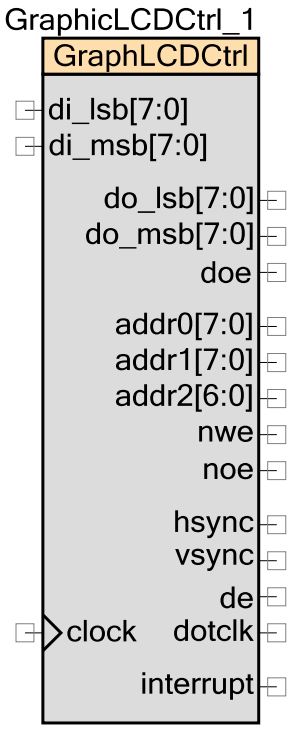
グラフィック LCD コントローラ (GraphicLCDCtrl) 1.61



特長

- 以下にしめす、HVGA 解像度までのフルプログラマブルスクリーンサイズをサポート:
  - QVGA (320x240) @ 60 Hz 16 bpp
  - WQVGA (480x272) @ 60 Hz 16 bpp
  - HVGA (480x320) @ 60 Hz 16 bpp
- 仮想スクリーンの操作に対応
- SEGGER emWin グラフィックライブラリ搭載インターフェース
- ブランキングの合間に読み書き操作を実行
- CPU の介入なしに、連続したタイミング信号をパネルに送信します
- 外部フレームバッファとして使用される最大 23 ビットアドレス、16 ビットデータの非同期 SRM デバイスに対応
- 水平および垂直ブランキングの合間の入り口と出口で選択可能な割込みパルスを生成

概要説明

グラフィック LCD コントローラ (GraphicLCDCtrl) コンポーネント は、LCD ドライバがあるが、LCD コントローラがない LCD パネルへのインターフェースです。このタイプのパネルには、フレームバッファがありません。フレームバッファを外部で提供する必要があります。

このコンポーネントは、16ビット幅の非同期 SRM デバイスを用いた、外部フレームバッファへのインタフェースでもあります。

このコンポーネントは、SEGGER emWin グラフィック ライブラリと一緒に動作するように設計されています。このグラフィックライブラリは Cypress によって提供され、Cypress のWebサイトwww.cypress.com/go/comp_emWinでご利用いただけます。このグラフィックライブラリには、テキストと画像の描画およびレンダリングのフル機能セットが用意されています。

GraphicLCDCtrl を使用する場合

GraphicLCDCtrl コンポーネントは、複数の LCD パネルに対応します。外部 SRAM 内で、制御信号を直接駆動し、フレームバッファを管理します。コンポーネントは、dotclk、hsync、vsync、および de 出力を生成し、SRAM からデータにアクセスし、制御を通して LCD 上に表示します。

フレームバッファの SRAM は、LCD パネルをリフレッシュしていない場合にのみ、読み書きのアクセスができます。リフレッシュ期間中に読み込みもしくは書き込みの要求があった場合、提供された API 関数は、リフレッシュがブランキング期間になるまで待機します。ブリンキング期間の間に、読み込みもしくは書き込みが行われます。

ブランキング期間の開始および終了を示すために、割り込みを使用することができます。この機能は、RTOS と組み合わせると特に有用です。RTOS は、ブランキング期間の開始および終了時に、フレームバッファにアクセスが必要なタスクをスワップインもしくはスワップアウトすることができます。

入出力接続

ここでは、GraphicLCDCtrl コンポーネントの入出力接続について説明します。一部の I/O は、シンボルの下に隠れている場合があります。そのシンボルで隠れる場合は、I/O の記述に記載されています。

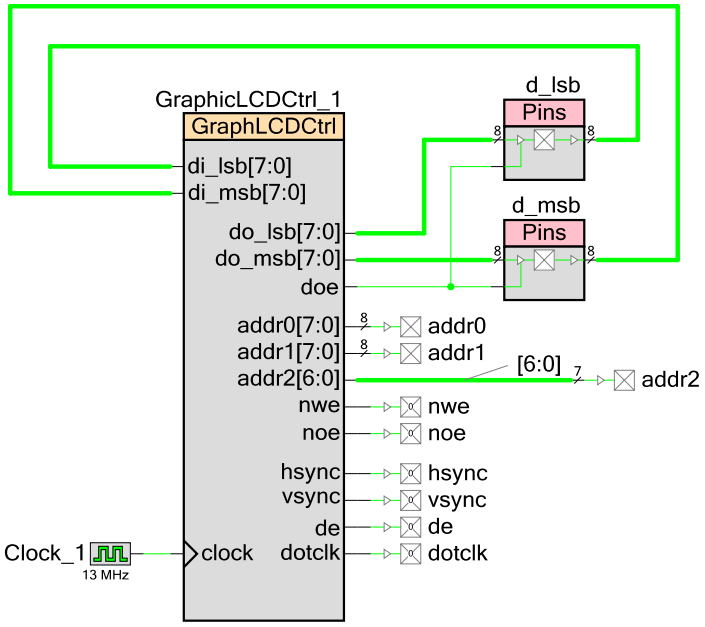
Input	非表示の場合	説明
di_lsb[7:0]	N	入力データバスの下位8ビット。読み込み処理中に使用されます。これらの信号をデバイスの入力ピンに接続し、これらのピンの「Input Synchronized (同期入力)」を無効にします。信号自身は同期出力信号に基づいて駆動されるので、既に同期されています。
di_msb[7:0]	N	入力データバスの上位8ビット。読み込み処理中に使用されます。これらの信号をデバイスの入力ピンに接続し、これらのピンの「Input Synchronized (同期入力)」を無効にします。信号自身は同期出力信号に基づいて駆動されるので、既に同期されています。
clock	N	このコンポーネントを駆動するクロック。周波数はdotclkの2倍です。

出力	非表示の場合	説明
do_lsb[7:0]	N	出力データバスの下位8ビット。書き込みトランザクション中のデータで使用されます。
do_msb[7:0]	N	出力データバスの上位8ビット。書き込みトランザクション中のデータで使用されます。
doe	N	PSoC内のデータバスコンポーネントの出力イネーブル。通常は、データバスの入力/出力ピン接続の出力イネーブルに接続されます。
addr0[7:0]	N	フレームバッファに接続されている、アドレスバスの下位8ビット。
addr1[7:0]	N	フレームバッファに接続されている、アドレスバスの中位8ビット。
addr2[6:0]	N	フレームバッファに接続されている、アドレスバスの上位7ビット。フレームバッファが必要とするデータビットの数は、使用するSRAMデバイスにより異なります。

出力	非表示の場合	説明
nwe	N	フレームバッファSRAMのアクティブLowの書き込みイネーブル。
noe	N	フレームバッファのアクティブLowの出力イネーブル。
de	N	パネルのデータイネーブル。
hsync	N	パネルの水平同期タイミング信号。
vsync	N	パネルの垂直同期タイミング信号。
dotclk	N	パネルを駆動するクロック。このクロックは、入力クロックの半分の周波数です。
割り込み	Y	エッジトリガの割り込み信号。割り込み生成が選択されていない場合、この出力は隠されています。

回路図マクロ情報

このマクロは、Optrex T-55343GD035JU-LW-AEN パネルへのインターフェースに対するデフォルトに合わせて設定されています。マクロに含まれているクロックは 13 MHz に設定されており、Optrex QVGA パネルに提供される dotclk は 6.5 MHz になります。



一般的には上位アドレスビットの一部だけが (addr2)、フレームバッファへの接続に使用されます。数は、使用される SRAM によって異なります。必要なアドレスビットの数に基づいて、以下の手順を使用してバスの幅を調整することができます。

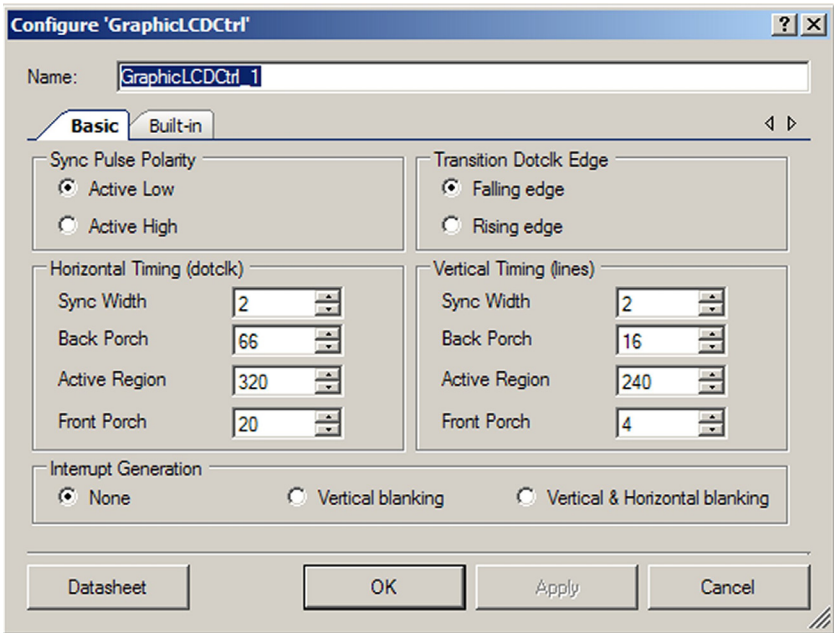
- addr2 出力ピンコンポーネントを設定し、「ピン数」を設定します。

- 出力ピンを駆動する信号で右クリックし、「Edit Name And Width (名前と幅を編集する)」を選択します。次に、「Left Index(左インデックス)」 を調整して出力ピンの幅を反映します。

「Input Synchronized (同期入力)」オプションは、すべてのデータ ピンで選択が解除され、すべてのピンの API 生成はオフになります。

コンポーネント・パラメータ

デザインの上に GraphicLCDCtrl コンポーネント をドラッグし、ダブルクリックして **Configure** ダイアログを開きます。デフォルトの GraphicLCDCtrl 設定は、Optrex パネルで動作の設定をします。



Configure GraphicLCDCtrl ダイアログには、以下のパラメータがあります。これらの設定全ては、コンパイル時に反映され、実行時にそれら設定を変更する必要はありません。これらはパネルの特性であり、フレームバッファ SRAM が使用されています。

同期パルスの極性

この設定により、hsync および vsync 信号は **Active High(アクティブハイ)** (生成されるパルスはハイパルス) または **Active Low(アクティブロー)** のいずれかです。この選択は hsync および vsync の両方の極性を制御します。デフォルト設定は **Active Low(アクティブロー)** です。

dotclkの遷移エッジ

dotclk はパネルに送信されるクロックで、これに基づいてパネルが動作します。遷移が**立ち上がりエッジ**に設定された場合、関連するすべての信号(hsync、vsync、de など)は dotclk の立ち上がりエッジで遷移します。**立ち下がりエッジ**に設定された場合、信号は立ち下がりエッジで遷移します。

一般的に、パネルが dotclk の片方のエッジにセットアップとホールドタイムを指定している場合、この設定を逆側のエッジに指定します。これにより、セットアップおよびホールドのサイクルが約 2 分の 1 クロックになります。デフォルトの設定は、**立ち下がりエッジ**です。

水平タイミング (dotclk)

- **Sync Width(同期間隔)** – 水平同期間隔を dotclk 単位で定義します。この値は、1 から 256 クロックサイクルの間で設定できます。デフォルトの設定は **2** です。
- **Back Porch(バックポーチ)** – 水平バックポーチ間隔を dotclk 単位で定義します。. この値は、6 から 256 クロックサイクルの間で設定できます。最低値の 6 は、スクリーンエリアがアクティブになる前に、読み込みもしくは書き込みアクセスが完了できないことを防ぐため、ステートマシンに十分な余裕を持って予告を発するために必要です。いくつかのパネルの設定では、バックポーチの期間を、同期シグナルの終了から、アクティブ期間の開始までの間と定義します。他のパネルでは、バックポーチの期間を、同期パルスの開始から、アクティブ期間の開始までの間と定義します。このコンポーネントでは、バックポーチの期間を、同期パルスの終了から、アクティブ期間の開始までの間と定義します。デフォルトの設定は **66** です。
- **Active Region(アクティブ領域)** – 水平アクティブ期間を dotclk 単位で定義します。アクティブ領域は、4 の倍数の設定を使用して実装されます。これにより、8 ビットカウンタのみを使用している場合でも、1024 x 1024 の大きさの領域を扱うことができます。頻繁に用いられるスクリーンサイズの全ては、垂直および水平方向の両方とも 4 の倍数です。この値には、4 から 1024 クロックサイクルの間の、4 の倍数が設定可能です。デフォルトの設定は **320** です。
- **Front Porch(フロントポーチ)** – 水平フロントポーチ幅を dotclk 単位で定義します。この値は、1 から 256 クロックサイクルの間で設定できます。デフォルトの設定は **20** です。

垂直タイミング(ライン)

- **Sync Width(Sync 幅)** – 垂直 Sync 幅をライン単位で定義します。この値は、1 から 256 クロックサイクルの間で設定できます。デフォルトの設定は **2** です。
- **Back Porch(バックポーチ)** – 垂直バックポーチ幅をライン単位で定義します。この値は、1 から 256 ラインの間で設定できます。いくつかのパネルの設定では、バックポーチの期間を、同期シグナルの終了から、アクティブ期間の開始までの間と定義します。他のパネルでは、バックポーチの期間を、同期パルスの開始から、アクティブ期間の開始までの間と定義します。このコンポーネントでは、バックポーチの期間を、同期パルスの終了から、アクティブ期間の開始までの間と定義します。デフォルトの設定は **16** です。
- **Active Region(アクティブ領域)** – 垂直アクティブ期間をライン単位で定義します。アクティブ領域は、4 の倍数の設定を使用して実装されます。これにより、8 ビットカウンタのみを使用している場合でも、1024 x 1024 の大きさの領域を扱うことができます。頻繁に用いられるスクリーンサイズの全ては、垂直および水平方向の両方とも 4 の倍数です。この値には、4 から 1024 ラインの間の、4 の倍数が設定可能です。デフォルトの設定は **240** です。

- **Front Porch(フロントポーチ)** – 垂直フロントポーチ間隔をライン単位で定義します。この値は、1 から 256 ラインの間で設定できます。デフォルトの設定は **4** です。

割り込みの生成

割り込みの生成の設定を定義します。デフォルトの設定は **None(なし)** です。

- **Vertical blanking(垂直ブランキング)**が設定されている場合、垂直ブランキング期間の開始時と終了時に割り込みパルスが生成されます。
- **Vertical & Horizontal blanking(垂直および水平ブランキング)**が設定されている場合、全てのアクティブ領域の開始時と終了時に割り込みパルスが生成されます。

  垂直アクティブ領域の間は、各ラインの開始時と終了時に割り込みパルスが生成されます。垂直ブランキング領域の間は、最後のアクティブラインの終わり、および最初のアクティブラインの始まりに、割り込みがそれぞれ一つずつ生成されます。

Clock Select（クロック選択）

このコンポーネントには、内部クロックはありません。クロックソースを必ず取りつけてください。提供するクロックの周波数は、パネルに出力する dotclk クロックが必要とする周波数の 2 倍である必要があります。

配置

GraphicLCDCtrl は UDB アレイにわたり配置され、配置情報のすべては *cyfitter.h* ファイルを通して API に提供されます。

リソース

リソースのタイプ				API メモリ（バイト）		
データパス セル	PLD	ステータス セル	Control/ Count7 セル	フラッシュ	RAM	ピン（外部入出力ごと）
7	11	1	1	421	6	45

アプリケーション プログラミング インタフェース

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

デフォルトにより、PSoC Creator はインスタンス名「GraphicLCDCtrl_1」を、与えられたデザインにおける、最初に与えられたコンポーネントに割り当てます。インスタンス名は、識別子の構文ルールに従った固有の値に変更できます。インスタンス名は、すべてのグローバル関数名、変数名、定数名のプリフィックスになります。便宜上、以下の表で「GraphicLCDCtrl」をインスタンス名を使用します。

関数	説明
GraphicLCDCtrl_Start()	GraphicLCDCtrl インタフェースを開始します。
GraphicLCDCtrl_Stop()	GraphicLCDCtrl インタフェースを無効化します。
GraphicLCDCtrl_Write()	フレームバッファへの書き込み操作を開始します。
GraphicLCDCtrl_Read()	フレームバッファからの読み込み操作を開始します。
GraphicLCDCtrl_WriteFrameAddr()	スクリーンをリフレッシュするときに使う、フレームバッファアドレスの始点を設定します。
GraphicLCDCtrl_ReadFrameAddr()	スクリーンをリフレッシュするときに使う、フレームバッファアドレスの始点を読み込みます。
GraphicLCDCtrl_WriteLineIncr()	隣接するライン間のアドレス間隔を設定します。
GraphicLCDCtrl_ReadLineIncr()	ライン間のアドレスのインクリメントを読み込みます。
GraphicLCDCtrl_Sleep()	GraphicLCDCtr の設定を保存して、無効にします。
GraphicLCDCtrl_Wakeup()	GraphicLCDCtrlの設定を復元し、有効にします。
GraphicLCDCtrl_Init()	コンポーネント パラメータを初期化するか、コンポーネント カスタマイザで提供される設定に復元します。
GraphicLCDCtrl_Enable()	GraphicLCDCtrl を有効にします。
GraphicLCDCtrl_SaveConfig()	GraphicLCDCtr の設定を保存します。
GraphicLCDCtrl_RestoreConfig()	GraphicLCDCtrlの設定を復元します。

グローバル変数

変数	説明
GraphicLCDCtrl_initVar	GraphicLCDCtrl_initVar は、グラフィック LCD コントローラが初期化されているかどうかを表します。初期値は0で、GraphicLCDCtrl_Start() が最初に呼び出された時に1になります。これにより、GraphicLCDCtrl_Start() ルーチーンを最初に呼び出した後、再度初期化する必要なしにコンポーネントを再起動できます。 コンポーネントの再初期化が必要な場合、GraphicLCDCtrl_Start() もしくは GraphicsLCDCtrl_Enable()関数を呼び出す前に、GraphicLCDCtrl_Init()関数を呼び出すこともできます。

void GraphicLCDCtrl_Start(void)

説明: アクティブモードのパワーのテンプレートビット、またはクロックのゲーティングを必要に応じて有効にします。これが推奨されているコンポーネント操作開始の方法です。GraphicLCDCtrl_Start()は、initVar 変数を設定し、GraphicLCDCtrl_Init()関数を呼び出し、さらに GraphicLCDCtrl_Enable() 関数を呼び出します。

パラメータ: なし

返り値: なし

副作用: initVar関数が既に設定されている場合、この関数は GraphicLCDCtrl_Enable()関数のみを呼び出します。

void GraphicLCDCtrl_Stop(void)

説明: アクティブモードのパワーのテンプレートビット、およびクロックのゲーティングを必要に応じて無効にします。

パラメータ: なし

返り値: なし

副作用: なし

void GraphicLCDCtrl_Write(uint32 addr, uint16 data)

説明: 提供されたアドレスとデータに基づいて、フレームバッファへの書き込みトランザクションを開始します。この書き込みはポスティッドライトであるため、この関数は、書き込みがインターフェースに対し実際に終了する前に戻ります。コマンドキューが埋まっている場合、この関数は、書き込み要求をキューに入れるだけのスペースが空くまで戻りません。

パラメータ: addr: コンポーネントのアドレスラインに送られるアドレス (addr2[6:0], addr1[7:0], addr0[7:0])

data: do_msb[7:0] (上位バイト) および do_lsb[7:0] (下位バイト) ピンに送られるデータ

返り値: なし

副作用: なし

uint16 GraphicLCDCtrl_Read(uint32 addr)

説明: フレームバッファからの読み込みトランザクションを開始します。現在ポスティッドされている書き込みが全て完了した後に、読み込みが実行されます。この関数は、読み込みが完了し、その後に読み込み値を返すまで待機します。

パラメータ: addr: コンポーネントのアドレスラインに送られるアドレス (addr2[6:0], addr1[7:0], addr0[7:0])

返り値: do_msb[7:0] (上位バイト) および do_lsb[7:0] (下位バイト) ピンから読み込まれたデータ

副作用: なし

void GraphicLCDCtrl_WriteFrameAddr(uint32 addr)

説明: スクリーンをリフレッシュするときに使う、開始フレームバッファアドレスを設定します。このレジスタは、垂直ブランキング期間ごとに読み込まれます。このレジスタの自動アップデートを行うためには、アクティブリフレッシュ期間に実行してください。

パラメータ: addr: アドレス フレームバッファの始点のアドレス

返り値: なし

副作用: なし

uint32 GraphicLCDCtrl_ReadFrameAddr(void)

説明: スクリーンをリフレッシュするときに使う、開始フレームバッファアドレスを読み取ります。

パラメータ: なし

返り値: フレームバッファの始点のアドレス

副作用: なし

void GraphicLCDCtrl_WriteLineIncr(uint32 incr)

説明: 隣接するライン間のアドレス間隔を設定します。デフォルト値は、ラインのディスプレイ幅です。この設定により、ラインを異なるワード境界に調整する、もしくはディスプレイ領域より大きい仮想ライン長を用いることが可能です。

パラメータ: incr: ライン間のアドレスインクリメント。最低でもラインのディスプレイサイズの値が必要です。

返り値: なし

副作用: なし

uint32 GraphicLCDCtrl_ReadLineIncr(void)

説明: ライン間のアドレスのインクリメントを読み込みます。

パラメータ: なし

返り値: ライン間のアドレスのインクリメント

副作用: なし

void GraphicLCDCtrl_Sleep(void)

説明: これは、コンポーネントのスリープを準備するのに推奨されるルーチンです。GraphicLCDCtrl_Sleep() ルーチーンは、現在のコンポーネントの状態を保存します。その後、GraphicLCDCtrl_Stop() および GraphicLCDCtrl_SaveConfig() 関数を呼び出し、ハードウェアの設定を保存します。

GraphicLCDCtrl_Sleep() 関数の呼び出しは、fCyPmSleep()関数もしくは CyPmHibernate() 関数を呼び出す前にしてください。電源管理関数については、*『PSoC Creator システム・リファレンスガイド』*を参照してください。

パラメータ: なし

返り値: なし

副作用: なし

void GraphicLCDCtrl_Wakeup(void)

説明: GraphicLCDCtrl_Sleep()関数が呼ばれた状態にコンポーネントを復元させる場合の、最も推奨されるルーチーンです。GraphicLCDCtrl_Wakeup() 関数は、GraphicLCDCtrl_RestoreConfig() 関数を呼び出し、設定を復元します。GraphicLCDCtrl_Sleep() が呼び出される前にコンポーネントが有効にされた場合、GraphicLCDCtrl_Wakeup() 関数はコンポーネントを再度有効にします。

パラメータ: なし

返り値: なし

副作用: GraphicLCDCtrl_Sleep() もしくは GraphicLCDCtrl_SaveConfig() 関数を呼び出す前に GraphicLCDCtrl_Wakeup()関数を呼び出した場合、予期せぬ挙動を示す場合があります。

void GraphicLCDCtrl_Init(void)

説明: コンポーネント パラメータを初期化するか、コンポーネント カスタマイザの保持する初期値に復元します。タイミング設定を定義するコンパイル時の設定が、カスタマイザの保持する値に保存されます。フレームバッファアドレスのランタイム設定が0になり、ラインインクリメントがディスプレイのラインサイズに設定されます。

パラメータ: なし

返り値: なし

副作用: このコンポーネントは、この関数呼び出し前にGraphicLCDCtrl_Stop で無効にする必要があります。そうしない場合、予期しない動作が発生する場合があります。これは、コンポーネントを以下のように初期化します。FIFO のデータをクリアしない。残りのコンポーネントハードウェア状態マシンをリセットしない。

void GraphicLCDCtrl_Enable(void)

説明: この関数はハードウェアの使用を開始し、コンポーネントの動作を開始します。GraphicLCDCtrl_Enable()を呼び出す必要はありません。コンポーネントの操作を開始するための推奨された方法である、GraphicLCDCtrl_Start()ルーチーンはこの関数を呼び出します。

パラメータ: なし

返り値: なし

副作用: なし

void GraphicLCDCtrl_SaveConfig(void)

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

パラメータ: なし

返り値: なし

副作用: なし

void GraphicLCDCtrl_RestoreConfig(void)

説明: この関数は、コンポーネントの設定と非保持レジスタを復元します。この関数はまた、コンポーネントのパラメータ値を GraphicLCDCtrl_Sleep() 関数を呼び出す前の状態に復旧します。

パラメータ: なし

返り値: なし

副作用: この API が GraphicLCDCtrl_SaveConfig の前に呼び出された場合、コンポーネントの設定はデフォルト設定に復元されます。フレームバッファアドレスのランタイム設定が0になり、ラインインクリメントがディスプレイのラインサイズに設定されます。

ファームウェア・ソースコードのサンプル

PSoC Creator は、[Find Example Project（サンプルプロジェクトを検索）] ダイアログに多数のサンプルプロジェクトを提供しており、そこには回路図およびサンプル コードが含まれています。コンポーネント固有の例を見るには、[Component Catalog] または回路図に置いたコンポーネント インスタンスからダイアログを開きます。一般例については、[Start Page] または **[File（ファイル）]** メニューからダイアログを開きます。必要に応じてダイアログにある **Filter Options** を使用し、選択できるプロジェクトのリストを絞り込みます。

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

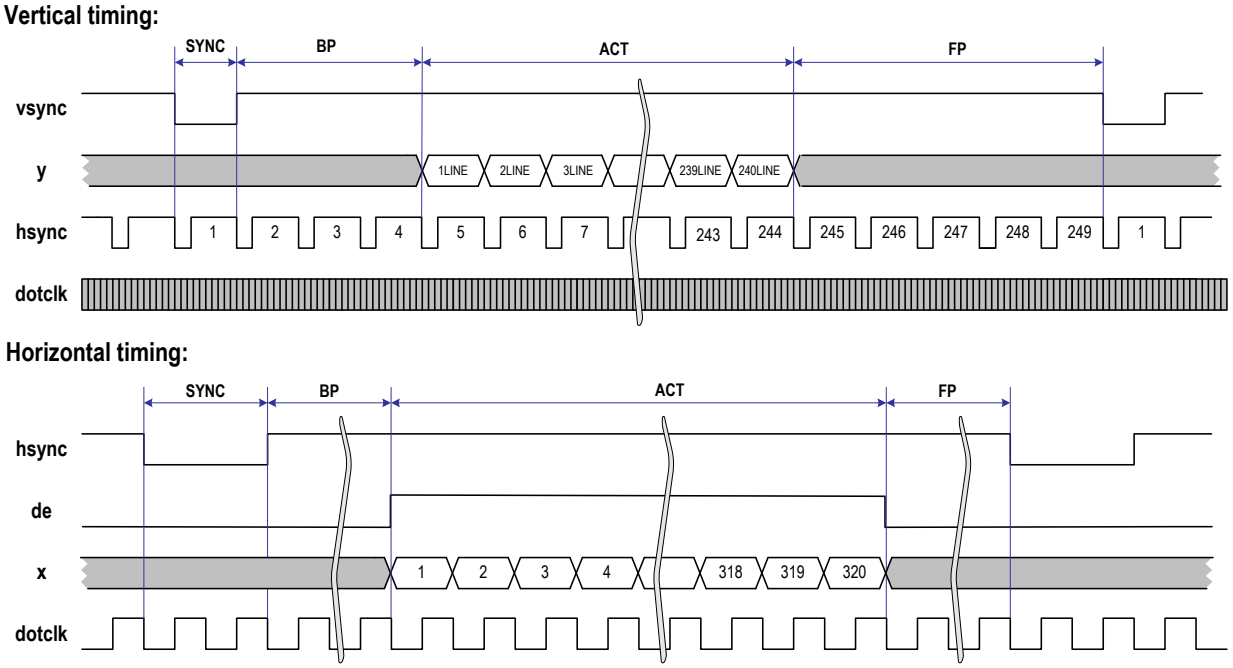
機能の説明

このコンポーネントは、CPU の介入なしに、連続したタイミング信号をパネルに送信します。リフレッシュ期間中に、このコンポーネントは、16 ビットのピクセルデータのフレームを通して、フレームバッファをスキャンする、読み込みリクエストを生成します。垂直および水平のブランキング期間中、コンポーネントはこのフレームバッファ インターフェースに対し、読み込みおよび書き込みを行うことができます。

スクリーンのリフレッシュとタイミング

フレームタイムの間、コンポーネントはvsyncに対して垂直タイミングパターンを生成します。フレームの各ラインを通して、コンポーネントはhsyncに対して、水平タイミングパターンを生成します。hsyncとvsyncに加えて、いくつかのパネルは、スクリーンのリフレッシュ中のアクティブ期間にアクティブHighとなるde (data enable)信号を要求します。リフレッシュ期間の各セグメントのタイミングに違いがあるものの、全てのパネルは、同じ方法で動作します。図 1 は、一般的パネルのタイミング図です。

図 1. 標準パネルタイミング



垂直信号の各フレームのシーケンス、および水平信号の各ラインのシーケンスは、以下のパターンに従います。

- Syncパルス: sync パルスが有効な期間
- バックポーチ: sync パルスが終了してから、アクティブディスプレイまでの期間
- Active: スクリーンに表示する期間
- フロントポーチ: アクティブディスプレイが終了してから、sync パルスが開始するまでの期間

アドレスの生成

スクリーンがリフレッシュされる間、コンポーネントは、画面のピクセルのアドレスを生成するフレームバッファをスキャンする必要があります。各ピクセルに対し、フレームバッファから 16 ビットを一度読み込む必要があります。各フレームの開始に関しては、フレームバッファのインデックスは、フレームバッファの既定の開始地点にリセットされます。この値の初期値は 0 であり、API 関数を使用して変更することができます。フレームバッファのアドレスは、リフレッシュ操作にのみ影響するものの、読み書きに関する API に影響を与えません。

フレームバッファ操作

コントローラのコンポーネントは、読み込みおよび書き込み操作を行うことができます。これらの操作には、以下のパラメータがあります:

- 「読み込み」もしくは「書き込み」
- Address(アドレス): 最大 23 ビットのアドレス
- Data: 16 ビットの値。書き込みは"do" (データ出力)で送信され、読み込みは "di" (データ入力)に合わせて読み込まれます。

このコンポーネントの実装では、23 ビットのアドレスと 1 ビットの読み/書きインジケータを組み合わせます。これにより、アドレスと操作種別を、コンポーネントに対し 3 バイトで送信できます。また、データパス FIFO において、操作タイプとアドレスを合わせて置くことができます。

読み書き操作は、垂直および水平ブランキング期間に行われます。

アイドル状態

フレームバッファ インタフェースにおいて読み込みも書き込みも行われていない場合、インターフェースはアイドル状態にあります。アイドル状態の制御信号は、読み込みのときの値と同じです。アイドル状態の間の出力ピンの値は以下の通りです:

- do: 影響なし (通常は、最後の状態のまま)
- doe: 0
- アドレス 影響なし (通常は、最後の状態のまま)
- nwe: 1
- noe 0

読み込みおよび書き込み操作の説明で明記されていない信号の値は、アイドル状態と同じです。

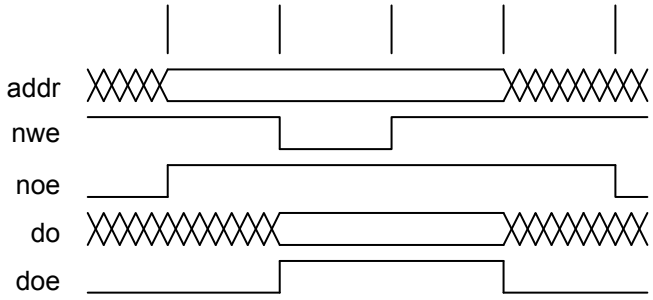
書き込み操作

このコンポーネントでは、書き込み操作が以下の 図 2タイミング図のように実装されています。この図によると、書き込み操作には 4 dotclk サイクルが必要です(全ての図は dotclk 単位)。このトランザクションは、直ちに、または別の読み取りまたは書き込みトランザクションに続いて実行できます。または、書き込みトランザクションの前または後で、アイドル状態になる場合があります。

CPU へのインターフェースにより、CPU がポスティッドライト要求することができます。(アドレスとデータを提供して書き込みを要求し、操作がフレームバッファに対して実際に完了する前に進みます) この実装では、ストールすることなく、4 つの書き込み要求を保持できます。

シグナルのスキューに関係なく、データバスがコンポーネントとフレームバッファの双方から同時に駆動されることがないように、noe および doe 信号パターンが作られていることに留意してください。

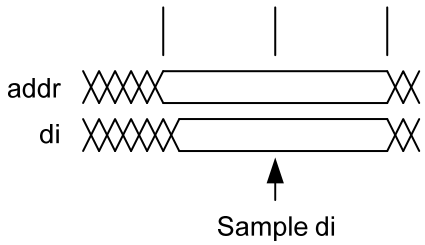
図 2. 書き込みトランザクションタイミングダイアグラム



読み込み操作

このコンポーネントでは、読み込み操作が以下の 図 3タイミング図のように実装されています。このトランザクションは、直ちに、または別の読み取りまたは書き込みトランザクションに続いて実行できます。または、書き込みトランザクションの前または後で、アイドル状態になる場合があります。

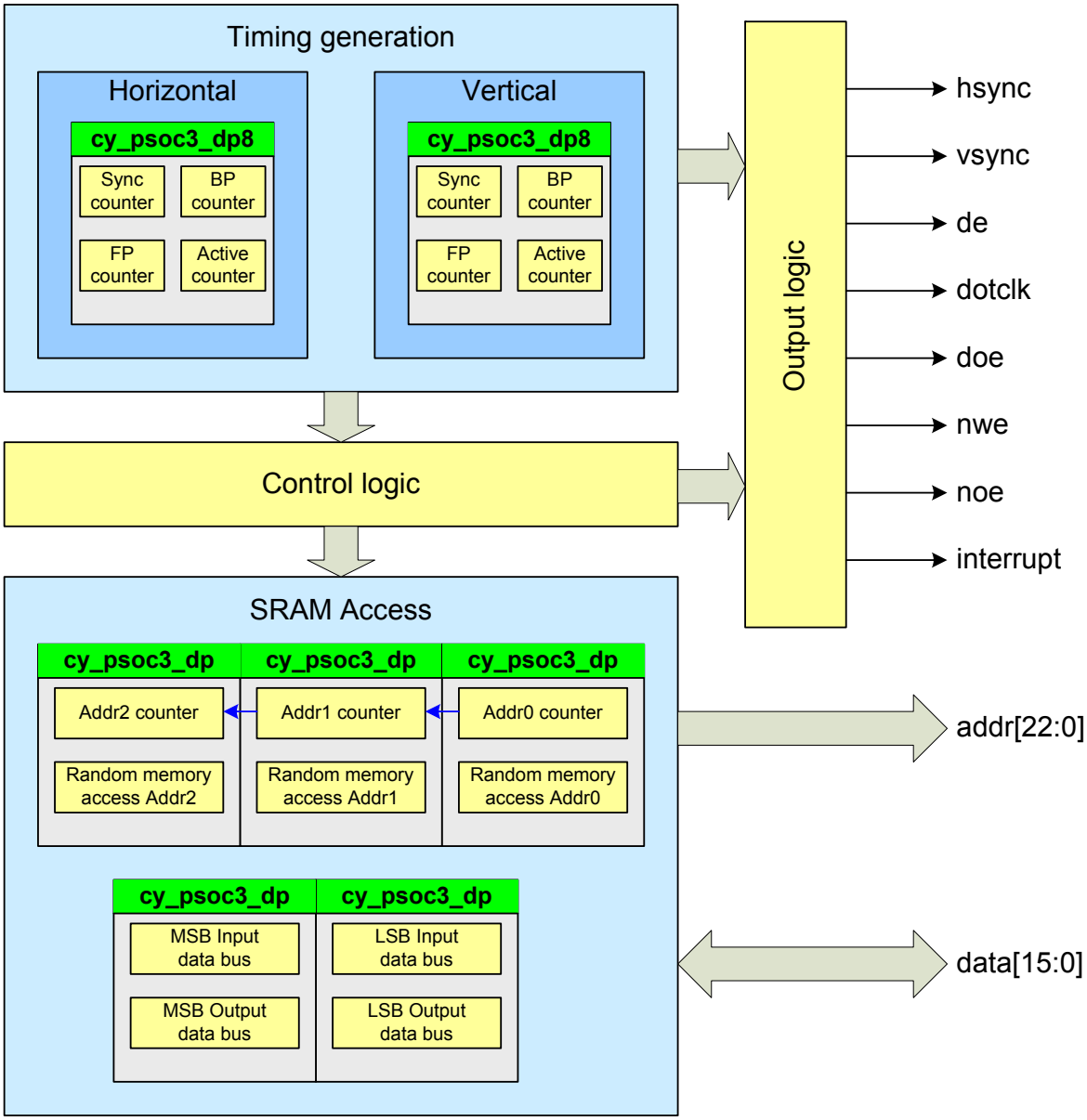
図 3. 読み取りトランザクションタイミングダイアグラム



ブロックダイアグラムと設定

GraphicLCDCtrl コンポーネントは、設定されたUDBの組み合わせとして実装されます。図 4 は、この実装です。

図 4. ブロック図



レジスタ

GraphicLCDCtrl_STATUS_REG

ビット	7	6	5	4	3	2	1	0
値	予約済み				v_blanking	h_blanking	avail	full

- full: コマンドおよびデータ FIFO がフルの場合に設定されます
- avail: データ読み込みが、CPU で有効になる場合に設定されます
- h_blanking: 水平ブランキング期間の間、設定されます
- v_blanking: 垂直ブランキング期間の間、設定されます

DC 電気的特性と AC 電気的特性

以下の値は、期待される性能を示しており、初期特性データを基にしています。

「公称ルーティングでの最大」タイミング特性

パラメータ	説明	Min	Typ	Max [1]	単位
f_{DOTCLK}	Dotclk周波数	–	–	20	MHz
f_{CLOCK}	コンポーネント クロック周波数	$2*f_{DOTCLK}$	–	–	MHz
t_{DOTCLK}	Dotclk 周期	$1/f_{DOTCLK}$	–	–	ns
t_{CKL}	Dotclk Low時間	–	0.5	–	$1/f_{DOTCLK}$
t_{CKH}	Dotclk High時間	–	0.5	–	$1/f_{DOTCLK}$
スクリーン リフレッシュおよびデータ操作タイミング					
t_{HSYNC}	水平syncパルス期間	1	–	256	t_{DOTCLK}
t_{HBP}	水平バックポーチ期間	6	–	256	t_{DOTCLK}
$t_{HACTIVE}$	水平アクティブ期間	4	–	1024	t_{DOTCLK}
t_{HFP}	水平フロントポーチ期間	1	–	256	t_{DOTCLK}

[1]これらの「公称値」は、通常ルーティング状況における、コンポーネントの最大安全動作周波数です。コンポーネントをより高いクロック周波数で実行することはできますが、タイミングの要求項目を STA の結果で検証する必要があります。

パラメータ	説明	Min	Typ	Max [1]	単位
t_{HBLANK}	水平ブランキング期間	–	t_{HSYNC} + t_{HBP} + t_{HFP}	–	t_{DOTCLK}
H_{CYCLE}	水平サイクル	–	t_{HBLANK} + $t_{HACTIVE}$	–	t_{DOTCLK}
t_{VSYNC}	垂直syncパルス期間	1	–	256	H_{CYCLE}
t_{VBP}	垂直バックポーチ期間	1	–	256	H_{CYCLE}
$t_{VACTIVE}$	垂直アクティブ期間	4	–	1024	H_{CYCLE}
t_{VFP}	垂直フロントポーチ期間	1	–	256	H_{CYCLE}
t_{VBLANK}	水平サイクル	–	t_{VSYNC} + t_{VBP} + t_{VFP}	–	H_{CYCLE}
V_{CYCLE}	垂直サイクル	–	t_{VBLANK} + $t_{VACTIVE}$	–	H_{CYCLE}
ピクセル タイミング					
t_{HV}	sync信号の立ち下がりエッジの位相差	–	t_{HFP}	–	t_{DOTCLK}
t_{VSYS}	垂直syncセットアップタイム	–	0.5	–	t_{DOTCLK}
t_{VSYH}	垂直syncホールドタイム	–	0.5	–	t_{DOTCLK}
t_{HSYS}	水平syncセットアップタイム	–	0.5	–	t_{DOTCLK}
t_{HSYH}	水平syncホールドタイム	–	0.5	–	t_{DOTCLK}
t_{DS}	LCDパネルへのデータ セットアップタイム	–	0.5	–	t_{DOTCLK}
t_{DH}	LCDパネルへのデータ ホールドタイム	–	0.5	–	t_{DOTCLK}
フレームバッファの操作タイミング					
t_{AS}	アドレス セットアップタイム	1	–	–	t_{DOTCLK}
t_{AH}	アドレス ホールド時間	–	2	–	t_{DOTCLK}
t_{PWE}	NWE パルス幅	–	1	–	t_{DOTCLK}
t_{DSW}	フレームバッファへのデータ セットアップタイム	–	1	–	t_{DOTCLK}
t_{DHW}	フレームバッファへのデータホールドタイム	–	1	–	t_{DOTCLK}
t_{CYCLE}	クロックサイクル期間				
	書き込みサイクル	4	–	–	t_{DOTCLK}
	読み込みサイクル	2	–	–	t_{DOTCLK}
t_{ACC}	データアクセス期間	–	1	–	t_{DOTCLK}

パラメータ	説明	Min	Typ	Max [1]	単位
t_{OH}	出力ホールドタイム	–	0	–	t_{DOTCLK}

「すべてのルーティングでの最大」タイミング特性

パラメータ	説明	Min	Typ	Max [2]	単位
f_{DOTCLK}	Dotclk周波数	–	–	10	MHz
f_{CLOCK}	コンポーネント クロック周波数	$2*f_{DOTCLK}$	–	–	MHz
t_{DOTCLK}	Dotclk 周期	$1/f_{DOTCLK}$	–	–	ns
t_{CKL}	Dotclk Low時間	–	0.5	–	$1/f_{DOTCLK}$
t_{CKH}	Dotclk High時間	–	0.5	–	$1/f_{DOTCLK}$
スクリーン リフレッシュおよびデータ操作タイミング					
t_{HSYNC}	水平syncパルス期間	1	–	256	t_{DOTCLK}
t_{HBP}	水平バックポーチ期間	6	–	256	t_{DOTCLK}
$t_{HACTIVE}$	水平アクティブ期間	4	–	1024	t_{DOTCLK}
t_{HFP}	水平フロントポーチ期間	1	–	256	t_{DOTCLK}
t_{HBLANK}	水平ブランキング期間	–	$t_{HSYNC} + t_{HBP} + t_{HFP}$	–	t_{DOTCLK}
H_{CYCLE}	水平サイクル	–	$t_{HBLANK} + t_{HACTIVE}$	–	t_{DOTCLK}
t_{VSYNC}	垂直syncパルス期間	1	–	256	H_{CYCLE}
t_{VBP}	垂直バックポーチ期間	1	–	256	H_{CYCLE}
$t_{VACTIVE}$	垂直アクティブ期間	4	–	1024	H_{CYCLE}
t_{VFP}	垂直フロントポーチ期間	1	–	256	H_{CYCLE}
t_{VBLANK}	水平サイクル	–	$t_{VSYNC} + t_{VBP} + t_{VFP}$	–	H_{CYCLE}
V_{CYCLE}	垂直サイクル	–	$t_{VBLANK} + t_{VACTIVE}$	–	H_{CYCLE}
ピクセル タイミング					
t_{HV}	sync信号の立ち下がりエッジの位相差	–	t_{HFP}	–	t_{DOTCLK}

[2]「すべてのルーティング」の最大値は、<公称値>/2 で最近似の整数に切り上げ/切り下げられます。コンポーネントインスタンスがこれら速度以下で動作する場合、このコンポーネントではミートタイミングを気にする必要がなくなります。

パラメータ	説明	Min	Typ	Max [2]	単位
t_{VSYS}	垂直syncセットアップタイム	–	0.5	–	t_{DOTCLK}
t_{VSYH}	垂直syncホールドタイム	–	0.5	–	t_{DOTCLK}
t_{HSYS}	水平syncセットアップタイム	–	0.5	–	t_{DOTCLK}
t_{HSYH}	水平syncホールドタイム	–	0.5	–	t_{DOTCLK}
t_{DS}	LCDパネルへのデータ セットアップタイム	–	0.5	–	t_{DOTCLK}
t_{DH}	LCDパネルへのデータ ホールドタイム	–	0.5	–	t_{DOTCLK}
フレームバッファの操作タイミング					
t_{AS}	アドレス セットアップタイム	1	–	–	t_{DOTCLK}
t_{AH}	アドレス ホールド時間	–	2	–	t_{DOTCLK}
t_{PWE}	NWE パルス幅	–	1	–	t_{DOTCLK}
t_{DSW}	フレームバッファへのデータ セットアップタイム	–	1	–	t_{DOTCLK}
t_{DHW}	フレームバッファへのデータホールドタイム	–	1	–	t_{DOTCLK}
t_{CYCLE}	クロックサイクル期間				
	書き込みサイクル	4	–	–	t_{DOTCLK}
	読み込みサイクル	2	–	–	t_{DOTCLK}
t_{ACC}	データアクセス期間	–	1	–	t_{DOTCLK}
t_{OH}	出力ホールドタイム	–	0	–	t_{DOTCLK}

図 5. スクリーン リフレッシュおよびデータ操作タイミング図

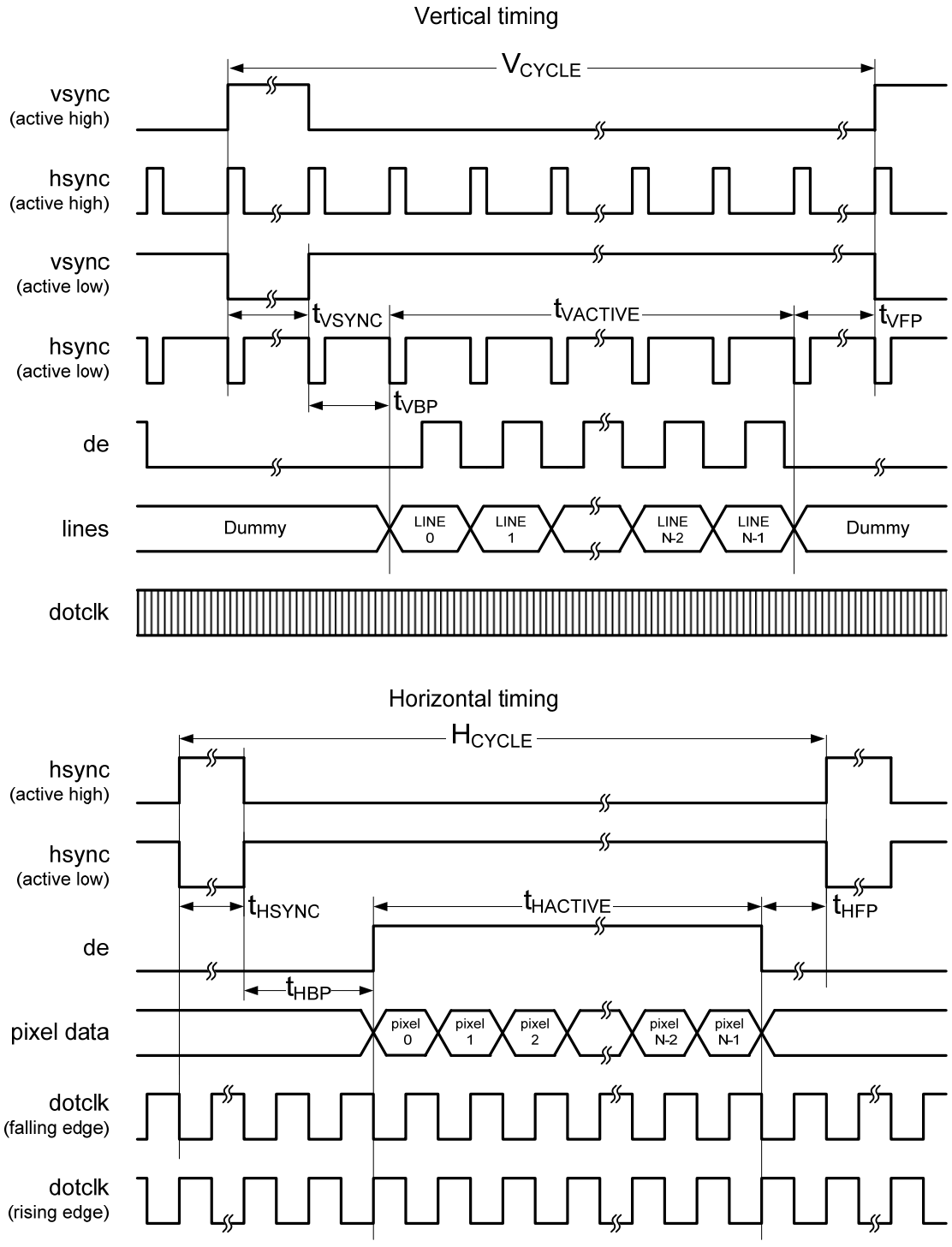


図 6. ピクセルタイミング図

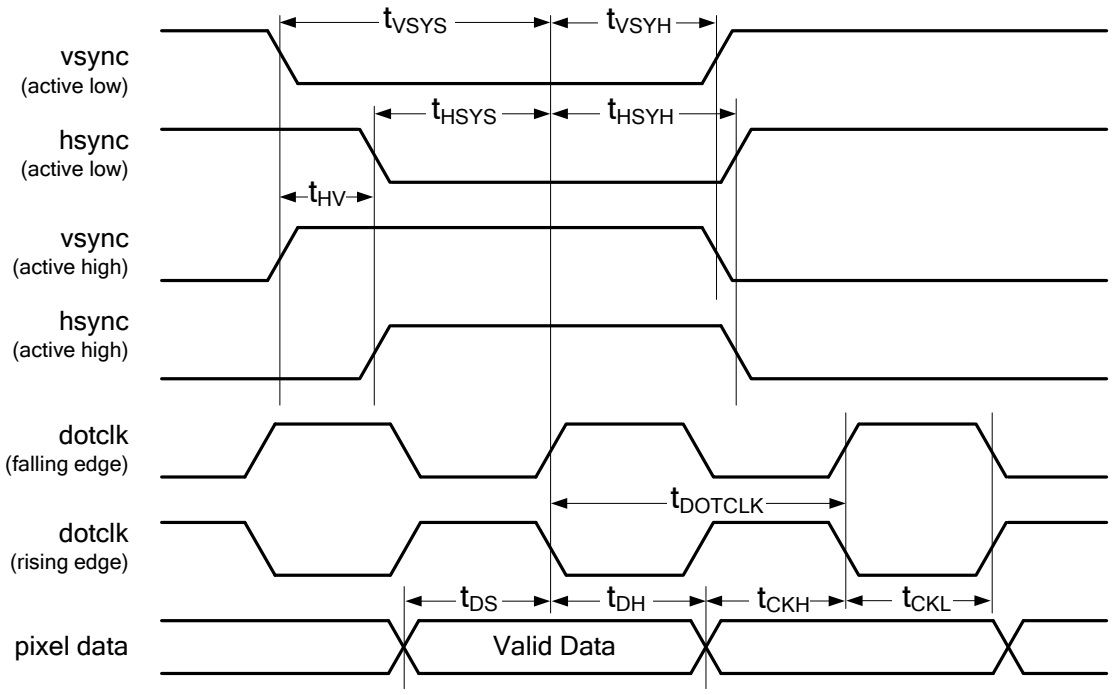
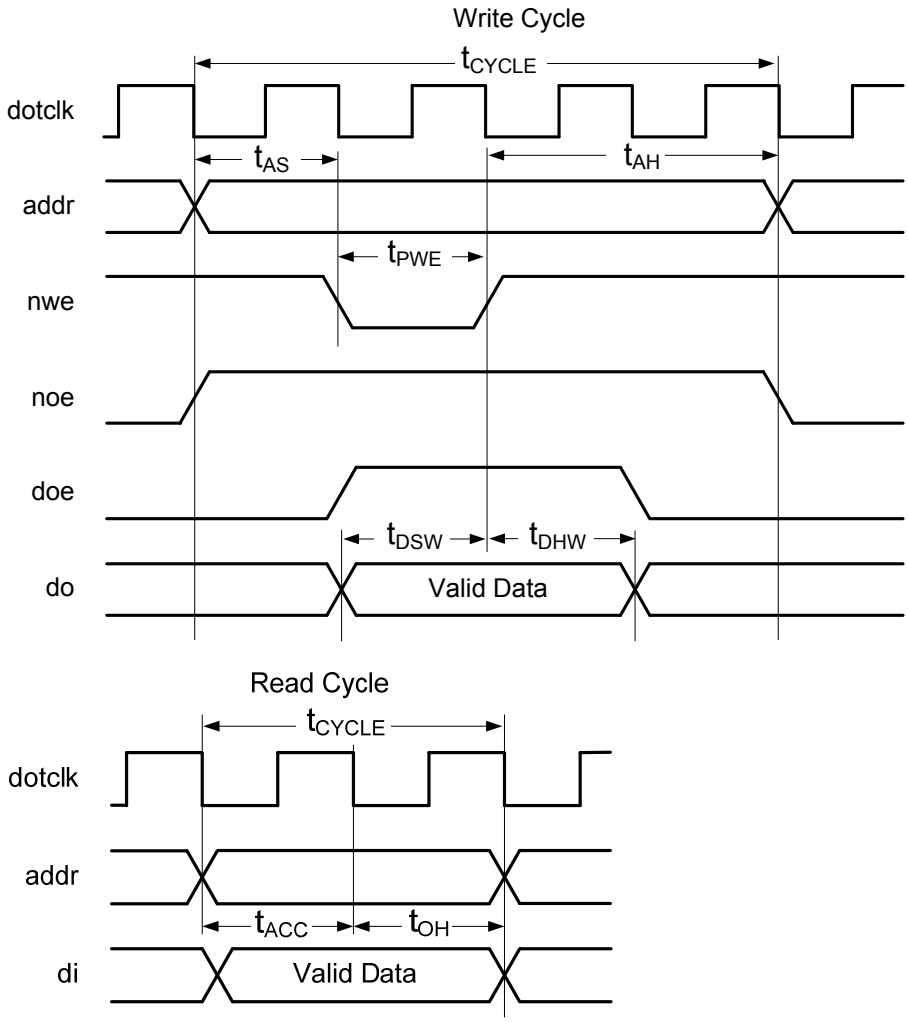


図 7. フレームバッファ データ操作タイミングダイアグラム



特性データ用の STA 結果の使用方法

公称ルーティング最大値は、静的タイミング分析 (STA) を使って、複数のテスト パスから収集されます。STA 結果を用いた場合、次の手法で設計の最大値を計算できます。

f_{clock}　Maximum Component Clock Frequency（最大コンポーネントクロック周波数）が、外付けクロックという名前のクロックサマリにタイミング結果として表示されます。下図は、クロック制限の例を示しています。

- **Clock Summary Section**

Clock	Type	Nominal Frequency	Required Frequency	Maximum Frequency	Violation
CLK	Sync	20.000 MHz	20.000 MHz	41.545 MHz	
ClockBlock/clk_bus	Async	60.000 MHz	60.000 MHz	N/A	
ClockBlock/dclk_0	Async	20.000 MHz	20.000 MHz	N/A	
CyBUS_CLK	Sync	60.000 MHz	60.000 MHz	N/A	
CyILO	Async	1.000 kHz	1.000 kHz	N/A	
CyIMO	Async	3.000 MHz	3.000 MHz	N/A	
CyMASTER_CLK	Sync	60.000 MHz	60.000 MHz	N/A	
CyPLL_OUT	Async	60.000 MHz	60.000 MHz	N/A	

残りのパラメータは、実装固有のものであり、クロックサイクル単位で表されます。これらは、2 つのカテゴリに分けることができます。

- コンポーネントの設定に用いられるパラメータ:

  スクリーン リフレッシュおよびデータ操作タイミングパラメータ

  f_{DOTCLK} パネルを駆動するクロック。このクロックは、入力クロックの半分の周波数です。このコンポーネントにより、パネルが遷移する信号の、dotclkのエッジを変更することができます。パラメータは、「立ち上がりエッジ」あるいは「立下りエッジ」のいずれかに設定できます。「立ち上がりエッジ」が設定された場合、全ての出力信号は、dotclkの立ち上がりエッジに合わせて変化します。「立ち下がりエッジ」が設定された場合、出力信号は、dotclkの立ち下がりエッジに合わせて変化します。これにより、パネルがこれらの信号を、セットアップおよびホールドタイムを満たすように、dotcklの逆のエッジに合わせてサンプルすることができます。

  t_{HSYNC} 水平 hsync パルスがアクティブな期間です。(dotclk 単位) 信号は、アクティブ High(生成されるパルスが正パルス)もしくはアクティブ Low(生成されるパルスが負パルス)のいずれかです。信号の極性は、コンポーネント カスタマイザで設定されます。

  t_{HBP} hsync パルスの終了から、アクティブ期間の開始までの期間です。(dotclk 単位)

  $t_{HACTIVE}$ 水平アクティブ期間(ディスプレイエリア)を定義します。(dotclk 単位)

  t_{HFP} アクティブディスプレイの終了から、hsync パルスの開始までの期間です。(dotclk 単位)

  t_{VSYNC} 垂直 sync パルスが有効な期間です。(H_{CYCLE} 単位) 信号は、アクティブ High(生成されるパルスが正パルス)もしくはアクティブ Low(生成されるパルスが負パルス)のいずれかです。信号の極性は、コンポーネント カスタマイザで設定されます。

  t_{HBP} vsync パルスの終了から、アクティブ期間の開始までの期間です。(H_{CYCLE} 単位)

  $t_{VACTIVE}$ 垂直アクティブ期間(ディスプレイエリア)を定義します。(H_{CYCLE} 単位)

  t_{HFP} アクティブディスプレイの終了から、vsync パルスの開始までの期間です。(H_{CYCLE} 単位)

  V_{CYCLE} 1 つの完全なフレームがアップデートされる期間です。t_{VSYNC}、t_{VBP}、$t_{VACTIVE}$、および t_{VFP} 期間の和として定義されます。

t_{VBLANK} フレーム期間内のブランキングライン数です。この期間中は、フレームバッファをアップデートすることができます。(コンポーネントが、フレームバッファに対し読み書き操作を開始します。) ブランキング期間中は、LCD パネルへのデータの流れはありません。この期間は、t_{VSYNC}、t_{VBP}、t_{VFP} 期間の和です。

H_{CYCLE} 1 つの水平ラインがアップデートされる期間です。t_{HSYNC}、t_{HBP}、$t_{HACTIVE}$、および t_{HFP} 期間の和として定義されます。

t_{HBLANK} 一つの水平ラインの中のブランキングピクセルの数です。この期間中は、フレームバッファをアップデートすることができます。(コンポーネントが、フレームバッファに対し読み書き操作を開始します。) ブランキング期間中は、LCD パネルへのデータの流れはありません。t_{HSYNC}、t_{HBP}、t_{HFP} 期間の和として定義されます。

- コンポーネントの実装によって固定されるパラメータ:

ピクセルタイミングのパラメータ

t_{DOTCLK} dotclk 信号の周期。

t_{CKL} コンポーネントは 50 % デューティーサイクル dotclk を生成します。

t_{CKH} コンポーネントは 50 % デューティーサイクル dotclk を生成します。

t_{VSYS} dotclk 信号のアクティブエッジの前に、vsync 信号が有効な時間の最小値。

t_{VSYH} dotclk 信号のアクティブエッジの後に、vsync 信号が有効な時間の最小値。

t_{HSYS} dotclk 信号のアクティブエッジの前に、hsync 信号が有効な時間の最小値。

t_{HSYH} dotclk 信号のアクティブエッジの後に、hsync 信号が有効な時間の最小値。

t_{VSYS}、t_{VSYH}、t_{HSYS}、t_{HSYH} パラメータは、垂直方向のタイミングについては dotclk と vsync の関係、水平方向のタイミングについては dotclk と hsync の関係で定義されます。パネルへ遷移する信号の、dotclk のエッジを変更することができます。これにより、パネルがこれらの信号を、セットアップおよびホールドタイムを満たすように、dotckl の逆のエッジに合わせてサンプルすることができます。これにより、t_{VSYS}、t_{VSYH}、t_{HSYS}、t_{HSYH} 信号が、dotclk サイクルのおよそ半分のセットアップおよびホールドタイムを確保できます。

t_{HV} sync 信号アクティブエッジの位相差。コンポーネントの実装では、垂直方向の計数は、水平フロントポーチの最初のサイクルで行われます。このため、vsync の hsync 前の位相差は、水平フロントポーチ期間(t_{HFP})に等しいです。

t_{DS} dotclk 信号のアクティブエッジの前に、パネルへの入力においてデータが有効な時間の最小値。

t_{DH} dotclk 信号のアクティブエッジの後に、パネルへの入力においてデータが有効な時間の最小値。

これらのパラメータを求めるには、フレームバッファとして用いられる SRAM のタイミングは、GraphicLCDCtrl コンポーネントの実装と合わせて考慮する必要があります。スクリーンがリフレッシュされる間、コンポーネントは、画面のピクセルのアドレスを生成するフレームバッファをスキャンします。フレームバッファへのアドレスは、dotclk のアクティブエッジに合わせて変化します。dotclk と

address 信号間の遅延はゼロに近いです。これらの二つの信号は、内部コンポーネントのクロックにより生成され、出力ピンへ伝搬します。これにより、dotclk サイクルのおよそ半分の間のホールドタイムを確保できます。セットアップタイムは、dotclk サイクルの水域の半分から、SRAM フレームバッファによる t_{AA} を引いた値です。t_{AA} は、対応する SRAM データシートに記載されています。

フレームバッファデータ操作パラメータ

t_{AS} nwe 信号の立ち下がりエッジの前に、アドレス信号が有効な時間の最小値。

t_{AH} nwe 信号の立ち上がりエッジの後に、アドレス信号が有効な時間の最小値。

t_{PWE} 書き込み信号のロー時間の最小パルス幅。

t_{CYCLE} 一つの操作(読み込みまたは書き込み)がフレームバッファへのインターフェースに行われる間の時間。

t_{DSW} 書き込み信号の立ち下がりエッジの前に、データが有効な時間の最小値。

t_{DHW} 書き込み信号の立ち上がりエッジの後に、データが有効な時間の最小値。

t_{ACC} アドレスが有効になった後、読み込み操作のためにデータがサンプリングされるまでの時間の最小値。

t_{OH} データがサンプリングされる dotclk のアクティブエッジの後にデータが有効であるべき時間の最小値。

コンポーネントの変更

ここでは、過去のバージョンからコンポーネントに加えられた主な変更を示します。

バージョン	変更の説明	変更の理由 / 影響
1.61	.cyre ファイルのすべてのコンポーネント API に CYREENTRANT キーワードを追加しました。	すべての API が真に再入可能とは限りません。コンポーネント API ソースファイルに含まれているコメントは、どの関数が候補であるかを示しています。 この変更には、安全な（フラグまたはクリティカル セクションによる並行コールから保護された）方法で使用される再入可能ではない関数のコンパイラの警告を制限する必要があります。
	コンポーネントの不正なタイミングパスをマークするためのタイミング制約を追加。	タイミング分析から使用されていないパスを削除。これにより、不正なタイミング妨害メッセージを防ぐことができます。
1.60.a	関連キットのリファレンスをデータシートから削除。	
1.60	DPクロックに対する再サンプルされたFIFOブロックステータス信号	これですべての PSoC 3 および PSoC 5 シリコンの同じタイミング結果で、コンポーネントが機能します。
	データシートのマイナーな編集と更新	

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

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

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

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

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