

PIM/m のメンテナンス系アーキテクチャと CSP

高橋勝己¹, 武田保孝¹, 村沢靖¹, 小森隆三¹, 杉野栄二²

1: 三菱電機 (株) 2: 北陸先端科学技術大学院大学

1 はじめに

並列推論マシン『PIM/m (Parallel Inference Machine / model M)』は、第五世代コンピュータプロジェクトの一環として開発を行なったものであり、先に開発した並列推論マシンのプロトタイプである『マルチ PSI』[1]の後継機として位置付けられている。従って、PIM/m の構成が 2 次元格子状であることや、各要素プロセッサが CISC タイプであることなどの基本設計はマルチ PSI を継承したものとになっている。しかし、PIM/m では、要素プロセッサの性能向上やプロセッサ数の増大に対応するために、パイプライン・アーキテクチャの採用や Refuge Stack の導入といったいくつかの改良を行なっている [2]。

『CSP (Console System Processor)』は、この PIM/m の立ち上げや、異常発生時の処理、ファームウェア / ソフトウェアのデバッグ支援など、PIM/m の開発や運用を行なうためのシステムである [3, 4]。この CSP は、マルチ PSI の CSP を PIM/m に合わせて改良したものである。

本稿では、PIM/m の CSP における要素プロセッサ内部の資源アクセス方法など、PIM/m の各要素プロセッサのメンテナンスを行なう機能について報告する。

2 PIM/m のハードウェア構成

PIM/m は、推論の高速処理のために専用に開発された推論型の要素プロセッサ (PE) を 2 次元格子状の高速ネットワークで接続した疎結合型並列マシンである。このマシンは 1 筐体 32PE (8 × 4) を単位として最大 8 筐体 256PE (16 × 16) まで、任意の大きさの長方形で、構成することができる。

図 1 は、PIM/m の運用時のシステム構成を示したものである。PIM/m の入出力機能は、SCSI によって接続された PSI-II によって提供され、FEP (Front-End Processor) と呼ばれる。PIM/m では、8PE ごとに SCSI が装備され、FEP だけでなく磁気ディスクなど様々な周辺機器を接続することができる。

PIM/m の立ち上げ、異常処理や PE の内部資源へのアクセスといった、PIM/m のメンテナンス機能は、メンテナンスバスで接続された CSP によって提供される。通常は FEP として用いられる PSI-II の 1 台を CSP として兼用する。

CSP と各 PE とを接続するメンテナンスバスは、CSP

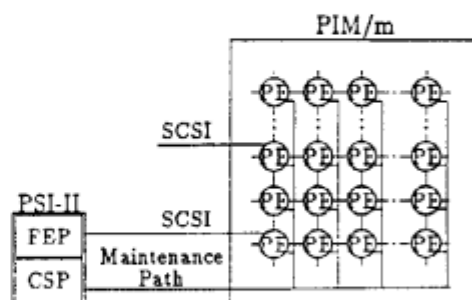


図 1: PIM/m の構成

がマスタ、PE がスレーブといった、シングルマスタ / マルチスレーブの低速バスであり、8bit のデータアドレス共用線と、幾つかの信号線からなっている。

PIM/m のメンテナンス機能はこのバスに対して、CSP がコマンドを送ることによって、実現されている。

3 CSP の主機能

CSP は、各 PE に用意された PE の起動や停止、エラー情報の保持や受渡しを行なうレジスタを用いて、個々の PE を制御するためのものであり、以下のような機能を持つ。

1. PIM/m の立ち上げ
電源 ON/OFF を含むシステムの立ち上げと初期化
2. 異常発生時の処理
異常時のメッセージ表示や、PE 停止の検出・報告と処置
3. プロセッサ構成の変更
PE が故障した場合に、その PE を除いたシステムの再構成を可能にする
4. PE 内部資源 (レジスタなど) の表示 / 変更
5. デバッグ機能
PIM/m での実行レベルは上位から、KL1、KL1-B 命令、マイクロ命令の 3 つがある。KL1 は、ユーザーのプログラムレベルであり、KL1-B 命令とは、それをコンパイラにかけた結果、出力される命令列である。マイクロ命令は KL1-B の各命令を 1 つずつ解釈実行するものである。CSP は、このそれぞれの実行レベルに対して、次のようなデバッグ機能を持つ。
 - (a) マイクロ命令レベル
ステップ実行 / 連続実行や指定マイクロアドレスによるブレイク、トレースなど
 - (b) KL1-B 命令レベル
ステップ実行 / 連続実行や設定条件によるブレイク、トレースなど

Maintenance Architecture and CSP on PIM/m
Katsumi Takahashi¹, Yasutaka Takeda¹, Yasushi Murasawa¹, Ryuzo Komori¹, Eiji Sugino²
1: Mitsubishi Electric Corporation. 2: Japan Advanced Institute of Science and Technology, Hokuriku.

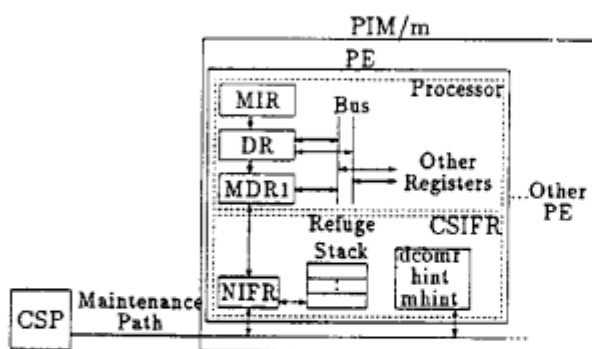


図2: PE内部のメンテナンス系の構成

イク。マイクロプログラムとの連係により実現
(c) KL1 命令レベル
スパイやトレース、インスペクタなど。同様に、
マイクロプログラムとの連係により実現

4 CSPによるPE内部資源へのアクセスとその高速化

4.1 1つのPEに対する内部資源へのアクセス

図2にPE内部のメンテナンス系の構成を示す。ここで、CSPが直接アクセスできるのは、CSIFR (Console System Interface Register) と呼ばれる、PEの起動や停止、エラー情報の保持をおこなうレジスタ群と3つのレジスタ (MIR, DR, MDR1) のみである。これ以外のレジスタへのアクセスは、この3つのレジスタを経由して行なわれる。

CSPがDRを経由してレジスタへの書き込みを行なう場合を例に、レジスタへの書き込み手順について述べる。この場合CSPは、MIRにDRの値を目的のレジスタに書き込むマイクロ命令をセットし、DRに書き込み値をセットした後、1ステップの実行を行なうことで、そのレジスタへの値の書き込みを実現している。この時、CSPがセットするマイクロ命令は、NIFRやMDR1、DRといったレジスタの内容を破壊しながら、MIRへ送られる。このため、PIM/mではその破壊される内容を退避させておくためのスタック、『Refuge Stack』が用意されている。

このRefuge Stackの効果は、構成PE台数の多い場合に発揮されるが、これに関しては次節で述べる。

CSPからあるPEのレジスタの書き込みを行なう手順を以下に示す。

1. PEの指定
2. 破壊されるレジスタ値の退避
3. レジスタに値を書き込むマイクロ命令のセット
4. 書き込む値のセット
5. マイクロ命令の1ステップ実行

6. 破壊されたレジスタ値の復元

CSPから行なうPE内部資源への書き込みは、この手順の2～6を繰り返すことで実現している。CSPから行なうPE内部資源への読み出しも、レジスタ値を読み出すマイクロ命令をセットすることで、同様に行なうことができる。

4.2 複数PEに対する内部資源へのアクセス

PIM/mの立ち上げ時には、パラメータの初期値設定など、複数のPEに対して同じ操作を行なう処理がほとんどである。このため、CSPは複数のPEと同時にアクセスするブロードキャスト機能を持っている。

PIM/mのプロトタイプであるマルチPSIのCSPにも同様の機能は存在したが、マルチPSIではRefuge Stackが用意されていなかったため、PEの内部資源へのアクセスの際に破壊されるレジスタの内容は、CSPがPEごとに読み出しCSP上に保存しなければならなかった。このため、マルチPSIでの複数PEの内部資源への書き込みは、PE数に比例する時間を必要とした。

一方、PIM/mでは、各PEがRefuge Stackを持っているため、破壊されるレジスタの退避/復元といった操作が、各PEごとに独立かつ同時に実行できるようになった。従って複数PEの内部資源への書き込みは、PE数に関係なく一定時間で終了することができる。

このようにRefuge Stackを用いることにより、プロセッサ数の増加による立ち上げ時間の増大を防いでいる。

5 おわりに

本稿では、PIM/mにおけるメンテナンス系アーキテクチャについて、特にCSPによるPE内部資源へのアクセスの高速化に焦点をあてて報告を行なった。この高速化では、各PEにRefuge Stackを持たせることで、PE数に比例しない、一定時間によるPE内部資源への書き込みが行なえるようになった。

また、PIM/mではこれと合わせて、マルチPSIのCSPが行なっていた各種情報の生成を各PEのマイクロプログラムによる生成に切替えることで、立ち上げ時間を大幅に短縮することを可能にした。PIM/mでは、この2つの改良によって、マルチPSI (64PE版) で約10分かかっていた立ち上げ時間が、256PEの場合でも3分程度で終了できるようになっている。

参考文献

- [1] Y. Takeda, H. Nakashima, K. Masuda, T. Chikayama and K. Taki: "A Load Balancing Mechanism for Large Scale Multiprocessor Systems and Its Implementation." Proc. Intl. Conf. on Fifth Generation Computer Systems 1988 (1988)
- [2] 中島、武田、中島: 『PIM/m 要素プロセッサのアーキテクチャ』, JSP'90.
- [3] 中島、田辺、酒: 『マルチPSIにおけるデバッグ機能・メンテナンス機能』, 第29回情報処理学会全国大会論文集 1B-9, 1984
- [4] 杉野、古市、野村、酒: 『マルチPSIにおけるデバッグ機能・メンテナンス機能』, JSP'89.