

PSI-IIシステムのファームウェア

池田 守宏 立野 裕和 山本 明 吉田 裕之 中島 克人 木村 康則 稲村 雄
(三菱電機) (神電気) (ICOT)

1. はじめに

第3世代コンピュータプロジェクトの一環として、現在 ICOTにおいてマルチPSIの要素プロセッサを開発中である。このプロセッサはKL0 処理系ファームウェアを実装し、スタンド・アロン・システムPSI-IIとしても用いられる。本稿ではPSI-IIファームウェアについて、モジュール構成、及び特徴あるいくつかの処理についてその方式を紹介する。

2. モジュール構成

図1にPSI-IIファームウェアのモジュール構成を示す。

2.1 Prolog専用機械命令(PMI)

PMI(Prolog Machine Instruction)は合計約120種の命令から成る。

- (1) 引数処理系命令：レジスタベースの引数処理を行う為の命令。put, get, unity の3系統の命令。
- (2) 制御系命令：述語呼出し用命令。call, execute等通常の述語呼出し命令の他KL0 特有の実行順序制御用の命令、及びESP のメソッド呼出し用の命令等。
- (3) インデキシング系命令：クローズ・インデキシングの為の命令。switch_on_term等。
- (4) カット系命令：普通のprologのカットの他、KL0 で追加されたりモート・カット（レベル番号付カット）、ESP のデモン・コンビネーション カット用のmethod_cut等。

2.2 組込み述語用命令(BLI)

BLI(Builtin Instruction)は合計約190種の命令から成る。

- (1) 一般組込命令：算術、論理演算、データ・タイプチェック、データ生成、データ操作、入出力等。一部の追加された命令を除いて、現在のPSI の仕様に準拠している。
- (2) システム制御用組込命令：プロセス管理、メモリ管理等OSに協力してシステム制御を行うための命令。多重論理空間[2]の制御もこれらの命令で行なわれる。ガベージコレクションもPSI と同様に組込み命令で実現している。

2.3 割込み処理

- (1) 割込み、プロセス切替処理：割込み処理ルーチンは原因を判別し、入出力割込み等の外部割込み、又はトラップの場合はPSI と同様プロセスの切替を行う。
- (2) メモリ割付け処理：グレイページのメカニズム[2]を利用した物理ページの割当てを、割込みチェックのタイミングで行う。
- (3) bind_hook 処理：フックされた変数に値が代入された時点でハードウェア割込みフラグがセットされており、ハンドラ起動可能な命令の切れ目を待って処理する。

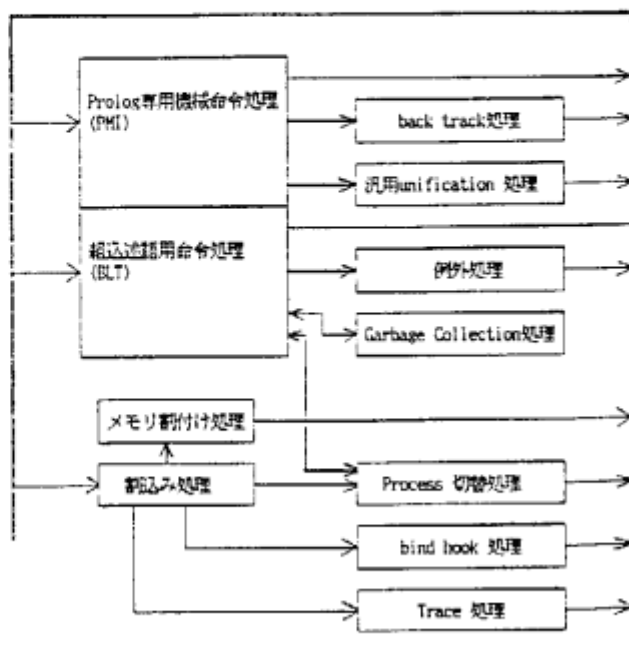


図1. PSI-IIファームウェアのモジュール構成

- (4) トレース処理：エクセプションのメカニズムを利用して、述語呼出し時にトレースハンドラを起動する。これにより、トレース処理プログラムが、被トレースプログラムと、並行して同一プロセス内で実行される。

3. 処理方式

3.1 構造体データのユニフィケーション処理

汎用ユニフィケーション処理の中で、構造体データどうしのユニフィケーションは、要素ごとのユニフィケーション命令(get_value)を要素数分繰返して実行することにより実現している。これによりファームウェア処理ルーチンが簡略化され、また命令の切れ目で統一的な割込みチェックが可能となる。

要素数分の繰返しのためにnext_element命令が設けられている。命令の処理アルゴリズムを図2に示す。NERは繰返し回数を保持するレジスタ。2つの引数レジスタが双方の要素アドレスを示す。get_value.next_elementから成るプログラムは、ファームウェア管理の主記憶上固定領域に置かれ、そのアドレスは固定でユニフィケーションルーチンから直接呼び出される。この種のファームウェア用の機械命令列をD_code(Designated Code)と称している。ユニ

Firmware of PSI-II System

Morihiro Ikeda¹, Hirokazu Tateno¹, Akira Yamamoto², Hiroyuki Yoshida², Katsuto Nakajima³, Yasunori Kimura¹, Yu Inamura¹

1. MITSUBISHI ELECTRIC Corp.

2. OKI ELECTRIC INDUSTRY Corp.

3. ICOT

```

Next_element  Ai,Aj,Lab
NER ← ;
if (NER!=0) { Ai := Ai+1 ;
              Aj := Aj+1 ;
              jump(Lab) ; } /* Loop again
else { LR := LR - 4 ; /* Local Top Pointer
      IAR := (LR) ; /* instruction address reg.
      NER := (LR+1) ; /* next element reg.
      Ai := (LR+2) ;
      Aj := (LR+3) ;
      jump(IAR) ; } /* return

```

図2. next_element命令処理内容

strf	IAR	---	Instruction Address reg.
strf	NER	---	next element reg.
	Ai	---	argument reg.
	Aj	---	argument reg.

図3. structure frame 構成

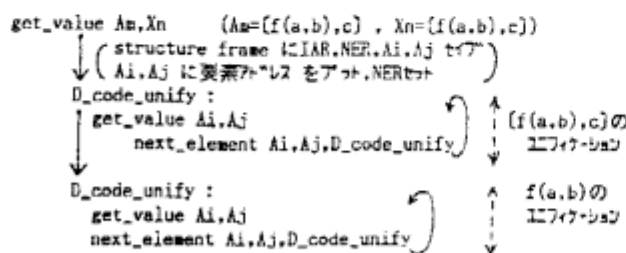


図4. 構造体どうしのユニフィケーション処理 例

ユニフィケーション用D_codeの呼び出し時、戻り用情報は、ローカルスタック上にstructure frame としてセーブされる。内容を図3に示す。図4に処理の例を示す。尚、現在の汎用ユニフィケーションルーチンでは、構造体要素がさらに構造体だった場合以後に、このD_codeを使用している。簡単な構造のデータは、ファームウェア内で処理し高速化をはかっている。

3.2. トレース処理

オペレーティングシステムがトレース機能を実現する為、ファームウェアでは以下のトレース処理機能を持つ。

(1) トレース・エクセプション

トレース処理のプログラムは、エクセプションのメカニズムを利用し、トレースエクセプションのハンドラとして記述する。従来のPSI ではトレースは、トレース・トラップを発生して、トラップ対応プロセスにプロセス切替を起こし、被トレース・プロセスとは別プロセスで動いていた。しかしこの場合プログラムのコンテキストが切替ってしまう為、トレーサ側から被トレース対象の述語の引数を見る場合、処理系の内部にも立入った複雑な処理をする必要があった。PSI-IIではトレース処理プログラムは被トレースプロセス内で、エクセプションハンドラとして動く為、引数はそのままトレーサに受渡し可能で、バックトラック等の実行制御も容易に行なえる。

(2) トレース・タイプ

トレース処理ルーチンでは、次に実行する機械命令の種類を判断し、エクセプションを起こすか否かを判断する。述語呼出し系の命令は11種に分類され(call, execute系命

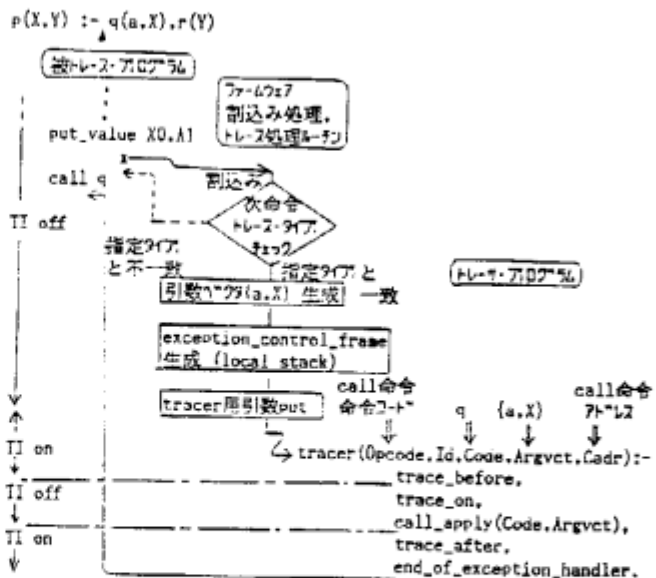


図5. トレース処理の流れ

令,method_call系命令等)、そのうちの種類のエクセプションを起こすかは予めプログラムで、トレースタイプを指定し選択できる。これによりESP のメソッド呼び出しのみのトレース、全ての述語呼び出しのトレース等、任意のレベルのトレース処理が可能となる。

(3) 実行制御

トレース実行中のプロセスは、被トレースプログラムとトレーサプログラムが並行して走る。この2つの状態を切分けるためにTI(Trace Inactive)のフラグを使用する。トレーサが走行中はTIをonにしてトレースエクセプションの発生を抑止する。図5にトレース処理の流れを模式的に示す。TI=offの場合述語呼出し(call)命令の直前でトレース処理ルーチンが走りトレーサをTI=on にして起動する。トレーサへの引数は、命令コード、エクセプションID、述語の引数をベクタに格納した引数ベクタ、呼出し元のアドレスがセットされる。またトレーサからのリターン用に必要な環境情報を、エクセプション・コントロール・フレームとしてローカル・スタック上にセーブする。トレーサの中で必要な処理をした後、被トレース述語を実行させる時は、TI=offに戻し、call_apply命令を使用する。

4. おわりに

本研究にあたり多くの貴重な助言をいただいたICOT及び関連メーカーの方々、またコーディング、テストに御協力いただいたエスイーティ(株)石川氏、川田氏、(株)フジシステム 佐藤氏、(株)エー・ディ・エス 吉田氏、に深く感謝する。

参考文献

- (1) : 中島ほか, マルチPSI 要素プロセッサPSI-IIのアーキテクチャ 第33回情報処理全国大会 1986
- (2) : 吉田ほか, マルチPSI 要素プロセッサPSI-IIのメモリ管理とプロセス管理 第33回情報処理全国大会 1986