

## サイプレスはインフィニオン テクノロジーズになりました

この表紙に続く文書には「サイプレス」と表記されていますが、これは同社が最初にこの製品を開発したからです。新規および既存のお客様いずれに対しても、引き続きインフィニオンがラインアップの一部として当該製品をご提供いたします。

## 文書の内容の継続性

下記製品がインフィニオンの製品ラインアップの一部として提供されたとしても、それを理由としてこの文書に変更が加わることはありません。今後も適宜改訂は行いますが、変更があった場合は文書の履歴ページでお知らせします。

## 注文時の部品番号の継続性

インフィニオンは既存の部品番号を引き続きサポートします。ご注文の際は、データシート記載の注文部品番号をこれまで通りご利用下さい。

## HyperRAM™の使用方法

著者: Nilesh Badodekar

関連製品ファミリ: S27KL0641 / S27KL0642 / S27KL0643

関連アプリケーションノート: [関連ドキュメント](#)を参照してください

AN226576 は、サイプレスの最新の高密度、高性能メモリを使用して設計するために必要な重要な概念の概要を説明し、システムで HyperRAM を使用する主な利点と一般的な使用例のシナリオをリストアップします。このアプリケーションノートは、市場をリードする SoC と同様なサイプレスの Traveo™ MCU を使用した HyperRAM を設計するためのリソースを提供します。

### 目次

1 はじめに.....	1	4.4 電力モード.....	9
2 HyperBus – 入門編.....	2	5 HyperRAM を使用した設計.....	11
3 Octal SPI (xSPI) – 入門編.....	4	6 HyperRAM – ピン数の少ない高性能システムメモリ.....	12
4 HyperRAM 製品の概要.....	6	7 関連ドキュメント.....	13
4.1 信号の説明.....	6	改訂履歴.....	14
4.2 一般的なシステム接続.....	7	セールス、ソリューションおよび法律情報.....	15
4.3 分散リフレッシュロジック.....	8		

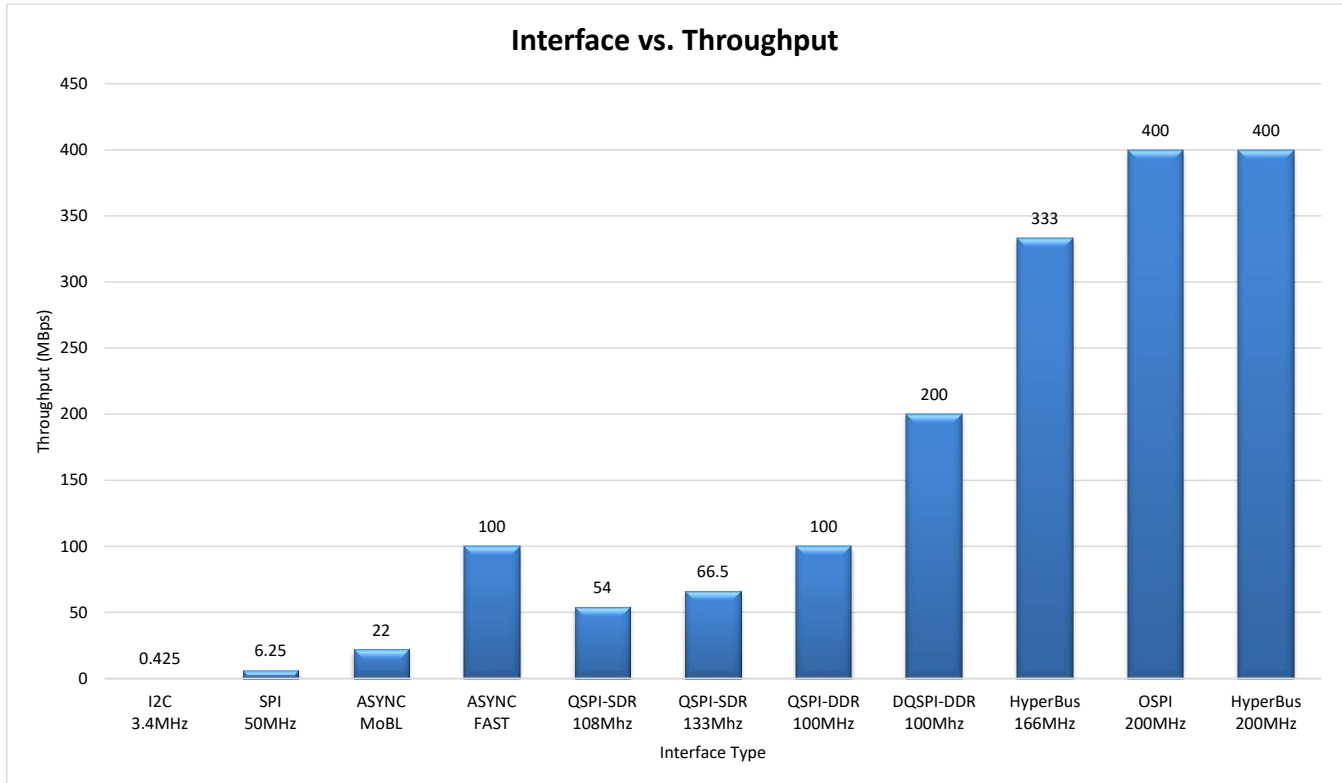
## 1 はじめに

HyperBus™ と OctalSPI (xSPI) HyperRAM™ は、どちらもサイプレスセミコンダクタがより幅広い HyperBus/xSPI メモリファミリの一部として導入した高性能 8 ビット幅のシリアルセルフリフレッシュ DRAM デバイスです。

シリアルインターフェースメモリの最大の利点は、メモリをホストコントローラに接続するために必要な信号の数が減ることであり、その結果、パッケージと多層 PCB のコストが削減されます。ただし、シリアルメモリは歴史的にデータスループットが低いか、またはランダムアクセス時間が長くなります。これは HyperBus や xSPI では緩和されており、クロック周波数レートの 2 倍のクロック周波数でソース同期データをキャプチャする高性能マルチ I/O DDR アーキテクチャを採用しています。

第 1 世代の HyperRAM デバイスは最大 166MHz の DDR クロック周波数をサポートしていましたが、第 2 世代のデバイスは現在 200MHz のクロック周波数をサポートしており、ピークデータ転送レートは 0.4GB/s です。図 1 に、異なるインターフェース間でのデータスループットの比較を示します。

図 1. インターフェース データ スループット



## 2 HyperBus – 入門編

HyperBus 製品は、高性能 HyperBus を使用して、ホストシステムのマスタと 1 つ以上のスレーブインターフェースを接続します。HyperBus は、マイクロプロセッサ、マイクロコントローラ、または ASIC デバイスをランダムアクセス NOR フラッシュメモリ、RAM、または周辺デバイスに接続するために使用されます。

HyperBus は、システムパフォーマンス、設計の容易さ、およびシステムコストの削減を強化しながら、パラレルおよびシリアルインターフェースメモリの両方のレガシー機能を利用するインターフェースです。

パラレルフラッシュと PSRAM は、長い間単純なインターフェースで高性能のランダムアクセスメモリの標準であり、組み込みシステムのコード実行とデータ保存に使用されてきました。ただし、パラレルインターフェースメモリには、45 以上の信号を含む個別の制御、アドレス、およびデータ接続を備えた多数の信号が必要です。一部のアドレス信号とデータ信号を多重化することで信号数を減らすパラレルインターフェースのバリエーションがありますが、それでも 20 以上の信号が含まれる場合があります。これらの数が多い信号数は高いデータスループットを提供しますが、ホストシステムプロセッサまたは ASIC 上の多くのコネクタによるコストと、信号配線の混雑に悩まされる多層 PCB による高コストを犠牲にします。

多くのシステムは、メモリへの接続に必要な信号数を減らすためにシリアルインターフェースメモリに移行しました。これにより、ホストシステム接続を他の機能でできるように解放し、パッケージと多層 PCB のコストを削減します。ただし、シリアルメモリは一般にデータスループットが低いか、ランダムアクセス時間が長い値です。このため、ランダムアクセスと高スループットのためにコードとデータを DRAM メモリに転送するだけのシリアルメモリの役割が必要になる場合があります。

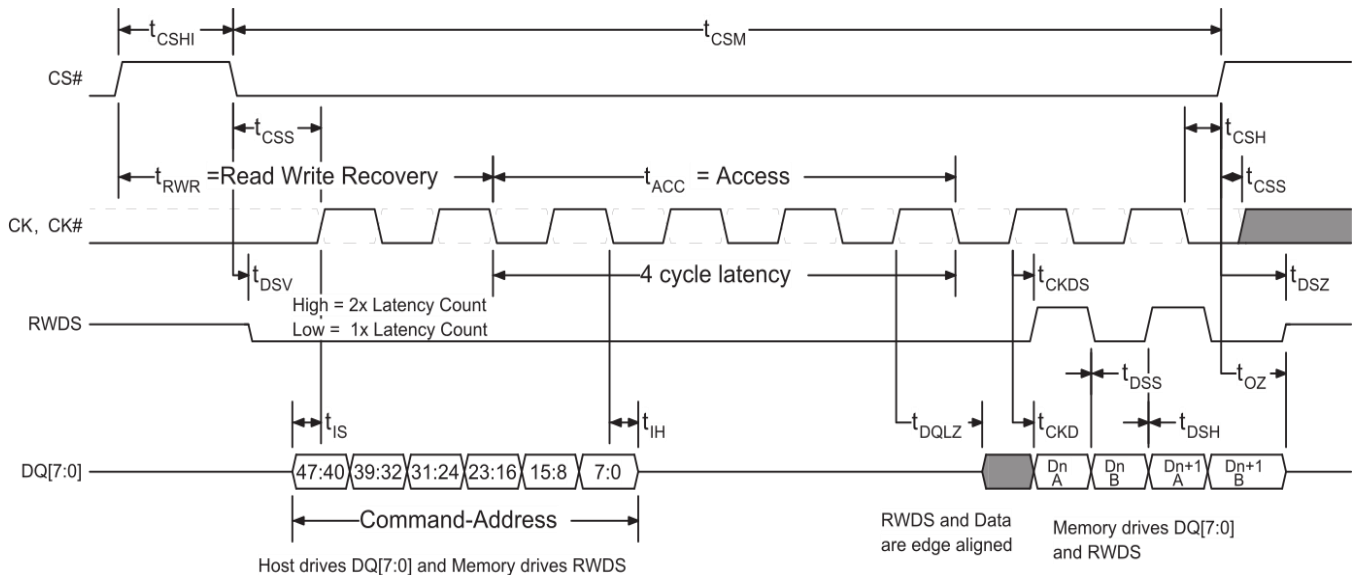
HyperBus は、信号数が少なく、ダブルデータレート (DDR) インターフェースを備えており、システム内のデバイス I/O 接続と信号ルーティングの輻輳の数を減らしながら、高い読み出しおよび書き込みスループットを実現します。

これらのデバイスは、8ビットデータバスを介してダブルデータレート (DDR) モードでデータおよびコマンド/アドレス情報を送信します。クロック入力信号は、コマンド/アドレス/データ情報を受信する際のデバイスによる信号キャプチャに使用されます。データストローブ (RWDS/DQS) は両方のインターフェースでの出力であり、データがデバイスからホストに転送されている時点を示します。RWDS/DQS は読み出し動作のデータ転送中に、クロックの立ち上りと立ち下りエッジを基準にします。

コマンド、アドレス、およびデータ情報は、8本の HyperBus DQ [7:0]信号を介して転送されます。クロック入力 (CK#, CK) は、DQ 信号でコマンド、アドレス、データを受信した際に HyperBus スレーブデバイスが情報をキャプチャするために使用されます。コマンド/アドレス/書き込みデータの値はクロックエッジにセンターアラインメントされ、リードデータの値は RWDS/DQS のトランジションにエッジアラインメントされます。

すべての HyperBus および OSPI 入力/出力は LVCMOS 互換であり、1.8V または 3.0V (公称) の電圧電源をサポートします。制御信号はマスタクロックを除き、すべてシングルエンドです。OSPI のマスタクロックはシングルエンドですが、HyperBus では 1.8V アーキテクチャの場合のみマスタクロックを差動にする必要があります。HyperBus および OSPI 命令プロトコルは、従来の業界標準のシリアルペリフェラルインターフェース (SPI) に従います。すべてのトランザクションは、チップセレクト (CS#) のアサートから始まります。これに続いて、CS# がアサートされなくなるまで、アクセスレイテンシのあるコマンドバイトとアドレスバイトの転送、および読み出しまたは書き込みデータ転送が行われます。

図 2. 読み出しトランザクション、単一の初期遅延カウンタ



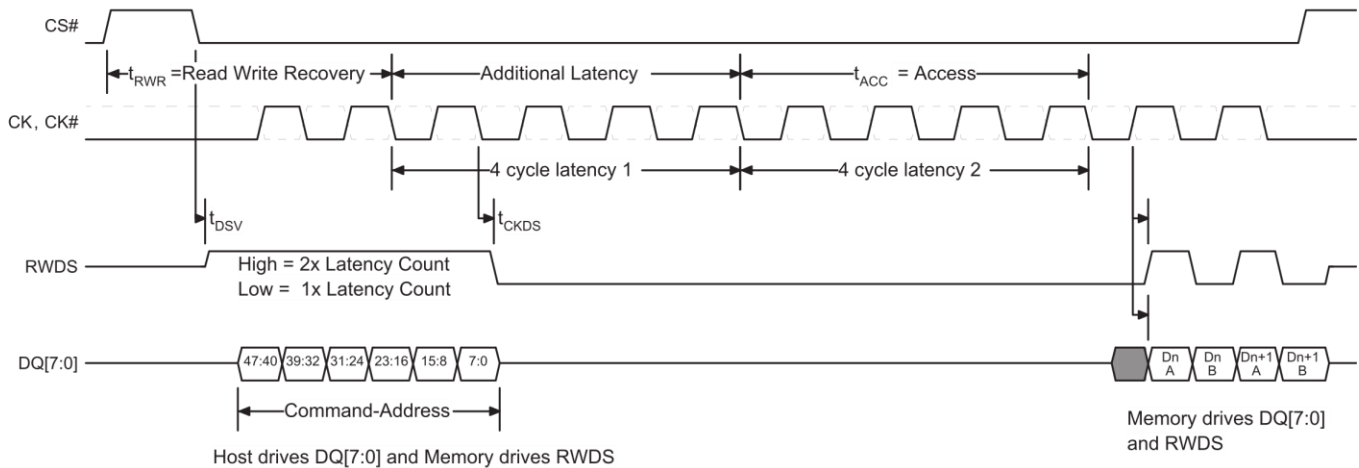
データストローブ (RWDS) は以下のことを示す双方向信号です。

- 読み出しトランザクションでデータが HyperRAM デバイスからマスタデバイスに転送を開始するタイミング (初期読み出し待ち時間)
- 読み出しトランザクション中に HyperRAM デバイスからマスタデバイスにデータが転送されるとき (ソース同期読み出しデータストローブとして)
- 書き込みトランザクションでマスタデバイスから HyperRAM デバイスへのデータ転送を開始する可能性がある場合 (初期書き込み遅延)
- 書き込みデータ転送中のデータマスキング。

読み出しまたは書き込みトランザクションの CA 転送部分では、RWDS は HyperRAM デバイスからの出力として機能し、トランザクションで追加の初期アクセス待ち時間が必要かどうかを示します。

読み出しデータ転送中、RWDS は、データ値が RWDS の遷移とエッジアラインされた読み出しデータストローブです。

図 3. 読み出しトランザクション、追加のレイテンシ カウント



書き込みデータ転送中、RWDS は、各データバイト転送が RWDS HIGH (無効でメモリ内のバイト位置の変更を防止) でマスクされているか、RWDS LOW (有効でメモリへの書き込み) でマスクされていないかを示します。ホストはデータマスクングを使用して、メモリ内の書き込みデータをバイトアラインしたり、1 回のバースト書き込みで複数のワードアラインされていない書き込みをマージできます。書き込みトランザクション中、データはクロック遷移に合わせてセンターアラインメントになります。

HyperBus プロトコルと関連するタイミングパラメータの詳細については、データシートを参照してください。

### 3 Octal SPI (xSPI) – 入門編

xSPI (Octal) は、8つの I/O をサポートする SPI 互換、低信号数、ダブルデータレート (DDR) インターフェースです。xSPI (Octal) の DDR プロトコルは、DQ 入力/出力信号でクロックサイクルごとに 2 データバイトを転送します。xSPI (Octal) での読み出しまたは書き込みトランザクションは、内部 RAM アレイで一連の 16 ビット幅の 1 クロックサイクルのデータ転送と、DQ 信号で 2 つの対応する 8 ビット幅 1/2 クロックサイクルのデータ転送で構成されます。xSPI は、外部メモリアンターフェースの JEDEC 規格 (JESD251) の 1 つとして採用されており、市場のマイクロコントローラや IP ベンダーに広く採用されています。

xSPI (Octal) の各トランザクションにはコマンドを含める必要がありますが、アドレスとデータはオプションです。トランザクションは次のような構造です。

- 各トランザクションは、CS# が LOW になることから始まり、CS# が HIGH を返すことで終わります。
- シリアルクロック (CK) は、ホストとメモリ間の各ビットまたはビットグループの転送をマークします。すべての転送はすべての CK エッジで発生します (DDR モード)。
- 各トランザクションには、実行するデバイス操作のタイプを選択する 16 ビットコマンドがあります。  
注: 16 ビットコマンドは、2 つの 8 ビットオペコードに基づきます。同じ 8 ビットオペコードがクロックの両エッジで送信されます。
- コマンドはスタンドアロンにすることも、アドレスビットを続けてデバイス内のメモリ位置を選択してデータにアクセスもできます。
- 読み出しトランザクションには、アドレスビットの後に待ち時間が必要であり、0~数 CK サイクルになる可能性があります。CK は、読み出しトランザクションの待機期間中もトグルを継続する必要があります。  
注: トランザクションのコマンドとアドレスの部分では、メモリは RWDS 信号を HIGH に駆動することで、最初のレイテンシ期間に追加される、必要なリフレッシュ時間 ( $t_{RFH}$ ) のために追加のレイテンシ期間が必要かどうかを示せます。
- レジスタへの書き込みトランザクションには、レイテンシ期間は必要ありません。

- メモリアレイへの書き込みトランザクションには、アドレスビットの後に待ち時間が必要であり、0～数 CK サイクルになる可能性があります。CK は、書き込みトランザクションの待機期間中もトグルを継続する必要があります。トランザクションのコマンドおよびアドレス部分の間、必要なリフレッシュ時間 ( $t_{RFH}$ ) に追加の遅延期間が必要かどうかを、メモリが示せることに注意してください。この時間とは、RWDS 信号を HIGH に駆動することによって初期遅延期間に追加される時間です。
- すべてのトランザクションで、コマンドビットとアドレスビットは、最上位ビット (MSb) が最初にあるデバイスでシフトされます。データバイト内の個々のデータビットも、最初にデバイス MSb にシフトインおよびシフトアウトされます。すべてのデータバイトは、最下位のアドレスバイトが最初に送信されて転送されます。

図 4. xSPI (8 進数) コマンドのみのトランザクション (DDR)

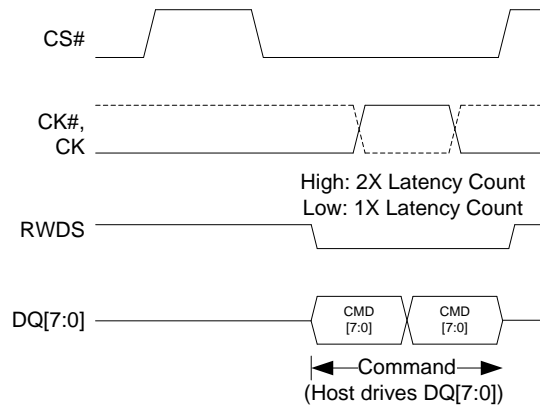


図 5. xSPI (8 進数) レイテンシトランザクションなしの書き込み (DDR) (レジスタ書き込み)

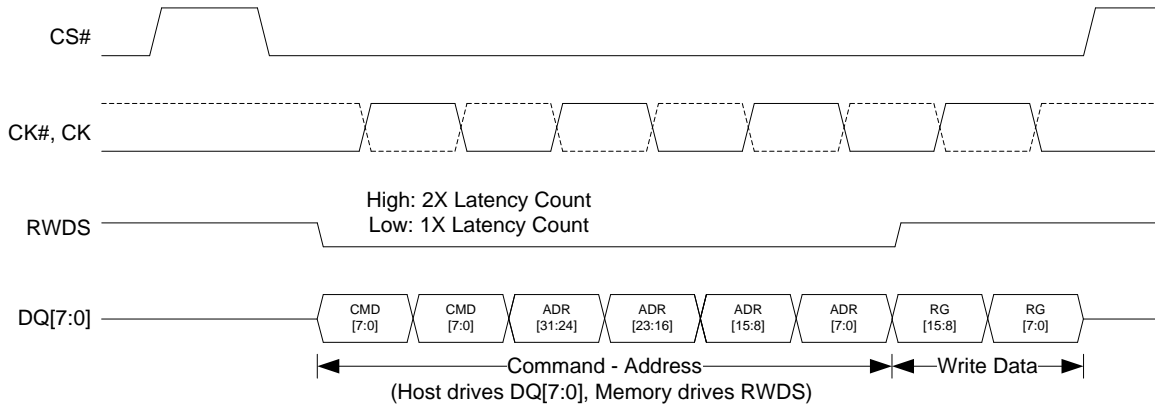


図 6. xSPI (8 進数) 1X レイテンシトランザクション (DDR) での書き込み (メモリアレイ書き込み)

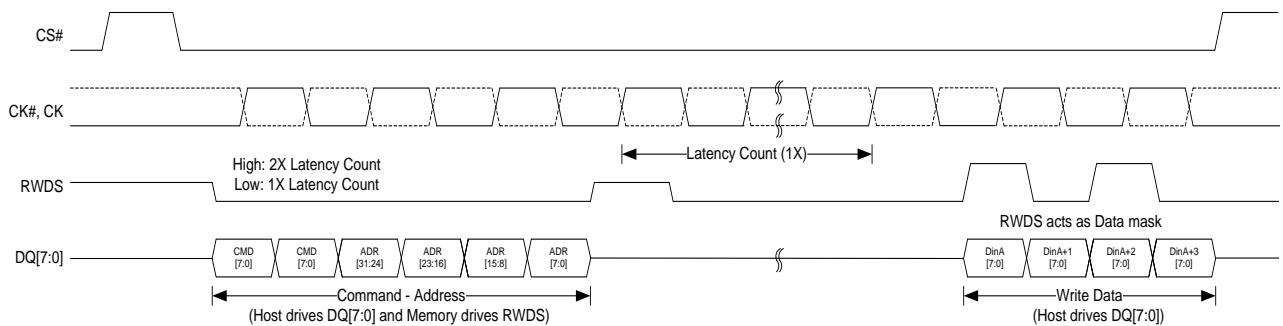
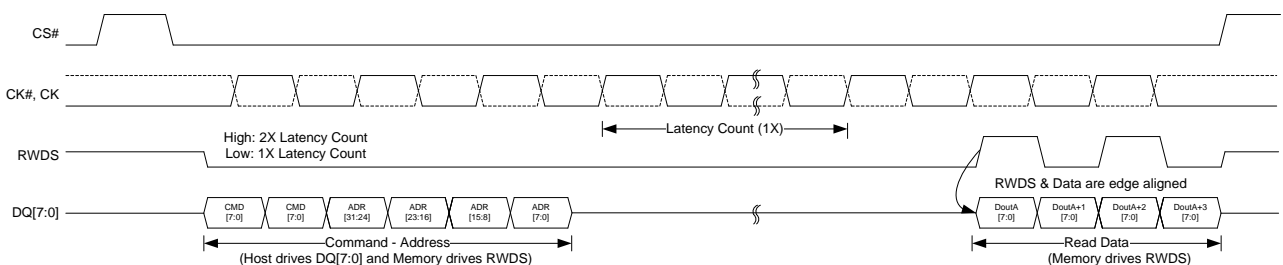


図 7. xSPI (8 進数) 1X レイテンシトランザクション (DDR) での読み出し (すべての読み出し)



標準としての xSPI は、いくつかのサイプレスセミコンダクタのフラッシュデバイスでサポートされます。しかし、このドキュメントで説明されている HyperRAM デバイスは、(8-8-8) 形式の操作のみをサポートします。すべてのオペコード、アドレス、データは 8 本の I/O で送信する必要があります。xSPI プロトコルと関連するタイミングパラメータの詳細については、xSPI HyperRAM のデータシートを参照してください。

## 4 HyperRAM 製品の概要

64Mb の HyperRAM デバイスは 1.8V または 3.0V のアレイと I/O、同期セルフリフレッシュダイナミック RAM (DRAM) です。HyperRAM デバイスは、ホストシステムへの HyperBus インターフェースを提供します。第 2 世代 HyperRAM は、xSPI インターフェースのサポートも追加しました。どちらのインターフェースにも 8 ビット (1 バイト) 幅の DDR データバスがあり、ワード幅 (16 ビットデータ) のアドレス境界のみを使用します。読み出しトランザクションは、各クロックサイクル中に 16 ビットのデータを提供します (両方のクロックエッジで 8 ビット)。書き込みトランザクションは、各クロックサイクルから 16 ビットのデータを取得します (各クロックエッジで 8 ビット)。

読み出しおよび書き込みトランザクションでは、ターゲットの行/列のアドレスを定義するために 3 クロックサイクルが必要であり、次に  $t_{ACC}$  の初期アクセス遅延が必要です。トランザクションのコマンド (CA) 部分で、メモリは、RWDS 信号を HIGH に駆動することにより、必要なリフレッシュ時間 ( $t_{RFH}$ ) の追加の待ち時間が初期待ち時間に追加されるかどうかを示します。読み出し (または書き込み) トランザクション中、初期のデータ値が出力 (入力) された後、後続のクロック サイクルで追加のデータはラップまたはリニア シーケンスに従って行から読み出されます (行に書き込まれます)。

リニアバーストモードで設定されている場合、デバイスはメモリアレイから次のシーケンシャル行を自動的にフェッチし、連続的なリニアバーストをサポートします。読み出しまたは書き込みデータ転送中にアレイの次の行に同時にアクセスすることで、第 1 世代 HyperRAM デバイスでは 333MB/s となるのに対し、第 2 世代 HyperRAM では  $400\text{MB/s}$  (1 バイト (8 ビットデータバス) \* 2 (データクロックエッジ) \* 200MHz = 400MB/s) の持続的なデータレートが得られます。

### 4.1 信号の説明

HyperBus と xSPI (Octal) インターフェースの両方に、同じ信号の命名規則があります。以下の表 1 は、HyperRAM デバイスのすべての信号をまとめたものです。

表 1. 信号説明のまとめ

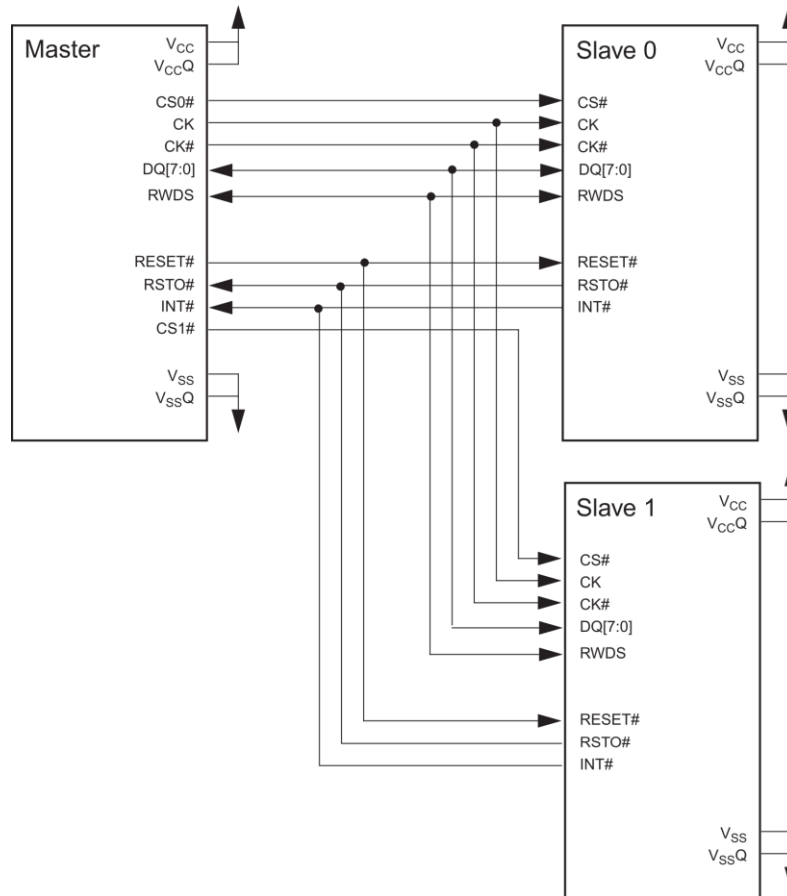
端子名	タイプ	説明
CS#	入力	チップセレクト。バストランザクションは、HIGH から LOW への遷移で開始します。バストランザクションは、LOW から HIGH への遷移で終了します。マスタデバイスには、スレーブごとに個別の CS#があります。
CK, CK#	入力	差動クロック。コマンド、アドレスとデータ情報は CK と CK#信号の交差に対してタイミングが定義される入力または出力です。 1.8V/3.0V I/O デバイスでは、差動クロックはオプションです。 シングルエンドクロック。CK#は使用されず、シングルエンド CK のみを使用されます。 クロックはフリーランニングである必要はありません。
DQ[7:0]	入出力	データ入力/出力。コマンド、アドレス、およびデータ情報は、読み出しおよび書き込みトランザクション中にこれらの信号で転送されます。
RWDS	入出力	読み出しデータ ストローブ。すべてのバストランザクションのコマンド/アドレス部の間、RWDS はスレーブ出力であり、追加の初期レイテンシが必要かどうかを示します。読み出しデータ転送中のスレーブ出力で、データは RWDS とエッジアライメントされます。書き込み時のデータ転送時にはスレーブ入力となり、データマスクとして機能します。(HIGH = 追加レイテンシ、LOW = 追加レイテンシなし)
RESET#	入力	ハードウェアリセット。LOW の場合、スレーブデバイスは自己初期化してスタンバイ状態に戻ります。RESET#が LOW の場合、RWDS と DQ [7:0]は High-Z 状態になります。スレーブの RESET#入力には弱いプルアップが含まれます。RESET#を未接続のままにすると、HIGH 状態にプルアップされます。
VCC	電源供給	アレイ電源
VCCQ	電源供給	入力/出力の電源
GND VSS	電源供給	アレイグランド
VSSQ	電源供給	入力/出力グランド
RFU	未接続	将来使用のための予約。内部的に接続されている、またはされていないかもしれません。信号/ボールの場所は、将来の互換性のために PCB ルーティングチャネルによって接続されておらず、使用されていないままにしておく必要があります。信号/ボールは将来的に信号で使用される可能性があります。

## 4.2 一般的なシステム接続

一般的な HyperBus システムでは、複数のスレーブメモリを個別のチップセレクト信号 (CS) を介してマスタに接続できます。図 8 に、一般的な構成を示します。スレーブの 1 つは HyperFlash デバイスにでき、もう 1 つは HyperRAM デバイスにできます。



図 8. 一般的なシステム構成



マスタは、HyperBus インターフェースまたは SoC に実装された IP をサポートするマイクロコントローラにできます。HyperRAM は本質的に DRAM デバイスであるため、セルはデータを保持するために定期的なリフレッシュサイクルが必要です。これは、分散リフレッシュロジックと呼ばれる内部セルフリフレッシュロジックによって行われます。

### 4.3 分散リフレッシュロジック

DRAM アレイでは、アレイ内のすべてのビットを定期的なリフレッシュする必要があります。これは、ホストシステムが特定の期限以内に各行内の 1 つの位置を読み書きすることで行えます。読み出しまたは書き込みアクセスは、ビットの行を内部バッファにコピーします。アクセスの最後に、バッファ内のビットがメモリ内の行に書き戻され、それによって DRAM メモリセルの行内のビットが再充電（リフレッシュ）されます。

HyperRAM デバイスには、行を自動的に更新する自己更新ロジックが含まれます。メモリがホストシステムによってアクティブに読み書きされない時にのみ行の自動リフレッシュを実行できます。リフレッシュが必要な時点でアクティブな読み書きが行われている場合は、リフレッシュロジックはこの読み書きの完了を待ちます。リフレッシュ動作が完了する前に新しい読み出しまたは書き込みが開始された場合、メモリは CA 期間中に RWDS HIGH を駆動します。これは、新しいアクセスを開始する前にリフレッシュ動作を完了させるために、新しいアクセスの開始時に追加の初期遅延時間が必要であることを示します。

表 2 に示すように、メモリアレイ全体に必要なリフレッシュ間隔は温度によって異なります。これは、すべての行がリフレッシュされなければならない時間です。すべての行のリフレッシュは、各インターバルの開始時に 1 つのバッチでアクセスしたり、複数の行を同時にグループ化して（バーストリフレッシュ）、インターバル全体に分散して実行したり、インターバル全体に均等に分散して 1 行リフレッシュしたりできます。セルフリフレッシュロジックは、メモリがバーストリフレッシュでホストアクセスを長い期間遅延させるような、長い期間リフレッシュ操作のバーストを行うのに忙しくないように、インターバル全体に単一行リフレッシュ操作を分散させることです。

表 2. 温度ごとのアレイ更新間隔

デバイス温度 (°C)	アレイの更新間隔 (ms)	アレイの行数	推奨される $t_{CSM}$ (μs)
85	64	8192	4
105	16	8192	1

分散リフレッシュ方式は、メモリが必要な分散リフレッシュを行えなくなるほど長いバースト トランザクションをホストが実行しないことが必要です。このことから、リフレッシュ ロジックがトランザクション間にリフレッシュを挿入できるように読み書きトランザクションの長さの上限値が設定されます。この制限は、CS# LOW 最大時間 ( $t_{CSM}$ ) と呼ばれます。 $t_{CSM}$  値は、アレイのリフレッシュ間隔をアレイの行数で割った値で決定され、この計算を半分に減らして、分散リフレッシュが必要になる直前に開始される最大長ホストアクセスによって分散リフレッシュ間隔を完全に逃すことができないようにします。 $t_{CSM}$  は必要な分散リフレッシュ間隔の半分に設定されています。このため、リフレッシュ操作を遅延させる一連の最大長ホストアクセスは、リフレッシュ間隔を行数で割った必要なレートの2倍のレートでリフレッシュ操作に追いつきます。

ホストシステムは、 $t_{CSM}$  に違反する前に各トランザクションを終了することにより、 $t_{CSM}$  値を尊重する必要があります。これは、ホストメモリコントローラロジックが  $t_{CSM}$  制限に達したときに長いトランザクションを分割するか、またはホストシステムのハードウェアまたはソフトウェアが  $t_{CSM}$  よりも長い単一の読み出しまたは書き込みトランザクションを実行しないことによって実行できます。

表 2 に示すように、アレイのリフレッシュ間隔は低温で長くなるため、 $t_{CSM}$  を増やしてトランザクションを長くできます。ホストシステムは、表の  $t_{CSM}$  値を使用して最大動作温度を取得するか、システム内の温度センサーから現在の動作温度を決定して、より長い分散リフレッシュ間隔を設定できます。

#### 4.4 電力モード

HyperBUS および xSPI HyperRAM デバイスはどちらも、以下の同一の電源モードをサポートします。

**アクティブ モード** : HyperRAM へのアクセスに使用される通常の動作モードは、アクティブモードとして定義されます。CS#信号を有効にすると、デバイスのアクティブモードが開始されます。すべてのデバイス操作は、アクティブモードで実行する必要があります。

**インターフェース スタンバイ** : スタンバイは、デバイスがデータ転送用にホストによって選択されていない間 (CS# = HIGH)、インターフェースのデフォルトの低電力状態です。この状態では、CS#と RESET#以外のすべての入力と出力は無視されます。

**アクティブクロック ストップ** : アクティブクロックストップ状態は、読み出しまたは書き込み動作のデータ転送部分の間、デバイスのインターフェース消費電力を  $I_{CC6}$  レベルに削減します。クロックが  $t_{ACC} + 30ns$  の間安定していると、デバイスは自動的にこの状態を有効にします。アクティブクロックストップ状態の間は、読み出しデータはラッチされ、常にデータバスにドライブされます。

アクティブクロックストップ状態は、ホストシステムクロックが停止してデータ転送を一時停止したときの消費電流を削減するために役立ちます。これらの延長されたデータ転送サイクル全体で CS#が LOW の場合でも、メモリデバイスのホストインターフェースは  $t_{ACC} + 30ns$  でアクティブクロックストップ電流レベルになります。これにより、データ転送が停止した場合にデバイスをより低い電流状態に移行できます。クロックのトグルによりデータ転送が再開されると、アクティブ読み書き電流は回復します。アクティブクロックストップ状態は、 $t_{CSM}$  制限に違反して使用してはいけません。 $t_{CSM}$  に違反する前に、CS#を HIGHにする必要があります。クロックは、LOW 状態である限り、アクティブなトランザクションのどの部分でも停止できます。レジスタアクセス中にクロックを停止しないようにすることを推奨します。詳細な仕様については、デバイスのデータシートを参照してください。

**ディープパワーダウン**：ディープパワーダウン (DPD) 状態では、消費電流は可能な限り低いレベル ( $i_{DPD}$ ) に駆動されます。DPD 状態に入るためには、デバイスコンフィグレーションレジスタの関連ビットに「0」を書き込みます。デバイスは  $t_{DPDIN}$  時間内に電力を削減し、すべてのリフレッシュ操作が停止します。DPD 状態の間、メモリスパースのデータは失われます (更新しないと無効になります)。CS#を LOW、次に HIGH に駆動すると、デバイスは DPD 状態を終了します。また、POR またはハードウェアリセットにより、デバイスは DPD 状態を終了します。スタンバイ状態に戻るためには、 $t_{EXTDPD}$  時間が必要です。POR の後にスタンバイ状態に戻るためには、他の POR と同様に、 $t_{VCS}$  時間が必要です。それらのイベントのいずれかで DPD を終了した後、デバイスの状態は POR を実行した後の状態と同じです。

**注**：xSPI (Octal) では、Deep Power Down トランザクションまたは Write Any register トランザクションを使用して DPD を入力できます。

**ハイブリッドスリープ**：この電源は、第 2 世代の HyperRAM デバイスでのみ使用できます。ハイブリッドスリープ (HS) 状態では、消費電流が減少します ( $i_{HS}$ )。HS 状態に入るためには、デバイスコンフィグレーションレジスタの関連ビットに「0」を書き込みます。このデバイスは、 $t_{HSIN}$  時間内に電力を削減します。メモリスパースとレジスタスペースのデータは、HS 状態の間保持されます。CS#を LOW にすると、デバイスは HS 状態を終了し、関連するビットは「1」になります。また、POR またはハードウェアリセットにより、デバイスはハイブリッドスリープ状態を終了します。

POR またはハードウェアリセットにより、メモリコアデータが失われる可能性のある更新が無効になることに注意してください。スタンバイ状態に戻るためには、 $t_{EXITHS}$  時間が必要です。これらのイベントのいずれかが原因で HS を終了した後、デバイスはハイブリッドスリープに入るのと同じ状態になります。ハイブリッドスリープ電源モードの詳細については、第 2 世代 HyperRAM デバイスのデータシートを参照してください。

**ハードウェアリセット**：RESET#入力は、デバイスをスタンバイ状態に戻すハードウェアメソッドを提供します。 $t_{RPH}$  の間、デバイスは  $I_{CC5}$  電流を引き出します。RESET#が  $t_{RPH}$  時間を超えて LOW に保持され続ける場合、デバイスには CMOS スタンバイ電流 ( $I_{CC4}$ ) が流れます。RESET#が LOW ( $t_{RP}$  中) の間、および  $t_{RPH}$  の間、バストランザクションは許可されません。

ハードウェアリセットは以下のことを行います。

- コンフィグレーションレジスタをデフォルト値に戻します
- RESET#が LOW のときにセルフリフレッシュ操作を停止します-メモリアレイデータは無効と見なされます
- デバイスを強制的にハイブリッドスリープ状態から終了させます
- デバイスを強制的にディープパワーダウン状態から終了させます

RESET#が HIGH に戻ると、セルフリフレッシュ操作が再開されます。セルフリフレッシュ操作は RESET# LOW の間に停止し、セルフリフレッシュ行カウンタはデフォルト値にリセットされるため、一部の行は表 2 に従って必要なアレイリフレッシュ間隔内にリフレッシュされない場合があります。ハードウェアリセット中またはリセット直後のアレイデータ。ホストシステムは、ハードウェアのリセット後に DRAM アレイデータが失われたと想定し、必要なデータをリロードする必要があります。

**ソフトウェアリセット**：ソフトウェアリセットは、デバイスをスタンバイ状態に戻すソフトウェアメソッドを提供します。 $t_{SR}$  の間、デバイスは  $I_{CC5}$  電流を引き出します。ソフトウェアリセットは次のことを行います。

- コンフィグレーションレジスタをデフォルト値に戻します
- ソフトウェアリセットプロセス中にセルフリフレッシュ操作を停止します - メモリアレイデータは無効と見なされます

ソフトウェアリセットが終了すると、セルフリフレッシュ操作が再開されます。セルフリフレッシュ操作が停止し、セルフリフレッシュ行カウンタがデフォルト値にリセットされるため、一部の行は、表 2 に従って必要なアレイリフレッシュ間隔内にリフレッシュされない場合があります。これにより、ソフトウェアのリセット中またはリセット直後に DRAM アレイデータが失われる可能性があります。ホストシステムは、ソフトウェアのリセット後に DRAM アレイデータが失われたと想定し、必要なデータをリロードする必要があります。

第 2 世代の HyperRAM と第 1 世代の HyperRAM デバイスの違いの詳細については、アプリケーションノート AN226137 – Migrating from S27KS0641 to S27KS0642 を参照してください。

## 5 HyperRAM を使用した設計

HyperBus および xSPI 規格は、少ないピン数、高スループットインターフェースの1つとして、業界全体で広く採用されています。サイプレスセミコンダクタは、HyperBus チップセットパーツと緊密に連携して、HyperBus インターフェースを SoC のハード IP として、または主要な FPGA ベンダーのソフト IP として有効にします。

表 3. HyperBus チップセットのサポート

パートナー	チップセット/プラットフォーム名	用途	HyperFlash	HyperRAM
サイプレス	Traveo S6J331x	自動車用クラスタ	•	•
	Traveo S6J335x	自動車用ゲートウェイ	•	•
	Traveo S6J326Cx	自動車用クラスタ	•	•
	Traveo S6J324Cx	自動車用クラスタ	•	•
	Traveo S6J327Cx	自動車用クラスタ	•	•
	Traveo S6J328Cx	自動車用クラスタ	•	•
	Traveo S6J32DAx	自動車用クラスタ	•	•
	Traveo S6J32BAx	自動車用クラスタ	•	•
	FM4 ファミリ S6E2DH シリーズ	産業用	•	•
Altera/Intel	MAX10	産業用	•	•
	Cyclone 10 LP	産業用	•	•
GCT	GDM7243i	IoT	•	•
Greewaves Technologies	GAP8	人工知能 (AI), IoT	•	•
Maxim	MAX32650	産業用, ポータブル医療, IoT	•	•
NXP	MAC57D5xxx	自動車用クラスタ	•	
	S32K148	自動車用ジェネリック/ボディ	•	
	S32V23x	自動車用 ADAS	•	
	Kinetis K80	産業用	•	
	Kinetis K82	産業用	•	
	Kinetis K28F	産業用	•	
	i.MX8 Family	自動車用インフォテインメント、産業用、コンシューマ	•	•
	i.MX RT1050	産業用	•	•
	i.MX RT1020	産業用	•	•
	i.MX RT1060	産業用	•	•
	i.MX RT106A	Alexa 音声サービス用の MCU ベースのソリューション	•	•
I.MX RT family	I.MXRT ファミリで HyperRAM をサポートする NXP アプリノート	•	•	
ルネサス	R-CAR D3	自動車用クラスタ	•	
	R-Car H3	自動車用インフォテインメント/ ADAS	•	
	R-CAR M3	自動車用インフォテインメント/ ADAS	•	
	R-Car V3M	自動車用 ADAS	•	
	RZ/A2M	産業用, AI	•	•
ST	STM32L4Rx	産業用	•	•
	Chorus SPC58 H	自動車用ゲートウェイ		•
Texas Instruments	Sitara AM6xxx	産業用、ネットワークング	•	•
Xilinx	VIRTEX Ultrascale+ (16nm)	通信	•	•
	KINTEX Ultrascale+ (16nm)	通信	•	•
	VIRTEX Ultrascale (20nm)	通信	•	•
	KINTEX Ultrascale (20nm)	通信 / 産業用	•	•
	VIRTEX-7 (28nm)	ネットワーク機器	•	•
	KINTEX-7 (28nm)	ネットワーク機器	•	•
	Zynq 7000	多種	•	•
	Zynq UltraScale+	産業用	•	•
	Artix-7	産業用	•	•

表 4. HyperBus 3rd Party 開発プラットフォーム

パートナー	3rd Party 開発プラットフォーム	用途	HyperFlash	HyperRAM
TrenzElectronics	TE0725 と Xilinx Artix-7 Xilinx Artix を搭載した TE0748	産業用 安全な SD	• •	• •
Devboards	Altera MAX10 を搭載した HyperMAX	産業用、医療、自動車	•	•

表 5. HyperBus メモリコントローラの IP

IP サプライヤー	リンク	Soft IP として FPGA に統合可能	SoC に統合可能
サイプレス	サイプレス	•	•
ケイデンス	ケイデンス		•
Mobiveil	Mobiveil		•
Synaptic Laboratories Ltd.	SynapticLabs	•	•

表 6. HyperBus メモリコントローラの検証 IP

IP サプライヤー	リンク	HyperFlash	HyperRAM
ケイデンス	ケイデンス	•	•

サイプレスセミコンダクタはまた、顧客がシステム設計を有利に開始できるようにするいくつかの設計 (IBIS および動作) モデルを提供します。

## 6 HyperRAM –ピン数の少ない高性能システムメモリ

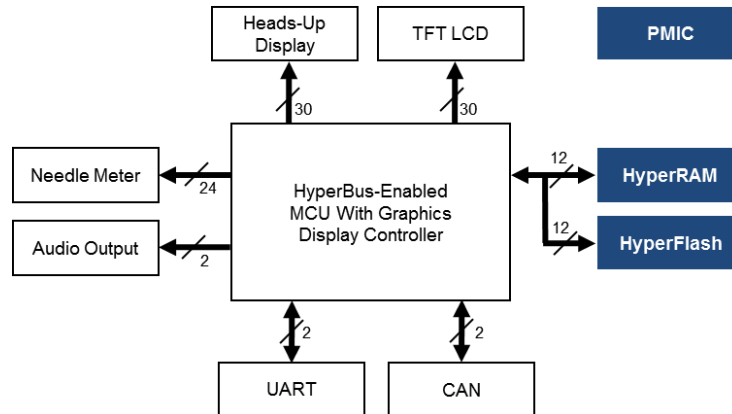
内部 RAM のサイズは、コントローラまたは FPGA/SoC を使用して複雑なアルゴリズムを実行する方法を決定する際の最も重要な要素の 1 つです。適切な RAM またはシステムメモリにより、プロセッサはコードをフェッチして実行できます。この RAM は、コンテキストとステートの格納領域を必要とする複数レベルのインサービスルーチンを実装するためにアプリケーションコードで使用される Stack/Heap ストレージとしても機能する場合があります。

アプリケーションの複雑さが増すにつれて、コントローラ/SoC はすぐにシステムメモリを使い果たします。設計者は、SDRAM または DDR RAM を使用して高速の外部 RAM を追加することにより、しばしばこの問題を解決します。このスケーラビリティは、アルゴリズムの複雑さを損なうことなく、数メガバイトの高性能 RAM を追加できることを意味します。しかし、SDRAM および DDR メモリには重大な欠点があります。SDRAM/DDR インターフェースは大量の I/O を使用するため、専用のインターフェースが必要となり、PCB 設計や製造コストが増加します。

最近の SoC/コントローラ設計のトレンドとして、NOR フラッシュメモリのデフォルト規格として HyperBus が採用されます。現在の HyperFlash デバイスは、インターフェースを 200MHz DDR で動作させながら最大 400Mbps のスループットを実現しており、SDRAM インターフェースでは 30~35 個の I/O が必要なのにに対し、必要な信号はわずか 12 本ですみます。HyperRAM は、HyperFlash と同じ高性能インターフェースを使用しますが、真の拡張 RAM として動作します。したがって、SoC/コントローラ/FPGA の内部メモリが不足している場合、既存の HyperBus インターフェースを再利用しながら、この高性能で少ピン数のシステムメモリを使用できます。HyperRAM は、頻繁な更新を必要とする高解像度のディスプレイ要素を格納するために、自動車クラスで広く使用されます。産業環境では、コントローラは HyperRAM を使用することで、ワーキングメモリを桁違いに増やせます。

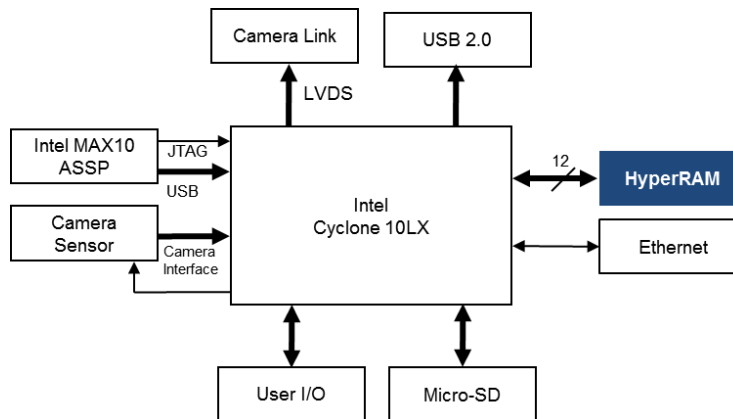
図 9 と図 10 に、システムでの HyperRAM の使用モデルを示します。自動車クラスは、いくつかのグラフィックレイヤの混合を実行して、車内の最終的なディスプレイを生成します。グラフィックの品質と複雑さは、コントローラに実装できるディスプレイバッファのサイズによって異なります。このようなシステムに高性能 HyperRAM を追加すると、ディスプレイのリフレッシュレートに不利益を課すことなく、いくつかの複雑なディスプレイ要素をロードおよび生成できます。

図 9. 自動車クラスターの DBuffer としての HyperRAM



HyperRAM のもう 1 つの使用モデルは、FPGA ベースのマシンビジョンシステムの拡張メモリです。一般的な FPGA の RAM リソースは限られます。これらのリソースは、イメージングアルゴリズムに必要な重要な処理を実行するためにより適切に使用されます。FPGA の構成とコードは、オンボードフラッシュまたは SD カードから HyperRAM にロードされ、最高のパフォーマンスを実現します。

図 10. マシンビジョンシステムの拡張メモリとしての HyperRAM



## 7 関連ドキュメント

- [AN211622](#) – HyperFlash および HyperRAM レイアウトガイド
- [AN209853](#) – HyperRAM™ のリフレッシュ間隔の最適化
- [AN218684](#) – HyperBus™ Memory: Guide to Efficient Data Access
- [AN226137](#) – Migrating from S27KS0641 to S27KS0642
- [HyperBus™ Specification Low signal Count, High-Performance DDR Bus](#)

## 著者と寄稿者について

名前: Nilesh Badodekar  
 役職: 上級スタッフ アプリケーション エンジニア  
 経歴: Nilesh Badodekar は、インドのカラグルルにある Indian Institute of Technology で視覚情報処理と組み込みシステムの修士号を取得しています。

## 改訂履歴

文書名: AN226576 – HyperRAM™の使用方法

文書番号: 002-32715

版数	変更内容
**	本版は英語版 002-26576 Rev. **について、CYPRESS DEVELOPER COMMUNITYの参画者によって日本語に翻訳されたドキュメントです。

## セールス、ソリューションおよび法律情報

### ワールドワイドな販売と設計サポート

サイプレスは、事業所、ソリューションセンター、メーカー代理店、および販売代理店の世界的なネットワークを保持しています。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

Arm® Cortex® Microcontrollers	<a href="http://cypress.com/arm">cypress.com/arm</a>
車載用	<a href="http://cypress.com/automotive">cypress.com/automotive</a>
クロック&バッファ	<a href="http://cypress.com/clocks">cypress.com/clocks</a>
インターフェース	<a href="http://cypress.com/interface">cypress.com/interface</a>
IoT (モノのインターネット)	<a href="http://cypress.com/iot">cypress.com/iot</a>
メモリ	<a href="http://cypress.com/memory">cypress.com/memory</a>
マイクロコントローラ	<a href="http://cypress.com/mcu">cypress.com/mcu</a>
PSoC	<a href="http://cypress.com/psoc">cypress.com/psoc</a>
電源用 IC	<a href="http://cypress.com/pmhc">cypress.com/pmhc</a>
タッチセンシング	<a href="http://cypress.com/touch">cypress.com/touch</a>
USB コントローラ	<a href="http://cypress.com/usb">cypress.com/usb</a>
ワイヤレス	<a href="http://cypress.com/wireless">cypress.com/wireless</a>

### PSoC®ソリューション

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6 MCU](#)

### サイプレス開発者コミュニティ

[コミュニティ](#) | [サンプルコード](#) | [Projects](#) | [ビデオ](#) | [ブログ](#) | [トレーニング](#) | [Components](#)

### テクニカルサポート

[cypress.com/support](http://cypress.com/support)

本書で言及するその他すべての商標または登録商標は、それぞれの所有者に帰属します。



Cypress Semiconductor  
An Infineon Technologies Company  
198 Champion Court  
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2019-2021. 本書面は、Cypress Semiconductor Corporation 及び Spansion LLC を含むその子会社 (以下「Cypress」という。) に帰属する財産である。本書面 (本書面に含まれ又は言及されているあらゆるソフトウェア若しくはファームウェア (以下「本ソフトウェア」という。)) を含む) は、アメリカ合衆国及び世界のその他の国における知的財産法及び条約に基づき Cypress が所有する。Cypress はこれらの法令及び条約に基づく全ての権利を留保し、本段落で特に記載されているものを除き、その特許権、著作権、商標権又はその他の知的財産権のライセンスを一切許諾しない。本ソフトウェアにライセンス契約書が伴っておらず、かつ Cypress との間で別途本ソフトウェアの使用方法を定める書面による合意がない場合、Cypress は、(1) 本ソフトウェアの著作権に基づき、(a) ソースコード形式で提供されている本ソフトウェアについて、Cypress ハードウェア製品と共に用いるためののみ、かつ組織内部でのみ、本ソフトウェアの修正及び複製を行うこと、並びに (b) Cypress のハードウェア製品ユニットに用いるためののみ、(直接又は再販売者及び販売代理店を介して間接のいずれかで) 本ソフトウェアをバイナリーコード形式で外部エンドユーザーに配布すること、並びに (2) 本ソフトウェア (Cypress により提供され、修正がなされていないもの) が抵触する Cypress の特許権のクレームに基づき、Cypress ハードウェア製品と共に用いるためののみ、本ソフトウェアの作成、利用、配布及び輸入を行うことについての非独占的で譲渡不能な一身専属的ライセンス (サブライセンスの権利を除く) を付与する。本ソフトウェアのその他の使用、複製、修正、変換又はコンパイルを禁止する。

**適用される法律により許される範囲内で、Cypress は、本書面又はいかなる本ソフトウェア若しくはこれに伴うハードウェアに関しても、明示又は黙示をとわず、いかなる保証 (商品性及び特定の目的への適合性の黙示の保証を含むがこれらに限られない) も行わない。**いかなるコンピューティングデバイスも絶対に安全ということはない。従って、Cypress のハードウェアまたはソフトウェア製品に講じられたセキュリティ対策にもかかわらず、Cypress は、Cypress 製品への権限のないアクセスまたは使用といったセキュリティ違反から生じる一切の責任を負わない。加えて、本書面に記載された製品には、エラーと呼ばれる設計上の欠陥またはエラーが含まれている可能性があり、公表された仕様とは異なる動作をする場合がある。適用される法律により許される範囲内で、Cypress は、別途通知することなく、本書面を変更する権利を留保する。Cypress は、本書面に記載のある、いかなる製品若しくは回路の適用又は使用から生じる一切の責任を負わない。本書面で提供されたあらゆる情報 (あらゆるサンプルデザイン情報又はプログラムコードを含む) は、参照目的のためのみに提供されたものである。この情報で構成するあらゆるアプリケーション及びその結果としてのあらゆる製品の機能性及び安全性を適切に設計、プログラム、かつテストすることは、本書面のユーザーの責任において行われるものとする。Cypress 製品は、兵器、兵器システム、原子力施設、生命維持装置若しくは生命維持システム、蘇生用の設備及び外科的移植を含むその他の医療機器若しくは医療システム、汚染管理若しくは有害物質管理の運用のために設計され若しくは意図されたシステムの重要な構成部分としての使用、又は装置若しくはシステムの不具合が人身傷害、死亡若しくは物的損害を生じさせるようなその他の使用 (以下「本目的外使用」という。) のためには設計、意図又は承認されていない。重要な構成部分とは、その不具合が装置若しくはシステムの不具合を生じさせるか又はその安全性若しくは実効性に影響すると合理的に予想できるような装置若しくはシステムのある構成部分をいう。Cypress 製品のあらゆる本目的外使用から生じ、若しくは本目的外使用に関連するいかなる請求、損害又はその他の責任についても、Cypress はその全部又は一部をとわず一切の責任を負わず、かつ Cypress はそれら一切から本書により免除される。Cypress は Cypress 製品の目的外使用から生じ又は本目的外使用に関連するあらゆる請求、費用、損害及びその他の責任 (人身傷害又は死亡に基づく請求を含む) から免責補償される。

Cypress, Cypress のロゴ, Spansion, Spansion のロゴ及びこれらの組み合わせ, WICED, PSoC, CapSense, EZ-USB, F-RAM, 及び Traveo は、米国及びその他の国における Cypress の商標又は登録商標である。Cypress のより完全な商標のリストは、[cypress.com](http://cypress.com) を参照すること。その他の名称及びブランドは、それぞれの権利者の財産として権利主張がなされている可能性がある。