EC84－46

高並列推論エンジンＰＩＥにおける相互結合網の構成

Interconnection Networks on Highly Parallel Inference Engine －ＰＩＥ－

坂井 修一[ω] 田中 英彦[ω] 元岡 達[ω]
Shuichi Sakai Hiehiko Tanaka Tohru Moto-oka

[ω] 東京大学 工学部
Faculty of Engineering,The Unversity of Tokyo

１．まえがき

高度な知識情報処理に対応する能力を持つものとして，論理型プログラムへの期待が高まっている。論理型プログラムは，プログラムの仕様と実行ストラテジを分離して検証することができることから，人間の思考を自然に記述できる利点があり，ソフトウェア生産性の向上をもたらすと考えられている。その反面で，現在の計算機上で，実用規模の知識情報処理システムを構築するのに十分な処理速度を得るのは難しい。一方，論理型プログラムには高い並列性が内在しており，並列処理により計算速度の向上が実現できると考えられる。この目的から，我々は高並列推論エンジンＰＩＥ（１）～（３）の開発を進めてきた。

ＰＩＥはゴール書換えモデルに基づいて論理型言語の処理を行うマシンである。ゴール書き換えモデルの他にも手続き指向型・関係表指向型などの方式のマシンが現在検討されている。どの処理方式を採った場合も，並列推論マシンにおいては，プロセッサ同士あるいはプロセッサ・メモリ間のデータの授受に携わる通信系の役割は重要であり，相互結合網の転送速度はシステムの全体性能を決定するといえる。

開発中の様々な推論向き計算機（（10）（11）（12）（13）など）において，相互結合網の実現形態に関して触れられたものが幾つか見受けられる。例えば，ＡＬＩＣＥ（10）では，デルタ網（７）（８）とリングバスを用いることが提案されている。しかし，一般に，特定の相互結合網を適用する根拠が十分明らかにされているとはいえず，また，通信量の正確な見積りが報告されている論文は少ない。すなわち，網の選択は未定であるか，またはなお流動的であるのが現状である。

本稿では，ＰＩＥの相互結合網とそれに関連する制御の方式を提案し，検討を加える。簡単に本稿の特徴を述べれば，以下のようになる。

① マシンの実行環境を想定して行ったシミュレーションの結果（通信頻度，１回のデータ転送量など）を，結合網設計の根拠として用いる。

② 転送される５種類のデータを，４つのグループに分類し，それぞれに対応する機能をもった結合網の設計を行う。

③ 処理ユニットへの負荷の分散状況に動的に適応する結合網を提案する （４．１，４．２）。

本稿は，まず高並列推論エンジンＰＩＥの処理方式と構成の概要を述べ（２章），次いでＰＩＥにおけるデータ転送の特徴を抽出し，必要な結合網の種類と機能分担を示す（３章）。続いて，各網の実現方式について述べ（４章），これに考察を加え，今後の検討課題を明らかにする（５章）。

２．高並列推論エンジンＰＩＥ

２．１ 処理方式と機能モジュール

ＰＩＥは，ゴール書換えモデルに基づいて，Prologに代表される論理型のプログラムをＯＲ並列に直接実行するマシンである。

ゴールは基本的に互いに独立であり，ゴールプール中に格納されている。プロセッサは，ゴールプール中のゴールを取り出し，単一化（Unification）を行って新たなゴールを生成し，これに縮退（Reduction）を施して，再びゴールプールに格納する。ＰＩＥでは，以上の操作

を繰り返して処理を進め，実行結果を得る。

マシン内でゴールは，ＡＮＤ関係で結ばれたゴールリテラル（ゴールフレーム，Goal Frame，ＧＦ）として表現される。マシンは，機能単位である推論ユニット（ＩＵ）（図１）が複数台結合された構成をとる。ＩＵは，定義節メモリ（ＤＭ），単一化プロセッサ（ＵＰ），メモリモジュール（ＭＭ），アクティビティコントローラ（ＡＣ）から成る。ＤＭには単一化を行う定義節，すなわちプログラムが格納される。ＵＰでは単一化・縮退が行われ，親ＧＦおよび定義節の持っている必要なデータをコピーすることによって新しい子ＧＦが生み出される。（単一化と縮退はパイプライン的に実行される。）ＭＭはゴールプールの役割を果し，ＵＰは自ＩＵ内のＭＭのみからＧＦを受け取る。新しいＧＦは，現在の負荷の分散状況に応じて，適当なＩＵのＭＭに送り込まれる。各ＧＦは推論木上の各ノードと対応づけられており，ＡＣはそのノード操作を行う。このようにＰＩＥではゴールの処理とその制御が分離されている。

ＰＩＥの初期モデル（ＰＩＥ―Ｉ）では，新たなＧＦはＵＰ内で完全コピーにより生成されるとし，シミュレーション評価などを行った。しかし，この方式は常に大きなＧＦを処理する必要があり，縮退・転送の両操作に大きなオーバヘッドを生じる（１）（２）。そこで，未定義変数部を含まない構造データ部分（Ground Instance，ＧＩ）を，構造メモリ（ＳＭ）に格納してＧＦ間で共有し，必要な場合のみＵＰによって呼び出す方式の導入を検討した（構造データ共有方式）。シミュレーションによって構造データを共有する効果を調べたところ，従来の完全コピー方式と比較して，ＧＦ長が約１／２，縮退時間も約１／２になることが示された（２）。

また，ＰＩＥ―Ｉでは，１つ１つのＤＭがすべての定義節を持つことを仮定していたが，扱う問題が大きくなると，これは現実的ではない。ＤＭのキャッシュ化と，定義節のプールである定義節構造メモリ（ＤＳＭ）の導入によって，この問題に対処することを検討している。

２．２　ＰＩＥ―Ⅱの全体構成

ＰＩＥにおけるＳＭの設置方法には２通りが考えられる。全ＩＵが１台のＳＭを共有する集中型と，各ＩＵごとにＳＭを設ける分散型である。分散型は，メモリアクセス競合が少なく，各バンクが並列動作を行える利点がある反面，

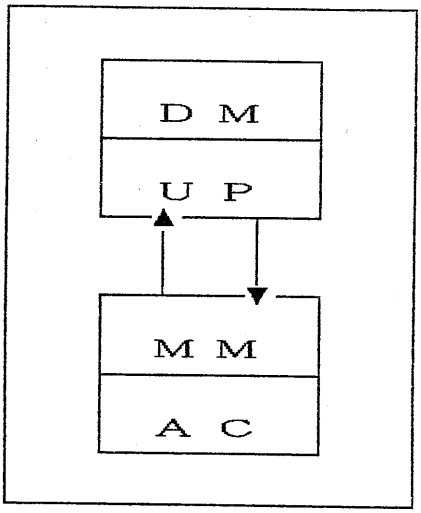


図１　推論ユニット
Fig.1　Inference Unit.

複数バンクにまたがるＧＩの読み出しに時間がかかり，ガーベジコレクション（ＧＣ）の手間も大きくなる欠点を持つ。

ＰＩＥでは，集中型ＳＭを採用する。このときＳＭの処理能力・アクセス競合の点から，１台のＳＭで賄えるＩＵの台数には限度があり，１６台程度が上限であることが示されている（２）。一方で，マシン全体のＩＵ台数としては数100 台から1000台程度を想定している。そこで，ＰＩＥ第２次モデル（ＰＩＥ－Ⅱ）ではマシンを階層的に構成することにし，構造データの共有は低いレベルのシステム内のＩＵ間でしか行わないことにする。

階層構成を採ることには，他にも処理の局所性を活かす利点がある。これによって，

①　ＧＦ転送時間の短縮

②　ノード情報（コマンド）転送時間の短縮

がもたらされる。

図２に，ＰＩＥ―Ⅱの全体構成を示す。レベル１システム（下位システム）は，ＩＵ１０数台とＳＭ１台より成り，この中でＧＦは構造データを共有する。システム内の負荷分散の監視と，他のレベル１システムとのインタフェースは，アクティビティマネジャ（ＡＭ）が司る。

１つのレベル１システムで処理しきれない仕事の場合，レベル２の結合網を介して，他のレベル１システムにＧＦが分配される。このＧＦは，ＧＩ部分を完全にコピーして持っており，レベル１システム間での構造データの共有は起らない。レベル２システムを構成するレベル１システムの個数は，目下６４程度を想定している。レベル２には，システムマネジャ（System Manager）があり，全システムの実行制御を司

っている。

なお，図中には示さなかったが，各レベル1システム内にはＤＳＭが1個置かれ，定義節プールの役割を果す。

（今後ＰＩＥとはＰＩＥ—Ⅱのことを指すものとする。）

2．3　レベル1—レベル2インタフェース

レベル1・レベル2間のインタフェース部では，通常の通信制御の他に，ＧＦと構造データの結合や切り分けなどの作業が必要である。また場合によっては，レベル1システム間で定義節の授受を行うことも考えられる。したがって，インタフェースの管理を行うＡＭと，ＳＭ・ＤＳＭは物理的にひとまとまりにして置く方が良いと考えられる。

言い換えれば，インタフェース部はＡＭ，ＳＭ，ＤＳＭを含み，これに作業用メモリであるインタフェースメモリ（ＩＭ）と，インタフェースプロセッサ（ＩＰ）を付加した構成となる（図3）。詳細は3．2で述べる。

3．ＰＩＥにおけるデータ転送と相互結合網

3．1　転送データの分類

ＰＩＥの相互結合網の設計に先立ち，マシンの機能モジュール間を移動するデータの特徴を把握し，要求される転送性能を明らかにする必要がある。

レベル1システム内で転送されるデータは，大きくわけて5種類ある（図4）。ゴールフレーム（ＧＦ），コマンド（ノード情報），構造データ，定義節，ＳＭアドレスがこれである。レベル2の結合網では，ＧＦ，コマンドおよび定義節が転送される。

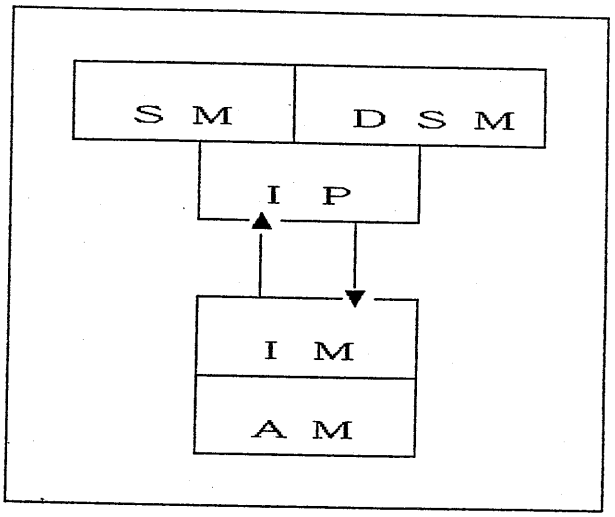


図3　レベル1－レベル2インタフェース部
Fig.3　Level 1 － Level 2　Interface Part.

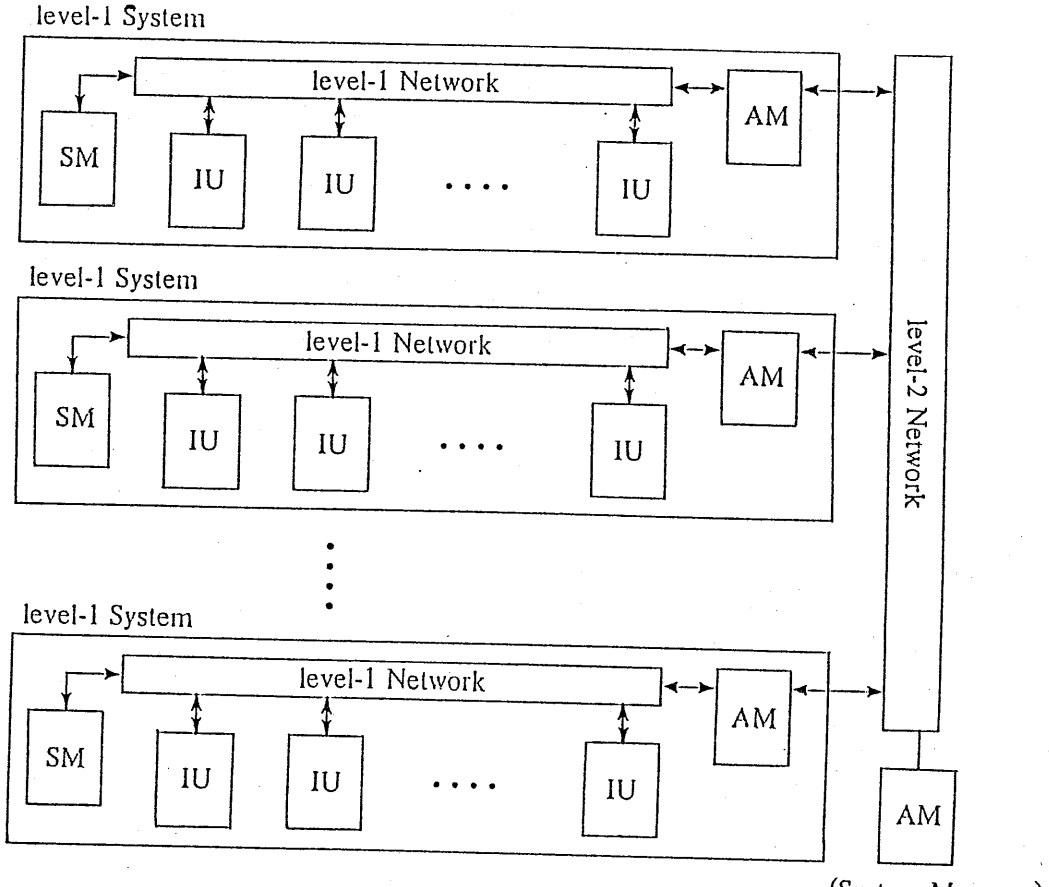


図2　ＰＩＥ－Ⅱの全体構成
Fig.2　Overview of ＰＩＥ－Ⅱ Architecture .

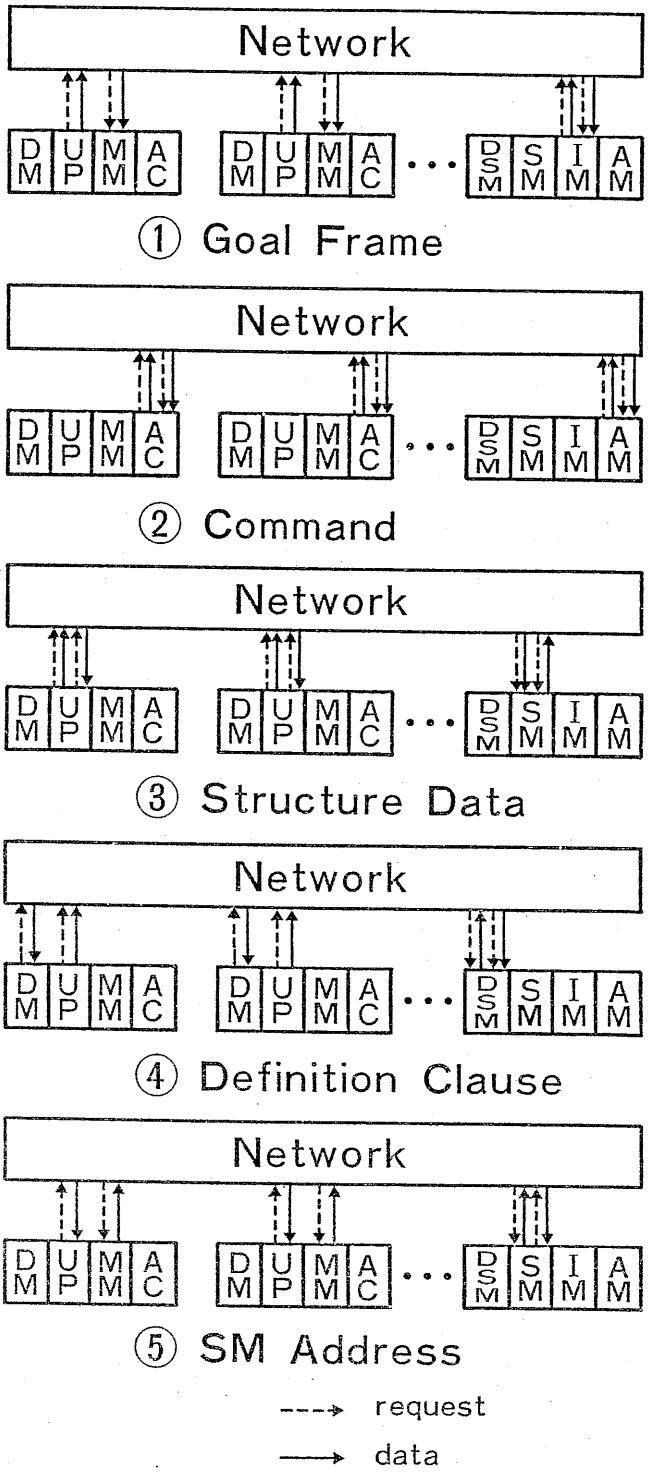


図4　レベル1システムにおけるデータ転送
Fig.4　Data Transfer in Level 1 System .

表1に各データの特徴を示した。大きさ・転送速度は，シミュレーションの結果得られた値である（1）（2）。

①　GF

UP内で生成されMMに送られるが，その際，同じレベル1システム内のMMに分配される場合と，IMを経由して他のレベル1へ分配される場合がある。

表1．レベル1システム内の転送データ
Table 1.　Transfer Data in Level 1 System.

データ		FROM	TO	大きさ	転送頻度*(/IU)
GF		UP UP IM	MM IM MM	数100B	1
コマンド		AC AC AM	AC AM AC	10～20数B	2～3
構造データ	△G	UP	SM	10～20B	0.5 ～ 1
	LFコマンド	UP	SM	約10B	0.1 ～ 1.4
	LFデータ	SM	UP	10～50B	0.1 ～ 1.4
定義節		UP DSM	DSM DM	数100B 数100B	<< 1 << 1
SMアドレス		SM MM	UP SM	約4B 約4B	0.5 ～ 1

＊UP内での1GFの平均処理時間Tu を単位時間とした値

本マシンでは，1GFの転送遅延は問題にならず，処理の流れを乱さないだけの高い転送スループットが要求される。

②　コマンド

AC間でやりとりされるノード情報であり，同じレベル1システムのAC間で転送が行われる場合と，IMを経由して他のレベル1へ分配される場合がある。数μsの低い転送遅延が要求される。

③　構造データ

UPで生成され，SMに転送・格納されるもの（△G）と，UPでの単一化に用いるためSMから読み出されるもの（追加読み出しデータ，LFデータ）がある。後者の読み出し時には，UPから追加読み出しコマンド（LFコマンド）を発行する必要があり，本稿においては，便宜上これも構造データの範疇に入れる。

追加読み出しには，数μsの早いレスポンスが要求される。

④　定義節

外部からの入力として与えられる場合と，UPにおいて生成される場合がある。網の立場から見れば，新しい定義節をDSMに格納する転送，DSM内の定義節をDMに読み出す転送に分類されるが，GF転送と比較して，どちらも頻度が小さい。

なお，DSMからDMへの転送は，一般にマルチキャストである（4．1参照）。

⑤　SMアドレス

SMアドレスは，以下の2つの目的で転送される。

(1)構造データ（△G）生成のとき，新しいGFにSM内の△Gを指すポインタを付加する。

(2)ガーベジコレクションのとき，MM中の生きているGFが参照している構造データに，（SM内で）印付けを行う。

前者では，UPで△Gが生成される速度に追随してアドレスの供給を行うことが必要である。また，表1中には(2)に関する転送頻度が示されていないが，これはシミュレーションによるデータが得られていないという意味であり，実際には，(1)の場合と同じくらいの頻度であると予想される。

①から⑤の転送頻度は，表1中では，UP内での1GFの平均処理時間Tuを単位時間とする相対値として与えられている。PIE－Iでは，この時間は約700クロック（試作UP（1）を用いた場合，約140μs）である。PIE－IIでは，この値は約1／2になると予想されている。

3．2　相互結合網の構成と役割分担

上述の5種（厳密には7種）のデータをやりとりするPIEの相互結合網の構成方式を述べる。本節では，結合網の種別と，通信における役割分担を示し，各網の具体的な実現方式については次章で議論することにする。

PIEの結合網としては，5種のデータそれぞれに，独立の結合網を割り当てる構成が最も自然で，かつ高い転送性能を期待できる方法と考えられる。しかし，網ハードウェア量が大きく必要になり，IUと網のインタフェースが複雑化する難点がある。そこで，大きさなどの点で共通点のあるGFと定義節は同一の網で，コマンド・構造データ・SMアドレスはそれぞれ独立な網で転送を行うことにする。このように，ハードウェアの許容範囲内で各種の転送データに個別に対応した網構成を採ることによって，各網の役割分担が明確になり，それぞれの通信機能が単純化され，転送の並列性も高まる。

したがって，レベル1システムは4種類の網を持つことになる（表2）。その内訳を以下に列挙して示す。

（Ⅰ）分配網（Distribution Network，DN）
　　GFと定義節をIUに分配する。

（Ⅱ）コマンド網（Command Network，CN）
　　AC間のコマンド転送を行う。

（Ⅲ）追加読み出し網（Lazy Fetch Network，LFN）
　　△G，LFコマンド，LFデータの転送を行う。

（Ⅳ）アドレス転送網（Address Transmission Network，AN）
　　SMアドレスの授受を行う。

同様にレベル2には，分配網（DN）とコマンド網（CN）がある。

IUと相互結合網の接続を図5に，レベル1－レベル2インタフェース部と結合網の接続を図6に示す。

表2．PIEの相互結合網
Table 2. Interconnection Networks on PIE

結合網	転送データ	特　　徴
DN	GF 定義節	高スループット マルチキャスト
CN	コマンド	低遅延
LFN	△G LFコマンド LFデータ	はやいレスポンス (数μs以下)
AN	SMアドレス	△G生成に追随するスループット

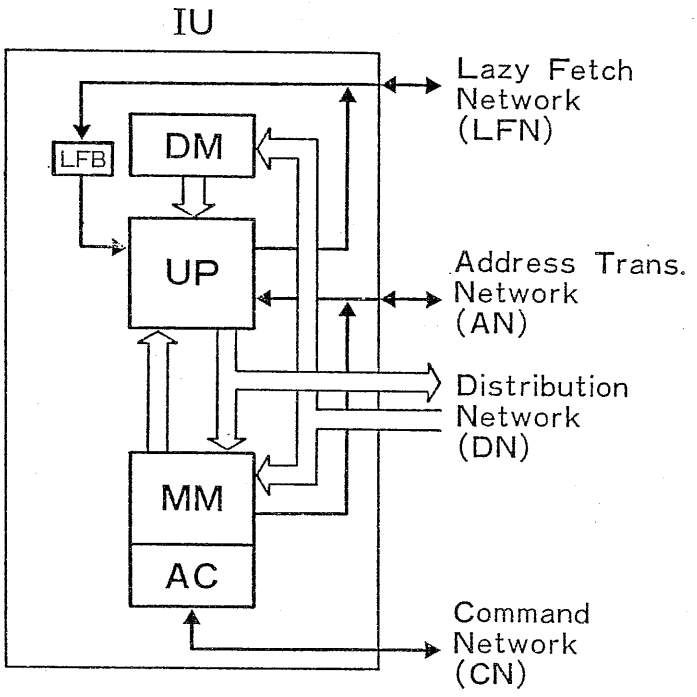


図5　IUと相互結合網
Fig.5　IU and the Interconnection Networks.

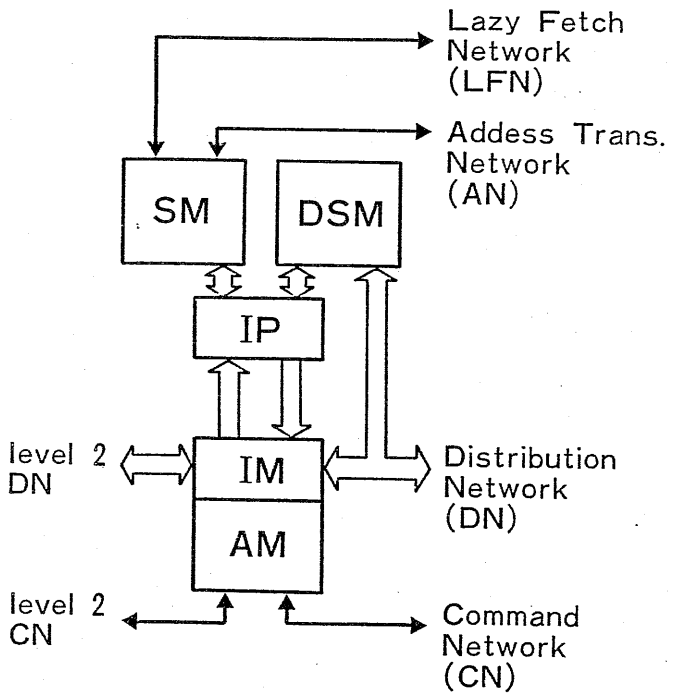


図6　レベル1－レベル2インタフェース部と相互結合網

Fig.6　Level 1 － Level 2 Interface Part and the Interconnection Networks.

4．各結合網の機能と実現方式

4．1　DN　（レベル1）

DNでは，GFと定義節が転送される。両方のデータとも数 100Bの大きさであり，本マシンの結合網の中では最も大きな容量が必要となる（表1，2）。

①　GFの転送

PIEでは，GFの分配によって負荷分散を実現する。したがって，

(1)転送時のスループットが高いこと

の他に，

(2)負荷の分散が均等であること

が要求される。

さらに，GF分配後のコマンドの転送を少なくするようにあらかじめ考慮しなくてはならない場合がある。例えば，notやguardを含むGFを多くのIUに分散させた場合，後に発生するコマンドの送受に要する時間が全体の処理時間に悪影響を及ぼすことが予想される。したがってこのようなGFはできる限り1IUに集中して置くべきである。

すなわち，3番目の要求として，

(3)後のコマンドの転送量が低く抑えられること

が挙げられる。

(1)(2)(3)は，必ずしも同時に満足され得る要求ではなく，場合によっては相矛盾することもある。3者のトレードオフを考慮したGFの行先制御が必要である。

②　定義節の転送

定義節は，処理の開始時に外部入力として与えられ，あるいは処理中にUPにより生成されてDSM内に格納される。DSMからの読み出しは，DMの要求によって起るが，1つのIUに必要な定義節（群）は，同じレベル1の他のIUにも必要である場合が多く，網はマルチキャスト機能を備えていなくてはならない。

DSMの読み出し要求は，10B程度のコマンドの形で与えられる。この読み出しコマンドの転送には，特殊な網を設けたり他の網を用いたりせず，DNを用いることにする。これは，網インタフェースの複雑化・ハードウェア量の増大を防ぐ意味からも妥当であり，また本コマンドは，転送量・転送頻度がともに小さいことから，GFや定義節の転送に悪影響を及ぼすことはないと考えられる。

①②より，DNでは，1対1のあらかじめ行先の決まっている通信の他に，行先を特定しない負荷の分配とマルチキャストという2種類の特殊な通信が行われることが示された。

最もハードウェアコストの低いDN（レベル1）の実現方法として，バスを用いる方式が挙げられる。しかし，要求されるスループットは1IU（1ポート）あたり数MB／s 以上であり，容量の点から適用が不可能である。一方，完全結合網やクロスバスイッチを用いれば高いスループットが得られるが，大きなハードウェア量が必要になる（O（N^2））。

格子型網・超立方体網・CCC網などの静的な結合トポロジー（5）の適用も考えられるが，これらは結合の局所性が大きく，問題をうまく分割してIUに割り当てないと，非局所的な通信が全体性能を低下させることになる。

レベル1のDNとしては，オメガ網（図7）（6）を適用する。オメガ網には結合の局所性がなく，したがって，プロセッサ間距離の大きな転送が処理の隘路となる危険がない。また，ハードウェア量もO（N log N）と妥当である。なお，オメガ網の構成単位であるスイッチング・ユニット（SU）は，ポート数4のものを用いる。

負荷（GF）の均等な分配を実現する方法として，AMによるIUの監視が挙げられる。しかし，完全に負荷の分散状況を監視することは，AMの処理オーバヘッド・CNの通信オーバヘ

ッドの点から現実的ではない。また，ＧＦの行先ＭＭをあらかじめ決めてしまうと，ＤＮ内の閉塞によって待たされるＧＦが多くなり，全体のスループットが低下する。

逆に，ＧＦの行先を完全にランダムに与えた場合，ＡＭ・ＣＮのオーバヘッドはなくなるが，負荷の不均等から処理効率が著しく低下する。さらにＤＮ内の閉塞の問題があり，このような環境では，理想的な場合の約１／２の転送スループットしか得られないことが解析されている（７）。

本マシンでは，この問題を以下に述べる方法で解決した。

ＤＮを構成するオメガ網には，ＧＦの転送経路とは逆向きに，すなわちＭＭからＵＰの向きに，ＧＦ転送の許可を与える信号線を設け，負荷を受け入れ可能なＭＭがあらかじめこの信号をオンにしておく。許可信号が立っている行先のうち，閉塞なしで到達できるものが選ばれ，ＧＦはそこへ転送される。このように簡単な制御機構を付加することで，スループットを高く保ちつつ負荷の均等な分配を実現する。

許可信号のオン・オフを決めるのは，ＩＵ内に現存する負荷の量であるが，その閾値は動的にＡＭが設定する。この作業は厳密に行う必要がなく，負荷の監視を行う周期も，ＧＦ転送時間に比較して十分大きく取れるため，ＡＭ・ＣＮのオーバヘッドはともに小さい。

４．２　負荷分散適応型ＳＵの設計

前節で述べたＤＮは，図８に示すＳＵで構成される。網は回線交換方式を考え，網制御は各ＳＵが独立に行う分散型である。

本ＳＵは，

① 行先を特定するモードと負荷分散を行うモードの２つの制御モードを持ち，ポートごとにモード指定を行う。

② マルチキャスト機能を持つ。

③ 同期式・非同期式の両方で使える。

などの特徴を持つ。

ＳＵは，データ部（Data Plane）と制御部（Control Plane）より成る。データ部は４入力４出力のクロスバスイッチであり，制御部からの信号（C_{00}～C_{33}）により，ゲートの開閉がなされる。

制御線としては通常見られるｒｅｑ，ａｃｋ，ｓｔｂの他に，ｒｍｄ（モード指定）とｌｄｃ（負荷受入れ可）が設けられている。前者は転送モードの指定（上記）を行い，後者は行先に

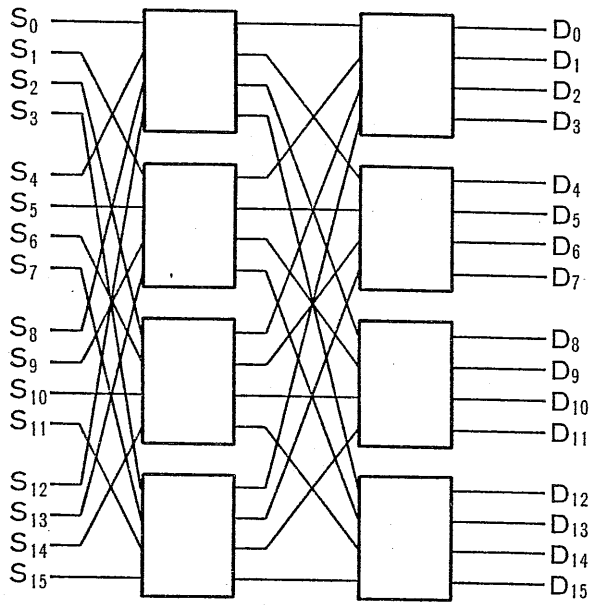


図７　４ポートのＳＵから構成されるオメガ網
Fig.7　Omega Network Consisting of 4 Port SU's.

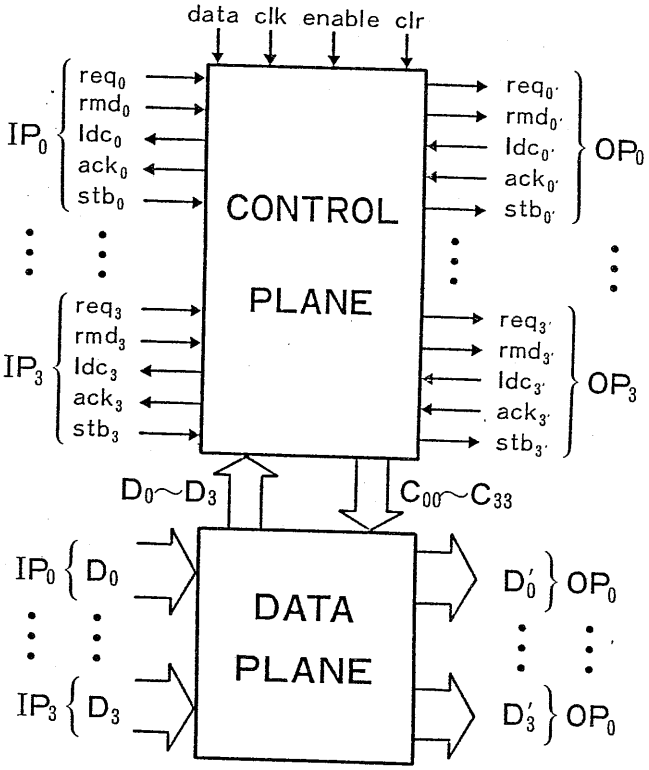


図８　負荷分散適応型ＳＵ
Fig.8　ＳＵ Adaptive for Load Balancing.

負荷を受入れることができるモジュールが存在するかどうかを示す。

制御部の入力ポート（ＩＰ）側には，ルーティング・コントローラがあり，モードに応じて出力ポート（ＯＰ）に転送要求を出す。これを受けたＯＰ側は，アービトレーションを行って，

適当なゲート制御信号（Ｃij）を立て，次段のＳＵに転送要求を出す。

行先を特定するモードでは，ソース側はreqを立てると同時にデータ線に行先モジュールの番地を出力し，各段のＳＵはこの情報をもとにルーティングを行う。負荷分散モードの場合，各段のＳＵはldcの立っている任意の空きＯＰを選択して経路を設定する。

本ＳＵの設計例を付録に示した（ＤＤＬによる記述）。４ポートの構成でデータ幅を９bit（１bitはパリティのチェックなどに用いる）とすると，ゲート数は600余り，入出力線の総数は116である。

なお，ＰＩＥのＤＮ（レベル１）に本ＳＵを適用する場合，規模の点から同期式を採るのが良いと考えられる。

４．３　ＣＮ　（レベル１）

ＣＮは，ＡＣ（またはＡＭ）間でコマンドの転送を行う網である（表１，２）。

コマンドの転送量は，レベル１全体で0.３～１．２ＫＢ／Ｔｕであり，比較的小さい（最大１５ＭＢ／ｓ程度）。一方で，数μsという低い転送遅延が要求される。

ＣＮは通常特定のＡＣ（ＡＭ）間の１対１の通信を行うが，負荷分散の制御に用いられる閾値（４．１参照）を設定する際には，同時に全ＭＭに情報を提供するのが望ましく，マルチキャストの機能が必要である。

以上のことから，レベル１のＣＮとして，総ハードウェア量の小さいバス方式を用いることにする（図９）。バスは，制御の高速性の点から個別要求論理とし，アービトレーションの柔軟性の点から集中管理型を考える。

各ＡＣ（ＡＭ）は，アービタに転送要求を出し，受理された（ａｃｋ）のを確認してから，コマンドをバス上に送り出す。その際，線数の節約のため，最初に行先アドレスを送る方式を採り，アドレスバスは設けない。転送の終了はＥＯＤ（End of Packet）信号によって知らされる。

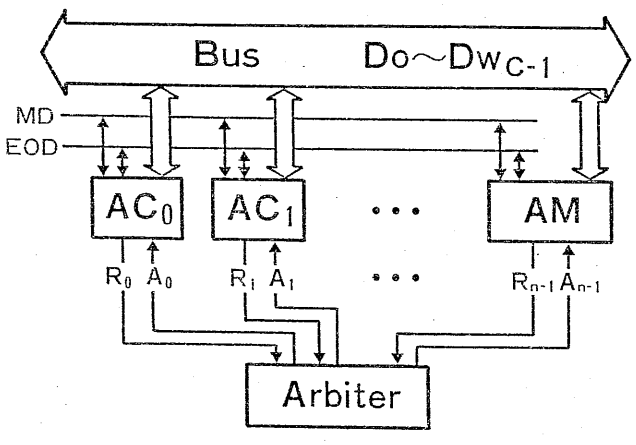
バス幅Ｗｃは，一般に大きいほど高い転送効率が得られると考えられるが，ハードウェア量の増大，データのスキューなどの問題がある。ＣＮの環境では，３Ｂ幅（Ｗｃ＝２４）で目標とする転送時間を達成できると考えられる。

４．４　ＬＦＮ　（レベル１）

ＬＦＮによって送られるデータは，△Ｇ，ＬＦコマンド，ＬＦデータの３種である（表１，２）。

３者の転送量を合計すると，レベル１全体で0.１～２ＫＢ／Ｔｕ程度である（最大２５ＭＢ／ｓ）。一方で，追加読み出しの応答（往復とＳＭ内の処理）は，単一化処理を中断して行われるため，十分に短い時間で実現されねばならない（数μs）。

このことと，現在検討中のＳＭが集中型であ

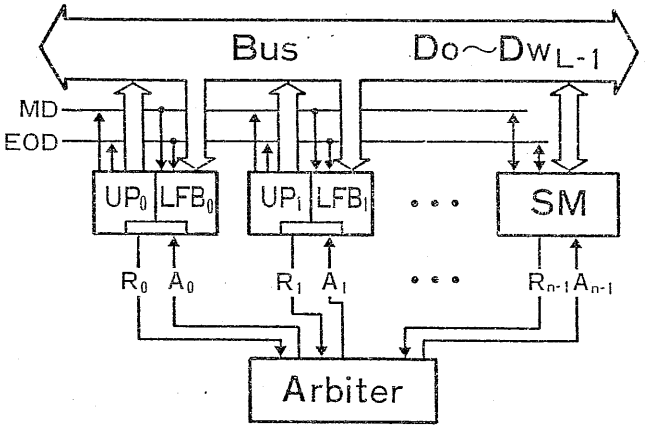


MD：Bus Mode　EOD：End of Data

Ri　：Request i　Ai　：Acknowledge i

図９　集中管理型個別要求論理バスによるＣＮの実現

Fig.9　CN Construction by Bus, with Integrated Control and Independent Request Signals.



MD：Bus Mode　EOD：End of Data

Ri　：Request i　Ai　：Acknowledge i

図１０　集中管理型個別要求論理バスによるＬＦＮの実現

Fig.10　LFN Construction by Bus, with Integrated Control and Independent Request Signals.

ることから，ＣＮと同様に，集中管理型個別要求論理バスによってＬＦＮを構成することにする（図１１）。バス幅W_Lは，５Ｂ（４０本）程度で十分に短いレスポンス時間を得ることができる。なお，転送要求が衝突した際に，後回しにされた追加読み出しの応答が遅くならないように，アービトレーション・転送・ＳＭの内部処理の３つの処理を重畳化する。

４．５　ＡＮ　（レベル１）

ＡＮは，ＳＭからＵＰに空きアドレスを提供し，使用中のＳＭアドレスをＭＭからＳＭに知らせるための結合網である（表１，２）。

各ＵＰ内にはＳＭの空きアドレスを蓄えるバッファがあり，△Ｇ生成時に１つずつこれが使われる。したがって，ＡＮでは転送遅延は問題とならず，△Ｇの生成・消滅に追随するスループットを得ることが問題となる。

これは，レベル１全体で0.1ＫＢ／Ｔｕ程度と小さい（約１．３ＭＢ／ｓ）。したがって，ビットシリアルな時分割多重チャネル方式のリングバス（図１１）をＡＮに適用することにする。

チャネルには，データの型や有効なデータが入っているか否かを示すヘッダを設ける（この場合，転送単位は固定長でよい）。

４．６　レベル２の結合網

レベル２には，ＤＮ・ＣＮの２種類の結合網があり，前者はＧＦと定義節，後者はコマンドの転送に携わる。

レベル２のＤＮによって転送されるＧＦは，構造データを完全にコピーして持つものであり，レベル１のそれと比較して大きい。定義節およびコマンドは，レベル１のそれとほぼ同じ大きさであると考えられる。転送頻度は３者ともにレベル１のものより低いと予想される。

２種類の網に関しては，レベル１システム間でやりとりされるデータのトラヒックを見積り，その結果から実現方式の検討を行う予定である。ＤＮとして，４．１で述べた負荷分散適応型のオメガ網を適用することが考えられ，またＣＮとして集中管理型個別要求論理バス（４．３）の適用が考えられる。

なお，目下レベル２システムは６４個のレベル１システム（全ユニット数１０２４台程度）より成る場合を想定しており，したがって，２種の網はともに６４ポートを持つ。

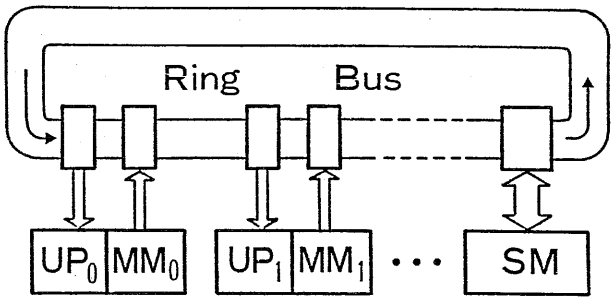


図１１．時分割多重チャネル型リングバスによるＡＮの実現

Fig.11　AN Construction by Time Slotted Multi-channel Ringbus.

５．考察

レベル１のＤＮとしてオメガ網を用いることを考えた。オメガ網は１ルートの多段結合網であり，一般にその欠点として，閉塞率が高いことと信頼性が低いことが指摘されている。本マシンの環境において前者を解決する方法を４．１に示した。

信頼性に関しては，パリティなどによるエラーチェックの他に，次のような対処法が考えられる。

①　ＳＵ内の論理回路の多重化

②　ＳＵの付加による網の多ルート化(4)(9)

このうち②は，例えば１段分のＳＵをオメガ網に追加し，故障部を迂回するルーティングを行うことなどが挙げられる。この場合，１組の入出力の組合せに対し，４つの経路が設定できる。

また，本稿では，負荷分散の大まかな監視をＡＭが行う場合を想定したが，負荷分散に関する制御をＤＮだけで行う方法が考えられる。また，相互結合網を考慮に入れた負荷分散の評価（解析・シミュレーション）を行い，ここで提案した方式の検証を行う必要がある。

ＰＩＥの高機能化（３）とともに，ＳＭに対する要求が増す可能性がある。高頻度のアクセスが起る環境では，ＳＭを多バンク化し，多段結合網でＩＵと接続する対処法が考えられる。

全体を通じて，ＰＩＥの実行環境を想定しての具体的なシミュレーション評価が必要である。

なお，各網のインタフェース部に関しては触れなかったが，データ転送の優先順位決定・エラーチェックなどを含めたインタフェースの設計は重要な課題である。

6．むすび

本稿では，高並列推論エンジンＰＩＥの相互結合網に関して，レベル１システムを中心に，その実現方式の検討を行い，ＰＩＥ－Ⅱの処理速度に十分に追随する網構成を示した。現在，相互結合網を考慮した負荷分散に関して，より一般的なマシン環境を想定した解析・シミュレーションを行っている。また，本稿で述べた結合網は，ＰＩＥ－Ⅱの階層的シミュレータ（３）に組み入れられ，全体的な評価が行われる予定である。

文　献

（１）Moto-oka,T., Tanaka,H., Aida,H., Hirata,K.,Maruyama,T. : “The Architecture of a Parallel Inference Engine -PIE-”, Proc. Int. Conf. of FGCS 1984 pp.479-488 (Nov. 1984).

（２）平田，田中，元岡：“ＰＩＥにおける構造メモリの構成について”，アーキテクチャワークショップインジャパン ’84シンポジウム（1984- 11）.

（３）丸山，平田，相田，田中，元岡：“高並列推論エンジンＰＩＥの階層的構成とそのシミュレータ”，本研究会技報（1984-12）.

（４）坂井，計，田中，元岡：“汎用スイッチング・ユニットを用いた高並列計算機の相互結合網”，信学技報，ＥＣ84-18,（1984-07）.

（５）Feng,T. : “A Survey of Interconnection Networks”, IEEE COMPUTER, 14, 12, pp.12-27 (Dec. 1981).

（６）Lawrie,D.H. : “Access and Alignment of Data in an Array Processor”, IEEE Trans. Comput. C-24, 12, pp. 1145-1155 (Dec.1975)

（７）Patel,J.H.: “Performance of Processor-Memory Interconnection for Multiprocessors” IEEE Trans. Comput. C-30, 10, pp.771-780 (Oct.1981).

（８）Dias,D.M., Jump,J.R.: “Anlysis and Simulation of Buffered Delta Networks”, IEEE Trans. Comput. C-30, 4, pp.273-282 (Apr.1981).

（９）Adams,Ⅲ,G.B., Siegel,H.J.: “The Extra Stage Cube: A Fault-Tolerant Interconnection Network for Supersystems”, IEEE Trans. Comput. C-31, 5,pp.443-454 (May.1982).

（10）Darlington,J., Reeve,M. : “ALICE and the Parallel Evaluation of Logic Programs”, Reserch Report, Imperial College, London (June 1983).

（11）Ciepielewski ,A., Haridi ,S. : “A Formal Model for OR-Parallel Execution of Logic Programs”, IFIP1983, pp.299-305 (Sep.1983).

（12）伊藤，益田，清水，来住，久野：“データフフロー方式並列推論マシンのアーキテクチャ”，Proc. of the Logic Prog. Conf. ’84, 7-1,（1984—03）.

（13）Hasegawa,R., Amamiya,M.,: “Parallel Execution of Logic Programs based on Dataflow Concept”, Proc. Int. Conf. of FGCS 1984, pp.507-516 (Nov. 1984).

付録　負荷分散適応型スイッチング・ユニットの構成例（ＤＤＬによる記述）

Appendix　Construction of a Switching Unit adaptive for Load Balancing (Example Described in ＤＤＬ).

```
<SYSTEM> CSU:                                   /* circuit switching SU */
  <TIME>
    CLK0<(10)>,                                 /* arbiter clock */
    CLK<(40,0)>.                                /* selector clock */
  <ENTRANCE>
    IPIN(4,12),                                 /* IP input */
    OPIN(4,2),                                  /* OP input */
    DATA,ENABLE,CLEAR.
  <TERMANAL>
    IREQ(4), IRMD(4), ILDC(4), IACK(4), ISTB(4), ID(4,9),
    OREQ(4), ORMD(4), OLDC(4), OACK(4), OSTB(4), OD(4,9),
    C(4,4), IIREQ(4,4), DREQ(4,4), DACK(4,4), ACK(4,4)
  .
  <BOOLEAN>
    <<I = 0 : 3>>
        ID(I) = IPIN(I,0:8),
        IREQ(I) = IPIN(I,9),
        IRMD(I) = IPIN(I,10),
        ISTB(I) = IPIN(I,11),
        OLDC(I) = OPIN(I,0),
        OACK(I) = OPIN(I,1),
        OD(I) = ID(0) & C(0,I) | ID(1) & C(1,I) |
                ID(2) & C(2,I) | ID(3) & C(3,I),
        ILDC(I) = |/(OLDC & ~OREQ)
      /* loop end */
    /* boolean end */

<AUTOMATON> IP4 : CLK :
  <REGISTER>
    RR(4,4),                                    /* routing register */
    DR(4,4),                                    /* distribution register */
    DLR(4,4)                                    /* dist. latch register */
  .
  <TERMINAL>
    DRE(4), DST(4,2)
```

```
<DELAY>
  IRE(4)<(5)>

.
<LOGIC>
  <<I = Ø : 3>>
    IACK = |/ACK(I)

  .
  IRE = IREQ & ~IRMD,
  DRE = IREQ &  IRMD,
  <<I = Ø : 3>>
   /* address converter */
    DST(I,Ø) = |/(RR(I) & ID(I,Ø) || ID(I,2) ||
                         ID(I,4) || ID(I,6)),
    DST(I,1) = |/(RR(I) & ID(I,1) || ID(I,3) ||
                         ID(I,5) || ID(I,7)),
    IIREQ(I,Ø) = ~DST(I,Ø) & ~DST(I,1) & IRE(I),
    IIREQ(I,1) = ~DST(I,Ø) &  DST(I,1) & IRE(I),
    IIREQ(I,2) =  DST(I,Ø) & ~DST(I,1) & IRE(I),
    IIREQ(I,3) =  DST(I,Ø) &  DST(I,1) & IRE(I),
    /* routing register update */

     |* ENABLE & IREQ(I) *|
       RR(I) <- RR(I,1:3) || DATA

     .,
     /* r.r.update end */
    /* address converter end */
    /* ditribution selector */
     DREQ(I) = DLR(I),
     |* ~&/ DACK(I) */
       DR(I) <- {~|/DR(I,Ø:2)) || DR(I,Ø:2),
       DLR(I) <- DRE(I) & DR(I)

     .
    /* distribution selector end */
   . /* loop end */
  . /* logic end */
<END> IP4.
```

```
<AUTOMATON> OP4:CLKØ:
  <REGISTER>
    AR(4,4)              /* arbitration register */
  .
  <TERMANAL>
    IC(4)
  .
  <BOOLEAN>
    IC = OLDC & (OREQ ~& ~ORMD),
    <<J = Ø : 3>>
      <<I = Ø : 3>>
        ACK(I,J) = C(I,J) & OACK(J)
      .,
      OSTB(J) = ISTB(Ø) & C(Ø,J) |ISTB(1) & C(1,J) |
                ISTB(2) & C(2,J) |ISTB(3) & C(3,J)
    . /* loop end */
  . /* booleazn end */
  <LOGIC>
    <<J = Ø : 3>>
      <<I = Ø : 3>>
        C(I,J) = (IC(J) & DREQ(I,J) | IIREQ(I,J))
                  & AR(J,I), /* AR is special */
        DACK(I,J) = DREQ(I,J) & C(I,J)
      .,  /* I loop end */
      OREQ(J) = C(Ø,J) | C(1,J) | C(2,J) | C(3,J),
      ORMD(J) = DACK(Ø,J) | DACK(1,J) |
                DACK(2,J) | DACK(3,J),
      |* ~OREQ(J) *|
        AR(J) <- ( ~|/ AR(J,Ø:2) ) || AR(J,Ø:2)
      .  /* if end */
    .  /* J loop end */
  .  /* logic end */
<END> OP4.

<END> CSU.
```