

GPIF2 用 AXI バスマスタ IP : GPIF2_Master

1. 概要

AMD/Xilinx vivado 開発ツールで利用できる AXI バスマスタ IP <GPIF2_Master> により、USB 制御コアを AXI バスマスタとし、既存の各種 AXI スレーブ IP を制御できます。また、MicroBlaze や Zynq ARM コア、DMA 等のバスマスタと混在も可能です。

WindowsPC や LinuxPC を AXI バスマスタと見立てて、FPGA を簡単に制御できる IP です。

【適用ボード】

AMD/Xilinx 製 FPGA を搭載する Smart-USB Sigma 製品

AX-Card7 (Artix7) , SX-Card7 (Spartan7)

KX-Card7 (Kintex7) , AX-Card7M (Artix7) , ZQ-Card (Zynq7000)

【対応開発ツール】

AMD/Xilinx vivado2022.2 以降

Block Design 環境で、既存の各種 AXI スレーブ IP や他のバスマスタと組み合わせて利用できます。

2. USB 制御回路の AXI バスマスタ化

2.1 提供するファイル

GPIF2_Master.v

|--- GPIF2_WFIFO.xci

|--- GPIF2_RFIFO.xci

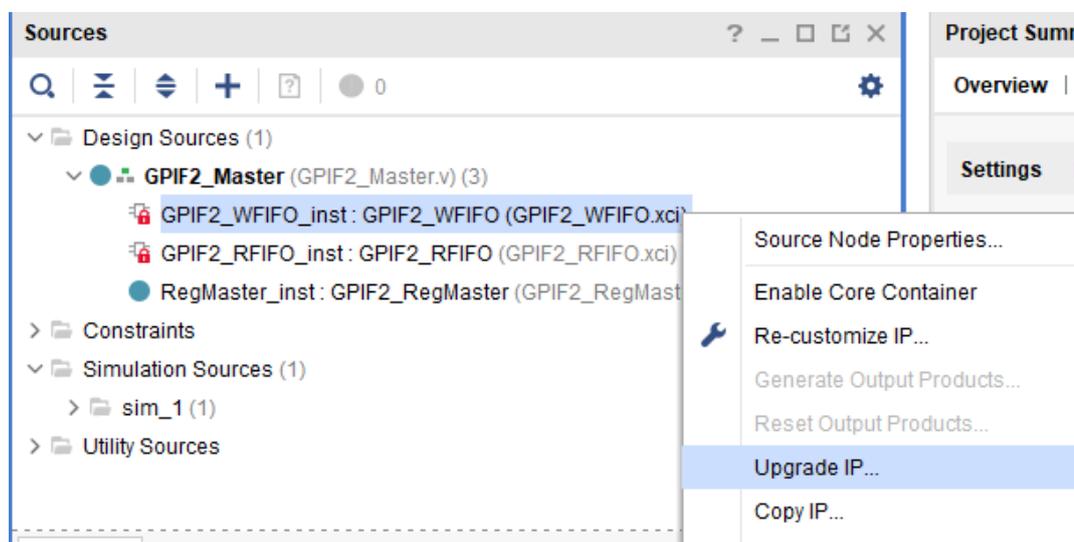
|--- GPIF2_RegMaster.v

vivado ツールの Block Design 環境で、GPIF2_Master を読み込んで利用することができます。

2.2 使い方 (Block Design で GPIF2_Master を利用する)

1 vivado で新規プロジェクトを作成し、上記の 4 つのファイルをプロジェクトに追加します。

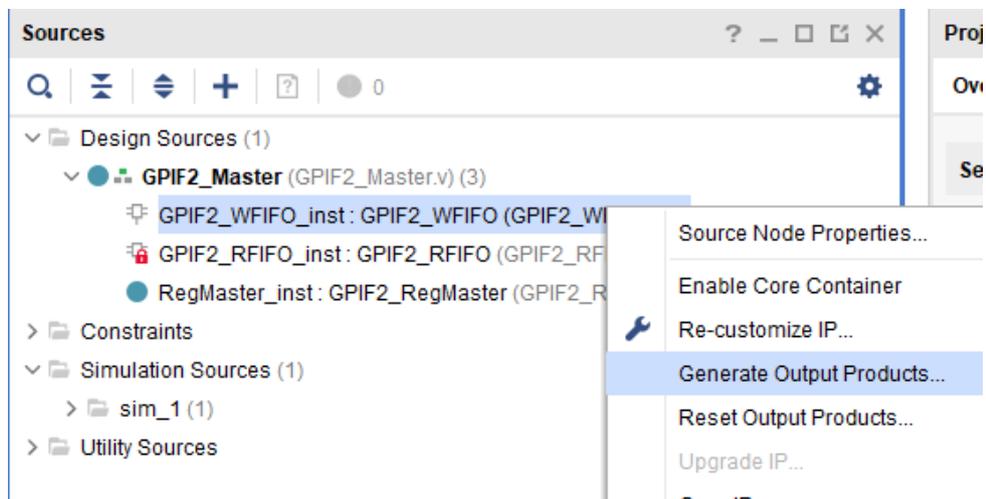
2 Source ウィンドウで GPIF2_WFIFO_inst を選択し、「Upgrade IP…」を実行します。



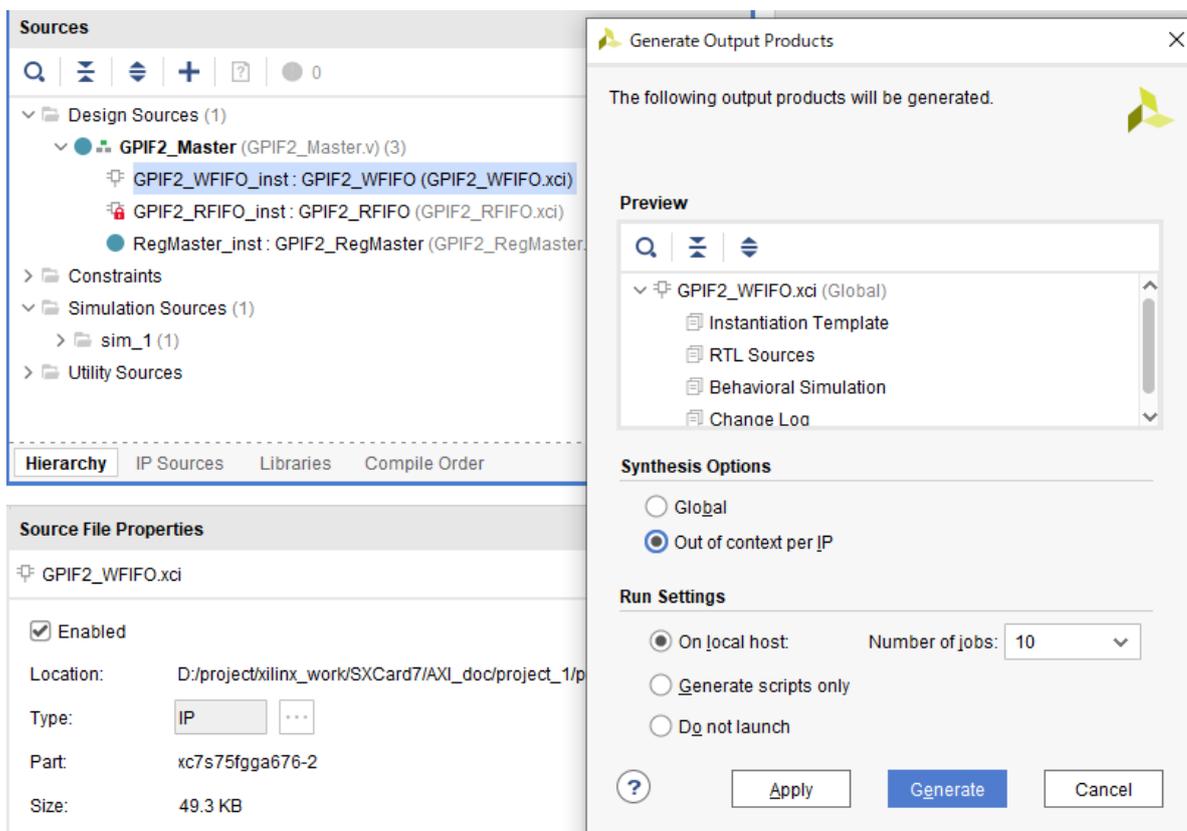
< 図 1. FIFO の Generate Output Products >

- 3 Upgrade IP 実施後、「Generate Output Products」を実行します。このとき、Synthesis Option は Out of context per IP にしてください。

同じ操作を GPIF2_RFIFO_inst にも実施します。

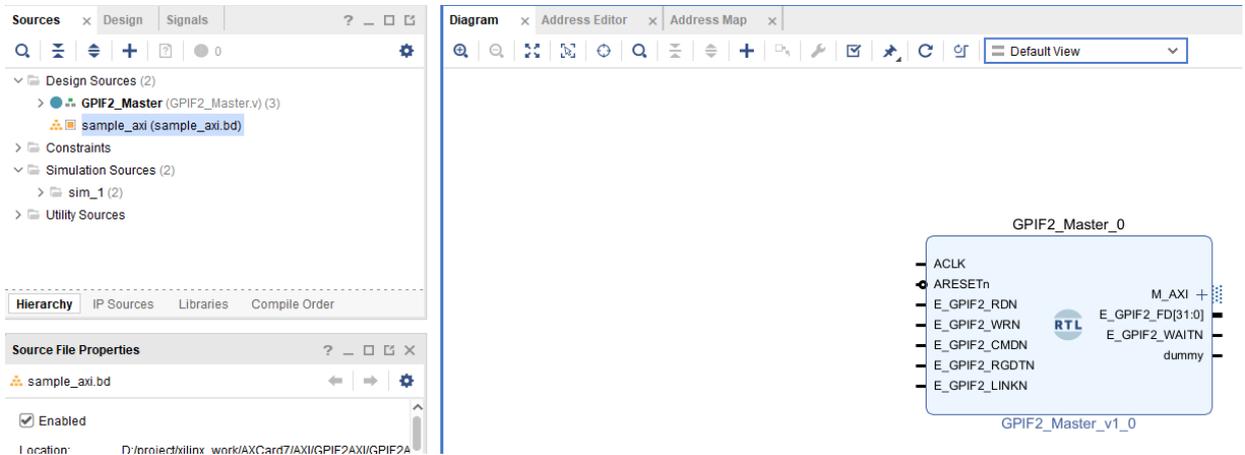


< 図 1.1. FIFO の Generate Output Products >



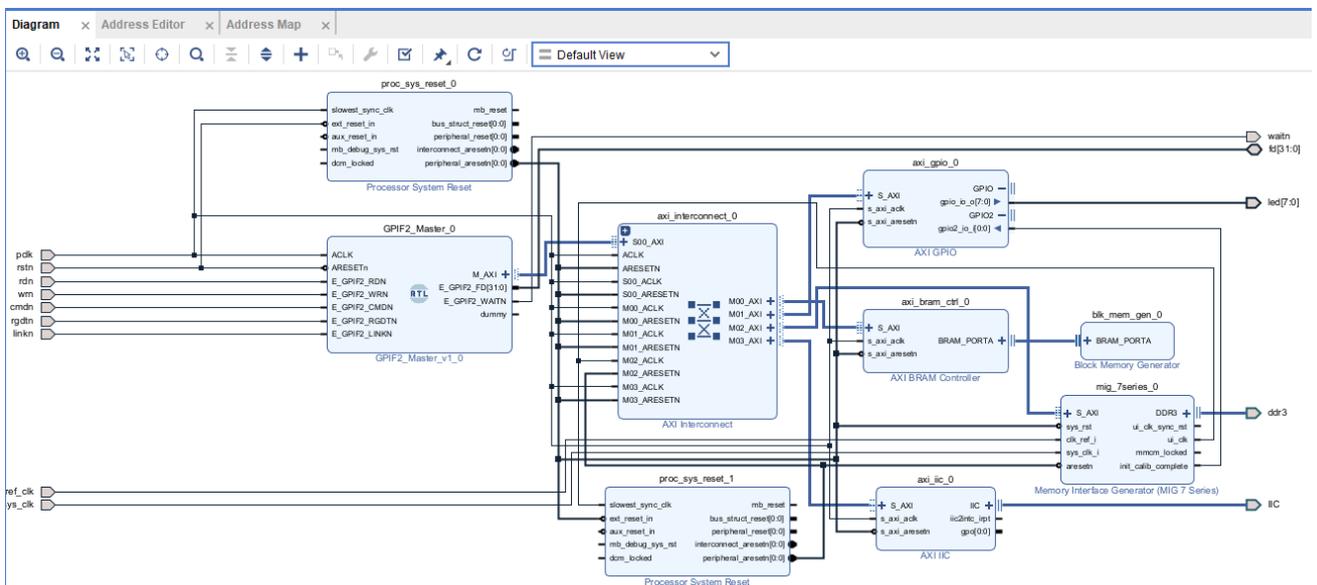
< 図 1.2. FIFO の Generate Output Products >

- 4 Flow Navigator の IP INTEGRATOR から「Create Block Design」をクリックしてエディタを起動します。
- 5 画面上で右クリック後「Add Module …」を選択し、GPIF2_Master を選択すると以下の画面になります。
これで USB 制御回路を AXI バスマスタとして利用できる準備が整いました。



< 図 2 . Block Design での GPIF2_Master 呼び出し >

- 6 GPIF2_Master の他に、AXI Interconnect IP を配置し、必要なスレーブ IP (IIC、GPIO、MIG、QSPI、BRAM 等) を追加して回路設計ができます。以下の例は、AX-Card7 システム開発ボードをターゲットに、DDR3 制御 (MIG) 、LED 制御 8 個 (GPIO) 、内蔵メモリ BRAM(32KB)、IIC を追加した例です。



< 図 3 . Block Design の完成 >

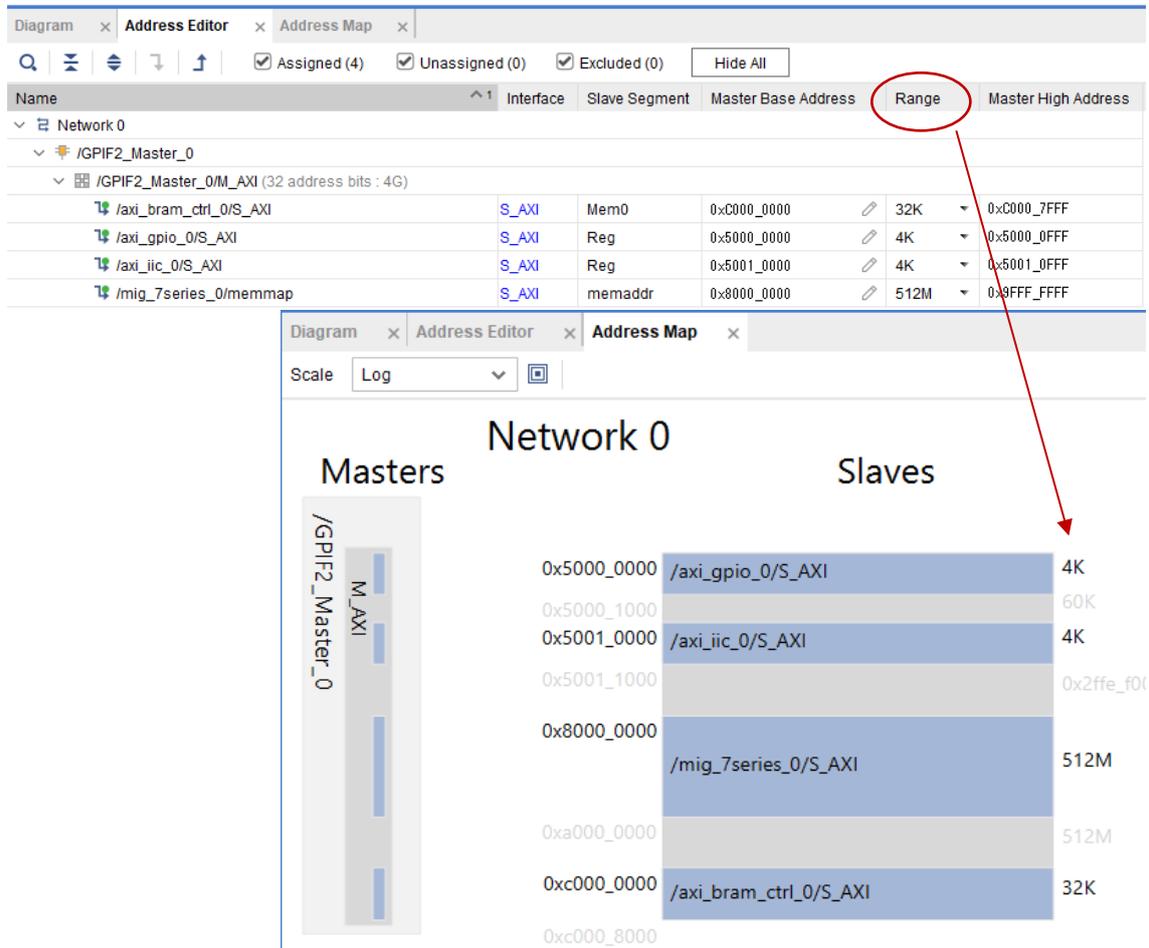
- 7 回路が完成したらキーボード「F6」キーを押して「Validate Design」を実行し、エラーのないことを確認後、「Generate Output Products」を実行してください。このとき、図 1.2 で示す画面と同じメニューが表示されるので、「Global」を選択してから Generate ボタンをクリックしてください。トップ回路を RTL 記述する場合は「Create HDL Wrapper…」を実行し、Block design をインスタンスとして利用します。

2.3 PC からの制御方法

従来の RefApp7 制御アプリを利用できません（23.2 月現在）。Python 等のツールで SUSlv.dll をインポートし制御します。

1 アドレスマップの確認

図 3 の画面でアドレスマップを確認します。



< 図 4. アドレスマップ >

LED を制御する GPIO 0x5000_0000、IIC 制御 0x5001_0000、DDR3 制御 0x8000_0000、内蔵メモリ制御 0xc000_0000 です。アドレス範囲は、Address Editor タブから変更できます。デフォルトで 64K です。

内蔵メモリの容量はこの画面で Range を選択することで変更できます。

2 レジスタ操作とメモリ操作の識別

従来の RefApp7 では IIC や LED を「レジスタ操作」、DDR3 や内蔵メモリを「メモリ操作」としていました。

レジスタ制御の場合、レジスタベースアドレス 0x1F_FFFF (d2097151)、メモリ制御の場合、メモリベースアドレス 0x1F_FFFE (d2097150)にそれぞれアドレスマップに示す先頭アドレスを書き込みます。

例えば、SUSlv_Reg_Write 関数で、レジスタ番号 0x1F_FFFF にデータ 0x5000_0000 を書き込んだ後、SUSlv_Reg_Write や SUSlv_Reg_Read で指定するレジスタ番号“0”で GPIO 制御できます。IIC ブロックは、オフセット 0x10000 なので、レジスタ番

号“65,536”で制御できます。IIC 内のサブレジスタは、それぞれ 65,536 + 各レジスタのオフセット がレジスタ番号となります。同様にメモリアドレス 0x1F_FFFE に 0x8000_0000 を WR すると、DDR3 メモリに対して SUSlv_Data_Write や SUSlv_Data_Read 関数で、データの書き込、読み出しができます。0xC000_0000 を設定すれば、内蔵メモリの BRAM (32kB) アクセスができます。

レジスタベースアドレス 0x1F_FFFF に 0x5001_0000 を WR すれば、IIC の先頭アドレスは、レジスタ番号“0”でアクセスすることができます。

メモリアドレス 0x1F_FFFE に 0xC000_0000 を設定すれば、内蔵メモリの BRAM (32kB) アクセスができます。

3. FPGA サンプルプロジェクト

AX-Card7 システム開発ボード向けに作成していますが、他のボードでもピンアサインの変更と DDR3 (MIG) を変更することで対応できます。

<無償ダウンロード>

https://www.prime-sys.co.jp/Download/Ref_Design/GPIF2M.zip

USB のデータ転送レート

RefApp7 の「転送レート測定」機能で計測した値と比較して、約 8%低下します。

PC 仕様 {Core I7-12700, 32GB RAM windows10-64bit, Intel H610 chipset} で運用した場合、RD/WR とともに約 340MB/s の転送レートです。

【備考】

2023.02.20 第 1 版発行

2023.04.27 第 2 版発行 (Generate Output Products のパラメータ設定変更)