

シンプルな計算機システムの開発に向けた挑戦

吉瀬 謙二^{1,a)}

受付日 2012年11月5日, 採録日 2013年4月5日

概要: プロセッサを含むハードウェア, オペレーティングシステム, ライブラリ, アプリケーションといった多くの層の横断的な教育を支援するプラットフォームの提供を目指して, MieruPC システムと呼ぶ「シンプルで分かりやすい計算機システム」の開発に取り組んでいる. 本システムでは, MieruPC の名前の由来となる「見える」という言葉のとおり, ハードウェアとソフトウェアの記述がよく見えてかつ変更できること, その開発プロジェクトでは, スタンドアロンで動作する魅力的な計算機とその開発環境を提供することを目指している. 本論文では, MieruPC システムを開発するプロジェクトの詳細, 工夫した点および得られた知見をまとめる.

キーワード: MieruPC, FPGA, 計算機システム

A Challenge to Make a Simple Computer System

KENJI KISE^{1,a)}

Received: November 5, 2012, Accepted: April 5, 2013

Abstract: The education across layers such as hardware, operating system, library and application is important. In order to support such education, we are developing “a simple and intelligible computer system”, named MieruPC system. In this system, as the Japanese word “MIERU” used as the part of MieruPC, the description of hardware and software can be clearly seen, and can be modified easily. In this paper, we describe the concept, the challenge, the implementation of MieruPC system, and the acquired knowledge through the project.

Keywords: MieruPC, FPGA, computer system

1. はじめに

計算機のハードウェアとソフトウェアの両面で深い知識を持つ技術者が求められている. そうした技術者の育成には, オペレーティングシステム (OS), アプリケーション, プロセッサを含むハードウェアといった多くの層の横断的な教育が有効である. また, そのような教育を支援する扱いやすいプラットフォームが必要とされている.

そのようなプラットフォームの提供を目指して, **MieruPC** システムと呼ぶ「シンプルで分かりやすい計算機システム」の開発に取り組んでいる. 本システムでは,

MieruPC の名前の由来となる「見える」という言葉のとおり, ハードウェアとソフトウェアの記述がよく見えてかつ変更できること, その開発プロジェクトでは, スタンドアロンで動作する魅力的な計算機とその開発環境を提供することを目指している. ここで, 本プロジェクトにおける「魅力的な計算機」を, 1980 年代に普及した MSX や PC-6601 を例とするパーソナルコンピュータに匹敵するディスプレイ, 入力装置, メモリ, 不揮発性の記憶装置を有して, グラフィカルなアプリケーションを作成できる計算機と定義する.

魅力的な計算機システムを手軽に扱える複雑さとコード量で実現し, 「計算機システムを理解できる, 作れる, 拡張できる」を体験するプラットフォームを提供することがプロジェクトの狙いである.

開発するスタンドアロンで動作する計算機を MieruPC

¹ 東京工業大学大学院情報理工学研究科
Graduate School of Information Science and Engineering,
Tokyo Institute of Technology, Meguro, Tokyo 152-8552,
Japan

a) kise@cs.titech.ac.jp

と呼ぶ。これに、OS、ソフトウェア開発環境 (SDK)、シミュレーション環境が加わり MieruPC システムとなる。

MieruPC は、FPGA ボード、マザーボード、液晶ユニット、キーボードなどで構成される。このハードウェアの上で、MieruOS と呼ばれる独自開発の OS が動作して、ファイルシステムやアプリケーションの実行を管理する。また、ソフトウェア開発環境である MieruSDK とハードウェアの動作を模倣するソフトウェアシミュレータを提供することでアプリケーションおよびシステムソフトウェアの開発を支援する。

本論文では、MieruPC システムを開発するプロジェクトの詳細、工夫した点および得られた知見をまとめる。本論文は、文献 [1] を修正したものである。2 章でプロジェクトの概要を説明する。3 章ではハードウェアに、4 章ではソフトウェアに焦点を当てる。5 章で議論し、6 章でまとめる。

2. MieruPC プロジェクト

2.1 プロジェクトのはじまり

「Computer Science を勉強しているのですが計算機システムを製作することはできますか?」という素朴な疑問に答えることを目的の 1 つとして、2008 年 4 月より、「MieruPC プロジェクト：中身の見える計算機システムを構築する教育・研究プロジェクト」を立ち上げ、スタンドアローンで動作する計算機としての MieruPC の開発を開始した。

図 1 に、2008 年 4 月に著者が描いた MieruPC のイメージを示す。この図には、重要な技術的な選択の結果が書き込まれている。モニター体型のデザイン。USB ではなくシンプルな PS/2 インタフェースのキーボードによる操作。プロセッサなどの主要なハードウェアの FPGA ボードへの実装。3.5 インチ程度の小型カラーディスプレイの搭載。SD カード (マルチメディアカード) によるプログラムやデータの読み書き。

このように列挙すると、普通の構成のように感じるかもしれないが、これらの選択が一般的だったわけではない。

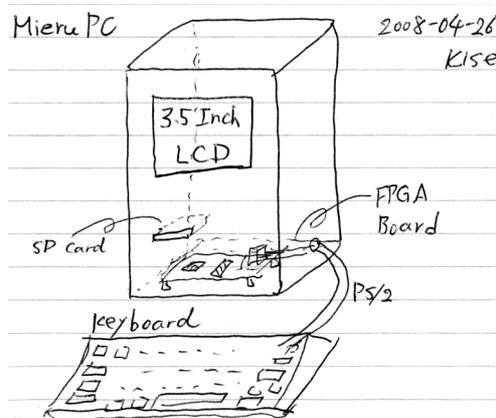


図 1 2008 年 4 月に描いた MieruPC のイメージ
Fig. 1 The image of MieruPC drawn in April, 2008.

著者の所属する学科では、学部 3 年生を対象とする実験において、マイクロプロセッサを FPGA に実装する。そこで用いるハードウェアを図 2 に示す。7 セグメント LED などに情報を表示して、その動作を確認する。このようなハードウェアを用いた実験は有意義なものであるが、計算機として見たときには出力装置の魅力に乏しい。FPGA ボードの VGA 端子を用いて、汎用のディスプレイに出力するという選択肢もあるが、ディスプレイのサイズが大きくなるため好ましくない。そこで、3.5 インチ (320 × 240 画素の液晶ユニット) 程度のカラーディスプレイを用いることにした。

計算機の内部や動作中の LED がよく見えるように、ケースには透明のアクリル板を用いることにした。図 1 のデザインに関しては、段ボールやアクリル板を加工して試作機を作成 (図 3) して検討した。これにより、モニター一体型ではなくセパレート型のデザインに落ち着くことになる。セパレート型としたことで、液晶ディスプレイを含む液晶ユニットと FPGA カードなど格納する MieruPC 本体が必要となった。教育の現場では机上のスペースが限られることから、手のひらにのる程度の大きさの計算機とした。

アプリケーションの 1 つとして、テキストエディタを用いる文章作成を想定し、キーボードが必要と考えた。USB 接

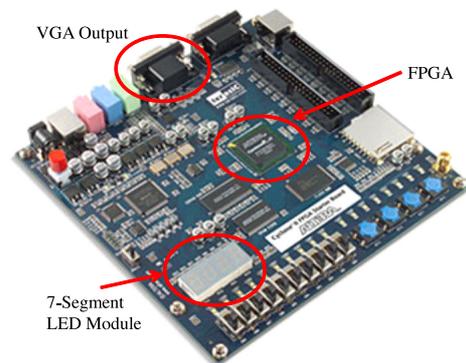


図 2 教育用の実験で用いられる FPGA 搭載ハードウェアの例
Fig. 2 Sample hardware with FPGA used in an experiment for education.

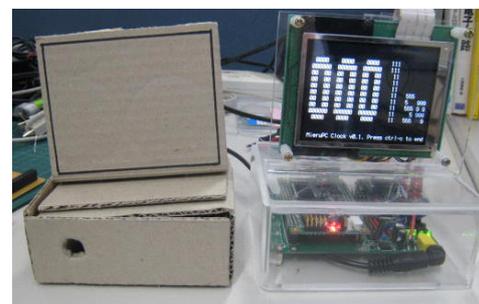


図 3 段ボールで作ったデザインの模型 (左) とアクリル板で作った試作機 (2008 年 7 月撮影)
Fig. 3 The design made from corrugated paper (left) and a prototype system (photography July, 2008).

続のキーボードが一般的だが、シンプルで分かりやすい計算機を目指すという方向性から、シンプルに制御できる PS/2 接続の市販されているキーボードを用いることにした。

近年の計算機システムでは主にマウスを用いた GUI (Graphical User Interface) を提供することが一般的である。しかしながら、シンプルなシステムでは GUI は不要と判断して、マウスの採用は見送った。

図 1 の FPGA ボードには、左端に FPGA が、中央にメインメモリとして利用する SRAM が描かれている。著者が計算機アーキテクチャを専門としてプロセッサ構成などを講義で扱っていることもあり、プロセッサの構成を変更できる計算機システムを作りたいという思いがあった。そのため、電子工作などで用いられる PIC (Peripheral Interface Controller) や SH (SuperH) などの組み込みマイコンではなく、FPGA を用いてハードウェアを実現することにした。

スタンドアロンで動作する計算機では、不揮発性の記憶装置が必要となる。スマートメディア、メモリスティックなども候補となったが、コンパクトなサイズとシリアル転送によるシンプルなデータ転送方式 (Serial Peripheral Interface, SPI モード) を提供していることから SD カードを用いることにした。

このような選択により方向性を定めて、プロジェクトを開始した。

2.2 プロジェクトのめざすところ

MieruPC プロジェクトでは、複数の大学の教員および学生が自由な意志で参加する形で開発を進めた。そのため、ゴールを見失わないように、箇条書きの明確な目的を設定している。それらを列挙する。2012 年 6 月時点において、達成したものに●、ある程度は達成したものに▲、達成していないものに□を付けている。

- ハードとソフトのすべてが見えるスタンドアロンで動作する計算機システムを作る！
- FPGA (あるいは VDEC^{*1} のチップなど) を活用して CPU の内部も見える・修正できる計算機システムを構築する。
- ▲ 大学 2, 3 年生が理解できる洗練された計算機システムの実現を目指す。
- ▲ 5 万行以内のテキスト形式で洗練された計算機システムを実装する。
- ▲ 必要最小限のシンプルなハードウェアで実現する。低価格で購入できる計算機システムとする。
- ▲ シンプルで洗練されたソフトウェア群を提供する。
- テキスト版テトリスが動く計算機システムを実現する。
- エディタでテキストファイルの編集ができる計算機シ

ステムを実現する。

- 新規性がない方が好ましい。開発は基本的に既存のものを参考にしながらゼロから構築する。
- ▲ 成果はフリーとしてどんどん公開する。参加したい人は歓迎する。どんどん担当してもらう。
- 貢献した人を讃える。論文をたくさん執筆する。ハードウェアを販売する。あまり厚くない本を出版する。
- 日本の情報工学を活性化する。日本発の MieruPC を世界に広める。

最後の 2 項目については達成していない。これらは長期的な目的であり、今後の課題である。

ある程度は達成できた▲の項目については、不足している部分を太字とした。3 番目の項目に関しては、大学 2, 3 年生が理解できるまでソースコードの記述が洗練されていない。同様に、4 番目、6 番目の項目に関しては、十分に洗練されていない。5 番目は販売価格が低価格とはいえない。10 番目はプロジェクト管理の難しさに起因する。その他、●とした 5 つの項目は十分に達成できた項目である。

2.3 プロジェクトのいま

プロジェクトの成果として、2012 年 6 月時点の最新版となる MieruPC-2010 システムを公開および販売している。図 4 は、画像を表示し、その上にキャラクタを表示するサンプルアプリケーションを動作させている MieruPC-2010 の写真である。MieruPC-2010 のサイズは 130 mm × 95 mm × 135 mm で、手のひらにのる計算機を実現している。

プロジェクトで構築したハードウェアを販売するために、2009 年 3 月に MieruPC 株式会社を設立した。商業的な成功とはいえないが、これまでに、100 台を超える MieruPC を出荷しており、これらは、様々な大学において活用されている。

開始時に掲げた目的の 40%以上を達成していることもあ



図 4 サンプルアプリケーションを実行中の MieruPC-2010
Fig. 4 A snapshot of MieruPC-2010 running sample application.

*1 東京大学大規模集積システム設計教育研究センター

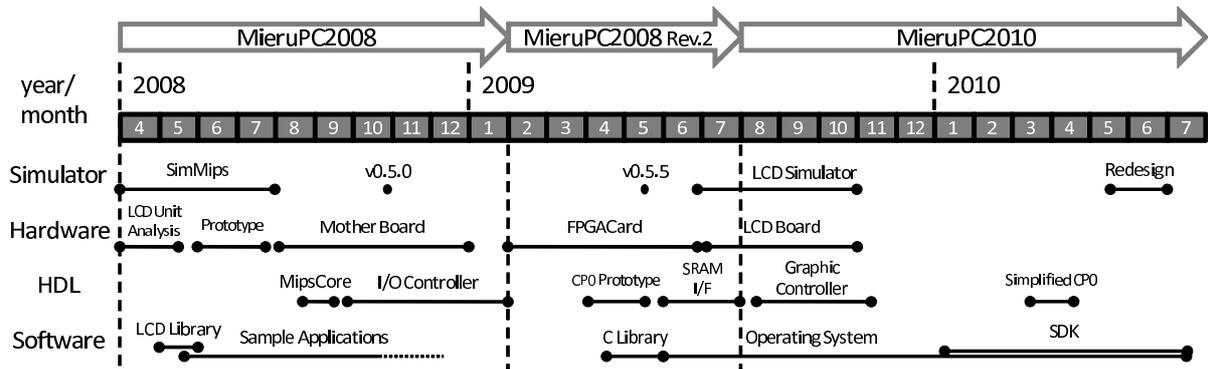


図 5 MieruPC プロジェクトの主な開発の履歴
 Fig. 5 The main development history of MieruPC project.

り、ハードウェア開発を含むプロジェクトとしては成功例の1つである、と主張したい。

2.4 プロジェクトのあしあと

図 5 に、本プロジェクトの主な開発履歴を示す。横軸を時間として、縦軸にはシミュレータ、ハードウェア、HDL、ソフトウェアに分類された開発内容の概要をまとめている。開発は大きく4つのフェーズに分類できる。

第1フェーズは2009年1月までの10カ月間であり、その成果はMieruPC-2008として製品化されている。第1フェーズでは、プロセッサシミュレータ SimMips [2] の実装、それをベースに開発した MIPS 命令セットに準拠する独自アーキテクチャのプロセッサ **MipsCore** と I/O コントローラ*2 というハードウェア開発と、対応するソフトウェアの開発、FPGA ボードや I/O コネクタを搭載するマザーボードの基板開発および試作機の開発などを行った。2008年7月時点の試作機が図3(右)である。

第2フェーズは2009年2月から7月までであり、その成果はMieruPC-2008 Rev.2 (Revision 2)として製品化されている。ここには、独自仕様のFPGAボードの開発、システム制御コプロセッサ (MIPS の CP0 に対応) の検討、C ライブラリの構築などが含まれる。

第3フェーズは2009年7月から2010年7月までであり、その成果はMieruPC-2010として製品化されている。ここでは、液晶ユニットの開発、簡略化されたシステム制御コプロセッサの搭載、独自仕様のOSの開発、MieruSDKの構築などを行った。

2010年8月以降が第4フェーズであるが、本論文ではこれには言及しない。

ここで示した各フェーズの進捗は予想を上回るものであった。「プロジェクトではスタートダッシュが大事」という思いから、開始時点から多くの学生を巻き込んで、様々

な要素技術の開発を迅速に行うように調整した。そして、何よりも、優秀な学生が実力を発揮したことが迅速な開発の鍵であった。

3. MieruPC システムのハードウェア

3.1 既製品から独自仕様のハードウェアへ

第1フェーズにおけるMieruPC-2008の開発では、既製品のハードウェアを多用して、主要な要素を迅速に開発した。

FPGA ボードとして、Spartan-3E XC3S1200Eを搭載するSUZAKU-Sを採用した。これに搭載されているFPGAに独自仕様のプロセッサ MipsCore と I/O コントローラを実装し、また、FPGA 内部のブロック RAM をメインメモリとして利用した。

液晶ユニットとして、コマンドインタープリタ型の液晶モジュール ITC-2432-035H を採用した。これは液晶のほかに液晶コントローラやマイコンを搭載するモジュールで、シリアル通信によるコマンドを受信すると、文字の表示やカーソルの移動といった、コマンドに対応する処理を実行する。5V 電源、グランド、シリアル通信の信号線の合計3本のみで、液晶ユニットとMieruPC本体を接続した。

コマンドインタープリタ型液晶ユニットの採用は重要な選択の1つであった。その主な利点は、MieruPC本体との接続ケーブルがシンプルになる*3こと、MieruPC本体にビデオメモリや液晶コントローラを搭載する必要がないことである。主な欠点は、液晶ユニットがプロセッサを持つ計算機システムとなりブラックボックスになる恐れがあること、シリアル転送の速度によって描画性能が制限されることである。これらの得失とプロジェクトの目的から、コマンドインタープリタ型液晶ユニットの採用を決めた。

マザーボードは、初期の段階では、ユニバーサル基板を用いて製作(図6の左)していたが、第1フェーズの後半で、図7に示すプリント基板としての実装に挑戦した。

*2 これら HDL による開発は、ハードウェアをソフトウェアとして記述するため、ハードウェア開発なのかソフトウェア開発なのか分類が難しい。本論文では、ハードウェア開発の一部ととらえる。

*3 液晶との一般的な接続には20本程度のフラットケーブルが必要になり、その取扱いは簡単ではない。

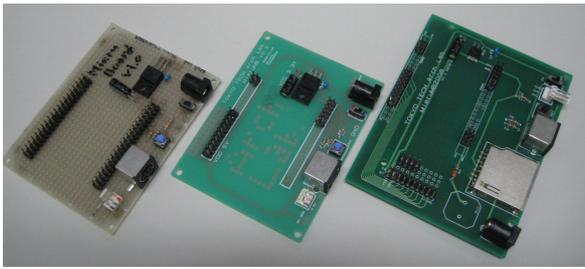


図 6 マザーボード MieruMB のいくつかの版
Fig. 6 Some version of mother board MieruMB.

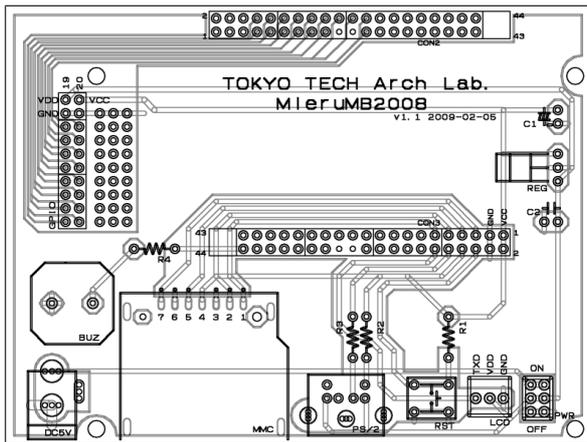


図 7 マザーボード MieruMB の基板設計データ
Fig. 7 The design layout of mother board MieruMB.

図 7 の基板設計データの左下から右に向かって、電源コネクタ、SD カードコネクタ、PS/2 コネクタ、リセットスイッチ、液晶ユニット用コネクタ、電源スイッチが配置されている。また、上側に FPGA ボードを搭載するピンがある。表と裏の 2 層で実現されたこのシンプルな基板の開発には、後に実施する独自仕様の FPGA ボードの開発準備という意図も含まれている。

第 2 フェーズで、独自仕様の FPGA ボードとして **FPGACard** (図 8) を開発した。FPGA に Spartan-3E XC3S250E (あるいは XC3S500E) を用い、メインメモリとして 512kB の非同期 SRAM を搭載する。これにより、メインメモリの容量が大幅に増えて、OS の搭載も現実的となった。この開発は挑戦的なもので、完成までに 5 回の基板の製造が必要となった。

第 3 フェーズで、独自仕様の液晶ユニットとして **MieruLCD** [3] を開発した。MieruLCD と MieruPC 本体との間のインタフェースとして、シンプルな RS-232C のシリアル通信 (通信速度 1Mbps) を採用した。MieruLCD は 2 つのモードを持つが、一般に利用されるコマンドモードでは、受信した文字列を「1 文字のコマンド文字 + 0 文字以上の引数」からなるコマンドの列として扱う。典型的なコマンドを表 1 に示す。一番左の列がコマンド文字を表す。たとえば、MieruPC 本体からシリアル通信で「FfcX10Y05DA」という文字列を送信すると、「色番号 0xfc (通常は黄色)

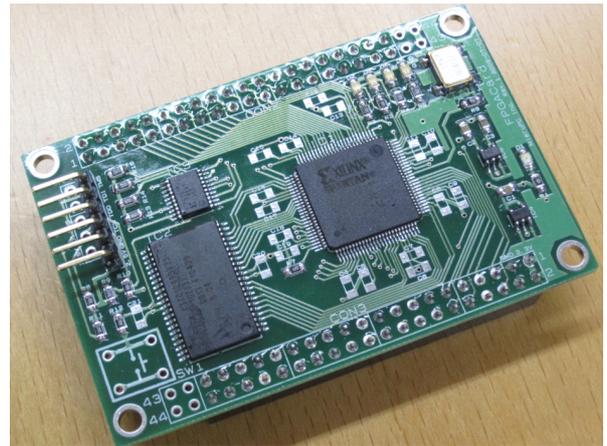


図 8 独自仕様の FPGA ボード (FPGACard)
Fig. 8 The original designed FPGA board (FPGACard).

表 1 MieruLCD が受け付ける典型的なコマンド
Table 1 Typical commands of MieruLCD.

コマンド名	引数	概要
C	ページ番号	指定ページをクリア
F	色 (16 進 2 桁)	前景色番号をセット
B	色 (16 進 2 桁)	背景色番号をセット
D	ASCII 1 文字	文字出力, カーソル移動
X	座標 (2 桁)	カーソル X 位置を移動
Y	座標 (2 桁)	カーソル Y 位置を移動
J	なし	カーソルを下に移動
d	なし	ドット出力
j	なし	描画位置を下に移動

で、カーソル位置 $(x,y) = (10,5)$ に文字 A を出力する」という意味を持つコマンド列となる。

マザーボード、FPGACard および MieruLCD の基板は設計、把握、変更の容易さから 2 層のデザインとした。また、誰でも利用しやすいように、回路図や設計データは無料の CAD を用いて作成した。

このように、独自仕様のハードウェア開発においては、シンプルなものから徐々に開発を進めていくインクリメンタルな手法が有効であった。

3.2 プロセッサ MipsCore と I/O コントローラ

FPGA に実装する MipsCore はキャッシュを持たないマルチサイクルのプロセッサとした。命令フェッチ、命令デコード、レジスタファイルアクセス、実行、メモリアクセス、ライトバックに対応する *fetch*, *decode*, *regfetch*, *execute*, *memaccess*, *writeback* の 6 ステップにより MIPS 命令を処理する。シンプルさを重視して、パイプライン処理は行っていない。

第 2 フェーズの FPGACard の開発によって利用できるようになった SRAM は 8 ビットアクセスであるため、MipsCore も、それまでの 32 ビットから 8 ビット単位のメモリアクセスに変更している。このため、*fetch* および

表 2 MieruPC-2010 本体の Verilog 記述の構成

Table 2 Verilog description of MieruPC-2010 main part.

ファイル名	行数	概要
MipsCore.v	645	MipsCore
Cp0.v	95	MIPS システム制御コプロセッサ
memcon.v	277	メモリコントローラ
mainmem.v	36	SRAM インタフェース
kbcon.v	177	PS/2 キーボードコントローラ
kbcon_table.v	527	キーコード変換 ROM
lcdcon.v	46	LCD コントローラ
mmcon.v	367	SD カードコントローラ
gpio.v	44	汎用 I/O
init.v	85	プログラムローダ
MieruPC.v	77	トップモジュール
define.v	197	定数などの定義
合計	2,626	

memaccess に要するサイクル数がそれぞれ5サイクルに増加した。乗除算の execute ステップは32サイクル、その他のステップには1サイクルを要する。このため、1命令の処理は、通常の命令で9サイクル、ロード/ストア命令で14サイクル、乗除算では40サイクルとなる。このように、標準で提供する MipsCore は、1つの命令の処理が完全に終了してから次の命令の処理を開始するとてもシンプルで低速なプロセッサとしている。このため、様々なプロセッサの高速化方式（キャッシュ、パイプライン処理など）を実装して、その効果を調べることができる。

第3フェーズのOSの開発にとともに、簡略化されたシステム制御コプロセッサ（MIPS CP0）を実装した。実現すべき機能をタイマ割込みとシステムコールの2つに絞り込んだため、CP0で定義されているレジスタのうち、COUNT、COMPARE（タイマ割込みに使用）、EPC（例外発生アドレス）の3つのレジスタと、SR（プロセッサ状態）、CAUSE（例外の発生理由）の2つのレジスタの一部ビットのみを実装している。また提供する例外はINT（割込み、タイマ割込みに限定）とSYSCALL（システムコール）の2つとした。

MieruPCでプロセッサからアクセス可能なI/Oは、すべてメモリマップドI/Oとして定義される。MipsCoreはハードウェア割込みを持たないため、キーボード処理などのレジスタの変化の検出にはポーリングを利用する。

表2に、MieruPC-2010本体におけるVerilog記述のファイル構成をまとめる。MipsCoreや簡略化されたCP0のほか、I/Oコントローラ、GPIO、プログラムローダを持ち、これらをトップモジュールが接続する。これらすべてをまとめたコード量は2,626行と少ない。kbcon_table.vのコード量が一見して多いように見られるが、実態は512ワードのROMをcase文で書き下したものである。

図9に、MieruPC-2010本体のブロック構成を示す。SD CardやSRAMといった図中の灰色のモジュールはFPGA

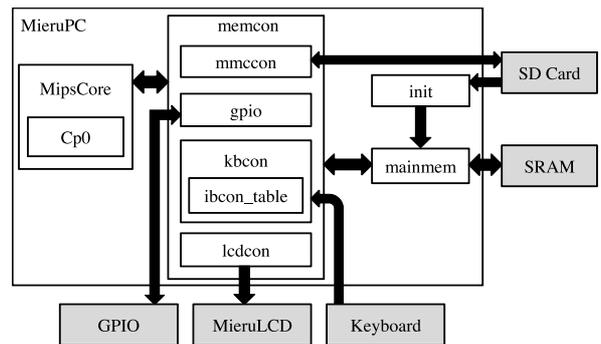


図 9 MieruPC-2010 本体のブロック構成

Fig. 9 Block diagram of MieruPC-2010 main part.

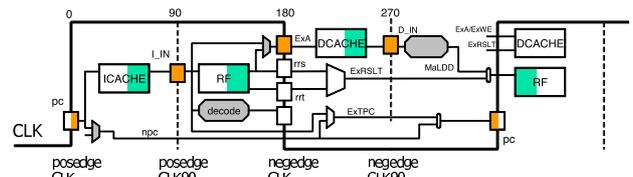


図 10 シングルサイクル版 MipsCore のデータパスとタイミング

Fig. 10 Datapath and timing of single-cycle MipsCore.

の外部のハードウェアである。図中の白色のモジュールがFPGAに実装される。それらのモジュールの名前は表2と同じものを用いている。たとえば、図中のCp0は、表2のCp0.vによって記述されるMIPSシステム制御コプロセッサである。図中の矢印がデータの流を表している。

3.3 ハードウェア量と動作速度

MieruPC-2010の本体が搭載するFPGAでは、ISE 11.5を利用した場合、Xilinx社の論理の大きさの尺度であるスライスを配置配線まで完了した状態で2,446個使用する。これは、FPGACardで主に利用するXC3S250Eが持つスライス2,448個のほぼ100%である。規模の大きいXC3S500Eが持つスライスの53%に相当する。小容量で安価なXC3S250EでもMieruPCは実現できるが、拡張を考慮するとXC3S500Eの利用が好ましい。

FPGAの内部ロジックは67MHzで動作するが、SRAMの安定アクセスのために54MHzで動作させている。先に述べたMipsCoreの命令実行に必要なサイクル数から、乗除算命令の実行頻度が少なく、ロード/ストア命令を20%含むアプリケーションを想定すると、MIPS値（Million Instruction Per Second）は5.4MIPSとなる。これは、1980年代の計算機の性能に相当し、工夫することで様々なアプリケーションを実装できるレベルといえる。

一方で、MipsCoreをマルチサイクルからシングルサイクルに改良しキャッシュを搭載することで、13MIPS（18MHz動作）を達成できることを確認している。図10に、シングルサイクル版MipsCoreのデータパスとタイミングを示す。位相の90度異なるクロックを用いたり、結果書

き込みと命令フェッチを同時に処理していたりと工夫を施している。

さらに、パイプライン処理を施すことで19 MIPS (36 MHz動作) まで性能が向上することも確認している。このように、プロセッサの動作周波数とアーキテクチャを変更できることはFPGAを用いたシステムの利点である。

4. MieruPC システムのソフトウェア

4.1 MieruOS

MieruPC-2010 で動作させる OS として、SD カードのファイルシステム管理、ユーザアプリケーションのロードと起動、システムコールの処理を提供する MieruOS を開発した。図 11 に、MieruOS のアーキテクチャを示す。図では、MieruOS が提供するモジュールを灰色とした。

扱うプロセスは、カーネルプロセス (図の左上) とアプリケーションプロセス (図の右上) の2つのみとした。カーネルのプログラムサイズは約 40 kB、使用するメモリは約 67 kB である。このため、カーネルプロセスが 512 kB のメインメモリのうち 128 kB の領域を利用し、アプリケーションプロセスが残りの 384 kB を利用することにした。

MieruPC が起動すると、まず、ハードウェアのプログラムローダが OS をロードし、プロセッサが実行を開始する。その後、プロセス管理やメモリ管理などの各種データ構造を初期化し、カーネルプロセスを起動する (図 (a))。カーネルプロセスではシェル (図 (b)) が動作し、ユーザからのコマンドを受付ける。

次に、ユーザからのコマンドによりアプリケーション実行が指示されると、プログラムローダ (図 (c)) により実行するプログラムイメージがメモリ上に展開され、アプリケーションプロセスとして起動 (図 (d)) される。アプリケーションが終了すると再びカーネルプロセスに戻り (図 (e))、新たなアプリケーションが実行可能になる。アプリ

ケーションではシステムコールを介して OS の機能を使用する (図 (f))。

図 11 では、システムの電源が投入されてから最初のアプリケーションの実行が開始するまでの流れを黒矢印で示している。

図 11 下に示す 4 つのモジュールの機能は次のとおりである。

メモリ管理 brk システムコールによりヒープ領域の増減が可能。malloc や free はライブラリが提供する。

ファイルシステム SD カード上の FAT ファイルシステムにアクセスできる。ファイルやディレクトリの読み書きが可能。

プログラムローダ ELF 形式の実行ファイルのサポート。
デバイスドライバ 各種デバイスのためのドライバ。

メモリマップについて述べる。MIPS のレジスタは 32 ビット長であるが、メモリアドレスには下位 24 ビットのみが利用され、上位の 8 ビットは無視される。先頭から 0x01ffff までの 128 kB を MieruOS が利用する。特に、0x000000 から 0x0000ff がスタートアップルーチン用、0x000100 から 0x0001ff までが例外ハンドラ用に予約されている。アプリケーションが 0x020000 から 0x07ffff までの 384 kB を利用する。その他、0x800100~0x8001ff がメモリマップド I/O のために利用される。

MieruOS は、C++ とアセンブラにより 3,672 行で記述されている。スタートアップルーチンやシステムコールの一部はアセンブラ、その他のほとんどの部分は C++ で記述されている。

4.2 ライブラリと SDK

アプリケーションは主に C 言語により記述する。ソースコードに MieruPC のライブラリファイル mierulib.h をインクルードしてコンパイルし、リンカスクリプトを用いてライブラリとリンクする。

図 12, 図 13 は、キーボードの入力に反応して “Hello world” を表示するアプリケーションと、提供されるライブラリのコードの一部である。

MieruPC のアプリケーションでは、図 12 の 5 行目の液晶ユニットを初期化する関数 lcd_ttyopen を最初に行う。7 行目でキーボードの入力をビジー状態で待ち続ける。何らかの入力があれば 8 行目の lcd_puts 関数が呼び出される。この関数は図 13 の 1~10 行目に定義され、文字列を 1 文字ずつ切り出し、文字を描画する液晶ユニットへのコマンド “D” とともに、5~6 行目で lcd_sendchar 関数に渡す。lcd_sendchar 関数では、16 行目でキーボードコントローラがビジーでなくなるまで待ち、17 行目で文字を送信する。さらに、7 行目で (図中では省略するが) カーソルの位置を 1 文字分移動させる。この処理を文字数だけ繰り返す。その結果、液晶ユニットの画面には “Hello world” が

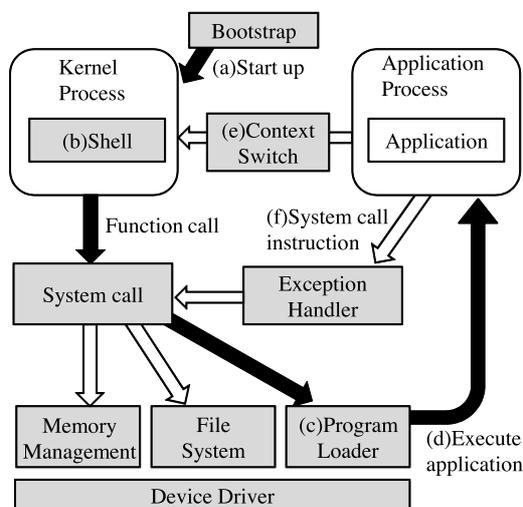


図 11 MieruOS のアーキテクチャ
Fig. 11 Architecture of MieruOS.

```

Application
1 #include <mierulib.h>
2
3 int main (void)
4 {
5     lcd_ttyopen(1);
6     while(1) {
7         while (!mpc_getchar(NULL));
8         lcd_puts("Hello world");
9     }
10 }
    
```

図 12 MieruPC のアプリケーションの例
 Fig. 12 An example of an application of MieruPC.

```

Library
1 void lcd_puts(char *s)
2 {
3     int i = 0;
4     while(s[i]) {
5         lcd_sendchar('D');
6         lcd_sendchar(s[i++]);
7         /* move internal cursor... */
8     }
9     lcd_nextline();
10 }
11
12 void lcd_sendchar(char c)
13 {
14     volatile char *ptr =
15         (volatile char *) MP_MMAP_LCD;
16     while (!*ptr);
17     *ptr = c;
18 }
    
```

図 13 MieruPC のライブラリの一部
 Fig. 13 A part of the libraries.

表示される。9 行目で改行コマンドを送信する `lcd_nextline` 関数を呼び出して、アプリケーションへと制御を戻す。この例のように、ライブラリもブラックボックスにならないように、シンプルで分かりやすい記述を心がけた。

アプリケーションの開発・検証を容易にするため、ライブラリ、シミュレータ、SD カードのイメージファイルを作成するツールなどを統合する SDK として MieruSDK [4] を整備した。

MieruSDK を利用してアプリケーションをテストする際は、まずアプリケーションをライブラリとともにクロスコンパイル、リンクして、バイナリを作成する。次に、OS

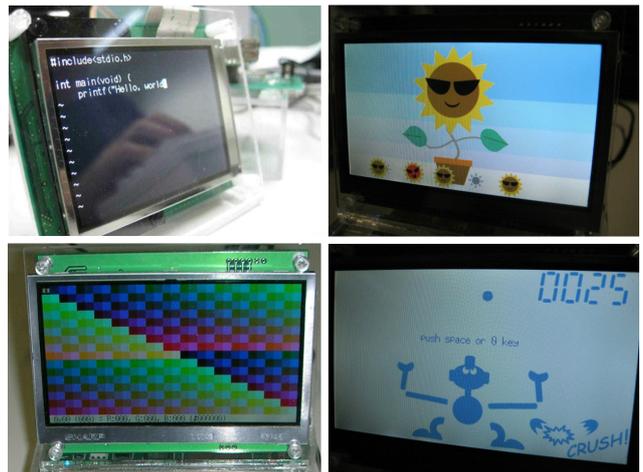


図 14 開発されたアプリケーションの例
 Fig. 14 Photo of developed applications.

カーネルとアプリケーションのバイナリ、起動時に実行するアプリケーションを記述した簡易シェルスクリプトを含む SD カードのイメージファイルを作成する。こうして作成されたイメージファイルを入力としてシミュレータを実行すると、OS の起動とともにテスト用のアプリケーションが実行される。

シミュレータで十分なテストを行ってから、SD カードにバイナリを書き込み、実機でテストする。MieruSDK はこうした一連の作業を自動化し、アプリケーションの開発を支援する。

開発されたいくつかのアプリケーションのスクリーンショットを図 14 に示す。左上はテキストエディタ、右上はハードウェアに音センサを追加してタイミングに合わせて音を鳴らすゲーム、右下はボールを受け止めるゲームの様子である。左下は、表示可能な色を分かりやすく表示し、選択した色番号を表示するアプリケーションである。

5. 議論

5.1 教育現場における MieruPC システム

MieruPC システムを教育の現場で利用する試みがなされている。

東京工業大学の計算機アーキテクチャの講義では MIPS アーキテクチャを題材として扱っており、その中で SimMips と MieruPC システムを取り入れた講義を行った。講義の最後には期末課題として、SimMips をインストールしたサーバを受講者向けに開放し、SimMips のソースコードの一部を変更する必要がある課題を課した。また、余裕のある人向けに MieruPC のアプリケーション開発を課題とした。SimMips の上で自らの作ったアプリケーションをテストしてもらい、完成したアプリケーションは本人のプレゼンテーションとともに MieruPC 実機の上で動作させた。

この課題は必須ではなかったが、受講者の約 10% の学生が独創性に富んだアプリケーションを提出した。SimMips

というソフトウェアによるシミュレーション環境だけではなく、「手のひらにのる実在する計算機システムとしての MieruPC で開発したアプリケーションが動作するという新鮮さ」が学習意欲の向上につながっている。

しかし、より多くの学生に広く触れてもらうためにはいくつかの課題がある。まず、シミュレータ上で動作していたアプリケーションが、いざ実機で動作させようとする環境の微妙な違いによりうまく動作しない、という場面が見受けられた。こうした課題に自宅で取り組む場合、実機でのテストの機会をどのように提供するかを検討する必要がある。また、意欲はあったがどう手をつけてよいかわからなかったという意見も見受けられ、ドキュメントや開発環境の整備も課題となった。

豊橋技術科学大学でも、高専生向けの組込みシステムに関する夏期講座として、MieruPC のアプリケーション開発の課題を課した。はじめに MieruPC の概要やアプリケーション開発の流れに関する説明をした後、5 日間で各自のアイデアに基づいたアプリケーションを開発させた。講座の後、学生には制作したアプリケーションの説明や講座の感想などを数枚のレポートの形で提出することを求めた。レポートでは「思いどおりに動いてくれたときに、自分でもこんなものが作れるんだ、と感動した」「パソコンから離れたものでプログラムを実行するのが良いなと思った」などの感想が得られている。

ハードウェアに手を入れる実験も進められている。東京農工大学では、集積回路工学の実験に MieruPC が利用されている。MieruPC のハードウェアを使い、はじめに液晶コントローラ、PS/2 キーボードコントローラと簡単な回路により、液晶モジュール上でスロットマシンを動作させる。ここからステップアップして、最終的に MIPS ベースのプロセッサを動作させるといった内容である。

5.2 MieruPC であるからこそその教育効果

実践的なプロジェクト実習の題材として、文部科学省先導的 IT スペシャリスト育成推進プログラムの 1 つである「情報理工実践プログラム」(東京大学、東京工業大学、国立情報学研究所) のソフトウェア開発プロジェクト実践に MieruPC が用いられた。

東京大学と東京工業大学の修士 1 年の学生各 2 名、合計 4 名が、通年の実習を通して MieruPC のハードウェア、ソフトウェアを拡張し、MieruPC Rev.ITsp と名付けられた計算機システムを構築した [4]。MieruPC-2010 で用いられる MieruOS と MieruSDK は、この実習で作成されたソフトウェアを拡張したものである。

MieruPC Rev.ITsp では、独自の拡張として、RS-232C シリアル通信による一般的な計算機 (PC) との通信機構を実装している。この拡張により、ウェブサービスを通じて取得した情報を、MieruPC Rev.ITsp が受信して画面に



図 15 MieruPC Rev.ITsp 向けのアプリケーションの例
Fig. 15 Photo of an application for MieruPC Rev.ITsp.

表示するアプリケーションが実現可能である。その例として、東京近郊の降雨量をグラフィカルに表示するアプリケーションの動作例を図 15 に示す。

MieruPC を用いたプロジェクト実習により、一般的なコンピュータと比べてハードウェアの制約が大きい中で、ある機能がどのようなパーツの組合せで提供されているのかという知識、また、所望の機能を実現するために、その制約に合わせて利点や欠点を考慮し、実装方法を選択する能力が身についたことが、文献 [4] に述べられている。

このように、ソフトウェアからハードウェアまでの様々なレイヤを見通すことができ、そのうえで、適切な実装方法を検討して実践する能力の習得こそが MieruPC が支援すべき教育項目であり、これを達成できたことは MieruPC であるからこそその教育効果といえる。

また、この例から、MieruPC プロジェクトの狙いである「計算機システムを理解できる、作れる、拡張できる」を体験するプラットフォームとして機能したことも分かる。このプロジェクト実習に参加した修士 1 年の 4 名は、ベースとなる MieruPC-2008 Rev.2 のシステムをよく理解し、そしてハードウェアおよびソフトウェアの両面からの拡張を行い、設定した計算機システムを作り上げることに成功している。

5.3 関連する実験・演習との比較

他大学で行われている FPGA をターゲットとする関連の実験・演習 [5] との着眼点および教育効果の相違について議論する。

東京大学情報科学科では、3 年生の冬学期から「ほんとうのコンピュータ自作」として CPU 実験を実施している。これは、4~6 名からなるチームを構成して、与えられた課題プログラムが高速に動作するように、コンパイラ、アセンブラなどのツールを含む独自のコンピュータを設計・製作する。

コンピュータを自作するという目的は同様であるが、東京大学の CPU 実験は、大規模な FPGA ボードを用いて、

できるだけ高速なコンピュータシステムを設計・製作する点でかなり敷居の高い実験である。

一方、MieruPC プロジェクトでは、アセンブラ、コンパイラはその見える対象には含まれない。FPGA ボードや液晶ユニットという主要なハードウェア要素の中身が見えるように、独自設計のハードウェアを提供する。ソフトウェアに関しては、OS、ライブラリを含む SDK、プロセッサシミュレータを提供する。これらは、高性能を目指すことなく、あえて改良の余地があるシンプルなコードを提供し、それを実験や演習に応じてアレンジして利用してもらうことを目指しており、その着眼点は大きく異なる。

いくつかの大学の実験では、市販の FPGA ボードにコンピュータを実装して、課題プログラムの動作速度を評価する。

京都大学情報工学科では、独自の SIMPLE アーキテクチャに準拠するコンピュータの方式設計および論理設計を行い、コンピュータ全体を FPGA ボードに実装し、ソートを課題プログラムとして性能を競っている。

熊本大学情報電気電子工学科の実験では、独自に開発した実験用基板に搭載される FPGA に、教育用マイクロプロセッサ KITE を実装する。

慶應大学情報工学科では、ハードウェア記述言語を用いて 16 ビットの CPU を設計し、FPGA ボードに実装してサンプルコードの動作確認を行う。

これらの実験では、コンピュータアーキテクチャの講義で学んだ CPU の深い理解とその応用を目的としている。一方、MieruPC プロジェクトでは、CPU、OS、ライブラリ、ハードウェア基板などの「ハードとソフトのすべてが見える」コンピュータシステムの提供と多くの層の横断的な教育を支援することを目指しており、狙いとする教育範囲が大きく異なっている。

5.4 関連研究、得られた知見、展開

研究・教育用ツールの例として、プロセッサシミュレータの SimpleScalar [6], SPIM [7] が有名である。SimpleScalar は主に、計算機アーキテクチャ研究で活用されているが、その複雑さのために計算機システムの教育には適していない。SPIM はテキスト形式のアセンブリを直接実行できるため、クロス開発環境が必要なく、命令セットやアセンブリプログラミングを手軽に学ぶには適している。しかしながら、オペレーティングシステムを含む計算機システムの教育には適していない。

FPGA を利用した研究向けのシステムとしては、RAMP (Research Accelerator for MultiProcessors) [8] が知られている。RAMP は FPGA の高速かつ柔軟な計算が可能であるという特徴を生かして、マルチプロセッサやマルチコアの高速なプロトタイピング環境を提供している。しかし、こうした動作速度を追求したシステムはその分複雑な実装

となっていることが多く、教育目的でそのまま用いることは難しい。

関連する試みとして、TINYCPU [9] がある。ここでは、16 ビットの独自命令セットのプロセッサを FPGA 上に実装している。また、クロスアセンブラとクロスコンパイラも提供されており、これらのコードおよび文法ファイルの合計は 400 行あまりにまとめられている。TINYCPU は、容易に理解できるコードで、複数の分野にまたがった教育を可能にするなど、MieruPC との共通点も多い。MieruPC プロジェクトは、シンプルさを保ちつつ、より実用的で魅力的な計算機システムを提供することを目標とする点が挑戦的である。

MieruPC プロジェクトを通じて得られた主な知見をまとめる。

- プロジェクト開始時のスタートダッシュが重要。優秀な学生が実力を発揮することが鍵であった。
- 独自仕様のハードウェア開発においては、シンプルなものから徐々に開発を進めていくインクリメンタルな手法が有効であった。
- 競争的な資金を獲得することなく、大学で中身の見える計算機システムを開発できる。MieruPC システムの開発は 3 年間で完了した。
- MieruPC が、「計算機システムを理解できる、作れる、拡張できる」を体験するプラットフォームとして機能することを確認した。

MieruPC の展開について述べる。MieruPC システムには、「安価でない」という欠点がある。そこで、MieruPC プロジェクトで開発したシステムをベースに、小型の液晶ユニットを採用するなどの工夫を施した安価な材料を採用する組み込みシステム用の教材 MieruEMB システムを開発した。動作の様子を図 16 に示す。このように、MieruPC プロジェクトの成果は様々なところで活用されている。



図 16 組み込みシステムの教材として開発した MieruEMB
Fig. 16 MieruEMB system for embedded system education.

6. おわりに

プロセッサを含むハードウェア、オペレーティングシステム、アプリケーションといった多くの層の横断的な教育を支援するプラットフォームの提供を目指して、MieruPCシステムの開発に取り組んでいる。本論文では、MieruPCシステムを開発するプロジェクトの詳細、工夫した点および得られた知見をまとめた。

「シンプルで分かりやすい」、「小型」、「安価」、「魅力的」といった要求は、ときに相反する複雑な関係にある。そのようななか、必要な機能と切り捨てる機能を一貫性を持って選択することが重要である。MieruPCプロジェクトでは、開発時からその普及を前提にして、これらの取捨選択を行った。プロジェクト成果の1つがMieruPC-2010システムである。

謝辞 はじまりのはじまりでは権藤克彦先生に、開発で三好健文さん、小林良太郎先生、中條拓伯先生に、並木美太郎先生のご支援に、河端麻紀子さんのご支援に、大部分の開発で藤枝直輝さんに、FPGACardで若杉祐太さんに、MipsCoreで渡邊伸平さんに、Rev.ITspで高前田伸也さん、佐野正浩さん、芝哲史さん、曹哲さん、川合秀実さん、笹田耕一さんに、開発で佐藤真平さん、森谷章さん、植原昂さん、森洋介さん、高橋朝英さん、棟岡朋也さん、山田裕介さん、佐野伸太郎さんに感謝します。

参考文献

- [1] 吉瀬謙二：シンプルでわかりやすい計算機システムの開発に向けた挑戦，組込みシステムシンポジウム (ESS2012)，pp.110–118 (2012).
- [2] 藤枝直輝，渡邊伸平，吉瀬謙二：研究・教育に有用な MIPS システムシミュレータ SimMips，情報処理学会論文誌，Vol.50, No.11, pp.2665–2676 (2009).
- [3] 藤枝直輝，吉瀬謙二：FPGA による高速で扱いやすい LCD 表示器の実装と評価，電子情報通信学会研究報告 RECONF2010-82，pp.193–198 (2011).
- [4] 佐野正浩，高前田伸也，芝 哲史，曹 哲，伊藤宗平，川合秀実，笹田耕一，吉瀬謙二：Mieru システムソフトウェア，情報処理学会研究報告 2010-ARC-189，No.16 (2010).
- [5] 中條拓伯：ハードウェアシステムを駆使した研究・教育の実践，SWoPP2010 BOF2 発表資料 (2010).
- [6] Burger, D. and Austin, T.M.: The SimpleScalar Tool Set, Version 2.0, Technical Report CS-TR-1997-1342, University of Wisconsin-Madison (1997).
- [7] Larus, J.R.: SPIM S20: A MIPS R2000 Simulator, Technical Report, Computer Sciences Department, University of Wisconsin-Madison (1990).
- [8] University of California at Berkeley: RAMP: Research Accelerator for Multiple Processors, available from (<http://ramp.eecs.berkeley.edu/>).
- [9] Nakano, K. and Ito, Y.: Processor, Assembler, and Compiler Design Education using an FPGA, *Asia-Pacific Workshop on Embedded System Education and Research*, pp.723–728 (2008).



吉瀬 謙二 (正会員)

1995年名古屋大学工学部電子工学科卒業。2000年東京大学大学院情報工学専攻博士課程修了。博士(工学)。同年電気通信大学大学院情報システム学研究科助手。2006年東京工業大学大学院情報理工学研究科講師。2011年同准教授。計算機アーキテクチャ、並列処理に関する研究に従事。電子情報通信学会，IEEE-CS，ACM 各会員。