

## Smart-USB Plus 製品対応

## 1word ハンドシェイク通信の概要

## 1. 概要

Smart-USB Plus 製品ファミリの各製品は、USB 制御用 IC としてサイプレス社の CY7C68013A デバイス(以下、USB 制御 IC)を採用しています。この USB 制御 IC と FPGA がボード上で接続しています。

このアプリケーション・ノートでは、USB 制御 IC と FPGA がハンドシェイクしながら、データ転送を実現するための FPGA 回路について解説します。

## 1.1 バースト・モードとの違い

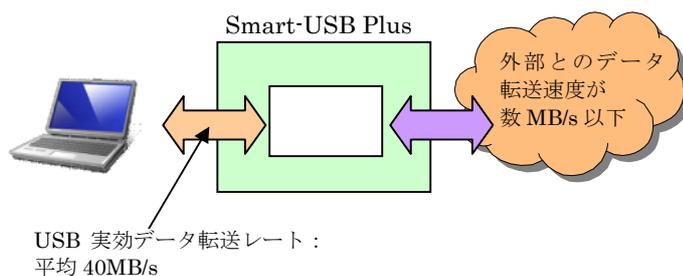
バーストモードは、USB 制御 IC と FPGA 間のデータ転送を最短の時間で連続して行うモードです。

ボード上の FPGA 外部メモリにデータを一旦蓄積する場合には、このモードが有効です。通常、高速にデータ転送する場合はこのモードを使います。

しかし、ボード上の外部メモリや FPGA 内部メモリを FIFO のようにバッファ・メモリとして利用し、連続したデータを取り扱う場合には注意が必要です。図 1 で示すように「PC=ボード」間のデータ転送レートと、「ボード=外部システム」間のデータ転送レートが一致していなければ、特定のタイミングで WAIT 制御が必要になります。

バースト・モードでは、512 バイト(48MHz クロック 256 個分)単位(最大 2,048 バイトの場合もあります)のデータを連続して転送します。この単位のデータ転送を複数回実施して、全データの転送を完了します。この間、バッファメモリが Empty または Full の場合、「PC=ボード」間のデータを一時停止しなければ正しいデータ転送ができなくなります。このため、WAITn 信号を利用して、ホスト PC にデータ転送を一時停止させ、正常にデータ転送を行います。

バーストモードでの WAITn 信号は、最大 1,024 クロック単位の期間中、アクティブにしても無視されます。このため、最低 4K バイトのバッファメモリが必要です(8KB 以上のバッファを推奨します)。



<図 1. データ転送レートの違い>

## 1.2 1Word ハンドシェイク通信とは

バースト・モードの場合、FPGA 内部または外部メモリにバッファが必要でしたが、1Word ハンドシェイク通信(以下、ワード・モード)では、1 ワード単位に WAITn 信号を利用してデータ転送することができます。

ホスト PC とボード間のデータ転送レートよりも低速なデータを扱う場合に便利な機能です。また、回路規模が小さくなることも利点の一つです。ロジックセル使用数はわずか 100 個未満、FPGA 全体の使用率は1%未満です。FPGA の内部メモリを他のロジックで全て消費した場合でも、WAITn 信号を使ったワード・モード転送により、連続したデータを長時間収集、生成することができます。

## 1.3 モードの切替方法

ワード・モード、バースト・モードにかかわらず、リファレンス制御アプリケーション(RefApp7.exe)でのメモリ操作は同じです。

RefApp7.exe のツールバーから、「メモリ転送モード切替」を選択し、リード時とライト時にそれぞれどちらのモードを選択するか選択できます。ボード電源投入後やボードリセット後はバースト・モードになります。



<図 2. RefApp7.exe での設定画面>

## (注意)

ワード・モードに設定後、ボードの電源を再投入したり、電圧低下やリセットスイッチの押下でシステムリセット状態になった場合、ワード・モードからバースト・モードになります。ワード・モードで運用する場合は、再設定してください。

## 【適用ボード】

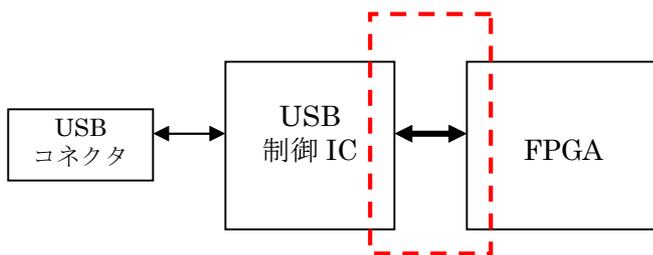
- ◆ CX-Card2(ファームウェア ver.0202 以上)
- ◆ SX-Card3(ファームウェア ver.0202 以上)
- ◆ SX-USB3(ファームウェア ver.0202 以上)
- ◆ CX-Card4(ファームウェア ver.0200 以上)
- ◆ ASUP-M3(ファームウェア ver.0200 以上)
- ◆ SX-Card6(ファームウェア ver.0200 以上)

## 2. インタフェース・プロトコルの概要

USB 制御 IC と FPGA 間のインタフェース・プロトコルは以下の信号線を使用して行います。図 3 で示す赤枠の部分の信号線です。

USB 制御 IC ピン名	信号名	FPGA からみた I/O 方向
IFCLK	IFCLK	入力
CTL[0]	RDn	入力
CTL[1]	WRn	入力
CTL[2]	CMDn	入力
CTL[3]	RGDTn	入力
RDY[0]	WAITn	出力
FD[15:0]	USB_DATA[15:0]	双方向

この他、未使用ピンとして RDY[5:1] が FPGA に接続しているボードもあります。この場合、FPGA コンパイル時にこれらのピンをトライステート状態(未使用ピンとして記述)にしてください。



<図 3. Smart-USB Plus 製品の基本構成>

### 2.1 信号線の概要

信号名 : IFCLK

信号方向 : USB 制御 IC ← 48MHz OSC → FPGA

機能 : USB 制御を行うための基本クロックです。外部のクロック源(水晶発振器)から USB 制御 IC と FPGA 間の両方に同位相のクロックを供給します。

信号名 : RDn ( CTL[0] ) (Active Low)

信号方向 : USB 制御 IC → FPGA

機能 : FPGA が USB 制御 IC にデータを送信する際、この信号に同期して FPGA が USB データバス (FD[15:0]) 上にデータを送出します。この信号は IFCLK の立ち上がりエッジに同期しています。

信号名 : WRn ( CTL[1] ) (Active Low)

信号方向 : USB 制御 IC → FPGA

機能 : USB 制御 IC が FPGA にデータを送信する際、この信号に同期して USB 制御 IC が USB データバス

(FD[15:0]) 上にデータを送出します。この信号は IFCLK の立ち上がりエッジに同期しています。

信号名 : CMDn ( CTL[2] ) (Active Low)

信号方向 : USB 制御 IC → FPGA

機能 : USB 制御 IC が FPGA に対してコマンドを発行していることを示します。この信号に同期して USB 制御 IC が USB データバス上にコマンドデータを送出します。受信する FPGA では、この信号により USB データバス上のデータが、コマンドであることを判断することができます。この信号は IFCLK の立ち上がりエッジに同期しています。

信号名 : RGDTn ( CTL[3] ) (Active Low)

信号方向 : USB 制御 IC → FPGA

機能 : USB 制御 IC が FPGA に対してコマンドを発行する際、この信号により、レジスタ・アクセスかメモリ・アクセスかを表示します。High レベルの場合、レジスタ・アクセスであることを表示し、Low レベルの場合、メモリ・アクセスであることを示します。この信号は IFCLK の立ち上がりエッジに同期しています。

信号名 : WAITn ( RDY ) (Active Low)

信号方向 : USB 制御 IC ← FPGA

機能 : USB 制御 IC とのデータ転送中(メモリ・アクセス)、FPGA 側でデータの転送を任意にウエイトすることができます。

### 2.2 アクセス制御

USB 制御 IC は、バルク転送(以下、メモリ転送)を実行することをメモリコマンドで示し、FPGA がそのコマンドを解釈してデータ転送を開始します。バースト・モード、ワード・モード共通です。

ホスト PC の制御アプリケーションでは、次の専用 API を発行することで、メモリ・アクセスを開始します。

FD[15:0] バス上のデータの流れは以下の通りです。

```

開始コマンド + データ 0 + データ 1 + ...
                ..... + データ n-1 + 終了コマンド
  
```

【対応する API (USB プロトコルを実現する専用関数)】

- \* SUP64\_Data\_Write  
(メモリデータの書き込み動作)
- \* SUP64\_Data\_Raed  
(メモリデータの読み出し動作)

Windows 上で動作する USB 制御アプリケーションが発行するこれらの API により、USB バルク転送を開始します。

## 【コマンド・データの内容】

Bit	7	6	5-2	1	0
内容	アクセス種別		不定	転送状態	転送モード

## アクセス種別

00: 8bit アクセス、 01: 16bit アクセス

10: 32bit アクセス、 11: 64bit アクセス

(注:コマンド・データの上位 8bit は不定データです)

## 転送状態 ;

"1": メモリ転送の開始を表示

"0": メモリ転送の終了を表示

## 転送モード ;

"1": メモリ・リードの開始または終了を表示

"0": メモリ・ライトの開始または終了を表示

CMDn 信号が Low レベルの時、RGDTn 信号が Low レベルになると、ホスト PC がボードに対してメモリ転送を行うことを示します。このとき、USB 制御 IC は FD[15:0]バス上にメモリ・コマンド(開始コマンド)を出力します。FPGA ではこのデータをデコードし、このコマンドがメモリ転送であることとアクセス種別を検出します。コマンド出力後、メモリリードであれば RDn、メモリライトであれば WRn 信号がアクティブになり、データ転送が始まります。すべてのデータを転送後、USB 制御 IC は終了コマンドを発行し、メモリ転送動作が完了します。

## 2.3 メモリ・ライト

(PC からボードへのデータ転送)

メモリライト時の手順を示します。図 4 のタイミングチャートを参照してください。

1. メモリデータ転送の前に、USB 制御アプリケーションでメモリの開始アドレスと転送するデータ量(バイト)を決めます。メモリの開始アドレスが固定されていれば、アドレスの設定は不要です。
2. USB 制御 IC がメモリコマンドを発行します。コマンド内容は、[b: 01000010]です。FPGA は①のタイミングでコマンドを解釈します。
3. ②のタイミング(State0)で FPGA がウェイト要求しているかどうか確認します。FPGA が出力する WAITn 信号をこのタイミングで High レベルにすれば、USB 制御 IC はウェイトしません。Low レベルにすると、USB 制御 IC は WRn、RDn、RGDTn の制御線をアクティブにしません。

(注)①と②のタイミングは規定していません。通常数 ms 以下です。

4. FPGA がウェイト要求していない場合、USB 制御 IC は③のタイミング(State1)で WRn、RGDTn 信号を 1 クロック分 Low レベルに変化させ、FD[15:0]データバス上にメモリ転送データ(DATA0)を出力します。FPGA は④のタイミング(State3)でデータを取り込みます。

この様に、基本の書き込みサイクルは 3 ステート(3 クロック)構成です。WAITn 制御を行わない場合、このサイクルで連続してデータを転送します。

5. FPGA 側で USB 制御 IC からデータを受け取る準備ができていない場合、⑤のタイミングで WAITn 信号が Low レベルであることを確認します。次のクロック⑤'のタイミングでも Low レベルなので、State0 が挿入され、State0 が 2 クロック分延長しています。⑤"のタイミングで WAITn 信号が High レベルになっているので、次ステートの State1 に移行し、State2 で FPGA がデータを受け取ります。
6. すべてのデータ転送が完了すると、USB 制御 IC は 2 項と同様にメモリコマンドを発行し、データ転送処理を完了します。コマンド内容は、「b:01000000」です。FPGA 回路では、このコマンドを無視することもできます。

## 2.4 メモリ・リード

(ボードから PC へのデータ転送)

メモリ・リード時の手順を示します。図 5 のタイミングチャートを参照してください。

1. メモリデータ転送の前に、USB 制御アプリケーションでメモリの開始アドレスと転送するデータ量(バイト)を決めます。メモリの開始アドレスが固定されていれば、アドレスの設定は不要です。
2. FPGA 内では、RDn 信号をイネーブルとして IFCLK クロック数をカウントすることで、データ転送量を計数することができます。これにより、正しいデータ転送量を把握することができます。(注:データ読み出し時の注意点を参照してください)
3. USB 制御 IC がメモリコマンドを発行します。コマンド内容は、[b: 01000011]です。FPGA は①のタイミングでコマンドを解釈します。
4. ②のタイミング(State0)で FPGA がウェイト要求しているかどうか確認します。FPGA が出力する WAITn 信号をこのタイミングで High レベルにすれば、USB 制御 IC はウェイトしません。Low レベルにすると、RDn、RGDTn の制御線をアクティブにしません。

(注)①と②のタイミングは規定していません。通常数 ms 以下です。

5. USB 制御 IC は③のタイミング(State1)で RDn、RGDTn 信号を Low レベルにします。FPGA では、State1 の状態を検出し、④のタイミング(State3)で FD[15:0] データバス上にメモリ転送データを出します。USB 制御 IC は⑤のタイミングでデータを取り込みます。  
これで 1 ワードのデータを転送する基本サイクルが終了したことになります。WAITn 信号がアクティブになっていなければ、このサイクルを繰り返し実行してデータ転送します。
6. ⑤のタイミングで USB 制御 IC がデータを取り込むのと同時に、次のサイクルの WAITn 制御信号を検出します。図 5 では、WAITn 信号が Low レベルになっているので、1 クロック後の⑥のタイミングで WAITn 信号レベルを検出します。この場合、High レベルなので、State1 に移行します。後は基本サイクルと同様です。  
この例では、1 クロック分 WAITn 制御を行ったことになります。
7. すべてのデータ転送が完了すると、USB 制御 IC は 2 項と同様にメモリコマンドを発行し、データ転送処理を完了します。コマンド内容は、「b:01000001」です。FPGA 回路では、このコマンドを無視することもできます。

## < データを読み出す場合の注意点 >

USB 制御 IC は、制御アプリケーション上で設定した設定値(転送バイト数)よりも 2,048 バイト多くボードからデータを読み出します。

これは、USB 制御 IC が FIFO 動作を基本に構成されているためです。

以下、ホスト PC からメモリリード操作が実行されたときの USB 制御 IC の詳細動作を説明します。

1. ホスト PC がメモリリードを実行。
2. USB 制御 IC がメモリリード開始コマンドをホスト PC から受け取る。
3. USB 制御 IC はメモリリード用に割り当てられた 2,048 バイト(512 バイトの 4 面バッファ)分のデータを FPGA から読み出す。
4. USB 制御 IC がホスト PC にデータを送信すると、バッファに空きができるので、USB 制御 IC が FPGA に対してデータを読み出します。512 バイト単位です。
5. メモリリード関数(SUP64\_Data\_Read)で設定したバイト数を転送し終わると、数 ms 後にホスト PC が USB 制御 IC にメモリリード終了コマンドを発行します。この

コマンドにより、USB 制御 IC のバッファがクリアされてデータ転送が終了します。

このとき、設定したバイト数のデータ転送が完了すると、USB 制御 IC のバッファは空になります。この時点で USB 制御 IC はデータ転送が完了したかどうかわかりません。このため、USB 制御 IC は FPGA から 2,048 バイト分のデータを設定バイト数以外に読み出します。設定バイト数よりも 2K バイト余計に読み出してしまい、メモリ終了コマンドにより、このデータがクリアされてしまいます。

例えば、1M バイトのメモリリードを行う場合、実際に読み出されるデータは 1M+2K バイト分のデータになります。

### 問題点:

FPGA に構築した回路構成が、FIFO からデータを読み出して、USB 制御 IC に転送するような方式やメモリリード後にクリアされるメモリ構成の場合、意図するデータに加えて 2K バイトの余分なデータが読み出されるため、データが消失し、システムの誤動作を招きます。

### 対策:

FPGA 側でデータ転送量を管理することでこの問題点を回避することができます。IFCLK をカウンタクロックとし、RDn 信号をイネーブルとしてデータ数をカウントすることで、最後の 2K バイト分の読み出し要求を無視することができます。

### 【サンプル回路】

このアプリケーションノートで解説する内容をベースにしたサンプル回路を提供しています。

SX-Card3 システム開発ボード用:

[http://www.prime-sys.co.jp/Download/SX\\_Card3](http://www.prime-sys.co.jp/Download/SX_Card3)

CX-Card2 システム開発ボード用:

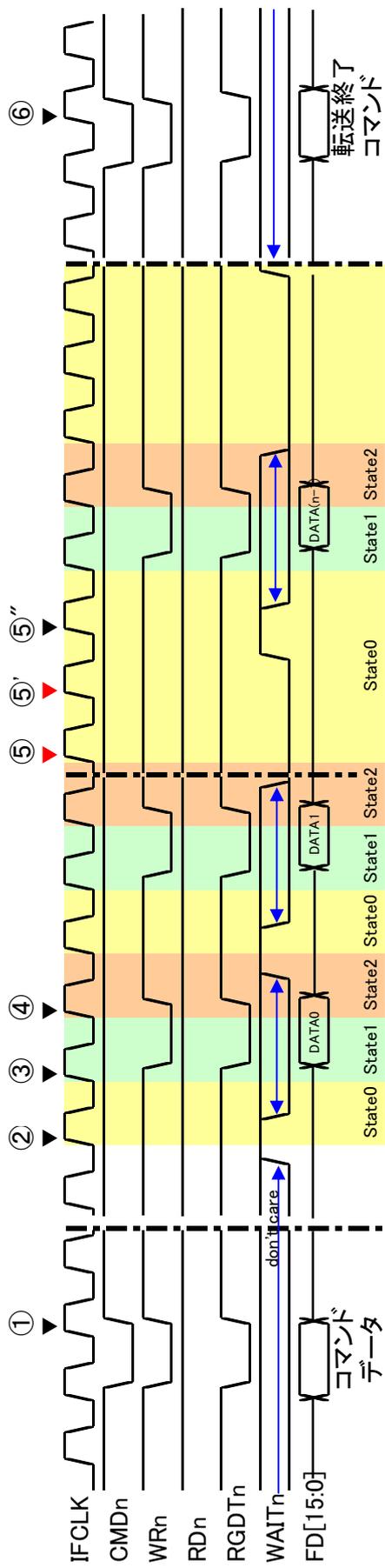
[http://www.prime-sys.co.jp/Download/CX\\_Card2](http://www.prime-sys.co.jp/Download/CX_Card2)

SX-Card6 システム開発ボード用:

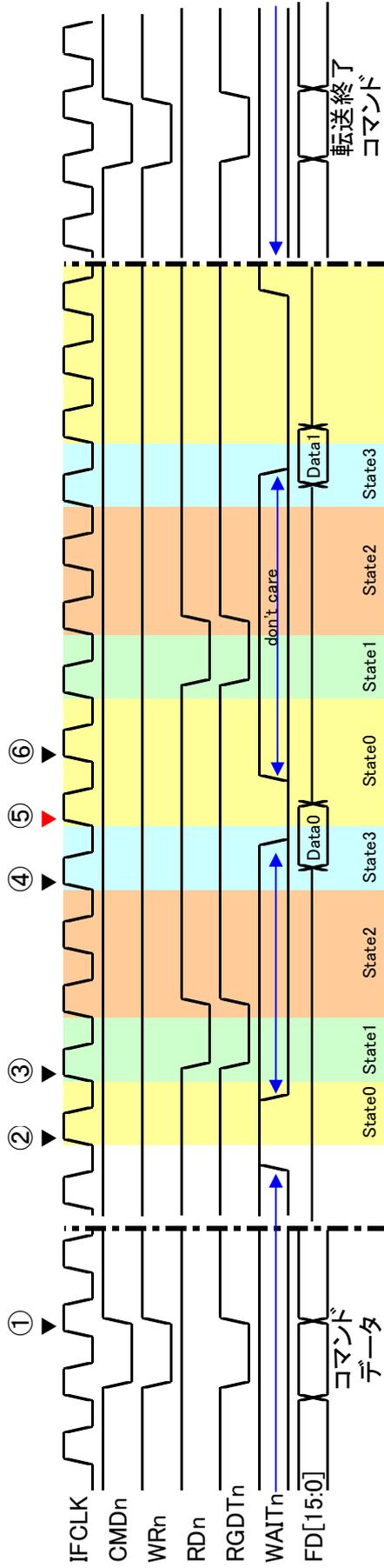
[http://www.prime-sys.co.jp/Download/SX\\_Card6](http://www.prime-sys.co.jp/Download/SX_Card6)

サンプル回路の内容:

FPGA 内にデバッグカウンタを実装しています。PC からデバッグカウンタのカウント値を読み出し、そのファイルを再度ボードに書き込むことで、ボード上の LED を点灯させています。制御アプリケーション RefApp7.exe の「メモリ操作」を利用して動作確認ができます。図 2 のメモリ操作画面でメモリ転送モードを Word に切り換えることを忘れずに行ってください。



〈図4. メモリ転送(PC→ボード:メモリ・ライト)〉



〈図5. メモリ転送(PC←ボード:メモリ・リード)〉