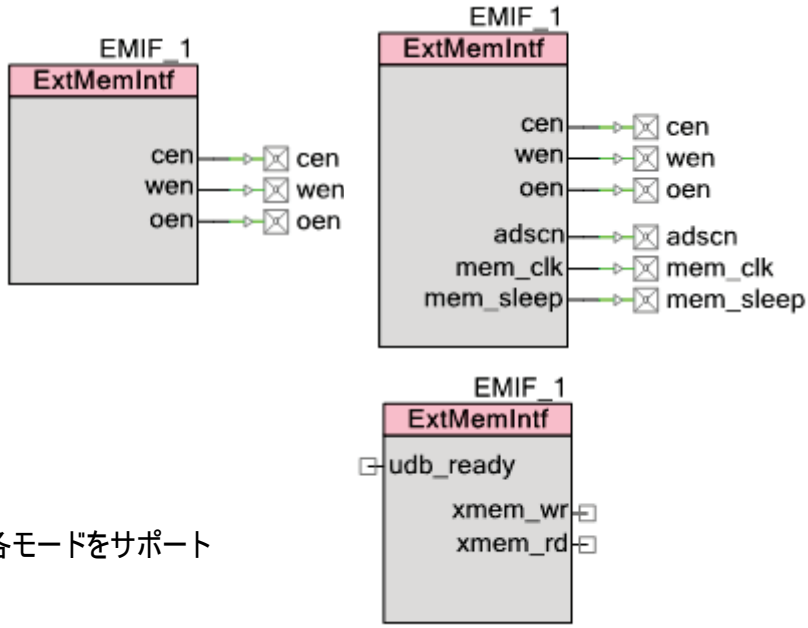
# External Memory Interface (EMIF)
1.0

## 特長




- 8/16/24 ビットアドレスバス幅
- 8/16 ビットデータバス幅
- 外部同期メモリをサポート
- 外部非同期メモリをサポート
- メモリ用カスタムインタフェースをサポート
- 外部メモリの速度を広範囲にサポート (5 ～ 200 ns)
- 外部メモリの電源オフ、スリープ、復帰の各モードをサポート

## 概要

EMIF コンポーネントにより CPU または DMA が PSoC 3 外部のメモリ IC にアクセスすることができます。これにより EMIF ハードウェアおよび、必要に応じて UDB と GPIO の設定が簡便化されます。EMIF は、任意のUDB を同期および非同期モードに設定する必要なく、同期および非同期メモリをコントロールできます。UDB モードでは、外部メモリコントロール信号を生成するように UDB を設定する必要があります。

### EMIF の用途

EMIF を使用して、使用可能なメモリ容量を拡張します。これにより、外付けメモリデバイスへのデータストレージを拡張することができます。通常、PSoC 3 に接続する外部メモリ IC はテキスト、オーディオ、またはビデオ コンテンツなど幅広いデータアレイを保存することを目的としています。その他の予想されるアプリケーションには、データロギングが含まれており、外部 LCD ピクセルデータのバッファとして使用されます。

# 入出力接続

ここでは、EMIF のさまざまな入出力接続について説明します。I/O 項目のアスタリスク (*) はその I/O が、説明に挙げられた条件において、回路シンボルに表示されない場合があることを示します。

## udb_ready – 入力*

AHB バス送信が終了したことを PHUB に示す、外部ロジックからの信号です (アクティブ HIGH)。この入力は **External Memory Type** パラメータが **Custom** に設定されていると表示されます。

## xmem_wr – 出力*

カスタムインタフェースへの汎用書き込み信号 (アクティブ HIGH)。AHB バス送信が終了したことを PHUB に示す、外部ロジックからの信号です (アクティブ HIGH)。この出力は **External Memory Type** パラメータが **Custom** に設定されていると表示されます。

## xmem_rd – 出力*

カスタムインタフェースへの汎用読み取り信号 (アクティブ HIGH)。この出力は **External Memory Type** パラメータが **Custom** に設定されていると表示されます。

## cen – 出力*

外部メモリに対するチップイネーブルです (アクティブ LOW)。この出力は **External Memory Type** パラメータが **Synchronous** または **Asynchronous** に設定されていると表示されます。

## wen – 出力*

外部メモリに対する書き込みイネーブルです (アクティブ LOW)。この出力は **External Memory Type** パラメータが **Synchronous** または **Asynchronous** に設定されていると表示されます。

## oen – 出力*

外部メモリに対する出力イネーブルです (アクティブ LOW)。この出力は **External Memory Type** パラメータが **Synchronous** または **Asynchronous** に設定されていると表示されます。

## mem_clk – 出力*

外部同期メモリに対するクロックです。この出力は **External Memory Type** パラメータが **Synchronous** に設定されていると表示されます。

### adscn – 出力*

外部同期メモリに対するアドレスストローブです (アクティブ LOW)。この出力は **External Memory Type** パラメータが **Synchronous** に設定されていると表示されます。

### mem_sleep – 出力*

外部同期メモリのスリープピンに対するスリープ信号です (アクティブ HIGH)。この出力は **External Memory Type** パラメータが **Synchronous** に設定されていると表示されます。

## 回路図マクロ情報

デフォルトで、PSoC Creator コンポーネントカタログには、EMIF コンポーネントの回路図マクロ実装が入っています。これらのマクロには、既に接続した出力ピンが含まれます。回路図マクロは、非同期外部メモリインタフェースおよび同期外部メモリインタフェースの両方で利用できます。

**図 1. 非同期外部メモリインタフェース回路図マクロ**

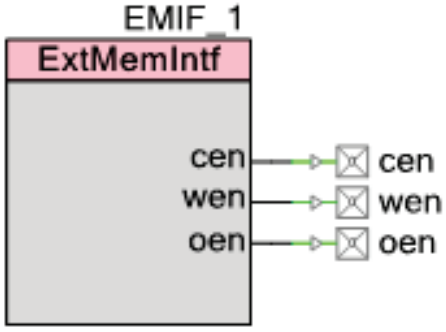
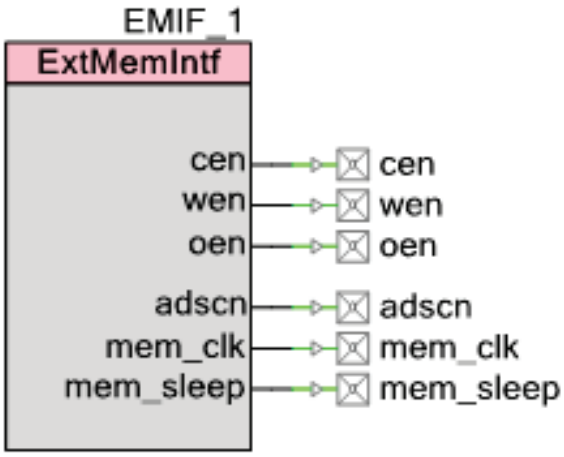


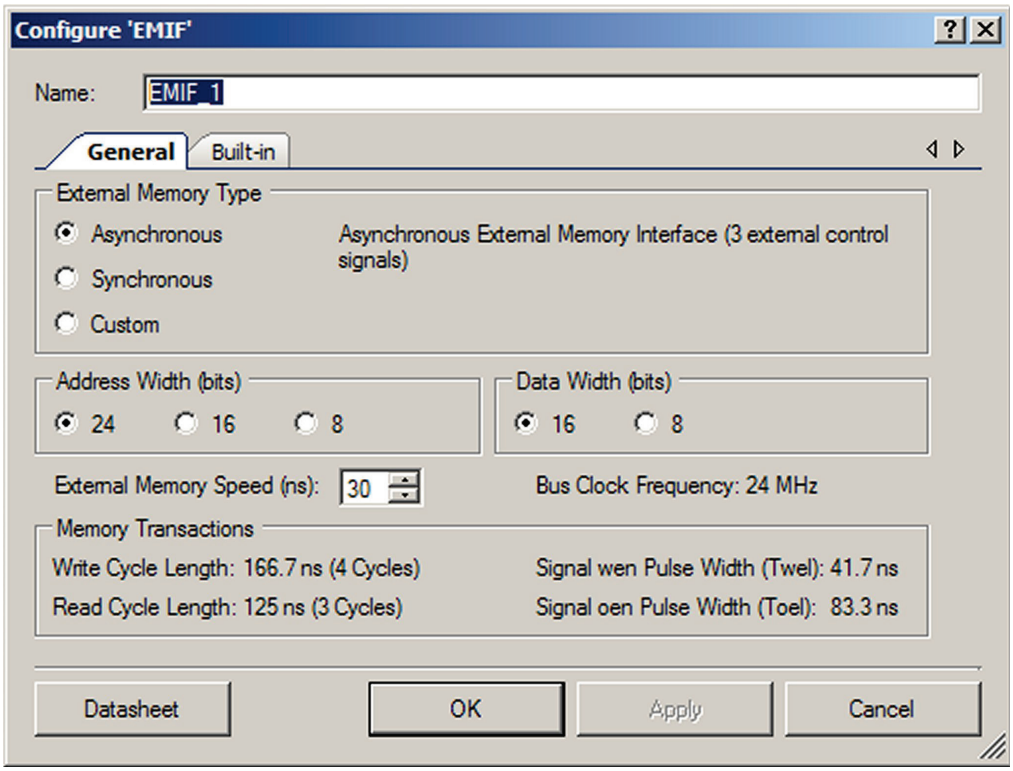
**図 2. 同期外部メモリインタフェース回路図マクロ**




# コンポーネント・パラメータ

EMIF コンポーネントを設計上にドラッグし、ダブルクリックして **Configure** ダイアログを開きます。

**図 3. Configure EMIF ダイアログ**



EMIF には、次のパラメータがあります。

### 外部メモリの種類

外部メモリの種類を決定します。デフォルトの外部メモリの種類は **Asynchronous** です。

### アドレス幅 (ビット)

外部メモリのアドレスバスのビット数を決定します。デフォルトのアドレスバス幅は **24 bits** です。

### データ幅 (ビット)

外部メモリのデータバスのビット数を決定します。デフォルトのデータバス幅は **16 bits** です。

### 外部メモリ速度 (ns)

外部メモリ速度を決定します (単位 ns)。デフォルトの外部メモリ速度は **30 ns** です。

### バスクロック周波数

選択したバスクロック周波数を表示します (単位 MHz)。

### メモリトランザクション

- **Write Cycle Length** – 書き込みサイクル長を表示します (単位 ナノ秒)。バスクロック周波数サイクルは括弧内に表示されます。
- **Read Cycle Length** – 読み取りサイクル長を表示します (単位 ナノ秒)。バスクロック周波数サイクルは括弧内に表示されます。
- **Signal wen Pulse Width (Twel)** – Twel パラメータ (書き込みイネーブル信号のパルス幅) を表示します (単位 ナノ秒)。詳しくは、「DC 電気的特性と AC 電気的特性」セクションを参照してください。
- **Signal oen Pulse Width (Toel)** – Toel パラメータ (書き込みイネーブル信号のパルス幅) を表示します (単位 ナノ秒)。詳しくは、「DC 電気的特性と AC 電気的特性」 セクションを参照してください。

## 配置

EMIF コンポーネントは EMIF ハードウェアブロック内に配置されます。

# リソース

外部メモリインタフェース (EMIF) は PSoC 3 ファミリの専用ハードウェアであり、UDB と組み合わせて使用することにより、外部メモリデバイスに接続できます。

| モード | デジタルブロック | API メモリ (バイト) | | ピン (外部入出力ごと) |
|---|---|---|---|---|
| | EMIF | フラッシュ | RAM | |
| Asynchronous (非同期) | 1 | 204 | 4 | データ幅 (8 ～16 ビット) + アドレス幅 (8 ～ 24 ビット) + 3 |
| 同期 | 1 | 219 | 4 | データ幅 (8 ～16 ビット) + アドレス幅 (8 ～ 24 ビット) + 6 |
| カスタム | 1 | 185 | 4 | データ幅 (8 ～16 ビット) + アドレス幅 (8 ～ 24 ビット) + ユーザ定義の制御信号 |

# アプリケーション プログラミング インタフェース

アプリケーション・プログラミング・インターフェース (API) ルーチンにより、ソフトウェアを使用してコンポーネントを設定できます。次の表は、各関数へのインターフェースとその説明を示しています。続くセクションでは、各関数について詳しく説明します。

デフォルトでは、PSoC Creator は、インスタンス名「EMIF_1」を設計上のコンポーネントの最初のインスタンスに割り当てます。コンポーネントのインスタンス名称は、識別子の文法ルールに従って固有の名前に変更できます。インスタンス名は、すべてのグローバル関数名、変数名、定数名のプリフィックスになります。読みやすいように、下表では「EMIF」というインスタンス名を使用しています。

| 関数 | 説明 |
|---|---|
| EMIF_Start() | EMIF_Init() および EMIF_Enable() を呼び出します |
| EMIF_Stop() | EMIF ブロックをディスエーブルにします。関連付けられている全ての I/O ポートおよびピンを HI-Z モードに返します |
| EMIF_Init() | EMIF 設定を初期化し、現在のカスタマイザ状態に復元します |
| EMIF_Enable() | EMIF ハードウェアブロック、関連付けられた I/O ポートおよびピンを有効にします |
| EMIF_ExtMemSleep() | 外部メモリのスリープ信号を HIGH に設定します。使用する外部メモリ IC の種類により、信号の反転が必要になる場合があります |
| EMIF_ExtMemWakeup() | 外部メモリのスリープ信号を LOW に設定します。使用する外部メモリ IC の種類により、信号の反転が必要になる場合があります |
| EMIF_SaveConfig() | EMIF 非保持レジスタのユーザ設定を保存します。このルーチンは EMIF_Sleep() によって呼び出され、スリープに入る前にコンポーネント設定を保存します |

| 関数 | 説明 |
|---|---|
| EMIF_Sleep() | EMIF 動作を停止し、ユーザ設定および EMIF のイネーブル状態を保存します |
| EMIF_RestoreConfig() | EMIF 非保持レジスタのユーザ設定を復元します。このルーチンは EMIF_Wakeup() により呼び出され、スリープを出る前にコンポーネント設定を復元します |
| EMIF_Wakeup() | ユーザ設定とイネーブル状態を復元します |

## void EMIF_Start(void)

**説明:** これは、コンポーネントの動作を開始する際に推奨される方法です。EMIF_Start() は EMIF_Init() 関数を呼び出してから、EMIF_Enable() 関数を呼び出します。

**パラメータ:** なし

**返り値:** なし

**副作用:** なし

## void EMIF_Stop(void)

**説明:** EMIF ブロックをディスエーブルにします。関連付けられている全ての I/O ポートおよびピンを HI-Z モードに返します。

**パラメータ:** なし

**返り値:** なし

**副作用:** なし

## void EMIF_Init(void)

**説明:** カスタマイザのコンフィギュアダイアログの設定に従って、コンポーネントを初期化または復元します。EMIF_Start() ルーチンが EMIF_Init() 関数を呼び出すので、この関数を呼び出す必要はありません。これはコンポーネントの動作を開始する際に推奨される方法です。

**パラメータ:** なし

**返り値:** なし

**副作用:** なし

## void EMIF_Enable(void)

**説明:** ハードウェアの使用を開始し、コンポーネントの動作を開始します。EMIF_Start() ルーチンが EMIF_Enable() 関数を呼び出すので、この関数を呼び出す必要はありません。これはコンポーネントの動作を開始する際に推奨される方法です。

**パラメータ:** なし

**返り値:** なし

**副作用:** なし

## void EMIF_ExtMemSleep(void)

**説明:** EMIF_PWR_DWN レジスタの「mem_pd」ビットを設定します。これにより、外部メモリのスリープ信号を HIGH に設定します。使用する外部メモリ IC の種類により、信号を反転する必要があることがあります。

**パラメータ:** なし

**返り値:** なし

**副作用:** なし

## void EMIF_ExtMemWakeup(void)

**説明:** EMIF_PWR_DWN レジスタの「mem_pd」ビットをリセットします。これにより、外部メモリのスリープ信号を LOW に設定します。使用する外部メモリ IC の種類により、信号を反転する必要があることがあります。

**パラメータ:** なし

**返り値:** なし

**副作用:** なし

## void EMIF_SaveConfig(void)

**説明:** この関数は、コンポーネントの設定を保存します。保持されないレジスタも保存します。この関数は、Configure ダイアログで定義されている、または該当する API で変更される、現在のコンポーネントパラメータ値も保存します。この関数は、EMIF_Sleep() 関数に呼び出されます。

**パラメータ:** なし

**返り値:** なし

**副作用:** なし

## void EMIF_Sleep(void)

**説明:** これは、コンポーネントのスリープを準備するのに推奨されるルーチンです。EMIF_Sleep() ルーチンは、現在のコンポーネントの状態を保存します。次に、EMIF_Stop() 関数を呼び出し、EMIF_SaveConfig() を呼び出してハードウェア構成を保存します。

CyPmSleep() および CyPmHibernate() 関数を呼び出す前に、EMIF_Sleep() 関数を呼び出してください。電源管理関数については、PSoC Creator *System Reference Guide* を参照してください。

**パラメータ:** なし

**返り値:** なし

**副作用:** なし

## void EMIF_RestoreConfig(void)

**説明:** この関数は、コンポーネントの設定を復元します。維持されないレジスタも復元します。この関数はまた、コンポーネントのパラメータ値を EMIF_Sleep() 関数を呼び出す前の状態に復旧します。

**パラメータ:** なし

**返り値:** なし

**副作用:** EMIF_Sleep()または EMIF_SaveConfig() 関数を呼び出す前に、この関数を呼び出した場合、予期しない動作を示すことがあります。

## void EMIF_Wakeup(void)

**説明:** これは、コンポーネントを EMIF_Sleep() が呼び出されたときの状態に復元するのに推奨されるルーチンです。EMIF_Wakeup() 関数は、ユーザー設定を復旧させるために EMIF_RestoreConfig() 関数を呼び出します。EMIF_Sleep() 関数が呼び出される前にコンポーネントがイネーブルになった場合、EMIF_Wakeup() 関数がコンポーネントを再度イネーブルにします。

**パラメータ:** なし

**返り値:** なし

**副作用:** 最初に EMIF_Sleep() または EMIF_SaveConfig() 関数を呼び出すことなく EMIF_Wakeup() 関数を呼び出すと、予期しない動作を示すことがあります。

## 定義

**CYDEV_EXTMEM_BASE –** パラメータを外部メモリに配置するための便宜上のマクロです。

## ファームウェア・ソースコードのサンプル

PSoC Creator は、Find Example Project ダイアログに数多くのサンプルプロジェクトを提供しており、そこには回路図およびコード例が含まれています。コンポーネント固有の例を見るには、コンポーネントカタログまたは回路図に置いたコンポーネントインスタンスからダイアログを開きます。一般例については、「Start Page (スタートページ)」または **File** メニューからダイアログを開きます。必要に応じてダイアログにある **Filter Options** を使用し、選択できるプロジェクトのリストを絞り込みます。

詳しくは、PSoC Creator ヘルプの「Find Example Project（サンプルプロジェクトを検索）」を参照してください。

## 機能の説明

このコンポーネントおよび PSoC Creator では一般的に、コード、変数、構造、アレイを外部メモリに配置するための直接のサポートを提供していません。ただし、このような目的で上級ユーザがこのコンポーネントまたは PSoC Creator を使用して、ソースコード、リンカスクリプト、またはその他のソースファイルを作成または変更することは可能です。

初期化した変数は外部メモリに配置しないでください。PSoC Creator が自動生成したコード、変数、構造およびアレイを外部メモリに配置しないでくださいさい。

PSoC 3 EMIF ハードウェアおよび関連する GPIO ポートは最大で 24 ビットのアドレスバス、および 8 ビットまたは 16 ビットのデータバスをサポートします。

アドレス、データ、および制御ラインに接続するため、外部メモリには最大で 6 つの GPIO ポートが必要です (1 ～ 3 ポートはアドレス、1 ～ 2 ポートはデータ、1 ポートは制御)。アドレスおよびデータでは、すべてのポートを選択してください。制御ポートで未使用のピンは GPIO として使用できます。

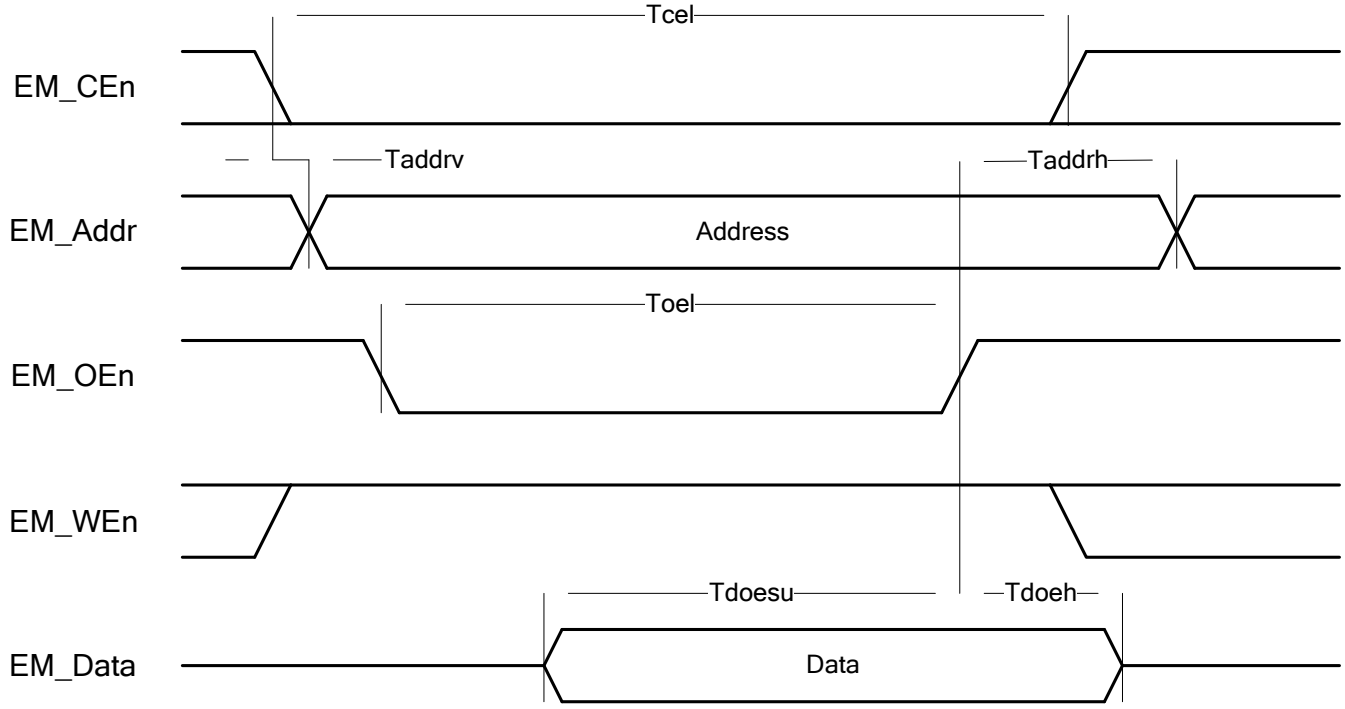
EMIF コンポーネントがプロジェクト回路図に配置されている場合、I/O ポートの指定をアドレスおよびデータに設定することが重要です。アドレスおよびデータのポートの指定は記号や回路図ではなく、Design-Wide Resources ウィンドウの Pins タブで実施します。制御信号はコンポーネントの記号で端子として表示され、端子はピンコンポーネントに通常の方法でルーティングしなければなりません。

EMIF には 3 種類 – 非同期、同期、カスタム (使用例は稀) があります。選択した種類により記号のサイズ、記号の端子の数、記号内のテキストの一部を決定します。コンポーネントのカスタムバージョンのみに入力端子 udb_ready が含まれます。xmem_wr および xmem_rd 信号を UDB ベースのロジックに接続することが予想されます。これら両方が外部 IC に対して適切な制御信号を生成し、udb_ready 端子を経由して PHUB にフィードバック信号を戻します。カスタムモード信号の適切なタイミングは、それぞれのユーザが決定してください。

## DC 電気的特性と AC 電気的特性

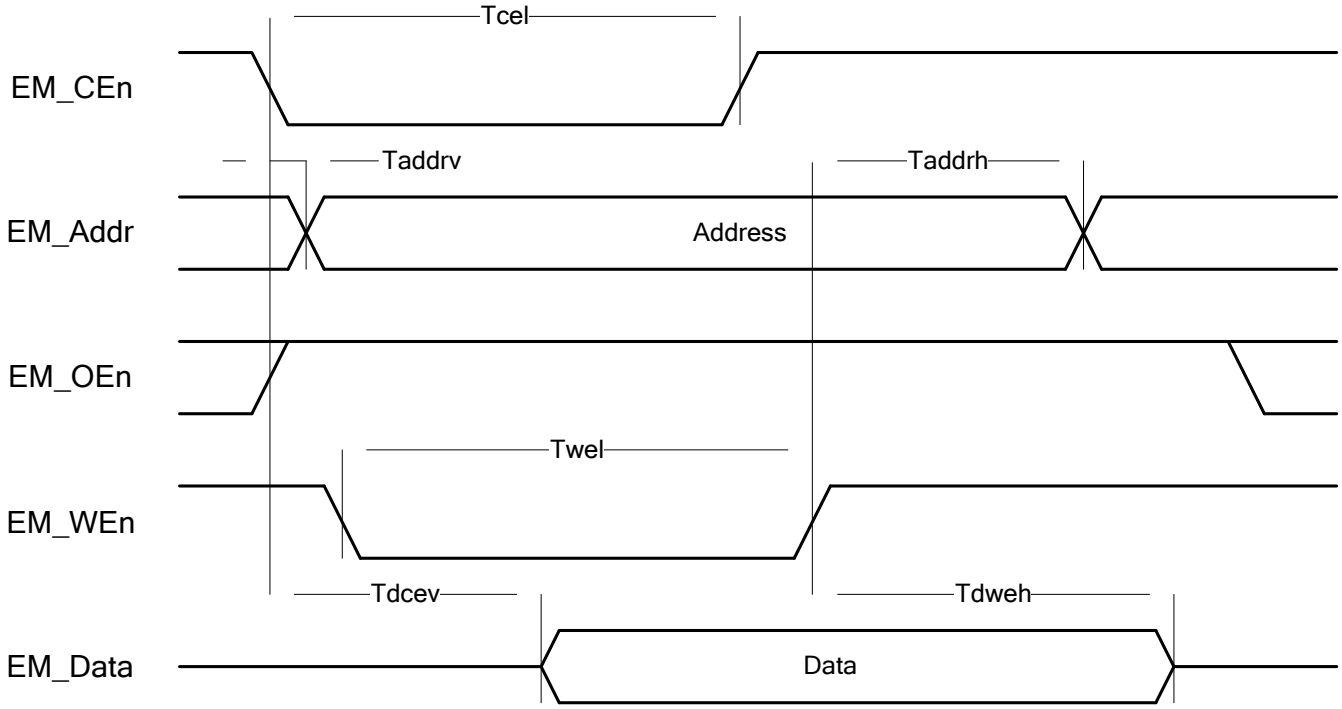
次の値は予想されるパフォーマンスを示唆したもので、初期の特性データに基づいています。下の表で別途指定されている場合を除き、$T_A$ = 25°C, $V_{DD}$ = 5.0 V、High Power、オペアンプ バイアス LOW、出力リファレンスはアナログ グランド = $V_{SSA}$ です。

## 図 4. 非同期リードサイクルの SRAM タイミング



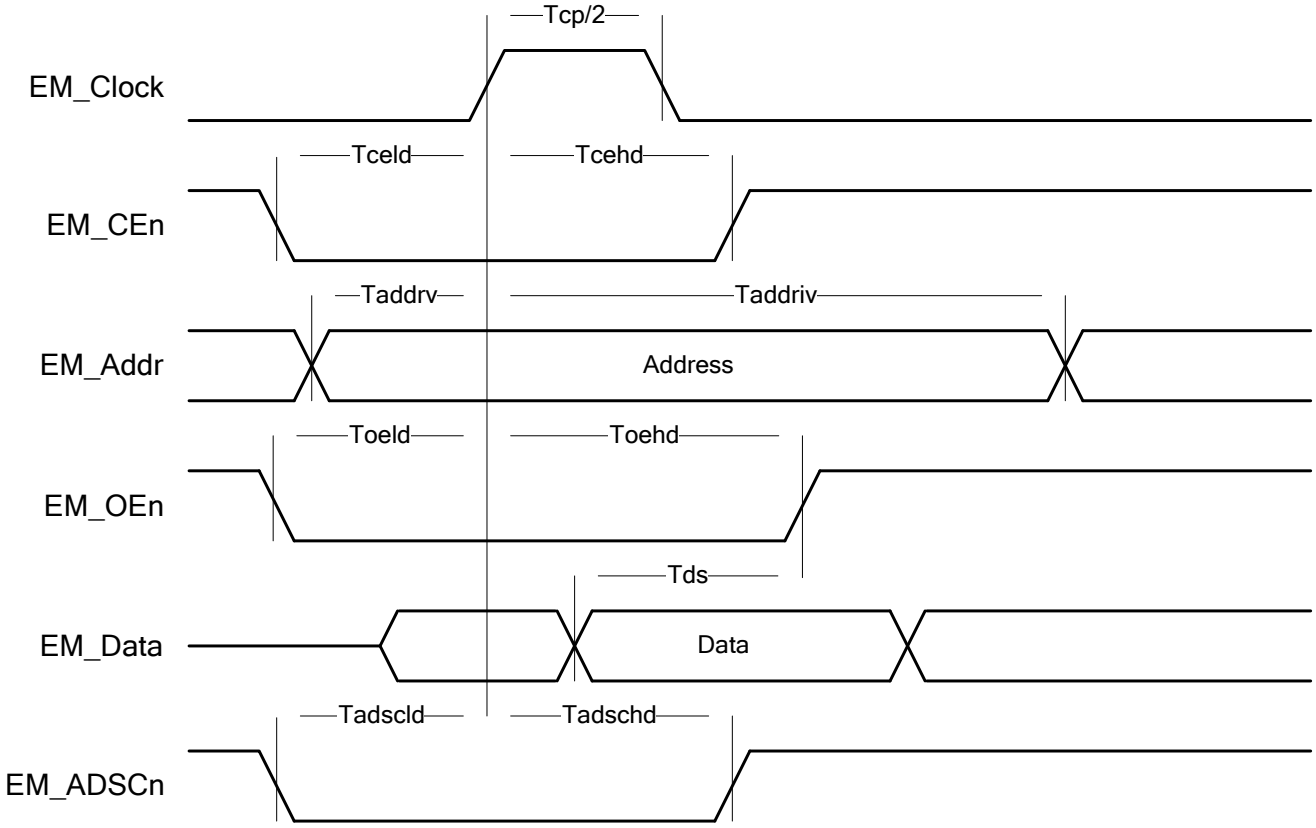

| 記号 | 説明 | 条件 | Min | Typ | Max | 単位 |
|---|---|---|---|---|---|---|
| T | EMIFクロック周期 | $V_{DDA}$ ≥ 3.3 V | 30.3 | - | - | ns |
| Tcel | EM_CEn の LOW 時間 | | 2T – 5 | - | 2T + 5 | ns |
| Taddrv | EM_CEn LOW から EM_Addr 有効までの時間 | | - | - | 5 | ns |
| Taddrh | EM_WEn HIGHの後のアドレスホールド時間 | | T | - | - | ns |
| Toel | EM_OEn の LOW 時間 | | 2T – 5 | - | 2T + 5 | ns |
| Tdoesu | EM_OEn HIGH までのデータセットアップ時間 | | T + 15 | - | - | ns |
| Tdoeh | EM_OEn HIGH の後のデータホールド時間 | | 3 | - | - | ns |

### 図 5. 非同期書き込みサイクルの SRAM タイミング




| 記号 | 説明 | 条件 | Min | Typ | Max | 単位 |
|---|---|---|---|---|---|---|
| T | EMIFクロック周期 | $V_{DDA}$ ≥ 3.3 V | 30.3 | - | - | ns |
| Tcel | EM_CEn の LOW 時間 | | T – 5 | - | T + 5 | ns |
| Taddrv | EM_CEn LOW から EM_Addr 有効までの時間 | | - | - | 5 | ns |
| Taddrh | EM_WEn HIGHの後のアドレスホールド時間 | | T | - | - | ns |
| Twel | EM_WEn の LOW 時間 | | T – 5 | - | T + 5 | ns |
| Tdcev | EM_CEn LOW からデータ有効までの時間 | | - | - | 7 | ns |
| Tdweh | EM_WEn HIGH の後のデータホールド時間 | | T | - | - | ns |

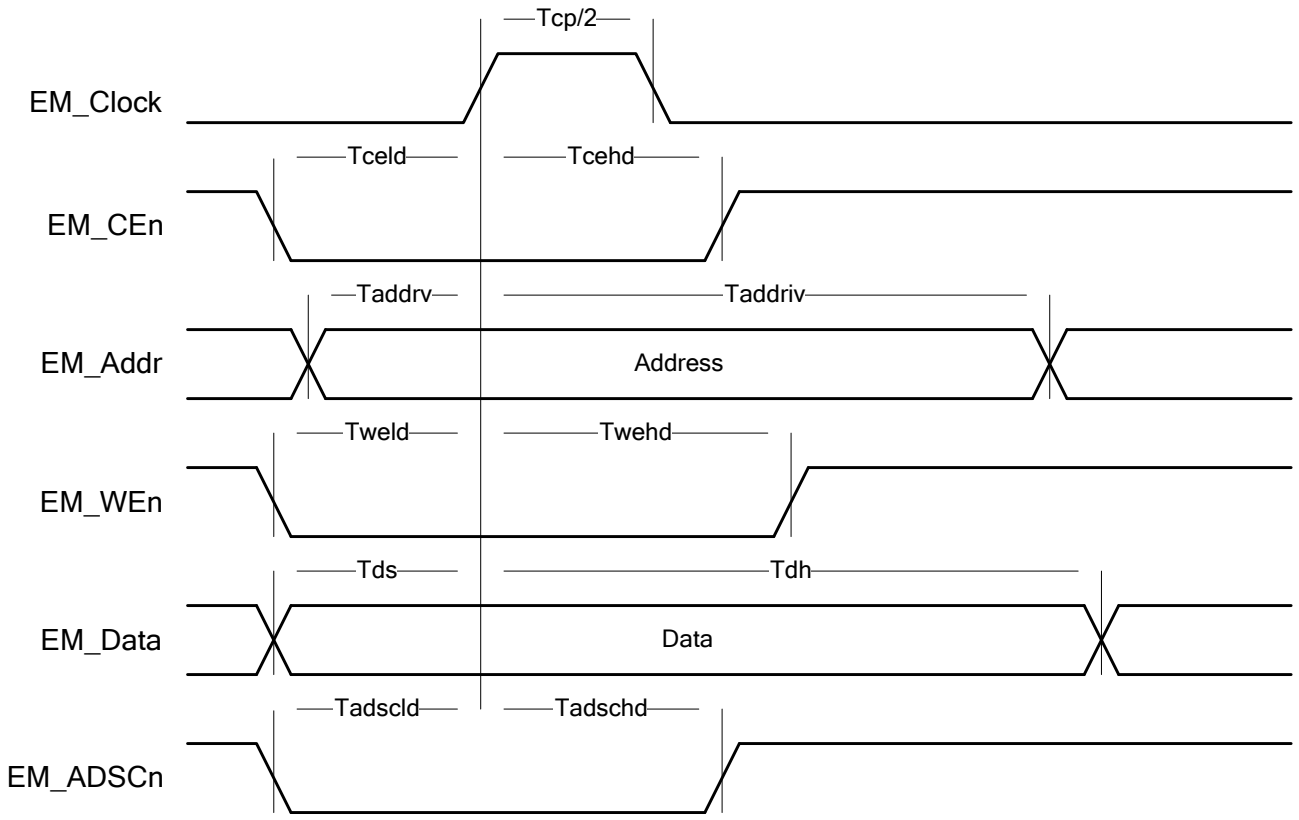
### 図 6. 同期読み取りサイクルのタイミング




| 記号 | 説明 | 条件 | Min | Typ | Max | 単位 |
|---|---|---|---|---|---|---|
| T | EMIFクロック周期 | $V_{DDA}$ ≥ 3.3 V | 30.3 | - | - | ns |
| Tcp/2 | EM_Clock パルス HIGH | | T/2 | - | - | ns |
| Tceld | EM_Clock LOW から EM_Clock HIGH までの時間 | | 5 | - | - | ns |
| Tcehd | EM_Clock HIGH から EM_CEn HIGH までの時間 | | T/2 – 5 | - | - | ns |
| Taddrv | EM_Addr 有効から EM_Clock HIGH までの時間 | | 5 | - | - | ns |
| Taddriv | EM_Clock HIGH から EM_Addr 無効まで | | T/2 – 5 | - | - | ns |
| Toeld | EM_OEn LOW から EM_Clock HIGH までの時間 | | 5 | - | - | ns |
| Toehd | EM_Clock HIGH から EM_OEn HIGH までの時間 | | T | - | - | ns |

| 記号 | 説明 | 条件 | Min | Typ | Max | 単位 |
|---|---|---|---|---|---|---|
| Tds | EM_OEn HIGH になるまでのデータ有効時間 | | T + 15 | - | - | ns |
| Tadscld | EM_ADSCn LOW から EM_Clock HIGH までの時間 | | 5 | - | - | ns |
| Tadschd | EM_Clock HIGH から EM_ADSCn HIGH までの時間 | | T/2 – 5 | - | - | ns |

## 図 7. 同期書き込みサイクルのタイミング




| 記号 | 説明 | 条件 | Min | Typ | Max | 単位 |
|---|---|---|---|---|---|---|
| T | EMIFクロック周期 | $V_{DDA} \geq 3.3$ V | 30.3 | - | - | ns |
| Tcp/2 | EM_Clock パルスが HIGH である時間 | | T/2 | - | - | ns |
| Tceld | EM_Clock LOW から EM_Clock HIGH までの時間 | | 5 | - | - | ns |

| 記号 | 説明 | 条件 | Min | Typ | Max | 単位 |
|---|---|---|---|---|---|---|
| Tcehd | EM_Clock HIGH から EM_CEn HIGH までの時間 | | T/2 – 5 | - | - | ns |
| Taddrv | EM_Addr 有効から EM_Clock HIGH までの時間 | | 5 | - | - | ns |
| Taddriv | EM_Clock HIGH から EM_Addr 無効まで | | T/2 – 5 | - | - | ns |
| Tweld | EM_WEn HIGH から EM_Clock HIGH までの時間 | | 5 | - | - | ns |
| Twehd | EM_Clock HIGH から EM_WEn HIGH までの時間 | | T/2 – 5 | - | - | ns |
| Tds | EM_Clock HIGH の前のデータ有効時間 | | 5 | - | - | ns |
| Tdh | EM_Clock HIGH の後のデータ有効時間 | | T | - | - | ns |
| Tadscld | EM_ADSCn LOW から EM_Clock HIGH までの時間 | | 5 | - | - | ns |
| Tadschd | EM_Clock HIGH から EM_ADSCn HIGH までの時間 | | T/2 – 5 | - | - | ns |

# コンポーネントの変更

バージョン 1.0 は EMIF コンポーネントの最初のリリースです。

Copyright © 2005-2012 Cypress Semiconductor Corporation 本文書に記載される情報は、予告なく変更される場合があります。Cypress Semiconductor Corporation は、サイプレス製品に組み込まれた回路以外のいかなる回路を使用することに対しても一切の責任を負いません。特許又はその他の権限下で、ライセンスを譲渡又は暗示することもありません。サイプレス製品は、サイプレスとの書面による合意に基づくものでない限り、医療、生命維持、救命、重要な管理、又は安全の用途のために仕様することを保証するものではなく、また使用することを意図したものでもありません。さらにサイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことを合理的に予想される、生命維持システムの重要なコンポーネンツとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

PSoC Designer™ 及び Programmable System-on-Chip™は、Cypress Semiconductor Corp.の商標、PSoC®は同社の登録商標です。本文書で言及するその他全ての商標又は登録商標は各社の所有物です。

全てのソースコード(ソフトウェア及び/又はファームウェア)は Cypress Semiconductor Corporation (以下「サイプレス」)が所有し、全世界(米国及びその他の国)の特許権保護、米国の著作権法並びに国際協定の条項により保護され、かつそれらに従います。サイプレスが本書面によるライセンシーに付与するライセンスは、個人的、非独占的かつ譲渡不能のライセンスであって、適用される契約で指定されたサイプレスの集積回路と併用されるライセンシーの製品のみをサポートするカスタムソフトウェア及び/又はカスタムファームウェアを作成する目的に限って、サイプレスのソースコードの派生著作物を複製、使用、変更、そして作成するためのライセンス、並びにサイプレスのソースコード及び派生著作物をコンパイルするためのライセンスです。上記で指定された場合を除き、サイプレスの書面による明示的な許可なくして本ソースコードを複製、変更、変換、コンパイル、又は表示することは全て禁止されます。

免責条項:サイプレスは、明示的又は黙示的を問わず、本資料に関するいかなる種類の保証も行いません。これには、商品性又は特定目的への適合性の黙示的な保証が含まれますが、これに限定されません。サイプレスは、本文書に記載される資料に対して今後予告なく変更を加える権利を留保します。サイプレスは、本文書に記載されるいかなる製品又は回路を適用又は使用したことによって生ずるいかなる責任も負いません。サイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことが合理的に予想される生命維持システムの重要なコンポーネンツとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

ソフトウェアの使用は、適用されるサイプレスソフトウェアライセンス契約によって制限され、かつ制約される場合があります。