

VirtualCRAMAS (SILS)へのISSレス技術の適用

Application of ISS-less technology to VirtualCRAMAS (SILS)

森	山	裕	Yuu MORIYAMA
深	澤	健	Takeshi FUKAZAWA
前	川	正	Masahiro MAEKAWA
北	村	章	Akira KITAMURA



要 旨

従来、ECU (Electric Control Unit) に搭載する車両制御ソフトウェアの検査品質向上のため、実車搭載前にHILS (Hardware In the Loop Simulator) による検査を行っていたが、車両制御システムの複雑化および開発の分業化などによりECUの入手やHILSの構築が困難になってきた。

そこでECUハードウェアをソフトで置き換えた仮想ECUを使用するSILS (Software In the Loop Simulator)を導入することで、車両制御ソフトウェアの検査品質・検査効率の向上やECUへの車両制御ソフトウェア実装前工程への検査前出しおよびより上流工程のロジック検証への対応が検討されているが、シミュレーション速度が十分でなく実用化への課題となっていた (テン技報第47号に関連論文掲載)。

今回、テン技報第47号掲載論文のSILSで用いた、マイコンを命令セット単位で忠実に模擬するISS (Instruction Set Simulator) 方式ではなく、車両制御ソフトウェア動作に必要なマイコン動作のみ模擬するISSレス方式を導入することで、HILSと同等の検査精度を保ちつつ、より高速動作可能なSILS「VirtualCRAMAS」を開発したので報告する。

Abstract

Conventionally, we have used HILS (Hardware In the Loop Simulator) for better inspection quality of vehicle control software in ECU (Electric Control Unit) before setting it up on an real vehicle. However, due to more complicated vehicle control systems and the division of development work, it has become difficult to obtain the ECU and build the HILS for the inspection.

There have been discussions on improving vehicle software inspection quality and/or efficiency, moving its inspection in ECU to an earlier stage than its actual setting, and verifying the logic in the upper process by adopting the SILS (Software In the Loop Simulator) that uses a virtual ECU instead of real ECU. However, SILS simulation speed is not fast enough and that has been an obstacle to practical use. [FUJITSU TEN TECHNICAL JOURNAL No. 47 in Japanese (No. 27 in English) carries the related article.]

In this paper, we explain our developed SILS "Virtual CRAMAS," which can inspect the software almost as accurately as the HILS does at faster speed. We developed it in the approach discussed here that simulates only microcomputer's operations necessary for the vehicle control software operation and using ISS-less technology, instead of adopting the ISS (Instruction Set Simulator) technology that faithfully simulates the microcomputer in instruction unit and was used for SILS introduced in the above-mentioned technical journal.

1

はじめに

近年、車両制御の高機能化、複雑化に伴いECU (Electric Control Unit) の数が増加し、さらに複数のECUが統合して車両システムの制御を行うようになっている。また、車両開発においては、期間の短縮・試作レス化の動きが加速しており、増加するソフトをいかに短期・効率的に開発し、品質を確保していくかが重要な課題となっている。そのようななかで、実ECUおよびHILS (Hardware In the Loop Simulator) などのハードウェアなしでECUのソフトウェア開発を支援するツールとして、SILS (Software In the Loop Simulator) が注目されている。

今回、マイコンを命令セット単位で忠実に模擬するISS (Instruction Set Simulator) 方式ではなく、車両制御ソフトウェア動作に必要なマイコン動作のみ模擬するISSレス方式を導入することで、より高速動作可能なSILS「VirtualCRAMAS」を開発したので報告する。

2

SILSの種類と用途

本章では、さまざまな種類のSILSとその用途について述べる。

なお、一般的には、HILS、SILSと並んで、MILS (Model in the Loop Simulator) およびPILS (Processor in the Loop Simulator) という用語も用いられるが、文献によって基準、用法が異なり統一されていないため、本論文では、フィードバックループの一部にハードウェアを用いるものを「HILS」、ハードウェアを用いず全てをソフトウェアでシミュレーションするものを「SILS」と表

現する。

2.1 エミュレーションとシミュレーション

「SILS」と一言でいってもそのなかにはさまざまな種類が存在する。ここでは、「シミュレーションレベル (精度)」という観点からSILSの分類を行う。対象とするのはマイコン (マイクロコンピュータ) を含むECU部分のシミュレーションレベルである。

まず、図1に示す実ECUのシステム構成からシミュレーションレベルの定義を行う。図1のようにECUでは周辺回路を含むハードウェア上でOS (Operation System), PF (Platform) ソフト、アプリ (アプリケーション) ソフトなどに階層化されたソフトウェアが動作している。前述したようにSILSではECUをすべてソフトウェアでシミュレーションするため、その実現方式はSILSの目的、用途を満たす範囲内で自由であり、それが多種、多様なSILSを生み出す要因である。

ところで、本論文のなかで「シミュレーション」という言葉を多く用いているがECU部分に限っては「エミュレーション」という意味を含んでいる。「エミュレーション」とは、あるシステムの一部を別システム上でそのまま動作させることである。例として、Windows用ソフトウェアをLinux上でそのまま動作させるものや、PC上でゲーム機のソフトをそのまま動作させるものなどがある。いずれにしても「そのまま」、「手を加えずに」というところがポイントとなる。

SILSのECU部分は、ECU内部の階層で、ある層以上をエミュレーション、それより下をシミュレーションで実現することで実装される。SILSのECU部分を設計する場合、

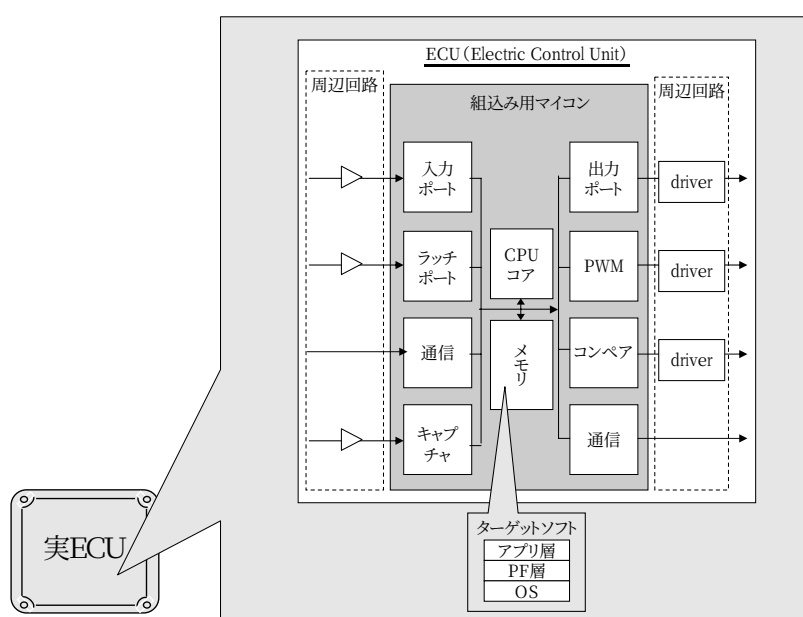


図1 実ECU構成

Fig.1 Structure of Real ECU

まず、ECUのどの部分を「そのまま」、「手を加えずに」実装するか、という点を決定する。この決定には、SILSの用途、すなわちユーザがどのような目的でどのように使用するかという要素が大きくかかわってくる。

当然、ユーザがデバッグの対象とする制御ソフトウェアとその上位の階層のものは「そのまま」、「手を加えずに」実装する、つまりエミュレーションする必要がある。この際、対象とする制御ソフトウェアより下位（物理側、ハード側）の部分は、上位がエミュレートする範囲で自由に実装（シミュレーション）することが可能である。

一般的にはエミュレーション部分を下位に持ってくるほど、より緻密な、実機に近いシミュレーションが可能（シミュレーションレベルが高い）である。その反面、設計・実装に高度な知識を必要とし、対応工数も大きくなる。逆にエミュレーション部分が上位であるほど、シミュレーションレベルが低い設計・実装が容易になる。

最終的にはユーザの用途・目的に合わせて上記内容を決するため、その都度シミュレーションレベルを変更するとシステムの互換性や再利用性が失われてしまう。

そこで、当社では、独自にSILSを3種類のシミュレーションレベルに分類し、それぞれのモジュール構成、I/Fを統一化することにより、効率的なシステム開発およびユーザサポートを実現する方式を開発した。以下でその3種類のシミュレーションレベルについて詳細に述べる。

2.2 3種類のSILS

図2に当社で定義した3種類のSILSの種類とその用途・特徴を示す。図2のように、シミュレーションレベル・用途に応じてそれぞれの種類を

- ・タイプ1型（RAM値渡し）
- ・タイプ2型（ISSレス）
- ・タイプ3型（ISS）

と定義した。

タイプ1は、「アプリソフト」のみを対象としたSILSである。「アプリソフト」が参照するRAM値領域以上をエミュレーションするため、構造は3種類の内、最もシンプルで動作速度も最も高速である。用途はシステム開発やECU開発のアプリケーション開発を想定している。

タイプ2型は、「アプリソフト、PFソフト」を対象としたSILSである。ただし、PFソフトの一部改変を必要とする。PFソフトが参照するレジスタ領域以上をエミュレーションするため、タイプ1型よりも複雑な構造で動作速度も遅くなる。用途はECU開発のアプリケーション開発やPF開発・検査を想定している。

タイプ3型は、「アプリソフト、PFソフト」を対象としたSILSである。マイクロコンピュータの命令セットをエミュレーションするため、実OSやPFソフトを含む全ソフトがオブジェクトのまま動作するが、構造が複雑になり、動作速度も3種類中最も遅い。用途は、マイコン開発を想定している。

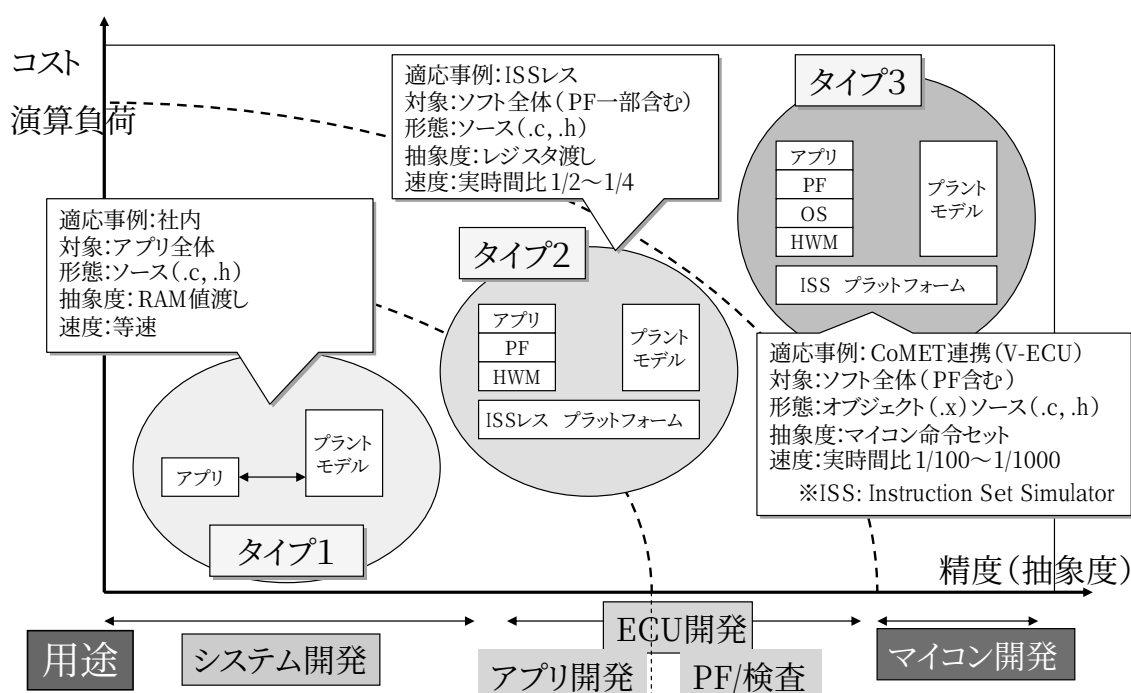


図2 SILSの種類と用途

Fig.2 Types and Use of Our SILS

3

システム構成

3.1 SILSの基本構成

それぞれのタイプは、シミュレーションの用途によってのみ選定され、対象とする制御ソフトの種類によって限定されない。従って、一つの機種開発においても、その開発プロセスおよびデバッグの目的により、これらのタイプを使い分け、組み合わせることが必要となる。そこで当社では、図3で示されるSILSを仮想ECU部分と仮想車両部分に分割し、仮想ECU部分のみ目的に応じたタイプに容易に変更できる方式を実現した。

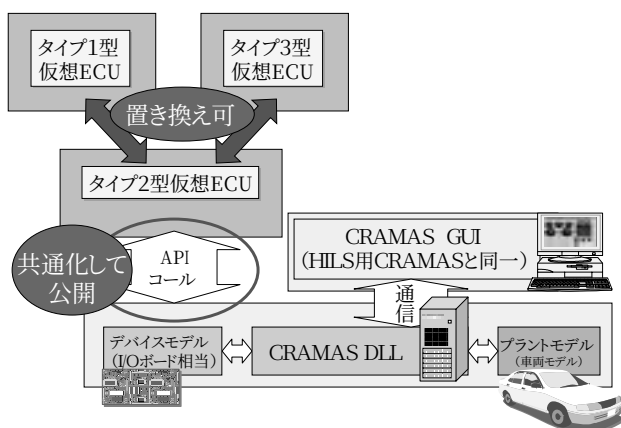


図3 SILS基本構成

Fig.3 Basic Structure of SILS

図3のように、SILSは「仮想ECU」と「仮想車両」の部分から構成される。このうち、仮想ECU部分は、用途に応じてタイプ1型からタイプ3型に切替え可能である。これに対して仮想車両部分は共通のものを扱い、シミュレーションの操作、計測値表示を行うGUIと接続される。

これにより、ユーザは、仮想ECU側を目的に応じて切替えながら、同じ車両モデル、テストコンテンツを用いて連続的なデバッグが可能になる。この時、シミュレーションの設定・操作を行うGUIをHILSと共通化することで、HILSとの車両モデル、テストコンテンツの共有も可能となった。また、複数のECUを統合したシステムをシミュレーションする場合、ECU毎にシミュレーション精度を変えて最適化することも可能になる。

CRAMAS部では、テン技報第47号掲載論文において、「タイプ3型SILS」について述べたため、本論文では「タイプ2型SILS」の構成、内部アルゴリズムなどについて詳細に説明する。

3.2 タイプ2型SILSの構成

図4にタイプ2型SILSの詳細な構成を示す。図4のように、システムは大きく分けて「ECUシミュレータ部」、「シミュレーションエンジン部」、「外部モデル実行部」、「シミュレーションコントロール・表示部」の四つの部分から構成される。

ECUシミュレータ部は、コントロールソフト、IOドライバを備えた「仮想マイコンコア部」およびタイマ・キャプチャ・コンペア機能を備えた「仮想マイコンリソース部」からなる。シミュレーションエンジン部は、イベント演算部・システムクロックからなる。

外部モデル実行部は、コントロール対象となる外部機構などを模擬する「プラントモデル」、I/Fボード機能を模擬する「I/Fボードモデル」の各モデルと、それらのモデルをコントロールする「モデルコントロール部」からなる。

シミュレーションコントロール・表示部は、シミュレーション全体をコントロールする「シミュレーションコントロール部」と、シミュレーション実行環境にデータを入力

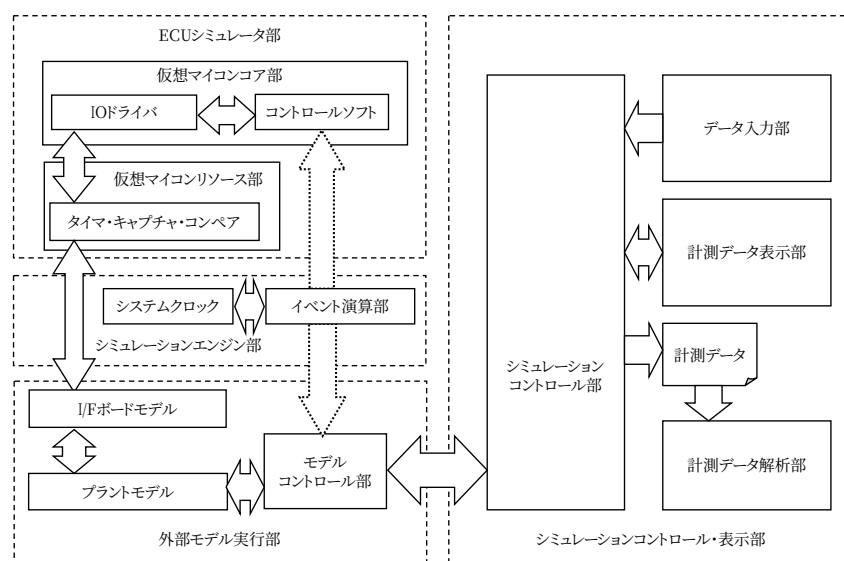


図4 タイプ2型SILS構成

Fig.4 Structure of SILS Type 2

する「データ入力部」、計測したシミュレーション結果を表示する「計測データ表示部」からなり、さらに、蓄積した計測データを詳細に解析する「計測データ解析部」を有する。

3.3 エンジン制御システムへの応用例

実際のエンジン制御システムをタイプ2型SILSで実現すると図5のような対応関係になる。

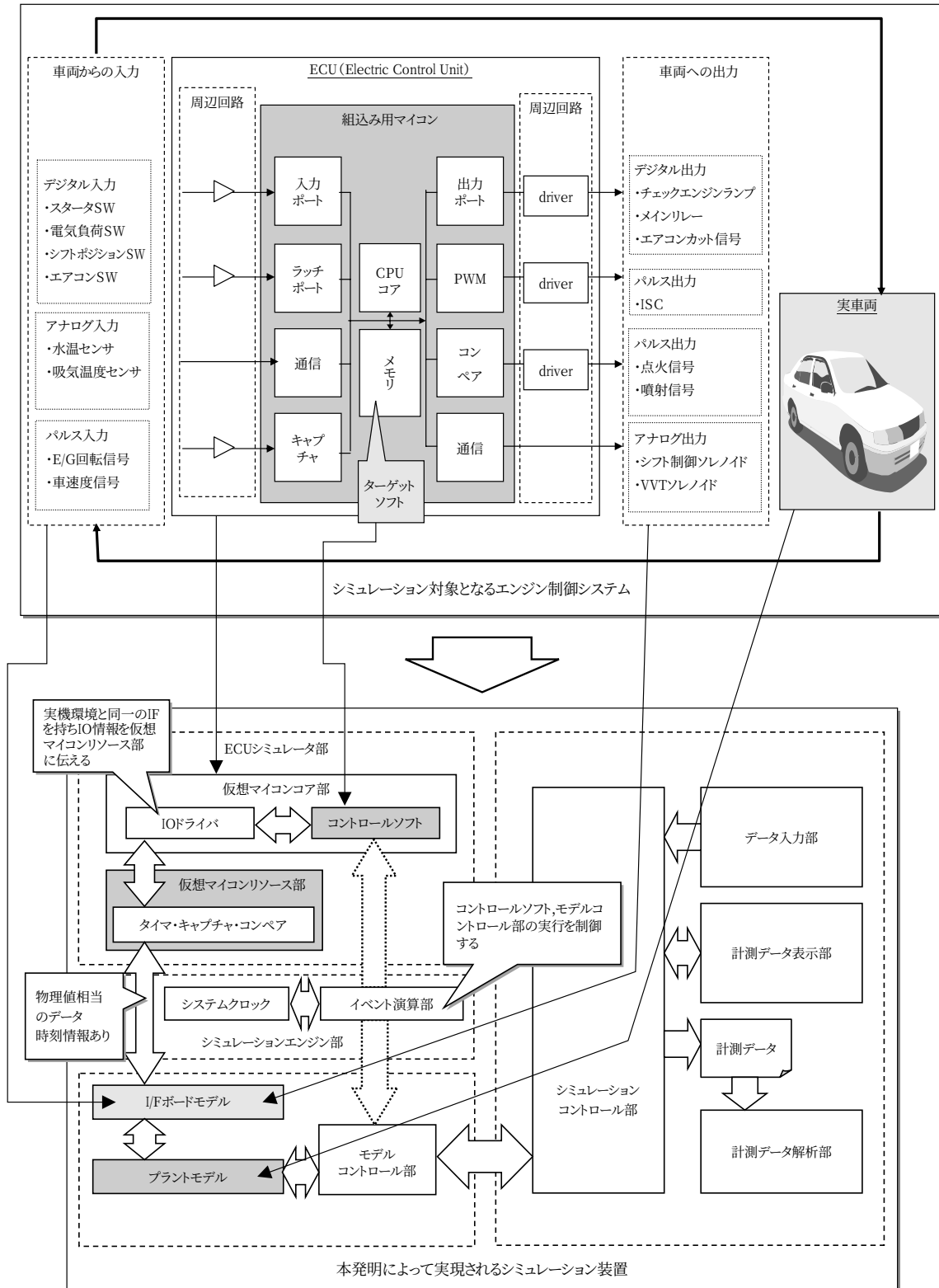


図5 タイプ2型SILSと実エンジン制御システムとの対応

Fig.5 Structure of SILS Type 2 Corresponding to Real Engine Control System

図5のように、実際のエンジン制御システムにおける「実車両」、「ECU」、「車両への入出力」は、タイプ2型SILSの「プラントモデル」、「ECUシミュレータ部」、「I/Fボードモデル」に対応する。

実システムでのマイコン周辺回路の処理を実現するため、「ECUシミュレータ部」内に「IOドライバ」を設けている。また、「ECUシミュレータ部」と「外部モデル実行部」との間に、「シミュレーションエンジン部」を設け、「コントロールソフト」と「モデルコントロール部」実行を制御している。この実行制御を時間同期処理ではなく、イベント処理で実現しているため、タイプ3型より高速なシミュレーション実行が可能となる。次章以降では、これらの高速化手法の詳細について述べる。

4

高速化手法

4.1 仮想マイコン周辺のIO処理

タイプ2型SILSでは、処理高速化のため「CPUモデル」「周辺回路モデル」などのハードウェアモデルを用いることなく、仮想マイコン周辺のIO処理を実現している。

仮想マイコン周辺のIO処理方式を図6に示す。

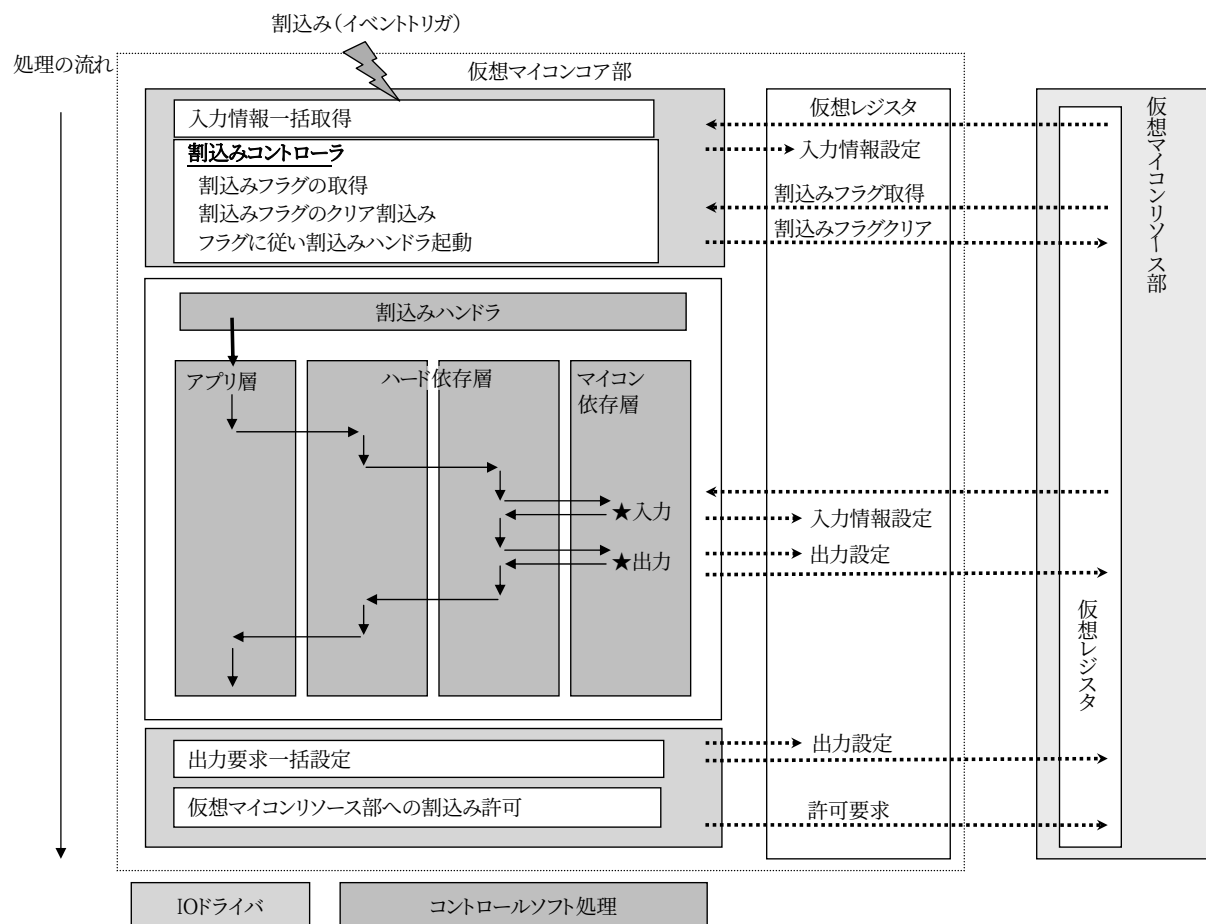


図6 仮想マイコン周辺IO処理

Fig.6 IO Processes on Virtual Microcomputer Periphery

4.2 イベント駆動による高速演算の実現

仮想レジスタを用いた仮想マイコン周辺のIO処理高速化とともに、タイプ2型SILSでは、マイコン内部の処理をイベント駆動することによりさらなる高速化を実現している。

図7にイベント駆動による高速化実現方法を示す。

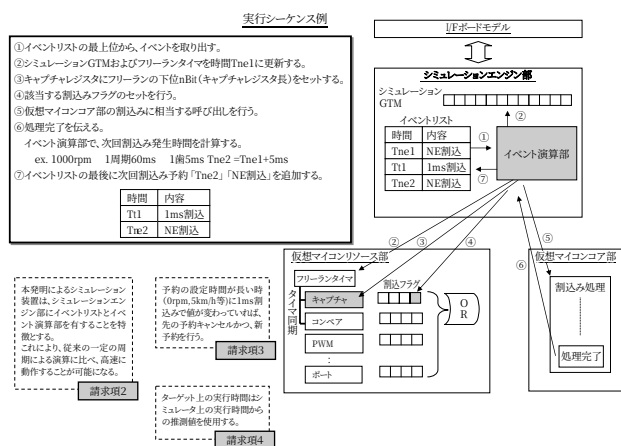


図7 イベント駆動による高速演算実現方法
Fig.7 Method for Event-Driven High Speed Computing

図7に示すイベント駆動の手順は以下の通りである。

- ① イベントリストの最上位から、イベントを取出す。
- ② シミュレーションGTM (Global Timer) およびフリーランタイマを時間Tne1に更新する。
- ③ キャプチャレジスタにフリーランの下位nBit (キャプチャレジスタ長) をセットする。
- ④ 該当する割込みフラグのセットを行う。
- ⑤ 仮想マイコンコア部の割込みに相当する呼出しを行う。
- ⑥ 処理完了を伝える。

イベント演算部で、次回割込み発生時間を計算する。

- ⑦ イベントリストの最後に次回割込み予約「Tne2」「NE割込」を追加する。

これにより、従来の一定の周期による演算に比べ、高速に動作することが可能になる。

4.3 タイプ2型SILSの時間概念

イベント駆動によりシミュレーション時のタイマ処理を図8に示す。

イベント駆動動作

通常のシミュレーション処理では一定の時間を刻むタイマが存在し(図の上段)、高速化する場合、高速な計算機を使用するか、ターゲットソースの動作以外の処理を軽くするしかなかった。

一方、イベント駆動の場合、一定のタイマは存在せず、次々と発生してくるイベントを順番に処理していく。これにより、大幅な処理時間の短縮を可能とするものである。

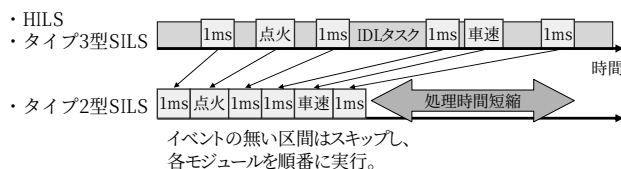


図8 イベント駆動時のタイマ処理
Fig.8 Event-Driven Process

図8のように、通常のシミュレーション処理では一定の時間を刻むタイマが存在し(図の上段)、高速化する場合、高速な計算機を使用するか、ターゲットソースの動作以外の処理を軽くするしかなかった。

一方、イベント駆動の場合、一定のタイマは存在せず、次々と発生してくるイベントを順番に処理していく。これにより、大幅な処理時間の短縮を可能とするものである。

各シミュレーション方式の、シミュレーション時間(仮想時間)と実時間の関係を図9に示す。

SILSの時間概念

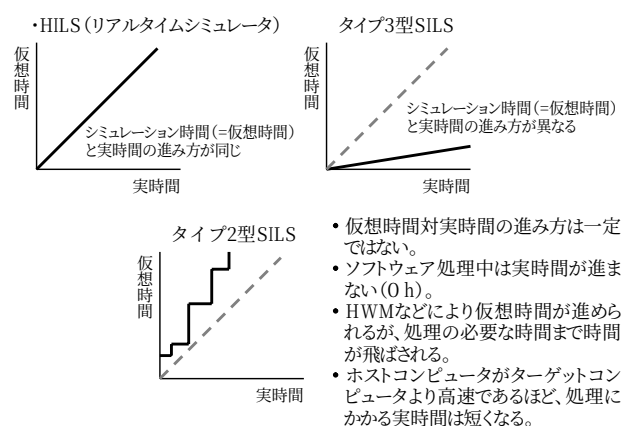


図9 仮想時間と実時間の関係

Fig.9 Relationship between Virtual Time and Real Time

図9のように、HILS (リアルタイムシミュレータ) では、仮想時間と実時間の進み方は完全に一致する。また、タイプ3型SILSでは、仮想時間と実時間の進み方が異なるが、仮想時間と実時間のずれは一定(線形)である。

これに対し、タイプ2型SILSでは、ソフトウェア処理中は実時間が進まず(0h)、イベント処理により仮想時間が進められるため、処理の必要な時間まで時間が飛ばされる。この結果、仮想時間と実時間のずれが一定にならない(非線形)。

5 タイプ2型SILSでのシミュレーション結果

5.1 仮想ECUコネクタ入出力検査

タイプ2型SILSの動作検証のため、実ECUで行われるのと同様のECUコネクタ入出力検査を実施した。第3章で述べたように、VirtualCRAMASではHILSとのテストコンテンツの共有が可能のため、今回、HILS用の検査シートをそのまま流用した。ただし、SILSではモデル化されていない電源系などの一部信号については検査を省略した。

検査結果の一部を図10に示す。

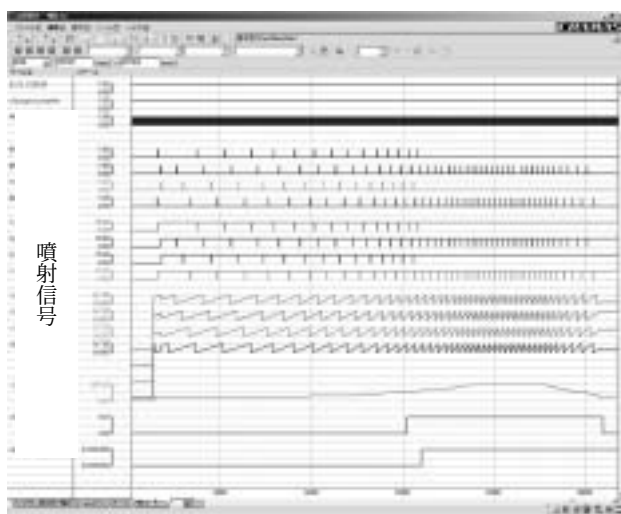


図10 ECUコネクタ入出力検査結果

Fig.10 Inspection Results of ECU Connector Input and Output

図10の結果を詳細に検証したところ、全ての信号が実ECUの検査基準を満たしていることが分かった。その他の信号についても一部を除いて実ECUの検査基準を満たしていた。実ECUの基準を満たしていなかった信号についても、SILSのモデリング方法や時間概念に起因するものであり、ECU制御ソフトの動作に問題ないことがわかった。

さらに、同じ検査項目におけるSILSとHILSの実行結果を図11に示す。

図11のように、SILSとHILSの実行結果は、ソフト動作検証に必要な信号波形、タイミングなどが一致していることがわかった。

以上のことから、タイプ2型SILSがECUコネクタ入出力検査に適用可能であることがわかった。

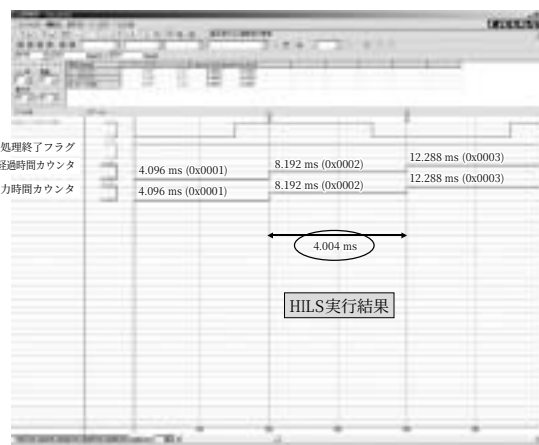
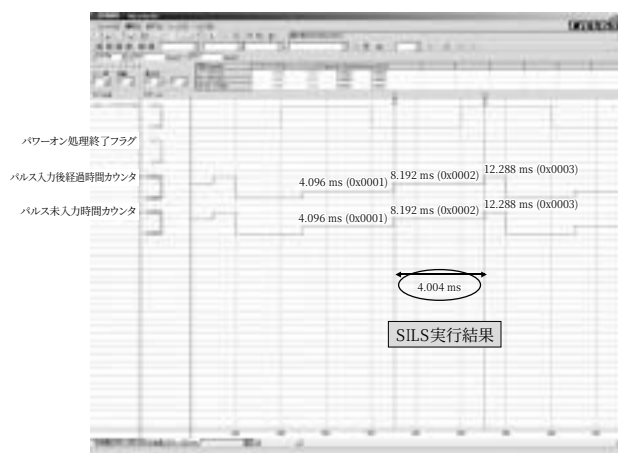


図11 SILS検査結果とHILS検査結果の比較

Fig.11 Results Comparison between SILS and HILS

5.2 高速化手法の効果

表1にタイプ2型SILS (ISSレスVirtualCRAMAS) と他のSILSのベンチマーク結果を示す。

表1のように、タイプ2型ISSレスVirtualCRAMASは、他のタイプ2型SILSと同等、タイプ3型SILSよりも高速なシミュレーションが可能であることがわかった。

表1 SILSベンチマーク結果
Table 1 Results of SILS Benchmark Comparison

	ISSレス VirtualCRAMAS	ISSレスSILS A	ISS SILS B	ISS SILS C	ISS SILS D
タイプ	タイプ2型 (ISSレス)	タイプ2型 (ISSレス)	タイプ3型 (ISS)	タイプ3型 (ISS)	タイプ3型 (ISS)
メーカー	当社	A社	B社	C社	D社
ISS有無	無し	無し	有り	有り	有り
速度	100MIPS	100MIPS	1MIPS	20~50MIPS	30MIPS

6

今後の課題

今回、VirtualCRAMASにISSレス方式を導入することで、HILSと同等の検査精度を保ちつつ、より高速動作可能なSILSを開発できた。高速化によりHILS/SILS混在環境による試作車レス開発にも応用可能となり、車両制御ソフト開発プロセスの大幅な効率化・改革が可能である。しかしながら、今後、本格運用を開始した際の注意点や課題も明らかになった。

6.1 実ECU・HILSとの違い

前述の通り、タイプ2型SILSでは、仮想レジスタやイベント駆動の導入によりシミュレーションの高速化を行っており、実ECUやHILSに対し、一部ハードウェアの動作や時間概念の差異が生じている。その結果、HILSと同等の検査を行った際、検査結果が一致しない場合が生じた。

従って、実機、または、HILSと同じ検査項目を実施する際、検査結果に差異が生じる項目について考慮しておく必要がある。例えば、SILS/HILS共通の検査シートを作成し、該当する項目には予めSILS/HILS双方の検査基準を明記するなどが考えられる。これらの差異を予め定義することで、SILS/HILSで共通の検査を実施することが可能となり、検査効率の向上が可能となる。

6.2 システム構築工数の低減

現状、タイプ2型SILS構築の際、仮想マイコンリソース部内のHWM部分や、I/Fボードモデル部分は、ターゲットとなるマイコンおよびECU周辺回路の仕様に基づきその都度、設計・実装している。

従って、システム全体を構築するのに大きな工数を要している。これらの工数を低減するため、ハードウェアモデルをライブラリ化しマイコンや周辺回路仕様に応じて組合わせて使用する方法などについて検討が必要である。

7

おわりに

今回、VirtualCRAMASにISSレス方式を導入することで、より高速動作可能なSILSを開発できた。

今後は、加速試験などに向けたさらなるシミュレーションの高速化と本格運用のための課題解決が必要である。

最後に、VirtualCRAMASを開発するに当たり、ご協力とご指導を頂いた関係者の皆様に心より感謝いたします。

筆者紹介



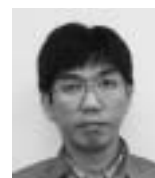
森山 裕
(もりやま ゆう)

1998年入社。以来、制御システム開発用シミュレータ (CRAMAS) の開発に従事。現在、AE本部 制御システム開発統括部CRAMAS部に在籍。



深澤 健
(ふかざわ たけし)

1980年入社。以来、自動車用電子機器および開発ツールの開発に従事。営業技術を経て、現在、AE本部 制御システム開発統括部CRAMAS部長。



前川 正博
(まえかわ まさひろ)

1990年入社。以来、自動車用電子機器および開発ツールの開発に従事。自動車制御ソフト開発を経て、現在、AE本部 制御システム開発統括部CRAMAS部チームリーダー。



北村 章
(きたむら あきら)

2004年入社。以来、自動車用電子機器および開発ツールの開発に従事。現在、AE本部 制御システム開発統括部CRAMAS部に在籍。