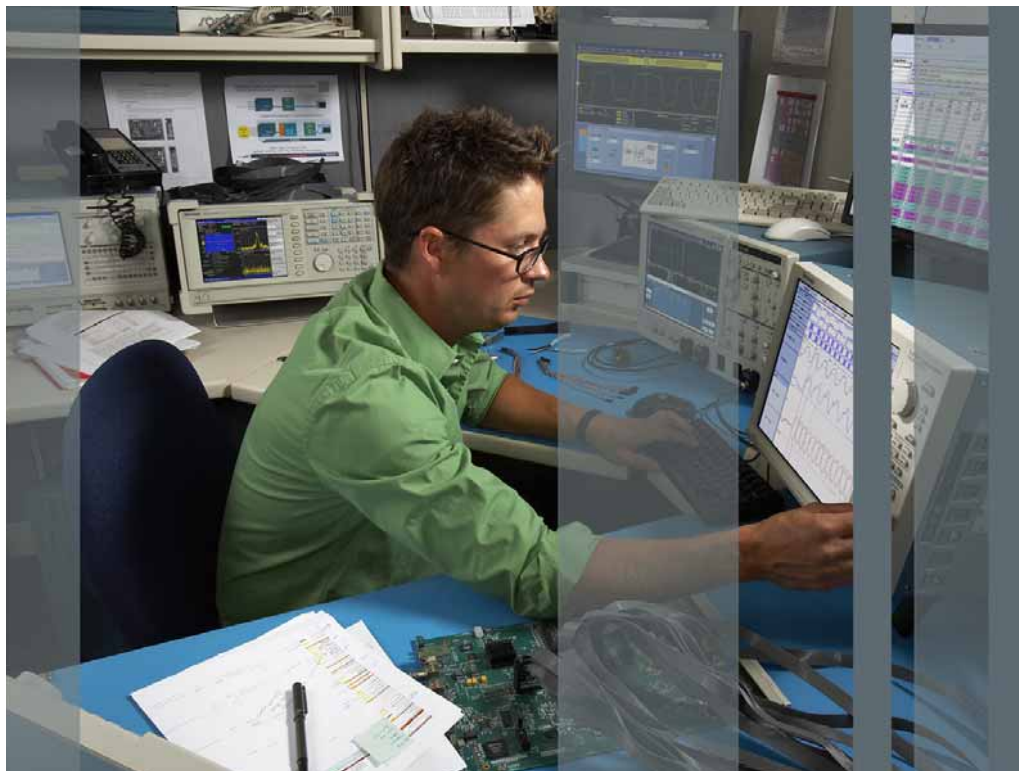




## DDR、DDR2、DDR3 SDRAMの コマンドとプロトコルの検証

### 複雑化、短期間化する設計サイクルと歩調を合わせる

大容量、高速、低電力、かつ物理的に小さいメモリを必要としているシステムは、コンピュータのメモリだけではなく、エンベデッド・システムのアプリケーションでも同様の要求があります。このアプリケーション・ノートでは、DDR、DDR2、DDR3 SDRAMのコマンドおよびプロトコルを検証する際のロジック・アナライザについて重点的に説明します。

## はじめに

DRAMの非同期動作は、同期型プロセッサとのインタフェースに際して多くの設計上の課題をもたらしました。SDRAM（同期型DRAM）は、DRAMの動作をコンピュータ・システムあるいはエンベデッド・システムのその他の部分と同期させ、シーケンス・ベースのメモリ動作において、異なるすべてのモードを定義する必要がないように設計されています。DDR SDRAMをはじめとするSDRAMの技術的進歩は、メモリのデータ・レート・パフォーマンスを積極的に向上させてきました。SDRAM、DDR SDRAM、DDR2 SDRAMおよびDDR3 SDRAMは、メモリ・コントローラ・ホストからメモリに送出されるメモリ・コマンドで制御されます。

## DDR、DDR2、DDR3 SDRAM

コンピュータで使用されるメモリには、大容量、高速化、低電力化、小型化などの要求が常にあります。これらのニーズにより、DRAM技術の進歩は加速されています。主要なDRAMは、SDRAM（Synchronous DRAM）、DDR（Double Data Rate）SDRAM、DDR2（Double Data Rate 2）SDRAM、DDR3（Double Data Rate 3）SDRAMなどによって発展してきました。

DDR（Double Data Rate）SDRAMでは、クロック・レート的高速化、データのバースト転送、1クロック・サイクルあたり2ビットのデータ転送により、メモリ・データ・レート性能を向上しています。

DDR2 SDRAMは、DDR SDRAMに対していくつかの改良が施されています。DDR2 SDRAMのクロック・レートはより高速になっており、メモリのデータ・レートも高速になっています。クロック・レートが高速になっているため、確実なメモリ動作のためにはシグナル・インテグリティが重要になります。クロック・レートが高速になると、回路基板の信号経路は伝送ラインになり、正しいレイアウトと信号経路の終端におけるターミネーションが重要になります。

DDR3 SDRAMはSDRAM技術の性能を改善し、拡張したもので、800Mbpsから始まっています。DDR3 SDRAMは、6種類のレベルのデータ・レートとクロック・スピードに対応しています。DDR3-800/1066/1333 SDRAMは2007年に実用化されており、DDR3-1600は2008年に、DDR3-1866/2133 SDRAMは2009年に実用化されると見込まれています。DDR3 SDRAMは1.5Vで動作します。DDR2 SDRAMの1.8V動作に比べて83%と小さいため、DDR3-1066 SDRAMはDDR2-800 SDRAMよりも低電力です。また、DDR2 SDRAMのデータDQドライバは18Ωですが、DDR3 SDRAMでは34Ωと高インピーダンスになっています。DDR3 SDRAMのメモリは512Mバイトから始まり、将来は8Gバイトになると見込まれています。DDR2 SDRAMと同様、DDR3 SDRAMのデータ出力は×4、×8、×16で構成されます。メモリ・サイズにもよりますが、DDR2 SDRAMでは4または8バンクですが、DDR3 SDRAMでは8バンクです。

| SDRAM     | データ・レート<br>MT/s | クロック<br>MHz | VDD<br>V |
|-----------|-----------------|-------------|----------|
| DDR-266   | 266             | 133         | 2.5      |
| DDR-333   | 333             | 166         | 2.5      |
| DDR-400   | 400             | 200         | 2.5      |
| DDR2-400  | 400             | 200         | 1.8      |
| DDR2-533  | 533             | 267         | 1.8      |
| DDR2-667  | 667             | 334         | 1.8      |
| DDR2-800  | 800             | 400         | 1.8      |
| DDR2-1066 | 1066            | 533         | 1.8      |
| DDR3-800  | 800             | 400         | 1.5      |
| DDR3-1066 | 1066            | 533         | 1.5      |
| DDR3-1333 | 1333            | 667         | 1.5      |
| DDR3-1600 | 1600            | 800         | 1.5      |

表1. SDRAMの規格

DDR2 SDRAMもDDR3 SDRAMも共に4つのモード・レジスタを持っています。DDR2 SDRAMでは2つのモード・レジスタが設定され、2つは予備となっています。DDR3 SDRAMでは、4つのモード・レジスタすべてを使用します。大きな違いは、DDR2 SDRAMのモード・レジスタでは読み込みでCASレイテンシを設定し、書き込みではモード・レジスタの読み込みレイテンシ設定-1に設定されます。DDR3 SDRAMのモード・レジスタでは、CASの読み込みレイテンシと書き込みレイテンシの両方を別々に設定します。DDR3 SDRAMでは8nプリフェッチ・アーキテクチャを使用しており、4クロック・サイクルで8データ・ワードを転送します。DDR2 SDRAMでは4nプリフェッチ・アーキテクチャを使用しており、2クロック・サイクルで4データ・ワードを転送します。DDR3 SDRAMのモード・レジスタはオンザフライ・バースト・チョップに対応するようプログラムされており、読み込みまたは書き込みコマンドでアドレス行12をローに設定することで、8データ・ワードを4データ・ワードに短くします。オンザフライ・バースト・チョップは、DDR2 SDRAMおよびDDR3 SDRAMに共通のアドレス行10をハイにして行われる読み込み時と書き込み時のオートプリチャージ機能の概念と似ています。

その他、DDR3 SDRAMで特長的なのが差動のデータ・ストロブDQSです。DDR2 SDRAMのデータ・ストロブは、モード・レジスタによってシングルエンドまたは差動にプログラムされています。DDR3 SDRAMには、アクティブ・ロー非同期RESET#という新しいピンがあり、現在のステートに関係なくSDRAMを既知のステートにすることでシステムの安定性を改善することができます。DDR3 SDRAMでは、DDR2 SDRAMと同じFBGAパッケージを使用しています。

DDR3 DIMMは、DIMM上にコマンド、クロック、アドレスのためのターミネーションを持っています。DDR2 DIMMを使用する多くのシステムでは、マザーボードでコマンド、クロック、アドレスを終端します。DIMM上のDDR3 DIMMターミネーションにより、フライバイ・トポロジが可能になり、SDRAMの各コマンド、クロック、アドレス・ピンは、DIMMの配線端で終端される1本の配線に接続されます。これによりシグナル・インテグリティが改善され、DDR2 DIMMのツリー・アーキテクチャに比べて高速な動作が可能になります。フライバイ・トポロジによる書き込みにおけるクロックCKとデータ・ストロブDQS間のタイミング・スキューを補償する新しいレベルの書き込み機能が、DDR3 SDRAMのメモリ・コントローラに導入されています。DDR3 DIMMはDDR2 DIMMと異なったキーになっており、間違ったDIMMがマザーボードに差し込めないようになっています。

SDRAMの設計の実装に際しては、確実なメモリ動作を保証するために、回路ボードの構造からソフトウェアの動作までの完全な検証と試験が必要になります。メモリ・システムの検証なくしては、製品の信頼性を保証できません。

ステートおよびタイミングを同時にアクイジションする機能を備えたロジック・アナライザは、メモリ・コマンドやプロトコルの検証およびデバッグに最適な装置です。また、20ps高分解能のタイミング・アクイジションにより、メモリ信号の電気特性も機能的にチェックできます。このアプリケーション・ノートは、メモリ・システムの正しいコマンドおよびプロトコルの動作を検証するためのロジック・アナライザの使用に焦点を当てています。

## DDR3のロジック・アナライザによるソリューション

当社のロジック・アナライザとNexus Technology社製のDDR3メモリ・サポート・パッケージにより、DDR3の読み込み／書き込みデータとDDR3コマンドを取込むことができます。TLA7BB4型ロジック・アナライザ・モジュールは、すべてのDDR3のスピードに対応できる唯一のモジュールです。全チャンネル同時に20ps (50GS/s)の高分解タイミング性能を持っており、グリッチ、タイミング問題、シグナル・インテグリティ問題を捕捉し、解析を可能にします。

このアプリケーション・ノートではほとんどの場合、64Meg×8ビット構成の512Mb DDR2 SDRAMを例にとっています。内部の構成は、16Meg×8ビット×4バンクです。

ここに紹介する例ではTLA7AA4型モジュールを使用しており、エンベデッド・システム、コンピュータ、ワークステーション、およびサーバで使用されるほとんどのSDRAM、DDR SDRAM、DDR2 SDRAM、DDR3 SDRAMに適用できますが、DDR3 SDRAMと多くのDDR2 SDRAMでは、TLA7BB4型モジュールの使用をお勧めします。推奨機種の詳細等については、当社担当営業またはお客様コールセンターまでご連絡ください。また、メモリの仕様、機能、および動作については、使用するメモリのデータ・シートをかならずご確認ください。

## ロジック・アナライザ・プロービング

ロジック・アナライザ・プロービングは、過去10年の間に進化してきました。最初、ロジック・アナライザ用プローブは、グラッパ・クリップを使用してICの脚に接続するか、または回路ボードに取付けられたスクエア・ピンに接続していました。デジタル設計が複雑化し、信号が高速になると、ロジック・アナライザによるプロービングはインピーダンスが制御されたクイック・コネクタ・タイプのMictorコネクタを回路ボード上で使用できるようになりました。

最先端のデジタル設計は、デジタル・クロック・レートをほぼ1GHzにまで押し上げました。その結果、プロービングの信号の忠実性がきわめて重要になりました。このようなきわめて高い周波数においては、コネクタのインピーダンスの不一致や、回路ボードのプローブ接続のためのフット・プリントの大きさが、問題を引き起こすことがあるため、プローブは高密度化されたコネクタレス接続タイプのプローブへと進化してきました。これらの高密度コネクタレス・プローブは、回路ボードのパッドあるいはトレースに直接接続するため、回路ボード上にコネクタを取り付けたり、それを使用する必要はありません。

回路ボードのMictorコネクタやコネクタレスのフット・プリントはいずれも、メモリ・システムを検証およびデバッグする際にロジック・アナライザをどのように使用するかについて、慎重な考察を必要とします。メモリ・システムに接続したロジック・アナライザ・プローブのシミュレーションを解析すれば、信号や、メモリに装着したプローブの影響を測定するのに最適なプロービング位置を知ることができます。このシミュレーションは、メモリ・システムの回路ボードを設計する前に行なうのが最良です。

ときには、最終製品の設計上、プロービングのテスト・ポイントを回路ボード上に残すことが許されない場合があります。そのような設計の検証およびデバッグ手段として、いくつかの方法が考えられます。第一は、ロジック・アナライザおよびオシロスコープのテスト・ポイントをフル装備した大きな評価用回路ボードを設計する方法です。評価とデバッグが済んだら、テスト・ポイントを減らすか、またはテスト・ポイントのない小さい回路ボードを設計します。第二は、Nexus Technology社などが提供しているロジック・アナライザで検証可能な専用DIMMを使用する方法です。DIMM、SO-DIMM、FB-DIMMの3つのタイプが用意されています。第三は、プロービング・インタポーザを使用する方法です。メモリの構成や信号の速度に応じて異なるタイプのものがあり、最も一般的なインタポーザは、メモリDIMMと回路ボード上のDIMMソケット間に設置されます。メモリICと回路ボード間に設置する別のタイプのインタポーザもあります。第四の方法として、プロービングのテスト・ポイントが回路ボードのエッジに隣接するエリアに位置するように回路ボードを設計します。最終的な設計では、回路ボードのテスト・ポイント・エリアをなくして回路ボードのサイズを小さくします。最後は、回路ボードにはテスト・ポイントが存在せず、回路ボードにハンダ付けしたアイソレーション抵抗の付いたワイヤにプローブを接続する方法です。

| コマンド          | SO# | RAS# | CAS# | WE# |
|---------------|-----|------|------|-----|
| Mode Register | 0   | 0    | 0    | 0   |
| Refresh       | 0   | 0    | 0    | 1   |
| Precharge     | 0   | 0    | 1    | 0   |
| Activate Row  | 0   | 0    | 1    | 1   |
| Write Column  | 0   | 1    | 0    | 0   |
| Read Column   | 0   | 1    | 0    | 1   |
| No Operation  | 0   | 1    | 1    | 1   |
| Deselect      | 1   | X    | X    | X   |

表2. SDRAMのコマンド

## ロジック・アナライザのチャンネルとグループの設定

SDRAMメモリ・コマンドは、メモリ・クロック（CK）の立上りエッジと同期して出力されます。コマンド信号は、チップ選択（SO# またはCS#）、行アドレス選択（RAS#）、列アドレス選択（CAS#）、ライト・イネーブル（WE#）の4つです。#記号は、これらがアクティブ・ロー信号であることを示しています（表2参照）。メモリのなかには、アクティブ・ハイ信号であるクロック・イネーブル（CKE）のような信号を使用しているものもあります。

メモリ・コマンドの検証は、ロジック・アナライザ・プローブの5つの信号（CK、SO#、RAS#、CAS#、WE#）を必要とします。TLAシリーズ・ロジック・アナライザの設定には、わずか1～2分しかかかりません。設定は、デフォルト・システムの設定から開始します。次に、ロジック・アナライザのSetupウィンドウで、5つのコマンド信号（CK、SO#、RAS#、CAS#、WE#）にプローブ

のチャンネルを割り当て、SO#、RAS#、CAS#、WE#チャンネルでコマンド・グループ名を作成します。コマンド・グループを作成すると、ロジック・アナライザのバスフォームの波形、Listingウィンドウ、トリガ、フィルタ、およびサーチでシンボリック・コマンド名を使用することが可能になります。

ロジック・アナライザのすべてのチャンネルが同じというわけではありません。クロック・チャンネルは外部クロッキング用で、クオリファイヤは有効なクロック・エッジの選別（クロック・クオリファイ）に使用します。その他のチャンネルは、通常のデータ・アクイジション・チャンネルです。たとえば、136チャンネルのTLA7AA4型ロジック・アナライザ・モジュールを使用する場合は、SDRAMメモリのCK信号をロジック・アナライザのCK1またはCK3の入力チャンネルに接続し、SDRAMメモリのコマンド信号（SO#、RAS#、CAS#、WE#）をロジック・アナライザのA1、A3、C3、またはE3プローブ・チャンネルに接続することが推奨されます。

これらの個々のロジック・アナライザ・プローブ・チャンネルを使用する理由は、1/2チャンネル・モードまたは1/4チャンネル・モードを使用して、より高速でより長いレコードを取込めるように設定を変更できることにあります。たとえば、TLA7BB4型において136のフル・チャンネル、内部タイミングを使用した場合、ロジック・アナライザの最大タイミング分解能は635ps (1.6GS/s)、チャンネル当たりの最大レコード長は64Mbとなります。1/4チャンネル・モードでは、使用可能なチャンネル数は本来のチャンネル数の4分の1に相当する34チャンネルと少なくなります。タイミング分解能は156ps (6.4GS/s) と4倍になり、チャンネル当たりのレコード長も128Mbと4倍になります。

最大限の柔軟性を確保するためには、フル・チャンネル、1/2チャンネル、および1/4チャンネルの内部クロッキング・モードで使用されるCK1、CK3、A1、A3、C3、およびE3プローブ・チャンネルからスタートするのがベストです。また、ステート・アクイジション・モードでは、TLA7BB4型ロジック・アナライザ・モジュールは、1/2チャンネルのアクイジション・モードを使用して、より高速のデータ・レートおよびより高速のクロック・レートの取込みが可能です。チャンネル数やメモリを減らしてタイミング分解能を高めるようにロジック・アナライザの設定を変更できるかどうかについては、ロジック・アナライザのマニュアルを参照してください。測定の設定でロジック・アナライザの全チャンネルを使用しない場合は、これらのプローブ・チャンネルを最初に使用します。

## ロジック・アナライザのスレッシュホールドの設定と信号のアクティビティ

ロジック・アナライザのスレッシュホールドは、測定対象の信号のタイプと、メモリのタイプによって異なります。ロジック・アナライザの比較回路は、設定されたスレッシュホールド電圧値に基づいて信号のロジック・ハイおよびロジック・ローを決定します。たとえば、DDR SDRAMはSO#、RAS#、CAS#、WE#のコマンド信号に対して0.9Vのスレッシュホールド電圧を設定します。0.9Vのスレッシュホールド電圧は、DDR2 SDRAMの $V_{REF(DC)}$ と同じです。DDR2 SDRAMのCKは差動信号で、ロジック・アナライザのスレッシュホールド電圧は0Vに設定されますが、50mVの小さなオフセットを使用する設計技術者もいます。メモリ・インタフェースでは、差動クロックとシングルエンドのデータ信号の使用がより一般的になりつつあります。そのために、当社のロジック・アナライザ・プローブは、差動クロック、差動オリファイヤ信号、およびシングルエンドのデータ信号をプロービングできるように設計されています。

DDR3 SDRAMの $V_{REF(DC)}$ は0.750Vであり、DDR2 SDRAMよりも低くなっています。

| Waveform     | Activity |
|--------------|----------|
| LA 1: Sample |          |
| LA 1: CLK    | ↑        |

図1. アクティブな矢印によって示されたDDR2 SDRAMのクロックの遷移

| Waveform     | Activity |
|--------------|----------|
| LA 1: Sample |          |
| LA 1: CLK    | —        |

| Waveform     | Activity |
|--------------|----------|
| LA 1: Sample |          |
| LA 1: CLK    | —        |

図2および図3. ロジック・ハイ・またはロジック・ロー状態にとどまり、DDR2 SDRAMクロックのエッジの遷移が起こっていないことが水平方向のアクティブなラインによって示されています。

検証に際しては、出荷時のデフォルト設定から始めました。まず最初に、Setupウィンドウでクロックとコマンド信号をプローブ・チャンネルに割り付けし、コマンド信号に対してコマンド・グループを作成し、スレッシュホールド電圧を適切に設定しました。トリガ設定は、trigger immediatelyに設定したままとしました。これは、最初に取り込まれたサンプルに対してトリガをかけることを意味し、trigger on anythingと同じです。Trigger immediatelyは、Runボタンが押されるのと同時にデータを取り込みます。Waveformウィンドウでは、表示波形にサンプル・マークとクロック信号が追加されます。Activityボタンをクリックすると、個々の波形ラベルの横に信号のアクティビティが表示されます。

アクティビティ・インジケータは、ロジック・アナライザでの取込みを開始する前にライブの信号をチェックできる優れた機能です。クロックに対しては、クロックの遷移を示す上下方向の矢印が表示されます（図1参照）。波形のアクティビティを示す矢印が表示されない場合（図2および図3）は、次のことをチェックします。

- メモリ・システムに電源が供給され、メモリ・システムが動作しているか
- 正しいプローブ・チャンネルがメモリ・システムの正しいテスト・ポイントに接続されているか
- ロジック・アナライザのスレッシュホールド電圧設定が差動クロック信号に対して正しいレベルであるか

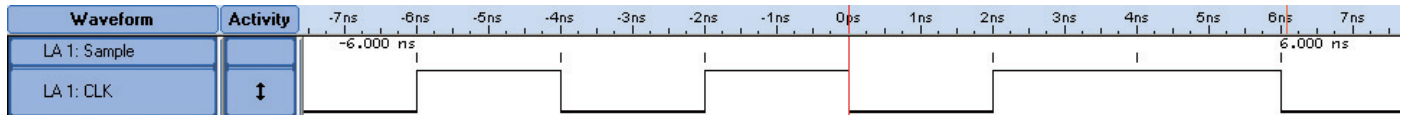


図4. 2nsのタイミング分解能で測定したDDR2 SDRAMのクロック

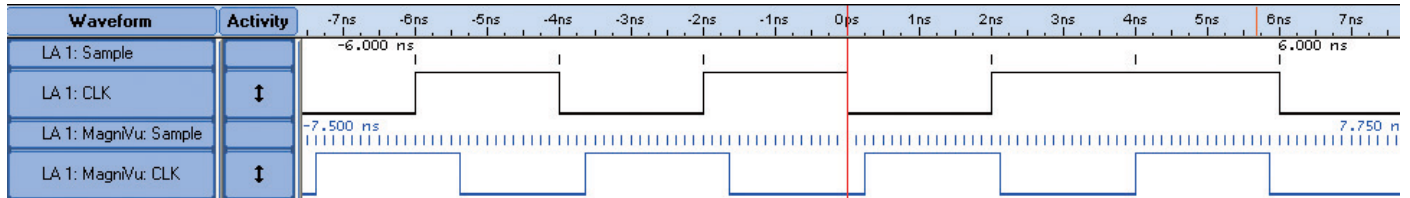


図5. DDR2 SDRAMのクロックを、上のディープ・タイミング波形で2nsのタイミング分解能で測定し、同時に下の波形でMagniVuの125psの高分解能タイミングで測定。125psのタイミング分解能ではクロック信号をより正確に表せることに注目してください。

## DDR2 SDRAMのクロック波形

Runボタンを押し、ロジック・アナライザでDDR2 SDRAMのクロックを取込みます。図4は、タイミング波形を示したものです。クロックを示す縦のサンプル・マークは、ロジック・アナライザのデフォルトである2nsというタイミング分解能で測定されていることを示しています。測定のタイミング分解能が高くなるほど、クロックを正確に測定できます。高いタイミング分解能でのクロックの測定方法には、2通りの方法があります。1つは、1/2チャンネルまたは1/4チャンネルのタイミング・モードを使用する方法です。もう1つの方法は、当社MagniVu®による125ps高分解能タイミングで取込む方法です。

MagniVuは、当社が特許を取得したデジタル・オーバ・サンプリング・ロジック・アナライザ・アーキテクチャで、個々のロジック・アナライザ・チャンネルごとに1つの高分解能サンプラを使用しています。TLAシリーズ・ロジック・アナライザは、どのアキュイジション・モードであっても、この高分解能サンプラによって取込まれた高分解能タイミング・データも保存します。つまり2つのロジック・アナライザ、すなわちMagniVuの高分解能タイミングと通常の汎用・タイミングあるいはステート・アキュイジションの機能

を1モジュールで実現していることになります。全てのデータには1つの高分解能サンプラ（MagniVuサンプラ）を通して取込まれているので、MagniVu高分解能タイミングと汎用・タイミングおよびステート・アキュイジションのデータは完全に時間相関がとれていることになります。

TLA7AA4型モジュールでは、MagniVuのサンプリング分解能は125psで、レコード長は16Kです。TLA7BB4型モジュールでは、MagniVuのサンプリング分解能は20psで、レコード長は128Kです。MagniVuでは、高分解能タイミングの取込みは常にオンになっており、データは保存されています。DDR2 SDRAMクロックの高分解能タイミング波形を表示するには、MagniVuボタンを押すだけです（図5参照）。

図5は、高分解能によるタイミングを示しています。DDR2 SDRAMのクロック信号は、同じロジック・アナライザ・プローブで2通りのタイミング分解能で測定されており、TLA7AA4型モジュールによるMagniVuの波形では、信号のエッジの不確定性が125ps以内であるのに対し、汎用・タイミングの波形ではエッジの不確定性は2nsです。MagniVuの125psの高分解能タイミングでは、信号をより正確に取込みます。

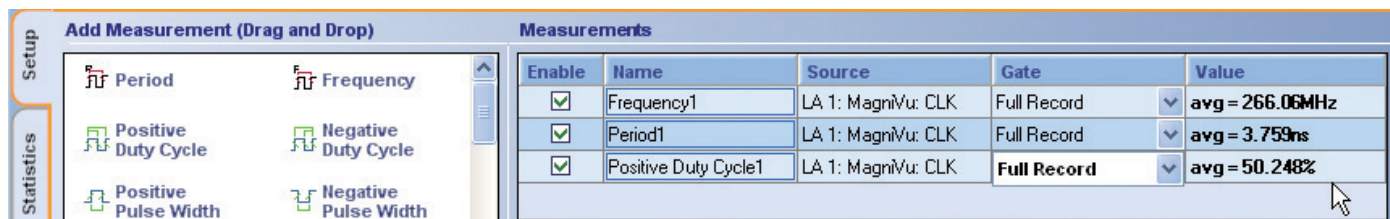


図6. DDR2 SDRAMのクロックの測定：周波数266.06MHz、周期3.759ns、正のデューティ・サイクル50.248%

## DDR2 SDRAMクロックの測定

ロジック・アナライザのドラッグ・アンド・ドロップ操作による自動測定機能を使用すると、DDR2 SDRAMのクロックの周波数、周期、およびデューティ・サイクルをすばやく測定できます。Waveformウィンドウの下にあるMeasurementタブで「周波数の測定」を選択し、MagniVu DDR2 SDRAMクロック波形までドラッグし、ドロップします。周期と正のデューティ・サイクルを測定する場合も同じように操作します。MagniVuの波形を使用する理由は、この波形が最高の時間分解能のデータだからです。測定の精度を上げるには、ゲートをDisplayからFull Recordに変更し測定範囲を広げます（図6参照）。測定メニューのStatisticsタブで、ポピュレーション数が、4（ゲートがDisplay時）から542（ゲートがFull Record時）に増加したことがわかります。

正のデューティ・サイクルの50.248%という値は、このメモリの45%～55%という仕様によく合致しています。たとえば、DDR2 533 SDRAMでは、クロック周波数は266.06MHzです。DDR2 SDRAMは通常、クロック周波数の2倍であるデータ・レートで表されます。

最高の精度で測定を行うための最良の方法は、当社の高性能オシロスコープを使用してクロック信号を測定する方法です。ジッタ、立上り時間、立下り時間、スペクトラム拡散クロッキング（SSC）といったクロックのその他の仕様は、DPOJETジッタ／タイミング解析ソフトウェアを搭載したオシロスコープを使用することで、より正確に測定できます。一般的にオシロスコープは高いアナログ帯域と高いサンプリング・レートを備えており、アナログ信号の測定に使用されますが、ジッタ／タイミング解析ソフトウェアの測定機能を利用することで、より高精度のアナログ測定、パラメータ測定を実現できます。これに対し、ロジック・アナライザのドラッグ・アンド・ドロップ操作による自動測定機能では、きわめて簡単かつ速やかにDDR2 SDRAMクロックの機能チェックを行うことができます。

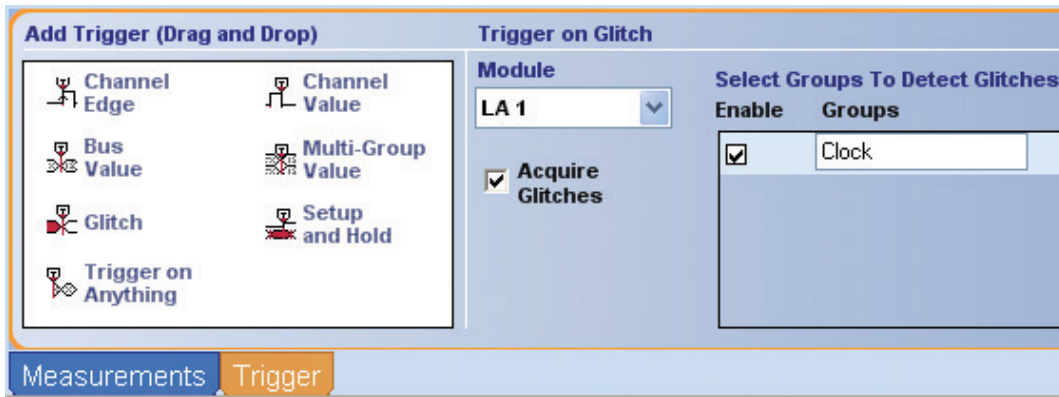


図7. ドラッグ・アンド・ドロップで簡単にトリガを設定し、DDR2 SDRAMのクロックのグリッチをチェック

### クロックのグリッチ・チェック

メモリを確実に動作させるためには、メモリ・コントローラ・ハブによって出力されるクリーンなDDR2 SDRAM用クロックが不可欠です。クロックのグリッチは、クロストーク、トレースのインピーダンス・エラー、信号の終端エラー、電源障害などによって発生します。

SDRAM、DDR SDRAM、DDR2、DDR3 SDRAMでは、ロジック・アナライザのグリッチ・トリガ機能を使用して、クロックのグリッチをすばやくチェックできます。Waveformウィンドウで、Measurementタブの右のTriggerタブを選択し、表示されたグリッチ・トリガ・アイコンをクロックのグループ波形上にドラッグし、ドロップします。この簡単な操作で、クロックのグリッチに対してのみトリガをかけるようにロジック・アナライザを設定したことになります（図7参照）。

当社のTLAシリーズ・ロジック・アナライザでは、グリッチは汎用・タイミングの1サンプル周期内に2つ以上のエッジが存在した場合として定義されています。TriggerウィンドウのAcquire Glitchesのチェック・ボックスで、赤いグリッチ識別フラグのオン／オフを切り替えられます。グリッチ・トリガでは、ロジック・アナライザはラン状態になった後、SDRAMクロック・ラインでグリッチに検出した場合にのみトリガがかかり、停止します。ロジック・アナライザのグリッチ・トリガをメモリのクロック・テストに

使用する際の注意点は、クロックのポジティブなパルス幅とネガティブなパルス幅のそれぞれが、汎用タイミングの最高速サンプル周期よりも大きくなければならないことです。

ロジック・アナライザをスタートさせてから昼食に出かけ、戻ったときに（あるいは週末、帰り際にロジック・アナライザをスタートさせて帰宅し、週明けに出勤してきたときに）ロジック・アナライザがまだデータ取込み中である場合は、SDRAMのクロック信号にグリッチが存在しなかったと確信できます。グリッチが存在した場合は、ロジック・アナライザとオシロスコープを統合したiView®機能が提供するディスプレイが、デバッグに必要なメモリ・プロトコルからデジタルおよびアナログ波形に至るまで、完全なメモリ・バスの波形表示機能を提供します。グリッチの検出によりロジック・アナライザにトリガがかかると、Waveformウィンドウのトリガ・ポイントの、汎用・タイミングのサンプル期間と同時間のMagniVu波形上に2つ以上のエッジが表示されます。クロックのグリッチをさらに検証するには、上記の統合システムを使用します。このオペレーションでは、まず、クロックのグリッチによりロジック・アナライザにトリガがかかります。ロジック・アナライザはさらに、グリッチのアナログ特性を取込むためにオシロスコープにトリガをかけます。ロジック・アナライザはオシロスコープが取込んだクロック波形のデータを取込み、オシロスコープと同様のアナログ波形をロジック・アナライザのデジタル波形と時間相関をとり表示します。



図8. ロジック・アナライザのPower Triggerプログラムを使用して、ロジック・アナライザの動作時間長とSDRAMのクロックのグリッチ数を計測する。

ドラッグ・アンド・ドロップ操作によるトリガ設定は、汎用タイミングのグループと個別の波形チャンネルに対してのみ使用できます。したがって、クロック波形をWaveformウィンドウに追加する場合は、クロック波形をプローブや名前ごとに追加するのではなく、クロック・グループとして追加しておく便利です。Waveformウィンドウでは、波形ラベルの左の拡張ボックスが波形のグループ表示であることを示します。波形グループに複数のチャンネルがある場合、波形グループはバスフォームの波形として表示されます。グループを展開（すべてのチャンネルを表示）するには、拡張ボックスのプラス記号をクリックします。グループ表示に戻すには、拡張ボックスのマイナス記号をクリックします。

どれくらいの頻度でグリッチが存在するかを知るためには、1. タイマをスタートさせる、2. ロジック・アナライザを停止させるまでグリッチ数を数える、の2つのステートを持つロジック・アナライザPower Triggerプログラムを作成します（図8参照）。タイマとカウンタの値は定期的に更新され、ロジック・アナライザのステータス・モニタに表示されます。最初のステートは、ロジック・アナライザが動作を開始すると同時にタイマをスタートさせます。第二のステートは、ロジック・アナライザを停止させるまでグリッチをカウントします。

当社のオシロスコープは、グリッチに対してトリガをかける機能も備えています。したがって、オシロスコープでクロックを測定する際には、グリッチに対してトリガがかかるようにトリガ機能を設定し、1～2分かけてクロック信号を検証してください。

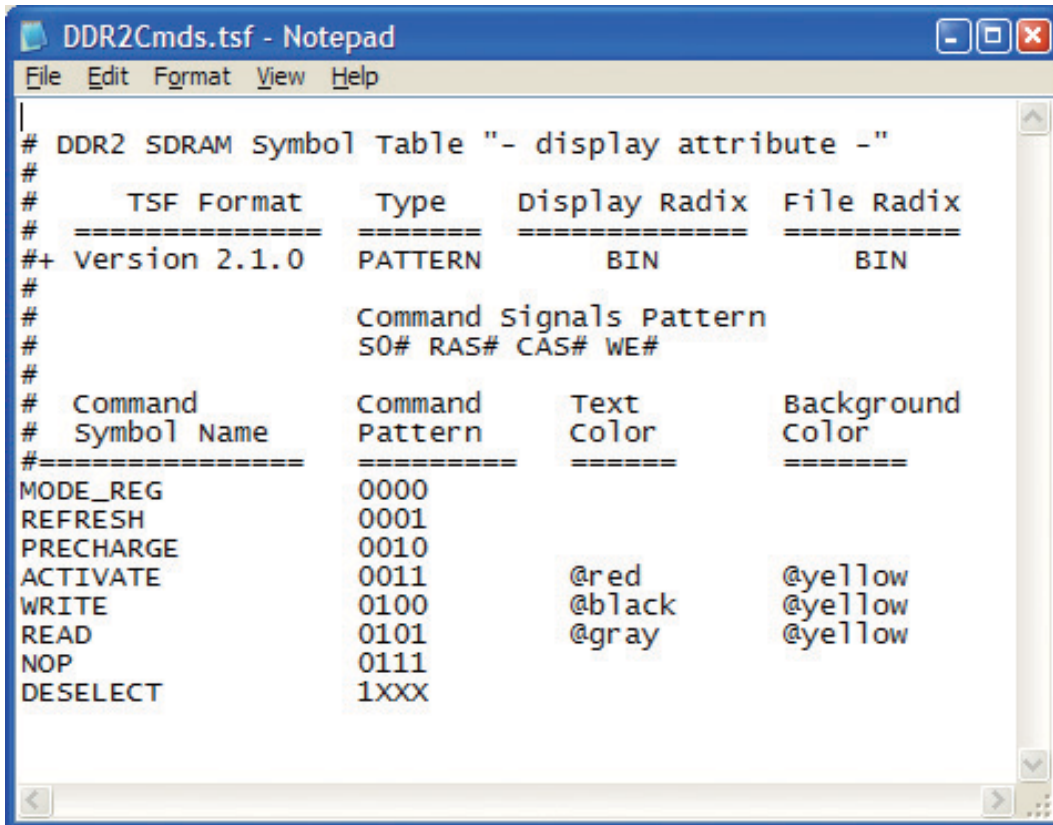


図9. DDR2 SDRAMのパターン・シンボル・ファイルの例

## SDRAMのコマンド信号の取込みおよび表示

メモリ・コントローラ・ハブは、SDRAMにクロックおよびメモリ・コマンドを送出します。SDRAMが導入される前は、RAS#、CAS#、およびWE#のエッジのタイミングとシーケンスがダイナミックRAM（DRAM）の動作モードを決定していました。同期型DRAM（SDRAM）は、SO#、RAS#、CAS#、およびWE#のコマンド信号をSDRAMクロックの立上りエッジでサンプリングします。クロックに同期した規則正しいSDRAMの同期オペレーションは、DRAMの非同期オペレーションに取って代わるものです。

SDRAMのコマンド・グループの表示の追加は、ロジック・アナライザのWaveformウィンドウで行います。直ちにトリガがかかるように設定した場合は、コマンド・グループの信号の変化が全く観測できないかもしれません。それは、メモリ・コントローラ・ハブがメモリの選択を解除している可能性があり、ロジック・アナライザのRunボタンを押したときにSDRAMにメモリ・コマンドが送出されなかったためです。

Activate Rowコマンドは、書き込みまたは読み込みのコマンド・シーケンスの最初のコマンドです。Activate Rowコマンドに対してロジック・アナライザのトリガをかけるためには、0011に相当するCommandグループ値に対してトリガがかかるようにロジック・アナライザを設定します。これは、表2に示すように、SO#=0、RAS#=0、CAS#=1、WE#=1となります。

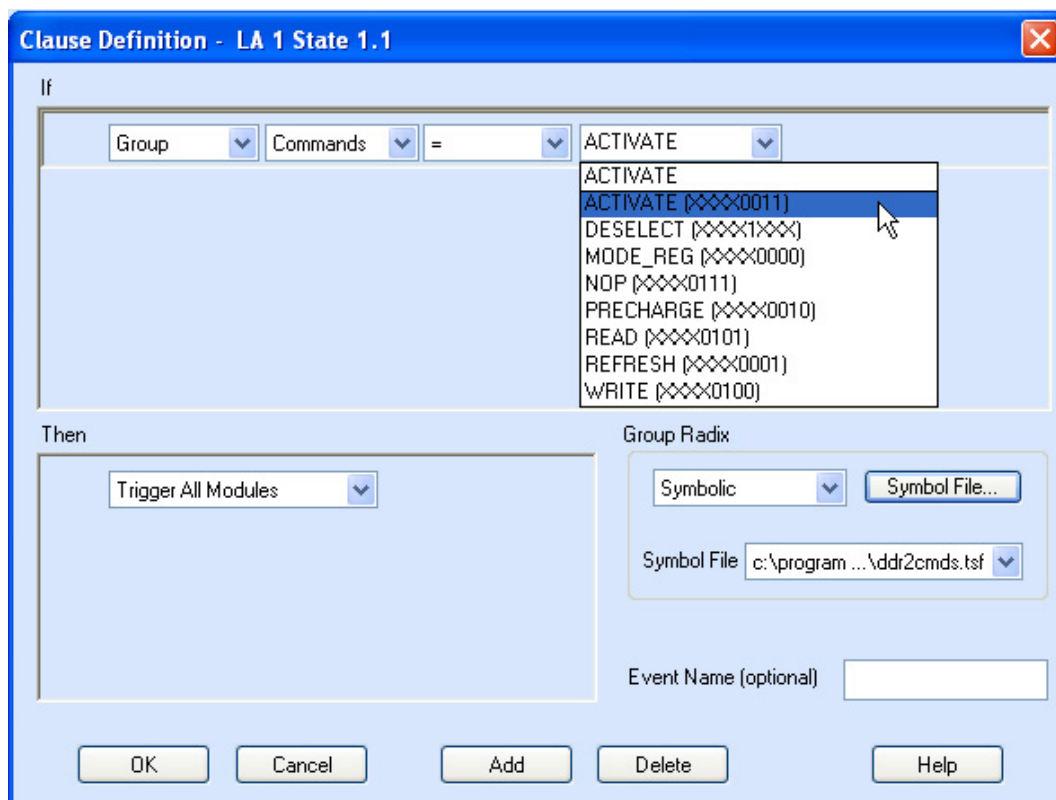


図10. ロジック・アナライザのPower TriggerプログラムでSDRAMのActivateコマンドに対してトリガをかける設定例

0011といったマジック・ナンバーを使った処理では間違いが起こりがちです。ロジック・アナライザは、2進数、8進数、16進数、10進数、符号付き10進数、記号など、複数のデータ・フォーマットをサポートしています。信号グループが、SDRAMのコマンド・グループのような論理的なステートを表す場合は、パターン・シンボルも使用できます。それぞれのパターン・シンボルは、色づけされています。表2に示されたSDRAMコマンド表を基に、Microsoft Notepadでテキスト・シンボル・ファイル（TFS）を作成し、DDR2Cmds.tsfファイルとして保存します（図9参照）。

このパターン・シンボルは、SDRAMのActivateコマンドでトリガがかかるようにロジック・アナライザを設定する際に使用されます（図10）。ロジック・アナライザのPower Triggerの設定では、コマンド・グループがActivateコマンドに等しいときにトリガがかかるようにロジック・アナライザが設定されます。Power Triggerの設定内でパターン・シンボルを使用するには、Group RadixをSymbolicに変更し、DDR2Cmds.tsfファイルを選択します。

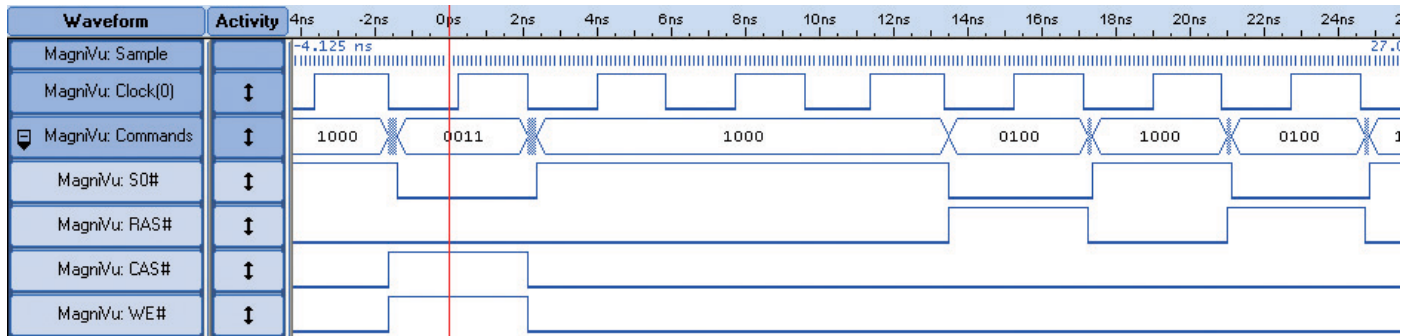


図11. CommandsグループのSDRAMのActivateコマンドに対してトリガ。Commandsグループは、SDRAMコマンドのSD#、RAS#、およびWE#信号を表示するために展開されています。

ロジック・アナライザに、SDRAMのActivateコマンド（バイナリ0011）でトリガがかかります（図11参照）。Commandsグループの波形は、Commandsバスのバイナリ値とともにバスフォームで表示され、信号のエッジの遷移は交差する線、または上下の縦線で示されます。

図11では、個々の信号を見るために、Commandsグループのラベルの左のプラス記号をクリックしてCommandsグループのバスフォームが展開されています。Commandsグループの表示形式はバイナリが選択されています。Opsにおける縦線は、トリガ・ポイントです。トリガ・ポイントでは、表1に示すように、Commandsグループはバイナリの0011で、Activateコマンドです。拡張されたコマンド波形は、トリガ・ポイントでSD#=0、RAS#=0、CAS#=1、WE#=1となります。

ロジック・アナライザのトリガ設定と同様に、Commandsグループのバスフォームの表示形式を変更します。Waveformのプロパティ内で、表示形式をシンボルに変更し、Power Triggerで使用したのと同じパターン・シンボルのDDR2Cmds.tsfファイルから読み込むように選択します。これにより、カラー表示のシンボルによるわかりやすいバスフォームを見ることができます（図12参照）。シンボルによるバスフォーム表示は、正しいSDRAMコマンド・プロトコル・シーケンスおよびタイミングの関係を検証するための強力なツールとなります。

## SDRAMの書き込み動作の検証

SDRAMの書き込み動作のプロトコル・シーケンスは、Activateコマンドと、その後のWriteコマンドで開始されます。行とバンクのアドレスを持ったActivateコマンドは、書き込みや読み込みのために特定のバンクの特定の行を開きます。その後、列とバンクのアドレスを持ったWriteコマンドが、書き込みのために、その開いた行の特定の列を開きます。開いている行がないバンクにWriteコマンドでアクセスしようすると、プロトコル・エラーになります。Writeコマンドの受け取り後、メモリはメモリ・コントローラ・ハブがデータを書き込むためのメモリ・ライト・サイクルに移行します。一般に、DDR2 SDRAMは4の倍数のデータ数で動作します。開いている行の書き込みが完了し、別の行にアクセスする場合は、Prechargeコマンドでその行を閉じる、あるいはオフにする必要があります。最も単純なDDR2 SDRAMコマンド・プロトコル・シーケンスは、Activate-Write-Prechargeです。書き込みを連続して行う場合のシーケンスは、Activate-複数のWrite-Prechargeとなります。書き込みの後、続けて読み込みを行うシーケンスは、Activate-Write-Read-Prechargeとなります。WriteとReadは、開いている行に対しては任意の順番で実行できます。

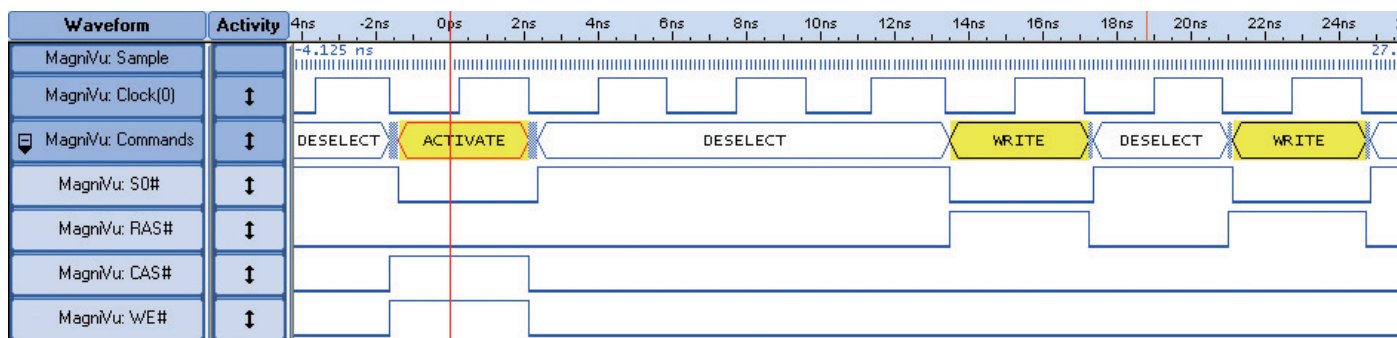


図12. SDRAMのActivateコマンドに対してトリガをかけ、Commandsグループのバス表示形式は色で識別されたコマンド・シンボルを使用。Commandsグループは、SO#、RAS#、CAS#、WE#の信号を表示するために展開されています。

図12は、Activateコマンドの後にWriteコマンドが続く書き込みシーケンスの冒頭を示したものです。DDR2 SDRAMは、ActivateコマンドとWriteコマンド間の最小時間 ( $t_{RCD}(MIN)$ ) が規定されています。また、この最小時間は、メモリの追加レイテンシにも依存します。与えられたCAS追加レイテンシ (AL) により、メモリの持つ帯域幅を連続して使用することによってコマンドおよびデータ・バスをより効率化できます。追加レイテンシにより、Activate-Write最小時間 ( $t_{RCD}(MIN)$ ) の前にWriteコマンドをメモリに送出することが可能になります。追加レイテンシの値は、メモリ・コントローラ・ハブにより、メモリの拡張モード・レジスタにプログラムされます。ここに示したDDR2 SDRAMの例では、追加レイテンシは使用していません。

DDR2 SDRAMの場合、追加レイテンシがゼロの場合には、Activateコマンドから $t_{RCD}(MIN)$ の時間内にWriteコマンドを送出してはなりません。ここに示した例では、 $t_{RCD}(MIN)$ は15nsに指定されています。 $t_{RCD}$ の値を検証する簡単な方法として、Activateコマンドの立上りクロック・エッジを右クリックしてそこにマーカを追加し、同様にWriteコマンドの立上りクロック・エッジを右クリックしてさらにマーカを追加する方法があります。見やすくするために、マーカをダブルクリックし、デフォルトの名前をActivateおよびWriteに変更できます。また、ActivateとWriteのマーカをクロックのエッジに調整する必要がある場合、マウスでマーカを選択し、クロックの波形上を水平方向に動かすと、マーカが自動的にクロック波形のエッジに移動します。マーカを波形のエッジに移動したくないときは、マーカを空白の波形スペース上で移動します。次に、デルタ時間測定ボックスを変更して、ActivateマーカとWriteマーカ間の時間差を表示させます。デルタ時間測定ボックスは、メニュー・バーから出して最適な場所に移動できます。

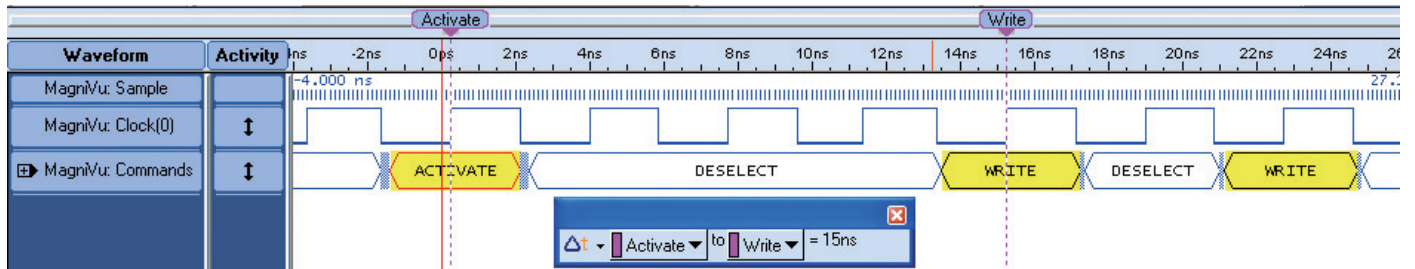


図13. ActivateコマンドとWriteコマンド間の $t_{RCD}$ の値（15ns）の測定

Writeコマンドは、Activateコマンドから15ns後に送出されています（図13参照）。もし測定値 $t_{RCD}$ が15ns未満で追加レイテンシゼロの場合は、メモリ・コントローラ・ハブがDDR2 SDRAM仕様を満たしていないことになります。

図13には、連続したWrite-Writeの動作の開始も示されています。2つ目のWriteコマンドは、最初のWriteコマンドから2クロック・サイクル後に送出されています。連続した2つのWriteコマンドの間には、Deselectコマンドがあります。メモリ・コントローラ・ハブが、2つのWriteコマンド間にDeselectコマンドを挟まずに送出すると、DDR2 DRAMプロトコル・エラーになります。DDR2 DRAMは、メモリ・コントローラ・ハブによってストローブされたデータを読み込むことで実際の書き込みを実行します。

DDR2 DRAMの最小バースト長は、4データ・ビットです。DDRは、ダブル・データ・レートの略で、クロック・サイクル当たり2つのデータ・ビットが書き込まれることを意味します。バースト長が4データ・ビットの書き込み動作を完了するためには、2つのクロック・サイクルが必要となります。これが、Writeコマンドの後にDeselectコマンドが存在する理由です。Deselectコマンドは、Writeコマンドの後に必要となる第二のサイクルのデータ書き込みのための時間を提供します。

DDR2 DRAMのもう1つの重要な仕様は、Prechargeコマンドが送出されてから行を開くActivateコマンドが送出されるまでの時間、最小 $t_{RP}$ です。これは、Prechargeコマンドに対してトリガがかかるようにロジック・アナライザの設定を変更し、PrechargeコマンドとActivateコマンド間の時間 $t_{RP}$ を測定することで簡単に検証できます。図13は、DDR2 DRAMの書き込み動作について示したのですが、DDR2 DRAMの読み込み動作にも同じプロトコルとタイミング検証方法を適用できます。

## ロジック・アナライザのステート・アキュイジションによるプロトコル検証

SDRAMのコマンド・プロトコル・シーケンスを取込み、検証するためのさらに効率的な手段として、ロジック・アナライザのステート・アキュイジションがあります。ステート・アキュイジションは、SDRAMクロックを使用してSDRAMコマンドをロジック・アナライザに取込みます。1つのSDRAMクロック・サイクルに1ラインのロジック・アナライザ・メモリが使用されるのに対し、タイミング・アキュイジションでは複数ラインのロジック・アナライザ・メモリが使用されます。ステート・アキュイジションは、外部クロックまたは同期アキュイジションとも呼ばれます。

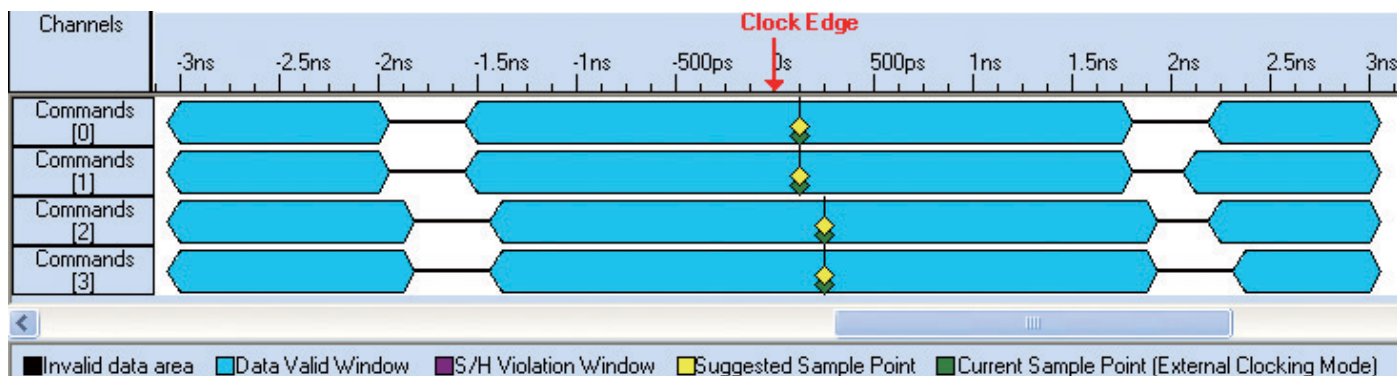


図14. ロジック・アナライザのAuto Deskew解析は、理想のサンプル・ポイントのポジションを黄色の菱形マークで提示します。この例では、現在のサンプル・ポイントのポジションを表す緑の菱形マークは、提示されたサンプル・ポイントのポジションと同じポジションに設定されています。

クロック同期の信号であるSDRAMコマンドには、SDRAMのクロック・エッジに対してセットアップとホールド時間仕様が存在します。これは、データ有効ウィンドウとも呼ばれます。メモリ・コマンド信号のデータ有効ウィンドウは、SDRAMクロックの立上りエッジを取り囲んでいます。ロジック・アナライザには、SDRAMクロックを持つSDRAMコマンドを取込むためのSDRAMコマンド信号用データ有効ウィンドウがあります。ロジック・アナライザは、ロジック・アナライザの個々のチャンネルのデータ有効ウィンドウを調整して、信号を確実に取込めるようにポジションを最適化できるという柔軟性があります。

ロジック・アナライザのAuto Deskew解析機能は、各チャンネルごとにロジック・アナライザのデータ有効ウィンドウの最適ポジションを検出できます。

Auto Deskew機能は、コマンド信号エッジのトランジションをサーチします。コマンド信号エッジのトランジションは、外部SDRAM立上りクロック・エッジを基準にします（図14参照）。黒の横線は、エッジのトランジションのためにデータが無効であることを示しています。青の太いバーは、エッジのトランジションがないデータ有効ウィンドウを示しています。黄色の菱形マークは、Auto Deskew解析により推奨されたロジック・アナライザの理

想サンプル・ポイント・ポジションを表しています。緑の菱形マークは、ロジック・アナライザの現在のサンプル・ポイント・ポジションを表し、ここでは提示されたサンプル・ポイント・ポジションと一致するように調整されています。Commandsの[3]、[2]、[1]、[0]のチャンネルは、SO#、RAS#、CAS#、WE#のコマンド信号を表しています。

ここに示した例では、SO#、RAS#、CAS#、WE#のコマンド信号は正しく動作し、よく似た時間帯にエッジが存在し、無効データ領域は狭くなっています。Chip Select SO#がアクティブでない場合、RAS#、CAS#、WE#のエッジはクロック・サイクル中のどの時点でも存在し得ます。その場合、Auto Deskewは、これらのエッジを観察し、クロック・エッジの周りに非常に狭いデータ有効ウィンドウを表示します。データ有効ウィンドウは、表示されない場合もあります。Auto Deskew解析は、常にChip Select SO#を含んで行います。それはSO#が常にアクティブな時だけコマンドが有効だからです。Auto Deskew解析でSDRAMのクロック・エッジの周りにデータ有効ウィンドウが表示されない場合は、SO#の提示サンプル・ポイントをRAS#、CAS#、およびWE#にも適用することからはじめてください。

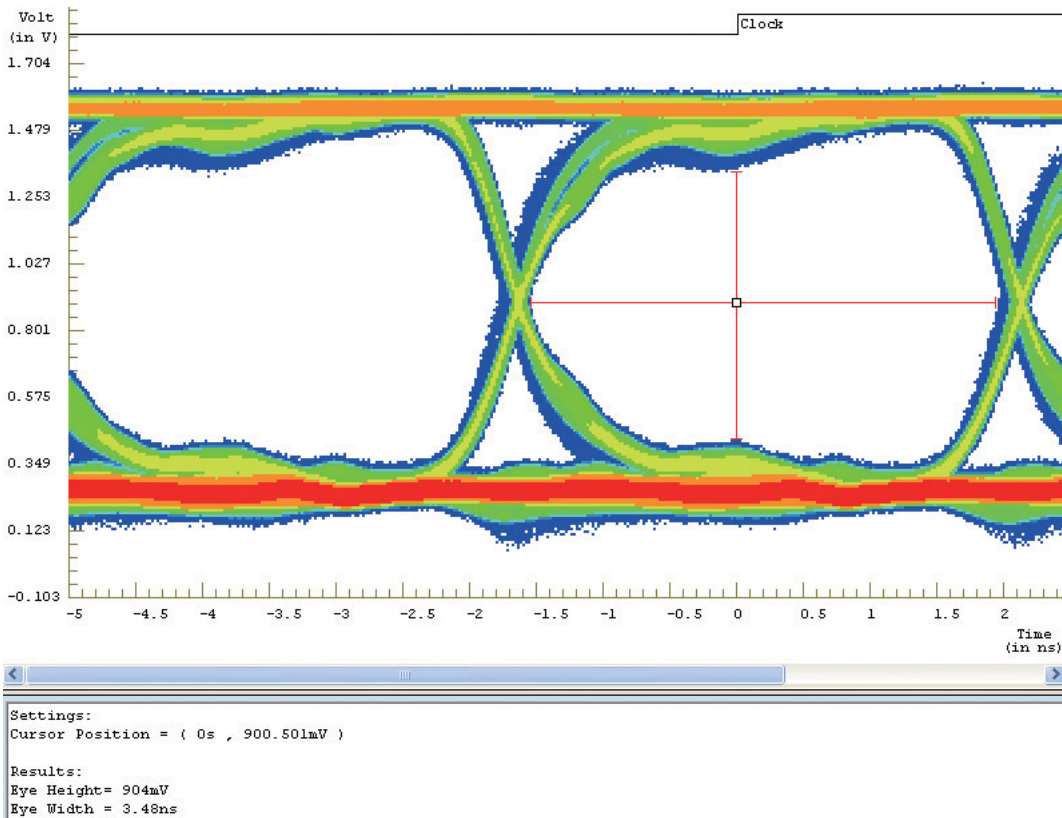


図15. SDRAMのクロックの立上りエッジのSO#、RAS#、CAS#、およびWE#の多チャンネル・アイ・ダイアグラム。  
カーソル位置のアイ・オープニングは、高さ904mV、幅3.48nsです。

推奨されたサンプル・ポイント・ポジションの検証方法には、少なくとも4つあります。その第一は、Auto Deskewのセットアップ/ホールド時間違反解析を使用する方法です。第二は、推奨されたサンプル・ポイント・ポジションで、セットアップ/ホールド時間違反でのトリガ機能を利用する方法です。第三は、CK、SO#、RAS#、CAS#、WE#のMagniVu®高分解能波形でデータを解析して評価する方法です。第四の方法は、ロジック・アナライザとオシロスコープを統合し、iVerify®解析機能を利用して複数チャンネルのアイ・ダイアグラムとして解析する方法です（図15参照）。

iCapture®マルチプレキシングと呼ばれる当社ロジック・アナライザのユニークな機能は、ロジック・アナライザのプロープを通してロジック・アナライザと外部のオシロスコープに同時に信号を導ける機能です。ロジック・アナライザは、iViewケーブルを使用してオシロスコープを制御し、トリガをかけ、オシロスコープの波形データを取込みます。iVerify解析機能により、ロジック・アナライザのディスプレイ上に複数チャンネルのアイ・ダイアグラムを作成できます。すべてのアイ・ダイアグラムを同時に解析することも、1つのチャンネルだけをハイライト表示にすることも可能です。図15は、SO#、RAS#、CAS#、およびWE#の複数チャンネル・アイ・ダイアグラムを示したものです。カーソルで測定したクロックの立上りエッジ点のアイ・オープニングは、高さ904mV、幅3.48nsです。このiVerify機能でのアイ・ダイアグラム解析は、SO#、RAS#、CAS#、WE#信号の詳細解析を迅速に実現できる最適な方法です。

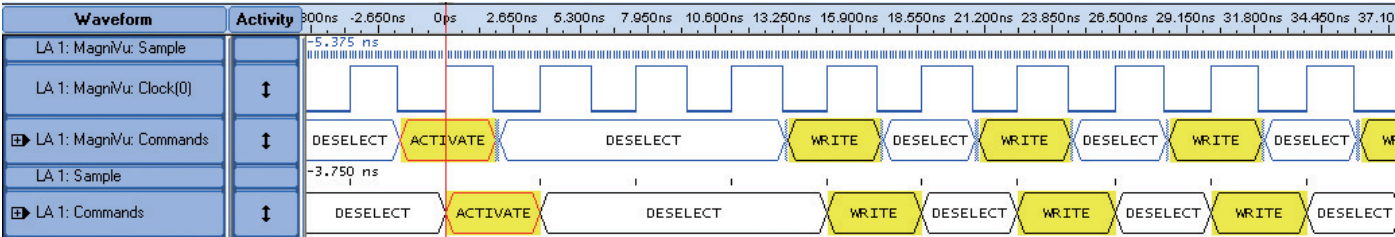


図16. Waveformウィンドウの一番下にあるのは、外部クロックによるCommandsのステートのバスフォームです。上の波形とバスフォームは、SDRAMのクロックとコマンドのMagniVuの125ps高分解能タイミング波形です。

Chip select SO#が非アクティブのときには、RAS#、CAS#、およびWE#はクロック・サイクル中のどこにでも変更できます。ただし、このような変更を行うと、メモリの選択が解除された際に発生したエッジの遷移によってアイ・ダイアグラムが無効になる場合があります。iVerifyのSetupメニューには、ロジック・アナライザの任意のチャンネルでクロックの立上りエッジをクオリファイするためのメニューが用意されています。したがって、有効なアイ・ダイアグラムを取込むために、必要なクロックの立上りエッジをクオリファイする信号としてSO#アクティブ・ローを指定します。

Waveformウィンドウの上段と中段には、MagniVuの125ps高分解能タイミングのSDRAMクロックとコマンド・グループが表示されています（図16参照）。下段はステート・アキュイジションのCommandsバスフォームで、そのデコードにはパターン・シンボ

ル・ファイルが使用されます。Commandsのステートのバスフォームの上に表示されているサンプル位置を示すティック・マークをよく見てください。ロジック・アナライザにCommandsバスをサンプリングさせているのはSDRAMクロック・エッジであるため、これらのティック・マークは、SDRAMクロックの立上りエッジでしか発生していません。そのため、ロジック・アナライザはSDRAMのクロック・サイクル中でコマンドを1回だけサンプリングし、CommandsのステートのバスフォームはSDRAMのクロックの立上りエッジにおいてのみ値が変化します。ステート・アキュイジションの利点は、DDR2 SDRAMがCommandsバスを見るのと同じ方法で、ロジック・アナライザでCommandsバスを検証することができることにあります。

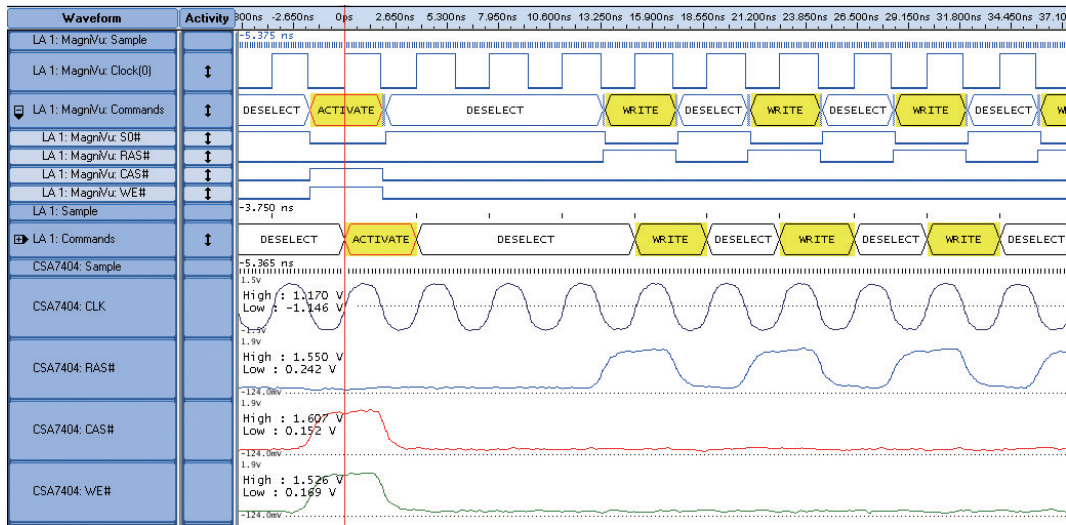


図17. ロジック・アナライザのiCapture対応のプロープでは、同時に3つの異なるタイプのSDRAMクロックおよびコマンド測定のための波形観測が可能です。上段の波形は、MagniVu による125psの高分解能タイミングです。中央のCommandsのバスフォームは、ステート・アクイジションのデータです。下の4つの波形は、iViewインタフェースを使用したオシロスコープで取込んだCLK、RAS#、CAS#、およびWE#波形です。

## iLink®ツール・セット：2つのパワフルな測定ツールの組み合わせ

ロジック・アナライザとオシロスコープは、長年にわたってデジタルのトラブルシューティングのためのツールとして選ばれてきましたが、この主要な2つの装置を統合することで実現される優れたメリットをご存知ないエンジニアも多いでしょう。

ロジック・アナライザは、デジタル情報の流れをチェックしながら回路の問題点を検出してトリガをかけ、関連イベントを取込むことでデバッグと検証を迅速化します。オシロスコープは、デジタルのタイミング・ダイアグラムの背後にあるアナログ特性に焦点を当て、生のアナログ波形を表示し、信号品質の問題を迅速に解析します。

当社のロジック・アナライザは、業界唯一のロジック・アナライザとオシロスコープの統合パッケージであるiLinkツール・セットをサポートしています。iLinkツール・セットは、当社のTLAシリーズのロジック・アナライザが備えている20ps分解能のMagniVu アクイジション、および先端のステート・マシン・ベースのトリガ・パワーと、最先端のオシロスコープを統合します。

iLinkツール・セットの一連のパワフルな機能により、ロジック・アナライザのディスプレイに時間相関の取れたデジタル信号とアナログ信号表示が可能になります。ロジック・アナライザが信号をデジタル形式で取込み、表示するのに対し、接続されたTDSシリーズ・オシロスコープは同じ信号をアナログ形式で取込み、ロジック・アナライザの画面に表示します。デジタルとアナログの表示を同時に観察すれば、アナログ領域におけるグリッチがいかにしてデジタル領域のタイミングの問題を引き起こしているかを簡単に検証できます。

iLinkツール・セットは、問題の検出とトラブルシューティングを迅速化することを目的に設計された包括的なパッケージです。

- iCapture®マルチプレキシングは、1本のロジック・アナライザ・プロープでデジタルとアナログの同時取込みを可能にします。
- iView®は、ロジック・アナライザの画面上に、オシロスコープのアナログ信号を時間相関をとりながら同時に表示します。
- iVerify®は、オシロスコープのアイ・ダイアグラム・データをロジック・アナライザに転送して、複数チャンネルのアナログ特性を一度に検証します。

当社のiCaptureとiViewは、ロジック・アナライザとオシロスコープ間のオペレーションを統合します。オシロスコープのCLK、RAS#、CAS#、およびWE#波形は、ロジック・アナライザのステート・アクイジションのCommandsバスフォーム、およびSDRAMのクロックとコマンドのMagniVu 125ps高分解能タイミングと時間的に相関付けられます（図17参照）。アナログ、デジタル・タイミング、およびデジタル・ステートの完全な波形表示は、iCaptureロジック・プローブを通じて得られます。ロジック・アナライザがオシロスコープにトリガをかけることも、オシロスコープがロジック・アナライザにトリガをかけることも可能です。当社のiCapture、iView、およびiVerifyは、iLinkツール・セットの一部であり、メモリの検証およびデバッグのための強力な機能を提供します。

Listingウィンドウは、Commandsのステート・データを解析するためのさらに効率的な手段です（図18参照）。最初のカラムにはサンプル番号が表示されます。2つ目のカラムには、前述のパターン・シンボル・ファイルでデコードされたCommandsのステート・データが表示されます。最後のカラムは、125ps分解能のタイムスタンプが表示されます。このタイムスタンプ表示では、先頭からの時間、トリガからの時間、前のサンプルからの時間（図18参照）、その他のバリエーションといった何通りもの設定が可能です。中央の赤い横線は、トリガ・ポイントを表しています。トリガの対象はActivateコマンドで、Activateコマンドの前には一連のDeselectコマンドが存在します。このActivateコマンドから、長期間のWrite-Write動作のシーケンスが始まっています。Activateコマンドの後ろの3つのDeselectコマンドは、Writeコマンドを送出するのに必要なDDR2 SDRAMの最小 $t_{RCD}$  (MIN) の要件を満たすためのものです。WriteコマンドとWriteコマンドの間のDeselectコマンドは、4データ・ビットのバースト・データをメモリに書き込むために、先行するWriteコマンドの後に時間を確保するためのものです。

| Sample | Commands | Timestamp |
|--------|----------|-----------|
| 65506  | DESELECT | 3.750 ns  |
| 65507  | DESELECT | 3.750 ns  |
| 65508  | DESELECT | 3.750 ns  |
| 65509  | DESELECT | 3.750 ns  |
| 65510  | DESELECT | 3.750 ns  |
| 65511  | DESELECT | 3.750 ns  |
| 65512  | DESELECT | 3.875 ns  |
| 65513  | DESELECT | 3.750 ns  |
| 65514  | DESELECT | 3.750 ns  |
| 65515  | DESELECT | 3.750 ns  |
| 65516  | DESELECT | 3.750 ns  |
| 65517  | DESELECT | 3.750 ns  |
| 65518  | DESELECT | 3.750 ns  |
| 65519  | DESELECT | 3.750 ns  |
| 65520  | DESELECT | 3.750 ns  |
| 65521  | DESELECT | 3.750 ns  |
| 65522  | DESELECT | 3.750 ns  |
| 65523  | DESELECT | 3.750 ns  |
| 65524  | DESELECT | 3.750 ns  |
| 65525  | DESELECT | 3.750 ns  |
| 65526  | DESELECT | 3.750 ns  |
| 65527  | DESELECT | 3.875 ns  |
| 65528  | DESELECT | 3.750 ns  |
| 65529  | DESELECT | 3.750 ns  |
| 65530  | DESELECT | 3.750 ns  |
| 65531  | DESELECT | 3.750 ns  |
| 65532  | DESELECT | 3.750 ns  |
| 65533  | DESELECT | 3.750 ns  |
| 65534  | DESELECT | 3.750 ns  |
| 65535  | ACTIVATE | 3.750 ns  |
| 65536  | DESELECT | 3.750 ns  |
| 65537  | DESELECT | 3.750 ns  |
| 65538  | DESELECT | 3.750 ns  |
| 65539  | WRITE    | 3.750 ns  |
| 65540  | DESELECT | 3.750 ns  |
| 65541  | WRITE    | 3.750 ns  |
| 65542  | DESELECT | 3.875 ns  |
| 65543  | WRITE    | 3.750 ns  |
| 65544  | DESELECT | 3.750 ns  |
| 65545  | WRITE    | 3.750 ns  |
| 65546  | DESELECT | 3.750 ns  |
| 65547  | WRITE    | 3.750 ns  |
| 65548  | DESELECT | 3.750 ns  |
| 65549  | WRITE    | 3.750 ns  |
| 65550  | DESELECT | 3.750 ns  |
| 65551  | WRITE    | 3.750 ns  |
| 65552  | DESELECT | 3.750 ns  |
| 65553  | WRITE    | 3.750 ns  |
| 65554  | DESELECT | 3.750 ns  |
| 65555  | WRITE    | 3.750 ns  |
| 65556  | DESELECT | 3.750 ns  |
| 65557  | WRITE    | 3.875 ns  |
| 65558  | DESELECT | 3.750 ns  |
| 65559  | WRITE    | 3.750 ns  |
| 65560  | DESELECT | 3.750 ns  |
| 65561  | WRITE    | 3.750 ns  |
| 65562  | DESELECT | 3.875 ns  |
| 65563  | WRITE    | 3.750 ns  |
| 65564  | DESELECT | 3.750 ns  |
| 65565  | WRITE    | 3.750 ns  |
| 65566  | DESELECT | 3.750 ns  |
| 65567  | WRITE    | 3.750 ns  |
| 65568  | DESELECT | 3.750 ns  |
| 65569  | WRITE    | 3.750 ns  |
| 65570  | DESELECT | 3.750 ns  |
| 65571  | WRITE    | 3.625 ns  |

図18. ListingウィンドウでのSDRAMコマンドのステート・アクイジションとコマンドのシンボル名の表示

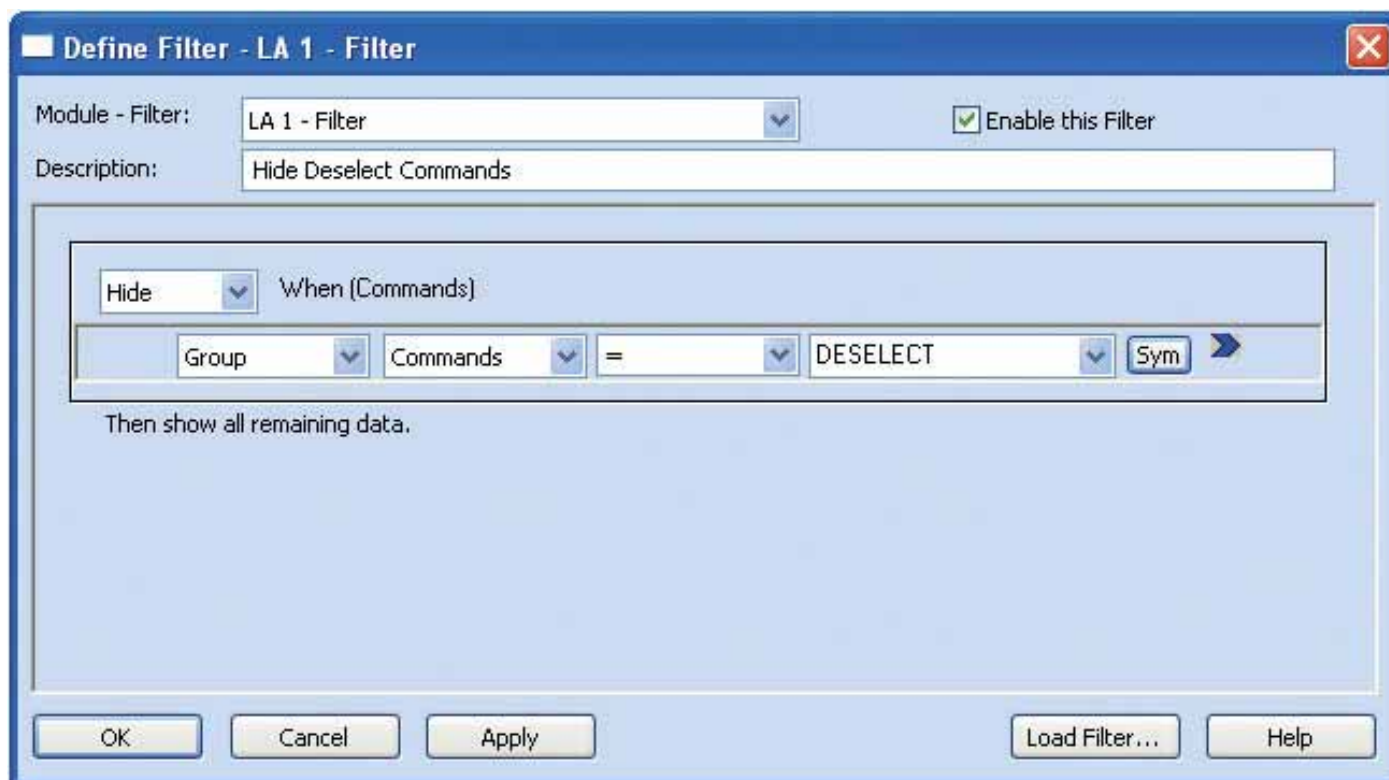


図19. SDRAMのすべてのDeselectコマンドを隠す表示フィルタ

解析をさらに効率化するために、表示フィルタを定義できます。たとえば、図18には数多くのDeselectコマンドが表示されていますが、ListingウィンドウでDeselectコマンドを表示しないようにしたほうが便利な場合があります。Deselectコマンドを表示しないようにする（隠す）には、Define Filterウィンドウ（図19参照）

を使用します。CommandsのグループがDeselectである場合には、コマンドを表示しません（hide）。表示フィルタは、取込んだデータの表示方法を変更するだけで、データそのものを変更するわけではありません。

Deselectコマンドを隠す表示フィルタを適用して、Listingウィンドウを表示してみます（図20参照）。Activateコマンドの前には約7.8 $\mu$ s間隔で多数のRefreshコマンドがあり、Activateコマンドの後には多数のWrite-Writeコマンドが続いています。タイムスタンプの値は、表示されている前のコマンドからの時間であることに注意してください。最初のWriteコマンドのタイムスタンプは15nsですが、これはWaveformウィンドウで測定した値です。このメモリの場合、Activate-Writeコマンドの最小時間tRCE（MIN）は15nsです。TLA7AA4型のタイムスタンプの分解能は125psです。TLA7BB4型では20psです。

### Refresh動作の検証

DRAMの基本をなす記憶素子は、個々のメモリ・セルのキャパシタです。メモリ・セルの電荷は、いずれ消滅し、メモリ・セルはそのデータを失います。したがって、メモリ・セルのキャパシタがデータを失う前に、メモリ・セルのデータを読み取り、同じデータをメモリ・セルに書き戻す必要があります。この動作はRefreshと呼ばれます。

| Sample | Filter Commands | Timestamp    |
|--------|-----------------|--------------|
| 580    | REFRESH         | 0 ps         |
| 2656   | REFRESH         | 7.802,125 us |
| 4732   | REFRESH         | 7.802,500 us |
| 6808   | REFRESH         | 7.802,500 us |
| 8884   | REFRESH         | 7.802,625 us |
| 10960  | REFRESH         | 7.802,375 us |
| 13036  | REFRESH         | 7.802,000 us |
| 15112  | REFRESH         | 7.802,250 us |
| 17188  | REFRESH         | 7.802,125 us |
| 19264  | REFRESH         | 7.802,125 us |
| 21340  | REFRESH         | 7.801,750 us |
| 23416  | REFRESH         | 7.801,750 us |
| 25492  | REFRESH         | 7.802,250 us |
| 27568  | REFRESH         | 7.802,000 us |
| 29644  | REFRESH         | 7.802,125 us |
| 31720  | REFRESH         | 7.802,250 us |
| 33796  | REFRESH         | 7.802,125 us |
| 35872  | REFRESH         | 7.802,375 us |
| 37948  | REFRESH         | 7.802,250 us |
| 40024  | REFRESH         | 7.802,750 us |
| 42100  | REFRESH         | 7.802,750 us |
| 44176  | REFRESH         | 7.802,000 us |
| 46252  | REFRESH         | 7.802,125 us |
| 48328  | REFRESH         | 7.802,250 us |
| 50404  | REFRESH         | 7.802,000 us |
| 52480  | REFRESH         | 7.801,625 us |
| 54556  | REFRESH         | 7.802,375 us |
| 56632  | REFRESH         | 7.801,875 us |
| 58708  | REFRESH         | 7.802,375 us |
| 60784  | REFRESH         | 7.802,625 us |
| 62860  | REFRESH         | 7.801,875 us |
| 64936  | REFRESH         | 7.802,500 us |
| 65535  | ACTIVATE        | 2.251,375 us |
| 65539  | WRITE           | 15.000 ns    |
| 65541  | WRITE           | 7.500 ns     |
| 65543  | WRITE           | 7.625 ns     |
| 65545  | WRITE           | 7.500 ns     |
| 65547  | WRITE           | 7.500 ns     |
| 65549  | WRITE           | 7.500 ns     |
| 65551  | WRITE           | 7.500 ns     |
| 65553  | WRITE           | 7.500 ns     |
| 65555  | WRITE           | 7.500 ns     |
| 65557  | WRITE           | 7.625 ns     |
| 65559  | WRITE           | 7.500 ns     |
| 65561  | WRITE           | 7.500 ns     |
| 65563  | WRITE           | 7.625 ns     |
| 65565  | WRITE           | 7.500 ns     |
| 65567  | WRITE           | 7.500 ns     |
| 65569  | WRITE           | 7.500 ns     |
| 65571  | WRITE           | 7.375 ns     |
| 65573  | WRITE           | 7.625 ns     |
| 65575  | WRITE           | 7.500 ns     |
| 65577  | WRITE           | 7.500 ns     |
| 65579  | WRITE           | 7.625 ns     |
| 65581  | WRITE           | 7.500 ns     |
| 65583  | WRITE           | 7.500 ns     |
| 65585  | WRITE           | 7.500 ns     |
| 65587  | WRITE           | 7.500 ns     |
| 65589  | WRITE           | 7.500 ns     |
| 65591  | WRITE           | 7.500 ns     |
| 65593  | WRITE           | 7.500 ns     |
| 65595  | WRITE           | 7.625 ns     |
| 65597  | WRITE           | 7.500 ns     |
| 65599  | WRITE           | 7.500 ns     |
| 65601  | WRITE           | 7.500 ns     |

図20. すべてのDeselectコマンドを非表示にしたListingウィンドウ。Activateコマンドでトリガがかかる前には、アクティブなメモリ・コマンドがRefreshコマンドだけであることがはっきりとわかります。Activateコマンドの後には、Write-Write動作のシーケンスが続きます。

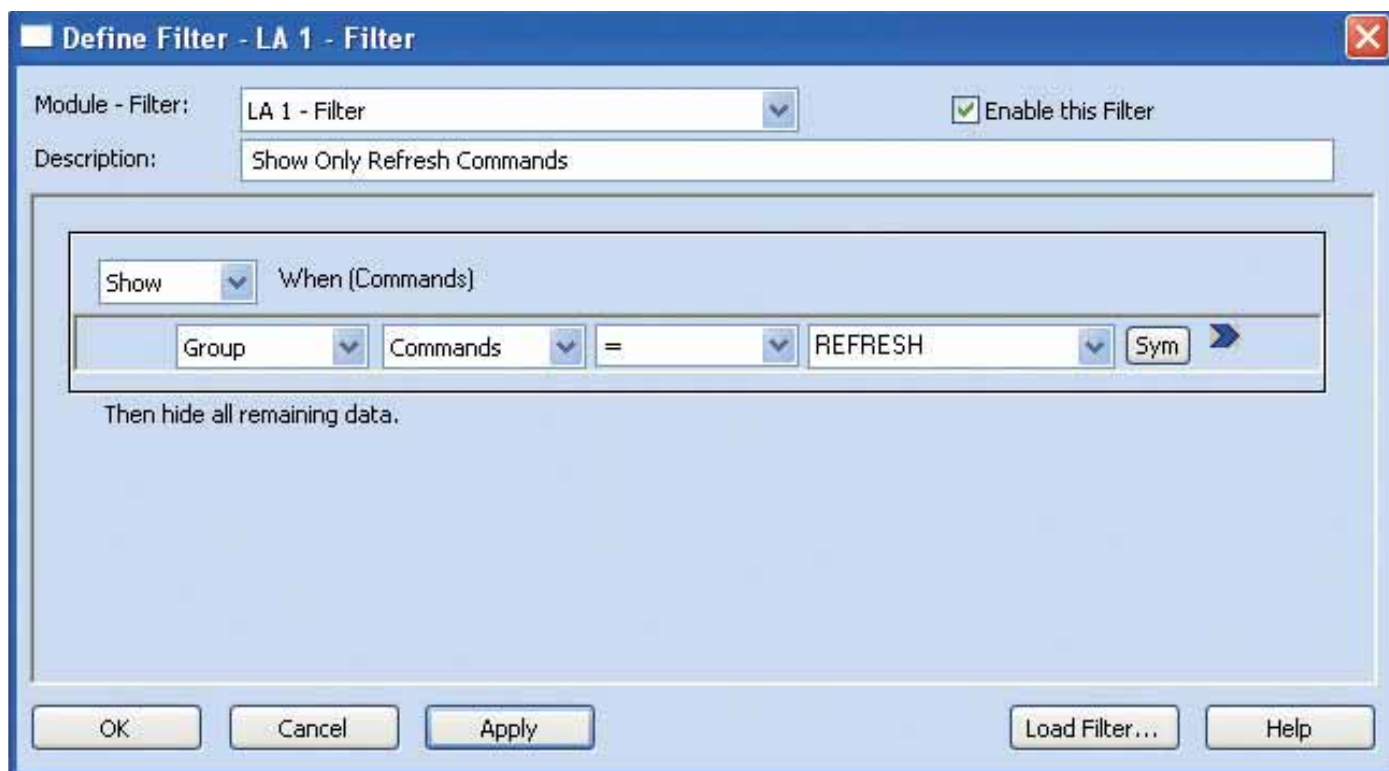


図21. この表示フィルタは、SDRAMのRefreshコマンドだけを表示するためのもので、Refresh動作の検証に役立ちます。

DRAMのすべてのメモリ・セルは、キャパシタの電荷が完全に減少する前にRefreshされる必要があります。メモリをRefreshするのは、メモリ・コントローラ・ハブの役割です。初期のDRAMでは、Refreshする行のアドレスはRefreshサイクル時にメモリ・コント

ローラ・ハブが提供する必要がありました。現在では、SDRAMには行のアドレスを生成する内部Refreshコントローラ（リフレッシュ・カウンタ）があり、SDRAMがRefreshコマンドを受け取るたびに、次にRefreshされる行アドレスが生成されます。

この例で使用している512Mb DDR2 SDRAMのRefreshのサイクル・タイムは64msで、Refreshのカウンタの幅は13ビット(8192)です。Refreshのサイクル・タイム(64ms)をRefreshカウント数(8192)で割ると、 $7.8125\mu\text{s}$ となります。これは、RefreshからRefreshまでの平均時間です。Refresh動作には若干の柔軟性があり、タスク間のスケジューリングやスイッチングを効率化するために、最大で8つの連続したRefreshサイクルが認められています。

以下では、Refreshサイクルが平均値の $7.8125\mu\text{s}$ であるかどうかを検証します。連続したRefreshサイクルの、サイクル間の最小時間 $t_{\text{RFC(MIN)}}$ の仕様は105nsです。この仕様は、Refresh-Activateコマンドの最小間隔にも適用されます。

Refreshコマンドだけを表示する表示フィルタを作成します(図21参照)。このフィルタをListingウィンドウに適用します(図22参照)。Listingウィンドウには、128Kステートのすべてのコマンドを含んだデータをフィルタに通した結果が表示されます。最初のRefreshコマンドは、サンプル580で取込まれ、最後のRefreshコマンドはサンプル129305で取込まれています。タイムスタンプは、個々のRefreshコマンド間の時間を表示しており、平均は約 $7.8\mu\text{s}$ です。

| Sample | Filter Commands | Timestamp    |
|--------|-----------------|--------------|
| 580    | REFRESH         | 0 ps         |
| 2656   | REFRESH         | 7.802,125 us |
| 4732   | REFRESH         | 7.802,500 us |
| 6808   | REFRESH         | 7.802,500 us |
| 8884   | REFRESH         | 7.802,625 us |
| 10960  | REFRESH         | 7.802,375 us |
| 13036  | REFRESH         | 7.802,000 us |
| 15112  | REFRESH         | 7.802,250 us |
| 17188  | REFRESH         | 7.802,125 us |
| 19264  | REFRESH         | 7.802,125 us |
| 21340  | REFRESH         | 7.801,750 us |
| 23416  | REFRESH         | 7.801,750 us |
| 25492  | REFRESH         | 7.802,250 us |
| 27568  | REFRESH         | 7.802,000 us |
| 29644  | REFRESH         | 7.802,125 us |
| 31720  | REFRESH         | 7.802,250 us |
| 33796  | REFRESH         | 7.802,125 us |
| 35872  | REFRESH         | 7.802,375 us |
| 37948  | REFRESH         | 7.802,250 us |
| 40024  | REFRESH         | 7.802,750 us |
| 42100  | REFRESH         | 7.802,750 us |
| 44176  | REFRESH         | 7.802,000 us |
| 46252  | REFRESH         | 7.802,125 us |
| 48328  | REFRESH         | 7.802,250 us |
| 50404  | REFRESH         | 7.802,000 us |
| 52480  | REFRESH         | 7.801,625 us |
| 54556  | REFRESH         | 7.802,375 us |
| 56632  | REFRESH         | 7.801,875 us |
| 58708  | REFRESH         | 7.802,375 us |
| 60784  | REFRESH         | 7.802,625 us |
| 62860  | REFRESH         | 7.801,875 us |
| 64936  | REFRESH         | 7.802,500 us |
| 67012  | REFRESH         | 7.851,500 us |
| 69100  | REFRESH         | 7.798,750 us |
| 71177  | REFRESH         | 7.806,125 us |
| 73252  | REFRESH         | 7.797,875 us |
| 75329  | REFRESH         | 7.806,000 us |
| 77404  | REFRESH         | 7.797,625 us |
| 79481  | REFRESH         | 7.805,875 us |
| 81556  | REFRESH         | 7.798,875 us |
| 83633  | REFRESH         | 7.806,500 us |
| 85708  | REFRESH         | 7.799,250 us |
| 87785  | REFRESH         | 7.807,000 us |
| 89860  | REFRESH         | 7.799,125 us |
| 91937  | REFRESH         | 7.805,750 us |
| 94012  | REFRESH         | 7.797,750 us |
| 96089  | REFRESH         | 7.805,250 us |
| 98164  | REFRESH         | 7.798,125 us |
| 100241 | REFRESH         | 7.806,125 us |
| 102316 | REFRESH         | 7.798,625 us |
| 104393 | REFRESH         | 7.806,125 us |
| 106468 | REFRESH         | 7.798,750 us |
| 108545 | REFRESH         | 7.806,000 us |
| 110620 | REFRESH         | 7.797,875 us |
| 112697 | REFRESH         | 7.806,250 us |
| 114772 | REFRESH         | 7.799,375 us |
| 116849 | REFRESH         | 7.806,375 us |
| 118924 | REFRESH         | 7.798,250 us |
| 121001 | REFRESH         | 7.805,250 us |
| 123076 | REFRESH         | 7.797,875 us |
| 125153 | REFRESH         | 7.805,500 us |
| 127228 | REFRESH         | 7.798,500 us |
| 129305 | REFRESH         | 7.806,250 us |

図22. ListingウィンドウにRefreshコマンドだけを表示し、Refresh-Refreshコマンドの時間を検証

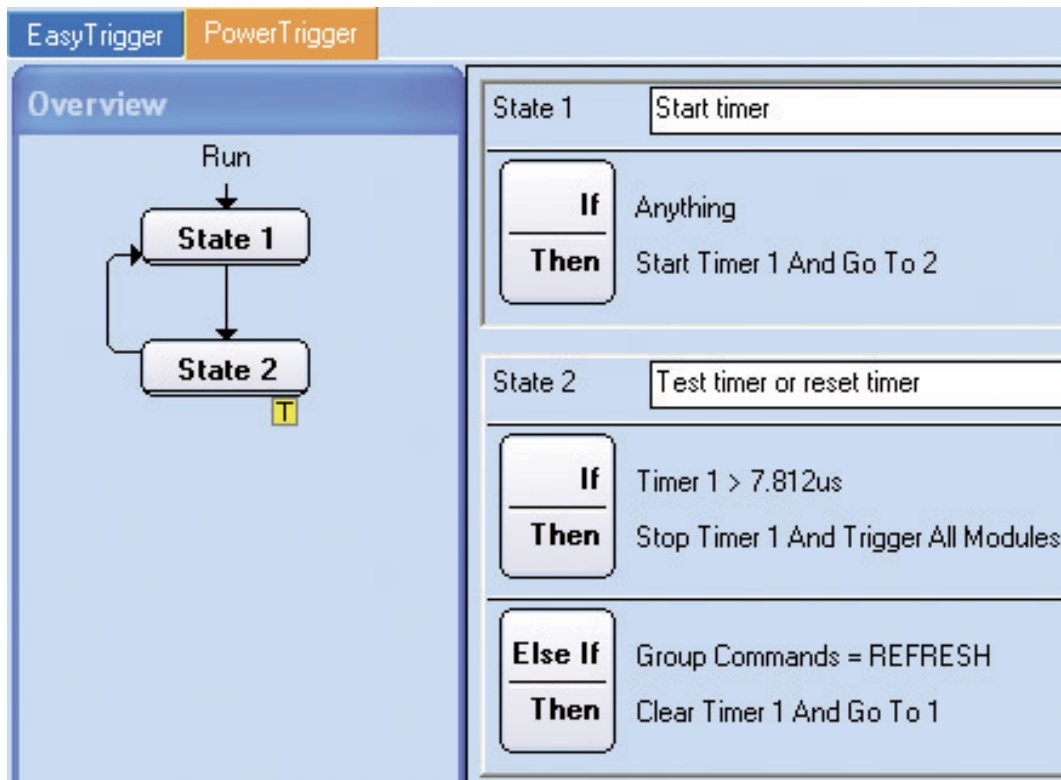
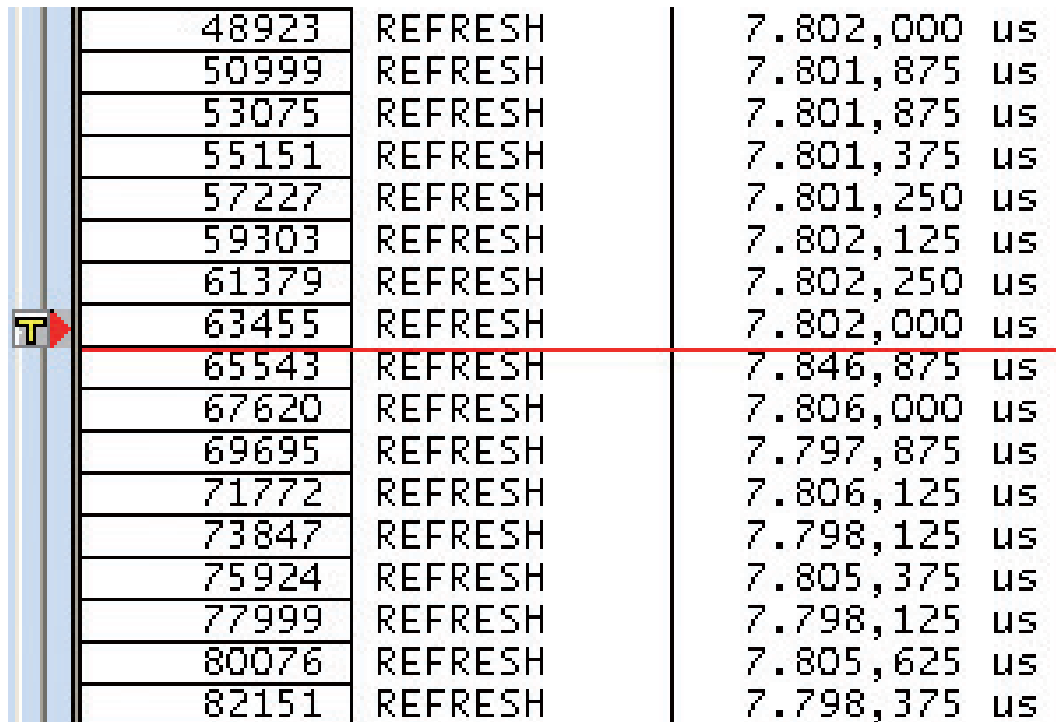


図23. Power Triggerプログラムで7.812 $\mu$ sより大きいRefresh-Refreshコマンドを検出

すべてのデータを取込み、目視でチェックする方法は、Refreshコマンド間の最大平均時間を検証する手段としてはあまり有効ではありません。もっと良い方法は、簡単なPower Triggerプログラムを作り、Refreshコマンド間の間隔が指定した時間長よりも大きい場合にのみトリガをかける方法です（図23参照）。最初のトリガ・ステートはタイマを起動させ、2つ目のステートはタイマの値が7.812 $\mu$ sより大きい、また、そのサンプル・ポイントにRefreshコマンドが存在するかをチェックします。タイマの値が7.812 $\mu$ sより大きい場合は、ロジック・アナライザにトリガがかかります（図24参照）。ここでは、Refresh-Refreshコマンドの時間長は

7.846,875 $\mu$ sとなっています。PowerTriggerプログラムの設定値を7.9 $\mu$ sに変更し、再びチェックを実行します。長時間放置しましたが、各Refresh-Refreshコマンドの時間長と7.9 $\mu$ sとを比較した結果、トリガはかかりませんでした。このことから、Refresh-Refreshコマンドの最大時間は7.9 $\mu$ s未満であることが確認できます。図24では、サンプル番号が連続していないことに注目してください。これは、表示フィルタが使用されていることをはっきりと示しています。図21のSDRAMのRefreshコマンドだけを表示する表示フィルタが使用されています。



|       |         |              |
|-------|---------|--------------|
| 48923 | REFRESH | 7.802,000 us |
| 50999 | REFRESH | 7.801,875 us |
| 53075 | REFRESH | 7.801,875 us |
| 55151 | REFRESH | 7.801,375 us |
| 57227 | REFRESH | 7.801,250 us |
| 59303 | REFRESH | 7.802,125 us |
| 61379 | REFRESH | 7.802,250 us |
| 63455 | REFRESH | 7.802,000 us |
| 65543 | REFRESH | 7.846,875 us |
| 67620 | REFRESH | 7.806,000 us |
| 69695 | REFRESH | 7.797,875 us |
| 71772 | REFRESH | 7.806,125 us |
| 73847 | REFRESH | 7.798,125 us |
| 75924 | REFRESH | 7.805,375 us |
| 77999 | REFRESH | 7.798,125 us |
| 80076 | REFRESH | 7.805,625 us |
| 82151 | REFRESH | 7.798,375 us |

図24. Power Triggerプログラムで7.812  $\mu$ sよりも大きい時間長のRefresh-Refreshコマンドを検出。この場合のRefresh間隔は7.846,875  $\mu$ s。

Power Triggerプログラムでは、カウンタ、タイマ、セットアップ/ホールド時間違反、グリッチ、16のステート、レンジ・リコグナイザ、ワード・リコグナイザ、トランジション・リコグナイザといったさまざまなトリガ・リソースを使用できます。Power Triggerプログラムのライブラリは、正しいメモリ動作を検証するための貴重なツールとなります。Power Triggerプログラムでは、たとえばActivateコマンドとWriteコマンド間の時間長を測定し、最小時間の違反に対してトリガをかけるように設定することもできます。

#### アドレスとバンクの動作

DDR2 SDRAMの動作を検証するのにこれまで使用した信号は、わずかに5つです。次に検証する信号は、Address信号とBank信号です。これらの信号はメモリ・コントローラ・ハブから送出され、SDRAMのコマンドと同様、SDRAMのクロックの立上りエッジでメモリに読み込まれます。

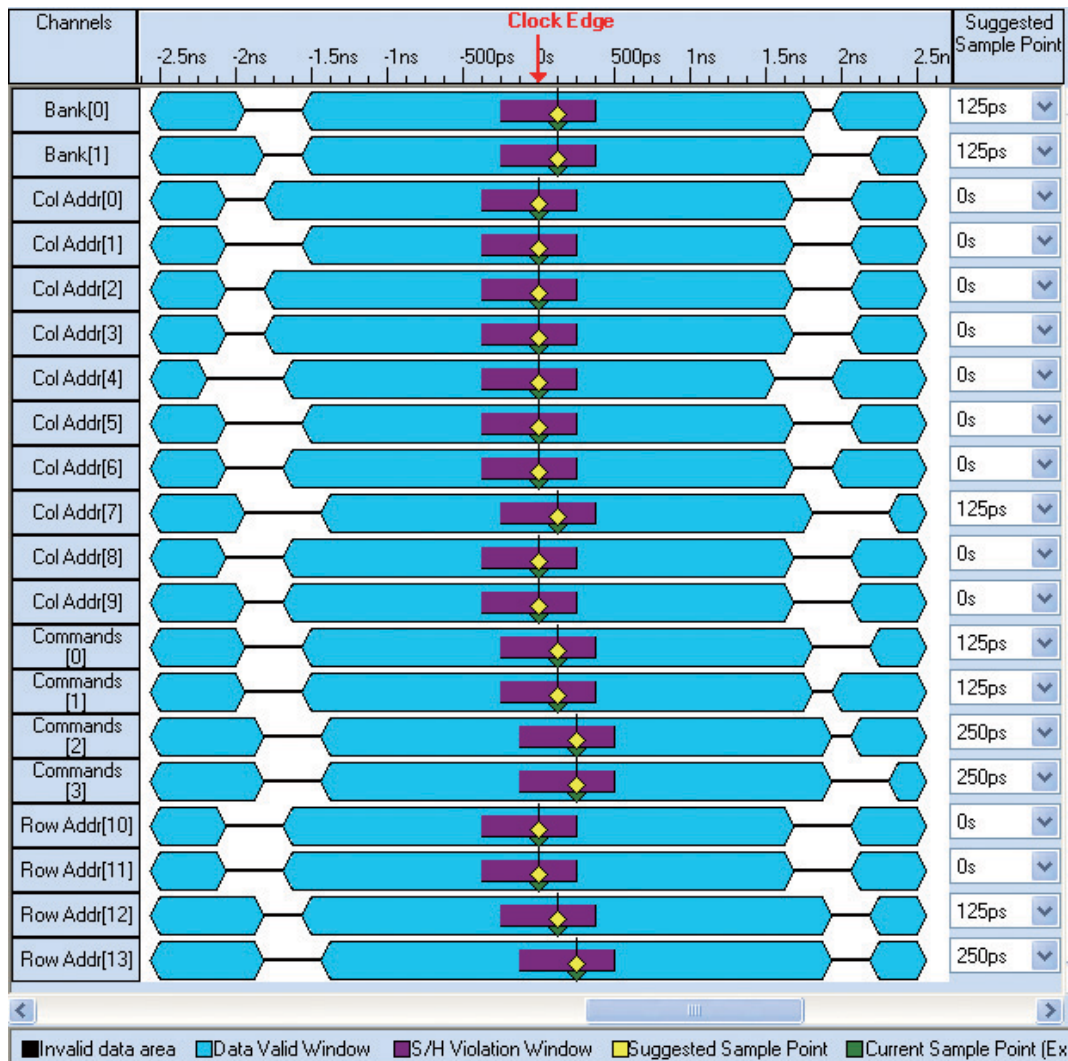


図25. バンク・アドレス、列アドレス、行アドレス、およびコマンドのデータ有効ウィンドウを表示するAuto Deskew解析

DDR2 SDRAMのサイズやバンク構成はさまざまです。512MbのDDR2 SDRAMは、A0～A13を行アドレス、BA0～BA1をバンク・アドレス、A0～A9を列アドレスとして使用します。列アドレスよりも行アドレスの方が多いため、メモリの消費電力を減らすためです。メモリの消費電力は、列の数の増加にともなって増加します。行アドレスと列アドレスは、メモリの同じ物理ピン上でマルチプレックス化されています。そのため、Activateコマンドは

A0～A13のすべてを行アドレスに使用します。Writeコマンドは、A0～A09を列アドレスに使用し、A10は自動プリ・チャージのオン/オフの切り替えに使用されます。自動プリ・チャージは、書き込みバーストの後で行を自動的にプリ・チャージし行を閉じます。自動プリ・チャージが選択されていない場合は、その後の書き込みおよび読み込みアクセスのために行は開いたままになります。

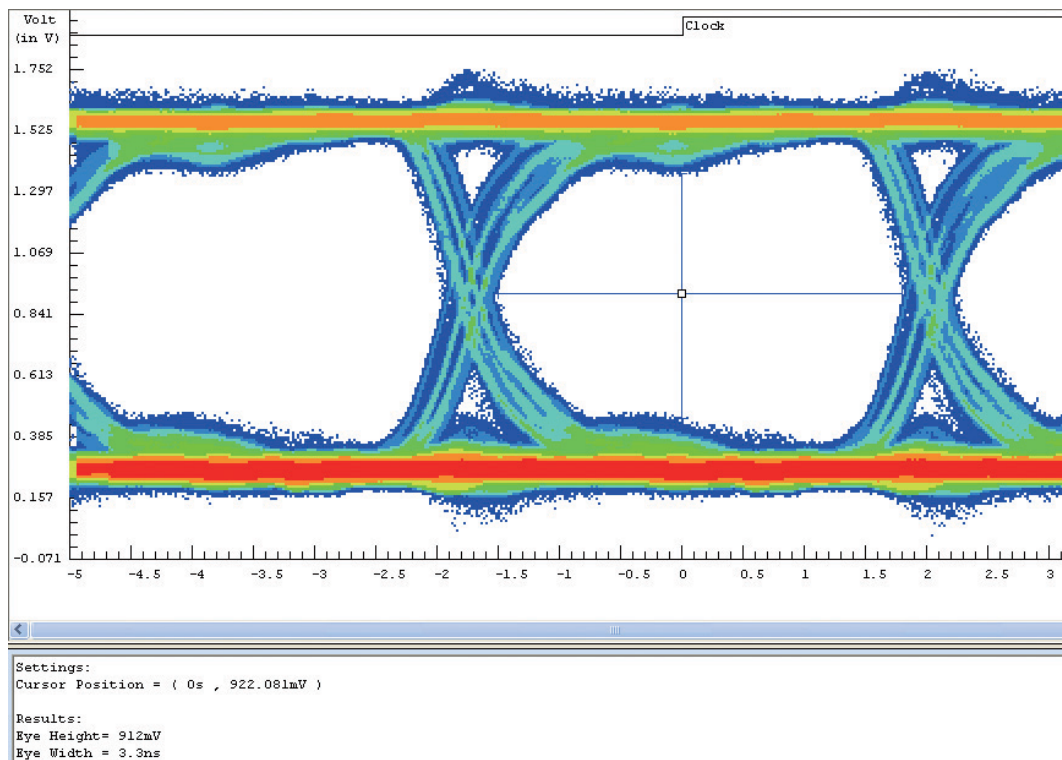


図26. バンク・アドレス、列アドレス、行アドレス、およびコマンドのデータ有効ウィンドウを表示するiVerity解析

メモリの使用に関する統計データは、書き込みより多くの読み込みが行われることや、次のメモリ・ロケーションがほかのどのメモリ・ロケーションよりも読み取られる可能性が高いことを示しています。これは、コンピュータは他のどんなメモリ動作よりもシーケンシャルな命令の読み取りが多いというコンセプトを裏付けています。ActivateコマンドとReadコマンドの間にはレイテンシが存在しますが、最初の読み込みコマンドを実行した後は、連続してReadコマンドを与えることにより、開いているメモリの行の終わりに達するまで最大データ・レートで連続したデータ読み出しが可能です。そのため、行アドレスは、メモリ・コントローラ・ハブの最下位アドレス・ビットに割り当てられます。

DDR2 SDRAMのアドレス・バスとバンク・アドレスのステート・アクイジションの際には、ロジック・アナライザのデータ有効ウィンドウ、あるいはサンプル・ポイントのポジションは、コマンド・バスの場合と同じ要領で設定します（図25および図26参照）。予想されるように、2つのバンク・アドレスと14のアドレスを追加すると、アイ・ダイアグラムはわずかに狭くなります。多チャンネルのアイ・ダイアグラムは、コマンド信号が4つしかない図14の3.48nsから、20の信号すべてを含む図26の3.3nsへと変化します。

| Sample | Filter Bank | Filter Row Addr | Filter Col Addr | Filter Commands | Timestamp    |
|--------|-------------|-----------------|-----------------|-----------------|--------------|
| 59337  | 3           | 2000            | 000             | REFRESH         | 7.801,875 us |
| 61413  | 3           | 2000            | 000             | REFRESH         | 7.801,625 us |
| 63489  | 3           | 2000            | 000             | REFRESH         | 7.802,250 us |
| 65535  | 3           | 2000            | 000             | ACTIVATE        | 7.689,875 us |
| 65539  | 3           | 2000            | 000             | WRITE           | 15.000 ns    |
| 65541  | 3           | 2004            | 004             | WRITE           | 7.500 ns     |
| 65543  | 3           | 2008            | 008             | WRITE           | 7.500 ns     |
| 65545  | 3           | 200C            | 00C             | WRITE           | 7.500 ns     |
| 65547  | 3           | 2010            | 010             | WRITE           | 7.625 ns     |
| 65549  | 3           | 2014            | 014             | WRITE           | 7.500 ns     |
| 65551  | 3           | 2018            | 018             | WRITE           | 7.500 ns     |
| 65553  | 3           | 201C            | 01C             | WRITE           | 7.500 ns     |
| 65555  | 3           | 2020            | 020             | WRITE           | 7.500 ns     |
| 65557  | 3           | 2024            | 024             | WRITE           | 7.500 ns     |
| 65559  | 3           | 2028            | 028             | WRITE           | 7.500 ns     |
| 65561  | 3           | 202C            | 02C             | WRITE           | 7.500 ns     |
| 65563  | 3           | 2030            | 030             | WRITE           | 7.625 ns     |
| 65573  | 3           | 2034            | 034             | PRECHARGE       | 37.500 ns    |
| 65577  | 3           | 2034            | 034             | REFRESH         | 15.125 ns    |
| 65608  | 3           | 2000            | 000             | ACTIVATE        | 116.500 ns   |
| 65612  | 3           | 2034            | 034             | WRITE           | 15.000 ns    |
| 65614  | 3           | 2038            | 038             | WRITE           | 7.625 ns     |
| 65616  | 3           | 203C            | 03C             | WRITE           | 7.500 ns     |
| 65618  | 3           | 2040            | 040             | WRITE           | 7.500 ns     |

図27. ListingウィンドウでのActivateコマンドに対するトリガとバンク・アドレス、行アドレス、列アドレスの表示例

Deselectコマンドをフィルタにより表示対象から除外し、Activateコマンドでロジック・アナライザにトリガをかけます（図27参照）。各データ・カラムのタイトル欄に"Filter"という単語が表示されていますが、これは、データにフィルタが適用されていることを示しています。最初のActivateコマンドと2つめのActivateコマンドはともに、2000（hex）という行アドレスと、3というバンク・アドレスにアクセスしています。最初のWriteコマンドはヘキサの000という列アドレスにアクセスし、メモリ・コントローラ・ハブはメモリに4つのデータ値をバースト状に書き込みます。2つめのWrite

コマンドでは列アドレスがヘキサの004になっていますが、これは000、001、002、003のアドレスには最初のWriteコマンドによってすでにデータが書きこまれているためです。Prechargeコマンドの直前のWriteコマンドの列アドレスは、ヘキサの030です。メモリ・コントローラ・ハブのPrechargeコマンドは、開いていたバンク3、行アドレスヘキサの2000の行を閉じます。Refreshサイクルが実行されると、Activateコマンドがバンク3でヘキサ2000の同じ行を開き、次のWriteコマンドが次の列アドレス034への書き込みを開始します。

タイムスタンプを使用すると、このコマンド・シーケンスのタイミングをすばやく検証できます。最初の3つのRefreshコマンドの間隔は、指定された平均値の $7.8\mu\text{s}$ です。Refresh-Activateコマンドの時間は $7.689,875\mu\text{s}$ で、仕様の最小値である $105\text{ns}$ を大幅に上回っています。Activate-Writeの時間は、仕様の最小値と同じ $15\text{ns}$ です。

連続したWrite-Writeの時間は仕様の最小値である $7.5\text{ns}$ で、これはクロック2サイクルに相当します。バースト長が4であれば、毎回の書き込み動作で4つのすべてのデータ・ビットをメモリに書き込むのに2クロック・サイクルを要します。このアプリケーション・ノートですでに述べたように、各Writeコマンド間にはDeselectコマンドが挿入されています。タイムスタンプは若干ばらつきますが、SDRAMのクロック周期が $3.759\text{ns}$ であるのに対して分解能が $125\text{ps}$ であるためです。

Write-Prechargeは、3つの時間からなっています。第一は書き込みレイテンシ (WL) で、長さは3クロック・サイクル ( $3.759\text{ns} \times 3 = 11.277\text{ns}$ ) です。第二は、バーストの4つのデータ・ビットをメモリに書き込むための2クロック・サイクルです ( $3.759\text{ns} \times 2 = 7.518\text{ns}$ )。3つ目は、書き込みのリカバリ・タイムの $15\text{ns}$  (MIN) です。これら3つの時間を足し合わせた値は、図27の実際の測定値である $37.5\text{ns}$ よりも小さい $33.795\text{ns}$  (MIN) になります。

Precharge-Refreshの時間長は $15.125\text{ns}$ で、これはPrechargeのコマンド周期の $15\text{ns}$  (MIN) に極めて近い値です。2つめのRefresh-Activateの時間長は $116\text{ns}$ で、要求仕様のRefresh-Activateコマンドの $105\text{ns}$  (MIN) よりも大きくなっています。Listingウィンドウの残りのWrite-Writeコマンドは、 $7.625\text{ns} \sim 7.500\text{ns}$ です。

最後のRefresh-Refreshの時間長は、いくつかの方法で検証できます。第一は、タイムスタンプの時間を足し合わせる方法です。第二は、もっと簡単で、エラーを生じにくいカーソルを使用する方法です。カーソル2を最後のRefreshに移動し、カーソル1を1つ前のRefresh、すなわちActivateコマンドのすぐ上のRefreshに移動します。次に、デルタ時間のリードアウトで、カーソル1からカーソル2までの時間を測定します。結果は、指定された平均リフレッシュ間隔である $7.847\mu\text{s}$ になります。

Listingウィンドウは、慎重に読み取る必要があります。列アドレスA0～A9は、行アドレスのA0～A9のアドレスと同じ物理ピンです。一方、行アドレスはA0～A13を使用します。したがってアドレス・バスは、Activateコマンドでは14ビットの行アドレスとして読み取り、Writeコマンドでは10ビットの列アドレスとして読み取ります。

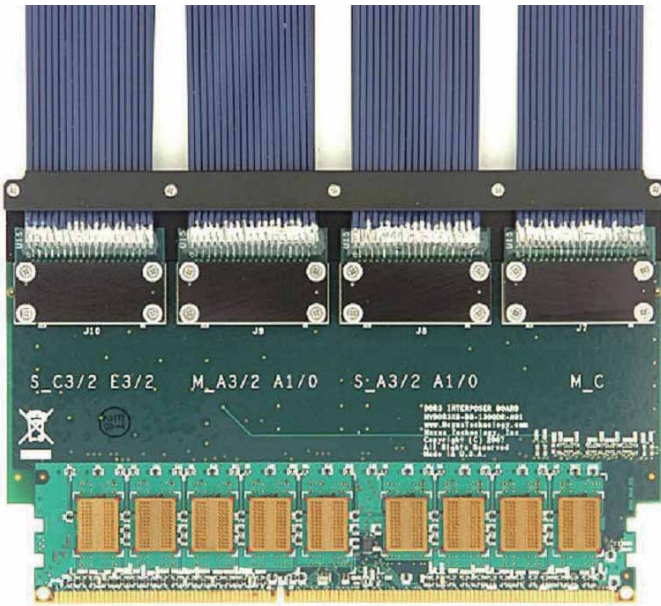


図28. JEDEC規格とロジック・アナライザのプローブ・ポイント用に設計された Nexus DDR3 NEXVu DIMM。優れたシグナル・インテグリティを提供すると共に、メモリ・コンポーネントにおける実際のアイ・パターンを取込むことができます。

## ロジック・アナライザ・メモリ・サポート

ロジック・アナライザ・メモリ・サポートは、ロジック・アナライザ設定のコンフィギュレーション、メモリ・アクイジションのためのカスタム・クロッキング設定、メモリ・データ解析ソフトウェア、およびモニタリング・リストを提供し、ロジック・アナライザによるメモリ・システム解析の操作性や効率を高めます。メモリ・サポートにはプロービング・ハードウェアが含まれる場合があります。Nexus Technology社 ([www.NexusTechnology.com](http://www.NexusTechnology.com)) は、ロジック・アナライザ用に幅広いメモリ・サポート製品を提供しており、当社のロジック・アナライザおよびオシロスコープ用の補完的な製品を提供するテクノロジクス・エンベデッド・システムズ・ツール・パートナーでもあります。当社は、Nexus Technology社のいくつかの製品を取り扱っています。

Nexus Technology社のDDR3/DDR2プロトコル違反ソフトウェアは、DDR2とDDR3のバスを自動的に解析し、プロトコル違反、その周波数違反をすばやく簡単に特定することができ、ロジック・アナライザ・メモリに取込まれたすべてのDDRコマンドを一覧することもできます。

Nexus Technology社のロジック・アナライザ・メモリ・サポート製品は、DDR、DDR2、DDR3、DIMM、SO-DIMM、FB-DIMM、および新しく登場したその他のメモリ技術に使用されます。プロービング・ハードウェアは、エンベデッド・メモリ・システムに直接プロービングする場合を除き、DIMM用インタポザからメモリを実装した試験専用のDIMMまで多岐に渡ります。Nexus Technology社では、メモリを実装した試験用のDIMMをNEXVu DIMMと呼んでいます（図28参照）。

DDR2 NEXVu DIMMは、通常のDIMMよりも高さがあるため、通常の高さのDIMMを隣接スロットに装着した状態でもロジック・アナライザ・プローブを接続できます。ロジック・アナライザ・プローブは、DDR2 NEXVu DIMMの表か裏のどちらかの面に接続します。2つのDDR2 NEXVu DIMMを隣接するスロットに装着する場合は、それぞれのDDR2 NEXVu DIMMの反対側の面にロジック・アナライザ・プローブを接続できます。

NEXVu DIMMは、接続するプロービング回路ボードのトレースの負荷効果を軽減するために、FBGA ICパッケージの近接の回路ボードの内層にアイソレーション抵抗を備えています。図28では、8つのDDR2 667 SDRAMとヒート・シンクを備えたアドバンスド・メモリ・バッファ（AMB）を見ることができます。

また、図28では、エッジ・コネクタからAMBに向かう24の高速差動信号のトレースも見ることができます。トレースは、FBDIMMの中心から遠ざかるほどトレース長が長くなることに注意してください。FB-DIMMメモリ・システムには、24の異なる信号のトレース長を等しくするための曲がりくねったトレース・パターンは存在しません。これは、回路ボードのレイアウトを単純化します。信号の数が少なく、回路ボードのレイアウトが単純になればなるほど、メモリ・システム当たりのFB-DIMMチャンネル数はUDIMMやRDIMMを使用した場合に比べて多くなります。回路ボードのトレース長の違いに起因する信号のスキューは、FBDIMMチャンネルの初期化の過程でAMBにより軽減されます。

| Sample | Address | FBDNXV2A Mnemonics                                                                                                                                                                                                         |
|--------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 30     | 00A4A   | MRS - MODE REGISTER SET<br>Normal MRS<br>PD Mode: Standard<br>Write Recovery: 6<br>DLL Reset: No<br>Operating Mode: Normal<br>Latency: 4<br>Burst Type: Interleaved<br>Burst: 4                                            |
| 31     | 10380   | MRS - MODE REGISTER SET<br>Extended MRS<br>RDQS Enable: No<br>DQS# Enable: Enable<br>OCD Operation: OCD Calibration><br>Rtt: Disabled<br>Additive Latency: 0<br>Output Drive Strength: 100%<br>DLL Enable: Enable (Normal) |

図29. DDR2 SDRAMのモード・レジスタ（MR）と拡張モード・レジスタの設定は、メモリ・コントローラ・ハブによるMode Registerコマンドの送出時にロジック・アナライザに取込まれます。

NEX-FBD-NEXVuでのデータの取込み

データの取込みを開始するための設定も簡単です。最初のステップは、ロジック・アナライザのデータ有効ウィンドウを最適に設定（Commandグループのサンプル・ポイント設定）することです。これは、このアプリケーション・ノートですでに説明したように、MagniVuまたはAuto Deskewを使用します。複数チャンネルのアイ・ダイアグラムで信号のデータ有効ウィンドウをチェックするには、iVerify®機能を使用します。

NEX-FBD-NEXVuのカスタム・クロッキングには、ロジック・アナライザのメモリを有効に使用するためのいくつかのクロッキング・モードがあります。最初のステップでは2つの選択を行います。1つは、DDR2 DRAMのクロックのすべての立上りエッジの取込みか、またはRAS#およびCAS#コマンドに合わせて取込むかの選択です。もう1つの選択は、Refreshサイクルを取込むか、または取込むRefreshサイクルの数を減らすかの選択です。

次のステップでは、Listingウィンドウにカラムを追加し、CASレイテンシ（CL）、バースト長（BL）、および追加レイテンシ（AL）を決定するためにSDRAMのモード・レジスタ（MR）と拡張モード・レジスタの設定を取込みます。これらのレジスタには、メモリ・コントローラ・ハブにより、Mode Registerコマンドの実行時にアドレスA0～13およびバンク・アドレスBA0～1を使用して、16ビットの値がプログラムされます。

モード・レジスタの値は、読み込みおよび書き込みのデータ有効ウィンドウの場所を探すのに必要になります。ここではモード・レジスタの値は、CASレイテンシが4、バースト長が4、追加レイテンシが0となります（図29参照）。これらの値は、書き込みおよび読み込みのデータ有効ウィンドウの場所を決定し、Listingウィンドウでロジック・アナライザの逆アセンブリ表示の設定をするのに使用されます。

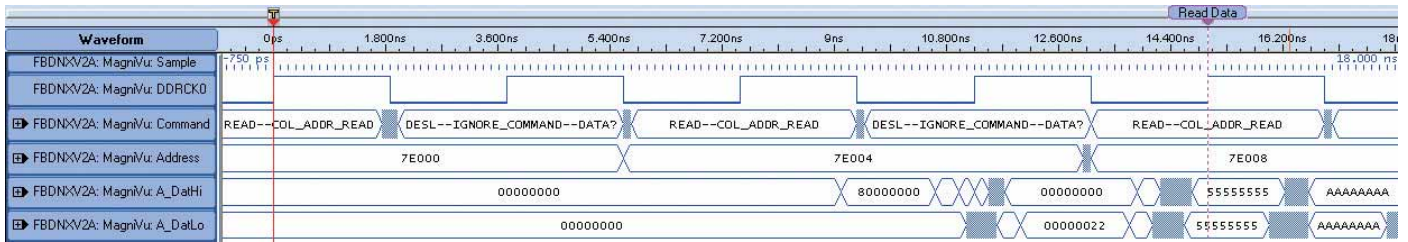


図31. Activateコマンド後の最初のReadコマンドに対するトリガ。読み込みレイテンシ（RL）分の4つのクロック・エッジをカウントして、Read Dataマーカーで示された最初の読み込みデータ・サイクルを検出。DDR2 SDRAMの64ビットの読み込みデータは、5555、5555、5555、5555(hex)とAAAA、AAAA、AAAA、AAAA(hex) が交互に入れ替わるパターンです。

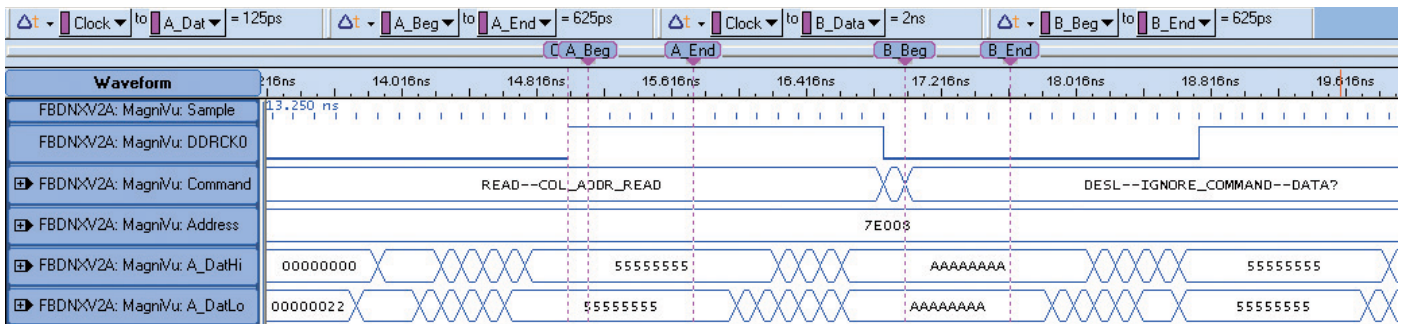


図32. MagniVu高分解能タイミングは、読み込みのデータ有効ウィンドウを測定してロジック・アナライザのサンプル・ポイント・ポジションを設定。DDR2 SDRAMの64ビットの読み込みデータは、5555、5555、5555、5555(hex)とAAAA、AAAA、AAAA、AAAA(hex) が交互に入れ替わるパターンです。

メモリ・サポートは、16ビットのモード・レジスタと拡張モード・レジスタのビット・デコード表示が可能です。図29は、サンプル30のモード・レジスタ設定がデコードされ、どのようなニーモニックになるかを示しています。図9で使ったシンボル表は、図29に示すような複雑な複数行のニーモニック・デコーディングを提供できません。ニーモニック・デコーディングは、メモリ・サポートの大きな長所です。

第三のステップでは、Activate-Readのコマンド・シーケンスに対してトリガをかけ、MagniVu®高分解能タイミング波形を使用して、クロック・サイクル当たり2つの読み込みデータ有効ウィンドウを決定します。通常は、最初の読み込みデータをActivate-Readのコマンド・シーケンスの頭で確認する方が任意のReadコマンドに対してトリガをかけるよりも簡単です。また、読み込みレイテンシが正しいことを確認するのも容易です。

読み込みレイテンシ（RL）は、読み込みのバースト・データの起点を決定します。CASレイテンシ（CL）に追加レイテンシ（AL）を加えて読み込みレイテンシ（RL）を決定し、ReadコマンドからのRL分のクロックの立上りエッジをカウントしてReadコマンドの最初の読み込みデータ・サイクルを検出します（図30参照）。2クロックのデータ読み込みサイクルと4クロックのデータ読み込みサイクルがあり、前者はバースト長が4の場合で、後者はバースト長が8のケースです。

メモリ・サポートは、64ビットのメモリ・データを32ビット・データのhiグループと、32ビット・データのlowグループに分割します（図31参照）。この場合、マージされた2つのTLA7A44型ロジック・アナライザ・モジュールはデータ・ラインに対して外部2Xクロッキング・モードで動作します。これは、ロジック・アナライザがサンプル・ポジションの異なる2つのメモリ・チャンネルを使用して、1つのデータI/Oライン上の1クロック・サイクル分のDDR2SDRAMデータを取込むことを意味します。読み込みデータを取込むために、それぞれ2つのサンプル・ポイントを持つ2つのグループからなる4つのサンプル・ポイントがSetupウィンドウのカスタム・クロッキングで設定されます。

Waveformウィンドウのマーカーが、クロックの立上りエッジにセットされ、Clockという名前に変更されています。図31では、マーカーは目に見える5本の縦の点線で示されています。2つめのA\_Begマーカーは、値が5555、5555、5555、5555である最初の読み込みデータ有効ウィンドウの起点より後ろにセットされます。3つめのA\_Endマーカーは、A\_Begマーカーから625ps後ろにセットされます。このA\_BegからA\_Endまでの625psの時間は、TLA7AA4型ロジック・アナライザ・モジュールで確実にデータを取込むために最低限必要な、読み込みデータが安定して、変化してはならない時間です。

| Sample | Address | FBDNXV2A Mnemonics          | FBDNXV2A DataHi | FBDNXV2A DataLo | FBDNXV2A BChkBits | Timestamp |
|--------|---------|-----------------------------|-----------------|-----------------|-------------------|-----------|
| 65530  | 4E3FC   | DESL - IGNORE COMMAND       | -----           | -----           | 00                | 3.750 ns  |
| 65531  | 4E3FC   | DESL - IGNORE COMMAND       | -----           | -----           | 00                | 3.875 ns  |
| 65532  | 4E3FC   | DESL - IGNORE COMMAND       | -----           | -----           | 00                | 3.625 ns  |
| 65533  | 4E000   | PRE - PRECHARGE SELECT BANK | -----           | -----           | 00                | 3.875 ns  |
| 65534  | 7E000   | ACTV - ROW ADDRESS STROBE   | -----           | -----           | 00                | 15.000 ns |
| 65535  | 7E000   | READ - COL ADDR READ        | -----           | -----           | 00                | 15.000 ns |
| 65536  | 7E000   | DESL - IGNORE COMMAND       | -----           | -----           | 00                | 3.750 ns  |
| 65537  | 7E004   | READ - COL ADDR READ        | -----           | -----           | 00                | 3.750 ns  |
| 65538  | 7E004   | DESL - IGNORE COMMAND       | -----           | -----           | 00                | 3.750 ns  |
| 65539  | 7E008   | READ - COL ADDR READ        | -----           | -----           | 00                | 3.750 ns  |
|        |         | READ DATA                   | 55555555        | 55555555        | 00                |           |
|        |         | READ DATA                   | AAAAAAAA        | AAAAAAAA        | 00                |           |
| 65540  | 7E008   | READ DATA                   | 55555555        | 55555555        | 00                | 3.750 ns  |
|        |         | READ DATA                   | AAAAAAAA        | AAAAAAAA        | 00                |           |
| 65541  | 7E00C   | READ - COL ADDR READ        | -----           | -----           | 00                | 3.750 ns  |
|        |         | READ DATA                   | 55555555        | 55555555        | 00                |           |
|        |         | READ DATA                   | AAAAAAAA        | AAAAAAAA        | 00                |           |
| 65542  | 7E00C   | READ DATA                   | 55555555        | 55555555        | 00                | 3.750 ns  |
|        |         | READ DATA                   | AAAAAAAA        | AAAAAAAA        | 00                |           |
| 65543  | 7E010   | READ - COL ADDR READ        | -----           | -----           | 00                | 3.750 ns  |

図33. Nexus Technology社のNEX-FBD-NEXVuのListingウィンドウに5555、5555、5555、5555とAAAA、AAAA、AAAA、AAAAが交互に入れ替わるパターンのDDR2 SDRAM読み込みデータが表示された例

ClockとA\_Begマーカ間のデルタ時間測定で示される125psは、カスタム・クロッキングのセットアップで設定されたセットアップ時間に相当します。同様に、B\_BegマーカとB\_Endマーカは、DDR2 SDRAMクロック・サイクルの2つ目のデータ有効ウィンドウを決定するのに使用されます。図31では、1つ目のロジック・アナライザの読み込みデータ有効ウィンドウAは、クロックの立上りエッジの125ps後から始まっています。2つ目のロジック・アナライザの読み込みデータ有効ウィンドウBは、クロックの立上りエッジから2ns後に始まっています。

TLA7AA4型ロジック・アナライザ・モジュールを3枚マージして使用している場合は、書き込みデータと読み込みデータの両方を取込みます。ロジック・アナライザの書き込みサンプル・ポイントのポジションを設定するには、Activate-Writeのコマンド・シーケンスに対してトリガをかけ、MagniVuの高分解能タイミング波形を使用して書き込みデータに対するロジック・アナライザのデータ有効ウィンドウを決定します。書き込みデータの場所を決定するには、書き込みレイテンシ (WL) は読み込みレイテンシ (RL) よりも1小さいことに注意してください。

DDR2 SDRAMのデータ・ラインには、1つのクロック・サイクルに対して4つのデータ有効ウィンドウがあります。ロジック・アナライザは、それぞれのクロック・サイクルの4つのすべてのデータ有効ウィンドウを取込みるように設定します。それには、マージされた3枚のTLA7AA4型ロジック・アナライザ・モジュールに対してカスタム・クロッキングをExternal 4Xモードに設定します。ロジック・アナライザは、データの取込みを完了した後、SDRAMのコマンドを解析してそれが書き込みサイクルであるか読み込みサイクルであるかを判断し、さらに、ロジック・アナライザの取込んだデータのどこに有効なデータ・サイクルが存在するかを追加レイテンシ (AL)、CASレイテンシ (CL)、およびバースト長 (BL) を使用して決定します。サイクルのタイプ、追加レイテンシ、CASレイテンシ、およびバースト長は、ListingウィンドウのMnemonicsカラムのDisassembly propertiesでユーザが設定します。

メモリ・サポートのListingウィンドウは、65539と65541のサンプルに関してニーモニックを3行で表示しています (図32参照)。DDR2 SDRAMの1つのクロック・サイクルで、メモリ・コントローラ・ハブはDDR2 SDRAMに対してReadコマンドを送出し、同じクロック・サイクルで前のReadコマンドから2つの64ビットのデータ読み込みが発生しています。NEX-FBD-NEXVuメモリ・サポートは、この解析機能を提供し、ニーモニックをデコードします。また、NEX-FBD-NEXVuは、ListingウィンドウのMnemonicsカラムのDisassembly propertiesで、「取込んだすべてのサイクルを表示」、「idleサイクルまたはwaitサイクルは非表示」、「Activate、Write、およびReadのサイクルを表示」、「WriteとReadのサイクルだけを表示」の4つの表示モードを選ぶことができます。

## まとめ

このアプリケーション・ノートでは、SDRAMのプロトコルのシーケンスおよびタイミングの検証に際してのロジック・アナライザの能力について説明してきました。SDRAMのクロックや4つのSDRAMコマンド信号をチェックするだけで、多くのことを検証できます。完全な検証には、Nexus Technology社の幅広いメモリ・サポート製品、メモリ・インターポーザ、NEXVu DIMMを使用します。

当社は、エンベデッド・システムやコンピュータの設計技術者がメモリ設計の電気試験や動作評価を迅速かつ正確に実施できるようにするために、業界最先端のオシロスコープ、真の差動TDR、およびロジック・アナライザとNexus Technology社のメモリ・サポート・パッケージのシステムを含めた包括的なツール・セットを提供しています。これらのツール・セットを使用することで、優れた操作性と性能が実現され、組み込みシステムやコンピュータ・メモリ・システムの評価、デバッグの理想的なソリューションとなります。

## Tektronix お問い合わせ先：

### 日本

本社 03-6714-3111  
SA営業統括部 03-6714-3004  
ビデオ計測営業部 03-6714-3005

大宮営業所 048-646-0711  
仙台オフィス 022-792-2011  
神奈川営業所 045-473-9871  
東京営業所 042-573-2111  
名古屋営業所 052-581-3547  
大阪営業所 06-6397-6531  
京都オフィス 075-323-9048  
福岡営業所 092-472-2626  
湘南カスタム・サービス・センタ 0120-7-41046

### 地域拠点

米国 1-800-426-2200  
中南米 52-55-542-4700  
東南アジア諸国／豪州 65-6356-3900  
中国 86-10-6235-1230  
インド 91-80-2227-5577  
欧州 44-0-1344-392-400  
中近東／北アフリカ 41-52-675-3777  
他30カ国

Updated 17 October 2007

### 詳細について

当社は、最先端テクノロジーに携わるエンジニアのために、資料を用意しています。当社ホームページ([www.tektronix.co.jp](http://www.tektronix.co.jp))または[www.tektronix.com](http://www.tektronix.com))をご参照ください。



TEKTRONIXおよびTEKは、Tektronix, Inc.の登録商標です。Microsoftは、米国Microsoft Corporationの登録商標です。記載された商品名はすべて各社の商標あるいは登録商標です。

04/08 FLG/WOW

52Z-19236-1



## 日本テクトロニクス株式会社

東京都港区港南2-15-2 品川インターシティB棟6階 〒108-6106  
製品についてのご質問・ご相談は、お客様コールセンターまでお問い合わせください。

**TEL 03-6714-3010 E-mail [ccc.jp@tektronix.com](mailto:ccc.jp@tektronix.com)**

電話受付時間／9:00～12:00・13:00～18:00 月曜～金曜（休祝日は除く）

当社ホームページをご覧ください。 [www.tektronix.co.jp](http://www.tektronix.co.jp)  
製品のFAQもご覧ください。 [www.tektronix.co.jp/faq/](http://www.tektronix.co.jp/faq/)

●記載内容は予告なく変更することがありますので、あらかじめご了承ください。

© Tektronix