

オリジナルプロセッサの開発支援のための基本入出力ハードウェアの提案

大柿 宏務[†] 大津 金光[†] 横田 隆史[†]

[†] 宇都宮大学 工学部 基盤工学科 情報電子オプティクスコース

1 はじめに

FPGA(Field Programmable Gate Array)の組み換え可能なハードウェアであり、それらが安価かつ大容量で利用可能になったことで、オリジナルの構成のプロセッサの開発が、研究や教育で行いやすい環境が整ってきた [1][2]. 開発者がプロセッサを実用的に動かそうとする場合、基本的な入出力装置が欠かせない. しかしプロセッサを作ることに興味を持っている開発者にとって、入出力装置を作るために更なる開発期間を要するのは、開発者のモチベーションを下げる原因になると考えられる. そこで本研究では、オリジナルプロセッサの開発者が、プロセッサそのものの開発に集中できるように、XILINX社のFPGAをターゲットとして、どのような構成のプロセッサに対しても汎用的に使える基本入出力ハードウェアと、できるだけ簡単に接続できるインタフェースを検討する.

2 基本入出力ハードウェア

プロセッサを動作させるための基本的な入出力装置として、プログラムやデータの転送およびデバッグを行うためのHost PC、プロセッサの操作を行うためのキーボードマウス、画面出力を行うためのビデオ出力を検討する. 本稿で想定する基本入出力ハードウェアを図1に示す.

基本入出力ハードウェアの要件として、FPGA内部のプロセッサの操作を外部のコンピュータに依存するのではなく、プロセッサを自身の制御の下で入出力を行うことをできることを想定する. プロセッサを動作させる際、外部から任意のプログラムをプロセッサにロードして、プログラムを実行し、その結果を確認することが必要になる. また、開発したプロセッサが意図した通り動作しない場合に、プロセッサをFPGAに実装した状態でデバッグが行える必要があると考える. これらプログラムやデータの転送、デバッグを行うためにHost PCとの接続が必要になる. FPGA上でプロセッサがユーザと直接情報のやり取りを行えるようにするため、キーボードとマウスとの接続の機能と、プロセッサで処理した結果をすぐに表示確認できるようにするためのビデオ表示機能が必要であると考え. これらの入出力装置をプロセッサと接続するためには、プロセッサと入出力装置との差異を解消するために入出力コントローラが必要になる. 例えば、情報機器間のシリアル通信でデータの送信をする際、パラレル信号をシリアル信号に変換、受信する際はその逆変換を行

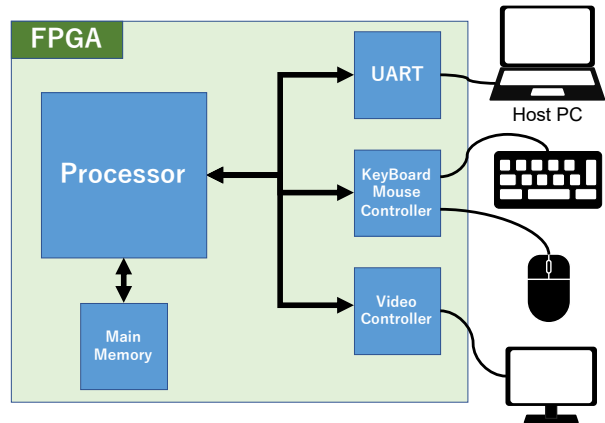


図 1: 基本入出力ハードウェアの構成

う UART や、キーボードのキー入力やマウスのクリック入力、移動幅の情報をシリアル信号で受け取ってパラレル信号に変換するコントローラ、ビデオ画面上のピクセルの水平位置情報、垂直位置情報と RGB の色情報を 1 ピクセルずつ順に出力するコントローラなどが必要になる. これらのコントローラを含む入出力コントローラは XILINX 社や DIGILENT 社が機能ブロックを IP 部品やソースコードとして提供している. これらの入出力コントローラとプロセッサを接続すれば、各入出力装置をプロセッサから使用可能となる. ここで、CPU とこれらの入出力コントローラをどのように接続するかが問題となる.

XILINX 社が提供している IP の UART[3][4] の接続インタフェースには AXI4[5] バス、とそれを簡略化した AXI4-Lite[6] バスが使用されている. AXI4-Lite は読み出しアドレス、読み出しデータ、書き込みアドレス、書き込みデータ、書き込み応答の 5 つのチャンネルをもち、それぞれのチャンネルでハンドシェイク通信を行うための ready と valid という信号と、それぞれのチャンネルでのデータ信号とアドレス信号、書き込みデータのストロブ信号、読み出し応答信号の合計 17 個の信号で構成されている. 読み出しと書き込み別々にハンドシェイク通信を行うことで、同時に読み書きを実現している. しかし、本稿で想定する基本入出力ハードウェアでは、必要となるバス転送性能が高くないため、読み書きの両方を同時に行う必要はないと考える. そのため、インタフェース仕様を単純化して、プロセッサ側に装備すべきインタフェースのハードウェア量を少なくした方がメリットが大きいと考えられる.

以上より、プロセッサに AXI4-Lite のインタフェース回路を持たせて、直接 IP 部品と接続させるのではな

Proposal to develop basic input/output hardware to support development of original processor

[†]Hiroomu Oogaki, Kanemitsu Ootsu and Takashi Yokota
Department of Fundamental Engineering, School of Engineering, Utsunomiya University ([†])

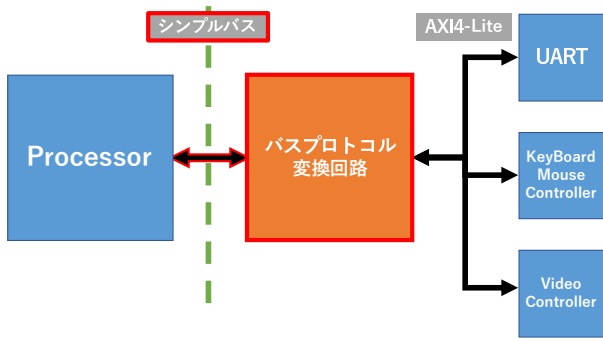


図 2: 基本入出力ハードウェアの構成

く、より簡単なバスインタフェースを用いてプロセッサと接続し、AXI4-Lite とのバスプロトコル変換回路を介して各入出力コントローラと接続する方式を採用する。図 2 に本稿で提案するシステム構成を示す。

3 プロセッサとのインタフェース仕様

バスプロトコル変換の回路の検討するにあたり、プロセッサとの接続がより簡単なバスインタフェース、シンプバスを考えた。図 3 にシンプバスの構成を示す。シンプバスは、読み出しアドレス、読み出しデータ、書き込みアドレス、書き込みデータ、データの指定を行う CMD、その CMD に対してバスのトランザクションがどういう状態かを知らせるための信号 STATE の合計 6 個の信号で構成している。CMD はデータの書き込みか読み出しかを指定する 2 ビットの信号である。STATE はバスの読み出し、および書き込みトランザクションが終了したことを知らせる 2 ビット信号である。想定した読み出しトランザクションは、読み出し先の入出力コントローラのアドレスをアドレスバスにセットし、同時に CMD でデータの読み出しを指定し、入出力装置からデータを読み出し、読み出したデータをデータバスにセットすると同時に STATE でデータの読み出しが終了したことを知らせる。書き込みトランザクションは、書き込み先の入出力装置のアドレスと書き込むデータをそれぞれのバスにセットし、同時に CMD でデータの書き込みを指定する。入出力装置にデータを書き込んだのち、STATE でデータの書き込みトランザクションが終了したことを知らせる。シンプバスインタフェースは、以上の CMD と STATE によりバストランザクションの制御を行う。

プロセッサの基本ワード長は、8 ビット、16 ビット、32 ビット、64 ビット等、選択肢があり、プロセッサの仕様に応じて変わる。それに伴ったバスプロトコル変換回路の変更を簡単に行えるように、バスプロトコル変換回路の HDL 記述の一部分を変更するだけで、回路全体に変更を反映できるようにする。実現方法としては、回路の記述を行うハードウェア記述言語 Verilog-HDL では可変部分をパラメータ化して記述することで、基本ワード長の変更を容易にする。図 3 のシンプ

シンプバス

信号名	ビット幅	方向	概要
CMD	2	1	書き込み/読み出しの指定
STATE	2	0	入出力装置の状態
RADDR	ADDR_WIDTH	1	読み出しアドレス
RDATA	DATA_WIDTH	0	読み出しデータ
WADDR	ADDR_WIDTH	1	書き込みアドレス
WDATA	DATA_WIDTH	1	書き込みデータ

・ CMD

R/W	概要
00	NOP : なし
10	READ : 読み出し
11	WRITE : 書き込み

・ STATE

STATE	概要
00	NOP : なし
10	READ DONE : 読み出し終了
11	WRITE DONE : 書き込み終了

図 3: シンプバスの構成

ルバスの表において ADDR と DATA はそれぞれアドレスバスとデータバスを表し、それぞれのビット幅を ADDR_WIDTH と DATA_WIDTH と指定していて、これデータバスとアドレスバスが可変であることを示す。

4 おわりに

本稿では、入出力装置の作るための開発期間を短縮するために、プロセッサと入出力装置が簡単に接続できるような基本入出力ハードウェアを提案した。その中で、AXI4-Lite をよりシンプなバスプロトコルに変換するハードウェアを前提に、プロセッサとの接続がより簡単なバスインタフェースの検討を行った。今後は、本稿で検討したインタフェースによってアクセス可能となる基本入出力ハードウェアの実装を進める。また、割込み制御機能についても仕様を検討し、実装する予定である。

謝辞

本研究は、一部 JSPS の科研費 20K11726 の援助による。

参考文献

- [1] 井澤裕司: "動かして分かる CPU の作り方 10 講", 技術評論社
- [2] 川田裕貴, 伊藤剛浩: "独自 CPU 開発で学ぶコンピュータのしくみ", 秀和システム
- [3] UART: "AXI UART 16550 v2.0 Product Guide (PG143)", XILINX, October 5, 2016
- [4] UART: "AXI UART Lite v2.0 Product Guide (PG142)", XILINX, April 5, 2017
- [5] AXI4: "Vivado Design Suite: AXI Reference Guide (UG1037)", XILINX, July 15, 2017
- [6] AXI4-Lite: "DS765 - LogiCORE IP AXI Lite IPIF (axi_lite_ipif) (v1.01a) Data Sheet (AXI)(DS765) (v1.01a)", XILINX, January 18, 2012