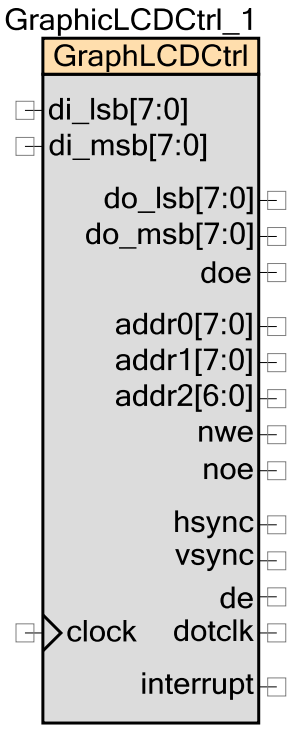
グラフィック LCD コントローラ (GraphicLCDCtrl)

1.60



特長

- 以下のように、HVGA 解像度までのスクリーンサイズをフルプログラム可能
  - 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
- ブランキング期間中の読み書き操作
- 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
- 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
- 垂直および水平のブランキング周期の開始と終了時に生成される割り込みパルスを選択可能

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。この種のパネルには、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。

GraphicLCDCtrl をいつ使用するか

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、SRAM からデータにアクセスし、 dotclk、 hsync、 vsync、 および de 出力の制御を通して LCD 上に表示します。

フレームバッファの SRAM は、LCD パネルのリフレッシュに使用されていない間のみ、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 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。

入力/出力 (I/O)の接続

この節には、GraphicLCDCtrl の様々な入力および出力の接続について記載されています。いくつかの I/O は、その I/O の解説内に記載されている条件によっては、見えないようになっています。

入力	不可視の可能性	説明
di_lsb[7:0]	N	入力データバスの下位8ビット。読み込み処理中のデータに使用。これらの信号は、デバイスの入力ピンに接続してください。また、これらのピンの「Input Synchronized」を無効にしてください。これらの信号は、同期された出力信号により駆動されているため、既に同期しています。
di_msb[7:0]	N	入力データバスの上位8ビット。読み込み処理中のデータに使用。16ビット インターフェースモードのときのみ存在。これらの信号は、デバイスの入力ピンに接続してください。また、これらのピンの「Input Synchronized」を無効にしてください。これらの信号は、同期された出力信号により駆動されているため、既に同期しています。

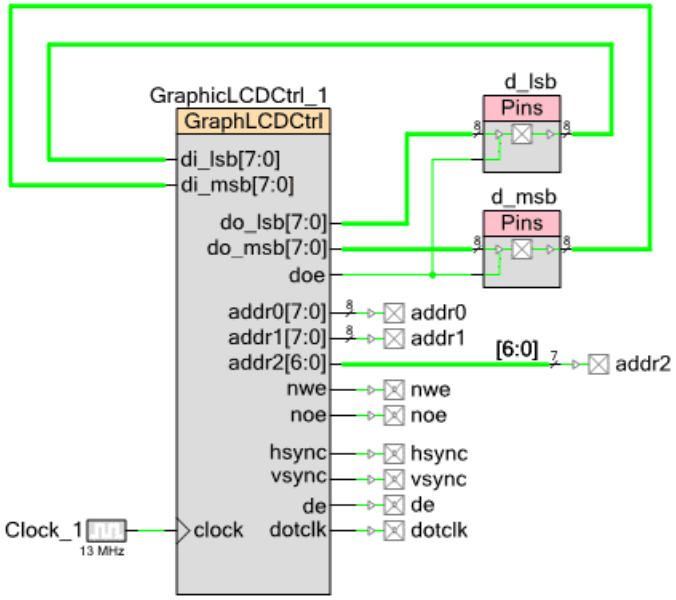
クロック	N	このコンポーネントを駆動するクロック。周波数はdotclkの2倍です。

出力	見えない状態	説明
do_lsb[7:0]	N	出力データバスの下位8ビット。書き込み処理中のデータに使用。
do_msb[7:0]	N	出力データバスの上位8ビット。書き込み処理中のデータに使用。
doe	N	PSoC内のデータバスコンポーネントのoutput enable。通常は、データバスのI/Oピン コンポーネントのoutput enable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。
nwe	N	フレームバッファSRAMのアクティブLowのwrite enable。
noe	N	フレームバッファのアクティブLowのoutput enable。
de	N	パネルのデータenable。
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、見えないようになっています。

回路マクロ情報

このマクロは、Cypress CY8CKIT-016 EBK に使用されている 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あ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 に提供される dotclk は 6.5 MHz になります。



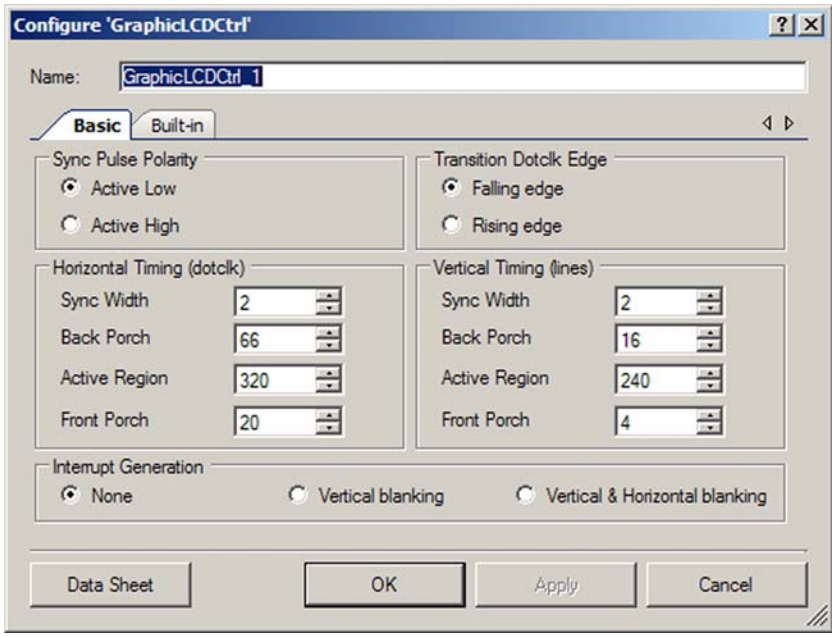
なお、一般的には上位アドレスビットの内いくつかのみ(addr2)が、フレームバッファへの接続に使用されます。この数は、使用される SRAM によって異なります。必要なアドレスビットの数によって、バスの幅を調整するために、たとえば以下の手順を行ってください。

- addr2 出力ピンコンポーネントの、「Number of Pins」を設定してください。
- 出力ピンを駆動する信号を右クリックし、[Edit Name And Width(名前と幅の編集)]を選択します。出力ピンの幅を反映させるため、[Left Index(左インデックス)]を調整します。

また、データピンの全てに対し[Input Synchronized]のチェックを解除し、全てのピンに対する API の生成を無効にします。

パラメータと設定

デザインの上に GraphicLCDCtrl コンポーネット をドラッグし、ダブルクリックして[設定]ダイアログを開きます。デフォルトの GraphicLCDCtrl 設定は、Optrex ボードと Cypress CY8CKIT-016 EBK と合わせて使用する設定になっています。



[Configure GraphicLCDCtrl (GraphicLCDCtrl の設定)]ダイアログには、以下のパラメータが含まれます。これらの設定全ては、コンパイル時に反映され、実行するときに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

これらの設定により、hsync および vsync 信号がアクティブ High かアクティブ Low になります。hsync および vsync の極性は共に、この設定一つで決定されます。デフォルト設定はアクティブ Low 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)

- **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。
- **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。
- **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。アクティブ期間は、4 の倍数に設定します。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。
- **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 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。
- **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。
- **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。アクティブ期間は、4 の倍数に設定します。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。
- **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「なし」です。

- 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。
- 「垂直および水平帰線消去」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。垂直ブランキング期間中は、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。

クロックの選択

このコンポーネントに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

リソースタイプ				API メモリ（バイト数）		ピン（外部 I/O あたり）
データパスセル	PLD	ステータスセル	コントロール / カウント7 セル	フラッシュ	RAM	
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 インタフェース(API)により、ソフトウェアを用いてコンポーネントを設定することが可能になります。以下の表に、各関数およびインターフェースの解説が記載されています。以下の節では、各関数に関するより詳細な説明が述べられています。

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。このインスタンスの名前を、識別子の命名規則に従った、任意の別のものに変えることができます。そのインスタンス名が、

全てのグローバル関数、変数、および定数記号の接頭語になります。読みやすくするため、以下の表で "GraphicLCDCtrl"をインスタンス名として用います。

関数	説明
void GraphicLCDCtrl_Start(void)	GraphicLCDCtrl インタフェースを開始します。
void GraphicLCDCtrl_Stop(void)	GraphicLCDCtrl インタフェースを無効化します。
void GraphicLCDCtrl_Write(uint32 addr, uint16 data)	フレームバッファへの書き込み操作を開始します。
uint16 GraphicLCDCtrl_Read(uint32 addr)	フレームバッファからの読み込み操作を開始します。
void GraphicLCDCtrl_WriteFrameAddr(uint32 addr)	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。
uint32 GraphicLCDCtrl_ReadFrameAddr(void)	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。
void GraphicLCDCtrl_WriteLineIncr(uint32 incr)	隣接するライン間のアドレス間隔を設定します。
uint32 GraphicLCDCtrl_ReadLineIncr(void)	ライン間のアドレスのインクリメントを読み込みます。
void GraphicLCDCtrl_Sleep(void)	GraphicLCDCtrlの設定を保存し、無効化します。
void GraphicLCDCtrl_Wakeup(void)	GraphicLCDCtrlの設定を復元し、有効化します。
void GraphicLCDCtrl_Init(void)	コンポーネント パラメータを初期化、もしくはコンポーネント カスタマイザの保持する初期値に復元します。
void GraphicLCDCtrl_Enable(void)	GraphicLCDCtrlを有効化します。
void GraphicLCDCtrl_SaveConfig(void)	GraphicLCDCtrlの設定を保存します。
void GraphicLCDCtrl_RestoreConfig(void)	GraphicLCDCtrlの設定を復元します。

グローバル変数

変数	説明
GraphicLCDCtrl_initVar	GraphicLCDCtrlが初期化されたか否かを示します。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、およびクロックのゲーティングを必要に応じて有効化します。これは、コンポーネントの操作を開始するにあたり、最も推奨される方法です。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

戻り値： 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化します。

パラメータ： なし

戻り値： 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操作を開始します。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： アドレス 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]).

データ 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

戻り値： 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操作を開始します。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、読み込みが完了するまで待機し、その後に読み込み値を返します。

パラメータ： アドレス 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]).

戻り値： 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、フレームバッファアドレスの始点を設定します。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： アドレス フレームバッファの始点のアドレス

戻り値： 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、フレームバッファアドレスの始点を読み込みます。

パラメータ： なし

戻り値： 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。

戻り値： 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

戻り値： 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： コンポーネントをスリープさせる場合の、最も推奨されるルーチーンです。
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。パワーマネージメントの関数についての詳細な情報については、PSoC Creator *システム リファレンスガイド(System Reference Guide)*を参照してください。

パラメータ： なし

戻り値： 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

戻り値： 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、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

戻り値： なし

副作用： このコンポーネントは、このAPIコールの前にGraphicLCDCtrl_Stop APIにより無効化される必要があります。そうしない場合、予期せぬ挙動を示す場合があります。. コンポーネントは、以下の例外を除いて再初期化されます。FIFOからデータはクリアされず、コンポーネント ハードウェア ステートマシンを再起動しません。

void GraphicLCDCtrl_Enable(void)

説明： ハードウェアを有効化し、コンポーネントの操作を開始します。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

戻り値： 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を保存します。非保持レジスタの内容も保存されます。この関数は、[設定]ダイアログにより定義された、もしくは適切なAPIにより変更された、現在のコンポーネントの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

戻り値： 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を復元します。非保持レジスタの内容も復元されます。この関数は、コンポーネントのパラメータ値を、GraphicLCDCtrl_Sleep()関数を呼び出す以前の値に復元します。

パラメータ： なし

戻り値： 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 ソースコードの例

CY8CKIT-016 Graphic LCD コントローラ キットに付随する例を参照してください。コントローラの初期化以外に、一般的にはこのコンポーネントは Segger emWin Graphics コンポーネントにより使用されます。

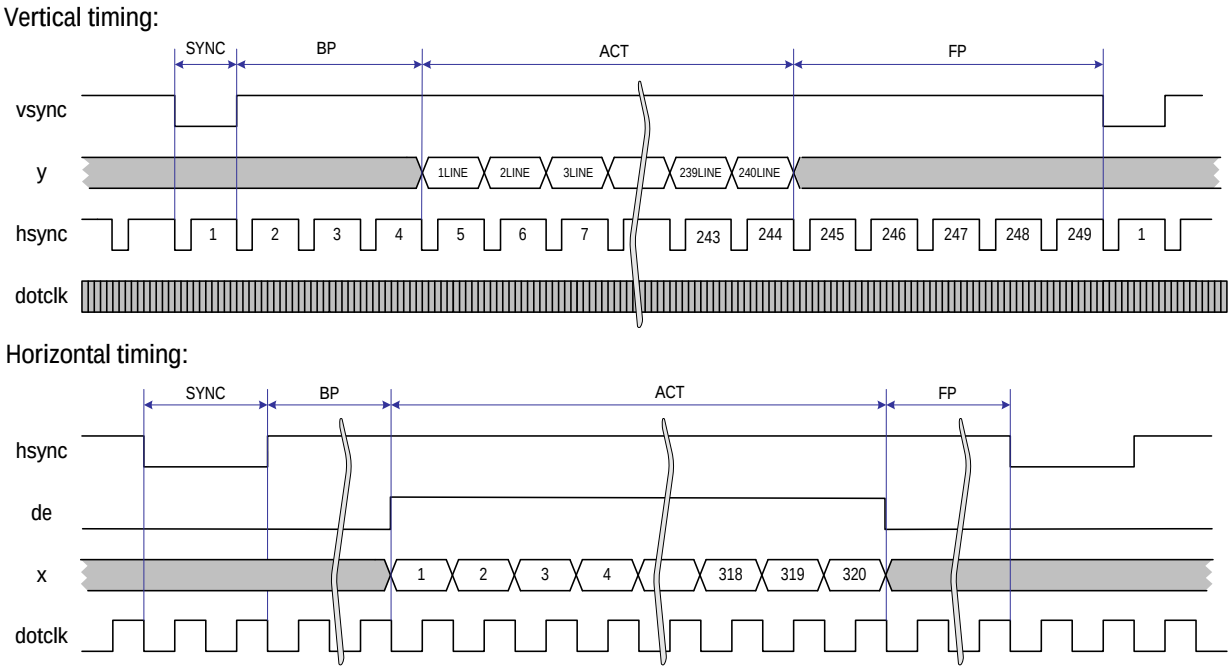
機能説明

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。以下は、一般的なパネルのタイミング図です。



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、以下のパターンにしたがいます。

- 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。
- アクティブ: 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 関数を通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」
- アドレス: 最大 23 ビットのアドレス。
- データ: 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/書き指示子を組み合わせます。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
- addr: 影響なし (通常は、最後の状態のまま)
- nwe: 1
- noe 0

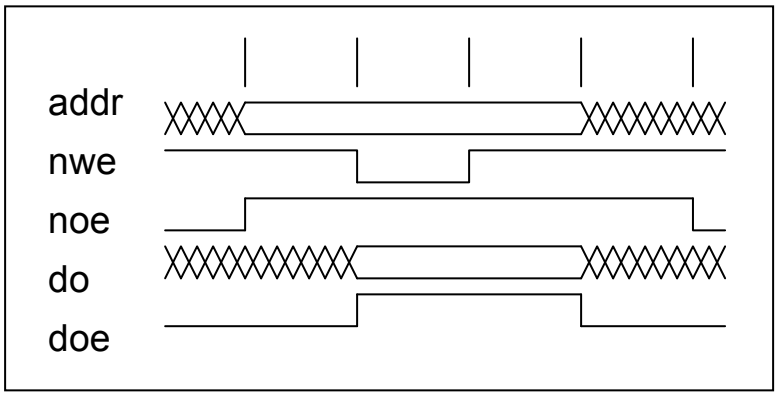
読み込みおよび書き込み操作の説明で明記されていない信号の値は、アイドル状態と同じです。

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)。この操作は、他の読み込みまたは書き込み操作と連続して行うことができるほか、アイドル状態から、もしくはアイドル状態へ移行することも可能です。

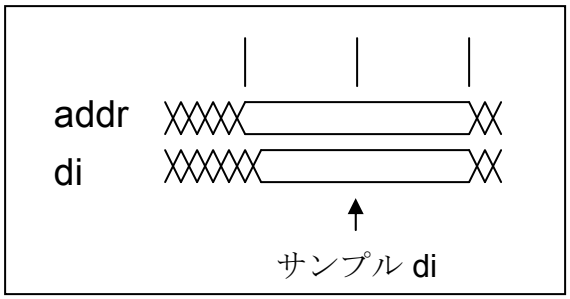
CPU へのインターフェースにより、CPU がポスティッドライト要求することができます。(アドレスとデータを提供して書き込みを要求し、操作がフレームバッファに対して実際に完了する前に進みます) この実装では、CPU がストールすることなく、4 つの書き込み要求を保持できます。

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。



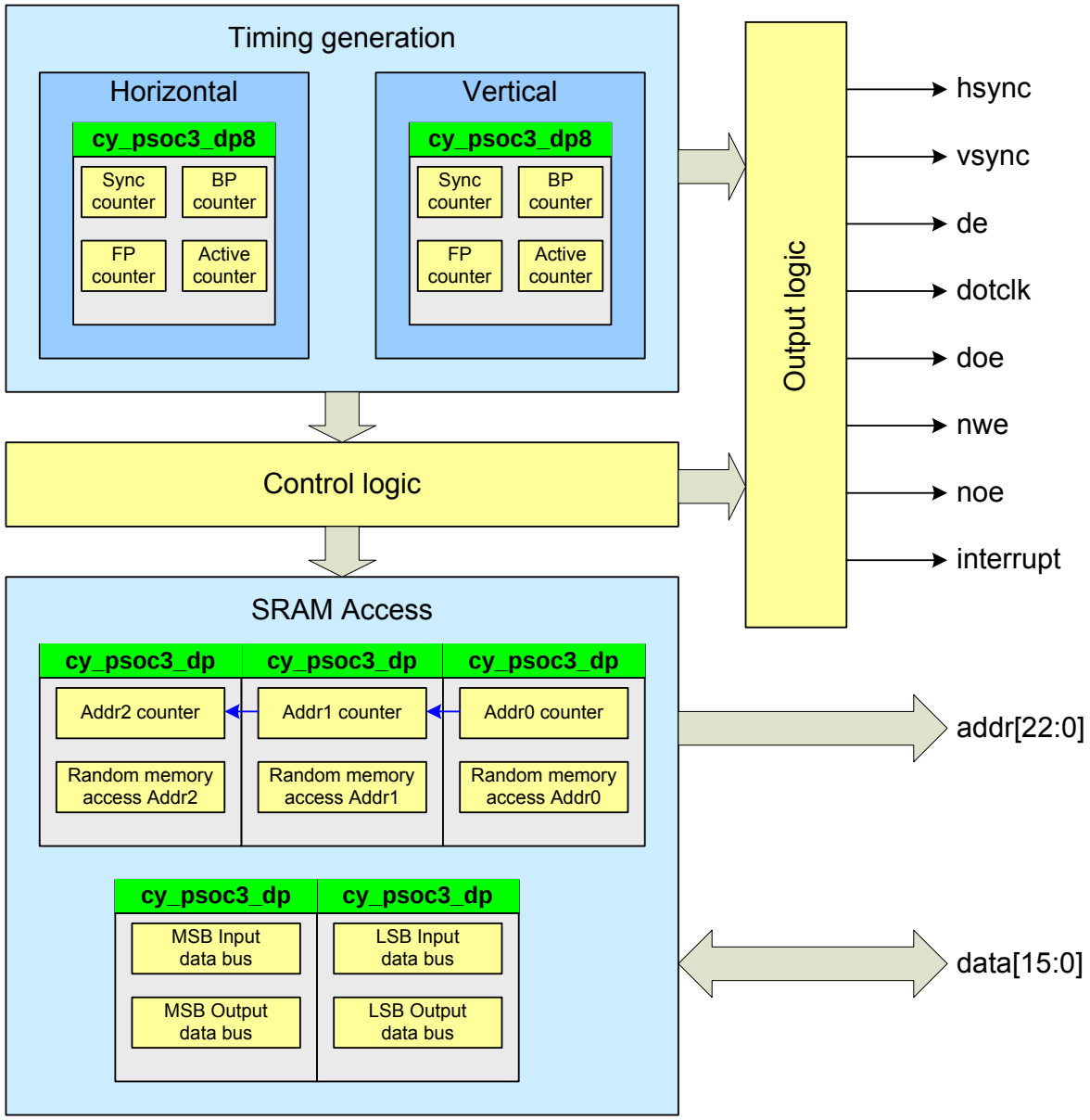
読み込み操作

このコンポーネントでは、読み込み操作が以下のタイミング図のように実装されています。この操作は、他の読み込みまたは書き込み操作と連続して行うことができるほか、アイドル状態から、もしくはアイドル状態へ移行することも可能です。


addr
di
サンプル di

ブロック図および設定

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。. 実装は以下のブロック図のように行われます。



レジスタ

GraphicLCDCtrl_STATUS_REG

ビット	7	6	5	4	3	2	1	0
値	reserved				v_blanking	h_blanking	avail	full

- full: コマンドおよびデータ FIFO が FULL 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。

備考

該当なし

DC 電気的特性と AC 電気的特性

以下の値は、期待されるパフォーマンスのもので、初期特性データに基づいています。

タイミング特性 「最大値、通常のルーティング」

パラメータ	説明	Min	Typ	Max [1]	単位
f_{DOTCLK}	Dotclk周波数	–	–	15	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同期パルス期間	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}

[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駆動することは可能ですが、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_{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}	水平同期セットアップ期間	–	0.5	–	t_{DOTCLK}
t_{HSYH}	水平同期待機期間	–	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 ホールドタイム	–	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}

タイミング特性 「全てのルーティングでの最大値」

パラメータ	説明	最小値	典型値	最大値 [1]	単位
f_{DOTCLK}	Dotclk周波数	–	–	7	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}

[1]「全てのルーティング」時の最大値は、<公称値>を 2 で割ったものを、四捨五入して整数にした値です。ユーザーにとって、コンポーネントをこの値の周波数以下で動作させているときに、タイミングの心配をしなくて済む、目安を示しています。

パラメータ	説明	最小値	典型値	最大値 [1]	単位
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 [4]					
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 [5]					
t_{AS}	アドレス セットアップタイム	1	–	–	t_{DOTCLK}
t_{AH}	アドレス ホールドタイム	–	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}

パラメータ	説明		最小値	典型値	最大値 [1]	単位
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 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

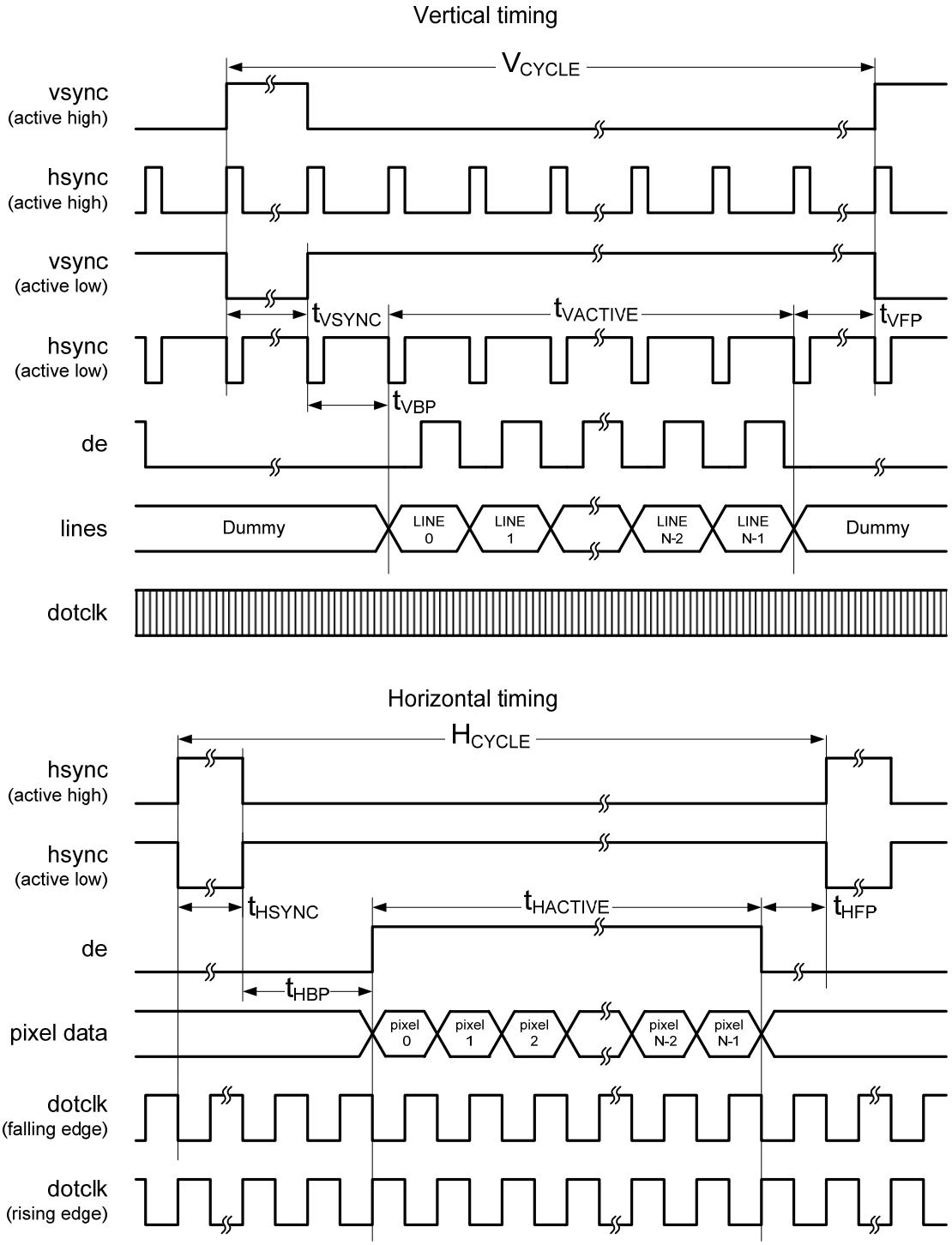


図 2 ピクセルタイミング図

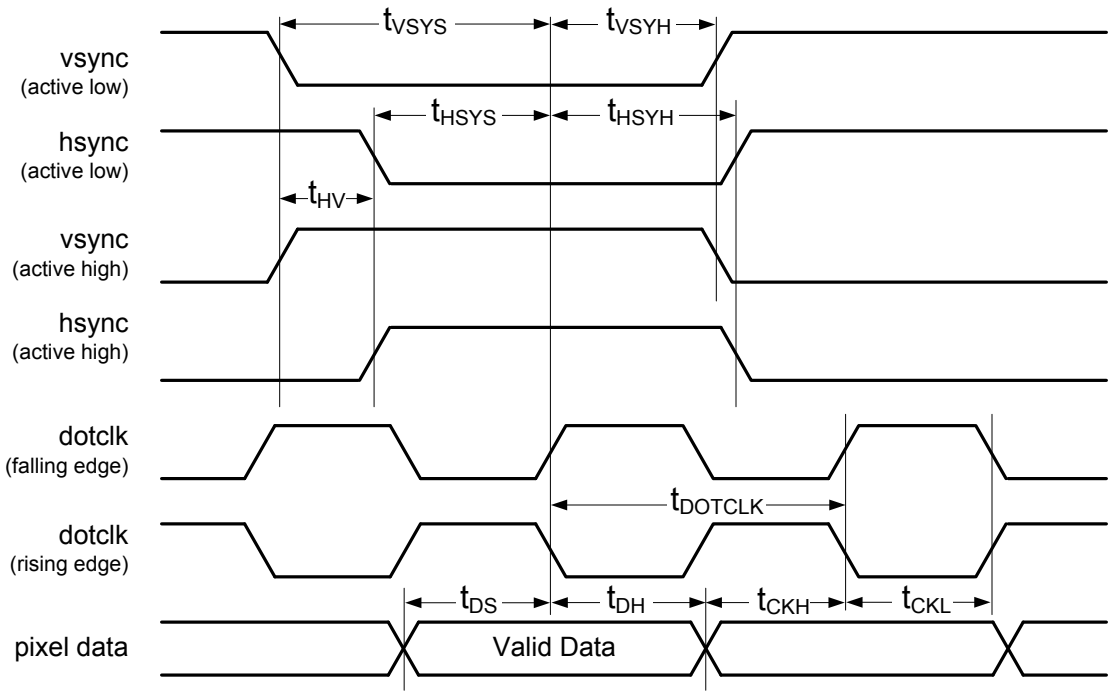
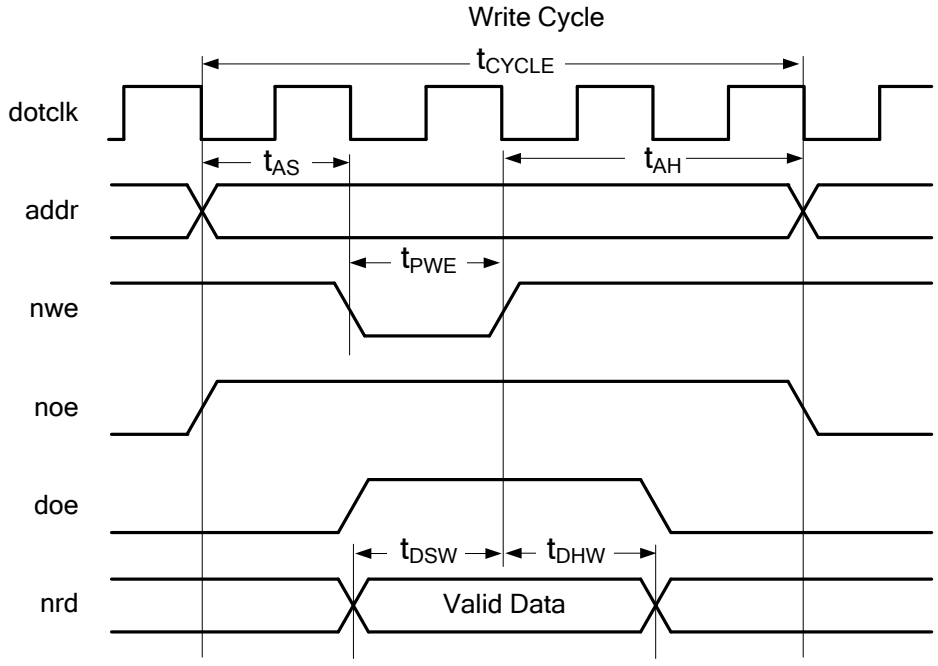
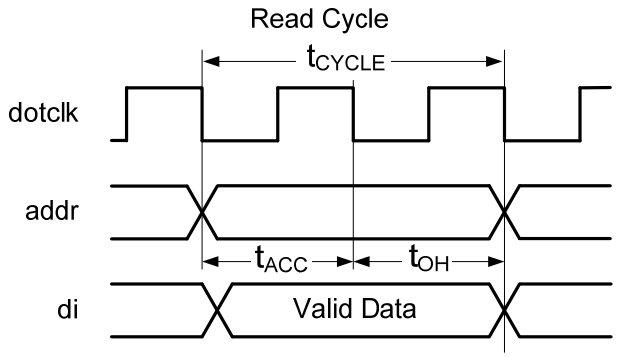


図 3 フレームバッファ データ操作タイミングダイアグラム





静的タイミング解析結果を用いて特性データ求める

公称ルートの最大値は、静的タイミング解析(STA)による複数のテストによるものです。各デザインの最大値は、STA 結果から以下の方法で求めることができます:

f_{clock}　コンポーネントの最大クロック周波数は、名づけられた外部クロックとして、クロックの摘要のタイミングの結果に表れます。以下の画像は、*_timing.html* における、クロックの制限の一例です:

-Clock Summary

Clock	Actual Freq	Max Freq	Violation
BUS_CLK	60.000 MHz	UNKNOWN	
Clock	20.000 MHz	27.820 MHz	

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は:

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の工ッジを変更することができます。この設定は、「立ち上がりエッジ」あるいは「立下りエッジ」のいずれかです。「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。「立ち上がりエッジ」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dotcklの逆の工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。(HCYCLE 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。(HCYCLE 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}、および 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}、 および 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は:

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、デューティ比 50%の dotclk を生成します。

t_{CKH} コンポーネントは、デューティ比 50%の 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。

注: t_{VSYS}、 t_{VSYH}、 t_{HSYS}、および 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。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。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}、および 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 シグナルの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 のタイミングが、GraphicLCDCtrl コンポーネントの実装と合わせて考慮される必要があります。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

口に近いです。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_{AHT}　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}　書き込み信号の口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点

この節には、コンポーネントにおける、前のバージョンからの主な変更点が記載されています。

バージョン	変更点の説明	変更の理由と影響
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	全てのPSoC3 およびPSoC5シリコンで、コンポーネントが同じタイミングで動作するようになりました。
	データシートの細かい修正および更新がありました。	

© Cypress Semiconductor Corporation, 2011. 本文書に記載されている情報は、事前の予告なしに変更される場合があります。サイプレス セミコンダクタ社では、サイプレス製品に統合されている以外の回路の使用については、一切の責任を負いません。また、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、含意することもありません。サイプレス製品は、サイプレスとの明示的な書面による合意がない限り、医療、生命維持、救命、重要な管理、または安全に関わる用途での使用を保証するものではなく、そのような使用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が重大な傷害を負うことが妥当に予測される、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は許可していません。生命維持システム用途にサイプレス製品を使用することは、製造者がそのように使用する上での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ます。

PSoC Designer™、Programmable System-on-Chip™、PSoC Express™はサイプレス セミコンダクタ社の商標であり、PSoC®はサイプレス セミコンダクタ社の登録商標です。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。

すべてのソースコード (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って保護され、それらの規定に従います。サイプレスが本書面により被免許者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、該当す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被免許者の製品のみをサポートするカスタム ソフトウェアやカスタム 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、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、本ソースコードのいかなる複製、変更、変換、コンパイル、または表示もサイプレスの明示的な書面による許可がない限り禁止されています。

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を含みますが、それらに限定されません。サイプレスは、本文書に記載した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、本文書に記載した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り生じる一切の責任を負いません。さらに、サイプレスは誤動作や故障によって使用者が重大な傷害を負うことが妥当に予測される、生命維持システムの重要なコンポーネントとしてのサイプレス製品の使用を許可していません。生命維持システム用途にサイプレス製品を使用することは、製造者がそのように使用する上での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、その規定に従います。