

連続データ収集用 FPGA リファレンス・デザイン

1. データ収集システムの構築

Smart-USB Plus 製品^(注1)を利用して簡単に連続したデータ収集ができるソリューションを無償提供します。FPGA リファレンス・デザインと Windows 対応のリファレンス制御アプリケーション”RefApp7”を利用すると、短時間で FPGA 設計が完了し、USB 経由で PC に収集したデータをファイル化することができます。ファイル化後に、別途、解析ソフトウェア等を利用してデータ解析が可能になります。

(注1) アルテラ FPGA を搭載したボードに限ります

【システム構成例】

AD コンバータを搭載したボードを Smart-USB Plus 製品と接続し、AD 変換後のデータを USB 経由で PC に連続して収集することができます。ここで提供する FPGA リファレンス・デザインでは、AD 変換された 16bit パラレルデータをそのまま PC に取り込み、バイナリファイル(.bin)化します。

FPGA リファレンス・デザインをカスタマイズして、特定のデータパターンからデータを取り込んだり(トリガ機能)、データを処理してから PC にデータ転送することもできます。



<図 1. システム構成概要>

【FPGA リファレンス・デザインの概要】

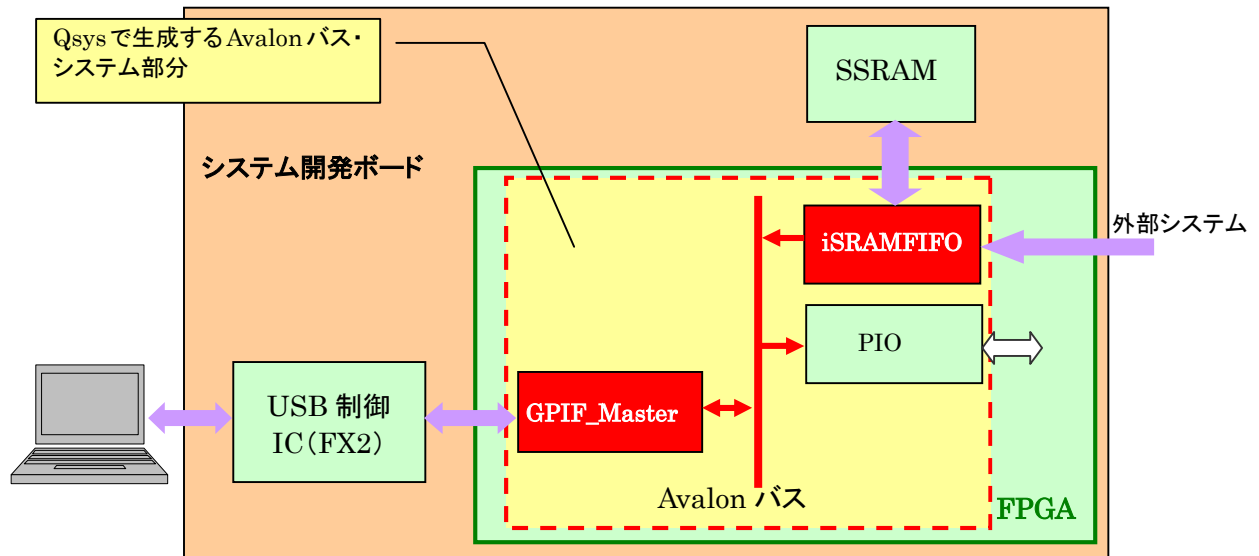
このリファレンス・デザインはアルテラ社 FPGA を搭載した Smart-USB Plus 製品向けです。開発ツール Quartus2(以下、Q2)に含まれる Qsys システム統合ツールを利用し、HDL コード記述を最小限に抑えることで、短時間にデータ収集システムを構築することができます。このため、FPGA 設計に不慣れなユーザでもデータ収集システム設計が可能です。

FPGA リファレンス・デザインを適用できる製品は、FPGA 周辺回路として同期 SRAM を搭載しているボードです。以下に該当するボード製品名を示します。

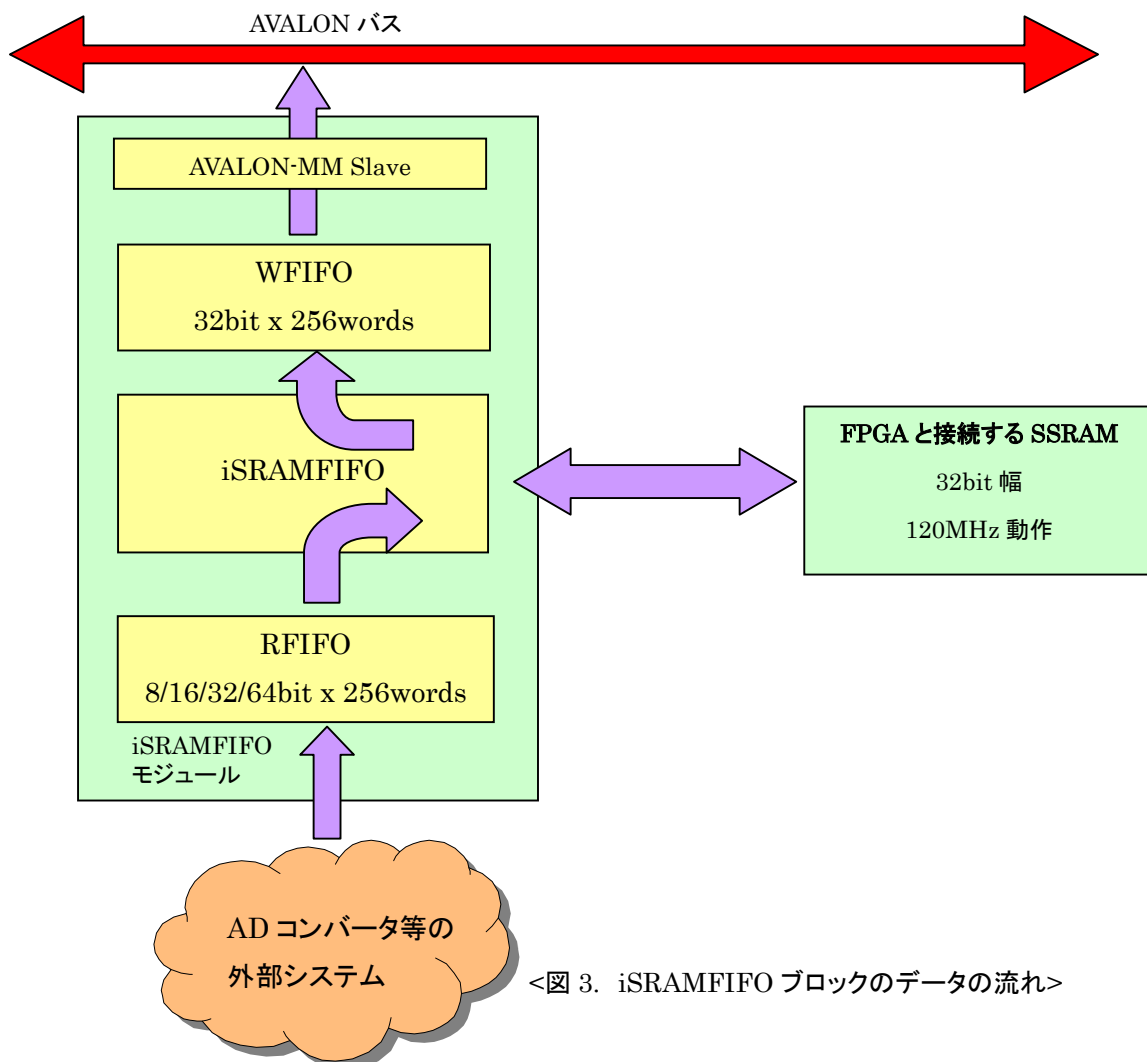
1. CX-USB2 システム開発ボード (Cyclone3-FPGA 搭載)
2. CX-Card2 システム開発ボード (Cyclone2-FPGA 搭載、IMG-Pro4 装着時)
3. SX-USB2 システム開発ボード (Stratix2-FPGA 搭載)
4. SX-USB3 システム開発ボード (Stratix3-FPGA 搭載)
5. System-SX システム開発ボード (Stratix-FPGA 搭載)
6. System-SX2 システム開発ボード (Stratix2-FPGA 搭載)

【FPGA リファレンス・デザインの構成】

FPGA に接続している同期 SRAM をデータバッファとして利用しながら、連続したデータを切れ目なく PC にデータ転送します。また、Qsys ツールを利用することで、FPGA の機能(図 2 のに示す赤色ブロック)については設計者が回路記述する必要がありません。回路記述が必要なのは FPGA のトップ回路(外部システムとの接続部分)と FPGA のピン設定(IO の電圧設定やピンアサイン)などです。



<図 2. システムブロック図>

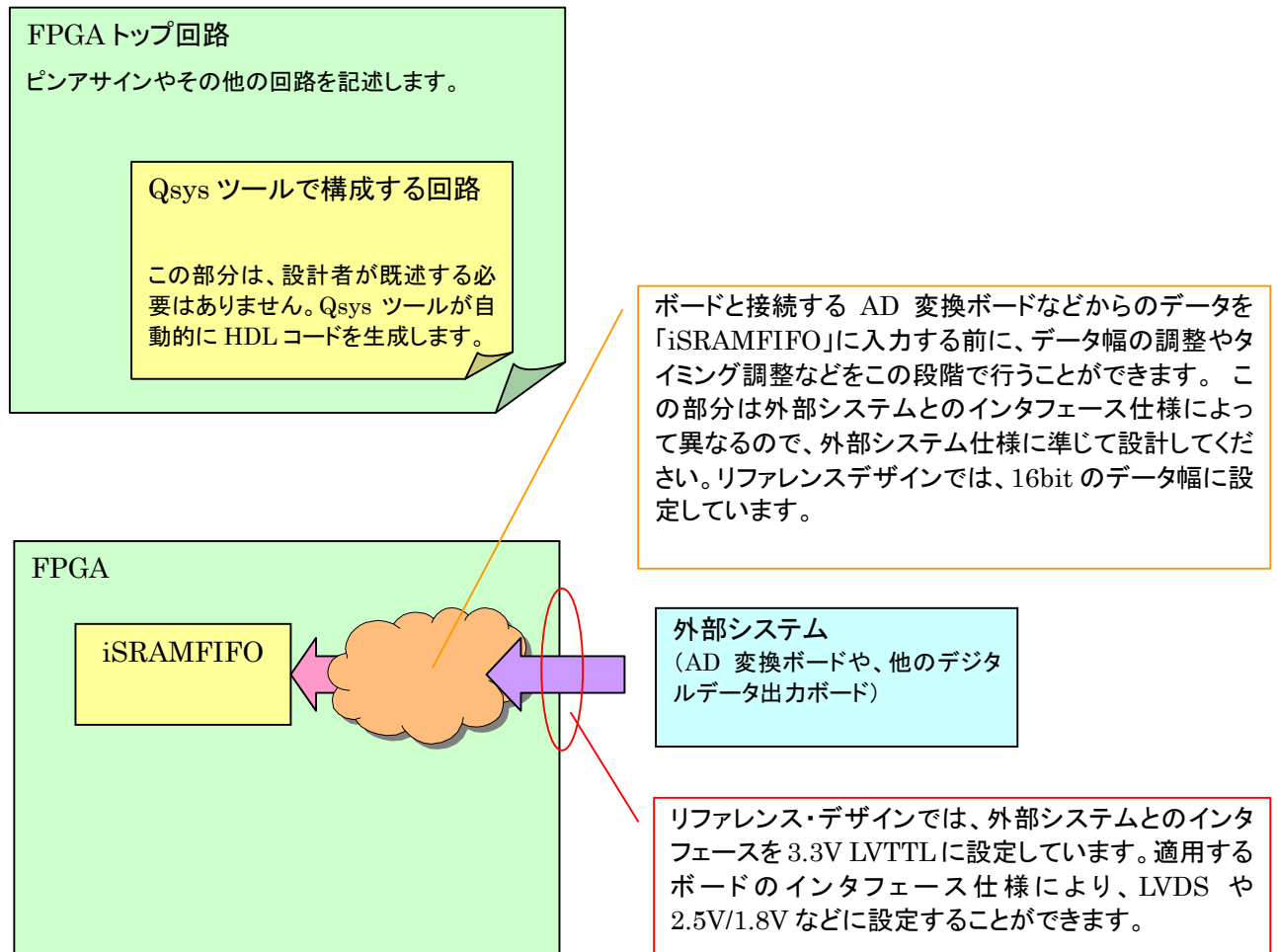


<図 3. iSRAMFIFO ブロックのデータの流れ>

2. 具体設計

2.1 FPGA 回路記述の概要

下図(図.4)のように、Qsys ツール上で構成する回路がデータ収集システムの主回路です。FPGA トップ回路では、ボード上の LED に状態表示させたり、図 2 で示す iSRAMFIFO ブロックにデータを入力する前に、データ処理をする回路を挿入することもできます。



<図 4. FPGA 回路記述の概要>

【FPGA リソース】

CX-USB2 システム開発ボードにリファレンス・デザインを適用した場合のリソース使用状況は以下のとおりです。

使用ロジックセル数	1,098/24,624 (4%)
使用内蔵メモリ(M9K)	14/66 個 (21%)

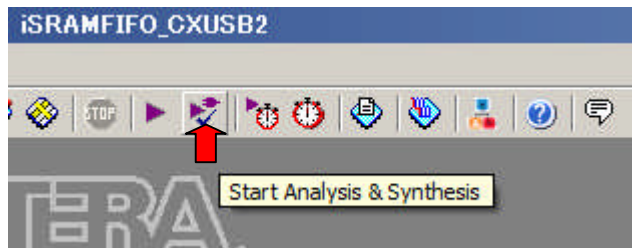
2.2 サンプル回路の入手

この資料で解説する CX-USB2 システム開発ボード用サンプル回路は、以下の URL からダウンロードできます。

http://www.prime-sys.co.jp/Download/GPIF_Avalon/iSRAMFIFO_CXUSB2_SUA010.zip (約 9MB)

ダウンロードするファイルは Quratus2 (以下、Q2) プロジェクトを丸ごと zip 形式のファイルに圧縮しています。使用した Q2 のバージョンは ver11.1SP2 です。

zip ファイルを解凍後、Q2 のツールバーにある、Start Analysis & Synthesis ボタンをクリックしてください。赤の矢印で指し示すボタンです。



Start Analysis & Synthesis が完了すると、Q2 の画面左側に見える Project Navigator にプロジェクトのファイル構成が表示されます。

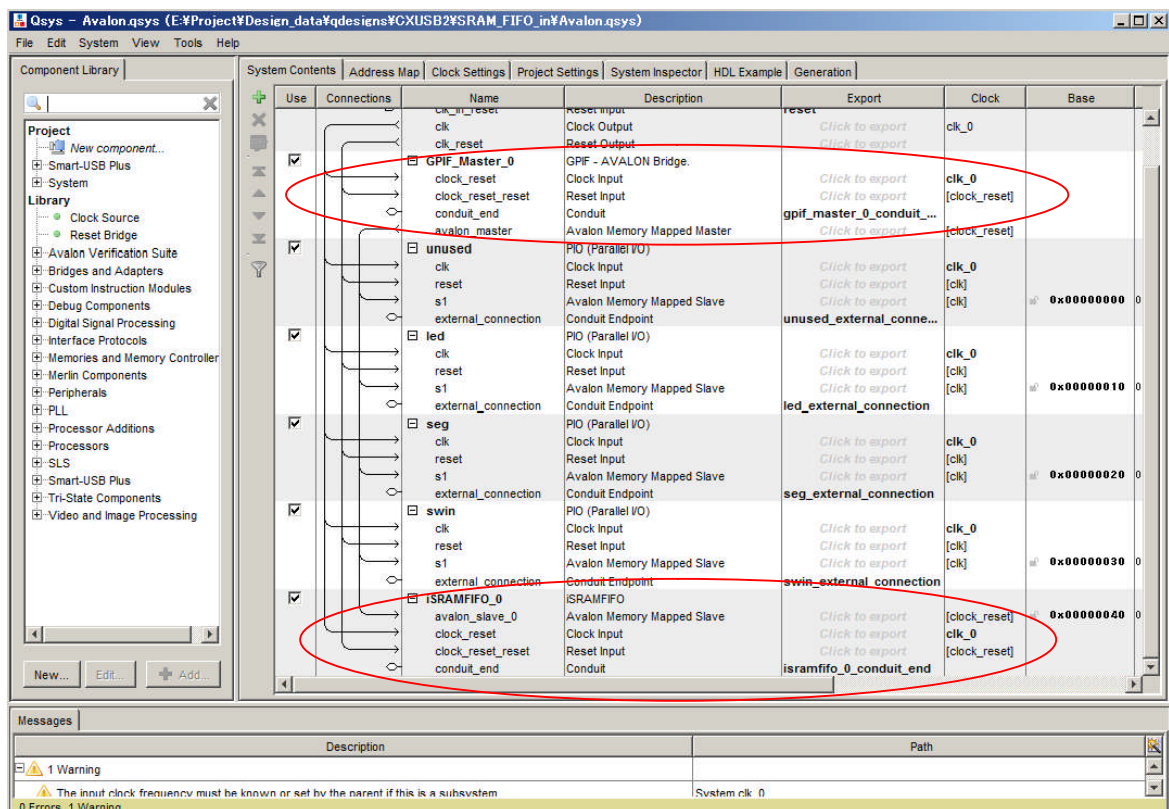
<図 5. Q2 のツールバー ボタン>

2.2 モジュールのファイル構成

Qsys ツールで用いる回路モジュールは、外部システムからのデータを取り込む iSRAMFIFO モジュールと、USB インタフェースをサポートする GPIF_Master モジュールで構成しています。これら 2 つのモジュールを Qsys ツール上で Avalon バスに接続します。リファレンス・デザインでは、すでにモジュールを接続した状態なので、新たにモジュールを追加する必要はありません。

Q2 ツールバーの“Tools”から“Qsys”を選択し、リファレンス・デザインのフォルダ内にある Avalon.qsys ファイルを選択してください。選択後、図 7 で示す画面が得られます。

赤枠で囲んだ部分が GPIF_Master モジュールと iSRAMFIFO モジュールです。



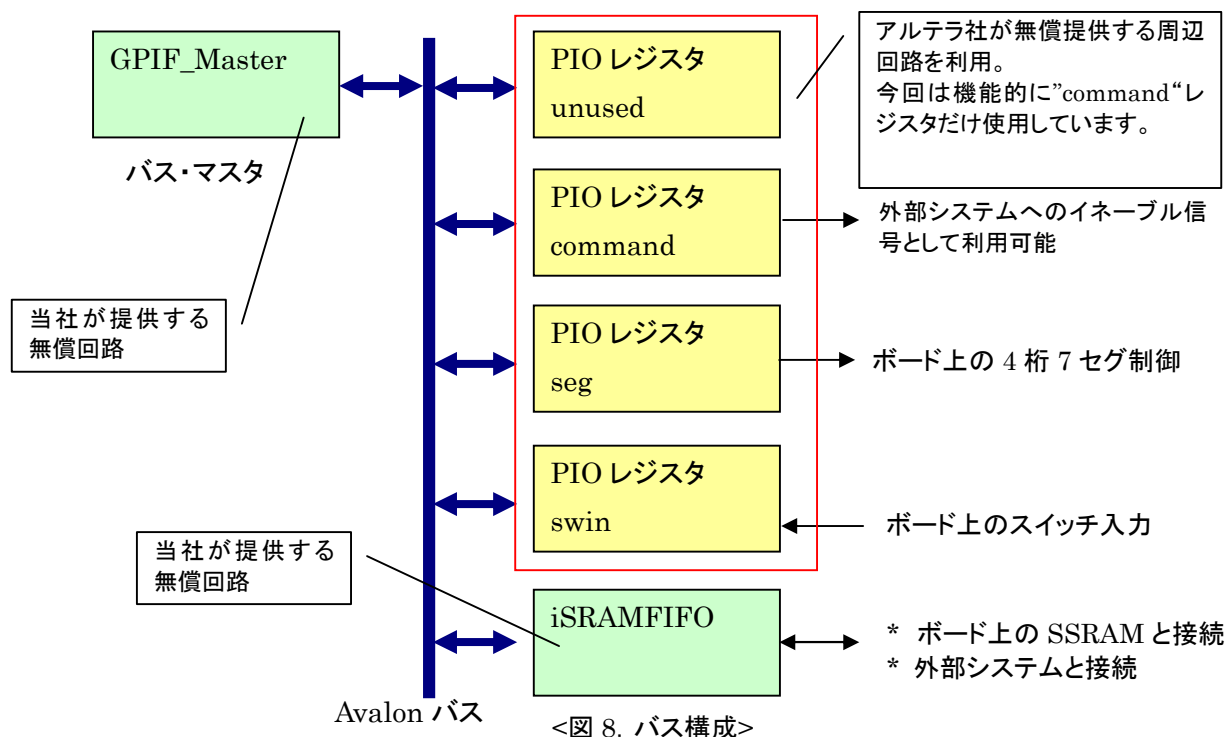
<図 7. Qsys ツール画面>

Qsys ツール内で、iSRAMFIFO モジュールのバス幅を変更したり、モジュールの追加、モジュール名の変更、アドレスの変更等を行った後には、必ず Qsys ツール画面の“Generation”タブから“Generation”ボタンをクリックして、Qsys ツールにシステム変更後の回路(HDLコード)を出力させてください。
Generate ボタンをクリックしないと Qsys ツール画面で変更した情報が、実際の HDL コードに反映されません。

(Qsys プロジェクトを新規に作る場合の注意点)

Qsys ツールを起動して、GPIF_Master ブロックと iSRAMFIFO ブロックを使った新しいシステムを設計することもできます。その際、Generate ボタンをクリックして HDL コードを生成したら、XXX.qip ファイルをプロジェクトに追加してから Q2 コンパイルをしてください。この qip ファイルを追加しないとコンパイルエラーになります。

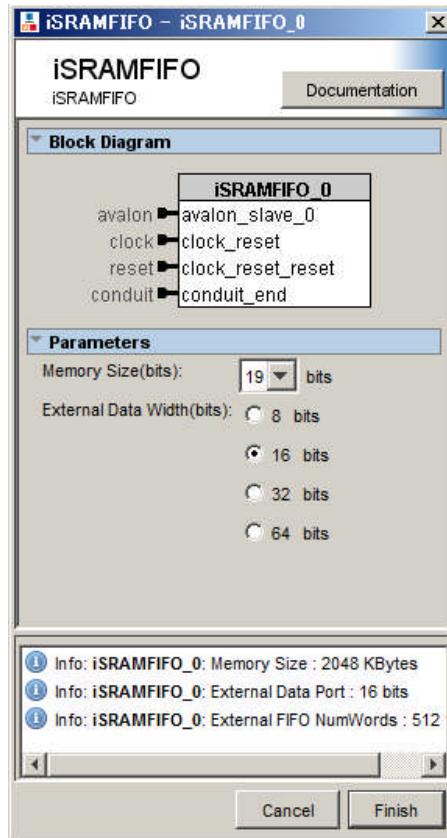
図 7 より、Avalon バスに接続しているモジュール構成は以下のようになります。



2.3 動作パラメータ

Qsys の GUI から、接続する SSRAM の容量と外部ポートのデータバス幅を変更できます。SSRAM の容量は 256KB～64MB の範囲内で設定し、外部ポートのデータバス幅は 8/16/32 bit のいずれかを選択することができます。このデータ幅設定により、外部ポート側の FIFO の深さが自動的に決まります。そのため、FPGA 内部メモリ構成に違いが発生する場合があります。たとえば、Cyclone3-FPGA を搭載する CX-USB2 ボードでは、図 9 の設定の場合、M9K メモリブロックを 2 個使います。

リファレンス・デザインは CX-USB2 用に設計しているため、SSRAM 容量は 16Mbit(アドレス 19bit)に設定しています。



<図 9. iSRAMFIFO モジュールの設定例)>

2.4 アドレス・マップ

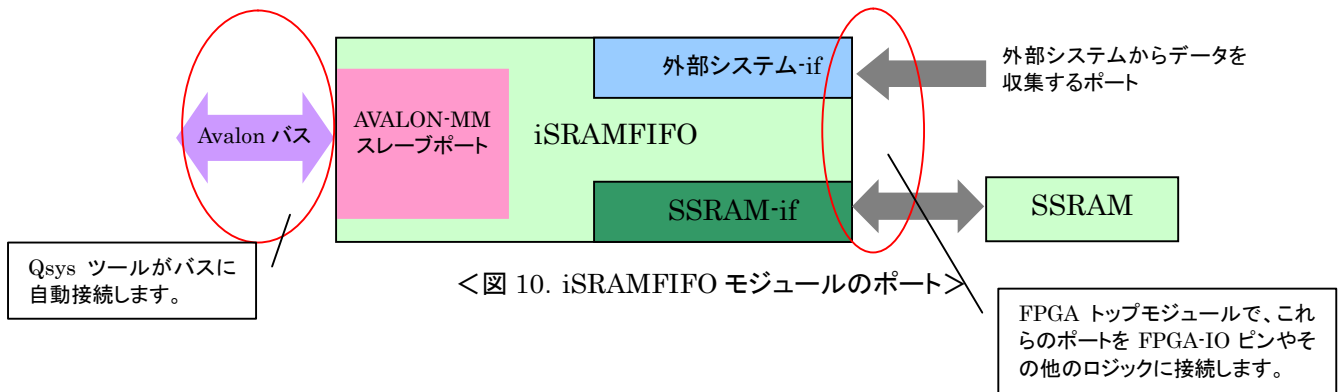
AVALON-MM マスタデバイス(GPIF_Master)から見た iSRAMFIFO モジュールのアドレスマップを示します。バスマスタ・モジュールから FIFO データポートへのアクセスは 32 ビットだけに対応しています。8 ビット/16 ビットのアクセスは行えません。Qsys で自動生成するのでユーザーがカスタマイズする必要はありません。

	iSRAMFIFO モジュールの場合
offset + 0	FIFO データポート(Read のみ)
offset + 4	FIFO ステータスポート(Read のみ)

<表 1. アドレスマップ>

2.5 iSRAMFIFO モジュール・ポートの詳細

ボード外部からデータを PC に収集するときに使用する「iSRAMFIFO モジュール」のポートを示します。図 10 で示す通り、モジュールとして 3 ポートを備え、この内 AVALON-MM スレーブ・ポートは、Qsys で自動生成されるモジュールには表示されません。FPGA プロジェクト内では、トップモジュールで「外部システム-if」ポートと「SSRAM-if」ポートをそれぞれ、FPGA 内部ロジックや FPGA ピンに配線します。



2.5.1 AVALON-MM スレーブ・ポート

AVALON-MM マスタ・デバイス(GPIF_Master)と接続する、Avalon-MM スレーブ・ポートの詳細を示します。Qsys で自動生成するのでユーザがカスタマイズする必要はありません。

【FIFO データポート】

GPIF_Master が、FIFO 化された SRAM からデータをリードするポートです。

ポート名	bit31～bit0
R/W 方向	Read のみ
ビットアサイン	FIFO データポート

<表 2. AVALON スレーブ・ポートの詳細(1)>

【FIFO ステータスポート】

RFIFO の各種ステータスを取得できます。
(注)iSRAMFIFO 全体のステータスではなく、AVALON バス側にある小容量の FIFO(RFIFO)のステータスです。

	bit31～bit18	bit17	bit16	bit15～bit9	bit8～bit0
ビットアサイン (Read のみ)	未使用(0 固定)	RFIFO_EMPTY	RFIFO_FULL	未使用(0 固定)	RFIFO_USEDW
RFIFO_EMPTY	1 : RFIFO が Empty であることを示します。 0 : RFIFO が Empty でないことを示します。 RFIFO が Empty の時に FIFO データポートを読み出すと、RFIFO にデータが書き込まれるまでバスがロックしてしまいます。注意してください。				
RFIFO_FULL	1 : RFIFO が Full であることを示します。 0 : RFIFO が Full になっていないことを示します。				
RFIFO_USEDW	RFIFO に書き込まれているデータ数を示します。 最大で 256 ワード(100h)です。				

<表 4. AVALON スレーブ・ポートの詳細(2)>

2.5.2 SSRAM-IF ポート

iSRAMFIFO モジュールとボード上の SSRAM を接続するためのポートです。データ幅は 32bit 固定ですが、アドレス線の本数は、図 7 で示した “Memory Size(bits)” の設定本数になります。

信号名	機能	属性	備考
ssram_clkin	iSRAMFIFO コアクロック入力	I	FPGA 内部のクロックを接続
ssram_clkout	SSRAM クロック出力	O	SSRAM へ供給数するクロック
ssram_ab	アドレス・バス	O	可変
ssram_db	データ・バス(32bit)	IO	32bit 固定
ssram_bwan	バイトアクセス	O	ボード上で Low レベル固定でも動作します。通常、FPGA から制御します。
ssram_bwbn	バイトアクセス	O	
ssram_bwcn	バイトアクセス	O	
ssram_bwdn	バイトアクセス	O	
ssram_wen	ライトイネーブル	O	FPGA から制御が必要
ssram_cken	クロックイネーブル	O	Low レベルに設定
ssram_zz	スリープ	O	Low レベルに設定
ssram_ftn	フロースルー／パイプラインモード切替	O	必ず、フロースルー・モードで運用 Low レベルに設定
ssram_lbon	リニアバースト	O	Low レベルに設定
ssram_adv	バーストアドレスカウンタ制御	O	Low レベルに設定
ssram_oen	アウトプットイネーブル	O	Low レベルに設定
ssram_csn	チップイネーブル	O	FPGA から制御が必要

<表 5. SSRAM-IF ポートの詳細>

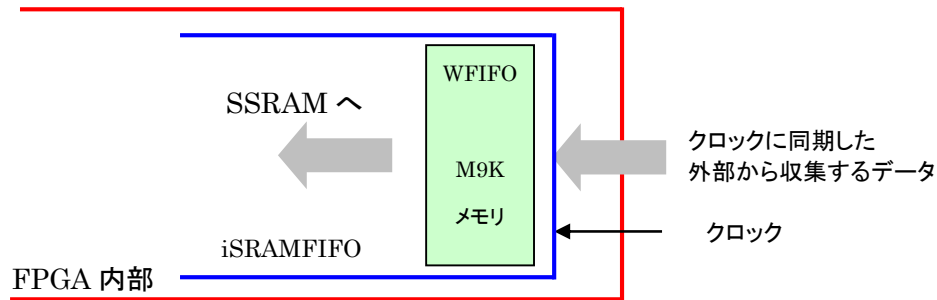
ボード製品により、SSRAM 制御の方法に違いがあります。FPGA のプロジェクト内で配線・接続しなければならない制御線と、ボード上のソルダ・パターンやジャンパ、ディップスイッチの設定を行う制御線があります。

SSRAM の動作モードは、フロースルー・モードに設定し、ボード上で FPGA に接続している SSRAM 制御線はすべて FPGA プロジェクト内で配線してください。それ以外の制御線は、備考欄の記載のようにボード上で Low レベルに設定します。

2.5.3 外部システム-IF ポート

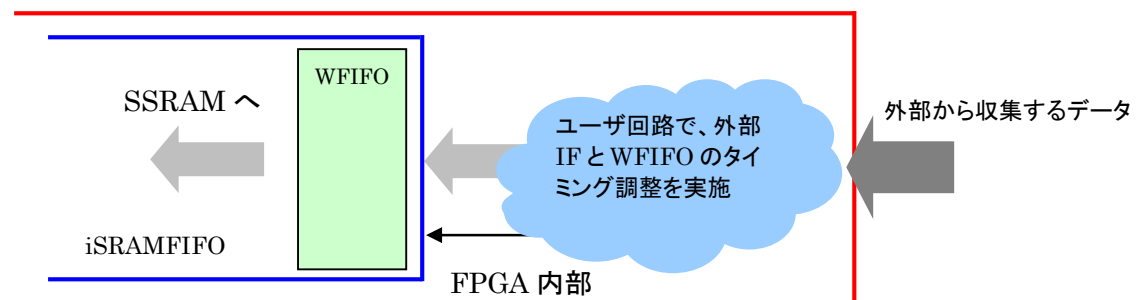
外部システム(例えば、AD コンバータ)からデータを収集するためのポートです。8/16/32 ビットのデータ幅に対応します。図 9.iSRAMFIFO モジュールの設定画面で示す GUI 画面で、External Data Width(bits) の設定値がデータ幅になります。

外部システム-IF ポートは、図 11 で示す様に FPGA の内蔵メモリを FIFO として使用しているのです、信号タイミング仕様は FPGA 製品内蔵メモリのタイミング仕様に準じます。



<図 11. FIFO 部のブロック図>

外部システムとのインタフェース仕様は、ユーザ回路を追加することで自由に変更することができます。下図で示す様に FPGA 内部の FIFO ブロック前段にユーザ回路を追加して、外部システムとのインタフェース仕様に適合させます。例えば、外部システムのインタフェースが 64bit のデータ幅の場合、iSRAMFIFO の外部システム-IF ポートでは直接対応できませんが、ユーザ回路により 32bit 2 ワードに組み下げることで対応することができます。



<図 12. 外部インタフェース仕様のカスタマイズ>

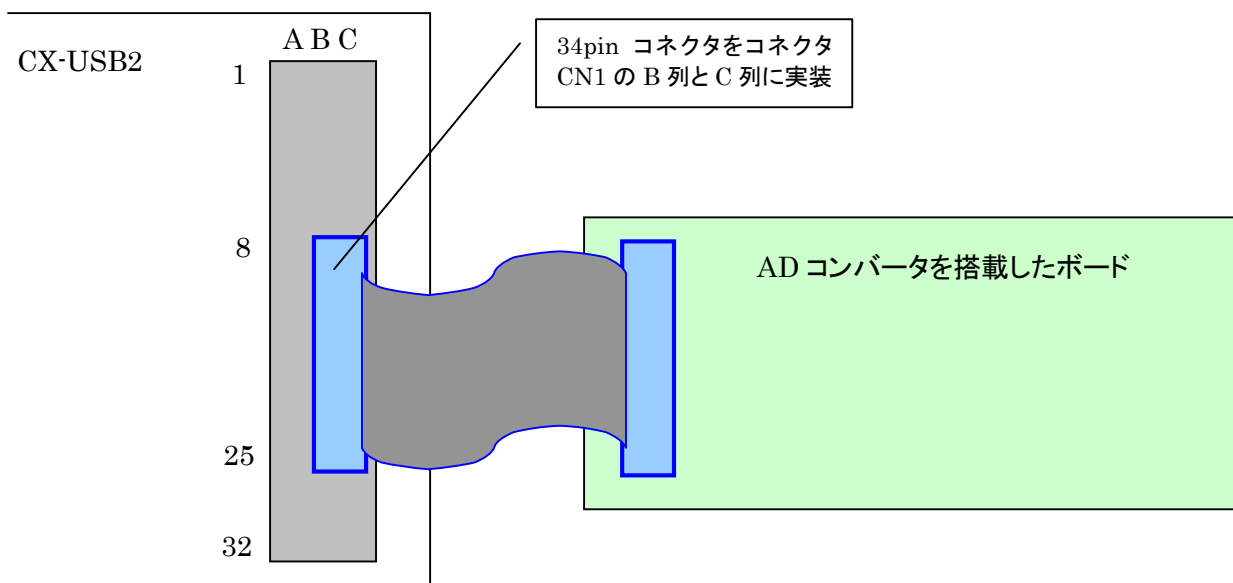
信号名	機能	属性	備考
wr_clk	WFIFO のライトクロック	I	
wr_data	WFIFO データ入力	I	8/16/32 ビット可変
wr_req	WFIFO-ライトリクエスト	I	H レベルでデータを WFIFO に書き込み
wr_empty	WFIFO-EMPTY	O	WFIFO に何もデータが書き込まれていない
wr_full	WFIFO-FULL	O	WFIFO に空き容量がない
wr_usedw	WFIFO に書き込まれているデータのワード数	O	信号本数 9~11(可変)

<表 6. 外部システム-IF ポートの詳細>

図 11 で示す WFIFO ブロックのタイミング仕様は、使用する FPGA のメモリ仕様になります。CX-USB2 システム開発ボードに適用する場合は、FPGA=Cyclone3 なので、M9K メモリのタイミング仕様になります。

【FPGA のピンアサイン】

CX-USB2システム開発ボード用リファレンス・デザインにおけるFPGAピン設定は、表6に示す外部システム・IFポートに対して行います。この例では、CX-USB2のCN1コネクタに34ピンのコネクタを実装し、該当するFPGAのIOピンをQ2ツール上で設定しています。ピン設定情報は、PIN_SET.tclファイルに記述しています。



<図 13. CX-USB2 と外部ボードとの接続>

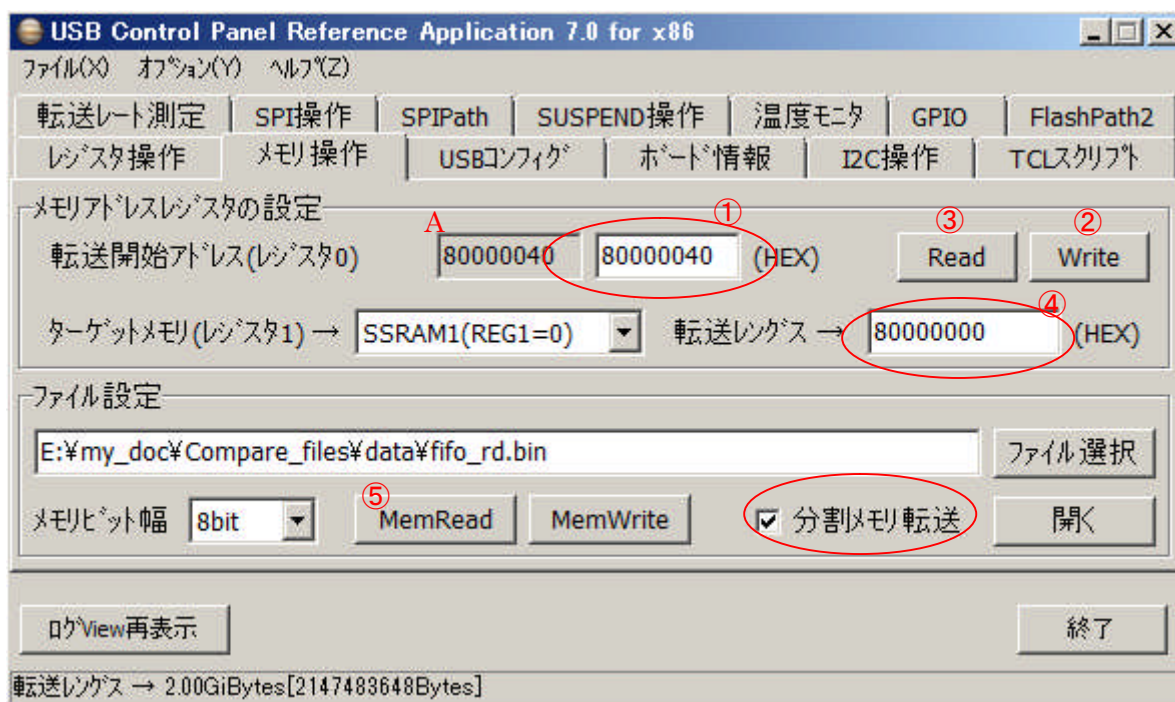
実際に外部システムとケーブル接続またはコネクタ接続する際には、外部システム・IFポートとの接続を考慮し、正しくFPGAピンアサインを行ってください。

3. モジュールの使用方法

3.1 制御アプリケーションの操作

GPIF-AVALONブリッジ(GPIF_Master)をバスマスタとして構成したFPGAプロジェクトでは、制御アプリケーション”RefApp7.exe“の「メモリ操作」画面で制御できます。

RefApp2の場合、PCに搭載したメインメモリの空き容量が転送データの限界になります。2GBを超えるようなデータ転送を行う場合には、RefApp7を利用してください。RefApp7では、「メモリライト」ボタンの右横に“分割メモリ転送”チェックボックスがあります。この機能にチェックをすることで、メインメモリの空き容量に関係なく、PCの空きHDD容量に依存したデータ転送ができます。詳細は、SmartUSB技術サポートサイトのページを参考にしてください。http://www.smartusb.info/article.php/Refapp7_mem_transfer



<図 14. 制御アプリケーション “RefApp7.exe” のメモリ操作画面>

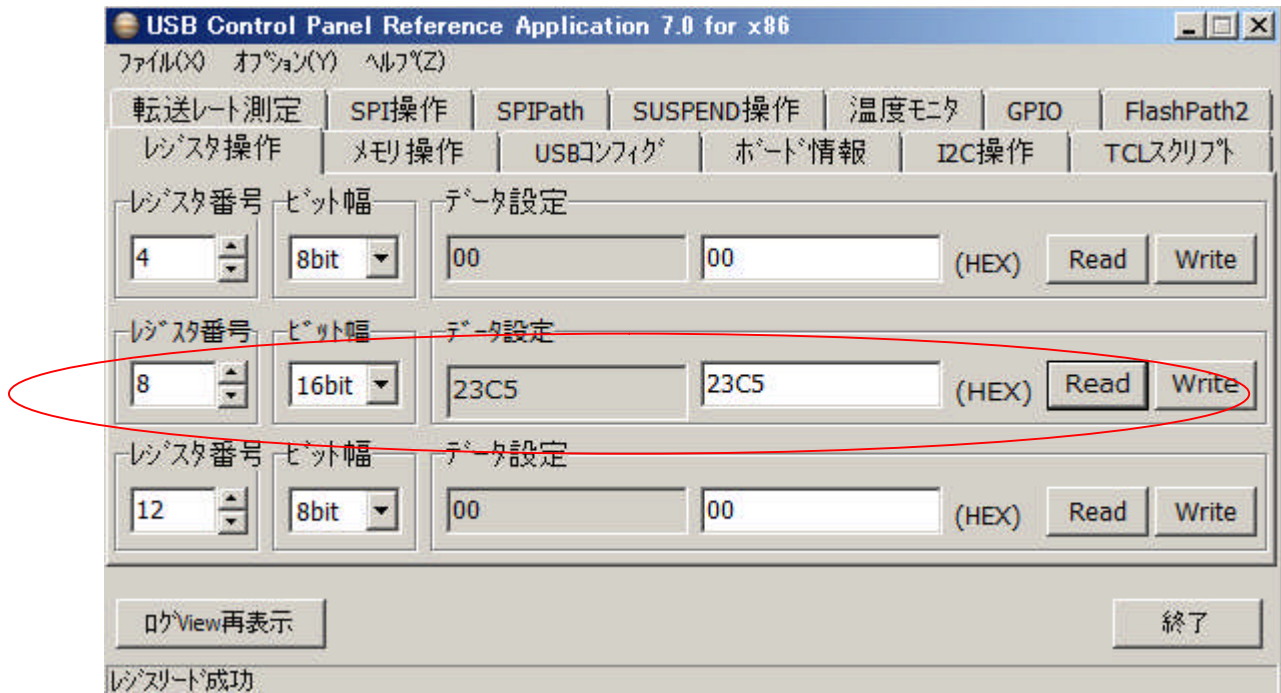
【設定手順】

- 1 「転送開始アドレス」に iSRAMFIFO モジュールのデータポート・アドレスを設定します。図 7 で示す Qsys のベースアドレスです。
但し、GPIF-AVALON ブリッジは、カレントアドレスをインクリメントしないモードを使用するので、メモリ転送開始アドレスの最上位の BIT31 には必ず「1」をセットしてください。
SRAMFIFO モジュールのデータポートが ”00000040”h の場合には”80000040”h と指定します。
- 2 Write ボタンをクリックして設定します。
- 3 Read ボタンをクリックして正しく設定したか確認してください。読み出したデータは図 14 中の“A”で示す窓に表示します。
- 4 PC に収集したいデータ量を指定します。図 14 の例では、「80000000” h)なので、2GB のデータを収集し、「ファイル設定」で指定したファイル“fifo_rd.bin”を生成します。必ず、分割メモリチェックボックスにチェックを入れてください。データ量が 1GB 未満なら分割メモリチェックボックスは、チェックしないでください。その方が高速に収集できます。
- 5 「メモリリード」ボタンをクリックするとデータ収集を開始し、No.4 で設定したデータ量を収集し終わるとファイル化してメモリリード動作を終了します。（注意）転送レングスは 512 バイト単位になるように設定してください。

3.2 レジスタ操作

図 8 のバス構成では、アルテラ社提供の無償 PIO モジュールを利用しています。今回の iSRAMFIFO モジュールを制御する上では、これらの PIO は不要ですが、ハードウェア・レジスタ・サンプルとしてデザインに追加しています。これらの PIO モジュールは、RefApp7 の「レジスタ操作」タブから値を設定し、外部システムや FPGA 内部回路への設定に利用することができます。

例えば、CX-USB2 の 4 桁 7 セグに数値を表示したい場合、レジスタ No.8 を 16bit 幅に設定し、16 進数でデータを設定 (WR) すると、ボード上の 7 セグに設定した数値が表示できます。レジスタ No.4 は汎用の 8bit レジスタ、レジスタ No.12 は 8bit 入力専用レジスタの設定になっています。



<図 15. レジスタ操作画面>

4. データ転送レート

USB インタフェースの実効データ転送レートは、約 40MB/s です。

制御ソフト名	デバイスドライバ	API (DLL)	USB 実効データ転送レート
RefApp7	Microsoft 社製 WinUSB デバイスドライバ	使用可能	40MB/s

<表 11. USB 実効データ転送レート>

(注) 制御アプリの種類、違いについては、こちらを御覧ください。

<http://www.prime-sys.co.jp/products/refapp.htm>

4.1 外部システムとのインタフェース速度

RefApp7を利用して、USBの実効データ転送レートが平均40MB/sの場合、iSRAMFIFOモジュールと外部システムのデータ転送帯域も約40MB/s以下に設定しないと、すべてのデータを正しく通信できなくなります。



<図 16. iSRAMFIFO モジュールを適用してデータ収集する場合のデータ転送レート>

しかし、iSRAMFIFOモジュールの外部システム-IFポートにある、FIFOのステータス信号(Full, Empty, usedw)を利用して、FIFOのオーバーフロー／アンダーフローを防ぎながら外部システムとインタフェースすることができれば、正しく通信することができます。

(重要)

外部システムから連続した長時間のデータを収集したり、外部システムへ連続したデータを出力する場合は、USBインタフェース側の実効データ転送レートを超えない範囲で、外部システムとのデータ帯域を決定する必要があります。

4.2 動作クロック

iSRAMFIFOモジュールには、SSRAMに供給するクロック(SRAMクロック)と外部システムから供給されるクロック(外部クロック)の2系統があります(注)。

SRAMクロックと外部クロックは任意の周波数で動作させる事ができますが、SRAMクロックは外部クロックに対して十分に早い周波数で動作させる様にしてください。SRAMクロックが外部クロックに対して十分に早くないと性能が低下します。リファレンス・デザインではSRAMクロックを120MHzで運用しています。

SRAMFIFOモジュールは、FPGA外部のSSRAMをFlowThrough(フロースルー)モードで使用しています。このため、ボード上にはアクセス速度200MHzのSSRAMを搭載しても、SRAMクロックの上限は133MHzになります(注2)。

(注1) 実際には、AVALONバス用のクロックもあります。

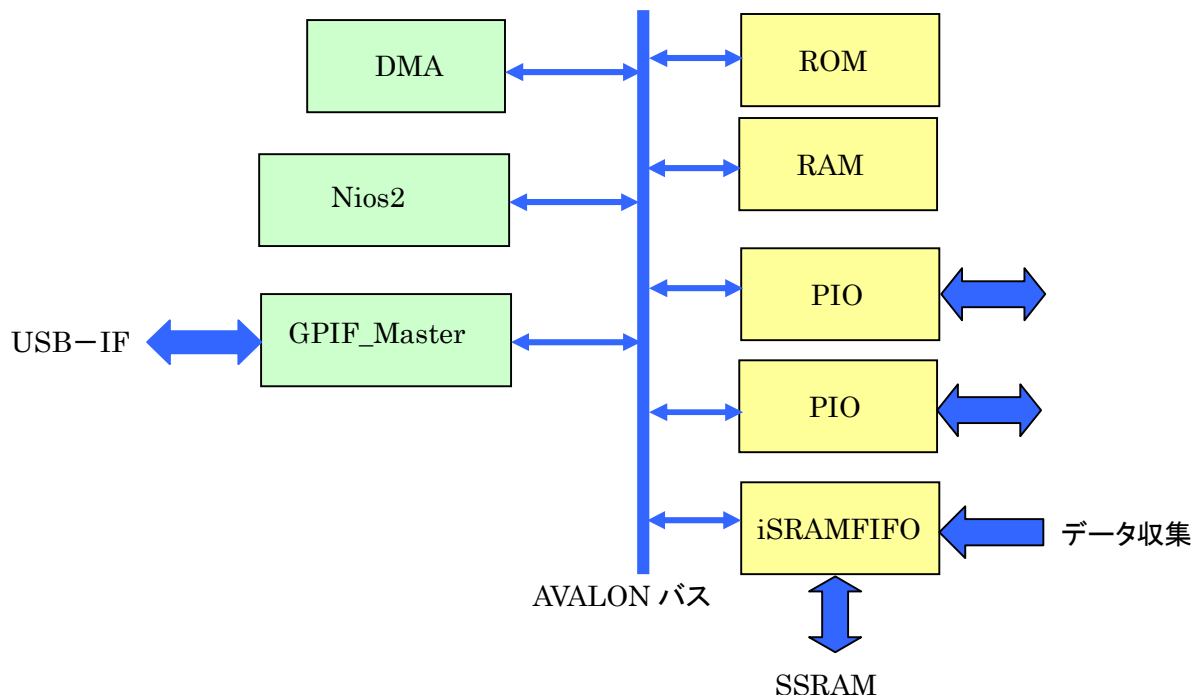
(注2) 200MHzでアクセスするためにはパイプラインモードで運用する必要がありますが、クロックレイテンシが増えるので、SSRAMFIFOモジュールではフロースルーモードを採用しています。

5. CPU を混載したシステム設計

Qsys システム統合ツールは、アルテラ社の Nios2 ソフト CPU をバス・マスタとした組み込みシステムを構築するためのツールです。当社では Nios2 CPU に代わるバス・マスタを開発し、GPIF_AVALON バスブリッジ回路を GPIF_Master モジュールとしてリリースしています。SRAM-FIFO モジュールは AVALON バス・スレーブなので、バス・マスタの GPIF_Master が AVALON バス仕様により SRAM-FIFO モジュールを制御しています。

GPIF_Master は複数のバス・マスタと共存できるので、Nios2-CPU も加えたシステムの開発ができます。iSRAMFIFO モジュールで収集したデータを Nios2 で処理しながら、処理結果を PC に転送するようなアプリケーション開発も可能です。

Nios2 ソフト CPU は、Nios2e コアが無償となり、無償版 Q2 ツールと組み合わせれば、Smart-USB Plus 製品ボードだけ購入することで、低価格で、短納期の組み込みシステム開発が可能になります。



< 図 17. Nios2 と混在したシステム開発の例 >

備考:

【関連資料】

[SUA006.pdf](#) 「GPIF-AVALON ブリッジ回路」解説

[SUA007.pdf](#) 「"SOPC Builder"を利用した Smart-USB Plus 製品用リファレンス回路 SRAM-FIFO モジュール」解説