

PIM用のKL1 処理系と その開発支援環境

ICOT

第1研究室

今 井 明

並列推論マシンPIM

共有メモリによる密結合 + 専用高速ネットワークによる結合

並列論理型言語 **KL1** を並列実行

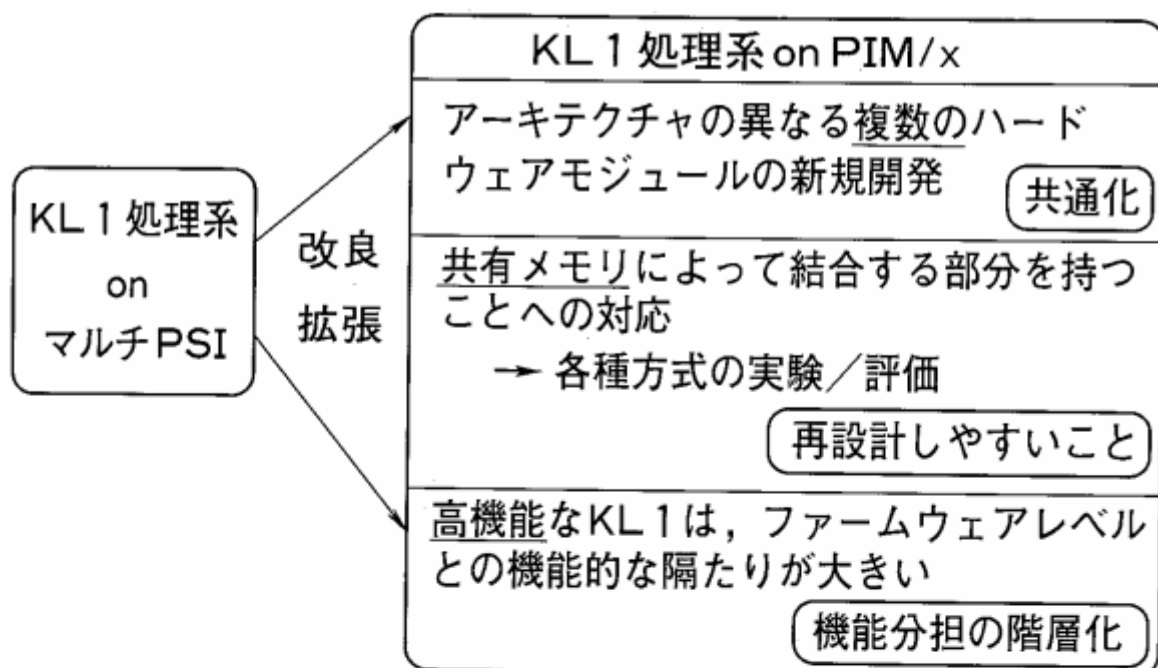


PIM上のオペレーティングシステムPIMOSを記述できる

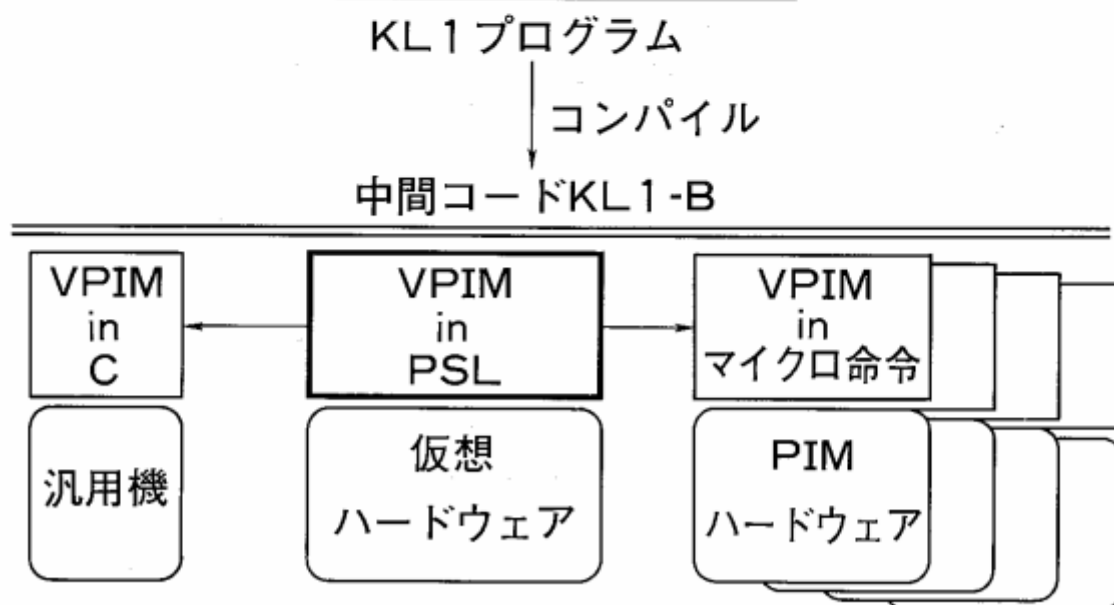
KL1 言語処理系で実現すべき項目

メタ管理機能
メモリ管理機能

PIMのKL1 言語処理系の開発方針



KL1 処理系の概要



処理系記述言語PSLで、PIMモジュールに共通のKL1言語処理系仕様VPIMを記述

処理系記述言語PSL (PIM System descriptive Language)

設計の目的

仕様書

PIM上のKL1処理系の実現方式の仕様書

実機ファームウェアのソース

仕様書のコンパイルにより，PIM実機での動作

——→ マクロ展開言語

シミュレータのソース

既存の言語に変換することで，シミュレータ(PIM/s)として動作

——→ 仕様の検証／評価

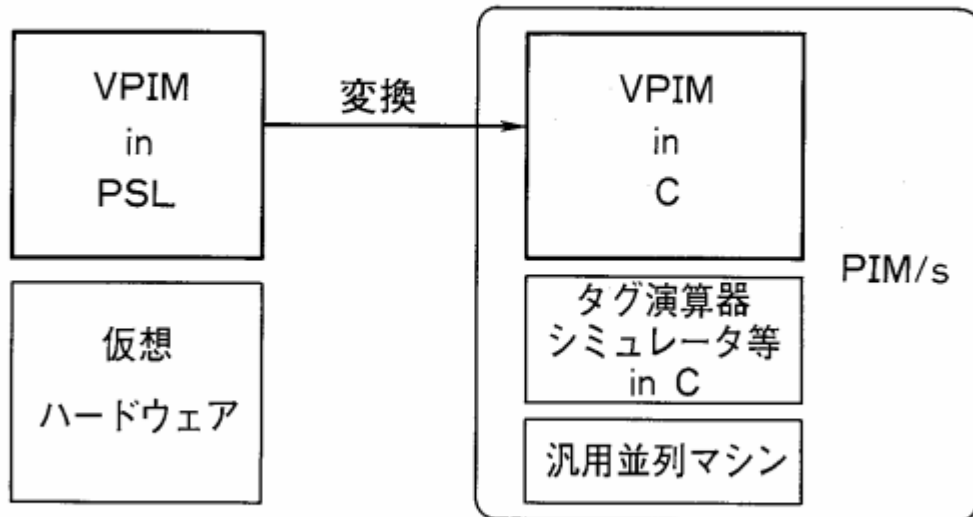
仮想KL1処理系VPIM (Virtual PIM)

PSLにより記述
徹底した階層構成

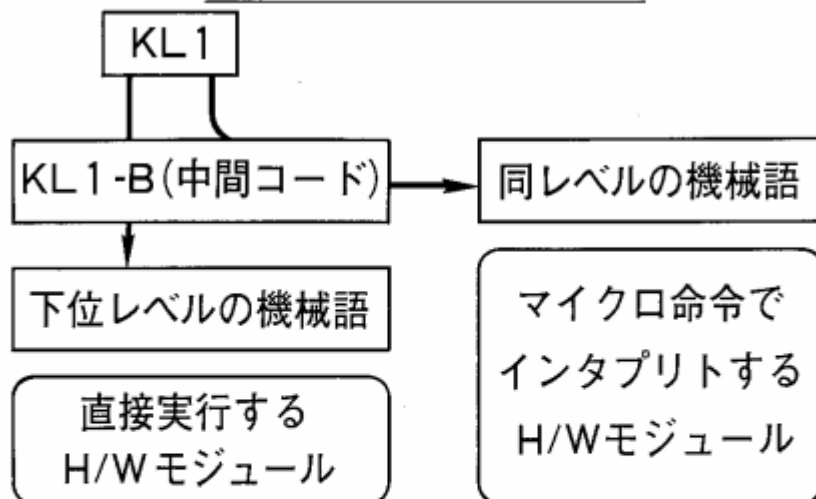
抽象機械語解釈レベル KL1-B解釈のトップレベル	↑上位 ↓下位
処理モジュールレベル イベント／クラスタ間メッセージ処理等	
構造モジュールレベル 構造体の定義／操作，GC等	
基本モジュールレベル 排他制御の基本処理，メモリマップ等	
操作プリミティブレベル(基本命令) メモリ操作，レジスタ間演算	

VPIMシミュレータPIM/s (PIM Simulator)

目的 VPIM仕様のテスト／デバッグ／評価を，
ハードウェア開発と独立に行なう



KL1コンパイラ



従来のコンパイル方式の

- ✕ ハードウェアレジスタの使用効率が悪い
- ✕ コンテキストスイッチの処理が重い

→ 改良したコンパイル技法(メモリベース方式)を開発

PSLコンパイラ

PSLによるVPIM記述を実機のファームウェアにコンパイルする

PSLの基本命令と実機の $\left\{ \begin{array}{c} \text{機械語} \\ \text{マイクロ命令} \end{array} \right\}$ 対応表を取り替えて、複数のハードウェアモジュールに対応

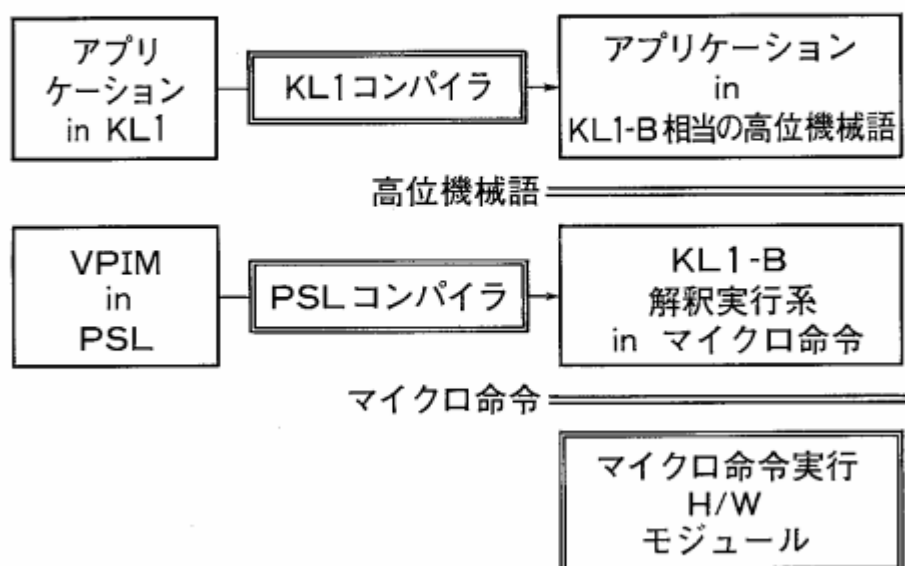
PSLエキスパンダ

PSLで記述したマクロを展開

→ 展開後のプログラムは、PSLコンパイラの入力になる

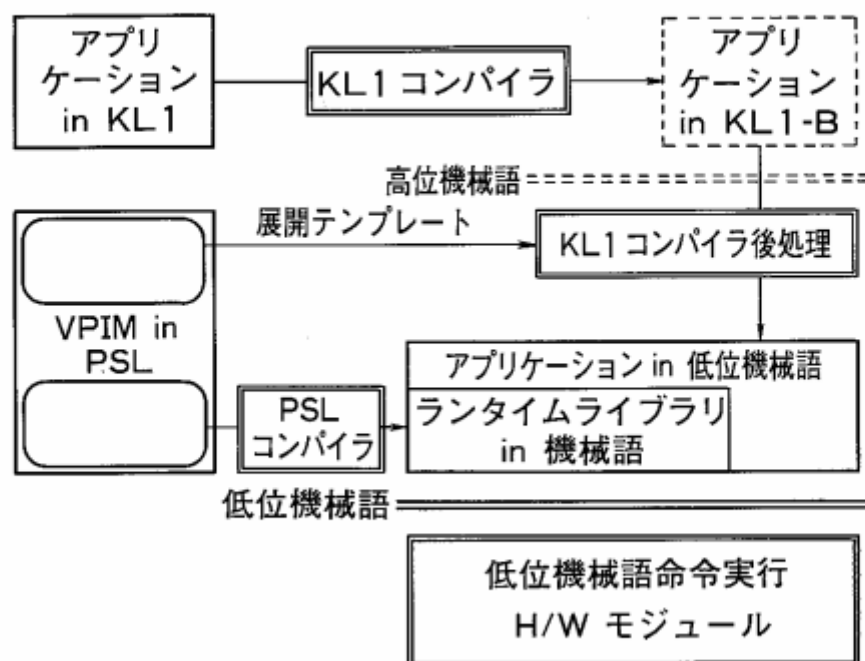
実機処理系の全体構成（1）

中間コードインタプリト方式



実機処理系の全体構成（2）

中間コード展開方式



開発支援環境

相互参照

PSLで記述したプログラムの定義参照情報データベースを作り，マクロ定義の検索や定義と参照間の整合性チェックを行なう

ユーザインタフェースの統合

各種開発支援ツールをエディタから統合的に呼び出せる

開発の現状

VPIM

複数クラスタのプロトタイプが，PIM/sとして稼働中

KL1 コンパイラ

改良方式のコンパイラをKL1にて記述した

→ 実機上のセルフコンパイラとして使える

PSL コンパイラ

プロトタイプ版は完成

コード生成の最適化を施した版を開発中

ま と め

- PIMのKL1処理系開発は，仮想マシンのKL1処理系VPIMを処理系記述言語PSLにて記述している
- KL1は，KL1コンパイラで中間コードもしくは，機械語命令に変換後実行する
- PSLコンパイラでVPIMをPIM実機の命令に変換する
- VPIMのテスト／評価は，汎用機上のシミュレータで行なう
- 開発支援環境を充実させている