

サイプレスの非同期 FIFO の理解

著者: Adithi Perepu

関連プロジェクト: なし

関連製品ファミリ: **CY7C421**

ソフトウェア バージョン: 該当なし

関連アプリケーション ノート: なし

本アプリケーション ノートの最新版または関連プロジェクト ファイルについては、
<http://www.cypress.com/go/AN1044> へアクセスしてください。

AN1044 は、非同期 FIFO **CY7C421** のアーキテクチャ、特長、拡張ロジックの概要を提供し、FIFO の共通問題およびその対策を説明します。

目次

はじめに.....	1
非同期 FIFO の概要.....	1
FIFO の読み出し／書き込み動作.....	2
共通 FIFO コンフィギュレーション.....	3
スタンドアロンおよび幅拡大のコンフィギュレーション.....	3
深さの拡大コンフィギュレーション (トークン パッシング メカニズム).....	5
再送信機能.....	7
アプリケーション.....	7
設計時の注意事項および対策.....	7
破損または繰り返しデータ.....	8
FIFO ロックアップ.....	8
欠陥または消失データ.....	9
繰り返しまたは順番誤ったデータ、偽の Full または Empty	9
Empty の読み出しと Full の書き込み.....	9
有効なパルス幅への違反.....	10
まとめ.....	11
ワールドワイドな販売と設計サポート.....	13

はじめに

本アプリケーション ノートはサイプレスの非同期 FIFO **CY7C421** の内部アーキテクチャを説明します。主なデバイス特長やアプリケーション、故障モード、一般的な問題状況とそのソリューションも含まれます。本アプリケーション ノートで指定されたタイミングのパラメーターはデバイス データシート「**CY7C421, 512 x 9 Asynchronous FIFO**」から取られます。

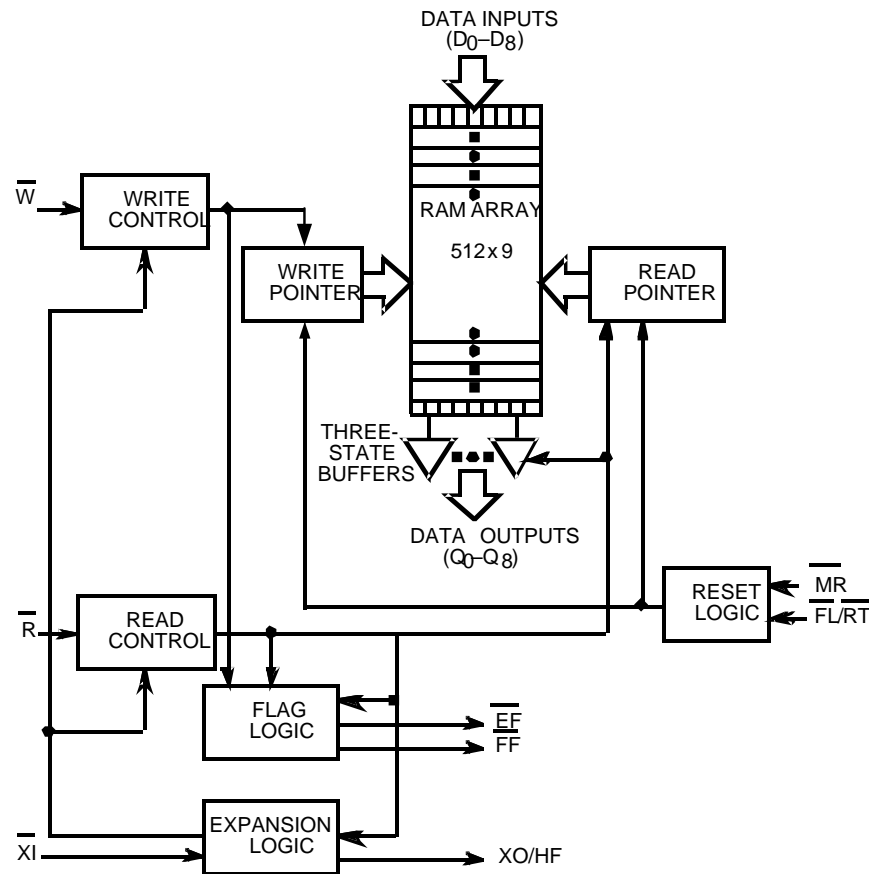
非同期 FIFO の概要

サイプレスの非同期 FIFO (**CY7C421**) は 512 ワード深さで、9 ビット幅です。このモノリシック デバイスは、15 ナノ秒までのアクセス時間と 25 ナノ秒までのサイクル タイムに対応しています。速度とパッケージのあり得る組み合わせについては、デバイス データシートを参照してください。

サイプレスの非同期 FIFO は、独立した読み出し／書き込み動作を可能にする専用の読み出し／書き込みポートの SRAM 種のインターフェースを採用します。FIFO は特別に設計されたデュアル ポートの SRAM セルを使用します。これらの SRAM セルは、両方のポートからの同時アクセスをサポートするために個別の読み出しと書き込みトランジスタを持っています。

CY7C421 非同期 FIFO は、データが書き込まれた順序と同じ順序で読み出されるように構成されます。Full、half-full、および empty のフラグは読み書き動作を安易にします。深さと幅を拡大するために追加のピンが用意されます。FIFO の論理ブロック図については、図 1 を参照してください。

図 1. FIFO ブロック図



FIFO の読み出し／書き込み動作

非同期 FIFO の読み出し／書き込みタイミングは図 2 と図 3 に示されます。読み出し動作は読み出しイネーブル (R) の立ち下がりエッジで開始されます。出力データバス (Q₀~Q₈) は R の立ち下がりエッジの t_A 期間後に有効なデータを提供します。この t_A 期間は FIFO の読み出しアクセス時間です。出力データバスは、R がアサートされてから t_{LZR} 期間後に高インピーダンス状態を終了します。読み出しがバス上で有効なデータが提供された後 (R の立ち下がりエッジの t_A 後) にのみ行われるよう注意する必要があります。R のデアサートは読み出し動作を終了します。

Q₀~Q₈ バス上のデータは R の立ち上がりエッジ後の t_{DVR} の間有効のままです。これは読み出しサイクル終了時の出力データホールド時間です。その後、内部回路は次の読み出し動作のためにそれ自体を準備します。この期間は t_{RR}、すなわち読み出し回復時間と呼ばれます。その後の読み出し動作はこの期間内に開始すべきではありません。t_{PR} で表す最小パルス幅は読み出しアクセスに必要で、読み出しアクセス時間 (t_A) に等しいです。

読み出しサイクル時間 (t_{RC}) は以下のように計算されます。

$$t_{RC} = \text{アクセス時間 (t}_A\text{)} + \text{読み出し回復時間 (t}_{RR}\text{)}$$

最大読み出し周波数は t_{RC} の逆数で、つまり、

$$\text{読み出し周波数 (最大値)} = 1/(t_A + t_{RR})$$

例えば、アクセスに 20ns かかり、読み出し回復に 10ns かかるサイプレス FIFO は、読み出しサイクルに 30ns かかる、つまり最大読み出しサイクル周波数が 33.3MHz です。

書き込み動作は読み出し動作と同様です。書き込み動作は書き込み信号 (W) のアサート時に開始され、W のデアサート時 (立ち上がりエッジ) に終了します。有効な書き込みを行うためには、入力データバス (D₀~D₈) は W の立ち上がりエッジ前の t_{SD} (セットアップ時間) の間安定し、そのエッジ後の t_{HD} (ホールド時間) の間有効のままです。有効な書き込み動作のためには、最小書き込みイネーブルパルス幅 (t_{PW}) が必要です。連続した書き込みサイクル間に書き込み回復時間 (t_{WR}) が必要です。

最大書き込み周波数は $1/(t_{PW} + t_{WR})$ です。例えば、書き込みストロブ幅が 15ns、書き込み回復時間が 10ns のデバイスの場合、書き込みサイクル時間は 25ns、つまり書き込み周波数は最大 40MHz です。

FIFO には個別の内部書き込みと読み出しカウンタ (ポインタ) が含まれます。書き込みポインタは常に次に書き込まれるワードを指し、読み出しポインタは常に現時点読み出される FIFO ワードを指します。各書き込みまたは読み出し動作は対応するカウンタを 1 位置インクリメントします。これらのカウンタの相対位置は empty、half-full、full のフラグで外部に示されるデバイスの状態を決定します。

図 2. 非同期読み出しタイミング

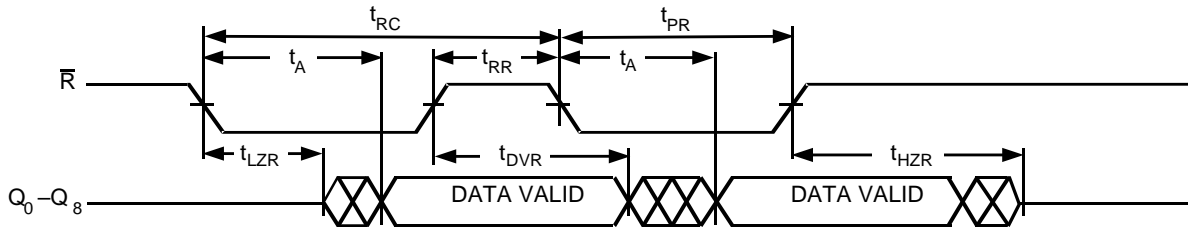
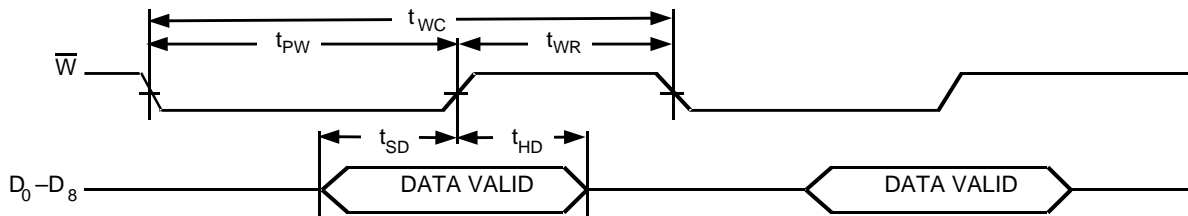


図 3. 非同期書き込みタイミング



共通 FIFO コンフィギュレーション

最小限の外部ロジックでより広い且つ／または深い FIFO を形成するために複数の非同期 FIFO をカスケード接続できます。複合フラグを生成するために OR ゲートを実装する外部ロジックが必要です。以下のセクションはスタンドアロン動作、幅拡大、および深さ拡大について説明します。

サイプレスの FIFO は拡大ロジック (幅と深さ両方の拡大) を実装するためにピン XI とピン XO を提供します。ピン XI とピン XO は 1 つの FIFO から別の FIFO へトークン パッシング方式で使用されます。FL はデータでロードされる最初の FIFO を示します。

スタンドアロンおよび幅拡大のコンフィギュレーション

図 4 にスタンドアロン コンフィギュレーションを示します。このコンフィギュレーションでは、XI (拡大入力) ピンは LOW に接続し、FL (最初にロード) ピンは HIGH に接続します。

図 4. スタンドアロン動作

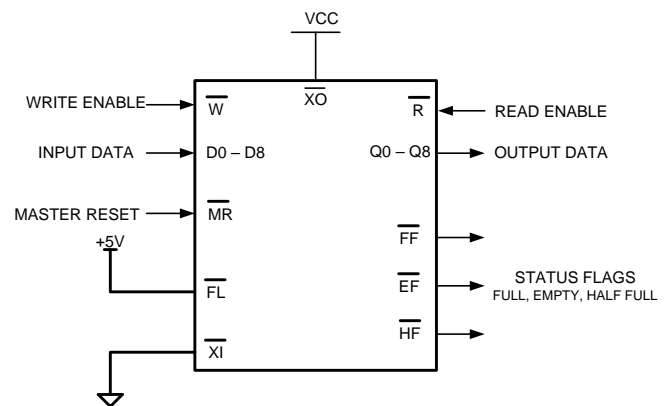
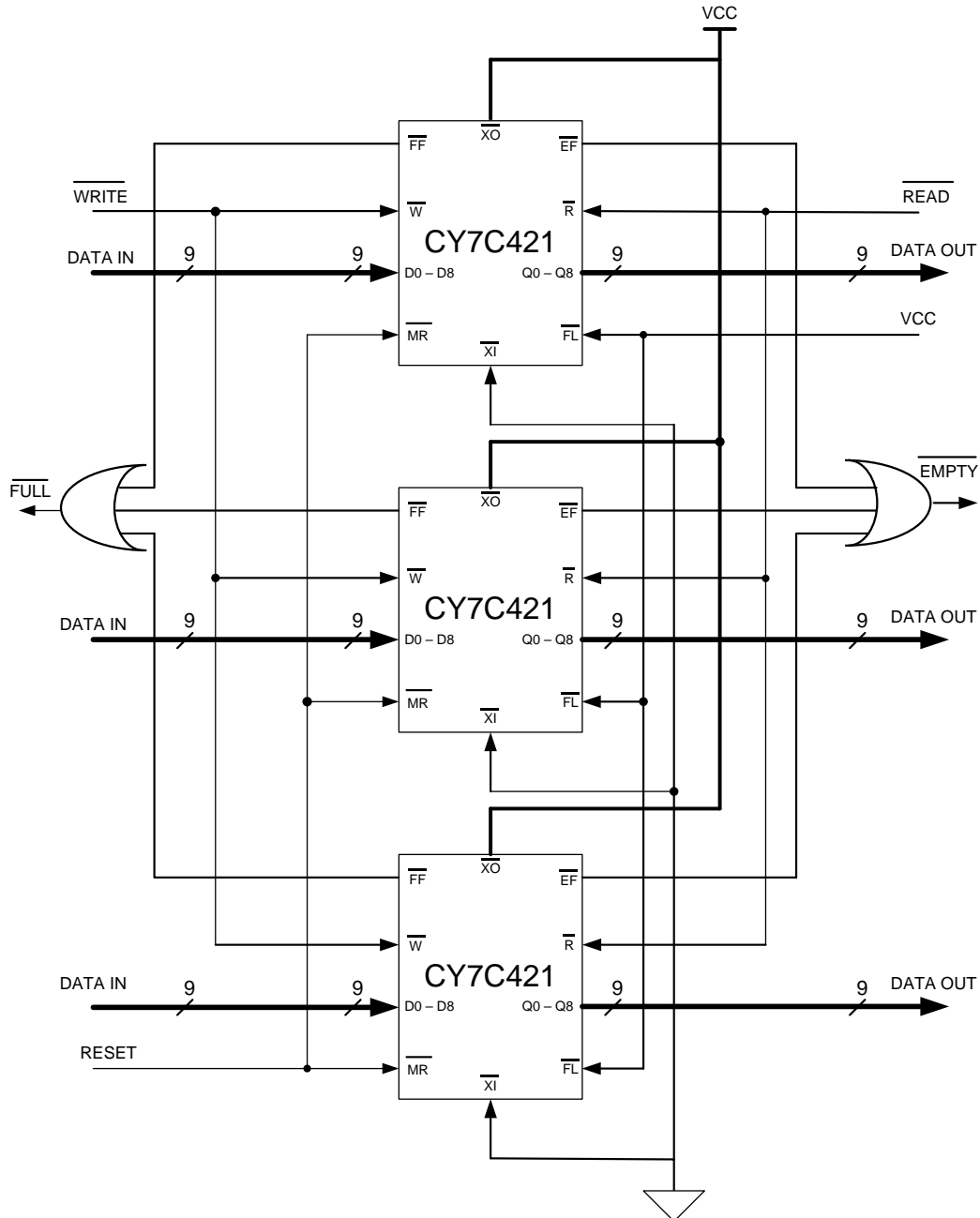


図 5 に幅拡大コンフィギュレーションを示します。スタンダードコンフィギュレーションと同様に、FIFO ピン XI は LOW に接続し、FL は HIGH に接続します。

幅拡大では、伝播遅延が設計の個々の FIFO が同時に full、half-full、または empty の状態に入ることを防ぐことがあります。そのため、幅拡大 FIFO の瞬間状態を適切に反映するために、複合フラグを外部で生成する必要があります。これは OR ゲートで実装できます。

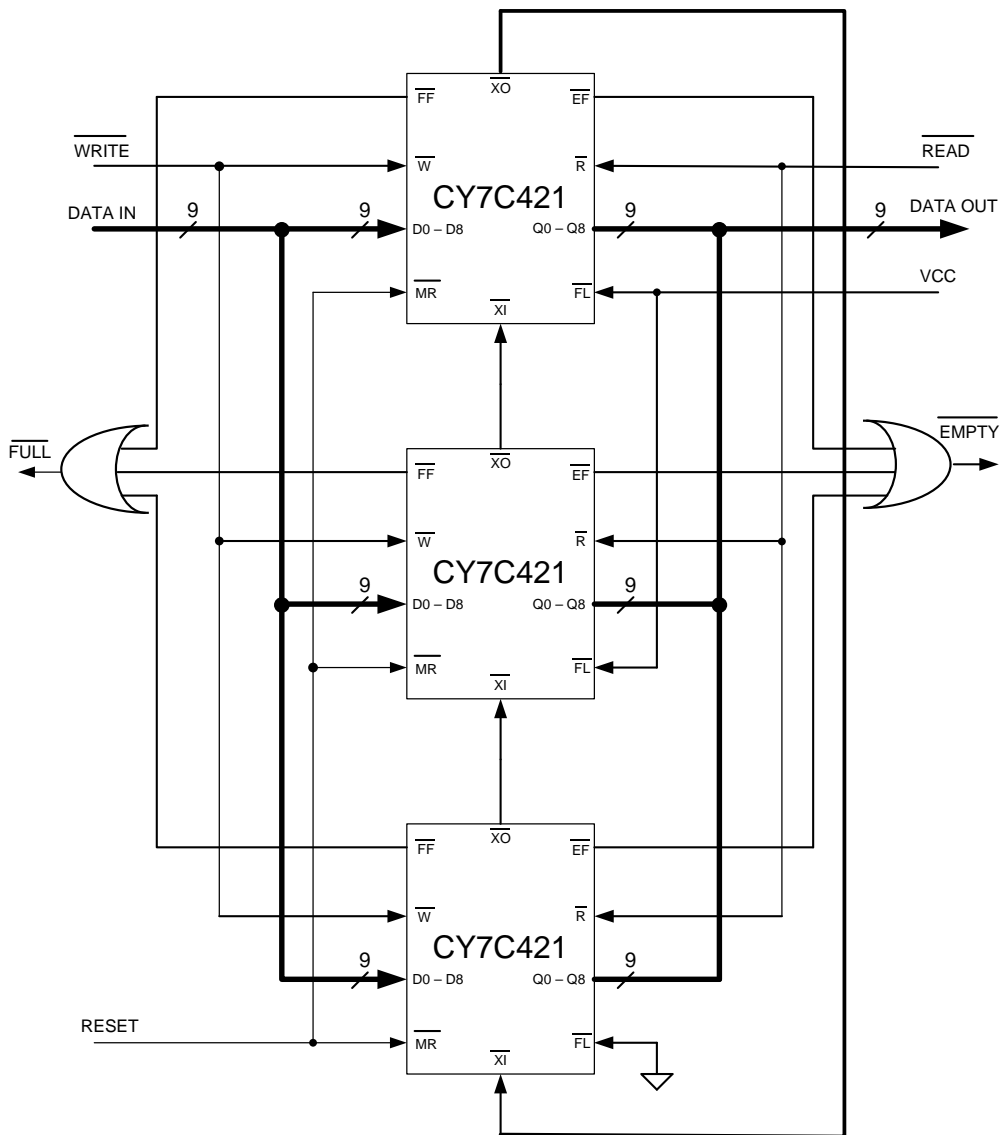
図 5. 幅の拡大



深さの拡大コンフィギュレーション (トークン パッシング メカニズム)

図 6 に深さの拡大を示します。このコンフィギュレーションでは、デバイスが最初書き込まれる FIFO であるようにそのデバイスの FL (最初にロード) ピンは LOW に接続される必要があります。その後、1 つのデバイスの XO (拡大出力) を次のデバイスの XI (拡大入力) ピンに接続することで FIFO をディジーチェーンでつなぐことができます。チェーンの最後のデバイスの XO 信号は最初のデバイスの XI ピンに接続され、よってトークン パッシング リングを形成します。

図 6. 深さの拡大 (トークン パッシング)



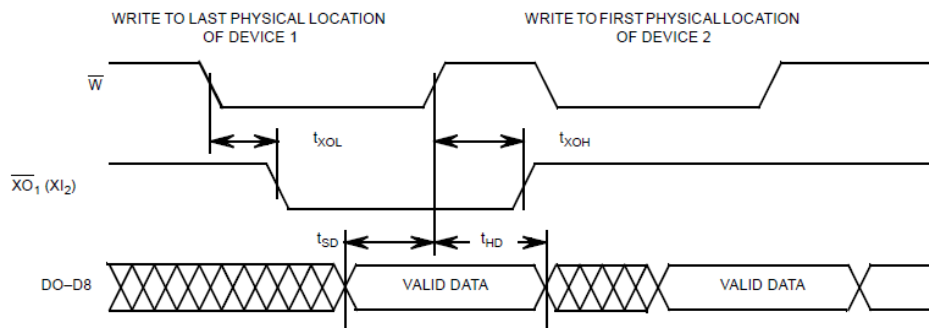
幅拡大された FIFO と同様に、empty と full の複合フラグを生成するために OR ゲートを使用できます。トークン パッシングは書き込みと読み出しプロセスの一貫性を保証します。読み出し／書き込みトークンは読み出し／書き込み動作でアクセスされるデバイスを決定します。書き込み動作のトークン パッシング手順では、最初の FIFO は満杯になるまで書き込まれます。内部書き込みポインターは書き込まれる位置を決定し、各書き込み後にインクリメントされます。書き込みポインターが最初の FIFO の最後の物理的な位置に到着した時、そのデバイスへ更なる書き込みは行えません。その時点で最初の FIFO は、XO から XI へのインターフェースを介してチェーンの次の FIFO へ書き込みトークンを渡します。2 番目のデバイスは書き込みトークンを持っており、満杯するまで将来の書き込みデータをすべて受信し、書き込みトークンをチェーンの次のデバイスへ渡します。

FIFO チェーンを満たすために十分な書き込みが行われた場合、最後のデバイスが最初のデバイスへ書き込みトークンを渡し戻すことに失敗します。理由は、full 状態の FIFO が書き込みトークンを受け取れないからです。読み出し動作が発生して内部の位置のいずれかを解放するまでは、FIFO チェーンへの更なる書き込みは許可されません。内部書き込みと読み出しカウンターの相対位置は、デバイスの状態を判断し、書き込み動作を通してデータを受け取れるかどうかを決定します。図 7 に書き込み動作のタイミングを示します。

書き込み手順と同様に、チェーンの最初の FIFO は読み出しトークンを持っています。FIFO チェーンが読み出される時、読み出しトークンを持っているデバイスは、デバイスの書き込みポインターに指定されるアドレスからのデータを供給します。すると、読み出しポインターはインクリメントされます。ポインターは FIFO が empty になるまで引き続きインクリメントされ、その後、読み出しトークンはチェーンの次のデバイスに渡されます。読み出しトークンは XO から XI へのインターフェースを介して渡されます。図 8 に読み出し動作のタイミングを示します。

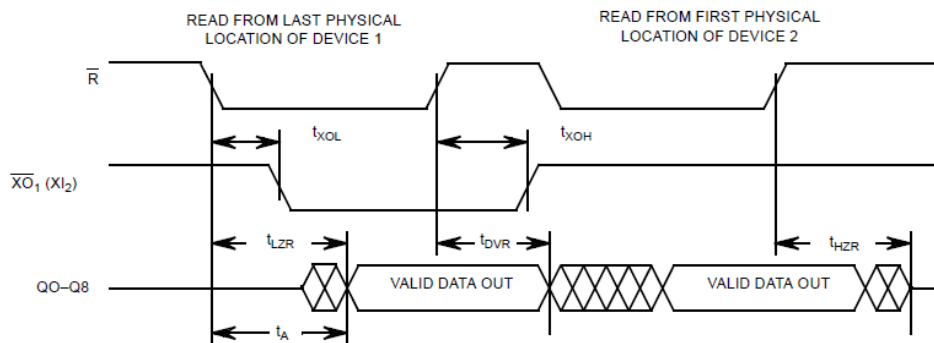
深さ拡大設計は幅拡大と同様に、FIFO チェーン全体の瞬間状態を反映するために複合の状態フラグを生成する必要があります。

図 7. 書き込み拡大タイミング



デバイス 1 の拡大出力 (\overline{XO}_1) はデバイス 2 の拡大入力 (XI_2) に接続されます。

図 8. 読み出し拡大タイミング



デバイス 1 の拡大出力 (\overline{XO}_1) はデバイス 2 の拡大入力 (XI_2) に接続されます。

再送信機能

再送信機能は、遠隔通信アプリケーションでデータ パケットを再送信するのに、ディスク ドライブでセクターを書き換えるのに役立ちます。これは、ワードやパターン発生器などと同様に FIFO の単一のデータ ブロックが外部に複数回送信される必要があるアプリケーションで特に役に立ちます。

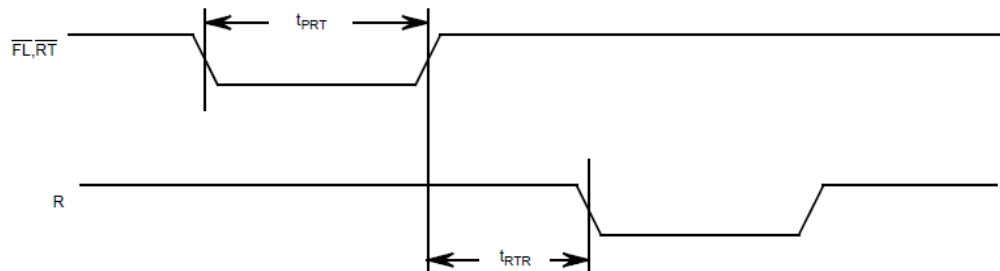
サイプレスの FIFO では、データを何回も再送信でき、また、FIFO が格納しているデータ量に関係なく再送信機能をいつでも利用できます。この点では、FIFO が full になった時、または必要より少ないワード数を持っている時に再送信が不可能な競争会社の FIFO 製品より優れています。

再送信動作では、読み出しポインターはその最初の位置にリセットされ、読み出しポインターが書き込みポインターに指定されるメモリ位置と同じ位置に到達するまで R ピンにパルスが発生します。再送信 (RT) ピンはスタンドアロンと幅拡大モードで利用可能です。深さ拡大モードは再送信機能をサポートせず、このピンは最初にロードされる FIFO を指定します。

再送信機能はアクティブ LOW パルスを再送信入力にアサートすることで開始され、内部読み出しカウンタを 0 にリセットします。R 入力はこの間に非アクティブに維持されます。そうしなければ、競合する要件が読み出しカウンタを破損する可能性があります。再送信プロセスは書き込みカウンタの状態や書き込みプロセスに影響を与えず、再送信と書き込みサイクルが重なったり同時に発生したりしても設計や使用ルールを違反しません。デバイスがロックアップされず、図 9 に示される再送信タイミング制約が満たされる限り、データは損失したり破損したりすることがありません。

書き込みと再送信が同時に実行されると、現時点でどのデータが FIFO に格納されているか、どのデータが読み出されているかを追跡することは複雑になることがあります。例えば、FIFO が書き込まれており、FIFO が half-full になると再送信機能がアクティブになるシナリオを考えてみましょう。FIFO は最初の位置からデータを再送信し／読み出し、その読み出しは empty の状態になるまで引き続きます。読み出しポインターが書き込みポインターに追い付き、(再送信がアクティブになった後 FIFO に書き込まれた) 新しいデータも読み出される前に、FIFO は 4 分の 3 埋まる場合があります。そのため、このようなシナリオではデータの追跡は困難になる場合があります。

図 9. 再送信タイミング



t_{PRT} は最小の再送信パルス幅です。

t_{RTR} は再送信回復時間です。これは違反してはならないタイミング ウィンドウです。

アプリケーション

FIFO は一時データまたは制御バッファを提供することで、異なるデータ速度で動作する 2 つのシステム間の通信を可能にします。

FIFO の一般的なアプリケーション例:

- プロセッサ間の通信
- ローカル エリア ネットワークを含む通信システム

- リアルタイム データのバッファリング用のデジタル信号処理ベースのシステム
- 高性能ディスク コントローラーを含む電子データ処理、CPU、周辺機器

設計時の注意事項および対策

以下のセクションでは、サイプレスの非同期 FIFO の設計際に注意すべき幾つかの点およびそれぞれの対策を説明します。

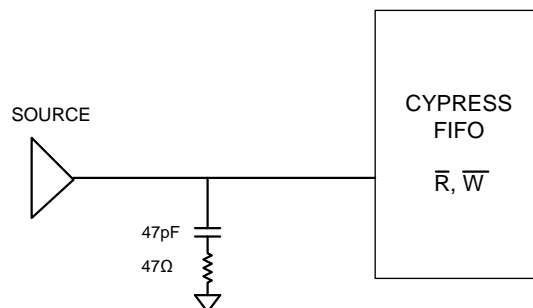
破損または繰り返しデータ

FIFO で発生する破損または繰り返しデータのもっとも一般的な原因は FIFO の W 入力にあるスプリアスのアクティブ信号 (グリッチ) です。書き込みグリッチが原因でデータ入力の論理レベルが FIFO に書き込まれます。これは、誤りデータがデバイスに書き込まれてしまう原因となります。有効なデータがデータ入力に存在していれば、書き込みグリッチはこのデータを重複させます。

書き込みグリッチは、インピーダンス不整合による電圧反射に起因することが多いです。インピーダンス不整合はインピーダンス整合終端ネットワークで除去できます。読み出し／書き込み信号の立ち上がり／立ち下がり時間が 2ns であることを前提とし、プリント回路基板 (PCB) 上で転送元から転送先への転送線が約 4 インチを超える場合、終端ネットワークを W と R 配線に使用することをお勧めします。R と W 信号の立ち上がり／立ち下がり時間が 2ns 未満の場合、1 インチくらい短い配線長は終端が必要です。

終端ネットワークは負荷インピーダンスを通常 50Ω 以下のプリント基板配線特性インピーダンスに一致させます (G-10 水晶エポキシ材でのマイクロストリップまたはストリップライン構築の場合)。電圧反射を最小限にするために、少し過減衰の終端をお勧めします。サイプレスは、読み出し／書き込みピンとグランド間を、47Ω 抵抗と直列に 47pF (最大値) コンデンサで接続することを推奨しています (図 10)。この終端ネットワークは高周波パルスのハイパス フィルターとして機能し、直流電力を消費しません。FIFO を 1 つ以上駆動する読み出しまたは書き込み配線には 1 つの終端ネットワークのみが必要です。転送元から電氣的に最も離れている入力でネットワークを接続します。複数の負荷の場合、最大ライン長を決定するために、「SRAM System Design Guidelines」ホワイト ペーパーを参照してください。

図 10. 推奨の終端ネットワーク

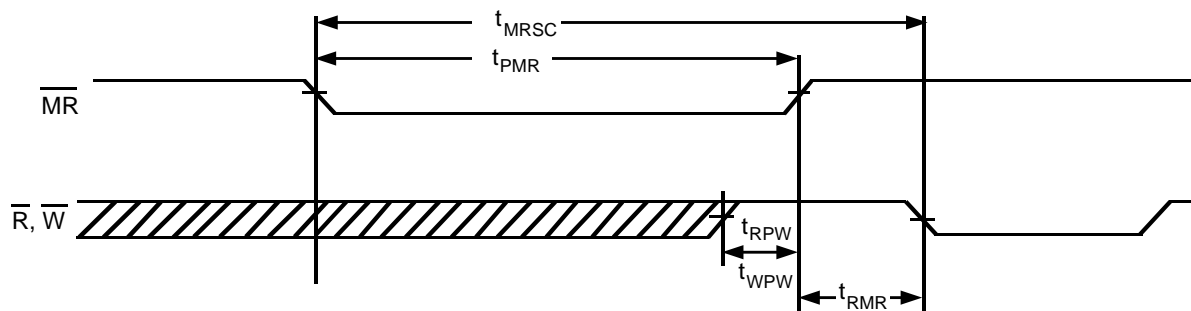


FIFO データの破損は、マスター リセットのタイミング制約違反によって引き起こされることがあります。図 11 に示すように、読み出しおよび書き出し信号は t_{RMR} (マスター リセット回復時間) の仕様を満たすように、MR (マスター リセット) の立ち上がりエッジごろでアサートされるべきではありません。FIFO が内部初期化プロセスを経て、リセットが終了した後セッティング期間を必要とするため、この制約は必要です。

FIFO ロックアップ

MR ピンでの短いノイズ パルスは FIFO を「部分的にリセット」させ、その結果、FIFO が応答しないことがあります。この問題を防ぐために、適切な終端スキームを転送配線に組み込む必要があります。

図 11. マスター リセット タイミング



欠陥または消失データ

R 入力でのグリッチは意図しない読み出し動作によりデータを消失させることがあります。これは内部読み出しカウンターを増分し、その結果、データ損失を引き起こします。これらのグリッチを除去するために、適切な終端スキームを R ラインに組み込む必要があります。

繰り返しまたは順番誤ったデータ、偽の Full または Empty

位置ずれの内部読み出しまたは書き込みポインタは繰り返しまたは順番誤ったデータ、偽の full および／または empty の状態を含むさまざまな症状を引き起こす可能性があります。位置ずれのポインタの 2 つの最も一般的な原因はマスター リセット違反と境界条件違反です。

境界条件は FIFO が full または empty のいずれかの状態として定義されます。FIFO がより大きいワードを作るために並列に接続されている時、特定の条件では個々の FIFO は読み出し／書き込み要求を無視するか応答することがあります。個々の FIFO が異なって動作するシステム レベルの現象はワード位置ずれです。この問題は、読み出しが書き込み動作の直後に続く場合は empty の状態で、書き込みが読み出し動作の直後に続く場合は full の状態で発生します。

Empty 境界での動作

リセットされて、empty になっている FIFO を考えてみましょう。empty のフラグはアクティブ (LOW) で、内部ロジックは読み出し動作を防止します。一般的に、読み出しと書き込みの信号が非同期です。書き込み動作が完了すると、FIFO の内部状態は empty から (empty+1) に遷移します。この間隔の間に、読み出し動作が認識されることも、認識されないこともあります。書き込みの前の読み出し動作は無視され、書き込みの後に続く読み出しは無視されません。FIFO はこれらの条件の中で読み出しを認識するかを決定します。この不確実なウィンドウの間、読み出しが無視されるかどうかを決定することはできません。1 つの FIFO では、この不確実性は許容できます。しかし、2 つ以上の FIFO がより大きいワードを作るために並列に接続されている場合、読み出しを無視するものもあれば、無視しないものもあることがあります。

Empty 境界での待機

図 13 に Empty 境界での読み出しに発生する問題を防止するタイミングを示します。FIFO から読み出す任意のデバイスは、R 信号の HIGH から LOW への変化を開始する前に、書き込み動作の完了後 t_{RAE} の間待機する必要があります。W 信号の立ち上がりエッジは書き込み動作の完了を示します。

Full 境界での動作

単一の FIFO が full になった時に同様の状態が発生します。full フラグはアクティブ (LOW) で、内部ロジックは書き込み動作を防止します。読み出し動作の後に書き込み動作が続くと、FIFO は full から「full-1」に遷移してから full に戻ります。FIFO が full から「full-1」に遷移している間、書き込み動作は認識されるか認識されないことがあります。FIFO は状態遷移に限られた時間を要するため不確実性の開口部が適用され、この瞬間に届く書き込みコマンドが無視される可能性があります。

このタイミングを満たす 1 つの方法としては、empty フラグがアクティブになっている時に読み出し動作が阻止されるように、複合 empty フラグ (EF) で読み出し動作をゲートします。ただし、empty の FIFO への最初の書き込みの前またはその間に R 信号は LOW になることがあり、データは出力に正しく伝播することに注意してください。

Full 境界での待機

図 13 に Full 境界での書き込みに発生する問題を防止するタイミングを示します。FIFO へ書き込む任意のデバイスは、W 信号の HIGH から LOW への変化を開始する前に、読み出し動作の終了後 t_{WAF} の間待機する必要があります。R 信号の立ち上がりエッジは読み出し動作の終了を示します。

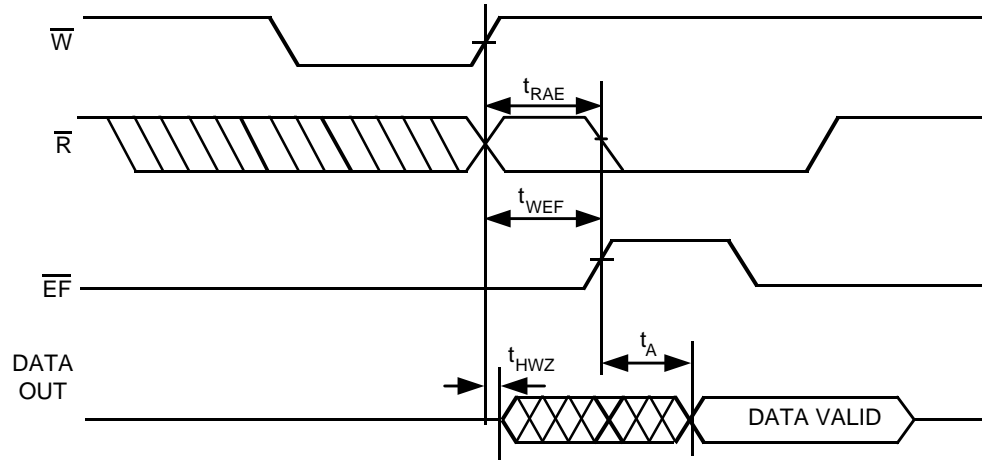
このタイミングを満たす 1 つの方法としては、full フラグがアクティブになっている時に書き込み動作が阻止されるように、複合 full フラグで (FF) 書き込み動作をゲートします。ただし、full の FIFO からの最初の読み出しの前またはその間に W 信号が LOW になることがあり、データは正しく書き込まれます。

Empty の読み出しと Full の書き込み

サイプレスの FIFO が empty になっている時、そのデータ出力は高インピーダンス状態になります。そのため、empty の FIFO から読み出そうとすると、予測できないデータが取得されます。外部ロジックは読み出しを防止し、読み出しポインタはインクリメントされません。

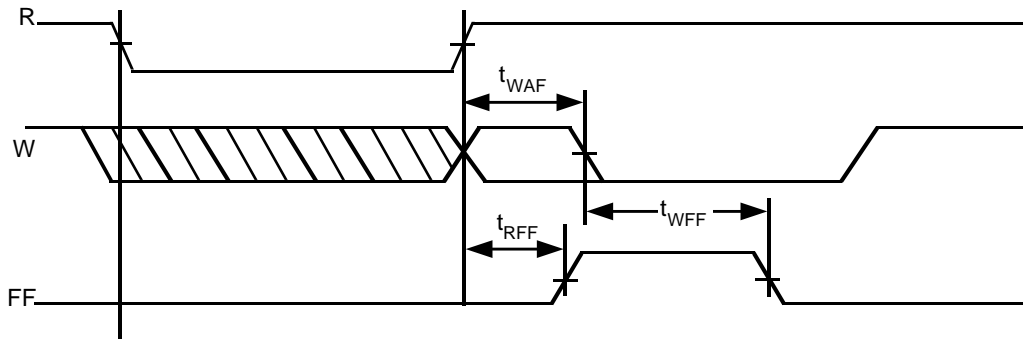
外部ロジックは full の FIFO への書き込みも防止し、書き込みポインタはインクリメントされません。

図 12. タイミング違反により読み出しが失敗



t_{RAE} は無効な読み出しウィンドウです。読み出し動作はこのウィンドウ以内に開始してはいけません。

図 13. タイミング違反により書き込みが失敗



t_{WAF} は無効な書き込みウィンドウです。書き込み動作はこのウィンドウ以内に開始してはいけません。

有効なパルス幅への違反

フラグが適切に使用されていない場合、この現象は empty または full 境界で発生する可能性があります。empty の FIFO からの読み出しを防止するためには empty のフラグを使用し、full の FIFO に書き込みを防ぐためには full フラグを使用してください。そうしない場合、実際の信号がデータシートの仕様を満たしていても読み出しまたは書き込みストロブの有効なパルス幅は違反されます。

FIFO が empty になり、読み出しパルスを受信している場合を考えてみましょう。FIFO が empty のため、この動作は無視されます。次の動作では、単一のワードが FIFO に書き込まれ、FIFO を (empty+1) に移動させます。一方、読み出し信号がアサートし続け、書き込み信号が読み出し信号の立ち上がりエッジの少し前に発生した場合、有効な最小 LOW 読み出しパルス幅への違反が発生します。

同様に、full になった FIFO に書き込もうとすることや、非同期的に読み出しを実行しようとすることによって最小書き込みパルス幅が違反されることがあります。empty と full のフラグは、これらの有効なパルス幅への違反を避けるために使用する必要があります。

まとめ

サイプレスの非同期 FIFO は、クロックドメインにわたってデータ同期のタスクを簡素化するデュアルポートデバイスです。状態フラグや再送信などの機能が組み込まれた簡潔なインターフェース、および幅／深さ拡大のサポートを提供することにより、これらのデバイスはプロセッサ間通信システムに最適です。

著者について

名前: Adithi Perepu
役職: アプリケーション エンジニア

改訂履歴

文書名: サイプレスの非同期 FIFO の理解 - AN1044

文書番号: 001-95840

版	ECN	変更者	発行日	変更内容
**	4722774	YYA	05/13/2015	これは英語版 001-25919 Rev. *G を翻訳した日本語版 001-95840 Rev. ** 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、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

車載	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
ワイヤレス/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

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 Phone : 408-943-2600
198 Champion Court Fax : 408-943-4730
San Jose, CA 95134-1709 Website : www.cypress.com

© Cypress Semiconductor Corporation, 2007-2015. 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はありません。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ス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ス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。