

極めて低コストで効率的な VDEC チップ試作・検証システムの開発と応用

若杉 祐太^{†1} 佐藤 真平^{†1} 植原 昂^{†1}
藤枝 直輝^{†1} 渡邊 伸平^{†1} 高前田 伸也^{†1}
森 洋介^{†1} 吉瀬 謙二^{†1}

計算機アーキテクチャ研究において実際にチップを試作し評価することは、研究に説得力を持たせる上で大いに重要である。しかしながらチップの試作・検証は多くのコストと手間を必要とする。本稿では、我々が開発した VDEC チップ試作・検証システムについて述べる。本システムは、非常に安価かつ少ない時間でチップの試作と検証をおこなうことを可能にする。また 32bit RISC プロセッサをはじめとする、実際の開発事例を紹介する。

Development of Low-Cost Verification Scheme for VDEC-assisted Prototype Chip and its Application

YUHTA WAKASUGI,^{†1} SHIMPEI SATO,^{†1} KOH UEHARA,^{†1}
NAOKI FUJIEDA,^{†1} SHIMPEI WATANABE,^{†1}
SHINYA TAKAMAEDA,^{†1} YOUSUKE MORI^{†1}
and KENJI KISE^{†1}

In computer architecture study, it is very important to implement an idea and prototype a chip. This helps us to demonstrate the practicality of our study. But prototyping chip needs much cost and time. In this paper, we describe our verification scheme for VDEC-assisted prototype chip. This scheme enables us to prototype and verify a chip efficiently. We introduce some case studies which use this scheme.

1. はじめに

計算機アーキテクチャ研究において、自らのアイデアを実装し回路面積や遅延、消費電力等の評価をおこなうことは研究に説得力を持たせる上で非常に重要な要素である。このためには実際にチップを試作すること¹⁾⁻³⁾、あるいは各種の EDA ツールを用いてシミュレーションをおこなうことが必要不可欠である。しかし、特に前者のチップ試作は、日本のアーキテクチャ研究者の間で十分に普及しているとは言い難い。

我々は、チップ試作のメリットは次の 3 つであると考えている。

- (1) FPGA 実装やシミュレーションよりも詳細な評価をおこなうことが可能である。
- (2) 幅広い分野の人に対するデモンストレーション効果が期待できる。
- (3) 実際に動くものが作れるということは、学生に対して良い動機付けになる。

これに対し、一般に考えられるチップ試作のデメリットは次の 3 つであろう。

- (1) 多くの手間と時間を要する。
- (2) 多くの予算を要する。
- (3) 正しく動作する保証がない。

我々が開発したチップ試作・検証システムはこれらのデメリットを抑えるものである。具体的には、ローム 0.18 μ m を利用し 2.5mm 角のチップを試作する場合に、チップ試作・検証を 40 万円、3 人月でおこなうことが可能になる。さらに、我々は本システムにより試作したチップのうち 3 種類の動作を確認している^{*1}。もちろん、本システムによる試作にはいくつかの制限がある。例えば、スタンダードセルベースのデジタル設計に限る、高い周波数での検証には向かない等であるが、先に述べたチップ試作のメリット (2), (3) は十分に享受できる。

本稿の構成を述べる。我々が開発するチップ試作・検証システムのうち、チップ試作の効率化についてを 2 章で、チップ検証の効率化・低コスト化についてを 3 章で述べる。4 章、5 章ではケーススタディとして本システムを用いて我々が開発したチップについて述べ、6 章で本稿をまとめる。

^{†1} 東京工業大学 大学院情報理工学研究所

Graduate School of Information Science and Engineering, Tokyo Institute of Technology

*1 4 種類のチップを試作。うち 1 つは動作確認中。

2. 効率的なチップ試作

本章では、東京大学大規模集積システム設計教育研究センター VDEC の試作サービスを利用した効率的なチップ開発フローについて述べる。

2.1 チップ開発フローのねらいと試作可能なチップ

現在、全国の高専・大学の構成員は VDEC を通して提供される EDA ツール、製造ファウンドリの設計ライブラリを利用してチップ試作をおこなうことができる。しかしチップ試作は、EDA ツールの使い方や各製造ファウンドリのテクノロジーに依存する部分などで多くのノウハウを必要とする。そのため、新たにチップ試作を始めるグループは VDEC 主催の講習会に参加する等して十分に勉強する必要があり、新規参入の敷居が高い。

我々のチップ設計フローは、EDA ツールに熟練していない学生でも短い期間でチップ開発ができるようにすることを目的としている。そのために、設計フロー内で設計者が作業すべきポイントを入念に選定し、それ以外の部分は標準スクリプト等で高度に自動化を施す。これにより設計フローはシンプルになり、ノウハウを持たない学生でも、余計な EDA ツールの操作に悩まされることなくチップの機能設計に専念できるようになる。なお、本フローはローム 0.18 μm 京都大学版デジタル設計フローを大いに参考にし、設計用スクリプトの多くを流用している。

設計フローのシンプル化により設計の自由度はいくらか制限される。本設計フローで試作可能なチップについて以下にまとめる。

- 製造プロセスはローム 0.18 μm を利用する。
ローム 0.18 μm は最もユーザーが多く情報を入手しやすい。また、設計のライブラリも充実している。ダイサイズは 2.4mm 角～10mm 角で、メタルは 5 層まで利用できる。
- 設計はスタンダードセルベースのデジタル設計に限る。
スタンダードセルのライブラリは、ロームが提供するものと京都大学が提供するものが利用できる。SRAM のマクロも提供されているが、我々のフローでは現在は利用できない。アナログ設計のマクロへの対応は今後の課題である。
- 階層設計が利用できる。
モジュール毎に配置配線し、上位階層でサブモジュールを結合する。複数の階層を持つチップが開発できる。
アーキテクチャ評価の初期段階に利用するならば、以上の制限は問題にならない。しかし、より本格的な評価をおこなうには CAM, RAM 等のアナログ設計のマクロへの対応が必須

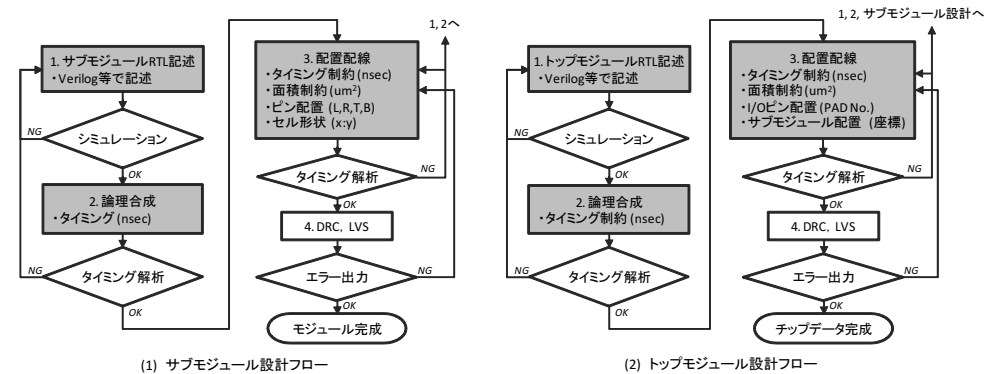


図 1 VDEC チップ開発フロー。

の課題である。

2.2 VDEC チップ設計フロー

我々が用いる VDEC のチップ設計フローを図 1 に示す。図 1 左はサブモジュールの設計フロー、右は最上位モジュールの設計フローである。網掛けの操作はチップ設計者が作業すべきポイントである。以下に、それぞれの操作の説明と留意点を示す。

1. RTL 記述

Verilog-HDL や VHDL といったハードウェア記述言語で RTL 記述を作成する。階層設計をおこなう場合は、この段階でモジュールの切り分けを考える。

2. 論理合成

階層設計をおこなう各サブモジュール毎に論理合成をおこなう。論理合成には標準スクリプトを用いるため、設計者はタイミング制約のみを最長パスの遅延 (nsec) により指定する。この段階では配線遅延は理想化して見積もるため、設計の制約よりもある程度短い遅延を指定する。タイミング解析の結果、所望の速度が得られない場合や、合成後のネットリストによる Verilog シミュレーションがうまくいかない場合には RTL の見直しをおこなう。

3. 配置配線

各サブモジュール毎に配置配線をおこなう。配置配線も論理合成同様に標準スクリプトを用いる。設計者は、目標とする最長パスの遅延 (nsec)、面積 (μm^2)、モジュールの入出力ピンの位置 (上下左右のどの辺から引き出すか)、セルの形状 (縦横比) を指定する。長方形以外のモジュールを設計する場合には標準スクリプトは利用できない。ターゲットの面積

表 1 チップ設計に使用するツール.

用途	ツール	バージョン
Verilog シミュレーション	Verilog-XL	05.10.004-s
論理合成	Design Compiler	A-2007.12-SP3
配置配線	Astro	Z-2007.03-SP7
DRC, LVS	Calibre	v2007.3.18.11

表 2 評価ボードを利用する場合の各手法の特徴

	市販のボードを利用	業者に設計を依頼	チップ毎に独自開発 (提案方式)
コスト	×	×	
手間			
仕様の柔軟性	×		
高周波数での検証			

は論理合成時に見積もられたセルの合計面積の 1.2~1.7 倍程度を指定する. 本フローでは, 配置配線後に容量抽出とタイミング解析を自動でおこなう. 解析の結果, タイミングを満たさない場合は, 配置配線時のスクリプトを High Effort 用に変更して, 再度配置配線をおこなうか, RTL 設計の見直しをおこなう.

最上位モジュールの配置配線の際には, 各サブモジュールの位置を座標で指定する. また, チップの入出力ピンの位置を PAD の番号で指定する.

その他の留意点

DRC, LVS はローム 0.18um 用のルールファイルが提供されているためそれを利用する. エラーが検出された場合には, 配置配線の制約を多少甘くすると消える場合が多い. どうしても数個エラーが残る場合は, エラー箇所を直接手書きで修正することもできる.

また使用する EDA ツールについて, 我々が利用しているものを表 1 に示す. VDEC からダウンロードできるツールは複数のバージョンがあるが, 対応する OS がそれぞれ異なるので注意が必要である. 我々は, これらのツールを x86-64 版 CentOS 4.6 にインストールして使用している.

3. 低コストで効率的なチップ検証システム

本章では, 我々が開発する低コストなチップ検証システムについて述べる.

3.1 検証システムのねらい

試作したチップの動作検証には, VDEC の各拠点に設置されているテストを利用する方法と, I/O 回路等が実装されたチップ評価用のボードを利用する方法がある. 特に後者の評価ボードを利用する方法は, 手で検証ができる点, I/O を利用した大規模な検証ができる点で魅力的である. しかし, 市販の評価ボード⁴⁾ は数十万円以上と非常に高価で, また構成が固定されているために柔軟性に欠ける場合がある. 業者にボード設計を依頼する場合には, オーダーメイドが可能で品質も期待できるが非常に高価である.

そこで我々は, 試作したチップ毎に小規模な評価ボードを独自開発するという手法をと

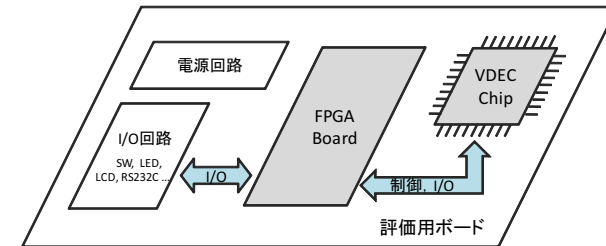


図 2 VDEC チップ評価用ボード.

る. この手法は一見すると非効率的に思われるが, 実際には 1 枚あたり数千円と非常に低コストで, かつ設計を再利用することで手間もそれほど必要としない.

それぞれの手法の特徴を表 2 にまとめる. 我々の手法はコストや柔軟性の面で優れるが, 高い周波数で検証を行いたい場合には適さない. なぜならば, 高周波の回路設計は特別の知識を必要とし, ボード設計の手間が増加するためである. しかし, 数 10MHz 程度での検証ならば専用の設計を行わずとも十分に可能である. また, 我々の手法では評価システムをチップの数だけ大量に生産することが可能である. さらに, それぞれのチップに専用でボードを設計するため必要最低限の機能のみを実装すればよく, 評価システムをコンパクトに抑えることができる. このことから, 評価システムを持ち運んでデモンストレーションを行いやすいというメリットも期待できる.

3.2 VDEC チップ評価用ボードの開発

我々が開発する VDEC チップ評価用ボードのブロック図を図 2 に示す. 主要なコンポーネントは, FPGA ボードを接続するためのコネクタと, VDEC のチップを実装するためのパターン, 電源回路, 各種 I/O 回路である. FPGA の柔軟性を生かして, チップの制御と I/O は FPGA ボードを介しておこなう. 各コンポーネントについてポイントとなる事項を以下に示す.

VDEC チップ

チップをボードに実装する方法として、ボードにソケットを実装する方法と基板表面にチップを直接実装する方法がある。前者はソケットが高価であるが、多くのチップを検証したい場合には適している。後者の表面実装は、低コストであるがチップを基板に実装する手間がかかる。我々は、チップのパッケージには 160 ピン程度の QFP(Quad Flat Package) を利用することが多く、それほど実装の手間を必要としないため表面実装を採用している。

FPGA ボード

評価用ボードにはコネクタのみを実装し市販の FPGA ボードを接続する。FPGA ボードの選定には注意が必要である。FPGA ボードは VDEC のチップと接続するため、それぞれの I/O 電圧が一致することが望ましい。例えばロームの 0.18 μ m でチップを試作した場合、I/O 電圧は 3.3V である。

チップと FPGA ボードの接続

チップの I/O ピンはクロックやリセット信号も含め、必要なものを FPGA に接続する。接続の際には、クロックは最短距離で優先的に引くとよい。また、リセットのように敏感な信号は、高速で ON/OFF しノイズ源となる信号^{*1}と並走して配線すべきではない。数 10MHz 程度での低い周波数の動作を目標とする場合でもこれらに注意する必要がある。

また、チップの入出力ピンは FPGA ボードとの接続に加えて、評価ボード上に汎用のピンヘッド等により引き出しておくことでデバッグの際に便利である。デバッグにはロジックアナライザが利用できる。

I/O 回路

チップ毎にボードを設計する場合、I/O 回路は必要なものを個別に選定できる。我々は、初期段階の検証には LED を、大量のデータの入出力を確認する場合にはインテグラル電子のコマンドインタプリタ液晶 ITC-2432-035H を用いている。後者は、シリアル通信の規格に則ってコマンドを転送することで液晶に文字を表示できるものである。コントローラが簡単に作成可能、接続する信号線は 1 本のみと手軽であることからデバッグ用途に適している。

電源回路

例えばロームの 0.18 μ m プロセスで試作した場合には、コア電圧として 1.8V、I/O 電圧として 3.3V を供給する必要がある。我々は、定電圧レギュレータにより入力電圧をそれぞ

*1 例えばクロック。

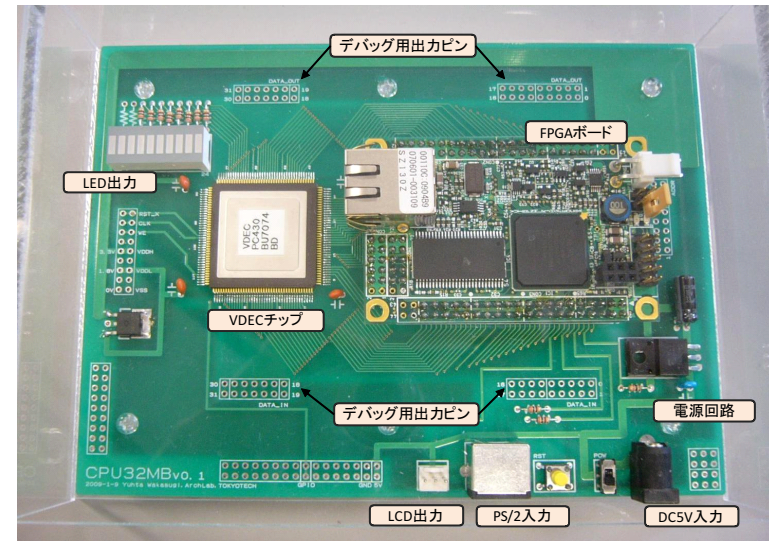


図 3 我々が開発した VDEC チップ評価用ボードの一例。

れの電圧に降圧して利用している。また、チップの電源ピン付近にノイズ除去用のコンデンサをいくつか実装している。

3.3 評価用ボードの開発例

図 3 に、我々が開発した VDEC チップ評価用ボードを示す。12cm × 16cm のコンパクトな 2 層両面基板に、前節で示したコンポーネントが実装されている。本ボードは我々が試作した 32bitRISC プロセッサ用に開発したものである。このチップの詳細は次章で示す。FPGA ボードには Atmark Techno 社の SUZAKU-S(Spartan3E, 120 万ゲート) を利用した。I/O には、PS/2 キーボード入力と、コマンドインタプリタ液晶用 LCD 出力、LED、リセット入力用のプッシュスイッチを用意した。また、ロジックアナライザを用いたデバッグ用に、チップの入出力は全て外部に引き出した。

ボードの設計は、フリーのプリント基板作成ツールである PCBE を利用し、プリント基板の製造は P 版.com を通じて株式会社インフローに依頼した。この程度の規模のボードならば、基板の製造にかかるコストは 1 枚あたり 2 千円弱である^{*2}。

*2 20 枚製造した場合。

4. ケーススタディ1：32bitRISC プロセッサ

2章,3章で述べた低コストで効率的なチップ試作・検証システムの利用例として32bitRISCプロセッサの開発事例を紹介する。

4.1 試作の概要

我々は、シンプルで効率的なメニーコアアーキテクチャM-Core⁵⁾を開発している。本試作では、メニーコアプロセッサの計算コア部分の初期検討として、MIPS32命令セットの大部分を実装するマルチサイクルプロセッサを設計した。チップの設計には京都大学提供版のスタンダードセルライブラリを利用し、ローム0.18 μm プロセスにより試作をおこなった。また、評価用ボードを設計し動作の確認をおこなった。

4.2 チップの開発

本試作の32bitCPUは、命令フェッチ、命令デコード、オペランドフェッチ、実行、メモリリード、メモリライト、ライトバックの7ステージを有するマルチサイクルプロセッサである。パイプライン化を施していない理由は、検討の初期段階であることとチップ試作の経験が浅いことから、デバッグのしやすさを優先したためである。チップ化する部分は計算コアのみとし、チップ外部に用意したメモリと接続する。実装する命令は、MIPS32命令セットの計算コア命令から浮動小数点と積和等の特殊命令を除いた70種である。

実装はVerilog-HDLによりおこない、シミュレーションにはVerilog-XLを用いた。シミュレーション結果の検証には、MIPSシステムシミュレータSimMips⁶⁾を用いた。検証の手順は次の通り。まず、テストプログラムをVerilogシミュレータと、SimMipsの両方で走らせる。そして、1命令実行するごとに汎用レジスタの値をログとして出力し、両者のログを比較する。我々は、C言語で記述した比較的大規模なテストプログラムを用い、先頭400万命令が一致するまでRTLの検証をおこなった。

DesignCompilerによる論理合成後のシミュレーションでは、スタンダードセルの面積が 0.30mm^2 、ロジックのみの遅延が 6.35nsec であった。クリティカルパスは実行ステージの32bit加算器である。Astroによる配置配線の際には、目標の面積をスタンダードセルの面積の1.3倍に、タイミング制約を 12nsec とした。配置配線、容量抽出後のシミュレーションでは面積が 0.39mm^2 、最長パスの遅延が 11.1nsec であった。

実際に試作したチップのダイ写真を図4に示す。チップの左辺がメモリからのデータの入力、右辺がメモリへのデータ出力である。アドレスは下辺に配置し、クロック、リセット、メモリのライトイネーブルは上辺にまとめた。

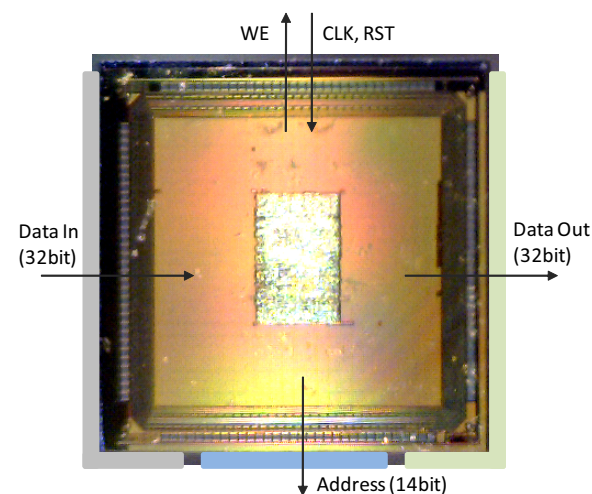


図4 32bitRISC プロセッサのダイ写真。ダイサイズは2.4mm角である。

4.3 チップの検証

チップの検証は評価用ボードを開発しそれを用いておこなった。開発した評価ボードは、2章で例として取り上げた図3のものである。32bitCPUはメモリを内蔵していないため、FPGA内部にブロックRAMを用いて52KBのメモリを構築し、これをメインメモリとして用いた。チップのクロックは、FPGAで発振器のクロックを倍周したものを利用した。

C言語を用いて、ナイトライダー^{*1}程度の簡単なプログラムを作成し実行したところ30MHzで正常に動作した。配置配線時のシミュレーション結果である90MHzより大幅に下回る理由は、本チップが外部のメモリを1サイクルでリード/ライトする設計であるためである。このように、クリティカルパスがボード上の配線に渡る場合は、チップの動作周波数の限界は測定できない。チップの遅延を詳細に測定したい場合は、別途テストを利用する必要がある。

4.4 開発工数とコスト

本試作に要した開発工数について述べる。まずチップ試作に関しては、修士課程の学生1

*1 いくつかのLEDを光が往復するように光らせるトイプログラム。

名と学部学生 2 名の計 3 名で 2 週間を要した。このうちの大部分は RTL の検証作業に費やしている。評価用ボードの設計は、修士課程学生 1 名が 1 週間を費やした。本ボードはゼロからの設計であったが、2 回目以降同様のパッケージで試作する場合には、ボードの設計を再利用することで開発工数を短縮できる。検証に要した時間を含めると、トータルの開発工数は 3 月程度である。

コストについて述べる。本試作はローム 0.18um プロセスにて 2.4mm 角のチップを 20 個製造し、そのうちの 15 個を QFP160 でパッケージングした。これらのコストはトータルで 30 万円程度である。ボード設計に要したコストは、20 枚を製造して 4 万円程度である。ボードに実装する細かな部品を合わせてもトータルの費用は 40 万円未満^{*1}で、この費用で評価システムをチップの個数である 15 セット構築できるのは非常に魅力的である。

以上まとめると本試作は 40 万円、3 人月と非常に低コストかつ効率的であったと言える。

4.5 VDEC チップを利用した計算機システムの構築

本チップの活用例として、シンプルな計算機システムを構築する。図 5 は対象とする計算機システムである。計算機の入出力として PS/2 キーボードとコマンドインタプリタ液晶を用いる。これらを利用するために、FPGA 上にキーボードコントローラ (PS/2 KB Controller) および、液晶コントローラ (LCD Controller) を実装する。これらの I/O はメモリマップド I/O を利用してプログラムから制御する。そのため、メモリマップド I/O を司る Memory Controller も FPGA に実装する。メインメモリには、FPGA 内のブロック RAM を 52KB のシングルポートメモリとして合成したものを利用する。

本計算機システムで動作するアプリケーションは C 言語で記述し、MIPS のクロスコンパイラでコンパイルすることで作成できる。コマンドインタプリタ液晶を利用するための C 言語のライブラリを用意している。図 6 は UNIX コマンドの `sl`^{*2} を本計算機システム向けにポータリングし、実機で動作させている様子である。高級言語で記述されたアプリケーションがそつ無く動作するさまは計算機としての資質を十分に感じさせてくれる。

5. ケーススタディ 2: オンチップルータ

5.1 概要

我々は、多機能オンチップルータによりメニーコアプロセッサのディペンダビリティを向

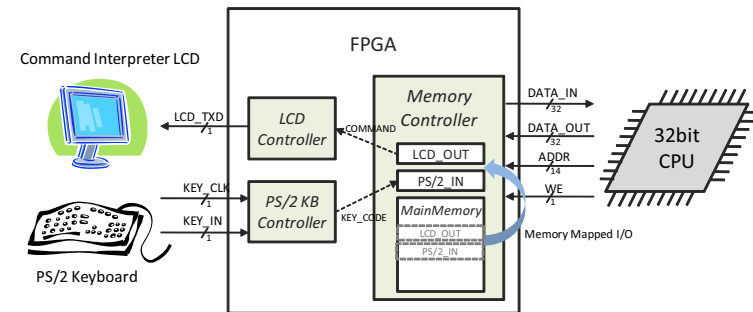


図 5 VDEC チップを利用したシンプルな計算機システム。

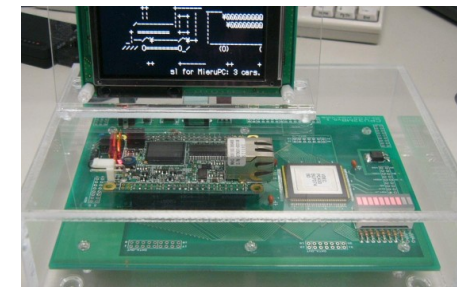


図 6 VDEC チップを利用した計算機システムでアプリケーション `sl` が動作する様子。

上する SmartCore システム⁷⁾ を提案している。本試作では SmartCore システムの主要な構成要素であるオンチップルータにターゲットを絞り、基本的なルータを 2 x 2 のメッシュ状に接続するチップを設計した。チップの設計には京都大学提供版のスタンダードセルライブラリを利用し、ローム 0.18um プロセスにより試作をおこなった。また、動作確認のために評価用ボードを開発した。

5.2 チップの開発

本試作のオンチップルータは、慶應大学松谷宏紀博士提供の RTL 記述をベースに開発したものである。ルータのアーキテクチャは、仮想チャネルのないシンプルな 3 サイクルルータで、4 エントリ x 5 本 (N,E,S,W,Core) の入力バッファをもつ。

DesignCompiler による論理合成後のシミュレーションでは、1 つのルータあたりスタンダードセルの面積が $0.15mm^2$ 、ロジックのみの遅延が最大で $2.3nsec$ であった。ルータは

*1 FPGA ボードは取り外し可能であるため、費用として計上していない。

*2 コンソール画面をアスキーアートで書かれた SL が駆け抜けるアニメーションプログラム。

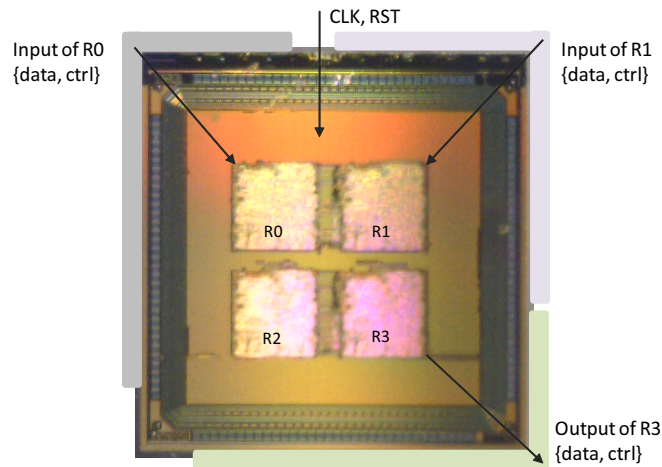


図 7 2 × 2 構成オンチップルータのダイ写真．ダイサイズは 2.4mm 角である．

クロスバスイッチが占める割合が大きく配線に多くの面積を必要とする．そのため、配置配線の際にはターゲットの面積をスタンダードセルの面積の 1.65 倍と大きめに設定した．タイミング制約を $10nsec$ として Astro で配置配線をおこない、配線遅延を含むシミュレーションをおこなった結果、1 つのルータあたりの面積が $0.25mm^2$ 、最大遅延は $8.4nsec$ であった．

チップ化の際には、これらのルータモジュールを 2.4mm 角のダイ内部に 4 つ配置した．実際に試作したチップのダイ写真を図 7 に示す．左上のルータ (R0) と右上のルータ (R1) のコアからの入力をチップの入力とし、右下のルータ (R3) のコアへの出力をチップの出力としている．クロックはチップの上辺から供給する．

5.3 チップの検証

チップの動作検証のために設計した評価ボードを図 8 に示す．32bitCPU 用に設計した図 3 の評価ボードと大部分の設計を共有していることが見て取れる．I/O には、スイッチ入力と LCD 出力、LED 出力を用意した．スイッチにより設定した値をルーティングし、LED に出力するというデモが可能である．

5.4 開発工数とコスト

本試作に要した開発工数について述べる．まずチップ試作に関しては、既存の RTL を利

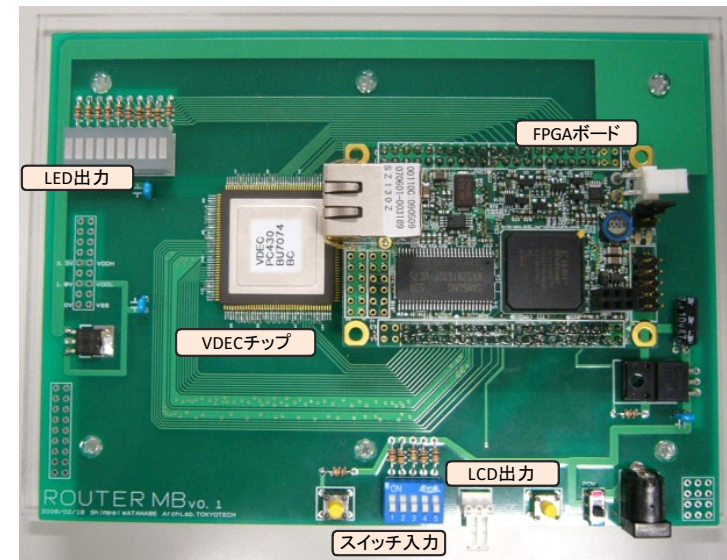


図 8 オンチップルータ評価用ボード．

用したこともあり修士課程の学生 1 名がおよそ 2 週間を費やすことで開発することができた．評価用ボードの設計に関しても、32bitCPU のボードの設計データを利用することで、3 日程度で完成することができた．

コストについては 32bitCPU と同様、チップとボードの製造にかかる費用を合わせておよそ 40 万円である．

6. おわりに

多くの手間とコストを要するというチップ試作のデメリットを軽減するために、低コストで効率的なチップ試作・検証システムを開発した．本システムを用いると、ローム 0.18um にて 2.5mm 角のチップを試作する場合のコストを、わずか 3 人月、40 万円に抑えることが可能になる．また、実際の開発事例を紹介し本システムが十分実用的であることを示した．

今後は、開発したチップ試作・検証システムをアーキテクチャ研究に積極的に利用し、よりリアリティのある研究を進めていくことが望まれる．

謝辞 本研究は東京大学大規模集積システム設計教育研究センターを通し、ローム株式会社、凸版印刷株式会社、シノプシス株式会社、日本ケイデンス株式会社、メンター株式会社の協力で行われたものである。また、本研究の一部は科学技術振興機構・戦略的創造研究推進事業 (CREST) 「アーキテクチャと形式的検証の協調による超ディペンダブル VLSI」の支援による。本稿中で紹介するオンチップルータの試作にあたり、RTL 記述を提供して頂いた慶応大学松谷宏紀博士に感謝いたします。

参 考 文 献

- 1) Chengjie Zang, Shigeki Imai, and Shinji Kimura: Issue Mechanism for Embedded Simultaneous Multithreading Processor, The 20th Workshop on Circuits and Systems in Karuizawa (KARUIZAWA-2007), pp.325-330 (2007).
- 2) 長谷川 揚平, 堤 聡, 中村 拓郎, 西村 隆, 佐野 徹, 加東 勝, 齊藤正太郎, 天野 英晴: 動的リコンフィギャラブルプロセッサ MuCCRA-1 の実装と評価. 先進的計算基盤システムシンポジウム (SACSI2007), pp.95-102 (2007).
- 3) 関直臣, Lei Zhao, 徐慧, 池淵大輔, 小島悠, 長谷川揚平, 天野英晴, 香嶋俊裕, 武田清大, 白井利明, 中田光貴, 宇佐美公良, 砂田徹也, 金井遵, 並木美太郎, 近藤正章, 中村宏: MIPS R3000 プロセッサにおける細粒度動的スリープ制御の実装と評価, 情報処理学会研究報告 2008-ARC-176, pp.71-76 (2008).
- 4) 三菱電機マイコン機器ソフトウェア株式会社: *MU200-SX*, <http://colordial.jp/powermedusa/products/sx.html>.
- 5) 植原昂, 佐藤真平, 森谷章, 藤枝直輝, 高前田伸也, 渡邊伸平, 三好健文, 小林良太郎, 吉瀬謙二: シンプルで効率的なメニーコアアーキテクチャの開発, 情報処理学会研究報告 2008-ARC-180, pp.39-44 (2008).
- 6) 藤枝直輝, 渡邊伸平, 吉瀬謙二: SimMips: 教育・研究に有用な Linux が動く 5000 行の MIPS システムシミュレータ, コンピュータシステム・シンポジウム (ComSys2008) 論文集, pp.143-150 (2008).
- 7) 吉瀬謙二, 植原昂, 佐藤真平: メニーコアプロセッサのディペンダビリティ向上と高性能化を目指す SmartCore システム, 情報処理学会研究報告 2008-ARC-180, pp.49-52 (2008).