

並列計算機ADENAシステムのプログラミング環境

5L-10

佐々木真司 若谷彰良 森康浩 岡本理

松下電器産業(株) 半導体研究センター

1. はじめに

ADENAシステムは、数値シミュレーションを目的とした並列計算機で京都大学と共同で開発している。ADENAシステムにおけるプログラミングはFORTRANに並列性を付加した言語であるADETRANによって行われるが、今回はADENAシステムでのプログラミングをサポートする環境について述べる。

2. ハードウェア構成

ADENAはホスト計算機(以下、ホスト)と256個のPE(Processing Element)およびADENAネットワークから構成されている。PEは図1に示すように、EPU(Element Processing Unit), TCU(Transfer Control Unit), LIM(Local Instruction Memory), LDM(Local Data Memory)から構成されており、ネットワークはSRC(Send Receive Controller), BMU(Buffer Memory Unit)から構成されている。

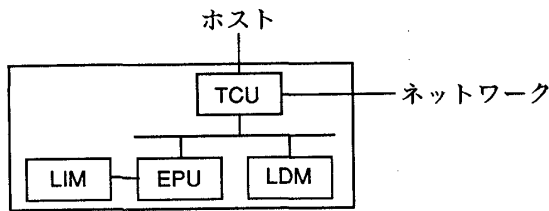


図1 PEの構成

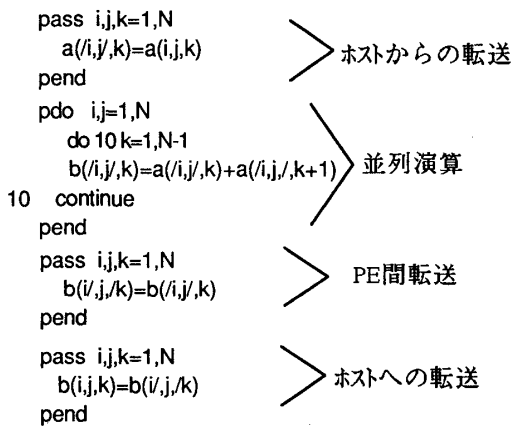


図2 ADETRANプログラム例

3. ソフトウェア構成

(1) ソースプログラム

ホストはUNIXをOSとするワークステーションで、この上でADETRANによりアプリケーションプログラムの開発を行う。ADETRANは、ADENAシステムでプログラミングを行うための高級言語で、基本的な文法はFORTRANに従い、並列演算やPE間転送などの処理を付加している。図2にその例を示す。

(2) システムプログラム

ADENA上での実行を行うため、図3に示す様にADETRANコンパイラ、ADENAリンクが用意され、またEPUアセンブラコードでのプログラム開発用にEPUアセンブラと逆アセンブラが用意されている。

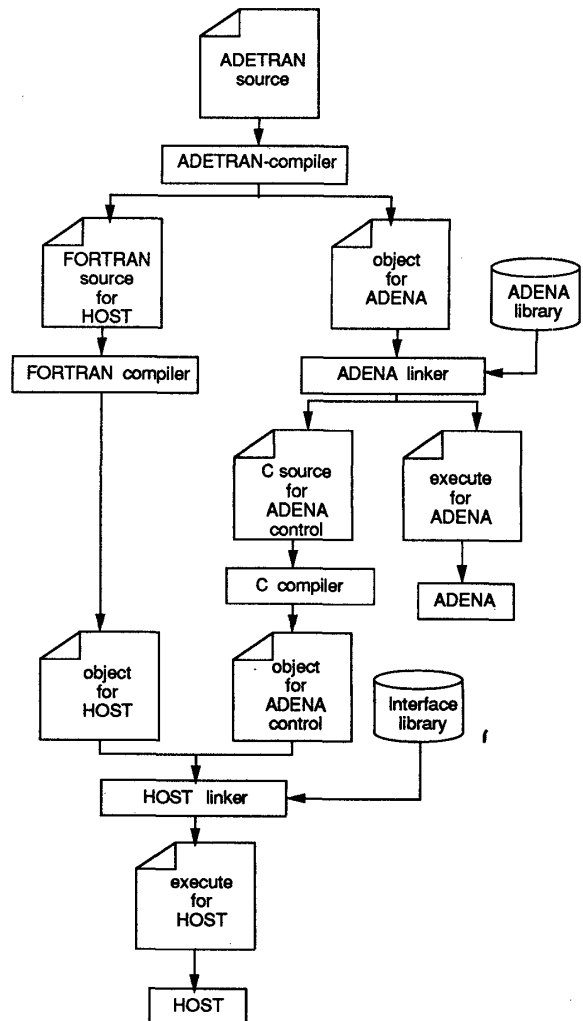


図3 ADETRANプログラムの流れ

Programming environment of the ADENA system  
 Shinji SASAKI, Akiyoshi WAKATANI,  
 Yasuhiro MORI, Tadashi OKAMOTO  
 Matsushita Electric Industrial Co.,Ltd.

### (3) ライブラリプログラム

ホストからADENAを制御するインターフェイスライブラリとしてADENAコントロールルーチンが用意され、ADENA上に配置されるADENAライブラリとして、スタートアップルーチン、割込み処理ルーチン、数学関数、ADENA サブルーチンパッケージ (FFT、三重対角ソルバー等) が用意されている。さらに、システム立ち上げ時にハードウェアをチェックするADENA診断プログラムが用意されている。

## 4. ハードウェアデバッグツール

モニタツールは主にハードウェアに依存するデバッグを目的とし、ハードウェアのレベルに応じて用意している。共通した機能として、LIM/LDMへのアクセス、PEの制御、PEの状態監視がある。

### (1) PEモニタ

実機とは別のシステムとして1PEで構成され、PEにおける演算処理制御を行い、1クロックごとのトレース機能がある。

### (2) ADENAモニタ

全PEとネットワークから構成され、ネットワークを用いた演算・転送処理制御を行い、転送コントローラTCUを自由に設定する機能がある。

### (3) システムモニタ

実機上のモニタで、ホストからADENAを制御する。コマンドとして用いることができる。

## 5. デバッグ方法

ADETRANプログラムのデバッグ方法を、エラー発生原因別に示す。

### (1) 実行前のエラー発生検出

- ・ 解法のアルゴリズム FORTRANプログラムによる解法確認 → ADETRAN化
- ・ 配列の宣言、引数、呼び出し等 ADETRANコンパイラ、ADENAリンカによるエラーチェック

### (2) 実行時のエラー発生検出

- ・ PE間転送 システムモニタ/ADENAモニタによる解析
- ・ 並列演算 システムモニタ/ADENAモニタ/PEモニタによる解析

実行時の値をチェックするためには、実行単位ごとに値をホスト上へ戻すよう、あらかじめプログラミングしておく必要があり、大規模なプログラムでは困難となる為、ADETRAN用のデバッグを使用する。

## 6. ADETRANデバッグ

ADENAシステムでのADETRANデバッグには、ADETRANデバッグを用意している。

### (1) 機能

システムモニタの機能に加え、ADENA内の論

理スレーブ配列へのアクセスと、並列演算実行単位でのブレークポイント設定が可能。

### (2) 構成

UNIX上のデバッガdbxと同環境でデバッグが実行できる構成としている。

### (3) ブレークポイント設定の実現

ブレークポイントの設定は、ハードウェアで用意されているユーザーフラグ機構によって実現している。

ユーザーフラグ機構は、ホスト計算機と全PEとの間に図4のような接続関係があり、ユーザーフラグの信号線はPE内のEPUレジスタの一部に反映されているため、条件分岐命令の条件として用いることができる。

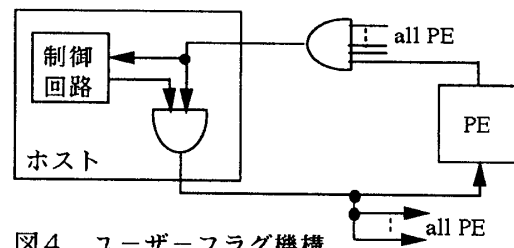


図4 ユーザーフラグ機構

この機構をデバッガ動作時のホスト-ADENA間の制御に用いることにより、ホスト上でデバッグ処理を実行しADENAの再起動を行なう場合に、ADENAを停止状態にしなくてもよい。ADENAを一度停止状態にしてからデバッグ処理を実行し再起動をかける方法に比べて高速化が図られている。

## 7. おわりに

現在、ADENAシステムは試作機が稼働し、各種アプリケーションソフトウェアの開発が行われている。

今回、ADENAシステムのプログラミング環境について述べたが、今後はさらに効率的なプログラミングができる環境の整備を行っていく予定である。

## 謝辞

本研究にあたり、御指導をいただきました京都大学 野木助教授、半導体研究センター超LSIデバイス研究所 間野所長、廉田主任技師に深く感謝致します。

## 参考文献

- (1) 谷川他、「並列計算機ADENA」、情報処理研究会報告(計算機アーキテクチャ)、CPSY88-11、1988年7月。
- (2) H.Kadota and T.Nogi, "VLSI Parallel Computer with Data Transfer Network: ADENA", International Conf. on Parallel Processing, I-317, Aug. 1989.
- (3) 若谷他、「並列計算機ADENAに対する最適化コンパイルの一手法」、情報処理研究会報告(計算機アーキテクチャ)、1989年11月。