AN64574

シリアル ペリフェラル インターフェース (SPI) nvSRAM を使用した設計

著者: Shivendra Singh
関連プロジェクト: あり
関連製品ファミリ: CY14C101PA/QxA, CY14B101PA/QxA, CY14E101PA/QxA
ソフトウェア バージョン: PSoC® Creator™ – 3.0 またはそれ以降
PSoC Designer™ – 5.2 またはそれ以降
関連アプリケーション ノート: なし

サイプレスのシリアル ペリフェラル インタフェース (SPI) nvSRAM は、0 サイクル遅延書き込み動作と無限の SRAM 書き込み回数を提供する高性能の不揮発性シリアル メモリです。SPI nvSRAM は、スレーブ SPI デバイスで、システムにおける nvSRAM にアクセスするための SPI マスター コントローラーを必要とします。このアプリケーション ノートは、ユーザーが SPI nvSRAM を使用して設計するために設計上の重要な考慮事項とファームウェア設計のヒントをいくつか提供します。PSoC1 用の関連プロジェクトおよび PSoC 3 と PSoC 4 用のライブラリ コンポーネントに関連付けられているプロジェクトも、標準的な SPI マスター コントローラーによる SPI nvSRAM へのアクセスを説明するプロジェクト例として提供されます。

目次

はじめに 1
SPI nvSRAM のコンフィギュレーション 2
　入力ピンのコンフィギュレーション 4
　ピン配置 5
SPI 動作モード 5
SPI nvSRAM オペコード 6
　SPI nvSRAM におけるアドレッシング 8
nvSRAM 動作 9
　ステータス レジスタ動作 9
　nvSRAM における SRAM 書き込み／読み出し動作 10
プロジェクト例の設定 12
まとめ 13
ワールドワイドな販売と設計サポート 15

はじめに

サイプレスの nvSRAM は、高速な SRAM セルと不揮発性セルを単一の nvSRAM セルに統合します。通常の動作モードでは、すべての読み出しおよび書き込みは nvSRAM の SRAM 部から読み出し、SRAM 部に書き込みが直接行われます。これは、EEPROM、フラッシュ、FRAM、MRAM やバッテリ バックアップ SRAM などの既存の不揮発性メモリ技術に比べてより高速な書き込みと読み出しを提供します。システムの電力喪失が発生した場合、SRAM からのデータは、デバイス V_{CAP} ピンに接続される小さなコンデンサに格納されているエネルギーを使用することで、不揮発性セルに自動的に転送されます。次回の電源投入サイクル中に、不揮発性セルからのデータは、SRAM アレイに自動的にリコールされ、ユーザーに示されます。nvSRAM の V_{CAP} ピンへ接続されるコンデンサは通常動作中に充電されます。

nvSRAM は、その不揮発性セルに 100 万回のアクセス可能回数を指定します。nvSRAM のアクセス可能回数は、データ転送が STORE 動作中に、SRAM セルから不揮発性セルに行われる時にのみ費やされます。nvSRAM の不揮発性 STORE は、デバイスの電源が所定の閾値レベル (V_{SWITCH}) の下に落ちた時に自動的に開始されるか、または、要望に応じてオペコードかハードウェア ($\overline{HSB}$) ピンを介して開始されます。SRAM セルは書き込みと読み出し動作に無限のアクセス可能回数を提供するため、nvSRAM は通常の動作中に書き込み回数を一切費やしません。不揮発性 STORE は、システムの電源障害が検出され、データを不揮発性セルの中に安全に移動する必要がある場合にのみ行われます。これは、nvSRAM のアクセス可能サイクルは、(いずれのリアルタイム アプリケーションでも 100 万回に達しそうもない) システムの電源障害またはシステム シャットダウン イベントの合計数に相当することを意味します。

SPI nvSRAM は、業界標準の 8 ピン SOIC と 16 ピン SOIC パッケージにて高速で低消費電力のシリアル nvSRAM を提供します。同じことを実行するには数十ミリ秒を要する EEPROM やフラッシュ メモリに対して、nvSRAM により、数十マイクロ秒で数百バイトを書き込むことが可能になります。電源喪失が発生した場合にランタイムの重要な情報を瞬時に保存することを必要とするデータロギング アプリケーションが多く有ります。この重要な情報は、コントローラーのランタイム実行状態、またはスクラッチ パッド データ、パラメーター設定、およびコントローラーによって測定される他の環境変数を含みます。

このアプリケーション ノートでは、SPI nvSRAM の接続およびすべての基準 SPI マスター コントローラーに適用可能な機能を詳しく説明します。このアプリケーション ノートにおいて提供されるハードウェアの推奨事項は、必須事項ではありませんが、適用すれば、より堅牢な全体デザインを実現します。システム レベルで SPI nvSRAM の動作を説明するために、タイミング図および PSoC 1 ベースの擬似コードを使用して幾つかのオペコードを説明します。

このアプリケーション ノートは、次のトピックを取り扱います。

- SPI nvSRAM の接続
- SPI 動作モード
- nvSRAM 動作

PSoC 1 ベースのプロジェクト (関連プロジェクト 1) と PSoC 3、PSoC 4 nvSRAM SPI ライブラリ コンポーネント (関連プロジェクト 2 と関連プロジェクト 3) は、このアプリケーション ノートと一緒に添付されます。

SPI nvSRAM のコンフィギュレーション

次の表に示すように、サイプレスは、異なるコンフィギュレーションおよびパッケージ オプションで、SPI nvSRAM をサポートします。

表 1. SPI nvSRAM のコンフィギュレーション

nvSRAM 型番	動作電圧 (Typ)	パッケージ	$\overline{WP}$ ピン	V_{CAP} ピン／AutoStore	$\overline{HSB}$ ピン／ハードウェア ストア	RTC
CY14CXXXQ1A	2.5 V	8 ピン SOIC	有	無／無	無／無	RTC 無し
CY14BXXXQ1A	3.0V					
CY14EXXXQ1A	5.0V					
CY14CXXXQ2A	2.5V	8 ピン SOIC	無	有／有	無／無	RTC 無し
CY14BXXXQ2A	3.0V					
CY14EXXXQ2A	5.0V					
CY14CXXXQ3A	2.5V	16 ピン SOIC	有	有／有	有	RTC 無し
CY14BXXXQ3A	3.0V					
CY14EXXXQ3A	5.0V					
CY14CXXXPA	2.5V	16 ピン SOIC	有	有／有	有	RTC
CY14BXXXPA	3.0V					
CY14EXXXPA	5.0V					

表 1 の「XXX」は nvSRAM 型番に容量オプションを提供するための領域を示します。XXX = 064 は 64 K ビット; XXX=256 は 256K ビット; XXX=512 は 512K ビット; XXX=101 は 1 M ビット; XXX=102 は 2M ビットの nvSRAM 容量。

SPI ホスト コントローラーと NVSRAM デバイス間の接続は、nvSRAM デバイスのコンフィギュレーションとパッケージのオプションによって異なります。図 2～図 4 は、異なるコンフィギュレーションとパッケージのオプションにおける SPI SRAM の詳細な回路図の接続を示します。SPI ホスト コントローラーと SPI nvSRAM 間のハードウェア接続は、特定のコンフィギュレーションとパッケージ オプション用のすべての容量にわたって同一のままです。

SPI nvSRAM デバイスの一般的なシステム レベルのコンフィギュレーションは、ページ 3 の図 1 に示されます。専用 SPI バスを持たないマイクロコントローラーでは、I／O 汎用ポートが接続用に使用されることもあります。

図 1. 一般的 SPI nvSRAM の接続

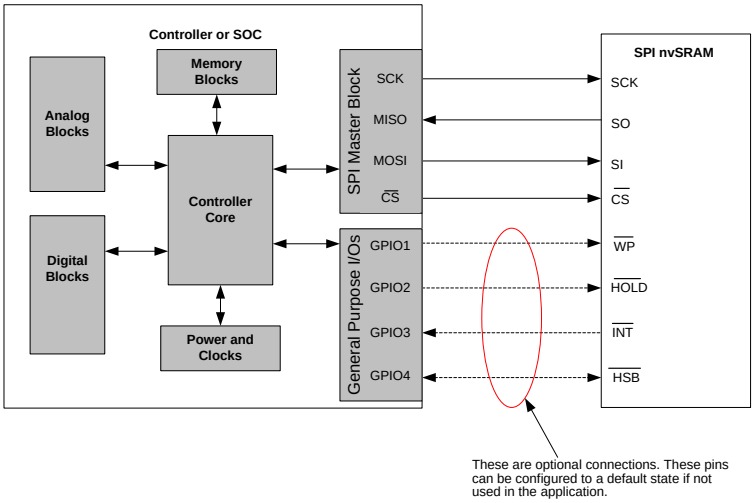


図 2. コントローラー付きの 8 ピン SPI nvSRAM インターフェース (V_{CAP} 無し)

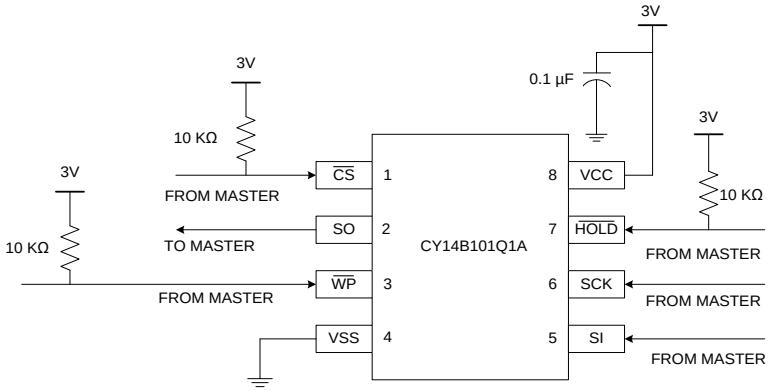


図 3. コントローラー付きの 8 ピン SPI nvSRAM インターフェース (V_{CAP} 付き)

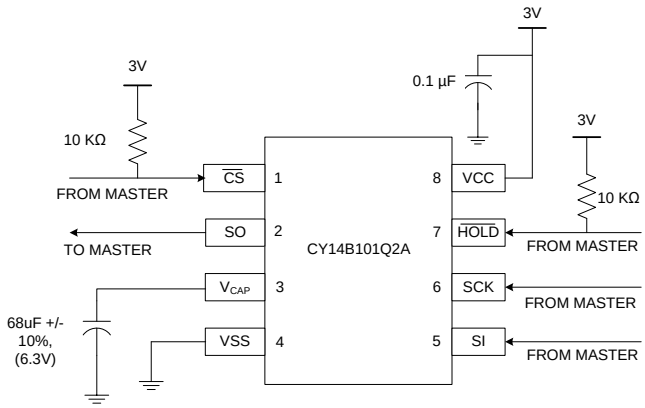
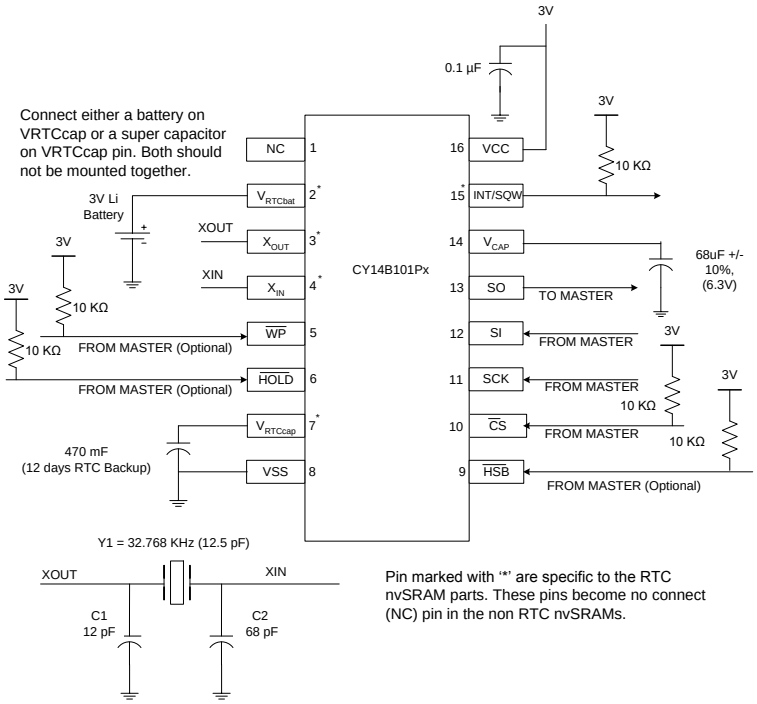


図 4. コントローラー付きの 16 ピン (RTC) SPI nvSRAM インターフェース



入力ピンのコンフィギュレーション

SPI nvSRAM は、デバイスの正常な動作のために、固定された論理状態 (HIGH または LOW) に適切にバイアスをかける必要がある制御入力ピンを多く持ちます。ピンが開放されたままの場合、高いスタンバイ電流を引き起こす中間レベル、または LOW あるいは HIGH 論理レベルに開放することが考えられます。信号が進む方向は、システムのノイズ、容量結合やリークなどのいくつかの要因に依存します。このため、入力回路からみえるレベルは比較的にランダムで、動作中に変わることもあります。このような予測できない入力レベルはデバイス動作に著しく影響する可能性があります。そのため、未使用の入力ピンは、アクティブ LOW 入力に対して HIGH などの適切な論理レベルに常に接続される必要があります。10kΩ の抵抗を、未使用の入力ピンをプルアップまたはプルダウンするのに使用することができます。

HOLDピン: SPI nvSRAM はアクティブ LOW 入力である HOLDピンを備え、ユーザーが途中でクロックを一時停止させ、動作中の通信を一時停止させることを可能とします。このピンが LOW に開放された場合、デバイスはもはや受信したクロック パルスに反応せず、通信が中断させられ、データが損失または破壊される恐れが有ります。使用されない場合、このピンは、これらの状態でノイズによる望ましくない事象を回避するために、10kΩ の抵抗でプルアップをしなければなりません。

CS ピン: マイクロコントローラーは常に、通常動作中にチップセレクト ピン (CS) を駆動する必要があります。これは、マイクロコントローラーの電源切断または電源投入時に開放される可能性があります。このピンは、これらの状態でノイズによる望ましくないコマンドを避けるために、10kΩ のプルアップ抵抗を持つ必要があります。

WP ピン: 書き込み防止 (WP)は、このピンを外部で LOW レベルに引き、メイン メモリとステータス レジスタへの書き込みを防止するために使用されるアクティブ LOW 入力信号です。ステータス レジスタの WPEN ビットは、WP ピンの機能を決定します。WPEN ビットが「1」にセットされた場合、WP ピンの制御が有効になります;「0」にセットされた場合、WPピンが無効になります。このピンは、これらの状態でノイズによる望ましくない事象を避けるために、10kΩ のプルアップ抵抗を持つ必要があります。

HSB ピン: HSB ピンは nvSRAM の双方向ピンです。それは出力として、不揮発性 STORE 動作中に nvSRAM のレディ／ビジー状態を提供します。nvSRAM がレディ状態にある時、ホスト コントローラーはデバイスのすべての機能にアクセスすることが可能です。nvSRAM がビジー状態にある時、読み出しステータス レジスタ以外のすべてのコマンドは禁止され、デバイスは、読み出しステータス レジスタ コマンドを通じて取得することができるステータス レジスタのRDYビットをセットします。入力ピンとして、HSB ピンは、コントローラーによりハードウェア STORE を外部から起動するために使用されます。このピンは、GPIO に接続されない場合、開放されたままにすることができます。内部弱プルアップは、通常動作時にHSBをHIGH に保つために設けられます。HSBがコントローラー

GPIO によって外部で制御された場合、このラインでノイズによる望ましくないトリガーを避けるために、外部の 10kΩ プルアップ抵抗を使用する必要があります。

V_{CAP}: V_{CAP} に接続されるコンデンサは電源喪失時に nvSRAM へ電源を供給して、SRAM から不揮発性素子にデータを記憶します。通常動作中に、デバイスは、コンデンサを充電するのに V_{CC} から電流を引き込みます。充電された電力はチップが一回の STORE 処理を実行するのに使用されます。V_{CC} ピンの電圧が V_{SWITCH} を下回ると、デバイスは V_{CC} と V_{CAP} ピンの接続を自動的に切ります。STORE 処理は、V_{CAP} コンデンサから供給される電力で起動されます。

正常な AutoStore 処理のために、V_{CAP} ピンに適切な値のコンデンサを常に接続します。選択したコンデンサの値は、デバイスのデータシートに規定された範囲内に入る必要があります。コンデンサを不適切に選択すると、デバイスの誤動作が発生する可能性があります。nvSRAM 製品用コンデンサ選択のガイドラインの詳細については、アプリケーション ノート「サイプレスの nvSRAM 用のストレージ コンデンサ オプション – AN43593」を参照してください。

ピン配置

以下のピンは、RTC 機能特有のものです。RTC 機能が使用されない場合、これらのピンは、基板上で開放されたままにすることができます。

INT ピン: これは RTC 部の出力ピンです。INT 出力は、RTC の nvSRAM デバイスの異なる機能を引き出すために多重化されます。RTC レジスタの設定と nvSRAM で定義された優先順位に応じて、INT ピンは、アラーム ステータス、ウォッチドッグ タイマーの状態、較正クロック出力と方形波出力のいずれかを引き出すために設定することができます。INT ピンは、割り込みステータス／制御レジスタ内の H/L ビットを設定することにより設定可能なドライバ出力です。H/L ビットが「1」にセットすると、INT ピンが HIGH アクティブになり、ドライバ モードがプッシュ プルになります。H/L ビットが「0」にセットされると、INT ピンはアクティブ LOW のオープン ドレイン出力であるため、論理 HIGH 状態に駆動するのに外部のプルアップ抵抗が必要です。従って、INT をアクティブ LOW モードで使用している間、INT ピンは 10kΩ プルアップ抵抗を使用することで V_{CC} にプルアップする必要があります。

V_{RTCbat} と V_{RTCcap} ピン: これらのピンは、nvSRAM デバイスの RTC 回路にバックアップ電源を供給するために使用され、システム電源 (V_{CC}) が落ちた時に RTC クロックの実行を維持します。V_{RTCbat} と V_{RTCcap} ピンは、V_{RTCbat} の非充電式バッテリまたは V_{RTCcap} ピンのスーパー コンデンサに接続する必要があります。使用されない場合、これらのピンが開放されたままにする必要があります。

注 : V_{RTCcap} ピンは通常の動作中にそれに接続されているスーパー コンデンサを充電するために使用されるため、このピンを直接 V_{SS} に短絡することはできません。このため、V_{RTCcap} をグランドに直接接続する (V_{SS}) と nvSRAM から過剰な電流が流れる可能性があります。

nvSRAM RTC の設計ガイドラインとベスト プラクティスについては、アプリケーション ノート「不揮発性スタティック ランダム アクセス メモリ (nvSRAM) リアルタイム クロック (RTC) の設計ガイドラインおよびベストプラクティス – AN61546」を参照してください。

SPI 動作モード

SPI nvSRAM は、SPI 通信の開始時に SPI マスターが設定するクロック極性 (CPOL) やクロック位相 (CPHA) に依存する SPI モード 0 (CPOL = 0、CPHA = 0) および SPI モード 3 (CPOL = 1、CPHA = 1) をサポートします。表 2 は、SPI マスターとスレーブによる MOSI と MISO ラインでの SPI クロッキングとデータ駆動に対して、すべての SPI モードをまとめたものです。nvSRAM の SPI モードは、マスター コントローラーの SPI モードに応じて自動的に設定されます。

表 2. SPI 動作モード

	モード 0 (CPOL=0;CPHA=0)	モード 1 (CPOL=0;CPHA=1)	モード 2 (CPOL=1;CPHA=0)	モード 3 (CPOL=1;CPHA=1)
SPI クロック (SCK) 開始論理レベル	LOW	LOW	HIGH	HIGH
MOSI 上の nvSRAM によりラッチされるデータ	SCK 立ち上りエッジ (↑)	SCK 立ち下がりエッジ (↓)	SCK 立ち下がりエッジ (↓)	SCK 立ち上りエッジ (↑)
MISO の nvSRAM により駆動されるデータ	SCK 立ち下がりエッジ (↓)	SCK 立ち上りエッジ (↑)	SCK 立ち上りエッジ (↑)	SCK 立ち下がりエッジ (↓)
SPI nvSRAM サポート	有	無	無	有

SPI nvSRAM オペコード

すべての SPI オペコード、アドレスおよびデータは 8 ビットデータ転送と考えられているため、すべての内部動作は実際にバイト幅です。すべてのトランザクションは$\overline{CS}$LOW で発生します。アドレス、制御、および入力データは SI ピンを介して入力され、出力データは SO ピンを介して出力されます。オペコードは、デバイスに対する制御権を提供します。SPI nvSRAM は、すべての読み出しおよび書き込み動作用に業界標準のオペコードをサポートします。また、nvSRAM 特有の NV 動作と高速 (104MHz) の SPI アクセス用の特別なオペコードをサポートします。SPI nvSRAM の特定の動作ごとに唯一なオペコードが割り当てられます。SPI nvSRAM 命令一覧とそれぞれのオペコードは表 3 に定義されます。

表 3. SPI nvSRAM オペコード

命令カテゴリ	命令名	オペレーション コード	CY14B101P/Qx		CY14C101PA/QxA、CY14B101PA/QxA、CY14E101PA/QxA	
			オペコード サポート	SPI 周波数	オペコード サポート	SPI 周波数
状態レジスタ制御命令	WREN	06H (0000 0110)	√	最大 40MHz	√	最大 104MHz
	WRDI	04H (0000 0100)	√	最大 40MHz	√	最大 104MHz
	RDSR	05H (0000 0101)	√	最大 40MHz	√	最大 40MHz
	FAST_RDSR	09H (0000 1001)	X	該当なし	√	最大 104MHz
	WRSR	01H (0000 0001)	√	最大 40MHz	√	最大 104MHz
SRAM READ および WRITE 命令	読み出し	03H (0000 0011)	√	最大 40MHz	√	最大 40MHz
	FAST_READ	0BH (00001011)	X	該当なし	√	最大 104MHz
	書き込み	02H (0000 0010)	√	最大 40MHz	√	最大 104MHz
RTC 命令 [注 1]	WRTC	12H (0001 0010)	√	最大 40MHz	√	最大 104MHz
	RDRTC	13H (0001 0011)	√	最大 25MHz	√	最大 25MHz
	FAST_RDRTC	1DH (00011101)	X	該当なし	√	最大 104MHz
NV 注 [注 2]	STORE	3CH (0011 1100)	√	最大 40MHz	√	最大 104MHz
	RECALL	60H (0110 0000)	√	最大 40MHz	√	最大 104MHz
	ASENB	59H (0101 1001)	√	最大 40MHz	√	最大 104MHz
	ASDISB	19H (0001 1001)	√	最大 40MHz	√	最大 104MHz
スリープ	SLEEP	B9H (1011 1001)	X	該当なし	√	最大 104MHz
連番	WRSN	C2H (1100 0010)	X	該当なし	√	最大 104MHz
	RDSN	C3H (1100 0011)	X	該当なし	√	最大 40MHz
	FAST_RDSN	C9H (1100 1001)	X	該当なし	√	最大 104MHz
デバイス ID 読み出し	RDID	9FH (1001 1111)	X	該当なし	√	最大 40MHz
	FAST_RDID	99H (1001 1001)	X	該当なし	√	最大 104MHz

注 1 : RTC 命令は RTC nvSRAM デバイス (CY14C101P/PA、CY14B101P/PA と CY14E101P/PA) 特有のものです。これらのコマンドは非 RTC デバイスには適用されません。

注 2 : これらのコマンドは、NV 動作実行のための nvSRAM 製品の固有のものです。

表 4 は、正常な動作に必要な関連データ バイトを伴ったオペコードを説明します。

表 4. SPI nvSRAM データ流れ形式

命令名	オペコード	SI におけるマスター転送	SO における nvSRAM 転送	コメント
WREN	06H	06h	-	このコマンドはステータス レジスタの WEN ビットをセットする
WRDI	04h	04h	-	ステータス レジスタの WEB ビット (セットされた場合) をクリアする
RDSR	05H	05h	StatusReg_Data	ステータス レジスタの内容を読み出す
FAST_RDSR	09H	09H、Dummy_Byte	StatusReg_Data	
WRSR	01H	01H、StatusReg_Data	-	WEN ビットはステータス レジスタに書き込む前にセットされる必要がある WEN は、$\overline{CS}$が HIGH になると、クリアされる
READ [注 3]	03H	03H、Add1、Add2、Add3	Data1、Data2、Data3、...、DataN	データ長 1～N を読み出す。N は整数の値のいずれかである。nvSRAM の内部アドレス カウンターは自動的に 1 ずつインクリメントする nvSRAM のカウントが最大のアドレッシング限界に達すると、開始アドレスにロールオーバーして、そこからデータを読み出し続ける$\overline{CS}$が HIGH になると、読み出しが終了する
FAST_READ [注 3]	0BH	0BH、Add1、Add2、Add3、Dummy_Byte	Data1、Data2、Data3、...、DataN	
WRITE [注 3]	02H	02H、Add1、Add2、Add3、Data1、Data2、Data3、...、DataN	–	WEN ビットは、nvSRAM メモリに書き込む前にセットされる必要がある。データ長 1～N を書き込む。N は整数の値のいずれかである。nvSRAM の内部アドレス カウンターは自動的に 1 ずつインクリメントする。nvSRAM のカウントが最大のアドレッシング限界に達すると、開始アドレスにロールオーバーして、以前書き込まれたデータをオーバラップすることでそこからデータを読み出し続ける。ファームウェアは、大量の書込み動作中にメモリ カウンターがロールオーバーすることにより上書きされるデータを処理しなければならない。$\overline{CS}$が HIGH になると、書き込みが終了する
WRTC [注 4]	12H	12H、Addr Data	–	「W」ビットは RTC フラグ レジスタにて「1」にセットし、WEN ビットはステータス レジスタにて「1」にセットする必要がある。$\overline{CS}$が HIGH になると、WEN はクリアされる
RDRTC [注 4]	13H	13H、Addr	Data	
FAST_RDRTC [注 4]	1DH	1DH、Addr Dummy_Byte	Data	
STORE	3CH	3CH	–	
RECALL	60H	60H	–	
ASENB	59H	59H	–	
ASDISB	19H	19H	–	
SLEEP	B9H	B9H	–	WEN ビットは、スリープ コマンドを起動する前にセットされる必要がある。$\overline{CS}$が HIGH になると、デバイスがスリープ コマンドを登録する。WEN は、$\overline{CS}$が HIGH になると、クリアされる

命令名	オペコード	SI におけるマスター転送	SO における nvSRAM 転送	コメント
WRSN	C2H	C2H、Data1、Data2、...、Data8	–	WEN ビットがセットされる必要がある。8 バイト連番を書き込む。$\overline{CS}$が HIGH になると、WEN はクリアされる
RDSN	C3H	C3H	Data1、Data2、Data3、...、Data8	8 バイト連番を読み出す
FAST_RDSN	C9H	C9H、Dummy_Byte	Data1、Data2、Data3、...、Data8	8 バイト連番を読み出す
RDID	9FH	9FH	Data1、Data2、Data3、Data4	デバイス ID 4 バイト
FAST_RDID	99H	99H、Dummy_Byte	Data1、Data2、Data3、Data4	デバイス ID 4 バイト

注 3：1Mb とそれより大きな容量の SPI nvSRAM は 3 バイトアドレスを使用します。より小さな容量の nvSRAM (512 キロビット以下) は 2 バイトアドレスを使用します。

注 4: RTC 命令は RTC nvSRAM デバイス (CY14C101PA、CY14B101PA と CY14E101PA) 特有のものです。これらのコマンドは非 RTC デバイスには適用されません。

SPI nvSRAM におけるアドレッシング

SPI ホスト コントローラーは、バイト単位で nvSRAM SPI と通信し、常に、バイト通信中に、最初のクロックサイクルで最上位ビットを、8 クロック サイクル目で最下位ビットを送信します。これは、コマンド、アドレスとデータ バイトを含むすべての SPI 通信のために良い状態を保持します。同様に、SPI nvSRAM が、読み出し動作中にデータ バイトを送信する時、データ バイト送信中、常に最初に最上位ビットを、最後に最下位ビットを送信します。図 5 は、SPI マスターによって 3 アドレスバイトを送信している間、SPI MOSI (マスター アウト スレーブ イン) ラインを介して送信されるアドレス ビットの例を示します。

図 5. SPI nvSRAM におけるアドレス ビット送信

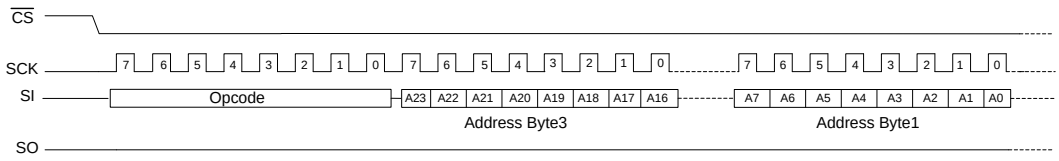


図 6 は異なる nvSRAM 容量用のアドレッシング方式を示します。A0 はアドレスの最下位ビット (LS ビット) です。

図 6. SPI nvSRAM オペコードおよびアドレッシング

Density	Opcode								Address Byte3 (MSB)								Address Byte2 (Intermediate Byte)								Address Byte1 (LSB)							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
256 Kbit	op	op	op	op	op	op	op	op	Not Applicable (2 Byte Addressing Only)								0	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
512 Kbit	op	op	op	op	op	op	op	op	Not Applicable (2 Byte Addressing Only)								A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1 Mbit	op	op	op	op	op	op	op	op	0	0	0	0	0	0	0	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
2 Mbit	op	op	op	op	op	op	op	op	0	0	0	0	0	0	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

最上位アドレス バイト (MSB) の未使用のビットは、ビットに影響せず、nvSRAM はこれらのビットの状態を無視します。ただし、ファームウェアで未使用のアドレス ビットを「0」にをセットするのは良い方法です。このアプローチにより、同じソケットでより大きな容量のデバイスに移行している間、ファームウェアを簡単にアップグレードできるようになります。

nvSRAM 動作

このセクションでは、タイミング図と PSoC1 特有の擬似コードと共に nvSRAM 動作について説明します。接頭辞「SPIM_SPIM_」で始まるすべての関数は、PSoC 1 固有の関数です。それらの表現と実装は、SPI マスターとして使用されるコントローラーに対して変更対象になります。

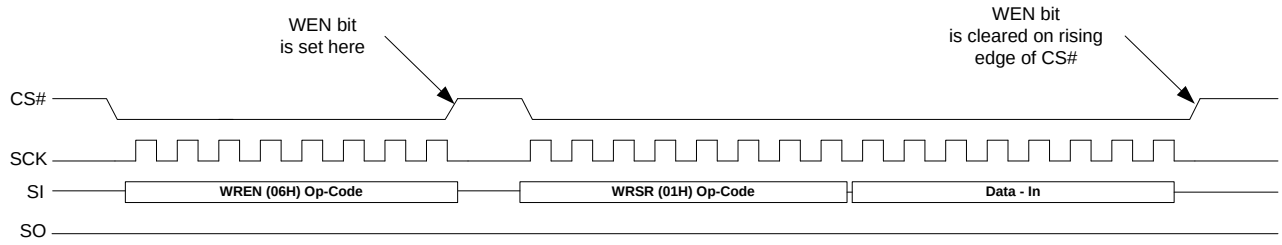
この節の範囲は、SPI nvSRAM に定義されるすべてのオペコードを含めていません。各オペコードの詳細については、デバイスのデータシートを参照してください。

ステータス レジスタ動作

書き込みステータス レジスタ: ステータス レジスタに書き込むためには、書き込む対象になるデータ バイトが続くステータス レジスタ書き込みオペコード (WRSR) コマンドを送信する必要があります。

- WREN オペコードを送信することにより、WEN ビットをセットします。
- ステータス レジスタへ書き込まれるデータ バイトが続く書き込みステータス レジスタ オペコード (WRSR) を送信します。ステータス レジスタの読み出し専用ビットが WRSR 動作から影響を受けないことに注意してください。ステータス レジスタの詳細については、デバイスのデータシートを参照してください。図 7 は、ステータス レジスタへの書き込みタイミング図を示します。

図 7. ステータス レジスタへの書き込み




```
/*******************PSoC1 Based Pseudo Code for Status Register Write***************/
#define CS_HI Port0_0(1)
#define CS_LO Port0_0(0)

void WRSR(BYTE data1) //User Define Function
{
    BYTE WREN=0x06;
    BYTE OPCODEWRSR=0x01;

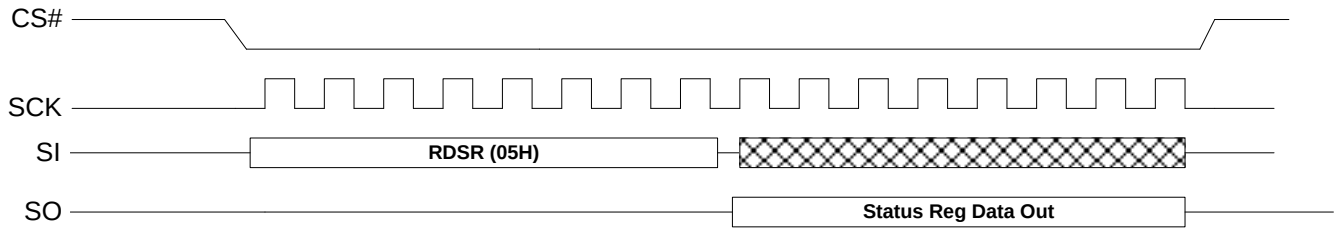
    CS_LO;
    while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
    SPIM_SendTxData(WREN); //This will set WEN='1'
    CS_HI;

    CS_LO;
    while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
    SPIM_SendTxData(OPCODEWRSR);//Send OPCODE for Status Register write

    while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
    SPIM_SendTxData(data1); //Send the data to be written into Status Register
    CS_HI;
}
```

読み出しステータス レジスタ: ステータス レジスタの内容を読み出すには、読み出しステータス レジスタ オペコード (RDSR) を送信する必要があります。その後、nvSRAM が SO ライン上のステータス レジスタの内容を送信し始めます。SPI nvSRAM は、チップ セレクト ピンを LOW にプルすることで、選択されたままにする必要があり、SPI クロックは、RDSR コマンドが続くステータス レジスタの内容を読み出すために利用可能でなければなりません。図 8 は、ステータス レジスタからの読み出しタイミング図を示します。

図 8. ステータス レジスタからの読み出し




```
/******** PSoC1 Based Pseudo Code for Status Register Read************************/
BYTE RDSR ()// User Define Function

{
    BYTE OPCODERDSR=0x05;
    BYTE data;

    CS_LO;
    while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
    SPIM_SendTxData(OPCODERDSR); //Send instruction

    while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
    SPIM_SendTxData(0x01); //Dummy write to generate CLK and read data

    while(!(SPIM_bReadStatus() & SPIM_SPIM_RX_BUFFER_FULL));
    data = SPIM_bReadRxData(); //Read Byte from Status Register
    CS_HI;

    return(data);
}
```

nvSRAM における SRAM 書き込み／読み出し動作

SRAM 書き込み: SPI nvSRAM の SRAM アレイに書き込むために、コントローラーは次のように書き込みコマンドを起動する必要があります。

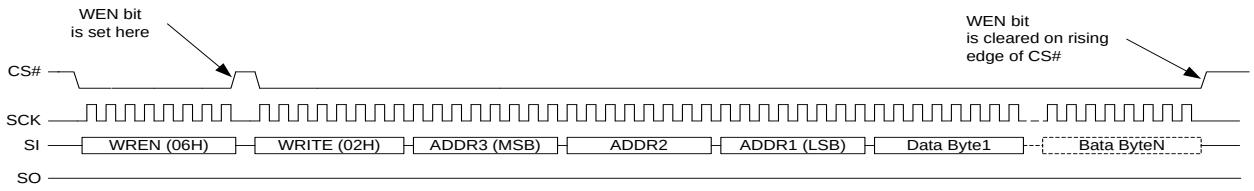
- 書き込みイネーブル ラッチ (WEN) ビットを設定するために、WREN オペコードを送信します。
- 書き込みオペコードを送信します。
- 最上位アドレス バイトを送信します。
- 中間アドレス バイト (3 バイト アドレッシングで) を送信します。
- 下位アドレス バイトを送信します。
- データ バイト／バイトを送信します。

nvSRAM へのいかなる書き込みコマンドも、書き込みイネーブル (WREN) 命令により先行される必要があります。デバイスが書き込みイネーブル(WEN=「0」) になっていない場合は、それは書き込み命令を無視し、$\overline{CS}$が HIGH 状態になるとスタンバイ状態に戻ります。シリアル通信を再開させるには、新しい$\overline{CS}$立ち下がりエッジが必要です。

書き込み命令 (WRSR、WRITE または WRTC)、または、nvSRAM の特別な命令 (STORE、RECALL、ASENB、ASDISB) のいずれかを終了した後、ステータス レジスタの WEN ビットは、書き込みサイクルの終わりに$\overline{CS}$ の立ち上がりエッジで「0」にクリアされます。これにより、不注意な書き込みが防止されます。

また、WREN といかなる WRITE 命令の間でステータス レジスタ (RDSR オペコード) を読み取っても、WEN ビットをクリアしないことに注意してください。WEN ビットが書き込む動作の前にセットされることを確認するために、WREN の直後にステータス レジスタを読み出すユーザーもいます。図 9 は、SRAM メモリへ書き込むためのタイミング図を示します。

図 9. SRAM への書き込み




```
/* PSoC1 Based Pseudo Code for nvSRAM write in burst mode. By sending tot_cnt =1, user
can write only 1 byte at a given address location*/

void nvSRAMBURSTWRITE(BYTE addr1, BYTE addr2, BYTE addr3, DWORD tot_cnt, BYTE*data) //
User Define Function
    {   BYTE WREN=0x06;
        BYTE OPCODEWRITE=0x02;
        DWORD count=0;

        CS_LO;
        while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
        SPIM_SendTxData(WREN); //Set WEN='1' prior to write
        CS_HI;

        CS_LO;
        while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
        SPIM_SendTxData(OPCODEWRITE); //Send OPCODE for Write into main memory
        while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
        SPIM_SendTxData(addr1); //Send MS Byte
        while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
        SPIM_SendTxData(addr2); //Send Intermediate Address Byte
        while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
        SPIM_SendTxData(addr3); //Send LS Byte of Address

        for(count=0; count< tot_cnt; count++)
        {
        while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
        SPIM_SendTxData(data[count+4]); //Byte written into main memory}
        CS_HI;
}
```

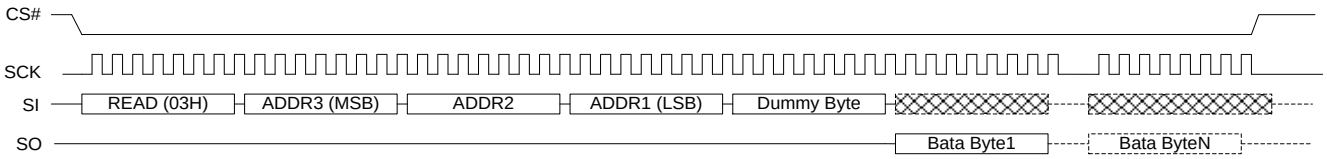
SRAM 読み込み: SPI nvSRAM の SRAM アレイから読み出すために、コントローラーは次の形式でオペコードとアドレスを送信する必要があります。

- 読み出しオペコードを送信します。
- 最上位アドレス バイトを送信します。
- 中間アドレス バイト (3 バイト アドレッシングで) を送信します。
- 下位アドレス バイトを送信します。

nvSRAM は、チップ セレクト信号 LOW をプルすることでデバイスが選択されたままにし、SPI クロックが存在する間、SO ライン上でデータを送信します。

SRAM 読み出しを開始するために、コントローラーは、読み出しアドレス バイトが続く読み出しオペコードを発行します。読み出しリクエストとアドレスを登録した後、nvSRAM は SO ピン上でデータを送信します。後続のデータ バイトには、単にデータ バイトを連続的にクロック出力しながら $\overline{CS}$を LOW に維持することでアクセスできます。これはバースト モード読み出しと呼ばれ、アドレスは、SPI nvSRAM デバイスによって自動的にインクリメントされます。$\overline{CS}$が HIGH にデアサートされると、データ出力が停止し、SO が高インピーダンス (HI-Z) 状態になります。図 10 は、SRAM から読み出すためのタイミング図を示します。

図 10 SRAM からの読み出し




```
/********************nvSRAM Read Burst Data*************************/
void  nvSRAMBURSTREAD(BYTE  addr1,  BYTE  addr2,  BYTE  addr3,  DWORD  tot_cnt,BYTE  *
readDataArr) User Define Function
    {
        BYTE readdata;
        BYTE data;
        BYTE OPCODEREAD=0x03;
        DWORD count=0;

        CS_LO;
        while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
        SPIM_SendTxData(OPCODEREAD); //Send Read Opcode
        while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
        SPIM_SendTxData(addr1); // Send MS Byte
        while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
        SPIM_SendTxData(addr2); // Send Intermediate Address Byte
        while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
        SPIM_SendTxData(addr3); // Send LS Byte of Address

        for(count=0; count<tot_cnt; count++)
        {
        while(!(SPIM_bReadStatus() & SPIM_SPIM_TX_BUFFER_EMPTY));
        SPIM_SendTxData(addr1); //Dummy write to generate CLK and read data
        while(!(SPIM_bReadStatus() & SPIM_SPIM_RX_BUFFER_FULL));
        readDataArr[count] = SPIM_bReadRxData();// Read data from nvSRAM
        }
        CS_HI;
}
```

プロジェクト例の設定

このアプリケーション ノートを伴う PSoC ベースのプロジェクトは、ターゲット アプリケーションに応じて変更することができます。PSoC のプログラム可能性と柔軟性により、SPI マスター機能に加えて、デバイスに他の機能を更に追加することが可能となります。このアプリケーションを伴うプロジェクトは、PSoC との SPI nvSRAM インターフェイスを示した1つの例のみです。

以下の設定は、添付のソース コードを変更せずに、実行するが必要です。表 5 はプロジェクト例のコネクションの詳細を示します。

- PSoC Designer™ – 5.2 またはそれ以降
- ハードウェア キット – CY8CKIT-001 (DVK1 開発基板)
- CY8C29 プロセッサ モジュール (CY8CKIT-008)
- プロジェクトを構築するために選択された PSoC 1 型番: CY8C29866-24AXI
- UART ボーレートの設定 – 19200 ボー／秒
- PSoC 1 コントローラーへの V_{DD} 電源 – 3.3 V

表 5. プロジェクト例の PSoC ポート コンフィギュレーション

nvSRAM 信号名	PSoC (マスター) 信号名	PSoC 1 I/O 割当て	信号方向
$\overline{CS}$	CS_n	Port0_1	PSoC 1 出力
SI	MOSI	Port0_1	PSoC 1 出力
SO	MISO	Port0_2	PSoC 1 入力
SCK	SCK	Port0_3	PSoC 1 出力
$\overline{WP}$	WP_n	Port0_4	PSoC 1 出力
$\overline{HOLD}$	HOLD_n	Port0_5	PSoC 1 出力
INT/SQW	INT	Port0_6	PSoC 1 入力
$\overline{HSB}$	HSB_n	Port0_7	PSoC 1 入力／出力 (プロジェクト例はこの PSoC 1 ピンを出力として設定する。それは、適切な駆動モード レジスタを設定することで出力／入力として動的に設定できる)
	UART TX	Port2_5	PSoC 1 出力 (UART 端末へ)
	UART RX	Port2_6	PSoC 1 入力 (UART 端末から)

まとめ

サイプレスの SPI nvSRAM は、すべての他の不揮発性 SPI メモリ製品と同様に、標準的 SPI アクセス プロトコルをサポートします。これによって、nvSRAM はすべての SPI マスター コントローラーとの互換性があり、システム開発サイクル時間が短くなります。すべての SPI オペコードは、nvSRAM 特有なものの一部を除いて、標準的 SPI メモリ製品オペコードと一致しています。これによって、SPI nvSRAM が、機能が同じ他のすべての不揮発性メモリ デバイスのドロップイン代替品となります。このアプリケーション ノートは、回路図、タイミング図およびコード例と共に、アプリケーションに SPI nvSRAM を設定する方法を示します。

改訂履歴

文書名: シリアル ペリフェラル インターフェース (SPI) nvSRAM を使用した設計 – AN64574

文書番号: 001-92721

版	ECN 番号	変更者	発行日	変更内容
**	4395629	HZEN	06/10/2014	これは英語版 001-64574 Rev. *G を翻訳した日本語版 001-92721 Rev. **です。
*A	4733695	HZEN	04/21/2015	これは英語版 001-64574 Rev. *H を翻訳した日本語版 001-92721 Rev. *A です。

ワールドワイドな販売と設計サポート

サイプレスは、世界中に事業所やソリューション センター、販売代理店を持っています。お客様の最寄りのオフィスについては、サイプレスのロケーション ページをご覧ください。

製品

車載用	cypress.com/go/automotive
クロック＆バッファ	cypress.com/go/clocks
インターフェース	cypress.com/go/interface
照明＆電源管理	cypress.com/go/powerpsoc cypress.com/go/plc
メモリ	cypress.com/go/memory
PSoC	cypress.com/go/psoc
タッチ センシング	cypress.com/go/touch
USB コントローラー	cypress.com/go/usb
無線／RF	cypress.com/go/wireless

PSoC®ソリューション

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

サイプレス開発者コミュニティ

コミュニティ | フォーラム | ブログ | ビデオ | トレーニング

テクニカル サポート

cypress.com/go/support

PSoC Designer はサイプレス セミコンダクタ社の商標であり、PSoC はサイプレス セミコンダクタ社の登録商標です。本書で言及するその他すべての商標または登録商標は、各社の所有物です。

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation、2010-2015. 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 (サイプレス セミコンダクタ社) 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一切の責任を負いません。サイプレス セミコンダクタ社は、特許またはその他の権利に基づくライセンスを譲渡することも、または含意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、または安全の用途のために使用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ト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。
このソースコード (ソフトウェアおよび／またはファームウェア) はサイプレス セミコンダクタ社 (以下「サイプレス」) が所有し、全世界の特許権保護 (米国およびその他の国)、米国の著作権法なら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り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り、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 ソフトウェアおよび／またはカスタム 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コピー、使用、変更そして作成するためのライセンス、ならびにサイプレスのソース コードおよ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、または表示することはすべて禁止します。
免責条項: サイプレスは、明示的または黙示的を問わず、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、商品性また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または回路を適用また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ト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。