

実時間並列処理計算機 CODA – 相互結合網 –

戸田賢二 西田健次 島田俊夫† 山口喜教

電子技術総合研究所 †名古屋大学

〒305 つくば市梅園 1-1-4

Email: toda@etl.go.jp Tel: 0298-58-5875 Fax: 0298-58-5882

あらまし 我々の開発している実時間処理用並列計算機 CODA の相互結合網の特徴及びその実現方式について紹介する。CODA の相互結合網は、優先度逆転を抑える優先度先送り方式の優先度制御機構を持ったバケット切り替え方式の 4×4 ルータチップ 48 個 (16 個 3 段) から構成されるデルタ網である。同ルータチップは 391 ピン, 10 万ゲート, 0.7 ミクロンのゲートアレイを使用し, 32 ビットの優先度制御を行ないつつポート当たり 1.9G bits / second のデータ転送能力を持つ。

和文キーワード: 優先度先送り方式, 実時間処理用相互結合網, 多段網, ルータチップ, 優先度キュー, 実時間並列アーキテクチャ

The Real-Time Parallel Architecture CODA – Interconnection Network –

Kenji TODA Kenji NISHIDA Toshio SHIMADA†
Yoshinori YAMAGUCHI

Electrotechnical Laboratory †Nagoya University
1-1-4 Umezono, Tsukuba 305, JAPAN

Abstract The features and implementation of an interconnection network for the real-time parallel architecture CODA are described. The interconnection network is a priority-inversion-free prioritized delta network composed of 48 (16 chips * 3 stages) packet switching 4-by-4 router chips. The chip is a 391 pins 100K gates 0.7 micron gate array. Its predicted data transmission rate is 1.9G bits/second per port with the priority forwarding control of a 32 bit priority.

英文 key words: Priority Forwarding Scheme, Real-Time Interconnection Network, Multistage Network, Priority Queue, Real-Time Parallel Architecture

1 はじめに

我々はセンサフュージョン応用を目指した実時間並列計算機 CODA の開発を進めている。センサフュージョン応用では、高いスループットと共にタスク処理に置いて正確な時間管理が要求されるため、CODA はシステム全体を通して優先度逆転が発生しないように設計されており、アドラインなどの時間を直接優先度として用いることを可能とする 32 ビットの広い優先度空間をサポートしている [CODA, PE93]。本稿では、CODA のプロセッサ間の相互結合網について、その動作方式及び性能について報告する。

2 多段網上での優先度逆転と優先度先送り方式

2.1 優先度逆転

実時間処理のための相互結合網への要求は、ノードサイズに対する規模拡張性があること、通信遅延に対する予測性があること、高い優先度のパケットに対しては低いものより短い通信遅延を与えること、優先度制御を行なうことによるスループットの低下が少ないこと、等がある。多段網は大規模な構成にも対応できる規模拡張性があるが、衝突パケット同士の優先度情報だけに基づく調停を行なうと、中間優先度のパケットによってブロックされている低い優先度のパケットの後ろから来た高い優先度のパケットがいつまでも待たされてしまうという優先度逆転現象が発生する。この幻想は、網上で複数かつネストして発生しう

2.2 優先度先送り方式

この優先度逆転を解消するために我々は、パケット転送がブロックされたら自分の優先度情報を次段のルータに送り、高い優先度のパケットが後ろで待たされている状況を知らせるといった優先度先送り方式を提案した [PF1]。優先度先送り方式には、先送りの方向、先送りする優先度情報の種類及び利用方法により複数のバリエーションが可能であるが、性能/実装コストの点で優れている基本優先度先送り方式（優先度先送りの方向はその段のパケットの出力方向、転送するのは優先度のみ、先送りされた優先度はブロッキング中に限り有効）について、性能評価と実現コストの検討を行なった [PF1, 2, 3, 4]。

BPF のルータ上での動作は次のものである。ただし、パケットバッファは入力ポート側にあると仮定し、入力ポート毎に以下に示すポート優先度を用意する。

- バッファが一杯でない場合、ポート優先度はバッファ中のパケットの最大優先度となる。
- 自分のパケットの出力がブロックされている場合、そのブロック先（次段のルータ）に自分のポート優先度を出力する。
- （バッファが一杯で）前段からのパケットをブロックしているルータは、送られてきた優先度と自分の

ポート優先度を比較し高い方を新たにポート優先度とする。

- パケットの衝突の際は、ポート優先度の高い方が優先される（等しい場合はラウンドロビン動作とする）

評価の結果 BPF では、最高優先度のパケットに対する最悪転送遅延が $O((\log_n N)^2)$ であり（ラウンドロビンは $O(N)$ ）、スループットの低下を招かず優先度に応じた転送遅延を与えることが明かになった。また、パケットの優先度として到着すべきアドラインを与え、時間的余裕のないパケットを優先する最小ラキシティ優先制御を行なったシミュレーションでは、パケットのラキシティがある程度分散している状況において、ラウンドロビンと比較して著しい性能向上を得た [PF4]。実現においても、パケットバッファの替わりに用いる優先度キューのコストが大半であり、ルータ間には特別な信号線は必要ない。

以上の評価結果より、CODA の相互結合網として優先度先送り方式を採用したルータの多段網を用いる。

3 CODA の全体構成と相互結合網

CODA は、プロセッサ、メモリ、センサ/アクチュエータインタフェースからなる基本プロセッサ 64 台をパケット切り替え型の相互結合網で接続した構成である（図 1 参照）。相互結合網はデルタ網であり、 4×4 ルータチップ 48 個（16 個の 3 段）から構成されている。本ルータチップは、優先度先送り機構を内蔵しており、網上での優先度逆転の発生を抑えることができる。また、基本プロセッサ側の網へのインタフェースにも優先度先送り機構を組み込み基本プロセッサと網の間での優先度逆転の発生を防いでいる。なお、優先度先送り方式を適用しても、同一経路上の同一優先度のパケットについては追い越しが発生せず、到着順番が保証される。

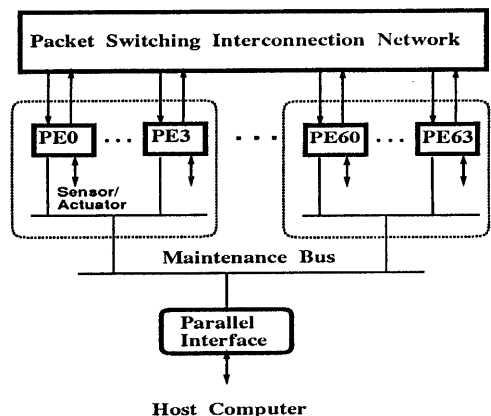


図 1: CODA の全体構成

4 ルータチップの特徴

ルータチップは、391ピン、10万ゲート、0.7ミクロンのゲートアレイ（実効線幅1.0ミクロン）であり、次の特徴を持つ。

- バケット切り替え型
バケット毎に、内蔵している目的アドレスへのルーティングを行なう。
- 4×4（4入力ポート、4出力ポート）
- データ幅38ビット（+1パリティビット）／ポート
- クロック同期動作（20ns 4段パイプライン）
ルータのデータ転送は網全体に供給されるクロックに同期して行なわれる。ポート当たりのデータ転送能力は、1.9G bits / second.
- 8バケット優先度キュー／入力ポート
バケットは38ビット幅4セグメント固定長。
- 仮想カットスルー機能
入力中のバケットは、出力先が空いている限り、ルータ内に一旦蓄えられることなく、出力される。出力先が塞がっていれば、内部のバッファに全体が蓄えられるため一つのバケットが複数の段を塞ぎ続けることはない。
- 優先度制御機能
BPF制御及び入力バッファとして優先度キューを使用（同一優先度バケットに対してはFIFO動作）。モード切り替えにより、非優先度制御も可能。この場合、ラウンドロビン制御で入力キューがFIFOとなる。
- 保守機能
チップのリセット、モード切り替え、内部レジスタ等の参照／設定、が行なえる。

バケットの一般形式を表1に示す。ルータの制御情報は第1セグメントだけで、残りの3セグメントはデータとして扱う。

表1: ルータの要求するバケット形式

Segment No.	38 bits × 4 segments
1	6 bit PE address, 32 bit priority
2	data
3	data
4	data

4.1 バケット転送について

4.1.1 バケット転送の規約

- 複数の連続するバケットの転送においては、バケット間にすき間（バケット転送を伴わない）のクロックが発生することはない。
- バケットは第一セグメントの転送が開始されれば第4セグメントの転送までが保証されており、途中でバケット転送が中断することはない。
- ルータ側は、バケット入力及び出力のタイミングに対し奇数クロックや偶数クロック等の概念を持たない。（すなわち、ルータ網に偶数クロックで入ってきたバケットの出力が始まるのが偶数クロックであるという保証はない。）1/2の周波数のプロセッサのクロックとの同期は、busy信号の制御によりプロセッサのバケットポートでとることが可能。

4.1.2 各ポートのハンドシェイク線のタイミング

各ポートは、busyとDSの2本のハンドシェイクのための信号を持つ。これらの信号のタイミングを入力ポートを例に説明する。

バケットポートバッファが一杯でない（少なくともバケットの第一セグメントが出力されたらそのバケットバッファのエレメントは空とみなす。逆に、少なくとも第一セグメントが入力されたらそのバッファのエレメントは使用中とみなす。）なら、busyをOFF、一杯ならONを前段に出力する。第4セグメントの入力の次のクロックからbusyがOFFの時、前段からのDSがONであれば、そのデータを第一セグメントとして取り込む。1クロック前のbusy信号で、現在のバケット転送の可否が決まる。第2セグメントから第4セグメントの転送については、busyもDSも無視する。

5 信号線と電源・グラウンドの検討

5.1 信号線数

入力ポート又は出力ポート毎に、2本のハンドシェイクピンが必要である。また、チップ毎にクロック1本、保守用に3本のピンを用いる。従って、チップ全体では、

- データI/O：38(data) + 1(parity) + 1(DS) + 1(busy) = 41/port, 41*8 = 328/chip
- チップ全体の制御：1(clock) + 1(mclock) + 1(min) + 1(mout) = 4/chip

合計 328 + 4 = 332

となる。391ピンCPGA（Ceramic Pin Grid Array）パッケージの許容信号ライン数は、332でありこの制限一杯である。

5.2 各信号線の仕様

- clock (入力, data sampling at rising edge)
動作クロック. 本クロック信号に同期して, セグメント単位でデータ転送を行なう.
- mclock (入力, data sampling at rising edge)
保守回路の動作クロック.
- min (入力, 正論理, pull up)
保守回路への入力. 各種コマンド及び内部状態の入力を行なう.
- mout (出力, 正論理)
保守回路からの出力. 内部状態の出力に用いる. パリティチェック状態でデータのバリティエラーを検出すると 1 になる (通常は mclock を停止して使用). リセット信号でクリアされる. 入力側で pull up していれば断線するとエラーとなる.
- データライン (入力側と出力側あり, ポート毎, 正論理, 入力は pull up)
- DS (データラインの信号と同方向, ポート毎)
Data Strobe, データラインの信号が有効であることを示す.
- busy (データラインの信号と反対方向, ポート毎)
バッファの状態を示す. バッファフルで busy が立つ.

尚, PE アドレス部 (6 ビット) は, 先頭 2 ビットをルータ内アドレス情報として用い, ルータチップから出力する際, 2 ビット単位でラウンドシフトする. 従って, パリティには影響しない.

5.3 ピンアサイン

ルータは 4×4 であり, 以下の信号線がある.

- 入力ポート: A, B, C, D
各入力ポートは, 40 ビット入力線 (38 ビットデータ, 1 ビットパリティ, 1 ビットデータストロブ) と 1 ビット出力線 (busy) を持つ.
- 出力ポート: W, X, Y, Z
各出力ポートは, 40 ビット出力線 (38 ビットデータ, 1 ビットパリティ, 1 ビットデータストロブ) と 1 ビット入力線 (busy) を持つ.
- クロック (clock), 保守用クロック (mclock), 保守用入力 (min)
- 保守用出力 (mout)

ピンアサインでは, 次の点に留意した.

- 出力信号と入力信号のブロック化
出力信号に囲まれた入力信号は安定性が悪い.

- クロックを各ポートから等距離に置く
スキュー減少のため.
- クロックを含めた制御信号は電源・グラウンドの近くに置く
信号の安定性確保のため.

この結果, チップの左半分は入力, 右半分は出力とした. 表 2 は左回りに各信号の配置を記述したものである. (チップの上下左右の各側面の中央には, 電源・グラウンドが作り付けで配置されている.)

5.4 電源・グラウンドについての検討

ルータでは, 入力信号, 出力信号とも同時に変化する場合が一般的であり, チップに対する十分な数の適正な配置の電源・グラウンドを設ける必要がある. 本章では, この点をデータブック [DB,GAPKG] のルール及びシミュレーションによって検討する.

ルータチップの出力信号ピンの合計は $328/2 + 1(\text{mout}) = 165$ でありこれらは同時変化する可能性がある. 入力信号ピンは $328/2 + 3(\text{clock, mclock, min}) = 167$ である.

以下, 必要な電源・グラウンドの数を見積もる. 但し, ドライブパッドは B4 (4mA) を仮定する.

- VDD/VSS 同時変化する出力 (SSO) のためのパッドペア
必要な SSO パッドペア = $165/10 = 16.5 \mapsto 17$,
従って合計 34 パッド. (ドライバとして B4R を使用した場合は, $16.5 * 0.4 = 6.6 \mapsto 7$, 合計 14 パッド.)
- 内部ロジックのためのグラウンドバス: VSS2
各側面の VSS2 パッド = $90K(\text{使用ゲート数}) * 0.5(\text{同時スイッチゲートの割合}) * 50(\text{MHz}) * 1.67 * 10^{**} * 5 / 4 = 9.4 \mapsto 10$, 合計 40 パッド.
- 入力レシーバ電源・グラウンドバス: VDD3/VSS3
入力が同時変化 (SSI) する場合: 必要な SSI パッドペア = $167/20 = 8.35 \mapsto 9$, 合計 18 パッド.

電源・グラウンドに必要なパッド数は総計, $34 + 40 + 18 = 92$, となる. ゲートキャパシティ 100K のデバイスのパッド数は 437 であることから, $437 - 332 = 105$ の電源・グラウンドパッドが許されるので, 92 は許容範囲である. CPGA 391 ピンのパッケージの仕様では, 信号ピンは最大 332 本, VDD に 29 本, VSS に 30 本であるが, VDD, VSS とも複数のパッドの接続が可能であるので, 上記必要な電源・グラウンドパッドが確保できる.

上記ピンアサインプランでパッド配線も問題なく, 電気特性のシミュレーション結果も良好であった. (検証のパラメータ: 総ゲート数 235,000, ゲート使用率: 36 パーセント同時変化ゲート: 35 パーセント, の条件で VSS2 25 本. 検証ソフト CMD.)

表 2: ルータチップピン配置

位置 (左回り)	信号の種類	入出力種別
上側面中央	W ポートの busy, X ポートの busy	入力
上側面左	A ポートの data 及び parity (39 ビット)	入力
左側面上	B ポートの data 及び parity (39 ビット)	入力
左側面中央	mclock, clock, min	入力
左側面下	C ポートの data 及び parity (39 ビット)	入力
下側面左	D ポートの data 及び parity (39 ビット)	入力
下側面中央	Y ポートの busy, Z ポートの busy	入力
下側面中央	D ポートの busy, C ポートの busy	出力
下側面右	Z ポートの data 及び parity (39 ビット)	出力
右側面下	Y ポートの data 及び parity (39 ビット)	出力
右側面中央	mout	出力
右側面上	X ポートの data 及び parity (39 ビット)	出力
上側面右	W ポートの data 及び parity (39 ビット)	出力
上側面中央	B ポートの busy, A ポートの busy	出力

6 ルータチップのパイプライン構成

本チップは、25MHz で動作予定のプロセッサクロックに同期させ倍の 50MHz クロックで、20ns 毎にデータ転送を行なう設計である。

6.1 優先度先送り方式と優先度キュー

優先度先送り方式は、自分のパケットバッファが一杯になった際、前段から送られてくる優先度をポート優先度と比較し、送られてきた方の優先度が高ければそれをポート優先度に格納する、という動作が必要である。しかし、この作業時間はパケットバッファとして用いる優先度キューの動作時間に隠蔽される。従って、本ルータチップ実現の技術ポイントは、同時入出力可能な高速優先度キューの実現にある。我々は、8 エントリ 32 ビット優先度という今回の条件に最も適しているものとして双方向シフトレジスタによる実現を提案し、CAD による設計を通して評価を行なった [PQ, VLPQ]。以下その実現方式を説明する。

6.2 優先度キューの実現

各レジスタは優先度を格納しており、新しく到着したパケットの優先度と各々自分の保持している優先度を比較する (この比較は、各レジスタ毎に用意されたコンパレータで同時に行なう)。その結果と合わせて出力の可否を見て、

1. データを保持する、
2. 新しいデータをロードする (パケット入力)、
3. データをフォワードシフトする (最右端のヘッドのレジスタからはパケット出力)、

4. データをバックワードシフトする、

という動作を選択する。表 3 と表 4 は各々、シフトレジスタのヘッド以外のものとヘッドの状態遷移である。同じ優先度の場合は、到着の順番を保持するようにしている。図中、X は don't care である。この方式では、自分の動作を決定する情報として、出力の可否と自分と左右のレジスタの優先度判定結果があればよい。

優先度情報をもつパケットヘッド (第 1 セグメント) 以外は、上記優先度キューからのポインタが指す 2 ポートメモリ (同時リード/ライト可能) の領域に保持する。以上の実現により異なるパケットに対する同時入出力の他、同一パケットについても、全体の到着を待たず次のルータへの出力が行なえる。

6.3 パイプライン構成

CAD によるシミュレーションにおいて、32 ビット 8 パケットの優先度キューのクリティカルパスの最大遅延は 24ns であった [VLPQ]。従って、20ns のステージ遅延を実現するため、優先度比較と、レジスタシフト及び書き込み/読み出しの 2 ステージに分ける。段間遅延 20ns の各ステージの動作は、

1. チップ外からのデータ入力
2. 優先度比較
3. レジスタシフト+データ入力及びデータ出力
4. 出力セレクト (ラウンドロビン制御含む)
5. チップ外へのデータ出力

の 5 段であるが、データ入力と出力はオーバーラップすることから、各ルータ当たり 4 段のパイプラインステージとな

表 3: 優先度キューの状態遷移表

Full	Output Enable	$D_{i+1} \geq D_{in}$	$D_i \geq D_{in}$	$D_{i-1} \geq D_{in}$	Action
X	1	1	1	1	Shift Forward
X	1	1	1	0	Load D_{in}
X	1	X	0	0	Hold
X	0	1	1	X	Hold
1	0	X	0	0	Hold
0	0	1	0	0	Load D_{in}
0	0	0	0	0	Shift Backward

表 4: 優先度キューヘッドの状態遷移表

Full	Output Enable	$D_i \geq D_{in}$	$D_{i-1} \geq D_{in}$	Action
X	1	1	1	Shift Forward
X	1	X	0	Load D_{in}
X	0	1	X	Hold
1	0	0	0	Hold
0	0	0	0	Load D_{in}

る。この構成では、非優先度動作モード（完全ラウンドロビン調停 + FIFO）では、優先度比較のステージを省けるであろう。

また、チップ外からのデータ入力と優先度比較、出力セレクトとチップ外へのデータ出力は統合できる可能性があるため、回路設計を行ないながら検討する。

プロセッサクロックが 20MHz となった場合は、ルータは 40MHz クロックで各段 25ns となる。この場合は、優先度比較のステージとレジスタシフト + データ入力及び出力のステージが融合できるため全体として 3 段パイプライン構成とすることが可能となる。

6.4 ゲート数

各入力ポート当たりのゲート数は、

- 優先度キューに 8K
 パケットの第 1 セグメント格納用の 39 ビット × 8 ワードのシフトレジスタ及び 33 ビットコンパレータ 8 組。
- 2 ポートメモリに 5K
 パケットの第 2 ~ 4 セグメント用。39 ビット × 24 ワードの 2 ポート (1r/1w) RAM であり、メモリシミュレーションで生成可能な 19 ビットと 20 ビットの 2 種類の RAM を組み合わせて使用。
- ロジックに 7K
 優先度先送り機構や同一優先度の際及び非優先度制御モードで用いられるラウンドロビン回路等を含む。

の 20K となり、ルータ全体では 80K 必要である。これにチップ全体として、保守用回路、エラー検出回路等も加わるため総計 90K 余りと推定される。

7 保守機能について

7.1 ルータチップへの保守機能の組み込みとピン数の制限

信号ピン数の検討にあるようにポート当たり 38 ビット（行き先 PE アドレス 6 ビット + 優先度 32 ビット）のデータ幅を実現すると、チップ全体の制御に使えるピン数はクロックを除き、3 本しかない。これらを用いてチップのリセット、パリティエラーの出力、優先度制御モード / 非優先度制御モードの切り替えの他、チップの動作テスト及び故障箇所発見のため内部状態の出力及び任意の状態の設定・参照等の保守機能を組み込む。

このため、3 本の保守用ピンは、mclock（入力）、min（入力）、mout（出力）として、mclock に同期して min のデータを読みとり、結果を mout から出力する。mout は、またチップ検査用（入力バッファを全て nand で結んでおき、入力バッファに全て 1 を入れ順次 0 の信号を増やしていくと出力がそれに依ってトグルすることを調べるもので、入力が正常に行なえることを確認できる）にゲートアレイメーカーから要求されている出力ピンも兼ねる。

min からのコマンド入力形式は現在検討中であるが、1 が連続した回数をコマンドにする方法や、1 の後定めた特定のビット数をコマンドとして解釈する（0 がそのビッ

ト数を越えて連続すれば強制的にコマンド受付状態に帰す)等が候補である。チップ組み込み回路の実現とホスト側のインタフェースの取り易さを考慮して決めたい。moutについても、データ出力中は、特定ビット毎に1を必ず立て、非出力状態と区別するようにしたい。

7.2 保守用コマンドの種類

保守用コマンドとして次のものを予定している：

- チップ検査状態 ON
全入力バッファの nand 出力を mout より出力しつづける。
- チップ検査状態 OFF
- チップリセット
パリティエラー検出 ON、優先度先送りモードとなる。
- パリティエラー検出 ON
パリティエラー検出時には、mount が 1 になる。パリティエラーを検出したチップはそのまま転送動作を続ける。パリティエラー信号はリセットがかかるかまで解除されない。
- パリティエラー検出 OFF
- 優先度制御モード
優先度先送り+優先度キュー。
- 非優先度制御モード
ラウンドロビン+ FIFO キュー。
- 内部状態設定 (複数)
コマンドで指定した入力ポートの状態を min からのデータに設定する。なお、出力ポート W, X, Y, Z の状態については、各々入力ポート A, B, C, D に統合して扱う。
- 内部状態出力 (複数)
mout から指定された入力ポートの内部状態を出力する。出力ポートについては、内部状態設定と同様に扱う。

7.3 保守のための内部状態の参照・設定方式

保守用回路以外の回路について、入力ポート単位で状態保持の FF 毎に裏 FF を設ける。裏 FF 同士は全体でシフトレジスタを構成し、状態参照時にはそこへ表 FF の値を取り込み、mclock に同期して mout から内部状態を出力する。状態設定時には、mclock に同期して min からデータを取り込み、最後に表 FF に値をセットする。裏 FF を用いることでパケット転送作業に影響を与えることなく状態の参照が可能となる。

7.4 ホストの機能について

各ルータチップ毎にホストから保守用信号 (mclock, min, mout) を用いアクセスできること。特に内部状態の設定・参照時には、mclock に同期して数百ビットの I/O があるためこれを扱える必要がある。また、ホストから clock, mclock の停止/再開が行なえることや、mout はパリティエラーの出力も兼ねているので、ホストからこの認識が行なえることも要求される。

8 今後の予定

今後 CAD による設計を進めつつ、細部を詰めて行きたい。この際の検討事項として重要なものには、39 ビットの 3 ポート (2r/1w) RAM の組み込み可能性の可否がある。これは、すでに述べた 2 ポート RAM と組み合わせて 4 ポート (3r/1w) RAM を実現するためであり、これが実装できれば、パケットバッファ (優先度キュー) からのパイプライン (異なる出力ポートに対してパケットのセグメントが 1 つずつずれる状態で同時に) 出力が可能となる。ゲートアレイメーカー側の見積りでは、優先度キュー+制御回路が 12K ゲート程度に抑えられれば、この方式の 4 ポート RAM が実装可能とのこと。設計を進めながら可能性を確かめる必要がある。全体で容量が少し足りない程度なら 3 ポートでの実現や入力バッファのサイズを現在の 8 パケットから縮小することも検討すべきであろう。

謝辞

現在の研究の機会を与えて下さった、柏木寛 電子技術総合研究所長並びに弓場敏嗣 情報アーキテクチャ部長に感謝致します。また、有益な意見を下さった情報アーキテクチャ部の皆様に感謝致します。特に、平木敬教授と児玉祐悦氏には保守用回路などについて貴重な助言を頂きました。

なお、本研究は、科学技術庁の平成 4 年度科学技術振興調整費による「センサフュージョンの基盤的技術の開発に関する研究」の一環として行ったものである。

参考文献

- [CODA] T. Shimada, K. Toda, K. Nishida, "Real-Time Parallel Architecture for Sensor Fusion", Journal of Parallel and Distributed Computing, Vol. 15, No. 2, pp. 143-152, June 1992.
- [PE93] 西田, 戸田, 島田, 山口, 「実時間並列処理計算機 CODA-プロセッサ」, RTP'93 資料, 1993 年 3 月.
- [PF1] 戸田, 西田, 坂井, 島田: 多段ネットワークにおける優先度制御方式の提案及びその評価, 情報処理学会 計算機アーキテクチャ研究会 (SWoPP'91), ARC89-22, 159/167 (1991).

- [PF2] 戸田, 西田, 坂井, 平木, 島田: 優先度先送り方式を用いたオメガネットワークの性能評価, 電子情報通信学会 コンピュータシステム研究会, CPSY91-53, 9/14 (1991).
- [PF3] 戸田, 西田, 内堀, 島田: 実時間用相互接続網の評価, 情報処理学会 オペレーティング・システム研究会 (SWoPP'92), 56-13, 97/102 (1992).
- [PF4] K. Toda, K. Nishida, S. Sakai, and T. Shimada, "A Priority Forwarding Scheme for Real-Time Multistage Interconnection Networks", Proc. of Real-Time Systems Symp., IEEE Computer Society TC Real-Time Systems, pp. 208-217, December 1992.
- [PQ] Nishida, K., et. al., "The Architecture of CODAR: A Parallel Processor for Real-Time Applications", Electrotechnical Laboratory, TR-91-19, 1991.
- [VLPQ] Nick Michell, Kenji Toda, Kenji Nishida, "A VLSI Priority Queue", ETL TR-91-27, Aug. 1991.
- [GADB] 0.7-Micron Array-Based Products Databook, LSI ロジック, 1990.
- [GAPKG] Package Selector Guide, LSI ロジック, 1991.