

FPGA を使った実践的システム設計の世界へようこそ

こうして“コンピュータ・システム技術 学習キット”は完成した！

デバッグが完了したある夜...

“コンピュータ・システム技術学習キット”(写真1)を見ながら...

編集者(以下“編”): いやぁ～、ついにここまで来ましたね！

筆者(以下“筆”): 来ましたねぇ～。

編: 絵も音も出るし、ストレージもネットワークもつながるし...。ちょっとやりすぎましたかね？

筆: いやぁ～それくらい機能が付いてないと、“オリジナル仕様パソコン”にならないよ(笑)。まぁ～パソコンを作るかどうかは置いて、今でも現役で使えるバスやインターフェースでないと、評価ボードとしてすぐに限界がきて、“おもちゃ”で終わっちゃうよ。

編: そうですね。

筆: アナログRGB回路とか、過去に何度も設計してるけど、やはり最初にあがってきた試作ボードに火を入れて、正常に画面が表示されたときはやっぱり感動するよね。

編: 自分で設計した回路が、すぐに目の前で動かせるというのは、FPGA の大きなポイントですよ。その感動をすべての読者にも感じて欲しいと思います。

バスの違いによるパフォーマンスの比較

編: さて、次は上に載るオプションCPUボードの準備ですね。

筆: 編集部経由で手配してもらった (CPU 名称はまだ秘密) の試作は、来週から始めるからもうちょっと待っててよ。

編: よろしくお願いします。そういえばそのCPUを使って、

PCIバス側からバースト転送した場合と、CPU内蔵DMAコントローラを使ってローカル・バス側からバースト転送した場合で、どちらがパフォーマンスがいいでしょうね。

筆: そういうときこそ、この学習キットを使って評価して欲しいね。FPGAはPCIバス側からもローカル・バス側からも32ビット・フルにつながってるから、このボードで実際にそれぞれの構成でパフォーマンス比較ができるし。

編: そういえばそのCPUはローカル・バスでもMPXモードがありますよね。通常モードとMPXモードとPCIの三つともえの比較はおもしろそうですね。

筆: しっ！MPXモードをもつCPUは数少ないから何のCPUかバレちゃうよ(笑)。

FPGA対応ソフトCPUコアいろいろ

編: オプションCPUカードを載せなくても、この学習キットだけでFPGA内にソフトCPUコアを入れても動きますよね。

筆: もちろん。デバイスがXilinxなんで、まずは順当にいったMicroBlazeだろうね。

編: ほかに、ある筆者さんにオリジナルCPUの設計をお願いして...。もちろんCPUだけでなく、Cコンパイラやデバッグまで準備してもらおうのが目標です。

筆: 『オリジナルのCPUを作りました！』という記事はよく見るけど、ソフトウェア開発環境まで揃ったものはなかなかないからね。期待してますよ。

振り返れば...

編: Iさん(筆者)に最初書いてもらった原稿って、1997年10月号特集のCPLDでPCIデバイスを設計する記事でしたよね。

筆: 懐かしいねぇ～。あの頃は数百マクロセルのCPLDで、ひい～こら言いながらPCIを設計してたんだなぁ～。

編: それが10年足らずで、100万ゲートのFPGAがポンポン使えてしまう時代になった...。

筆: バースト対応バスマスタ入れても、PCI部分の使用率が10%とかだもんね。笑っちゃうよね。

編: それからSDRAMコントローラだの、IDEコントローラだの、いろいろなバスやインターフェースの設計記事を書いていただきました。

筆: それらを特集1回でまとめたよね。

編: はい。2003年1月号の『作りながら学ぶコンピュータシステム技術』(図1)です。あれからもう3年ですよ。

筆: 早いねぇ～。ついこないだのような気がするよ。あの特集の敗因は...



写真1 “コンピュータ・システム技術学習キット”

本学習キットの詳細はAppendix2を参照

編：ま、負けたんですか!?. あの特集は大好評でものすごく売れたんですよ(汗).

筆：いやそういう意味じゃなくて(笑), あのシステム構成だと複数枚数のPCI 評価ボードを使わざるを得なかったから、トータルとしてコストが高くなっちゃう点が問題だったということで...

編：その反省点を踏まえて、新たにこのボードを設計したということですね.

筆：設計したもなにも、このボードの仕様にバンバン注文してきたの、そっちじゃん.

編：そ、それを言われると...(汗). 次から次へと無理な注文に応えていただき、本当にありがとうございます.

オリジナル仕様パソコンを“創る”ぞ!

筆：最近、昔の8ビット・パソコンをFPGAで復元(?)させて、Webで公開してる人がいるよね.

編：よく見ますね. 実は編集部にもその手の投稿の話もあるんですが、BIOSやソフトウェアのライセンスに問題があるので、さすがにうちでは取り上げるのは難しいですね.

筆：そうすると、やはり新規にオリジナルな仕様で設計するのが一番だね. このボードを使ってPCIできました、HDD読み書きできました、Ethernetがつながりました...だけじゃつまらないから、最終的にはオリジナル仕様パソコンを“創る”ところまでいかないとね!

シリアル・コントローラくらいささっと作れるように!

筆： 세미나とかの講師をしてるんだけど、組み合わせ回路でLEDを光らすぐらいはできるんだけど、何個かの状態をもったステート・マシンの設計ができない受講生が多いよね.

編：そうなんですか.

筆：あと、同期回路設計でクロックのエッジ検出とかがわからない人. 直前のクロックでの状態をフラグにでも保存しておいて、現在の信号の状態と一致していれば変化なし、一致しなければ変化ありで、それがエッジを検出したことになるってことに、気がつかないみたい.

編：クロック・エッジで動作するというと、VHDLだと、

```
if (CLOCK' event and CLOCK = '1') then
    を使うことしか頭に思いつかないのかな?
```

筆：だから、バースト転送対応PCI イニシエータとか、ウルトラDMA 対応 IDE コントローラとかの解説より、もっともあっと基本的なコントローラの設計方法の解説から始めないとダメかなぁと...

編：シリアル・コントローラくらいにシンプルなやつからですか?

筆：本当はシリアル・コントローラだって奥が深いんだけどねえ~. データ・ビットを真ん中の1回だけでサンプリングするんじゃなくて、複数ビットをサンプリングして多数決とるとか.



図1 2003年1月号特集『作りながら学ぶコンピュータシステム技術』CPUにSH-4を使い、PCIバス上に各種インターフェースをインストールしてオリジナル・パソコンを構築するまでを解説

本特集の内容について

編：じゃ次の特集は、シリアル・コントローラの作り方を、ステップ・バイ・ステップで徹底的に詳しく解説するというのはどうでしょうか.

筆：そうですねえ~. たとえば手動操作でクロックを与えて、8ビットLEDの表示が1ビットずれて表示されるのを目視で確認するとか...

編：それいいですね! すごく短いHDLソースをベースにその動作を確認したら、次は一つ機能を追加してその動作を見て、さらに機能を追加して...、そして最終的に調歩同期式シリアル・コントローラを作るという展開ですね.

筆：でも、そうすると、それだけでかなりページ取っちゃうよ. この学習キットはほかにもいっぱいコネクタ載ってるのに...

編：それはそうですが、1回の特集ですべてのインターフェースを解説するのはハナっから無理です. なので、最初はもっとも基礎的で汎用的で一般的にどこでも使うシリアル・コントローラを解説しましょう. あともう一つ取り上げるとしたら、何がいいですかね?

筆：まあ~ 読者が興味を引くとしたら、見た目にも派手? なアナログRGB出力かな.

編：アナログRGBだと...ディスプレイがちゃんと同期するとか、赤だけ表示してみるとか...そういうレベルからの解説ですね.

筆：そうそう. カラーバー表示は重要!(笑).

編：では次の特集はそういう方向で...で、締切りは 日ですよ!

筆：何い~!?

P

1

2

3

App1

4

5

App2

6