

SDRAM をメインメモリとする

MIPS-CPU の FPGA 化

FPGA implementation of MIPS CPU with SDRAM as a main memory

森川 良[†], 堀田 厚生^{††}

Ryo MORIKAWA[†], Atsuo HOTTA^{††}

Abstract In the year of 2003, our laboratory designed CPU of MIPS architecture, one of the representative RISC architectures, in order to study design methodology of a CPU, and implemented it in FPGA. The CPU adopted SRAMs in the FPGA as a main memory.

As a next step, a MIPS-CPU with a SDRAM as a main memory has been studied. SDRAM controller module has been designed to cope with the specific features of SDRAMs such as refreshing, address multiplexing, access latency and so on. Designed CPU module has been implemented in a FPGA. The highest operation frequency is found to be 82.50MHz with a program of small steps.

1. はじめに

1.1 研究の背景

FPGA(Field Programmable Gate Array)は、デバイス内部の回路情報をプログラミングすることができる主に SRAM ベースの LSI である。ハード開発において半導体 IP(Intellectual Property)の動作を実デバイス上で容易に検証できるため、システム LSI に求められている開発期間の短縮、開発コストの低減というメリットを企業にもたらす。従来では、一つの製品の開発段階において試作品に不具合等を発見した場合、再度初めから試作機を作らなければならなかった。そこで ASIC の開発段階において FPGA が使用されるようになった。また、企業の製品開発という分野のみに留まらず、大学等の研究機関においても FPGA を用いた研究が数多く行われている²⁾³⁾⁴⁾⁵⁾⁶⁾。

FPGA 化の高集積化が進み、現在では様々な規格の入出

カインターフェイスや、DSP などにおいて様々な IP が提供されている。また、最近ではマイクロプロセッサなども登場している。例として、Altera 社からは NIOS があり、Xilinx 社からは MicroBlaze と PicoBlaze が提供されている。⁵⁾ これらは、FPGA ベンダである各社が開発しているため、それぞれ同社製 FPGA に適した構成となっている。

このように、FPGA 単体における性能向上および IP 蓄積という点も FPGA が広い分野で普及する理由の一つであり、他方ではソフトウェアデザインツールの普及という点も挙げられる。FPGA の大手ベンダの一つである Altera 社を例に挙げると、ユニバーシティプログラムと呼ばれる制度があり、教育機関の教員や学生が無償で FPGA/CPLD 開発ツールに触れることができる。

そして、近年においては雑誌の付録としても FPGA/CPLD が登場し、開発ツールも機能制限はあるが同梱されている(ライセンスは各自入手)。このようなことから、FPGA/CPLD による回路設計がより身近になっていると言える。

[†] 愛知工業大学大学院 工学研究科 (豊田市)

^{††} 愛知工業大学 電子工学専攻 (豊田市)

1・2 従来の研究

FPGA を用いた研究は数多く行われている。IC 内の回路情報を HDL 等を用いて容易に書き換えることができるという利点を活かし、各々の理論を実現させる手段として用いられる例や、同様の利点を活かし LSI 教育用として用いられる例等が挙げられる。

角田ら²⁾は、FPGA の性能向上に伴い、高性能プロセッサを実装するためにはプロセッサの仕様と FPGA 上での実装規模の関係を把握することの必要性を唱え、32 ビット RISC プロセッサの設計を行い、論理合成を行った。

次に荒木ら³⁾は、組み込み機器に適したリアルタイム OS 内で必要とされている機能について着目したプロセッサを Verilog-HDL で記述し、FPGA に実装した状態のシミュレーションを行った。この結果、提案したプロセッサシステムの特徴と特性を確認し、ボトルネックとなるバスの条件と、提案したプロセッサシステムにおけるプロセッサ数の実用上の上限を示した。

また、田中ら⁴⁾は FPGA と DSP という代表的なハードウェアとソフトウェアを用いて実際に実機で動作させると共に、その処理性能を学生に競わせ、トレードオフ解析に必要な知識を得ることができる学習方法を紹介している。FPGA 内にマイクロプロセッサを実現し、それをソフトウェアで制御することで、計算機アーキテクチャ、システムソフトウェア、及び集積回路設計といった多くの分野の知識を同時に体験及び学習することを目的としている。ハードウェア・ソフトウェアトレードオフを考察する場合に、ハードウェア面においては FPGA の柔軟性が有効であると考えられるからである。

1・3 研究の目的

平成 15 年度、我々の研究室において RISC タイプ CPU の代表的なものの一つである MIPS アーキテクチャの CPU(以後 MIPS)の設計を行い、FPGA に実装し、動作の評価を行った。

そこで、本研究ではその次の段階の一つとして、SDRAM をメインメモリとして採用し容量の拡大を図る。その手順として、SDRAM コントローラ設計、コントローラ部と MIPS の結合、の過程を経て行う。また、それぞれの過程において FPGA での実機検証を行う。

本研究で、設計・論理合成・シミュレーション・実機解析において使用するソフトウェアは、Quartus2 ver4.2 を用いる。ハードウェアは、Altera 社の FPGA、Stratix シリーズの一つである EP1S10F780C7ES と、Micron 社の 2Kwordx32bit 構成の 64Mbit の SDRAM が実装されている CQ 出版社 Stratix 評価キットを使用する。コントローラ等、回路設計には Verilog-HDL を用いる。

2. SDRAM コントローラ

2・1・1 DRAM

DRAM はメモリセル内部にあるコンデンサに電荷を蓄えることによりデータを保持する。構造が単純であるため、大容量かつ安価という利点を持つ。しかしながら、コンデンサに蓄えられた電荷はリークにより時間の経過と共に失われるといった欠点も持つ。このデータ消失を避けるために、定期的なチャージを行うリフレッシュ動作を必要とする。

2・1・2 SDR(Single Data Rate)SDRAM

SDR_SDRAM はクロック同期式 DRAM である。主な特徴として、コマンドによる制御、リフレッシュ、初期化等が挙げられる。

SDRAM の内部状態はコマンドを入力することにより制御される。コマンドは、NOP、Active、Read、Write、Precharge、Refresh、Load Mode Register 等がある。アドレス、データは各コマンドのタイミングに合わせて入出力される。アドレスは、アドレス・マルチプレクス方式により、アドレスを行列に分け、時間差で入力することにより、アドレス・バス幅の削減を図っている。また、データ・バスは双方向バスで構成されており、そのため、入出力でのタイミング調整が重要になる。

初期化は、メモリデバイスへの電源投入後の必須事項とも呼べるもので、例えば、電源投入後 100 μ 秒間はアクセス等何も行わず、その後、リフレッシュ動作を 2 度行うと言ったものである。この過程を無視した場合、その後のデータの読み書きというメモリ本来の役割における正常動作が保障されない。また、この初期化の過程においてコマンドの部分で触れた Load Mode Register 命令において、読み書き動作の設定を行う。設定の主立ったものとして、バースト長やキャス・レイテンシがある。図 1 にデータを読み出す場合を例に挙げる。バースト長を 4 としたとき、一つのアドレスを与える事で、そこから連続する四つ分のデータを排出するといったものである。また、キャス・レイテンシはリードコマンドが挿入されて、データが排出されるまでの遅れ時間であり 1 とした場合、コマンド挿入から 1 クロック遅れて排出される。このレイテンシは動作周波数に大きく左右される。

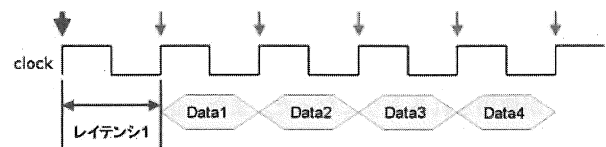


図 1 データ読み出し例

2・2 コントローラ部概要

上述した SDRAM の特徴を踏まえて、コントローラ部に必要と思われる機能をまとめると、

- ・初期化、またそれに伴う動作設定

- ・コマンド、アドレス、データの適時排出
- ・データ双方向バスの制御
- ・リフレッシュ動作の制御

などが挙げられる。

これらの考えを基に設計したコントローラ部のブロック図を図2に示す。図に示すように、イニシャライズ・モジュール、セレクト・モジュール、ステートマシン・モジュール、三つのモジュールで構成している。

イニシャライズ・モジュールでは、先述した初期化、またそれに伴う動作設定のコマンド排出を行っている。今回はバースト長を1とし、連続データでは無く単一のデータの読み書きを行うよう設定した。キャス・レイテンシは2に設定した。

セレクト信号はイニシャライズ・モジュールが初期化に関する一連の作業を終了したことをセクタへ示し、セクタはイニシャライズ・モジュールの作業が終了するまではイニシャライズ・モジュールからの入力を出し、作業完了を示すセレクト信号を受け取るとステートマシン・モジュールからの入力を出し出す。

そして、ステートマシン・モジュールでは、データの読出し、書き出し、リフレッシュに関するコマンド、アドレス、データの適時排出を行う。また、双方向バスの制御もここで行う。

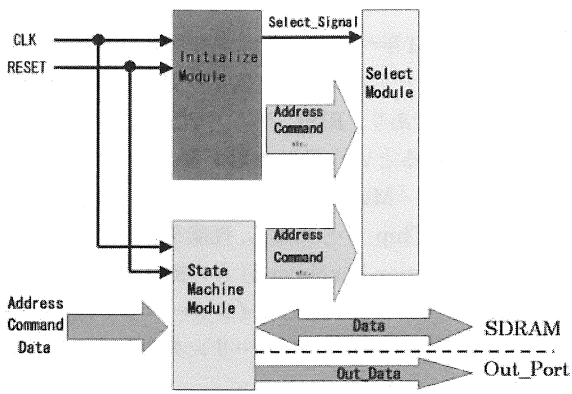


図2 コントローラ部ブロック図

続いて、コントローラの状態遷移を図3に示す。これは、初期化処理が終了しメモリ本来の役割を行う際のコントローラ部ステートマシンの状態遷移を表す。各状態において、それぞれの名前と同義のコマンドを排出し、アドレス、データの排出等のタイミング制御も行っている。NOPの状態ではコントローラは常に待機しており、外部（コントローラに対する入力）から読み書き要求が出された場合、図4（）内の数値はバス幅）にて示すステートマシン・モジュール内の各レジスタへ、アドレス、データ、読み書きといった要求内容を格納する。ここでは、要求、アドレス、データの情報はクロックに同期して全て同時に入ってくるものとする。そして、Active状態に遷移し、先程蓄えた

アドレスから、コマンドと共にアドレスを排出し、NOP_Aへと遷移する。ここで、要求内容を判別し、各処理へ遷移していく。今回設計したコントローラではNOPから再びNOPへ戻るまでが一つの作業の単位となる。また、要求を受け付け、再びNOP状態に戻るまでは、外部からの（読み書き）要求は受け付けないこととし、読み書き処理中にモジュール内部のリフレッシュカウンタから、リフレッシュ要求が出された場合は一連の作業が終了した後に処理される。そのため、リフレッシュカウンタは、規格で定められた間隔よりも、一作業分（NOPから再びNOPまで戻る間分）早く要求を出すようにしている。

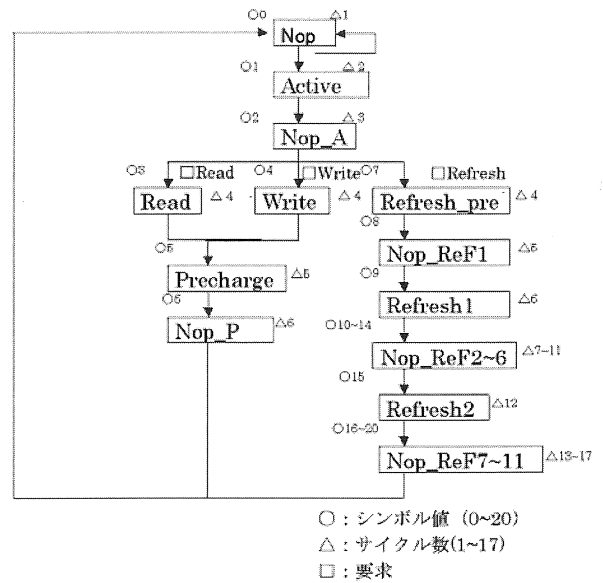


図3 コントローラ状態遷移

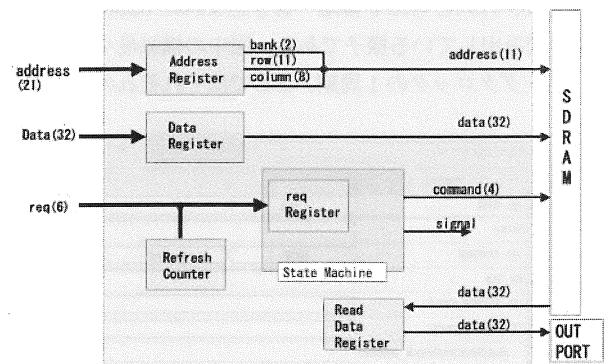


図4 ステートマシン・モジュール内部構成

2・3 論理合成

先述したコントローラ部に、データの読み書き要求を出し、またそれと同時にアドレス、データをコントローラ部へ排出する動作検証用のテスト・モジュールを加え論理合成を行う。テスト・モジュールの排出するデータは、当研

研究室において MIPS CPU を設計した際に動作検証用に用いた簡単なプログラムの一つ (データ 5555AAAAh 内の 1 であるビットをカウントするプログラム) をデータとしている。テスト・モジュールとしての動作は、このデータをメモリへ書き込み、全ての書き込み作業が完了した後、読み出す作業を要求するものである。

「Quartus II ver.4.2」による論理合成結果を表 1 に示す。対象デバイスは Altera 社 FPGA Stratix シリーズ EP1S10F780C7ES である。実機動作検証のためロジアナ IP を使用する。これを利用することにより FPGA 内の余った LE (Logic Element)、RAM を利用して内部信号を解析することができる。ロジアナ IP を用いると、ロジアナ IP 無しの状態よりも最大動作周波数が低下する。ロジアナ IP は本来回路動作においては必要が無い回路であるので両方の結果を示しておく。

表 1 論理合成結果

	ロジアナIPあり	ロジアナIPなし
最大動作周波数	101.09MHz	121.20MHz
使用LE (全10570個)	1400個(13%)	330個(3%)
使用内部メモリ (全920448bit)	208,384bit (22%)	1536(1%)
使用PLL(全6個)	1	1

2・4 実機テスト

実機動作検証結果の一部を図 5(a)(b)に示す。図 5(a)は論理合成結果の最大動作周波数を参考に 99.00MHz で駆動させた結果の一部である。図 5(b)は 99.00MHz より少し落とし周波数を 88.00MHz で動作させた結果の一部である。それぞれ、行列ともに 0 番地へ書き込まれた 32bit データ 22h を読み出している様子である。図中の横軸最小幅はサンプリングクロックの 1 周期、ここではそれぞれの駆動周

波数である。commd はコマンドを表し、MData_inout は双方向データバスの動きを表している。inout_enable の信号が 0 であるときはメモリへの出力となり、1 であるときはメモリからの入力となる。ここで注目すべき点は、両図の inout_enable の立ち下がり直前の MData_inout 値、立ち下がり直後の RData_keep の値である。RData_keep は MData_inout からのメモリ読み出しデータを保持するために設けたレジスタの値である。99.00MHz 動作の場合共に 22h という正しい値が読み出されていないことが分かる。これは、表 1 に示すロジアナ IP ありの場合の論理合成結果の最大動作周波数に満たない周波数で誤動作が起きていることを示している。この原因の一つとして、論理合成時に FPGA 外部に位置する SDRAM の情報が含まれていないことが挙げられる。FPGA 内部においては、Altera 社のデバイスに関する LE や配線遅延等の情報を基に論理合成結果が算出されるわけであるが、あくまでも FPGA 内の情報に過ぎない。そのため、FPGA 外部の情報、メモリ・FPGA 間の配線遅延等が論理合成結果に反映されず、実際の最大動作周波数より高い結果が算出される原因の一つとなっていると考える。よって、今回製作したコントローラの最大動作周波数は、ロジアナ IP の有無に関係無く実機において確認できた範囲、88.00MHz とする。

3 CPU との結合

3・1 MIPS

MIPS とは 1981 年～1984 年にスタンフォード大学のジョン・L・ヘネシーらのグループにより開発された CPU アーキテクチャであり、RISC タイプで限定された数の命令を高速実行するという設計方針が採用されている。MIPS の名前は「Microprocessor without Interlocked Pipeline Stages Chip」から由来し、性能を評価する MIPS 「Million Instruction Per Second」との掛け言葉でもある。MIPS は元々ハードウェアとしてインターロック機構を持たず、コンパイラの動的スケジューリングによりストール

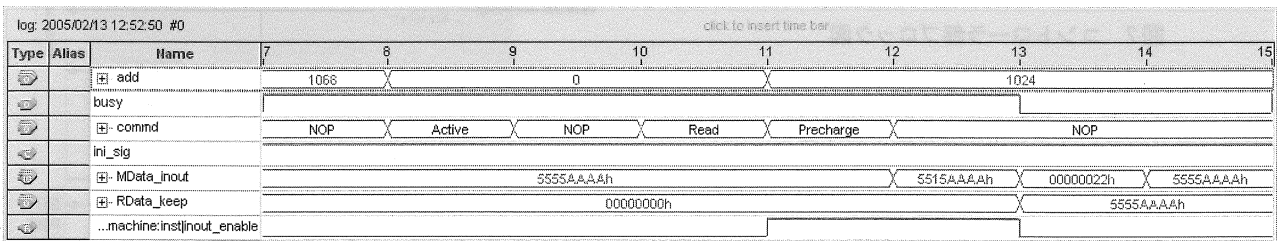


図 5(a) 動作周波数 99.00MHz 読み出し例

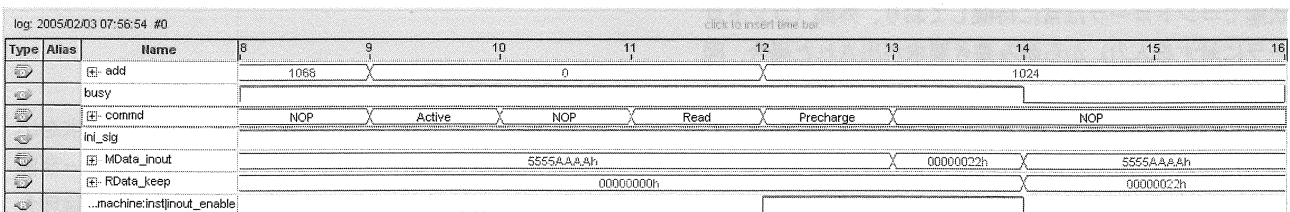


図 5(b) 動作周波数 88.00MHz 読み出し例

処理を行っていた。ただし、今ではパイプライン段数の増加によりあまり高速化にならないためハードウェア的にストール処理を行っている。

当研究室で MIPS 命令セットアーキテクチャに基づいて設計した CPU の命令は整数演算を中心に 25 命令に限定したものをを用いている。

halt 命令は本来 MIPS の命令にはないが、プログラム停止の都合上付加している。

今回の SDRAM コントローラの結合対象となる CPU はマルチサイクル・タイプのものとした。主な理由として、パイプラインはマルチサイクル・タイプにおいての処理が成り立つという条件の下で構成されている。そのため、マルチサイクルにおいての (1 命令あたりの処理時間等) 動作検証が重要な意味を持つためである。

設計した SRAM をメモリとする CPU の一連の命令処理の流れは、R 形式命令の場合 FETCH1、FETCH2、DECODE、EXE、WB とした。

3・2 タイミング

CPU と SDRAM コントローラの接続を図 6 に示す。昨年度設計した CPU に接続されている SRAM の代わりにコントローラを接続する。コントローラによって、CPU は SRAM、SDRAM を意識することなく今までと同じ方式でメモリへアクセスすることができる。しかしながら、タイミングという面からみると、SRAM がデータ読出しに 2

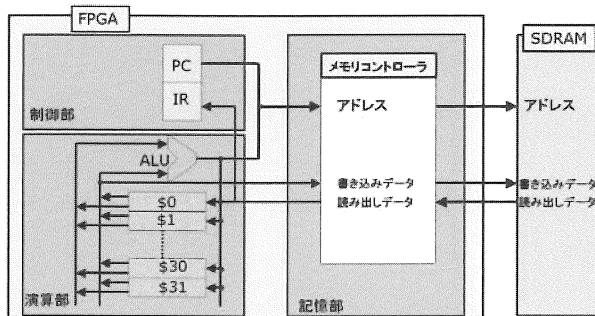


図 6 CPU・SDRAM コントローラ接続

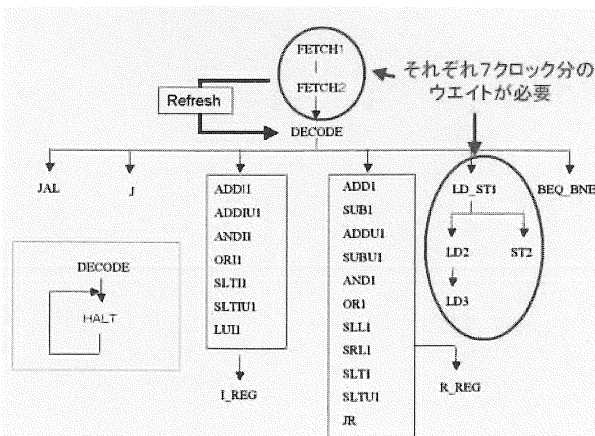


図 7 CPU の命令実行におけるウェイト挿入部

クロックかかるのに対して、SDRAM では 7 クロックかかってしまう。そのために、CPU の命令処理工程において、ウェイトを挿入しなければならない。ウェイト挿入部分について図 7 を示す。

CPU がメモリアクセスを行うフェッチステージにおいて 7 クロック分のウェイトを挿入する。また、メモリからデータを読み出す、書き込むというロード・ストア命令の実行ステージにおいてもウェイトを挿入しなければならない。このウェイト挿入がタイミングという面においての CPU 部の変更点である。

次に、SDRAM における状態を考慮したとき、SDRAM コントローラはリフレッシュ動作中、他のアクセスを拒否する。もし、このとき CPU がアクセスすることを前提に処理を進めていれば命令、データを得ることができず誤動作となる。そのため、CPU とコントローラ間において、リフレッシュ動作をどこで行わせるかの強調が必要となる。そこで、コントローラ部はリフレッシュカウンタからリフレッシュが行われるまで要求を出し続けるようにし、CPU 側では要求有無判別の地点をフェッチでの最後のウェイト部分で行う。そのため、リフレッシュカウンタには CPU 一命令実行分の余裕を持たせた。

3・3 論理合成

コントローラ設計時の動作確認作業と同様に、メモリへプログラムを書き込むためのモジュールを付加した。書き込むプログラムは先述したものの他に、乗算プログラム (16 ビット×16 ビット (49×13))、ユークリッドの互除法プログラム (入力データ 18 と 12) の三つを用いる。それぞれのテスト・モジュールを付加した論理合成により得られた平均結果を表 2 に示す。

表 2 論理合成平均結果

CPU with SDRAM	ロジアナIPなし
最大動作周波数	101.05MHz
使用LE (全10570個)	1548個
使用内部メモリ (全920448bit)	3883bit
使用PLL(全6個)	1

3・4 実機検証

第 2 節同様、ロジアナ IP を用いて実機検証を行い、3 プログラムとも 82.50MHz においての正常動作を確認した。これ以上の周波数では SDRAM からの読出しデータが壊れてしまい、それを命令として読み込んだ CPU はこちらの意図する動きはなさない。この 82.50MHz という結果

は、コントローラ単体において 88.00MHz であり、それに CPU モジュールが付加され全体的に動作周波数が減少したという考えからも正しいと言える。また、論理合成結果の最高動作周波数の比較において、第 2 節と同様の結果となった。原因についても第 2 節と同様の事が考えられる。

4 結言

本研究では、CPU 設計手法とその応用を研究する目的で、SDRAM をメインメモリとする MIPS CPU の FPGA 化を行った。そのために以下のことを行った。

- 1) SDRAM コントローラを製作し、簡単なプログラムの書き込み、読み出しを行い、正常動作することを確認した。
- 2) 同コントローラを使用し、SDRAM とマルチサイクル方式の CPU の結合を行い、CPU として正常動作することを確認した。

最大動作周波数は、コントローラ単体では 88.00MHz、CPU 結合時では、82.50MHz であった。

参考文献

- 1) 特許庁"特許からみた「プログラマブル・ロジック・デバイス(Programmable Logic Device:PLD)」の展望"
http://www.jpo.go.jp/shiryoutoushin/chousa/tt1301_062.htm

- 2) 角田一幸,渡邊誠也,正木亮"32 ビット RISC プロセッサ FPGA 上への実装"平成 12 年度 電気・情報関連学会中国支部連合大会
- 3) 田中康一郎,林悠平"FPGA/DSP ベースシステムによる組み込みシステム設計教育"電子情報通信学会論文誌 D-I Vol.J87-D-I No.6, pp.640~pp.648 2004
- 4) 安永守利,高見知親"FPGA を用いたナノ秒オーダ画像認識ハードウェア"電子情報通信学会論文誌 D-II Vol.84-D-II No.10, pp.2280~pp.2292 2001
- 5) 荒木英夫,久津輪敏朗"FPGA による組み込み制御を目的としたプロセッサシステムの実装と評価"電子情報通信学会論文誌 C Vol.J86-C No.8, pp.799~pp.807 2003
- 6) 小堀友義,丸山勉"FPGA を用いた格子ガスオートマトンの高速計算"情報処理学会論文誌 Vol.42 No.7 2001
- 7) CQ 出版,"トランジスタ技術" p136 11,2004
- 8) 堀田厚生"半導体の基礎理論" p203 技術法論社 2000
- 9) Gerry Kane 著,前川守 監訳,"mips RISC アーキテクチャ-R2000/R3000" pp.5~pp.22 共立出版株式会社 1992
- 10) David A.Patterson,John L.Hennessy 著,成田光彰訳"コンピュータの構成と設計"上下巻 日系 BP 社 1999
(受理 平成 17 年 3 月 17 日)