並列オブジェクト指向トータルアーキテクチャA-NET
7D−6　－ハードウェアの設計方針－

吉永 努　寺岡 孝司　茂木 久　佐々木 昌　馬場 敬信
yoshi@infor.utsunomiya-u.ac.jp
宇都宮大学工学部

1　はじめに

A-NET 計算機は，並列オブジェクト指向言語 A-NETL を高速実行するために設計された高並列計算機である [1, 3, 4]．320KB の局所メモリを持つトポロジ独立なノードプロセッサをネットワークにより結合した分散メモリ形態をとり，メッセージ駆動方式でプログラムを実行する．各ノードプロセッサは，メソッドを実行する PE と通信を制御するルータからなる．プロトタイプ PE の試作においては，命令セットレベルの設計変更にも柔軟に対応できるよう，LSI ビルディングブロックを用いたマイクロプログラム制御方式を採用した．ルータは，各種ネットワークトポロジによる結合を可能とするため，プログラマブル素子を用いて通信制御装置を構成するとともに，ノードプロセッサ間での非同期的なパケット交換に対応できる構成とした．

本稿では，A-NET 計算機の設計方針とハードウェア構成の概略について述べる。

2　設計方針

A-NET の最大の特徴は，高並列プログラムの開発容易性を重視して並列オブジェクト指向実行モデルを核概念としたソフトウェア指向アプローチを採用し，高並列ソフトウェアの開発環境とその実行マシンを統合的に検討していることである．したがって，ハードウェアの設計方針も，ソフトウェアからの影響を強く受けたものとなっている．マシン全体の設計方針は次の 2 点である．

(1) 高並列計算機の構成要素として，VLSI 指向の専用ノードプロセッサを開発する．

(2) ノードプロセッサの構成をネットワークトポロジ独立とし，応用によって適切なトポロジを選択できるようにする．

第 1 の点は，汎用マイクロプロセッサと UNIX オペレーティングシステムなどをベースとした最近の商用超並列マシンと異なり，我々のアプローチが独自の並列処理記述言語および実行モデルから発していることによる．必要最小限のハードウェア構成で実行に最適なアーキテクチャを構築し，ノードプロセッサの簡単化とスケーラビリティを確保しようとするものである．ただし，このことは特定の応用分野のみを想定していることとは異なる．並列オブジェクト指向は，汎用の実行モデルである．

第 2 点は，高並列プログラムにおける多数オブジェクトの定義法と負荷分散のコストをソフトウェア／ハードウェア両面から検討した結果による．これまでの経験から，多くの問題において同一のメソッドを持つ多数のオブジェクトを同時に起動することが有効であることが分かってきた．A-NETL では，ユーザが柔軟性と実行効率を選択できるよう，静的／動的な多数オブジェクトの生成機能を設けるとともに，トポロジ独立なオブジェクトの最適割り付けをソフトウェアサポートしている [2]．

3　A-NET 計算機のハードウェア構成

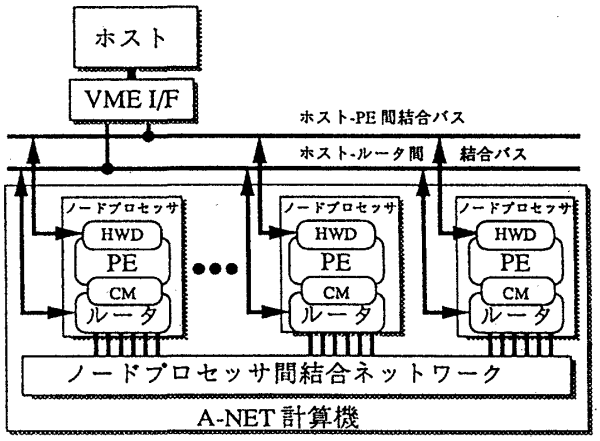
図 1 に A-NET 計算機の全体の構成概略とホストとの結合の様子を示す．A-NET 計算機の各ノードプロセッサは，ルータの持つ 6 本のリンクにより相互に結合される．結合トポロジは基本的に任意であるが，現在までにバイナリ n キューブ，2D, 3D メッシュ，トリーなどに対応するルータのロジックを検討している．各ノードプロセッサは，PE，ルータ，PE-ルータ間共有メモリ（CM），デバッグ用付加回路（HWD）からなる．PE は，OS およびユーザ定義のオブジェクトを割り付ける 280KB のローカルメモリを持ち，メッセージを受理してそれに対応するメソッドを実行する [5]．ルータは，オブジェクト間のメッセージパッシングとオブジェクト生成に伴うコード転送を制御する．

PE とルータの結合法には，

(1) メッセージ送受信バッファを共有メモリに持ち，完全に並列動作する．

(2) 送受信バッファどちらかを共有し，半独立に動作する．

(3) 必要に応じてメモリを占有し，交互に動作する．



HWD: ハードウェアデバッグ用付加回路　CM : PE-ルータ間共有メモリ

図1　A-NET計算機とホスト結合

A Parallel Object-Oriented Total Architecture A-NET,
– Design Principles –,
Tsutomu YOSHINAGA, Takashi TERAOKA, Hisashi MOGI,
Syo SASAKI and Takanobu BABA
Faculty of Engineering, Utsunomiya University

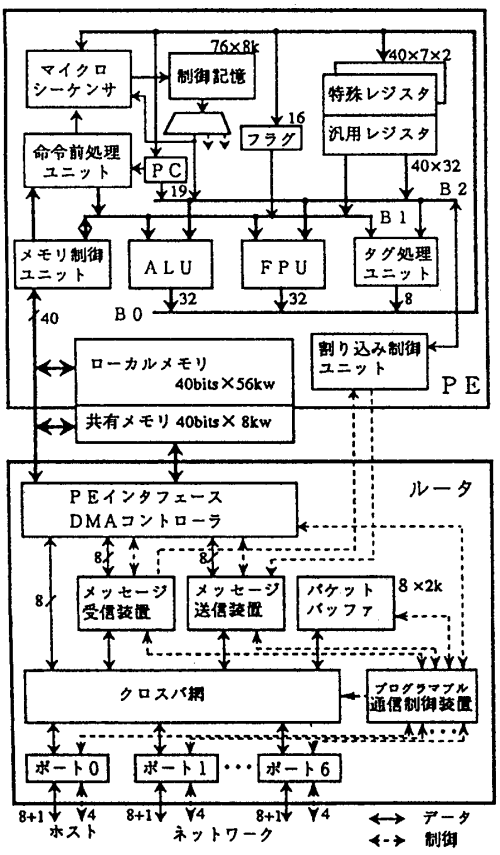


図2．ノードプロセッサの構成

などが考えられる．我々は，はじめ(1)の方法を用いたが，受理したメッセージをOSにより共有メモリからローカルメモリにコピーするなどの処理が必要となり，メッセージ受信時のPE，ルータの並列動作の恩恵が少ないと判断して(2)の方法をとることにした．すなわち，メッセージ送信バッファとメモリの空き領域情報などを40KBのCM上に共有し，自ノード宛メッセージの受信時以外はPEとルータが並列に動作するものである．

HWDは，PEの動作をホストから検証するための付加回路である．本来，ノードプロセッサはルータのホスト用ポートを介してのみホストと接続される訳であるが，プロトタイプPEのハードウェアデバッグを行う目的でルータを経由せずに，ホストから直接PE内部ファシリティの読み書き，実行制御ができるようにする．これに使用するモニタは，ホスト上にCを用いて作成した．

ホストにはSONY NEWS-1750を用い，VMEインタフェースを使用してA-NET計算機と接続する．プロトタイプにおいては，このインタフェース部に各ノードプロセッサからの割り込み要求を調停する回路を設け，16ノードまでをバス結合する．

4 ノードプロセッサのアーキテクチャ

図2にノードプロセッサのブロック図を示す．以下に，PEとルータの構成概略について述べる．

4.1 PE

PEは，VLSI化を目標としてできるだけ簡単な構成とし，オンチップメモリを想定したローカルメモリ中心のアーキテクチャになっている．したがって，キャッシュメモリや大容量レジスタファイルは持たず，機械命令のオペランドも基本的にメモリ上のデータが指定される．命令セットには，メッセージ送受信などのA-NETL指向のものを定義し，マイクロシーケンサやALUなどにAMD29300シリーズのLSIビルディングブロックを用いて設計した．これにより，機械命令レベルの設計変更にも柔軟に対応できる構成となっている．実際，シミュレータによる実行評価データなどを検討して，これまで何度か命令セットの改良，拡張を行っている．ローカルメモリは，1語40ビットであり，演算部は8ビットのタグ処理と32ビットの算術論理演算を並列に行う．

4.2 ルータ

ルータには，プログラマブル通信制御装置（PCC）を設け，そのロジックを書き換えることによりネットワークトポロジ可変性に対応する．ルータ−ルータ間でのメッセージ交換は，適応型／バーチャル・カットスルー方式で行う．これは，メッセージの経路選択をPCCで一括して行うため，各ポートでパケットをバッファリングする必要があることなどによる．

また，ルータ内でのデータ転送をできるだけ並列に行うため，ポートやメッセージセンダ／レジーバなどのブロックをクロスバ網で結合するとともに，各ブロックに小規模のシーケンサを持たせている．

5 おわりに

A-NET計算機の設計方針とその構成概要について述べた．現在，PEは基板のレイアウト設計を行っているが，IC数190個程度で4層，約50cm×50cmとなる予定である．またルータは，各ブロックの論理設計を行っている段階であるが，それぞれのシーケンサを22V10程度のPLDで実現すると実装が困難であることが分かったため，今後高密度のデバイスを用いてハードウェアの集積化を行う必要がある．

参考文献

[1] Baba, T. et al.: A Parallel Object-Oriented Total Architecture: A-NET, *Proc. Supercomputing '90*, pp.278-285 (1990).

[2] Baba, T. et al.: A Network-Topology Independent Task Allocation Strategy for Parallel Computers, *Proc. Supercomputing '90*, pp.878-887 (1990).

[3] 馬場:超並列マシンへの道，情報処理，Vol.32, No.4, pp.348-364 (1991).

[4] Yoshinaga, T. and Baba, T.: A Parallel Object-Oriented Language A-NETL and Its Programming Environment, *Proc. COMPSAC '91*, pp.459-464 (1991).

[5] Yoshinaga, T. and Baba, T.: A Local Operating System for the A-NET Parallel Object-Oriented computer, *J. Inf. Process.* (to appear).