

USB3.0-IP ホスト側リファレンス・デザイン説明書

Rev1.11J 2015/05/14

このドキュメントは Altera 製あるいは Xilinx 製評価ボードと USB3.0-IP デモボードを使って USB3.0-IP ホスト側コアを評価するリファレンス・デザインに関して説明したものです。本リファレンス・デザインによる実機ボード評価の具体的な手順については以下の USB3.0-IP コア専用 Web ページを参照してください。

Altera 版 USB3.0-IP コア専用 Web サイト: http://www.dgway.com/USB3-IP_A.html

Xilinx 版 USB3.0-IP コア専用 Web サイト: http://www.dgway.com/USB3-IP_X.html

なお、本ドキュメントにおいて Altera 版は Cyclone IV GX FPGA 開発キット(型番:DK-DEV-4CGX150N)用デザインで、Xilinx 版は Spartan-6 FPGA SP605 評価キット(型番:EK-S6-SP605-G)用デザインで説明しています。

1. リファレンス・デザインを活用・応用するための推奨される知識

本リファレンス・デザインを理解し、活用・応用するためには、以下の知識を要することが推奨されます。

1. Altera 設計ツール QuartusII/Qsys/NiosII Software Build Tools for Eclipse、あるいは、Xilinx 設計ツール ISE/Vivado/EDK/SDK に関する基本的な操作方法についての知識
2. C 言語に関する基本的な知識
3. USB3.0 の動作に関する知識
4. FAT システムに関する知識

Altera/Xilinx FPGA 設計ツールに関しては、活用・応用する範囲によって必要とするツールの知識が異なります。

また、リファレンス・デザインでは PHY 層として、外付け PHY チップ(TI 社 TUSB1310A)を採用しており、当該チップの各種技術資料も参照することを推奨します。

TUSB1310A 紹介 Web サイト: <http://www.ti.com/product/TUSB1310A>

USB3.0 の動作に関しては、USB2.0 までの知識を活用することが可能であり、ソフトウェアに関しましてはほぼ同じと言えます。本リファレンス・デザインでは、USB の Mass Storage Class (Bulk-Only)を制御するホストシステムを実現しており、当該クラスの知識はデザインの理解に役立ちます。ハードウェア(プロトコル,シリアル転送)の詳細や USB3.0 と USB2.0 の差異等については、各種解説書、web サイト、雑誌などをご参照ください。

本リファレンス・デザインでは、マスタストレージを応用動作させる為に、ファイル・システムとして FAT システムを使っております。本リファレンス・デザインの FAT システムはフリーソフトウェア(FatFs)を移植して使用しております。当該システムについては作者(赤松武史氏)の web サイト、雑誌記事をご参照ください。ただし、本コアおよびリファレンス・デザインに関連して作者本人に問い合わせることはご遠慮ください。

技術参照例:

CQ 出版社 Interface 誌 2010 年3月号 : USB3.0 の特徴と USB2.0 との比較

CQ 出版社 Interface 誌 2010 年9月号 : USB マス・ストレージ・クラスの制御手順

CQ 出版社 Interface 誌 2010 年9月号 : FatFs と極小 Petit-FatFs の使い方

CQ 出版社 Interface 誌 2010 年9月号 : FAT ファイル・システムの詳細

FatFs web サイト):

FatFs 汎用 FAT ファイルシステム・モジュール(http://elm-chan.org/fsw/ff/00index_j.html)

2. ハードウェアの説明

2-1 Altera 版リファレンス・デザインのハードウェア

Altera 版リファレンス・デザインの全体構成を図 1 に示します。全体で USB ホストシステムを構成しています。

1. USB3.0-IP ホスト側コア: USB3.0 のリンク層,プロトコル層を含みます。
2. usb_rp_c4gx: FPGA 全体のトップモジュール
3. pipe_to_iob_c4gx: コアの PIPE インターフェースと FPGA の IO を接続し、外付け PHY チップと接続します。
4. nios_qsys64_c4gx_rp: NIOS2 プロセッサや DDR2 コントローラ(atlmemddr)を含み、コアとのレジスタアクセスと DMA アクセスを行います。
5. ソフトウェア: デバイス・フレームワークより上位の制御(規格書の第9章以降)はソフトウェアで構成を管理します。
ここでは、Mass Storage Class を制御するホストシステムを実現しております。
6. デモ・ボード: DesignGateway 社製 HSMC デモ基板(型番:AB08-USB3HSMC)です。
外付け PHY チップ(TI 社 TUSB1310A)を搭載し、HSMC コネクタ経由で CycloneIV GX ボードと接続します。

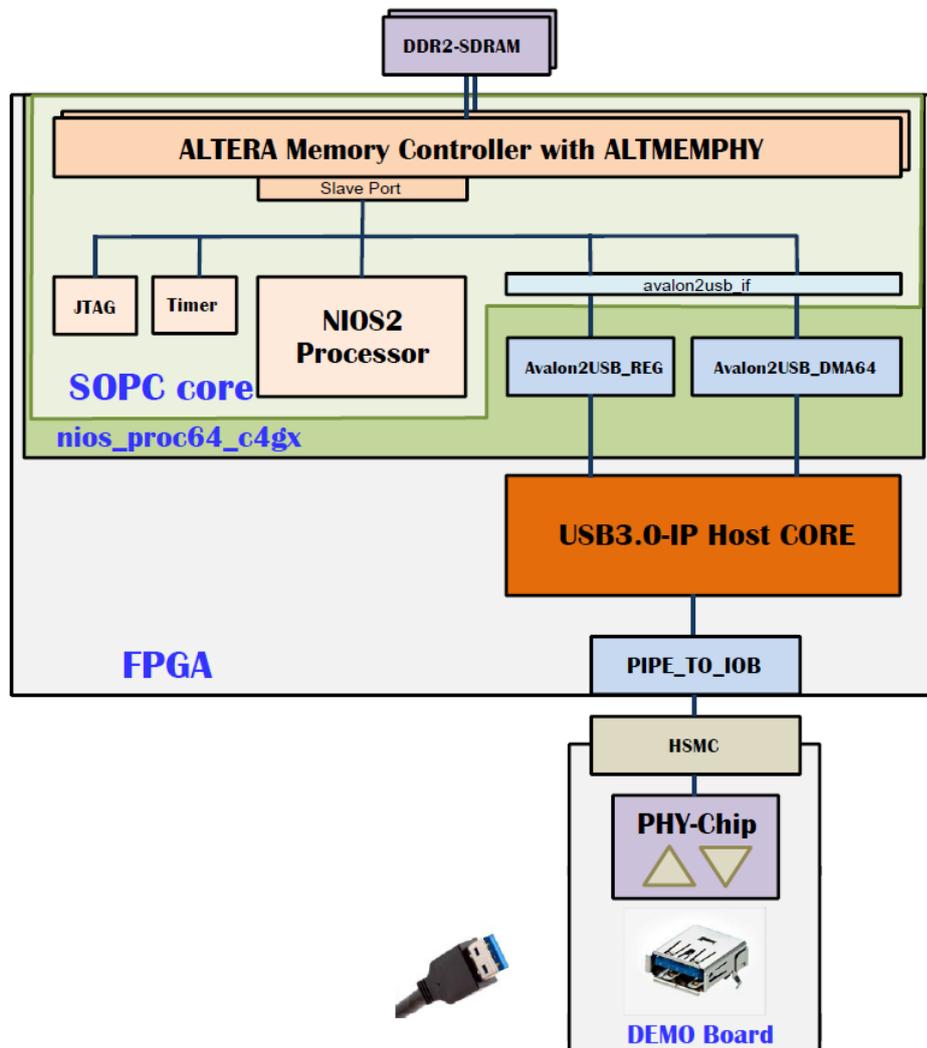


図 1: Altera 版ホスト・リファレンス・デザインの全体構成

- USB3.0-IP ホスト側コア

コアに関しましては、データ・シートを参照してください。

- usb_rp_c4gx

コアとリファレンス・デザインの各モジュールの接続を行っています。

また、PHY チップへの各種設定信号の出力もここで行っています。

本リファレンス・デザインのデモ・ボードでは、デバイスモードと兼用の A タイプの USB コネクタを使用する都合上、USB3.0 のシリアル信号(TX/RX)の+/-を反対に接続しています。ただし、PHY チップの+/-の扱いの反転はコアが自動的に判定し O_LANE_POL を出力します。したがって、コアからの O_LANE_POL 信号を PHY の RX_POLARITY へ接続します。なお TX 側は、USB3.0 の仕様上デバイス側にて+/-反転に対する処理が自動的に行われます。

また、PHY チップの Strapping Options 設定を行うため、Options 設定中の状態と通常動作中の状態の切替えもここで行っております。

- pipe_to_iob_c4gx

コアと外付け PHY チップを接続するモジュールです。構成を図 2 に示します。

この部分の動作周波数は 250MHz(4nS)と非常に高速ですが、本モジュールにより接続を行います。

要点としては、

⇒ 受信信号のタイミングとクロックは、Place & Router の機能によりタイミング調整を行う。必要に応じて PLL の phase_shift 機能を併用する。

⇒ TX_CLK は PLL の phase_shift 機能により、内部クロックに対し約 180 度位相調整して出力する。

⇒ 入力 FF からコア、コアから出力 FF 間は、必要に応じて、FF を数珠つなぎにしてタイミングを合わせる。

具体的な実装方法についてはリファレンス・デザイン内の本モジュールを参照してください。

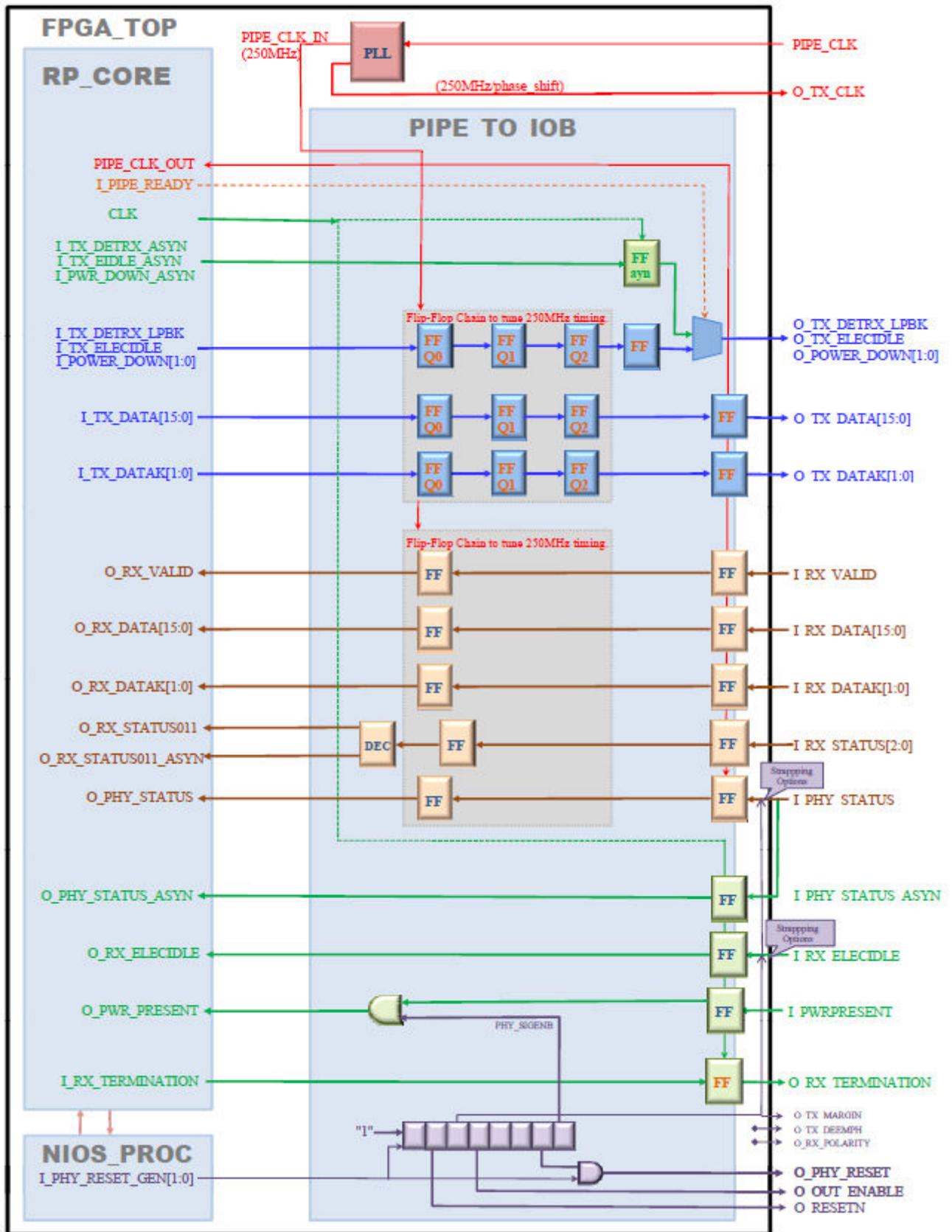


図 2: PIPE_TO_IOB の構成

- nios_qsys64_c4gx_rp

NIOS2 プロセッサと DDR2 コントローラ(altmemddr)を含み、NIOS2 から Avalon バス経由でレジスタアクセスを行う場合の接続モジュール(Avalon2USB_REG)と altmemddr から Avalon バス経由で DMA アクセスを行う場合の接続モジュール(Avalon2USB_DMA64/Avalon2USB_DMA)で構成されます。なお、Cyclone IV GX 版では、DDR2 コントローラは2つ用いています。1つあたり 64MB の DDR2 メモリと接続され、2つで 128MB となります。

nios_qsys64_c4gx_rp 全体の概略は全体構成を図 1 に示しましたので、それを参照ください。

Avalon2USB_DMA64 の概略構成を図 3 に示します。Avalon バスの動作は ALTERA の当該資料、コア側インターフェースの動作はデザイン・シートを参照ください。Cyclone IV GX 版では、Avalon バスは 64bit(@75MHz)、コアは 32/64bit(@125MHz)ですので、途中の PFIFO によりその変換を行っています。

詳細はリファレンス・デザイン中のソースを参照ください。

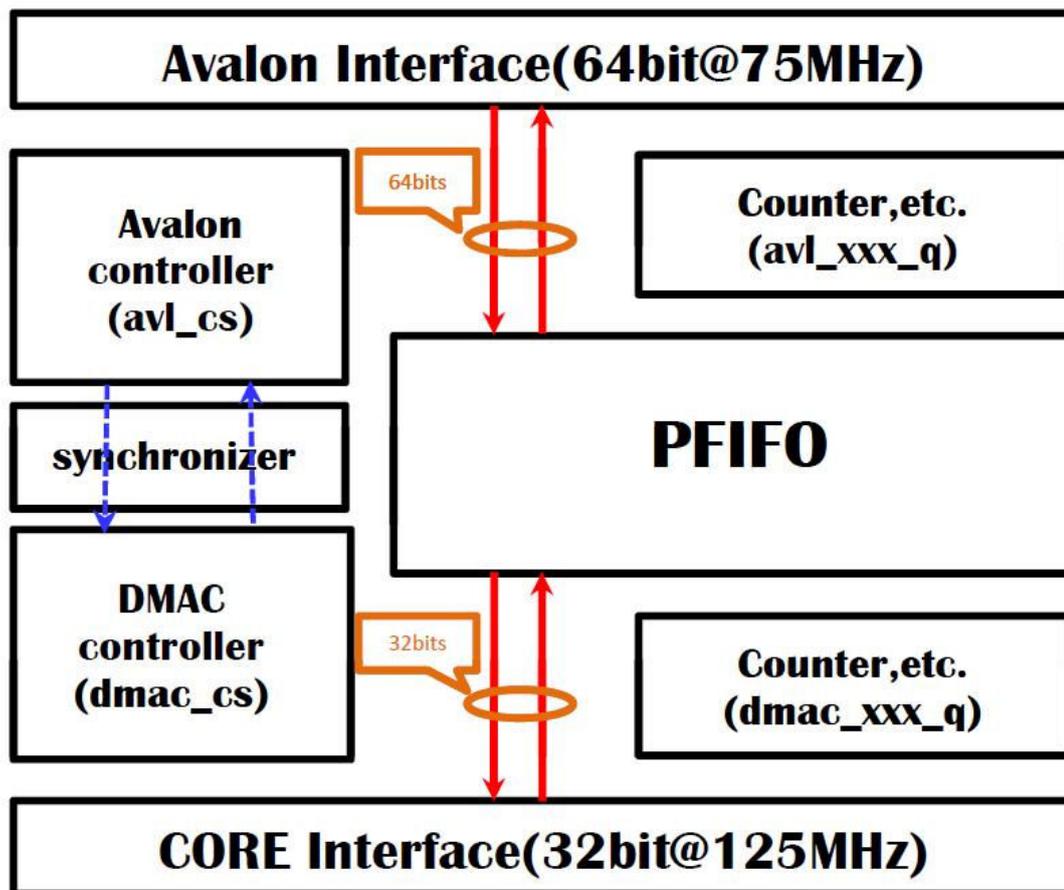


図 3: Avalon2USB_DMA64 の概略構成

2-2Xilinx 版リファレンス・デザインのハードウェア

Xilinx 版リファレンス・デザインの全体構成を図 4 に示します。全体で USB ホストシステムを構成しています。

1. USB3.0-IP ホスト側コア: USB3.0 のリンク層,プロトコル層を含みます.
2. usb_ep_sp605: FPGA 全体のトップモジュール
3. pipe_to_job_sp605: コアの PIPE インターフェースと FPGA の IO を接続し、外付け PHY チップと接続します.
4. mb_proc_npi64_dma64_sp605: Microblaze や DDR3(mpmc)を含み、コアとのレジスタアクセスと DMA アクセスを行います.
5. ソフトウェア: デバイス・フレームワークより上位の制御(規格書の第9章以降)はソフトウェアで構成を管理します。
ここでは、Mass Storage Class を制御するホストシステムを実現しております。
6. デモ・ボード: DesignGateway 社製 FMC デモ基板(型番:AB07-USB3FMC)です。
外付け PHY チップ(TI 社 TUSB1310A)を搭載し、FMC コネクタ経由で SP605 評価ボードと接続します。

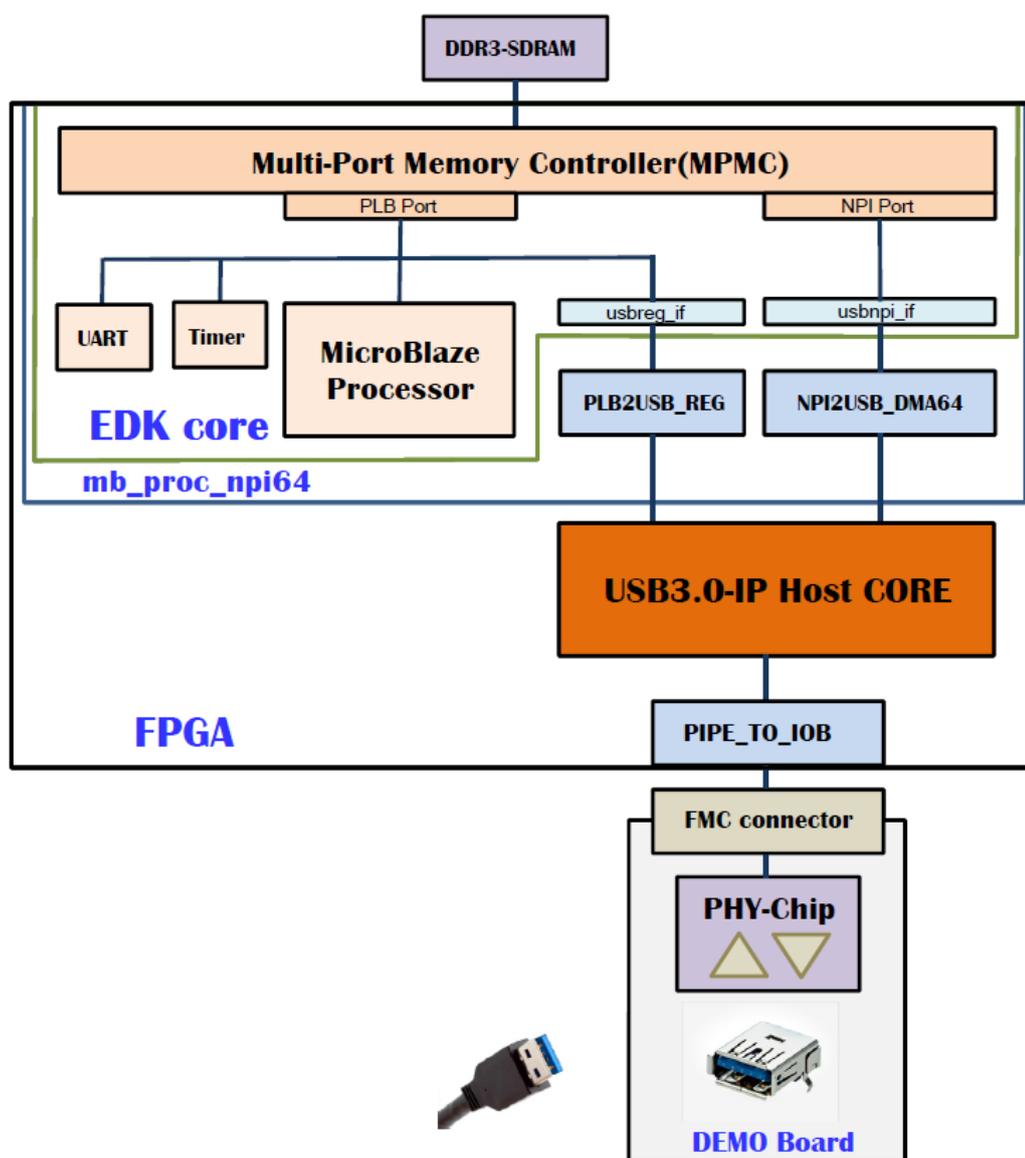


図 4: Xilinx 版ホスト・リファレンス・デザインの全体構成

- USB3.0-IP ホスト側コア

コアに関しましては、データ・シートを参照してください。

- usb_ep_sp605

コアとリファレンス・デザインの各モジュールの接続を行っています。

また、PHY チップへの各種設定信号の出力もここで行っています。

本リファレンス・デザインのデモ・ボードでは、デバイスモードと兼用の A タイプの USB コネクタを使用する都合上、USB3.0 のシリアル信号(TX/RX)の+/-を反対に接続しています。ただし、PHY チップの+/-の扱いの反転はコアが自動的に判定し O_LANE_POL を出力します。したがって、コアからの O_LANE_POL 信号を PHY の RX_POLARITY へ接続します。なお TX 側は、USB3.0 の仕様上デバイス側にて+/-反転に対する処理が自動的に行われます。

また、PHY チップの Strapping Options 設定を行うため、Options 設定中の状態と通常動作中の状態の切替えもここで行っております。

- pipe_to_iob_sp605

コアと外付け PHY チップを接続するモジュールです。構成を図 5 に示します。

この部分の動作周波数は 250MHz(4nS)と非常に高速ですが、本モジュールにより接続を行います。

要点としては、

⇒ DCM_SP の phase_shift 機能により、受信信号のタイミングとクロックを合わせる。

⇒ TX_CLK は ODDR2 より出力する。

⇒ 入力 FF(IOB 内)からコア、コアから出力 FF(IOB 内)間は、コアと IOB 間の FPGA 内の距離に応じて、FF を数珠つなぎにしてタイミングを合わせる。

具体的な実装方法についてはリファレンス・デザイン内の本モジュールを参照してください。

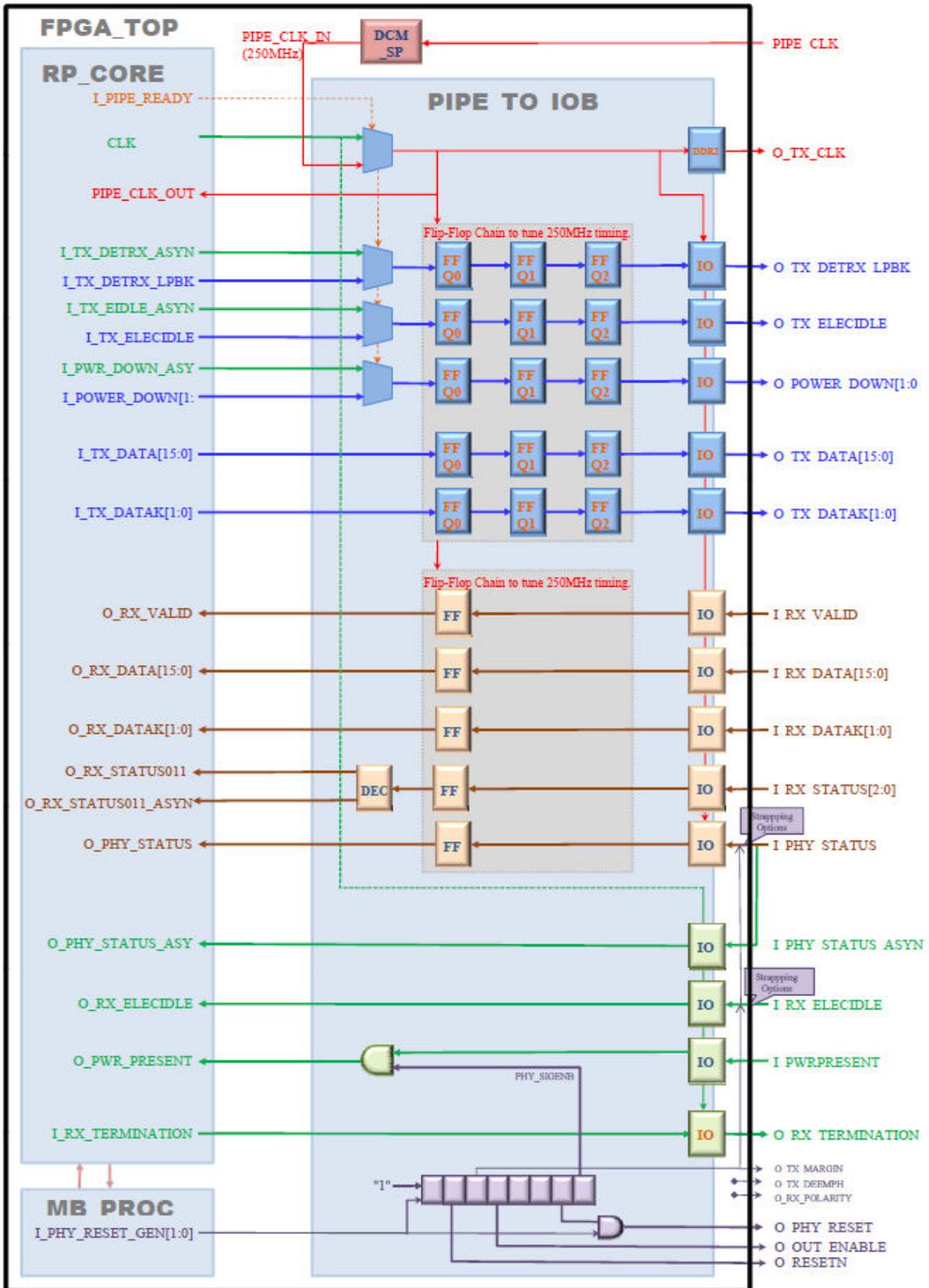


図 5: PIPE_TO_IOB の構成

- mb_proc_npi64_dma64_sp605

MicroBlaze(以下 MBZ)と DDR2(mPMC: multi-port memory controller)を含み、MBZ から PLB バス経由でレジスタアクセスを行う場合の接続モジュール(PLB2USB_REG)と mPMC から NPI バス(64bit)経由で DMA アクセスを行う場合の接続モジュール(NPI2USB_DMA64)で構成されます。

mb_proc_npi64 全体の概略は全体構成を図 4 に示しましたので、それを参照ください。

NPI2USB_DMA64 の概略構成を図 6 に示します。NPI バスの動作は Xilinx の当該資料、コア側インターフェースの動作はデザイン・シートを参照ください。NPI バスは 64bit(@83MHz)、コアは 32/64bit(@125MHz)ですので、途中の PFIFO によりその変換を行っています。

詳細はリファレンス・デザイン中のソースを参照ください。

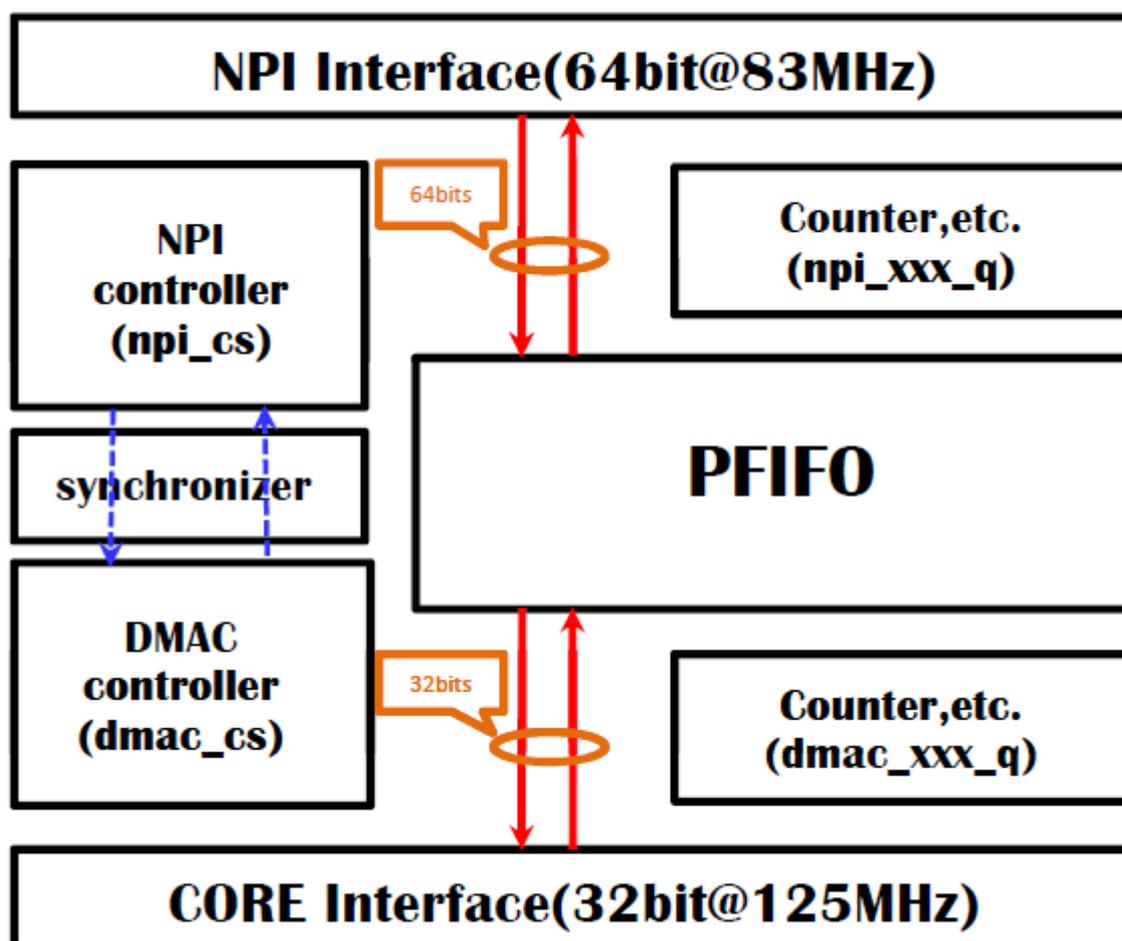


図 6: NPI2USB_DMA64 の概略構成

3. ソフトウェアの説明

本リファレンス・デザインのソフトウェアにて、各ソースの概略構成は以下となっています。また、コアのレジスタおよび概略動作の流れについてはデータ・シートを参照してください。更に詳細はソース・コードを参照してください。

- usb_core_rp.h: 全体のヘッダファイル.
- usb_com_rp.c: コマンド・シェル
- usb_bus_ctrl_rp.c: バス状態, 割り込み状態(ポーリング)等の制御
- usb_dev_ctrl_rp.c: クラス判別前のデバイス・リクエストの制御
- usb_hub_ctrl.c: ハブクラス用のデバイス・リクエストの制御
- usb_msc_ctrl_rp.c: マスストレージのデバイス・リクエストの制御および SCSI(ATA)コマンド制御
- usb_msc_ctrl_rw.c: SCSI(ATA)コマンドの内のデータのリード・ライト制御
- usb_rp0_ctrl.c: RPO[ルートポイント・ゼロ]によるデバイス・リクエスト(setup)の制御
- usb_rpn_ctrl.c: Bulk-only を構成する RPO[ルートポイント・アウト]と RPI[[ルートポイント・イン]の制御]
- usb_bffr_ctrl_rp.c: 各ログやバッファ領域の制御. UART 出力の制御

◆ Xilinx 版 SP605 デザインの注意点

Xilinx 版 SP605 リファレンス・デザインのソフトウェアでは、コマンド・シェルのサイズが大きい関係上、microblaze の LMB(local mempry bus)に接続可能なオンチップ RAM の容量(Max64KB)を超えてしまいます。そこで PLB バスに追加のオンチップ RAM (Max64KB)を接続しています。テキストコードの配置を制御する場合は、Linker Script をマニュアルで変更します。この方法についてはザイリンクスのアンサー(#21639)を参照ください。

- lscript.ld: テキストコードの配置を制御する為の Linker Script

本リファレンス・デザインでは、マスストレージを応用動作させる為に、FAT システムを使っております。なお、本リファレンス・デザインの FAT システムはフリーソフトウェア(FatFs)を移植して使用しております。FatFs 関連の各ソースは、./FatFs 配下に格納されています。ほぼオリジナルのまま使用しておりますが、以下の追加・変更を行っています。

- usb_rp_api.c: 本コア用の FatFs の下位レイア・インタフェース
- ffconf_h: 一部の define を変更。(Changed by Design Gateway で検索)
- ff.h: 本コアでは R/W 用のデータは 256 バイト境界に配置する必要がある。FatFs 用のワークバッファをポインタ変数とし初期化時に 256 バイト境界に配置する。(Changed by Design Gateway で検索)

4. 改版履歴

リビジョン	日付	内容
1.0	2012/01/06	第 1 版のリリース
1.1J	2015/05/12	Altera 版と Xilinx 版を同ドキュメントに統合
1.11J	2015/05/14	6page,7page の全体トップモジュール名を修正

Copyright: 2012 Design Gateway Co.,Ltd.