

CISCからRISCへのOS移行のためのハードウェア技術

2G-7

内堀 郁夫

(株)東芝 情報処理・機器技術研究所

1.はじめに

高度産業用コンピュータVL2000では、RISCをベースとしたVLIWアーキテクチャを採用した。しかし従来機種であるTOSBAC G8000シリーズは、かなり複雑なCISCアーキテクチャを持っているため、VL2000の開発に当たっては、G8000からのソフトウェア移行性を如何に保証するかが、技術的なポイントの1つであった。

2.OSの移植

高級言語で記述されたアプリケーションプログラムでは、アーキテクチャの変更が直接影響することは少ない。むしろ問題となるのは、システムコール呼び出しのインタフェースなど、OSの構造に依存する部分である。もともとリアルタイム制御の分野では、OSの構造に深く依存する部分が多いので、従来CISCマシン上で動作しているOSをRISCマシンに移植することが必須となる。そのために、OSの記述に用いられている、システム記述言語の互換コンパイラを開発したが、その実現にあたってのハードウェアとしてのポイントは、大きくは次の2点である。

(1)ハードウェアの構成

制御レジスタ類の構成などを、CISCアーキテクチャを意識したものとする。

(2)ファームウェアモードの導入

超特権を持つ動作モードを導入し、マイクロコード機能の一部をカバーする。

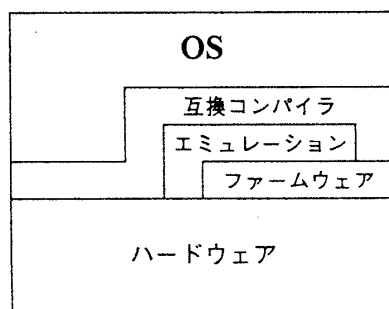


図1 OS移植の構造

3.ハードウェアの構成

RISCベースのVLIW方式といっても、これは主として演算器の形態である。実際のCPUアーキテクチャは、制御レジスタ類の構成や、割り込みの方式などで肉付けされて成り立っている。OS内部では、こうした制御系のハードウェア構成への依存する部分が多く、簡単には移行できないのが普通である。OSの基本構造を維持するために、次の点について考慮した。

(1)割り込みマスク等、直接ハードウェア動作を規定するレジスタ類の構成を、従来機から継承した。

(2)割り込みの基本動作・優先度、あるいはアドレス変換方式を従来機から継承した。

4.ファームウェアモードの導入

ファームウェアといえば、マイクロコードと同義に用いられる場合が多いが、ここでいう「ファームウェア」は、マイクロコードではなく、いわば超特権のエミュレーションプログラムのことである。多くのCISC型マシンはマイクロコード制御であるから、ハードウェアのRISC化に伴って、マイクロコードで行っている処理の大部分を通常のプロセスに展開する必要がある。しかし一部には単純に展開できない部分もある。これは、OSあるいは、一部アプリケーションプログラムにも、マイクロコードの特性を期待している部分があるためである。その特性とは、大きくは次の2つである。

(1)不可分性

一連の処理を不可分の機能単位として、通常のプロセスに提供している。つまり非同期の割り込みがあっても、処理の中間状態がみえることはない。

(2)不可視性

ハードウェア動作上のイベントを、通常のプロセスからはみえないレベルで処理することがある。

CISCマシン上で構築されたOSは、例えば割り込みの基本動作やアドレス変換方式でも、不可分・不可視を期待しているが、こういった処理を完全にハードウェア化するのは困難である。こうした部分をカバーするのが、ファームウェアモード、つまりマイクロコードの特性をプログラムレベルで実現するしくみである。ファームウェア自体は、主記憶装置上にあり、通常の命令セットで記述されている点では、通常のプロセスと何等変わるところはないが、通常のプロセスには、ハードウェアの一部として振る舞うことができる。通常のプロセスとは、次のような相違点がある。

(1)物理アドレス空間上の特定領域に常駐しており、この

Operating System Migration from CISC to RISC

Ikuro UCHIHORI

TOSHIBA Corporation

領域を保護するための専用ハードウェアがある。他のプログラムからは、専用のコール命令、もしくは割り込みで呼び出される。

- (2)ファームウェアモード中は、原則としてすべての割り込みが禁止される。
- (3)ファームウェアモードのプログラムは、すべての保護を無視できる。
- (4)通常のプログラムからは操作できないハードウェア資源を操作することができる。
- (5)専用の作業用レジスタが用意されている。

以上で明かなように、専用ハードウェアも多く付加されている。これらによって、ファームウェアモードとして必要な特性を保証しているのである。ファームウェアによって、次のような機能が実現されている。

(1)割り込み発生処理

ハードウェアが発生する割り込みを受けて、OSの割り込みハンドラに受け渡す処理である。プログラムとプログラムの狭間にあつて、不可分・不可視であるため、ファームウェア処理とした(図2)。

(2)アドレス変換処理

従来マイクロコードで閉じていた処理。通常、プログ

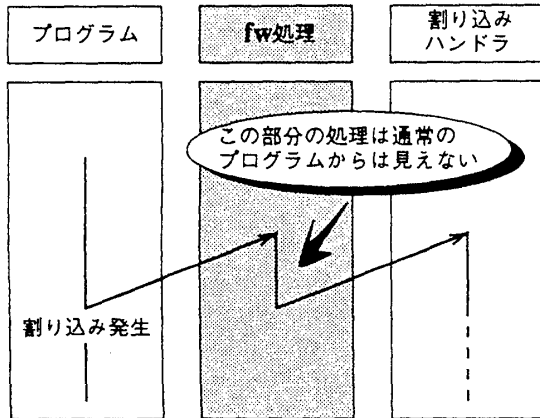


図2 割り込み発生時のfw動作

ラムに同期して起動するが、OSはその動作を一切意識していないため、ファームウェア処理とした(図3)。

(3)入出力処理

入出力機器を制御する処理。従来機種用の入出力チャンネルを使用可能とするために、チャンネルからみたCPUアーキテクチャの相違点を吸収する必要からファームウェア処理とした。

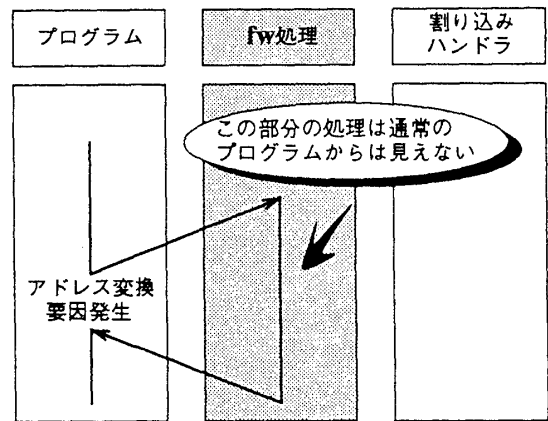


図3 アドレス変換処理時のfw動作

(4)RAS関連処理

CISCマシンが有していた、多くのRAS関連機能を継承する。各種ハードウェア異常発生時の処理、あるいはマルチプロセッサ時のプロセッサダウン情報の受け渡しを行う。

5.おわりに

CISCからRISCへのアーキテクチャ変更にあたり、ソフトウェア移行性を保証すべく、以上述べてきたような施策をとった。この結果として、CISCマシン上で動作していたOSの大部分を、再コンパイルするだけで、移植することができた。一部にはアセンブラレベルの記述も含むOSに関してもこの成果を得たことで、今後移植されるであろう、各種アプリケーションプログラムに対しても「ソース互換」をほぼ完全な形で保証できることを、証明できたと考える。

参考文献

1)森 良哉

「ミニコンピュータ・ワークステーションの命令セットアーキテクチャ」

情報処理,29,12,pp.1412-1419(1988)

2)山田晃智,他

CISCからRISCへのオブジェクトトランスレーション
情報処理学会第45回全国大会(1992)