

小型ソフトコア基板用USBインタフェースの開発

植竹 大輔[†] 片下敏宏[‡] 中島俊夫[‡] 戸田賢二[‡]

[†]栃木県産業技術センター 〒105-0123 栃木県宇都宮市刈沼 367-1

[‡]産業技術総合研究所 〒305-8561 茨城県つくば市梅園 1-1-1 中央第2

E-mail: [†]duetake01@pref.tochigi.jp [‡]{t-katashita,skywalker-nakajima,k-toda}@aist.go.jp

あらまし 近年、ユビキタスコンピューティングの概念が提唱され、その研究が盛んに行われており、その実装の一つとしていわゆる“組込機器”の開発が注目されている。この組込機器の開発において、最近の多品種少量生産や機器の高機能化、製品サイクルの短縮化などの制限により、従来の開発手法ではコストが増大する傾向にある。その対応策として我々は組込機器の開発環境整備を進めており、本稿ではその一環として行った FPGA 上で動作する USB インターフェイスコアの開発について述べる。

キーワード ユビキタスコンピューティング、ソフトコア、USB

Development of USB Interface for small soft-core

Daisuke UETAKE[†] Toshihiro KATASHITA[‡] Toshio Nakajima[‡] and Kenji TODA[‡]

[†] Industrial Technology Center of Tochigi Prefecture 367-1 Karinuma, Utsunomiya-shi, Tochigi, 321-3224 Japan

[‡] National Institute of Advances Industrial Science and Technology

AIST Tsukuba Central 2, Tsukuba, Ibaraki, 305-8568

E-mail: [†]duetake01@pref.tochigi.jp, [‡]{t-katashita,skywalker-nakajima,k-toda}@aist.go.jp

Abstract Recently the ubiquitous computing is actively studied. The environment of developing embedded systems are emphasized. On the embedded device development, the cost is increased caused by short time to the market and the high performance demands. We plan to build the development environment of the embedded devices. In this paper, we describe developing USB interface environment: USB interface board, the interface core between CPU and USB chip, and device driver for uCLinux.

Keyword ubiquitous computing, soft-core, USB

1. はじめに

1989年にXeroxパロアルト研究所が提唱したユビキタスコンピューティングの概念は、情報機器の進化と普及につれ注目されている。人間が存在を意識せずに偏在し各々が連携して動作するコンピュータは今後更に増加をすると考えられる。

一方で、その実装の一つとしていわゆる“組込”もまた研究開発が盛んである。これは一義的に定義することは難しい言葉であるが、一般的には産業機器や家電などに内蔵され、特定の機能を提供するコンピュータ

とされている。

現時点の組込機器では、利用できるハードウェア資源は制限され、その機能も限られたものであるが、今後は、拡張のための外部接続バスへの対応や多様な機能の追加など、高機能化・多機能化によるよりいっそうの製品差別化が必要となると考えられる。このような多品種少量生産への対応には、高機能化・高付加価値の実現や、製品開発サイクルの短縮化も課題の一つである。これらを同時に満たすには、従来の開発手法ではコストが大きく増大してしまう。

組込機器開発を取り巻く環境も変化しつつあり、その一つに組込に用いるデバイスの高性能化低コスト化がある。ムーアの法則に示される様に、プロセッサやメモリなどの半導体製品の高性能化と価格の低下は進んでおり、以前と同程度のコストで高い処理能力を持つデバイスを利用できるようになっているため、更に製造コストを下げることも、同程度のコストで以前より高い機能の提供も、双方共に期待できる。

また、組込機器の受け持つ処理のなかでも、特に処理能力を必要とする機能を実現するためには、ハードウェアの利用が必要となるが、設計・試作や少数生産などのケースにおいては CPLD、FPGA のような論理プログラマブルデバイスを利用することが有効であると考えられ、これらのデバイスも技術の進歩とともに低廉化し利用しやすいようになってきている。ハードウェア記述言語によるハードウェア設計は、設計をよりいっそう抽象化し、設計期間の短縮や変更・更新を容易にするというメリットも享受することができる。

更に、高機能を実現するために必要な OS やハードウェアコアなどについても、オープンなものが有志によって開発されており、採用例も増えてきており高い信頼性を期待できるものが自由に利用できる。これらは開発の利便性を向上に寄与している。

そして、USB という接続規格がパーソナルコンピュータの普及と同時に広まってきている。これは活線挿抜の実現など、手軽に使用できる利点があり、最近では殆どの PC でチップセットへとコントローラが内蔵されている。普及による一般化により低価格化を実現し多種多様の対応デバイスが販売されている。その為、既成の製品機器へと対応出来るので、カメラ等の外部機器と接続し機能拡張を実現できる。仕様が一般に公開されており、コントロールチップやケーブルやコネクタなども安価で多数存在し入手性も良好なので、開発にも使いやすい。

そのような現状を踏まえ、本研究では開発環境整備の一環として FPGA の CPU コアで動作する USB の接続インターフェイスを開発した。

本論文では二章で組込機器を取り巻く環境について具体的に述べ、三章では作成した USB インターフェイスについて述べる。

2. 組込機器の開発における背景

本章では、組込機器の開発環境を取り巻く状況について述べる。

2.1. オープンハードウェアソフトコア

ソフトウェアの世界ではいわゆるオープンソースの伸張が著しく様々な分野において、ソースが公開さ

れ自由に利用できる極めて有用なソフトウェアが無数に公開する。

同様にハードウェアでも、オープンにされているものが多数存在している。

OpenCores.org(<http://www.opencores.org/>) などが代表例として挙げられる。他にも FPGA や ASIC の IP (Intellectual Property, 再利用可能な設計リソース) コアをオープンなライセンスの元に配布しているプロジェクトが多く存在し、CPU コアなどの大規模プロジェクトなども配布されている。種類も充実してきており、役立つ IP コアを利用することが出来る。

2.2. 組込用 OS

組込 OS には ITRON、Windows Embedded、VxWorks、Linux など多くのものが存在している。これらは利用対象が必要とする機能によって差別化されている。

ITRON

東京大学坂村健氏による TRON プロジェクトのサブプロジェクトの一つ。ITRON は Industrial TRON の略であり、自由に利用できる組込機器向けリアルタイム OS の仕様を策定している。実装としては、商用のものや、オープンソースとして配布されている TOPPERS などが存在する。

Windows Embedded

Microsoft Windows の組込向け OS である。必要な機能だけを選択することができ、ゲーム機や ATM、POS システム (Point of Sales System, 販売時点商品管理システム) などの OS などとして利用されている。

VxWorks

組込向けに Wind River 社がリリースしているリアルタイム OS。高信頼性を要する航空宇宙分野などで特に利用されているほか、幅広い用途に用いられている。

Linux

ヘルシンキ大学の学生であった Linus Torvalds 氏によるオープンソースとして開発されている UNIX 互換 OS。ソースを無償で公開し世界中のボランティアによって改良とメンテナンスが行われている。Kernel を再構築する際に機能を選択する事により、限定されたハードウェア資源の元でも軽快かつ安定して動作する。また、多くの環境へと移植されている。

なお、バージョン 2.6 から noMMU をサポートし、MMU を持たない CPU でも動作するようになった。

uCLinux

MMUを持たない CPU の組み込み用として開発された軽量小型の Linux バリエーションである。メモリ保護機能やプロセス毎の仮想メモリを持たず、システムコールが一部異なる(例えば fork が存在しない)が、多くの既存ソフトウェアが動作する、GNU ツールを利用したクロス開発が容易、などという Linux の利点は保持している。様々な CPU に対して移植が行われており、MMU を持たない組込開発においては活用されている。

2.3. 外部接続バス USB

USB(Universal Serial Bus)は 11Mbps(Full Speed - 1.1) または 480Mbps(High Speed - 2.0)を最大速度として機器間を接続する規格である[1]。最近のコンピュータに搭載されており、様々な対応機器が開発されている。

USB の機器間は Hub による階層構造で接続され、Master/Slave プロトコルを用いて機器間のコミュニケーションを行う。階層構造の中に Host デバイスの存在が必要な為、Slave コントローラしか存在しない場合は動作することが出来ない。

USB における通信は SOF(Start of Frame)を先頭とした 1ms 周期でなされており、これを一単位として Frame と呼ぶ。Frame は Transaction と呼ばれる通信単位で構成されており、SETUP、IN、OUT の種別を持っている(図2)。

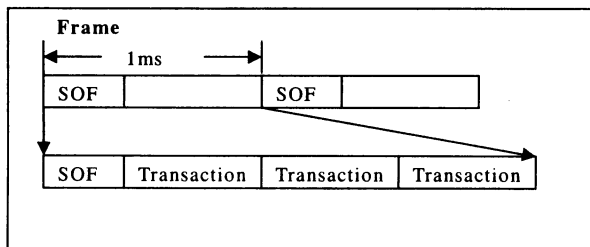


図1 USBのフレーム構造

SETUP	転送の準備
IN	ホストへのデータ転送
OUT	エンドポイントへのデータ転送

図2 Transactionの種別

また、USBの転送モードを図3に示す。Control、Interrupt、Bulk、Isochronousと四つの転送モードを持ち、それぞれが要求される動作に応じて適切なモードを選択して動作する。

- **Control 転送**
半二重通信、USBの基本的動作の為の通信
- **Interrupt 転送**
一定周期にデータ量小の通信。HID デバイスなど
- **Bulk 転送**
信頼性重視、時間保証無し。ストレージなど
- **Isochronous 転送**
再送なし、データ量保証。音声や映像など

図2 USBの転送モード

3. 設計と実装

以上述べてきたような、低コスト化、開発の迅速性、高性能・高機能化の要求とオープンハードウェアやオープンソフトウェアという流れを踏まえ、本稿では開発環境整備の一助としてFPGAのCPUコアで動作する組み込みOS上で利用可能なUSBインターフェイスコアの作成を行った。

今回の開発では USB コントローラを搭載し機器を接続する基板の設計、MicroBlaze を仲介する IP コアの開発、また、それを uCLinux 上で動作させるデバイスドライバの開発/改造を行う。

3.1. 作成環境

FPGA 開発基板は Suzaku (アットマークテクノ社)を用いることとした[6]。主な仕様を以下に示す。

FPGA	Xilinx Spartan-3(XC3S400 FT256)
動作周波数	3.6864MHz(FPGA 内部で通信)
コンフィギュレーション LSI	TE7720
Ethernet	10Base-T/100Base-TX
シリアル	UART 115.2kbps
基板サイズ	72×47mm

本基板は Spartan-3 上に [2] ソフト CPU コア MicroBlaze を搭載することで uCLinux を動作させることが出来る[3]。

MicroBlaze とは Xilinx 社より提供されている 32bitRISC ソフトプロセッサコアである。IBM 社の CoreConnect バスアーキテクチャが採用され OPB(On chip peripheral bus)や PLB(Processor Local Bus)などにより外部機器のインターフェイス設計が容易である。また Linux や TOPPERS/JSP などの各 OS の移植も進んでいる。MicroBlaze の開発環境は充実しており、容易に開発を行う事が出来るという利点がある。

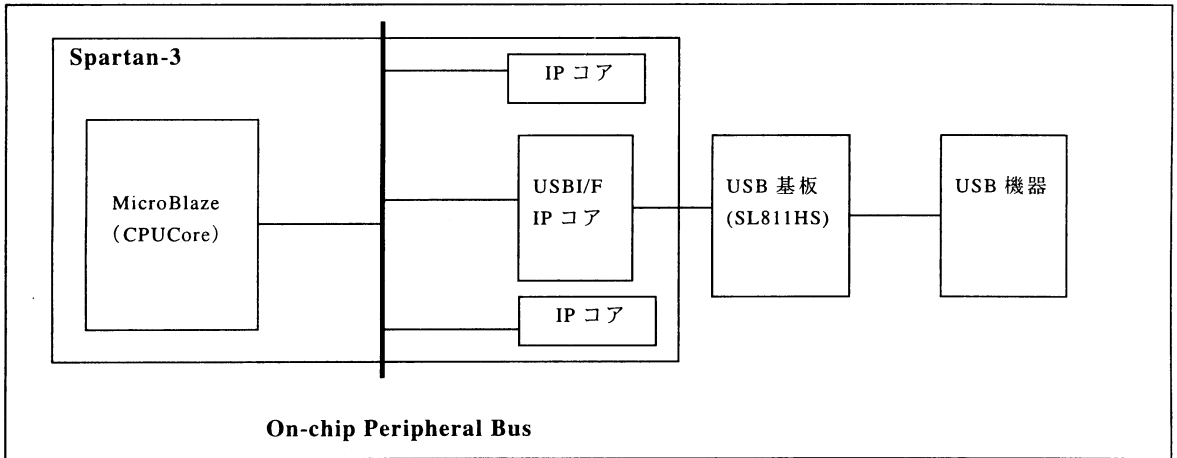


図4 USB インターフェイスコアの概念図

今回、ボード単独で動作する状況も想定し、外部デバイスを接続して動作可能なホストモードで動作することとした。コントローラはホストモードを持つ SL811HS(Cypress 社)を用いることとした。このデバイスはホスト/スレーブ両モードで動作し、SOF(Start of Frame)の自動生成などもサポートしている。SL811HS は 256 バイトの内部 RAM を持ち、00H-0FH までがコントロールレジスタ、10H-FFH がメモリバッファとなっている[5]。

3.2. 実装

基本的なデザインを構成したブロック図を図4に示す。MicroBlaze と IP コアは OPB で接続され、接続された SL811HS を制御し、取得したデータを OPB へと仲介するインターフェイスとして動作する。これを uCLinux から制御し USB 対応デバイスとして動作する事になる。OPB とは、IBM が開発した規格であり、MicroBlaze で利用されているものである[4]。

設計した内容で実際に試作を行った。

3.2.1. USB 基板の作成

基板は EagleCAD を用いて設計を行った。これはドイツの CadSoft Computer, Inc. がリリースしている商用 PCB-CAD であるが、非商用・評価目的において、層数と基板サイズの制限のみで無料で利用できるこれを用いて FPGA の外部ピンと USB コントローラチップを接続する回路ならびに基板を設計し、プリント基板を実際に作成した(図5)。

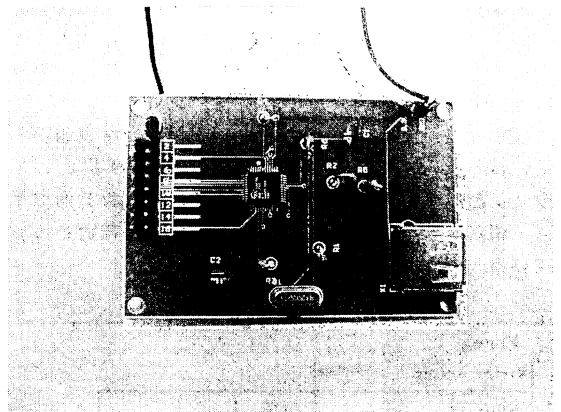


図5 試作したインターフェイス基板

3.2.2. ソフトコアの実装

インターフェイス部分のソフトコアは VHDL によって記述した。USB インターフェイス IP コアの動作はチップが要求するシーケンスで必要に応じレジスタやバッファに書き込み読み込みを行い、FIFO を介し OPB との通信を行う動作をする。

デバッグは chipscope で行った。これは Xilinx 社の提供するツールで、事前に指定したポートの挙動をサンプリングし結果を FPGA 内部の ram に蓄積することにより、PC で内部動作を監視することが出来るものである。これで入出力を確認しながら IP コアの開発を進めた。

3.2.3. Linux デバイスドライバ

デバイスドライバは PC 上に Linux の開発環境を用意し、カーネルのソースツリーの中に用意されているものを基にクロス開発を行った。

カーネルのソースツリーには SL811 のドライバが既に存在するので、ドライバから分離されているハードウェア依存部(asm/sl811-hw.h)を作成した。これはハードウェアへアクセスする部分の動作が記述されるべきファイルであり、uClinux 上から適切なアドレスへアクセスするようコーディングを行った。

4. おわりに

今後、ユビキタスコンピューティングとその実装としての組込が興隆するとともに、開発者は高機能かつ低コストでの製品製作を両立することを求められるようになると思込まれる。

我々は工数の削減と高機能化を同時に実現可能な環境整備を進めており、今回の USB 基板の開発は、その一環である。今回開発した基板データ、VHDL コードなど開発リソースのオープンでの公開を検討しておりこれらが製品開発コストの低減に寄与出来れば幸いである。

文 献

- [1] Compaq, Intel, Microsoft, NEC, Universal Serial Bus Specification, September 1998
- [2] Xilinx, Spartan-3 Datasheet, July 2004
- [3] Xilinx, MicroBlaze Processor Reference Guide, August 2004.
- [4] IBM, On-Chip Peripheral Bus Architecture Specifications Version 2.1, April 2001
- [5] Cypress SL811HS Datasheet, March 2002
- [6] 株式会社アットマークテクノ, SUSAKU Hardware Manual Version 1.0.1, June 2004