

組込みソフト開発支援のためのシステムシミュレーション環境

6N-8

岡島正明 五十嵐真悟 小山徳章 安田剛
株式会社東芝 研究開発センター

1 はじめに

組込みマイコンシステムは、それ自体ではソフトウェア開発環境を持たない場合が多く、開発環境の整った別の計算機上でターゲット用ソフトウェアの開発を行うクロス開発が一般的に行われている。

また、組込みマイコンシステムの開発は、ハードウェアとソフトウェアの開発を並行して行う場合が多く、ハードウェアの開発の遅れがソフトウェアのテスト、デバッグに直接影響を及ぼすことになる。そこで、ハードウェアがなくてもソフトウェアのテスト、デバッグを行うことができる環境が望まれる。

本稿では、開発機上で、ターゲットハードウェアを含めた組込みシステム全体のシミュレーションを可能とする方式について述べる。

2 シミュレーション環境構築の課題

組込みシステムのソフトウェアは、ターゲットハードウェアの制御を行うことを目的とするものであるため、入出力命令が頻繁に使用される。しかし、従来のシミュレーション環境においては、CPUシミュレータによってターゲットCPUの動作を模擬するのみで、入出力部のシミュレーションは行わないものが一般的であった。そのため、入出力部は、デバッグを用いて入出力命令時に値をセットする方法で対処するといったアドホックな対応でしのいでいた。そこで、制御対象の動作や表示/入力用デバイスなどのヒューマンインタフェース(HI)の模擬を行う部分まで含めたシミュレーション方式が望まれる。

ターゲットCPUの動作および入出力までをシミュレートする環境を単一のプロセスで実装した場合、非同期入出力のシミュレーションが難しい、あるいは入出力待ち状態が生じるために全体のスループットが低下するといった問題がある。そのため、入出力部のシミュレータは別プロセスとして実装する方法が提案されている[1]。しかし、プロセス間通信は通常の命令実行に比較すると非常に低速なため、システムの一部ではなく、組込みシステム全体のシミュレーションを行う場合、シミュレーション実行のボトルネックとなってしまう。

また、組込み用マイコンの多くは、プログラムの解釈実行を行うCPUコアのみではなく、入出力制御用の周辺回路(シリアルコントローラ、A/Dコンバータなど)が同一チップ上に配置される(図1)。

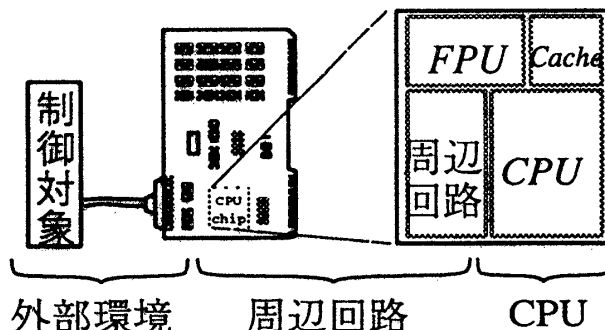


図1: 組込みシステムの構成

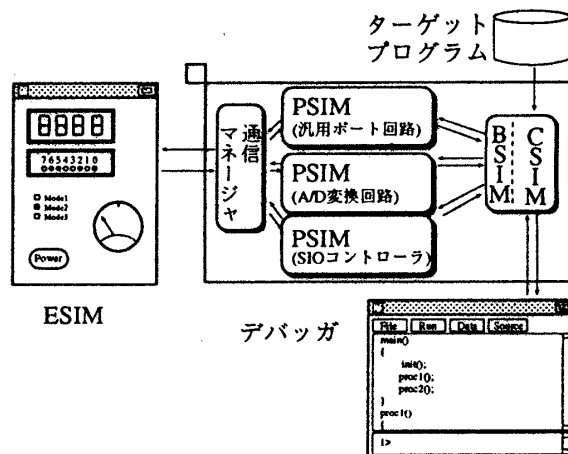


図2: シミュレーション環境の全体構成

CPUシミュレーション部と入出力シミュレーション部からなる方式[2]では、入出力シミュレーション部に周辺回路機能を作り込む必要がある。そのため、ターゲットシステムには依存しない部分である周辺回路のシミュレーション機能を、ターゲットシステム毎に作成しなくてはならないこととなる。

3 シミュレーション方式

2節であげた課題の解決のため、我々は図2のように構成するシミュレーション方式を採用した。

本方式は、大きく3つの要素によってシミュレーション環境を構築することにより、シミュレーションを行う。

- CSIM(Core Simulator): CPUコアのシミュレートを行うもので、ターゲットシステム用に開発されたプログラムの解釈および

An Architecture of System Simulation Environment for Developing Embedded Microcomputer Systems.

OKAJIMA Masaaki, IGARASHI Masato, KOYAMA Noriaki, YASUDA Takeshi.

Research&Development Center, Toshiba Corp.

び実行を行う。

- **PSIM(Peripheral SIMulator):**
周辺回路のシミュレートを行うもので、ターゲット CPU の各周辺回路に対応して複数のものから構成する。
- **ESIM(Environment SIMulator):**
制御対象および HI などの外部環境をシミュレートする。

上記の構成要素以外に、CSIM と PSIM の間のやりとりを仲介する BSIM、PSIM と ESIM の間のやりとりを仲介する通信マネージャを用意している。BSIM は、バスに相当する。通信マネージャは、周辺回路と制御対象とをつなぐ入出力ポートに相当する。また、デバッグは、シミュレーション環境でのターゲットプログラムの実行を制御する。

シミュレーション実行は、次のように行う。

1. デバッグを通じてシミュレーション実行要求を CSIM に通知する。
2. CSIM はターゲット用プログラムの解釈/実行を行う。
3. プログラム中に入出力命令があらわれた場合には、CSIM から BSIM を介して、その入出力命令の対象となる PSIM にシミュレートを要求する。
4. 入出力命令がポートを介して外部機器を制御するものであった場合には、PSIM から通信マネージャを介して ESIM に対して制御値の送信を行う。

本シミュレーション方式では、PSIM および ESIM により制御対象まで含めた入出力の模擬を行うことで、ターゲットシステム全体のシミュレーションを可能としている。また、入出力部分を、汎用性を持つ周辺回路 (PSIM) と、ターゲットシステム毎に異なる外部環境 (ESIM) の二つに分けているので、ターゲットが異なっても、入出力部全てではなくシステム依存部である外部環境のみの作成でよい。

さらに、4節で述べるように、CSIM、PSIM、ESIM はそれぞれ並行に動作する実装方式を採用した。また、PSIM および ESIM を、両者の特性を考慮した別の実装法とする。このことにより、シミュレーション実行の非同期入出力のシミュレーションを可能とするとともにシミュレーション実行速度が向上する。

4 並行動作主体の実現

実際の組込みマイコンシステムにおける CPU コア、各周辺回路、外部環境はそれぞれ物理的に並行に動作する。そのため、シミュレーション環境においても、CSIM、各 PSIM、ESIM も並行に動作することが望ましい。このことは、特に非同期入出力のシミュレーションなどの実現のために重要である。

複数の制御フローを並行に実行する方式としては、スレッドを用いる方式とプロセスを用いる方式が考えられる。

両者を比較すると、以下のような特徴が挙げられる。

- スレッド間通信は、プロセス間通信に比較して通信コストが非常に小さい。
- 並行に動作するものをスレッドで実装するのは、アドレス空間を共有している、同一実行モジュールとなるなどといったことから、スレッドを用いるよりプロセスによる実装のほうが高い独立性を得られる。

また、CSIM、PSIM および ESIM の特徴として、以下のようなことがいえる。

- CPU コアと周辺回路の間はアクセスが頻繁に生じるため、高速なデータの受渡しを要求される。
- 外部環境は製品分野毎に異なるので、作成を容易とするため、独立性の高いものが望ましい。また、MMI など比較的低速な応答でも十分であるものが多い。

以上のことから、通信の高速性を要求されるが、CPU チップという形態での提供が可能な CSIM および各 PSIM は同一プロセス中の別スレッドで、また、高速性を要求されないが独立性の高さを要求される ESIM は別プロセスとして実現するのが妥当であると考えられる。そのため、CSIM、PSIM は同一プロセス中の別スレッド、ESIM はそれとは別のプロセスとして実現することとした。

5 おわりに

組込みマイコンシステム全体を開発機上でシミュレートすることを可能とするシミュレーション方式を提案した。CSIM、PSIM、ESIM から構成し、入出力部を製品分野に依存しない PSIM と製品分野依存となる ESIM に分けた。また、両者を分けたことにより、高速性の要求の高い部分 (PSIM) と、独立性の要求の高い部分 (ESIM) に別の実装法を用いることができ、高速性およびターゲット非依存部の再利用という要求を満たすことが可能となった。このシミュレーション方式を用いることで、開発機上でもモジュールテストレベルではなくシステム全体のシミュレーションが可能となり、ソフトウェア開発期間の短縮が期待できる。

現在、本方式によるシステムシミュレーション環境の実装を終了し、評価中であるが、組込みソフトのテスト、デバッグに十分効果がある見通しを得ている。

参考文献

- [1] 栗山他: “シミュレーションによるリアルタイムシステム開発環境”, 計測と制御 Vol.31, No.7, 1992
- [2] 島袋他: “機器組込み型マイコンソフトの設計・テスト支援方式”, 情報処理学会第 46 回全国大会, 1993