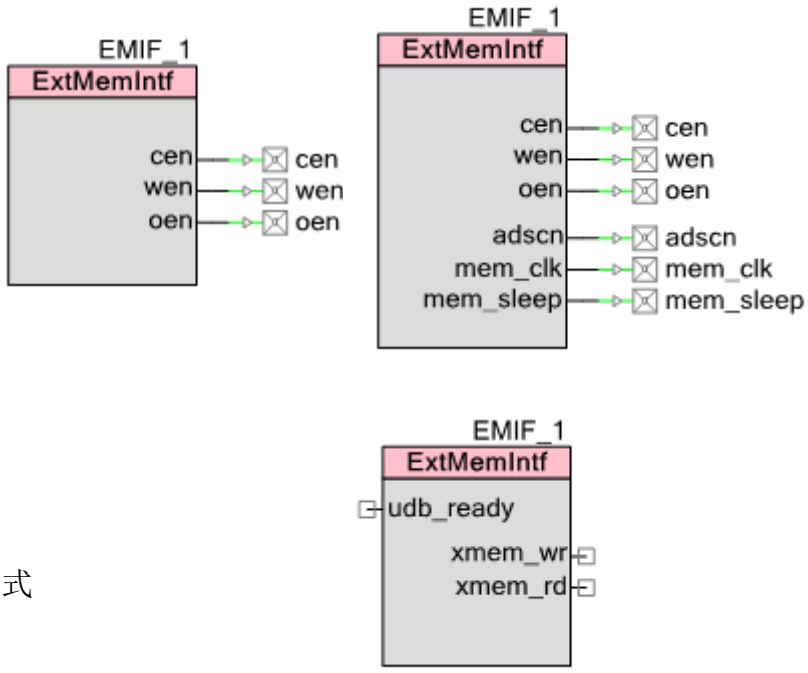
外部存储器接口 (EMIF)

1.0

特性

- 8 位、16 位、24 位地址总线宽度
- 8 位、16 位数据总线宽度
- 支持外部同步存储器
- 支持外部异步存储器
- 支持存储器自定义接口
- 支持外部存储器速度变化范围 (5-200 ns)
- 支持外部存储器断电、睡眠和唤醒模式



概述

EMIF 组件可以通过 CPU 或 DMA 访问 PSoC 3 外部存储器 IC。它简化了 EMIF 硬件及 UDB 和 GPIO 的所需设置。EMIF 可以控制同步或异步存储器，无需在同步和异步模式下配置任何 UDB。在 UDB 模式下，必须配置 UDB，用来生成外部控制信号。

使用 EMIF 时

EMIF 用于扩展可用存储器空间。这支持在片下存储器器件中扩展数据存储。外部存储器 IC 连接至 PSoC 3，通常旨在保存大量数据阵列，例如，文本、音频或视频内容。其他预期应用包括数据记录，这些应用可用作外部 LCD 像素数据的缓冲区

输入/输出连接

本节介绍 EMIF 的各种输入和输出连接。I/O 列表中的星号 (*) 表示该 I/O 是可隐藏 I/O，其隐藏条件在该 I/O 的说明中。

udb_ready — 输入*

外部逻辑信号向 PHUB 指示 AHB 总线数据操作已完成（高电平有效）。将 **External Memory Type**（外部存储器类型）参数设置为 **Custom**（自定义）时，此输入可见。

xmem_wr — 输出*

自定义接口的通用写入信号（高电平有效）。外部逻辑信号向 PHUB 指示 AHB 总线数据操作已完成（高电平有效）。将 **External Memory Type**（外部存储器类型）参数设置为 **Custom**（自定义）时，此输入可见。

xmem_rd — 输出*

自定义接口的通用读取信号（高电平有效）。将 **External Memory Type**（外部存储器类型）参数设置为 **Custom**（自定义）时，此输入可见。

cen — 输出*

外部存储器芯片使能（低电平有效）。将 **External Memory Type**（外部存储器类型）参数设置为 **Synchronous**（同步）或 **Asynchronous**（异步）时，此输出可见。

wen — 输出*

外部存储器写使能（低电平有效）。将 **External Memory Type**（外部存储器类型）参数设置为 **Synchronous**（同步）或 **Asynchronous**（异步）时，此输出可见。

oen — 输出*

外部存储器输出使能（低电平有效）。将 **External Memory Type**（外部存储器类型）参数设置为 **Synchronous**（同步）或 **Asynchronous**（异步）时，此输出可见。

mem_clk — 输出*

用于外部同步存储器的时钟。将 **External Memory Type**（外部存储器类型）参数设置为 **Synchronous**（同步）时，此输出可见。

adscn — 输出*

外部同步存储器的地址探针（低电平有效）。将 **External Memory Type**（外部存储器类型）参数设置为 **Synchronous**（同步）时，此输出可见。

mem_sleep — 输出*

外部同步存储器睡眠引脚的睡眠信号（高电平有效）。将 **External Memory Type**（外部存储器类型）参数设置为 **Synchronous**（同步）时，此输出可见。

原理图宏信息

默认情况下，PSoC Creator 组件目录包含 EMIF 组件的原理图宏实现。这些宏已经包含连接的输出引脚。原理图宏即可用于异步和同步类型的外部存储器接口。

图 1. 异步外部存储器接口原理图宏

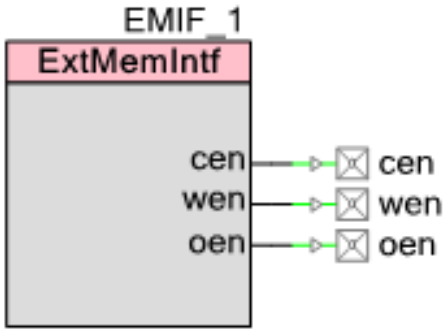
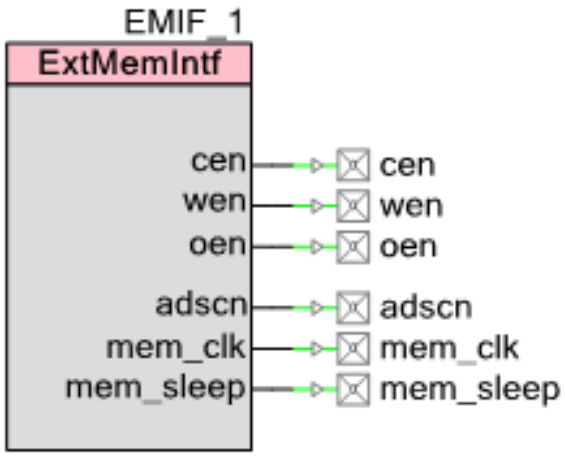


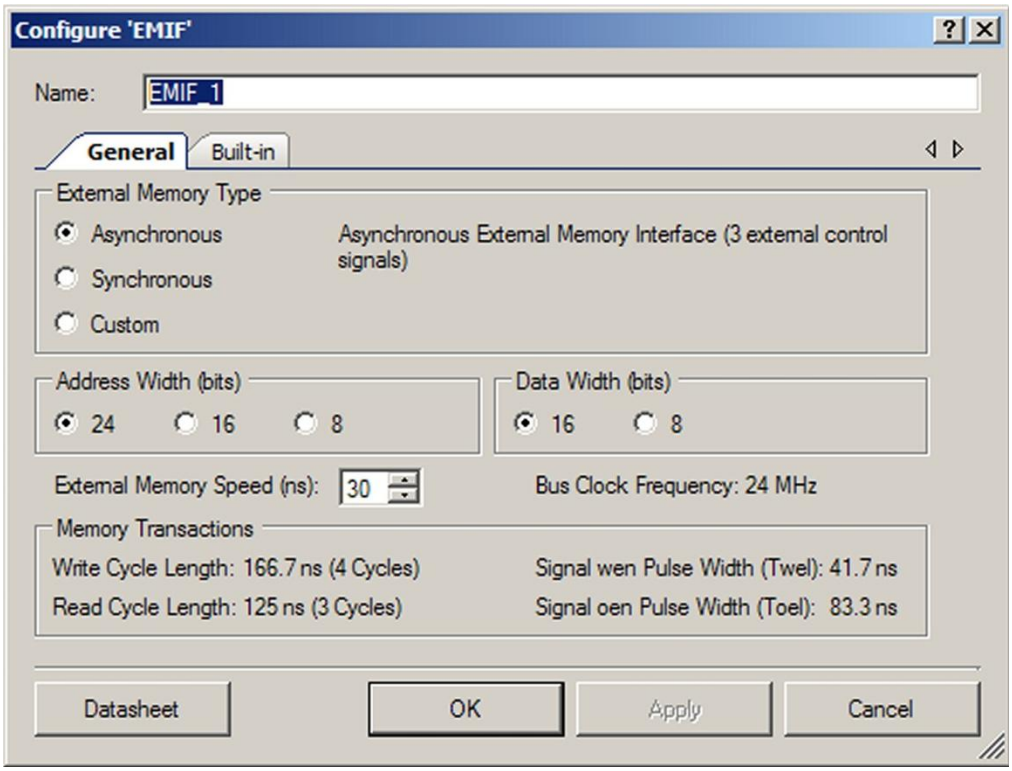
图 2. 同步外部存储器接口原理图宏



元件参数

将 EMIF 拖入设计中，双击该组件，打开 **Configure**（配置）对话框。

图 3. 配置 EMIF 对话框



EMIF 提供下列参数。

外部存储器类型

确定外部存储器类型。默认外部存储器类型是**异步**。

地址宽度（位）

确定外部存储器地址总线的位数。默认地址总线宽度为 **24 位**。

数据宽度（位）

确定外部存储器数据总线的位数。默认数据总线宽度为 **16 位**。

外部存储器速度 (ns)

确定外部存储器的速度 (ns)。默认外部存储器速度为 **30 ns**。

总线时钟频率

显示所选总线时钟频率（兆赫兹）。

存储器数据操作

- **写入周期长度 —** 表示写入周期的长度（纳秒）。总线时钟频率周期显示在括号中。
- **读取周期长度 —** 表示读取周期的长度（纳秒）。总线时钟频率周期显示在括号中。
- **信号 Wen 脉宽 (Twel) —** 表示 Twel 参数（启用信号脉宽的写入）（纳秒）。有关详细信息，请参见直流和交流电气特性这一章节。
- **信号 Oen 脉宽 (Toel) —** 表示 Toel 参数（启用信号脉宽的写入）（纳秒）。有关详细信息，请参见直流和交流电气特性这一章节。

放置

EMIF 组件放置在 EMIF 硬件模块中。

资源

外部存储器接口 (EMIF) 是一 PSoC 3 系列专用硬件模块，与 UDB 一起使用，支持与外部存储器器件连接。

模式	数字模块	API Memory（API 存储器）（字节）		Pins（引脚）（每个外部 I/O）
	EMIF	Flash（闪存）	RAM	
异步	1	204	4	数据（8-16 位）+ 地址宽度（8-24 位）+ 3
同步	1	219	4	数据（8-16 位）+ 地址宽度（8-24 位）+ 6
自定义	1	185	4	数据（8-16 位）+ 地址宽度（8-24 位）+ 用户定义的控制信号

应用程序编程接口

应用程序编程接口 (API) 子程序允许您使用软件配置组件。下表列出了每个函数的接口，并进行了说明。以下各节将更详细地介绍每个函数。

默认情况下，PSoC Creator 将实例名称“EMIF_1”分配给指定设计中组件的第一个实例。您可以将其重命名为遵循标识符语法规则的任何唯一值。实例名称会成为每个全局函数名称、变量和常量符号的前缀。出于可读性考虑，下表中使用的实例名称为“EMIF”。

函数	说明
EMIF_Start()	Calls EMIF_Init() and EMIF_Enable()。
EMIF_Stop()	禁用 EMIF 模块。返回所有相关的 I/O 端口和引脚至 HI-Z 模式
EMIF_Init()	初始化或恢复当前自定义程序状态的 EMIF 配置
EMIF_Enable()	启用 EMIF 硬件模块、相关 I/O 端口和引脚
EMIF_ExtMemSleep()	设置外部存储器睡眠信号为高电平；注意：根据所使用的外部存储器 IC 的类型，可能需要反转该信号
EMIF_ExtMemWakeup()	设置外部存储器睡眠信号为低电平；注意：根据所使用的外部存储器 IC 的类型，可能需要反转该信号
EMIF_SaveConfig()	保存 EMIF 非保留寄存器的用户配置。通过 EMIF_Sleep() 调用此子程序，以便在进入睡眠模式前保存组件配置
EMIF_Sleep()	停止 EMIF 操作，保存用户配置与 EMIF 的启用状态
EMIF_RestoreConfig()	恢复 EMIF 非保留寄存器的用户配置。通过 EMIF_Wakeup() 调用此子程序，以便在退出睡眠模式时恢复组件配置
EMIF_Wakeup()	恢复用户配置和启用状态

void EMIF_Start(void)

说明： 这是开始执行组件操作的首选方法。EMIF_Start() 调用 EMIF_Init() 函数，然后再调用 EMIF_Enable() 函数。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

void EMIF_Stop(void)

说明： 禁用 EMIF 模块。返回所有相关的 I/O 端口和引脚至 HI-Z 模式

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

void EMIF_Init(void)

说明： 根据自定义程序“配置”对话框设置来初始化或恢复组件。无需调用 EMIF_Init()，因为 EMIF_Start() 子程序会调用此函数，这是开始组件操作的首选方法。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

void EMIF_Enable(void)

说明： 激活硬件并开始执行组件操作。无需调用 EMIF_Enable()，因为 EMIF_Start() 子程序会调用此函数，这是开始组件操作的首选方法。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

void EMIF_ExtMemSleep(void)

说明： 在 EMIF_PWR_DWN 寄存器中设置‘mem_pd’位。这将外部存储器睡眠信号设置为高电平。根据所使用的外部存储器 IC 的类型，可能需要反转该信号。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

void EMIF_ExtMemWakeup(void)

说明： 在 EMIF_PWR_DWN 寄存器中复位‘mem_pd’位。这将外部存储器睡眠信号设置为低电平。根据所使用的外部存储器 IC 的类型，可能需要反转该信号

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

void EMIF_SaveConfig(void)

说明： 此函数保存组件配置。它将保存非保存寄存器。此函数还将保存当前“配置”对话框中定义的或通过相应 API 修改的组件参数值。通过 EMIF_Sleep() 函数调用此函数。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

void EMIF_Sleep(void)

说明： 这是准备组件睡眠的首选子程序。EMIF_Sleep() 子程序保存当前组件状态。然后调用 EMIF_Stop() 函数和 EMIF_SaveConfig() 以保存硬件配置。

调用 EMIF_Sleep() 函数，然后再调用 CyPmSleep() 或 CyPmHibernate() 函数。有关电源管理函数的更多信息，请参考 PSoC Creator *System Reference Guide*（《系统参考指南》）。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

void EMIF_RestoreConfig(void)

说明： 此函数用恢复组件配置。这将恢复非保留寄存器。此外，此函数还将组件参数值恢复为调用 EMIF_Sleep() 函数之前的状态。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： 如果在调用此函数之前未调用 EMIF_Sleep() 或 EMIF_SaveConfig() 函数，可能产生意外行为。

void EMIF_Wakeup(void)

说明： 这是用来将组件恢复到调用 EMIF_Sleep() 时的状态的首选子程序。EMIF_Wakeup() 函数调用 EMIF_RestoreConfig() 函数以用来恢复该配置。如果在调用 EMIF_Sleep() 前启用该组件，则 EMIF_Wakeup() 函数还将重新启用该组件。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： 如果在调用 EMIF_Wakeup() 函数之前未调用 EMIF_Sleep() 或 EMIF_SaveConfig() 函数，可能产生意外行为。

定义

CYDEV_EXTMEM_BASE — 用于在外部存储器中定位参数的便利宏。

固件源代码示例

PSoC Creator 在“查找示例项目”对话框中提供了大量包括原理图和代码示例的示例项目。要获取组件特定的示例，请打开组件目录中的对话框或原理图中的组件实例。要获取通用的示例，请打开 Start Page（开始页）或 **File**（文件）菜单中的对话框。根据需要，使用对话框中的 **Filter Options**（滤波器选项）可缩小可选项目的列表。

有关更多信息，请参见 PSoC Creator 帮助中的“Find Example Project（查找示例项目）”主题。

功能描述

通常，此组件与 PSoC Creator 均不提供用于定位外部存储器中的代码、变量、结构或阵列的任何外部支持。然而，在此组件或 PSoC Creator 中却没有阻止高级用户为此目的而创建或修改源代码、链接脚本或其他任何源文件的任何程序。

初始化变量不应放置在外部存储器中。请勿在外部存储器中定位通过 PSoC Creator 自动生成的代码、变量、结构和阵列。

PSoC 3 EMIF 硬件和相关 GPIO 端口支持高达 24 位的地址总线和 8 位或 16 位数据总线。

外部存储器最多需要 6 个 GPIO 端口，用来连接地址、数据和控制行：1-3 个端口用于连接地址；1-2 个端口用于连接数据，1 个端口用于连接控制行。对于地址和数据来说，必须选择整个端口。控制端口中未使用的引脚仍然可以用作 GPIO。

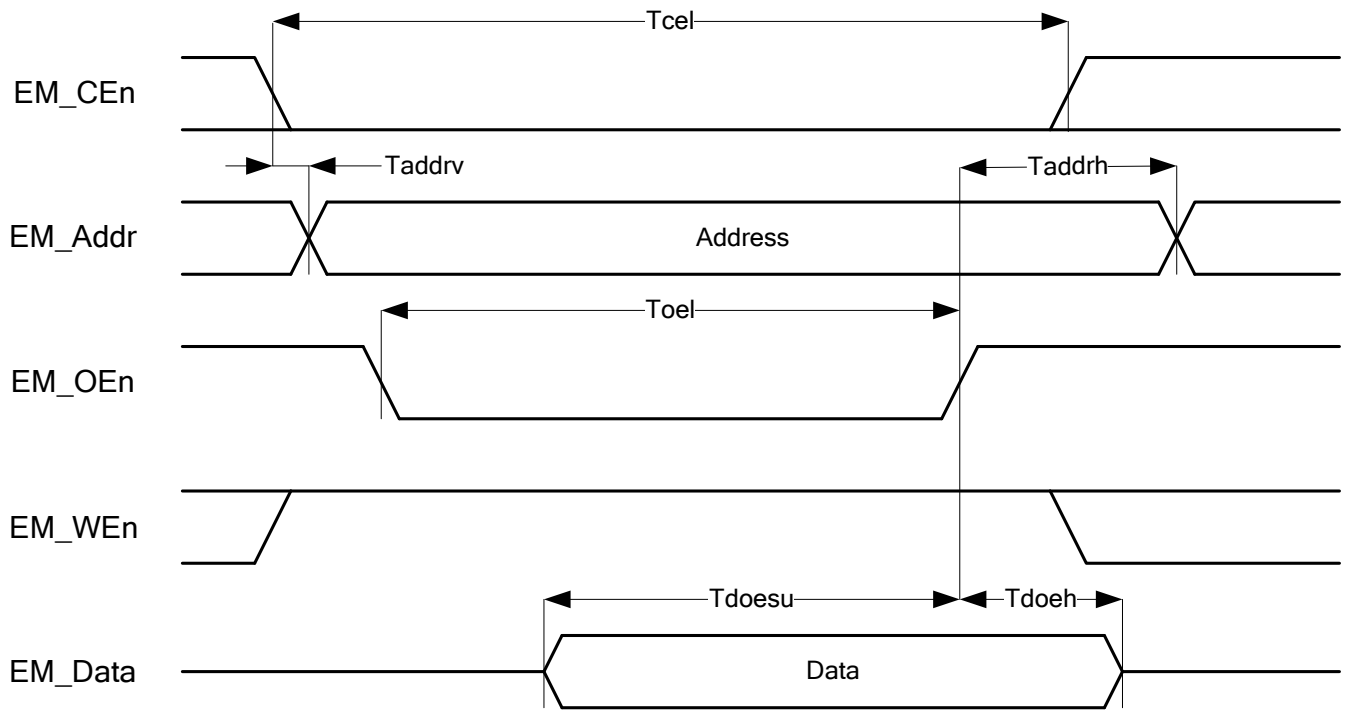
重要的是，当 EMIF 组件放置在项目原理图上时，一定要为地址和数据设置 I/O 端口标示。不需在符号或原理图上标示地址和数据端口；而是在 Design-Wide Resources（设计范围资源）窗口的 Pins（引脚）选项卡中完成该标示。控制信号在组件符号上显示为终端；通常情况下，这些终端不应路由至引脚组件。

共有 3 种 EMIF 类型 — 异步、同步或自定义（不常使用）。所选类型用来确定符号大小、符号中的终端数及符号中的某些文本。自定义组件版本是包含输入终端 udb_ready 的唯一版本。据预计，xmem_wr 与 xmem_rd 信号将与基于 UDB 的逻辑绑定，二者均向外部 IC 生成相应的控制信号，同时提供通过 udb_ready 终端返回至 PHUB 的反馈信号。您可以为自定义模式信号确定相应的时序。

直流和交流电气特性

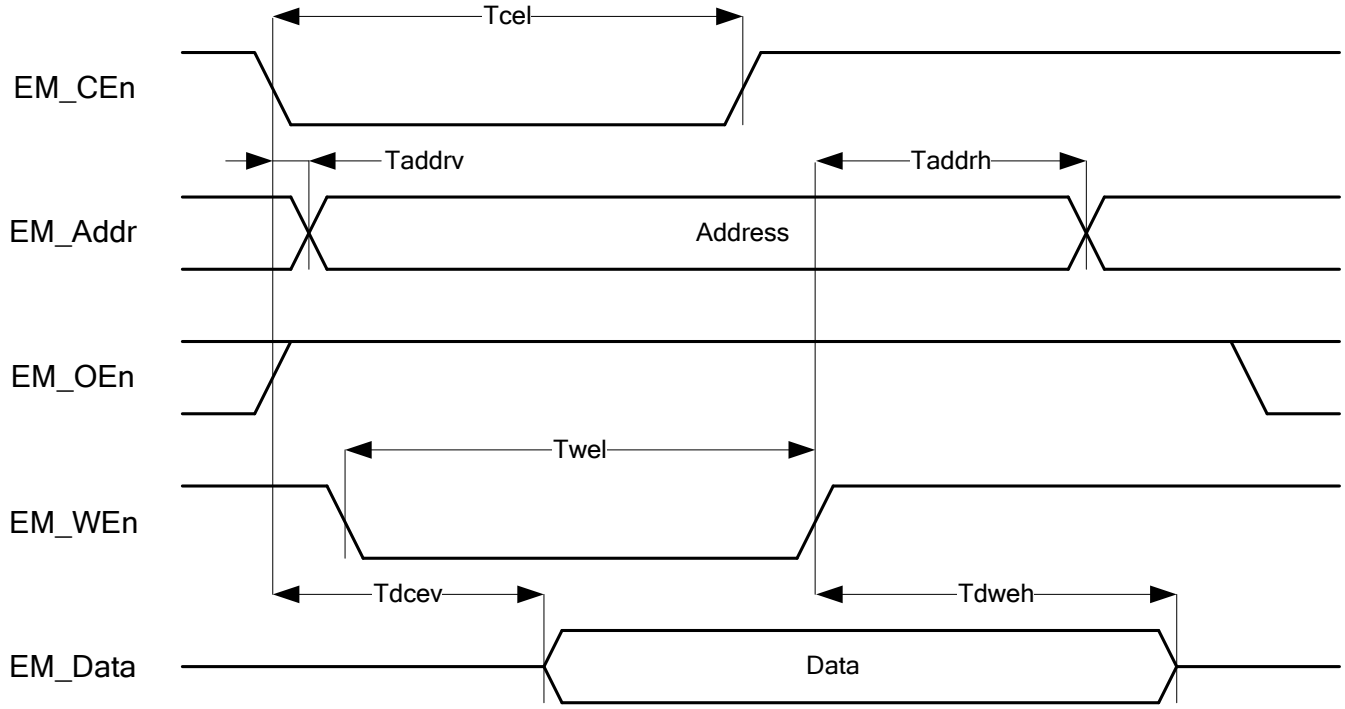
下列值表示期望的性能，它们基于初始特性数据。除非下表中另外指定，否则条件均为：T_A = 25°C、V_{DD} = 5.0 V、功耗高、运算放大器偏压低、以模拟接地电压为参考的输出电压 = V_{SSA}。

图 4. 异步读取周期 SRAM 时序



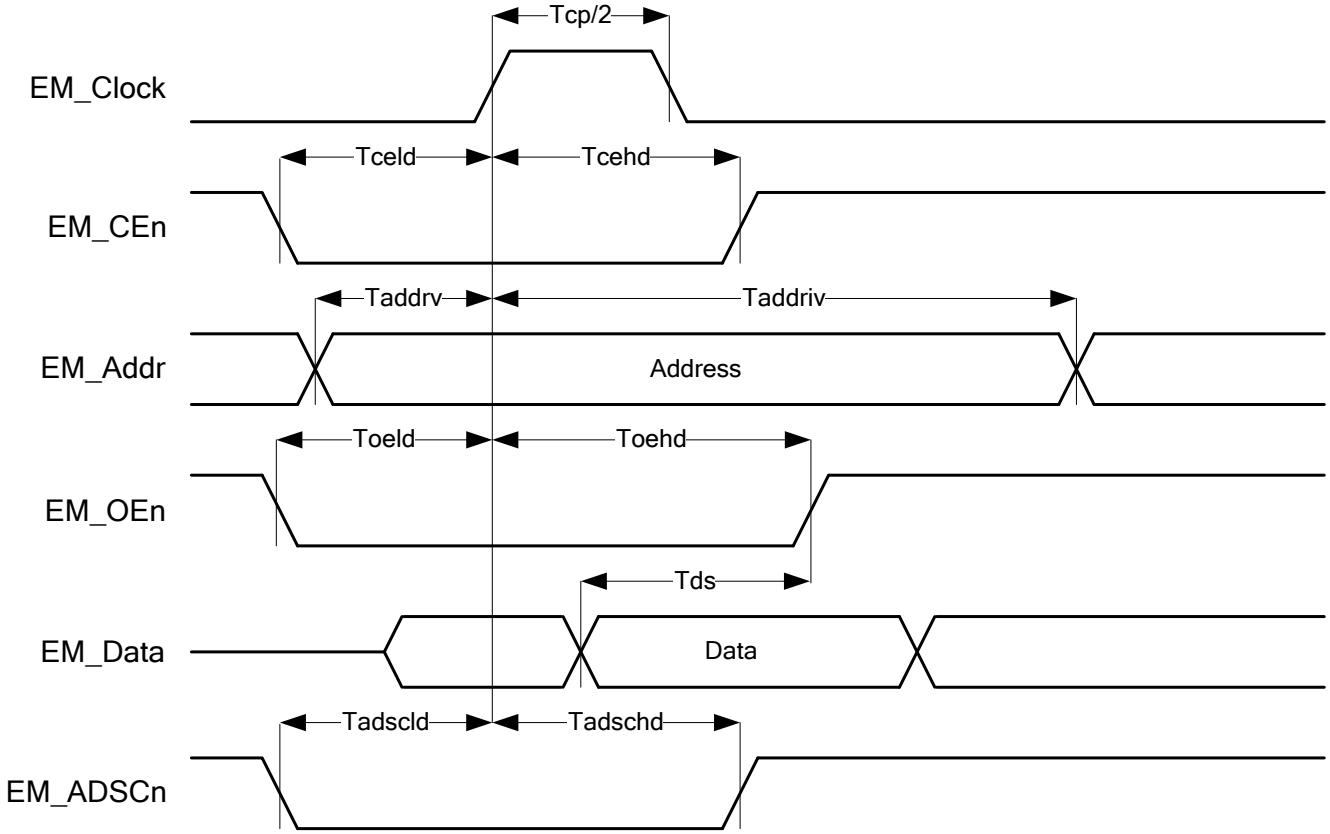
符号	说明	条件	最小值	典型值	最大值	单位
T	EMIF 时钟周期	V_{DDA} ≥ 3.3 V	30.3	-	-	ns
Tcel	EM_CEn 为低的时间		2T – 5	-	2T + 5	ns
Taddrv	从 EM_CEn 为低到 EM_Addr 有效经过的时间		-	-	5	ns
Taddrh	EM_WEn 为高后的地址保持时间		T	-	-	ns
Toel	EM_OEn 为低的时间		2T – 5	-	2T + 5	ns
Tdoesu	从读取数据出现到 EM_OEn 为高的建立时间		T + 15	-	-	ns
Tdoeh	EM_OEn 为高后的数据保持时间		3	-	-	ns

图 5. 异步写入周期 SRAM 时序



符号	说明	条件	最小值	典型值	最大值	单位
T	EMIF 时钟周期	V_{DDA} ≥ 3.3 V	30.3	-	-	ns
Tcel	EM_CEn 为低的时间		T – 5	-	T + 5	ns
Taddrv	从 EM_CEn 为低到 EM_Addr 有效经过的时间		-	-	5	ns
Taddrh	EM_WEn 为高后的地址保持时间		T	-	-	ns
Twel	EM_WEn 为低的时间		T – 5	-	T + 5	ns
Tdcev	从 EM_CEn 为低到数据有效经过的时间		-	-	7	ns
Tdweh	EM_WEn 为高后的数据保持时间		T	-	-	ns

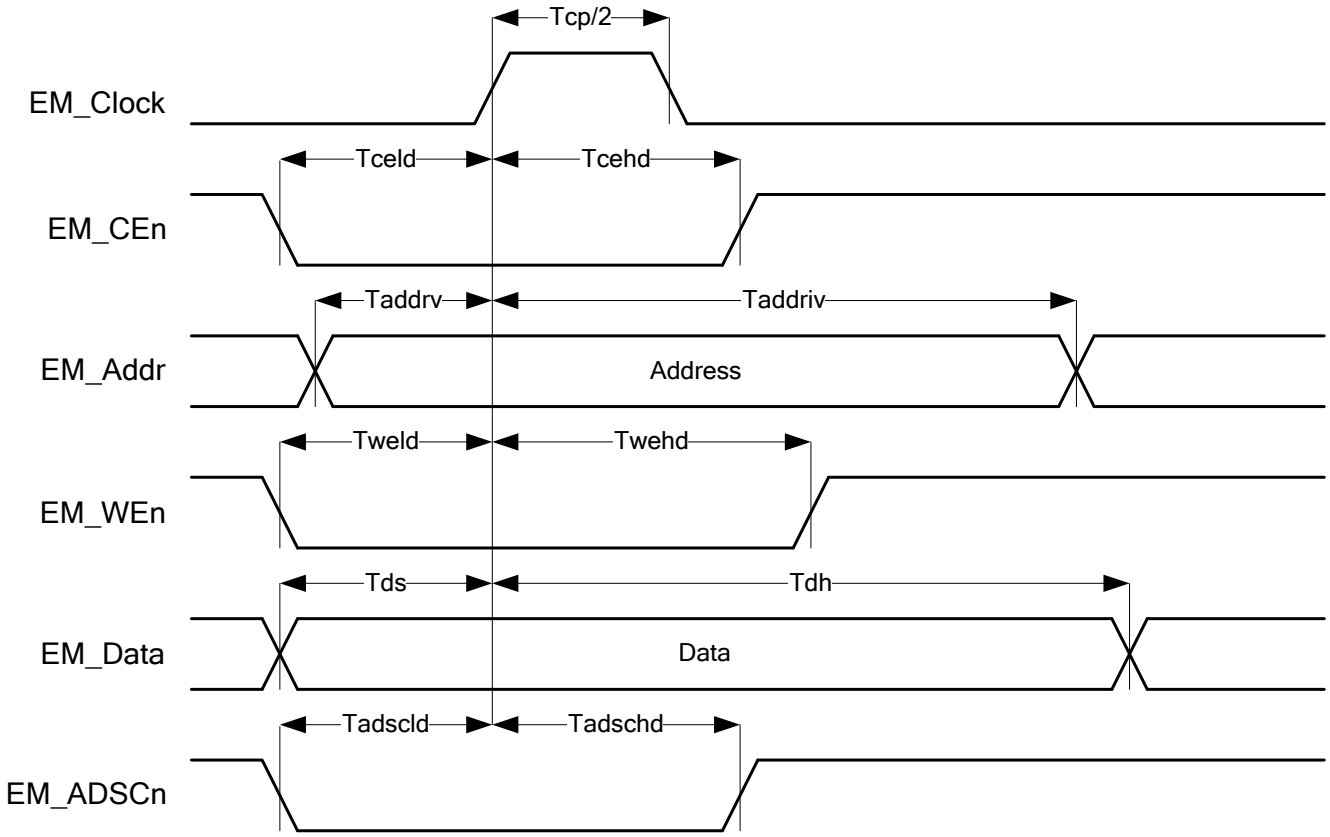
图 6. 同步读循环时序



符号	说明	条件	最小值	典型值	最大值	单位
T	EMIF 时钟周期	V_{DDA} ≥ 3.3 V	30.3	-	-	ns
Tcp/2	EM_Clock 脉冲为高		T/2	-	-	ns
Tceld	从 EM_CEn 为低到 EM_Clock 为高经过的时间		5	-	-	ns
Tcehd	从 EM_Clock 为高到 EM_CEn 为高经过的时间		T/2 — 5	-	-	ns
Taddrv	从 EM_Addr 有效到 EM_Clock 为高经过的时间		5	-	-	ns
Taddriv	从 EM_Clock 为高到 EM_Addr 无效经过的时间		T/2 — 5	-	-	ns
Toeld	从 EM_OEn 为低到 EM_Clock 为高经过的时间		5	-	-	ns
Toehd	从 EM_Clock 为高到 EM_OEn 为高经		T	-	-	ns

	过的时间					
Tds	EM_OEn 为高之前数据有效的时间		T + 15	-	-	ns
Tadscld	从 EM_ADSCn 为低到 EM_clock 为高经过的时间		5	-	-	ns
Tadschd	从 EM_Clock 为高到 EM_ADSCn 为高经过的时间		T/2 — 5	-	-	ns

图 7. 同步写循环时序



符号	说明	条件	最小值	典型值	最大值	单位
T	EMIF 时钟周期	V_{DDA} ≥ 3.3 V	30.3	-	-	ns
Tcp/2	EM_Clock 脉冲为高		T/2	-	-	ns
Tceld	从 EM_CEn 为低到 EM_Clock 为高经过的时间		5	-	-	ns
Tcehd	从 EM_Clock 为高到 EM_CEn 为高经过		T/2 — 5	-	-	ns

	的时间					
Taddrv	从 EM_Addr 有效到 EM_Clock 为高经过的时间		5	-	-	ns
Taddriv	从 EM_Clock 为高到 EM_Addr 无效经过的时间		T/2 — 5	-	-	ns
Tweld	从 EM_WEn 为低到 EM_Clock 为高经过的时间		5	-	-	ns
Twehd	从 EM_Clock 为高到 EM_WEn 为高经过的时间		T/2 — 5	-	-	ns
Tds	EM_Clock 为高之前数据有效的时间		5	-	-	ns
Tdh	EM_Clock 为高后数据无效的时间		T	-	-	ns
Tadscld	从 EM_ADSCn 为低到 EM_clock 为高经过的时间		5	-	-	ns
Tadschd	从 EM_Clock 为高到 EM_ADSCn 为高经过的时间		T/2 — 5	-	-	ns

组件更改

版本 1.0 是 EMIF 组件的首次发行版。

© 赛普拉斯半导体公司，2012。此处所包含的信息可能会随时更改，恕不另行通知。除赛普拉斯产品的内嵌电路之外，赛普拉斯半导体公司不对任何其他电路的使用承担任何责任。也不根据专利或其他权利以明示或暗示的方式授予任何许可。除非与赛普拉斯签订明确的书面协议，否则赛普拉斯产品不保证能够用于或适用于医疗、生命支持、救生、关键控制或安全应用领域。此外，对于可能发生运转异常和故障并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统中，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

CyDesigner™、Programmable System-on-Chip™ 和 PSoC Express™ 是赛普拉斯半导体公司的商标；PSoC® 是赛普拉斯半导体公司的注册商标。此处引用的所有其他商标或注册商标归其各自所有者所有。

所有源代码（软件和/或固件）均归赛普拉斯半导体公司（赛普拉斯）所有，并受全球专利法规（美国和美国以外的专利法规）、美国版权法以及国际条约规定的保护和约束。赛普拉斯据此向获许可者授予适用于个人的、非独占性、不可转让的许可，用以复制、使用、修改、创建赛普拉斯源代码的派生作品、编译赛普拉斯源代码和派生作品，并且其目的只能是创建自定义软件和/或固件，以支持获许可者仅将其获得的产品依照适用协议规定的方式与赛普拉斯集成电路配合使用。除上述指定的用途之外，未经赛普拉斯的明确书面许可，不得对此类源代码进行任何复制、修改、转换、编译或演示。

免责声明：赛普拉斯不针对此材料提供任何类型的明示或暗示保证，包括（但不仅限于）针对特定用途的适销性和适用性的暗示保证。赛普拉斯保留在不做出通知的情况下对此处所述材料进行更改的权利。赛普拉斯不对此处所述之任何产品或电路的应用或使用承担任何责任。对于可能发生运转异常和故障并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统中，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

产品使用可能受适用的赛普拉斯软件许可协议限制。