

システムオンチップ時代の組み込みシステム開発

田丸 喜一郎

システムオンチップ時代と言われしばらく経つが、やっと現実のものとして身近になってきた。半導体の集積度は、高性能な組み込みマイコンさえチップの数パーセントに収めてしまう。アナログ部品や DRAM メモリ、FLASH メモリも1チップに混載する技術も確立した。要素部品となるIPの標準化や流通機構の構築も VSI コンソーシアムなどで進んでいる。

システムオンチップは、システムの大半を単一の半導体チップに実装することで、高性能で安価な組み込みシステムを提供することを可能にしたが、その一方で開発の後戻りが許されない状況を組み込みシステムの開発者に要求する。一部の部品を開発の終盤になって差し替えることが困難だからである。組み込みシステムに対する Time To Market の要求も厳しくなっている。半導体メーカーがいくら努力してチップの開発期間を短縮しても、チップを作り直すことはスケジュール的に致命的である。

このため、組み込みシステムの開発を上流工程から系統的支援する設計手法や設計環境の重要性が増している。コデザイン手法、コベリフィケーション手法などである。システムオンチップの時代では、システム設計の工程でハードウェアとソフトウェアの切り分けを適切に行えるかが最大のポイントである。同時に、ソフトウェアで実装する機能が要求する処理量やハードウェアで実装する機能のチップ面積・消費電力の高い見積もり精度が要求される。

従来は、比較的単純にハードウェアとソフトウェアの切り分けが行えた。開発工程も初期の段階からハードウェアの開発フローとソフトウェアの開発フローが分かれた。ハードウェアで実装される機能は Verilog や VHDL などのハードウェア記述言語で、ソフトウェアで実装される機能は C や C++などのプログラミング言語で記述されるのが一般的であった。

システムオンチップ時代では、ハードウェアとソフトウェアを切り分ける前、システム設計段階で機能を実装し評価・検証することが求められる。それでも、単一の言語ですべての機能を記述するのは効率が悪い。「帯に短し襷に長し」である。このため、ハードウェアで実装するかソフトウェアで実装するかによって記述言語を選ぶのではなく、記述すべき処理の特性に応じて記述言語を選択する。組み込みシステムの機能は、イベント処理とデータ処理に大きく分けられる。それに応じて記述言語を選ぶ。

イベント処理は状態遷移図や状態遷移表で、データ処理はデータフローグラフで記述するのが1つの解である。記述された機能を候補となるハードウェアアーキテクチャにマッピングして評価し、ハードウェアアーキテクチャを決定する。その後、ハードウェアにマッピングされた機能に対応する記述はハードウェア記述言語に、ソフトウェアにマッピングされた機能に対応する記述はプログラミング言語に変換する。

ハードウェアアーキテクチャでも、ソフトウェアで実装されるイベント処理を組み込みマイコンで、データ処理を DSP やメディアプロセッサで実装することが1つのトレンドとなっている。処理の特性に応じて記述言語を選択することは、これら実行プラットフォームの選択とも相性がよい。

ZIPC は状態遷移表によるソフトウェア開発を支援するツールである。もちろん、万能ではないし、完全でもない。コデザイン環境やコシミュレーション環境との連携、オブジェクト指向の支援などなど、まだまだ発展の余地がある。今後どのように発展し、システムオンチップ時代の標準ツールの1つとして成長していくか、楽しみなツールである。

[たまる きいちろう]

ZIPC Ver.5.0 リリースのお知らせ

ZIPC Ver. 5.0 とは...

ZIPC Ver.5.0 は、拡張階層化状態遷移表設計手法 Ver.2.0 をドキュメンテーションの中心として、設計、プログラミング、ターゲット試験までのソフトウェア開発のライフサイクルを支援するツールです。従来のエディタ群に新たに TC (Timing Chart) エディタを加えた他、VIP (Visual Interface Prototyping) によるビジュアルで直感的な試験環境、 μ ITRON 対応、日本電気(株)/富士通(株)/松下電器(株)などの ICE への対応により幅広いデバイス選択肢を手に入れる事ができます。

拡張階層化状態遷移表設計手法 Ver. 2.0 (Extended Hierarchy State Transition Matrix Design Method)

制御系ソフトウェアの開発に大きく貢献する「拡張階層化状態遷移表 (EHSTM: Extended Hierarchy State Transition Matrix) 設計手法」が Ver.2.0 となり、並列状態、同期状態、状態階層化、ステートドリブン、アクティビティ等、現場要求事項を実装しました。

ZIPC Ver. 5.0 で追加された新機能

■ TC エディタ (TC: Timing Chart)

タイミングチャートを記述するためのエディタです。Ver.5.0 に新たに加わった新機能で、TC エディタの実装により正確な時間情報を記述する事ができます。

■ ZIPC ビジュアルインターフェイスプロトタイプ (VIP: Visual Interface Prototyping)

Visual Basic (Microsoft 社製)、ZIPC シミュレータとリンクする事によって Visual Basic で書いた画像を STM 設計書上でシミュレーションを可能にします。Ver.5.0 に新たに加わった新機能です。

ZIPC Ver. 5.0 で強化された機能

■ ZIPC シミュレータ (SIM: Simulator)

STM 設計書上で状態遷移、RTOS や割り込み動作のシミュレーションが行えます。また、VIP とのリンクによりビジュアルプロトタイプシミュレーション、 μ ITRON 対応により μ ITRON 仕様のシステムコールを使用する事ができます。

■ ZIPC エミュレータ (EMU: Emulator)

日本電気(株) (V850/78K0/78K4)、富士通(株) (FR/FMC-16/FMC-8)、松下電器(株) (M102H) のインサーキットエミュレータ (ICE) とリンクし STM 設計書レベルのターゲットデバッグを行う事ができます。

★ その他のツールも、さらに機能アップ! ★

動作環境

日本語 Windows95/NT 4.0 が快適に動作する環境 (Pentium 133 MHz 以上、24MB メモリ以上)

* 紙面に記載された社名、製品名は各会社の商標または登録商標です。

詳細については、キャッツ(株)までお問い合わせ下さい。