Smart-USB Plus 製品対応

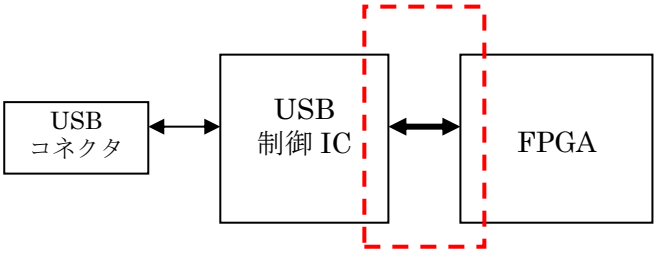
USB インタフェース・プロトコルの概要

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 とインタフェースを実現するための FPGA 回路について解説します。図 1 に示す破線で囲んだ部分です。また、製品に添付する CD-ROM 内に収録する mnl_Smart-USB_Plus.pdf を併せてご覧ください。

【適用ボード】

- Smart-USB Plus 製品ファミリのボード



<図 1. Smart-USB Plus 製品の基本構成>

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。 アクセス種別は 2 種類あり、FPGA 内に実装したハードウエア・レジスタへの書き込み、読み出しをする「レジスタ・アクセス」とメモリなどに蓄えたデータを読み出したり、メモリに書き込む動作をする「メモリ・アクセス」があります。

以下の表に記載しない未使用ピンとして 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。

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

2.1 信号線の概要

信号名 ：IFCLK
信号方向 ：USB 制御 IC←48MHzOSC→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。ただし、512 バイト単位です。

3. データ転送プロトコル

取り扱うことのできるデータビット幅は、8 / 16 / 32 / 64 bit の 4 種類です。それぞれ、USB データバスの FD[15:0] 信号線上に出力します。物理的なデータ線が 16 本なので、16bit を超えるデータは 16bit 単位に分割して転送します。

3.1 レジスタ・アクセス

3.1.1 アクセス概要

各ビット長でアクセスを行うレジスタ・アクセス手順を示します。このレジスタ・アクセスは、USB プロトコルで定義する「コントロール転送」を利用しています。

【対応する API（USB プロトコルを実現する専用関数）】
（使用するデバイスドライバが PsiCeb2.sys の場合）
- **CebRegisterWrite**
（レジスタへの書き込み動作）
- **CebRegisterRead**
（レジスタからの読み出し動作）

（使用するデバイスドライバが WinUSB.sys の場合）
- **SUP64_Reg_Write**
（レジスタへの書き込み動作）
- **SUP64_Reg_Read**
（レジスタからの読み出し動作）

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、以下のアクセスを行います。
USB 制御 IC は、FPGA にコマンド・データを送出し、同時に、このコマンドがレジスタ・アクセスであることを FPGA に表示します。コマンド・データはレジスタ No.（レジスタ・アドレス）を含みます。次に、レジスタ・ライトならばコマンド・データで表示したレジスタに書き込みたいデータを WRn 信号とともに FPGA に出力します。レジスタ・リードなら、FPGA に RDn 信号を出力して、FPGA が RDn に同期したデータを USB 制御 IC に出力します。
レジスタ No.設定は 0～16,383 番まで設定できます。詳細は、SUA011　レジスタ・アクセスの拡張機能をご覧下さい。

＊8bit データ・アクセスの場合

コマンド
　＋ データ [15:0] 上位 8bit は無視

＊16bit データ・アクセスの場合

コマンド
　＋ データ [15:0]

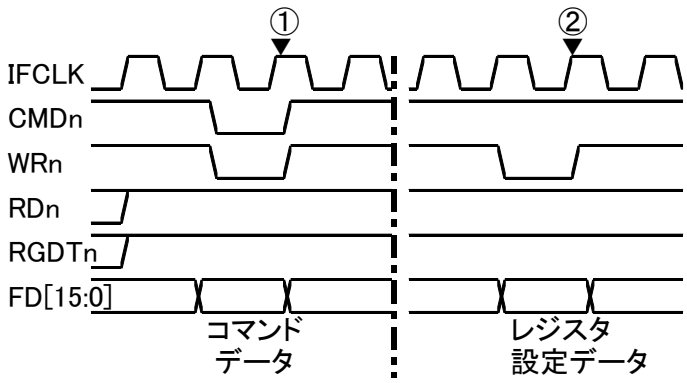
＊32bit データ・アクセスの場合

コマンド
　＋ データ [15:0]
　　＋ データ [31:16]

＊64bit データ・アクセスの場合

コマンド
　＋ データ [15:0]
　　＋ データ [31:16]
　　　＋ データ [47:32]
　　　　＋ データ [63:48]

3.1.2 アクセス・タイミング



<図 2. 基本レジスタ・アクセス・タイミング>

具体例 1：FPGA に構成したハードウエア・レジスタ（8bit）に任意の値を書き込む

USB 制御アプリケーションが専用 API の CebRegisterWrite または SUP64_Reg_Write を発行すると、USB 制御 IC は各制御信号をアクティブにし、USB データバス（FD[15:0]）上にデータを出力します。FPGA 側では図 2 に示す①のクロックエッジで FD[15:0]のデータを取り込みます。この値がレジスタのアドレス（番号）であり、コマンド・データになります。コマンド・データにはビット幅を示すアクセス種別と、レジスタ番号を含みます。FPGA ではこのアクセス種別でビット幅を判断し、レジスタ番号をデコードすることで、所望のビット幅でレジスタにアクセスすることができます。図 2 では、レジスタのビット幅が 8bit または 16bit の場合のタイミングを示します。

【コマンド・データの内容】

Bit	7	6	5	4	3	2	1	0
内容	アクセス種別		レジスタ No.指定（63～0）					

アクセス種別

00: 8bit アクセス、 01: 16bit アクセス

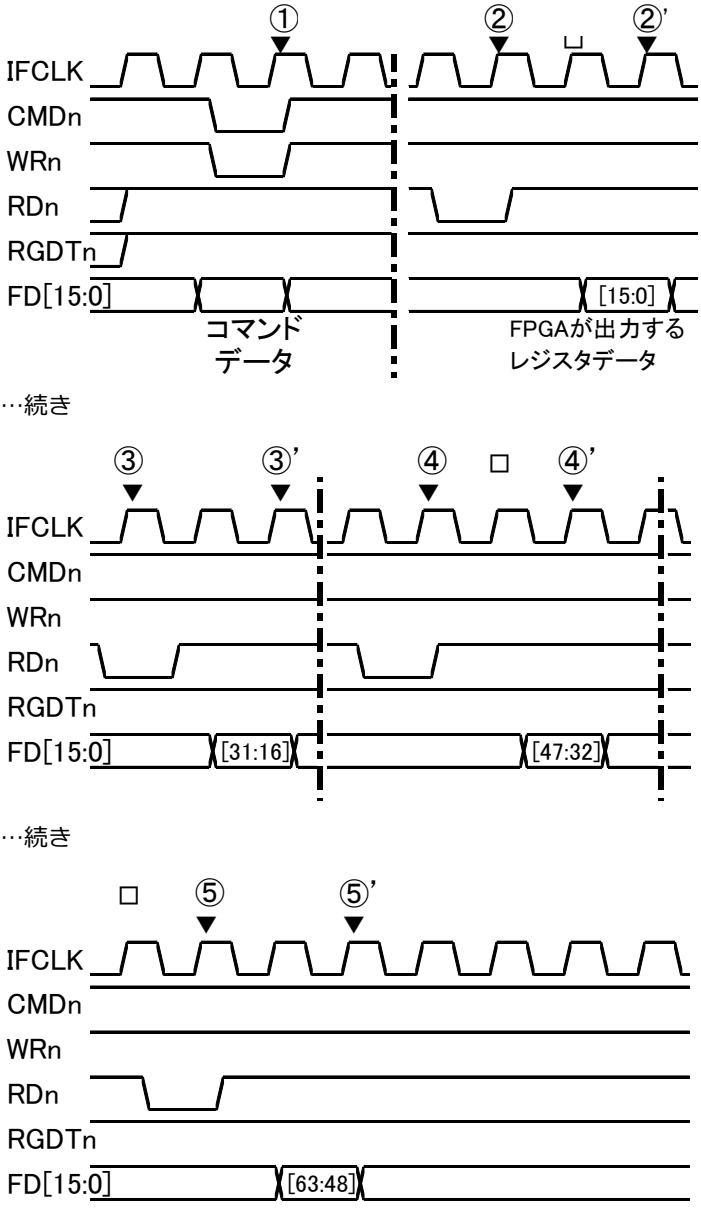
10: 32bit アクセス、 11: 64bit アクセス

（注:コマンド･データの上位 8bit もアドレス設定できます。SUA012 レジスタ・アクセスの拡張機能をご覧下さい）

FPGA はコマンド・データを受信後、②のタイミングで FPGA 内のレジスタに設定する任意のデータを受信します。WRn 信号を信号取り込みの条件として、FD[15:0]バス上のデータを FPGA 内に設定したレジスタに書き込みます。

32bit や 64bit アクセスの場合も同様に、USB 制御 IC からコマンド・データに続き、レジスタに設定するデータが 2 度または 4 度にわたり FPGA に出力されます。コマンド・データとレジスタ設定データの時間間隔は規定していませんが、通常数 5～8us の時間間隔があります。また、32bit/64bit アクセス時のレジスタ設定データの時間間隔も規定していません。通常、6us 以下です。

具体例 2：FPGA に構成したハードウエア・レジスタ（64bit）からデータを読み出す



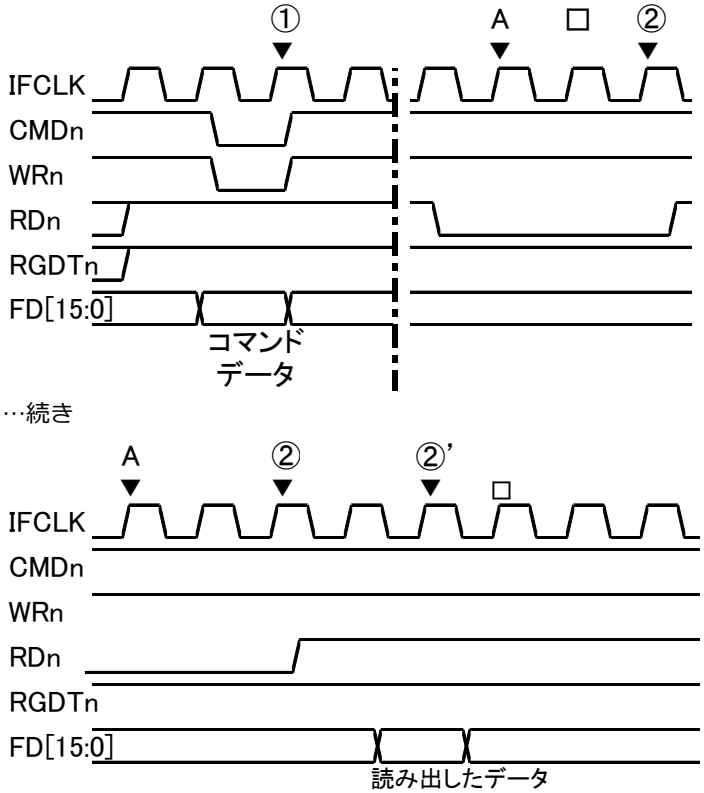
<図 3. 64bit レジスタ・アクセス（Rd）タイミング>

USB 制御アプリケーションが API の CebRegisterRead または SUP64_Reg_Read を発行すると、USB 制御 IC は各制御信号をアクティブにします。FPGA 側では①のタイミングで FD[15:0]バス上のデータをコマンド・データとして受信します。ここまでは、レジスタ・ライトと同じ動作です。FPGA ではレジスタ No.をデコードし、データを出力する準備が整います。

コマンド解釈終了後、②のタイミングで FPGA が RDn 信号を検出し、レジスタに保持しているデータを FD[15:0]バス上に出力します。2 クロック後の②'のタイミングで USB 制御 IC がデータを読み込みます。以下、③、④、⑤のタイミングで FPGA が FD[15:0]バス上にデータを出力し、それぞれ 2 クロック後の③'、④'、⑤'のタイミングで USB 制御 IC がデータを読み込みます。

FPGA が RDn 信号を検出してから、2 クロック後の立ち上がりエッジまでに FD[15:0]バス上にデータを出力できない場合、USB 制御 IC にデータを読み込ませるタイミングを遅らせることができます。

具体例 3：FPGA に構成したハードウエア・レジスタ（16bit）からデータを読み出す（2 クロック Wait）



<図 4. 16bit アクセス（Rd）タイミング、ウェイト有>

通常のレジスタリード・アクセスの場合、①のタイミングでコマンド解釈を実行後、「A」のタイミングで RDn 信号を FPGA が検出し、②のタイミングでレジスタデータを FD[15:0]バス上に出力します。しかし、レジスタ・アクセスを使った FIFO アクセスなどで、データ出力が②のタイミングに間に合わない場合もあります。その場合、USB 制御アプリケーションソフトの「GPIF 設定」画面から、ウェイト設定することでレジスタ・アクセスのタイミングを調整することができます。

図 4 のタイミングチャートでは、「GPIF 設定」画面でウェイト数を 3 に設定した場合です。RDn 信号が通常 1 クロック分の幅でアクティブになるところ、3 クロック分アクティブになっています。

FPGA 側では、「A」のタイミングで、FD[15:0]バスにデータを出力する準備に取りかかり、ウェイトしている間、データを用意して②'のタイミングで USB 制御 IC がデータを読み取ることができるようにします。

USB 制御アプリケーションソフトの「GPIF 設定」画面では、レジスタへの WR、RD アクセスに対して、0～255 までのクロック数の間、WRn/RDn 信号をそれぞれウェイトする機能があります。

3.2 メモリ・アクセス

3.2.1 アクセス概要

大量のデータを高速にデータ転送することのできる"USB バルク転送"を利用したデータ転送方式を示します。

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。

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 バルク転送を実現する専用関数）】

（使用するデバイスドライバが PsiCeb2.sys の場合）

＊ CebDataWrite
（メモリデータの書き込み動作）
＊ CebDataRead
（メモリデータの読み出し動作）

（使用するデバイスドライバが WinUSB.sys の場合）

＊ SUP64_Data_Write
（メモリデータの書き込み動作）
＊ SUP64_Data_Read
（メモリデータの読み出し動作）

【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。

3.2.2 メモリ・ライト

（PC からボードへのデータ転送）

メモリライト時の一般的な手順を示します。図 5、6 のタイミングチャートを参照してください。図 6 はデータ転送全体を表しています。WRn/RGDTn が Low アクティブになっている箇所が、512 バイト単位でデータ転送を行っている部分です。その一部を拡大したものが図 5 です。

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: 01xxxx10]です。FPGA は①のタイミングでコマンドを解釈します。
3. ②のタイミングで FPGA がウェイト要求しているかどうか確認します（ウェイトの詳細は後述の 3.2.4 項を参照してください）。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、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 1ms 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のタイミングで WRn、RGDTn 信号を 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を出力します。FPGA は④のタイミングからデータを取り込みます。
5. ④のタイミング以降、USB 制御 IC は 512 バイト単位でデータを扱うので、WRN,RGDTn 信号も IFCLK が 256 クロックごとに Low レベルに変化します。しかし、USB 制御 IC の内蔵メモリ状況により、最大で 2048 バイトの連続したアクセスが発生する場合があります。

6． 512 バイト単位で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。

3.2.3 メモリ・リード

（ボードから PC へのデータ転送）

メモリリード時の一般的な手順を示します。図 7 のタイミングチャートを参照してください。図 7-1 はデータ転送全体を表しています。RDn/RGDTn が Low アクティブになっている箇所が、512 バイト単位でデータ転送を行っている部分です。その一部を拡大したものが図 7 です。

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：データを読み出す場合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のタイミングで FPGA がウェイト要求しているかどうか確認します（ウェイトの詳細は 3.2.4 項を参照してください）。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、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 1ms 以下です。
5． FPGA がウェイト要求していない場合、USB 制御 IC は③のタイミングで RDn、RGDTn 信号を Low レベルにします。④のタイミングで FPGA がこの状態を検出し、⑤のタイミングで FD[15:0]データバス上にメモリ転送データを出力します。USB 制御 IC は⑥のタイミングでデータを取り込みます。
6． メモリライト時と同様に、⑥のタイミング以降 USB 制御 IC は 512 バイト単位でデータを扱うので、RDn、RGDTn 信号も IFCLK が 256 クロックごとに Low レベルに変化します。しかし、USB 制御 IC の内蔵メモリ状況により、最大で 4,096 バイトの連続したアクセス(1,024 クロック)が発生する場合があります。
7． 512 バイト単位で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。

3.2.4 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 512 バイトの FIFO を 4 面内蔵する構成のためです。
以下、ホスト PC からメモリリード関数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、内蔵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． メモリリード関数で設定したバイト数を転送し終わると、通常 1ms 以下のタイミングでホスト 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。
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。ホストからメモリリード終了コマンドを受信していないためです。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てしまいます。図 7-1 を参照してください。
2K バイト余計に読み出してしまったデータは、ホストから発行されたメモリリード終了コマンドにより、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。

P.10 以降の参考資料では、最小転送単位の 512 バイトを読み出したときの実際のタイミング波形を示しています。

問題点：
FPGA に実装した回路構成が、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。

3.2.5 メモリアクセス時の WAIT 制御

FPGA が USB 制御 IC に対して出力する WAITn 信号を使ってウエイト制御ができます。ホスト PC からボードへデータ転送する場合（メモリ･ライト）、ボードがホスト PC からのデータを受け取れない時には、WAIT 制御することでデータ転送を一時的に停止し、データを失うことなく正しくデータ転送することができます。ボードからホスト PC へのデータ転送（メモリ･リード）の場合も同様です。

USB 制御 IC は、図 8 で示す②のタイミングで WAITn 信号の Low/High レベルの検出を行います。High レベルの時、その後の WAITn 信号レベルは don't care です。また、WAIT するタイミングは、②で示すように、WRn/RDn/RGDTn の各信号レベルが High レベルの時だけです。すなわち、実際にデータ転送が行われている時に WAIT 制御はできません。512 バイト単位のデータ転送が終わった時点（図 6,7 の青矢印で示した区間）で、USB 制御 IC は FPGA からの WAIT 要求の有無を確認します。WAIT 要求がなければ、引き続きデータ転送を行います。

図 8-1 で示すように、WAITn=Low レベルの間、ホスト PC からボードに対してデータの読み出しが行われません。

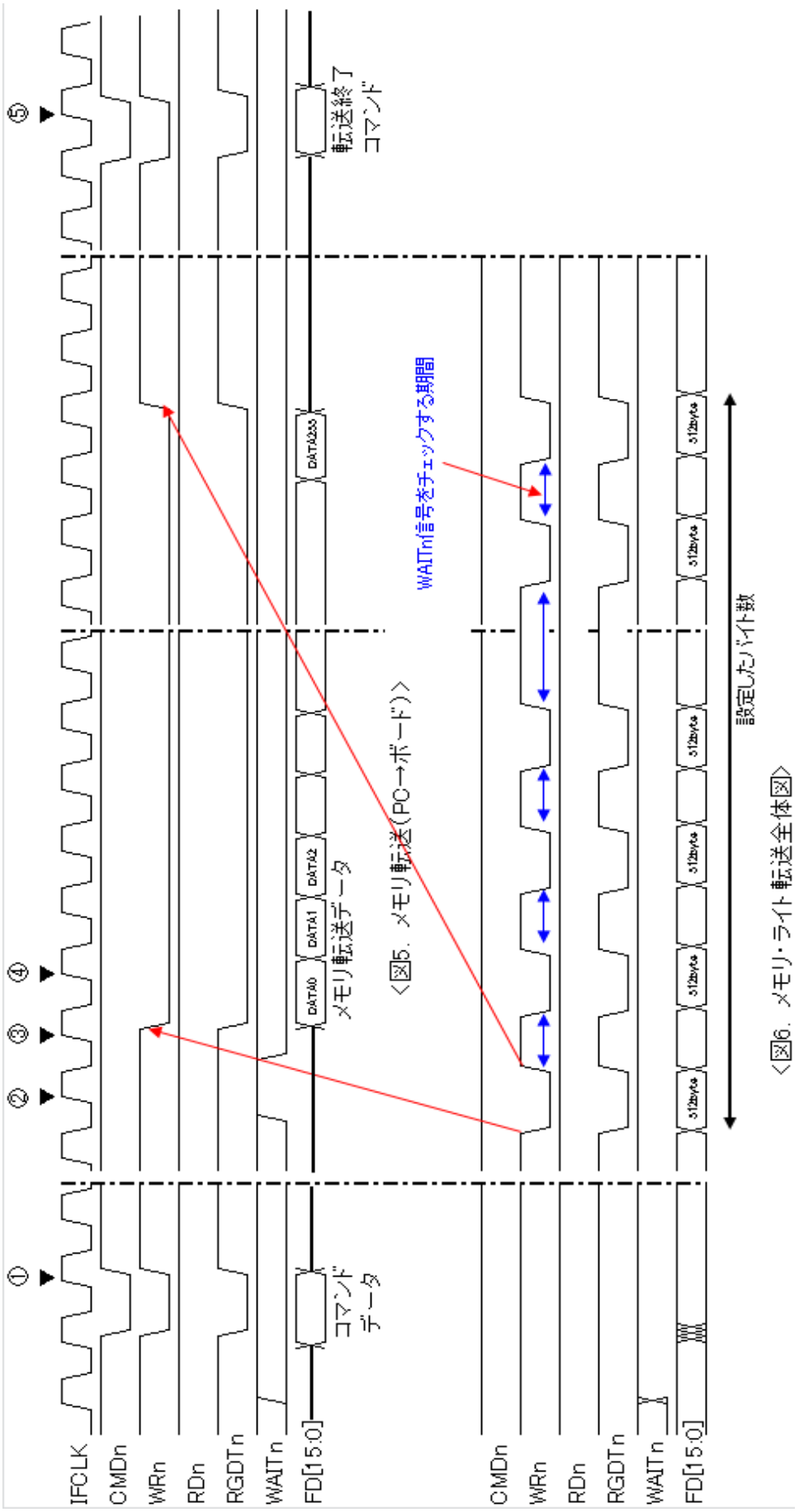
このように、WAITn 信号を利用した WAIT 制御は、ボード外に低速なメモリを接続している場合や、外部システムからのデータ転送速度が USB の実効データ転送速度よりも低速な場合に便利な機能です。

制御アプリケーションの RefApp2.exe および RefApp3.exe では、WAITn 信号を Low レベルにする時間の最大値を 10 秒に設定しています。10 秒を超えるとタイムアウトエラーになります。タイムアウトの設定時間は、API の CebSetTimeOut 関数で設定することができます。
WinUSB.sys デバイスドライバを使用する RefApp7.exe 制御アプリケーションには、このタイムアウトの設定がありません。

【WAIT 制御の注意点】
ホスト PC からボードに対して、2,048 バイト以下のデータ転送（書込）する場合、WAIT 制御には注意が必要です。 USB 制御 IC には 512 バイトのバッファが 4 面あり、このバッファを利用して FPGA とのデータ転送を行います。
転送データ量が 2,048 バイト以下の場合、ホスト PC はデータ転送がすべて完了したと判断します。このため、USB 制御 IC と FPGA 間で WAIT 制御していた場合、ホスト PC はすべてのデータ転送が完了している状態なのに、ボード側では WAIT 制御が続いている状態になる場合があります。
この状態で、ホスト PC は次のボード制御コマンドを発行できるので、ボード側で WAIT 制御中にもかかわらず、ホスト

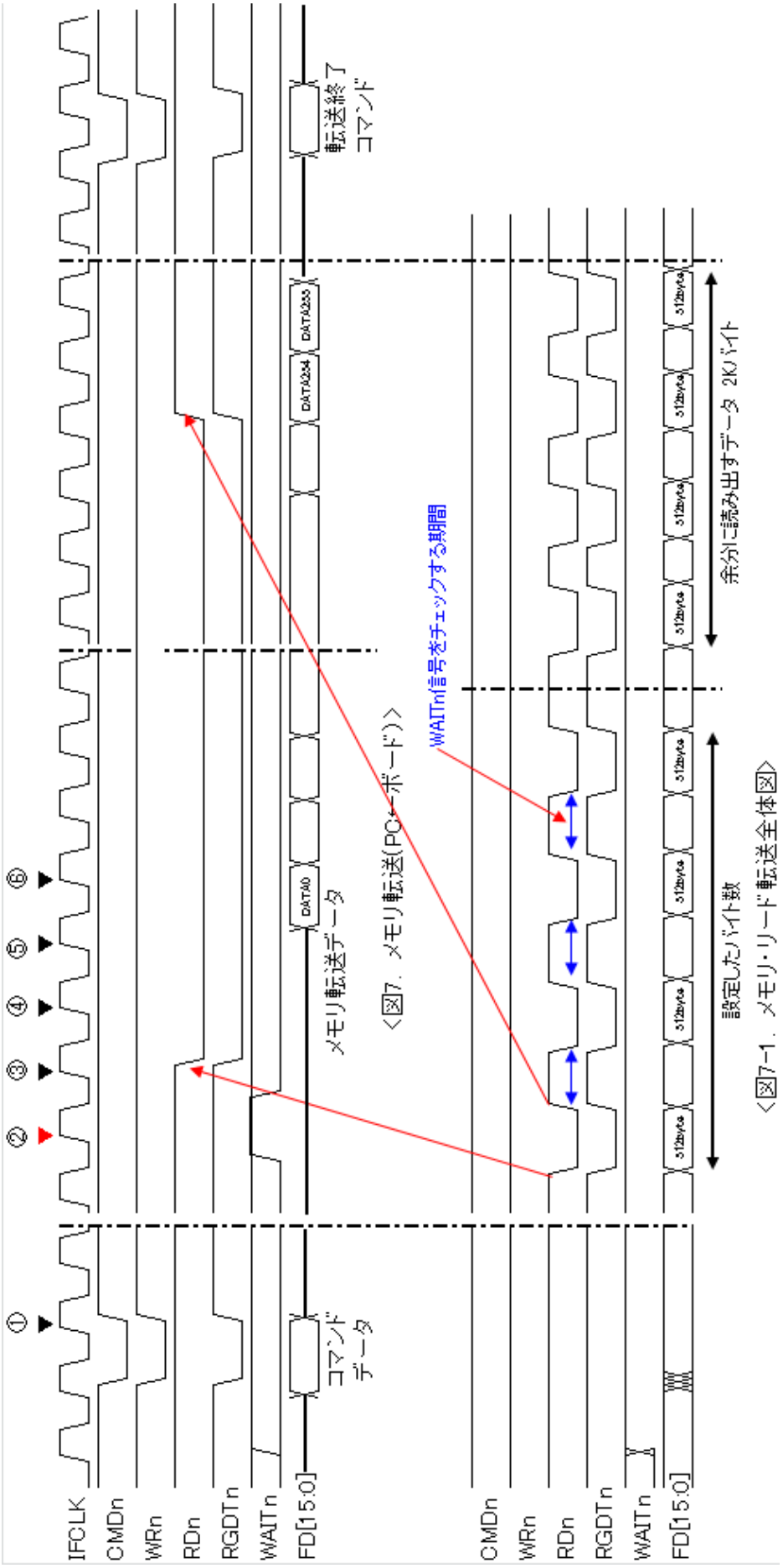
PC 側の制御ソフトがコマンドを発行してしまうと、ボード側の USB ファームウエアが誤動作する原因になります。

<対策>
実効データが 2,048 バイト以下の場合でも、データ転送量を 2,560 バイト（A00hex）に設定すれば、問題ありません。FPGA 回路では 2,560 バイト分のデータが送られてくる動作になりますが、実効データだけ受信できるような回路にすることで対処できます。

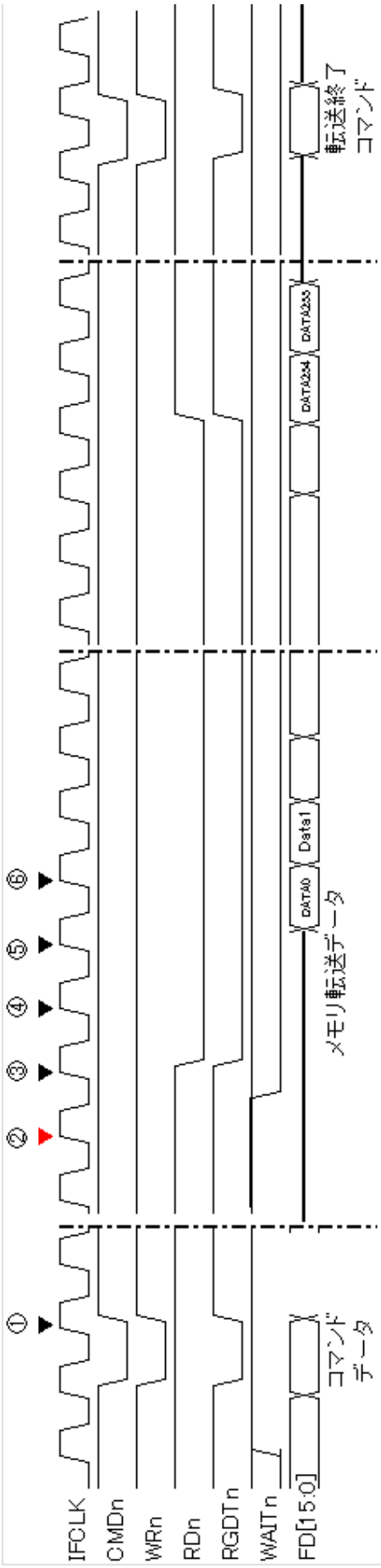


〈図5. メモリ転送(PC→ボード)〉

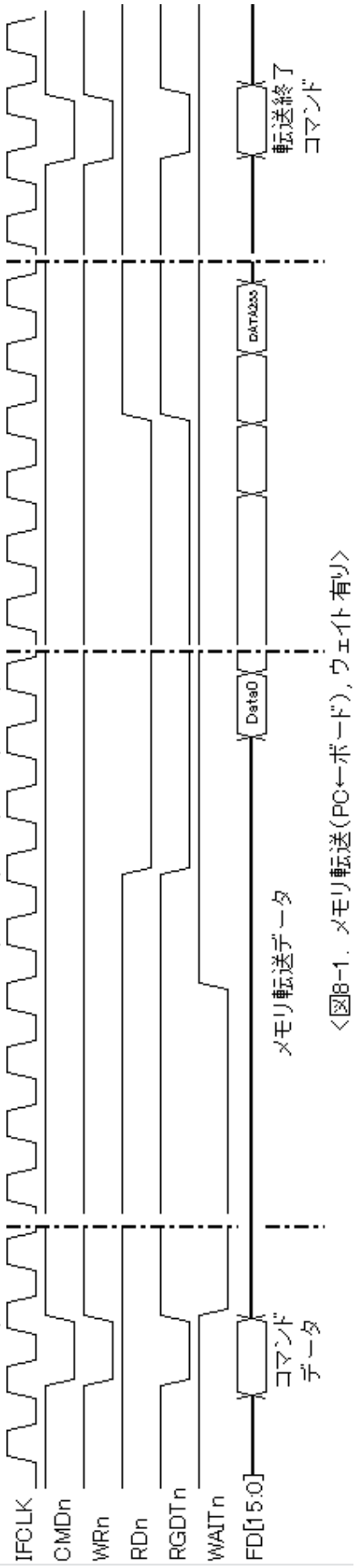
〈図6. メモリ・ライト転送全体図〉



〈図7-1. メモリ・リード転送全体図〉



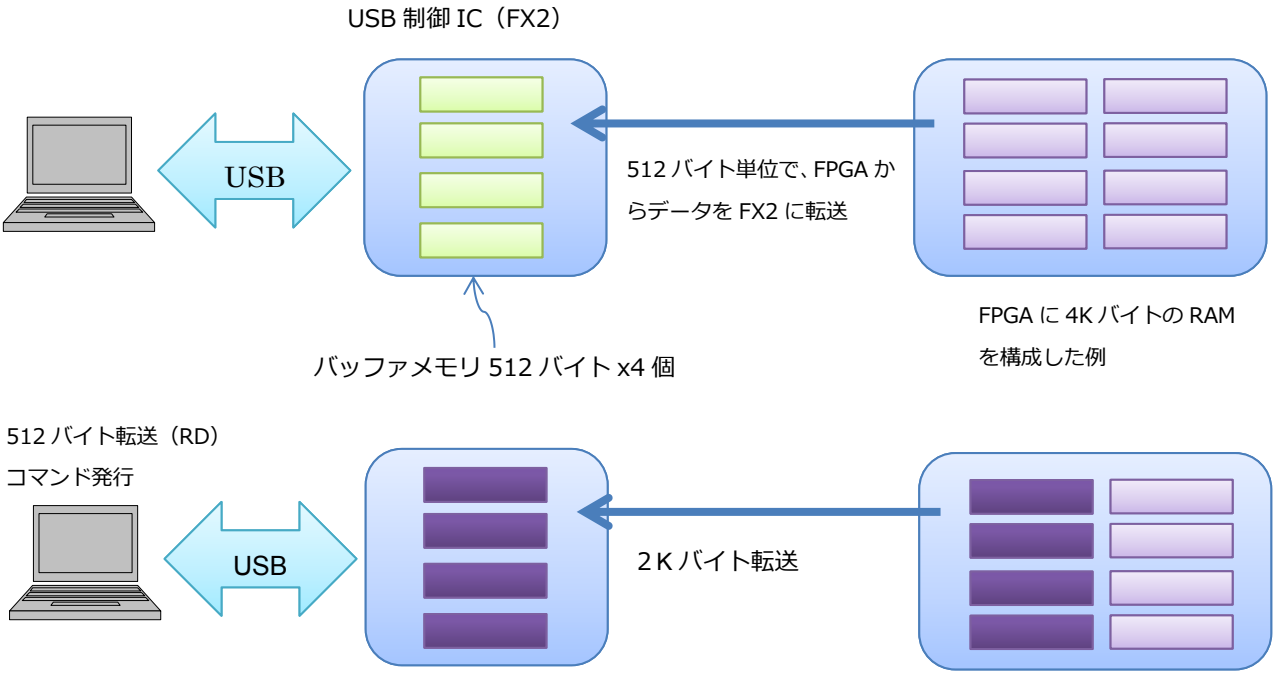
〈図8. メモリ転送(PC←ボード)、ウェイトなし〉



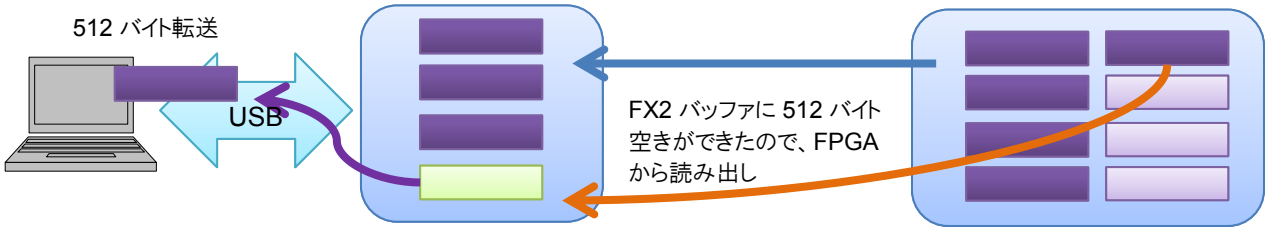
〈図8-1. メモリ転送(PC←ボード)、ウェイト有り〉

メモリリードの注意点

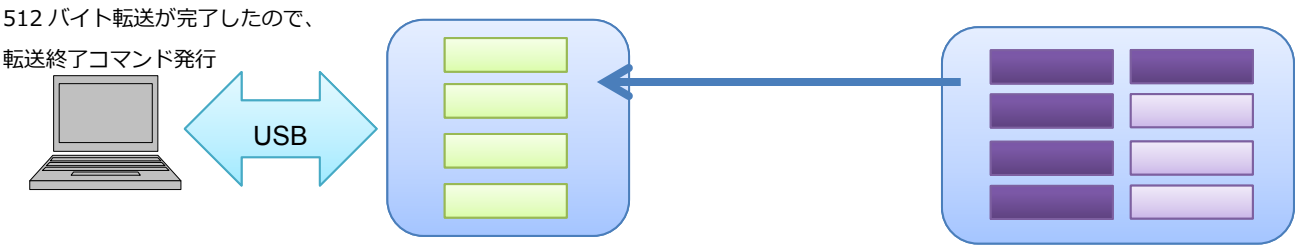
USB 制御 IC（以下 FX2）と FPGA 間のインタフェースでは、メモリリード時に注意すべき動作があります。
RD コマンド時に指定したデータ量(512 バイト単位)を転送した後に、ボード内では２K バイト余計に読み出します。



PC アプリから 512 バイト転送（RD）のコマンドを発行しましたが、ボード側では、FX2 内の２K バイトバッファに常にデータを溜めようとします。このため、実際には、FX2 は FPGA から 2K バイト読み出します。図 2.参照。

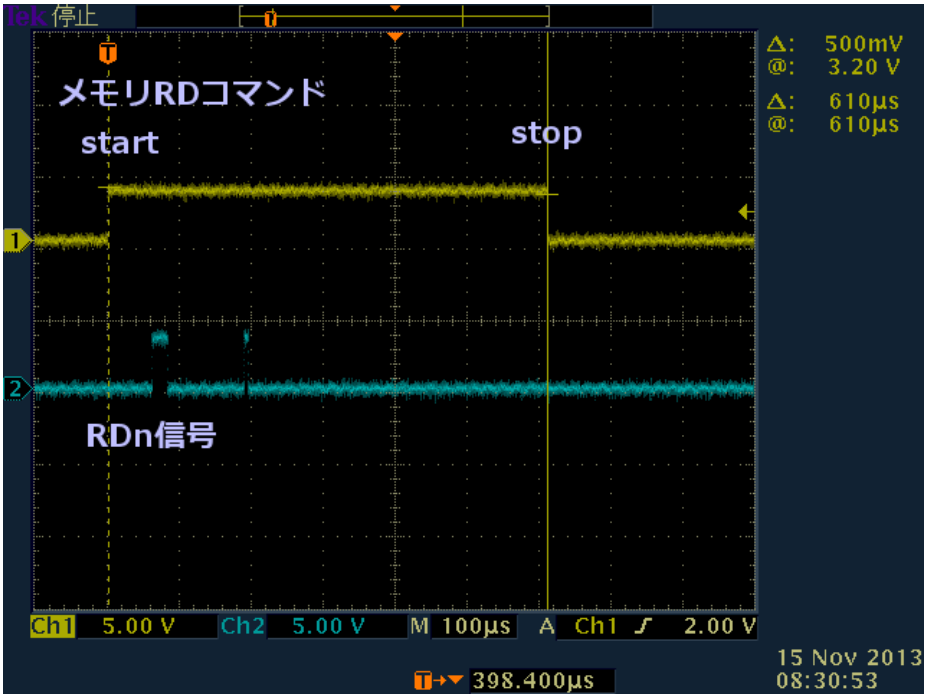


PC に 512 バイトのデータが転送されますが、FX2 のバッファに空きができたので、FX2 は FPGA からデータを読み出します。図 3 参照。



余分な２K バイトが FX2 に読み出されますが、PC アプリでは指定した 512 バイトデータの受信が完了したので、転送終了コマンドを発行します。FX2 はこのコマンドを受信して、バッファ内のデータをクリアします。
FPGA では、指定した転送量に加えて２K バイトのデータが余分に読み出されることになります。

<図 9. メモリリード時のデータの流れ>

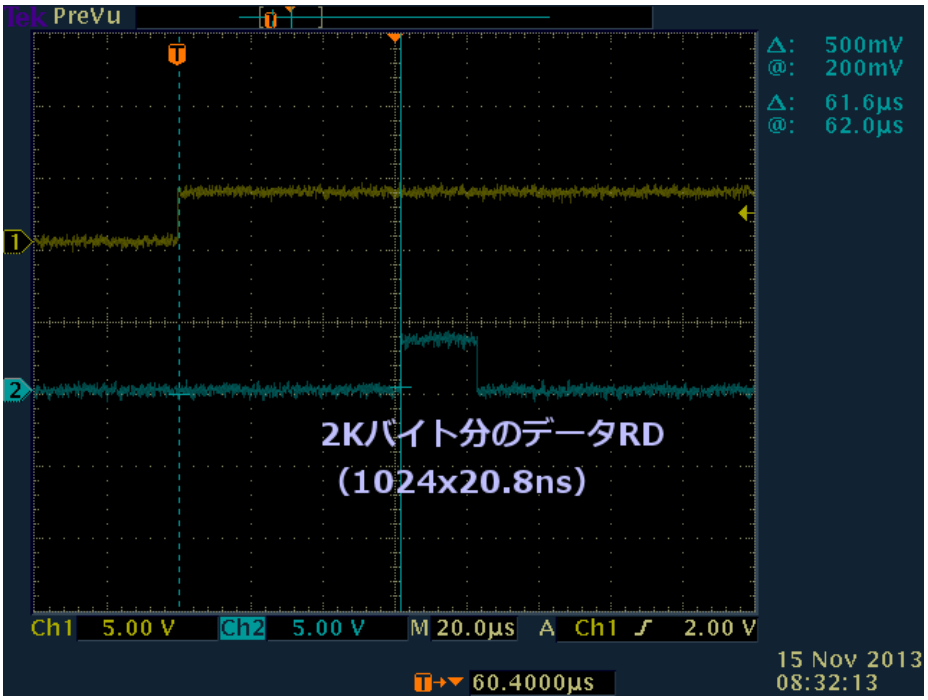


Ch1（黄色）：メモリ RD 転送開始コマンドと終了コマンドの時間間隔を表します。
転送開始コマンドでトリガしています。

Ch2（水色）：制御線　RD n （観測ピンでは反転出力）

512 バイトの転送を行った状態の例です。1 回のメモリ転送に約 610us 要していますが、一定の値ではありません。
比較的短い場合のタイミングです。

図 10．512 バイトのデータ転送（ボード→PC）　全体タイミング



転送開始コマンドから、実際に RDn 信号が出力されるまでの時間間隔は、約 62us です。

USB 制御 IC の 2K バッファが空なので、最初に 2K バイト分一気に読み出します。
16bit 幅のデータ線なので、1 クロック(20.8ns)で 2 バイト転送します。
このため、2K バイトのデータ転送時間は 1024x20.8ns≒21.3us になります。

図 11．FX2 が最初に読み出す 2 K バイトデータ・タイミング

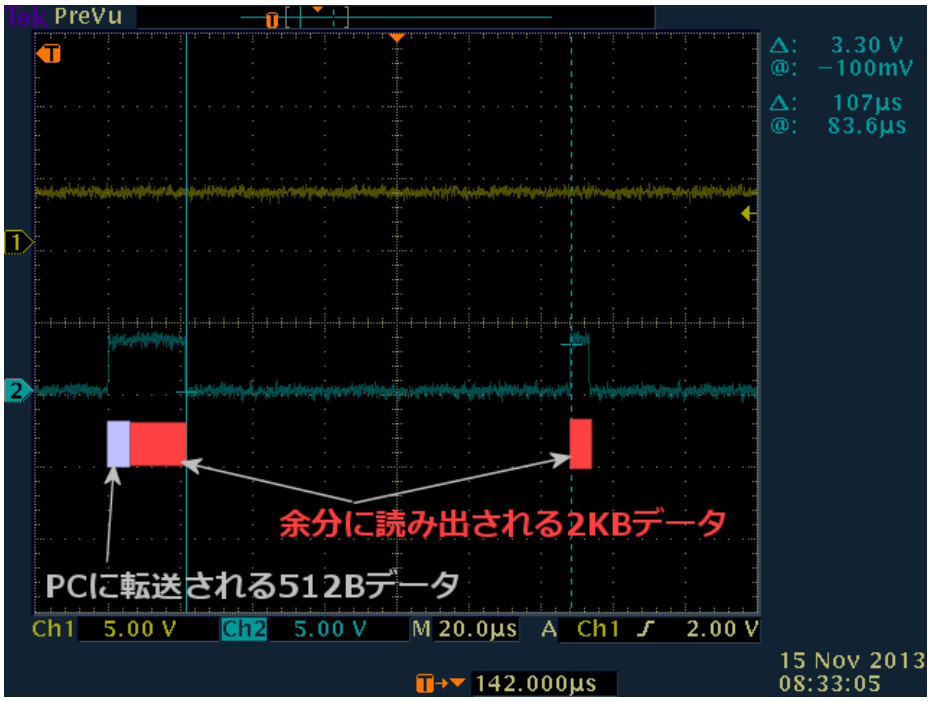


図 12. PC が 512 バイト読み出した後に、FX2 に転送される 2K バイトデータ

USB 制御 IC のバッファに 2K バイト読み込んだ後、実際に PC に転送された 512B 分のデータ領域が空きます。 このため、512B 分を追加で読み出します（ファームウエアの動作仕様です）。

左図赤いブロック部分が FPGA から読み出してしまう余分なデータです。 1536＋512=2,048 バイト。

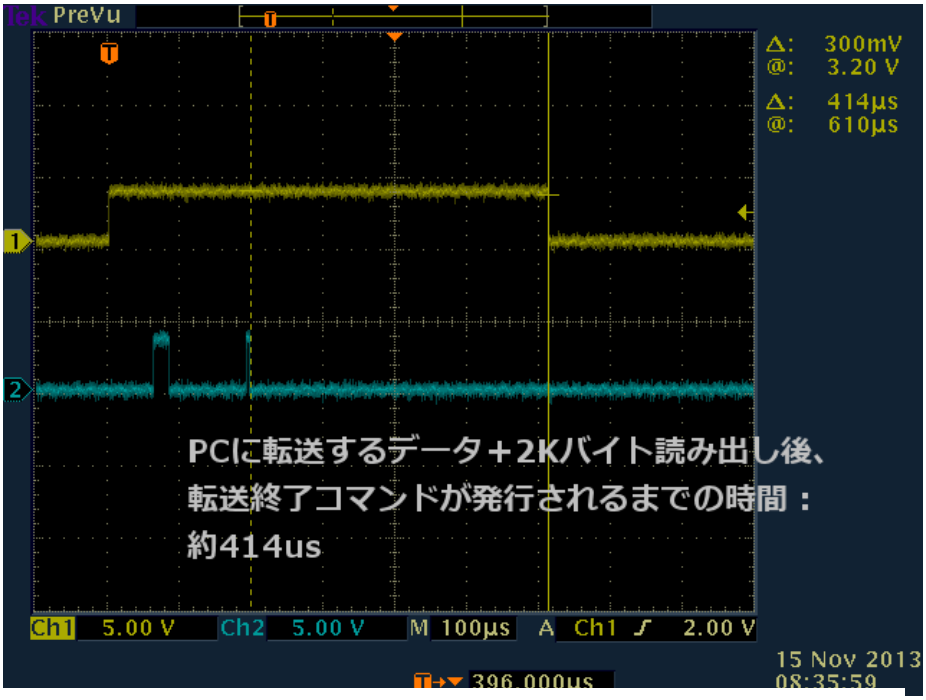


図 13. PC がボードから 512 バイ読み出す全体タイミング

余分なデータ 2K を読み出し後、転送終了コマンドが発行されるまでの時間間隔は、約 414us です。
この時間は、実際に USB 制御 IC から PC に USB 経由でデータ転送され、ホスト PC は転送が完了したことを受けて、転送終了コマンドを発行するまでの時間です。このため、PC にデータ転送する時間がかかれば、転送終了コマンド発行タイミングも遅れます。

この例では、実際に転送した 512B データ転送終了後から、メモリ転送が完了（メモリ終了コマンド発行）するまで、約 542us になります。