

# システムLSIソフトウェアの開発環境

## Total Software Environment for System LSI Development

### あらまし

ネットワーク社会の高度化によるシステムの多様化ならびにメディア処理の高速化の要求などに伴って、ハード/ソフトともにシステムLSIには従来の開発スタイルから飛躍的に向上した高設計品質と短TATを同時に満足する開発環境が要求されている。

とりわけ数100万ステップを超える組込みソフトウェア開発の分野では開発環境の良否がシステムLSIを使用したシステム全体の開発ならびにビジネスへ甚大な影響を与えるようになりつつある。

本稿では、組込みソフト開発で今後必須となるUML言語を中心とした上流CASEツールと従来からの開発環境であるSoftuneとの連携、基盤技術の要となるコンパイラ技術、組込みJava技術および現在急激に増大しているメディア、プロファイルに対応したミドルウェアライブラリについて紹介する。

### Abstract

To meet the demand for faster media processing on today's diverse, networked systems, an environment for quickly designing high-quality hardware, software and system LSIs is needed. In particular, when developing embedded software containing millions of steps for system LSIs, the quality of the development environment has a significant effect on the product. This report describes the linkage between the high-level CASE tools based on the UML language that will be essential for the future development of embedded software and Softune, which is a conventional development environment. It also describes the compiler technology required for software platform development, embedded Java technology, and middleware libraries capable of supporting the rapidly increasing media and profiles.



五十嵐 純  
(いがらし じゅん)

システムLSIソフトウェア部基盤ソフトウェア開発部 所属  
現在、マイコン開発支援システムの企画、開発に従事。



松浪邦彦  
(まつなみ くにひこ)

システムLSIソフトウェア部基盤ソフトウェア開発部 所属  
現在、マイコン用コンパイラの開発に従事。



根岸康之  
(ねぎし やすゆき)

システムLSIソフトウェア部ミドルウェア開発部 所属  
現在、組込みJavaの開発に従事。



後藤 純  
(ごとう きよし)

システムLSIソフトウェア部ミドルウェア開発部 所属  
現在、マイコン用ミドルウェアの企画、開発に従事。

## ま え が き

組込みソフトウェア開発では年々増大するプログラム規模に加えて開発短TAT ( Turn Around Time ) の要求からハード部分のソフト化が発生している。ソフトが大規模化するにつれてソフト障害によるシステムダウン、商品の回収などの問題が生じておりその品質が厳しく問われるようになってきている。このように数100万ステップを超える組込みソフトウェア開発の分野では、開発環境の良否がシステムLSIを使用したシステム全体の開発ならびにビジネスへ甚大な影響を与えている。この問題に対処するには従来からの開発環境に加えて以下に述べる上流開発ツールとの連携が今後重要となる。またメディア処理高速化の要求により並列処理プロセッサの採用が必須となるが、要となるのはコンパイラ技術である。

本稿では、上記の連携、技術に加え富士通の組込みJava技術とミドルウェアライブラリについて紹介する。

## 上流CASEツールと統合開発環境との連携

富士通では、CやC++で組込みソフトウェアの開発を行えるよう十分な性能と操作性を備えた統合開発環境Softuneを提供している<sup>①</sup>。本開発環境だけでも十分プログラム開発は可能であるが、顧客のアプリケーションの更なる大規模化、高機能化ならびに開発期間短縮、高品質化などの要求を満足するためには、今後上流設計にCASEツールの導入が必須となる。まず商品を企画する際の各部門間での製品仕様やプログラム仕様に関する共通言語が必要となる。この段階で最近注目されている言語としてUML ( Unified Modeling Language ) があり、オブジェクト指向的に要求事項を分析しながら状態遷移図や状態遷移表などで仕様設計を行っていくことができる。これらのCASEツールで作成した仕様書が上流工程の技術者から下流工程の技術者へ情報伝達が行われ、最終的には富士通のマイクロコンピュータ上で動作するプログラムにまでブレイクダウンされる。この過程を設計者がUMLや状態図・表を見ながら手作業で行うことももちろん可能であるが、より正確に短時間で行うためには、CASEツールの仕様書データから直接SoftuneのC/C++ソースへの自動変換が求められる。

富士通ではこの要求に応えるため、上・中・下流の3層でCASEツールに連携させた開発プロセスを提供している。

さらにこれらのツール連携ソリューションだけにとどまらず、顧客へのコンサルテーションを含めたモデルを提案して顧客および富士通が取り組むべき領域を明確にし、その解答となる手法やCASEツール、コンサルティングなどを提示している。以下に具体的に自動車のボディ電装部品など主に状態の遷移を制御することを主とする機器の開発向けのCASEツールとSoftuneの連携を例に説明する。

連携の一例として主に自動車ボディ機器部品の開発によく使用されているStatemateMAGNUM ( I-Logix社 )、ZIPC ( キャッツ社 )、Rhapsody in MicroC : RiMC ( I-Logix社 ) の連携を以下に示す。

図-1は上・中・下流の3層と各CASEツールの位置付けを表している。最上流では、状態遷移図のCASEツールであるStatemateMAGNUMが使われており主に自動車メーカーが部品メーカーへの発注の際の仕様書の役割として使用されている。受注した部品メーカーはこの仕様書をもとに富士通マイクロコントローラを使用した組込み製品を開発するためにそのプログラムの仕様を設計しなければならない。上流のStatemateMAGNUMで作成された状態遷移図には、組込み用マイクロコントローラの割り込みやポートアドレスなどの情報は記述されていない。むしろ抽象度を高くして記載してあるために担当者同士の情報交換がやり易いという利点がある。この上流の仕様書データは、一旦、中流に位置付けされたCASEツールの仕様書データに変換される。富士通では顧客に選択幅を持たせる意味から中流フェーズに二つのCASEツールを用意した。一つは状態遷移図のツールであるRhapsody in MicroCであり上流のStatemateMAGNUMのデータをそのままインポートできる。もう一つは状態遷移表のCASEツールであるZIPCであり、上流のRhapsody in MicroCのデータをZIPCのデータにコンバータで変換する。

RiMCとZIPCの大きな違いは状態遷移の表し方が図か表かという点である。両者とも出来上がった仕様書からSoftune用のCソースを自動生成できる。

顧客はこれら中流CASEツールを使用して、マイクロコントローラ特有の割り込み時の処理やタイマ起動処理の情報、ポートアドレス情報や初期化情報などを仕様書に追加する形で盛り込み、組込みプログラムの設計を行う。これらのCASEツールは状態遷移の論理が正しく記載されているか確認を行うシミュレーション機能を備えており、シミュレーションによる論理の確認が終わればC

## 二つのパス(RiMC or ZIPC経由)での連携

富士通の提案するSoftune-CASE連携開発環境

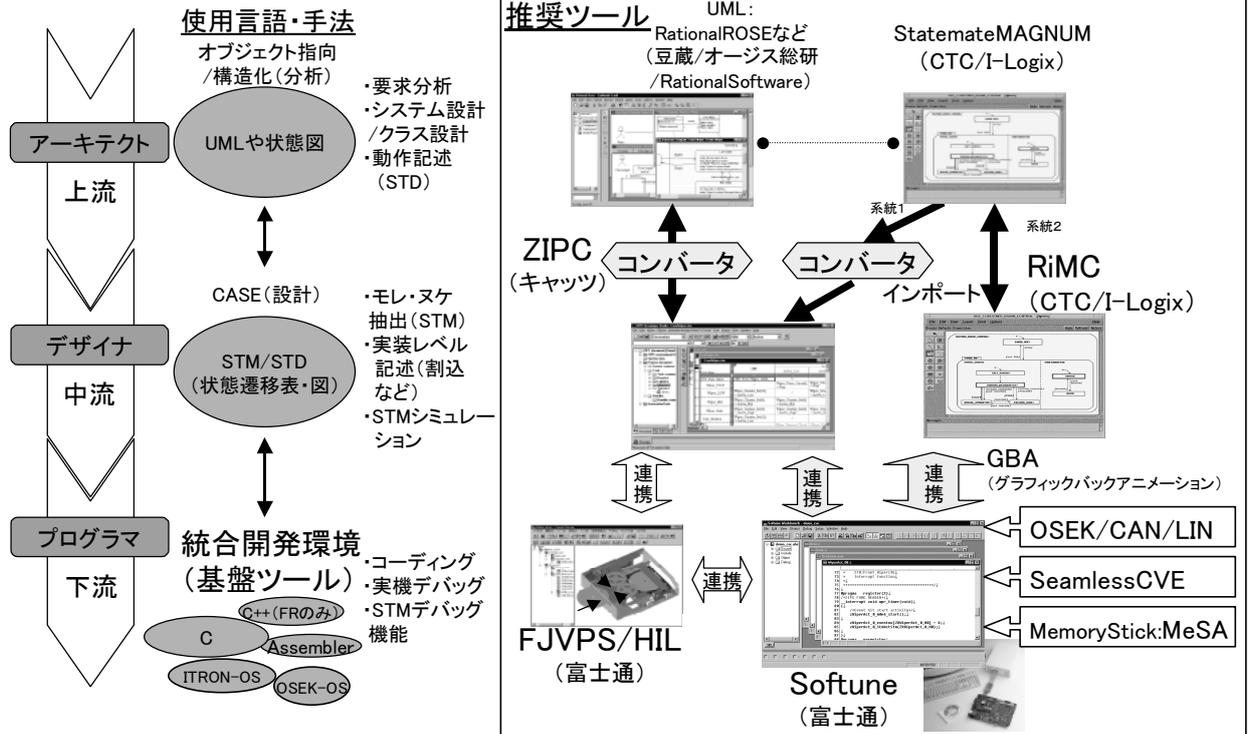


図-1 上・中・下流の3層と各CASEツールの位置付け  
Fig.1-Three types of CASE layer and example of the link of tools.

ソース自動生成機能によりSoftune用のCソースプログラムが生成される。

最下流のSoftuneでは中流CASEツールで生成されたCソースをコンパイル、リンクして実行形式のプログラムを作成しICE (インサーキットエミュレータ) を使って動作検証を行う。ここで注目すべき点は、SoftuneでのCプログラムの実行動作確認の際に中流CASEツールとが連動して、現在どの状態であるかを中流CASEツール上の図や表がハイライト表示してアニメーション動作する点である。これにより顧客はCソースではなく自分が書いた仕様書を見ながらプログラムの検証デバッグを行うことができる。

### UMLツールとの連携の例

現在、UMLで記述した仕様書のうち状態図の部分は、SoftuneのCプログラムまで実際に変換ができる。Rational ROSE (ラショナルソフトウェア社) で書いた状態図をキヤッツ社から提供されているコンバータによりZIPCの状態遷移表に変換することができる。

### CASE連携ソリューションの更なる充実：トータルソリューションの提供

顧客が組込み機器のソフトウェアを開発する際のプロ

セスと富士通が提供すべきソリューションの関係を図-2に示す。顧客が組込み製品のプログラムを開発する場合は、要求仕様定義、システム詳細仕様、プログラム設計、ソフトウェア実装、というV字型の左の設計フェーズのラインを経由してソフトウェアに実装する。

その後、モジュール統合テスト、システム統合テスト、システム受入テスト、というV字型の右の検証フェーズのラインを経由して製品の完成となる。この検証フェーズでは設計時に使用した設計書に関連づけられて検証できることが望ましい。

また、システムLSIを採用した場合には、システムLSIの設計のフェーズとして、アーキ選定、仕様記述、RTL論理合成、チップ/実機ボード評価・検証というフェーズが存在することになる。このようにマイクロコントローラを使う顧客は図のようなW型V字プロセスに沿って開発を行うことになる。前述のStateMATE MAGNUM や UML , ZIPC , RiMC や , Softuneなどはそれぞれ図で示した位置に存在するツールである。

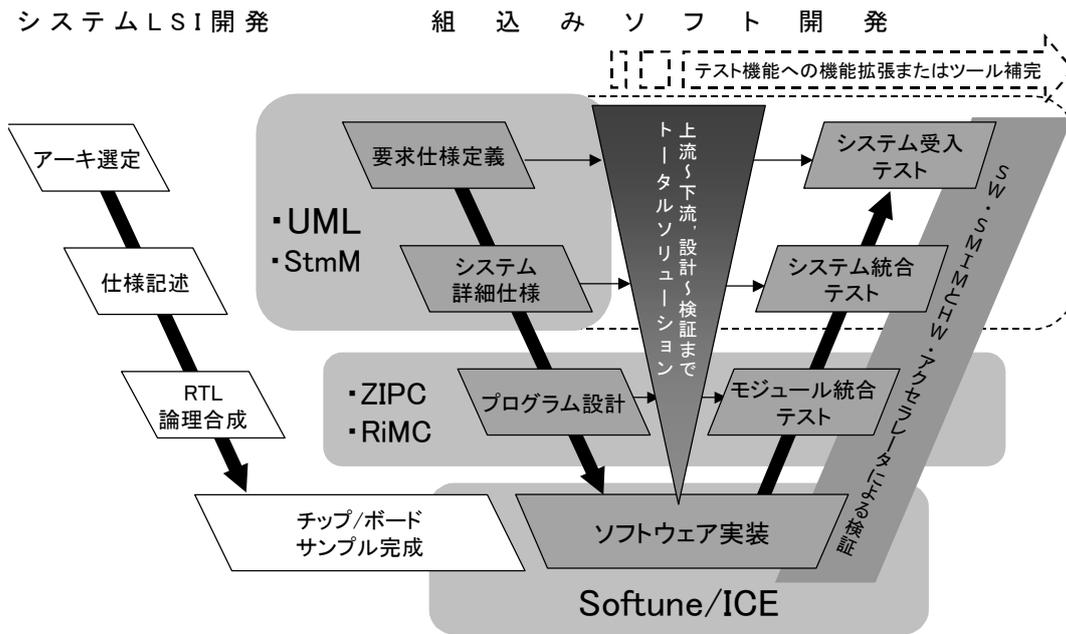


図-2 組み込み開発型VプロセスとCASE連携のドメインの関係  
Fig.2-The V model solution for embedded system development.

## CASE導入の問題点と富士通が提供するソリューション

顧客はこれらのツールを使って上流から下流まで設計し、検証を行うが、実際にはツールを導入しただけでは開発は進まない。CASEツールの場合、ツールの操作性よりもそのCASEの手法そのものの導入や運用がうまくいくかどうか導入の成否に大きく影響する。すなわちCASEツールを使用する顧客に対しては、マイクロコントローラやツールの提供だけでなく、このV字型モデルの上流～下流、設計～検証までを遂行する際に発生する問題に対して適宜対応できるソリューションを持っていることが望まれる。

CASEの導入の際に問題となる点は主に、(1) どのCASEツールがよいか分からない、(2) 手法のマスタが難しい、(3) 開発人員の不足、の3点がある。これらの問題を抱えた顧客に対応できるよう、各CASEツールや各手法に関するエキスパートを備えたツールメーカーやコンサルティング会社と協調することが重要である。著者は各CASEツールメーカーやコンサルティング会社との協調体制を構築し、CASEを導入する際の顧客の問題に対応できるよう努力している。

さらに将来的には先に述べた、各個別V字型モデルの領域にとどまらず、これらをネットワークで統合した拡張システムを記述する必要性が生じてくるであろう。そのような場合をあらかじめ想定して富士通ではUMLを

ベースとしたオブジェクト指向の開発プロセスを無理なく、これらの拡張システムに構築できる体系に展開可能なように今後のツール連携・推進をしていく。

## コンパイラ技術

### FR500用コンパイラの最適化技術

マルチメディア処理が要求されるデジタル民生製品では、従来のRISCプロセッサの10倍以上の性能が必要とされる。富士通では、このようなデジタル民生機器をターゲットにした組み込み用VLIWプロセッサFR500を開発するとともに、このハードウェアの機能/性能を効率よく利用することが可能なコンパイラ“FR500 Softune C/C++ Compiler”を開発した<sup>(2)</sup>。

### FR500の性能を引き出す最適化

FR500はVLIWプロセッサであり、同時に四つの演算を行うことができる。また、条件付で実行できるPredicated命令、例外の発生を保留できるNE (Non-Excepting) 命令、メディア処理に最適な各種メディア命令および単精度浮動小数点演算命令を持つ。メディア命令および浮動小数点演算命令にはデータ処理を効率良く行うためのSIMD命令がある<sup>(2)</sup>。FR500 Softune C/C++ CompilerはこれらFR500の特徴を利用し、性能を引き出すために以下に述べる最適化をサポートしている。

- ・条件付実行制御 (Conditional Execution Control)

- ・投機的実行制御 (Speculative Execution Control)
- ・命令スケジューリング (Instruction Scheduling)
- ・トレーススケジューリング (Trace Scheduling)
- ・クロスファイル最適化 (Cross File Optimization)
- ・ベクトル化 (Vectorization)

## (1) 条件付実行制御 (Conditional Execution Control)

FR500は同時に四つの演算を行うことが可能であるが、実際のプログラムには多くの分岐命令が含まれているため、同時実行可能な命令を見つけ出すことは難しい (図-3左)。Predicated命令は分岐を命令の実行条件として表現し、プログラム中から分岐を排除することができ

る (図-3右)。

このような最適化を条件付実行制御といい、この最適化により、プログラム中から分岐をなくし、より高いILP (命令並列度) を実現し、効率よくVLIWの各命令スロットを使用することができる。

## (2) 投機的実行制御 (Speculative Execution Control)

プログラムがメモリから値をロードする際に、キャッシュミスによる遅延はプログラムの性能を劣化させるため、メモリからの値のロードとその値を使った演算はできるだけ離れているべきである。実際のプログラム中には多くの分岐命令があるため、このような命令の引き離

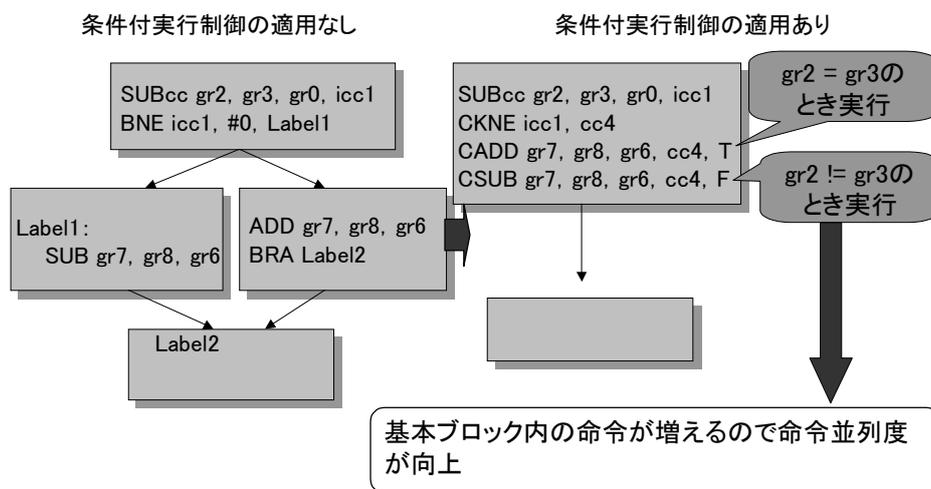


図-3 条件付実行制御の例

Fig.3-Flow of conditional execution control.

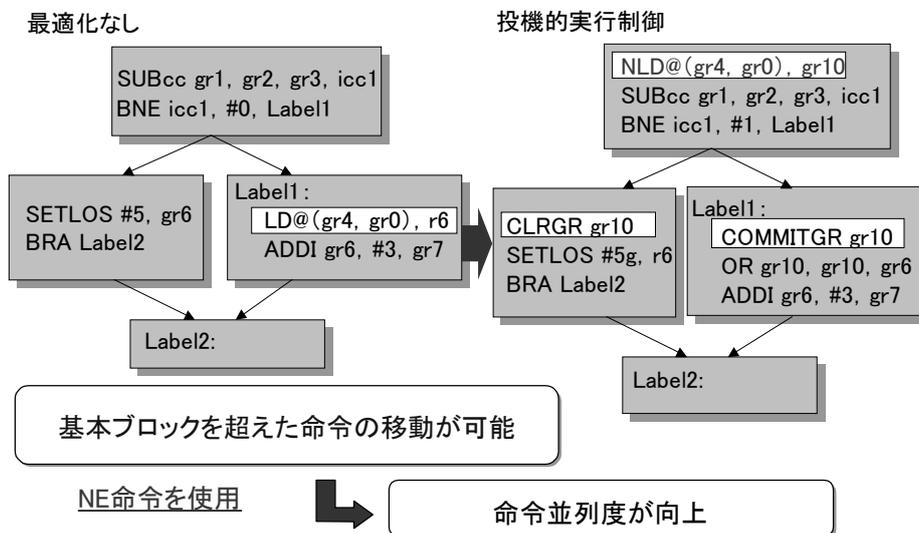


図-4 投機的実行制御の例

Fig.4-Flow of speculative execution control.



ことができる

## (6) ベクトル化 (Vectorization)

FR500は、メディア処理に最適な各種メディア命令および単精度浮動小数点演算命令をもつ。メディア命令および浮動小数点命令演算にはデータ処理を効率良く行うためのSIMD (Single Instruction stream/Multiple Data stream) 命令がある。コンパイラはこれらメディアSIMD命令、および単精度浮動小数点演算SIMD命令をそれぞれベクトル長4・ベクトル長2のベクトル命令として出力する (図-6)。

### 評価結果

画像処理プログラムであるJPEGデコードプログラムを利用してこれらの最適化の効果を検証した (図-7)。従来の最適化の性能をスカラ演算性能と位置付け、命令スケジューリング、投機的実行制御、トレーススケジューリングの各最適化による実行性能の向上をFR-Vデザインキットを使用して実測した。

すべての最適化を実施した性能はスカラ性能に比べて1.74倍である。またそのときのILP (命令並列度) は1.70となっている。JPEGは主に整数演算で構成されているが、FR500で使用できる整数演算用のスロットは二つのみであり、ILPが1.70ということは使用可能なスロットの85%を使用していることを意味しており、非常に効率のよいオブジェクトが生成されている。

このように、FR500 Softune C/C++ Compilerを使用することにより、VLIWプロセッサであるFR500のアーキテクチャを意識することなく、C/C++言語で記述された既存のアプリケーションプログラムを効率よく動作させることができる。このため、従来では性能面で難しかった組み込み分野でのメディア処理のソフトウェア化を実現することが可能となった。

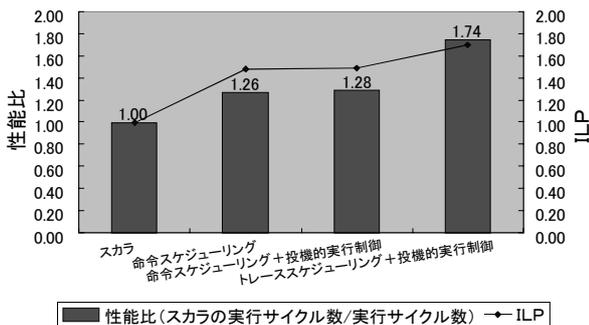


図-7 性能比とILPの推移

Fig.7-Effect of performance improvement with each scheduling method.

## 組み込みJava技術

富士通では、組み込み分野向けOSとしてμITRON仕様リアルタイムOS: REALOSシリーズを8ビットマイコンから32ビットマイコンまで提供している。インターネットの普及により組み込み分野向けOSにおいても、ネットワークのサポートが必要となってきている。さらにネットワーク環境でのコンテンツ開発や配信でのJavaの普及により組み込み分野でのJava実行環境が重要になってきた。富士通ではRISCマイコンSPARC LiteにJava実行環境として、インタプリタ方式とJITコンパイラ方式を開発した<sup>(3)</sup>。しかし、これらを評価した結果、ソフトウェアによるJava実行環境は組み込み分野で要求される実行速度とメモリ使用量との両立をさせることは困難と判断した。これらの背景を図-8に示す。

### Javaチップによる問題解決

このように既存のマイコンでは、Javaバイトコードとマイコンの機械命令との対応が1:nとなり、Javaの実行は重い処理である。そこでJavaバイトコードを機械命令とするJavaチップにより組み込み分野で要求される実行速度向上とメモリ使用量削減を図った<sup>(4)-(6)</sup>。

### Javaチップと関連製品

JavaチップはpicoJava-IIコアをベースにし、周辺回路を組み込んだ。

picoJava-IIの命令セットは、Javaバイトコードに対しスーパーセットであり、Javaだけでなくアセンブリ言語、C/C++言語にも対応できる。

Javaチップの基本ソフトウェアとして、リアルタイムOSとその上にJava実行環境として組み込み向けの

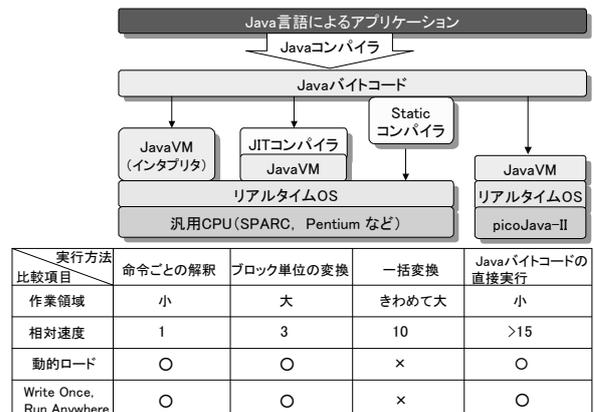


図-8 Javaチップの優位性

Fig.8-Java chip superiority table among others for Java applications.

PersonalJavaを $\mu$ ITRON上を実装した。PersonalJavaはPDAなど向けにGUIをサポートしている。このためグラフィックライブラリを開発し、PersonalJavaのwindow systemであるTruffleをpicoJavaリファレンスボード(図-9)用に変更した<sup>(4)</sup>。

なお、PersonalJavaにはJavaインタプリタが組込まれているがJavaチップはJavaバイトコードを直接実行するため、ソフトによるインタプリタをスキップするよう改造した<sup>(5),(6)</sup>。JavaVMはインタプリタのほかにクラスローダ、スレッド管理、バイトコード検証、JNI(Javaネイティブインタフェース)、ガーベジコレクタの機能を含むが、これらについてはJavaチップでもソフトウェアでの対応が必要なため開発した。このように

Java実行環境はパソコンのようにGUIやネットワークをサポートしたものになり、個々のドライバやソフトを寄せ集めてのJava実行環境構築は煩雑になる。このため $\mu$ ITRONと各種ドライバ、PersonalJavaをパッケージ化し、図-10のようにJTRONサポートパッケージ：J-REALOS/PJとして提供している<sup>(7)</sup>。

Javaチップによる効果

Javaチップ上にリアルタイムOSとJavaVMを実装することにより、ベンチマークテストでのJava実行性能はパソコン上のPersonalJavaエミュレーション環境に比べ同一周波数換算で約15倍高速であった。また、C実行性能はDhrystone2.1 0.94 MIPS/MHzで、JavaチップはJavaとCの実行を両立させている。また、ブラウザやゲームなどのJavaアプリケーションでの評価ではチップの推奨動作周波数において十分な応答性、快適さで実行できた。JavaチップはJava実行速度と実行に必要なメモリの削減を解決し、資源の限られた組込み機器においてもJavaが実用可能なことを示した。さらに、C/C++コンパイラの利用により、既存Cソースの流用が可能になり、アプリケーションやドライバの開発工数を削減できる。また、Javaのアプリケーションは、Javaの特徴である“Write Once, Run Anywhere”により、特定の開発ホストに制限されずに開発でき、作成したアプリケーションは実機がなくてもJava実行環境があれば動作を確認できるので、従来からのクロス開発ツールの入手や供給による制限がなくなる。さらに、従来のプラットフォームにクローズした開発に比べ、APIの標準化により、オープンな開発が可能になり、JavaはP

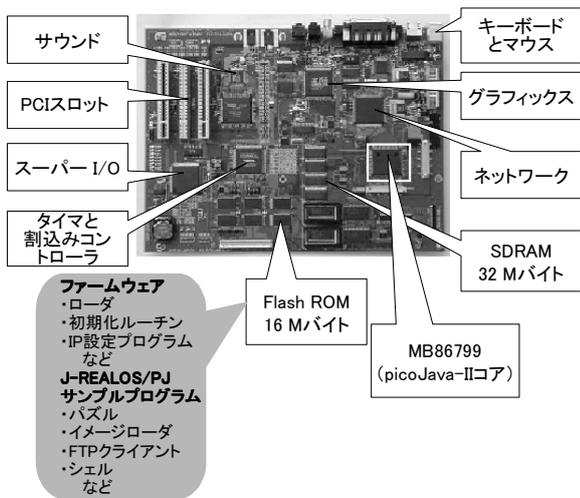


図-9 リファレンスボードの概要  
Fig.9-Photo and diagram of pico Java evaluation board.

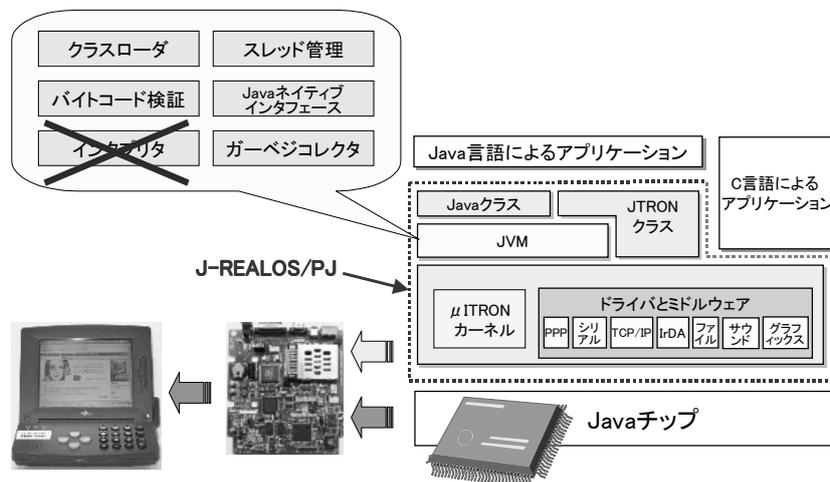


図-10 J-REALOS/PJ  
Fig.10-Configuration of J-REALOS/PJ.

ラットフォーム共通という特徴からほかのプラットフォーム向けに開発したアプリケーションの利用も可能になる。

## 組込みJavaの課題と今後の方向

Javaが普及するための条件でもあるネットワークインフラはブロードバンドとIPv6の普及により整いつつある。組込み分野でのJavaは携帯電話や一部の家電機器や制御機器に搭載されてきたが、自動販売機などにも搭載され新たなサービスが可能になった。また、従来のリアルタイム処理プログラムはアセンブリ言語、C/C++言語で記述されてきたが、Javaのリアルタイム処理対応も仕様が策定されリファレンスとなる実装が行われている。組込み向けのPersonalJava/EmbeddedJavaの後継であるJava2 ME CDC/CLDCでは、分野ごとのProfileを導入しAPIを専用にしサイズ縮小を図っている。組込み分野向けのJava開発が進み、Javaチップの利用は様々な組込み分野で本格化しつつある。富士通は、これらの組込みJava分野に対してコストパフォーマンスの高いJava実行環境を提供していく。

## ミドルウェアライブラリ

近年、情報家電や様々な組込み機器において静止画・動画・オーディオなどのメディア処理が必要となっている。富士通ではこれらのメディア処理ソフトウェアをミドルウェアとして提供することにより、組込みシステムの構築をサポートしている。メディア処理向け命令を強化したVLIW型プロセッサFR-Vとミドルウェアを組み合わせることで、高性能なシステムをフレキシブルに構築できるようになる。

提供するメディア処理系ミドルウェアには次のものがある。

### (1) 静止画

- ・JPEGエンコード/デコード
- ・JPEG2000エンコード/デコード

### (2) 動画

- ・MPEG1ビデオデコード
- ・MPEG4ビデオ (Simple Profile LV3) エンコード/デコード

### (3) オーディオ

- ・MPEG-1 オーディオデコード (Layer2)
- ・MPEG-2AACデコード
- ・MPEG-4AACデコード
- ・MP3デコード

### ・AC-3デコード

メディア処理系以外にも、GUI・描画処理の構築を支援する2次元グラフィックス・3次元グラフィックス、インターネット接続を可能にするTCP/IP、ブラウザ・セキュリティ機構、Java、機器間の短距離無線通信を可能とするBluetoothなどの開発も進めている。富士通はこれらミドルウェアを開発/提供することにより、情報家電や様々な組込みシステムの構築を支援していく。

## む す び

富士通はマイクロプロセッサを幅広く取り揃えており、それに対するソフトウェア開発環境として、使いやすく高機能な統合開発環境Softuneを提供している。この環境をベースに基盤技術の要となるコンパイラ技術の開発はもちろんのこと今後必須となる上流ツールとの連携の充実を図っている。

また組込み分野でのJava実行環境のサポート、各種ミドルウェアライブラリの開発に今後も積極的に取り組んでいく。

## 参考文献

- (1) 五十嵐純ほか：システムLSIソフトウェアの開発環境と実行環境．*FUJITSU*，Vol.49，No.2，p.135-140（1998）．
- (2) 高橋宏政：システムLSI用VLIWプロセッサコア．*FUJITSU*，Vol.52，No.4，p.368-373（2001）．
- (3) 世界で初めてJavaチップを核とした組み込み用途向け「Javaソリューション」を提供開始。  
<http://pr.fujitsu.com/jp/news/2000/04/5.html>
- (4) 根岸康之ほか：Javaチップ搭載リファレンスボードの構成と使い方．*Interface*，12月号，p.108-119（2000）．
- (5) Takeshi Eto：Java™ Technology in Silicon：A Review of the Hardware and Software of Fujitsu's Technology for the Java Platform．  
<http://servlet.java.sun.com/javaone/javaone2000/>
- (6) Takashi Aoki et al.：“On the Software Virtual Machine for the Real Hardware Stack Machine”  
Proceedings of the Java[™] Virtual Machine Research and Technology Symposium (JVM '01)，221，USENIX Association．
- (7) <http://pr.fujitsu.com/jp/news/2000/11/9.html>  
JTRON2.1仕様準拠JTRONサポートパッケージ「J-REALOS/PJ」を提供開始