

FPGA を用いたマルチプロセッサシステムテストベッドの実装

尾形 航^{*1} 山本 泰平^{*2} 水尾 学^{*2} 木村啓二^{*1} 笠原博徳^{*1}

^{*1} 早稲田大学 理工学部 電気電子情報工学科

〒169 東京都新宿区大久保 3-4-1

^{*2} 株式会社写真化学 システム機器事業部 電子デバイス部

〒602 京都府京都市上京区東掘通り一条上ル堅富田町 436-5

マルチプロセッサシステムのアーキテクチャ研究において対象アーキテクチャを評価するには、ソフトウェアでエミュレートを行う、或いは実際にハードウェアを製作して評価する等の手法が取られてきた。しかし前者の手法では評価に膨大な時間を要するという難点があり、一方後者の実際に製作する方法では多大な費用がかかる、開発から実際に稼動して評価に入るまでに相当の期間を要する、また一度製作した機体を改造して別のアーキテクチャの評価を行う事が事実上不可能、等の問題があった。これらを解決する高速・浮動小数点演算機能を含む高機能・過大でない費用で実現可能・容易にアーキテクチャを改変可能なアーキテクチャエミュレータの製作を計画し、S-RAM 型の大規模な FPGA を用いて実装した。アーキテクチャの中心となる CPU は 32 ビット同時処理でかつ浮動小数点演算機能を含み最高 30MIPS/FLOPS のパフォーマンスを発揮して SPEC や Perfectclub といった実用に即したベンチマークプログラムを走行させる能力を持つ。S-RAM 型の FPGA の特性を活かし異なる複数のアーキテクチャ向けにデータ転送、同期機構、スケジューリング機構等の性能向上に有効と思われる機構を追加するなど、さまざまな試行を容易に行える。

Implementation of FPGA Based Architecture Test Bed For Multi Processor System

W. Ogata^{*1} T. Yamamoto^{*2} M. Mizuo^{*2} K. Kimura^{*1} H. Kasahara^{*1}

^{*1} Department of Electrical, Electronics and Computer Engineering, Waseda University

55N-05-05A, 3-4-1 Okubo, Shinjuku-ku Tokyo 169 Japan

{ogata,kimura,kasahara}@oscar.elec.waseda.ac.jp

^{*2} System Equipment Division, Electronics Device Department, SHASIN KAGAKU CO., LTD.

Ichijo-Agaru Higashi Horikawa-Dori Kamigyo-ku Kyoto 602 Japan

There are two major approaches to evaluate new multi processor system architecture, the evaluation with software emulator on WS or high-performance PC. The other is the evaluation on hardware emulator. However, it takes very long time on the evaluation with the software emulator to evaluate new architecture using large benchmark programs. On the other hand, it is expensive to develop a machine. Also, it takes long time to develop new machine. Even if we have hardware, it is very difficult to modify it. So, evaluation of different architecture is difficult by using the same hardware. To cope with these problems, we have been planning an architecture emulator, which is high-performance with floating point arithmetic unit; not so expensive cost, reconfigurable easily. It is combination of S-RAM based large scale FPGAs (Field Programmable Gate Array). On the emulator using FPGAs, 32-bit CPU-CORE including Floating-Point Unit with 30MIPS/MFLOPS is implemented. This offers us to run large application (benchmark) programs such as SPEC or Perfect-Benchmark. With the feature of S-RAM based FPGA, we can implement different architectures or we can implement special mechanism for efficient parallel processing such as data transfer control unit, synchronization unit, and task scheduling unit and so on.

1 はじめに

従来よりアーキテクチャの研究に際して対象アーキテクチャの性能評価を行うには大別して (1) WS やメインフレーム上にソフトウェアでエミュレータを構成する (2) 実際に対象アーキテクチャをハードウェアとしてインプリメントするなどの手法が取られているが、ソフトウェアエミュレーションは必然速度が非常に鈍くサンプルプログラムをシミュレータにかけて評価するには多大な時間を要する。特に近年は SPEC や Perfectclub 等の実用に即したアプリケーションプログラムを用いた評価が必須となり評価結果を得る迄の時間も長くなる。当研究室の例では Pentium-II(233MHz)パソコン上で簡易なシミュレータを構築した場合は単一 CPU のエミュレーションで約 200KHz 相当、メンテナンス性を重視して C++で構築した場合は 10KHz 相当のパフォーマンスとなり、マルチプロセッサシステムでは更にパフォーマンスが漸減する。この為一つの評価データを取る為に数時間～数日を要する事も珍しくなく研究の妨げとなる。

一方ハードウェアで対象アーキテクチャを製作する方法では専用ハードウェア製作に多額の費用がかかる、ハードウェア製作に多大な時間を要する、ハードウェアの変更が困難という問題がある。学生にハードウェアを製作させるには配線の引き回し等の実装やノウハウを欠き、安定して動作する機体を得るのは困難であるし外部の業者に製作を委託するには多額の予算が必要となり、また、対象アーキテクチャを設定して予備評価、回路の設計、基板の設計、実装、デバッグ、調整を経て評価に至るまでに多大な時間を要する。そして、一度製作した機体のアーキテクチャを変更して別のものにする事は事実上困難であり、複数のアーキテクチャを比較検討する上で致命的な障害となる。これらの問題を解決し、十分な演算速度を持つ・アーキテクチャの設定/変更が容易である・アーキテクチャを設定してから評価を行なうまでの時間を短縮できる・過大でない予算で十分実現可能であるといった要素を満たすシステムを実現するために、プログラマブルなデバイスを使用してアーキテクチャを評価するシステムの実現について検討した。図 1 に各種エミュレーションのパフォーマンスの対比を示した。原点より上方へは CPU 当たりの性能を示し、右方へはエミュレーションする CPU 台数を示す。下の斜線で囲まれた網掛け部分がソフトウェアエミュレータのパフォーマンス、上方の矩形の網掛け部分がハードウェアを製作した場合のパフォーマンスを示す。今回開発したアーキテクチャエミュレータの CPU(以下 Type-X と称する) 1 基のパフォーマンス並びに実装したエミュレータの 4 個の CPU を稼働させたときのも合わせて示す (4 Type-X)。

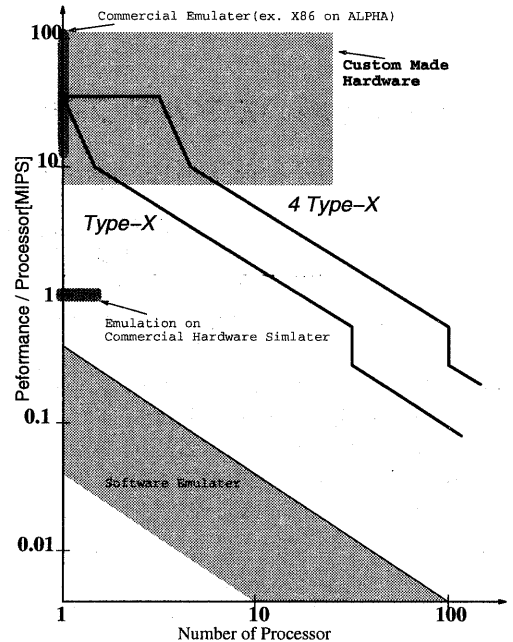


図 1. 各種手法の比較

2. プログラマブルデバイスベースの

ハードウェアエミュレータ

実際にエミュレータを実現する為に下記の検討を行った。

2.1 十分な演算速度を持つ この目的を達成するにはハードウェアを主体としたシステムを構築するのが適当であり、大前提でもある。以下はこの前提の下に検討を行う。

2.2 アーキテクチャの設定変更が容易である アーキテクチャ研究には特にこの特長が欠かせない。廉価なヒューズ型・アンチヒューズ型の機種があるが再プログラムが不可能であり複数のアーキテクチャを試行する目的には不適である。また電氣的書換え可能な機種もあるが回数に制限があることや専用のプログラマに差し替える必要があるなど不利である。本稿では大規模な S-RAM ベースの機種を使用し多数のアーキテクチャを存分に試行することを検討する。

2.3 アーキテクチャを設定してから評価を行うまでの時間を短縮できる 前項にも関連するが、特に複数のアーキテクチャを比較する場合には対象アーキテクチャを設計し、これをシステムに反映してすぐに評価に移れる特性が必要であり、消去の手間やデバイスの差し替えの必要がなくオン・ボードでプログラミングできる S-RAM ベースのプログラマブル・ロジック・デバイスが望ましい。

2.4 過大でない予算で十分実現可能である プログラマブル・ロジック・デバイスなど高機能の LSI を採用する場合には、単にデバイスの単価のみならずプログラミング用の

ツールや回路設計用プログラム等の CAD の導入・維持費用も考慮に入れなければならない、全体のコストを勘案する必要がある。

上記の条件を満たすプログラマブルデバイスを検討し、本稿では入手性なども考慮して ALTERA 社 FLEX 10K シリーズを選択した。

3. 使用するプログラマブルデバイス

本設計では米 ALTERA 社の開発したプログラマブルデバイス FLEX10K シリーズを使用する。これはコンフィグレーション情報を S-RAM に貯える。4 入力 LUT (Look Up Table) と FF の組を 8 組纏めた LAB (Logic Array Block) や、S-RAM 機能を設定できる EAB (Embedded Array Block) を基本要素に持ち、これらの要素間を高速バスで縦横に接続する。1LAB がほぼ 8 ビットの加算器とレジスタ (フリップフロップ) に相当し、EAB に 2048 ビットの RAM や ROM を設定できる。本稿ではこの中でも公称値 10 万ゲートの機種 (EPF10K100) を使用する。

4 プログラマブルロジックデバイスベースの CPU

手許で回路をインプリメント出来るプログラマブル・ロジック・デバイスの特性を学校の教育用システムに活かし[1][2]、これを使用して小型の CPU を実装する学生実験や実習の例が既に幾つか報告されている。大規模なものでは DLX[3]と呼ばれる簡易な 32bit-RISC プロセッサをプログラマブル・ロジック・デバイスを使用して製作した例もある。一方、アーキテクチャ研究を中心として対象アーキテクチャの性能評価を行う立場からは先にも述べたように大型の実用アプリケーションを処理できる能力が欠かせない。とりわけ実用アプリケーションでは実数を処理する機会が多く、これを効率的に処理する工夫が必要となる。

4.1 浮動小数点演算処理を含む

プログラマブル・ロジック・デバイスベースの CPU

プログラマブル・ロジック・デバイスを使用した CPU など浮動小数点演算回路まで含めて公表されたインプリメント例は殆ど無い。以下に浮動小数点演算機能を持つ CPU をプログラマブルデバイスに実現する方法について検討する。

a) 整数演算命令に依るエミュレーション 整数演算命令を組み合わせる指数部の処理、仮数部のシフトや演算を行ってエミュレートする方法では CPU は整数演算命令のみをサポートすれば良く、ハードウェア面ではもっとも簡便に済む利点があるが性能が極端に低下する不利を被らざるを得ない。

b) シーケンサやマイクロプログラミングによる浮動小数点演算 使用素子数に強い制限があった初期のコンピュータシステムや FPU で少ない回路で複雑な演算を提供するうえで非常に有効だが速度の面で不利である上マイクロプログラム等を構成する必要があるなど別の手間をかけなければならない。

c) 浮動小数点演算補助命令の装備 仮数部処理命令・指数部処理命令など複数の補助命令を用意しこれらを組み合わせて浮動小数点演算を実現するが演算パイプラインを構成した場合には一連の補助命令間でパイプラインハザードを生じてパフ

ォーマンスを悪化させる。また指数部や仮数部の演算回路を各々用意するなら、浮動小数点演算回路そのものを用意するのと比べて回路規模に大差がなくなる。

d) 浮動小数点演算回路の装備 浮動小数点演算回路をそのままインプリメントする場合に最も問題になるのが回路規模である。指数部と仮数部につき別途回路が必要になる他、仮数部のシフトを行うバレルシフタ、乗算命令で使用される乗算器は特に大規模な回路となるが、今後のアーキテクチャ研究においては十分な素子数を前提として浮動小数点演算のインプリメントすることも妥当と判断する。また、LUT の特性を活かした実装を行って回路資源を節約し、カスタム LSI に比べて容量で劣る FPGA 上で実現する工夫をする。

4.2 プログラマブルデバイス上での

浮動小数点演算器の実現

十分な素子数を使用できるという前提の下に浮動小数点演算回路そのものをインプリメントする。IEEE-754 の単精度 (指数部 8bit, 符号 1bit, 仮数部 24bit) に準拠したデータタイプを処理する演算回路をインプリメントするが例外処理、NaN などでは当初のインプリメントではサポートしない。また誤差の丸め処理などは当初のインプリメントでは省略する。

4.2.1 浮動小数点加減算器

浮動小数点加減算での作業は、指定された演算の種類、2 数の符号や指数部の差により幾つかの類形に分けられる。同符号数の和、あるいは異符号数の差を求める時には単に桁合わせをして加算をすればよい。同符号数の差、あるいは異符号数の和を求める時には各々の数の指数部の差によって 3 つ (何れが大きいかも区別すると 5 つ) の類形を処理することになる。

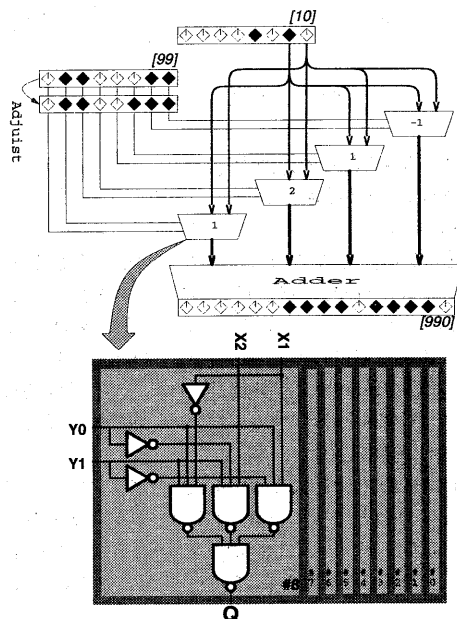


図 2. LUT の特性を活かした部分積

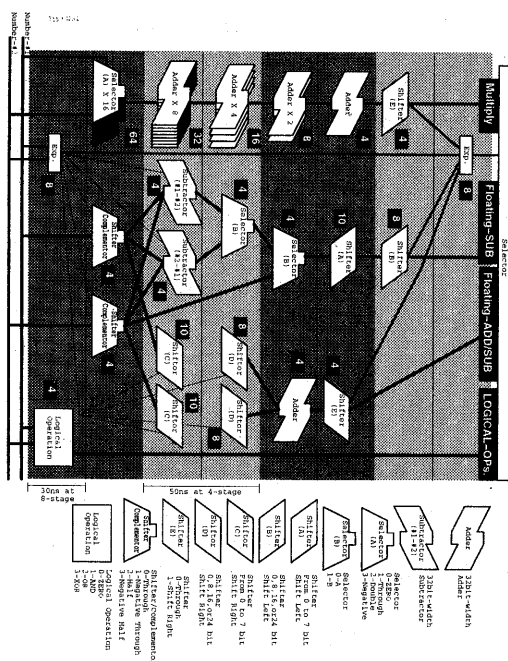


図 3. 整数/浮動小数点演算器の構成

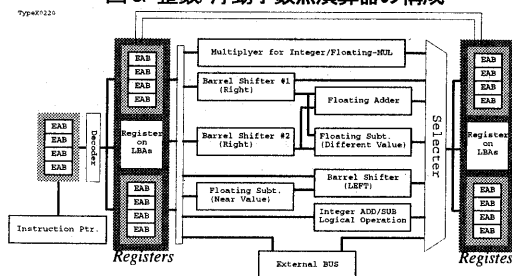


図 4. 32bitCPU 全体の構成

4.2.2 整数·浮動小数点乗算器

乗算器は最も回路資源を要求し、この節約がプログラマブルデバイス上にインプリメントする際のポイントとなる。

a) **通常の加算** 通常の加算による方法では第 1 段で個々の部分積(32×32→1024 項)を表現する為に 128LAB、そして 32 項を加え合わせる為に第 2 段階以下各段に各々 64,32,16,8,4LAB を用いて 124LAB、計 252LBA を要する。これは EPF10K100 の回路資源の半分近くで、他の実験を行うための回路を併せてインプリメントする事を考えても削減出来れば望ましい。

b) **BOOTH のアルゴリズム** 或いは **BOOTH のアルゴリズム**がある。乗数を 2 ビットずつに区切って各々の部分項を 0,1,2,N の符合で代表して(2 次の BOOTH のデコーダ)、3 ビットの情報で被乗数の加工を指示する。カスタム LSI 等

では加算器の数を減らす等の利点があるが、セクタへの入力が5本になり4入力LUTにおいては1ビットの処理の為にLUT2個を要し実装効率が悪くなる。

c) 4 入力 LUT の特性を活かした実装 上記の手法を一部改変したものを用いて 4 入力 LUT で効率的に処理できるようにする[4]。各部分積に 0,1,2 を乗じる所までは同じだが、3 を乗じる際に、被乗数に-1 を乗じた値を出力すると同時に上位の部分積(当該部分積の 4 倍の数を扱う)にて被乗数に乗じる部分項に 1 を加える。これで被乗数に 4 倍を乗じた数値から被乗数に 1 倍を乗じた数値を引き、結果として被乗数に 3 を乗じた数値を得られる。尚、被乗数に-1 を乗じた値を得るには、2 の補数を利用する。4 入力 LUT では、その内の 2 入力を被乗数の入力(一方は被乗数を其の仮入力し、もう一方は被乗数を 1 ビット左にシフトしたものを入力する)とし、残る 2 本の入力で被乗数に乗ずる部分項 0,1,2,-1 を選択する。これにより 1 ビット当たり 1 個の LUT で済み(図 2)、単純な加算の場合や BOOTH のアルゴリズムを使用する場合に比べて約半分の資源で済む。これらを組み合わせて整数浮動小数点演算器を構成した様子を図 3 に、また、32 ビット CPU 全体の様子を図 4 に示す。

5. 実装

実際にエミュレータを実装した様子を以下に示す。実装には株式会社写真化学の協力を頂き、プログラマブルデバイス評価用ボードに改良を加えていただいた。図 5 に単一プログラマブルデバイスのモジュール(型番 VS-BOARD AM4-RAM)の写真を示す。これはデバイス一個と 512Kbit S-RAM8 個、72pin-SIMM メモリスロット 2 基を持ち、このモジュールで完結したコンピュータシステム(PE)を構成する。S-RAM をキャッシュメモリとして、また SIMM メモリを主記憶として使用しメモリの階層構造をエミュレートする事も可能である。このモジュールを 4 枚、メインボード(VS-BOARD AB3)上に装着し、ボード上に実装されたネットワーク用デバイスを介して相互に接続する。

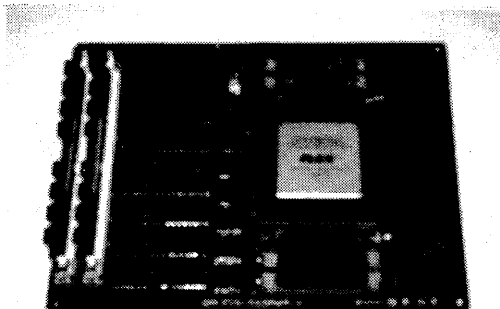


図5. 単一モジュールの写真

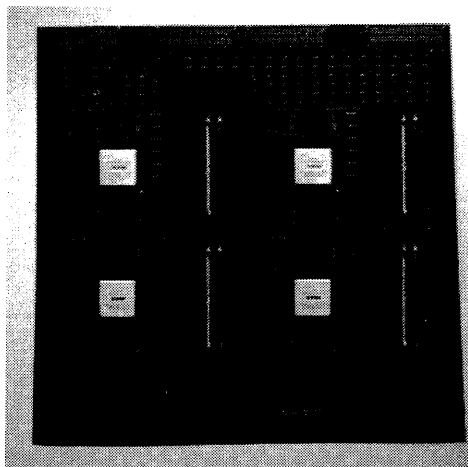


図6 システム全体の構成

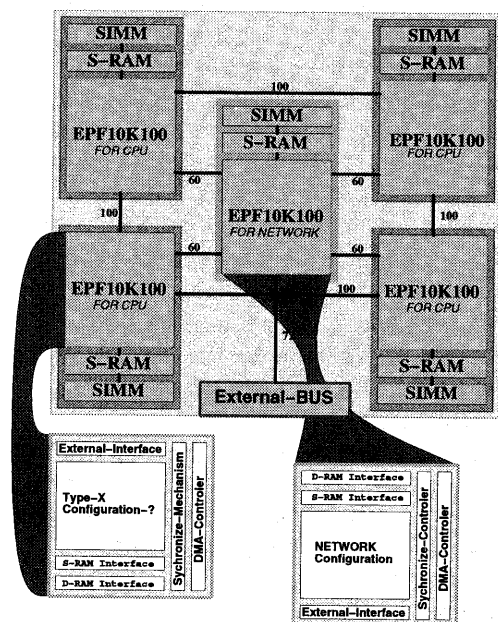


図7 システムの概略図

これにより、全体としてマルチプロセッサシステムを構成することが出来る。

図7に全体の概略図を示す。ネットワーク用デバイスにもS-RAMとSIMMが接続されており、これらをマルチプロセッサシステムの集中共有メモリとして使用することも可能である。

6 応用

実装したシステムの応用について述べる。コンフィギュレーションを自在に変更して異なるアーキテクチャを試行する利点を活かして、各アーキテクチャの利点等を比較検討するこ

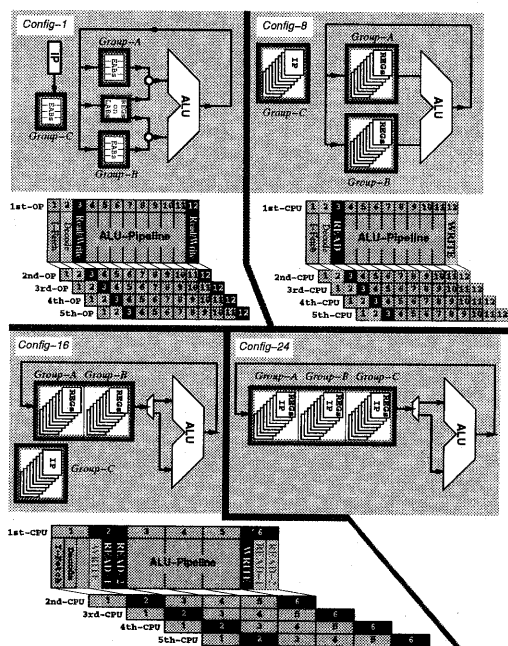


図8 パイプラインの構成と複数CPUのエミュレーション

とも容易に出来る。

6.1 パイプラインの構成と複数CPUのエミュレーション

アーキテクチャ変更の例として命令/パイプラインの変更の例を示す。

a) パイプラインの変更

図8のConfiguration-1は最もピッチを短くした場合のコンフィギュレーションを示し最もパフォーマンスが向上するが其の一方でパイプラインハザードが生じ易くなる。パイプライン構成を変更し、最適なパイプライン段数を調査する事も容易にできる。

b) 複数のCPUのエミュレート

パイプライン技術の派生として複数のCPUのコンテキストを順次パイプラインに投入してマルチプロセッサシステムをエミュレートする事も考えられる。図8のConfiguration8~24は、その場合のパイプラインの構成を示したもので今回実装したシステムでは単一デバイスで最大24CPU(1CPUに32ビット幅レジスタ30本割りを割り当てた場合)をエミュレート可能であり、システム全体では96CPUのエミュレートを行える。この場合でも1CPU当たり1乃至2MIPS/MFLOPSのパフォーマンスを維持し、最適化を施したソフトウェアエミュレーションでは単一CPUのエミュレートで1MIPSを達成できるか否かという状況に比べると格段にパフォーマンス向上を望める。

C) スーパースカラRISC, VLIWのインプリメント

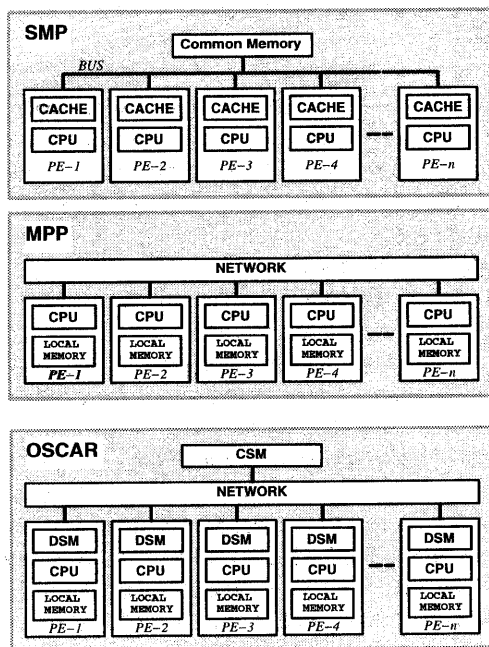
今回実装したエミュレータはCPU用に4個のデバイスを搭載しており、見方を変えれば4基のALU, FPUを用いることが出来る。これらはネットワークを担当するデバイスで相互に多

数の配線で結ばれており、全体として複数の演算器を持つ単一のCPUのエミュレーションを行える。これを発展させてスーパースカラ型RISCやVLIWのアーキテクチャエミュレーションに応用できる。

D) シングルチップ・マルチプロセッサのインプリメント
本来個々のデバイスに個別のCPUをインプリメントする前提でエミュレータを設計しており、多数の配線で結ばれるネットワーク担当デバイスに共有レジスタをインプリメントし、全体でシングルチップ・マルチプロセッサをエミュレートする。

6.2 メモリ構成

従来は並列処理マシンの構成はSMP型(図9上)、すなわち共有メモリに複数のプロセッサが接続された形式、或はMPP(図9中)に見られるように分散メモリを個々のプロセッサに接続し、プロセッサ間のデータ転送はネットワークを介して行なう形式のものがある。当研究室で並列処理マシンに適したメモリ構成として提案しているものを図9下に示す。これは、コンパイラがどのように変数を配置するかを解析した結果、集中共有メモリ並びに分散共有メモリの両方を持つアーキテクチャが望ましいとする結果を得て構成したものである。既にこのアーキテクチャを持つマルチプロセッサシステムOSCARを製作、稼働させてきたが、本エミュレータにより多くのベンチマークを走行させて充実した評価を得る。



DSM : Distributed Shared Memory
CSM : Centralized Shared Memory

図9. メモリ構成

7 まとめ

LUTの特徴を活かす工夫を施して現在入手できるFPGAの内、大型の機種の上に単精度の浮動小数点演算機能を含む32bit同時処理で最大30MIPS/MFLOPSの性能を持つCPUをインプリメントできることを示した。また、これをベースに過大でない予算ながら改変可能で高いパフォーマンスを持つエミュレータを実装し、その応用例を示した。

本稿で述べたエミュレータ(マルチプロセッサシステムテストベッド)は昨年中に実装を終了し、現在主に論理方面のデバッグ中である。設計上(プログラマブルデバイスのCADによる予測)は30MIPS/MFLOPSを達成する見込みであり、一部の回路を動作させて試験した範囲でも同様の速度を実現できるものと思われる。近日中に浮動小数点演算機能付き32bitRISC型のコンフィギュレーションで稼働を開始し計算機アーキテクチャの研究に資する。

従来は専門のベンダなどでなければ自らの研究に向けたアーキテクチャのマシンを製作したり、提案したアーキテクチャを容易に実機にする機会はまず訪れなかった。然し大容量のプログラマブルデバイス、機種が提供される様になり、これらの研究にも容易に手をつけられるようになり、非常に好ましい状況になっている。最先端のプロセッサは500MHzで稼動し、GIPS以上の性能を発揮する。これに比べれば本設計で述べたシステムの様な設計ではパフォーマンスとしては10分の1~100分の1でしかない。然しソフトウェアエミュレータに比較すれば格段の性能を発揮しアーキテクチャ研究に資するところは大きくあると思われる。或いは数年前の初期のi486DX(33~66MHz)やPentium(66MHz)に比べれば、特に浮動小数点演算や乗算の類のパフォーマンスで優る計算機を自ら設計できるという事を考えても本設計の意義はあるのではないかとと思われる。

最後に、実際の機材の製作を快く引き受けて頂き、提供して頂いた(株)写真化学様の皆様にも紙上を借りて御礼申し上げます。特に窓口となって頂いた水尾様、そして実際の設計の指揮を執っていただいた山本様には大変お世話になりました。ありがとうございました。

参考文献

- [1]末吉井上奥村久我「教育用32ビットRISCマイクロプロセッサDLX-FPGAと教材ボードの開発」第3回FPGA/PLD Design Conference & Exhibit 応用技術論文集p579-588(1995)
- [2]鶴田阿部「16bit RISC プロセッサ pecc の設計と評価」第3回FPGA/PLD Design Conference & Exhibit 応用技術論文集p579-588(1995)
- [3]DAPatterson, J.L.Hennessy, "Computer Architecture: A Quantitative Approach (Second Edition)", Morgan Kaufmann Publishers, Inc.(1996)
- [4]尾形山本水尾「FLEX10Kを用いたマルチプロセッサシステムシミュレータの実装とこれを用いた評価」ALTERA PLD WORLD '97 技術論文集 p.117-138 (1997)