

汎用エンジン RM-II の構成

富田 昌宏[†] 澄川 文徳[†]
菅 沼 直 昭[†] 平野 浩 太 郎[†]

電氣的に書替え可能な FPGA をメモリと組み合わせることによって、複数の用途への適用を可能とする汎用エンジンの概念に基づいて開発された RM-II (Reconfigurable Machine-II) について述べる。汎用エンジンは、FPGA 上に実現するワイヤード論理によって専用エンジンに近い性能を得る一方で、複数の応用に対応可能である点に特徴をもつ。最初のプロトタイプである RM-I において問題となった規模と柔軟性の不足を解決することを RM-II 開発の主眼とした。実現可能な回路規模およびメモリ容量を倍増させるとともに、FPGA 間の配線を変更するための FPGA の導入によって柔軟性を高めた。通常は一つの CPU で行う処理を複数の FPGA に分割して実現することが一般的な汎用エンジンでは、データパスが複数の FPGA にまたがるが多く、多様な通信を短時間で完了する必要がある。さらに、FPGA の外部端子数に関する制約も厳しい。そこで、FPGA とメモリの接続線を流用して X/Y 方向のバスと接続するクロスバスを導入することで、多様な転送を多くの場合 1 クロックで実現した。論理シミュレーション、論理設計誤りの診断と画像処理に適用した。その結果、RM-I の約 2 倍の性能が得られることを確認した。

A Reconfigurable Machine : RM-II

MASAHITO TOMITA,[†] FUMINORI SUMIKAWA,[†]
NAOAKI SUGANUMA[†] and KOTARO HIRANO[†]

This paper presents a Reconfigurable Machine-II (RM-II) based on the concept of Reconfigurable Machine, capable of efficiently implementing a wide range of computationally complex algorithms on its flexible architecture combining FPGA's and memories. RM-II has been built to provide higher flexibility and larger capacity than the first prototype: RM-I. RM-II doubles the target circuit size and memory storage, incorporating one FPGA for flexible interconnection among other FPGA's. On a Reconfigurable Machine, where one process is often implemented by using multiple FPGA's, a datapath may cross several FPGA's. In addition, existing FPGA's have limited number of external pins. Thus, a cross-bus is employed for data exchange among multiple FPGA's using connections between FPGA's and memories. RM-II has been applied to logic simulation, diagnosis, and image processing. The results have shown about twice performance as high as that on RM-I.

1. はじめに

大規模化する論理回路の設計期間を短縮するために、CAD における特定の処理を高速に実行する専用エンジンが提案・開発されてきた¹⁾。これらのエンジンは、ソフトウェアによる処理と比較して 2 桁から 3 桁の処理速度向上を達成している。しかし反面では、特定の処理向けに徹底した専用化が図られているために処理の柔軟性に欠け、また大きな筐体を必要とするという問題がある¹⁾。一方、汎用の並列計算機を用いた並列処理によって、高速処理を達成する試み²⁾も行われている。この方式では、プログラム言語レベルで

の粒度の粗い並列性に頼らざるをえないため、専用エンジンに比べて処理速度が低下する。マイクロプログラム制御を導入したエンジン³⁾は、ワイヤード論理を用いた専用エンジンと汎用並列計算機の間で位置付けられるが、その構造上の制約から確保できる柔軟性には限界がある。以上のことから、CAD の処理を高速化する場合、柔軟性・汎用性と処理性能間のトレード・オフから逃れることは不可能であった。

この問題を解決するために、われわれは電氣的に書替え可能な FPGA をメモリと組み合わせることによって、複数の用途への適用を可能とする汎用エンジン (Reconfigurable Machine) の概念を提案し、そのプロトタイプとして RM-I を開発した⁴⁾⁻⁶⁾。図 1 にその構成を示す。対象とするアルゴリズムをハードウェア上に実現する実行モジュールと、ホスト・コンピュー

[†] 神戸大学大学院自然科学研究科
The Graduate School of Science and Technology,
Kobe University

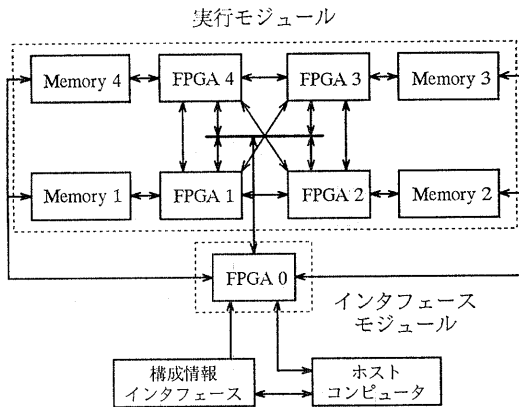


図 1 RM-I の構成
Fig. 1 RM-I architecture.

たとの通信を行うインタフェース・モジュールから成る。各 FPGA 間は、おもにデータ転送に利用する共通バスと、データ転送と制御の両方に利用可能な完全結合の信号線で結ばれている。

汎用エンジンは、RM-I の例で示されるように、電氣的に内部の論理を書き替えることができる複数の FPGA (Field Programmable Gate Array)⁷⁾ とメモリによって構成される。ゲートレベルでプログラム可能とすることによって、ワイヤード論理を用いたエンジンを柔軟に構築可能とし、高い柔軟性と高い処理性能の両立を実現する。

汎用エンジン上での動作手順の例を図 2 に示す。まず、ホスト・コンピュータから送られる構成情報 (Configuration Data) によって FPGA の内部機能を設定する。次に、必要に応じて汎用エンジンのメモリに初期データを設定する。汎用エンジンは、ホスト・コンピュータからの指示によって処理を開始する。結果がメモリに格納される場合は、インタフェース・モジュールを介してホスト・コンピュータに転送する。

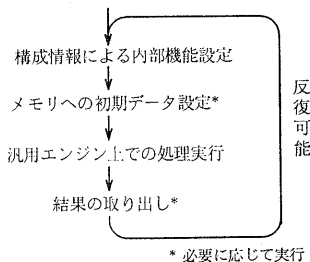


図 2 汎用エンジン上での動作手順例
Fig. 2 Processing sequence on Reconfigurable Machine.

異なる処理を実行する際には、対応する別の構成情報によって FPGA を再設定する。

汎用エンジンと同様の概念に基づいて開発されたマシンとして、Splash⁸⁾ と AnyBoard⁹⁾ が挙げられる。Splash では 32 個の FPGA を直線的に並べるリニア・アレイ構造を採用して、各 FPGA 間に隣接する FPGA からアクセス可能なメモリを配置しており、一次元のパターン・マッチング処理に適している。しかし、FPGA 間の結合形態がリニア・アレイで固定されている点と、任意のメモリへのアクセスが困難な点等によって、応用分野が限定される。一方の AnyBoard では 5 個の FPGA が直線的に配置され、18 ビットの共通バスで接続する。中央部の 3 個の FPGA はそれぞれ RAM (128K×8 ビット) のデータ線と接続するが、必要な外部端子数を節約するため、左端の FPGA から出力される共通のアドレスによってのみアクセスされる。そのため、独立したアドレスで各メモリを参照するパイプライン処理には適していない。

汎用エンジンの概念は、搭載する FPGA の数やメモリ容量、FPGA 間接続、メモリ配置、外界とのインタフェース等を規定するものではない⁴⁾。これらは、対象とする複数の処理がもつ特性を考慮して決定すべきであり、その結果、CAD 用途に限らず、信号処理・画像処理等の広い分野への応用も考えられる。したがって、Splash や AnyBoard も汎用エンジンに含まれるが、柔軟性の不足によってその適用範囲が限られると考えられる。

本稿では、RM-I の問題点を明らかとするとともに、その解決を図るべく開発した柔軟性の高い汎用エンジン: RM-II の構成について述べる。

2. 汎用エンジン RM-II の仕様

汎用エンジンの一実現例である RM-I の問題点を示した上で、RM-II の仕様を示す。特に、RM-II の結合網に求められる項目について詳しく述べる。

2.1 RM-I の問題点

汎用エンジンの最初の実現例である RM-I は、比較的小規模の問題を対象として設計された。FPGA には、電氣的に書き換え可能な Xilinx 社の XC 3090⁷⁾ を用いた。XC 3090 は、論理機能を実現する 320 個の CLB (Configurable Logic Block) と、入出力バッファの役目を果たす 144 個の IOB (Input/Output Block) によって構成される。

論理設計に含まれる誤りを自動的に特定する論理診

表 1 RM-I と RM-II の仕様比較
Table 1 Specifications of RM-I and RM-II.

項目	RM-I	RM-II
実現可能な回路規模	約20Kゲート	約45Kゲート
総メモリ容量	384Kバイト	768Kバイト
実行モジュールに搭載する FPGA	XC 3090×4	XC 4005×9 (結線変更用に1個)
メモリバンク数	4	8
メモリビット構成	32K×24ビット	32K×24ビット
FPGA 間の配線	固定	可変
結合網	完全結合の固定配線 共通バス	変更可能な配線 クロスバス

断手法と論理シミュレーションについて、高速に実行するエンジンが RM-I 上に構築された。その結果、従来の専用エンジンに近い処理速度の向上効果が確認された。しかし、以下の問題点も明らかとなった。

- FPGA 間の配線が固定されているため、FPGA の端子割当に関する柔軟性に欠ける。
- FPGA 間の完全結合により、モジュールの拡張や複数モジュールの結合が困難となる。
- メモリバンク数の不足により、パイプライン処理の途中で待機時間が必要となる場合がある⁶⁾。

2.2 RM-II の仕様

前述の問題点を解決するべく決定した RM-II の仕様を RM-I との比較を含めて表 1 に示す。RM-II では、実現可能な回路規模、ならびに総メモリ容量を RM-I の約 2 倍とした。バンク数を 8 とすることにより、パイプライン段数に関する制約を緩和する。さらに、FPGA 間の配線を変更するための FPGA を用意することにより、柔軟性を向上させる。XC 3090 の上位モデルである XC 4005¹⁰⁾を FPGA として用いることにより、配線効率および内部資源利用率の向上と、システム・クロックの高速化を図る。

2.3 RM-II の結合網に対する要求

バンク数の増加によって、並列処理効率の向上が期待される。たとえば論理シミュレーションを実現する場合、入力信号値、出力信号値、接続情報、イベントを管理するタイム・ホイール、素子の機能等、多くの情報を格納する必要がある。これらを多くのメモリバンクに分散することにより、各テーブルに独立してアクセスすることができる。その一方で、バンク数の増加によって RM-I のような完全結合の実現は困難となる。また、共通バスによるデータ転送では、通信量

の増加への対応が困難となる。

汎用エンジンの結合網に求められる項目を列挙する。

- 一対多の通信を実現できる。
- 複数の独立した通信間で競合する割合が低い。
- 遅れ時間が短い。
- 占有する FPGA の外部端子数が少ない。

汎用エンジン上では、通常は単一のプロセッサによって処理される内容が、一般に複数の FPGA に割り当てられる。よって、データバスが複数の FPGA に分岐する場合があります。一対多の通信を一対一の通信と同程度の時間内に行う必要がある。また、対象とする処理内容をワイヤード論理で実現することによってクロック単位の並列性を抽出し、処理速度の向上を図る。そのためには、通信の遅れ時間が 1 クロック以内であることが望ましい。さらに、XC 4005 では IOB の数、すなわち割当て可能な外部端子数が 112 と比較的少ない。32K×24 ビットのメモリへの接続に必要な端子数を除くと、約 70 となる。FPGA 間接続を可変とするための FPGA への接続をも考慮すると、残された端子数はわずかである。よって、多くの外部端子を占有することなく、通信を実現する必要がある。

以上の要求項目を考慮して、FPGA 間を複数の縦横のバスで結合するとともに、FPGA とメモリとの接続を流用することによって外部端子の占有を最小限に抑える、クロスバスによる結合を採用した。クロスバス結合や、ハイパー・クロスバス結合¹¹⁾と類似しているが、クロスバス結合では複数の一対多の通信を、多くの場合 1 クロックで完了する。

3. RM-II の構成

図 3 に RM-II の構成を示す。インタフェース・モジュールの構成については RM-I と同様である。以下、実行モジュールにおける、クロスバス結合を用いたデータバス、アドレスバスと、それらの制御について述べる。さらに、RM-II を支援するソフトウェア環境について述べる。

3.1 データバスとクロスバス・スイッチ

図 3 における二重線と太線は、それぞれ X および Y 方向について各 3 本のデータバス (24 ビット) を示す。アドレスバスについては後述する。FPGA 1~8 はそれぞれ対応する MEM 1~8 のメモリバンクに接続する。各 FPGA とメモリバンクを接続する信号線に SW 1~8 のスイッチを付加することにより、デー

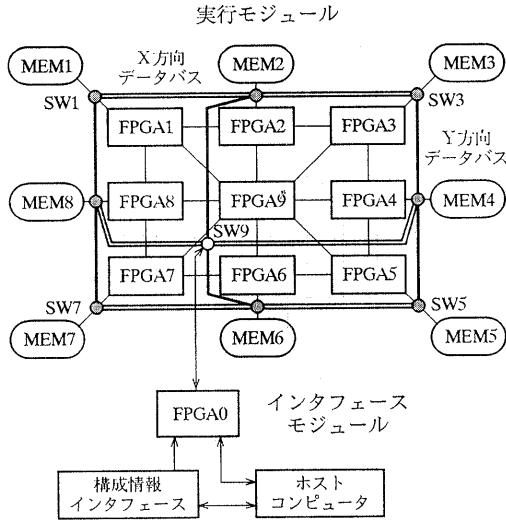


図 3 RM-II の構成
Fig. 3 RM-II architecture.

タバスと接続する。FPGA とメモリバンクとの接続信号線を流用することで、FPGA の外部端子を新たに占有することなくデータ転送路を確保できる。よって、残りの約 70 本の信号線を FPGA 間の結線に利用することができる。さらに、FPGA-FPGA 間の転送に留まらず、FPGA-メモリ、メモリ-メモリ間で転送を行うことも可能となる。ただし、FPGA 9 は FPGA 1~8 の間の配線変更や制御部の実現に利用され、対応するメモリバンクをもたない。

図 4 に、クロスバスを構成するためのスイッチの内部を示す。SW 1~8 については、図 4 (a) に示すように、3 分岐の任意の一つが信号源となり、他の 2 分岐の両方または一方に出力できる、3 組の双方向バッファによって構成される。FPGA とメモリについて

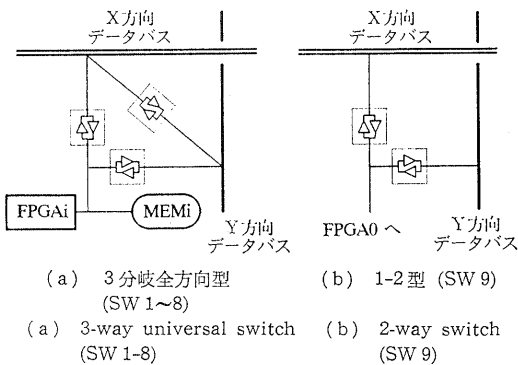


図 4 クロスバス・スイッチの構成
Fig. 4 Cross-bus switches.

も、それぞれ双方向の出力を利用する。1クロック内での転送を実現するため、各スイッチはレジスタ要素を含まない。

SW 9 については、図 4 (b) に示すように 1 対 2 の関係で双方向バッファを用いて接続している。FPGA 0 と X/Y 方向バスとのデータの授受を実現するとともに、X 方向バスと Y 方向バス間でデータを転送することも可能である。また、図には明示していないが、外周のバスを介して他のモジュールと交信することも考えられる。

3.2 アドレスバス

RM-I ではデータ転送用のバスのみが用意されていた。そのため、バスを介してアドレスを指定後、他のメモリバンクに格納されているデータを受け取るためには最低 2 クロックを要した。RM-II では、メモリ・アドレスを転送するバスを用意することによって、この操作を 1 クロックで完了する。図 5 にその構成を示す。アドレスバス (15 ビット) についてはデータバスに比べると利用頻度が低いことを考慮して、X/Y 方向に 1 行/列おきに間引いて配置している。各頂点 (SW 1, 3, 5, 7) については図 4 (a) で示した 3 分岐全方向型スイッチを利用するが、その他の部分には、1 組の双方向バッファで構成される単純な一対一のスイッチを用いる。SW 0 を介した FPGA 0 との接続は、ホスト・コンピュータからメモリバンクへの初期データ設定に利用される。通常、初期データ設定においては共通のアドレスが送られる。現在は、FPGA 1~3 を結ぶ X 方向アドレスバスに接続しているが、これには必然性はない。

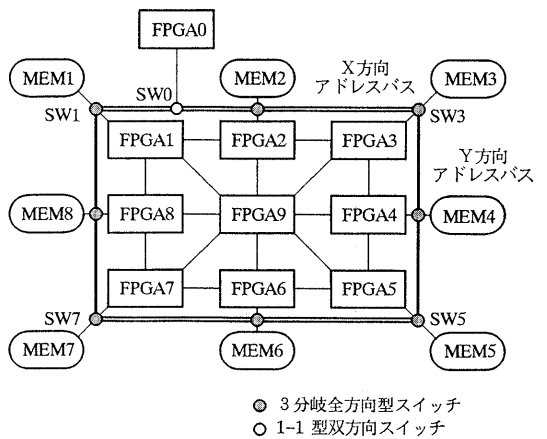


図 5 アドレスバスの構成
Fig. 5 Address busses.

アドレスを送る必要がない場合には、アドレスバスを第2のデータバスとして利用することもできる。ただし、この場合は FPGA-FPGA 間のデータ転送に限られる。

3.3 クロスバス・スイッチの制御

各スイッチの制御方法については特に規定を設けず、アプリケーションの開発者に委ねられている。RM-II の上で実現する回路を設計する際にあらかじめ静的に決定しておく方法と、動的に制御する回路を埋めこむ方法がある。動的制御の場合は、要求を受け付けて許可を与えるタイミングを決定しておく必要がある。例えば、1クロックの間にメモリを2回アクセスできるクロック周波数を選択した場合は、次に示すように前半をローカルなメモリのアクセスに、後半を必要に応じて通信に利用できる。

(1) クロック前半

- 対応するメモリバンク、すなわち FPGA_i からは MEM_i へのアクセスのみ認める。
- クロスバスを介した転送要求を受け付ける。

(2) クロック後半

- クロック前半で受け付けた要求信号をもとに（例えば FPGA 9 が）アービトレーションを行う。その結果に基づいて、クロスバスを介したデータ転送を行う。

ただし、これはあくまで一例であり、汎用エンジン上に実現するアプリケーションに応じて、適切なクロック周波数と制御方式を選択することが望まれる。

3.4 ソフトウェア環境

RM-II のアプリケーションを開発するためのソフトウェア環境について述べる。図6に構成情報を作成するまでの流れを示す。現状では Xilinx 社が提供しているソフトウェアをそのまま利用している。

(1) 動作レベル・シミュレーション

まず、RM-II による処理の対象とするアルゴリズムを動作レベルで表現し、シミュレーションを行う。現状ではC言語を用いて表現しているが、VHDL 等のハードウェア記述言語を用いることも考えられる。

(2) 各メモリ、FPGA への割付け

アルゴリズムで必要とする記憶要素、論理ブロックを割り付けるべきメモリバンクと FPGA を人手で決定する。同時に、FPGA 間結線とクロスバス上の転送経路の割付けについても人手で決定する。実際には、

(1)と並行して割付けを行う場合が多い。割付けが具体化した段階で、レジスタ転送レベルのシミュレーション

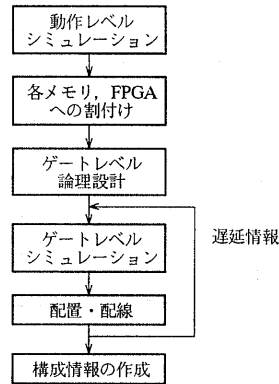


図6 RM-II のアプリケーション開発
Fig. 6 Design flow for RM-II applications.

によって、性能評価とデータの衝突の有無に関する検証を行う。

この工程は、(1)の結果とともに、アプリケーションの性能に大きな影響を及ぼす。一方で、人手による割付けの必要性は設計上の負担となっており、その自動化が課題となっている。

(3) ゲートレベルの論理設計とシミュレーション

各 FPGA に割り付けられた論理ブロックをゲートレベルに詳細化する。階層的論理設計が支援されている。レジスタや加算器等については、Xilinx 社で用意されたマクロや、ユーザが独自に作成したマクロの利用が可能である。必要ならばこの段階で、仮想遅延を用いた論理シミュレーションを行うことができる。

(4) 配置・配線

Xilinx 社が提供している配置・配線プログラムによって、自動的に FPGA 内部の配置・配線を行う。FPGA 1個の配置・配線に要する時間は実現する回路によって大きく異なるが、i486 DX 2 (66 MHz) を用いたパーソナル・コンピュータ上で処理を行うと、RM-I で利用していた XC 3090 では 3~6 時間を要していたが、RM-II で採用した XC 4005 については約 20 分で終了する。RM-II 上のすべての FPGA の設計変更を行う場合でも約 200 分で完了することになる。また、これまでの数十例に対する配置・配線において、未結線は発生しなかった。

配置・配線の結果、詳細な遅延情報が得られるので、必要ならば遅延を考慮したシミュレーションを行う。

(5) 構成情報の作成

合計 10 個の FPGA の配置・配線データをもと

に、構成情報を作成する。現状では、Motorola EXORMACS と呼ばれる ROM データのフォーマットを利用している。

4. クロスバス結合の評価

RM-II が採用するクロスバス結合によるデータベースと、交差部のスイッチ SW 9 を省いた結合方式を対象として、複数の独立したデータ転送を同時に行う際に、競合をおこさずに1クロックで完了させることが可能な組合せの割合を評価した。表 2 に結果を示す。同時に転送するデータの数 2~4 に対して、すべての組合せを計算機上で列挙した上で、競合の有無を判定した。ただし、各 FPGA はメモリと接続する一組の外部端子 (24 ビット) を利用してクロスバスに接続するため、一つの FPGA が同時に複数のデータ転送に関与することはないとする。「各データ転送に関与する FPGA 数の組合せ」について、例えば、(2, 6) は、2 個の FPGA 間の転送と同時に、それ以外の 6 個の FPGA 間の転送が発生することを示す。よって (2, 2, 2, 2) は、2 個の FPGA 間の転送が互いに独立して 4 組同時に発生することを示す。「FPGA の組合せ数」は、「FPGA 数の組合せ」に対して考えられる FPGA 1~8 の割当ての組合せ数を表す。クロスバスを利用する場合、2 種類の同時転送について、FPGA 数の組合せが (2, 2) に対しては 100%、つまりすべての組合せに対して競合を発生せずに 1 クロックで完了さ

表 2 1 クロックで転送可能な組合せの割合
Table 2 Ratio of possible combinations with concurrent data transfers.

同時転送データ数	各データ転送に関与する FPGA 数の組合せ	FPGA の組合せ数	1 クロックで転送可能な組合せの割合 (%)	
			SW 9 を除去	クロスバス
2	(2, 2)	210	97.6	100.0
	(2, 3)	560	90.7	100.0
	(2, 4)	420	79.7	93.9
	(2, 5)	168	64.3	81.0
	(2, 6)	28	50.0	64.3
	(3, 3)	280	78.6	94.3
	(3, 4)	280	65.7	87.1
	(3, 5)	56	50.0	71.4
3	(4, 4)	35	51.4	77.1
	(2, 2, 2)	420	42.4	73.3
	(2, 2, 3)	840	26.6	50.0
	(2, 2, 4)	210	19.5	39.5
4	(2, 3, 3)	280	16.4	30.7
	(2, 2, 2, 2)	105	7.6	15.2

表 3 同時転送データ数と 1 クロックで転送可能な組合せの割合

Table 3 Number of concurrent data transfers and ratio of possible combinations.

同時転送データ数	1 クロックで転送可能な組合せの割合 (%)	
	SW 9 を除去	クロスバス
2	79.5	92.8
3	27.9	51.2
4	7.6	15.2
平均値	54.4	72.0

* 平均値=FPGA の組合せ総数に対する割合

せることが可能である。一方、(4, 4) の場合は 77.1% となる。SW 9 を除去した場合は、それぞれ 97.6%、51.4% に低下する。

同時転送データ数ごとに整理した結果を、表 3 に示す。同時転送データ数が 2 以上のすべての組合せのなかで、72% については競合なく 1 クロックで転送可能であることが明らかとなった。特に、SW 9 を除いた結果との比較から、交差部のスイッチの効果が大きいことがわかる。

この結果は、ランダムに通信要求が発生した場合に対応している。3.4 節で述べたように、現状では手作業で転送経路を決定しているため、事前に可能な限り競合を避けるように FPGA への割付けを行うことで、処理性能の向上を図る。例えば、4 組の一对一の転送を同時に行う必要がある場合、各ペアを X/Y のデータベース上に配置することで、競合を回避できる。しかし、人手ですべての場合をつくすのは困難であり、その自動化が求められている。

5. 性能評価

RM-II の実装結果と性能について述べる。性能評価にあたり、RM-II 単体での性能のほか、論理シミュレーション、論理設計誤りの診断処理、画像処理を RM-II 上で実現し、性能を評価した。

5.1 RM-II の実装

RM-II のハードウェアを 1 ボード (450 mm×350 mm) 上にワイヤ・ラッピングで実現した。FPGA として Xilinx 社の XC 4005 を 10 個用いたほか、メモリバンクに 256 K ビットの SRAM (20 ns) を 24 個、またクロスバス・スイッチ等に 8 ビットのトライステート・バッファ 74 AC 245 を 119 個利用している。クロック周波数は、2, 4, 5, 8, 10 MHz から選択可

能である。

5.2 RM-II の単体性能

アプリケーションに依存しない性能評価の観点から、構成情報の転送時間と、クロスバスの最大伝搬遅延時間を評価した。

現状では、115 K バイトの構成情報をデージー・チェーン接続された FPGA の信号線を介して書き込んでおり、9.5 秒を要している。I/O タイミングの調整と各 FPGA への並列書き込みによって、0.14 秒程度まで短縮可能である。

クロスバスの最大伝搬遅延時間については、10 MHz の信号をアドレスおよびデータバスに与え、5 段のスイッチを通過する時間を測定した。その結果 30 ns 以下となり、比較的高速なデータ転送を行えることが確認された。

5.3 論理シミュレータへの応用

RM-I 上に構築した LSIM (Logic Simulator)⁶⁾と同様のアルゴリズムによる、割当遅延方式のゲートレベル論理シミュレータ LSIM-II を RM-II 上に構築した。イベント伝搬と素子評価の二つのフェーズのそれぞれについてパイプライン処理を行う。LSIM では、メモリバンク数の不足によってイベント当たり 2 クロックの待機時間が必要であった。RM-II ではメモリバンク数を倍増させたため、LSIM-II ではこの待機時間をなくすことが可能となった。

表 4 に LSIM-II の処理速度を示す。LSIM と同様に 4 MHz のクロックを用いた。特に平均ファンアウト数が少ない INV 1 K、OKI¹²⁾の回路例で処理速度向上の効果が大きい。これらの例では、イベント当たり約 2 クロックで処理を完了する。この点では専用エンジンによる SP(Simulation Processor)¹³⁾と同等と考えられる。

5.4 論理診断エンジンへの応用

ゲートレベルの回路に含まれる論理設計誤りを追跡・特定する論理診断の一手法である。拡張 X-伝搬

表 4 論理シミュレータ LSIM-II の処理速度
Table 4 Processing speed of logic simulator: LSIM-II.

回路名	ゲート数	処理速度 (万イベント/秒)	LSIM に対する 速度比
DDA 8	584	150	1.35
INV 1 K	1,000	180	1.86
OKI	420	186	1.91
演算器	156	179	1.38

表 5 論理診断エンジンの性能比較
Table 5 Performance comparison with Logic
Diagnosis Engines.

項目	LDE	LDE'	LDE-II
メモリサイクル/クロック	2	1	1
状態数	16	24	16
処理クロック数の比	1	1.85	0.95
クロック周波数	4 MHz	8 MHz	8 MHz
処理速度比	1	1.08	2.1

法における多値シミュレーション処理を、RM-I 上で高速に行う論理診断エンジン: LDE'⁴⁾がすでに開発されている。ここでは、RM-II 上で同様の処理を行う LDE-II について述べる。

表 5 に文献 4) と同一の 7 回路例に対する実験の結果に基づく性能比較を示す。LDE では 4 MHz のクロックを採用して各クロック当たり 2 回のメモリ・サイクルを用意することで、read-modify-write を行っていた。LDE-II ではクロック周波数を倍の 8 MHz としたが、遅延の問題で現状ではクロック当たり 1 メモリ・サイクルとしている。表の LDE' は、RM-I を用いて LDE-II と同様のクロックおよびメモリサイクルを利用した場合の仮想的な性能を示す。LDE' では状態数が 16 から 24 に増加し、処理に要するクロック数が 1.85 倍となるため、処理速度の向上は 1.08 倍にとどまると見積もられる。一方 LDE-II では、バンク数の増加とクロスバスの利用によって状態数を減らすことが可能となり、メモリ・サイクル数の減少にもかかわらず 2.1 倍の処理速度を得た。

5.5 Wavelet 変換エンジンへの応用

RM-II が論理設計の CAD 以外の分野にも適用可能であることを確認するため、画像に対する Wavelet 変換¹⁴⁾に応用した。実現した Haar Wavelet 変換は、隣接 4 画素間で加減算を行い、新たな 4 画素の値を得る操作を反復する。RM-II 上の 8 バンクのメモリのうち原画像格納に 4 バンク、変換後の画像格納に 4 バンクを割り当てた。また、1 ピクセルの画像データが 8 ビットで表せることから、各バンクの 1 ワードで 2 画素分のデータを表現した。その結果、256×256 画素の画像に対して 1.64 msec で処理が完了し、11 MIPS の計算機と比べて約 60 倍の速度を得た。ただ現状では、データの授受に 80 msec を要しており、実用化のためにはインタフェースの点でさらなる工夫が必要である。

6. 拡張性に関する検討

RM-I では FPGA 間を共通バスと完全結合の配線で結んでいるため、FPGA を追加することは困難であった。RM-II では RM-I よりも拡張が容易な結合形態を採用している。そこで、今後の拡張の可能性に関して、(実行) モジュール単体の規模拡張と、複数モジュールの結合による拡張の 2 方式を例にとり、検討を加える。

(1) モジュールの規模拡張

モジュール内部のメモリまたはゲートの容量が不足する場合、図 7 に示すように拡張することが考えられる。ハッチングを施した矩形は FPGA とメモリの組合せを、無地の矩形は配線変更用の FPGA を表す。

図 7 (a) は、現在の RM-II の構成をそのまま拡張した例であり、メモリ容量を重視して拡張する必要がある場合に適している。たとえば、5×5 の FPGA 配列を採用すると、メモリバンクの数を 21 に増やすことができる。その一方で、特に奇数番目のクロスバスにおける通信競合の増加が懸念される。

図 7 (b) に示す構成では、メモリバンク数は 13 となるが、ゲート容量については同図 (a) と同様に 25/9 = 2.78 倍に拡張できるため、ゲート容量を重視した拡張に適している。また、メモリに直結する FPGA と配線変更用の FPGA を交互に配置することで、FPGA 間の配線を柔軟かつ有効に利用できる可能性がある。

(2) 複数モジュールの結合

モジュールの規模をむやみに拡張することは、クロ

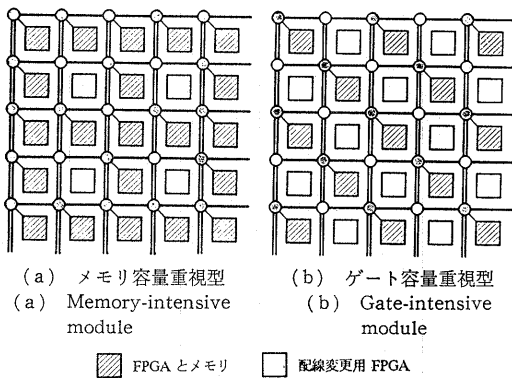


図 7 モジュール拡張形態の例
Fig. 7 Enhanced processing modules.

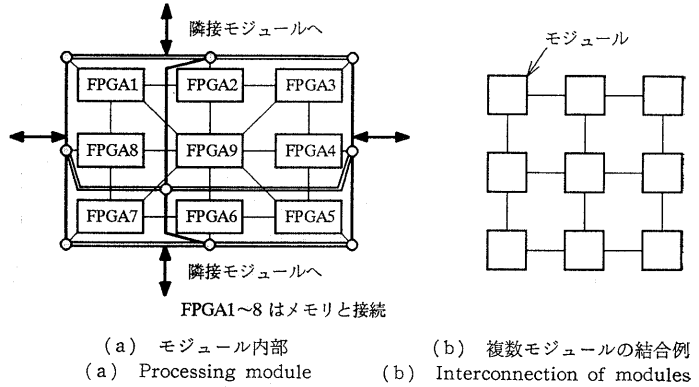


図 8 複数モジュールの結合による拡張例
Fig. 8 An example of parallel processing modules.

バスにおける通信競合の可能性が高くなる点で望ましくない。これまでに実験対象とした複数の応用例については、現状の RM-II のモジュール構成のままで、まとまった処理単位を実現することが可能であった。この場合、モジュールを単位として、従来の一般的な並列計算機における処理ユニットを構築できると考えられる。図 8 に例を示す。モジュールの内部を示す図 8 (a) は、現在の RM-II の構成と同一である。図では省略しているが、FPGA 1~8 はそれぞれメモリバンク MEM 1~8 と接続する。外周部のバスを介して、隣接するモジュールとの通信を行う。また、外部端子数に余裕がある FPGA 2, 4, 6, 8 が、隣接モジュールとの通信を制御する。モジュール内部では、2.3 節で述べた要求を満足するためにクロスバスを必要とするが、モジュール間については一般的な並列計算機における結合形態を採用することも可能と考える。図 8 (b) では、モジュール間にメッシュ結合を適用する例が示されている。

応用によって大きく異なるが、適切な処理粒度の境界値が存在して、それより細かい粒度をもつ処理で必要とするデータの転送には、クロスバスが適していると考えられる。一方で、その境界値を越える粗粒度の処理で必要となるデータの転送についてもクロスバスを利用すると、接続する FPGA 数の増加にともない、通信競合の可能性が高まると予想される。

よって、クロック単位でのデータ転送を目的としたクロスバスをそのまま拡張することは、必ずしも性能向上に寄与せず、処理の粒度に応じて他の一般的な結合網と階層的に組み合わせる形態で拡張することが望ましいと考えられる。今後、処理の粒度と適切な拡張方

式の関係について評価する必要がある。

7. おわりに

FPGA とメモリを組み合わせることによって、複数の処理を高速に実行する汎用エンジン RM-II の構成について述べた。最初のプロトタイプである RM-I で問題となった規模と柔軟性の不足に対応して、複数の同時データ転送のためのクロスバスと配線変更のための FPGA を用意する点を特徴とする。論理シミュレーション、論理設計誤りの診断処理、画像処理を RM-II 上で実現し、性能を評価した。バンク数の増加によって高まる通信要求をクロスバスによって満足した結果、RM-I の約 2 倍の性能を得ることが可能となった。

今後の課題として、ホスト・コンピュータとのデータ転送に要する時間の短縮と、設計支援環境の構築、さらに対象とする処理と適切な拡張方式の関係についての評価が挙げられる。

参考文献

- 1) Blank, T.: A Survey of Hardware Accelerators Used in Computer-Aided Design, *IEEE Design and Test of Computers*, Vol. 1, No. 3, pp. 21-39 (1984).
- 2) Tham, K.Y.: Parallel Processing for CAD Applications, *IEEE Design and Test of Computers*, Vol. 4, No. 5, pp. 13-17 (1987).
- 3) Agrawal, P. et al.: MARS: A Multiprocessor-Based Programmable Accelerator, *IEEE Design and Test of Computers*, Vol. 4, No. 5, pp. 29-36 (1987).
- 4) 菅沼直昭, 村田之広, 富田昌宏, 平野浩太郎: 汎用エンジンの開発と論理診断への応用, DA シンポジウム '92, pp. 89-92 (1992).
- 5) Suganuma, N., Murata, Y., Nakata, S., Nagata, S., Tomita, M. and Hirano, K.: Reconfigurable Machine and Its Application to Logic Diagnosis, *International Conf. on Computer Aided Design*, pp. 373-376 (1992).
- 6) 澄川文徳, 永田真一, 菅沼直昭, 富田昌宏, 平野浩太郎: 汎用エンジン RM-I による論理シミュレーション, 第 45 回情報処理学会全国大会論文集, Vol. 6, pp. 159-160 (1992).
- 7) プログラマブル・ゲートアレイ データブック, ザイリンクス社 (1990).
- 8) Gokhale, M., Holmes, W., Kopser, A., Lucas, S., Minnich, R. and Lopresti, D.: Building and Using a Highly Parallel Programmable Logic Array, *Computer*, Vol. 24, No. 1, pp. 81-89 (1991).

- 9) Van den Bout, D.E., Morris, J.N., Thomae, D., Labrozzi, S., Wingo, S. and Hallman, D.: AnyBoard: An FPGA-Based, Reconfigurable System, *IEEE Design and Test of Computers*, Vol. 9, No. 3, pp. 21-30 (1992).
- 10) The XC 4000 Data Book, Xilinx, Inc. (1991).
- 11) 中越順二, 田中輝雄, 濱中直樹, 面田耕一郎: 並列計算機 H2P の要素プロセッサ間非同期データ転送方式, 第 38 回情報処理学会全国大会論文集, pp. 1488-1489 (1989).
- 12) 菊池原秀行, 今村真人, 青柳洋介, 浜崎良二, 白木 昇: HLS: 論理シミュレーション専用計算機 (6) 性能評価, 第 41 回情報処理学会全国大会論文集, Vol. 6, pp. 137-138 (1990).
- 13) Saitoh, M., Iwata, K., Nakamura, A., Kakegawa, M., Masuda, J., Hamamura, H., Hirose, F. and Kawato, N.: Logic Simulation System Using Simulation Processor (SP), *Proc. 25th Design Automation Conference*, pp. 225-230 (1988).
- 14) Rioul, O. and Vetterli, M.: Wavelet and Signal Processing, *IEEE SP*, Vol. 8, No. 4, pp. 14-38 (1991).

(平成 5 年 9 月 16 日受付)

(平成 6 年 2 月 17 日採録)



富田 昌宏 (正会員)

1960 年生。1983 年 3 月東京大学工学部精密機械工学科卒業。助手を経て 1989 年東京大学講師。1990 年 5 月より神戸大学大学院自然科学研究科講師。現在に至る。工学博士。

主に論理装置の CAD, 情報処理工学に関する研究に従事。IEEE, 電子情報通信学会, プリント回路学会各会員。



澄川 文徳 (学生会員)

1971 年生。1993 年 3 月神戸大学工学部電子工学科卒業。現在, 同大学大学院工学研究科修士課程に在学中。論理装置の CAD, 並列処理アーキテクチャの研究に従事。



菅沼 直昭 (正会員)

1965 年生。1989 年 3 月神戸大学工学部電子工学科卒業。1991 年 3 月同大学大学院工学研究科修士課程修了。1994 年 3 月同大学院自然科学研究科博士課程修了。工学博士。論理装置の CAD, 設計自動化に関する研究に従事。



平野浩太郎

1935年生. 1960年3月大阪大学工学部通信工学科卒業. 1965年同大学大学院博士課程修了. 同年神戸大学工学部電気工学科助教授. 現在は, 同学部電気電子工学科教授, 同大学大学院自然科学研究科教授を兼任. 工学博士. デジタル信号処理, デジタル通信, LSI設計の研究に携わる. IEEE 上級会員, Eta Kappa Nu, 電子情報通信学会, システム制御情報学会各会員.
