	WO	0x018	**I2C 控制清除寄存器。**当向该寄存器的位写 1 时，I²C 控制寄存器中的相应位清零。写 0 时对 I²C 控制寄存器的相应位没有影响。	-	表 959
MMCTRL	R/W	0x01C	**监控模式控制寄存器。**	0x00	表 960
ADR1	R/W	0x020	**I2C 从属地址寄存器 1。**包含 7 位从属地址，用于从机模式下 I²C 接口操作，不用于主机模式下。最低有效位确定从机是否响应通用调用地址。	0x00	表 961
ADR2	R/W	0x024	**I2C 从属地址寄存器 2。**包含 7 位从属地址，用于从机模式下 I²C 接口操作，不用于主机模式下。最低有效位确定从机是否响应通用调用地址。	0x00	表 961
ADR3	R/W	0x028	**I2C 从属地址寄存器 3。**包含 7 位从属地址，用于从机模式下 I²C 接口操作，不用于主机模式下。最低有效位确定从机是否响应通用调用地址。	0x00	表 961
DATA_BUFFER	RO	0x02C	**数据缓冲寄存器。**每次从总线接收到 9 个位 （8 位数据和 ACK 或 NACK）后， DAT 移位寄存器的 8 MSB 的内容将自动传输到 DATA_BUFFER。	0x00	表 963

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 951. 寄存器简介：I²C0（基址 0x400A 1000） *（续）*

名称	访问类型	地址偏移	描述	复位值[1]	参考
MASK0	R/W	0x030	**I2C 从属地址掩码寄存器 0**。该掩码寄存器与 ADR0 共同决定地址匹配。与通用调用地址（‘0000000’）相比时，掩码寄存器不产生任何影响。	0x00	表 964
MASK1	R/W	0x034	**I2C 从属地址掩码寄存器 1**。该掩码寄存器与 ADR0 共同决定地址匹配。与通用调用地址（‘0000000’）相比时，掩码寄存器不产生任何影响。	0x00	表 964
MASK2	R/W	0x038	**I2C 从属地址掩码寄存器 2**。该掩码寄存器与 ADR0 共同决定地址匹配。与通用调用地址（‘0000000’）相比时，掩码寄存器不产生任何影响。	0x00	表 964
MASK3	R/W	0x03C	**I2C 从属地址掩码寄存器 3**。该掩码寄存器与 ADR0 共同决定地址匹配。与通用调用地址（‘0000000’）相比时，掩码寄存器不产生任何影响。	0x00	表 964

[1] 复位值仅反映存储在所用位中的数据，不包括保留位的内容。

表 952. 寄存器简介：I²C1（基址 0x400E 0000）

名称	访问类型	地址偏移	描述	复位值[1]	参考
CONSET	R/W	0x000	**I2C 控制置位寄存器。**当向该寄存器的位写 1 时，I²C 控制寄存器中的相应位置位。写 0 时对 I²C 控制寄存器的相应位没有影响。	0x00	表 953
STAT	RO	0x004	**I2C 状态寄存器。**在 I²C 工作期间，该寄存器提供详细的状态码，允许软件决定需要执行的下一步操作。	0xF8	表 954
IDAT	R/W	0x008	**I2C 数据寄存器。**在主 / 从发送模式期间，要发送的数据写入该寄存器。在主机或从机接收模式期间，可从此寄存器读取已经接收的数据。	0x00	表 955
ADR0	R/W	0x00C	**I2C 从属地址寄存器 0。**包含 7 位从属地址，用于从机模式下 I²C 接口操作，不用于主机模式下。最低有效位确定从机是否响应通用调用地址。	0x00	表 956
SCLH	R/W	0x010	**SCH 占空比寄存器高半字。**决定 I²C 时钟的高电平时间。	0x04	表 957
SCLL	R/W	0x014	**SCL 占空比寄存器低半字。**决定 I²C 时钟低电平时间。SCLL 和 SCLH 共同决定 I²C 主机产生的时钟频率及从机模式下所用的时间。	0x04	表 958
CONCLR	WO	0x018	**I2C 控制清除寄存器。**当向该寄存器的位写 1 时，I²C 控制寄存器中的相应位清零。写 0 时对 I²C 控制寄存器的相应位没有影响。	-	表 959
MMCTRL	R/W	0x01C	**监控模式控制寄存器。**	0x00	表 960
ADR1	R/W	0x020	**I2C 从属地址寄存器 1。**包含 7 位从属地址，用于从机模式下 I²C 接口操作，不用于主机模式下。最低有效位确定从机是否响应通用调用地址。	0x00	表 961
ADR2	R/W	0x024	**I2C 从属地址寄存器 2。**包含 7 位从属地址，用于从机模式下 I²C 接口操作，不用于主机模式下。最低有效位确定从机是否响应通用调用地址。	0x00	表 961
ADR3	R/W	0x028	**I2C 从属地址寄存器 3。**包含 7 位从属地址，用于从机模式下 I²C 接口操作，不用于主机模式下。最低有效位确定从机是否响应通用调用地址。	0x00	表 961
DATA_ BUFFER	RO	0x02C	**数据缓冲寄存器。**每次从总线接收到 9 个位（8 位数据和 ACK 或 NACK）后，DAT 移位寄存器的 8 MSB 的内容将自动传输到 DATA_BUFFER。	0x00	表 963

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 952. 寄存器简介：I²C1 （基址 0x400E 0000）*（续）*

名称	访问类型	地址偏移	描述	复位值[1]	参考
MASK0	R/W	0x030	**I2C 从属地址掩码寄存器 0**。该掩码寄存器与 ADR0 共同决定地址匹配。与通用调用地址 (‘0000000’) 相比时，掩码寄存器不产生任何影响。	0x00	表 964
MASK1	R/W	0x034	**I2C 从属地址掩码寄存器 1**。该掩码寄存器与 ADR0 共同决定地址匹配。与通用调用地址 (‘0000000’) 相比时，掩码寄存器不产生任何影响。	0x00	表 964
MASK2	R/W	0x038	**I2C 从属地址掩码寄存器 2**。该掩码寄存器与 ADR0 共同决定地址匹配。与通用调用地址 (‘0000000’) 相比时，掩码寄存器不产生任何影响。	0x00	表 964
MASK3	R/W	0x03C	**I2C 从属地址掩码寄存器 3**。该掩码寄存器与 ADR0 共同决定地址匹配。与通用调用地址 (‘0000000’) 相比时，掩码寄存器不产生任何影响。	0x00	表 964

[1] 复位值仅反映存储在所用位中的数据，不包括保留位的内容。

42.7.1 I²C 控制设置寄存器

CONSET 寄存器控制 CON 寄存器中位的设置，这些位控制 I²C 接口的操作。向该寄存器的位写 1 会使 I²C 控制寄存器中的相应位置位。写入 0 无效。

表 953. I²C控制设置寄存器（CONSET – 地址0x400A 1000 (I2C0)和0x400E 0000 (I2C1)）位描述

位	符号	描述	复位值
1:0	-	保留。用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
2	AA	断言确认标志。	
3	SI	I²C 中断标志。	0
4	STO	停止标志。	0
5	STA	起动标志。	0
6	I2EN	I²C 接口使能。	0
31:7	-	保留。未定义从保留位读取的值。	-

I2EN I²C 接口使能。当 I2EN 为 1 时， I²C 接口使能。可通过将 1 写入 CONCLR 寄存器中的 I2ENC 位将 I2EN 清除。当 I2EN 为 0 时， I²C 接口禁能。

当 I2EN 为“0”时，忽略 SDA 和 SCL 输入信号，I²C 模块处于“不可寻址”的从机状态，STO 位强制为 “0”。

I2EN 不应用于暂时释放 I²C 总线，因为当 I2EN 复位时，I²C 总线状态丢失。应使用 AA 标志代替。

STA 是起始标志。该位置位时，I²C 接口进入主机模式并发送一个起始条件，如果已经处于主机模式，则发送一个重复起始条件。

当 STA 为 1 且 I²C 接口没有处于主机模式时，它将进入主机模式，校验总线并在总线空闲时产生一个起始条件。如果总线不空闲，则其等待 “停止”条件 （这将释放总线），并在延迟内置时钟发生器的半个时钟周期后产生 “起动”条件。当 I²C 接口已经处于主机模式且已发送或接收了数据时，它会发送一个重复起始条件。STA 可在任意时间置位，包括 I²C 接口处于可寻址的从机模式时也可置位。

可通过将 1 写入 CONCLR 寄存器中的 STAC 位将 STA 清除。当 STA 为 0 时，将不会生成任何 “起动”条件或 “重复起动”条件。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

STA 和 STO 都置位时，如果接口处于主机模式下，则向 I²C 总线发送一个停止条件，然后再发送一个起始条件。如果I²C接口处于从模式，则产生内部停止条件，但不发送到总线上。

STO 是停止标志。在主机模式下，该位置位会使 I²C 接口发送一个停止条件，或在从机模式下从错误状态中恢复。当主机模式下 STO=1 时，向 I²C 总线发送停止条件。当总线检测到停止条件时， STO 自动清零。

在从机模式下，设置此位可从错误状态中恢复。在此情况下，没有 “停止”条件传送到总线。硬件的行为就好像已经接收到 “停止”条件并切换到 “未寻址的”从机接收器模式。STO 标志由硬件自动清除。

SI 是 I²C 中断标志。当 I²C 状态改变时该位置位。不过由于进入状态 F8 时，无需中断服务程序处理任何事宜，便不会设置 SI。

SI 处于设置状态时，SCL 线路串行时钟的低电平时间段有延长，且会中止串行传送。当 SCL 为“高电平”时，其不受 SI 标志状态的影响。SI 必须通过软件复位，通过将 1 写入 CONCLR 寄存器的 SIC 位实现。

AA 是应答标志位。当设置为 1 时，在以下情况下，SCL 线上的确认时钟脉冲期间会返回确认 （SDA 为低电平）：

1. 已接收到从属地址寄存器中的地址。
2. 在设置 ADR 中的通用调用位 (GC) 时，已经接收到通用调用地址。
3. 当 I²C 接口处于主接收模式时，接收到一个数据字节。
4. 当 I²C 接口处于可寻址的从机接收模式时，接收到一个数据字节。

可通过将 1 写入 CONCLR 寄存器中的 AAC 位清除 AA 位。当 AA 为 0 时，在以下情况下，SCL 线上的确认时钟脉冲期间会返回确认 （SDA 为高电平）：

1. 当 I²C 接口处于主接收模式时，接收到一个数据字节。
2. 当 I²C 处于可寻址的从机接收模式时，接收到一个数据字节。

42.7.2 I²C 状态寄存器

每个 I²C 状态寄存器反映相应 I²C 接口的情况。 I²C 状态寄存器为只读。

表 954. I²C 状态寄存器 （STAT – 地址 0x400A 1004 (I2C0) 和 0x400E 0004 (I2C1)）位描述

位	符号	描述	复位值
2:0	-	这些位未使用且始终为 0。	0
7:3	状态	这些位提供关于 I²C 接口的实际状态信息。	0x1F
31:8	-	保留。未定义从保留位读取的值。	-

三个最低有效位始终为 0。作为字节时，状态寄存器内容表示状态码。有 26 种可能的状态码。当状态码为 0xF8 时，没有相关信息可用且 SI 位未置位。其它所有 25 个状态码符合定义的 I²C 状态。当进入这些状态中的任一状态时，SI 位将被设置。关于状态码的完整列表，参见表表 969 ～表 974。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

42.7.3 I²C 数据寄存器

此寄存器包含要发送的数据或刚接收的数据。 SI 位置位后，仅在此寄存器没有进行字节移位时 CPU 才可以对其进行读写操作。只要 SI 位进行了置位， DAT 中的数据就保持稳定不变。 DAT 中的数据始终从右向左移位：要发送的第一位是 MSB （位 7），在收到一个字节后，已接收数据的第一位放在 DAT 的 MSB 上。

表 955. I²C 数据寄存器 （DAT － 0x400A 1008 (I2C0) 和 0x400E 0008 (I2C1)）位描述

位	符号	描述	复位值
7:0	数据	此寄存器保留已接收或要发送的数据值。	0
31:8	-	保留。未定义从保留位读取的值。	-

42.7.4 I²C 从机地址寄存器 0

该寄存器可读 / 写，只有在 I²C 接口设置为从机模式时才可用。在主机模式下，此寄存器无效。 ADR 的 LSB 为广播呼叫位。在此位设置后，会识别广播呼叫地址 (0x00)。

如果此寄存器包含 0x00，则 I²C 将不确认总线上的任何地址。复位时此寄存器将被清除到此禁用状态。另请参见表 962。

表 956. I²C 从机地址寄存器 0（ADR0 － 地址 0x400A 100C (I2C0) 和 0x400E 000C (I2C1)）位描述

位	符号	描述	复位值
0	GC	通用调用使能位。	0
7:1	地址	从机模式的 I²C 器件地址。	0x00
31:8	-	保留。未定义从保留位读取的值。	-

42.7.5 I²C SCL 高、低占空比寄存器

表 957. I²C SCL 高电平占空比寄存器（SCLH － 地址 0x400A 1010 (I2C0) 和 0x400E 0010 (I2C1)）位描述

位	符号	描述	复位值
15:0	SCLH	SCL 高电平时段选择的计数。	0x0004
31:16	-	保留。未定义从保留位读取的值。	-

表 958. I²C SCL 低电平占空比寄存器（SCLL － 地址 0x400A 1014 (I2C0) 和 0x400E 0014 (I2C1)）位描述

位	符号	描述	复位值
15:0	SCLL	SCL 低电平时段选择的计数。	0x0004
31:16	-	保留。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

42.7.5.1 选择适当的 I²C 数据速率和占空比

软件必须为寄存器 SCLH 和 SCLL 设置值，以选择适当的数据速率和占空比。 SCLH 定义 SCL 高电平时间的 C_PCLK 周期数，SCLL 定义 SCL 低电平时间 I2C_PCLK 的周期数。频率由下面公式确定 （I2C_PCLK 是外围 I2C 时钟的频率）：

(10)

$$I^2C_{bitfrequency} = \frac{I2CPCLK}{I2CSCLH + I2CSCLL}$$

SCLL 和 SCLH 的值必须确保数据速率在适当的 I²C 数据速率范围内。各寄存器的值必须大于或等于 4。表 959 给出了根据 I2C_PCLK 频率和 SCLL 及 SCLH 值计算出来的 I²C 总线速率的示例。

表 959. 用于所选 I²C 时钟值的 SCLL + SCLH 值

I²C 模式	I²C 位频率	I2C_PCLK (MHz)								
		6	8	10	12	16	20	30	40	50
		SCLH + SCLL								
标准模式	100 kHz	60	80	100	120	160	200	300	400	500
快速模式	400 kHz	15	20	25	30	40	50	75	100	125
超快速模式	1 MHz	-	8	10	12	16	20	30	40	50

SCLL 和 SCLH 值不必相同，软件可通过设置这两个寄存器设置 SCL 的不同占空比。例如，I²C 总线规范定义在快速模式和超快速模式 I²C 下不同值对应的 SCL 低电平时间和高电平时间。

42.7.6 I²C 控制清除寄存器

CONCLR 寄存器控制 CON 寄存器中位的清除，这些位控制 I²C 接口的操作。向该寄存器的位写 1 会使 I²C 控制寄存器中的相应位清零。写入 0 无效。

表 960. I²C 控制清除寄存器 （CONCLR – 地址 0x400A 1018 和 0x400E 0018 (I2C1)）位描述

位	符号	描述	复位值
1:0	-	保留。用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
2	AAC	断言确认清除位。	
3	SIC	I²C 中断清零位。	0
4	-	保留。用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
5	STAC	START 标志清除位。	0
6	I2ENC	I²C 接口禁能位。	0
7	-	保留。用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
31:8	-	保留。未定义从保留位读取的值。	-

AAC 是声明应答清零位。将 1 写入此位会清除 CONSET 寄存器中的 AA 位。写入 0 无效。

SIC 是 I²C 中断清零位。将 1 写入此位会清除 CONSET 寄存器中的 SI 位。写入 0 无效。

STAC 是起始标志清零位。将 1 写入此位会清除 CONSET 寄存器中的 STA 位。写入 0 无效。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

I2ENC 是 I²C 接口禁能位。将 1 写入此位会清除 CONSET 寄存器中的 I2EN 位。写入 0 无效。

42.7.7 I²C 监控模式控制寄存器

该寄存器控制监控模式，监控模式允许 I²C 块在不需实际参与通信或干扰 I²C 总线的情况下监控 I²C 总线上的流量。

表 961. I²C 监控模式控制寄存器（MMCTRL – 地址 0x400A 101C (I2C0) 和 0x400E 001C (I2C1)）位描述

位	符号	值	描述	复位值
0	MM_ENA		监控模式使能。	0
		0	监控模式禁用。	
		1	I²C 块将进入监控模式。在此模式下， SDA 输出将强制为高电平。这可防止 I²C 块向 I²C 数据总线输出任何类型的数据（包括 ACK） 根据 ENA_SCL 位状态，也可以将输出强制为高电平，以防止模块控制 I²C 时钟线。	
1	ENA_SCL		SCL 输出使能。	0
		0	如果此位被清除为 ‘0’ ，则当模块处于监控模式时将强制 SCL 输出为高电平。如上所述，这可防止模块控制 I²C 时钟线。	
		1	该位置位时， I²C 块将以与正常操作中相同的方法控制时钟线。这意味着，作为从机设备，I²C 模块可 “延长”时钟线（使其为低电平），直到它有时间响应 I²C 中断为止 [1]	
2	MATCH_ALL		选择中断寄存器匹配。	0
		0	如果此位被清除，则只有当四个 （最多）上述地址寄存器中之一出现匹配时，才会生成中断。也就是说，就地址识别而言模块会作为普通从机响应。	
		1	此位设为 ‘1’ 且 I2C 处于监控模式时，收到任意地址时均会产生中断。将使该器件可以监控总线上的所有流量。	
31:3	-	-	保留。未定义从保留位读取的值。	-

[1] 当 ENA_SCL 位清除且 I²C 不能再延迟总线时，中断响应时间就变得很重要。为了使器件在这些情况下能有更多时间对 I²C 中断作出响应，就需要使用 DATA_BUFFER 寄存器 （第 42.7.9 节）来保存接收到的数据，保存时间为发送完一个 9 位字的时间。

注：如果 MM_ENA 为‘0’（即，如果模块不处于监控模式），则 ENA_SCL 和 MATCH_ALL 位无效。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

42.7.7.1 监控模式的中断

模块处于监控模式时，所有中断都将正常发生。这意味着第一个中断会在检测到地址匹配时发生（如果已设置 MATCH_ALL 位，则接收的任何地址都会发生中断，否则只有在地址与四个地址寄存器中之一匹配时才会发生中断）。

地址匹配检测后，对于从机写传输，在接收每个字节后都会产生中断，对于从机读传输，则在模块“认为”已发送每个字节后产生中断。在此第二种情况下，数据寄存器实际上包含了总线上其它从机发送的数据，该从机实际上是由主机寻址的。

在所有这些中断发生后，处理器可读取数据寄存器以查看总线上实际传送的数据。

42.7.7.2 监控模式中的仲裁丢失

在监控模式下，I²C 块不能响应总线主机的信息请求或发布应答，而是由总线上的其它从机响应。就我们的模块而言，这很可能会导致丢失仲裁的状态。

软件应当意识到，模块处于监控模式且不应响应检测到的任何仲裁状态丢失。

42.7.8 I²C 从机地址寄存器

这些寄存器可读 / 写，只有在 I²C 接口设置为从机模式时才可用。在主机模式下，此寄存器无效。 ADR 的 LSB 为广播呼叫位。在此位设置后，会识别广播呼叫地址 (0x00)。

如果这些寄存器包含 0x00，则 I²C 将不确认总线上的任何地址。复位时所有四个寄存器（包括 ADR0，参见表 956）都将清除到此禁用状态。

表 962. I²C 从机地址寄存器 （ADR – 地址 0x400A 1020 (ADR1) 到 0x400A 1028 (ADR3) (I2C0) 和 0x400E 0020 (ADR1) 到 0x400E 0028 (ADR3) (I2C1)）位描述

位	符号	描述	复位值
0	GC	通用调用使能位。	0
7:1	地址	从机模式的 I²C 器件地址。	0x00
31:8	-	保留。未定义从保留位读取的值。	-

42.7.9 I²C 数据缓冲寄存器

在监控模式下，如果 ENA_SCL 没有置位，则 I²C 块就不能延长时钟（使总线延迟）。这意味着处理器读取总线接收数据内容的时间有限。如果处理器读 DAT 移位寄存器，则在接收的数据被新数据覆写之前，它如平常一样只有一位时间响应中断。

为了给处理器提供更多时间来响应，将增加一个新的 8 位只读 DATA_BUFFER 寄存器。在总线上每接收九个位（8 位数据加 ACK 或 NACK）后，会自动将 DAT 移位寄存器的 8 MSB 内容传输到 DATA_BUFFER，这意味着处理器有九位传输时间响应中断并在数据被覆写前读取。

处理器仍能像往常一样直接读 DAT，并且 DAT 的行为不会有任何改变。

尽管 DATA_BUFFER 寄存器主要用于 ENA_SCL 位 = ‘0’的监控模式，但它也可用于在任何操作模式下随时读取。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 963. I²C 数据缓冲寄存器（DATA_BUFFER – 地址 0x400A 102C (I2C0) 和 0x400E 002C (I2C1)）位描述

位	符号	描述	复位值
7:0	数据	此寄存器保留 DAT 移位寄存器的 8 MSB 内容。	0
31:8	-	保留。未定义从保留位读取的值。	-

42.7.10 I²C 掩码寄存器

四个掩码寄存器每个都包含七个有效位 (7:1)。这些掩码寄存器与之关联的 ADRn 寄存器相比，其设置为 ‘1’ 的任何位都会导致已接收地址的相应位上的自动比较。换句话说，决定地址匹配时不考虑 ADRn 寄存器中被掩码的位。

复位时，所有掩码寄存器位都被清除为 ‘0’。

掩码寄存器对与通用调用地址 (“0000000”) 的比较不产生任何影响。

掩码寄存器的位 (31:8) 和位 (0) 未使用且不应写入。读取这些位时将始终返回零。

发生地址匹配中断时，处理器必须读数据寄存器 (DAT) 以确定实际导致匹配的已接收地址。

表 964. I²C 掩码寄存器（MASK – 地址 0x400A 1030 (MASK0) 到 0x400A 103C (MASK3) (I2C0) 和 0x400E 0030 (MASK0) 到 0x400E 003C (MASK3) (I2C1)）位描述

位	符号	描述	复位值
0	-	保留。用户软件不应向保留位写入 1。读取此位时始终返回零。	0
7:1	MASK	掩码位。	0x00
31:8	-	保留。未定义从保留位读取的值。	-

42.8 I²C 操作模式

在给定的应用中，I²C 块可作为主机、从机或同时作主机和从机。在从机模式，I²C 硬件查找其 4 个从属地址中的任何一个地址及通用调用地址。如果检测到这些地址之一，则请求中断。如果处理器希望成为总线主机，则在进入主机模式前，硬件将一直等待，直到总线空闲，这样就不会中断可能的从机操作。如果在总机模式下丢失总线仲裁，则 I²C 模块将立即切换到从机模式并在同一串行传输中检测自身的从机地址。

42.8.1 主发送器模式

此模式下，数据从主机发送到从机。在进入主发送模式前，必须按表 965 所示初始化 CONSET 寄存器。I2EN 必须置 1 以使能 I²C 功能。如果 AA 位为 0，则当另一个器件为总线上的主机时，I²C 接口不会对任何地址作出应答，因此不能进入从机模式。STA、STO 和 SI 位必须为 0。通过将 1 写入 CONCLR 寄存器中的 SIC 位将 SI 位清除。写从属地址后应清除 STA 位。

表 965. 用于配置主机模式的 CONSET

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	0	-	-

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

发送的第一个字节包含接收设备的从属地址 （7 位）和数据方向位。在此模式下，数据方向位 (R/W) 应为 0，表示 “写入”。发送的第一个字节包含从属地址和 “写” 位。一次发送 8 位数据。每发送完一个字节后，接收一个确认位。输出 “起动” 和 “停止” 条件指示串行传输的开始和结束。

转件置位 STA 位时，I²C 接口将进入主发送模式。一旦总线空闲，I²C 逻辑就会发送起始条件。发送 “起动” 条件后， SI 位置位， STAT 寄存器中的状态码为 0x08。此状态码用于引导状态服务例程，该例程将从属地址和 “写” 位加载到 DAT 寄存器，然后清除 SI 位。通过将 1 写入到 CONCLR 寄存器中的 SIC 位清除 SI。

当发送完从属地址和 R/W 位并接收确认位后，SI 位再次设置，此时主机模式下可能的状态码为 0x18、0x20 或 0x38，或者如果使能了从属模式（将 AA 设置为 1），则为 0x68、0x78 或 0xB0。各状态码的相应操作见表 969 ～表 974。

图 156. 主发送器模式中的格式

42.8.2 主接收器模式

在主机接收器模式下，从从机发送器接收数据。传输的发起方式与在主机发送器模式下相同。发送完起始条件后，中断服务程序必须将从属地址和数据方向位装入 I²C 数据寄存器 (DAT)，然后清零 SI 位。在此情况下，数据方向位 (R/W) 应为 1 以表示 “读取”。

发送完从属地址和数据方向位并接收到确认位后，SI 位被设置，状态寄存器将显示状态码。对于主机模式，可能的状态码为 0x40、 0x48 或 0x38。对于从机模式，可能的状态码为 0x68、 0x78 或 0xB0。更多详细信息，请参阅表 970。

图 157. 主接收器模式的格式

经过一个重复起始条件后， I²C 可切换到主发送模式。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 158. 发送重复起始信号后，主接收器切换至主发送器

42.8.3 从属接收器模式

在从机接收器模式下，从主机发送器接收数据字节。要初始化从机接收器模式，对任一从机地址寄存器 (ADR0-3) 进行写操作并写 I²C 控制置位寄存器 (CONSET)，如表 966 所示。

表 966. 用于配置从机模式的 CONSET

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	1	-	-

I2EN 必须置 1 以使能 I²C 功能。 AA 位必须设置为 1 以确认其自身的从属地址或通用调用地址。 STA、 STO 和 SI 位都设置为 0。

初始化 ADR 和 CONSET 后，I²C 接口开始等待，直到被其自身地址或后跟数据方向位的通用地址寻址为止。如果方向位为 0 (W)，则其进入从机接收器模式。如果方向位为 1 (R)，则其进入从机发送器模式。接收到地址和方向位后， SI 位设置，并可从状态寄存器 (STAT) 读取一个有效状态码。关于状态码和操作请见表 973。

图 159. 从属接收器模式的格式

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

42.8.4 从属发送器模式

接收和处理第一个字节的方式与从机接收器模式下相同。但在此模式下，方向位为 1，指示读操作。串行数据通过 SDA 发送，同时通过 SCL 输入串行时钟。“起动”和“停止”条件被识别为串行传输的开始和结束。在特定应用中， I^2C 可作为主机 / 从机。在从机模式下，I^2C 硬件查找其自身的从机地址和通用调用地址。如果检测到这些地址之一，则请求中断。当微控制器希望成为总线主机时，在进入主机模式前，硬件将一直等待，直到总线空闲，这样就不会中断可能的从机操作。如果在总机模式下丢失总线仲裁，则 I^2C 接口将立即切换到从机模式并在同一串行传输中检测自身的从机地址。

图 160. 从属发送器模式的格式

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

42.9 I²C 执行和操作

图 161 所示为片内 I²C 总线接口的执行流程，下面章节将对图中各模块进行描述。

图 161. I²C 串行接口功能框图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

42.9.1 输入滤波器与输出级

输入信号与内部时钟同步，小于三个时钟的峰值将被滤出。

I^2C 输出是一个特殊管脚，是为符合 I^2C 规范而设计的。

42.9.2 地址寄存器 ADR0 ～ ADR3

当作为从发送器或接收器时，这些寄存器可装入 7 位从属地址（7 个最高有效位），I^2C 块将对这些地址作出响应。LSB (GC) 用于使能通用调用地址 (0x00) 识别。使能了多个从机地址后，且已接收到自己的从机地址时，接收的实际地址可从 DAT 寄存器读取。

42.9.3 地址掩码寄存器， MASK0 ～ MASK3

四个掩码寄存器每个都包含七个有效位 (7:1)。这些掩码寄存器与之关联的 ADRn 寄存器相比，其设置为‘1’的任何位都会导致已接收地址的相应位上的自动比较。换句话说，决定地址匹配时不考虑 ADRn 寄存器中被掩码的位。

发生地址匹配中断时，处理器必须读数据寄存器 (I2DAT) 以确定实际导致匹配的接收地址。

42.9.4 比较器

比较器将接收到的 7 位从属地址与其自身的从属地址（ADR 中的 7 个最高有效位）进行比较，它还将第一次收到的 8 位字节与通用调用地址 (0x00) 进行比较。如果发现相等，则设置相应的状态位并请求中断。

42.9.5 移位寄存器， DAT

此 8 位寄存器包含一个要发送的串行数据字节或一个刚接收到的字节。 DAT 中的数据始终从右向左移位。要发送的第一位是 MSB （位 7），在收到一个字节后，已接收数据的第一位放在 DAT 的 MSB 上。当数据被移出时，总线上的数据同时移入； DAT 始终包含总线上出现的最后一个字节。因此，在仲裁丢失的情况下，会用 DAT 中的正确数据进行从主机发送器到从机接收器的跳变。

42.9.6 仲裁与同步逻辑

在主发送模式下，仲裁逻辑校验每个发送的逻辑 1 在 I^2C 总线上是否真正以逻辑 1 出现。如果总线上另一个器件否定逻辑 1 并将 SDA 线拉低，则仲裁丢失， I^2C 块立即由主发送器转换成从接收器。I^2C 块将继续输出时钟脉冲（在 SCL 上），直到发送完当前串行字节为止。

主机接收器模式下也可能丢失仲裁。在该模式下，只有在 I^2C 模块向总线返回一个 “无应答”：（逻辑 1）才会发生仲裁丢失。当总线上另一设备将此信号拉低时，仲裁丢失。由于这只会在串行字节结束时出现，因此 I^2C 块不再产生时钟脉冲。图 162 所示为仲裁过程。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

(1) 另一设备发送串行数据。

(2) 另一个器件通过将 SDA 线拉低撤消了该 I²C 主机发送的一个逻辑 （虚线）。仲裁丢失，该 I²C 进入从接收模式。

(3) 该 I²C 为从接收模式，但仍产生时钟脉冲，直到当前字节发送完为止。该 I²C 不会为下一个字节产生时钟脉冲。赢得仲裁后， SDA 上的数据由新主机产生。

图 162. 仲裁过程

同步逻辑将使串行时钟发生器与来自另一设备的 SCL 线上的时钟脉冲同步。如果两个或多个主机设备产生时钟脉冲，则标记持续时间由产生最短标记的设备确定，间隔持续时间由产生最长间隔的设备确定。图 163 所示为同步过程。

(1) 另一个器件在 I²C 定时一个完整的高电平时间前将 SCL 线拉低。其它设备有效地确定了 （较短）高电平期间。

(2) 另一个器件在 I²C 定时一个完整的低电平时间并释放 SCL 后继续将 SCL 线拉低。I²C 时钟发生器必须等待，直到 SCL 变为高电平。其它设备有效地确定了 （较长）低电平期间。

(3) SCL 线被释放，时钟发生器开始定时高电平时间。

图 163. 串行时钟同步

从机可延长间隔持续时间以减慢总线主机，也可以延长间隔持续时间以实现信号握手目的。可在每位或一个完整字节传输后延长低电平时间。发送或接收到一个字节后，I²C 块将延长 SCL 低电平时间且已发送应答位。串行中断标志 (SI) 被设置，且延长继续进行直到串行中断标志被清除。

42.9.7 串行时钟生成器

当 I²C 块处于主发送或主接收模式时，可编程时钟脉冲发生器提供 SCL 时钟脉冲。当 I²C 块处于从机模式时，时钟脉冲发生器关闭。I²C 输出时钟频率和占空比可通过 I²C 时钟控制寄存器编程。详情请参见关于 SCLL 和 SCLH 寄存器的描述。除非总线与上述其它 SCL 时钟源同步，输出时钟脉冲的占空比均会按照预先编程确定的情况出现。

42.9.8 时序与控制

定时和控制逻辑为串行字节处理产生定时和控制信号。该逻辑块为 DAT 提供移位脉冲，可使能比较器、产生并检测起始和停止条件、接收并发送应答位、控制主 / 从机模式，还包含中断请求逻辑并监控 I²C 总线状态。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

42.9.9 控制寄存器 ICONSET 和 CONCLR

I^2C 控制寄存器包含用于控制以下 I^2C 块功能的位：串行传输的起动和重启、串行传输终止、比特率、地址识别及确认。

I^2C 控制寄存器的内容可能读出为 CONSET。写 CONSET 可置位 I^2C 控制寄存器中的相应位。反之，写 CONCLR 将清零 I^2C 控制寄存器中的相应位。

42.9.10 状态解码器与状态寄存器

状态解码器获取所有内部状态位并将其压缩成 5 位代码，该代码与各 I^2C 总线状态一一对应。该 5 位代码可用于产生向量地址，以快速处理各种服务例程。每个服务例程处理一个特定的总线状态。如果使用 I^2C 块的所有 4 种模式，则存在 26 种可能的总线状态。串行中断标志设置（通过硬件）后，该 5 位状态码被锁存到状态寄存器的 5 个最高有效位并保持稳定，直到中断标志被软件清除。状态寄存器的三个最低有效位始终为零。如果状态码用作服务例程的向量，则这些例程由八个地址位置替代。对于大多数的服务例程，八个字节的代码足够（参见本节的软件示例）。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

42.10 I²C 操作模式详解

四种操作模式为：

- 主机发送器
- 主机接收器
- 从机接收器
- 从机发送器

各模式下数据传输操作如图 164、图 165、图 166、图 167 和图 168 所示。表 967 说明了介绍 I²C 操作模式的图中所使用缩写的含义。

表 967. 用于描述 I²C 操作的缩写

缩写	描述
S	起始条件
SLA	7 位从属地址
R	读取位 （SDA 为高电平）
W	写入位 （SDA 低电平）
A	应答位 （SDA 低电平）
$\overline{A}$	未确认位 （SDA 为高电平）
数据	8 位数据字节
P	“停止”条件

在图 164 ～图 168 中，圆圈用来指示串行中断标志何时被置位。圆圈中的数字显示 STAT 寄存器中保留的状态码。在这些点，必须执行服务例程以继续或完成串行传输。这些服务例程不是至关重要的，因为串行传输被挂起，直到串行中断标志被软件清除。

当进入串行中断例程时， STAT 中的状态码用于分支到相应的服务例程。对于每个状态代码，需要的软件操作以及后面串行传输的详细情况见表 969 ～表 975。

42.10.1 主发送器模式

在主发送模式中，向从接收器发送数据字节 （见图 164）。 CON 必须按如下所述进行初始化，然后才能进入主机发送器模式：

表 968. 用于初始化主发送模式的 CONSET

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	x	-	-

I²C 速率也必须在 I2SCLL 和 I2SCLH 寄存器中配置。必须将 I2EN 设置为逻辑 1 来使能 I²C 块。如果 AA 位复位，则当另一个器件正变成总线主机时， I²C 模块将不会应答其自身的从机地址或通用调用地址。也就是说，如果 AA 位复位，则 I²C 接口就不能进入从机模式。STA、 STO 和 SI 必须复位。

现在即可通过设置 STA 位进入主机发送器模式。一旦总线空闲，I²C 逻辑会立即测试 I²C 总线并产生一个起始条件。当 “起动”条件发送后，串行中断标志 (SI) 设置，状态寄存器 (STAT) 中的状态码为 0x08。中断服务例程使用此状态码进入相应的状态服务例程，该例程将从属地址和数据方向位 (SLA+W) 载入 DAT。随后必须复位 CON 中的 SI 位，之后才能继续串行传输。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

从属地址和方向位发送完且接收到确认位后，串行中断标志 (SI) 再次置位，STAT 中可能存在许多状态码。主机模式下有 0x18、0x20 或 0x38，如果使能了从机模式（AA= 逻辑 1），则有 0x68、0x78 或 0xB0。表 969 中详细介绍了每个状态代码对应的操作。在发送完重复起始条件 （状态 0x10）后， I²C 块通过将 SLA+R 装入 DAT 切换到主接收模式。

表 969. 主发送器模式

状态码 (STAT)	I²C 总线和硬件的状态	应用软件响应					I²C 硬件执行的下一个操作
		至 / 自 DAT	至 CON				
			STA	STO	SI	AA	
0x08	“起动”条件已发送。	加载 SLA+W ；清除 STA	X	0	0	X	将发送 SLA+W ；将接收 ACK 位。
0x10	“重复起动”条件已发送。	加载 SLA+W 或	X	0	0	X	同上。
		加载 SLA+R ；清除 STA	X	0	0	X	将发送 SLA+R ； I²C 模块将切换为 MST/REC 模式。
0x18	SLA+W 已发送；ACK 已接收。	加载数据字节或	0	0	0	X	将发送数据字节；将接收 ACK 位。
		无 DAT 操作或	1	0	0	X	将发送 “重复起动”。
		无 DAT 操作或	0	1	0	X	将发送 “停止”条件； STO 标志将复位。
		无 DAT 操作	1	1	0	X	“停止”条件之后将发送 “起动”条件；STO 标志将复位。
0x20	已发送 SLA+W ； 已接收 NOT ACK。	加载数据字节或	0	0	0	X	将发送数据字节；将接收 ACK 位。
		无 DAT 操作或	1	0	0	X	将发送 “重复起动”。
		无 DAT 操作或	0	1	0	X	将发送 “停止”条件； STO 标志将复位。
		无 DAT 操作	1	1	0	X	“停止”条件之后将发送 “起动”条件；STO 标志将复位。
0x28	DAT 中的数据字节已发送； ACK 已接收。	加载数据字节或	0	0	0	X	将发送数据字节；将接收 ACK 位。
		无 DAT 操作或	1	0	0	X	将发送 “重复起动”。
		无 DAT 操作或	0	1	0	X	将发送 “停止”条件； STO 标志将复位。
		无 DAT 操作	1	1	0	X	“停止”条件之后将发送 “起动”条件；STO 标志将复位。
0x30	DAT 中的数据字节已发送； NOT ACK 已接收。	加载数据字节或	0	0	0	X	将发送数据字节；将接收 ACK 位。
		无 DAT 操作或	1	0	0	X	将发送 “重复起动”。
		无 DAT 操作或	0	1	0	X	将发送 “停止”条件； STO 标志将复位。
		无 DAT 操作	1	1	0	X	“停止”条件之后将发送 “起动”条件；STO 标志将复位。
0x38	SLA+R/W 或数据字节操作中仲裁丢失。	无 DAT 操作或	0	0	0	X	I²C 总线将被释放；进入不可寻址从机模式。
		无 DAT 操作	1	0	0	X	当总线空闲时将发送 “起动”条件。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 164. 主发送模式下的格式和状态

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

42.10.2 主接收器模式

在主接收模式中，主机所接收的数据字节来自从发送器（见图 165）。传输的初始化与主机发送器模式下相同。发送完 “起动”条件后，中断服务例程必须将 7 位从属地址和数据方向位 (SLA+R) 载入 DAT 中。随后必须清除 CON 中的 SI 位，之后才能继续串行传输。

从属地址和数据方向位发送完且接收到确认位后，串行中断标志 (SI) 再次置位，STAT 中可能存在许多状态码。主机模式下为 0x40、0x48 或 0x38，从机模式 (AA = 1) 为 0x68、0x78 或 0xB0。表 970 中详细介绍了每个状态代码对应的操作。在发送完重复起始条件 （状态 0x10）后， I²C 模块通过将 SLA+W 装入 DAT 切换到主机发送器模式。

表 970. 主接收器模式

状态码 (STAT)	I²C 总线和硬件的状态	应用软件响应					I²C 硬件执行的下一个操作
		至 / 自 DAT	至 CON				
			STA	STO	SI	AA	
0x08	“起动”条件已发送。	加载 SLA+R	X	0	0	X	将发送 SLA+R ；将接收 ACK 位。
0x10	“重复起动”条件已发送。	加载 SLA+R 或	X	0	0	X	同上。
		加载 SLA+W	X	0	0	X	将发送 SLA+W ； I²C 块将切换为 MST/TRX 模式。
0x38	在 NOT ACK 位中丢失仲裁。	无 DAT 操作或	0	0	0	X	I²C 总线将被释放； I²C 模块进入从机模式。
		无 DAT 操作	1	0	0	X	当总线空闲时将发送 “起动”条件。
0x40	SLA+R 已发送； ACK 已接收。	无 DAT 操作或	0	0	0	0	将接收数据字节；将返回 NOT ACK 位。
		无 DAT 操作	0	0	0	1	将接收数据字节；将返回 ACK 位。
0x48	SLA+R 已发送； NOT ACK 已接收。	无 DAT 操作或	1	0	0	X	将发送 “重复起动”条件。
		无 DAT 操作或	0	1	0	X	将发送 “停止”条件； STO 标志将复位。
		无 DAT 操作	1	1	0	X	“停止”条件之后将发送 “起动”条件；STO 标志将复位。
0x50	数据字节已接收；ACK 已返回。	读取数据字节或	0	0	0	0	将接收数据字节；将返回 NOT ACK 位。
		读取数据字节	0	0	0	1	将接收数据字节；将返回 ACK 位。
0x58	数据字节已接收；NOT ACK 已返回。	读取数据字节或	1	0	0	X	将发送 “重复起动”条件。
		读取数据字节或	0	1	0	X	将发送 “停止”条件； STO 标志将复位。
		读取数据字节	1	1	0	X	“停止”条件之后将发送 “起动”条件；STO 标志将复位。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 165. 主接收模式下的格式和状态

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

42.10.3 从属接收器模式

在从接收模式中，从机所接收的数据字节来自主发送器（见图 166）。要初始化从机接收器模式，必须按如下所示加载 ADR 和 CON：

表 971. 从机接收器模式下的 ADR 使用

位	7	6	5	4	3	2	1	0
符号	自身的 7 位从属地址							GC

高 7 位是主机寻址时 I²C 模块将要响应的地址。如果设置了 LSB (GC)，则 I²C 模块将响应通用调用地址 (0x00)；否则它将忽略该通用调用地址。

表 972. 用于初始化从接收模式的 CONSET

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	1	-	-

I²C 总线速率的设置不影响从机模式中的 I²C 块。必须将 I2EN 设置为逻辑 1 来使能 I²C 块。AA 位必须置位以使能 I²C 块来应答其自身从属地址或通用调用地址。 STA、 STO 和 SI 必须复位。

当初始化 ADR 和 CON 后，I²C 模块一直等待，直至被自身的从机地址寻址，之后是数据方向位，该数据方向位必须为 “0” (W)，以便 I²C 模块在从机接收器模式下工作。接收到其自身的从属地址和 W 位后，将设置串行中断标记 (SI)，并可从 STAT 读取一个有效状态码。此状态码用作状态服务例程的向量。表 973 中详细介绍了每个状态代码对应的操作。如果当 I²C 块在主机模式中时仲裁丢失，也可进入从接收模式 （见状态 0x68 和 0x78）。

如果 AA 位在传输过程中复位，则在接收完下一个数据字节后 I²C 块将向 SDA 返回一个非应答 （逻辑 1）。当 AA 复位时， I²C 块不响应其自身的从属地址或通用调用地址。但是，I²C 总线仍被监控，而且，地址识别可随时通过置位 AA 来恢复。这就意味着 AA 位可临时将 I²C 块从 I²C 总线上分离出来。

表 973. 从机接收器模式

状态码 (STAT)	I²C 总线和硬件的状态	应用软件响应					I²C 硬件执行的下一个操作
		至 / 自 DAT	至 CON				
			STA	STO	SI	AA	
0x60	已接收自身的 SLA+W，已返回 ACK。	无 DAT 操作或	X	0	0	0	将接收数据字节并返回 NOT ACK。
		无 DAT 操作	X	0	0	1	将接收数据字节并返回 ACK。
0x68	作为主机，在 SLA+R/W 中仲裁丢失；自身的 SLA+W 已接收， ACK 已返回。	无 DAT 操作或	X	0	0	0	将接收数据字节并返回 NOT ACK。
		无 DAT 操作	X	0	0	1	将接收数据字节并返回 ACK。
0x70	通用调用地址 (0x00) 已接收； ACK 已返回。	无 DAT 操作或	X	0	0	0	将接收数据字节并返回 NOT ACK。
		无 DAT 操作	X	0	0	1	将接收数据字节并返回 ACK。
0x78	作为主机，在 SLA+R/W 中仲裁丢失；通用调用地址已接收， ACK 已返回。	无 DAT 操作或	X	0	0	0	将接收数据字节并返回 NOT ACK。
		无 DAT 操作	X	0	0	1	将接收数据字节并返回 ACK。
0x80	先前已使用自身的 SLV 地址进行寻址； DATA 已接收； ACK 已返回。	读取数据字节或	X	0	0	0	将接收数据字节并返回 NOT ACK。
		读取数据字节	X	0	0	1	将接收数据字节并返回 ACK。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 973. 从机接收器模式 *（续）*

状态码 (STAT)	I²C 总线和硬件的状态	应用软件响应 至 / 自 DAT	至 CON STA	STO	SI	AA	I²C 硬件执行的下一个操作
0x88	先前已使用自身的 SLA 进行寻址； DATA 字节已接收； NOT ACK 已返回。	读取数据字节或	0	0	0	0	切换到未寻址的 SLV 模式；不识别自身 SLA 或通用调用地址。
		读取数据字节或	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA ；如果 ADR[0] = 逻辑 1，将识别通用调用地址。
		读取数据字节或	1	0	0	0	切换到未寻址的 SLV 模式；不识别自身 SLA 或通用调用地址。当总线空闲时将发送 “起动”条件。
		读取数据字节	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA ；如果 ADR[0] = 逻辑 1，将识别通用调用地址；当总线空闲后发送起始条件。
0x90	先前已使用通用调用进行寻址； DATA 字节已接收； ACK 已返回。	读取数据字节或	X	0	0	0	将接收数据字节并返回 NOT ACK。
		读取数据字节	X	0	0	1	将接收数据字节并返回 ACK。
0x98	先前已使用通用调用进行寻址； DATA 字节已接收； NOT ACK 已返回。	读取数据字节或	0	0	0	0	切换到未寻址的 SLV 模式；不识别自身 SLA 或通用调用地址。
		读取数据字节或	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA ；如果 ADR[0] = 逻辑 1，将识别通用调用地址。
		读取数据字节或	1	0	0	0	切换到未寻址的 SLV 模式；不识别自身 SLA 或通用调用地址。当总线空闲时将发送 “起动”条件。
		读取数据字节	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA ；如果 ADR[0] = 逻辑 1，将识别通用调用地址；当总线空闲后发送起始条件。
0xA0	当作为 SLV/REC 或 SLV/TRX 静态寻址时，已接收 “停止”条件或 “重复起动”条件。	无 STDAT 操作或	0	0	0	0	切换到未寻址的 SLV 模式；不识别自身 SLA 或通用调用地址。
		无 STDAT 操作或	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA ；如果 ADR[0] = 逻辑 1，将识别通用调用地址。
		无 STDAT 操作或	1	0	0	0	切换到未寻址的 SLV 模式；不识别自身 SLA 或通用调用地址。当总线空闲时将发送 “起动”条件。
		无 STDAT 操作	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA ；如果 ADR[0] = 逻辑 1，将识别通用调用地址；当总线空闲后发送起始条件。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

图 166. 从机接收模式下的格式和状态

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

42.10.4 从属发送器模式

在从发送模式中，向主接收器发送数据字节（见图 167）。数据传输的初始化与从机接收器模式下相同。当初始化 ADR 和 CON 后，I²C 块一直等待，直至被自身的从机地址寻址，之后是数据方向位，该数据方向位必须为“1”(R)，以便 I²C 模块在从机发送器模式下工作。接收到其自身的从属地址和 R 位后，将设置串行中断标记 (SI)，并可从 STAT 读取一个有效状态码。该状态代码用作状态服务程序的向量，每个状态代码的对应操作详见表 974。如果当 I²C 块在主机模式中时仲裁丢失，也可进入从发送模式（见状态 0xB0）。

如果 AA 位在传输过程中复位，则 I²C 块将发送最后一个字节并进入状态 0xC0 或 0xC8。I²C 块切换到非寻址的从机模式，如果继续传输，它将忽略主接收器。从而主机接收器作为串行数据接收所有 1。当 AA 复位时， I²C 块不响应其自身的从属地址或通用调用地址。但是， I²C 总线仍被监控，而且，地址识别可随时通过置位 AA 来恢复。这就意味着 AA 位可临时将 I²C 块从 I²C 总线上分离出来。

表 974. 从属发送器模式

状态码 (STAT)	I²C 总线和硬件的状态	应用软件响应					I²C 硬件执行的下一个操作
		至 / 自 DAT	至 CON				
			STA	STO	SI	AA	
0xA8	自身的 SLA+R 已接收；ACK 已返回。	加载数据字节或	X	0	0	0	将发送最后一个数据字节并接收 ACK 位。
		加载数据字节	X	0	0	1	将发送数据字节；将接收 ACK。
0xB0	作为主机，在 SLA+R/W 中仲裁丢失；自身的 SLA+R 已接收，ACK 已返回。	加载数据字节或	X	0	0	0	将发送最后一个数据字节并接收 ACK 位。
		加载数据字节	X	0	0	1	将发送数据字节；将接收 ACK 位。
0xB8	DAT 中的数据字节已发送； ACK 已接收。	加载数据字节或	X	0	0	0	将发送最后一个数据字节并接收 ACK 位。
		加载数据字节	X	0	0	1	将发送数据字节；将接收 ACK 位。
0xC0	DAT 中的数据字节已发送； NOT ACK 已接收。	无 DAT 操作或	0	0	0	0	切换到未寻址的 SLV 模式；不识别自身 SLA 或通用调用地址。
		无 DAT 操作或	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA ；如果 ADR[0] = 逻辑 1，将识别通用调用地址。
		无 DAT 操作或	1	0	0	0	切换到未寻址的 SLV 模式；不识别自身 SLA 或通用调用地址。当总线空闲时将发送 “起动”条件。
		无 DAT 操作	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA ；如果 ADR[0] = 逻辑 1，将识别通用调用地址；当总线空闲后发送起始条件。
0xC8	DAT 中最后一个数据字节已发送 (AA = 0)；ACK 已接收。	无 DAT 操作或	0	0	0	0	切换到未寻址的 SLV 模式；不识别自身 SLA 或通用调用地址。
		无 DAT 操作或	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA ；如果 ADR[0] = 逻辑 1，将识别通用调用地址。
		无 DAT 操作或	1	0	0	0	切换到未寻址的 SLV 模式；不识别自身 SLA 或通用调用地址。当总线空闲时将发送 “起动”条件。
		无 DAT 操作	1	0	0	01	切换到不可寻址 SLV 模式；识别自身 SLA ；如果 ADR.0 = 逻辑 1，将识别通用调用地址；当总线空闲后发送起始条件。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 167. 从发送模式下的格式和状态

42.10.5 其它状态

还有两种 STAT 代码与已定义的 I²C 硬件状态不对应 （见表 975）。论述如下。

42.10.5.1 STAT = 0xF8

此状态码表示没有任何可用的相关信息，因为串行中断标记 SI 尚未置位。这种情况在其它状态和 I²C 块还未开始执行串行传输之间出现。

42.10.5.2 STAT = 0x00

该状态代码表示在 I²C 串行传输过程中出现了总线错误。当 “起动” 或 “停止” 条件发生在格式帧的非法位置上时会导致总线错误。此类非法位置的示例有在地址字节、数据字节或确认位的串行传输期间。当外部干扰影响到内部 I²C 块信号时也会产生总线错误。发生总线错误时 SI 设置。要从总线错误中恢复， STO 标志必须设置且 SI 必须清除。这使得 I²C 模块进入 “非寻址” 的从机模式 （已定义的状态）并清除 STO 标志 （CON 中的其他位不受影响）。 SDA 和 SCL 线被释放 （不发送 “停止” 条件）。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 975. 其它状态

状态码 (STAT)	I²C 总线和硬件的状态	应用软件响应					I²C 硬件执行的下一个操作
		至 / 自 DAT	至 CON				
			STA	STO	SI	AA	
0xF8	无可用的相关状态信息； SI = 0。	无 DAT 操作	无 CON 操作				等待或继续进行当前传输。
0x00	由于非法的 “起动” 或 “停止” 条件，在 MST 或所选的从机模式期间总线出错。当外部干扰使 I²C 块进入未定义的状态时也出现 0x00 状态。	无 DAT 操作	0	1	0	X	在 MST 或寻址的 SLV 模式下，仅内部硬件受影响。在所有情况下，总线被释放、I²C 块切换到非寻址的 SLV 模式。STO 复位。

42.10.6 某些特殊情况

I²C 硬件可以处理串行传输过程中出现的以下几种特殊情况：

- 来自两个主机的同时 “重复起动” 条件
- 仲裁丢失后进行数据传输
- 强制访问 I²C 总线
- SCL 或 SDA 低电平妨碍 I²C 总线的操作
- 总线错误

42.10.6.1 来自两个主机的同时 “重复起动” 条件

在主机发送器模式或主机接收器模式下可以产生 “重复起动” 条件。如果此时另一个主机同时产生重复起始条件，就出现特殊情况 （参见图 168）。因为两台主机发送相同的数据，所以任一台主机都不会丢失仲裁，直到出现了这种情况。

如果 I²C 硬件在产生重复起始条件之前在 I²C 总线上检测到重复起始条件，则它将释放总线，并且不产生中断请求。如果另一个主机通过产生停止条件来释放总线，则 I²C 块将发送一个正常的起始条件 （状态 0x08），并开始重新进行完整的串行数据传输。

图 168. 来自两个主机的同时 “重复起动” 条件

42.10.6.2 仲裁丢失后进行数据传输

在主发送模式和主接收模式中仲裁可能会丢失（见图 162）。仲裁丢失由 STAT 中的如下状态指示：0x38、 0x68、 0x78 和 0xB0 （参见图 164 和图 165）。

如果 CON 中的 STA 标志由服务于这些状态的例程设置，则当总线再次空闲时，会在不受 CPU 干扰下发送 “起动” 条件 （状态 0x08），并可开始重试整个串行传输。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

42.10.6.3 强制访问 I²C 总线

在一些应用中，未控制源可能会导致总线挂起。在这种情况下，干扰、总线临时中断或 SDA 和 SCL 之间的临时短路都可能导致总线挂起。

如果不可控制源产生了一个多余的起始条件或掩码了一个停止条件，则 I²C 总线一直保持忙碌状态。如果 STA 标志置位且在相应的时间内未访问总线，那么 I² 总线有可能会被强制访问。这通过在 STA 标志仍被设置的情况下设置 STO 标志来实现。不发送 “停止” 条件。I²C 硬件的操作就好像是接收到停止条件一样，可以发送起始条件。 STO 标志由硬件清除（参见图 169）。

图 169. 强制访问忙 I²C 总线

42.10.6.4 SCL 或 SDA 低电平妨碍 I²C 总线的操作

如果 SDA 或 SCL 线被总线上任何一个器件拉低， I²C 总线可能挂起。如果 SCL 线被总线上的器件妨碍 （拉低），不能继续串行传输，这时必须通拉低 SCL 总线的器件来处理。

通常， SDA 线可能被总线上的另一与当前总线主机不同步的设备（由于丢失时钟或将噪声脉冲感测为时钟）阻塞。在这种情况下，可通过在 SCL 线上发送其它时钟脉冲解决问题（参见图 170）。虽然 I²C 接口没有专门用来检测总线挂起的定时器，但可以用系统的其它定时器来完成。检测时，软件可在 SCL 上强制时钟 （可要求多达 9 个），直到 SDA 被不良设备释放。此时，从机可能还是不同步，所以还要发送一个起始条件以确保所有 I²C 外设同步。

(1) 尝试发送一个起始条件不成功。
(2) SDA 线被释放。
(3) 尝试发送一个起始条件成功。进入状态 08H。

图 170. 从因 SDA 低电平而导致总线受阻的状态中恢复

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

42.10.6.5 总线错误

当格式帧的非法位置上检测到 “启动” 或 “停止” 条件时会出现总线错误。非法位置的示例有在地址字节、数据位或确认位的串行传输期间。

仅当 I²CC 硬件作为主机或被寻址的从机进行串行传输时，它才对总线错误有反应。检测到总线错误时，I²C 块会立即切换成非寻址的从机模式，并释放 SDA 和 SCL 线，设置中断标志，并将 0x00 装入状态寄存器。该状态代码可用作状态服务程序的向量，尝试再次终止串行传输或从错误状态中恢复，如表 975 所示。

42.10.7 I²C 状态服务程序

本节将介绍不同 I²C 状态服务程序都必须执行的操作。其中包括：

- 复位后 I²C 块的初始化。
- I²C 中断服务
- 支持 4 种 I²C 操作模式的 26 种状态服务程序。

42.10.8 初始化

在初始化示例中，I²C 块可在主机模式和从机模式中使能。每种模式都使用缓冲区进行发送和接收。初始化例程执行以下功能：

- 将器件自身的从属地址和通用调用位 (GC) 载入 I2ADR
- 置位 I²C 中断使能位和中断优先级位
- 通过同时设置 CON 中的 I2EN 和 AA 位使能从机模式，通过加载 SCLH 和 SCLL 寄存器定义串行时钟频率 （对于主机模式）。主机例程必须在主程序中启动。

这时，I²C 硬件开始在 I²C 总线上检查自身的从属地址和通用调用。如果检测到通用调用或自身的从属地址，则请求中断并将相应的状态信息载入 STAT。

42.10.9 I²C 中断服务

当进入 I²C 中断时，STAT 含有一个状态代码，可识别要执行的 26 个状态服务中的其中一个。

42.10.10 状态服务程序

每个状态程序都是 I²C 中断程序的组成部分，分别用来处理 26 种状态。

42.10.11 配合实际应用的状态服务

状态服务示例演示了响应 26 个 I²C 状态代码必须执行的典型操作。如果 4 种 I²C 操作模式中有一种或几种没被用到，则模式的相关的状态服务可被忽略，只要小心处理，那些状态就不会出现。

在应用中，可能需要在 I²C 操作过程中执行一些超时处理，来限制无效总线或丢失服务程序。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

42.11 软件示例

42.11.1 初始化程序

将 I²C 接口初始化用作从机和 / 或主机的例子。

1. 将自身的从属地址载入 ADR，如果需要，使能通用调用识别。
2. 使能 I²C 中断。
3. 将 0x44 写入 CONSET 以设置 I2EN 和 AA 位，使能从机功能。对于仅主机功能，将 0x40 写入 CONSET。

42.11.2 启动主机发送功能

通过建立缓冲区、指针和数据计数开始主机发送操作，然后启动 “起动”。

1. 初始化主机数据计数器。
2. 建立数据将发送到的从属地址，并添加写位。
3. 将 0x20 写入 CONSET 以设置 STA 位。
4. 建立要在主机发送缓冲区中发送的数据。
5. 初始化主机数据计数器以匹配正在发送的消息长度。
6. 退出。

42.11.3 启动主机接收功能

通过建立缓冲区、指针和数据计数开始主机接收操作，然后启动 “起动”。

1. 初始化主机数据计数器。
2. 建立数据将发送到的从属地址，并添加读位。
3. 将 0x20 写入 CONSET 以设置 STA 位。
4. 建立主机接收缓冲区。
5. 初始化主机数据计数器以匹配要接收的消息长度。
6. 退出。

42.11.4 I²C 中断程序

确定 I²C 的状态和处理该状态的状态程序。

1. 从 STA 中读出 I²C 的状态。
2. 使用状态值分支到 26 个可能状态例程之一。

42.11.5 无指定模式的状态

42.11.5.1 状态：0x00

总线错误。进入未寻址的从机模式并释放总线。

1. 将 0x14 写入 CONSET 以设置 STO 和 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 退出。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

42.11.5.2 主机状态

状态 08 和状态 10 用于主机发送模式和主机接收模式。R/W 位决定下一状态是在主机发送模式中还是主机接收模式中。

42.11.5.3 状态：0x08

“起动”条件已发送。将发送从属地址 + R/W 位和接收 ACK 位。

1. 将从属地址和 R/W 位写入 DAT。
2. 将 0x04 写入 CONSET 以设置 AA 位。
3. 将 0x08 写入 CONCLR 以清除 SI 标志。
4. 建立主机发送模式数据缓冲区。
5. 建立主机接收模式数据缓冲区。
6. 初始化主机数据计数器。
7. 退出。

42.11.5.4 状态：0x10

“重复起动”条件已发送。将发送从属地址 + R/W 位和接收 ACK 位。

1. 将从属地址和 R/W 位写入 DAT。
2. 将 0x04 写入 CONSET 以设置 AA 位。
3. 将 0x08 写入 CONCLR 以清除 SI 标志。
4. 建立主机发送模式数据缓冲区。
5. 建立主机接收模式数据缓冲区。
6. 初始化主机数据计数器。
7. 退出。

42.11.6 主发送状态

42.11.6.1 状态：0x18

之前的状态为状态 8 或状态 10，从属地址 + 写位已发送，并接收了 ACK。将发送第一个数据字节并接收 ACK 位。

1. 将来自主机发送缓冲区的第一个数据字节载入 DAT。
2. 将 0x04 写入 CONSET 以设置 AA 位。
3. 将 0x08 写入 CONCLR 以清除 SI 标志。
4. 主机发送缓冲区指针加递增。
5. 退出。

42.11.6.2 状态：0x20

已发送从属地址 + 写位，已接收 NOT ACK。将发送 “停止”条件。

1. 将 0x14 写入 CONSET 以设置 STO 和 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 退出。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

42.11.6.3 状态：0x28

已发送数据，已接收 ACK。如果发送的数据是最后一个数据字节则发送一个“停止”条件，否则发送下一个数据字节。

1. 主机数据计数器递减，如果发送的不是最后一个数据字节则跳至第 5 步。
2. 将 0x14 写入 CONSET 以设置 STO 和 AA 位。
3. 将 0x08 写入 CONCLR 以清除 SI 标志。
4. 退出。
5. 将来自主机发送缓冲区的下一个数据字节载入 DAT。
6. 将 0x04 写入 CONSET 以设置 AA 位。
7. 将 0x08 写入 CONCLR 以清除 SI 标志。
8. 主机发送缓冲区指针加递增。
9. 退出。

42.11.6.4 状态：0x30

已发送数据，已接收 NOT ACK。将发送 “停止”条件。

1. 将 0x14 写入 CONSET 以设置 STO 和 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 退出。

42.11.6.5 状态：0x38

在从属地址 + 写位或数据期间仲裁已丢失。总线已释放且进入未寻址的从机模式。当总线再次空闲时将发送一个新的 “起动”条件。

1. 将 0x24 写入 CONSET 以设置 STA 和 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 退出。

42.11.7 主接收状态

42.11.7.1 状态：0x40

之前的状态为状态 08 或状态 10。从属地址 + 读位已发送，并接收了 ACK。将接收数据并返回 ACK。

1. 将 0x04 写入 CONSET 以设置 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 退出。

42.11.7.2 状态：0x48

已发送从属地址 + 读位，已接收 NOT ACK。将发送 “停止”条件。

1. 将 0x14 写入 CONSET 以设置 STO 和 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 退出。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

42.11.7.3 状态：0x50

数据已接收，ACK 已返回。将从 DAT 读取数据。将接收其它数据。如果这是最后一个数据字节，则返回 NOT ACK，否则返回 ACK。

1. 从 DAT 读取数据字节到主机接收缓冲区中。
2. 主机数据计数器递减，如果发送的不是最后一个数据字节则跳至第 5 步。
3. 将 0x0C 写入 CONCLR 以清除 SI 标志和 AA 位。
4. 退出。
5. 将 0x04 写入 CONSET 以设置 AA 位。
6. 将 0x08 写入 CONCLR 以清除 SI 标志。
7. 主机发送缓冲区指针加递增。
8. 退出。

42.11.7.4 状态：0x58

数据已接收， NOT ACK 已返回。将从 DAT 读取数据。将发送 “停止”条件。

1. 从 DAT 读取数据字节到主机接收缓冲区中。
2. 将 0x14 写入 CONSET 以设置 STO 和 AA 位。
3. 将 0x08 写入 CONCLR 以清除 SI 标志。
4. 退出。

42.11.8 从接收状态

42.11.8.1 状态：0x60

自身的从属地址 + 写位已接收， ACK 已返回。将接收数据并返回 ACK。

1. 将 0x04 写入 CONSET 以设置 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 建立从机接收模式数据缓冲区。
4. 初始化从机数据计数器。
5. 退出。

42.11.8.2 状态：0x68

作为总线主机时，在从属地址和 R/W 位中仲裁已丢失。自身的从属地址 + 写位已接收，ACK 已返回。将接收数据并返回 ACK。在总线再次空闲后设置 STA 以重启主机模式。

1. 将 0x24 写入 CONSET 以设置 STA 和 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 建立从机接收模式数据缓冲区。
4. 初始化从机数据计数器。
5. 退出。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

42.11.8.3 状态：0x70

通用调用已接收， ACK 已返回。将接收数据并返回 ACK。

1. 将 0x04 写入 CONSET 以设置 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 建立从机接收模式数据缓冲区。
4. 初始化从机数据计数器。
5. 退出。

42.11.8.4 状态：0x78

作为总线主机时，在从属地址 +R/W 位中仲裁已丢失。广播呼叫已接收且 ACK 已返回。将接收数据并返回 ACK。在总线再次空闲后设置 STA 以重启主机模式。

1. 将 0x24 写入 CONSET 以设置 STA 和 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 建立从机接收模式数据缓冲区。
4. 初始化从机数据计数器。
5. 退出。

42.11.8.5 状态：0x80

先前已使用自身的从属地址进行寻址。数据已接收且 ACK 已返回。将读取其它数据。

1. 从 DAT 读取数据字节到从机接收缓冲区中。
2. 从机数据计数器递减，如果发送的不是最后一个数据字节则跳至第 5 步。
3. 将 0x0C 写入 CONCLR 以清除 SI 标志和 AA 位。
4. 退出。
5. 将 0x04 写入 CONSET 以设置 AA 位。
6. 将 0x08 写入 CONCLR 以清除 SI 标志。
7. 从机发送缓冲区指针加递增。
8. 退出。

42.11.8.6 状态：0x88

先前已使用自身的从属地址进行寻址。数据已接收且 NOT ACK 已返回。将不保存接收的数据。进入未寻址的从机模式。

1. 将 0x04 写入 CONSET 以设置 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 退出。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

42.11.8.7 状态：0x90

先前已使用通用调用进行寻址。数据已接收， ACK 已返回。将保存接收的数据。将只接收第一个数据字节和 ACK。将接收其它数据和 NOT ACK。

1. 从 DAT 读取数据字节到从机接收缓冲区中。
2. 将 0x0C 写入 CONCLR 以清除 SI 标志和 AA 位。
3. 退出。

42.11.8.8 状态：0x98

先前已使用通用调用进行寻址。数据已接收，NOT ACK 已返回。将不保存接收的数据。进入未寻址的从机模式。

1. 将 0x04 写入 CONSET 以设置 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 退出。

42.11.8.9 状态：0xA0

已接收 “停止”条件或 “重复启动”条件，但仍作为从机寻址。将不保存数据。进入未寻址的从机模式。

1. 将 0x04 写入 CONSET 以设置 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 退出。

42.11.9 从发送状态

42.11.9.1 状态：0xA8

自身的从属地址 + 读位已接收， ACK 已返回。将发送数据，将接收 ACK 位。

1. 将来自从机发送缓冲区的第一个数据字节载入 DAT。
2. 将 0x04 写入 CONSET 以设置 AA 位。
3. 将 0x08 写入 CONCLR 以清除 SI 标志。
4. 建立从机发送模式数据缓冲区。
5. 从机发送缓冲区指针加递增。
6. 退出。

42.11.9.2 状态：0xB0

作为总线主机时，在从属地址和 R/W 位中仲裁丢失。自身的从属地址 + 读位已接收， ACK 已返回。将发送数据，将接收 ACK 位。在总线再次空闲后设置 STA 以重启主机模式。

1. 将来自从机发送缓冲区的第一个数据字节载入 DAT。
2. 将 0x24 写入 CONSET 以设置 STA 和 AA 位。
3. 将 0x08 写入 CONCLR 以清除 SI 标志。
4. 建立从机发送模式数据缓冲区。
5. 从机发送缓冲区指针加递增。
6. 退出。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

42.11.9.3 状态：0xB8

已发送数据，已接收 ACK。将发送数据，将接收 ACK 位。

1. 将来自从机发送缓冲区的数据字节载入 DAT。
2. 将 0x04 写入 CONSET 以设置 AA 位。
3. 将 0x08 写入 CONCLR 以清除 SI 标志。
4. 从机发送缓冲区指针加递增。
5. 退出。

42.11.9.4 状态：0xC0

已发送数据，已接收 NOT ACK。进入未寻址的从机模式。

1. 将 0x04 写入 CONSET 以设置 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 退出。

42.11.9.5 状态：0xC8

最后一个数据字节已发送， ACK 已接收。进入未寻址的从机模式。

1. 将 0x04 写入 CONSET 以设置 AA 位。
2. 将 0x08 写入 CONCLR 以清除 SI 标志。
3. 退出。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

43.1 本章导读

所有 LPC43xx 器件上均可使用 ADC0 和 ADC1。 ADC 通道数在小封装上有限制。

表 976. 不同封装的 ADC 通道

模拟功能	引脚	LBGA256	TFBGA180	TFBGA100	LQFP208	LQFP144	LQFP100
ADC0_0	P4_3	是	是	-	是	是	-
ADC0_1	P4_1	是	是	-	是	是	-
ADC0_2	PF_8	是	-	-	-	-	-
ADC0_3	P7_5	是	是	-	是	是	-
ADC0_4	P7_4	是	是	-	是	是	-
ADC0_5	PF_10	是	-	-	是	-	是
ADC0_6	PB_6	是	是	-	-	-	-
ADC1_0	PC_3	是	-	-	是	-	-
ADC1_1	PC_0	是	是	-	是	-	-
ADC1_2	PF_9	是	-	-	是	-	-
ADC1_3	PF_6	是	-	-	是	-	-
ADC1_4	PF_5	是	-	-	是	-	-
ADC1_5	PF_11	是	-	-	是	-	是
ADC1_6	P7_7	是	是	-	是	是	-
ADC1_7	PF_7	是	-	-	是	-	-
DAC	P4_4	是	是	-	是	是	-
ADC 输入通道 0。在 10 位 ADC0/1 和 DAC 之间共享。	ADC0_0/ ADC1_0/DAC	是	是	是	是	是	是
ADC 输入通道 1。在 10 位 ADC0/1 之间共享。	ADC0_1/ ADC1_1	是	是	是	是	是	是
ADC 输入通道 2。在 10 位 ADC0/1 之间共享。	ADC0_2/ ADC1_2	是	是	是	是	是	是
ADC 输入通道 3。在 10 位 ADC0/1 之间共享。	ADC0_3/ ADC1_3	是	是	是	是	是	是
ADC 输入通道 4。在 10 位 ADC0/1 之间共享。	ADC0_4/ ADC1_4	是	是	-	是	是	-
ADC 输入通道 5。在 10 位 ADC0/1 之间共享。	ADC0_5/ ADC1_5	是	是	-	是	是	-
ADC 输入通道 6。在 10 位 ADC0/1 之间共享。	ADC0_6/ ADC1_6	是	是	-	是	是	-
ADC 输入通道 7。在 10 位 ADC0/1 之间共享。	ADC0_7/ ADC1_7	是	是	-	是	是	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

43.2 基本配置

ADC0 和 ADC1 配置如下：

- 参见表 977 以了解时钟与电源控制。
- ADC0 通过 ADC0_RST 复位 （复位 # 40）。
- ADC1 通过 ADC1_RST 复位 （复位 # 41）。
- ADC0 中断连接到 NVIC 中的中断插槽 #17 上。
- ADC1 中断连接到 NVIC 中的中断插槽 #21 上。
- 连接到 GPDMA 的方法是使用 CREG 模块中的 DMAMUX 寄存器（表 41），并使能 DMA 通道配置寄存器中的 GPDMA 通道第 19.6.20 节。
- 可选择外部引脚(ADCTRIG0/1)、MOTOCON PWM MCOA2输出作为ADC0/1的转换启动 （参见表 981）。
- ADC 转换启动通过 GIMA （参见表 138）连接至定时器或 SCT 输出。
- ADC0和ADC1输入与数字功能多路复用，需要使用ENAIO0/1寄存器对其引脚进行配置（参见第 15.4.6 节和第 15.4.8 节）。
- ADC1 通道 7 连接到带隙基准 （参见第 15.4.8.1 节）。

表 977. ADC0/1 时钟和电源控制

	基本时钟	分支时钟	工作频率	备注
ADC0 时钟	BASE_APB3_CLK	CLK_APB3_ADC0	最大 204 MHz	用于寄存器接口和ADC0转换率。
ADC1 时钟	BASE_APB3_CLK	CLK_APB3_ADC1	最大 204 MHz	用于寄存器接口和ADC1转换率。

43.3 功能要点

- 10 位逐次逼近型模数转换器。
- 输入在 8 个引脚中多路复用。
- 掉电模式。
- 测量范围：0 至 3.3 V。
- 10 位转换时间 = 2.45 μs。
- 用于单个或多个输入的连发转换模式。
- 输入引脚或定时器匹配信号跳变的选择性转换。
- 每个 A/D 通道的独立结果寄存器可减少中断开销。
- 连接至带隙基准 （参见第 15.4.8.1 节）。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

43.4 简介

A/D 转换器的基础时钟由 APB 时钟提供 (CLK_APB4_ADC0/1)。每个转换器包括一个可编程分频器，将此时钟调整到逐次逼近所需的 4.5 MHz （最大）时钟。一个完全准确的转换要求有 11 个这样的时钟。

43.5 引脚说明

表 978 简要总结了每个 ADC 相关引脚。

表 978. ADC 引脚说明

引脚功能	类型	描述
ADC0_[7:0]/ ADC1_[7:0]	输入	共用的模拟输入。A/D 转换单元可以测量任何这种输入信号的电压。输入在纯模拟引脚上的 ADC0 和 ADC1 之间共用。 **注：**ADC0 引脚与 DAC0 引脚共用。
ADC0_[6:0]	输入	模拟输入。从多路复用模拟 / 数字引脚至 ADC0 的输入。A/D 转换单元可以测量任何这种输入信号的电压。这些引脚不与 ADC1 共用。
ADC1_[7:0]	输入	模拟输入。从多路复用模拟 / 数字引脚至 ADC1 的输入。A/D 转换单元可以测量任何这种输入信号的电压。这些引脚不与 ADC0 共用。
ADCTRIG0	输入	ADC0/1 的启动输入。
ADCTRIG1	输入	ADC0/1 的启动输入。
VDDA	电源	模拟电源。也是两个 ADC 的参考电压 VREF。
VSSA	接地	模拟接地。

43.6 寄存器描述

ADC0 的寄存器地址如表 979 所示。

表 979. 寄存器简介：ADC0 （基址 0x400E 3000）

名称	访问类型	地址偏移	描述	复位值 [1]	参考
CR	R/W	0x000	A/D 控制寄存器。必须对 AD0CR 寄存器进行写操作，使其在 A/D 转换发生前选择操作模式。	0x0000 0000	表 981
GDR	R0	0x004	A/D 全局数据寄存器。包含最新的 A/D 转换的结果。	-	表 982
-	-	0x008	保留。	-	
INTEN	R/W	0x00C	A/D 中断使能寄存器。此寄存器包含使能位，该使能位使得每个 A/D 通道的 DONE 标志被包含或排除在生成 A/D 中断的过程中。	0x0000 0100	表 983
DR0	RO	0x010	A/D 通道 0 数据寄存器。此寄存器包含通道 0 中最新完成的转换的结果。	-	表 984
DR1	RO	0x014	A/D 通道 1 数据寄存器。该寄存器包含通道 1 中最新完成的转换的结果。	-	表 984
DR2	RO	0x018	A/D 通道 2 数据寄存器。该寄存器包含通道 2 中最新完成的转换的结果。	-	表 984
DR3	RO	0x01C	A/D 通道 3 数据寄存器。该寄存器包含通道 3 中最新完成的转换的结果。	-	表 984

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 979. 寄存器简介：ADC0 （基址 0x400E 3000） *（续）*

名称	访问类型	地址偏移	描述	复位值 [1]	参考
DR4	RO	0x020	A/D 通道 4 数据寄存器。该寄存器包含通道 4 中最新完成的转换的结果。	-	表 984
DR5	RO	0x024	A/D 通道 5 数据寄存器。该寄存器包含通道 5 中最新完成的转换的结果。	-	表 984
DR6	RO	0x028	A/D 通道 6 数据寄存器。该寄存器包含通道 6 中最新完成的转换的结果。	-	表 984
DR7	RO	0x02C	A/D 通道 7 数据寄存器。该寄存器包含通道 7 中最新完成的转换的结果。	-	表 984
STAT	RO	0x030	A/D 状态寄存器。此寄存器包含全部 A/D 通道的 DONE 和 OVERRUN 标志，以及 A/D 中断标志。	0	表 985

[1] 复位值仅反映存储在所用位中的数据，不包括保留位的内容。

表 980. 寄存器简介：ADC1 （基址 0x400E 4000）

名称	访问类型	地址偏移	描述	复位值 [1]	参考
CR	R/W	0x000	A/D 控制寄存器。必须对 AD1CR 寄存器进行写操作，使其在 A/D 转换发生前选择操作模式。	0x0000 0000	表 981
GDR	R0	0x004	A/D 全局数据寄存器。包含最新的 A/D 转换的结果。	-	表 982
-	-	0x008	保留。	-	
INTEN	R/W	0x00C	A/D 中断使能寄存器。此寄存器包含使能位，该使能位使得每个 A/D 通道的 DONE 标志被包含或排除在生成 A/D 中断的过程中。	0x0000 0100	表 983
DR0	RO	0x010	A/D 通道 0 数据寄存器。此寄存器包含通道 0 中最新完成的转换的结果。	-	表 984
DR1	RO	0x014	A/D 通道 1 数据寄存器。该寄存器包含通道 1 中最新完成的转换的结果。	-	表 984
DR2	RO	0x018	A/D 通道 2 数据寄存器。该寄存器包含通道 2 中最新完成的转换的结果。	-	表 984
DR3	RO	0x01C	A/D 通道 3 数据寄存器。该寄存器包含通道 3 中最新完成的转换的结果。	-	表 984
DR4	RO	0x020	A/D 通道 4 数据寄存器。该寄存器包含通道 4 中最新完成的转换的结果。	-	表 984
DR5	RO	0x024	A/D 通道 5 数据寄存器。该寄存器包含通道 5 中最新完成的转换的结果。	-	表 984
DR6	RO	0x028	A/D 通道 6 数据寄存器。该寄存器包含通道 6 中最新完成的转换的结果。	-	表 984
DR7	RO	0x02C	A/D 通道 7 数据寄存器。该寄存器包含通道 7 中最新完成的转换的结果。	-	表 984
STAT	RO	0x030	A/D 状态寄存器。此寄存器包含全部 A/D 通道的 DONE 和 OVERRUN 标志，以及 A/D 中断标志。	0	表 985

[1] 复位值仅反映存储在所用位中的数据，不包括保留位的内容。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

43.6.1 A/D 控制寄存器

A/D 控制寄存器提供的位可用于选择要转换的 A/D 通道、 A/D 计时、 A/D 模式和 A/D 起动启动。

表 981. A/D 控制寄存器 （CR - 地址 0x400E 3000 (ADC0) 和 0x400E 4000 (ADC1)）位描述

位	符号	值	描述	复位值
7:0	SEL		选择要采样和转换的ADC[7:0]引脚。位0选择引脚ADC0，位1选择引脚ADC1，...，位 7 选择引脚 ADC7。在软件控制模式中，这些位中只应有一位为 1。在硬件扫描模式中， 1-8 位的值都为 1。所有的零都相当于 0x01。	0
15:8	CLKDIV		ADC 时钟除以 CLKDIV 值加 1 来产生用于 A/D 转换器的时钟，该时钟的频率应小于等于 4.5 MHz。通常软件应编程该域中的最小值来得到 4.5 MHz 或稍低于 9 MHz 的时钟，但某些情况下 （例如高阻抗模拟电源）可能需要更低的时钟。	0
16	BURST		控制连发模式	0
		0	转换由软件控制并要求有 11 个时钟。	
		1	AD 转换器以 CLKS 字段选择的速率来重复转换，1s 内扫描（如果有必要）SEL 字段中被选的引脚。起动后进行的第一次转换对应于 SEL 字段中的最低有效位 1，然后是 （适用情况下）更高编号的 1 位 （引脚） 。清除此位可终止重复转换，但是清除此位时正在执行的转换仍将完成。 **注意事项：**当 BURST = 1 时 START 位必须为 000，否则转换无法启动。	
19:17	CLKS		此字段选择连发模式下每个转换使用的时钟数量，以及 ADDR 的 LS 位中的决定结果准确性的位的数量，其数量介于11个时钟（10 位）和4个时钟（3 位）之间。	000
		0x0	11 个时钟 /10 位	
		0x1	10 个时钟 /9 位	
		0x2	9 个时钟 /8 位	
		0x3	8 个时钟 /7 位	
		0x4	7 个时钟 /6 位	
		0x5	6 个时钟 /5 位	
		0x6	5 个时钟 /4 位	
		0x7	4 个时钟 /3 位	
20	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
21	PDN		电源模式	0
		0	A/D 转换器处于掉电模式。	
		1	A/D 转换器可以运行。	
23:22	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
26:24	START		BURST 位为 0 时，控制 A/D 转换的启动。	0
		0x0	未开始 （将 PDN 清除为 0 时应使用该值）。	
		0x1	现在启动。	
		0x2	当位 27 选择的边沿出现在 CTOUT_15 时开始转换（结合定时器输出 15、ADC start0）。	
		0x3	当位 27 选择的边沿出现在 CTOUT_8 时开始转换 （结合定时器输出 8、 ADC start1）。	
		0x4	当位 27 选择的边沿出现在 ADCTRIG0 输入时开始转换 (ADC start3)。	
		0x5	当位 27 选择的边沿出现在 ADCTRIG1 输入时开始转换 (ADC start4)。	
		0x6	当位27选择的边沿出现在Motocon PWM输出MCOA2时开始转换(ADC start5)。	
		0x7	保留。	

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 981. A/D 控制寄存器 （CR - 地址 0x400E 3000 (ADC0) 和 0x400E 4000 (ADC1)）位描述

位	符号	值	描述	复位值
27	EDGE		控制用于启动转换的已选信号的上升沿或下降沿。该位仅在 START 字段包含 0x2-0x6 时有效。	0
		0	上升沿。	
		1	下降沿。	
31:28	-	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

43.6.2 A/D 全局数据寄存器。

A/D 全局寄存器包含最新的 A/D 转换的结果。这包括数据、 DONE 和溢出标志以及与数据相关的 A/D 通道的数量。

表 982. A/D 全局数据寄存器 （CR - 地址 0x400E 3004 (ADC0) 和 0x400E 4004 (ADC1)）位描述

位	符号	描述	复位值
5:0	-	保留。这些位始终读为零。	0
15:6	V_VREF	当 DONE 为 1 时，此字段包含一个二进制分式，该分式代表 SEL 字段选择的 ADCn 引脚的电压，除以 VDDA 引脚的参考电压。字段中的零表示 ADCn 输入引脚电压比 VSSA 上的电压小、相等或接近，而 0x3FF 表明 ADCn 输入引脚电压与 VDDA 上的电压接近、相等或大于它。	-
23:16	-	保留。这些位始终读为零。	0
26:24	CHN	这些位包含 LS 位转换的源通道。	-
29:27	-	保留。这些位始终读为零。	0
30	OVERRUN	在连发模式中，如果在产生 V_VREF 位结果的转换之前，一个或多个转换的结果丢失或被覆写，则此位为 1。	0
31	DONE	当模数转换完成后，此位被设置为1。当读取此寄存器且对AD0/1CR 寄存器进行写操作时，该位被清除。如果在对 AD0/1CR 进行写操作时，某一转换仍在执行，则此位将被设置并开始新的转换。	0

43.6.3 A/D 中断使能寄存器

使用此寄存器可控制在转换完成时哪些 A/D 通道产生中断。例如，可能需要使用一些 A/D 通道不断地对传感器执行转换来监视它们。无论何时需要，都可通过应用程序来读取最新的结果。在这种情况下，不希望在某些 A/D 通道的每次转换结束时都有中断。

表 983. A/D中断使能寄存器（INTEN - 地址0x400E 300C (ADC0)和0x400E 400C (ADC1)）位描述

位	符号	描述	复位值
7:0	ADINTEN	使用这些位可控制在转换完成时哪些 A/D 通道产生中断。当位 0 为 1 是， A/D 通道 0 中的转换完成将产生中断，当位 1 为 1， A/D 通道 1 中的转换完成将产生中断，等等。	0x00
8	ADGINTEN	当该位为 1 时，使能 ADDR 中的全局 DONE 标志来产生中断。当该位为 0 时，只有 ADINTEN 7:0 使能的单个 A/D 通道将产生中断。	1
31:9	-	保留。始终为 0。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

43.6.4 A/D 数据寄存器

A/D 数据寄存器保存 A/D 转换完成时的结果，也包括表明转换何时完成及转换超限何时发生的标志。

表 984. A/D 数据寄存器 （DR – 地址 0x400E 3010 (DR0) 到 0x400E 302C (DR7) (ADC0) ；0x400E 4010 (DR0) 到 0x400E 402C (DR7) (ADC1)）位描述

位	符号	描述	复位值
5:0	-	保留。始终为 0。	0
15:6	V_VREF	当 DONE 为 1 时，此字段包含一个二进制分式，该分式代表表 981 所选的 ADCn 输入引脚的电压，除以 VDDA 引脚的电压。字段中的零表示 ADCn 输入引脚电压比 VDDA 上的电压小、相等或接近，而 0x3FF 表明 ADCn 输入引脚电压与 VDDA 上的电压接近、相等或大于它。	-
29:16	-	保留。始终为 0。	0
30	OVERRUN	在连发模式中，如果在产生此寄存器中V_VREF位的结果的转换之前，一个或多个转换的结果丢失或被覆写，则此位为 1。读取该寄存器将清除此位。	0
31	DONE	当 A/D 转换完成后，此位被设置为 1。读取寄存器时将清除此位。	0

43.6.5 A/D 状态寄存器

利用 A/D 状态寄存器可以同时检查所有 A/D 通道的状态。 AD0/1DRn 寄存器中出现的每个 A/D 通道 n 的 DONE 和 OVERRUN 标志都镜像在 ADSTAT 中。中断标志（所有 DONE 标志的逻辑 OR）也可在 ADSTAT 中找到。

表 985. A/D 状态寄存器 （STAT - 地址 0x400E 3030 (ADC0)， 0x400E 4030 (ADC1)）位描述

位	符号	描述	复位值
7:0	DONE	这些位镜像了出现在每个A/D通道的结果寄存器中的DONE状态标志。	0
15:8	OVERUN	这些位镜像了出现在每个 A/D 通道的结果寄存器中的 OVERRRUN 状态标志。读取 ADSTAT 允许同时检查所用 A/D 通道的状态。	0
16	ADINT	此位为 A/D 中断标志。当任何单个 A/D 通道 Done 标志被断言并使能，以通过 ADINTEN 寄存器促成 A/D 中断时，该位为 1。	0
31:17	-	保留。始终为 0。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

43.7 操作

43.7.1 硬件启动转换

如果 ADCR 中的 BURST 位为 0，且 START 字段包含介于 0x2 和 0x6 的任何值，则 A/D 转换器在所选引脚或定时器信号有跳变发生时，将开始转换。选项包括两个 ADCTRIG 外部输入引脚，一个来自 motocon PWM 的输出和两个结合的定时器输出 （参见第 43.6.1 节）。

43.7.2 中断

当 ADSTAT 寄存器中的 ADINT 位为 1 时，有中断请求送至 NVIC。当已使能中断 （通过 ADINTEN 寄存器）的 A/D 通道的任何 DONE 位为 1 时， ADINT 位为 1。软件可以使用与 ADC 对应的 NVIC 的中断使能位来控制此结果是否处于中断。要清除对应的 DONE 标志，必须读取产生中断的 A/D 通道的结果寄存器。

43.7.3 DMA 控制

DMA 传输请求从 ADC 中断请求线生成。要产生 DMA 传输，必须满足与产生中断相同的条件。在 DMA 读取请求通道的 A/D 数据寄存器 (DR[7:0]) 后，会清除待定的 DMA 请求。读取全局数据寄存器 (GDR) 不会清除任何待定的 DMA 请求。

DMA 传输只支持连发请求。连发的大小可在 DMA 通道控制寄存器中设置为预定义连发大小之一 （参见表 279）。如果 ADC 通道的数量与 DMA 支持的预定义连发大小之一都不相等 （DMA 适用大小为 1、 4、 8），则将连发大小设置为 1。

DMA 传输长度决定 DMA 中断何时产生。传输长度可以设置为正在转换的 ADC 通道的数量（参见第 19.6.19 节）。非连续的通道可使用分散 / 聚集链表由 DMA 进行传输 （参见表 19.8.5）。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

44.1 本章导读

所有 LPC43xx 器件上均可使用 DAC。

44.2 基本配置

DAC 的配置如下：

- 必须使能 DAC 转换器寄存器 （表 989）中的 DMA_ENA 位来获得有效的 DAC 输出。
- 参见表 986 以了解时钟与电源控制。
- DAC 通过 DAC_RST 复位 （复位 # 42）。
- DAC 中断连接到 NVIC 中的中断插槽 # 0 上。
- 连接到 GPDMA 的方法是使用 CREG 模块中的 DMAMUX 寄存器（表 41），并使能 DMA 通道配置寄存器中的 GPDMA 通道第 19.6.20 节。

表 986. DAC 时钟和电源控制

	基本时钟	分支时钟	工作频率	备注
DAC 寄存器接口时钟和 DMA 计数器速率时钟。	BASE_APB3_CLK	CLK_APB3_DAC	高达 204 MHz	-

44.3 功能要点

- 10 位分辨率
- 采用单调性设计 （电阻串结构）
- 转换速度可控
- 可针对速度和功耗进行优化
- 低功耗
- 最大更新率为 1 MHz
- DMA 支持

44.4 引脚说明

表 987 简要总结了每个 DAC 相关引脚。

表 987. DAC 引脚描述

引脚	类型	描述
DAC	输出	模拟输出。DACR 写入新值后，经过所选的建立时间，此引脚上的电压（相对于 V_{SSA}）为 VALUE/1024 ≤ VDDA。DAC 引脚与 ADC0 和 ADC1 的通道 0 输入引脚共享。
VDDA	电源	模拟电源和电压参考。该引脚为D/A转换器提供电压参考水平VREF。
VSSA	-	地线。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

44.5 寄存器描述

表 988. 寄存器简介：DAC （基址 0x400E 1000）

名称	访问类型	地址偏移	描述	复位值	参考
CR	R/W	0x000	DAC 寄存器。保存转换数据。	0	表 989
CTRL	R/W	0x004	DAC 控制寄存器。	0	表 990
CNTVAL	R/W	0x008	DAC 计数器值寄存器。	0	表 991

44.5.1 D/A 转换器寄存器

此读 / 写寄存器包括要转换为模拟输出值的数字值和一个权衡性能与功耗的位。

表 989. D/A 转换器寄存器 （CR- 地址 0x400E 1000）位说明

位	符号	值	描述	复位值
5:0	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-
15:6	VALUE		此字段写入新值后，经过所选的建立时间， DAC 引脚上的电压（相对于 V_{SSA}）为 VALUE/1024 ≤ VDDA。	0
16	BIAS		建立时间	0
		0	建立时间更短，功耗更高；允许最大更新率为 1 MHz。	
		1	建立时间更长，功耗更低；允许最大更新率为 400 kHz。	
31:17	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

44.5.2 D/A 转换器控制寄存器

此读 / 写寄存器使能 DMA 操作并控制 DMA 定时器。

表 990. D/A 控制寄存器 （CTRL - 地址 0x400E 1004）位描述

位	符号	值	描述	复位值
0	INT_DMA_REQ		DMA 请求	0
		0	只要对 DACR 寄存器进行写操作此位就被清除。	
		1	定时器超时时，此位由硬件进行设置。	
1	DBLBUF_ENA		DMA 双缓冲	0
		0	禁用 DACR 双缓冲。	
		1	当此位与 CNT_ENA 位都被设置时， DACR 寄存器中的双缓冲功能将使能。写入 DACR 寄存器的内容将被写入前缓冲区然后在下一次计数器超时时传输到 DACR 中。	
2	CNT_ENA		DMA 超时	0
		0	禁用超时计数器操作。	
		1	使能超时计数器操作。	
3	DMA_ENA		使能 DMA	0
		0	禁止 DMA 访问。	
		1	为 DAC 使能 DMA 连发请求输入 15 （参见表 260）。	
31:4	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

44.5.3 D/A 转换器计数器值寄存器

此读 / 写寄存器包含中断 /DMA 计数器的重载值。

表 991. D/A 转换器计数器值寄存器 （CNTVAL - 地址 0x400E 1008）位描述

位	符号	描述	复位值
15:0	VALUE	DAC 中断 /DMA 定时器的 16 位重载值。	0
31:16	-	保留。	-

44.6 功能说明

44.6.1 DMA 计数器

当 DACCTRL 中的计数器使能位 CNT_ENA 被设置后，16 位的计数器将开始以 CLK_APB3_DAC 所选的速率，从预先编程到 DACCNTVAL 寄存器中的值开始进行倒计数。每当计数器达到零时就会递减，并按照 DACCNTVAL 的值重新加载，DMA 请求位 INT_DMA_REQ 将在硬件中被设置。

注意，DACCTRL 和 DACCNTVAL 寄存器中的内容是可进行读写访问的，但是定时器本身既不可读也不可写。

如果 DACCTRL 寄存器中的 DMA_ENA 位被设置，则 DAC DMA 请求将被发送到 GPDMA。DMA_ENA 位被清除后，复位后为默认状态， DAC DMA 请求将被封锁。

44.6.2 双缓冲

只有当 DACCTRL 中的 CNT_ENA 和 DBLBUF_ENA 位都被设置时，双缓冲才使能。在这种情况下，对 DACR 寄存器的任何写操作都将只会加载前缓冲区，前缓冲区与 DACR 寄存器分享其寄存器地址。只要计数器达到零且 DMA 请求被设置，DACR 自身都将从前缓冲区加载。同时，将使用 COUNTVAL 寄存器的值重载计数器。

读取 DACR 寄存器将只会返回 DACR 寄存器自身的内容，不返回前缓冲区寄存器的内容。

如果 CNT_ENA 或 DBLBUF_ENA 位有一个为 0，对 DACR 地址的任何写入都将直接进入 DACR 寄存器。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

图 171. 使用 DMA 中断和定时器的 DAC 控制

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

45.1 本章导读

闪存编程 API 在带有片内闪存的器件上均可用。对于没有闪存的器件，支持较少的在系统编程 (ISP) 命令集 （参见表 995）。有关无闪存器件启动过程的详细信息，参见第 5 章。

45.2 简介

引导装载程序控制复位后的初始操作，并提供闪存编程工具。这可能是清空设备的初始编程，已编程设备的擦除和重编程，或者是通过运行系统中的应用程序对闪存编程。

45.3 功能要点

- 在线系统编程：在线系统编程 (ISP) 是使用引导装载程序软件和 UART0 串口，对片内闪存进行编程或重编程。当器件位于最终用户端时，可执行此操作。
- 对于没有片内闪存的器件，ISP 允许向片内 SRAM 加载数据并执行来自片内 SRAM 的代码。
- 在应用编程：在应用编程 (IAP) 的功能是按照终结用户应用程序代码对片内闪存进行擦除及写入操作。
- 闪存签名生成：嵌入式硬件可以为一系列闪存地址和整个闪存生成签名。

45.4 描述

器件每次上电或复位时，闪存引导加载程序代码即被执行一次。加载程序还可以执行 ISP 命令处理程序或用户应用程序代码。复位后引脚 P2_7 的低电平将被视为起动 ISP 命令处理程序的外部硬件请求。假设在 RESET 引脚上产生上升沿时，电源引脚处于标称的电平，那么在最多 3 ms 后，会对 P2_7 引脚信号进行采样并确定是继续用户代码还是产生 ISP 处理程序。如果 P2_7 采样为低电平且看门狗上溢标志被设置，则忽略起动 ISP 命令处理程序的外部硬件请求。如果没有执行 ISP 命令处理程序的请求（复位后 P2_7 采样为高电平），则搜索有效的用户程序。如果找到有效的用户程序，则会将执行控制转交给它。如果没找到有效的用户程序，将调用自动波特率例程。

引脚 P2_7 被用作 ISP 的硬件请求信号，因此需要特别注意它。由于复位后 P2_7 处于高阻抗，用户需要提供外部硬件 （上拉电阻器或其他器件）将引脚置于已定义状态，这非常重要。否则，可能意外进入 ISP 模式。

在加电复位后进入 ISP 模式时， IRC 和 PLL1 会用于产生 96 MHz 的 CCLK 信号。UART0 引脚被设置为 P2_0 和 P2_1。

确定主机的波特率后，测试字符串 “Synchronized”将被发送到主机。信号握手成功后，ISP 进入命令中断模式。

硬件闪存签名生成功能内置到闪存中，此功能可用于创建签名，然后可使用该签名来验证闪存内容。闪存签名生成的详细信息如第 45.11 节所述。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

45.4.1 复位后的存储器映射

复位后，当用户程序开始执行时，中断向量将被设置为指向闪存的开始（参见图 3）。

45.4.1.1 有效用户代码的判定标准

保留的 Cortex-M4 异常向量位置 7（向量表中偏移量 0x 001C）应此包含表中条目 0 到 6 的校验和的二补数。这将使表中头 8 个条目的校验和为 0。启动引导程序代码将求闪存扇区 0 的前 8 个位值的校验和。如果结果为 0，则执行控制将被转交到用户代码。

如果签名无效，则自动波特率例程通过串行端口 0 与主机进行同步。主机应当发送一个同步字符 ?(0x3F) 并等待响应。主机侧的串口设置应是 8 个数据位，1 个停止位，没有奇偶校验。自动波特率例程按其自身的频率测量接收到的同步字符的位时间，并对串口的波特率发生器进行编程。它还向主机发送一个 ASCII 字符串（"Synchronized<CR><LF>"）。作为响应，主机应当发送同一字符串（"Synchronized<CR><LF>"）。自动波特率例程检查接收到的字符以验证同步。如果通过验证，则向主机发送 “OK<CR><LF>” 字符串。主机应通过发送器件运行所使用的晶体频率（单位 kHz）进行响应。例如，如果运行在 10MHz，主机的响应当为 “10000<CR><LF>”。在接收到晶振频率后再向主机发送 “OK<CR><LF>”。如果同步未得到验证，则自动波特率例程再次等待同步字符。在用户调用 ISP 的情况下要使自动波特率正确工作， CCLK 频率应当大于或等于 10 MHz。

接收到晶体频率后，将立即初始化器件并调用 ISP 命令处理程序。因为安全方面的原因，在执行将导致闪存擦除 / 写入操作的命令和 "Go" 命令之前必需有 "Unlock" 命令。其他命令可直接执行，不需要先执行解锁命令。每个 ISP 会话都要求执行一次 Unlock 命令。解锁命令在 第 972 页上的章节 45.8 “ISP 命令” 中进行了说明。

45.4.2 通信协议

所有 ISP 命令都应作为单一 ASCII 字符串发送。这些字符串应使用回车 (CR) 和 / 或换行 (LF) 控制字符来终止，多余的 <CR> 和 <LF> 将被忽略。所有 ISP 响应都是以 <CR><LF> 终止的 ASCII 字符串形式发送。数据以 UU 编码格式发送和接收。

45.4.2.1 ISP 命令格式

"Command Parameter_0 Parameter_1 … Parameter_n<CR><LF>" "Data" （Data 只适用于写命令）。

45.4.2.2 ISP 响应格式

"Return_Code<CR><LF>Response_0<CR><LF>Response_1<CR><LF> … Response_n<CR><LF>" "Data" （Data 只适用于读命令）。

45.4.2.3 ISP 数据格式

数据流采用 UU 编码格式。 UU 编码算法将 3 字节的二进制数据转换成 4 字节的可打印的 ASCII 字符集。它比十六进制格式更有效，十六进制格式将 1 字节的二进制数据转换成 2 字节的 ASCII 十六进制。发送 20 个 UU 编码行后，发送端应发送校验和。 UU 编码行的长度不应超过 61 个字符（字节），也就是它可容纳 45 个数据字节。接收端应将其与已接收字节的校验和比较。如果校验和匹配，接收器应当响应 “OK<CR><LF>” 来继续下一次发送。如果校验和不匹配，则接收器应当响应 “RESEND<CR><LF>” 。作为响应，发送端应重新发送字节。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

45.4.2.4 ISP 流量控制

为了防止由缓冲区超限造成的数据丢失，使用软件 XON/XOFF 流控制模式。当数据迅速到达时，会发送 ASCII 控制字符 DC3 (0x13) 以阻止数据流。数据流通过发送 ASCII 控制字符 DC1(0x11) 来恢复。主机也应支持同样的流控制模式。

45.4.2.5 ISP 命令中止

可通过发送 ASCII 控制字符 "ESC" (0x1B) 来终止命令。此功能在 “ISP 命令” 部分没有作为命令被记录。接收到转义码后， ISP 命令处理程序将等待新的命令。

45.4.2.6 IAP 过程中的中断

片内闪存在擦除 / 写操作期间无法访问。用户应用程序代码开始执行时，来自用户闪存区的中断向量变为有效。用户应在使用闪存擦除 / 写入 IAP 调用前，禁用中断或是确保 RAM 中的用户中断向量有效且中断处理程序驻留在 RAM 中。 IAP 代码不使用或禁用中断。

45.4.2.7 ISP 命令处理程序使用的 RAM

ISP 命令使用片内 RAM 的 0x1000 0118 到 0x1000 01FF。用户可以使用此区，但其内容在复位时可能会丢失。Flash 编程命令使用片内 RAM 最顶端的 32 字节。协议栈位于 RAM 的前 32 字节，可使用的最大协议栈为 256 字节，向下增长。

45.4.2.8 IAP 命令处理程序使用的 RAM

Flash 编程命令使用片内 RAM 最顶端的 32 字节。在分配给用户的协议栈空间中，可使用的最大协议栈为 128 字节，向下增长。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

45.5 带闪存的 LPC43xx 器件的启动过程流程图

(1) 有关晶振频率处理的详情，请参见 第 985 页上的章节 45.9.9 “重新调用 ISP”

(2) 有关基于 CRP 设置的可用 ISP 命令的更多详细信息，参见 第 45.7 节 “代码读保护 (CRP)”

图 172. 带闪存的 LPC43xx 器件的启动过程流程图

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

45.6 扇区号

一些 IAP 和 ISP 命令在 “扇区” 操作，需指定扇区号。下表给出了 LPC43xx 设备的区段号和内存地址间的对应关系。 IAP 和 ISP 例程位于引导 ROM 中。

表 992. 闪存配置

闪存存储体	扇区号	扇区大小 [kB]	起始地址	结束地址	LPC43x2	LPC43x3	LPC43x5	LPC43x7
A	0	8	0x1A00 0000	0x1A00 1FFF	x	x	x	x
A	1	8	0x1A00 2000	0x1A00 3FFF	x	x	x	x
A	2	8	0x1A00 4000	0x1A00 5FFF	x	x	x	x
A	3	8	0x1A00 6000	0x1A00 7FFF	x	x	x	x
A	4	8	0x1A00 8000	0x1A00 9FFF	x	x	x	x
A	5	8	0x1A00 A000	0x1A00 BFFF	x	x	x	x
A	6	8	0x1A00 C000	0x1A00 DFFF	x	x	x	x
A	7	8	0x1A00 E000	0x1A00 FFFF	x	x	x	x
A	8	64	0x1A01 0000	0x1A01 FFFF	x	x	x	x
A	9	64	0x1A02 0000	0x1A02 FFFF	x	x	x	x
A	10	64	0x1A03 0000	0x1A03 FFFF	x	x	x	x
A	11	64	0x1A04 0000	0x1A04 FFFF	x		x	x
A	12	64	0x1A05 0000	0x1A05 FFFF	x		x	x
A	13	64	0x1A06 0000	0x1A06 FFFF	x			x
A	14	64	0x1A07 0000	0x1A07 FFFF	x			x
B	0	8	0x1B00 0000	0x1B00 1FFF		x	x	x
B	1	8	0x1B00 2000	0x1B00 3FFF		x	x	x
B	2	8	0x1B00 4000	0x1B00 5FFF		x	x	x
B	3	8	0x1B00 6000	0x1B00 7FFF		x	x	x
B	4	8	0x1B00 8000	0x1B00 9FFF		x	x	x
B	5	8	0x1B00 A000	0x1B00 BFFF		x	x	x
B	6	8	0x1B00 C000	0x1B00 DFFF		x	x	x
B	7	8	0x1B00 E000	0x1B00 FFFF		x	x	x
B	8	64	0x1B01 0000	0x1B01 FFFF		x	x	x
B	9	64	0x1B02 0000	0x1B02 FFFF		x	x	x
B	10	64	0x1B03 0000	0x1B03 FFFF		x	x	x
B	11	64	0x1B04 0000	0x1B04 FFFF			x	x
B	12	64	0x1B05 0000	0x1B05 FFFF			x	x
B	13	64	0x1B06 0000	0x1B06 FFFF				x
B	14	64	0x1B07 0000	0x1B07 FFFF				x

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

45.7 代码读保护 (CRP)

代码读取保护是一种允许用户使能不同的系统安全等级的机制，从而可以限制片内闪存的访问和 ISP 的使用。需要时，通过在闪存 0x000002FC 位置对特定模式进行编程来调用 CRP。 IAP 命令不受代码读取保护影响。

重要事项：任何 CRP 更改都仅在设备经过一个电源周期后才变为有效。

表 993. 代码读保护选项

名称	0x000002FC 中的编程模式	描述
CRP1	0x12345678	禁止通过 JTAG 引脚访问芯片。此模式允许使用下列 ISP 命令和限制更新部分闪存： • 由于ISP代码使用SRAM，写入RAM命令不能访问SRAM中0x10000200 以下的内容，参见第 45.4.2.7 节。 • 读取内存操作：禁用。 • 复制 RAM 到闪存命令：不能对扇区 0 进行写操作。 • Go 命令：禁用。 • 擦除扇区命令：可擦除除了扇区 0 的任何单独扇区，或一次性擦除所有扇区。 • 比较命令：禁用。 此模式在要求 CRP 且需要更新闪存字段但不能擦除所有扇区时有用。禁用比较命令，因此在部分闪存更新的情况下，二级加载程序应执行校验和机制来验证闪存的完整性。
CRP2	0x87654321	除了以下项外，它与 CRP1 相似： • 写入 RAM 命令：禁用。 • 复制 RAM 到闪存：禁用。 • 擦除命令：只允许擦除所有扇区。
CRP3	0x43218765	与 CRP2 相似，但如果闪存扇区 0 中存在有效用户代码，则禁用通过拉低 P2_7 电平进入 ISP。 此模式有效禁止了使用 P2_7 引脚覆写 ISP。用户应用程序应通过使用 IAP 调用或者使用 UART0 调用 ISP 来进行闪存更新。 **注意：如果选择了 CRP3，此后该器件将无法执行厂商测试。**

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 994. 代码读保护硬件 / 软件的相互作用

CRP 选件	用户代码有效性	P2_7 引脚复位状态	是否使能JTAG	LPC43xx 是否进入 ISP 模式	ISP模式下部分闪存更新
无	无	X	有	有	有
	有	高电平	有	无	不适用
	有	低电平	有	有	有
CRP1	无	x	无	有	有
	有	高电平	无	无	不适用
	有	低电平	无	有	有
CRP2	无	x	无	有	无
	有	高电平	无	无	不适用
	有	低电平	无	有	无
CRP3	无	x	无	有	无
	有	x	无	无	不适用

如果已使能了任何 CRP 模式且通过 ISP 允许访问芯片，不受支持或受限的 ISP 命令将被终止，同时返回代码 CODE_READ_PROTECTION_ENABLED。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

45.8 ISP 命令

下列命令被 ISP 命令处理程序接受。对于每条命令都支持详细的状态码。当接收到未定义的命令时，命令处理程序发送返回码 INVALID_COMMAND。命令和返回码都采用 ASCII 格式。

只有当接收到的 ISP 命令执行完毕且可通过主机给出新的 ISP 命令时， ISP 命令处理程序才会发送 CMD_SUCCESS。"Set Baud Rate"、"Write to RAM"、"Read Memory" 和 "Go" 不遵守此规则。

表 995. ISP 命令总结

ISP 命令	用法	无闪存器件	有闪存器件	说明见：
解锁	U < 解锁代码 >	是	是	表 996
设置波特率	B < 波特率 > < 停止位 >	是	是	表 997
回标	A < 设定 >	是	是	表 999
写入 RAM	W < 起始地址 > < 字节数 >	是	是	表 1000
读取内存	W < 地址 > < 字节数 >	是	是	表 1001
准备写操作扇区	P < 起始扇区号 > < 结束扇区号 >	否	是	表 1002
复制 RAM 到闪存	C < 闪存地址 > <RAM 地址 > < 字节数 >	否	是	表 1003
运行	G < 地址 > < 模式 >	是	是	表 1004
擦除扇区	E < 起始扇区号 > < 结束扇区号 >	否	是	表 1005
空白检查扇区	I < 起始扇区号 > < 结束扇区号 >	否	是	表 1006
读取器件 ID	J	是	是	表 1007
读取引导代码版本	K	是	是	表 1009
读取序列号	N	是	是	表 1010
比较	M < 地址 1> < 地址 2> < 字节数 >	否	是	表 1011

45.8.1 解锁 < 解锁代码 >

表 996. ISP 解锁命令

命令	U
输入	解锁代码：23130_{10}
返回码	CMD_SUCCESS | INVALID_CODE | PARAM_ERROR
描述	此命令用于解锁闪存 “写入”、“擦除” 和 “运行” 命令。
示例	"U 23130<CR><LF>" 解锁闪存写入 / 擦除以及运行命令。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

45.8.2 设置波特率 < 波特率 > < 停止位 >

表 997. ISP 设置波特率命令

命令	B
输入	波特率：9600 | 19200 | 38400 | 57600 | 115200 终止位：1 | 2
返回码	CMD_SUCCESS | INVALID_BAUD_RATE | INVALID_STOP_BIT | PARAM_ERROR
描述	此命令用于更改波特率。命令处理程序发送 CMD_SUCCESS 返回码后，新的波特率生效。
示例	“B 57600 1<CR><LF>”设置串口波特率 57600bps 和 1 个停止位。

表 998. 可能的 ISP 波特率和 CCLK 频率之间的相关性 （单位 MHz）

ISP 波特率与 CCLK 频率	9600	19200	38400	57600	115200
< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >
< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >
< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >
< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >
< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >
< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >
< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >
< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >
< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >	< 待定 >

[1] 复位后 ISP 入口使用片内 IRC 和 PLL1 以 CCLK = 96 MHz 的速率来运行设备。

45.8.3 回应 < 设定 >

表 999. ISP 回应命令

命令	A
输入	设置：打开 = 1 | 关闭 = 0
返回码	CMD_SUCCESS | PARAM_ERROR
描述	回标命令的默认设置是 ON。当其为 ON 时， ISP 命令处理程序将接收到的串行数据发送回主机。
示例	“A 0<CR><LF>”将回应关闭。

45.8.4 写 RAM < 起始地址 > < 字节数 >

只有接收到 CMD_SUCCESS 返回码后，主机才应发送数据。当发送完 20 个 UU 编码行之后主机应当发送校验和。校验和是通过把原始数据 （UU 编码前）字节的值叠加起来产生的，当发送完 20 个 UU 编码行后该值会复位。任何 UU 编码行的长度不应超过 61 个字符（字节），也就是说，它可以携带 45 个数据字节。当数据少于 20 个 UU 编码行时，校验和应当按照实际发送的字节数进行计算。 ISP 命令处理程序将其与已接收字节的校验和作比

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

较。如果校验和匹配，那么 ISP 命令处理程序响应 “OK<CR><LF>” 来继续下一次发送。如果校验和不匹配，那么 ISP 命令处理程序响应 “RESEND<CR><LF>”。作为响应，主机应当重新发送字节。

表 1000. ISP 写 RAM 命令

命令	W
输入	**起始地址：**要写入数据字节的 RAM 地址。此地址应是字边界。 **字节数：**写入的字节数。此数字应是 4 的倍数。
返回码	CMD_SUCCESS | ADDR_ERROR （地址不在字边界）| ADDR_NOT_MAPPED | COUNT_ERROR （字节数不是 4 的倍数） | PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
描述	此命令用于下载数据到 RAM。数据应为 UU 编码格式。使能了代码读取保护等级 CRP2 或 CRP3 后，此命令被锁定。
示例	"W 268435968 4<CR><LF>" 向地址 0x1000 0200 写入 4 个字节数据。

45.8.5 读存储器 < 地址 > < 字节数 >

数据流后是命令成功返回代码。发送完 20 个 UU 编码行之后发送校验和。校验和是通过把原始数据 （UU 编码前）字节的值叠加起来产生的，当发送完 20 个 UU 编码行后该值会复位。任何 UU 编码行的长度不应超过 61 个字符 （字节），也就是说，它可以携带 45 个数据字节。当数据少于 20 个 UU 编码行时，校验和按照实际发送的字节数进行计算。主机应将其与已接收字节的校验和比较。如果校验和匹配，那么主机应当响应 “OK<CR><LF>” 来继续下一次发送。如果校验和不匹配，主机应当响应 “RESEND<CR><LF>” 。作为响应， ISP 命令处理程序再次发送数据。

表 1001. ISP 读存储器命令

命令	R
输入	**起始地址：**被读出数据字节的地址。此地址应是字边界。 **字节数：**读出的字节数。此数字应是 4 的倍数。
返回码	CMD_SUCCESS，后面是 < 实际数据 （UU 编码） > | ADDR_ERROR （地址不在字边界）| ADDR_NOT_MAPPED | COUNT_ERROR （字节数不是 4 的倍数）| PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
描述	此命令用于从 RAM 或闪存中读取数据。使能了任何等级的代码读取保护后，此命令将被锁定。
示例	"R 268435968 4<CR><LF>" 从地址 0x1000 0200 读取 4 个字节数据。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

45.8.6 准备写操作的扇区 < 起始扇区号 > < 结束扇区号 >

此命令使闪存写 / 擦除操作分为两步。

表 1002. ISP 准备写操作命令的扇区

命令	P
输入	**起始扇区号。** **结束扇区号：**应当大于或等于起始扇区号。
返回码	CMD_SUCCESS | BUSY | INVALID_SECTOR | PARAM_ERROR
描述	此命令必须在执行 "Copy RAM to Flash" 或 "Erase Sector(s)" 命令前执行。 "Copy RAM to Flash" or "Erase Sector(s)" 命令的成功执行会使得相关扇区被重新保护。要准备单一的扇区，使用相同的 “起始” 和 “结束” 扇区号。
示例	“P 0 0<CR><LF>” 准备 Flash 扇区 0。

45.8.7 将 RAM 内容复制到闪存 < 闪存地址 > <RAM 地址 > < 字节数 >

表 1003. ISP 复制命令

命令	C
输入	**闪存地址 (DST)：**要写入数据字节的目标 Flash 地址。目标地址应为 256 字节边界。 **RAM 地址 (SRC)：**读出数据字节的源 RAM 地址。 **字节数：**写入的字节数。应为 256 | 512 | 1024 | 4096。
返回码	CMD_SUCCESS | SRC_ADDR_ERROR （地址不在字边界） | DST_ADDR_ERROR （地址不在正确的边界） | SRC_ADDR_NOT_MAPPED | DST_ADDR_NOT_MAPPED | COUNT_ERROR （字节数不是 256 | 512 | 1024 | 4096） | SECTOR_NOT_PREPARED_FOR WRITE_OPERATION | BUSY | CMD_LOCKED | PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
描述	此命令用于闪存编程。在此命令前执行应 “准备写操作扇区” 命令。一旦复制命令成功执行，受影响的扇区将自动被重新保护。使能了代码读取保护等级 CRP2 或 CRP3 后，此命令被锁定。当使能读取保护等级 CRP1 时，除了扇区 0，其他各扇区都可进行写操作。
示例	"C 0 268468224 512<CR><LF>" 将 RAM 地址 0x1000 8000 开始的 512 字节复制到闪存地址 0。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

45.8.8 运行 < 地址 > < 模式 >

表 1004. ISP 运行命令

命令	G
输入	**地址：**代码执行起始的 Flash 或 RAM 地址。此地址应在字边界。 **模式 （保留，用于反向兼容性）：**T （在拇指模式下执行程序） |A （禁用）。
返回码	CMD_SUCCESS | ADDR_ERROR | ADDR_NOT_MAPPED | CMD_LOCKED | PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
描述	此命令用于执行驻留在 RAM 或闪存中的程序。一旦此命令执行成功，则可能不能返回 ISP 命令处理程序。使能了任何等级的代码读取保护后，此命令将被锁定。
示例	"G 0 T<CR><LF>" 跳转到地址 0x0000 0000 处。

使用 GO 命令时，代码执行从指定地址 （假设其为可执行地址）开始，设备保留为该 ISP 代码而进行的设置。这意味着与芯片复位后直接进入用户代码相比，有些特性会有所变化。最重要的是，主 PLL 将开始运行并被连接和配置，以产生频率接近 14.7456 MHz 的 CPU 时钟。

45.8.9 擦除扇区 < 起始扇区号 > < 结束扇区号 >

表 1005. ISP 擦除扇区命令

命令	E
输入	**起始扇区号。** **结束扇区号：**应当大于或等于起始扇区号。
返回码	CMD_SUCCESS | BUSY | INVALID_SECTOR | SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION | CMD_LOCKED | PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
描述	此命令用于擦除片内闪存的一个或多个扇区。使能代码读取保护等级 CRP3 后，此命令被锁定。当使能读取保护等级 CRP1 时，除了扇区 0，其他各扇区都可被擦除。启用 CRP1 和 CRP2 时，所有扇区可一次性擦除。
示例	“E 2 3<CR><LF>”擦除 Flash 扇区 2 和 3。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

45.8.10 扇区查空 < 起始扇区号 > < 结束扇区号 >

表 1006. ISP 扇区查空命令

命令	I
输入	**起始扇区号：** **结束扇区号：**应当大于或等于起始扇区号。
返回码	CMD_SUCCESS | SECTOR_NOT_BLANK （后跟 < 第一个非空字的偏移量 > < 非空字的内容 >） | INVALID_SECTOR | PARAM_ERROR |
描述	此命令用于空白检查片内闪存的一个或多个扇区。
示例	“I 2 3<CR><LF>”对 Flash 扇区 2 和 3 进行查空。

45.8.11 读器件标识号

表 1007. ISP 读器件标识命令

命令	J
输入	无。
返回码	CMD_SUCCESS 后跟 ASCII 格式的器件标识号 （见 表 1008 “LPC43xx 器件标识号”）。
描述	此命令用于读取器件识别号。器件识别号映射到设备系列的特征子集。此编号作为技术修订的结果，通常将不会改变。

表 1008. LPC43xx 器件标识号

设备	ASCII/dec 代码	Hex 代码
< 待定 >	< 待定 >	< 待定 >

45.8.12 读取启动代码版本号

表 1009. ISP 读 Boot 代码版本号命令

命令	K
输入	无
返回码	CMD_SUCCESS，后跟 2 字节 ASCII 格式的引导代码版本号。将其解释为 < 字节 1 （主） >.< 字节 0 （次） >。
描述	此命令用于读取引导代码版本号。

45.8.13 读取设备序列号

表 1010. ISP 读取设备序列号命令

命令	N
输入	无。
返回码	CMD_SUCCESS，后跟 4 个十进制 ASCII 组的设备序列号，每个都代表一个 32 位的值。
描述	此命令用于读取设备序列号。序列号可用于唯一性地确定所有 LPC43xx 设备的单一单元。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

45.8.14 比较 < 地址 1> < 地址 2> < 字节数 >

表 1011. ISP 比较命令

命令	M
输入	**地址 1(DST)**：要比较的数据字节的起始 Flash 或 RAM 地址。此地址应是字边界。 **地址 2(SRC)**：要比较的数据字节的起始 Flash 或 RAM 地址。此地址应是字边界。 **字节数**：待比较的字节数；计数值应当为 4 的倍数。
返回码	CMD_SUCCESS | （源数据和目标数据相等） COMPARE_ERROR | （后跟第一个不匹配的偏移） COUNT_ERROR （字节数不是 4 的倍数）| ADDR_ERROR | ADDR_NOT_MAPPED | PARAM_ERROR |
描述	此命令用于比较两个位置的内存内容。使能了任何等级的代码读取保护后，此命令将被锁定。
示例	"M 8192 268435968 4<CR><LF>" 将 RAM 地址 0x1000 0200 开始的 4 个字节与闪存地址 0x2000 开始的 4 个字节进行比较。

45.8.15 ISP 返回代码

表 1012. ISP 返回代码总览

返回码	助记符	描述
0	CMD_SUCCESS	命令执行成功。只有当主机给出的命令成功执行完成时，才由 ISP 处理程序发送。
1	INVALID_COMMAND	无效命令。
2	SRC_ADDR_ERROR	源地址不在字边界。
3	DST_ADDR_ERROR	目标地址不在正确的边界。
4	SRC_ADDR_NOT_MAPPED	源地址没有映射在内存映像图中。适用时应考虑计数值。
5	DST_ADDR_NOT_MAPPED	目标地址没有映射在内存映像图中。适用时应考虑计数值。
6	COUNT_ERROR	计数的字节不是 4 的倍数或允许值。
7	INVALID_SECTOR	扇区号无效或结束扇区号比起始扇区号大。
8	SECTOR_NOT_BLANK	扇区非空。
9	SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION	准备写操作扇区的命令没被执行。
10	COMPARE_ERROR	源数据和目标数据不相等。
11	BUSY	闪存编程硬件接口忙。
12	PARAM_ERROR	参数数量不足或参数无效。
13	ADDR_ERROR	地址不在字边界。
14	ADDR_NOT_MAPPED	地址没有映射在内存映像图中。适用时应考虑计数值。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 1012. ISP 返回代码总览 *（续）*

返回码	助记符	描述
15	CMD_LOCKED	命令被锁定。
16	INVALID_CODE	解锁代码无效。
17	INVALID_BAUD_RATE	无效的波特率设置。
18	INVALID_STOP_BIT	无效的终止位设置。
19	CODE_READ_PROTECTION_ENABLED	代码读取保护使能。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

45.9 IAP 命令

对于应用编程， IAP 例程应通过寄存器 r0 中的字指针来调用，此指针指向包含命令代码和参数的内存 (RAM)。 IAP 命令的结果在指向寄存器 r1 的结果表中返回。用户可以把寄存器 r0 和 r1 中的指针赋予相同的值，如此便能将命令表复用来存放结果。参数表应足够大，以便在结果数超过参数数量时可以保存所有的结果。参数传递见图 173。参数和结果的数目根据 IAP 命令而有所不同。传递给 “复制 RAM 到闪存” 命令的最大参数数量为 5。结果最大数量为 4，通过 “读取设备序列号” 命令返回。当接收到未定义的命令时，命令处理程序发送状态码 INVALID_COMMAND。 IAP 例程驻留在 0x1FFF 1FF0 位置。

IAP 函数可使用 C 语言通过下列方法调用。

定义 IAP 位置入口点。由于 Cortex-M4 只使用拇指模式，要设置 IAP 位置的位 0。

```
#define IAP_LOCATION 0x1FFF1FF1
```

定义 IAP 函数的数据架构或指针来传递 IAP 命令表和结果表：

```
unsigned long command[5];
unsigned long result[5];
```

或

```
unsigned long * command;
unsigned long * result;
command=(unsigned long *) 0x...
result= (unsigned long *) 0x...
```

定义函数类型的指针，它有两个参数并返回 void。注意，IAP 返回的结果带有驻留在 R1 中的表的基址。

```
typedef void (*IAP)(unsigned int [],unsigned int[]);
IAP iap_entry;
```

设置函数指针：

```
iap_entry=(IAP) IAP_LOCATION;
```

无论何时想调用 IAP，你都可以使用下面的语句。

```
iap_entry (command, result);
```

IAP 调用可通过使用符号定义文件功能来进一步简化，该功能由 ADS （ARM 开发工具包）中 ARM 链接器支持，也可以使用汇编代码来调用 IAP 例程。

按照 ARM 规范 （ARM 拇指程序调用标准 SWS ESPC 0002 A-05），最多 4 个参数可在寄存器 r0、 r1、 r2 和 r3 中分别传递。其它参数在协议栈上传递。最多 4 个参数可在寄存器 r0、r1、r2 和 r3 中分别返回。其它参数通过内存间接返回。有些 IAP 调用要求的参数多于 4 个。如果使用 ARM 推荐的模式进行参数传递 / 返回，则可能由于来自不同供应商的 C 编译器之间的执行差异而产生问题。推荐的参数传递模式降低了这种风险。

写操作和擦除操作期间，无法访问闪存。会导致闪存写 / 擦除操作的 IAP 命令使用片内 RAM 顶部的 32 位空间来进行执行操作。如果应用程序中允许进行 IAP 闪存编程操作，则用户程序不应使用此空间。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 1013. IAP 命令总览

IAP 命令	命令代码	说明见：
准备写操作扇区	50_{10}	表 1014
复制 RAM 到闪存	51_{10}	表 1015
擦除扇区	52_{10}	表 1016
空白检查扇区	53_{10}	表 1017
读取器件 ID	54_{10}	表 1018
读取引导代码版本	55_{10}	表 1019
读取设备序列号	58_{10}	表 1020
比较	56_{10}	表 1021
重新调用 ISP	57_{10}	表 1022

图 173. IAP 参数传递

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

45.9.1 准备写操作扇区

此命令使闪存写 / 擦除操作分为两步。

表 1014. IAP 准备写操作命令的扇区

命令	准备写操作扇区
输入	**命令代码：50 （十进制）** **参数 0：**起始扇区号 **参数 1：**结束扇区号 （应当大于或等于起始扇区号）。
返回码	CMD_SUCCESS | BUSY | INVALID_SECTOR
结果	无
描述	此命令必须在执行 "Copy RAM to Flash" 或 "Erase Sector(s)" 命令前执行。 "Copy RAM to Flash" or "Erase Sector(s)" 命令的成功执行会使得相关扇区被重新保护。要准备单一的扇区，使用相同的 “起始” 和 “结束” 扇区号。

45.9.2 复制 RAM 到闪存

表 1015. IAP 将 RAM 内容复制到闪存命令

命令	复制 RAM 到闪存
输入	**命令代码：51 （十进制）** **参数 0(DST)：**要写入数据字节的目标 Flash 地址。该地址应当为 256 字节边界。 **参数 1(SRC)：**从中读取数据字节的源 RAM 地址。此地址应是字边界。 **参数 2：**写入的字节数。应为 256 | 512 | 1024 | 4096。 **参数 3：**CPU 时钟频率 (CCLK)，单位 kHz。
返回码	CMD_SUCCESS | SRC_ADDR_ERROR （地址不以字为边界） | DST_ADDR_ERROR （地址不在正确的边界） | SRC_ADDR_NOT_MAPPED | DST_ADDR_NOT_MAPPED | COUNT_ERROR （字节数不是 256 | 512 | 1024 | 4096） | SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION | BUSY |
结果	无
描述	此命令用于闪存编程。受影响的扇区应先通过调用 “准备写操作扇区” 命令准备好。一旦复制命令成功执行，受影响的扇区将自动被重新保护。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

45.9.3 擦除扇区

表 1016. IAP 擦除扇区命令

命令	擦除扇区
输入	**命令代码：52 （十进制）** **参数 0：**起始扇区号 **参数 1：**结束扇区号 （应当大于或等于起始扇区号）。 **参数 2：**CPU 时钟频率 (CCLK)，单位 kHz。
返回码	CMD_SUCCESS | BUSY | SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION | INVALID_SECTOR
结果	无
描述	此命令用于擦除片内闪存的一个或多个扇区。要擦除单一的扇区，使用相同的“起始”和“结束”扇区号。

45.9.4 空白检查扇区

表 1017. IAP 扇区查空命令

命令	空白检查扇区
输入	**命令代码：53 （十进制）** **参数 0：**起始扇区号 **参数 1：**结束扇区号 （应当大于或等于起始扇区号）。
返回码	CMD_SUCCESS | BUSY | SECTOR_NOT_BLANK | INVALID_SECTOR
结果	**结果 0：**状态代码为 SECTOR_NOT_BLANK 时第一个非空字位置的偏移量。 **结果 1：**非空字位置的内容。
描述	此命令用于空白检查片内闪存的一个或多个扇区。要空白检查单一的扇区，使用相同的“起始”和“结束”扇区号。

45.9.5 读取器件识别号

表 1018. IAP 读取器件识别号命令

命令	读取器件识别号
输入	**命令代码：54 （十进制）** **参数：**无
返回码	CMD_SUCCESS |
结果	**结果 0：**器件标识号。
描述	此命令用于读取器件识别号。返回的值是器件 ID 的十六进制格式。参见表 1008“LPC43xx 器件标识号”。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

45.9.6 读取启动代码版本号

表 1019. IAP 读 Boot 代码版本号命令

命令	读 Boot 代码版本号
输入	**命令代码：55 （十进制）** **参数：**无
返回码	CMD_SUCCESS |
结果	**结果 0：**2 字节 Boot 代码版本号 （ASCII 格式）。将其解释为 < 字节 1 （主） >.< 字节 0 （次） >
描述	此命令用于读取引导代码版本号。

45.9.7 读取设备序列号

表 1020. IAP 读取设备序列号命令

命令	读取设备序列号
输入	**命令代码：58 （十进制）** **参数：**无
返回码	CMD_SUCCESS |
结果	**结果 0：**设备识别号的前 32 位字 （在最下面的地址） **结果 1：**设备识别号前 33-64 位字 **结果 2：**设备识别号前 65-96 位字 **结果 3：**设备识别号前 97-128 位字
描述	此命令用于读取设备识别号。序列号可用于唯一性地确定所有 LPC43xx 设备的单一单元。

45.9.8 比较 < 地址 1> < 地址 2> < 字节数 >

表 1021. IAP 比较命令

命令	比较
输入	**命令代码：56 （十进制）** **参数 0(DST)：**要比较的数据字节的起始 Flash 或 RAM 地址。此地址应是字边界。 **参数 1(SRC)：**要比较的数据字节的起始 Flash 或 RAM 地址。此地址应是字边界。 **参数 2：**待比较的字节数；计数值应当为 4 的倍数。
返回码	CMD_SUCCESS | COMPARE_ERROR COUNT_ERROR （字节数不是 4 的倍数）| ADDR_ERROR | ADDR_NOT_MAPPED
结果	**结果 0：**当状态代码为 COMPARE_ERROR 时第一个不匹配字节的偏移地址。
描述	此命令用于比较两个位置的内存内容。 **当源或目标地址包含从地址 0 开始的前 64 字节中的任意一个地址时，比较的结果可能不正确。因为前 64 字节是可以重新映射到 RAM 中的。**

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

45.9.9 重新调用 ISP

表 1022. 重新调用 ISP

命令	比较
输入	**命令代码：57 （十进制）**
返回码	无
结果	**无。**
描述	该命令用来调用 ISP 模式中的引导装载程序。它映射引导向量，将时钟频率设置为 96 MHz，配置 UART0 引脚 U0_RX 和 U0_TX，并复位 TIMER1 和 U0FDR（参见表 817）。当内部闪存中存在有效的用户程序且不可访问 P2_7 引脚以强制进入 ISP 模式时，可使用此命令。此命令不禁用 PLL1，因此当器件运行到 PLL1 外时，可以调用启动引导程序。在这种情况下，自动波特率信号握手后， ISP 工具必须传递 PLL1 输出频率。 另外一种选择是在调用此 IAP 前，禁用 PLL1 并选择 IRC 作为时钟源。这种情况下，ISP 发送的频率将被忽略，且使用 IRC 和 PLL1 产生一个 14.748 MHz 的时钟。

45.9.10 IAP 状态代码

表 1023. IAP 状态代码小结

状态码	助记符	描述
0	CMD_SUCCESS	命令执行成功。
1	INVALID_COMMAND	无效命令。
2	SRC_ADDR_ERROR	源地址不在字边界。
3	DST_ADDR_ERROR	目标地址不在正确的边界。
4	SRC_ADDR_NOT_MAPPED	源地址没有映射在内存映像图中。适用时应考虑计数值。
5	DST_ADDR_NOT_MAPPED	目标地址没有映射在内存映像图中。适用时应考虑计数值。
6	COUNT_ERROR	计数的字节不是 4 的倍数或允许值。
7	INVALID_SECTOR	扇区号无效。
8	SECTOR_NOT_BLANK	扇区非空。
9	SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION	准备写操作扇区的命令没被执行。
10	COMPARE_ERROR	源数据和目标数据不相同。
11	BUSY	闪存编程硬件接口忙。

45.10 JTAG 闪存编程接口

调试工具可以将部分闪存文件写入 RAM，然后使用适当的偏移重复执行 IAP 调用 "Copy RAM to Flash"。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

45.11 闪存签名生成

闪存模块包含内置的签名生成器。此生成器可从一系列闪存中生成 128 位的签名。典型的用途是根据计算签名验证闪存的内容 （例如编程期间）。

用于生成签名的地址范围必须在闪字边界对齐，例如，128 位边界。一旦开始，签名生成将独立完成。在签名生成过程中，不允许出于其它目的访问闪存，而且试图读取会导致断言等待状态，直至签名生成完成。签名生成期间，可执行闪存外部代码 （例如内部 RAM）。如果中断向量表被重新映射到闪存外的其它内存中，这也可包括中断服务。发起签名生成的代码也应放在闪存外面。

45.11.1 用于生成签名的寄存器描述

表 1024. 寄存器简介：FMC （基址 0x4008 4000）

名称	描述	访问类型	复位值	地址	参考
FMSSTART	签名起始地址寄存器	R/W	0	0x4008 4020	表 1025
FMSSTOP	签名停止地址寄存器	R/W	0	0x4008 4024	表 1026
FMSW0	128 位签名字 0	R	-	0x4008 402C	表 1027
FMSW1	128 位签名字 1	R	-	0x4008 4030	表 1028
FMSW2	128 位签名字 2	R	-	0x4008 4034	表 1029
FMSW3	128 位签名字 3	R	-	0x4008 4038	表 1030
FMSTAT	签名生成状态寄存器	R	0	0x4008 4FE0	第 45.11.1.3 节
FMSTATCLR	签名生成状态清除寄存器	W	-	0x4008 4FE8	第 45.11.1.4 节

45.11.1.1 签名生成地址和控制寄存器

这些寄存器控制自动签名生成，可为闪存内容的任何部分生成签名。用于生成签名的地址范围通过向起始地址寄存器(FMSSTART)写入起始地址，以及向停止地址寄存器(FMSSTOP)写入停止地址来定义。起始地址和停止地址必须与 128 位边界对齐，而且可通过将字节地址除以 16 得到。

签名生成通过设置 FMSSTOP 寄存器中的 SIG_START 位开始。在单一写操作中，SIG_START 位的设置通常与签名停止地址相结合。

表 1025 和表 1026 分别显示了 FMSSTART 和 FMSSTOP 寄存器中的位分配。

表 1025. 闪存模块签名起始寄存器 (FMSSTART - 0x4008 4020) 位说明

位	符号	描述	复位值
31:17	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
16:0	START	签名生成起始地址 （对应于 AHB 字节地址位 [20:4]）。	0

表 1026. 闪存模块签名停止寄存器 (FMSSTOP - 0x4008 4024) 位描述

位	符号	值	描述	复位值
31:18	-		保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
17	SIG_START		签名生成起始控制位	0
		0	签名生成停止。	
		1	发起签名生成。	
16:0	STOP		除以 16 的 BIST 停止地址 （对应于 AHB 字节地址 [20:4]）。	0

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

45.11.1.2 签名生成结果寄存器

签名生成结果寄存器返回内置生成器生成的闪存签名，此 128 位签名由四个寄存器 FMSW0、 FMSW1、 FMSW2 和 FMSW3 反映。

生成的闪存签名可用于验证闪存内容。可将生成的签名与预期签名相比，从而节省时间和代码空间。生成签名的方法在第 45.11.2 节中说明。

表 1030 分别显示了寄存器 FMSW0、 FMSW1、 FMSW2 和 FMSW3 的位分配。

表 1027. FMSW0 寄存器位描述 （FMSW0，地址：0x4008 402C)

位	符号	描述	复位值
31:0	SW0[31:0]	128 位签名的字 0 （位 31 到 0）。	-

表 1028. FMSW1 寄存器位描述 （FMSW1，地址：0x4008 4030)

位	符号	描述	复位值
31:0	SW1[63:32]	128 位签名的字 1 （位 63 到 32）。	-

表 1029. FMSW2 寄存器位描述 （FMSW2，地址：0x4008 4034)

位	符号	描述	复位值
31:0	SW2[95:64]	128 位签名的字 2 （位 95 到 64）。	-

表 1030. FMSW3 寄存器位描述 （FMSW3，地址：0x4008 4038)

位	符号	描述	复位值
31:0	SW3[127:96]	128 位签名的字 3 （位 127 到 96）。	-

45.11.1.3 闪存模块状态寄存器 (FMSTAT - 0x0x4008 4FE0)

FMSTAT 只读寄存器提供了一种确定签名生成是否完成的方法。签名生成的完成情况可通过轮询 FMSTAT 中的 SIG_DONE 位来检查。在开始签名生成操作前， SIG_DONE 应通过 FMSTATCLR 寄存器清除，否则其状态可能表示先前操作已完成。

表 1031. 闪存模块状态寄存器 (FMSTAT - 0x4008 4FE0) 位描述

位	符号	描述	复位值
31:2	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
2	SIG_DONE	其值为 1 时，先前启动的签名生成已完成。有关清除此标志的信息，参见 FMSTATCLR 寄存器说明。	0
1:0	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

45.11.1.4 闪存模块状态清除寄存器 (FMSTATCLR - 0x0x4008 4FE8)

FMSTATCLR 寄存器用于清除签名生成完成标志。

表 1032. 闪存模块状态清除寄存器 (FMSTATCLR - 0x0x4008 4FE8) 位描述

位	符号	描述	复位值
31:2	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用
2	SIG_DONE_CLR	向此位写入 1 将清除 FMSTAT 寄存器中的签名生成完成标志 (SIG_DONE) 。	0
1:0	-	保留，用户软件不应向保留位写入 1。未定义从保留位读取的值。	不适用

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

45.11.2 签名生成的算法和步骤

签名生成

可为闪存内容的任何部分生成签名。用于生成签名的地址范围通过向 FMSSTART 寄存器写入起始地址，以及向 FMSSTOP 寄存器写入停止地址来定义。

签名生成通过向 FMSSTOP.MISR_START 写入 1 开始。签名生成的开始通常与停止地址的定义相结合，后者在同一寄存器的另一个字段 FMSSTOP.FMSSTOP 中完成。

签名生成占用的时间与用于生成签名的地址范围成正比。为签名生成而读取闪存时，使用了自定时读取机制且不依赖任何可配置的闪存定时设置。安全的签名生成持续时间估计为：

时间 = int((60 / tcy) + 3) x (FMSSTOP - FMSSTART + 1)

通过软件启动签名生成时，持续时间以 AHB 时钟周期为单位， tcy 是一个 AHB 时钟的时间，单位为 ns。软件可通过轮询 FMSTAT 中的 SIG_DONE 位来确定签名完成的时间。

通过 JTAG 启动签名生成时，持续时间以 JTAG tck 周期为单位， tcy 是一个 JTAG 时钟的时间，单位为 ns。这种情况下不可轮询 FMSTAT 中的 SIG_DONE 位。

签名生成后，可从寄存器 FMSW0 到 FMSW3 中读取 128 位的签名。此 128 位签名反映从闪存中读取的正确数据。此 128 位签名反映闪存奇偶校验位并检查位值。

内容验证

从寄存器 FMSW0 到 FMSW3 读取的签名必须与参考签名相等。图 174 给出了推算参考签名的算法。

```
sign = 0
FOR address = FMSTART.FMSTART TO FMSTOP.FMSTOP
{
        FOR i = 0 TO 126
                nextSign[i] = f_Q[address][i] XOR sign[i+1]
                nextSign[127] = f_Q[address][127] XOR sign[0] XOR sign[2] XOR
                        sign[27] XOR sign[29]
        sign = nextSign
}
signature128 = sign
```

图 174. 生成 128 位签名的算法

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

46.1 本章导读

所有 LPC43xx 器件的调试功能都相同。并行跟踪端口不适用于 100 引脚封装。

46.2 基本配置

- 选择调试模式：
  - DBGEN 引脚位于高电平时，进入 ARM 调试模式。
  - DBGEN 引脚位于低电平时，进入 JTAG BSDL。
- 内核频率必须小于 80 MHz 才可使用 SWO 功能。
- ETM 跟踪捕获速度受跟踪引脚特性所限。在引脚配置寄存器中为多路复用跟踪引脚设置了 EHS 位时 （表 123），最大速度为 80 MHz。
- 对于 JTAG 编程，引脚 RTC_ALARM 必须设为低电平。
- 在 CREG 模块中使能 ETB SRAM （参见第 9.4.5 节）。

46.3 功能要点

- 支持标准 JTAG 和 ARM 串行调试接口模式。
- 可直接对所有存储器、寄存器和外设进行调试。
- 调试阶段不需要目标资源。
- 跟踪端口提供 CPU 指令跟踪功能。可通过 4 位跟踪数据端口或“串行线浏览器”输出。
- 八个断点。六个指令断点，也可用于重新映射代码补丁的指令地址。两个数据比较器，可用于重新映射文字值补丁的地址。
- 四个数据观察点，也可用作跟踪启动器。
- “仪表跟踪宏单元”允许额外的软件控制跟踪。

46.4 描述

调试和跟踪功能集成到 ARM Cortex-M4 中。除了标准 JTAG 调试和并行跟踪功能外，还支持串行调试接口和跟踪功能。 ARM Cortex-M4 的配置可支持最多八个断点和四个观察点。

JTAG 默认使用 LPC43xx 进行调试。进入 JTAG 调试模式后，调试工具可切换到 “串行调试接口”模式。

可使用 4 位并行接口或 “串行线输出”执行跟踪。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

46.5 引脚说明

表 1033 至表 1035 显示了与调试和跟踪有关的各种引脚功能。其中有些功能与其它功能共享引脚，因此不能同时使用。使用 JTAG 端口就不能使用 “串行调试接口” 和 “串行线输出”。使用并行跟踪需要 5 个引脚，这些引脚可以是用户应用的一部分，限制了对相应功能的调试能力。使用 “串行线输出” 进行跟踪则无此限制，但带宽有限。

表 1033. JTAG 引脚描述

引脚名称	类型	描述
TCK	输入	**JTAG 测试时钟。**此引脚是 JTAG 调试模式中调试逻辑的时钟。
TMS	输入	**JTAG 测试模式选择。** TMS 引脚选择 TAP 状态机中的下一状态。
TDI	输入	**JTAG 测试数据输入。**这是移位寄存器的串行数据输入。
TDO	输出	**JTAG 测试数据输出。**这是移位寄存器的串行数据输出，数据在 TCK 信号的负边沿移出设备。
$\overline{\text{TRST}}$	输入	**JTAG 测试复位。** $\overline{\text{TRST}}$ 引脚可用于复位调试逻辑内的测试逻辑。

表 1034. 串行调试接口引脚说明

引脚名称	类型	描述
SWDCLK	输入	**串行接口时钟。**此引脚是 “串行调试接口” 模式中调试逻辑的时钟。
SWDIO	输入/输出	**串行调试接口数据输入 / 输出。**外部调试工具使用 SWDIO 引脚与 Cortex-M4 CPU 进行通信并控制它。
SWO	输出	**串行线输出。**SWO 引脚也可提供 ITM 和 / 或 ETM 数据，供外部调试工具评估。 **注：**内核频率必须小于 80 MHz 才可使用 SWO。

表 1035. 并行跟踪引脚描述

引脚名称	类型	描述
TRACECLK	输入	**跟踪时钟。**外部调试工具使能跟踪后，此引脚为 TRACEDATA 引脚上的跟踪数据提供采样时钟。
TRACEDATA[3:0]	输出	**跟踪数据位 3 至 0。**外部调试工具使能跟踪时，这些引脚将提供 ETM 跟踪数据。然后，调试工具可以解析压缩信息并提供给用户。

46.6 调试连接

LPC43xx 为 JTAG 和串行调试接口 (SWD) 提供专用引脚。调试会话开始后，该器件将处于 JTAG 调试模式。进入调试模式后，调试器可将设备切换到 SWD 模式。

从目标板到调试器的连接可能有所变化。添加到新电路板设计的调试连接器的选择取决于将要使用的调试工具。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

46.6.1 ARM 标准 JTAG 连接器 （20 引脚）

图 175 显示标准 JTAG 连接器。 ARM 标准 JTAG 连接器支持 20 引脚 (0.1") 连接器的串行线和 JTAG 接口模式。它可用于访问所有 SWD、 SWV 和 JTAG 信号。

图 175. ARM 标准 JTAG 连接器

46.6.2 Cortex 调试连接器 （10 引脚）

图 176 显示 Cortex 调试连接器（10 引脚）该连接器支持极小连接器中的串行线和 JTAG。10 引脚 Cortex 调试连接器可访问所有 SWD、 SWV 和 JTAG 信号。

图 176. Cortex 调试连接器

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

46.6.3 Cortex 调试 + ETM 连接器 （20 引脚）

如果将要使用调试跟踪功能，则还有 “调试和跟踪” 连接器规范，如图 177 所示。这种小型 20 引脚 (0.05") 连接器可以访问 SWD、 SWV、 JTAG 和 ETM （4 位）信号。

图 177. Cortex 调试 + ETM 连接器

46.7 调试注意事项

注意事项：用户应了解调试期间的某些限制。特别重要的是，由于 Cortex-M4 集成的限制，LPC43xx 不能以常规方式从 “深度睡眠” 和 “掉电” 模式中唤醒。在调试期间不建议使用这些模式。

另外，调试模式会改变 Cortex-M4 CPU 处理掉电模式的方式。这可能使设备级别的电源模式不同于正常模式操作。这些差异意味着在调试过程中不应该进行功率测量，其结果会高于正常运行的情况。

在调试期间，无论 CPU 何时停止，“系统节拍定时器” 和 “重复性中断定时器” 都将自动停止。其他外设不受影响。如果 “重复性中断定时器” 的配置使时钟速率低于 CPU 时钟速率，则在某些调试操作中 （如单步骤运行）， RIT 可能不会按预期递增。

代码读取保护使能时，调试禁用。

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

46.8 调试存储器重新映射

在芯片复位之后，部分 Boot ROM 映射至地址 0，以便它可以自动执行。Boot ROM 将镜像切换至指向 0x1000 0000 或 0x1C00 0000（从 EMC 启动时）或者 0x8000 0000（从 SPIFI 启动时）。可以使用 M4 存储器映射寄存器从地址 0x0000 0000 开始执行编码 （表 39）。

寄存器映射通常不向用户公开。但是，如果调试器在复位后立即停止了 CPU 执行， Boot ROM 仍然映射至地址 0，则可能造成混淆。理想情况是，调试器应在此时自动修正映射，以便用户无需处理该问题。

46.9 JTAG TAP 识别

JTAG TAP 控制器包含设备 ID，调试软件可用它来识别设备的一般类型。

表 1036. JTAG TAP 识别

模式	ID 代码
JTAG 模式	0x4BA0 0477
SWD 模式	0x2BA0 1477
Cortex-M0 协处理器	0x0BA0 1477

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

47.1 本章导读

ARM Cortex-M0 和 ARM Cortex-M4 规范的详情请见以下文档：

- *Cortex-M4 器件通用用户指南*
- *Cortex-M0 器件通用用户指南*

47.2 Cortex-M4 指令集汇总

表 1037 显示 Cortex-M4 指令集，带以下用于计算周期数的缩略语：

- P = 流水线补充所需的周期数。其范围在 1-3 内，取决于目标指令的对齐和宽度，以及处理器能否在早期设法推测地址。
- B = 执行屏障操作所需的周期数。对于 DSB 和 DMB，最小周期数是 0。对于 ISB，最小周期数和流水线补充所需的数量相同。
- N = 寄存器列表中需要加载或存储的寄存器数量，包括 PC 或 LR。
- W = 等待合适事件所花费的周期数。

表 1037. Cortex-M4 指令集汇总

操作	描述	汇编程序	周期
传输	寄存器	MOV Rd, <op2>	1
	16 位立即数	MOVW Rd, #<imm>	1
	立即数到高位	MOVT Rd, #<imm>	1
	至 PC	MOV PC, Rm	1 + P
加法	加法	ADD Rd, Rn, <op2>	1
	加到 PC	ADD PC, PC, Rm	1 + P
	进位加法	ADC Rd， Rn， <op2>	1
	格式地址	ADR Rd, <label>	1
减法	减法	SUB Rd, Rn, <op2>	1
	借位减法	SBC Rd, Rn, <op2>	1
	反向	RSB Rd, Rn, <op2>	1
乘法	乘法	MUL Rd, Rn, Rm	1
	累加乘法	MLA Rd, Rn, Rm	1
	乘减	MLS Rd, Rn, Rm	1
	带符号的长数	SMULL RdLo, RdHi, Rn, Rm	1
	不带符号的长数	UMULL RdLo, RdHi, Rn, Rm	1
	带符号的长数累加	SMLAL RdLo, RdHi, Rn, Rm	1
	不带符号的长数累加	UMLAL RdLo, RdHi, Rn, Rm	1

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 1037. Cortex-M4 指令集汇总 *（续）*

操作	描述	汇编程序	周期
除法	带符号	SDIV Rd, Rn, Rm	2 至 12[1]
	不带符号	UDIV Rd, Rn, Rm	2 至 12[[1]
饱和	带符号	SSAT Rd, #<imm>, <op2>	1
	不带符号	USAT Rd, #<imm>, <op2>	1
比较	比较	CMP Rn, <op2>	1
负数	CMN Rn, <op2>	1	
逻辑	AND	AND Rd, Rn, <op2>	1
	异或	EOR Rd, Rn, <op2>	1
	OR	ORR Rd, Rn, <op2>	1
	OR NOT	ORN Rd, Rn, <op2>	1
	位清除	BIC Rd, Rn, <op2>	1
	取反传输	MVN Rd, <op2>	1
	与测试	TST Rn, <op2>	1
	异或测试	TEQ Rn, <op1>	
移位	逻辑左移	LSL Rd, Rn, #<imm>	1
	逻辑左移	LSL Rd, Rn, Rs	1
	逻辑右移	LSR Rd, Rn, #<imm>	1
	逻辑右移	LSR Rd, Rn, Rs	1
	算术右移	ASR Rd, Rn, #<imm>	1
	算术右移	ASR Rd, Rn, Rs	1
循环	循环右移	ROR Rd, Rn, #<imm>	1
	循环右移	ROR Rd, Rn, Rs	1
	带扩展	RRX Rd, Rn	1
计数	首位的 0	CLZ Rd, Rn	1
负载	字	LDR Rd, [Rn, <op2>]	2[2]
	至 PC	LDR PC, [Rn, <op2>]	2[2] + P
	半字	LDRH Rd, [Rn, <op2>]	2[2]
	字节	LDRB Rd, [Rn, <op2>]	2[2]
	带符号半字	LDRSH Rd, [Rn, <op2>]	2[2]
	带符号字节	LDRSB Rd, [Rn, <op2>]	2[2]
	用户字	LDRT Rd, [Rn, #<imm>]	2[2]
	用户半字	LDRHT Rd, [Rn, #<imm>]	2[2]
	用户字节	LDRBT Rd, [Rn, #<imm>]	2[2]
	用户带符号半字	LDRSHT Rd, [Rn, #<imm>]	2[2]
	用户带符号字节	LDRSBT Rd, [Rn, #<imm>]	2[2]
	相对 PC	LDR Rd,[PC,#<imm>]	2[2]
	双字	LDRD Rd, Rd, [Rn, #<imm>]	1 + N
	多重	LDM Rn, {<reglist>}	1 + N
	包括 PC 的多重	LDM Rn, {<reglist>, PC}	1 + N + P

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

表 1037. Cortex-M4 指令集汇总 *（续）*

操作	描述	汇编程序	周期
存储	字	STR Rd, [Rn, <op2>]	2[2]
	半字	STRH Rd, [Rn, <op2>]	2[2]
	字节	STRB Rd, [Rn, <op2>]	2[2]
	带符号半字	STRSH Rd, [Rn, <op2>]	2[2]
	带符号字节	STRSB Rd, [Rn, <op2>]	2[2]
	用户字	STRT Rd, [Rn, #<imm>]	2[2]
	用户半字	STRHT Rd, [Rn, #<imm>]	2[2]
	用户字节	STRBT Rd, [Rn, #<imm>]	2[2]
	用户带符号半字	STRSHT Rd, [Rn, #<imm>]	2[2]
	用户带符号字节	STRSBT Rd, [Rn, #<imm>]	2[2]
	双字	STRD Rd, Rd, [Rn, #<imm>]	1 + N
	多重	STM Rn, {<reglist>}	1 + N
压栈	压栈	PUSH {<reglist>}	1 + N
	用链路寄存器压栈	PUSH {<reglist>, LR}	1 + N
出栈	出栈	POP {<reglist>}	1 + N
	出栈和返回	POP {<reglist>, PC}	1 + N + P
信号量	加载独占	LDREX Rd, [Rn, #<imm>]	2
	加载独占半字	LDREXH Rd, [Rn]	2
	加载独占字节	LDREXB Rd, [Rn]	2
	存储独占	STREX Rd, Rt, [Rn, #<imm>]	2
	存储独占半字	STREXH Rd, Rt, [Rn]	2
	存储独占字节	STREXB Rd, Rt, [Rn]	2
	清除独占监控器	CLREX	1
分支	条件性	B<cc> <label>	1 或 1 + P[3]
	非条件性	B <label>	1 + P
	带链路	BL <label>	1 + P
	带交换	BX Rm	1 + P
	带链路和交换	BLX Rm	1 + P
	如果为 0，则分支	CBZ Rn, <label>	1 或 1 + P[3]
	如果为非 0，则分支	CBNZ Rn, <label>	1 或 1 + P[3]
	字节表分支	TBB [Rn, Rm]	2 + P
	半字表分支	TBH [Rn, Rm, LSL#1]	2 + P
	状态变化	超级用户调用	SVC #<imm>
	If-then-else	IT... <cond>	1[4]
	禁用中断	CPSID <flags>	1 或 2
	使能中断	CPSIE <flags>	1 或 2
	对特殊寄存器进行读操作	MRS Rd, <specreg>	1 或 2
	对特殊寄存器进行写操作	MSR <specreg>, Rn	1 或 2
	断点	BKPT #<imm>	-

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 1037. Cortex-M4 指令集汇总 *（续）*

操作	描述	汇编程序	周期
延长	带符号半字到字	SXTH Rd, <op2>	1
	带符号字节到字	SXTB Rd, <op2>	1
	不带符号半字	UXTH Rd, <op2>	1
	不带符号字节	UXTB Rd, <op2>	1
位字段	提取不带符号	UBFX Rd, Rn, #<imm>, #<imm>	1
	提取带符号	SBFX Rd, Rn, #<imm>, #<imm>	1
	清除	BFC Rd, Rn, #<imm>, #<imm>	1
	插入	BFI Rd, Rn, #<imm>, #<imm>	1
反向	字中的字节	REV Rd, Rm	1
	两个半字中的字节	REV16 Rd, Rm	1
	带符号低半字	REVSH Rd, Rm	1
	字位	RBIT Rd, Rm	1
提示	发送事件	SEV	1
	等待事件	WFE	1 + W
	等待中断	WFI	1 + W
	无操作	NOP	1
屏障	指令同步	ISB	1 + B
	数据存储器	DMB	1 + B
	数据同步	DSB <flags>	1 + B

[1] 根据输入操作数中的前导 1 和前导 0 的数量，除法运算使用提前终止将所需的周期数最小化。

[2] 相邻加载和存储单指令可以通过流水线传输其地址和数据阶段，使这些指令在单个执行周期内完成。

[3] 如果不采用分支，条件分支将在单个周期内完成。

[4] IT 指令可以折叠到之前的 16 位 Thumb 指令上 , 实现在零周期内执行。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 1038. Cortex-M4 DSP 指令集汇总

操作	描述	汇编程序	周期
乘法	带 32 位最高有效位累加的 32 位乘法	SMMLA	1
	带 32 位最高有效位减法的 32 位乘法	SMMLS	1
	返回 32 位最高有效位的 32 位乘法	SMMUL	1
	带舍入 32 位最高有效位累加的 32 位乘法	SMMLAR	1
	带舍入 32 位最高有效位减法的 32 位乘法	SMMLSR	1
	返回舍入 32 位最高有效位的 32 位乘法	SMMULR	1
带符号乘法	Q 设置：带 32 位累加的 16 位带符号乘法，低位乘低位	SMLABB	1
	Q 设置：带 32 位累加的 16 位带符号乘法，低位乘高位	SMLABT	1
	带 64 位累加的 16 位带符号乘法，低位乘低位	SMLALBB	1
	带 64 位累加的 16 位带符号乘法，低位乘高位	SMLALBT	1
	带单一 64 位累加器的双 16 位带符号乘法	SMLALD{X}	1
	带 64 位累加的 16 位带符号乘法，高位乘低位	SMLALTB	1
	带 64 位累加的 16 位带符号乘法，高位乘高位	SMLALTT	1
	产生 32 位结果值的带符号 16 位乘法，低位乘低位	SMULBB	1
	产生 32 位结果值的带符号 16 位乘法，低位乘高位	SMULBT	1
	产生 32 位结果值的带符号 16 位乘法，高位乘低位	SMULTB	1
	产生 32 位结果值的带符号 16 位乘法，高位乘低位	SMULTT	1
	返回 32 位最高有效位的 16 位乘 32 位带符号乘法，低位	SMULWB	1
	返回 32 位最高有效位的 16 位乘 32 位带符号乘法，高位	SMULWT	1
	返回差值的双 16 位带符号乘法	SMUSD{X}	1
	Q 设置：带 32 位累加的 16 位带符号乘法，高位乘低位	SMLATB	1
	Q 设置：带 32 位累加的 16 位带符号乘法，高位乘高位	SMLATT	1
	Q 设置：带单一 32 位累加器的双 16 位带符号乘法	SMLAD{X}	1
	Q 设置：带 32 位累加的 16 位乘 32 位带符号乘法，低位	SMLAWB	1
	Q 设置：带 32 位累加的 16 位乘 32 位带符号乘法，高位	SMLAWT	1
	Q 设置：带 32 位累加的双 16 位带符号乘减	SMLSD{X}	1
	Q 设置：带 64 位累加的双 16 位带符号乘减	SMLSLD{X}	1
	Q 设置：双 16 位乘法之和	SMUAD{X}	1
不带符号乘法	产生 64 位结果、带双 32 位累加的 32 位不带符号乘法	UMAAL	1
饱和	Q 设置：双 16 位饱和	SSAT16	1
	Q 设置：双 16 位不带符号饱和	USAT16	1

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 1038. Cortex-M4 DSP 指令集汇总 *（续）*

操作	描述	汇编程序	周期
打包和拆分	打包带移位低位的半字高位	PKHTB	1
	打包带移位高位的半字低位	PKHBT	1
	提取 8 位并将值用符号扩展到 32 位	SXTB	1
	双重提取 8 位并将各个值用符号扩展到 16 位	SXTB16	1
	提取 16 位并将值用符号扩展到 32 位	SXTH	1
	提取 8 位并将值用 0 扩展到 32 位	UXTB	1
	双重提取 8 位并将值用 0 扩展到 16 位	UXTB16	1
	提取 16 位并将值用 0 扩展到 32 位	UXTH	1
	提取 8 位至 32 位不带符号加法	UXTAB	1
	双重提取 8 位至 16 位不带符号加法	UXTAB16	1
	提取 16 位至 32 位不带符号加法	UXTAH	1
	提取 8 位到 32 位带符号加法	SXTAB	1
	双重提取 8 位至 16 位带符号加法	SXTAB16	1
	提取 16 位到 32 位带符号加法	SXTAH	1
其他数据处理	选择基于 GE 位的字节	SEL	1
	四 8 位不带符号绝对差值的不带符号和	USAD8	1
	带 32 位累加的四 8 位不带符号绝对差值的不带符号和	USADA8	1
加法	双 16 位不带符号饱和加法	UQADD16	1
	四 8 位不带符号饱和加法	UQADD8	1
	Q 设置：饱和加法	QADD	1
	Q 设置：双 16 位饱和加法	QADD16	1
	Q 设置：四 8 位饱和加法	QADD8	1
	Q 设置：饱和倍和加	QDADD	1
	GE 设置：四 8 位带符号加法	SADD8	1
	GE 设置：双 16 位带符号加法	SADD16	1
	带二等分结果的双 16 位带符号加法	SHADD16	1
	带二等分结果的四 8 位带符号加法	SHADD8	1
	GE 设置：双 16 位不带符号加法	UADD16	1
	GE 设置：四 8 位不带符号加法	UADD8	1
	带二等分结果的双 16 位不带符号加法	UHADD16	1
	带二等分结果的四 8 位不带符号加法	UHADD8	1

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 1038. Cortex-M4 DSP 指令集汇总 *（续）*

操作	描述	汇编程序	周期
减法	Q 设置：饱和倍和减	QDSUB	1
	双 16 位不带符号饱和减法	UQSUB16	1
	四 8 位不带符号饱和减法	UQSUB8	1
	Q 设置：饱和减法	QSUB	1
	Q 设置：双 16 位饱和减法	QSUB16	1
	Q 设置：四 8 位饱和减法	QSUB8	1
	带二等分结果的双 16 位带符号减法	SHSUB16	1
	带二等分结果的四 8 位带符号减法	SHSUB8	1
	GE 设置：双 16 位带符号减法	SSUB16	1
	GE 设置：四 8 位带符号减法	SSUB8	1
	带二等分结果的双 16 位不带符号减法	UHSUB16	1
	带二等分结果的四 8 位不带符号减法	UHSUB8	1
	GE 设置：双 16 位不带符号减法	USUB16	1
	GE 设置：四 8 位不带符号减法	USUB8	1
并行加法和减法	带交换的双 16 位不带符号饱和加法和减法	UQASX	1
	带交换的双 16 位不带符号饱和减法和加法	UQSAX	1
	GE 设置：带交换的双 16 位加法和减法	SASX	1
	Q 设置：带交换的双 16 位加法和减法	QASX	1
	Q 设置：带交换的双 16 位减法和加法	QSAX	1
	带二等分结果的双 16 位带符号加法和减法	SHASX	1
	带二等分结果的双 16 位带符号减法和加法	SHSAX	1
	GE 设置：带交换的双 16 位带符号减法和加法	SSAX	1
	GE 设置：带交换的双 16 位不带符号加法和减法	UASX	1
	带二等分结果和交换的双 16 位不带符号加法和减法	UHASX	1
	带二等分结果和交换的双 16 位不带符号减法和加法	UHSAX	1
	GE 设置：带交换的双 16 位不带符号减法和加法	USAX	1

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

47.3 Cortex-M0 指令集汇总

表 1039. Cortex-M0 指令集汇总

操作	描述	汇编程序	周期
传输	8 位立即数	MOVS Rd, #<imm>	1
	Lo 至 Lo	MOVS Rd, Rm	1
	任意至任意	MOV Rd, Rm	1
	任意至 PC	MOV PC, Rm	3
加法	3 位立即数	ADDS Rd, Rn, #<imm>	1
	全部寄存器 Lo	ADDS Rd, Rn, Rm	1
	任意至任意	ADD Rd, Rd, Rm	1
	任意至 PC	ADD PC, PC, Rm	3
	8 位立即数	ADDS Rd, Rd, #<imm>	1
	带进位	ADCS Rd, Rd, Rm	1
	立即数到 SP	ADD SP, SP, #<imm>	1
	从来自 SP 的地址	ADD Rd, SP, #<imm>	1
	从来自 PC 的地址	ADR Rd, <label>	1
减法	Lo 和 Lo	SUBS Rd, Rn, Rm	1
	3 位立即数	SUBS Rd, Rn, #<imm>	1
	8 位立即数	SUBS Rd, Rd, #<imm>	1
	带进位	SBCS Rd, Rd, Rm	1
	SP 中的立即数	SUB SP, SP, #<imm>	1
	求补	RSBS Rd, Rn, #0	1
乘法	乘法	MULS Rd, Rm, Rd	32
比较	比较	CMP Rn, Rm	1
	负数	CMN Rn, Rm	1
	立即数	CMP Rn, #<imm>	1
逻辑	AND	ANDS Rd, Rd, Rm	1
	异或	EORS Rd, Rd, Rm	1
	OR	ORRS Rd, Rd, Rm	1
	位清除	BICS Rd, Rd, Rm	1
	取反传输	MVNS Rd, Rm	1
	与测试	TST Rn, Rm	1
移位	逻辑左移立即数	LSLS Rd, Rm, #<shift>	1
	逻辑左移寄存器	LSLS Rd, Rd, Rs	1
	逻辑右移立即数	LSRS Rd, Rm, #<shift>	1
	逻辑右移寄存器	LSRS Rd, Rd, Rs	1
	算术右移	ASRS Rd, Rm, #<shift>	1
	算术右移寄存器	ASRS Rd, Rd, Rs	1
循环	循环右移寄存器	RORS Rd, Rd, Rs	1

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 1039. Cortex-M0 指令集汇总 *（续）*

操作	描述	汇编程序	周期
负载	字，立即数偏移量	LDR Rd, [Rn, #<imm>]	2
	半字，立即数偏移量	LDRH Rd, [Rn, #<imm>]	2
	字节，立即数偏移量	LDRB Rd, [Rn, #<imm>]	2
	字，寄存器偏移量	LDR Rd, [Rn, Rm]	2
	半字，寄存器偏移量	LDRH Rd, [Rn, Rm]	2
	带符号半字，寄存器偏移量	LDRSH Rd, [Rn, Rm]	2
	字节，寄存器偏移量	LDRB Rd, [Rn, Rm]	2
	带符号字节，寄存器偏移量	LDRSB Rd, [Rn, Rm]	2
	相对 PC	LDR Rd, <label>	2
	相对 SP	LDR Rd, [SP, #<imm>]	2
	多重，不包括基数	LDM Rn!, {<loreglist>}	1 + N[1]
	多重，包括基数	LDM Rn, {<loreglist>}	1 + N[1]
存储	字，立即数偏移量	STR Rd, [Rn, #<imm>]	2
	半字，立即数偏移量	STRH Rd, [Rn, #<imm>]	2
	字节，立即数偏移量	STRB Rd, [Rn, #<imm>]	2
	字，寄存器偏移量	STR Rd, [Rn, Rm]	2
	半字，寄存器偏移量	STRH Rd, [Rn, Rm]	2
	字节，寄存器偏移量	STRB Rd, [Rn, Rm]	2
	相对 SP	STR Rd, [SP, #<imm>]	2
	多重	STM Rn!, {<loreglist>}	1 + N[1]
压栈	压栈	PUSH {<loreglist>}	1 + N[1]
	用链路寄存器压栈	PUSH {<loreglist>, LR}	1 + N[1]
出栈	出栈	POP {<loreglist>}	1 + N[1]
	出栈和返回	POP {<loreglist>, PC}	4 + N[2]
分支	条件性	B<cc> <label>	1 或 3[3]
	非条件性	B <label>	3
	带链路	BL <label>	4
	带交换	BX Rm	3
	带链路和交换	BLX Rm	3
延长	带符号半字到字	SXTH Rd, Rm	1
	带符号字节到字	SXTB Rd, Rm	1
	不带符号半字	UXTH Rd, Rm	1
	不带符号字节	UXTB Rd, Rm	1
反向	字中的字节	REV Rd, Rm	1
	两个半字中的字节	REV16 Rd, Rm	1
	带符号的低位半字	REVSH Rd, Rm	1
状态变化	超级用户调用	SVC <imm>	-[4]
	禁用中断	CPSID i	1
	使能中断	CPSIE i	1
	对特殊寄存器进行读操作	MRS Rd, <specreg>	4
	对特殊寄存器进行写操作	MSR <specreg>, Rn	4

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 1039. Cortex-M0 指令集汇总 *（续）*

操作	描述	汇编程序	周期
提示	发送事件	SEV	1
	等待事件	WFE	2[5]
	等待中断	WFI	2[5]
	产生	YIELD[6]	1
	无操作	NOP	1
屏障	指令同步	ISB	4
	数据存储器	DMB	4
	数据同步	DSB	4

[1] N 是元素数量。

[2] N 是包括 PC 在内的出栈列表中的元素数量，假定加载或存储不会产生 HardFault 异常。

[3] 若采用，为 3，若不采用，为 1。

[4] 周期计数取决于内核和调试配置。

[5] 不包括等待中断或事件所花费的时间。

[6] 作为 NOP 运行。

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

48.1 缩略词

表 1040. 缩略词

首字母缩略词	描述
ADC	模数转换器
AES	高级加密标准
AHB	高级高性能总线
APB	高级外设总线
API	应用程序编程接口
BOD	掉电检测
CAN	控制器局域网
CMAC	基于密文的消息认证码
CSMA/CD	带检测冲突的载波侦听多路存取
DAC	数模转换器
DC-DC	直流对直流
DMA	直接存储器访问
GPIO	通用输入 / 输出
IRC	内部 RC
IrDA	红外数值标准协会
JTAG	联合测试行动小组
LCD	液晶显示器
LSB	最低有效位
MAC	介质访问控制
MCU	微控制器
MIIM	介质独立接口管理
n.c.	未连接
OHCI	开放式主控制器接口
OTG	On-The-Go
PHY	物理层
PLL	锁相环
PMC	电源模式控制
PWM	脉冲宽度调制器
RIT	重复性中断定时器
RMII	精简的媒体独立接口
SDRAM	同步动态随机存取存储器
SIMD	单指令多数据
SPI	串行外设接口
SSI	串行同步接口
SSP	同步串口
TCP/IP	传输控制协议 / 因特网协议

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 1040. 缩略词 *（续）*

首字母缩略词	描述
TTL	晶体管 - 晶体管逻辑
UART	通用异步收发器
ULPI	UTMI+ 低引脚接口
USART	通用同步 / 异步收发器
USB	通用串行总线
UTMI	USB2.0 收发器宏单元接口

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

48.2 法律信息

48.2.1 用户手册状态

文档状态 [1]	产品状态 [2]	定义
客观用户手册	开发	该文档包含产品开发客观规范的数据。
初版用户手册	验证	该文档含有初始规范的数据。
产品用户手册	产量	该文档含有产品规范。

[1] 请在开始或完成设计之前查看最新发布文件。

[2] 自本文件发布以来，文件中的器件产品状态可能已发生变化；如果存在多个器件，则可能存在差异。欲了解最新产品状态信息，请访问 http://www.nxp.com。

48.2.2 定义

初稿 — 本文仅为初稿版本。内容仍在内部审查，尚未正式批准，可能会有进一步修改或补充。恩智浦半导体对本文信息的准确性或完整性不做任何说明或保证，并对因使用此信息而带来的后果不承担任何责任。

48.2.3 免责声明

有限担保和责任 — 本文中的信息据信是准确和可靠的。但是，恩智浦半导体对此处所含信息的准确性或完整性不做任何明示或暗示的说明或保证，并对因使用此信息而导致的后果不承担任何责任。

在任何情况下，对于任何间接、意外、惩罚性、特殊或衍生性损害（包括但不限于利润损失、积蓄损失、业务中断、因拆卸或更换任何产品而产生的开支或返工费用），无论此等损害是否基于侵权行为（包括过失）、担保、违约或任何其他法理，恩智浦半导体均不承担任何责任。

对于因任何原因给客户带来的任何损害，恩智浦半导体对本文所述产品的总计责任和累积责任仅限于恩智浦*商业销售条款和条件*所规定的范围。

修改权利 — 恩智浦半导体保留对本文所发布的信息（包括但不限于规格和产品说明）随时进行修改的权利，恕不另行通知。本文件将取代并替换之前就此提供的所有信息。

适宜使用 — 恩智浦半导体产品并非设计、授权或担保适合用于生命保障、生命关键或安全关键系统或设备，亦非设计、授权或担保适合用于在恩智浦半导体产品失效或故障时会导致人员受伤、死亡或严重财产或环境损害的应用。恩智浦半导体对在此类设备或应用中加入和 / 或使用恩智浦半导体产品不承担任何责任，客户需自行承担因加入和 / 或使用恩智浦半导体产品而带来的风险。

应用 — 本文件所载任何产品的应用只用于例证目的。此类应用如不经进一步测试或修改用于特定用途，恩智浦半导体对其适用性不做任何说明或保证。

客户负责自行利用恩智浦半导体的产品进行设计和应用，对于应用或客户产品设计，恩智浦半导体无义务提供任何协助。客户须自行负责检验恩智浦半导体的产品是否适用于其规划的应用和产品，以及是否适用于其第三方客户的规划应用和使用。客户须提供适当的设计和操作安全保障措施，以降低与应用和产品相关的风险。

对于因客户应用或产品的任何缺陷或故障，或者客户的第三方客户的应用或使用导致的任何故障、损害、开支或问题，恩智浦半导体均不承担任何责任。客户负责对自己基于恩智浦半导体的产品的应用和产品进行所有必要测试，以避免这些应用和产品或者客户的第三方客户的应用或使用存在任何缺陷。恩智浦不承担与此相关的任何责任。

出口管制 — 本文件以及此处所描述的产品可能受出口法规的管制。出口可能需要事先经相关主管部门批准。

48.2.4 商标

注意：所有引用的品牌、产品名称、服务名称以及商标均为其各自所有者的资产。

I²C 总线 — 标志是恩智浦的商标。

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

48.3 表

表 1. 订购信息 6
表 2. 订购选项 6
表 3. ARM Cortex-M0 时钟和电源控制 8
表 4. 命令列表 11
表 5. 消息列表 11
表 6. 命令响应 11
表 7. IPC 示例 12
表 8. LPC4350/30/20/10 SRAM 配置 13
表 9. OTP 存储器描述（OTP 基址 0x4004 5000） 18
表 10. OTP 存储组别 3，字 0 – 客户控制数据（地址偏移 0x030） 19
表 11. OTP 存储组别 3，字 1 - 通用 OTP 存储组别 2，字 0 或 USB ID（地址偏移 0x034） 19
表 12. OTP 存储组别 3，字 2 - 通用 OTP 存储器 2，字 1 或以太网 MAC（地址 0x038） 19
表 13. ROM 驱动器指针（主要 API 输入点 0x1040 0100） 20
表 14. OTP 功能分配 21
表 15. OTP BOOT_SRC 位已编程时的引导模式 . . . 23
表 16. OTP BOOT_SRC 位为零时的启动模式 23
表 17. 映像标头 26
表 18. 引导流程时序参数 31
表 19. 安全 API 调用 35
表 20. NVIC 引脚描述 41
表 21. 中断源与 Cortex-M4 NVIC 的连接 42
表 22. 中断源与 Cortex-M0 NVIC 的连接 44
表 23. 事件路由器时钟和电源控制 45
表 24. 事件路由器输入 46
表 25. 事件路由器引脚描述 47
表 26. 寄存器简介：事件路由器（基址 0x4004 4000） 47
表 27. 电平配置寄存器（HILO – 地址 0x4004 4000）位描述 47
表 28. EDGE 和 HILO 寄存器设置组合 50
表 29. 边沿配置寄存器（EDGE - 地址 0x4004 4004）位描述 50
表 30. 清除事件使能寄存器（CLR_EN – 地址 0x4004 4FD8）位描述 52
表 31. 事件设置使能寄存器（SET_EN – 地址 0x4004 4FDC）位描述 53
表 32. 事件状态寄存器（STATUS – 地址 0x4004 4FE0）位描述 54
表 33. 事件使能寄存器（ENABLE – 地址 0x4004 4FE4）位描述 55
表 34. 清除事件状态寄存器（CLR_STAT – 地址 0x4004 4FE8）位描述 56
表 35. 设置事件状态寄存器（SET_STAT – 地址 0x4004 4FEC）位描述 57
表 36. CREG 时钟和电源控制 58
表 37. 寄存器简介：配置寄存器（基址 0x4004 3000） 59
表 38. CREG0 寄存器（CREG0，地址 0x4004 3004）位描述 60
表 39. 存储器映射寄存器（M4MEMMAP，地址 0x4004 3100）位描述 61
表 40. CREG5 控制寄存器（CREG5，地址 0x4004 3118）位描述 61
表 41. DMA 复用寄存器（DMAMUX，地址 0x4004 311C）位描述 62
表 42. ETB SRAM 配置寄存器（ETBCFG，地址 0x4004 3128）位描述 64
表 43. CREG6 控制寄存器（CREG6，地址 0x4004 312C）位描述 65
表 44. M4 TXEV 清除寄存器（M4TXEVENT，地址 0x4004 3130）位描述 66
表 45. 器件 ID 寄存器（CHIPID，地址 0x4004 3200）位描述 66
表 46. Cortex-M0 TXEV 清除寄存器（M0TXEVENT，地址 0x4004 3400）位描述 66
表 47. 存储器映射寄存器（M0APPMEMMAP，地址 0x4004 3404）位描述 66
表 48. 存储器保持 69
表 49. 寄存器简介：功耗模式控制器 (PMC)（基址 0x4004 2000） 69
表 50. 硬件睡眠事件使能寄存器（PD0_SLEEP0_HW_ENA – 地址 0x4004 2000）位描述 69
表 51. 掉电模式寄存器（PD0_SLEEP0_MODE – 地址 0x4004 201C）位描述 70
表 52. PMC 功耗模式的典型设置 70
表 53. CGU 时钟和电源控制 71
表 54. CGU0 基础时钟 73
表 55. 带可选择输入的时钟发生器的时钟源 73
表 56. 输出级的时钟源 74
表 57. CGU 引脚描述 75
表 58. 寄存器简介：CGU（基址 0x4005 0000） . . 76
表 59. FREQ_MON 寄存器（FREQ_MON，地址 0x4005 0014）位描述 79
表 60. XTAL_OSC_CTRL 寄存器（XTAL_OSC_CTRL，地址 0x4005 0018）位描述 80
表 61. PLL0USB 状态寄存器（PLL0USB_STAT，地址 0x4005 001C）位描述 80
表 62. PLL0USB 状态寄存器（PLL0USB_CTRL，地址 0x4005 0020C）位描述 80
表 63. PLL0USB M 分频器寄存器（PLL0USB_MDIV，地址 0x4005 0024）位描述 82
表 64. PLL0USB NP 分频器寄存器（PLL0USB_NP_DIV，地址 0x4005 0028）位描述 83
表 65. PLL0AUDIO 状态寄存器（PLL0AUDIO_STAT，地址 0x4005 002C）位描述 83
表 66. PLL0AUDIO 状态寄存器（PLL0AUDIO_CTRL，地址 0x4005 0030C）位描述 84
表 67. PLL0AUDIO M 分频器寄存器（PLL0AUDIO_MDIV，地址 0x4005 0034）位描述 85
表 68. PLL0 AUDIO NP 分频器寄存器（PLL0AUDIO_NP_DIV，地址 0x4005 0038）位描述 86
表 69. PLL0AUDIO 小数分频器寄存器

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

（PLL0AUDIO_FRAC，地址 0x4005 003C）位描述 86
表 70. PLL1 状态寄存器 （PLL1_STAT，地址 0x4005 0040）位描述 87
表 71. PLL1_CTRL 寄存器 （PLL1_CTRL，地址 0x4005 0044）位描述 87
表 72. IDIVA 状态寄存器 （IDIVA_CTRL，地址 0x4005 0048）位描述 89
表 73. IDIVB/C/D 控制寄存器 （IDIVB_CTRL，地址 0x4005 004C； IDIVC_CTRL，地址 0x4005 0050； IDIVC_CTRL，地址 0x4005 0054）位描述 90
表 74. IDIVE 状态寄存器 （IDIVE_CTRL，地址 0x4005 0058）位描述 91
表 75. BASE_SAFE_CLK 控制寄存器 （BASE_SAFE_CLK - 地址 0x4005 005C）位描述 92
表 76. BASE_USB0_CLK 控制寄存器 （BASE_USB0_CLK - 地址 0x4005 0060）位描述 92
表 77. BASE_PERIPH_CLK 控制寄存器 （BASE_PERIPH_CLK - 地址 0x4005 0064）位描述 93
表 78. BASE_USB1_CLK 控制寄存器 （BASE_USB1_CLK - 地址 0x4005 0068）位描述 94
表 79. BASE_M4_CLK 至 BASE_UART3_CLK 控制寄存器 （BASE_M4_CLK 至 BASE_UART3_CLK，地址 0x4005 006C 至 0x4005 00A8）位描述 . . 95
表 80. BASE_OUT_CLK 控制寄存器 （BASE_OUT_CLK，地址 0x4005 00AC）位描述 96
表 81. BASE_APLL_CLK 控制寄存器 （BASE_APLL_CLK - 地址 0x4005 00C0）位描述 97
表 82. BASE_CGU_OUT0_CLK 至 BASE_CGU_OUT1_CLK 控制寄存器 （BASE_CGU_OUT0_CLK 至 BASE_CGU_OUT1_CLK，地址 0x4005 00C4 至 0x4005 00C8）位描述 98
表 83. PLL 工作模式 100
表 84. HP0/1_Mode 寄存器中的 DIRECTL 和 DIRECTO 位设置 101
表 85. 480 MHz 输出时钟的 PLL0 （用于 USB）设置 108
表 86. 12 MHz 输入的 PLL0AUDIO 分频器设置 . . . 108
表 87. 绕过小数分频器的 12 MHz 的 PLL0AUDIO 分频器设置 110
表 88. CCU 时钟和电源控制 112
表 89. CCU1 分支时钟 113
表 90. CCU2 分支时钟 114
表 91. 寄存器简介：CCU1 （基址 0x4005 1000） . 115
表 92. 寄存器简介：CCU2 （基址 0x4005 2000） . 118
表 93. CCU1/2 功耗模式寄存器 （CCU1_PM，地址 0x4005 1000 和 CCU2_PM，地址 0x4005 2000）位描述 119
表 94. CCU1 基础时钟状态寄存器 （CCU1_BASE_STAT，地址 0x4005 1004）位描述 119
表 95. CCU2 基础时钟状态寄存器 （CCU2_BASE_STAT，地址 0x4005 2004）位描述 120
表 96. CCU1 分支时钟配置寄存器 （CLK_XXX_CFG，地址 0x4005 1100、 0x4005 1104、 ...、 0x4005 1A00）位描述 121
表 97. CCU1 分支时钟配置寄存器 （CLK_EMCDIV_CFG，地址 0x4005 1478）位描述 121
表 98. CCU2 分支时钟配置寄存器 （CLK_XXX_CFG，地址 0x4005 2100、 0x4005 2200、 ...、 0x4005 2800）位描述 122
表 99. CCU1 分支时钟状态寄存器 （CLK_XXX_STAT，地址 0x4005 1104、 0x4005 110C、 ...、 0x4005 1A04）位描述 123
表 100. CCU2 分支时钟状态寄存器 （CLK_XXX_STAT，地址 0x4005 2104、 0x4005 2204、 ...、 0x4005 2804）位描述 123
表 101. RGU 时钟和电源控制 124
表 102. 复位输出配置 125
表 103. 复位优先级 127
表 104. 寄存器简介：RGU （基址：0x4005 3000） . 128
表 105. 复位控制寄存器 0 （RESET_CTRL0，地址 0x4005 3100）位描述 130
表 106. 复位控制寄存器 1 （RESET_CTRL1，地址 0x4005 3104）位描述 131
表 107. 复位状态寄存器 0 （RESET_STATUS0，地址 0x4005 3110）位描述 133
表 108. 复位状态寄存器 1 （RESET_STATUS1，地址 0x4005 3114）位描述 134
表 109. 复位状态寄存器 2 （RESET_STATUS2，地址 0x4005 3118）位描述 136
表 110. 复位状态寄存器 3 （RESET_STATUS3，地址 0x4005 311C）位描述 138
表 111. 复位活动状态寄存器 0 （RESET_ACTIVE_STATUS0，地址 0x4005 3150）位描述 139
表 112. 复位活动状态寄存器 1 （RESET_ACTIVE_STATUS1，地址 0x4005 3154）位描述 141
表 113. 复位外部状态寄存器 0 （RESET_EXT_STAT0，地址 0x4005 3400）位描述 144
表 114. 复位外部状态寄存器 1 （RESET_EXT_STAT1，地址 0x4005 3404）位描述 144
表 115. 复位外部状态寄存器 2 （RESET_EXT_STAT2，地址 0x4005 3408）位描述 145
表 116. 复位外部状态寄存器 4 （RESET_EXT_STAT4，地址 0x4005 3410）位描述 145
表 117. 复位外部状态寄存器 5 （RESET_EXT_STAT5，地址 0x4005 3414）位描述 145
表 118. 复位外部状态寄存器 x （RESET_EXT_STATx，地址 0x4005 34xx）位描述 145
表 119. 复位外部状态寄存器 y （RESET_EXT_STATy，地址 0x4005 34yy）位描述 146
表 120. 引脚描述 148
表 121. SCU 时钟和电源控制 198
表 122. 寄存器简介：系统控制单元 (SCU)（基址 0x4008

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

6000） 201
表 123. 标准电平驱动引脚的引脚配置寄存器 （SFS，地址 0x4008 6000 (SPSP0_0) 至 0x4008 67AC (SFSPF_11)）位描述 208
表 124. 高驱动能力引脚的引脚配置寄存器 （SFS，地址 0x4008 60C4 (SFSP1_17) 至 0x4008 650C (SFSPA_3)）位描述 209
表 125. 高速引脚的引脚配置寄存器 （SFS，地址 0x4008 618C (SPSP3_3)； 0x4008 6C00 (SFSCLK0) 至 0x4008 6C0C (SFSCLK3)）位描述 210
表 126. 引脚 USB1_DP/USB1_DM 寄存器引脚配置 （SFSUSB，地址 0x4008 6C80）位描述 . . 211
表 127. 开漏 I²C 总线引脚寄存器的引脚配置 （SFSI2C0，地址 0x4008 6C84）位描述 . . 212
表 128. ENAIO0 寄存器控制的引脚 213
表 129. ADC0 功能选择寄存器 （ENAIO0，地址 0x4008 6C88）位描述 213
表 130. ENAIO1 寄存器控制的引脚 214
表 131. ADC1 功能选择寄存器 （ENAIO1，地址 0x4008 6C8C）位描述 215
表 132. ENAIO2 寄存器控制的引脚 216
表 133. 模拟功能选择寄存器 （ENAIO2，地址 0x4008 6C90）位描述 216
表 134. EMC 功能选择寄存器 （EMCDELAYCLK，地址 0x4008 6D00）位描述 217
表 135. 引脚中断选择寄存器 0 （PINTSEL0，地址 0x4008 6E00）位描述 217
表 136. 引脚中断选择寄存器 1 （PINTSEL1，地址 0x4008 6E04）位描述 219
表 137. GIMA 时钟和电源控制 221
表 138. GIMA 输出 222
表 139. 寄存器简介：GIMA （基址：0x400C 7000） 224
表 140. 定时器 0 CAP0_0 捕获输入多路复用器 （CAP0_0_IN，地址 0x400C 7000） 位描述 225
表 141. 定时器 0 CAP0_1 捕获输入多路复用器 （CAP0_1_IN，地址 0x400C 7004） 位描述 226
表 142. 定时器 0 CAP0_2 捕获输入多路复用器 （CAP0_2_IN，地址 0x400C 7008） 位描述 226
表 143. 定时器 0 CAP0_3 捕获输入多路复用器 （CAP0_3_IN，地址 0x400C 700C） 位描述 227
表 144. 定时器 1 CAP1_0 捕获输入多路复用器 （CAP1_0_IN，地址 0x400C 7010） 位描述 227
表 145. 定时器 1 CAP1_1 捕获输入多路复用器 （CAP1_1_IN，地址 0x400C 7014） 位描述 228
表 146. 定时器 1 CAP1_2 捕获输入多路复用器 （CAP1_2_IN，地址 0x400C 7018） 位描述 228
表 147. 定时器 1 CAP1_3 捕获输入多路复用器 （CAP1_3_IN，地址 0x400C 701C） 位描述 229
表 148. 定时器 2 CAP2_0 捕获输入多路复用器 （CAP2_0_IN，地址 0x400C 7020） 位描述 229
表 149. 定时器 2 CAP2_1 捕获输入多路复用器 （CAP2_1_IN，地址 0x400C 7024） 位描述 230
表 150. 定时器 2 CAP2_2 捕获输入多路复用器 （CAP2_2_IN，地址 0x400C 7028） 位描述 230
表 151. 定时器 2 CAP2_3 捕获输入多路复用器 （CAP2_3_IN，地址 0x400C 702C） 位描述 231
表 152. 定时器 3 CAP3_0 捕获输入多路复用器 （CAP3_0_IN，地址 0x400C 7030） 位描述 231
表 153. 定时器 3 CAP3_1 捕获输入多路复用器 （CAP3_1_IN，地址 0x400C 7034） 位描述 232
表 154. 定时器 3 CAP3_2 捕获输入多路复用器 （CAP3_2_IN，地址 0x400C 7038） 位描述 232
表 155. 定时器 3 CAP3_3 捕获输入多路复用器 （CAP3_3_IN，地址 0x400C 703C） 位描述 233
表 156. SCT CTIN_0 捕获输入多路复用器 （CTIN_0_IN，地址 0x400C 7040） 位描述 234
表 157. SCT CTIN_1 捕获输入多路复用器 （CTIN_1_IN，地址 0x400C 7044） 位描述 234
表 158. SCT CTIN_2 捕获输入多路复用器 （CTIN_2_IN，地址 0x400C 7048） 位描述 235
表 159. SCT CTIN_3 捕获输入多路复用器 （CTIN_3_IN，地址 0x400C 704C） 位描述 235
表 160. SCT CTIN_4 捕获输入多路复用器 （CTIN_4_IN，地址 0x400C 7050） 位描述 236
表 161. SCT CTIN_5 捕获输入多路复用器 （CTIN_5_IN，地址 0x400C 7054） 位描述 236
表 162. SCT CTIN_6 捕获输入多路复用器 （CTIN_6_IN，地址 0x400C 7058） 位描述 237
表 163. SCT CTIN_7 捕获输入多路复用器 （CTIN_7_IN，地址 0x400C 705C） 位描述 237
表 164. ADC 启动输入多路复用器 （VADC_TRIGGER_IN，地址 0x400C 7060）位描述 238
表 165. 事件路由器输入 13 多路复用器 （EVENTROUTER_13_IN，地址 0x400C 7064） 位描述 239
表 166. 事件路由器输入 14 多路复用器 （EVENTROUTER_14_IN，地址 0x400C 7068） 位描述 239
表 167. 事件路由器输入 16 多路复用器 （EVENTROUTER_16_IN，地址 0x400C 706C）

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

位描述 240
表 168. ADC 启动 0 输入多路复用器 （ADCSTART0_IN，地址 0x400C 7070）位描述 240
表 169. ADC 启动 1 输入多路复用器 （ADCSTART1_IN，地址 0x400C 7074）位描述 241
表 170. 不同引脚封装的 GPIO 引脚 242
表 171. GPIO 时钟和电源控制 242
表 172. 寄存器简介：GPIO 引脚中断 （基址：0x4008 7000） 245
表 173. 寄存器简介：GPIO 组 0 中断 （基址 0x4008 8000） 245
表 174. 寄存器简介：GPIO 组 1 中断 （基址 0x4008 9000） 246
表 175. 寄存器简介：GPIO 端口 （基址 0x400F 4000） 246
表 176. 引脚中断模式寄存器 （ISEL，地址 0x4008 7000）位描述 249
表 177. 引脚中断电平 （上升沿中断使能）寄存器 （IENR，地址 0x4008 7004）位描述 249
表 178. 引脚中断电平 （上升沿中断）设置寄存器 （SIENR，地址 0x4008 7008）位描述 249
表 179. 引脚中断电平 （上升沿中断）清除寄存器 （PCIENR，地址 0x4008 700C）位描述 . . . 250
表 180. 引脚中断有效电平 （下降沿中断使能）寄存器 （IENF，地址 0x4008 7010）位描述 250
表 181. 引脚中断有效电平 （下降沿中断）设置寄存器 （SIENF，地址 0x4008 7014）位描述 250
表 182. 引脚中断有效电平 （下降沿中断）清除寄存器 （CIENF，地址 0x4008 7018）位描述 251
表 183. 引脚中断上升沿寄存器 （RISE，地址 0x4008 701C）位描述 251
表 184. 引脚中断下降沿寄存器 （FALL，地址 0x4008 7020）位描述 251
表 185. 引脚中断状态寄存器 （IST，地址 0x4008 7024）位描述 252
表 186. GPIO 分组中断控制寄存器 （CTRL，地址 0x4008 8000 (GROUP0 INT) 和 0x4008 9000 (GROUP1 INT)）位描述 252
表 187. GPIO 分组中断端口极性寄存器 （PORT_POL，地址 0x4008 8020 (PORT_POL0) 至 0x4008 803C (PORT_POL7) （组 0 INT）和 0x4008 9020 (PORT_POL0) 至 0x4008 903C (PORT_POL7) （组 1 INT））位描述 253
表 188. GPIO 分组中断端口 n 使能寄存器 （PORT_ENA，地址 0x4008 8040 (PORT_ENA0) 至 0x4008 805C (PORT_ENA7) （组 0 INT）和 0x4008 9040 (PORT_ENA0) 至 0x4008 905C (PORT_ENA7) （组 1 INT））位描述 253
表 189. GPIO 端口字节引脚寄存器 （B，地址 0x400F 4000 (B0) 至 0x400F 00FC (B255)）位描述 253
表 190. GPIO 端口字引脚寄存器 （W，地址 0x400F 5000 (W0) 至 0x400F 13FC (W255)）位描述 254
表 191. GPIO 端口方向寄存器（DIR，地址 0x400F 6000 (DIR0) 至 0x400F 601C (DIR7)）位描述 . . . 254
表 192. GPIO 端口掩码寄存器 （MASK，地址 0x400F 6080 (MASK0) 至 0x400F 609C (MASK7)）位描述 254
表 193. GPIO 端口引脚寄存器（PIN，地址 0x400F 6100 (PIN0) 至 0x400F 611C (PIN7)）位描述 . . . 254
表 194. GPIO 屏蔽端口引脚寄存器 （MPIN，地址 0x400F 6180 (MPIN0) 至 0x400F 619C (MPIN7)）位描述 255
表 195. GPIO 端口设置寄存器 （SET，地址 0x400F 6200 (SET0) 至 0x400F 621C (SET7)）位描述 255
表 196. GPIO 端口清除寄存器 （CLR，地址 0x400F 6280 (CLR0) 至 0x400F 629C (CLR7)）位描述 255
表 197. GPIO 端口切换寄存器 （NOT，地址 0x400F 6300 (NOT0) 至 0x400F 632C (NOT7)）位描述 255
表 198. 边沿和电平触发引脚的引脚中断寄存器 257
表 199. SGPIO 时钟和电源控制 259
表 200. SGPIO 引脚描述 261
表 201. 寄存器简介：SGPIO （基址 0x4010 1000） 262
表 202. 引脚多路复用器配置寄存器 （OUT_MUX_CFG0 至 15，地址 0x4010 1000 至 0x4010 103C）位描述 264
表 203. 输出引脚多路复用 264
表 204. 输出使能控制 265
表 205. SGPIO 多路复用器配置寄存器 （SGPIO_MUX_CFG0 至 15，地址 0x4010 0040 至 0x4010 007C）位描述 266
表 206. SGPIO 多路复用器 268
表 207. 位串多路复用器配置寄存器 （SLICE_MUX_CFG0 至 15，地址 0x4010 1080 至 0x4010 10BC）位描述 269
表 208. 位串数据寄存器 （REG0 至 15，地址 0x4010 10C0 至 0x4010 10FC）位描述 269
表 209. 位串数据影子寄存器 （REG_SS0 至 15，地址 0x4010 110 至 0x4010 113C）位描述 270
表 210. 重新加载寄存器 （PRESET0 至 15，地址 0x4010 1140 至 0x4010 117C）位描述 270
表 211. 下行计数器寄存器 （COUNT0 至 15，地址 0x4010 1180 至 0x4010 11BC）位描述 . . . 270
表 212. 位置寄存器 （POS0 至 15，地址 0x4010 11C0 至 0x4010 11FC）位描述 270
表 213. 位串 A 掩码寄存器 （MASK_A，地址 0x4010 1200）位描述 271
表 214. 位串 H 掩码寄存器 （MASK_H，地址 0x4010 1204）位描述 271
表 215. 位串 I 掩码寄存器 （MASK_I，地址 0x4010 1208）位描述 271
表 216. 位串 P 掩码寄存器 （MASK_P，地址 0x4010 120C）位描述 271
表 217. GPIO 输入状态寄存器 （GPIO_INREG，地址 0x4010 1210）位描述 271
表 218. GPIO 输出控制寄存器 （GPIO_OUTREG，地址 0x4010 1214）位描述 272
表 219. GPIO 输出使能寄存器 （GPIO_OENREG，地址 0x4010 1218）位描述 272
表 220. 位串计数使能寄存器 （CTRL_ENABLED，地址 0x4010 121C）位描述 272
表 221. 位串计数禁用寄存器 （CTRL_DISABLED，地址

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

0x4010 1220）位描述 272
表 222. 移位时钟中断清除掩码寄存器 273
表 223. 移位时钟中断设置掩码寄存器 （SET_EN_0，地址 0x4010 1F04）位描述 273
表 224. 移位时钟中断使能寄存器 （ENABLE_0，地址 0x4010 1F08）位描述 273
表 225. 移位时钟中断状态寄存器 （STATUS_0，地址 0x4010 1F0C）位描述 273
表 226. 移位时钟中断清除状态寄存器 （CTR_STATUS_0，地址 0x4010 1F10）位描述 273
表 227. 移位时钟中断设置状态寄存器 （SET_STATUS_0，地址 0x4010 1F14）位描述 274
表 228. 捕获时钟中断清除掩码寄存器 （CLR_EN_1，地址 0x4010 1F20）位描述 274
表 229. 捕获时钟中断设置掩码寄存器 （SET_EN_1，地址 0x4010 1F24）位描述 274
表 230. 捕获时钟中断使能寄存器 （ENABLE_1，地址 0x4010 1F28）位描述 274
表 231. 捕获时钟中断状态寄存器 （STATUS_1，地址 0x4010 1F2C）位描述 274
表 232. 捕获时钟中断清除状态寄存器 （CTR_STATUS_1，地址 0x4010 1F30）位描述 274
表 233. 捕获时钟中断设置状态寄存器 （SET_STATUS_1，地址 0x4010 1F34）位描述 275
表 234. 模式匹配中断清除掩码寄存器 （CLR_EN_2，地址 0x4010 1F40）位描述 275
表 235. 模式匹配中断设置掩码寄存器 （SET_EN_2，地址 0x4010 1F44）位描述 275
表 236. 模式匹配中断使能寄存器 （ENABLE_2，地址 0x4010 1F48）位描述 275
表 237. 模式匹配中断状态寄存器 （STATUS_2，地址 0x4010 1F4C）位描述 275
表 238. 模式匹配中断清除状态寄存器 （CTR_STATUS_2，地址 0x4010 1F50）位描述 275
表 239. 模式匹配中断设置状态寄存器 （SET_STATUS_2，地址 0x4010 1F54）位描述 276
表 240. 输入位中断清除掩码寄存器 （CLR_EN_3，地址 0x4010 1F60）位描述 276
表 241. 输入中断设置掩码寄存器 （SET_EN_3，地址 0x4010 1F64）位描述 276
表 242. 输入中断使能寄存器（ENABLE_3，地址 0x4010 1F68）位描述 276
表 243. 输入中断状态寄存器 （STATUS_3，地址 0x4010 1F6C）位描述 276
表 244. 输入位中断清除状态寄存器 （CTR_STATUS_3，地址 0x4010 1F70）位描述 276
表 245. 移位时钟中断设置状态寄存器 （SET_STATUS_3，地址 0x4010 1F74）位描述 277
表 246. 位串 I/O 多路复用 282
表 247. LPC43xx 上的 SGPIO 应用 283
表 248. I2S 5.1 的 SGPIO 位串映射 284
表 249. I2S 5.1 的 SGPIO 设置， OUT_MUX_CFG 寄存器 285
表 250. I2S 5.1 的 SGPIO 设置， SGPIO_MUX_CFG 寄存器 285
表 251. I2S 5.1 的 SGPIO 设置， SLICE_MUX_CFG 寄存器 285
表 252. I2S 5.1 的 SGPIO 设置，位串 285
表 253. I2S 5.1 的 SGPIO 设置 （主机模式，引脚 8） 286
表 254. I2S 5.1 的 SGPIO 设置 （主机模式，引脚 9） 286
表 255. 相机接口的 SGPIO 位串映射 288
表 256. 相机接口的 SGPIO 设置 （OUT_MUX_CFG 寄存器） 288
表 257. 相机接口的 SGPIO 设置 （SGPIO_MUX_CFG 寄存器） 288
表 258. 相机接口的 SGPIO 设置 （SLICE_MUX_CFG 寄存器） 288
表 259. GPDMA 时钟和电源控制 290
表 260. 外围设备至 DMA 控制器的连接以及匹配流控制信号 291
表 261. 寄存器简介：GPDMA （基址 0x4000 2000） 294
表 262. DMA 中断状态寄存器 （INTSTAT，地址 0x4000 2000）位描述 296
表 263. DMA 中断终结计数请求状态寄存器 （INTTCCLEAR，地址 0x4000 2004）位描述 296
表 264. DMA 中断终结计数请求清除寄存器 （INTTCCLEAR，地址 0x4000 2008）位描述 296
表 265. DMA 中断错误状态寄存器 （INTERRSTAT，地址 0x4000 200C）位描述 297
表 266. DMA 中断错误清除寄存器 （INTERRCLR，地址 0x4000 2010）位描述 297
表 267. DMA 原始中断终结计数状态寄存器 （RAWINTTCSTAT，地址 0x4000 2014）位描述 297
表 268. DMA 原始错误中断状态寄存器 （RAWINTERRSTAT，地址 0x4000 2018）位描述 297
表 269. DMA 使能通道寄存器 （ENBLDCHNS，地址 0x4000 201C）位描述 298
表 270. DMA 软件连发请求寄存器 （SOFTBREQ，地址 0x4000 2020）位描述 298
表 271. DMA 软件单次请求寄存器 （SOFTSREQ，地址 0x4000 2024）位描述 298
表 272. DMA 软件最后一个连发请求寄存器 （SOFTBREQ，地址 0x4000 2028）位描述 299
表 273. DMA 软件最后一个单次请求寄存器 （SOFTLSREQ，地址 0x4000 202C）位描述 299
表 274. DMA 配置寄存器 （CONFIG，地址 0x4000 2030）位描述 299
表 275. DMA 同步寄存器 （SYNC，地址 0x4000 2034）位描述 300
表 276. DMA 通道源地址寄存器 （CSRCADDR，

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

0x4000 2100 (C0SRCADDR) 至 0x4000 21E0 (C7SRCADDR)）位描述 300
表 277. DMA 通道目标地址寄存器 （CDESTADDR，0x4000 2104 (C0DESTADDR) 至 0x4000 21E4 (C7DESTADDR)）位描述 301
表 278. DMA 通道链接列表项寄存器 （CLLI，0x4000 2108 (C0LLI) 至 0x4000 21E8 (C7LLI)）位描述 301
表 279. DMA 通道控制寄存器 （CCONTROL，0x4000 210C (C0CONTROL) 至 0x4000 21EC (C7CONTROL)）位描述 302
表 280. DMA 通道配置寄存器 （CCONFIG，0x4000 2110 (C0CONFIG) 至 0x4000 21F0 (C7CONFIG)）位描述 304
表 281. 流控制和传输类型位 306
表 282. 字节顺序特性 308
表 283. DMA 请求信号使用情形 313
表 284. SDIO 的时钟与电源控制 319
表 285. SDIO 引脚说明 320
表 286. 寄存器简介：SDMMC （基址：0x4000 4000） 321
表 287. 控制寄存器 （CTRL，地址 0x4000 4000）位描述 322
表 288. 电源使能寄存器 （PWREN，地址 0x4000 4004）位描述 324
表 289. 时钟分频器寄存器 （CLKDIV，地址 0x4000 4008）位描述 324
表 290. SD 时钟源寄存器 （CLKSRC，地址 0x4000 400C）位描述 324
表 291. 时钟使能寄存器 （CLKENA，地址 0x4000 4010）位描述 325
表 292. 超时寄存器 （TMOUT，地址 0x4000 4014）位描述 325
表 293. 卡类型寄存器 （CTYPE，地址 0x4000 4018）位描述 326
表 294. 块大小寄存器 （BLKSIZ，地址 0x4000 401C）位描述 326
表 295. 字节计数寄存器 （BYTCNT，地址 0x4000 4020）位描述 326
表 296. 中断掩码寄存器 （INTMASK，地址 0x4000 4024）位描述 326
表 297. 指令参数寄存器 （CMDARG，地址 0x4000 4028）位描述 327
表 298. 指令寄存器 （CMD，地址 0x4000 402C）位描述 328
表 299. 响应寄存器 0 （RESP0，地址 0x4000 4030）位描述 330
表 300. 响应寄存器 1 （RESP1，地址 0x4000 4034）位描述 330
表 301. 响应寄存器 2 （RESP2，地址 0x4000 4038）位描述 330
表 302. 响应寄存器 3 （RESP3，地址 0x4000 403C）位描述 330
表 303. 掩码中断状态寄存器 （MINTSTS，地址 0x4000 4040）位描述 331
表 304. 原始中断状态寄存器 （RINTSTS，地址 0x4000 4044）位描述 332
表 305. 状态寄存器 （STATUS，地址 0x4000 4048）位描述 333
表 306. FIFO 阈值水印寄存器 （FIFOTH，地址 0x4000 404C）位描述 334
表 307. 卡检测寄存器 （CDETECT，地址 0x4000 4050）位描述 335
表 308. 写保护寄存器 （WRTPRT，地址 0x4000 4054）位描述 336
表 309. CIU 卡已传输字节计数寄存器 （TCBCNT，地址 0x4000 405C）位描述 336
表 310. 主机至 BIU-FIFO 已传输字节计数寄存器 （TBBCNT，地址 0x4000 4060）位描述 . . 336
表 311. 去抖动计数寄存器 （DEBNCE，地址 0x4000 4064）位描述 336
表 312. UHS-1 寄存器 （UHS_REG，地址 0x4000 4074）位描述 337
表 313. 硬件复位 （RST_N，地址 0x4000 4078）位描述 337
表 314. 总线模式寄存器 （BMOD，地址 0x4000 4080）位描述 337
表 315. 查询需求寄存器 （PLDMND，地址 0x4000 4084）位描述 338
表 316. 描述符表基地址寄存器 （DBADDR，地址 0x4000 4088）位描述 338
表 317. 内部 DMAC 状态寄存器 （IDSTS，地址 0x4000 408C）位描述 339
表 318. 内部 DMAC 中断使能寄存器 (IDINTEN，地址 0x4000 4090）位描述 340
表 319. 当前主机描述符地址寄存器 （DSCADDR，地址 0x4000 4094）位描述 340
表 320. 当前缓冲描述符地址寄存器 （BUFADDR，地址 0x4000 4098）位描述 340
表 321. SEND_AUTO_STOP 位 341
表 322. CMD 寄存器的无数据指令设置 346
表 323. 单块或多块读取操作的 CMD 寄存器设置 . . 348
表 324. 单块或多块写入操作的 CMD 寄存器设置 . . 349
表 325. CMDARG 寄存器的参数 350
表 326. CMDARG 寄存器的参数 352
表 327. CMDARG 寄存器的参数 353
表 328. CMD 寄存器设置 353
表 329. BLKSIZ 寄存器 354
表 330. BYTCNT 寄存器 354
表 331. CMDARG 寄存器的参数 354
表 332. CMD 寄存器设置 355
表 333. BLKSIZ 寄存器 355
表 334. BYTCNT 寄存器 355
表 335. SD/MMC DMA DESC0 描述符 360
表 336. SD/MMC DMA DESC1 描述符 361
表 337. SD/MMC DMA DESC2 描述符 361
表 338. SD/MMC DMA DESC3 描述符 362
表 339. PBL 和水印级 365
表 340. 不同封装的 EMC 引脚 366
表 341. EMC 时钟和电源控制 367
表 342. 存储组别选择 369
表 343. EMC 引脚描述 370
表 344. 寄存器简介：外部存储控制器 （基址 0x4000 5000） 370
表 345. EMC 控制寄存器 （CONTROL – 地址 0x4000 5000）位描述 372

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 346. EMC 状态寄存器 （STATUS – 地址 0x4000 5004）位描述 373
表 347. EMC 配置寄存器 （CONFIG – 地址 0x4000 5008）位描述 374
表 348. 动态控制寄存器 （DYNAMICCONTROL – 地址 0x4000 5020）位描述 374
表 349. 动态存储器刷新定时器寄存器 （DYNAMICREFRESH – 地址 0x4000 5024）位描述 375
表 350. 动态存储器读配置寄存器 （DYNAMICREADCONFIG – 地址 0x4000 5028）位描述 376
表 351. 动态存储器预充电命令周期寄存器 （DYNAMICRP – 地址 0x4000 5030） 位描述 376
表 352. 动态存储器有效到预充电命令周期寄存器 （DYNAMICRAS – 地址 0x4000 5034） 位描述 377
表 353. 动态存储器自刷新退出时间寄存器 （DYNAMICSREX – 地址 0x4000 5038） 位描述 377
表 354. 动态存储器最后一个数据输出到有效时间寄存器 （DYNAMICAPR – 地址 0x4000 503C） 位描述 377
表 355. 动态存储器数据输入到有效命令时间寄存器 （DYNAMICDAL – 地址 0x4000 5040） 位描述 378
表 356. 动态存储器写入恢复时间寄存器 （DYNAMICWR - 地址 0x4000 5044）位描述 378
表 357. 动态存储器有效到有效命令周期寄存器 （DYNAMICRC – 地址 0x4000 5048） 位描述 378
表 358. 动态存储器自动刷新周期寄存器 （DYNAMICRFC – 地址 0x4000 504C） 位描述 379
表 359. 动态存储器退出自刷新寄存器 （DYNAMICXSR – 地址 0x4000 5050）位描述 379
表 360. 动态存储器有效组 A 至有效组 B 时间寄存器 （DYNAMICRRD - 地址 0x4000 5054） 位描述 379
表 361. 动态存储器加载模式寄存器到有效命令时间 （DYNAMICMRD – 地址 0x4000 5058） 位描述 380
表 362. 静态存储器延长等待寄存器 （STATICEXTENDEDWAIT – 地址 0x4000 5080）位描述 380
表 363. 动态存储器配置寄存器 （DYNAMICCONFIG，地址 0x4000 5100 (DYNAMICCONFIG0)、 0x4000 5120 (DYNAMICCONFIG1)、 0x4000 5140 (DYNAMICCONFIG2)、 0x4000 5160 (DYNAMICCONFIG3)） 位描述 381
表 364. 地址映射 381
表 365. 动态存储器 RASCAS 延迟寄存器 （DYNAMICRASCAS，地址 0x4000 5104 (DYNAMICRASCAS0)、 0x4000 5124 (DYNAMICRASCAS1)、 0x4000 5144 (DYNAMICRASCAS2)、 0x4000 5164 (DYNAMICRASCAS3)）位描述 383
表 366. 静态存储器配置寄存器 （STATICCONFIG，地址 0x4000 5200 (STATICCONFIG0)、 0x4000 5220 (STATICCONFIG1)、 0x4000 5240 (STATICCONFIG2)、 0x4000 5260 (STATICCONFIG3)）位描述 384
表 367. 静态存储器写使能延迟寄存器 （STATICWAITWEN，地址 0x4000 5204 (STATICWAITWEN0)、 0x4000 5224 (STATICWAITWEN1)、 0x4000 5244 (STATICWAITWEN2)、 0x4000 5264 (STATICWAITWEN3)）位描述 385
表 368. 静态存储器输出使能延迟寄存器 （STATICWAITOEN，地址 0x4000 5208 (STATICWAITOEN0)、 0x4000 5228 (STATICWAITOEN1)、 0x4000 5248 (STATICWAITOEN2)、 0x4000 5268 (STATICWAITOEN3)）位描述 385
表 369. 静态存储器读延迟寄存器 （STATICWAITRD，地址 0x4000 520C (STATICWAITRD0)、 0x4000 522C (STATICWAITRD1)、0x4000 524C (STATICWAITRD2)、 0x4000 526C (STATICWAITRD3)）位描述 386
表 370. 静态存储器页面模式读延迟寄存器 （STATICWAITPAGE，地址 0x4000 5210 (STATICWAITPAGE0)、 0x4000 5230 (STATICWAITPAGE1)、 0x4000 5250 (STATICWAITPAGE2)、 0x4000 5270 (STATICWAITPAGE3)）位描述 386
表 371. 静态存储器写延迟寄存器 （STATICWAITWR，地址 0x4000 5214 (STATICWAITWR0)、 0x4000 5234 (STATICWAITWR1)、0x4000 5254 (STATICWAITWR2)、 0x4000 5274 (STATICWAITWR3)）位描述 387
表 372. 静态存储器周转期延迟寄存器 （STATICWAITTURN，地址 0x4000 5218 (STATICWAITTURN0)、 0x4000 5238 (STATICWAITTURN1)、 0x4000 5258 (STATICWAITTURN2)、 0x4000 5278 (STATICWAITTURN3)）位描述 387
表 373. SPIFI 时钟和电源控制 394
表 374. SPIFI 闪存映射 395
表 375. SPIFI 引脚描述 395
表 376. 支持的 QSPI 设备 396
表 377. USB0 时钟和电源控制 397
表 378. USB 相关的首字母缩略词 398
表 379. 固定的端点配置 399
表 380. USB 数据包大小 399
表 381. USB0 引脚描述 400
表 382. 寄存器访问缩写 401
表 383. 寄存器简介：USB0 OTG 控制器 （寄存器基址 0x4000 6000） 401
表 384. CAPLENGTH 寄存器 （CAPLENGTH – 地址 0x4000 6100）位描述 403
表 385. HCSPARAMS 寄存器 （HCSPARAMS – 地址 0x4000 6104） 位描述 404
表 386. HCCPARAMS 寄存器 （HCCPARAMS - 地址 0x4000 6108）位描述 404

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 387. DCIVERSION 寄存器 （DCIVERSION - 地址 0x4000 6120）位描述 404
表 388. DCCPARAMS （地址 0x4000 6124） 405
表 389. 设备模式中的 USB 指令寄存器 （USBCMD_D - 地址 0x4000 6140）位描述 405
表 390. 主机模式中的 USB 指令寄存器 （USBCMD_H - 地址 0x4000 6140）位描述 – 主机模式 . . . 406
表 391. 帧列表大小值 408
表 392. 设备模式的 USB 状态寄存器（USBSTS_D - 地址 0x4000 6144）寄存器位描述 408
表 393. 主机模式的 USB 状态寄存器（USBSTS_H - 地址 0x4000 6144）寄存器位描述 409
表 394. 设备模式中的 USB 中断寄存器 （USBINTR_D - 地址 0x4000 6148）位描述 411
表 395. 主机模式中的 USB 中断寄存器 （USBINTR_H - 地址 0x4000 6148）位描述 412
表 396. 设备模式中的 USB 帧索引寄存器 （FRINDEX_D - 地址 0x4000 614C）位描述 413
表 397. 主机状态的 USB 帧索引寄存器 （FRINDEX_H - 地址 0x4000 614C）位描述 413
表 398. 用于帧列表索引的位数 414
表 399. 设备模式中的 USB 设备地址寄存器 （DEVICEADDR - 地址 0x4000 6154） 位描述 414
表 400. 主机模式中的 USB 周期性列表基址寄存器 （PERIODICLISTBASE - 地址 0x4000 6154）位描述 415
表 401. 设备模式中的 USB 端点列表地址寄存器 （ENDPOINTLISTADDR - 地址 0x4000 6158）位描述 415
表 402. 主机模式中的 USB 异步列表地址寄存器 （ASYNCLISTADDR - 地址 0x4000 6158） 位描述 415
表 403. 主机模式中的 USB TT 控制寄存器（TTCTRL - 地址 0x4000 615C）位描述 416
表 404. USB 连发大小寄存器 （BURSTSIZE – 地址 0x4000 6160）位描述 - 设备 / 主机模式 . . . 416
表 405. 主机模式中的 USB 传输缓冲区填充调谐寄存器 （TXFILLTUNING - 地址 0x4000 6164） 位描述 417
表 406. USB BINTERVAL 寄存器 （BINTERVAL - 地址 0x4000 6174）位描述 417
表 407. USB 端点 NAK 寄存器 （ENDPTNAK - 地址 0x4000 6178）位描述 418
表 408. USB 端点 NAK 使能寄存器 （ENDPTNAKEN - 地址 0x4000 617C）位描述 418
表 409. 设备模式中的端口状态和控制寄存器 （PORTSC1_D - 地址 0x4000 6184） 位描述 419
表 410. 主机模式中的端口状态和控制寄存器 （PORTSC1_H - 地址 0x4000 6184） 位描述 421
表 411. PORTSC1 寄存器中 PE 和 SUSP 位指定的端口状态 425
表 412. OTG 状态和控制寄存器 （OTGSC – 地址 0x4000 61A4）位描述 425
表 413. 设备模式中的 USB 模式寄存器 （USBMODE_D – 地址 0x4000 61A8）位描述 428
表 414. 主机模式中的 USB 模式寄存器 （USBMODE_H – 地址 0x4000 61A8）位描述 429
表 415. USB 端点设置状态寄存器 （ENDPTSETUPSTAT - 地址 0x4000 61AC）位描述 429
表 416. USB 端点启动寄存器 （ENDPTPRIME - 地址 0x4000 61B0）位描述 430
表 417. USB 端点刷新寄存器 （ENDPTFLUSH – 地址 0x4000 61B4）位描述 430
表 418. USB 端点状态寄存器 （ENDPTSTAT – 地址 0x4000 61B8）位描述 431
表 419. USB 端点完成寄存器（ENDPTCOMPLETE – 地址 0x4000 61BC）位描述 432
表 420. USB 端点 0 控制寄存器 （ENDPTCTRL0 - 地址 0x4000 61C0）位描述 432
表 421. USB 端点 1 到 5 控制寄存器 （ENDPTCTRL – 地址 0x4000 61C4 (ENDPTCTRL1) 到 0x4000 61D4 (ENDPTCTRL5)）位描述 . . . 433
表 422. 直接连接全速和低速设备的处理 438
表 423. 拆分状态机属性 439
表 424. 端点性能和特点 443
表 425. 当前 dTD 指针 444
表 426. 设置缓冲区 445
表 427. 下一个 dTD 指针 445
表 428. dTD 标记 446
表 429. dTD 缓冲页指针列表 447
表 430. 设备控制器端点初始化 451
表 431. 设备控制器失速响应矩阵 452
表 432. 可变长度传输协议示例 (ZLT = 0) 454
表 433. 可变长度传输协议示例 (ZLT = 1) 454
表 434. 中断 / 批量端点总线响应矩阵 454
表 435. 控制端点母线响应矩阵 456
表 436. 同步端点母线响应矩阵 458
表 437. 设备错误矩阵 462
表 438. 高频中断事件 463
表 439. 低频中断事件 463
表 440. 错误中断事件 463
表 441. USB1 时钟和电源控制 467
表 442. USB1 引脚描述 469
表 443. 寄存器访问缩写 470
表 444. 寄存器简介：USB1 主机 / 设备控制器 （寄存器基址 0x4000 7000） 470
表 445. CAPLENGTH 寄存器 （CAPLENGTH – 地址 0x4000 7100）位描述 472
表 446. HCSPARAMS 寄存器 （HCSPARAMS – 地址 0x4000 7104） 位描述 472
表 447. HCCPARAMS 寄存器 （HCCPARAMS - 地址 0x4000 7108）位描述 473
表 448. DCIVERSION 寄存器 （DCIVERSION - 地址 0x4000 7120）位描述 473
表 449. DCCPARAMS （地址 0x4000 7124） 473
表 450. 设备模式中的 USB 指令寄存器 （USBCMD_D - 地址 0x4000 7140）位描述 474
表 451. 主机模式中的 USB 指令寄存器 （USBCMD_H - 地址 0x4000 7140）位描述 475
表 452. 帧列表大小值 477
表 453. 设备模式的 USB 状态寄存器（USBSTS_D - 地址 0x4000 7144）寄存器位描述 477
表 454. 主机模式的 USB 状态寄存器（USBSTS_H - 地址

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

0x4000 7144）寄存器位描述 479
表 455. 设备模式中的 USB 中断寄存器 （USBINTR_D - 地址 0x4000 7148）位描述 481
表 456. 主机模式中的 USB 中断寄存器 （USBINTR_H - 地址 0x4000 7148）位描述 482
表 457. 设备模式中的 USB 帧索引寄存器 （FRINDEX_D - 地址 0x4000 714C）位描述 483
表 458. 主机模式中的 USB 帧索引寄存器 （FRINDEX_H - 地址 0x4000 714C）位描述 483
表 459. 用于帧列表索引的位数 483
表 460. 设备模式中的 USB 设备地址寄存器 （DEVICEADDR - 地址 0x4000 7154）位描述 484
表 461. 主机模式中的 USB 周期性列表基址寄存器 （PERIODICLISTBASE - 地址 0x4000 7154）位描述 484
表 462. 设备模式中的 USB 端点列表地址寄存器 （ENDPOINTLISTADDR - 地址 0x4000 7158）位描述 485
表 463. 主机模式中的 USB 异步列表地址寄存器 （ASYNCLISTADDR - 地址 0x4000 7158）位描述 485
表 464. 主机模式中的 USB TT 控制寄存器（TTCTRL - 地址 0x4000 715C）位描述 486
表 465. 设备 / 主机模式中的 USB 突发大小寄存器 （BURSTSIZE – 地址 0x4000 7160）位描述 486
表 466. 主机模式中的 USB 传输缓冲区填充调谐寄存器 （TXFILLTUNING - 地址 0x4000 7164）位描述 487
表 467. USB ULPI 视口寄存器 （ULPIVIEWPORT – 地址 0x4000 7170）位描述 488
表 468. 设备 / 主机模式中的 USB BINTERVAL 寄存器 （BINTERVAL – 地址 0x4000 7174）位描述 489
表 469. 设备模式中的 USB 端点 NAK 寄存器 （ENDPTNAK – 地址 0x4000 7178）位描述 489
表 470. 设备模式中的 USB 端点 NAK 使能寄存器 （ENDPTNAKEN – 地址 0x4000 717C）位描述 490
表 471. 设备模式中的端口状态和控制寄存器 （PORTSC1_D - 地址 0x4000 7184）位描述 491
表 472. 主机模式中的端口状态和控制寄存器 （PORTSC1_H - 地址 0x4000 7184）位描述 493
表 473. PORTSC1 寄存器中 PE 和 SUSP 位指定的端口状态 497
表 474. 设备模式中的 USB 模式寄存器 （USBMODE_D – 地址 0x4000 71A8）位描述 497
表 475. 主机模式中的 USB 模式寄存器 （USBMODE_H – 地址 0x4000 71A8）位描述 498
表 476. USB 端点设置状态寄存器 （ENDPTSETUPSTAT - 地址 0x4000 71AC）位描述 499
表 477. USB 端点启动寄存器 （ENDPTPRIME - 地址 0x4000 71B0）位描述 499
表 478. USB 端点刷新寄存器 （ENDPTFLUSH – 地址 0x4000 71B4）位描述 500
表 479. USB 端点状态寄存器 （ENDPTSTAT – 地址 0x4000 71B8）位描述 500
表 480. USB 端点完成寄存器（ENDPTCOMPLETE – 地址 0x4000 71BC）位描述 501
表 481. USB 端点 0 控制寄存器 （ENDPTCTRL0 - 地址 0x4000 71C0）位描述 501
表 482. USB 端点 1 到 3 控制寄存器 （ENDPTCTRL – 地址 0x4000 71C4 (ENDPTCTRL1) 到 0x4000 71CC (ENDPTCTRL3)）位描述 . . . 502
表 483. __WORD_BYTE 类结构 507
表 484. _BM_T 类结构 507
表 485. _CDC_ABSTRACT_CONTROL_MANAGEMENT_DESCRIPTOR 类结构 . . . 508
表 486. _CDC_CALL_MANAGEMENT_DESCRIPTOR 类结构 508
表 487. _CDC_HEADER_DESCRIPTOR 类结构 . . 508
表 488. _CDC_LINE_CODING 类结构 508
表 489. _CDC_UNION_1SLAVE_DESCRIPTOR 类结构 509
表 490. _CDC_UNION_DESCRIPTOR 类结构 509
表 491. _DFU_STATUS 类结构 509
表 492. _HID_DESCRIPTOR 类结构 509
表 493. _HID_DESCRIPTOR::_HID_DESCRIPTOR_LIST 类结构 510
表 494. _HID_REPORT_T 类结构 510
表 495. _MSC_CBW 类结构 510
表 496. _MSC_CSW 类结构 511
表 497. _REQUEST_TYPE 类结构 511
表 498. _USB_COMMON_DESCRIPTOR 类结构 . . . 511
表 499. _USB_CORE_DESCS_T 类结构 512
表 500. _USB_DEVICE_QUALIFIER_DESCRIPTOR 类结构 512
表 501. _USB_DFU_FUNC_DESCRIPTOR 类结构 513
表 502. _USB_INTERFACE_DESCRIPTOR 类结构 513
表 503. _USB_OTHER_SPEED_CONFIGURATION 类结构 514
表 504. _USB_SETUP_PACKET 类结构 514
表 505. _USB_STRING_DESCRIPTOR 类结构 . . . 515
表 506. _WB_T 类结构 515
表 507. USBD_API 类结构 515
表 508. USBD_API_INIT_PARAM 类结构 516
表 509. USBD_CDC_API 类结构 518
表 510. USBD_CDC_INIT_PARAM 类结构 519
表 511. USBD_CORE_API 类结构 523
表 512. USBD_DFU_API 类结构 526
表 513. USBD_DFU_INIT_PARAM 类结构 526
表 514. USBD_HID_API 类结构 529
表 515. USBD_HID_INIT_PARAM 类结构 529
表 516. USBD_HW_API 类结构 535
表 517. USBD_MSC_API 类结构 542
表 518. USBD_MSC_INIT_PARAM 类结构 542
表 519. 以太网时钟和电源控制 545
表 520. 以太网引脚描述 547
表 521. 寄存器简介：以太网 MAC 和 DMA （基址 0x4001 0000）. 548

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 522. MAC 配置寄存器 （MAC_CONFIG，地址 0x4001 0000）位描述 549
表 523. MAC 帧滤波寄存器 （MAC_FRAME_FILTER，地址 0x4001 0004）位描述 552
表 524. MAC Hash 表高位寄存器 （MAC_HASHTABLE_HIGH，地址 0x4001 0008）位描述 553
表 525. MAC Hash 表低位寄存器 （MAC_HASHTABLE_LOW，地址 0x4001 0008）位描述 553
表 526. MAC MII 地址寄存器 （MAC_MII_ADDR，地址 0x4001 0010）位描述 554
表 527. CSR 时钟范围值 555
表 528. MII 数据寄存器 （MAC_MII_DATA，地址 0x4001 0014）位描述 555
表 529. MAC 流控制寄存器 （MAC_FLOW_CTRL，地址 0x4001 0018）位描述 556
表 530. MAC VLAN 标签寄存器 （MAC_VLAN_TAG，地址 0x4001 01C）位描述 557
表 531. MAC 调试寄存器 （MAC_DEBUG，地址 0x4001 0024）位描述 557
表 532. MAC 远程唤醒帧滤波寄存器 （MAC_RWAKE_FRFLT，地址 0x4001 0028）位描述 558
表 533. MAC PMT 控制和状态寄存器 （MAC_PMT_CTRL_STAT，地址 0x4001 002C）位描述 559
表 534. MAC 中断状态寄存器 （MAC_INTR，地址 0x4001 0038）位描述 559
表 535. MAC 中断掩码寄存器 （MAC_INTR_MASK，地址 0x4001 003C）位描述 560
表 536. MAC 地址 0 高位寄存器 （MAC_ADDR0_HIGH，地址 0x4001 0040）位描述 560
表 537. MAC 地址 0 低位寄存器 （MAC_ADDR0_LOW，地址 0x4001 0044）位描述 560
表 538. MAC IEEE1588 时间戳控制寄存器 （MAC_TIMESTP_CTRL，地址 0x4001 0700）位描述 561
表 539. 时间戳快照对寄存器位的依存性 562
表 540. 亚秒递增寄存器 （SUBSECOND_INCR，地址 0x4001 0704）位描述 562
表 541. 系统时间秒寄存器 （SECONDS，地址 0x4001 0708）位描述 563
表 542. 系统时间纳秒寄存器 （NANOSECONDS，地址 0x4001 070C）位描述 563
表 543. 系统时间秒更新寄存器 （SECONDSUPDATE，地址 0x4001 0710）位描述 563
表 544. 系统时间纳秒更新寄存器 （NANOSECONDSUPDATE，地址 0x4001 0714）位描述 564
表 545. 时间戳加数寄存器 （ADDEND，地址 0x4001 0718）位描述 564
表 546. 目标时间秒寄存器 （TARGETSECONDS，地址 0x4001 071C）位描述 564
表 547. 目标时间纳秒寄存器 （TARGETNANOSECONDS，地址 0x4001 0720）位描述 565
表 548. 系统时间高字秒寄存器 （HIGHWORD，地址 0x4001 0724）位描述 565
表 549. 时间戳状态寄存器 （TIMESTAMPSTAT，地址 0x4001 0728）位描述 565
表 550. PPS 控制寄存器 （PPSCTRL，地址 0x4001 072C）位描述 566
表 551. 辅助时间戳纳秒寄存器 （AUXNANOSECONDS，地址 0x4001 0730）位描述 566
表 552. 辅助时间戳秒寄存器 （AUXSECONDS，地址 0x4001 0734）位描述 566
表 553. DMA 总线模式寄存器 （DMA_BUS_MODE，地址 0x4001 1000）位描述 567
表 554. 可编程连发长度设置 568
表 555. DMA 发送轮询请求寄存器 （DMA_TRANS_POLL_DEMAND，地址 0x4001 1004）位描述 569
表 556. DMA 接收轮询请求寄存器 （DMA_REC_POLL_DEMAND，地址 0x4001 1008）位描述 569
表 557. DMA 接收描述符列表地址寄存器 （DMA_REC_DES_ADDR，地址 0x4001 100C）位描述 569
表 558. DMA 发送描述符列表地址寄存器 （DMA_TRANS_DES_ADDR，地址 0x4001 1010）位描述 570
表 559. DMA 状态寄存器 （DMA_STAT，地址 0x4001 1014）位描述 570
表 560. DMA 操作模式寄存器 （DMA_OP_MODE，地址 0x4001 1018）位描述 572
表 561. DMA 中断使能寄存器 （DMA_INT_EN，地址 0x4001 101C）位描述 574
表 562. DMA 遗漏帧和缓冲区溢出计数寄存器 （DMA_MFRM_BUFOF，地址 0x4001 1020）位描述 576
表 563. DMA 接收中断看门狗定时器寄存器 （DMA_REC_INT_WDT，地址 0x4001 1024）位描述 576
表 564. DMA 当前主机发送描述符寄存器 （DMA_CURHOST_TRANS_DES，地址 0x4001 1048）位描述 577
表 565. DMA 当前主机接收描述符寄存器 （DMA_CURHOST_REC_DES，地址 0x4001 104C）位描述 577
表 566. DMA 当前主机发送缓冲区地址寄存器 （DMA_CURHOST_TRANS_BUF，地址 0x4001 1050）位描述 577
表 567. DMA 当前主机接收缓冲区地址寄存器 （DMA_CURHOST_TRANS_BUF，地址 0x4001 1054）位描述 577
表 568. 发送和接收 DMA 的优先级方案 580
表 569. 最小 PTP 时钟频率周期 586
表 570. 普通时钟：快照的 PTP 消息 588
表 571. 端对端透明时钟：快照的 PTP 消息 589
表 572. IEEE 1588-2008 中定义的消息格式 589
表 573. 控制和状态所需的 IPv4-UDP PTP 帧字段 . . 590
表 574. 控制和状态所需的 IPv6-UDP PTP 帧字段 . . 591
表 575. 控制和状态所需的以太网 PTP 帧字段 592
表 576. 发送描述符字 0 (TDES0) 608

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 577. 发送描述符字 1(TDES1) 611
表 578. 发送描述符字 2 (TDES2) 611
表 579. 发送描述符字 3 (TDES3) 611
表 580. 发送描述符字 6 (TDES6) 611
表 581. 发送描述符字 7 (TDES7) 611
表 582. 接收描述符字段 0 (RDES0) 613
表 583. 接收描述符字段 1 (RDES1) 615
表 584. 接收描述符字段 2 (RDES2) 615
表 585. 接收描述符字段 3 (RDES3) 615
表 586. 接收描述符字段 4 (RDES4) 616
表 587. 接收描述符字段 6 (RDES6) 617
表 588. 接收描述符字段 7 (RDES7) 617
表 589. LCD 时钟和电源控制 618
表 590. LCD 控制器引脚 621
表 591. 单面板 STN 显示器所使用的引脚 621
表 592. 双面板 STN 显示器所使用的引脚 622
表 593. TFT 显示器所使用的引脚 622
表 594. 寄存器简介：LCD 控制器（基址：0x4000 8000) 623
表 595. 水平定时寄存器 （TIMH，地址 0x4000 8000）位描述 624
表 596. 垂直定时寄存器 （TIMV，地址 0x4000 8004）位描述 625
表 597. 时钟和信号极性寄存器 （POL，地址 0x4000 8008）位描述 626
表 598. 行结束控制寄存器 （LE，地址 0x4000 800C）位描述 627
表 599. 上面板帧基址寄存器 （UPBASE，地址 0x4000 8010）位描述 628
表 600. 下面板帧基址寄存器 （LPBASE，地址 0x4000 8014）位描述 628
表 601. LCD 控制寄存器 （CTRL，地址 0x4000 8018）位描述 628
表 602. 中断掩码寄存器 （INTMSK，地址 0x4000 801C）位描述 630
表 603. 原始中断状态寄存器 （INTRAW，地址 0x4000 8020）位描述 631
表 604. 掩码中断状态寄存器 （INTSTAT，地址 0x4000 8024）位描述 631
表 605. 中断清除寄存器 （INTCLR，地址 0x4000 8028）位描述 632
表 606. 上面板当前地址寄存器 （UPCURR，地址 0x4000 802C）位描述 632
表 607. 下面板当前地址寄存器 （LPCURR，地址 0x4000 8030）位描述 632
表 608. 颜色调色板寄存器 （PAL，地址从 0x4000 8200 (PAL0) 至 0x4000 83FC (PAL255)）位描述 633
表 609. 游标图像寄存器 （CRSR_IMG，地址 0x4000 8800 (CRSR_IMG0) 到 0x4000 8BFC (CRSR_IMG1)）位描述 633
表 610. 游标控制寄存器 （CRSR_CTRL，地址 0x4000 8C00）位描述 634
表 611. 游标配置寄存器 （CRSR_CFG，地址 0x4000 8C04）位描述 634
表 612. 游标调色板寄存器 0 （CRSR_PAL0，地址 0x4000 8C08）位描述 635
表 613. 游标调色板寄存器 1 （CRSR_PAL1，地址 0x4000 8C0C）位描述 635
表 614. 游标 XY 位置寄存器 （CRSR_XY，地址 0x4000 8C10）位描述 636
表 615. 游标剪切位置寄存器 （CRSR_CLIP，地址 0x4000 8C14）位描述 636
表 616. 游标中断掩码寄存器 （CRSR_INTMSK，地址 0x4000 8C20）位描述 636
表 617. 游标中断清除寄存器 （CRSR_INTCLR，地址 0x4000 8C24）位描述 637
表 618. 游标原始中断状态寄存器 （CRSR_INTRAW，地址 0x4000 8C28）位描述 637
表 619. 游标掩码中断状态寄存器 （CRSR_INTSTAT，地址 0x4000 8C2C）位描述 637
表 620. 小端字节、小端像素顺序的 FIFO 位 641
表 621. 大端字节、大端像素顺序的 FIFO 位 642
表 622. 小端字节、大端像素顺序的 FIFO 位 643
表 623. RGB 模式数据格式 644
表 624. TFT 模式的调色板数据存储。 645
表 625. STN 彩色模式的调色板数据存储。 645
表 626. STN 单色模式的调色板数据存储。 646
表 627. STN 单色模式的调色板数据存储。 647
表 628. 32 x 32 游标的地址 649
表 629. 32 x 32 像素游标格式的像素映射缓冲 650
表 630. 64 x 64 像素游标格式的像素映射缓冲 650
表 631. 像素编码 651
表 632. 由 2 2/3 像素数据驱动的彩色显示 651
表 633. STN 单面板模式的 LCD 面板连接 658
表 634. STN 双面板模式的 LCD 面板连接 659
表 635. TFT 面板的 LCD 面板连接 660
表 636. SCT 时钟和电源控制 661
表 637. SCT 输入和输出 663
表 638. 寄存器简介：状态可配置定时器 （基址 0x4000 0000） 665
表 639. SCT 配置寄存器 （CONFIG - 地址 0x4000 0000）位描述 669
表 640. SCT 控制寄存器 （CTRL - 地址 0x4000 0004）位描述 670
表 641. SCT 限值寄存器 （LIMIT - 地址 0x4000 0008）位描述 671
表 642. SCT 终止条件寄存器 （HALT - 地址 0x4000 000C）位描述 671
表 643. SCT 停止条件寄存器 （STOP - 地址 0x4000 0010）位描述 672
表 644. SCT 启动条件寄存器 （START - 地址 0x4000 0014）位描述 672
表 645. SCT 计数器寄存器 （COUNT - 地址 0x4000 0040）位描述 672
表 646. SCT 状态寄存器 （STATE - 地址 0x4000 0044）位描述 673
表 647. SCT 输入寄存器 （INPUT - 地址 0x4000 0048）位描述 673
表 648. SCT 匹配 / 捕获寄存器模式寄存器 （REGMODE - 地址 0x4000 004C）位描述 674
表 649. SCT 输出寄存器 （OUTPUT - 地址 0x4000 0050）位描述 675
表 650. SCT 双向输出控制寄存器 （OUTPUTDIRCTRL - 地址 0x4000 0054）位描述 675
表 651. SCT 冲突解决寄存器 （RES - 地址 0x4000 0058）位描述 677

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 652. SCT DMA 0 请求寄存器 （DMAREQ0 - 地址 0x4000 005C）位描述 679
表 653. SCT DMA 1 请求寄存器 （DMAREQ1 - 地址 0x4000 0060）位描述 679
表 654. SCT 标志使能寄存器 （EVEN - 地址 0x4000 00F0）位描述 680
表 655. SCT 事件标志寄存器 （EVFLAG - 地址 0x4000 00F4）位描述 680
表 656. SCT 冲突使能寄存器 （CONEN - 地址 0x4000 00F8）位描述 680
表 657. SCT 冲突标志寄存器（CONFLAG - 地址 0x4000 00FC）位描述 681
表 658. SCT 匹配寄存器 0 至 15 （MATCH – 地址 0x4000 0100 (MATCH0) 至 0x4000 4013C (MATCH15)）位描述（REGMODEn 位 = 0） 681
表 659. SCT 捕获寄存器 0 至 15 （CAP - 地址 0x4000 0100 (CAP0) 至 0x4000 013C (CAP15)）位描述（REGMODEn 位 = 1） 682
表 660. SCT 匹配重新载入寄存器 0 至 15 （MATCHREL – 地址 0x4000 0200 (MATCHRELOAD0) 至 0x4000 023C (MATCHRELOAD15)）位描述（REGMODEn 位 = 0） 682
表 661. SCT 捕获控制寄存器 0 至 15 （CAPCTRL – 地址 0x4000 0200 (CAPCTRL0) 至 0x4000 023C (CAPCTRL15)）位描述（REGMODEn 位 = 1） 682
表 662. SCT 事件状态掩码寄存器 0 至 15 （EVSTATEMSK - 地址 0x4000 0300 (EVSTATEMSK0) 至 0x4000 0378 (EVSTATEMSK15)）位描述 683
表 663. SCT 事件控制寄存器 0 至 15 （EVCTRL - 地址 0x4000 0304 (EVCTRL0) 至 0x4000 037C (EVCTRL15)）位描述 683
表 664. SCT 输出设置寄存器 0 至 15 （OUTPUTSET – 地址 0x4000 0500 (OUTPUTSET0) 至 0x4000 0578 (OUTPUTSET15)）位描述 684
表 665. SCT 输出清除寄存器 0 至 15 （OUTPUTCL - 地址 0x4000 0504 (OUTPUTCL0) 至 0x4000 057C (OUTPUTCL15)）位描述 684
表 666. 事件条件 687
表 667. DMA 半字存取的替换地址映射 688
表 668. SCT 配置示例 692
表 669. 定时器 0/1/2/3 时钟和电源控制 694
表 670. 定时器 / 计数器功能描述 695
表 671. 定时器 0/1/2/3 输入和输出 696
表 672. 寄存器简介：定时器 0/1/2/3 （寄存器基址 0x4008 4000 (TIMER0)、 0x4008 5000 (TIMER1)、 0x400C 3000 (TIMER2) 和 0x400C 4000 (TIMER3)） 698
表 673. 定时器中断寄存器 IR （IR - 地址 0x4008 4000 (TIMER0)、 0x4008 5000 (TIMER1)、 0x400C 3000 (TIMER3) 和 0x400C 4000 (TIMER4)）位描述 699
表 674. 定时器控制寄存器 TCR （TCR - 地址 0x4008 4004 (TIMER0)、 0x4008 5004 (TIMER1)、 0x400C 3003 (TIMER2) 和 0x400C 4004 (TIMER3)）位描述 700
表 675. 定时器计数器寄存器 TC （TC - 地址 0x4008 4008 (TIMER0)、 0x4008 5008 (TIMER1)、 0x400C 3008 (TIMER2) 和 0x400C 4008 (TIMER3)）位描述 700
表 676. 定时器预分频寄存器 PR （PR - 地址 0x4008 400C (TIMER0)、 0x4008 500C (TIMER1)、 0x400C 300C (TIMER2) 和 0x400C 400C (TIMER3)）位描述 700
表 677. 定时器预分频计数器寄存器 PC （PC - 地址 0x4008 4010 (TIMER0)、 0x4008 5010 (TIMER1)、 0x400C 3010 (TIMER2) 和 0x400C 4010 (TIMER3)）位描述 701
表 678. 定时器匹配控制寄存器 MCR （MCR - 地址 0x4008 4014 (TIMER0), 0x4008 5014 (TIMER1), 0x400C 3014 (TIMER2), 0x400C 4014 (TIMER3)）位描述 701
表 679. 定时器匹配寄存器 MR0 至 3 （MR，地址 0x4008 4018 (MR0) 至 0x4008 4024 (M3) (TIMER0)、0x4008 5018 (MR0) 至 0x4008 5024 (MR3) (TIMER1)、 0x400C 3018 (MR0) 至 0x400C 8024 (MR3) (TIMER2)、 0x400C 4018 (MR0) 至 0x400C 4024 (MR3) (TIMER3)）位描述 702
表 680. 定时器捕获控制寄存器 （CCR - 地址 0x4008 4028 (TIMER0), 0x4008 5020 (TIMER1), 0x400C 3028 (TIMER2), 0x400C 4028 (TIMER3)）位描述 703
表 681. 定时器捕获寄存器 CR0 至 3 （CR，地址 0x4008 402C (CR0) 至 0x4008 4038 (CR3) (TIMER0)、0x4008 502C (CR0) 至 0x4008 5038 (CR3)(TIMER1)、 0x400C 302C (CR0) 至 0x400C 3038 (CR3) (TIMER2) 和 0x400C 402C (CR0) 至 0x400C 4038 (CR3) (TIMER3)）位描述 704
表 682. 定时器外部匹配寄存器 （EMR - 地址 0x4008 403C (TIMER0)、 0x4008 503C (TIMER1)、 0x400C 303C (TIMER2) 和 0x400C 403C (TIMER3)）位描述 704
表 683. 外部匹配控制 705
表 684. 定时器计数控制寄存器 CTCR （CTCR - 地址 0x4008 4070 (TIMER0), 0x4008 5070 (TIMER1), 0x400C 3070 (TIMER2), 0x400C 4070 (TIMER3)）位描述 706
表 685. PWM 时钟和电源控制 709
表 686. MOTOCON PWM 引脚描述 712
表 687. 寄存器简介：电动机控制脉冲宽度调制器 (MCPWM) （基址 0x400A 0000） 712
表 688. MCPWM 控制读取地址 (CON - 0x400A 0000) 位描述 713
表 689. MCPWM 控制设置地址 (CON_SET - 0x400A 0004) 位描述 715
表 690. MCPWM 控制清除地址 (CON_CLR - 0x400A 0008) 位描述 715
表 691. MCPWM 捕获控制读取地址 (CAPCON - 0x400A 000C) 位描述 716
表 692. MCPWM 捕获控制设置地址 (CAPCON_SET - 0x400A 0010) 位描述 717
表 693. MCPWM 捕获控制清除寄存器 （CAPCON_CLR

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

- 地址 0x400A 0014）位描述 718
表 694. MCPWM 定时器 / 计数器 0 至 2 寄存器 （TC - 0x400A 0018 (TC0)、 0x400A 001C (TC1)、 0x400A 0020) (TC2)）位描述 718
表 695. MCPWM 限值 0 至 2 寄存器 （LIM - 0x400A 0024 (LIM0)、 0x400A 0028 (LIM1) 和 0x400A 002C (LIM2)）位描述 719
表 696. MCPWM 匹配 0 至 2 寄存器 （MAT - 地址 0x400A 0030 (MAT0)、 0x400A 0034 (MAT1) 和 0x400A 0038 (MAT2)）位描述 719
表 697. MCPWM 死区寄存器 （DT - 地址 0x400A 003C）位描述 720
表 698. MCPWM 通信模式寄存器 （CP - 地址 0x400A 0040）位描述 721
表 699. MCPWM 捕获读取地址 （CAP - 0x400A 0044 (CAP0)、 0x400A 0048 (CAP1) 和 0x400A 004C 9CAP2)）位描述 721
表 700. 电机控制 PWM 中断 721
表 701. MCPWM 中断使能读取地址 (INTEN - 0x400A 0050) 位描述 722
表 702. MCPWM 中断使能设置寄存器 （INTEN_SET - 地址 0x400A 0054）位描述 723
表 703. PWM 中断使能清除寄存器 （INTEN_CLR - 地址 0x400A 0058）位描述 723
表 704. MCPWM 计数控制读取地址 (CNTCON - 0x400A 005C) 位描述 724
表 705. MCPWM 捕获控制设置地址 (CNTCON_SET - 0x400A 0060) 位描述 726
表 706. MCPWM 计数控制清除地址 (CNTCON_CLR - 0x400A 0064) 位描述 726
表 707. MCPWM 中断标志读取地址 (INTF - 0x400A 0068) 位描述 727
表 708. MCPWM 中断标志设置地址 (INTF_SET - 0x400A 006C) 位描述 728
表 709. MCPWM 中断标志清除地址 (INTF_CLR - 0x400A 0070) 位描述 729
表 710. MCPWM 捕获清除地址 (CAP_CLR - 0x400A 0074) 位描述 729
表 711. QEI 时钟和电源控制 736
表 712. QEI 引脚描述 738
表 713. 寄存器简介：QEI （基址 0x400C 6000） . . 738
表 714. QEI 控制寄存器 （CON - 地址 0x400C 6000）位描述 739
表 715. QEI 中断状态寄存器 （STAT - 地址 0x400C 6004）位描述 739
表 716. QEI 配置寄存器 （CONF - 地址 0x400C 6008）位描述 740
表 717. QEI 位置寄存器 （POS - 地址 0x400C 600C）位描述 740
表 718. QEI 最大位置寄存器 （MAXPOS - 地址 0x400C 6010）位描述 740
表 719. QEI 位置比较寄存器 0（CMPOS0 - 地址 0x400C 6014）位描述 740
表 720. QEI 位置比较寄存器 1（CMPOS1 - 地址 0x400C 6018）位描述 741
表 721. QEI 位置比较寄存器 2 （CMPOS2 - 地址 0x400C 601C）位描述 741
表 722. QEI 索引计数寄存器 （INXCNT - 地址 0x400C 6020）位描述 741
表 723. QEI 索引比较寄存器 0 （INXCMP0 - 地址 0x400C 6024）位描述 741
表 724. QEI 定时器载入寄存器 （LOAD - 地址 0x400C 6028）位描述 741
表 725. QEI 定时器寄存器 （TIME - 地址 0x400C 602C）位描述 741
表 726. QEI 速度寄存器 （VEL - 地址 0x400C 6030）位描述 742
表 727. QEI 速度捕获寄存器 （CAP - 地址 0x400C 6034）位描述 742
表 728. QEI 速度比较寄存器 （VELCOMP – 地址 0x400C 6038）位描述 742
表 729. 相位 A 输入寄存器上的 QEI 数字过滤器 （FILTERPHA - 0x400C 603C）位描述 . . . 742
表 730. 相位 B 输入寄存器上的 QEI 数字过滤器 （FILTERPHB - 0x400C 6040）位描述 . . . 742
表 731. 索引输入寄存器上的 QEI 数字过滤器 （FILTERINX - 0x400C 6044）位描述 742
表 732. QEI 索引接受窗口寄存器 (WINDOW - 0x400C 6048) 位描述 743
表 733. QEI 索引比较寄存器 1 （INXCMP1 - 地址 0x400C 604C）位描述 743
表 734. QEI 索引比较寄存器 2 （INXCMP2 - 地址 0x400C 6050）位描述 743
表 735. QEI 中断使能清除寄存器 （IEC – 地址 0x400C 6FD8）位描述 743
表 736. QEI 中断使能设置寄存器 （IES – 地址 0x400C 6FDC）位描述 744
表 737. QEI 中断状态寄存器 （INTSTAT - 地址 0x400C 6FE0）位描述 744
表 738. QEI 中断使能寄存器 （IE - 地址 0x400C 6FE4）位描述 745
表 739. QEI 中断状态清除寄存器 (CLR - 0x400C 6FE8) 位描述 746
表 740. QEI 中断状态设置寄存器 （SET - 地址 0x400C 6FEC）位描述 747
表 741. 编码器状态 748
表 742. 编码器状态跳变 [1] 748
表 743. 编码器方向 749
表 744. RIT 时钟和电源控制 751
表 745. 寄存器简介：重复性中断定时器 (RIT) （基址 0x400C 0000） 752
表 746. RI 比较值寄存器 （COMPVAL – 地址 0x400C 0000）位描述 752
表 747. RI 掩码寄存器 （MASK - 地址 0x400C 0004）位描述 753
表 748. RI 控制寄存器 （CTRL - 地址 0x400C 0008）位说明 753
表 749. RI 计数器寄存器 （COUNTER - 地址 0x400C 000C）位描述 753
表 750. 报警定时器时钟和电源控制 755
表 751. 寄存器简介：报警定时器 （基址 0x4004 0000） 756
表 752. 向下计数器寄存器 (DOWNCOUNTER - 0x4004 0000) 位描述 756
表 753. 预设值寄存器 (PRESET - 0x4004 0004) 位描述 756

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 754. 中断清除使能寄存器 (CLR_EN - 0x4004 0FD8) 位描述 756
表 755. 中断设置使能寄存器 (SET_EN - 0x4004 0FDC) 位描述 757
表 756. 中断状态寄存器 (STATUS - 0x4004 0FE0) 位描述 757
表 757. 中断使能寄存器 (ENABLE - 0x4004 0FE4) 位描述 757
表 758. 中断清除状态寄存器 (CLR_STAT - 0x4004 0FE8) 位描述 757
表 759. 中断设置状态寄存器 (SET_STAT - 0x4004 0FEC) 位描述 757
表 760. WWDT 时钟和电源控制 758
表 761. 寄存器简介：看门狗定时器 （基址 0x4008 0000） 760
表 762. 看门狗模式寄存器 (MOD - 0x4008 0000) 位描述 761
表 763. 看门狗工作模式选择 762
表 764. 看门狗定时器常量寄存器 (TC - 0x4008 0004) 位描述 762
表 765. 看门狗输入寄存器 (FEED - 0x4008 0008) 位描述 762
表 766. 看门狗定时器值寄存器 (TV - 0x4008 000C) 位描述 763
表 767. 看门狗定时器警告中断寄存器 (WARNINT - 0x4008 0014) 位描述 763
表 768. 看门狗定时器窗口寄存器 (WINDOW - 0x4008 0018) 位描述 763
表 769. RTC 时钟和电源控制 766
表 770. RTC 引脚描述 767
表 771. 寄存器简介：RTC （基址 0x4004 6000） . . 768
表 772. 寄存器简介：REGFILE （基址 0x4004 1000） 768
表 773. 中断位置寄存器 （ILR – 地址 0x4004 6000）位描述 769
表 774. 时钟控制寄存器 （CCR – 地址 0x4004 6008）位描述 769
表 775. 计数器递增中断寄存器 （CIIR – 地址 0x4004 600C）位描述 770
表 776. 报警掩码寄存器 （AMR – 地址 0x4004 6010）位描述 770
表 777. 完整时间寄存器 0 （CTIME0 – 地址 0x4004 6014）位描述 771
表 778. 完整时间寄存器 1 （CTIME1 – 地址 0x4004 6018）位描述 771
表 779. 完整时间寄存器 2 （CTIME2 – 地址 0x4004 601C）位描述 772
表 780. 时间计数器关系和值 772
表 781. 时间计数器寄存器 772
表 782. 秒寄存器 （SEC – 地址 0x4004 6020）位描述 773
表 783. 分钟寄存器 （MIN – 地址 0x4004 6024）位描述 773
表 784. 小时寄存器 （HRS – 地址 0x4004 6028）位描述 773
表 785. 月日期寄存器 （DOM – 地址 0x4004 602C）位描述 773
表 786. 周日期寄存器 （DOW – 地址 0x4004 6030）位描述 773
表 787. 年日期寄存器 （DOY – 地址 0x4004 6034）位描述 773
表 788. 月寄存器 （MONTH – 地址 0x4004 6038）位描述 774
表 789. 年寄存器 （YEAR – 地址 0x4004 603C）位描述 774
表 790. 校准寄存器 （CALIBRATION – 地址 0x4004 6040）位描述 774
表 791. 报警寄存器 775
表 792. 报警秒寄存器 （ASEC – 地址 0x4004 6060）位描述 775
表 793. 报警分钟寄存器 （AMIN – 地址 0x4004 6064）位描述 775
表 794. 报警小时寄存器 （AHRS – 地址 0x4004 6068）位描述 775
表 795. 报警月日期寄存器 （ADOM – 地址 0x4004 606C）位描述 776
表 796. 报警周日期寄存器 （ADOW – 地址 0x4004 6070）位描述 776
表 797. 报警年日期寄存器 （ADOY – 地址 0x4004 6074）位描述 776
表 798. 报警月寄存器 （AMON – 地址 0x4004 6078）位描述 776
表 799. 报警年寄存器 （AYRS – 地址 0x4004 607C）位描述 776
表 800. USART0/2/3 时钟和电源控制 778
表 801. USART0/2/3 引脚描述 781
表 802. 寄存器简介：USART0/2/3 （基址：0x4008 1000、 0x400C 1000、 0x400C 2000） . . . 782
表 803. USART 接收缓冲寄存器在 DLAB = 0、只读时 （RBR – 地址 0x4008 1000 (USART0)、0x400C 1000 (USART2)、0x400C 2000 (USART3)）位描述 783
表 804. USART 发送器保持寄存器在 DLAB = 0、只写时 （THR – 地址 0x4008 1000 (USART0)、0x400C 1000 (USART2)、0x400C 2000 (USART3)）位描述 783
表 805. USART 除数锁存器 LSB 寄存器在 DLAB = 1 时 （DLL – 地址 0x4008 1000 (USART0)、0x400C 1000 (USART2)、0x400C 2000 (USART3)）位描述 784
表 806. USART 除数锁存器 MSB 寄存器在 DLAB = 1 时 （DLM – 地址 0x4008 1004 (USART0)、0x400C 1004 (USART2)、0x400C 2004 (USART3)）位描述 784
表 807. USART 中断使能寄存器在 DLAB = 0 时 （IER – 地址 0x4008 1004 (USART0)、 0x400C 1004 (USART2)、 0x400C 2004 (USART3)）位描述 784
表 808. USART 中断识别寄存器在只读时 （IIR – 地址 0x4008 1008 (USART0)、 0x400C 1008 (USART2)、 0x400C 2008 (USART3)）位描述 785
表 809. USART 中断处理 786
表 810. USART FIFO 控制寄存器在只写时 （FCR – 地址 0x4008 1008 (USART0)、 0x400C 1008 (USART2)、 0x400C 2008 (USART3)）

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

位描述 787
表 811. USART 线路控制寄存器 （LCR – 地址 0x4008 100C (USART0)、 0x400C 100C (USART2)、 0x400C 200C (USART3)） 位描述 788
表 812. USART 线路状态寄存器在只读时 （LSR – 地址 0x4008 1014(USART0)、 0x400C 1014 (USART2)、 0x400C 2014 (USART3)） 位描述 789
表 813. USART 暂存寄存器（SCR – 地址 0x4008 101C (USART0)、 0x400C 101C (USART2)、 0x400C 201C (USART3)）位描述 790
表 814. 自动波特率控制寄存器 （ACR – 地址 0x4008 1020 (USART0)、 0x400C 1020 (USART2)、 0x400C 2020 (USART3)） 位描述 791
表 815. IrDA 控制寄存器 （ICR – 地址 0x4000 8024）位描述 794
表 816. IrDA 脉冲宽度 794
表 817. USART 小数分频器寄存器 （FDR – 地址 0x4008 1028 (USART0)、 0x400C 1028 (USART2)、 0x400C 2028 (USART3)） 位描述 795
表 818. 小数分频器设置查找表 797
表 819. USART 过采样寄存器 （OSR – 地址 0x4008 102C (USART0)、 0x400C 102C (USART2)、 0x400C 20402C (USART3)） 位描述 798
表 820. USART 半双工使能寄存器 （HDEN – 地址 0x4008 1040 (USART0), 0x400C 1040 (USART2), 0x400C 2040 (USART3)） 位描述 798
表 821. USART 智能卡接口控制寄存器 （SCICTRL – 地址 0x4008 1048(USART0), 0x400C 1048 (USART2), 0x400C 2048 (USART3)） 位描述 799
表 822. USART RS485 控制寄存器 （RS485CTRL – 地址 0x4008 104C (USART0), 0x400C 104C (USART2), 0x400C 204C (USART3)） 位描述 800
表 823. USART RS485 地址匹配寄存器 （RS485ADRMATCH – 地址 0x4008 1050 (USART0)、 0x400C 1050 (USART2)、 0x400C 2050 (USART3)）位描述 801
表 824. USART RS485 延迟值寄存器 （RS485DLY – 地址 0x4008 1054 (USART0)、 0x400C 1054 (USART2)、 0x400C 2054 (USART3)） 位描述 801
表 825. USART 同步模式控制寄存器 （SYNCCTRL – 地址 0x4008 1058 (USART0)、 0x400C 1058 (USART2)、 0x400C 2058 (USART3)） 位描述 801
表 826. USART 发送使能寄存器 （TER – 地址 0x4008 1030 (USART0)、 0x400C 1030 (USART2)、 0x400C 205C (USART3)） 位描述 803
表 827. UART1 时钟与电源控制 808
表 828. UART1 引脚描述 809
表 829. 寄存器简介：UART1 （基址 0x40082000） 810
表 830. UART1 接收缓冲寄存器 （当 DLAB = 0 (RBR - 地址 0x4008 2000）位描述 811
表 831. UART1 发送保持寄存器 （当 DLAB = 0 时） （THR - 地址 0x4008 2000）位描述 811
表 832. UART1 除数锁存器 LSB 寄存器 （当 DLAB = 1 时）（DLL - 地址 0x4008 2000）位描述 . . . 812
表 833. UART1 除数锁存器 MSB 寄存器 （当 DLAB = 1 时）（DLM - 地址 0x4008 2004）位描述 . . 812
表 834. UART1 中断使能寄存器 （当 DLAB = 0 时） （IER - 地址 0x4008 2004）位描述 812
表 835. UART1 中断识别寄存器 （IIR - 地址 0x4008 2008）位描述 813
表 836. UART1 中断处理 814
表 837. UART1 FIFO 控制寄存器 （FCR - 地址 0x4008 2008）位描述 815
表 838. UART1 线路控制寄存器 （LCR - 地址 0x4008 200C）位描述 816
表 839. UART1 调制解调器控制寄存器 （MCR - 地址 0x4008 2010）位描述 817
表 840. 调制解调器状态中断生成 818
表 841. UART1 线路状态寄存器 （LSR - 地址 0x4008 2014）位描述 819
表 842. UART1 调制解调器状态寄存器 （MSR - 地址 0x4008 2018）位描述 821
表 843. UART1 暂存寄存器 （SCR - 地址 0x4008 2014） 位描述 821
表 844. 自动波特率控制寄存器 （ACR - 地址 0x4008 2020）位描述 822
表 845. UART1 小数分频器寄存器 （FDR - 地址 0x4008 2028）位描述 825
表 846. 小数分频器设置查找表 827
表 847. UART1 发送使能寄存器 （TER - 地址 0x4008 2030）位描述 828
表 848. UART1 RS485 控制寄存器 （RS485CTRL - 地址 0x4008 204C）位描述 828
表 849. UART1 RS485 地址匹配寄存器 （RS485ADRMATCH – 地址 0x4008 2050）位描述 829
表 850. UART1 RS485 延迟值寄存器 （RS485DLY – 地址 0x4008 2054）位描述 829
表 851. UART1 FIFO 电平寄存器 （FIFOLVL – 地址 0x4008 2058）位描述 830
表 852. SSP0/1 时钟和电源控制 833
表 853. SSP 引脚描述 834
表 854. 寄存器简介：SSP0 （基址 0x4008 3000） 834
表 855. 寄存器简介：SSP1 （基址 0x400C 5000） 835
表 856. SSP 控制寄存器 0 （CR0 - 地址 0x4008 3000 (SSP0)、 0x400C 5000 (SSP1)）位描述 . . 836
表 857. SSP 控制寄存器 1 （CR1 - 地址 0x4008 3004 (SSP0)、 0x400C 5004 (SSP1)）位描述 . . 837
表 858. SSP 数据寄存器 （DR - 地址 0x4008 3008 (SSP0)、 0x400C 5008 (SSP1)）位描述 . . 837
表 859. SSP 状态寄存器 （SR - 地址 0x4008 300C (SSP0)、 0x400C 500C (SSP1)）位描述 . . 838
表 860. SSP 时钟预分频寄存器 （CPSR - 地址

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

0x4008 3010 (SSP0)、0x400C 5010 (SSP1)）位描述 838
表 861. SSP 中断掩码设置 / 清除寄存器 （IMSC - 地址 0x4008 3014 (SSP0)、0x400C 5014 (SSP1)）位描述 839
表 862. SSP 原始中断状态寄存器 （RIS - 地址 0x4008 3018 (SSP0)、 RIS - 0x400C 5018 (SSP1)）位描述 839
表 863. SSP 掩码中断状态寄存器 （MIS - 地址 0x4008 301C (SSP0)、 0x400C 501C (SSP1)）位描述 840
表 864. SSP 中断清除寄存器 （ICR - 地址 0x4008 3020 (SSP0)、 0x400C 5020 (SSP1)）位描述 . . 840
表 865. SSP DMA 控制寄存器 （DMACR - 地址 0x4008 3024 (SSP0)、0x400C 5024 (SSP1)）位描述 841
表 866. SPI 的时钟与电源控制 848
表 867. SPI 引脚描述 850
表 868. 寄存器简介：SPI （基址 0x4010 0000） . . . 850
表 869. SPI 控制寄存器 （CR - 地址 0x4010 0000）位描述 851
表 870. SPI 状态寄存器 （SR - 地址 0x4010 0004）位描述 852
表 871. SPI 数据寄存器 （DR - 地址 0x4010 0008）位描述 852
表 872. SPI 时钟计数器寄存器 （CCR - 地址 0x4010 0010）位描述 852
表 873. SPI 测试控制寄存器 （TCR - 地址 0x4010 0010）位描述 853
表 874. SPI 测试状态寄存器 （TSR - 地址 0x4010 0014）位描述 853
表 875. SPI 中断寄存器 （INT - 地址 0x4010 001C）位描述 853
表 876. SPI 数据和时钟相位的关系 854
表 877. I2S 的时钟与电源控制 857
表 878. 引脚描述 860
表 879. 寄存器简介：I2S0 （基址 0x400A 2000） . . 861
表 880. 寄存器简介：I2S1 （基址 0x400A 3000） . . 861
表 881. I2S 数字音频输出寄存器 （DAO - 地址 0x400A 2000 (I2S0) 及 0x400A 3000 (I2S1)）位描述 862
表 882. I2S 数字音频输入寄存器 （DAI - 地址 0x400A 2004 (I2S0) 及 0x400A 3004 (I2S1)）位描述 862
表 883. 发送 FIFO 寄存器 （TXFIFO - 地址 0x400A 2008 (I2S0) 及 0x400A 3008 (I2S1)）位描述 863
表 884. I2S 接收 FIFO 寄存器 （RXFIFO - 地址 0x400A 200C (I2S0) 及 0x400A 300C (I2S1)）位描述 863
表 885. I2S 状态反馈寄存器 （STATE - 地址 0x400A 2010 (I2S0) 及 0x400A 3010 (I2S1)）位描述 863
表 886. I2S DMA 配置寄存器 1 （DMA1 - 地址 0x400A 2014 (I2S0) 及 0x400A 3014 (I2S1)）位描述 864
表 887. I2S DMA 配置寄存器 2 （DMA2 - 地址 0x400A 2018 (I2S0) 及 0x400A 3018 (I2S1)）位描述 864
表 888. I2S 中断请求控制寄存器 （IRQ - 地址 0x400A 201C (I2S0) 及 0x400A 301C (I2S1)）位描述 865
表 889. I2S 发送时钟速率寄存器 （TXRATE - 地址 0x400A 2020 (I2S0) 及 0x400A 3020 (I2S1)）位描述 865
表 890. I2S 接收时钟速率寄存器 （RXRATE - 地址 0x400A 2024 (I2S0) 及 0x400A 3024 (I2S1)）位描述 866
表 891. I2S 发送时钟速率寄存器 （TXBITRATE - 地址 0x400A 2028 (I2S0) 及 0x400A 3028 (I2S1)）位描述 866
表 892. I2S 接收时钟速率寄存器 （RXBITRATE - 地址 0x400A 202C (I2S0) 及 0x400A 302C (I2S1)）位描述 866
表 893. I2S 发送模式控制寄存器 （TXMODE - 地址 0x400A 2030 (I2S0) 及 0x400A 3030 (I2S1)）位描述 867
表 894. I2S 接收模式控制寄存器 （RXMODE - 地址 0x400A 2034 (I2S0) 及 0x400A 3034 (I2S1)）位描述 867
表 895. I2S 发送模式 869
表 896. I2S 接收模式 872
表 897. FIFO 级别比较的条件 874
表 898. DMA 和中断请求产生 875
表 899. STATE 寄存器中的状态反馈 875
表 900. C_CAN 的时钟与电源控制 876
表 901. C_CAN 引脚描述 877
表 902. 寄存器简介：C_CAN0 （基址 0x400E 2000） 878
表 903. 寄存器简介：C_CAN1 （基址 0x400A 4000） 879
表 904. CAN 控制寄存器 （CNTL，地址 0x400E 2000 (C_CAN0) 和 0x400A 4000 (C_CAN1)）位描述 881
表 905. CAN 状态寄存器 （STAT，地址 0x400E 2004 (C_CAN0) 和 0x400A 4004 (C_CAN1)）位描述 882
表 906. CAN 错误计数器 （EC，地址 0x400E 2008 (C_CAN0) 和 0x400A 4008 (C_CAN1)）位描述 883
表 907. CAN 位计时寄存器 （BT，地址 0x400E 200C (C_CAN0) 和 0x400A 400C (C_CAN1)）位描述 883
表 908. CAN 中断寄存器 （INT，地址 0x400E 2010 (C_CAN0) 和 0x400A 4010 (C_CAN1)）位描述 884
表 909. CAN 检测寄存器 （TEST，地址 0x400E 2014 (C_CAN0) 和 0x400A 4014 (C_CAN1)）位描述 884
表 910. CAN 波特率预分频器扩展寄存器 （BRPE，地址 0x400E 2018 (C_CAN0) 和 0x400A 4018 (C_CAN1)）位描述 885
表 911. 消息接口寄存器 887
表 912. 消息 RAM 中消息对象的结构 887
表 913. CAN 信息接口指令请求寄存器 （IF1_CMDREQ，地址 0x400E 2020 (C_CAN0) 和 0x400A 4020 (C_CAN1)）位描述 888

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

表 914. CAN 信息接口指令请求寄存器（IF2_CMDREQ，地址 0x400E 2080 (C_CAN0) 和 0x400A 4080 (C_CAN1)）位描述 888
表 915. CAN 信息接口指令掩码寄存器写入方向（IF1_CMDMSK_W，地址 0x400E 2024 (C_CAN0) 和 0x400A 4024 (C_CAN1)）位描述 889
表 916. CAN 信息接口指令掩码寄存器写入方向（IF2_CMDMSK_W，地址 0x400E 2084 (C_CAN0) 和 0x400A 4084 (C_CAN1)）位描述 890
表 917. CAN 信息接口指令掩码寄存器读取方向（IF1_CMDMSK_R，地址 0x400E 2024 (C_CAN0) 和 0x400A 4024 (C_CAN1)）位描述 891
表 918. CAN 信息接口指令掩码寄存器读取方向（IF2_CMDMSK_R，地址 0x400E 2084 (C_CAN0) 和 0x400A 4084 (C_CAN1)）位描述 892
表 919. CAN 信息接口指令掩码 1 寄存器（IF1_MSK1，地址 0x400E 2028 (C_CAN0) 和 0x400A 4028 (C_CAN1)）位描述 893
表 920. CAN 信息接口指令掩码 1 寄存器（IF2_MSK1，地址 0x400E 2088 (C_CAN0) 和 0x400A 4088 (C_CAN1)）位描述 893
表 921. CAN 信息接口指令掩码 2 寄存器（IF1_MSK2，地址 0x400E 202C (C_CAN0) 和 0x400A 402C (C_CAN1)）位描述 893
表 922. CAN 信息接口指令掩码 2 寄存器（IF2_MSK2，0x400E 208C (C_CAN0) 和 0x400A 408C (C_CAN1)）位描述 894
表 923. CAN 信息接口指令仲裁 1 寄存器（IF1_ARB1，地址 0x400E 2030 (C_CAN0) 和 0x400A 4030 (C_CAN1)）位描述 894
表 924. CAN 信息接口指令仲裁 1 寄存器（IF2_ARB1，地址 0x400E 2090 (C_CAN0) 和 0x400A 4090 (C_CAN1)）位描述 894
表 925. CAN 信息接口指令仲裁 2 寄存器（IF1_ARB2，地址 0x400E 2034 (C_CAN0) 和 0x400A 4034 (C_CAN1)）位描述 895
表 926. AN 信息接口指令仲裁 2 寄存器（IF2_ARB2，地址 0x400E 2094 (C_CAN0) 和 0x400A 4094 (C_CAN1)）位描述 895
表 927. CAN 信息接口信息控制寄存器（IF1_MCTRL，地址 0x400E 2038 (C_CAN0) 和 0x400A 4038 (C_CAN1)）位描述 896
表 928. CAN 信息接口信息控制寄存器（IF2_MCTRL，地址 0x400E 2098 (C_CAN0) 和 0x400A 4098 (C_CAN1)）位描述 897
表 929. CAN 信息接口数据 A1 寄存器（IF1_DA1，地址 0x400E 203C (C_CAN0) 和 0x400A 403C (C_CAN1)）位描述 898
表 930. CAN 信息接口数据 A1 寄存器（IF2_DA1，地址 0x400E 209C (C_CAN0) 和 0x400A 409C (C_CAN1)）位描述 899
表 931. CAN 信息接口数据 A2 寄存器（IF1_DA2，地址 0x400E 2040 (C_CAN0) 和 0x400A 4040 (C_CAN1)）位描述 899
表 932. CAN 信息接口数据 A2 寄存器（IF2_DA2，地址 0x400E 20A0 (C_CAN0) 和 0x400A 40A0 (C_CAN1)）位描述 899
表 933. CAN 信息接口数据 B1 寄存器（IF1_DB1，地址 0x400E 2044 (C_CAN0) 和 0x400A 4044 (C_CAN1)）位描述 899
表 934. CAN 信息接口数据 B1 寄存器（IF2_DB1，地址 0x400E 20A4 (C_CAN0) 和 0x400A 40A4 (C_CAN1)）位描述 899
表 935. CAN 信息接口数据 B2 寄存器（IF1_DB2，地址 0x400E 2048 (C_CAN0) 和 0x400A 4048 (C_CAN1)）位描述 899
表 936. CAN 信息接口数据 B2 寄存器（IF2_DB2，地址 0x400E 20A8 (C_CAN0) 和 0x400A 40A8 (C_CAN1)）位描述 900
表 937. CAN 发送请求 1 寄存器（TXREQ1，地址 0x400E 2100 (C_CAN0) 和 0x400A 4100 (C_CAN1)）位描述 900
表 938. CAN 发送请求 2 寄存器（TXREQ2，地址 0x400E 2104 (C_CAN0) 和 0x400A 4104 (C_CAN1)）位描述 900
表 939. CAN 新数据 1 寄存器（ND1，地址 0x400E 2120 (C_CAN0) 和 0x400A 4120 (C_CAN1)）位描述 901
表 940. CAN 新数据 2 寄存器（ND2，地址 0x400E 2124 (C_CAN0) 和 0x400A 4124 (C_CAN1)）位描述 901
表 941. CAN 中断挂起 1 寄存器（IR1，地址 0x400E 2140 (C_CAN0) 和 0x400A 4140 (C_CAN1)）位描述 901
表 942. CAN 中断挂起 2 寄存器（IR2，地址 0x400E 2144 (C_CAN0) 和 0x400A 4144 (C_CAN1)）位描述 901
表 943. CAN 信息有效 1 寄存器（MSGV1，地址 0x400E 2160 (C_CAN0) 和 0x400A 4160 (C_CAN1)）位描述 902
表 944. CAN 信息有效 2 寄存器（MSGV2，地址 0x400E 2164 (C_CAN0) 和 0x400A 4164 (C_CAN1)）位描述 902
表 945. CAN 时钟分频器寄存器（CLKDIV，地址 0x400E 2180 (C_CAN0) 和 0x400A 4180 (C_CAN1)）位描述 902
表 946. 初始化发送对象 910
表 947. 初始化接收对象 910
表 948. C_CAN 位时间参数 914
表 949. I2C0/1 时钟及电源控制 915
表 950. I^2C 总线引脚描述 917
表 951. 寄存器简介：I^2C0（基址 0x400A 1000）. . 917
表 952. 寄存器简介：I^2C1（基址 0x400E 0000）. . 918
表 953. I^2C 控制设置寄存器（CONSET – 地址 0x400A 1000 (I2C0) 和 0x400E 0000 (I2C1)）位描述 919
表 954. I^2C 状态寄存器（STAT – 地址 0x400A 1004 (I2C0) 和 0x400E 0004 (I2C1)）位描述 . . . 920
表 955. I^2C 数据寄存器（DAT – 0x400A 1008 (I2C0) 和 0x400E 0008 (I2C1)）位描述 921
表 956. I^2C 从机地址寄存器 0（ADR0 – 地址 0x400A 100C (I2C0) 和 0x400E 000C (I2C1)）位

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

描述 921
表 957. I²C SCL 高电平占空比寄存器 （SCLH – 地址 0x400A 1010 (I2C0) 和 0x400E 0010 (I2C1)）位描述 921
表 958. I²C SCL 低电平占空比寄存器 （SCLL – 地址 0x400A 1014 (I2C0) 和 0x400E 0014 (I2C1)）位描述 921
表 959. 用于所选 I²C 时钟值的 SCLL + SCLH 值 . . . 922
表 960. I²C 控制清除寄存器 （CONCLR – 地址 0x400A 1018 和 0x400E 0018 (I2C1)）位描述 922
表 961. I²C 监控模式控制寄存器 （MMCTRL – 地址 0x400A 101C (I2C0) 和 0x400E 001C (I2C1)）位描述 923
表 962. I²C 从机地址寄存器（ADR – 地址 0x400A 1020 (ADR1) 到 0x400A 1028 (ADR3) (I2C0) 和 0x400E 0020 (ADR1) 到 0x400E 0028 (ADR3) (I2C1)）位描述 924
表 963. I²C 数据缓冲寄存器 （DATA_BUFFER – 地址 0x400A 102C (I2C0) 和 0x400E 002C (I2C1)）位描述 925
表 964. I²C 掩码寄存器 （MASK –地址 0x400A 1030 (MASK0) 到 0x400A 103C (MASK3) (I2C0) 和 0x400E 0030 (MASK0) 到 0x400E 003C (MASK3) (I2C1)）位描述 925
表 965. 用于配置主机模式的 CONSET 925
表 966. 用于配置从机模式的 CONSET 927
表 967. 用于描述 I²C 操作的缩写 933
表 968. 用于初始化主发送模式的 CONSET 933
表 969. 主发送器模式 934
表 970. 主接收器模式 936
表 971. 从机接收器模式下的 ADR 使用 938
表 972. 用于初始化从接收模式的 CONSET 938
表 973. 从机接收器模式 938
表 974. 从属发送器模式 941
表 975. 其它状态 943
表 976. 不同封装的 ADC 通道 953
表 977. ADC0/1 时钟和电源控制 954
表 978. ADC 引脚说明 955
表 979. 寄存器简介：ADC0 （基址 0x400E 3000） 955
表 980. 寄存器简介：ADC1 （基址 0x400E 4000） 956
表 981. A/D 控制寄存器 （CR - 地址 0x400E 3000 (ADC0) 和 0x400E 4000 (ADC1)）位描述 957
表 982. A/D 全局数据寄存器 （CR - 地址 0x400E 3004 (ADC0) 和 0x400E 4004 (ADC1)）位描述 958
表 983. A/D 中断使能寄存器 （INTEN - 地址 0x400E 300C (ADC0) 和 0x400E 400C (ADC1)）位描述 958
表 984. A/D 数据寄存器 （DR – 地址 0x400E 3010 (DR0) 到 0x400E 302C (DR7) (ADC0)；0x400E 4010 (DR0) 到 0x400E 402C (DR7) (ADC1)）位描述 959
表 985. A/D 状态寄存器 （STAT - 地址 0x400E 3030 (ADC0)， 0x400E 4030 (ADC1)）位描述 . . 959
表 986. DAC 时钟和电源控制 961
表 987. DAC 引脚描述 961
表 988. 寄存器简介：DAC （基址 0x400E 1000） . 962
表 989. D/A 转换器寄存器 （CR- 地址 0x400E 1000）位说明 962
表 990. D/A 控制寄存器 （CTRL - 地址 0x400E 1004）位描述 962
表 991. D/A 转换器计数器值寄存器 （CNTVAL - 地址 0x400E 1008）位描述 963
表 992. 闪存配置 969
表 993. 代码读保护选项 970
表 994. 代码读保护硬件 / 软件的相互作用 971
表 995. ISP 命令总结 972
表 996. ISP 解锁命令 972
表 997. ISP 设置波特率命令 973
表 998. 可能的 ISP 波特率和 CCLK 频率之间的相关性（单位 MHz） 973
表 999. ISP 回应命令 973
表 1000. ISP 写 RAM 命令 974
表 1001. ISP 读存储器命令 974
表 1002. ISP 准备写操作命令的扇区 975
表 1003. ISP 复制命令 975
表 1004. ISP 运行命令 976
表 1005. ISP 擦除扇区命令 976
表 1006. ISP 扇区查空命令 977
表 1007. ISP 读器件标识命令 977
表 1008. LPC43xx 器件标识号 977
表 1009. ISP 读 Boot 代码版本号命令 977
表 1010. ISP 读取设备序列号命令 977
表 1011. ISP 比较命令 978
表 1012. ISP 返回代码总览 978
表 1013. IAP 命令总览 981
表 1014. IAP 准备写操作命令的扇区 982
表 1015. IAP 将 RAM 内容复制到闪存命令 982
表 1016. IAP 擦除扇区命令 983
表 1017. IAP 扇区查空命令 983
表 1018. IAP 读取器件识别号命令 983
表 1019. IAP 读 Boot 代码版本号命令 984
表 1020. IAP 读取设备序列号命令 984
表 1021. IAP 比较命令 984
表 1022. 重新调用 ISP 985
表 1023. IAP 状态代码小结 985
表 1024. 寄存器简介：FMC （基址 0x4008 4000） 986
表 1025. 闪存模块签名起始寄存器 (FMSSTART - 0x4008 4020) 位说明 986
表 1026. 闪存模块签名停止寄存器 (FMSSTOP - 0x4008 4024) 位描述 986
表 1027. FMSW0 寄存器位描述 （FMSW0，地址：0x4008 402C) 987
表 1028. FMSW1 寄存器位描述 （FMSW1，地址：0x4008 4030) 987
表 1029. FMSW2 寄存器位描述 （FMSW2，地址：0x4008 4034) 987
表 1030. FMSW3 寄存器位描述 （FMSW3，地址：0x4008 4038) 987
表 1031. 闪存模块状态寄存器 (FMSTAT - 0x4008 4FE0) 位描述 987

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

表 1032. 闪存模块状态清除寄存器 (FMSTATCLR - 0x0x4008 4FE8) 位描述 987
表 1033. JTAG 引脚描述 990
表 1034. 串行调试接口引脚说明 990
表 1035. 并行跟踪引脚描述 990
表 1036. JTAG TAP 识别 993
表 1037. Cortex-M4 指令集汇总 994
表 1038. Cortex-M4 DSP 指令集汇总 998
表 1039. Cortex-M0 指令集汇总 1001
表 1040. 缩略词 1004

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

48.4 图

图 1. LPC43xx 框图 7
图 2. 双核框图 9
图 3. 系统存储器映射 （有关所有外围设备的详细地址，请参见图 4） 15
图 4. 外围设备存储器映射 （有关存储器模块的详细地址，请参见图 3） 16
图 5. LPC43xx AHB 多层矩阵连接 17
图 6. OTP 驱动器指针结构 20
图 7. 启动流程 (仅适用于无片上闪存的器件) 25
图 8. CMAC 生成 27
图 9. UART 启动流程（仅适用于无片上闪存的器件） 28
图 10. EMC 启动流程（仅适用于无片上闪存的器件） 29
图 11. SPI 启动流程（仅适用于无片上闪存的器件） 30
图 12. SPIFI 启动流程（仅适用于无片上闪存的器件） 30
图 13. USB 启动流程（仅适用于无片上闪存的器件） 31
图 14. 引导流程时序 32
图 15. AES 驱动器指针结构 34
图 16. AES 解密流程 36
图 17. CMAC 生成 38
图 18. AES 字节顺序 39
图 19. 事件路由器框图 46
图 20. CGU 和 CCU1/2 框图 72
图 21. CGU 框图 75
图 22. PLL0 框图 100
图 23. 带小数分频器的 PLL0 103
图 24. PLL1 框图 104
图 25. RGU 框图 124
图 26. RGU 复位结构 128
图 27. I/O 管脚的框图 199
图 28. GIMA 和外设之间的连接 222
图 29. GIMA 输入阶段 224
图 30. SGPIO 本地输出引脚多路复用器配置 266
图 31. SGPIO 框图 277
图 32. 一个位串的基本操作 278
图 33. 级联互连 280
图 34. SGPIO_MUX_CFG 位串多路复用器设置 282
图 35. 5.1 通道 I2S 输出映射至 SGPIO 位串 284
图 36. I2S 配置 284
图 37. SGPIO 相机接口配置 287
图 38. DMA 控制器框图 291
图 39. LLI 示例 317
图 40. SD/MMC 功能框图 320
图 41. 双缓冲描述符结构 359
图 42. 链状描述符结构 360
图 43. EMC 框图 (SDRAM) 368
图 44. EMC 框图 (SRAM) 369
图 45. 32 位存储组别外部存储器接口（位 MW = 10） 392
图 46. 16 位存储组别外部存储器接口（位 MW = 01） 392
图 47. 8 位存储组别外部存储器接口（位 MW = 00） 393
图 48. 高速 USB OTG 框图 398
图 49. USB 控制器模式 403
图 50. 端点队列头组织 442
图 51. 端点队列头数据结构 444
图 52. 设备状态图 449
图 53. 端点队列头图 459
图 54. 软件链接指针 460
图 55. 设备功率状态图 464
图 56. 主机 /OTG 电源状态图 465
图 57. 带 ULPI 的 USB1 框图 468
图 58. 带内部全速 PHY 的 USB1 框图 469
图 59. USB 设备驱动器指针结构 507
图 60. 以太网框图 546
图 61. 中断生成 575
图 62. 唤醒帧滤波寄存器 578
图 63. 网络时间同步 582
图 64. 使用微调方式的系统更新 584
图 65. 支持点对点路径校正的时钟的传播延迟计算 . . 587
图 66. 描述符环状和链状结构 596
图 67. 默认模式下的 TxDMA 操作 599
图 68. OSF 模式下的 TxDMA 操作 601
图 69. 接收 DMA 操作 604
图 70. 发送器描述符字段 607
图 71. 发送描述符获取 （读取） 608
图 72. 接收描述符字段 （备用配置） 612
图 73. LCD 控制器框图 639
图 74. 游标移动 647
图 75. 游标剪切 648
图 76. 游标图像格式 649
图 77. 加电和掉电序列 654
图 78. STN 显示器的水平时序 655
图 79. STN 显示器的垂直时序 656
图 80. TFT 显示器的水平时序 656
图 81. TFT 显示器的垂直时序 657
图 82. SCT 框图 662
图 83. SCT 计数器和选择逻辑 663
图 84. 匹配逻辑 685
图 85. 捕获逻辑 685
图 86. 事件选择 686
图 87. 输出限幅 i 686
图 88. SCT 中断生成 686
图 89. SCT 配置示例 692
图 90. 定时器周期，其中 PR = 2、 MRx = 6，并且使能了在匹配时同时执行中断和复位操作 707
图 91. 定时器周期，其中 PR = 2、 MRx = 6，并且使能了在匹配时同时执行中断和停止操作 707
图 92. 定时器框图 708
图 93. MCPWM 框图 711
图 94. 边沿对齐的 PWM 波形，无死区且 POLA = 0 . 730
图 95. 中心对齐的 PWM 波形，无死区且 POLA = 0 . 731
图 96. 边缘对齐的 PWM 波形，有死区且 POLA = 0 . 731
图 97. 中心对齐的 PWM 波形，有死区且 POLA = 0 . 732
图 98. 三相直流模式采样波形 734
图 99. 三相交流模式采样波形，边缘对齐的 PWM 模式 735

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

图 100. 编码器接口框图 737
图 101. 正交编码器基本操作 749
图 102. 重复性中断定时器 (RIT) 框图 752
图 103. 看门狗功能框图 764
图 104. 使能窗口模式时提前看门狗喂狗 765
图 105. 使能窗口模式时正确的看门狗喂狗 765
图 106. 看门狗警告中断 765
图 107. RTC 功能框图 767
图 108. USART 框图 780
图 109. 自动波特率 a) 模式 0 和 b) 模式 1 的波形图 . . 793
图 110. 设置 USART 分频器的算法 796
图 111. USART 串行接口协议 803
图 112. 以同步从模式发送数据 804
图 113. 典型智能卡应用 806
图 114. 智能卡 T = 0 波形 807
图 115. 自动 RTS 功能时序 818
图 116. 自动 CTS 功能时序 819
图 117. 自动波特率 a) 模式 0 和 b) 模式 1 的波形图 . . 824
图 118. 设置 UART 分频器的算法 826
图 119. UART1 功能框图 832
图 120. 德州仪器同步串行帧格式：a) 单帧和 b) 连续 / 背靠背 2 帧传输 841
图 121. CPOL=0 且 CPHA=0 时的 SPI 帧格式 （a) 单帧和 b) 连续帧传输） 842
图 122. CPOL=0 且 CPHA=1 时的 SPI 帧格式 843
图 123. CPOL = 1 且 CPHA = 0 时的 SPI 帧格式 （a) 单帧和 b) 连续帧传输） 844
图 124. CPOL = 1 且 CPHA = 1 时的 SPI 帧格式 845
图 125. Microwire 帧格式 （单帧传输） 846
图 126. Microwire 帧格式 （连续传送） 847
图 127. Microwire 帧格式建立及保持时间 847
图 128. SPI 功能框图 849
图 129. SPI 数据传输格式（CPHA = 0 和 CPHA = 1） 854
图 130. WS 信号连接 859
图 131. 简单 I2S 配置及总线时序 860
图 132. 典型的发送器主机模式 （有或没有 MCLK 输出） 870
图 133. 典型发送器主机模式，带或不带 MCLK 输出，使用音频 PLL 870
图 134. 发送器主机模式共享接收器的参考时钟 870
图 135. 4 线发送器主机模式共享接收器的位时钟和 WS 870
图 136. 典型的发送器从机模式 871
图 137. 发送器从机模式共享接收器的参考时钟 871
图 138. 4 线发送器从机模式共享接收器的位时钟和 WS 871
图 139. 典型的接收器主机模式 （有或没有 MCLK 输出） 873
图 140. 典型接收器主机模式，带或不带 MCLK 输出，使用音频 PLL 873
图 141. 接收器主机模式共享发送器参考时钟 873
图 142. 4 线接收器主机模式共享发送器的位时钟和 WS 873
图 143. 典型的接收器从机模式 874
图 144. 接收器从机模式共享发送器的参考时钟 874
图 145. 4 线接收器从机模式共享发送器的位时钟和 WS 874
图 146. 不同 I2S 模式的 FIFO 内容 875
图 147. C_CAN 功能框图 877
图 148. 消息对象传送的功能框图 886
图 149. 静默模式下的 CAN 内核 904
图 150. 环回模式下的 CAN 内核 905
图 151. 结合了静默模式的环回模式下的 CAN 内核 . . 905
图 152. 消息对象传送的功能框图 907
图 153. 将消息从 FIFO 缓冲区读取到消息缓冲区 912
图 154. 位时序 914
图 155. I2C 总线配置 916
图 156. 主发送器模式中的格式 926
图 157. 主接收器模式的格式 926
图 158. 发送重复起始信号后，主接收器切换至主发送器 927
图 159. 从属接收器模式的格式 927
图 160. 从属发送器模式的格式 928
图 161. I2C 串行接口功能框图 929
图 162. 仲裁过程 931
图 163. 串行时钟同步 931
图 164. 主发送模式下的格式和状态 935
图 165. 主接收模式下的格式和状态 937
图 166. 从机接收模式下的格式和状态 940
图 167. 从发送模式下的格式和状态 942
图 168. 来自两个主机的同时 “重复起动”条件 943
图 169. 强制访问忙 I2C 总线 944
图 170. 从因 SDA 低电平而导致总线受阻的状态中恢复 944
图 171. 使用 DMA 中断和定时器的 DAC 控制 964
图 172. 带闪存的 LPC43xx 器件的启动过程流程图 . . 968
图 173. IAP 参数传递 981
图 174. 生成 128 位签名的算法 988
图 175. ARM 标准 JTAG 连接器 991
图 176. Cortex 调试连接器 991
图 177. Cortex 调试 + ETM 连接器 992

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

48.5 内容

第 1 章：简介信息

1.1 简介 3
1.2 功能要点 3
1.3 订购信息 （无闪存的器件 LPC4350/30/20/10） . . 6
1.4 功能框图 7

第 2 章：LPC43xx ARM Cortex-M0 协处理器和内核间通信 (IPC)

2.1 本章导读 8
2.2 基本配置 8
2.3 简介 8
2.4 简介 8
2.4.1 硬件 9
2.4.2 中断处理 9
2.5 IPC 协议描述 9
2.5.1 IPC 队列 10
2.5.2 协议 10
2.5.3 示例 12

第 3 章：LPC43xx 存储器映射

3.1 本章导读 13
3.2 基本配置 13
3.3 存储器配置 13
3.3.1 片内静态 RAM 13
3.3.2 位带机制 14
3.3.3 掉电模式下的存储器保持 14
3.3.4 存储器保护单元 (MPU) 14
3.4 存储器映射 15
3.5 AHB 多层矩阵配置 17

第 4 章：LPC43xx 一次性可编程 (OTP) 存储器和 API

4.1 本章导读 18
4.2 功能要点 18
4.3 简介 18
4.4 寄存器描述 18
4.5 OTP API 20
4.5.1 OTP 功能分配 21
4.5.2 OTP API 调用 21

第 5 章：LPC43xx Boot ROM

5.1 本章导读 22
5.2 功能要点 22
5.3 功能说明 22
5.3.1 AES 兼容设备 24
5.3.2 启动流程 (仅适用于无片上闪存的器件) 24
5.3.3 启动映像格式 26
5.3.4 启动映像创建 26
5.3.4.1 CMAC 26
5.3.4.2 UART 引导模式 27
5.3.4.3 EMC 引导模式 29
5.3.4.4 SPI 引导模式 29
5.3.4.5 SPIFI 引导模式 30
5.3.4.6 USB 启动模式 30
5.3.5 引导流程时序 31
5.3.6 ISP 32

第 6 章：LPC43xx 安全 API

6.1 本章导读 33
6.2 功能要点 33
6.3 简介 33
6.4 AES API 34
6.4.1 AES 功能分配 34
6.5 功能说明 36
6.5.1 AES 解密 36
6.5.2 使用 AES 硬件加速的 CMAC 37
生成子密钥 37
生成 CMAC 标签 37
验证 CMAC 标签 38
6.5.3 AES 密钥的使用 38
6.5.4 字节顺序 39
6.5.5 在深度掉电模式下存储 AES 密钥 40

第 7 章：LPC43xx 可嵌套向量中断控制器 (NVIC)

7.1 本章导读 41
7.2 基本配置 41
7.3 功能要点 41
7.4 简介 41
7.5 引脚描述 41
7.6 中断源 42
7.6.1 Cortex-M4 中断源 42
7.6.2 Cortex-M0 中断源 44

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

第 8 章：LPC43xx 事件路由器

8.1 本章导读 45
8.2 基本配置 45
8.3 简介 45
8.4 事件路由器输入 46
8.5 引脚描述 47
8.6 寄存器描述 47
8.6.1 电平配置寄存器 47
8.6.2 边沿配置寄存器 50
8.6.3 清除事件使能寄存器 52
8.6.4 设置事件使能寄存器 53
8.6.5 事件状态寄存器 54
8.6.6 事件使能寄存器 55
8.6.7 清除事件状态寄存器 56
8.6.8 设置事件状态寄存器 57

第 9 章：LPC43xx 配置寄存器 (CREG)

9.1 本章导读 58
9.2 基本配置 58
9.3 功能要点 58
9.4 寄存器描述 59
9.4.1 CREG0 控制寄存器 60
9.4.2 ARM Cortex-M4 存储器映射寄存器 61
9.4.3 CREG5 控制寄存器 61
9.4.4 DMA 复用寄存器 62
9.4.5 ETB SRAM 配置寄存器 64
9.4.6 CREG6 控制寄存器 64
9.4.7 Cortex-M4 TXEV 事件清除寄存器 66
9.4.8 器件 ID 寄存器 66
9.4.9 Cortex-M0 TXEV 事件清除寄存器 66
9.4.10 ARM Cortex-M0 存储器映射寄存器 66

第 10 章：LPC43xx 电源管理控制器 (PMC)

10.1 本章导读 67
10.2 基本配置 67
10.3 简介 67
10.3.1 工作模式 67
10.3.2 睡眠模式 68
10.3.3 深度睡眠模式 68
10.3.4 掉电模式 68
10.3.5 深度掉电模式 68
10.3.6 掉电模式下的存储器保持 69
10.4 寄存器描述 69
10.4.1 硬件睡眠事件使能寄存器 PD0_SLEEP0_HW_ENA 69
10.4.2 掉电模式寄存器 PD0_SLEEP0_MODE 69
10.5 功能说明 70
10.5.1 运行时编程 70

第 11 章：LPC43xx 时钟产生单元 (CGU)

11.1 本章导读 71
11.2 基本配置 71
11.3 功能要点 71
11.4 简介 71
11.5 引脚描述 75
11.6 寄存器描述 76
11.6.1 频率监控寄存器 78
11.6.2 晶体振荡器控制寄存器 80
11.6.3 PLL0USB 寄存器 80
11.6.3.1 PLL0USB 状态寄存器 80
11.6.3.2 PLL0USB 控制寄存器 80
11.6.3.3 PLL0USB M 分频器寄存器 82
11.6.3.4 PLL0USB NP 分频器寄存器 83
11.6.4 PLL0AUDIO 寄存器 83
11.6.4.1 PLL0AUDIO 状态寄存器 83
11.6.4.2 PLL0AUDIO 控制寄存器 84
11.6.4.3 PLL0AUDIO M 分频器寄存器 85
11.6.4.4 PLL0AUDIO NP 分频器寄存器 86
11.6.4.5 PLL0AUDIO 小数分频器寄存器 86
11.6.5 PLL1 寄存器 87
11.6.5.1 PLL1 状态寄存器 87
11.6.5.2 PLL1 控制寄存器 87
11.6.6 整数分频器寄存器 A 89
11.6.7 整数分频器寄存器 B、 C、 D 90
11.6.8 整数分频器寄存器 E 91
11.6.9 BASE_SAFE_CLK 控制寄存器 92
11.6.10 BASE_USB0_CLK 控制寄存器 92
11.6.11 BASE_PERIPH_CLK 控制寄存器 93
11.6.12 BASE_USB1_CLK 控制寄存器 94
11.6.13 BASE_M4_CLK 至 BASE_UART3_CLK 控制寄存器 95
11.6.14 BASE_OUT_CLK 寄存器 96
11.6.15 BASE_APLL_CLK 寄存器 97
11.6.16 BASE_CGU_OUT0_CLK 至 BASE_CGU_OUT1_CLK 寄存器 98
11.7 功能描述 99
11.7.1 32 kHz 振荡器 99
11.7.2 IRC 99
11.7.3 晶体振荡器 99
11.7.4 PLL0 （PLL0USB 和 PLL0AUDIO） 99
11.7.4.1 功能要点 99
11.7.4.2 PLL0 描述 99
11.7.4.3 PLL0 工作模式的使用 100
11.7.4.3.1 正常模式 100
11.7.4.3.2 模式 1a：正常工作模式，无需后分频器和预分频器 101

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

11.7.4.3.3 模式 1b：正常工作模式，需要后分频器，无需预分频器 101
11.7.4.3.4 模式 1c：正常工作模式，无需后分频器，需要预分频器 101
11.7.4.3.5 模式 1d：正常工作模式，需要后分频器和预分频器 101
11.7.4.3.6 模式 3：掉电模式 (pd) 102
11.7.4.4 USB0 的设置 102
11.7.4.5 使用说明 102
11.7.5 PLL0AUDIO 的小数分频器 102
11.7.6 PLL1 103
11.7.6.1 功能要点 103
11.7.6.2 PLL1 描述 104
11.7.6.3 锁定检测器 104
11.7.6.4 掉电控制 104
11.7.6.5 可选择的反馈分频器时钟 104
11.7.6.6 直接输出模式 105
11.7.6.7 分频器比率编程 105
预分频器 105
后分频器 105
反馈分频器 105
更改分频器值 105
11.7.6.8 频率选择 105
整数模式 105
非整数模式 106
直接模式 106
掉电模式 106
11.8 CGU 配置示例 107
11.8.1 深度睡眠和掉电模式的 CGU 编程 107
11.8.2 对 CGU 进行编程以使 I2S 的外围设备时钟速率达到 30 MHz 107
11.8.3 针对 USB 应用的 PLL0USB 设置 108
11.8.4 音频应用的 PLL0AUDIO 设置 108
11.8.4.1 使用小数分频器 108
11.8.4.2 绕过小数分频器 110

第 12 章：LPC43xx 时钟控制单元 (CCU)

12.1 本章导读 112
12.2 基本配置 112
12.3 功能要点 112
12.4 简介 113
12.5 寄存器描述 115
12.5.1 电源模式寄存器 119
12.5.2 基础时钟状态寄存器 119
12.5.3 CCU1/2 分支时钟配置寄存器 121
12.5.4 CCU1/2 分支时钟状态寄存器 123

第 13 章：LPC43xx 复位产生单元 (RGU)

13.1 本章导读 124
13.2 基本配置 124
13.3 简介 124
13.3.1 复位等级 127
13.4 寄存器概述 128
13.4.1 RGU 复位控制寄存器 130
13.4.2 RGU 复位状态寄存器 132
13.4.3 RGU 复位活动状态寄存器 139
13.4.4 复位外部状态寄存器 143
13.4.4.1 CORE_RST 的复位外部状态寄存器 0 144
13.4.4.2 PERIPH_RST 的复位外部状态寄存器 1 144
13.4.4.3 MASTER_RST 的复位外部状态寄存器 2 . . . 145
13.4.4.4 WWDT_RST 的复位外部状态寄存器 4 145
13.4.4.5 CREG_RST 的复位外部状态寄存器 5 145
13.4.4.6 PERIPHERAL_RESET 的复位外部状态寄存器 145
13.4.4.7 MASTER_RESET 的复位外部状态寄存器 . . 146

第 14 章：LPC43xx 引脚配置

14.1 本章导读 147
14.2 引脚说明 147

第 15 章：LPC43xx 系统控制单元 (SCU)/IO 配置

15.1 本章导读 198
15.2 基本配置 198
15.3 简介 198
15.3.1 数字引脚功能 199
15.3.2 数字引脚模式 199
15.3.3 输入缓冲 200
15.3.4 可编程干扰滤波器 200
15.3.5 可编程转换速率 200
15.3.6 高速引脚 200
15.3.7 高驱动能力引脚 200
15.3.8 I^2C0 总线引脚 200
15.3.9 USB1 USB1_DP/USB1_DM 引脚 200
15.3.10 EMC 信号延迟控制 200
15.3.11 引脚复用 201
15.4 寄存器描述 201
15.4.1 标准电平驱动引脚的引脚配置寄存器 208
15.4.2 高驱动能力引脚的引脚配置寄存器 209
15.4.3 高速引脚的引脚配置寄存器 210
15.4.4 USB1 引脚 USB1_DP/USB1_DM 的引脚配置寄存器 211
15.4.5 开漏 I^2C 总线引脚的引脚配置寄存器 212
15.4.6 ADC0 功能选择寄存器 213
15.4.7 ADC1 功能选择寄存器 214
15.4.8 模拟功能选择寄存器 216
15.4.8.1 测量带隙 216
15.4.9 EMC 时钟延迟寄存器 217
15.4.10 引脚中断选择寄存器 0 217
15.4.11 引脚中断选择寄存器 1 219

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

第 16 章：LPC43xx 全局输入多路复用器阵列 (GIMA)

16.1 本章导读 221
16.2 基本配置 221
16.3 简介 221
16.3.1 GIMA 事件输入选择 221
16.3.2 GIMA 时钟同步 224
16.4 寄存器描述 224
16.4.1 定时器 0 CAP0_0 捕获输入多路复用器 (CAP0_0_IN) 225
16.4.2 定时器 0 CAP0_1 捕获输入多路复用器 (CAP0_1_IN) 226
16.4.3 定时器 0 CAP0_2 捕获输入多路复用器 (CAP0_2_IN) 226
16.4.4 定时器 0 CAP0_3 捕获输入多路复用器 (CAP0_3_IN) 227
16.4.5 定时器 1 CAP1_0 捕获输入多路复用器 (CAP1_0_IN) 227
16.4.6 定时器 1 CAP1_1 捕获输入多路复用器 (CAP1_1_IN) 228
16.4.7 定时器 1 CAP1_2 捕获输入多路复用器 (CAP1_2_IN) 228
16.4.8 定时器 1 CAP1_3 捕获输入多路复用器 (CAP1_3_IN) 229
16.4.9 定时器 2 CAP2_0 捕获输入多路复用器 (CAP2_0_IN) 229
16.4.10 定时器 2 CAP2_1 捕获输入多路复用器 (CAP2_1_IN) 230
16.4.11 定时器 2 CAP2_2 捕获输入多路复用器 (CAP2_2_IN) 230
16.4.12 定时器 2 CAP2_3 捕获输入多路复用器 (CAP2_3_IN) 231
16.4.13 定时器 3 CAP3_0 捕获输入多路复用器 (CAP3_0_IN) 231
16.4.14 定时器 3 CAP3_1 捕获输入多路复用器 (CAP3_1_IN) 232
16.4.15 定时器 3 CAP3_2 捕获输入多路复用器 (CAP3_2_IN) 232
16.4.16 定时器 3 CAP3_3 捕获输入多路复用器 (CAP3_3_IN) 233
16.4.17 SCT CTIN_0 捕获输入多路复用器 (CTIN_0_IN) 234
16.4.18 SCT CTIN_1 捕获输入多路复用器 (CTIN_1_IN) 234
16.4.19 SCT CTIN_2 捕获输入多路复用器 (CTIN_2_IN) 235
16.4.20 SCT CTIN_3 捕获输入多路复用器 (CTIN_3_IN) 235
16.4.21 SCT CTIN_4 捕获输入多路复用器 (CTIN_4_IN) 236
16.4.22 SCT CTIN_5 捕获输入多路复用器 (CTIN_5_IN) 236
16.4.23 SCT CTIN_6 捕获输入多路复用器 (CTIN_6_IN) 237
16.4.24 SCT CTIN_7 捕获输入多路复用器 (CTIN_7_IN) 237
16.4.25 VADC 启动输入多路复用器 (VADC_TRIGGER_IN) 238
16.4.26 事件路由器输入 13 多路复用器 (EVENTROUTER_13_IN) 239
16.4.27 事件路由器输入 14 多路复用器 (EVENTROUTER_14_IN) 239
16.4.28 事件路由器输入 16 多路复用器 (EVENTROUTER_16_IN) 240
16.4.29 ADC 启动 0 输入多路复用器 (ADCSTART0_IN) 240
16.4.30 ADC 启动 1 输入多路复用器 (ADCSTART1_IN) 241

第 17 章：LPC43xx GPIO

17.1 本章导读 242
17.2 基本配置 242
17.3 功能要点 243
17.3.1 GPIO 引脚中断功能要点 243
17.3.2 GPIO 分组中断功能要点 243
17.3.3 GPIO 端口功能要点 243
17.4 简介 244
17.4.1 GPIO 引脚中断 244
17.4.2 GPIO 分组中断 244
17.4.3 GPIO 端口 244
17.5 寄存器描述 245
17.5.1 GPIO 引脚中断寄存器描述 249
17.5.1.1 引脚中断模式寄存器 249
17.5.1.2 引脚中断电平 （上升沿中断）使能寄存器 . . . 249
17.5.1.3 引脚中断电平 （上升沿中断）设置寄存器 . . . 249
17.5.1.4 引脚中断电平 （上升沿中断）清除寄存器 . . . 250
17.5.1.5 引脚中断有效电平 （下降沿中断使能）寄存器 250
17.5.1.6 引脚中断有效电平 （下降沿中断）设置寄存器 250
17.5.1.7 引脚中断有效电平 （下降沿中断）清除寄存器 251
17.5.1.8 引脚中断上升沿寄存器 251
17.5.1.9 引脚中断下降沿寄存器 251
17.5.1.10 引脚中断状态寄存器 252
17.5.2 GPIO 组 0/ 组 1 中断寄存器描述 252
17.5.2.1 分组中断控制寄存器 252
17.5.2.2 GPIO 分组中断端口极性寄存器 253
17.5.2.3 GPIO 分组中断端口使能寄存器 253
17.5.3 GPIO 端口寄存器描述 253
17.5.3.1 GPIO 端口字节引脚寄存器 253
17.5.3.2 GPIO 端口字引脚寄存器 253
17.5.3.3 GPIO 端口方向寄存器 254
17.5.3.4 GPIO 端口掩码寄存器 254
17.5.3.5 GPIO 端口引脚寄存器 254
17.5.3.6 GPIO 屏蔽端口引脚寄存器 255
17.5.3.7 GPIO 端口设置寄存器 255
17.5.3.8 GPIO 端口清除寄存器 255

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

17.5.3.9 GPIO 端口切换寄存器 255
17.6 功能描述 256
17.6.1 读取引脚状态 256
17.6.2 GPIO 输出 256
17.6.3 I/O 的掩码 256
17.6.4 GPIO 中断 257
17.6.4.1 引脚中断 257
17.6.4.2 分组中断 257
17.6.5 推荐用法 257

第 18 章：LPC43xx 串行 GPIO (SGPIO)

18.1 本章导读 259
18.2 基本配置 259
18.3 功能要点 259
18.4 简介 259
18.4.1 中断 260
18.5 引脚说明 261
18.6 寄存器描述 262
18.6.1 引脚多路复用器配置寄存器 （OUT_MUX_CFG0 至 15） 263
18.6.2 SGPIO 多路复用器配置寄存器 （SGPIO_MUX_CFG0 至 15） 266
18.6.3 位串多路复用器配置寄存器 （SLICE_MUX_CFG0 至 15） 268
18.6.4 位串数据寄存器 （REG0 至 15） 269
18.6.5 位串数据影子寄存器 （REG_SS0 至 15） . . . 269
18.6.6 重新加载寄存器 （PRESET0 至 15） 270
18.6.7 下行计数器寄存器 （COUNT0 至 15） 270
18.6.8 位置寄存器 （POS0 至 15） 270
18.6.9 位串 A 掩码寄存器 (MASK_A) 271
18.6.10 位串 H 掩码寄存器 (MASK_H) 271
18.6.11 位串 I 掩码寄存器 (MASK_I) 271
18.6.12 位串 P 掩码寄存器 (MASK_P) 271
18.6.13 GPIO 输入状态寄存器 (GPIO_INREG) 271
18.6.14 GPIO 输出控制寄存器 (GPIO_OUTREG) . . . 272
18.6.15 GPIO 输出使能寄存器 (GPIO_OENREG) . . . 272
18.6.16 位串计数使能寄存器 (CTRL_ENABLED) 272
18.6.17 位串计数禁用寄存器 (CTRL_DISABLED) . . . 272
18.6.18 移位时钟中断清除掩码寄存器 (CLR_EN_0) . 273
18.6.19 移位时钟中断设置掩码寄存器 (SET_EN_0) . . 273
18.6.20 移位时钟中断使能寄存器 (ENABLE_0) 273
18.6.21 移位时钟中断状态寄存器 (STATUS_0) 273
18.6.22 移位时钟中断清除状态寄存器 (CTR_STATUS_0) 273
18.6.23 移位时钟中断设置状态寄存器 (SET_STATUS_0) 274
18.6.24 捕获时钟中断清除掩码寄存器 (CLR_EN_1) 274
18.6.25 捕获时钟中断设置掩码寄存器 (SET_EN_1) 274
18.6.26 捕获时钟中断使能 (ENABLE_1) 274
18.6.27 捕获时钟中断状态寄存器 (STATUS_1) 274
18.6.28 捕获时钟中断清除状态寄存器 (CTR_STATUS_1) 274
18.6.29 捕获时钟中断设置状态寄存器 (SET_STATUS_1) 275
18.6.30 模式匹配中断清除掩码寄存器 (CLR_EN_2) 275
18.6.31 模式匹配中断设置掩码寄存器 (SET_EN_2) 275
18.6.32 模式匹配中断使能 (ENABLE_2) 275
18.6.33 模式匹配中断状态寄存器 (STATUS_2) 275
18.6.34 模式匹配中断清除状态寄存器 (CTR_STATUS_2) 275
18.6.35 模式匹配中断设置状态寄存器 (SET_STATUS_2) 276
18.6.36 输入位中断清除掩码寄存器 (CLR_EN_3) . . . 276
18.6.37 输入位匹配中断设置掩码寄存器 (SET_EN_3) 276
18.6.38 输入位匹配中断使能 (ENABLE_3) 276
18.6.39 输入位匹配中断状态寄存器 (STATUS_3) . . . 276
18.6.40 输入位匹配中断清除状态寄存器 (CTR_STATUS_3) 276
18.6.41 输入位匹配中断设置状态寄存器 (SET_STATUS_3) 277
18.7 功能说明 277
18.7.1 级联 278
18.7.2 模式匹配 281
18.7.3 引脚复用 281
18.7.4 位串多路复用器 281
18.7.5 内部连接 283
18.8 示例 283
18.8.1 多通道 I2S 284
18.8.1.1 I2S 位串选择 284
18.8.1.2 I2S 位串配置 285
18.8.1.3 I2S 位串编程 287
18.8.2 相机接口示例 287
18.8.2.1 相机接口位串配置 288

第 19 章：LPC43xx 通用 DMA (GPDMA) 控制器

19.1 本章导读 290
19.2 基本配置 290
19.3 功能要点 290
19.4 简介 291
19.5 DMA 系统连接 291
19.5.1 DMA 请求信号 293
19.5.2 DMA 响应信号 294
19.6 寄存器描述 294
19.6.1 DMA 中断状态寄存器 296
19.6.2 DMA 中断终结计数请求状态寄存器 296
19.6.3 DMA 中断终结计数请求清除寄存器 296
19.6.4 DMA 中断错误状态寄存器 297
19.6.5 DMA 中断错误清除寄存器 297
19.6.6 DMA 原始中断终结计数状态寄存器 297
19.6.7 DMA 原始错误中断状态寄存器 297

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

19.6.8 DMA 使能通道寄存器 298
19.6.9 DMA 软件连发传送请求寄存器 298
19.6.10 DMA 软件单次请求寄存器 298
19.6.11 DMA 软件最后一个连发请求寄存器 299
19.6.12 DMA 软件最后一个单次请求寄存器 299
19.6.13 DMA 配置寄存器 299
19.6.14 DMA 同步寄存器 300
19.6.15 DMA 通道寄存器 300
19.6.16 DMA 通道源地址寄存器 300
19.6.17 DMA 通道目标地址寄存器 301
19.6.18 DMA 通道链接列表项寄存器 301
19.6.19 DMA 通道控制寄存器 301
19.6.19.1 保护和访问信息 303
19.6.20 通道配置寄存器 304
19.6.20.1 锁定控制 306
19.6.20.2 流控制和传输类型 306
19.7 功能说明 307
19.7.1 DMA 控制器功能描述 307
19.7.1.1 AHB 从机接口 307
19.7.1.2 控制逻辑和寄存器组 307
19.7.1.3 DMA 请求和响应接口 307
19.7.1.4 通道逻辑和通道寄存器组 307
19.7.1.5 中断请求 307
19.7.1.6 AHB 主机接口 307
19.7.1.6.1 总线和传输宽度 307
19.7.1.6.2 字节顺序特性 307
19.7.1.6.3 错误条件 310
19.7.1.7 通道硬件 310
19.7.1.8 DMA 请求优先级 310
19.7.1.9 中断生成 310
19.8 使用 DMA 控制器 311
19.8.1 编程 DMA 控制器 311
19.8.1.1 使能 DMA 控制器 311
19.8.1.2 禁用 DMA 控制器 311
19.8.1.3 使能 DMA 通道 311
19.8.1.4 禁用 DMA 通道 311
禁用 DMA 通道且丢失 FIFO 中的数据 311
禁用 DMA 通道且不丢失 FIFO 中的数据 311
19.8.1.5 设置新的 DMA 传输 311
19.8.1.6 中止 DMA 通道 312
19.8.1.7 编程 DMA 通道 312
19.8.2 流控制 312
19.8.2.1 外围设备到存储器或存储器到外围设备 DMA 流 313
19.8.2.2 外围设备到外围设备 DMA 流 313
19.8.2.3 存储器到存储器 DMA 流 314
19.8.3 中断请求 314
19.8.3.1 硬件中断序列流 315
19.8.4 地址产生 315
19.8.4.1 跨边界的字对齐传输 315
19.8.5 分散 / 聚集 315
19.8.5.1 链接列表项 316
19.8.5.1.1 编程 DMA 控制器，用于分散 / 聚集 DMA . . . 316
19.8.5.1.2 分散 / 聚集 DMA 示例 316

第 20 章：LPC43xx SD/MMC 接口

20.1 本章导读 319
20.2 基本配置 319
20.3 功能要点 319
20.4 简介 319
20.5 引脚说明 320
20.6 寄存器描述 321
20.6.1 控制寄存器 (CTRL) 322
20.6.2 电源使能寄存器 (PWREN) 324
20.6.3 时钟分频器寄存器 (CLKDIV) 324
20.6.4 SD 时钟源寄存器 (CLKSRC) 324
20.6.5 时钟使能寄存器 (CLKENA) 325
20.6.6 超时寄存器 (TMOUT) 325
20.6.7 卡类型寄存器 (CTYPE) 326
20.6.8 块大小寄存器 (BLKSIZ) 326
20.6.9 字节计数寄存器 (BYTCNT) 326
20.6.10 中断掩码寄存器 (INTMASK) 326
20.6.11 指令参数寄存器 (CMDARG) 327
20.6.12 指令寄存器 (CMD) 328
20.6.13 响应寄存器 0 (RESP0) 330
20.6.14 响应寄存器 1 (RESP1) 330
20.6.15 响应寄存器 2 (RESP2) 330
20.6.16 响应寄存器 3 (RESP3) 330
20.6.17 掩码中断状态寄存器 (MINTSTS) 331
20.6.18 原始中断状态寄存器 (RINTSTS) 332
20.6.19 状态寄存器 (STATUS) 333
20.6.20 FIFO 阈值水印寄存器 (FIFOTH) 334
20.6.21 卡检测寄存器 (CDETECT) 335
20.6.22 写保护寄存器 (WRTPRT) 336
20.6.23 CIU 卡已传输字节计数寄存器 (TCBCNT) . . . 336
20.6.24 主机至 BIU-FIFO 已传输字节计数寄存器 (TBBCNT) 336
20.6.25 去抖动计数寄存器 (DEBNCE) 336
20.6.26 UHS-1 寄存器 (UHS_REG) 337
20.6.27 硬件复位 (RST_N) 337
20.6.28 总线模式寄存器 (BMOD) 337
20.6.29 查询需求寄存器 (PLDMND) 338
20.6.30 描述符表基地址寄存器 (DBADDR) 338
20.6.31 内部 DMAC 状态寄存器 (IDSTS) 339
20.6.32 内部 DMAC 中断使能寄存器 (IDINTEN) 340
20.6.33 当前主机描述符地址寄存器 (DSCADDR) . . . 340
20.6.34 当前缓冲描述符地址寄存器 (BUFADDR) . . . 340
20.7 功能说明 340
20.7.1 自动停止。 340
20.7.2 软件 / 硬件限制 342
20.7.3 编程序列 343
20.7.3.1 初始化 343
20.7.3.2 枚举的卡协议栈 344
20.7.3.3 时钟编程 345
20.7.3.4 带或不带响应序列的无数据指令 345
20.7.3.5 数据传输指令 346
20.7.3.6 单块或多块读取 347
20.7.3.7 单块或多块写入 348
20.7.3.8 流读取 349
20.7.3.9 流写入 350
20.7.3.10 传输过程中发送 Stop 或 Abort 350

本文件所提供的全部信息均遵循法律声明的规定。 © NXP B.V. 2012. 保留所有权利。

20.7.4 挂起或恢复序列 351
20.7.4.1 Read_Wait 序列 352
20.7.4.2 CE-ATA 数据传输指令 352
20.7.4.2.1 复位和设备恢复 352
20.7.4.2.2 ATA 任务文件传输 352
20.7.4.2.3 使用 RW_MULTIPLE_BLOCK (RW_BLK) 执行 ATA 有效载荷传输 354
20.7.4.2.4 发送指令完成信号禁用 356
20.7.4.2.5 指令完成信号超时后进行恢复 356
20.7.4.2.6 减少 ATA 指令集 356
20.7.4.3 控制器 /DMA/FIFO 复位使用 358
20.7.4.4 错误处理 358
20.7.5 DMA 描述符 359
20.7.5.1 SD/MMC DMA 描述符 360
20.7.5.1.1 SD/MMC DMA 描述符 DESC0 360
20.7.5.1.2 SD/MMC DMA 描述符 DESC1 361
20.7.5.1.3 SD/MMC DMA 描述符 DESC2 361
20.7.5.1.4 SD/MMC DMA 描述符 DESC3 361
20.7.5.2 初始化 362
20.7.5.3 主机总线连发访问 362
20.7.5.4 主机数据缓冲区对齐 362
20.7.5.5 缓冲区大小计算 362
20.7.5.6 传输 362
20.7.5.7 接收 363
20.7.5.8 中断 363
20.7.5.9 中止 364
20.7.5.10 FBE 案例 364
20.7.5.11 FIFO 上溢和下溢 364
20.7.5.12 PBL 和水印级编程 364

第 21 章：LPC43xx 外部存储控制器 (EMC)

21.1 本章导读 366
21.2 基本配置 366
21.3 功能要点 367
21.4 简介 368
21.5 存储组别选择 369
21.6 引脚说明 370
21.7 寄存器描述 370
21.7.1 EMC 控制寄存器 372
21.7.2 EMC 状态寄存器 373
21.7.3 EMC 配置寄存器 374
21.7.4 动态存储器控制寄存器 374
21.7.5 动态存储器刷新定时器寄存器 375
21.7.6 动态存储器读配置寄存器 376
21.7.7 动态存储器预充电命令周期寄存器 376
21.7.8 动态存储器有效到预充电命令周期寄存器 . . .377
21.7.9 动态存储器自刷新退出时间寄存器 377
21.7.10 动态存储器最后一个数据输出到有效时间寄存器 377
21.7.11 动态存储器数据输入到有效命令时间寄存器 .378
21.7.12 动态存储器写恢复时间寄存器 378
21.7.13 动态存储器有效到有效命令周期寄存器 378
21.7.14 动态存储器自动刷新周期寄存器 379
21.7.15 动态存储器退出自刷新寄存器 379
21.7.16 动态存储器有效组 A 到有效组 B 时间寄存器 .379
21.7.17 动态存储器加载模式寄存器到有效命令时间 .380
21.7.18 静态存储器延长等待寄存器 380
21.7.19 动态存储器配置寄存器 381
21.7.20 动态存储器 RAS 和 CAS 延迟寄存器 383
21.7.21 静态存储器配置寄存器 383
21.7.22 静态存储器写使能延迟寄存器 385
21.7.23 静态存储器输出使能延迟寄存器 385
21.7.24 静态存储器读延迟寄存器 386
21.7.25 静态存储器页面模式读延迟寄存器 386
21.7.26 静态存储器写延迟寄存器 387
21.7.27 静态存储器周转期延迟寄存器 387
21.8 功能说明 388
21.8.1 AHB 从机寄存器接口 388
21.8.2 AHB 从机存储器接口 388
21.8.2.1 存储器事务字节序 388
21.8.2.2 存储器传输大小 388
21.8.2.3 写保护存储器区域 388
21.8.3 管脚接口 388
21.8.4 数据缓冲区 389
21.8.4.1 写缓冲区 389
21.8.4.2 读缓冲区 389
21.9 低功耗 390
21.9.1 低功耗 SDRAM 深度睡眠模式 390
21.9.2 低功耗 SDRAM 部分阵列刷新 390
21.10 外部静态存储器接口 391
21.10.1 32 位宽存储组别连接 391
21.10.2 16 位宽存储组别连接 392
21.10.3 8 位宽存储组别连接 393

第 22 章：LPC43xx SPI 闪存接口 (SPIFI)

22.1 本章导读 394
22.2 基本配置 394
22.3 功能要点 394
22.4 简介 394
22.5 引脚说明 395
22.6 支持的 QSPI 设备 395

第 23 章：LPC43xx USB0 主机 / 设备 /OTG 控制器

23.1 本章导读 397
23.2 基本配置 397
23.3 功能要点 397
23.4 简介 398
23.4.1 功能框图 398
23.4.2 关于 USB OTG 398
23.4.3 USB 首字母缩略词和缩写 398
23.4.4 发送和接收缓冲区 399
23.4.5 固定的端点配置 399

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

23.5 引脚说明 400
23.6 寄存器描述 401
23.6.1 寄存器使用 402
23.6.2 设备 / 主机功能寄存器 403
23.6.3 USB 指令寄存器 (USBCMD) 405
23.6.3.1 设备模式 405
23.6.3.2 主机模式 406
23.6.4 USB 状态寄存器 (USBSTS) 408
23.6.4.1 设备模式 408
23.6.4.2 主机模式 409
23.6.5 USB 中断寄存器 (USBINTR) 411
23.6.5.1 设备模式 411
23.6.5.2 主机模式 412
23.6.6 帧索引寄存器 (FRINDEX) 413
23.6.6.1 设备模式 413
23.6.6.2 主机模式 413
23.6.7 设备地址 （DEVICEADDR – 设备）和周期性列表基址 （PERIODICLISTBASE – 主机）寄存器 414
23.6.7.1 设备模式 414
23.6.7.2 主机模式 414
23.6.8 端点列表地址寄存器 （端点 LISTADDR – 设备）和异步列表地址 （ASYNCLISTADDR – 主机）寄存器 415
23.6.8.1 设备模式 415
23.6.8.2 主机模式 415
23.6.9 TT 控制寄存器 (TTCTRL) 415
23.6.9.1 设备模式 415
23.6.9.2 主机模式 415
23.6.10 突发大小寄存器 (BURSTSIZE) 416
23.6.11 传输缓冲区填充调谐寄存器 (TXFILLTUNING) 416
23.6.11.1 设备控制器 416
23.6.11.2 主机控制器 416
23.6.12 BINTERVAL 寄存器 417
23.6.13 USB 端点 NAK 寄存器 (ENDPTNAK) 417
23.6.13.1 设备模式 417
23.6.13.2 主机模式 418
23.6.14 USB 端点 NAK 使能寄存器 (ENDPTNAKEN) 418
23.6.14.1 设备模式 418
23.6.14.2 主机模式 419
23.6.15 端口状态和控制寄存器 (PORTSC1) 419
23.6.15.1 设备模式 419
23.6.15.2 主机模式 421
23.6.16 OTG 状态和控制寄存器 (OTGSC) 425
23.6.17 USB 模式寄存器 (USBMODE) 427
23.6.17.1 设备模式 428
23.6.17.2 主机模式 429
23.6.18 USB 端点设置状态寄存器 (ENDPSETUPSTAT) 429
23.6.19 USB 端点启动寄存器 (ENDPTPRIME) 430
23.6.20 USB 端点刷新寄存器 (ENDPTFLUSH) 430
23.6.21 USB 端点状态寄存器 (ENDPTSTAT) 431
23.6.22 USB 端点完成寄存器 (ENDPTCOMPLETE) . 431
23.6.23 USB 端点 0 控制寄存器 (ENDPTCTRL0) 432
23.6.24 端点 1 到 5 控制寄存器 433
23.7 功能说明 435
23.7.1 OTG 核 435
23.7.2 主机数据结构 435
23.7.3 主机操作模式 435
23.7.4 ATX_RGEN 模块 435
23.7.5 ATX 收发器 435
23.7.6 操作模式 435
23.7.7 SOF/VF 指示器 435
23.7.8 硬件辅助 436
23.7.8.1 自动复位 436
23.7.8.2 数据脉冲 436
23.7.8.3 B 断开到 A 连接 （跳变到 A 外设状态） 436
23.8 与 EHCI 标准的偏差 437
23.8.1 嵌入式事务转换器功能 437
23.8.1.1 功能寄存器 437
23.8.1.2 操作寄存器 437
23.8.1.3 探索 438
23.8.1.4 数据结构 438
23.8.1.5 操作模式 439
23.8.1.5.1 微帧管道 439
23.8.1.6 拆分状态机 439
23.8.1.7 异步事务计划和缓冲区管理 439
23.8.1.8 定期事务计划和缓冲区管理 439
23.8.1.9 多事务转译器 440
23.8.2 设备操作 440
23.8.2.1 USBMODE 寄存器 440
23.8.2.2 寄存器文件的非零字段 440
23.8.2.3 SOF 中断 440
23.8.3 EHCI 其它变型 441
23.8.3.1 探索 441
23.8.3.1.1 端口复位 441
23.8.3.1.2 端口速度检测 441
23.9 设备数据结构 442
23.9.1 端点队列头 (dQH) 442
23.9.1.1 端点性能和特点 442
23.9.1.2 传输重叠 444
23.9.1.3 当前 dTD 指针 444
23.9.1.4 设置缓冲区 444
23.9.2 端点传输描述符 (dTD) 445
23.9.2.1 确定同步 IN 端点数据包的数目 447
示例 1 447
示例 2 447
23.10 设备操作模型 447
23.10.1 设备控制器初始化 447
23.10.2 端口状态和控制 448
23.10.3 总线复位 450
23.10.4 挂起 / 恢复 450
23.10.4.1 挂起 450
23.10.4.1.1 操作模式 451
23.10.4.2 恢复 451
23.10.5 管理端点 451
23.10.5.1 端点初始化 451
23.10.5.2 停止 452
23.10.5.3 数据切换 452
23.10.5.3.1 数据切换复位 452
23.10.5.3.2 数据切换禁止 452
23.10.6 数据包传输操作模式 453
23.10.6.1 启动发送端点 453
23.10.6.2 启动接收端点 453

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

23.10.7 中断 / 批量端点操作模式 453
23.10.7.1 中断 / 批量端点总线响应矩阵 454
23.10.8 控制端点操作模式 455
23.10.8.1 设置阶段 455
23.10.8.1.1 使用设置锁定机制处理设置数据包 455
23.10.8.1.2 使用绊网机制处理设置数据包 455
23.10.8.2 数据阶段 456
23.10.8.3 状态阶段 456
23.10.8.4 控制端点母线响应矩阵 456
23.10.9 同步端点操作模式 457
TX 数据包停用 457
RX 数据包停用 457
23.10.9.1 同步管道的同步化 458
23.10.9.2 同步端点母线响应矩阵 458
23.10.10 管理队列头 459
23.10.10.1 队列头初始化 459
23.10.10.2 设置传输的操作模式 459
23.10.11 使用传输描述符管理传输 460
23.10.11.1 软件链接指针 460
23.10.11.2 建立传输描述符 460
23.10.11.3 执行传输描述符 461
链接列表为空 461
链接列表不为空 461
23.10.11.4 传输完成 461
23.10.11.5 刷新 / 取消启动端点 462
23.10.11.6 设备错误矩阵 462
23.10.12 服务中断 462
23.10.12.1 高频中断 463
23.10.12.2 低频中断 463
23.10.12.3 错误中断 463
23.11 USB 功率优化 463
23.11.1 USB 功率状态 463
23.11.2 设备功率状态 464
23.11.3 主机功率状态 465
23.11.4 Susp_CTRL 模块 466

第 24 章：LPC43xx USB1 主机 / 设备控制器

24.1 本章导读 467
24.2 基本配置 467
24.2.1 全速模式，无外部 PHY 467
24.2.2 高速模式，有 ULPI 接口 467
24.3 功能要点 467
24.4 简介 468
24.5 引脚说明 469
24.6 寄存器描述 470
24.6.1 设备 / 主机功能寄存器 472
24.6.2 USB 指令寄存器 (USBCMD) 473
24.6.2.1 设备模式 474
24.6.2.2 主机模式 475
24.6.3 USB 状态寄存器 (USBSTS) 477
24.6.3.1 设备模式 477
24.6.3.2 主机模式 479
24.6.4 USB 中断寄存器 (USBINTR) 481
24.6.4.1 设备模式 481
24.6.4.2 主机模式 482
24.6.5 帧索引寄存器 (FRINDEX) 483
24.6.5.1 设备模式 483
24.6.5.2 主机模式 483
24.6.6 设备地址 (DEVICEADDR) 和周期性列表基址 (PERIODICLISTBASE) 寄存器 484
24.6.6.1 设备模式 484
24.6.6.2 主机模式 484
24.6.7 端点列表地址寄存器 (ENDPOINTLISTADDR) 和异步列表地址 (ASYNCLISTADDR) 寄存器 . . 485
24.6.7.1 设备模式 485
24.6.7.2 主机模式 485
24.6.8 TT 控制寄存器 (TTCTRL) 485
24.6.8.1 设备模式 485
24.6.8.2 主机模式 486
24.6.9 突发大小寄存器 (BURSTSIZE) 486
24.6.10 传输缓冲区填充调谐寄存器 (TXFILLTUNING) 486
24.6.10.1 设备控制器 486
24.6.10.2 主机控制器 486
24.6.11 USB ULPI 视口寄存器 (ULPIVIEWPORT) . . 487
24.6.12 BINTERVAL 寄存器 489
24.6.13 USB 端点 NAK 寄存器 (ENDPTNAK) 489
24.6.13.1 设备模式 489
24.6.13.2 主机模式 490
24.6.14 USB 端点 NAK 使能寄存器 (ENDPTNAKEN) 490
24.6.14.1 设备模式 490
24.6.14.2 主机模式 490
24.6.15 端口状态和控制寄存器 (PORTSC1) 490
24.6.15.1 设备模式 490
24.6.15.2 主机模式 493
24.6.16 USB 模式寄存器 (USBMODE) 497
24.6.16.1 设备模式 497
24.6.16.2 主机模式 498
24.6.17 USB 端点设置状态寄存器 (ENDPSETUPSTAT) 499
24.6.18 USB 端点启动寄存器 (ENDPTPRIME) 499
24.6.19 USB 端点刷新寄存器 (ENDPTFLUSH) 500
24.6.20 USB 端点状态寄存器 (ENDPTSTAT) 500
24.6.21 USB 端点完成寄存器 (ENDPTCOMPLETE) . 501
24.6.22 USB 端点 0 控制寄存器 (ENDPTCTRL0) . . . 501
24.6.23 端点 1 到 3 控制寄存器 502
24.7 功能说明 504

第 25 章：LPC43xx USB API

25.1 本章导读 505
25.2 简介 505
25.3 USB 驱动器函数 505
25.4 调用 USB 设备驱动器 506
25.5 USB API 507
25.5.1 __WORD_BYTE 507
25.5.2 _BM_T 507

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

25.5.3 _CDC_ABSTRACT_CONTROL_MANAGEMENT_DESCRIPTOR 508
25.5.4 _CDC_CALL_MANAGEMENT_DESCRIPTOR 508
25.5.5 _CDC_HEADER_DESCRIPTOR 508
25.5.6 _CDC_LINE_CODING 508
25.5.7 _CDC_UNION_1SLAVE_DESCRIPTOR 509
25.5.8 _CDC_UNION_DESCRIPTOR 509
25.5.9 _DFU_STATUS 509
25.5.10 _HID_DESCRIPTOR 509
25.5.11 _HID_DESCRIPTOR::_HID_DESCRIPTOR_LIST 510
25.5.12 _HID_REPORT_T 510
25.5.13 _MSC_CBW 510
25.5.14 _MSC_CSW 511
25.5.15 _REQUEST_TYPE 511
25.5.16 _USB_COMMON_DESCRIPTOR 511
25.5.17 _USB_CORE_DESCS_T 512
25.5.18 _USB_DEVICE_QUALIFIER_DESCRIPTOR 512
25.5.19 _USB_DFU_FUNC_DESCRIPTOR 513
25.5.20 _USB_INTERFACE_DESCRIPTOR 513
25.5.21 _USB_OTHER_SPEED_CONFIGURATION 514
25.5.22 _USB_SETUP_PACKET 514
25.5.23 _USB_STRING_DESCRIPTOR 515
25.5.24 _WB_T 515
25.5.25 USBD_API 515
25.5.26 USBD_API_INIT_PARAM 516
25.5.27 USBD_CDC_API 518
25.5.28 USBD_CDC_INIT_PARAM 519
25.5.29 USBD_CORE_API 523
25.5.30 USBD_DFU_API 526
25.5.31 USBD_DFU_INIT_PARAM 526
25.5.32 USBD_HID_API 529
25.5.33 USBD_HID_INIT_PARAM 529
25.5.34 USBD_HW_API 535
25.5.35 USBD_MSC_API 542
25.5.36 USBD_MSC_INIT_PARAM 542

第 26 章：LPC43xx 以太网

26.1 本章导读 545
26.2 基本配置 545
26.3 功能要点 545
26.4 简介 546
26.5 引脚说明 547
26.6 寄存器描述 548
26.6.1 MAC 配置寄存器 549
26.6.2 MAC 帧滤波寄存器 552
26.6.3 MAC Hash 表高位寄存器 553
26.6.4 MAC Hash 表低位寄存器 553
26.6.5 MAC MII 地址寄存器 554
26.6.6 MAC MII 数据寄存器 555
26.6.7 MAC 流控制寄存器 556
26.6.8 MAC VLAN 标签寄存器 557
26.6.9 MAC 调试寄存器 557
26.6.10 MAC 远程唤醒帧滤波寄存器 558
26.6.11 MAC PMT 控制和状态寄存器 559
26.6.12 MAC 中断状态寄存器 559
26.6.13 MAC 中断掩码寄存器 560
26.6.14 MAC 地址 0 高位寄存器 560
26.6.15 MAC 地址 0 低位寄存器 560
26.6.16 MAC IEEE1588 时间戳控制寄存器 561
26.6.17 亚秒递增寄存器 562
26.6.18 系统时间秒寄存器 563
26.6.19 系统时间纳秒寄存器 563
26.6.20 系统时间秒更新寄存器 563
26.6.21 系统时间纳秒更新寄存器 564
26.6.22 时间戳加数寄存器 564
26.6.23 目标时间秒寄存器 564
26.6.24 目标时间纳秒寄存器 565
26.6.25 系统时间高字秒寄存器 565
26.6.26 时间戳状态寄存器 565
26.6.27 PPS 控制寄存器 566
26.6.28 辅助时间戳纳秒寄存器 566
26.6.29 辅助时间戳秒寄存器 566
26.6.30 DMA 总线模式寄存器 567
26.6.31 DMA 发送轮询请求寄存器 569
26.6.32 DMA 接收轮询请求寄存器 569
26.6.33 DMA 接收描述符列表地址寄存器 569
26.6.34 DMA 发送描述符列表地址寄存器 570
26.6.35 DMA 状态寄存器 570
26.6.36 DMA 操作模式寄存器 572
26.6.37 DMA 中断使能寄存器 574
26.6.38 DMA 遗漏帧和缓冲区溢出计数寄存器 576
26.6.39 DMA 接收中断看门狗定时器寄存器 576
26.6.40 DMA 当前主机发送描述符寄存器 577
26.6.41 DMA 当前主机接收描述符寄存器 577
26.6.42 DMA 当前主机发送缓冲区地址寄存器 577
26.6.43 DMA 当前主机接收缓冲区地址寄存器 577
26.7 功能说明 578
26.7.1 电源管理块 578
26.7.1.1 远程唤醒帧寄存器 578
滤波器 i 字节掩码 578
滤波器 i 命令 579
滤波器 i 偏移量 579
滤波器 i CRC-16 579
26.7.1.2 远程唤醒探测 579
26.7.1.3 魔法包检测 579
26.7.1.4 掉电期间系统注意事项 580
26.7.2 DMA 仲裁程序功能 580
26.7.3 IPC 接收校验和卸载引擎 581
26.8 IEEE 1588-2002 时间戳 582
26.8.1 基准时间源 583
26.8.2 系统时间寄存器模块 583
26.8.3 发送路径功能 585
26.8.4 接收路径功能 585
26.8.5 时间戳误差容限 586
26.8.6 基准时间时钟的频率范围 586
26.9 IEEE 1588-2008 高级时间戳 586
26.9.1 点对点 PTP 透明时钟 (P2P TC) 消息支持 . . . 587
26.9.2 时钟类型 588

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

26.9.2.1 普通时钟 588
26.9.2.2 边界时钟 588
26.9.2.3 端对端透明时钟 588
26.9.2.4 PTP 处理和控制 589
26.9.2.4.1 IPv4 上的 PTP 帧 590
26.9.2.4.2 IPv6 上的 PTP 帧 591
26.9.2.4.3 以太网上的 PTP 帧 592
26.9.3 基准时间源 592
26.9.3.1 48 位秒字段 592
26.9.3.2 固定每秒脉冲输出 593
26.9.3.3 灵活的每秒脉冲输出 593
26.9.3.4 PPS 起始和停止时间 593
26.9.3.5 PPS 宽度和间隔 594
26.9.3.6 带外部事件的辅助快照 594
26.9.4 发送路径功能 594
26.9.5 接收路径功能 594
26.9.6 辅助快照 595
26.10 DMA 控制器说明 595
26.10.1 初始化 596
26.10.1.1 主机总线连发访问 597
26.10.1.2 主机数据缓冲区对齐 597
示例：缓冲区读操作 597
示例：缓冲区写操作 597
26.10.1.3 缓冲区大小计算 597
26.10.1.4 DMA 仲裁程序 598
26.10.2 传输 598
26.10.2.1 TxDMA 操作：默认（非 OSF）模式 598
26.10.2.2 TxDMA 操作：OSF 模式 600
26.10.2.3 发送帧处理 602
26.10.2.4 发送轮询暂停 602
26.10.2.5 接收 603
26.10.2.6 接收描述符获得 605
26.10.2.7 接收帧处理 605
26.10.2.8 接收进程挂起 605
26.10.2.9 中断 605
26.10.2.10 DMA 错误响应 606
26.10.3 以太网描述符 606
26.10.3.1 发送描述符 607
26.10.3.2 接收描述符 612

第 27 章：LPC43xx LCD

27.1 本章导读 618
27.2 基本配置 618
27.3 功能要点 618
27.4 简介 619
27.4.1 可编程参数 619
27.4.2 硬件游标支持 619
27.4.3 所支持的 LCD 面板 620
27.4.3.1 TFT 面板 620
27.4.3.2 彩色 STN 面板 620
27.4.3.3 单色 STN 面板 620
27.5 引脚说明 621
27.5.1 信号使用 621
27.5.1.1 单面板 STN 显示器所使用的信号 621
27.5.1.2 双面板 STN 显示器所使用的信号 622
27.5.1.3 TFT 显示器所使用的信号 622
27.6 寄存器描述 623
27.6.1 水平定时寄存器 624
27.6.1.1 水平定时限制 624
27.6.2 垂直定时寄存器 625
27.6.3 时钟和信号极性寄存器 626
27.6.4 行结束控制寄存器 627
27.6.5 上面板帧基址寄存器 627
27.6.6 下面板帧基址寄存器 628
27.6.7 LCD 控制寄存器 628
27.6.8 中断掩码寄存器 630
27.6.9 原始中断状态寄存器 631
27.6.10 掩码中断状态寄存器 631
27.6.11 中断清除寄存器 632
27.6.12 上面板当前地址寄存器 632
27.6.13 下面板当前地址寄存器 632
27.6.14 颜色调色板寄存器 632
27.6.15 游标图像寄存器 633
27.6.16 游标控制寄存器 634
27.6.17 游标配置寄存器 634
27.6.18 游标调色板寄存器 0 635
27.6.19 游标调色板寄存器 1 635
27.6.20 游标 XY 位置寄存器 635
27.6.21 游标剪切位置寄存器 636
27.6.22 游标中断掩码寄存器 636
27.6.23 游标中断清除寄存器 637
27.6.24 游标原始中断状态寄存器 637
27.6.25 游标掩码中断状态寄存器 637
27.7 LCD 控制器功能描述 638
27.7.1 AHB 接口 639
27.7.1.1 AMBA AHB 从机接口 639
27.7.1.2 AMBA AHB 主机接口 639
27.7.2 双 DMA FIFO 和相关的控制逻辑 640
27.7.3 像素串行器 640
27.7.4 RAM 调色板 645
27.7.5 硬件游标 646
27.7.5.1 游标操作 646
27.7.5.2 游标大小 647
27.7.5.3 游标移动 647
27.7.5.4 游标 XY 定位 647
27.7.5.5 游标剪切 648
27.7.5.6 游标图像格式 649
27.7.6 灰阶器 651
27.7.7 上面板和下面板格式器。 651
27.7.8 面板时钟生成器 652
27.7.9 定时控制器 652
27.7.10 STN 和 TFT 数据选择 652
27.7.10.1 STN 显示器 652
27.7.10.2 TFT 显示器 652
27.7.11 中断生成 652
27.7.11.1 主机总线错误中断 652
27.7.11.2 垂直比较中断 653
27.7.11.2.1 下一基址更新中断 653
27.7.11.2.2 FIFO 下溢中断 653
27.7.12 LCD 加电和掉电序列 653

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

27.8 LCD 时序图 655
27.9 LCD 面板信号的使用 658

第 28 章：LPC43xx 状态可配置定时器 (SCT)

28.1 本章导读 661
28.2 基本配置 661
28.3 功能要点 661
28.4 简介 662
28.5 引脚说明 663
28.6 寄存器描述 665
28.6.1 SCT 配置寄存器 669
28.6.2 SCT 控制寄存器 670
28.6.3 SCT 限值寄存器 671
28.6.4 SCT 终止条件寄存器 671
28.6.5 SCT 停止条件寄存器 671
28.6.6 SCT 启动条件寄存器 672
28.6.7 SCT 计数器寄存器 672
28.6.8 SCT 状态寄存器 672
28.6.9 SCT 输入寄存器 673
28.6.10 SCT 匹配 / 捕获寄存器模式寄存器 674
28.6.11 SCT 输出寄存器 674
28.6.12 SCT 双向输出控制寄存器 675
28.6.13 SCT 解决冲突寄存器 677
28.6.14 SCT DMA 请求 0 和请求 1 寄存器 679
28.6.15 SCT 标志使能寄存器 680
28.6.16 SCT 事件标志寄存器 680
28.6.17 SCT 冲突使能寄存器 680
28.6.18 SCT 冲突标志寄存器 680
28.6.19 SCT 匹配寄存器 0 至 15（REGMODEn 位 =0） 681
28.6.20 SCT 捕获寄存器 0 至 15（REGMODEn 位 = 1） 681
28.6.21 SCT 匹配重新载入寄存器 0 至 15（REGMODEn 位 = 0） 682
28.6.22 SCT 捕获控制寄存器 0 至 15（REGMODEn 位 = 1） 682
28.6.23 SCT 事件状态掩码寄存器 0 至 15 682
28.6.24 SCT 事件控制寄存器 0 至 15 683
28.6.25 SCT 输出设置寄存器 0 至 15 684
28.6.26 SCT 输出清除寄存器 0 至 15 684
28.7 功能说明 685
28.7.1 匹配逻辑 685
28.7.2 捕获逻辑 685
28.7.3 事件选择 685
28.7.4 输出生成 686
28.7.5 中断生成 686
28.7.6 清除预分频器 687
28.7.7 匹配事件与 I/O 事件 687
28.7.8 DMA 操作 687
28.7.9 匹配 / 捕获寄存器的替换地址 688
28.7.10 SCT 操作 689
28.7.10.1 配置 SCT 689
28.7.10.1.1 配置计数器 689
28.7.10.1.2 配置匹配寄存器和捕获寄存器 689
28.7.10.1.3 配置事件和事件响应 689
28.7.10.1.4 配置多个状态 690
28.7.10.1.5 其他选项 690
28.7.10.2 操作 SCT 691
28.7.10.3 在不使用状态的情况下配置 SCT 691
28.7.10.4 示例 692

第 29 章：LPC43xx 定时器 0/1/2/3

29.1 本章导读 694
29.2 基本配置 694
29.3 功能要点 694
29.4 简介 695
29.5 引脚说明 695
29.6 DMA 连接 698
29.7 寄存器描述 698
29.7.1 定时器中断寄存器 699
29.7.2 定时器控制寄存器 700
29.7.3 定时器计数器寄存器 700
29.7.4 定时器预分频寄存器 700
29.7.5 定时器预分频计数器寄存器 700
29.7.6 定时器匹配控制寄存器 701
29.7.7 定时器匹配寄存器 (MR0 - MR3) 702
29.7.8 定时器捕获控制寄存器 702
29.7.9 定时器捕获寄存器 (CR0 - CR3) 704
29.7.10 定时器外部匹配寄存器 704
29.7.11 定时器计数控制寄存器 705
29.7.12 DMA 操作 706
29.8 定时器操作示例 707
29.9 架构 708

第 30 章：LPC43xx 电动控制 PWM(MOTOCONPWM)

30.1 本章导读 709
30.2 基本配置 709
30.3 简介 709
30.4 功能要点 709
30.5 简介 709
30.5.1 框图 711
30.6 引脚说明 712
30.7 寄存器描述 712
30.7.1 MCPWM 控制寄存器 713
30.7.1.1 MCPWM 控制读取地址 713
30.7.1.2 MCPWM 控制设置地址 715
30.7.1.3 MCPWM 控制清除地址 715
30.7.2 PWM 捕获控制寄存器 716
30.7.2.1 MCPWM 捕获控制读取地址 716
30.7.2.2 MCPWM 捕获控制设置地址 717
30.7.2.3 MCPWM 捕获控制清除地址 718
30.7.3 MCPWM 定时器 / 计数器 0-2 寄存器 718

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

30.7.4 MCPWM 限值 0-2 寄存器 719
30.7.5 MCPWM 匹配 0-2 寄存器 719
30.7.5.1 边缘对齐模式中的匹配寄存器 719
30.7.5.2 中心对齐模式中的匹配寄存器 720
30.7.5.3 0 和 100% 的占空比 720
30.7.6 MCPWM 死区寄存器 720
30.7.7 MCPWM 通信模式寄存器 720
30.7.8 MCPWM 捕获读取地址 721
30.7.9 MCPWM 中断寄存器 721
30.7.9.1 MCPWM 中断使能读取地址 722
30.7.9.2 MCPWM 中断使能设置地址 723
30.7.9.3 MCPWM 中断使能清除地址 723
30.7.10 MCPWM 计数控制寄存器 724
30.7.10.1 MCPWM 计数控制读取地址 724
30.7.10.2 MCPWM 计数控制设置地址 726
30.7.10.3 MCPWM 计数控制清除地址 726
30.7.11 MCPWM 中断标志寄存器 727
30.7.11.1 MCPWM 中断标志读取地址 727
30.7.11.2 MCPWM 中断标志设置地址 728
30.7.11.3 MCPWM 中断标志清除地址 729
30.7.12 MCPWM 捕获清除地址 729
30.8 功能说明 730
30.8.1 脉冲宽度调制 730
无死区的边缘对齐 PWM 730
无死区的中心对齐 PWM 730
死区计数器 731
30.8.2 影子寄存器和同步更新 732
30.8.3 紧急停机 (ABORT) 732
30.8.4 捕获事件 732
30.8.5 外部事件计数 （计数器模式） 733
30.8.6 三相直流模式 733
30.8.7 三相交流模式 734
30.8.8 中断 735

第 31 章：LPC43xx 正交编码器接口 (QEI)

31.1 本章导读 736
31.2 基本配置 736
31.3 功能要点 736
31.4 简介 736
31.5 引脚说明 738
31.6 寄存器描述 738
31.6.1 控制寄存器 739
31.6.1.1 QEI 控制寄存器 739
31.6.1.2 QEI 状态寄存器 739
31.6.1.3 QEI 配置寄存器 740
31.6.2 位置、索引和定时器寄存器 740
31.6.2.1 QEI 位置寄存器 740
31.6.2.2 QEI 最大位置寄存器 740
31.6.2.3 QEI 位置比较寄存器 0 740
31.6.2.4 QEI 位置比较寄存器 1 741
31.6.2.5 QEI 位置比较寄存器 2 741
31.6.2.6 QEI 索引计数寄存器 741
31.6.2.7 QEI 索引比较寄存器 0 741
31.6.2.8 QEI 定时器重新载入寄存器 741
31.6.2.9 QEI 定时器寄存器 741
31.6.2.10 QEI 速度寄存器 742
31.6.2.11 QEI 速度捕获寄存器 742
31.6.2.12 QEI 速度比较寄存器 742
31.6.2.13 相位 A 输入寄存器上的 QEI 数字过滤器 742
31.6.2.14 相位 B 输入寄存器上的 QEI 数字过滤器 742
31.6.2.15 索引输入寄存器上的 QEI 数字过滤器 742
31.6.2.16 QEI 索引接受窗口寄存器 743
31.6.2.17 QEI 索引比较寄存器 1 743
31.6.2.18 QEI 索引比较寄存器 2 743
31.6.3 中断寄存器 743
31.6.3.1 QEI 中断使能清除寄存器 743
31.6.3.2 QEI 中断使能设置寄存器 744
31.6.3.3 QEI 中断状态寄存器 744
31.6.3.4 QEI 中断使能寄存器 745
31.6.3.5 QEI 中断状态清除寄存器 746
31.6.3.6 QEI 中断状态设置寄存器 747
31.7 功能说明 748
31.7.1 输入信号 748
31.7.1.1 正交输入信号 748
31.7.1.2 数字输入过滤 749
31.7.2 位置捕获 749
31.7.3 速度捕获 750
31.7.4 速度比较 750

第 32 章：LPC43xx 重复性中断定时器 (RIT)

32.1 本章导读 751
32.2 基本配置 751
32.3 功能要点 751
32.4 简介 751
32.5 寄存器描述 752
32.5.1 RI 比较值寄存器 752
32.5.2 RI 掩码寄存器 753
32.5.3 RI 控制寄存器 753
32.5.4 RI 计数器寄存器 753
32.6 RI 定时器操作 754

第 33 章：LPC43xx 报警定时器

33.1 本章导读 755
33.2 基本配置 755
33.3 简介 755
33.4 寄存器描述 756
33.4.1 向下计数器寄存器 756
33.4.2 预设值寄存器 756
33.4.3 中断清除使能寄存器 756
33.4.4 中断设置使能寄存器 757
33.4.5 中断状态寄存器 757

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

33.4.6 中断使能寄存器 757
33.4.7 清除状态寄存器 757
33.4.8 设置状态寄存器 757

第 34 章：LPC43xx 窗口化看门狗定时器 (WWDT)

34.1 本章导读 758
34.2 基本配置 758
34.3 功能要点 758
34.4 应用 759
34.5 描述 759
34.5.1 调试模式中的 WWDT 特性 759
34.5.2 掉电模式中的 WWDT 特性 759
34.6 时钟 760
34.7 寄存器描述 760
34.7.1 看门狗模式寄存器 760
34.7.2 看门狗定时器常数寄存器 762
34.7.3 看门狗输入寄存器 762
34.7.4 看门狗定时器值寄存器 763
34.7.5 看门狗定时器报警中断寄存器 763
34.7.6 看门狗定时器窗口寄存器 763
34.8 功能框图 764
34.9 看门狗定时示例 765

第 35 章：LPC43xx 实时时钟 (RTC)

35.1 本章导读 766
35.2 基本配置 766
35.3 功能要点 766
35.4 简介 767
35.5 引脚说明 767
35.6 寄存器描述 768
35.6.1 中断位置寄存器 769
35.6.2 时钟控制寄存器 769
35.6.3 计数器递增中断寄存器 770
35.6.4 报警掩码寄存器 770
35.6.5 完整时间寄存器 771
35.6.5.1 完整时间寄存器 0 771
35.6.5.2 完整时间寄存器 1 771
35.6.5.3 完整时间寄存器 2 772
35.6.6 时间计数器组 772
35.6.6.1 闰年计算 774
35.6.6.2 校准寄存器 774
35.6.7 报警寄存器组 775
35.7 功能说明 777
35.7.1 校准步骤 777
向后校准 777
向前校准 777

第 36 章：LPC43xx USART0_2_3

36.1 本章导读 778
36.2 基本配置 778
36.3 功能要点 778
36.4 简介 779
36.5 引脚说明 781
36.6 寄存器描述 782
36.6.1 USART 接收缓冲寄存器 783
36.6.2 USART 发送器保持寄存器 783
36.6.3 USART 除数锁存 LSB 和 MSB 寄存器 783
36.6.4 USART 中断使能寄存器 784
36.6.5 USART 中断识别寄存器 785
36.6.6 USART FIFO 控制寄存器 787
36.6.6.1 DMA 操作 788
USART 接收器 DMA 788
USART 发送器 DMA 788
36.6.7 USART 线路控制寄存器 788
36.6.8 USART 线路状态寄存器 789
36.6.9 USART 暂存寄存器 790
36.6.10 USART 自动波特率控制寄存器 790
36.6.10.1 自动波特率 791
36.6.10.2 自动波特率模式 792
36.6.11 IrDA 控制寄存器 (USART3) 793
36.6.12 USART 小数分频器寄存器 (U0FDR - 0x4000 8028) 794
36.6.12.1 波特率计算 795
36.6.12.1.1 示例 1：USART_PCLK = 14.7456 MHz、 BR = 9600 797
36.6.12.1.2 示例 2：USART_PCLK = 12 MHz、 BR = 115200 797
36.6.13 USART 过采样寄存器 797
36.6.14 USART 半双工使能寄存器 798
36.6.15 USART 智能卡接口控制寄存器 799
36.6.16 USART RS485 控制寄存器 800
36.6.17 USART RS485 地址匹配寄存器 801
36.6.18 USART1 RS485 延迟值寄存器 801
36.6.19 USART 同步模式控制寄存器 801
36.6.20 USART 发送使能寄存器 803
36.7 功能说明 803
36.7.1 同步模式 803
36.7.1.1 同步从机模式 803
接收 803
传输 804
36.7.1.2 同步主机模式 805
36.7.2 RS-485/EIA-485 模式的操作 805
RS-485/EIA-485 正常多点模式 (NMM) 805
RS-485/EIA-485 自动地址检测 (AAD) 模式 . . 805
RS-485/EIA-485 自动方向控制 806
RS485/EIA-485 驱动器延迟时间 806
RS485/EIA-485 输出反转 806
36.7.3 智能卡模式 806
36.7.3.1 智能卡设置步骤 807

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

第 37 章：LPC43xx UART1

37.1 本章导读 808
37.2 基本配置 808
37.3 功能要点 808
37.4 引脚说明 809
37.5 寄存器描述 810
37.5.1 UART1 接收器缓冲寄存器（当 DLAB = 0 时）. 811
37.5.2 UART1 发送器保持寄存器（当 DLAB = 0 时）. 811
37.5.3 UART1 除数锁存器 LSB 和 MSB 寄存器（当 DLAB = 1 时）. 811
37.5.4 UART1 中断使能寄存器（当 DLAB = 0 时）. 812
37.5.5 UART1 中断识别寄存器 813
37.5.6 UART1 FIFO 控制寄存器 815
37.5.6.1 DMA 操作 816
UART 接收器 DMA 816
UART 发送器 DMA 816
37.5.7 UART1 线路控制寄存器 816
37.5.8 UART1 调制解调器控制寄存器 817
37.5.9 自动流控制 817
37.5.9.1 自动 RTS 817
37.5.9.2 自动 CTS 818
37.5.10 UART1 线路状态寄存器 819
37.5.11 UART1 调制解调器状态寄存器 820
37.5.12 UART1 暂存寄存器 821
37.5.13 UART1 自动波特率控制寄存器 821
37.5.14 自动波特率 822
37.5.15 自动波特率模式 823
37.5.16 UART1 小数分频器寄存器 824
37.5.16.1 波特率计算 825
37.5.16.1.1 示例 1：PCLK = 14.7456 MHz、 BR = 9600 827
37.5.16.1.2 示例 2：PCLK = 12 MHz、 BR = 115200 . . 827
37.5.17 UART1 发送使能寄存器 827
37.5.18 UART1 RS485 控制寄存器 828
37.5.19 UART1 RS-485 地址匹配寄存器 829
37.5.20 UART1 RS-485 延迟值寄存器 829
37.5.21 RS-485/EIA-485 模式的操作 829
RS-485/EIA-485 正常多点模式 (NMM) 829
RS-485/EIA-485 自动地址检测 (AAD) 模式 . . 829
RS-485/EIA-485 自动方向控制 830
RS485/EIA-485 驱动器延迟时间 830
RS485/EIA-485 输出反转 830
37.5.22 UART1 FIFO 电平寄存器 830
37.6 架构 830

第 38 章：LPC43xx SSP0/1

38.1 本章导读 833
38.2 基本配置 833
38.3 功能要点 833
38.4 简介 833
38.5 引脚说明 834
38.6 寄存器描述 834
38.6.1 SSP 控制寄存器 0 835
38.6.2 SSP 控制寄存器 1 836
38.6.3 SSP 数据寄存器 837
38.6.4 SSP 状态寄存器 838
38.6.5 SSP 时钟预分频寄存器 838
38.6.6 SSP 中断掩码设置 / 清除寄存器 838
38.6.7 SSP 原始中断状态寄存器 839
38.6.8 SSP 掩码中断状态寄存器 839
38.6.9 SSP 中断清除寄存器 840
38.6.10 SSP DMA 控制寄存器 840
38.7 功能说明 841
38.7.1 德州仪器同步串行帧格式 841
38.7.2 SPI 帧格式 842
38.7.2.1 时钟极性 (CPOL) 及相位 (CPHA) 控制 842
38.7.2.2 CPOL=0， CPHA=0 时的 SPI 格式 842
38.7.2.3 CPOL=0， CPHA=1 时的 SPI 格式 843
38.7.2.4 CPOL = 1 且 CPHA = 0 时的 SPI 格式 844
38.7.2.5 CPOL = 1 且 CPHA = 1 时的 SPI 格式 845
38.7.3 国家半导体 Microwire 帧格式 846
38.7.3.1 Microwire 模式下 CS 相对于 SK 的建立和保持时间要求 847

第 39 章：LPC43xx SPI

39.1 本章导读 848
39.2 基本配置 848
39.3 功能要点 848
39.4 简介 848
39.5 引脚说明 850
39.6 寄存器描述 850
39.6.1 SPI 控制寄存器 851
39.6.2 SPI 状态寄存器 852
39.6.3 SPI 数据寄存器 852
39.6.4 SPI 时钟计数器寄存器 852
39.6.5 SPI 测试控制寄存器 853
39.6.6 SPI 测试状态寄存器 853
39.6.7 SPI 中断寄存器 853
39.7 功能说明 853
39.7.1 SPI 数据传送 853
39.7.2 一般信息 855
39.7.3 主机操作 855
39.7.4 从机操作 856
39.7.5 异常状况 856

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

第 40 章：LPC43xx I2S 接口

40.1 本章导读 857
40.2 基本配置 857
40.3 功能要点 858
40.4 简介 859
40.4.1 I2S 连接方案 859
40.4.2 GIMA 的 I2S 连接 859
40.5 引脚描述 860
40.6 寄存器描述 861
40.6.1 I2S 数字音频输出寄存器 862
40.6.2 I2S 数字音频输入寄存器 862
40.6.3 I2S 发送 FIFO 寄存器 863
40.6.4 接收 FIFO 寄存器 863
40.6.5 I2S 状态反馈寄存器 863
40.6.6 I2S DMA 配置寄存器 1 864
40.6.7 I2S DMA 配置寄存器 2 864
40.6.8 I2S 中断请求控制寄存器 865
40.6.9 I2S 发送时钟速率寄存器 865
40.6.9.1 变速率发生器的说明 865
40.6.10 I2S 接收时钟速率寄存器 866
40.6.11 I2S 发送时钟位速率寄存器 866
40.6.12 I2S 接收时钟位速率寄存器 866
40.6.13 I2S 发送模式控制寄存器 867
40.6.14 I2S 接收模式控制寄存器 867
40.7 功能描述 868
40.7.1 I^2S 的发送和接收接口 868
40.7.2 I^2S 操作模式 869
40.7.3 FIFO 控制器 874

第 41 章：LPC43xx C_CAN

41.1 本章导读 876
41.2 基本配置 876
41.3 功能要点 876
41.4 简介 877
41.5 引脚描述 877
41.6 寄存器描述 878
复位时的寄存器值 878
读取 / 写入操作的时序 878
41.6.1 CAN 协议寄存器 881
41.6.1.1 CAN 控制寄存器 881
41.6.1.2 CAN 状态寄存器 882
41.6.1.3 CAN 错误计数器 883
41.6.1.4 CAN 位时序寄存器 883
41.6.1.5 CAN 中断寄存器 884
41.6.1.6 CAN 测试寄存器 884
41.6.1.7 CAN 波特率预分频器扩展寄存器 885
41.6.2 消息接口寄存器 886
41.6.2.1 消息对象 887
41.6.2.2 CAN 消息接口命令请求寄存器 887
41.6.2.3 CAN 消息接口指令掩码寄存器 889
写传输方向 889
读传输方向 891
41.6.2.4 IF1 和 IF2 消息缓冲寄存器 893
41.6.2.4.1 CAN 消息接口命令掩码 1 寄存器 893
41.6.2.4.2 CAN 消息接口命令掩码 2 寄存器 893
41.6.2.4.3 CAN 消息接口命令仲裁 1 寄存器 894
41.6.2.4.4 CAN 消息接口命令仲裁 2 寄存器 895
41.6.2.4.5 CAN 消息接口消息控制寄存器 896
41.6.2.4.6 CAN 消息接口数据 A1 寄存器 898
41.6.2.4.7 CAN 消息接口数据 A2 寄存器 899
41.6.2.4.8 CAN 消息接口数据 B1 寄存器 899
41.6.2.4.9 CAN 消息接口数据 B2 寄存器 899
41.6.3 消息处理器寄存器 900
41.6.3.1 CAN 发送请求 1 寄存器 900
41.6.3.2 CAN 发送请求 2 寄存器 900
41.6.3.3 CAN 新数据 1 寄存器 900
41.6.3.4 CAN 新数据 2 寄存器 901
41.6.3.5 CAN 中断挂起 1 寄存器 901
41.6.3.6 CAN 中断挂起 2 寄存器 901
41.6.3.7 CAN 消息有效 1 寄存器 902
41.6.3.8 CAN 消息有效 2 寄存器 902
41.6.4 CAN 定时寄存器 902
41.6.4.1 CAN 时钟分频器寄存器 902
41.7 功能描述 903
41.7.1 复位后 C_CAN 控制器状态 903
41.7.2 C_CAN 操作模式 903
41.7.2.1 软件初始化 903
41.7.2.2 CAN 消息传输 903
41.7.2.3 禁用自动重传 (DAR) 904
41.7.2.4 测试模式 904
41.7.2.4.1 静默模式 904
41.7.2.4.2 环回模式 905
41.7.2.4.3 结合静默模式的环回模式 905
41.7.2.4.4 基本模式 906
41.7.2.4.5 引脚 CAN_TXD 的软件控制 906
41.7.3 CAN 消息处理器 907
41.7.3.1 管理消息对象 908
41.7.3.2 IFx 寄存器和消息 RAM 之间的数据传输 908
41.7.3.3 在 CAN 内核中的移位寄存器和消息缓冲区之间发送消息 908
41.7.3.4 收到消息的验收过滤 909
41.7.3.4.1 接收数据帧 909
41.7.3.4.2 接收远程帧 909
41.7.3.5 接收 / 发送优先级 909
41.7.3.6 配置发送对象 910
41.7.3.7 更新发送对象 910
41.7.3.8 配置接收对象 910
41.7.3.9 处理收到的消息 911
41.7.3.10 FIFO 缓冲区配置 911
41.7.3.10.1 使用 FIFO 缓冲区接收消息 911
41.7.3.10.2 从 FIFO 缓冲区读取 911
41.7.4 中断处理 913
41.7.5 位时序 913
41.7.5.1 位时序和比特率 913

本文件所提供的全部信息均遵循法律声明的规定。

© NXP B.V. 2012. 保留所有权利。

第 42 章：LPC43xx I2C 总线接口

42.1 本章导读 915
42.2 基本配置 915
42.3 功能要点 915
42.4 应用 915
42.5 简介 916
42.5.1 I²C 超快速模式 916
42.6 引脚说明 917
42.7 寄存器描述 917
42.7.1 I²C 控制设置寄存器 919
42.7.2 I²C 状态寄存器 920
42.7.3 I²C 数据寄存器 921
42.7.4 I²C 从机地址寄存器 0 921
42.7.5 I²C SCL 高、低占空比寄存器 921
42.7.5.1 选择适当的 I²C 数据速率和占空比 922
42.7.6 I²C 控制清除寄存器 922
42.7.7 I²C 监控模式控制寄存器 923
42.7.7.1 监控模式的中断 924
42.7.7.2 监控模式中的仲裁丢失 924
42.7.8 I²C 从机地址寄存器 924
42.7.9 I²C 数据缓冲寄存器 924
42.7.10 I²C 掩码寄存器 925
42.8 I²C 操作模式 925
42.8.1 主发送器模式 925
42.8.2 主接收器模式 926
42.8.3 从属接收器模式 927
42.8.4 从属发送器模式 928
42.9 I²C 执行和操作 929
42.9.1 输入滤波器与输出级 930
42.9.2 地址寄存器 ADR0 ～ ADR3 930
42.9.3 地址掩码寄存器， MASK0 ～ MASK3 930
42.9.4 比较器 930
42.9.5 移位寄存器， DAT 930
42.9.6 仲裁与同步逻辑 930
42.9.7 串行时钟生成器 931
42.9.8 时序与控制 931
42.9.9 控制寄存器 ICONSET 和 CONCLR 932
42.9.10 状态解码器与状态寄存器 932
42.10 I²C 操作模式详解 933
42.10.1 主发送器模式 933
42.10.2 主接收器模式 936
42.10.3 从属接收器模式 938
42.10.4 从属发送器模式 941
42.10.5 其它状态 942
42.10.5.1 STAT = 0xF8 942
42.10.5.2 STAT = 0x00 942
42.10.6 某些特殊情况 943
42.10.6.1 来自两个主机的同时 “重复起动”条件 943
42.10.6.2 仲裁丢失后进行数据传输 943
42.10.6.3 强制访问 I²C 总线 944
42.10.6.4 SCL 或 SDA 低电平妨碍 I²C 总线的操作 . . . 944
42.10.6.5 总线错误 945
42.10.7 I²C 状态服务程序 945
42.10.8 初始化 945
42.10.9 I²C 中断服务 945
42.10.10 状态服务程序 945
42.10.11 配合实际应用的状态服务 945
42.11 软件示例 946
42.11.1 初始化程序 946
42.11.2 启动主机发送功能 946
42.11.3 启动主机接收功能 946
42.11.4 I²C 中断程序 946
42.11.5 无指定模式的状态 946
42.11.5.1 状态：0x00 946
42.11.5.2 主机状态 947
42.11.5.3 状态：0x08 947
42.11.5.4 状态：0x10 947
42.11.6 主发送状态 947
42.11.6.1 状态：0x18 947
42.11.6.2 状态：0x20 947
42.11.6.3 状态：0x28 948
42.11.6.4 状态：0x30 948
42.11.6.5 状态：0x38 948
42.11.7 主接收状态 948
42.11.7.1 状态：0x40 948
42.11.7.2 状态：0x48 948
42.11.7.3 状态：0x50 949
42.11.7.4 状态：0x58 949
42.11.8 从接收状态 949
42.11.8.1 状态：0x60 949
42.11.8.2 状态：0x68 949
42.11.8.3 状态：0x70 950
42.11.8.4 状态：0x78 950
42.11.8.5 状态：0x80 950
42.11.8.6 状态：0x88 950
42.11.8.7 状态：0x90 951
42.11.8.8 状态：0x98 951
42.11.8.9 状态：0xA0 951
42.11.9 从发送状态 951
42.11.9.1 状态：0xA8 951
42.11.9.2 状态：0xB0 951
42.11.9.3 状态：0xB8 952
42.11.9.4 状态：0xC0 952
42.11.9.5 状态：0xC8 952

第 43 章：LPC43xx 10 位 ADC0/1

43.1 本章导读 953
43.2 基本配置 954
43.3 功能要点 954
43.4 简介 955
43.5 引脚说明 955
43.6 寄存器描述 955
43.6.1 A/D 控制寄存器 957
43.6.2 A/D 全局数据寄存器。 958
43.6.3 A/D 中断使能寄存器 958
43.6.4 A/D 数据寄存器 959
43.6.5 A/D 状态寄存器 959

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

43.7 操作 960
43.7.1 硬件启动转换 960
43.7.2 中断 960
43.7.3 DMA 控制 960

第 44 章：LPC43xx DAC

44.1 本章导读 961
44.2 基本配置 961
44.3 功能要点 961
44.4 引脚说明 961
44.5 寄存器描述 962
44.5.1 D/A 转换器寄存器 962
44.5.2 D/A 转换器控制寄存器 962
44.5.3 D/A 转换器计数器值寄存器 963
44.6 功能说明 963
44.6.1 DMA 计数器 963
44.6.2 双缓冲 963

第 45 章：LPC43xx 闪存编程 /ISP 和 IAP

45.1 本章导读 965
45.2 简介 965
45.3 功能要点 965
45.4 描述 965
45.4.1 复位后的存储器映射 966
45.4.1.1 有效用户代码的判定标准 966
45.4.2 通信协议 966
45.4.2.1 ISP 命令格式 966
45.4.2.2 ISP 响应格式 966
45.4.2.3 ISP 数据格式 966
45.4.2.4 ISP 流量控制 967
45.4.2.5 ISP 命令中止 967
45.4.2.6 IAP 过程中的中断 967
45.4.2.7 ISP 命令处理程序使用的 RAM 967
45.4.2.8 IAP 命令处理程序使用的 RAM 967
45.5 带闪存的 LPC43xx 器件的启动过程流程图 . . . 968
45.6 扇区号 969
45.7 代码读保护 (CRP) 970
45.8 ISP 命令 972
45.8.1 解锁 < 解锁代码 > 972
45.8.2 设置波特率 < 波特率 > < 停止位 > 973
45.8.3 回应 < 设定 > 973
45.8.4 写 RAM < 起始地址 > < 字节数 > 973
45.8.5 读存储器 < 地址 > < 字节数 > 974
45.8.6 准备写操作的扇区 < 起始扇区号 > < 结束扇区号 > 975
45.8.7 将 RAM 内容复制到闪存 < 闪存地址 > <RAM 地址 > < 字节数 > 975
45.8.8 运行 < 地址 > < 模式 > 976
45.8.9 擦除扇区 < 起始扇区号 > < 结束扇区号 > . . . 976
45.8.10 扇区查空 < 起始扇区号 > < 结束扇区号 > . . . 977
45.8.11 读器件标识号 977
45.8.12 读取启动代码版本号 977
45.8.13 读取设备序列号 977
45.8.14 比较 < 地址 1> < 地址 2> < 字节数 > 978
45.8.15 ISP 返回代码 978
45.9 IAP 命令 980
45.9.1 准备写操作扇区 982
45.9.2 复制 RAM 到闪存 982
45.9.3 擦除扇区 983
45.9.4 空白检查扇区 983
45.9.5 读取器件识别号 983
45.9.6 读取启动代码版本号 984
45.9.7 读取设备序列号 984
45.9.8 比较 < 地址 1> < 地址 2> < 字节数 > 984
45.9.9 重新调用 ISP 985
45.9.10 IAP 状态代码 985
45.10 JTAG 闪存编程接口 985
45.11 闪存签名生成 986
45.11.1 用于生成签名的寄存器描述 986
45.11.1.1 签名生成地址和控制寄存器 986
45.11.1.2 签名生成结果寄存器 987
45.11.1.3 闪存模块状态寄存器 (FMSTAT - 0x0x4008 4FE0) 987
45.11.1.4 闪存模块状态清除寄存器 (FMSTATCLR - 0x0x4008 4FE8) 987
45.11.2 签名生成的算法和步骤 988
签名生成 988
内容验证 988

第 46 章：LPC43xx JTAG，串行调试接口 (SWD) 和跟踪功能

46.1 本章导读 989
46.2 基本配置 989
46.3 功能要点 989
46.4 描述 989
46.5 引脚说明 990
46.6 调试连接 990
46.6.1 ARM 标准 JTAG 连接器 （20 引脚） 991
46.6.2 Cortex 调试连接器 （10 引脚） 991
46.6.3 Cortex 调试 + ETM 连接器 （20 引脚） 992
46.7 调试注意事项 992
46.8 调试存储器重新映射 993
46.9 JTAG TAP 识别 993

第 47 章：LPC43xx ARM Cortex M0/M4 参考

47.1 本章导读 994
47.2 Cortex-M4 指令集汇总 994
47.3 Cortex-M0 指令集汇总 1001

本文件所提供的全部信息均遵循法律声明的规定。
© NXP B.V. 2012. 保留所有权利。

第 48 章：补充信息

48.1 缩略词 1004
48.2 法律信息 1006
48.2.1 用户手册状态 1006
48.2.2 定义 1006
48.2.3 免责声明 1006
48.2.4 商标 1006
48.3 表 1007
48.4 图 1025
48.5 内容 1027

注意：关于本文及相关产品的重要说明详见“法律信息”一节。

© NXP B.V. 2012. 保留所有权利。

有关详细信息，请访问：http://www.nxp.com
欲咨询销售办事处地址，请发送电子邮件至：salesaddresses@nxp.com

发布日期：2012 年 2 月 1 日
文档号：UM10503