

第3回

組み込みシステム開発の
要素技術と標準化

高田広章 名古屋大学
大学院情報科学研究科情報システム学専攻
hiro@ertl.jp

組み込みシステムは、制御対象の機械・機器の多様性を反映して、システムの規模・性質のいずれの面からもきわめて多様である。また、組み込みシステム開発に使われる要素技術（プロセッサ、リアルタイム OS、ネットワークなど）もきわめて多様である。本稿では、まず組み込みシステムの特性とその分化について述べた後、組み込みシステム開発の要素技術が多様である理由について説明する。また、組み込みシステム向けの要素技術について概観し、その最近の動向を述べるとともに、標準化の動きについて解説する。

組み込みシステムの特性とその分化

組み込みシステムとは、各種の機械や機器に組み込まれて、その制御を行うコンピュータシステムのことをいう。また、機器の一部となるコンピュータシステムで、機器と一体で製品（1つのシステム）と見なされるものと定義される場合もある。いずれの定義によっても、どこまでを組み込みシステムと呼ぶかには曖昧なところがあり、極端に言えば、パソコンやサーバ機などの汎用コンピュータ以外は、すべて組み込みシステムであると捉えることもできる。

このように広く捉えると、組み込みシステムは、システムの規模・性質のいずれの面からもきわめて多様であるが、組み込みシステム向けの技術という分野が成立するのは、組み込みシステムの多くに共通する特性があるためである。すべての組み込みシステムに共通とまでは言えないが、多くに共通する特性として、次の4つを挙げる^{1), 2)}ことができる。

- (1) 専用化されたシステム
- (2) 厳しいリソース制約
- (3) 高い信頼性
- (4) リアルタイム性

最近の傾向として、これまで組み込みシステムに分類していたシステムの中で、これらの特性がそれほど当ては

まらず、むしろ汎用システムに近い性質を持ったものが目立つようになってきたことが挙げられる。

その代表格として、携帯電話機やカーナビゲーションシステムを挙げるができる。たとえば携帯電話機を例にとると、すでに電話をかけるための専用システムとは言えなくなっている。また、10MBを超えるメモリを内蔵し、メモリ制約も緩やかになってきている。リアルタイム性が必要な無線系の制御や音声データの処理については、ベースバンドチップと呼ばれるLSIで実現されているため、メインプロセッサ上での処理にはそれほどリアルタイム性は必要なくなっている。

もちろん、上に挙げた4つの特性が典型的に当てはまる組み込みシステムも数多くあり、これらのことから、組み込みシステムの分化が進みつつあると考えられる。

もう1つ注意を促したいのは、上述したベースバンドチップの中にも、プロセッサが（場合によっては複数）組み込まれており、それだけでもコンピュータシステムと呼べるだけの複雑性を持ってきていることである。そのためこのようなLSIは、システムLSIあるいはSoC (System on Chip) と呼ばれており、そこに組み込まれるプロセッサで動作するソフトウェアも、組み込みソフトウェアと呼ばれている。このような「LSIへの組み込みソフトウェア」は、従来の（機器への）組み込みソフトウェアと特性が似ているものの、求められる要素技術には違いがある。そこで最近では、組み込みソフトウェアと区別して、ハードウェア依存ソフトウェア (hardware-

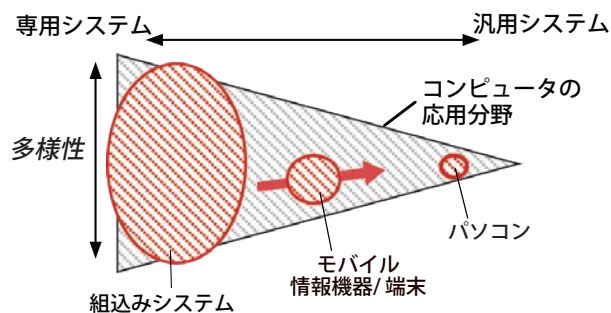


図-1 組込みシステムの多様化と分化

dependent software) などと呼ばれる場合もある。

組込みシステム開発の要素技術の多様性

上に述べた組込みシステムの4つの特性の中で、(2)～(4)の3つは、システム開発を困難にする要因である。一方、(1)の「専用化されたシステム」という特性は、システムを開発する上で有利に働く特性であり、他の3つの特性を満たすために活用することができる。すなわち、システムをあるアプリケーションに特化することで、リソース使用量を減らしたり、信頼性やリアルタイム性を向上させることが可能である。このことから、組込みシステム技術とは、専用化されたシステムであるという特性を活かして、(2)～(4)の制約を満たすための技術であると位置付けることができる。

しかしながら、あるアプリケーション専用のシステムを構築するために、それに特化した要素技術を用いる方法は、開発コストが非常に高くなるために一般には受け入れられない。そこで、組込みシステム向けの要素技術は、共通の特性を持ったアプリケーションをドメインとして括り出し、それを狙って開発されるのが一般的である。また、アプリケーションに最適化するための仕組みを、それぞれの要素技術に持たせるアプローチが採られることも多い。たとえば、アプリケーションの要求に応じて、命令セットやバス幅などが変更できるプロセッサ(構成可能プロセッサ)がこの例である。すなわち、専用システムを開発するための汎用(または多目的)技術が求められているわけである。

このように、特定のアプリケーションドメインを狙っ

て開発するとはいえ、ドメインごとの要求の違いは大きく、組込みシステム向けの要素技術はきわめて多様である。たとえば、組込みシステムに使われるプロセッサは、規模的には8～64ビットまでのものが幅広く使われているし、性能要件の面でも多様で、高速化に加えて、低消費エネルギーやコード効率も重視して設計される。また、組込みシステム向けのリアルタイムOSは、製品化されているものだけでも100種類を下らないと言われているし、ネットワークについても、アプリケーションドメインに特化した技術が数多くある。

これらは、パソコン向けのプロセッサ、OS、ネットワークが、いずれも数種類に収束しつつあるのと対照的であるが、汎用システムの要素技術が、多様なアプリケーションからの最大公約数的な要求に合致するように設計されていることを考えると、自然に理解することができる。

この様子を図示したのが図-1である。この図で組込みシステムを大きく示してあるのは、システムおよび要素技術の多様性を示している。一方市場規模的には、汎用システムよりも(かなり)小さい。言うなれば組込みシステム技術は、広い技術領域に薄く分布しているわけである。またこの図では、携帯電話機を代表とするモバイル情報機器/端末が、汎用システムに近付いていることも示されている。

組込みシステム向けの要素技術と標準化

この章では、組込みシステム向けの要素技術の概要と動向について解説し、その標準化の動きについて述べる。なお、同様の内容に関して、過去に文献2)でも解説し

ているため、本稿ではそれ以降の動向に重点を置く。また、組込みソフトウェアの開発技術に関しては、文献3)やこの連載の他の記事に譲ることにして、ここでは扱わない。

■ プロセッサ

前述の通り、組込みシステムには非常に多くの種類のプロセッサが用いられている。命令セットアーキテクチャにも多くの種類があるが、同じ命令セットアーキテクチャを持ったプロセッサでも、各種のアプリケーション向けにシリーズ展開されている。また、DSP (Digital Signal Processor) やメディアプロセッサなど、用途専用のプロセッサも広く使われている。さらに、前述の構成可能プロセッサが適用されるケースもある。

これらの専用プロセッサは、音声や画像などのメディア処理、ネットワーク処理、暗号処理などの分野で有効性が高いと考えられる。メディア処理においては、同じ演算処理を繰り返し行うために、演算を並列に行う命令を設けるなどの方法で、その演算処理を高速化する方法が有効である。それに対して、ネットワーク処理 (特にプロトコル処理) と暗号処理の分野では、ビット演算 (および、ビットフィールド演算) の占める比率が高い。これらの演算は、汎用プロセッサで実行すると多くの命令を必要とするのに対して、その処理に専用のビット演算命令は容易に実装できるという特性がある。

これらの専用プロセッサ技術は、汎用プロセッサと専用ハードウェアの間を埋める技術であると捉えることもできる。従来は、ある機能を実現する際に、汎用プロセッサ上でソフトウェアで実現するか、専用ハードウェアで実現するかという両極端の選択肢しかなかったものが、用途専用プロセッサや構成可能プロセッサにより、その中間的な選択肢が出てきたということができる。このように選択肢が増えることは、システム設計サイドからは工夫の余地が増える反面、最適選択が困難になることも意味する。そのため、どのような性質を持ったシステムにどのようなプロセッサが向いているのかの目安が求められる。

最近の傾向としては、1つのチップ上に複数のプロセッサを搭載したチップマルチプロセッサが使用されるケースが増えていることが挙げられる。この理由として、アプリケーションに処理の並列度があるのであれば、高速なプロセッサ1つよりも、低速なプロセッサ複数でシステムを実現した方が消費エネルギーが抑えられ、コスト的にも安くすむことが大きい。また、処理の特性に合致した複数種類のプロセッサを1つのチップ上に集積したヘテロジニアスマルチプロセッサも、携帯電話機やPDAなどに広く使われている。

今後の注目される技術として、プロセッサの命令セットや構成をシステム出荷後に変更することができる動的再構成可能プロセッサを挙げることができる。動的再構成可能プロセッサについては、研究・開発が進められており、製品に適用したという報告もある。しかし、有効となるアプリケーションドメインが明確でないことに加えて、ソフトウェア開発環境の問題など、本格的な実用化までに課題も残っている。

このようにプロセッサの分野では、技術はますます多様化する方向に向かっており、標準化の動きは見られない。これは、高級言語のコンパイラが用意されていれば、ソフトウェアを異なるプロセッサ間で移植することは可能であり、性能を犠牲にしてまでプロセッサを標準化する意義が低いためと考えられる。

■ システムLSIとFPGA

上述の携帯電話機やデジタルテレビ、デジタルカメラなどの組込みシステムでは、そのアプリケーション (または、その一部の機能) 専用に開発された大規模・複雑なシステムLSIが使用されることが多くなっている。システムLSIには、1つまたは複数のプロセッサに加えて、そのアプリケーション用の演算回路や周辺デバイス、メモリなどを1チップ上に集積している。これにより、基板上に実装すべき部品の点数を減らせ、コストダウンや高信頼化、物理的なサイズの縮小が図れるばかりでなく、外部バスを駆動しないことによる低消費エネルギー化やノイズ対策の容易化など、メリットが大きい。

一方、システムLSIの開発には大きなコストがかかるため、大量に使用される見込みがない用途向けには、システムLSIを開発するのは難しい。このような分野では、近年、大規模化・高性能化が著しいFPGA (Field Programmable Gate Array) を用いることが多くなっている。FPGAとは、内部のメモリ等を書き換えることで回路構成を変更できるLSIであり、単一品種のLSIでアプリケーションごとの専用回路を構成することができる。

これまでのFPGAの使われ方は、プロセッサと一緒に基板上に載せ、周辺回路をFPGA上で実現するというかたちが多かったが、FPGAの大規模化により、FPGA上にプロセッサを実現することも容易になってきた。特に、FPGA上で実現しやすいように最適化して設計されたプロセッサの場合、FPGAのリソース使用が少なく、複数のプロセッサを1つのFPGAに載せることも容易である。さらに、FPGAの中にプロセッサの回路ブロックを組み込んだLSIも開発されており、これらを用いると、アナログ回路を除くコンピュータシステム全体を1つのFPGAに載せることも現実的になってきた。

■ プログラミング言語と言語処理系

組込みシステム開発に使われるプログラミング言語としては、依然としてC言語が主流である。

C++やJavaは、規模の大きい組込みシステムには適用されているものの、主流と言える状況ではない。これにはいくつかの原因があると思われるが、最大の原因は、言語や言語処理系を十分に理解して使わないと、予期しないオーバーヘッドが生じることと考えられる。たとえばC++では、使用しないことを明示的に記述しない限り、例外処理のためのコードが無視できないサイズとなる。また、テンプレートを迂闊に用いると、コードサイズが急激に大きくなる場合がある。このような問題を解決するために、C++を組込みシステムに用いるためのサブセットであるEmbedded C++(EC++)が、日本のメーカーを中心に提案され、ISOによる標準化検討が進んでいる。

Javaについては、携帯電話機のように、マシン独立なバイトコードをダウンロードして使用する機器では活用されているが、それ以外では、ガベージコレクションによってリアルタイム性が阻害される問題と、仮想マシンのオーバーヘッドが無視できず、それほど活用されていないのが現状である。

一方で、まだまだアセンブリ言語が使われる場面も多い。これは、割り込み処理など、ハードウェアに密接に関連する部分でアセンブリ言語を使わざるを得ないという理由もあるが、用途専用の特殊なプロセッサなどで、十分な性能を持ったコンパイラが使用できないケースもある。

■ リアルタイムOS

組込みシステムにはリアルタイムOSが使われることが多い。国内でも最も広く使われているのが、ITRON仕様に準拠したリアルタイムOSであるが、最近では、組込みLinux(Linuxを組込みシステム向けに修正したもの)が適用されるケースも増えている。言うまでもなく、Linuxは元々は汎用システム向けに開発されたOSであり、前述したような汎用システムに近い性質を持った組込みシステムに適用するのは自然な流れである。

標準化の観点では、組込みシステムの要素技術の中で、リアルタイムOSは最も標準化が進んでいる分野であるということが出来る。ITRON仕様は、組込みシステム向けの標準リアルタイムOS仕様として広く使われているし、自動車制御システム向けにはOSEK/VDX OS仕様が標準化されている。また、組込みLinux技術に関する標準化も進められている。

ITRON仕様やOSEK/VDX OS仕様のようなリアルタイムOS(これを、リアルタイムカーネルと呼ぶ)は、組

込みLinuxのような汎用OSを組込みシステム向けに修正したOSとは、構造・機能・性能の面で大きな違いがある。

ハードウェア資源の管理はOSの最も重要な役割の1つであるが、リアルタイムカーネルは、プロセッサ、メモリ、タイマといった、どのようなコンピュータシステムにも備わっているハードウェア資源のみを管理する。逆に言うと、ファイルシステムやプロトコルスタックなど、各種のI/Oデバイスを扱うための機能は持っていない。これらの機能が必要な場合には、リアルタイムカーネルの上に、ミドルウェアというかたちで実現する。これは、すべての組込みシステムが共通に持っているI/Oデバイスがないことに由来している。たとえば、ストレージデバイスを持たない組込みシステムは数多くあるが、そのようなシステムでは、OSがファイルシステムを持っていても意味がない。

リアルタイムカーネルのもう1つの特徴として、保護機能を持っていないことが多いことが挙げられる。OSの保護機能とは、あるアプリケーションに障害がある場合でも、他のアプリケーションやOS自身にそれを波及させないための機能である。組込みソフトウェアは、機器に固定されて提供され、機器の制御のみを目的に開発されるのが通常であるため、ソフトウェアの検証が完了すれば、保護機能は不要ということが言える。

しかし近年、ソフトウェアの大規模化・複合化により、ソフトウェアを完全に検証することが困難になっている。また、障害原因を切り分ける意味でも、保護機能が望まれるケースが多くなっている。それを受けて、ITRON仕様やOSEK/VDX OS仕様に対して、保護機能を追加するための拡張仕様が標準化・実装されている。

ここでの課題の1つは、厳格なリアルタイム性を維持したまま、メモリ保護機能を実現するのは容易でないことである。厳格なリアルタイム性を求められるシステムでは、最悪実行時間の予測が難しくなるという理由から、キャッシュすら使われていないものが多い。現在のプロセッサのメモリ管理ユニット(MMU)は、アドレス変換結果をキャッシュすることで動作を高速化しているが、この機構ではやはり最悪実行時間の予測が難しくなる。

そのため、ITRON仕様とOSEK/VDX OS仕様のいずれの保護機能拡張においても、アドレス変換が不要な仕様となっている。すなわち、多重アドレス空間を設けずに、論理アドレスと物理アドレスを一致させ、アクセス保護のみを実現するアプローチをとっている。このアプローチであれば、プロセッサの持つメモリ管理ユニットを適切に設計することで、厳格なリアルタイム性を維持したままメモリ保護を実現することができる。

メモリ保護以外に重要となる保護機能として、プロ

セッサ時間を保護するための機能を挙げるができる。リアルタイムシステムにおいては、あるアプリケーションがプロセッサを使い過ぎたために、他のアプリケーションが時間制約を満たせなくなる事態を防ぐことが望まれる。そこで、アプリケーションが使える時間を制限し、それを超えてアプリケーションが実行するのを防ぐ機能があると有用である。これが、プロセッサ時間の保護機能である。

これは、逆に見ると、あるアプリケーションが使えるプロセッサ時間を予約しているとも見える。このことから、プロセッサのリザーベーション機能と呼ばれる場合もある。これを実現したOSの例として、時間制約が厳格でないソフトリアルタイムシステム向けに、Linuxにリザーベーション機能を持たせたOSがある。また、このような機能を標準化しようという動きもある。

■ ネットワーク

近年、組込みシステムもネットワークに接続される場合が増えている。組込みシステムのネットワーク接続は、大きく次の2つの目的に分けて考えることができる。

1つめは、制御対象の機械・機器の状態をネットワークを通してパソコンなどから監視したり、逆に状態を変更することを目的に、組込みシステムをネットワーク接続するケースである。この場合には、パソコンなどと互換性のあるネットワーク技術を使う必要があり、EthernetやTCP/IPプロトコルが用いられることが多い。

2つめの目的は、機械・機器を制御する機能自身を、分散システムとして実現するものである。これが最も進んでいるのが、自動車の中の組込みシステムである。現在の自動車には、多いものでは1台で50～70個ものコンピュータが使われており、それらがネットワークを介して協調動作している。

この場合においては、ネットワークが機械・機器の中で閉じているため、アプリケーションごとの要求に最適化したネットワーク技術を使うことができる。特に、機械・機器を制御するシステムの場合には、厳格なリアルタイム性が求められるのが通常で、Ethernetのような最悪ケースの保証ができないようなネットワーク技術は、そのままでは適用できない。

自動車内のネットワークを例にとると、サブシステムごとの要求にあわせて、1台の自動車に3～5種類のネットワーク技術が使用される場合がある。サブシステムごとの要求の例として、たとえば自動車の走行制御のためのネットワークでは、個々のデータが小さいために転送レートは低くてもよい一方で、速い応答性(数ミリ秒から十数ミリ秒程度)が求められる。また、厳格に時間制約を満たすことが求められ、高い信頼性が求められる。

自動車内のネットワークの分野においても、国際的な標準化が進行している。従来は、自動車メーカーごとに独自のネットワーク規格を持っているケースが多かったが、ネットワーク技術の高度化により、メーカーごとの規格では開発コストが大きくなり過ぎることが、その大きな理由である。

現在、自動車内ネットワークの国際標準規格として広く使用されているのが、CAN(Controller Area Network)である。CANは、複数のノードを1つのバスに接続する共有メディア型のネットワークで、非破壊ビットワイズアービトラージと呼ばれる方式で、優先度の高いメッセージから順にバスに送出される仕組みとなっている。最大のビットレートは1Mbpsで、その場合にはバスの最大伝送距離が50mとなる。制御用のデータを送ることを想定しているため、1つのメッセージのボディのサイズは1～8バイトとなっている。またCANは、自動車内だけでなく、FA機器などの産業用デバイスのネットワークにも使用されている。

さらに最近では、より速い応答性と高い信頼性を達成できるネットワークとして、FlexRayと呼ばれる規格の標準化が進んでいる。FlexRayは、CANと同様共有メディア型のネットワークであるが、各ノードがデータを送信できる時間帯をあらかじめ(設計時に)決めるタイムトリガ型の通信をサポートしている。そのために、ネットワークに接続された各ノードのクロックを同期させる機能を持っている。また、2重化バス構成やバスガーディアンを設けることで、耐故障性を実現している。

組込みシステムのネットワーク接続が進んでいることに伴って、組込みシステムの分野でもセキュリティの確保が重要になってきている。実際、インターネットに接続して使用するDVDレコーダで、セキュリティ上の問題が見つかったという事例が出てきている。組込みシステムのセキュリティ技術は、今後の重要な研究テーマである。

■ コデザインとシステムレベル設計

組込みシステムにおいては、ソフトウェアのみならずハードウェアも専用に開発することが多いため、ソフトウェア開発とハードウェア設計を協調して行うことで最適化を図るハードウェア/ソフトウェアコデザインの技術が重要である。

コデザインには、大きく分けて2つのアプローチがある。1つは、プロセッサの項で述べた、動作させるソフトウェアに最適化した専用のプロセッサを用いるというアプローチである。もう1つは、システムの機能の中で、どの部分をソフトウェア(プロセッサ)で実現し、どの部分をハードウェア(専用回路)で実現するかを最適化

するというアプローチである。

第2のアプローチに向けては、システム設計の段階においてはソフトウェアとハードウェアを分離せずに設計を進め、なるべく遅い段階で両者の切り分けを決める手法が有力である。このような手法を、ソフトウェアとハードウェアを区別しない「システム」というレベルで設計を進めることから、システムレベル設計と呼ぶ⁴⁾。

システムレベル設計を行うためにまず問題となるのは、ソフトウェアとハードウェアを区別しない記述を行うためのシステムレベル言語である。システムレベル言語を開発するアプローチとしては、従来のハードウェア記述言語 (HDL) を拡張する方法と、CやC++といったプログラミング言語をベースにする方法がある。標準化されているシステムレベル言語としては、前者の例としてSystemVerilog、後者の例としてSystemCとSpecCを挙げることができる。また、複数の言語を組み合わせるシステムレベルの記述を行おうとするアプローチもある。

組込みシステム分野へのお誘い

組込みシステム技術は、自動車、情報家電、工作機械など、我が国が強い国際競争力を持っている産業を支えるきわめて重要な技術となっている。にもかかわらず、特に大学において、組込みシステム技術を専門にしている研究者はまだ少数にとどまっている。一方海外においては、ACMが組込みシステムに関する論文誌 (Transactions in Embedded Computing Systems) を発刊したり、SIG (Special Interest Group on Embedded Systems) を設立するなど、組込みシステムに関する研究が急激に活性化している。

本稿で述べた通り、組込みシステムの要素技術はきわめて多様であり、多様な技術が必要とされている。そのため大学にとっては、開発した技術が実用化される可能性が高いという意味で、魅力のある分野ではないかと考えている。また、我が国発の技術が国際的にもイニシアティブをとれる可能性の高い分野でもある。実際、ITRON仕様やEC++などのように、我が国発の標準化活動で成功を取っているものも出てきているが、より多くの標準技術を我が国から発信していくべきである。そのためにも、1人でも多くの大学の研究者に、組込みシステム分野の研究に取り組んでいただきたいと考えている。

一方で、大学においてこの分野に取り組むには、産業界における組込みシステム開発の現状や課題などが、ほとんどオープンにされていないという問題がある。産業界で組込みシステム開発に取り組んでいる方には、産学連携を促進する意味でも、このような情報を発信することをお願いしたい。

この4月から、情報処理学会の中に組込みシステム研究グループを新設し、活動を開始する。組込みシステム技術に興味をお持ちの研究者・技術者には、ぜひこの研究グループの活動に参加いただくと幸いです。

参考文献

- 1) 高田広章: 組込みシステム開発技術の現状と展望, 情報処理学会論文誌, Vol.42, No.4, pp.930-938 (Apr. 2001).
- 2) 高田広章: 組込みソフトウェア開発の現状と課題, bit, 2000年11月号, pp.3-16 (2000).
- 3) 特集 組み込みソフトウェア開発技術, 情報処理, Vol.45, No.7, pp.675-715 (July 2004).
- 4) 特集 システムレベルデザイン, 情報処理, Vol.45, No.5, pp.449-499 (May 2004).

(平成17年3月14日受付)

