

## 動的データ駆動型処理システム Qv-x の視覚的評価支援環境

岩田 誠† 坪田 浩乃‡ 田村 俊之‡ 小守 伸史‡ 寺田 浩詔†

†大阪大学 工学部 情報システム工学科 ‡三菱電機(株) 半導体基礎研究所  
〒 565 吹田市山田丘 2-1 〒 664 伊丹市瑞原 4-1

E-mail: {iwata, terada}@ise.eng.osaka-u.ac.jp {tsubota, tamura, komori}@lsi.melco.co.jp

あらまし

本稿では、データ駆動パラダイムに基づく図的仕様記述体系の研究の一環として、動的データ駆動型プログラムの作成・評価を視覚的に支援する環境の構成法を提案する。本環境は、現在開発中の数種のスーパーパイプライン型データ駆動プロセッサシステムを前提として、a)プログラムの作成・機能検証、b)マルチプロセッサ上での最適な実現法の検討、c)新しいデータ駆動型アーキテクチャの検討、の効果的な支援を目標としている。

本稿では、1) 動的データ駆動型プログラムとその同時並行パイプライン動作の階層的視覚化、2) プログラムの動作に対応したプロセッサの振舞いの視覚化手法、ならびに、3) これらを実現するためのエミュレーション・シミュレーション手法について提案し、これらを統合化した支援環境の構成法を述べる。

和文キーワード 動的データ駆動型プロセッサ、スーパーパイプライン、視覚的性能評価、超並列処理

### A Visual Evaluation Environment for Dynamic Data-Driven Processing System Qv-x

Makoto IWATA† Hirono TSUBOTA‡ Toshiyuki TAMURA‡  
Shinji KOMORI‡ Hiroaki TERADA†

†Department of Information Systems Engineering, Faculty of Engineering, Osaka University  
2-1 Yamadaoka, Suita, Osaka 565, Japan

‡Semiconductor Research Lab., Mitsubishi Electric Co.  
4-1 Mizuhara, Itami city, Hyogo 664, Japan

Abstract

This paper proposes a visual performance evaluation tool for our superpipelined dynamic data-driven processors Qv-x. The proposed tool supports production and debugging of data-driven programs, evaluation of optimal implementations on a multiprocessor system and investigation into advanced data-driven architectures.

In this paper, a visualization scheme for the multidimensional behavior of dynamic data-driven programs and datadriven processor Qv-x are introduced. An implementation of the proposed visual environment for the Qv-x simulator and some practical Qv-x processor chips follows.

英文 key words Dynamic data-driven processor, super-pipeline, visual performance evaluation, massively parallel processing

## 1 はじめに

データ駆動原理は、細粒度から粗粒度にいたる並列処理方式の基礎としてだけではなく、システムの自然な表現能力を有している。特に、動的データ駆動原理は、トークンの追い越しをも許す柔軟なパイプライン型並列処理を可能にする、非常に自然な自律分散的な多重処理原理である [1]。このため、半永久的な入力系列(群)に対する処理が要求される、トランザクション処理や通信処理などの効果的な実現の手段となっている。

筆者らは既に、このような動的データ駆動原理の並列処理能力とシステム表現能力に着目して、

- スループットを重視した数種のスーパーパイプライン型動的データ駆動型プロセッサ Qv-x(Qv-1, Qv-2m, Qv-3s) [2] を開発し、同時に、
- これらのソフトウェア環境として、システムの発注者にも容易に理解可能な図的な仕様記述からデータ駆動型プログラムを直接生成する体系の研究 [3, 4] を進め、

ソフトウェアからハードウェアに至るまで一貫してデータ駆動パラダイムに基づいたシステム構成法の確立を目指している。

本稿では、このデータ駆動パラダイムに基づくシステム構成法の研究の一環として、動的データ駆動型プログラムならびにアーキテクチャの開発・評価を視覚的に支援する環境の構成法を述べる。

これまでにも並列処理プログラムを対象にしたシステムソフトウェアが各所で研究されているが [5]、その多くは、逐次代入型言語の並列化という立場をとっている。

これに対して、本評価環境は、仕様記述から生成されたデータ駆動型プログラムを、我々が共同開発した数種の Qv-x マルチプロセッサシステム上で高度並列に実行する体系を前提として、

- a) Qv-x プログラムの作成・機能検証、
- b) マルチプロセッサ上での最適実現法の検討、
- c) 新しいデータ駆動型アーキテクチャの検討、

の効果的な支援を目標としている。本稿では、特に、

- 1) 動的データ駆動型プログラムとその同時並行・パイプライン並列動作の階層的視覚化、
- 2) プログラムの動作に対応したマルチプロセッサの振舞いの視覚化手法、ならびに、

- 3) これらを効果的に実現するためのエミュレーション・シミュレーション手法、

を提案し、これらを統合化した支援環境の一構成法を述べる。

以下、図的仕様記述体系における Qv-x 評価環境の位置付けならびに要件を、Qv-x のハードウェア構成の特徴を加味して検討した後、Qv-x の視覚的評価支援手法ならびにその一実現法について述べる。

## 2 Qv-x の支援環境の要件

筆者らが検討中の仕様記述処理体系 AESOP (Advanced Environment for Software Production) [3,4] の大きな特徴は、実行機械の物理的制約に依存しない抽象的な動的データ駆動型プログラム (ADP: Abstract Data-Driven Program) を多面的かつ図的な仕様記述から直接生成できることにある。この ADP によって、ソフトウェアが本来表現すべき、論理的なデータ構造とその処理アルゴリズムが純粹に定義される。このため、ADP 水準で、仕様記述の検証や部品化を可能にし、かつ、異なる実行機械に対する可搬性が維持される。

しかし、この ADP を数種のデータ駆動型プロセッサ Qv-x 上で高度並列に実行するには、ADP から特定の Qv-x システムの物理的構成に最適化した実行形式プログラムに変換する手法の確立が必要になる。すなわち、各 Qv-x の機能的制約(命令セット、演算精度)や、物理的資源制約(プロセッサ数、各種記憶容量など)を加味した最適化が要求される。

また、仕様記述の記述水準、すなわち、ADP の記述水準を向上させるには、特定の機械に最適化した高機能かつ高性能なプログラム部品を蓄積し、ADP ではこれらをブラックボックス化して扱える必要がある。

このような最適化手法の検討ならびに部品の蓄積を効率良く行なうために、本研究では、現存あるいは将来に開発すべき数種の Qv-x を前提として、そのプログラムならびにアーキテクチャの開発、機能検証、性能評価を、視覚化手法を利用して効果的に支援する手法を検討している。

以下、本章では、Qv-x プロセッサの特徴であるスーパーパイプラインの基本構成と、その性能評価のポイントを述べ、この視覚的評価支援手法の要件を整理する。

## 2.1 Qv-x のスーパーパイプライン構成

データ駆動型処理方式では、(1). 入力データ組が揃った処理機能のみが実行に移され、(2). 一旦、起動された処理機能は、お互いに独立に実行可能であり、(3). 処理の遅れ (latency) に対する耐性がある、という特徴から、徹底したパイプライン処理が可能である。さらに、動的データ駆動方式では、トークンの追い越しをも許す、柔軟なストリーム型データ処理能力を備えているので、パイプラインの充足が原理的に容易である。

これらのことから、筆者らが共同開発した Qv-x のハードウェア実現は、図 1 に示すように、

- (i). データ駆動型処理実行に必要な機能要素 (待ち合わせ記憶部 (MM), 関数的処理部 (FP), プログラム記憶部 (PM)) と FIFO 型緩衝記憶 (QB) を、すべて多段のパイプライン型処理機構として実現し、さらに、
- (ii). これらの機能要素をふたたびパイプライン型に環状接続した、

スーパーパイプライン型のシステム構成として設計されている。これらのパイプライン型処理機構はすべて自己同期型データ転送・処理機構により実現され、大域的なシステムクロックやバスを排除した、完全な自律分散制御がハードウェア水準で実現されている。このため、従来のノイマン型アーキテクチャの大規模高集積化に伴うクロックスキューの問題や配線長増大の問題が原理的に解決されている。

このようなスーパーパイプライン構成を効果的に機能させる最大のポイントは、パイプライン内のパケット流にボトルネックを生じさせないことである。

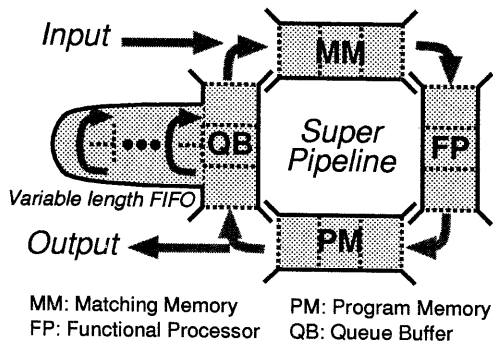


図 1: Qv-x のスーパーパイプライン構成の概念図

## 2.2 Qv-x プログラムとアーキテクチャの評価方針

### (a). Qv-x プログラムの機能検証

データ駆動型プログラムでは、データ依存関係によってのみ処理実行が進行するため、ブレイクポイントで、サスペンド・レジュームしたとしても、プログラムの実行結果には影響を与えないという特徴がある。すなわち、従来の逐次プロセスの並列プログラムのデバッグ手法の重大な問題である、プローブ効果 [6] が発生することは少ない。唯一非決定的な動作をするのは、データ依存関係が合流 (merge) する箇所のみである。これに関しては、静的に予め容易に検出可能であるため、この箇所にブレイクポイントを設定して観察すれば、プログラムの正しさが容易に確認できる。

### (b). Qv-x プログラムの性能評価

AESOP における ADP の実行性能の評価に際しては、実行機械に依存しない処理アルゴリズムの実行に本質的に必要な演算のみを対象に比較検討を行なうべきである。Qv-x プログラムには、データ駆動に固有の COPY 命令や SYNC 命令、動的データ駆動に固有のタグ操作命令が含まれている。このため、実効的な命令のみを対象にした性能 (単位時間あたりの処理率) の平均値や時間の変遷を抜粋できる必要がある。

### (b). プロセッサの性能評価

動的データ駆動方式を採用した Qv-x は、メモリアクセスやプロセッサ間通信の遅延に対する耐性を原理的に有している。しかし、これは、プログラム構造に依存する特性でもあり、プログラム最適化の鍵の一つにもなる。このため、種々の支配要因 (メモリアクセス遅延時間、機能分散形態など) に対する遅延耐性を比較検討できる必要がある。

また、Qv-x では、プログラムの並列性やプロセッサ間通信量がパイプラインの容量を越えても、FIFO 型緩衝記憶 (QB) により一定量のパケットを緩衝でき、QB がオーバーフローしない限り、パイプライン流量を最大に維持できる。しかし、QB 内のパケット量に応じて、パケットが QB 通過時間が変化するため、これがプログラム全体の性能を律速する。例えば、遅延ループのある信号処理などの場合には、プ

プログラム全体のスループット向上の鍵となる。つまり、Qv-x 上での効果的なプログラムの実行には、プロセッサ内のパイプライン流量をできる限り最大に維持し、かつ、QB 内の平均パケット量を極小化できる、プログラムに最適化する手法を実験的に検討できる必要がある。

### (c). マルチプロセッサの性能評価

Qv-x アーキテクチャでは、自己同期型の緩衝記憶 (QB) 付きルータチップにより、多数の Qv-x プロセッサを相互接続すれば、容易にマルチプロセッサ・システムに拡張可能である。例えば、50MFLOPS の最大性能を有する Qv-2m (Q version 2 developed by Mitsubishi: 通称 RAPID) は、最大 256 プロセッサを接続できるよう設計されている [7]。このようなマルチプロセッサの性能評価のためには、各プロセッサの稼働状況やプロセッサ間通信の状況を的確に観測でき、その結果に基づきパケット流量のボトルネックを解消するための種々の最適化手法を実験的に検討できる必要がある。

## 3 Qv-x の視覚的評価支援手法

本評価支援手法は、上述の各種の機能・性能評価の効果的支援のために、次の視覚化方針を採用した。

- Qv-x プログラム上のトークンや Qv-x プロセッサ上のパケットの多次元的振舞を、相互に関連づけた視覚化。
- マジカル数  $7 \pm 2$  を考慮した、部分的振る舞いや統計情報の視覚化。

以下では、これらの視覚化方針にしたがって、プログラムならびにアーキテクチャの開発、機能検証、性能評価の支援手法を述べる。

### 3.1 階層的図的プログラムの作成・機能デバッグ支援

動的データ駆動型プログラムは、ノード (節点) とアーク (有向枝) から成る階層的な有向グラフにより図 2 のように表現できる。また、その動作は有向グラフ上のデータ依存関係に従った多数のトークンの振舞として現れる。すなわち、ある瞬間には、(1). 同時並行性により同一のタグを持つトークンが複数のアーク上に存在し、(2). パイプライン並列性によ

り異なるタグを持つトークンが同一アーク上に存在し、次の瞬間には、それらの一部が消費され、別のアーク上に新たなトークンが生成される、といったように、多次元的にトークンが振る舞う。

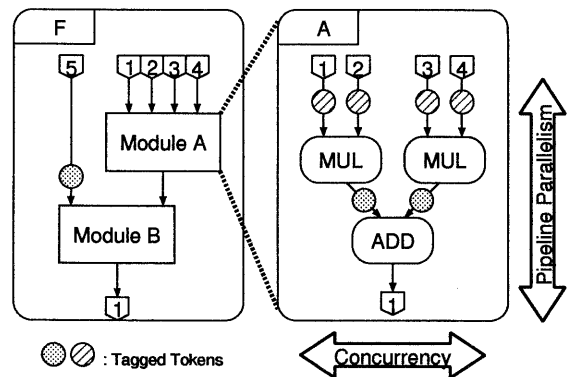


図 2: データ駆動型プログラムとその動作

したがって、プログラムの機能デバッグの際には、このように多次元的に振る舞う多数のトークンを追跡し、バグの原因を特定し解消しなければならない。これらの作業を効果的に支援するには、多数のトークンを適切な側面から抜粋して提示することが必要である。

以上のことから、本支援手法では、プログラム上でのトークンの振る舞いが直観的に把握できるよう、図的データ駆動型プログラムとその上でのトークンフローのアニメーションを行なう機能を提供する。さらに、多次元的なトークンの振る舞いを種々の側面で観測するために、以下のような切口で、部分的な視覚化機能を提供する。

- 機能階層 機能的にまとまった単位の階層的プログラムの編集と表示、機能階層のツリー表示
- タグ タグ付きトークンのタグ値によるフィルタリング機能
- 処理構造 特定の処理構造 (選択構造、遅延ループ、共有関数など) の明示的表示

さらに、これらの機能階層、タグ値、および、処理構造毎に部分的にアニメーションを行なうウィンドウ上で、以下の支援機能を提供する。

- トークンの行き先ポート、データ値、および、それらに対するマスクの設定を許すブレイク機能
- 非決定的実行の可能性がある箇所を明示機能

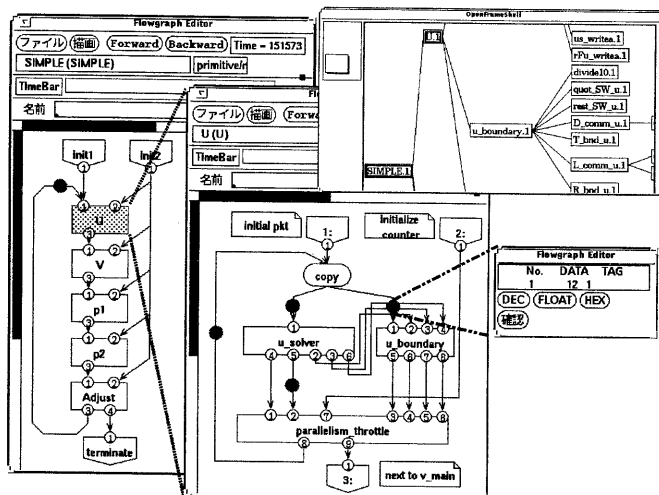


図 3: Qv-x の視覚的プログラミング支援環境の使用例

- (c). 実行履歴に基づくバックワード・アニメーション機能
- (d). 履歴依存データ記憶部 (DM) のデータ値の遷移状況の視覚的表示機能

図 3 は、本視覚的プログラム支援環境を利用して、流体シミュレーション (CFD) プログラムの作成・デバッグを行なっている様子を示している。右上の図がプログラムの機能階層のツリーの表示であり、この中の節点を選択すれば、対応する階層の実行アニメーションが行なわれる。また、トークンを選択すれば、そのパケットの内容が右図のようにポップアップされる。

### 3.2 単一プロセッサの性能評価支援

Qv-x プロセッサ内での並列性制御や待ち合わせ記憶の有効利用法などの最適な実現法の検討のためには、プロセッサ内の物理的構成がプログラムの実行性能に及ぼす影響を解析する必要がある。本評価環境が前提としているデータ駆動型プロセッサ Qv-x は基本的には、前述したように、パケットの待ち合わせ記憶部 (MM)、関数的処理部 (FP)、プログラム記憶部 (PM) からなる環状パイプライン構造からなり、さらに、キューバッファ (QB) が設けられ、パケット流量の揺らぎを緩衝できる。

このパケット流量は、(1). プロセッサ外部からの入力パケットの流量によるパイプライン並列性の変

動、(2). プログラムの持つ同時並行型並列性の変動、(3). 待ち合わせ記憶部 (MM) におけるハッシュ衝突の回数、に伴って、増減する。

したがって、Qv-x プログラムを最適化する際には、これらが適切に制御されているかどうかを観測する必要がある。このため本支援手法では、(1). MM の占有状況や競合中のハッシュ・アドレスに対応するトークンの探索、(2). 入力データの投入間隔やプログラム構造を修正した時の性能 (入出力パケット流量、応答時間、QB 内のパケット量、FP の実効的な稼働率、DM アクセス率) の比較、が可能でなければならない。

また、プロセッサ上でのパケット内の行き先ノードのみでは、プログラムの論理的構造との対応を直観的に把握することが困難である。このため、プロセッサ上でのパケットの振る舞いを階層的データ駆動型プログラム上で振る舞いに容易に対応づけられる機能を提供している。

### 3.3 マルチプロセッサの性能評価支援

Qv-x マルチプロセッサは各プロセッサ毎にメモリを有する、分散メモリ型である。本アーキテクチャ上で最適化を施すには、特定のプロセッサ間通信路ならびに特定のプロセッサがボトルネックにならないように、

- プロセッサトポロジとルーティング手法

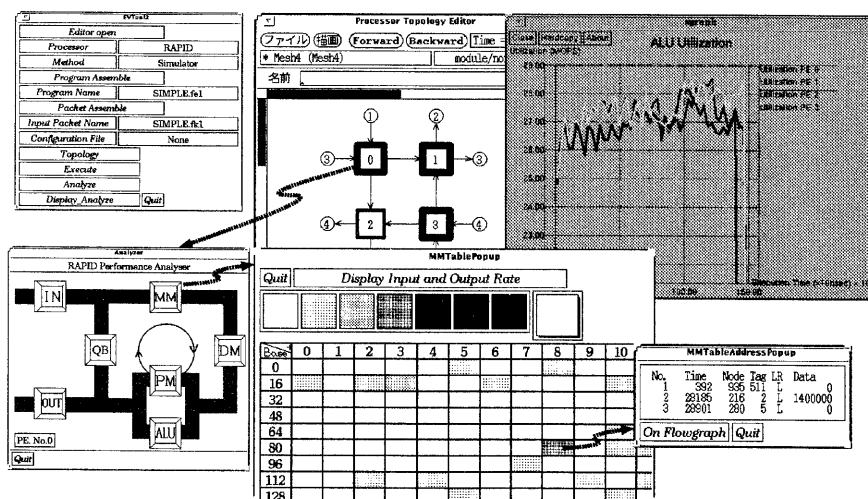


図 4: Qv-x マルチプロセッサの視覚的評価支援環境の使用例

- プロセッサトポロジ上へのプログラム・データの負荷・機能分散

を適切に選択しなければならない。

したがって、本支援手法では、プロセッサ・トポロジとその上へのプログラム・データの分割・配置を任意に指定できるインターフェースを提供する。さらに、その上での実行状況を的確に把握できるようにするため、

- 各プロセッサの ALU 稼働率、QB 長、ハッシュ衝突、DM アクセス率の時間的推移
- 各プロセッサ間通信路の packets 流量 (直接通信、Global DM アクセスの比率、通過 packets)

を視覚的にアニメーション表示する機能を提供する。

図 4 は、図 3 に示した CFD プログラムを、トラス状に相互接続された 4 台の RAPID 上で実行した時の、性能評価の様子を示している。図中左上は、本環境の主操作パネルであり、中央上に、4PE のトポロジを表示し、この上で、各プロセッサの ALU 稼働率とプロセッサ間通信量をアニメーション表示している。また、右上には、各プロセッサの ALU 稼働率の時間的推移をグラフ表示している。また、特定のプロセッサの動作を詳細に観測したい場合には、左下のようなプロセッサ構成上で種々の性能評価が可能である。例えば、右下は、MM のハッシュメモリの占有状況を視覚的に表示し、特定のハッシュアドレスの内容を右下のようにポップアップして見ることが可能である。

### 3.4 新規アーキテクチャの性能評価支援

現状の Qv-x プロセッサは、汎用プロセッサとして、あるいは、専用プロセッサとして実用化するには、目的に応じて最適なアーキテクチャに発展できる余地が充分にある。このため本支援手法では、これらの最適なアーキテクチャの探索を実験的に検討可能にすることも目的としている。例えば、

- 新しい命令セット (演算命令、タグ処理命令を含む)
- 巡回パイプラインの要素機能の構成
- packets 流量の制御機構 (活性 packets の退避、復帰機構)
- PM / DM のスワッピング機構

などの性能評価を支援する予定である。これらの支援機能の実現法としては、データ駆動型プログラムがハードウェアアルゴリズムを自然に表現できる特徴を活用して、現行のプロセッサ上で上記ハードウェア機構をシミュレートする手法が当面有望であると考えている。

## 4 Qv-x の視覚的評価環境の一構成法

### 4.1 エミュレーション・シミュレーション手法

前章では、Qv-x のプログラムならびにアーキテクチャの性能評価を視覚化手法を用いて支援する手法を述べたが、これらの機能を実用的に利用可能な

ように実現するには、以下の課題を解決する必要がある。

- 実行履歴のデータ量が膨大である。
- シミュレータによる実行時間が長い。
- Qv-x プロセッサからモニタできる情報量が少ない。

したがって、本環境の実現に際しては、高速実行が可能な少数のデータ駆動型プロセッサとチップ内の詳細なデータ収集が可能なシミュレータを有機的に活用して、大規模なマルチプロセッサをエミュレートする手法を採用した。図 5 は、実現した本視覚的評価環境の機能構成を示している。

本手法では、評価対象のマルチプロセッサ上に分割・配置された、データ駆動型プログラムのマクロなブロックを、少数の VLSI プロセッサ上で入れ換えながら実行することにより、実用的規模のプログラムを任意の接続構造を持つマルチプロセッサ上で実行した場合の実験的評価が可能になる。また、実行時の状況を詳細に解析したいブロックのみをシミュレータで実行し、それ以外のブロックを VLSI プロセッサ上で実行することにより、柔軟なデータ収集も実現される。

このような実現法の採用によって、機能的デバッグの支援が効率的に実現される。また、ブロックをノードとするマクロなデータ依存グラフが非巡回グラフである場合、例えば、信号処理プログラムのような場合には、性能評価の支援も可能である。

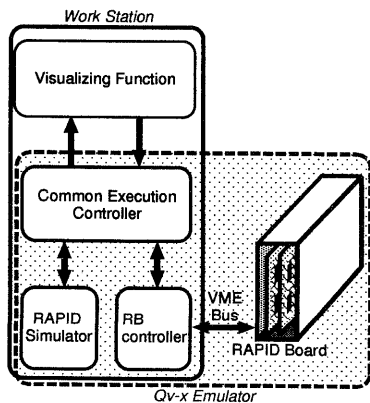


図 5: Qv-x の視覚的評価環境の機能構成

## 4.2 データベース化機能

本環境は、前述のような多面的な評価支援のために、複数の視覚化機能が互いに情報を授受し、プログラム情報や実行状況情報を加工しながら、連係して並列に動作する系である。評価対象プログラムが実用的規模になれば、大量のトレース情報のデータ処理能力が要求される。このため、本環境の実現にあたっては、データ駆動型プロセッサシステム上への将来の移植を考慮して、データ駆動パラダイムに基づく仕様化を行ない、これをワークステーション上で模擬する形態で実現した。

また、複数の Qv-x プロセッサ、および、シミュレータ・エミュレータに対する、可搬性を付与するために、評価環境中のデータ構造・機能を可能な限り共通化するように設計した。すなわち、(1). 実行可能プログラム形式、(2). トレースデータ形式、(3). 各種の解析データ形式 (トークン流データ、QB 長データなど)、を共通化して、データベース化を行なった。

## 5 おわりに

本稿では、スーパーパイプライン型動的データ駆動型プロセッサ Qv-x 上でのプログラムならびにアーキテクチャの性能評価・開発を視覚的に支援する環境の一構成法について述べた。

本稿に述べた視覚的評価支援手法は、シミュレータと少数の Qv-x プロセッサを有機的に結合して実現されているため、機能的評価は可能であるが、性能に関しては制限付きでしか評価できない。これに関しては、リアルタイムモニタ機能などのハードウェアサポート [8] も含めて、より効率的に性能評価を支援可能な手法を並行して検討中である。

現段階ではまだ、マルチプロセッサの視覚的支援機能の一部と新規アーキテクチャの開発支援機能が未完成であり、今後、上記のハードウェアサポートも含めて実用的な支援ツールとして実現する予定である。

また一方、Qv-x に採用している自己タイミング型パイプライン機構は、環状あるいは格子状に相互接続すれば、汎用的な集積化機能ブロックとして構成できる可能性がある [9]。データ駆動型プログラムがこのようなハードウェア・アルゴリズムを容易に模擬できることを利用すれば、本視覚的支援環境を自律分散型ハードウェア機構の視覚的シミュレータと

して将来的に発展できると考えている。

謝辞 御指導、御支援頂いた Qv-x 研究の関係各位ならびに御協力頂いた寺田研究室の各位に深く感謝する。特に、Qv-1 ならびに Qv-3s のシミュレータ・エミュレータを提供していただいたシャープ (株) の関係各位に感謝の意を表します。

なお、本研究の一部は、文部省科学研究費 (一般 B 05452363, 試験 B-1 06555110) の援助を受けて行なったものである。

## 参考文献

- [1]. 寺田: VLSI 向きデータ駆動型プロセッサ, 信学誌, Vol.72, No.7, pp.742-749 (1989).
- [2]. H.Terada, M.Iwata et al.: *Superpipelined Dynamic Data-Driven VLSI Processors*, Proc. ACM ISCA Workshop on Data-Flow Computing, (1992) (to be published).
- [3]. M.Iwata and H.Terada: *Multilateral Diagrammatic Specification Environment based on Data-Driven Paradigm*, Proc. ACM ISCA Workshop on Data-Flow Computing, (1992) (to be published).
- [4]. 岩田, 寺田: 図的仕様記述からのデータ駆動プログラムの生成手法, 情報処理学会計算機アーキテクチャ研究会, 94-ARC-107-8, pp.57-64 (1994).
- [5]. 笠原: 並列処理のためのシステムソフトウェア, 情報処理, Vol.34, No.9, pp.1134-1138 (1993).
- [6]. C.E.McDowell and D.P.Helmbold: *Debugging Concurrent Programs*, ACM Computing Surveys, Vol.21, No.4, pp.598-622 (1989).
- [7]. S. Komori et al.: *A 50 MFLOPS Superpipelined Data-Driven Microprocessor*, Proc. ISSCC '91, pp.92-93 (1991).
- [8]. 田村 他: データ駆動型プロセッサ RAPID のソフトウェア開発環境, 信学会, 94-CPSY (1994).
- [9]. 寺田, 西川 他: VLSI 向きデータ駆動型プロセッサ: Q-x, 信学論 (D), Vol.J71-D, No.8, pp.1383-1390 (1988).