モデルベース開発のポイント「３段検証」
ＺＩＰＣ／高速Ｃシミュレータ／そして富士通マイコン

富士通株式会社
ＬＳＩ事業本部設計共通技術統括部
第２プラットフォーム開発部・技師
五十嵐　純

１．はじめに

富士通はかなり早い時期からキャッツ社ＺＩＰＣと連携したソリューションを提供し、更に、顧客様の動向に沿った開発環境を提供するため常に進化し続けている。

当初は、ＺＩＰＣでシミュレーション検証を行い、コード生成後は弊社マイコンのＳＯＦＴＵＮＥ－ＩＣＥで実機検証を行うという２段検証で十分であった。しかし近年、モデルベース開発やシミュレーション環境が実用ベースになるにつれ、複雑なモデルから出たＣコードをいきなり実機で検証するのではなく、その中間にマイコン・シミュレータによるＣコード論理検証を加えた３段検証が重要となってきている。

これからのモデルベース開発における、この３段検証の効果や、それにキャッツ／富士通がどう役立つかについて説明する。

２．富士通の考え方の根底
「フロントローディング」
「Ｖ字開発環境ビジョン」

システムの開発においては、「高品質」「低コスト」「短納期」の追求が永遠のテーマである。

以前から富士通ではそれに対する一つの解答として「フロントローディング」という考え方を提示している。フロントローディングとはシミュレータを使った検証の前倒しのことである。従来、マイコンを使ったシステムを開発する場合、ソフトウェアの検証は、マイコンが搭載されたハード（ボードや周辺の入出力装置）が出来た後、実際にそれにソフトを搭載してソフトとハードの検証を同時に行っていた。開発言語はＣまたはアセンブラであり、仕様書を正確に書くよりは、システム設計者兼プログラマが正確なプログラムを書いてその検証をハード・ソフト合わせた環境で行えば大体要求仕様に合ったシステムが完成してしまう場合すらあった。４～８ビットマイコン程度の規模が小さなシステムでは、このやり方はきわめて効率的、且つ現実的と言える場合もある。しかし、16～32ビットマイコンになりシステム規模が大きくなると、ハードを作る事自体が難しいため、ソフト開発者がハードを使った検証を行う工程になってもまだハードができておらず待ち時間ができてしまう事がある。また、プログラム構造も複雑で大規模になるため、大昔のようにＣ言語ソースプログラムそのものが仕様書、などという訳には到底いかない。仕様設計をきちんと行ってからでないと、まともなプログラムを書く事はできない。また、ハードの試作機ができてきてからハード・ソフトを合体させ検証を始めては見たものの、複雑なシステムの場合、不具合が見つかった時にそれがハード起因のものかソフト起因のものかが判りにくく時間ばかり過ぎてゆく。そしてもしハードに原因が有った場合、ハードの作り直しとなりまた待ち時間が発生したり、ソフトの仕様修正という手戻りが発生していた。

そこで、ハードが出来る前の段階からシミュレータを使ってソフトの検証を前倒しして、上流から品質を作り込もうという考えがフロントローディングである。

フロントローディングでは、上流でＣＡＳＥツールを使い要求仕様を分析しながらモデリングし、ソフトウェア仕様書を作成し、その仕様書レベルでシミュレーション検証を十分行ってから、自動コード生成機能によりマイコンのＣコードを生成し、マイコン環境を使った検証に移る。これにより上流から品質を作り込め、後工程での検証・修正作業が減り、全体的なコストダウンと開発費削減を行う事ができる訳である。

富士通ではこのフロントローディングの考え方を、ソフトだけでなくＳｏＣ、メカも含めた全体システムの開発に取り入れ、各種モデリングツールや各種手法を使い、上流から品質を作り込み検証しながら開発することで、高品質・低コスト・短納期で完成させるための開発環境を提供することをビジョンとしている。（図１）

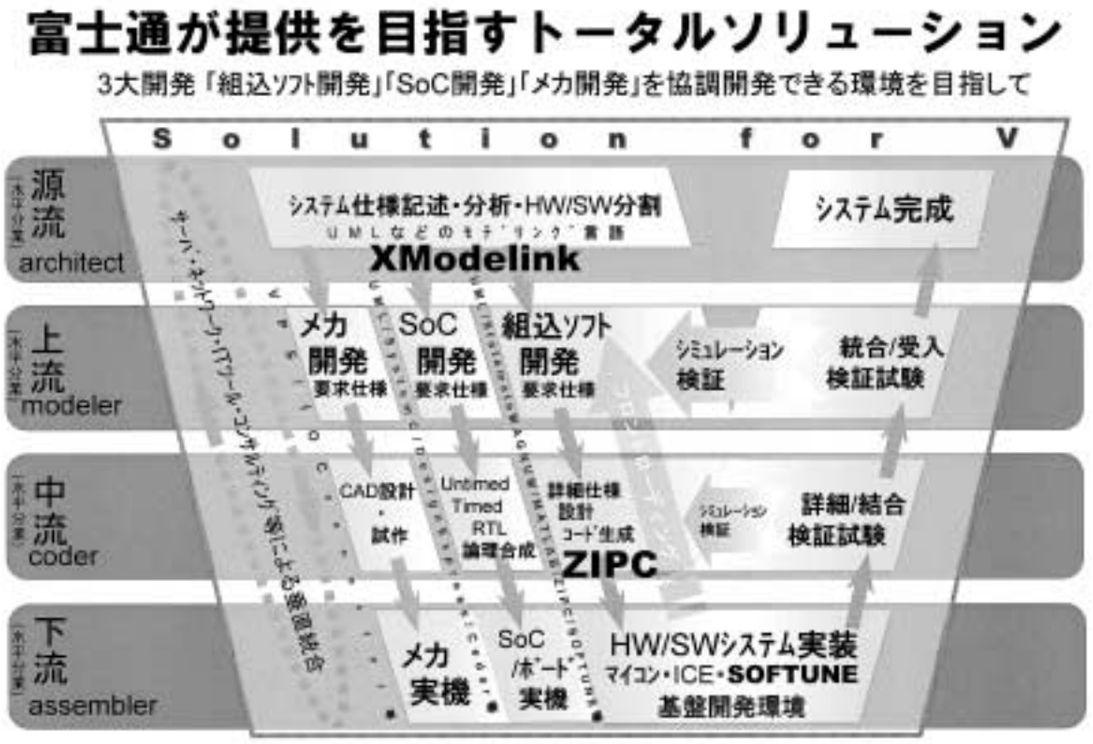

図１　富士通の開発環境ビジョン

３．３段検証理論

さてこの「フロントローディング」「Ｖ字開発環境ビジョン」の実践において、いかにソフトの検証を前倒ししていくかについて、検証の粒度に着目して考察してみる。（図２）

Ｖ字開発プロセスのなかでのソフト検証の粒度は単純に大きく分けるとまずは「仕様での検証」と「実機での検証」の二つの粒度レベルに分けられる。つまり、要求仕様に沿って作成した仕様書を十分検証する第１フェーズと、次にその仕様書を基に自動コード生成等でＣコードを生成しＩＣＥやマイコン実機で検証する第２フェーズ、の２つである。

例えば、ＺＩＰＣで良い仕様書が書ければ、あとはＺＩＰＣでコード生成したものを実機で検証すれば“システム完成”となるはずである。

しかし、システムの規模が大きくなるほど、この２つの検証だけではうまく行かない場合が多くなってきている。マイコンのハード・ソフトを合わせた検証を行って不具合が検出された時に、マイコンのハード（ボードや周辺機器）が悪いのか、ソフトの造りが悪いのかが判別しにくい場合が発生する。一度それが発生するとハード・ソフトが複雑に絡まった実機検証試験のジャングルに迷い込んでしまい、時間とコストを費やしてしまうことになる。

これは最上流のモデル検証と最下流の実機検証では検証の粒度のレベルが違いすぎるために

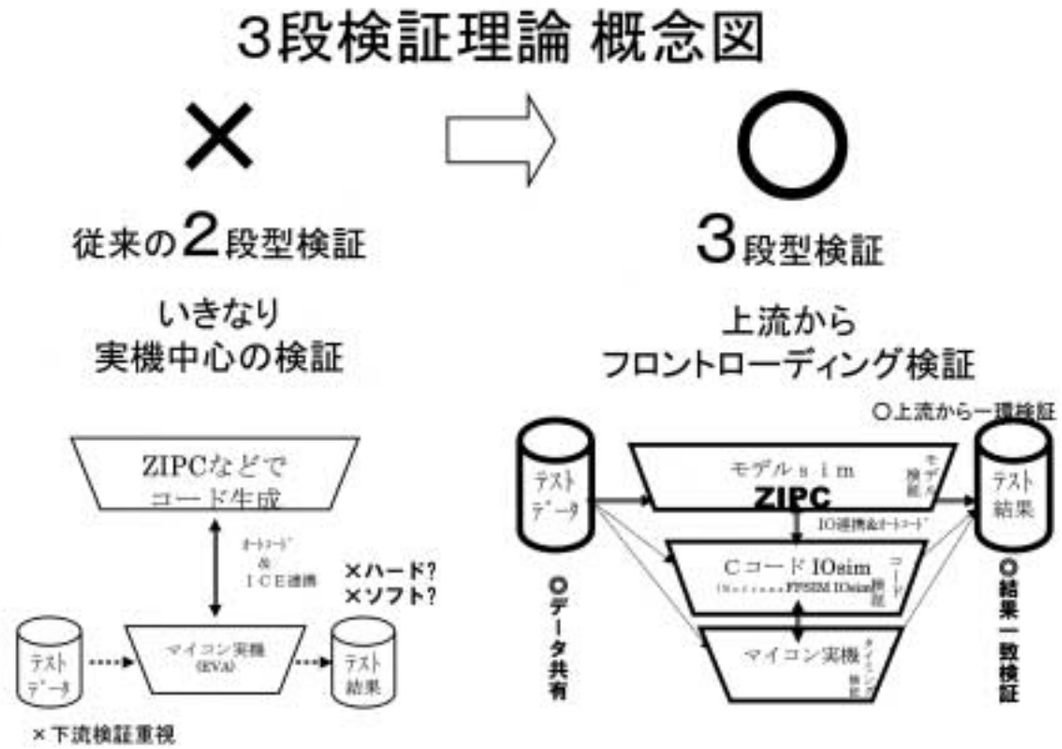

図２　３段検証のメリット

起こる。

これを避けるためには、上流から下流までの粒度の格差を補う必要がある。つまり、マイコンを使った検証（前述の第２フェーズ）の粒度をさらに「Ｃコード検証」と「実機での検証」の二つに分け、検証粒度のバランスを適正化するべきである。

このような背景により「１．モデル仕様」「２．Ｃコード」「３．実機」の３段の検証モデルができあがった。

３段が高品質・低コスト・短納期を目指す開発環境としてちょうど良いバランスであると考える。２段の検証では品質が確保しにくく、また４段以上では工程が延びコストが嵩む。

もちろん、実際の開発時には、厳密にはこの３つ以上の検証工程が存在しているので異論がある方もいると思うが、フロントローディングの考え方を推進するためにもここでは検証プロセスを敢えて「３」という覚えやすい魔法の数字でモデリングして説明させていただく。

４．３段検証の実際の手順とツール

３段検証の概念モデル図（図２）に沿って実際に使用するツールと検証手順を説明する。

●ステップ１！　ＺＩＰＣ：モデル・シミュレーション検証

ＺＩＰＣで状態遷移モデルの仕様を設計し、まずはＺＩＰＣシミュレータでその仕様が正しいかを検証する。この時点では仕様書だけの検証となる。

十分な検証が終わった時点でＺＩＰＣジェネレータにより富士通マイコン用のＣコードを生成する。ＺＩＰＣのＳＯＦＴＵＮＥ連携機能により、ＺＩＰＣから自動的にＳＯＦＴＵＮＥのプロジェクト形式でソースコードが生成される。このプロジェクト生成機能は他社半導体メーカツールへのコード生成にはない独自な機能である。これにより富士通のマイコンであれば次の検証フェーズにスムーズに移行できる。

●ステップ２！　ＳＯＦＴＵＮＥ高速シミュレータ：Ｃコード論理検証

上流のＺＩＰＣでのモデル仕様書の検証が終わったあと自動コード生成されたＣコードは、いきなり実機で検証するのではなく、まずはＳＯＦＴＵＮＥのシミュレータデバッガにより論理的に検証する。

従来のＳＯＦＴＵＮＥシミュレータではマイコンのコア命令しかシミュレートしなかったが新規開発したＳＯＦＴＵＮＥ高速シミュレータ「ＦＦｓｉｍ」では、マイコンに内蔵しているＩ／Ｏの割り込みシミュレーションを作り込むためのＡＰＩを用意している。富士通側でこのＡＰＩを使って一部のマイコン品種のＩ／Ｏシミュレーション機能を「ＩＯｓｉｍ」というモジュールで提供する。このＦＦｓｉｍ　ＩＯｓｉｍを用いることで、ＺＩＰＣから生成されたＣコードをマイコンのＩ／Ｏも含めて検証できる。ただし、このＦＦｓｉｍ　ＩＯｓｉｍは単にＩ／Ｏの順番の検証を保証するだけであり、実際のマイコンで動いたときの正確な時間は保証しない、という割り切った仕様のツールである。しかし、Ｉ／Ｏを含めた検証ができるため、ユーザが書いた殆どすべてのＣコードソース部分をなんの手も加えずにそのまま動作検証することができる。

このフェーズでは上流のＺＩＰＣとＳＯＦＴＵＮＥシミュレータが連携している点が非常に重要である。連携していることで、上流モデリングツール（ＺＩＰＣ）で使用した検証試験データをそのままＣコードシミュレータでの検証に使用することができる。

●ステップ３！　富士通マイコン：実機を使ったタイミング検証

最後に、ＦＦｓｉｍ　ＩＯｓｉｍを使って論理の検証が終わったコードをいよいよ実マイコンのＩＣＥ上で動作検証を行う。このフェーズでは正確な時間測定やタイミングの検証を行うことが出来る。ＺＩＰＣエミュレータによりＳＯＦＴＵＮＥのＩＣＥとの連携を行っているので、モデルの検証に使ったデータをそのまま使用することができる。例えばある状態からある状態に遷移する処理に実際には何μ秒掛かるか等も正確に測定可能である。

また、最上流での検証データをそのまま使用することができ、各検証レベル毎に検証結果の一致検証を行うことで検証の精度を確認することができる。

５．富士通の新しい強み：Ｉ／Ｏシミュレータ

このように、３段検証の２段目のＣシミュレータで十分なＣコードの検証を行うためには、マイコン内部のＩ／Ｏまでシミュレーションできなければならないと考え、富士通は新規にＩ／Ｏシミュレータを開発した。他社半導体メーカでは、Ｉ／Ｏシミュレータは開発に手間がかかる割りにはうまく使い方を提示できていないのか、いまひとつ開発に消極的なようである。

富士通のＩ／Ｏシミュレータ（ＩＯｓｉｍ）はＳＯＦＴＵＮＥ高速シミュレータ（ＦＦｓｉｍ）のオプション機能として用意している外部ＡＰＩを利用して開発してある。（図３）

例えばユーザプログラム内に、

１）リロードタイマの初期値をレジスタに設定し、

２）マイコン内のリロードタイマの制御ビットを操作し、

３）リロードタイマをスタートさせ、

４）タイマ値が０になると割り込み発生を検知し、

５）それに応じた動作をする、

という順番にＣコードで記述してあるとする。従来版のＳＯＦＴＵＮＥシミュレータでは、タイマの割込みに関しては、一定時間間隔で一回または連続して発生させる機能があり、上記４）５）の割込みのシミュレーション検証は出来た。しかし、１）２）３）の初期設定や制御ビット操作が書かれていてもそれは無視され、それと関係なくシミュレータで設定されたタイマの割り込みを発生させるだけのものであった。

新規開発のＩＯｓｉｍを使えば、このプログラムに書かれた論理の順番どおりにシミュレーション動作する。さらにリロードタイマに限らず、マイコンに内蔵されているＩ／Ｏリソース（Ａ／Ｄコンバータ、ＰＰＧ，ＣＡＮ，ＬＩＮ，ＵＡＲＴ，ウォッチドッグタイマ、などなど）

公開APIにより外部I/Fを自由に開発でき CoSim検証へ発展が可能

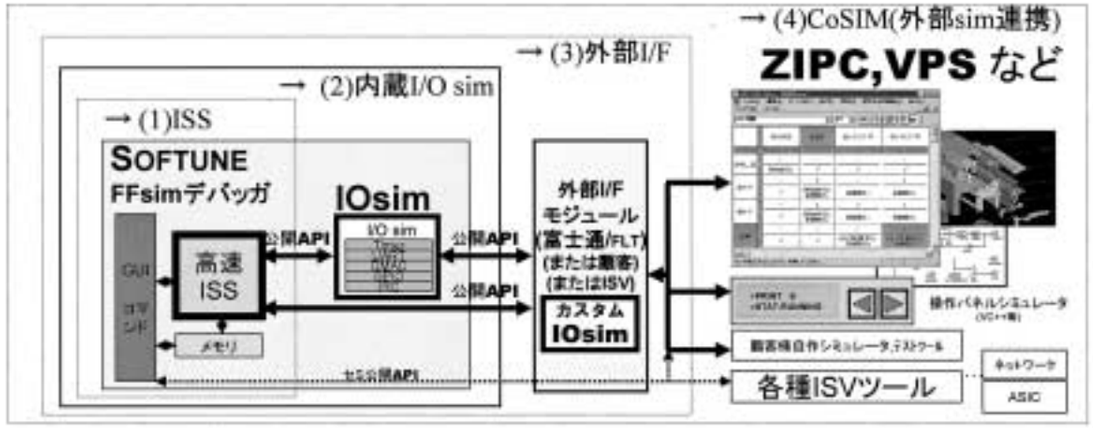

図３　SOFTUNE高速シミュレータとIOsim

すべてに関するシミュレーションを行っている。これによりこれらのＩ／Ｏリソースの制御ビットやレジスタを操作し、割り込み処理するプログラム部分も検証することが可能となる。つまりユーザプログラムの殆どの部分についてこのＦＦｓｉｍ　ＩＯｓｉｍだけで全コード走破検証を行うことができる。

ただしこのＦＦｓｉｍ　ＩＯｓｉｍでは、時間に関しては正確なシミュレーションを行わないように作り込んである。これは時間に関して正確なシミュレーションをすればするほどシミュレータ自体の速度が遅くなってしまい、結局はソフトの検証に使えないものになる恐れがあるからである。また、いくら正確に時間をシミュレートできたとしてもユーザは結局最終的には実マイコンや実ボードで周辺のＩ／Ｏも含めて時間測定をし直さなければならないからである。時間に関する正確性はないがＩ／Ｏの操作手順に関する論理検証をできるだけ高速に行えるもの、と割り切った開発・提供を行っていく。

Ａ／ＤコンバータやＵＡＲＴの様に、外界からのデータ入出力を必要とするような検証には、ＩＯｓｉｍと外界とのデータのやりとり用の外部ＡＰＩをＩＯｓｉｍに持たせることで実現する。例えば、図４のように外部にアナログ値入力用ＧＵＩを置き、いろいろなアナログデータ値を簡単に入力しながら検証ができるようにできる。または、ＺＩＰＣ　ＶＩＰのような外界ＩＯのプロトタイプツールや自作のテストベンチツールからこのＡＰＩを呼び出すことができるため、データ入出力方法のバリエーションは無数にあると言える。

６．ＩＣＥにはないＩ／Ｏシミュレータのメリット

さらに、Ｉ／Ｏシミュレータならではの検証メリットもある。
通常、マイコンには予約ビットと呼ばれる将来のマイコンの仕様改版時に備え、０か１固定に設定しておかなければならないビットがある。しかしながらこの予約ビットを間違って設定してあった場合でも必ずしも実際のマイコンやＩＣＥで動作させてもエラーとして検出できるようにはなっていない。しかしこれがＩＯｓｉｍであれば、予約ビットに関してもマイコンのハードウェアマニュアルに書かれたとおりシミュレーションしているため、そのビットに間違った値が書き込まれた時点でアラームを出すことができる。これにより、いままでの実機／ＩＣＥ中心の検証では潜在してしまっていた予約ビット操作の論理ミスも排除することが可能となり、従来より十分品質の良いコードとした後で、実機検証フェーズに移りタイミング検証を行えるため、最終製品の品質がより向上することになる。（図５）

７．実機とシミュレータのバランスの良い使い分けが今後重要

この３段検証の核となるＳＯＦＴＵＮＥ高速シミュレータは、上流のツールや、ＩＯシミュレータや、外部ＧＵＩなどとのデータ入出力インタフェースＡＰＩを備えている。今後は、これらのＡＰＩを使い、お客様毎のニーズに合わせて柔軟にカスタマイズ対応をして、それぞれの顧客のかゆいところに手がとどくツールにしていく。

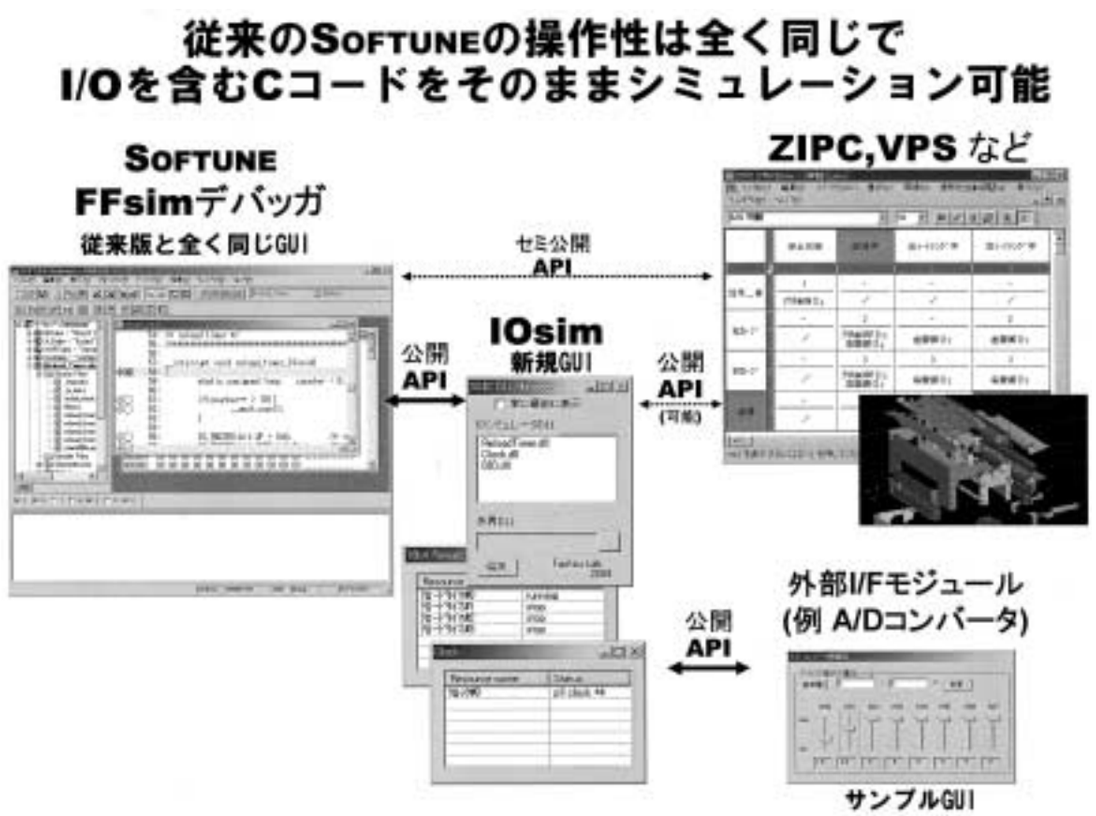

図４　SOFTUNE　IOsimデモ画面

また、シミュレータだけではなく、例えばモニタデバッガやオンチップ・デバッグサポート機能や評価ボード等の実機検証環境の充実も重要でありこれにも取り組んでいく。（図６）

８．ツールの連携だけが重要なのではない

良い組込みシステムを開発するためには、ここで述べてきたツールや連携機能の提供だけが重要なのではない。さらに大事なのは、これらのツールや連携をどう使って開発し、どう検証を行うかの「開発プロセス」の支援であると考えている。

富士通ではここで述べた３段検証の例題作成やデモを行えるようになることを目的に、教育用の題材作りにも取り組んでいる。その一例として、情報処理学会が主催するＭＤＤロボット・チャレンジという、マイコン制御の小型飛

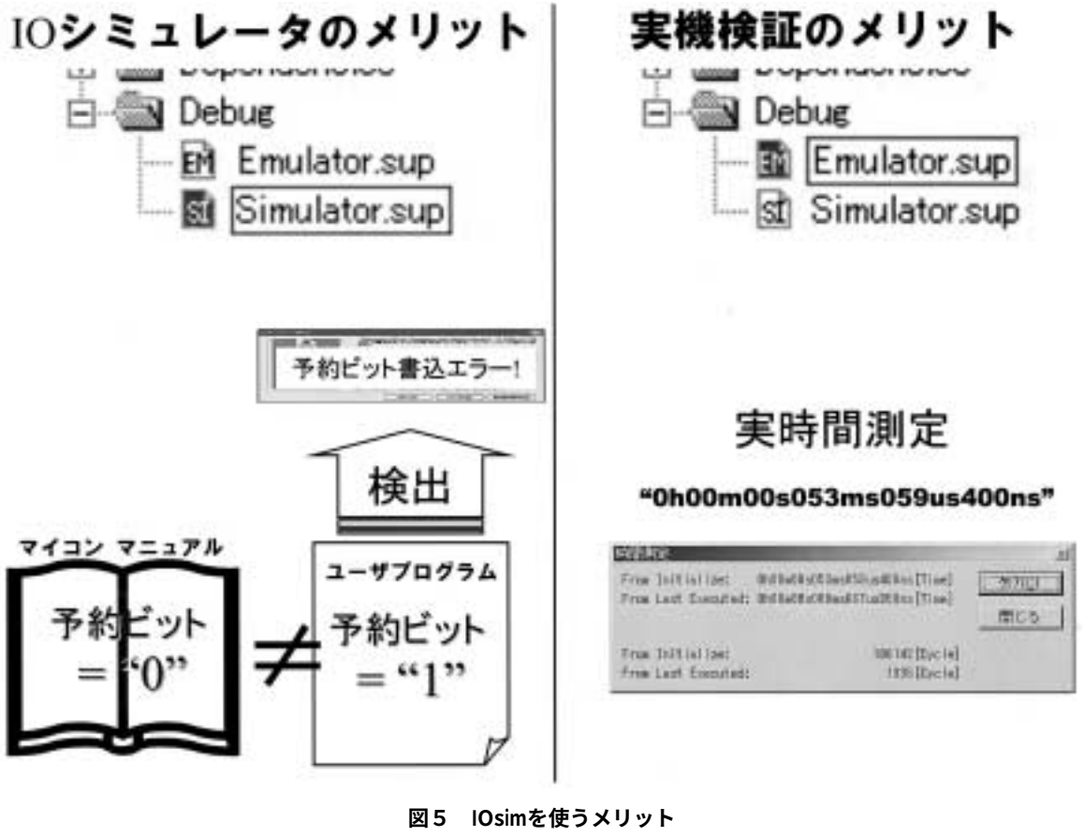

図５　IOsimを使うメリット

行船システムの開発コンテストに、富士通デバイス株式会社が、キャッツ、専修大学飯田研究室とチームを組み、参加している。ＺＩＰＣと富士通マイコンを使ったハード・ソフト両面の開発の競技であるため、ここで開発されたソフトウェアモデルやハードウェア・ボードを今後もデモや展示会などで分かりやすく説明しながら、モデリング開発や、３段検証の実践を普及させていく。

また、マイコンのＩ／Ｏシミュレーションそのものも状態遷移の固まりである。この点から我々自身のＩＯシミュレータの開発そのものにもＺＩＰＣが活用できないかも検討してみたいテーマである。

９．まとめ

今回は、ＳＯＦＴＵＮＥ高速シミュレータを核としてＺＩＰＣなどの上流ツールと下流の実機検証環境との３段階の繋がりに着目して説明した。

富士通はV字開発環境ビジョンに沿って常に進化中であり、それを実現させるためには、今後もＺＩＰＣやキャッツ社との連携が欠かせない。

引き続き、キャッツ社＋富士通で産み出す連携ソリューションに期待していただきたい。

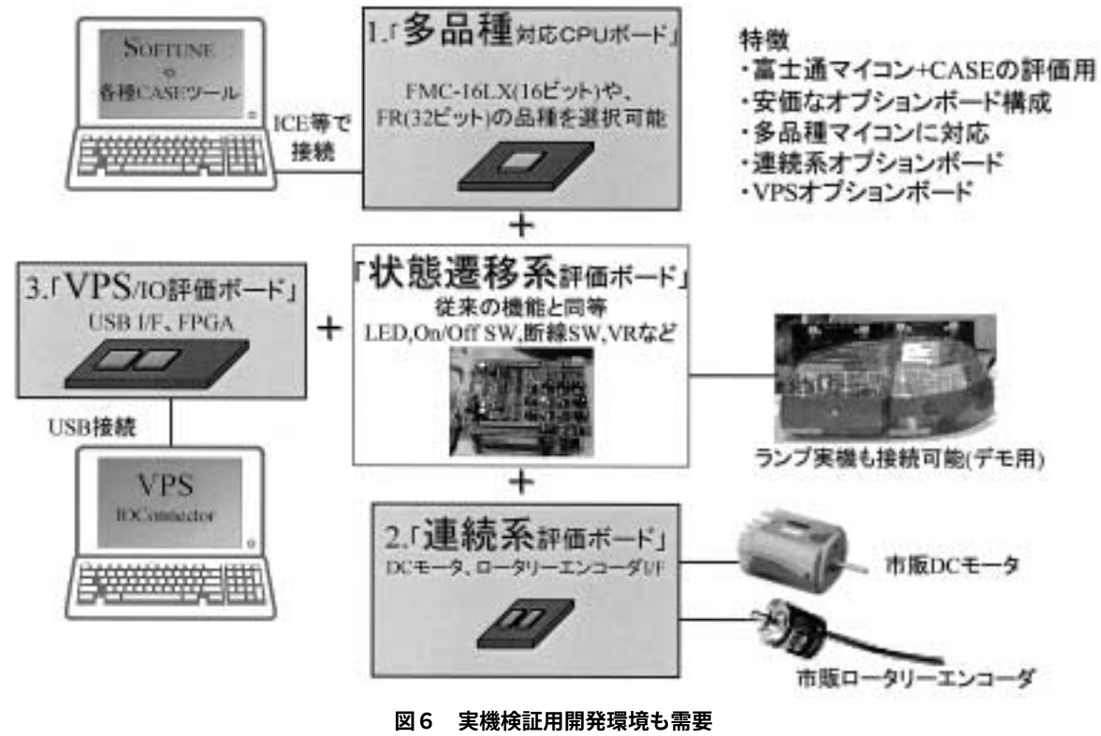

図６　実機検証用開発環境も需要