

USB2.0/1.1 機器エミュレーションシステムとその応用 —FPGA を用いた USB 機器総合開発の提案—

清水 喜弘[†]

[†] (株) 図研 SoC 事業部ビジネスデベロップメント課 〒222-8505 横浜市港北区新横浜 3-1-1

E-mail: [†] shim@zuken.co.jp

あらまし 近年の USB 機器開発においてソフトウェアはますます複雑になっているにも関わらず、開発期間は短縮を求められています。そこで我々は独自に開発した USB2.0/1.1 デバイスコントローラ IP コアと PCI IP コアを核に、これらに FPGA のプロトタイプ能力とソフトウェア技術を組み合わせることで USB 機器のソフト/ハードウェア開発を強力にバックアップする新しいシステム “EmuAid-USB” を開発しました。本稿では、USB 機器開発に携わるファームウェア技術者とデバイスドライバ技術者、さらにはハードウェア技術者のための USB2.0/1.1 ファンクションモデラ “EmuAid-USB” の開発事例とその応用について紹介します。

キーワード USB デバイス(ファンクション)開発支援, ソフト/ハードウェアエミュレーション技術, IP(USB2.0/1.1, PCI)

Proposal for USB2.0/1.1 Function Modeling System —an Application of FPGA to the USB Emulation System—

Yoshihiro SHIMIZU[†]

[†] SoC Business Unit, Zuken Inc. 3-1-1Shin-Yokohama, Kouhoku-ku, Yokohama 222-8505 Japan

E-mail: [†] shim@zuken.co.jp

Abstract Despite the increasing complexity of USB function system, market does not allow engineers to spend enough time for development. To satisfy market's requirement (i.e. engineers' requirement), brand-new system called “EmuAid-USB” has been released. Applying our original IP cores combined with our knowledge of FPGA prototyping and driver coding, “EmuAid-USB” became as an useful software/hardware development tool for USB function system. With this article, USB2.0/1.1 function modeler “EmuAid-USB”'s actual development example and its application are introduced, not only for the firmware and device-driver engineers, but also for the hardware engineers as well.

Keyword USB function system development support, Software / hardware emulation technology, IP(USB2.0/1.1, PCI)

1. 現状の問題点

USB 機器開発においてファームウェアは HW が無くては開発できず、さらに USB ホストドライバや USB ホストアプリケーションは HW とファームウェアができていなくては開発できません(図1)。よって否応無くシリアルな開発スケジュールを強いられます(図2)。

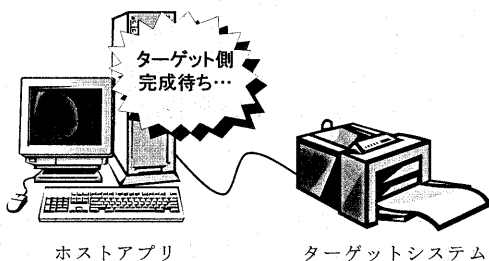


図1: ホスト側の開発が遅れる

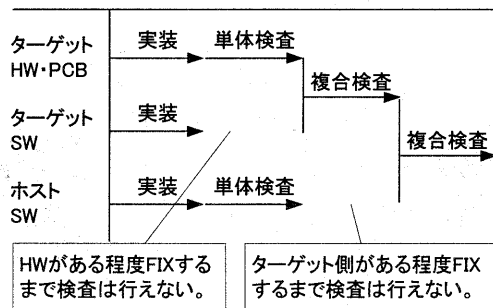


図2: スケジュール上の無駄

またターゲットシステムと USB ホストアプリケーションの接続性は、通常それぞれが独立してある程度設計された後、接続試験によって確認されます。この時点では、ターゲットシステムと USB ホストアプリケーションの品質が共に低いために、バグを発見した場

合の原因特定だけでもかなりの労力を必要とします(図3)。せめて片側だけでも品質が高ければ問題解決もスムーズになりますが、現状の開発現場ではこのような旧来の手法で開発が進められています。

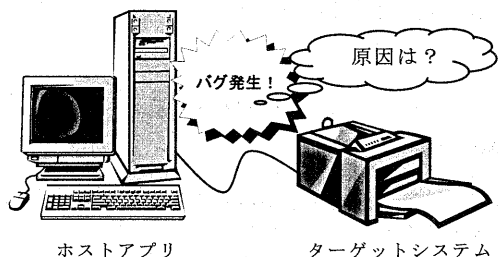


図3：デバッグの困難さ

本論文では、このような USB 機器開発における開発スケジュールやデバッグ手法の諸問題を FPGA を用いたシステムを利用することで解決する、USB 機器システムの新しい設計手法について提案します。

2. 解決策の模索

問題解決のためにはまずシステム開発の全体を、どのようにして並行して進めるかが課題となります。ファームウェアを容易に開発し、早期に USB ホストアプリケーションの開発に着手できる環境を構築しなくてはなりません。

そこで我々は、ユーザ所望のターゲットアプリケーションのモデルを Windows 上で動作させる環境を作成しました (EmuAid-USB)。EmuAid-USB で作成した Windows 上のファームウェアモデルから USB ターゲット機能を実装した PCI ボードを制御し、USB ホストに対して USB 通信の作用を与えます (図4)。この環境により、実機完成前に USB ホストアプリケーションの開発を進められます。

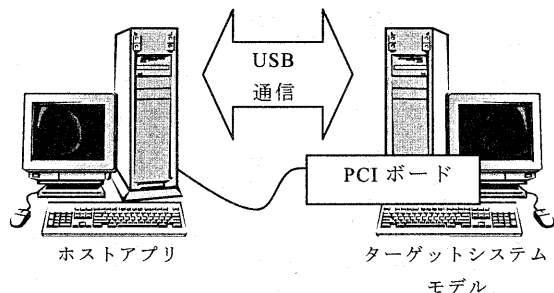


図4：ターゲットシステムのモデリング

また、この手法で設計した場合、実機評価の時点で USB ホストアプリケーションの品質はもちろん、ファームウェアの品質もこの手法を用いない場合に比べて

高くなっていることが期待されるので、実機評価の期間短縮が望めます(図5)。

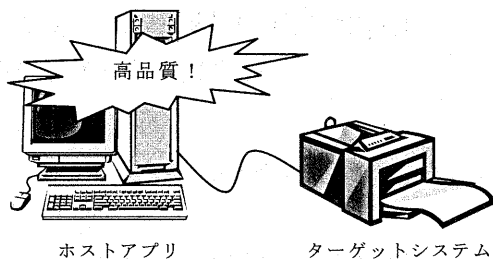


図5：開発初期段階のホストアプリ高品質化

さらに作成したモデルは実機評価実施時のリファレンスモデルとして大きく貢献します。問題が見つかった場合、リファレンスモデルと実際のターゲットファームウェアを比較することで、デバッグ作業を円滑に進められることが期待できます(図6)。

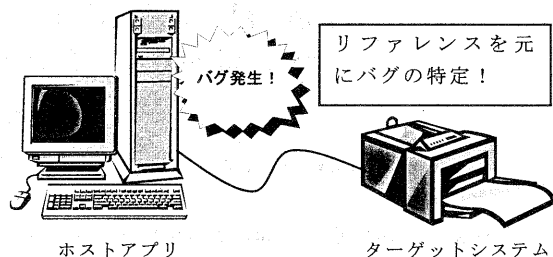


図6：開発後期のリファレンス

このモデリング環境を利用した場合、作業の並列化が実現し、スケジュール上のメリットが生まれます(図7)。旧来の手法(図2)との違いに注目してください。

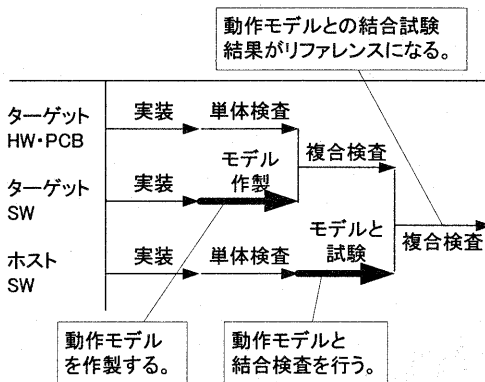


図7：スケジュール面のメリット

さて、このようなエミュレーションシステムを構築した場合の重要な問題として、実機システムとの同一

性が上げられます。USB プロトコルに関しては全く問題ありません。実機との唯一かつ大きな違いはデータ転送速度です。ファームウェアのモデリングに Windows を採用しているためファームウェアの割り込み遅延時間が実機に比べてかなり大きくなります。一般的な RTOS 環境では割り込み遅延時間を数 μs 以内で保証しています。しかしながら Windows は割り込み遅延時間を保証していません。我々の独自の調査では Pentium III 1.4GHz Windows 2000 日本語環境で $50\mu\text{s}$ ~ $500\mu\text{s}$ 程度の割り込み遅延時間が確認されました。つまり HW は実機と同様に動作しているものの、HW の 1 転送 (FIFO 1 つ分のデータを転送する) 間の時間がどうしても大きく開いてしまいます (図 8, 図 9)。

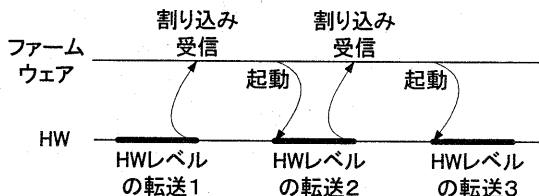


図 8 : RTOS 環境のデータ転送イメージ

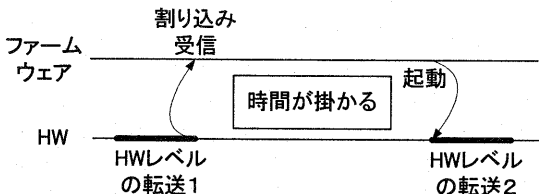


図 9 : Windows 環境でのデータ転送イメージ

しかしながらこの割り込み遅延時間が保証できないことによる弊害はありません。USB 機器は実際の使用環境では USB ハブに繋がれることが多いです。ユーザがハブに接続していないつもりでいても PC や USB ホストカード内部でハブが使われることもあります。USB 機器は USB ハブに接続されると確保できる通信帯域が狭くなります。つまり、USB 機器設計において速度を前提とした設計はしてはならないことが分かります (USB に限った話ではありません)。ちなみに EmuAid-USB は動作環境によって異なりますが 30-50MBps 程度の転送レートを出しています。よほどの特殊用途でない限り実機との速度相違による弊害はありません。

3. EmuAid-USB システム構成と詳細

以上で述べた、Windows 環境で実現する USB 機器エミュレーションのためのシステム、EmuAid-USB の機能詳細について説明します。

システムには USB と PCI の機能が必要です。USB デバイスコントローラ機能は図研製 USB2.0/1.1 コア (以下“Z-core USB^{※1}”)を採用しました。まず Z-core USB について説明します。

3.1. Z-core USB

汎用的に通用するエンドポイント構成を実現するために USB デバイスコントローラ機能は Z-core USB を採用しました。必然的に FPGA を利用することになります。FPGA を採用することで、ユーザ独自の回路を挿入したエミュレーション環境の実現という拡張性を持たせることが可能です。この点については第 5 節にて説明します。Z-core USB は図研が独自に開発した UTMI 規格準拠インタフェースを持つ USB2.0/1.1 デバイスコントローラ IP コアです。当然 USB2.0/1.1 デバイスコントローラの規格に則っていますが、加えて次のような特徴があります。

USB 規格では仮想的通信経路“エンドポイント”が定義されています。エンドポイントは本数(1-15)や性質 (Bulk, Interrupt, Isochronous)を設計するデバイスごとに自由にアレンジできます。Z-core USB はエンドポイントの本数と FIFO の段数やサイズといった HW 依存のエンドポイント設定をユーザが GUI から指定し、その指定に従った Verilog ソースコードを生成することができます (図 10)。またローカルバスインタフェースの性質も図研規格と AMBA バス対応から選択でき、さらにバス幅(16Bit, 32Bit)の選択もできます。

一般的な USB デバイスコントローラ IP ではエンドポイントの性質(Bulk, Interrupt, Isochronous)は HW ロジックに組み込まれていることが多いのですが、Z-core USB ではファームウェアから指定可能ですので SW による手軽さと自由度が大きいという特徴もあります。

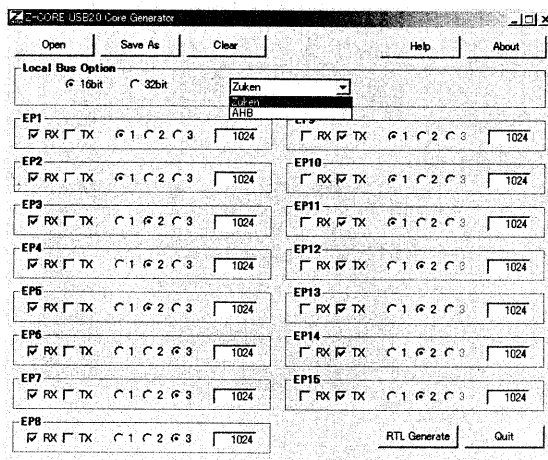


図 10 : Z-core USB Core Generator

3.2. EmuAid-USB システム概要

Z-core USB と同じく独自に開発した PCI コア (“Z-core PCI^{※2}”) を実装した FPGA, さらに USB 通信に必要な部品を備えた PCI ボードを開発しました (図 11~図 13). この PCI ボードを制御することで USB 通信部を動作させることができます.

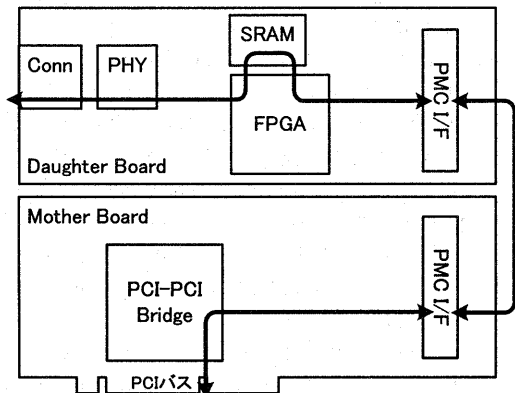


図 11: EmuAid-USB PCI ボード概要

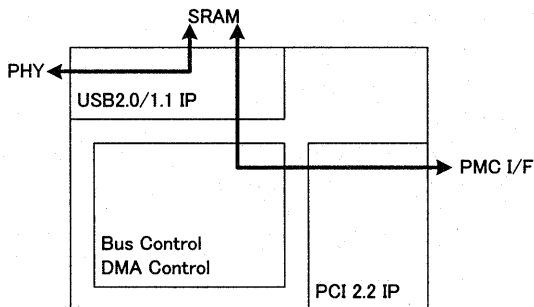


図 12: EmuAid-USB FPGA 内部ブロック図

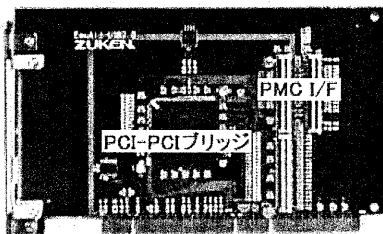
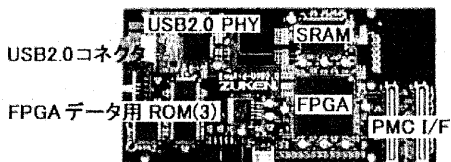


図 13: EmuAid-USB 外観

さらにこの PCI ボードを使用して Z-core USB を Windows 環境から制御する, API も併せて開発しました (図 14). USB 通信に必要な API を DLL として公開しています. ユーザはこの API を使ってエンドポイントの性質(Bulk, Interrupt, Isochronous)の指定やデータの送受信, コンフィグレーションルーチンを実装し, 手軽に USB2.0/1.1 デバイスの動作モデルを作成できます. API の使用において, HW 仕様や PCI ドライバ制御方法を意識する必要は全くありません. Win32 コンパイラ(Visual C++, C++ Builder 等)を用いて実機システムのモデルを簡単に作成できます.

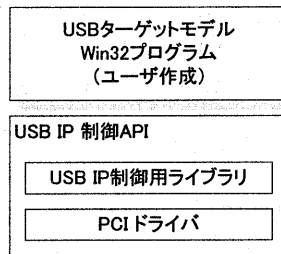


図 14: API 構成

出荷時のエンドポイント構成は以下のようになっています. よほど特殊な用途でなければこのエンドポイント構成で全く問題ありませんが, 必要があれば構成はすぐに変えられます. エンドポイント構成を変えた場合も API はそのまま使用可能です. なおエンドポイント 0 はデフォルトエンドポイントなので変更は出来ません.

EP 番号	方向	FIFO 段数	バッファサイズ
1-2	OUT	1	1024 Bytes
3-5	OUT	2	1024 Bytes
6-8	OUT	3	1024 Bytes
9-11	IN	1	1024 Bytes
12-15	IN	2	1024 Bytes

4. システム使用事例

ここでは EmuAid-USB を使用した USB 機器システム開発事例を紹介します.

4.1. 計測器分野

EmuAid-USB の販売当初は計測器分野で高い支持を得ることが出来ました. 計測器開発における現状の問題を示します.

- ・ 近年計測器の要求性能はますます高くなり, 計測器システムの HW が出来上がるまでの期間が非常に長い.
- ・ 計測器の高性能化に伴い, ホストアプリケーションも高性能化が求められている. しかしながら HW ができあがらないことはホストアプリケーションの検査は実施できない.

このようなユーザ要求に答えるために EmuAid-USB を利用しホストアプリケーションの開発を HW の開発と平行して進めることに成功しました (図 15)。

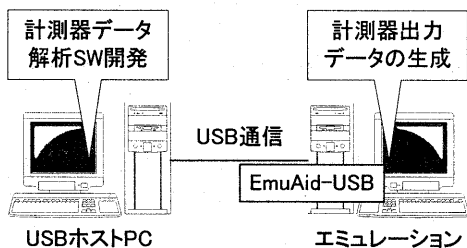


図 15: 計測器システムへの応用例

4.2. プリンタ等

実機システムでは USB ホストアプリケーション担当者が望む異常系の動作を実行することが難しい場合があります。EmuAid-USB を使用して異常系の動作を実行する (図 16) ことで、USB ホストアプリケーションの検査カバレッジ率を上げることが可能です。

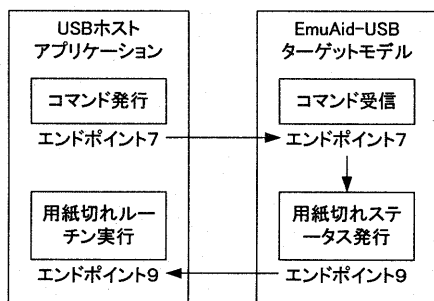


図 16: プリンタシステムへの応用

当然正常系の検査にも使用できます。ターゲットモデルは PC ですから、ターゲットモデルが受信したデータの期待値との比較や、ディスプレイ表示による画質確認、プリンタへの出力、ネットワークへの送信等ユーザのアイデアによる幅広い応用が可能です。

5. HW の拡張的課題

ここまでの議論で当初の USB 機器開発における諸問題を解決できます。しかしいくつかの新たな問題や更なる高度な要求が発生します。ここでは主に FPGA の特色を生かした応用について説明します。

5.1. 内蔵 FPGA を使用した HW 拡張

EmuAid-USB 上の FPGA にユーザ独自のロジックを挿入することで、HW 的拡張性を持たせたエミュレーションを実行できます。

セキュリティ機能を搭載した USB ストレージシステムを例に挙げます。暗号ロジックはデータの出入りに一番近い USB コアに直結した場合を想定します

(図 17)。USB ホストドライバが暗号ロジックを正しく制御できるかといった、ホストアプリケーションとファームウェアモデル、暗号ロジックの実機評価が可能になります。モデルプログラムをネットワークモデルにすれば、暗号ネットワーク USB 機器のエミュレーションも可能です。

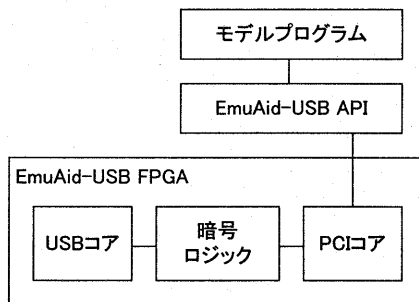


図 17: HW 拡張の例

5.2. さらに大掛りな HW 拡張

前述の EmuAid-USB 内蔵 FPGA を使用した HW 拡張では FPGA に搭載可能なゲート数の制限や、新たなインタフェースが必要な場合対応できないといった問題があります。

そこでエスケーエレクトロニクス株式会社の汎用 HW ロジックエミュレーション環境 VS-SYSTEM KM-1^{※3}を使用した拡張環境を準備しました (図 18)。KM-1 はベースボード上に汎用的な拡張性を持たせたバスを走らせ、そのベースボード上にユーザロジックを搭載した FPGA を載せてシステム検証が行えるプラットフォームです。EmuAid-USB 内蔵 FPGA では補い切れない HW 拡張性を KM-1 上に展開することでこの問題を回避します。

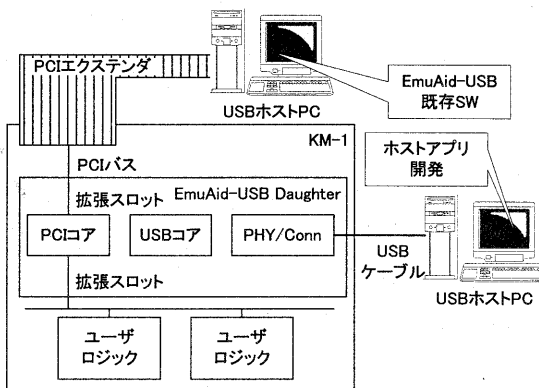


図 18: KM-1 を用いた EmuAid-USB 環境

このアイデアを用いた、HW ロジックまでも含めた

システムエミュレーションの例を示します (図 19)。
この例は画像処理システムのエミュレーションで、以下の機能を開発初期段階にエミュレーションできます。

- ・ 画像処理ロジックモジュールの実動作検証
- ・ ホストアプリケーションの画像表示検査
- ・ モデルプログラムの仕様妥当性検査

HW ができ上がっていない開発初期段階に、画像処理ロジックを実際に動作させながら USB 通信を行い、視覚的検証ができるメリットは多大了。

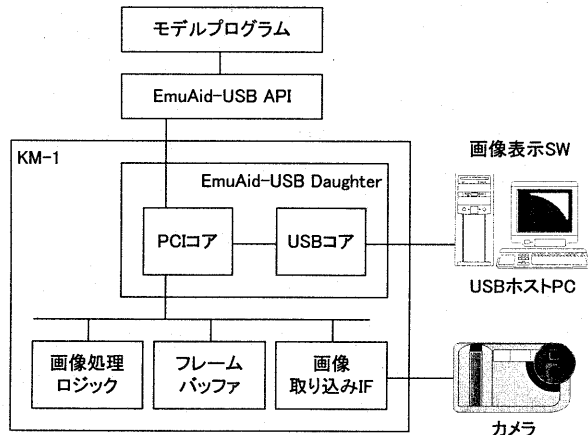


図 19 : KM-1 を用いた画像処理システム

6. SW の拡張的話題

EmuAid-USB は Windows 上でファームウェアのモデリングを実行します。しかしながら実機でのファームウェア作成において Windows を使用することはまずありえないと思われます。そこで EmuAid-USB API と同様のインタフェースを持った RTOS ミドルウェアを用意しました。

また今後はある USB クラスに特化したモデルプログラムをあらかじめ用意しておくことで、そのクラスの USB システム開発を支援する環境も提供します。

7. おわりに

USB 機器は、当然ながら USB ケーブルを介した通信を行うシステムです。EmuAid-USB は USB 機器を USB ケーブルの上流・下流に分けることで、新しい設計手法を確立することに成功しました。2002 年 10 月リリース以来数十ユーザに受け入れられ、明確な成果を出しています。

EmuAid-USB は使用するユーザのアイデアによって EmuAid-USB の開発者も予想しない様々な用途に応用できるシステムです。本研究会で発表することによって、新しく面白いアイデアを模索できればと願っております。

今後も更なる機能の拡張や新しい使用方法の提案を研究していきます。

文 献

- [1] Z-core USB 図研開発 USB IP コア
<http://www.zuken.co.jp/soc/doc/ip/index.html>
- [2] Z-core PCI 図研開発 PCI IP コア
<http://www.zuken.co.jp/soc/doc/ip/index.html>
- [3] KM-1 エスケーエレクトロニクス株式会社
<http://www.zuken.co.jp/soc/doc/verify/index.html>
- [4] USB Implementers Forum, Universal Serial Bus Specification Revision 2.0, April 27, 2000
- [5] Microsoft, Windows 2000 Driver Development Kit, June 28, 2000